Application of service composition mechanisms to Future Networks architectures and Smart Grids

Ramon Martín de Pozuelo Genís

UNIVERSITAT
RAMON LLULL

DOCTORAL THESIS

Title	**Application of service composition mechanisms to Future Networks architectures and Smart Grids**
Presented by	**Ramon Martín de Pozuelo Genís**
Centre	**Escola Tècnica Superior d'Enginyeria Electrònica i Informàtica La Salle**
Department	**Departament d'Informàtica**
Directed by	**Dr. Agustín Zaballos Diego y Dr. Francesc Pinyol Margalef**

Application of service composition mechanisms to Future Networks architectures and Smart Grids

by
Ramon Martín de Pozuelo

Grup de Recerca en Internet Technologies & Storage
Departament d'Informàtica
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull

Thesis director:
Dr. Agustín Zaballos Diego

Thesis co-director:
Dr. Francesc Pinyol Margalef

laSalle
RAMON LLULL UNIVERSITY

Acknowledgements / Agradecimientos /Agraïments

Diría sin pestañear que la realización de esta tesis ha supuesto un antes y un después en mi vida, no solo profesional, sino personal. Para bien y para mal ha condicionado mi vida en estos últimos años y seguramente mi futuro, pero considero que lo mejor de todo ha sido que me ha llevado a conocer y entablar relación con multitud de personas maravillosas, algunos de los cuales se han convertido en grandes amigos.

Primero de todo agradecer el soporte de La Salle – Universidad Ramon Llull para realizarla. Más de media vida he pasado en la institución de La Salle entre estudiante de primaria, secundaria, universidad, y después como trabajador, y es algo que deja su huella y siempre llevaré conmigo.

Después querría agradecérselo principalmente a Agustín Zaballos, el que ha sido mi director de tesis estos últimos años. Parte de esta tesis es suya. Sin él, creo que en algún momento hubiera tirado la toalla. Pero más allá de eso, le querría agradecer todo este tiempo que hemos pasado juntos. Tenemos formas de ser y de trabajar diferentes, pero le considero mi mentor a nivel profesional y científico y en estos años me ha hecho crecer en muchos sentidos. Gracias por la confianza depositada, siempre he intentado estar a la altura de ella.

Voldria fer menció també al meu co-director de tesi, Francesc Pinyol, i al Gabriel Fernández, que em van engrescar a començar aquesta aventura del doctorat i em van acompanyar els primers anys. El Gabriel ja m'havia convençut anys enrere per entrar al departament de Televisió Digital i endinsar-me en el món de la recerca. Moltes gràcies a ells i els altres membres del departament que em van acompanyar aquells primers anys com a becari a La Salle.

Agrair també a la Fundació i2cat i tots els participants del projecte TARIFA, que va marcar l'inici d'aquest doctorat. Entre ells he de fer mencions especials al Sebastià Sallent per l'aposta per aquest projecte innovador, a l'Albert Vidal per la moltes vegades complicada feina de coordinació, al Josep Paradells, per la seva clarividència en el disseny de solucions i la pressa de decisions i al Jesus Alcober per l'oportunitat que em va obrir de col·laborar als comitès d'estandardització ISO/IEC i AENOR.

Y por encima de todos ellos, de todas formas, he de agradecérselo a mis compañeros de batallas en el proyecto, Alberto José, Martín y Yury. Con ellos he compartido grandísimos momentos profesionales y personales y se convirtieron ya en familia. ¡Grandes Tariferos!

I have to thank also Dr. Yi Hwa (ETRI, Korea), Dr. Bernd Reuther (University of Kaiserslautern, Germany), Dr. Panos Georgatsos (CERTH, Greece) for sharing literature, appreciated assistance and collaboration during the project of TARIFA and subsequent work.

Y si TARIFA fue el inicio, he de agradecérselo también a otras personas claves en el desarrollo de esta tesis en los últimos años. En especial los que son o han sido mis compañeros del departamento, Alan, Julia, Josep Mª, Joan, David y Víctor, así como todos los becarios que han participado con nosotros en el desarrollo de los proyectos. Siempre han facilitado el trabajo, y sin ellos no hubiéramos podido realizar otros proyectos como FINESCE, que han resultado fundamentales para la definición final de la tesis.

Però seguint amb altres companys que són part d'aquesta tesi no em podria oblidar de l'Àngels, amb qui he compartit gran part d'aquest període professional i de la vida en general, i ha estat un gran suport moral durant tots aquests anys. Ella ha estat la veu de la consciència en molts casos i em va reconduir en moments en que no tenia molt clar quin era el camí adequat. Moltes gràcies a ella, i també a tots els altres membres d'aquesta família de La Salle amb qui he compartit grans vivències durant aquesta etapa, el Raúl, Lluís, Marc Freixes, Marc Arnela, Xuti, Xavi Sevillano, Xavi Valero, Marcos, Àlex, Ale (i més que em deixaré).

Per anar acabant, li agraeixo també a la Marta, per sobreviure la que ha estat la part més intensa de la tesi, per recolzar-me en tot moment, però també per donar-me canya i aportar-me les forces necessàries per acabar-la i començar una nova etapa junts. Aquest és un dia encara més feliç perquè el puc compartir amb tu.

Y finalmente quería dar las gracias a todos mis amigos y familia por su comprensión y apoyo durante este largo periodo. En especial a mis hermanos Juan Luis y Esther y a mis padres, Luis Joaquín y Mariquel, por su respaldo incondicional, por permitirme realizar los estudios que me han llevado hoy hasta aquí, así como transmitirme desde pequeño los valores del esfuerzo, del trabajo bien hecho y motivarme siempre para dar lo mejor de mí.

"Science fiction does not remain fiction for long. And certainly not on the Internet"

– Vinton Cerf

Prologue

Although the difficulties that present to build over a clean-slate network architecture, I strongly believe in the need of the Internet science and research on novel network architectures. I was inspired by proposals made at the beginning of this century by the same people who provided the groundings of the current Internet. Those ones that were there defining the principles, operation and protocols since its initiation, who changed the world, who changed every aspect of my life and the whole society, people like David D. Clark, Robert Braden, Scott Schenker, Van Jacobson, John Day, or Vint Cerf who have my complete respect and admiration, are the same that now advocate for continue researching on novel ways of doing things in Internet. Internet is working and its functioning is critical for the current-day economy and security of the whole World, but even if it looks like repairing the damage of an Airbus A380 in the middle of the air, they defend the need of changes in the operation of current Internet in order to move forward in its development, and adapt to the requirements that we have today and future applications that were never thought 40 years ago and will be possible now. They look for architectures that provide easier management and service deployment mechanisms in the network. They proclaim that the end-to-end principle is not valid anymore, and the network is increasingly providing more and more services and there could be other easier ways to build communications through new technologies such as Software Defined Networks and novel clean-slate network architectures.

Hence, all abovementioned made sense to me and got me on building over the conceptualization of a service-oriented Future Networks architectures, even when I knew it will be a much more difficult challenge than optimize something already working. It is a risk that we knew we were taking. Those novel architectures revised and proposed are still in a research phase, they are part of the Internet science work, and still will be many years until we will see them running at large scale. However, at this point, I have the feeling that the work done was completely worth it and this research helped me think different and grow as a researcher and as a person in general. It makes me consider that alternatives are always possible, even in those cases when something is massively established and people could say you are foolish for thinking different.

Abstract

This thesis revolves around the hypothesis the service composition methodology and mechanisms and how they can be applied to different fields of application in order to efficiently orchestrate flexible and context-aware communications and processes. More concretely, it focuses on two fields of application that are the context-aware media distribution and smart grid services and infrastructure management, towards a definition of a Software-Defined Utility (SDU), which proposes a new way of managing the Smart Grid following a software-based approach that enable a much more flexible operation of the power infrastructure. Hence, it reviews the context, requirements and challenges of these fields, as well as the service composition approaches. It makes special emphasis on the combination of service composition with Future Network (FN) architectures, presenting a service-oriented FN proposal for creating context-aware on-demand communication services. Service composition methodology and mechanisms are also presented in order to operate over this architecture, and afterwards, proposed for their usage (in conjunction or not with the FN architecture) in the deployment of context-aware media distribution and Smart Grids. Finally, the research and development done in the field of Smart Grids is depicted, proposing several parts of the SDU infrastructure, with examples of service composition application for designing dynamic and flexible security for smart metering or the orchestration and management of services and data resources within the utility infrastructure.

Contents

Glossary

ACK	Acknowledgement
ACM	Association for Computing Machinery
ADA	Advanced Distribution Automation
AI	Artificial Intelligence
AM	Atomic Mechanism
AS	Atomic Service
BB	Building Block
BER	Bit Error Rate
BPEL	Business Process Execution Language
CBR	Condition Based Routing
CCN	Content-Centric Networking
CM	Content Management
CoAP	Constrained Application Protocol
CRC	Cyclic Redundancy Check
CREQ	Communication Request
CRESP	Communication Response
CRSV	Communication Reservation
CS	Composed Service
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DCF	Distributed Coordination Function
DER	Distributed Energy Resources
DHT	Distributed Hash Table
DMS	Distribution Management System
DNS	Domain Name Server
DSO	Distribution System Operator
e2e	End-2-End
EMS	Environmental Management System
ESN	End Service Node
FEC	Forward Error Correction
FI	Future Internet
FIA	Future Internet Assembly
FIDEV	Finesce Device
FIND	Future Internet Design
FINESCE	Future INtErnet Smart Utility ServiCEs
FIDEV	FINESCE Device

FN	Future Network
FP6	Framework Programme 6
FP7	Framework Programme 7
GENI	Global Environment for Network Innovations
GPS	Global Positioning System
HCDM	Hybrid Cloud Data Management
HTTP	HyperText Transfer Protocol
ICN	Information-Centric Networking
ICT	Information and Communication Technologies
ID	Identifier
IDEV	INTEGRIS Device
IEEE	Institute of Electrical and Electronics Engineers
IMS	IP Multimedia Subsystem
IN	Intermediate Node
INTEGRIS	INTelligent Electrical GRId Sensor communications
IP	Internet Protocol
IPTV	Television over IP
IoT	Internet of Things
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LISP	Locator/ID Separator Protocol
MAC	Medium Access Control
MDC	Multiple Description Coding
MOS	Mean Opinion Score
MPLS	Multiprotocol Label Switching
MQTT	Message Queue Telemetry Transport
MVC	Multiview Video Coding
NAT	Network Address Translator
NGN	Next Generation Network
NS-2	Network Simulator 2
NSF	National Science Foundation
OASIS	Organization for the Advancement of Structured Information Standards
P2P	Peer-to-Peer
PEAQ	Perceptual Evaluation of Audio Quality
PN	Provider Node
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service

RAM	Random Access Memory
RBA	Role-Based Architecture
RN	Requester Node
RTCP	Real Time Control Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SG	Smart Grid
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNR	Signal to Noise Ration
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service-Oriented Computing
SSIM	Structural Similarity
SVC	Scalable Video Coding
TARIFA	The Autonomic Redesign of the Internet Future Architecture
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery Language
UDP	User Datagram Protocol
WF	WorkFlow
WoE	Web of Energy
WoT	Web of Things
WSDL	Web Service Description Language
XML	eXtensible Markup Language

List of figures

List of tables

1 Introduction and motivation

In 2005, different high-profile Internet research leaders, "Making the World (of Communications) a Different Place" [Clark et al., 2005] advocated for a shared common vision of how the networking field might be materially different in 10 years. They sketched some objectives and challenges the networking community should address in order to enhance the possibilities of networking in the future and, hence, deepen our understanding of the field. More than ten years after, many articles, initiatives and projects have been done inspired by the thinking written by David D. Clark, Robert Braden and others, but changing how Internet-scale communications work is a really long run. It is difficult to change something that works and is the cornerstone of current-day society and economy, even if those researchers that help to create it point out some fundamental faults in their own invention.

However, I firmly believe that most of their proposed topics are relevant to the research community as they provide insights into the most urging challenges facing the communications field today and in the long term, and how networking could (and should) be different (and better) in the future. Thus, I consider this discussion can provide the basis of a framework or vision to agree on a common purpose for the evolution of the networking field (and a research agenda to get there).

This introductory chapter gives a general vision of the different domains in which this thesis presents contributions as well as summarizing its objectives. To understand the path followed in the elaboration of this thesis, the focus on the different research lines and relationships between them it is also important to know the story behind it, especially the research projects in which I have participated that guided the different phases of the thesis.

The first inspiration for this thesis comes from the need of facing some faults or inefficiencies in current Internet architecture that were pointed out by David D. Clark and other Internet gurus some years before. It motivated me to participate in a project called TARIFA ("The Atomic Redesign of the Internet Future Architecture") [TARIFA] that started in 2010 and was funded by i2cat Foundation[1]. The goal of the project was to define an architecture for Internet of the Future, together with members of i2cat and research groups from other universities (Universitat Politècnica de Catalunya – UPC –, Universitat Pompeu Fabra – UPF –). It wanted to put in common the perspectives of the different researchers in the field of Future Networks[2] (FN) around Barcelona and provide a consolidated solution.

[1] i2cat Foundation (http://www.i2cat.net/en) is a non-profit research and innovation centre which promotes mission-oriented R+D+i activities on advanced Internet architectures, applications and services.

[2] Considering Future Networks architectures as evolutionary and especially "clean-slate" approaches to network architecture resilient, trustworthy and energy-efficient and designed to support open access, increasing heterogeneity of end-points (multimode devices, people, things) and networks (ad-hoc networks, opportunistic networks, networks of networks), with seamless and generalised handover, in support of the complete range of services and applications.

At that time, when I joined this group of researchers I had finished my MSc in Computer Networks and Telecommunications and I was collaborating with the Media Technologies Group (GTM) of La Salle – Ramon Llull University since 2006 in different media related projects. It is also worth to mention two of those projects that enable my early conference publications [Enrich et al., 2008] [Domingo et al., 2008] [Martin de Pozuelo et al., 2010] and they also influenced in a way or another the development of the subsequent thesis.

The first of them was SUIT [SUIT] ("Scalable, Ultra-fast and Interoperable Interactive Television") was a FP6 project funded by European Commission that aimed to provide flexible and more efficient interactive television, integrating broadcast and Internet communication networks and using scalable video formats. At the end, the project focused on the limitations of broadcast network for offering interactive TV, and the need of Internet network for providing enhanced media services. It helped me in my early research stage to understand the media distribution domain, their requirements, limitations and challenges in their gradual migration to IP and Internet.

The second one was i3Media [i3MEDIA], a CENIT[3] project that seeks solutions for the creation of technology to deal with intelligent contents, focused on the automation of heavy industry processes involved in the creation of audiovisual content. Concretely, I worked in the multimedia content adaptation process and developed an automatic Interactive TV service generator from DVB-PCF[4] [Martín de Pozuelo et al., 2010] [Peng et al., 2010] templates. This work, although focused in TV services and based on templates, made me realize of the potential of automatic interactive TV service composition.

With this wealth of knowledge and experiences I started my PhD, seeking to contribute in TARIFA project. Because of my background, my main initial focus was on how a new architecture for Internet should be defined in order to consider by default the efficient and flexible delivery of media services. However, besides this, in the course of the project I started to concern on other topics (e.g. cybersecurity, IoT) and domains (e.g. Smart Grids) that were later covered during my PhD.

In short, the authors of "Making the World (of Communications) a Different Place" guided the initial research of this thesis above all, claiming in their article for a network that should be ubiquitous, secure, robust, self-managing, pervasive and adaptable to changes in context. Hence, I specially embrace the following of their arguments as some of the grounds for my research:

[3] CENIT program was launched in 2006 by the Spanish Ministry of Economy and Competitiveness to stimulate public - private cooperation in industrial research through funding, through grants of up to 50% of consortium projects large-scale and long-range scientific and technical oriented planned research in technological areas future and potential international projection, whose purpose is the generation of new knowledge that can be useful for creating new products, processes or services or for the integration of technologies of strategic interest.

[4] DVB-PCF: Digital Video Broadcasting – Portable Content Format

1. "In 10 years, there should be a ubiquitous, low cost, open infrastructure suited for communication with low-cost computing devices such as sensors and controllers."
   It has been widely projected that by 2020 the number of connected smart objects will reach 10 billion and represent the 46 percent of the total devices and connections [CISCO, 2016]. The recent growth of the Internet has fostered the interaction of many heterogeneous technologies under a common environment such as the Internet of Things (IoT). However, the way to access IoT devices from a single platform is undoubtedly one of the biggest headaches for researchers. Nowadays, most IoT solutions are vertical, tending to address specific application domains at given spatial densities. Information is addressed by specific means and, in the majority of the cases, becomes available only to registered users.

2. "In 10 years, any physical object should be able to tag itself in a way that links it to relevant information and functions in cyberspace."

   In a unified IoT ecosystem, data and services should be semantically interoperable and made available across IoT domains through open, consistent and user-friendly interfaces, thus achieving significant efficiencies in service development and data consumption. There is still lot of work to do in this field in order to provide an IoT ecosystem able to embrace all existing and emerging stand-alone solutions in a "Web of Things" (WoT) [Guinard et al. 2011] and provide open and extensible APIs for service production and consumption.

3. "In 10 years, the Internet should be augmented with a new set of mechanisms for diagnosis and configuration, which can improve the usability of the Internet, reduce the need for manual intervention, and provide a linkage between application intentions and network behaviour."

   The network grows exponentially, and the complexity in its management is accordingly increasing. It is important that Internet continues its evolution (or revolution) including mechanisms for automatically adapting its communications and controlling the performance of the services offered. It means to provide methods for calculating, detecting, establishing and adapting optimal network services according to context information, and monitoring the service performance and the network beneath it. It applies to end services offered to users, but also to network management services. In current-day, it is usual that user and service mobility change the communication conditions during service provisioning, and it may cause a service deterioration or corruption. One of the well-known problems of current Internet architecture is the lack Quality of Service (QoS) guarantee at network layer [Clark et al., 2007]. Those mechanisms to maintain QoS at Internet-scale without complex and vendor dependent configurations are not yet available, allowing, for example, seamless handover between heterogeneous access network technologies will be provided, supporting transparent service continuity. More than 12 years after the authors' statement, a generic and adaptive functionality for the control and management of network and service resources is still needed.

4. “In 10 years, our communications infrastructure should be based on an architecture that provides a coherent framework for cybersecurity, robust operation in the face of attack, and a trustworthy environment for services and applications.“

   As networks become an integral part of corporations and everyone’s lives, advanced network security technologies are being developed to protect data and preserve privacy. The security of communication and data networks was not considered a priority some years ago, perhaps because nobody was able to predict such an impressive growth of the use of data networks and their significant importance worldwide. The design of protocols, devices and networks was more focused on their operational function rather than providing systems that fulfilled security requirements. However, this trend has radically changed by now. The thread of cyberattacks is worrying many different critical infrastructure environments such as Smart Grids (at electricity generation, transmission, and distribution), finance (banking), public health (hospitals, ambulances), transportation systems, alarm and security services, and Smart Cities services in general. [Yusta et al., 2011]. The inclusion and dependence on Internet of all these critical services have become the cybersecurity a very prolific line of research with many efforts dedicated to it.

Considering that their article was almost 10 years old at the time I read it, and most of their statements were still in progress it clearly motivated me to contribute to the research in Internet architectures and all the gaps that were identified. Actually, it can be said that current Internet was originally designed to interconnect a small group of researchers, and it is anticipated that may not be capable of supporting the near future set of usages, services, constraints and requirements that it will have to face. It worked very well for what they needed at that time. As John Day [Day, 2007] usually says: “The biggest problem with the ARPANET was we got too much right to begin with”. But in current-day, the Internet for the future cannot be longer seen the same way, as a set of links, routers and protocols. It is expected to be seen as a network of applications, services [Moreno-Vozmediano et al., 2013] [Papazoglou et al., 2007] [Chandrashekar et al., 2003], information and contents (resources) [Ahlgren et al., 2012] [Trossen et al., 2012] [Ghodsi et al., 2011]. New distributed software systems have become more dynamic, allowing transparent distribution, self-reconfiguration, portability, etc. Based on that, new paradigms deviated from the end-to-end principle have emerged, such the IoT. In addition, the continuous evolution of applications and services are increasing current networks complexity, adding more and diverse requirements (e.g. mobility, security or multihoming) that are not efficiently covered by current TCP/IP protocol stack.

The current picture of the networks shows a large, heterogeneous, dynamic and complex distributed system. Lots of patches aimed to amend different issues that have arisen during last years. Current networks have to deal with new services, applications and computing paradigms such as new modes of interaction, identification, context-awareness, energy efficiency, seamless service discovery and composition, mobility, ubiquity, etc.

The development of a new network architecture has been discussed for some time now. Several proposals are considered in this sense, evolutionary (incremental) approaches and revolutionary (clean-slate).

Taking a clean-slate approach, FN architectures define revolutionary ways to create communication based, for example, on Software-Defined Networks (SDN), Information Centric Networks (ICN), Service-oriented Architecture (SOA) concepts, or a mix of all of these new proposals to change the behavior and operation of networks, and finally at whole Internet-scale.

The work of this thesis relies on the concept of a context-aware and service-oriented FN architecture, which will aim at providing services taking into account the changing conditions of the context and thus, offering customized communication and seamless delivery of data. It was first analyzed in a generic way, although always thinking in media distribution services (that was, as explained, my initial background), but was gradually shifting to a different context: the Smart Grids. New research interests come analyzing the need of integration of IoT networks and cybersecurity integration efforts during TARIFA project. However, an external factor was the trigger of the scope extension and research field movement. The research group in which I was involved during my first years of PhD, Media Technologies Research Group (GTM) made some structural changes and I moved to the Group of Internet Technologies and Storage (GRITS) of La Salle Engineering – Ramon Llull University as well. This group dedicates their research efforts to heterogeneous network interconnection, especially but not exclusively in Internet of Things (IoT) scenarios for Smart Grids and Smart Cities and the required multiservice (data, audio, video, etc.) data networks. More concretely, it focuses on designing communication protocol standards, data storage and management for smart heterogeneous networks, which imply tailored QoS-aware routing protocols and cybersecurity solutions for the deployment of smart electricity grids machine to machine (M2M) communications.

During the last years of my PhD, involved in GRITS, I participated in different projects. I contributed in the end of the FP7 INTEGRIS [INTEGRIS] project, and I had a leading role in the National funded projects SHECloud, MBTAP, MBTAP wireless optimization and FP7 FI-PPP[5] FINESCE [FINESCE]. Although all these projects and my role in them are further detailed in *Chapter 7, Section 7.2 Research Outputs*, a brief description of them is provided here:

- INTEGRIS was a European Commission's FP7 project proposed the development of a novel and flexible ICT infrastructure based on a hybrid Power Line Communication-wireless integrated communications system able to completely and efficiently fulfill the communications requirements foreseen for the Smart Electricity Networks of the future.

[5] FI-PPP: Future Internet Public Private Partnership program launched in 2011 by European Commission within the FP7 with 270M€ of funding. The main goal is to advance a shared vision for harmonized European technology platforms and their implementation, as well as the integration and harmonization of the relevant policy, legal, political and regulatory frameworks.

- SHECloud (Smart Hybrid Enterprise Cloud) project selected in the call for projects AEESD[6] (Strategic Action Digital Economy and Society) 2013 from the Ministerio de Industria, Energía y Turismo. We participated with the companies Abiquo[7], Claranet[8] and MediaCloud[9], and its goal was to create an intelligent orchestrator for the interoperability of public and private clouds, automatizing the migration of workloads (virtual machines and other resources) among clouds in a Hybrid Cloud environment.
- MBTAP (Media Bus Transfer Protocol) and MBTAP wireless optimization projects are two projects selected in the calls for projects 2014 and 2015 in which we collaborated with Video Stream Networks (VSN)[10] in the definition of a reliable and secure transport protocol, optimized for large data transfers; especially over Long Fat Networks (LFNs) [Mathis et al., 2006] [Katabi et al., 2002]. The second project focuses on adaptation and optimization of the protocol to work over heterogeneous networks considering wireless segments, and it is still ongoing.
- FINESCE project represents the Smart Energy use case of the FI-PPP second phase. FINESCE contributed to the development of an open IT-infrastructure to be used to develop and offer new app-based solutions in all fields of the Future Internet related to the energy sector. The consortium of 31 partners, coordinated by Ericsson, included leading energy and ICT operators, manufacturers and service providers as well as research organizations and SMEs from all over Europe. Continuing the developments undertaken in INTEGRIS project, we aimed to integrate the IEC61850 [IEC 61850] Virtual Distribution Substation concept, by orchestrating the communication elements of the IDEV (INTEGRIS Device) prototypes developed in INTEGRIS.

I did not forget the media distribution topic in projects such as MBTAP and SHECloud, but projects focus on Smart Grid communications such as INTEGRIS and FINESCE meant a significant change in the context that I was usually working with. They provided me a broader vision and open a new field that has become a significant part of this thesis and my current research.

Actually, this shift provided new elements to my initial hypothesis and objectives. At first, I wanted to focus on analyzing the viability of service composition methodology for creating real-time adapted network communications. Although that definition was quite generic, my

[6] http://www.minetur.gob.es/PortalAyudas/AEESD/Paginas/Index.aspx

[7] Abiquo (http://www.abiquo.com/): Founded in 2006, Abiquo is one of the original pioneers of cloud computing, focusing their efforts on hybrid cloud management solutions for enterprises.

[8] Claranet (http://www.claranet.com/): Founded in 1996, Claranet is one of Europe's leading managed service providers.

[9] MediaCloud (http://www.mdcloud.cat/): A cloud digital infrastructure supplier part of Mediapro group, a leading provider of technical services for the audiovisual industry, producer and distributor of audiovisual content, film manager and distributor of sporting events, and interactive content producer.

[10] Video Stream Networks (https://www.vsn-tv.com): VSN is a global technology company specialized in providing advanced solutions for the broadcast and media sector. It offers solutions based on standard IT infrastructure solving the needs of creation, distribution and management of audiovisual contents in TV channels, public institutions, IPTV, universities, contents distributors and news agencies.

initial idea was always to test it for providing efficient media distribution services. The evolution in my research career during these years make me realize of the potential of the composition methodology and mechanisms in other fields such as Smart Grids and Smart Cities, and tackling other issues like seamless IoT and heterogeneous network integration or flexible security services deployment.

Therefore, the main hypothesis became that:

"Service composition methodology and mechanisms can be applied to media distribution and Smart Grid fields in order to efficiently orchestrate flexible and context-aware communications and processes."

Why service composition? Service composition is the technology that supports the composition of those activities required to reuse and combine existing services to enrich current services and to create new services. It is the process of constructing a complex Composite Service (CS) from fundamental pieces or functionalities, also called Atomic Services (AS), to achieve a specific task. This technology provides a natural way of combining existing services including both atomic and composite services. Such kind of recursive composition of composite services is one of the most attractive and challengeable features of the service composition, allowing to rapidly and easily create new services. Thus, the service composition can provide benefits on an improved usability of existing services, faster time for service creation and reduced time to market for new services. *Chapter 3: Service Composition* will summarize the research already done in this field and how it influenced my work.

In my opinion, service composition will enable enhanced and adapted solutions to offer to this two fields that are highly dynamic and define specific and critical requirements for each communication. Both fields characteristics and requirements will be detailed later in *Chapter 2: Future Networks for context-aware media distribution and Smart Grids? Two different fields?*.

I consider that FN designs should incorporate service composition capabilities and define a specific framework that enable them. It will contribute to create scalable, modular, and service-aware services regardless they operate for delivering media or for interconnecting smart objects within a smart electricity distribution grid. In this framework, the necessary functionality for establishing communications in any node connected to the network (user devices and network elements) is not fixed but dynamically composed, as appropriate to user service requirements, network transfer capabilities and surrounding context in the user and the network environments. In essence, a service-oriented paradigm is followed. Communications are accomplished by assembling appropriate services each performing a specific communication function. As such, service functionalities can be combined to create higher level communication services, which in turn can be combined with other services as well to enrich existing services or to create new composed ones, until the whole spectrum of required functionality for end-user communications is in place.

Nowadays, users are consuming very diverse services and resources (information heterogeneity) through very different devices. That is, consume services through different access networks wired or wireless (ADSL, UMTS, WiFi, etc.) and through very diverse devices (high-end devices such as a PC or low-end devices such as mobile phones) under dynamic conditions. In this heterogeneous and dynamic environment, FNs are expected to provide efficient service provisioning in a personalized manner, taking into consideration the specific context of each element involved in the communication process (application, node, link). This will enable to create and provide future rich and innovative services.

In addition, we are not talking only about *human users* any more. The advent of low-cost, low-profile devices along with the advancement in communication technologies has enabled a plethora of connected IoT smart objects to gain intelligence and ability to interact with the surrounding environment and the humans. The billions of connected smart objects are highly heterogeneous, ranging from wired cameras, wireless sensors and actuators to RFID tags, smart home appliances and wearables. Smart objects are essentially building blocks for services. Accessing them, their data, events and actuation capabilities with seamless, efficient and cost-effective, yet secure and accountable, means is expected to profoundly transform our environment and lives. Unprecedented opportunities for value-added services will emerge, while businesses will thrive by opening new frontiers in a variety of application domains.

In this scenario, context-aware service composition can play an important role. Service composition has traditionally been viewed as a static process performed manually by the developer at design time. However, service composition shifted towards operating at run-time, orchestrated by a centralized or distributed composition manager component. As mobile environments become increasingly common, composition is performed on an ad-hoc basis, in a distributed, peer-to-peer way.

As stated in the hypothesis, the main objective of this research is to analyze the viability of service composition methodology for creating real-time adapted network communications. It will provide the background in order to develop an orchestration mechanism that creates a workflow, processes all information retrieved from service and additional context (user, node, and network) and creates a workflow of fundamental/atomic services. However, several aspects should be previously considered in its design:

- Field of knowledge: First of all, it was important to analyse the different fields of knowledge in which the composition approach is proposed. The specific context, characteristics and singularities of the field may change the perspective and vary the methodology and the mechanisms to be used. Therefore, a review of the requirements, goals and challenges in media distribution and Smart Grids is given in chapter 2.
- Building Block selection: In the framework of composing network services, the level of granularity and the idiosyncrasy of the fundamental building blocks that should be used is not yet defined. Even the names used for defining the fundamental blocks are diverse [Braden et al., 2003] [SILO] [Schwerdel et al., 2009] [Bouabene et al., 2010] [Netserv] (building blocks, atomic services, functions, functional blocks, etc.) depending on the proposal. Although the concepts proposed on those projects are based in the same groundings, not only the names of the building blocks vary, but also their level of

complexity. Some approaches use functions as much indivisible as possible, atomic, as building blocks. That permits to define with a high level of granularity the resultant composed service. Contrarily, other projects or initiatives tend to design coarser services, that limits the level of flexibility and definition of the composed service, but it facilitates and speeds up the composition process. In addition, depending on the composition context, the level of granularity can be higher or lower, and the abovementioned projects only focused on Future Networks, not considering other fields and composition opportunities such as those that can arise in Smart Grid fields (e.g. cybersecurity or Internet of Things services).

- Composition methodology: Different mechanisms are used in order to combine building blocks. Depending on the level of dependability of user interaction in the process are usually classified as manual, automatic and self-automatic approaches, relying on templates, instances, or any previous knowledge that guide the composition process in one direction or another. The level of flexibility and dynamicity of the composed service has a big dependency on the method used to combine and orchestrate them. Greedy solutions could be used looking for the most suited composed service among all possible solutions. However, the time to resolve the composed service is prohibitive when number of possible building blocks combinations raise and the communication establishment time is critical. Heuristics can be used to reduce that time and the number of possible solutions. A first review of the different composition philosophies and techniques that are followed by the projects is given in Chapter 2.
- Prerequisites and effects: Composition process may also be guided by requisites and desired effects pre-defined by the user, application or the orchestrator agent. These preconditions are a set of the heuristics that can limit the number of possible combinations, and, hence, reduce the composition process time. However, not all the former network service composition proposals define them in the same way (some not even use them at all).

However, several challenges might be faced for designing an integral solution for the service composition in the FNs that allows to overcome the current technologies and deficiencies.

A new concept and definition of services for the FNs will be closely connected to the innovation of heterogeneous environments formed by different kind of networks and users with different requirements. The design of a FN architecture as proposed in this thesis will allow adopting futuristic capabilities. Complex and personalized users' requirements introduce the need of networks able to be self-configurable and self-evolvable. Considering this, the service composition technology should be also extended to cover possible changes derived from the service and network evolution.

New features such as data and service identification, context-awareness, seamless service discovery and composition, etc. are required in order to meet the new demanded services and modes of interaction. The lack of these features is withering as well the evolution of networks and slowing down or stopping solutions for known open issues like mobility, flexibility, security, etc. A service-aware architecture should help the deployment of clean-slate solutions in all these aspects.

The proposal presented in this thesis advocates for defining a service composition methodology an easy way of creating new complex services over FNs, discovering the service capabilities available in the network, processing their composability restriction rules, and finally chaining them sequentially or concurrently in a desired workflow. This process implies the definition of each composite services that user requests as a bag of atomic services, and the orchestrator should process all network instances of these services and rank them for creating the optimum workflow in the network. In this sense, there are several input parameters to take into account in the service composition: QoS Requirements (minimum and optimum), AS needed (and preferred by the service requester) and context constraints and preferences. These parameters should be matched with the information of the services that is retrieved from other network nodes: QoS capabilities, AS availability, and context information (by means of nodes, services, users and network profiles).

However, depending on the context of the field of action, the requirements can drastically change during the service operation, but the same methodology can be followed for providing dynamic, flexible, context-aware services. In this thesis, first I considered them in a generic way over a FN and then I applied them on use cases of context-aware media distribution and Smart energy services.

Actually, to assess the application viability of service composition concept on FN architecture context became a challenge by itself, because there is not any FN architecture yet that is able to work at service level or support context-awareness capabilities as described later in this thesis. Therefore, Chapter 4 summarizes how a FN architecture should be in order to support Service-Oriented Internet and context-awareness concepts, and exposes the resulting architecture (that was designed during TARIFA project) based on these principles.

In this context, service composition methodology is defined and tested in Chapter 5 from two perspectives: (1) an isolated service composition point of view and (2) considering its operation over a Service-Oriented FN architecture.

Besides this, in order to assess the application of context-aware service composition in other more mature fields, the usage of the same methodology and framework, is also analysed and tested in other fields, such as media streaming adaptation (Section 5.6.3) and security in the smart metering infrastructure of a Smart Grid (Chapter 5.7).

1.1 Objectives

According to the main hypothesis stated in previous section, this thesis has the objective of exploring the applicability of service-oriented perspective in the fields of media distribution and Smart Grids. More concretely it wants to define a service composition methodology that, in conjunction with novel Future Network architectures enables the orchestration of dynamic, flexible and context-aware communications and processes. After that the proposed solutions are analysed for specific use cases for enhancing the efficiency or offering advanced features applicable to media distribution and smart grid services.

Now, at this point after the motivation and research statement has been exposed, concrete objectives should be defined. Those objectives have been extracted during the development of the thesis along the different phases that I have faced in order to explore the application of the service-oriented paradigm to different fields. As mentioned before, the starting point appeared on analyzing the current Internet operation and behavior, and identifying a set of flaws and malfunctions that were accentuated by current-day media distribution requirements. Indeed, the first challenges came in completely understanding the novel media problems and challenges, and then to get into the service-oriented perspective and how it can be applied to mitigate the current situation. In the meanwhile, other opportunities and potentialities of application appeared in fields such as the IoT, and more concretely the Smart Grid. That made me extend the scope of this thesis and led to an exploratory research aiming at highlighting the fundamental aspects and requirements of this additional field of application.

Therefore, the specific scientific objectives of this thesis are the following:

Objective 1. Service composition:

Study and propose solutions in order to enable the provisioning of Future Internet services by means of context-aware service composition to provide adapted and personalized services, dealing with high dynamic and heterogeneous environments. Specifically, the research done in this field aims at providing solutions for the provisioning in Future Networks (for Future Media Internet and Future Internet of Things services). Service composition on FNs topic will address the requirements services, innovative business models, access methods, devices and interactions, exploring service-oriented paradigm application to FNs and design a service composition methodology and mechanisms that can be deployed over a service-oriented FN architecture.

The shift of paradigm, from layered models to non-layered/modular SOA-based models poses new challenges to protocol design; requiring extensive research on feasible techniques for building up optimized protocols from scratch according to different criteria, ranging from context-awareness to energy efficiency. Hence, we should evaluate which models are best suited to achieve certain behaviour (in terms of functionality, QoE, desired QoS, user preferences, mobility support, etc.). This reasoning should take into consideration the characteristics of the surrounding context (device capabilities, available network and computation resources, location and environment, etc.) covering the solving of the following issues:

1. which behaviour and outcome is desired,
2. which functions are most suitable to achieve desired outcome,
3. where they should be allocated,
4. which mechanism (protocol, language/ontology) will be used to allocate and configure functions along communication path.

Other issues to be solved are:

5. discovery of desired services according to their semantic description

6. mechanisms for context interchange
7. resource reservation
8. ontologies for describing context characteristics (node, link, service, etc.), QoS agreements, service/resource description, locator/identifier schemes, etc.

In order to analyze all those aspects, fist milestone to accomplish is to *revise all the former projects and initiatives on FNs and find or design a modular service-oriented FN architecture that enables service provisioning flexibility* (besides fulfilling current and future networks requirements). Taking into account the amount of FN clean-slate architectures defined in previous projects and initiatives, the foundations of that architecture are already in the literature, although the formalization of a FN architecture that inherently works with flexible and dynamic service provisioning was not undertaken yet at the time of this thesis.

Secondly, *research and definition of negotiation protocols and distributed service composition mechanisms* are also necessary. A lightweight and QoS-aware service negotiation protocol that integrates semantic service discovery and context-aware routing is a required tool to jointly work with the composition mechanisms over the service-oriented FN. This protocol should be generic and agnostic to the underlying technologies. Hence, the envisaged protocol should help the interconnection of different networks (technologies and architectures) and allow end-users and applications to look for and to reach any type of service through highly heterogeneous networks. It would permit users to discover and negotiate services they currently may not have access to, i.e. located in other networks or accessible through gateways and middle boxes (which means a great cost in configuration and maintenance.

And finally, *define a service-oriented framework for composing network functionalities* that gets integrated with the previous work. This task will involve some of the following aspects.

- Specification of the atomic functions required (starting with a limited set for specific use cases such as context-aware media service distribution).
- Identification of the relationships among functions in order to compose services.
- Specification of the atomic mechanisms, which will allow to deploy the atomic services.
- Validation of the specified atomic mechanisms.
- Specification of the relationship among atomic functions (rules, agreements, etc. which will be inputs and constraints in order to compose context-aware services) will also be defined in order to allow their composition and application to specific use cases.

Objective 2. Context-aware Media Distribution:

The cost-effective, efficient and scalable delivery of media streaming services from across the Internet –at home or on the go, for live or on-demand content, with the ability to stop and go and other viewing options– requires context- and network-aware solutions, rather than agnostic, overlay or application level solutions as is the current practice.

Indeed, the quality of the user experience, the perceived simplicity of accessing and interacting with systems and services, and the effective and acceptable hiding of the complexity of underlying technologies are determining factors for success or failure of these novel services. In order to maximize the quality of experience (QoE) of users, multimedia resources must be adapted to the specific context where they are going to be consumed.

In addition, users access Internet services through application layer protocols, which have not been designed to capture and convey the lower-level (hardware, firmware) and networking characteristics of the devices by which the users access the Internet. This obviously results in poor QoE and a waste of Internet resources.

Existing solutions present several shortcomings that should be solved in order to improve the experience of the final user such as: collecting the user information, the search and retrieval of the heterogeneous media, the complexity of personalization and contextualization aspects and a more immersive user interface adapted depending on the media content, the user profile and the device characteristics. To overcome this deficiency, some current practices [Xue et al., 2011] rely on dedicated client-server protocols or on a manual configuration. It can be argued, however, that the user should be the focus of service provisioning instead, in order to obtain the requested content always adapted to the preferences of this specific user. This assumption implies that all relevant user parameters should be considered when establishing a communication session. Thus, we need mechanisms to gather information related to the services' execution requirements, in order to provide seamless access to services and maximize users' QoE.

This thesis aims to contribute as well to the *integration of service composition mechanisms in the media service distribution in order to provide more efficient media transport and delivery, besides context-aware and personalization capabilities*. It proposes the research for a solution that adopts and integrates different techniques for estimating the level of quality of the media (subjective and objective) and combines them to transcoding tools that enable to provide the content transparently adapted to the user and network requirements.

Therefore, the following topics are tackled in this thesis:

1. How service-oriented FN architecture and the service composition framework can offer a good solution for the selection of context-aware media services,
2. which services should be defined in the context of media transport, and how they should be designed,
3. which metrics and which cost function offer a good option for selecting the most efficient composition depending on the user and application.

Objective 3. Software Defined Utility infrastructure

Extending the scope of the thesis, this third objective focuses on the Smart Grid communications and processes. Indeed, prior to the research undertaken related to this objective, a great amount of work was dedicated to gain the necessary knowledge to perform adequately study the requirements focusing mainly on communication protocols, and data management processes in the Smart Grid and the energy sector.

Deriving from the research analysis on the application of service composition for Smart Grid, several other different aspects are targeted. Actually, the overall objective was to define potential mechanisms for a future Software Defined Utility [Martín de Pozuelo et al., 2016]. Hence, this thesis not only analyzed how service composition techniques can apply, but also how other FN trends and novel ICT solutions can facilitate and simplify Smart Grids operation and management automation. In that sense, different issues were expected to be studied in this thesis.

First, the security mechanisms of the Smart Grid and more concretely the Smart Metering infrastructure were analyzed, identifying a deep lack of flexibility although different off-the-shelf available options can be selected. Therefore, a *research on how service composition methodology can be applied to improve the flexibility and context adaptation of the Smart Metering security services* is missing. Additional studies were also derived, revolving around the study of the cybersecurity and data management of the Smart Grid, such as the evaluation of context-aware systems as orchestrators of the communications QoS, reliability, data storage allocation or cybersecurity mechanisms that should be provided in different regions of the Smart Grid.

Moreover, another challenge was identified in the integration and monitoring of the heterogeneous data generated by every device on the Smart Grid (e.g., wired and wireless sensors, smart meters, distributed generators, dispersed loads, synchrophasors, wind turbines, solar panels, and communication network devices) into a single interface. Although the latest developments on the IoT (Internet of Things) field have definitely contributed to the physical connection of such an overwhelming amount of smart devices, several issues have arisen when attempting to provide a common management and monitoring interface for the whole Smart Grid [Zaballos et al., 2011] [Navarro et al., 2014]. Following the work done and the experiences acquired during INTEGRIS and FINESCE projects, the aim is to implement an ICT infrastructure, based on the IoT paradigm, to handle the Smart Grid storage and communications requirements [Oualmakran et al., 2011] [Herzog, 2014] to manage the whole Smart Grid and link it with end-users using a WoT-based approach, which results in a new bridge between the IoT and WoT (Web of Things).

Finally, research was also undertaken focusing on a flexible data management system for the Smart Grid. A distributed storage system is proposed, based on a context-aware dynamic configuration of the nodes that collect and store the data gathered from the Smart Grid at distribution level. In addition, a specific challenge has been studied on this topic, regarding the application of hybrid (private and public) clouds to store the huge amount of information collected in the Smart Grid environment. In this sense, the research seek an orchestrator for the hybrid cloud in Smart Grid environments, as a system that decides in which cloud the resources and data collected by the Smart Grid should be placed according to context information (e.g. security or latency requirements).

In summary, the following aspects are covered:

1. how service composition mechanisms can be adopted in the Smart Grids context,
2. how to design service-oriented reasoner to offer context-aware Smart Grid services,
3. which components of the Smart Grid can be redesigned in a service-oriented manner in order to define a Software Defined Utility.

1.2 Organization

This document is organized in 7 chapters. The structure of the document is as follows:

- *Chapter 1: Introduction and motivation.* In this chapter, the thesis is introduced by addressing the motivation, research statement and objectives.
- *Chapter 2: Context-aware media distribution and Smart Grids: Two different fields of application?.* Before going into details of the work done in the thesis, this chapter presents the different fields in which the service composition methodology has been explored during the development of the thesis. It introduces the reader to these fields and details their general context, requirements, challenges and opportunities that were considered.
- *Chapter 3: Service Composition.* This chapter summarizes the research previously done regarding service composition process, from the Service Oriented Architecture (SOA) paradigm definition, to its application to other initiatives and projects proposing FN architectures.
- *Chapter 4: A Semantic Context-Aware Network Architecture*. Before going into deep on the details of the service composition framework proposed in the thesis, this chapter gives a brief review of the service-oriented FN architecture model that was considered for the subsequent research and development related to FNs.
- *Chapter 5: Framework and mechanisms of context-aware service composition over a Future Network architecture.* This chapter elaborates the methodology proposed for the service composition process, starting from the services definition, and going through the service discovery, scoring and chaining processes. It also details the experimentation and evaluation of the framework that was tested from an isolated point of view, and afterwards over a preliminary service-oriented FN architecture implementation. At the end, it also includes the work undertaken exploring its application for context-aware media service distribution.
- *Chapter 6: Service composition on Smart Grid: from context-aware network management to flexible application deployment.* This chapter depicts the research and development done in the field of Smart Grids, analyzing how the Service-oriented paradigm can change the way we integrate ICT in the electrical grid and their components. It details the different actions that were undertaken in projects such as FP7 INTEGRIS and FP7 FI-PPP FINESCE and their derived work towards defining a Software Defined Utility infrastructure, with examples such the application of service composition for designing dynamic and flexible security for smart metering, or the orchestration of services and resources within the utility infrastructure.

- *Chapter 7: Conclusions, research outputs and future work*. Finally, this chapter exposes the conclusions of the thesis, specifies the contributions done during its elaboration and discuss the future directions to be taken.

In addition, the Figure 1.1 exposes the path followed to develop the thesis, highlighting some milestones, publications, participation in projects and other relevant outputs.

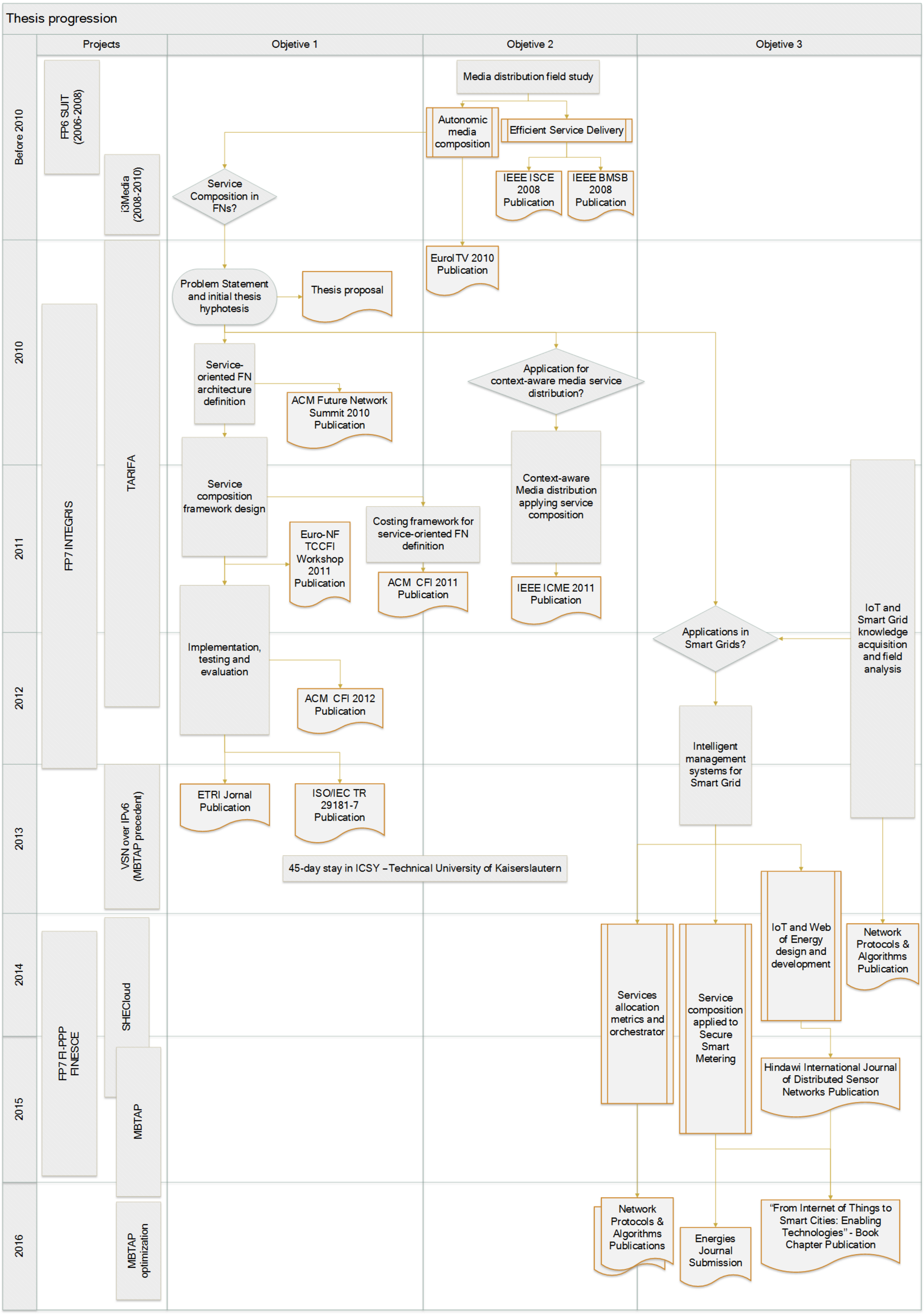


Figure 1.1. Thesis progression chart

2 Context-aware media distribution and Smart Grids: Two different fields of application?

Getting started is not easy. I do not like when someone starts explaining something expecting that all the audience are experts and know from A to Z the topic that is being exposed. It is difficult to tackle this issue in a PhD thesis in which explaining from the very beginning of everything you have learnt of a research topic would be probably outrageous.

It could be done starting directly explaining about service composition and its principles, but I preferred to invite the lecturer to get into the topic the same way I did: by acquiring the knowledge of the different fields of application in which I did my research.

It is important to take some time to overview *media distribution* and *Smart Grids* fields, making the lecturer understand which are their main characteristics and challenges. As was explained in the Chapter 1, I did a gradual migration (or integration) from one field to the other. I started realizing the potential application of the service composition tools to other fields such as the configuration of the Smart Grids and IoT communications and management processes. That shift was confirmed two years after I started my PhD, when I moved from Media Technologies Group (GTM) to the Research Group on Internet Technologies and Storage (GRITS).

Thus, the answer to the chapter title would be a ‘yes’, they are two fields with so different characteristics and challenges to be explained separately in this overview chapter. However, later along the rest of the thesis you will find some synergies between them (in the same way I found them doing my research).

2.1 Media distribution

Media distribution field actually encompasses a bunch of topics, but a coarse classification could be made into two different aspects, all the issues related to media/content, and all the other related to networks (the mean of distribution).

In order to limit the scope of this section, it will be focused on the network part. Taking into consideration the objective 2 of the thesis (apply service composition tools for providing efficient context-aware media distribution), the network context, the evolution that networks have been suffering in the recent years and the future challenges become especially relevant [Azcorra et al., 2009].

Currently there are over a billion Internet users and with machines and appliances, also connecting to the Internet (the “Internet of Things”) the number of users will increase rapidly. One of the most significant reasons for the fast growth of the Internet comes from multimedia and, concretely, from video. Current Internet traffic is increasing very fast mainly due to the proliferation of media capable devices, media services and their demand by Internet users.

Some analyses [CISCO, 2016] confirm that video traffic will keep growing at a tremendous pace. Next statements show some numbers and predictions in Internet video growth:

- *It would take more than 5 million years to watch the amount of video that will cross-global IP networks each month in 2020. Every second, a million minutes of video content will cross the network by 2020.*
- *Globally, IP video traffic will be 82 percent of all IP traffic (both business and consumer) by 2020, up from 70 percent in 2015. Global IP video traffic will grow threefold from 2015 to 2020, a CAGR (compound annual growth rate) of 26 percent. Internet video traffic will grow fourfold from 2015 to 2020, a CAGR of 31 percent.*
- *Internet video surveillance traffic nearly doubled in 2015, from 272 petabytes per month at the end of 2014 to 516 petabytes per month in 2015. Internet video surveillance traffic will increase tenfold between 2015 and 2020. Globally, 3.9 percent of all Internet video traffic will be due to video surveillance in 2020, up from 1.5 percent in 2015.*
- *Virtual reality traffic quadrupled in 2015, from 4.2 petabytes (PB) per month in 2014 to 17.9 PB per month in 2015. Globally, virtual reality traffic will increase 61-fold between 2015 and 2020, a CAGR of 127 percent.*
- *Internet video to TV grew 50 percent in 2015. This traffic will continue to grow at a rapid pace, increasing 3.6-fold by 2020. Internet video to TV will be 26 percent of fixed consumer Internet video traffic in 2020.*
- *Consumer video-on-demand (VoD) traffic will nearly double by 2020. The amount of VoD traffic in 2020 will be equivalent to 7.2 billion DVDs per month.*
- *Internet gaming traffic will grow sevenfold from 2015 to 2020, a CAGR of 46 percent. Globally, Internet gaming traffic will be 4 percent of consumer Internet traffic in 2020, up from 2 percent in 2015.*
- *Content delivery networks (CDNs) will carry nearly two-thirds of Internet traffic by 2020. Sixty-four percent of all Internet traffic will cross CDNs by 2020 globally, up from 45 percent in 2015.*
- *Content providers and distributors could adopt P2P as a distribution mechanism. There has been a strong case for P2P as a low-cost content-delivery system (CDS) for many years, yet most content providers and distributors have opted for direct distribution, which offer live video streaming through P2P and have had great success. If content providers in other regions follow suit, traffic could rapidly become highly symmetric.*

The sum of all forms of IP video, which includes Internet video, IP VoD, video files exchanged through file sharing, video-streamed gaming, and video conferencing, will continue to be in the range of 80 to 90 percent of total IP traffic. Globally, IP video traffic will account for 82 percent of traffic by 2020 (Figure 2.1).

Indeed, because of its gigantic growth, the Internet is reaching some technological and operational limits imposed by its architecture in its attempt to give full support to the new requirements introduced by the increasing number of new services, applications and contents. Future Media Internet group of the European Commission envisaged [Azcorra et al., 2009] a scenario with a heavy increase of user-generated content (UGC), convergence of different forms of media, natural multi-sensory interfaces, virtual learning environments, etc.

Figure 2.1. Global IP Traffic by Application Category [CISCO, 2016]

These new applications demand for high bandwidth and place new types of traffic demands and constraints on network architectures. Thus, there is a worldwide concern in the networking research community about the future of Internet and its manageability, security, openness to innovation and scalability.

2.1.1 Media networks characteristics, scenarios and protocols

Together with the abovementioned, it is relevant to remark that, in general, many communication networks supporting media applications are characterized by a wide variability in packet loss, delay, and throughput. Moreover, there is a high variety of receiving devices with different resources and capabilities commonly connected to a network, which results on poor Quality of Experience to the end-user [Brooks et al. 2010].

If we analyze the traffic characteristics requirements of modern media networks, it can be said that we need to achieve a minimum bandwidth –2Mbps for HD video streaming– with a low latency –less than 150ms–, jitter –less than 30ms– and error rate –less than 1%– (Figure 2.2).

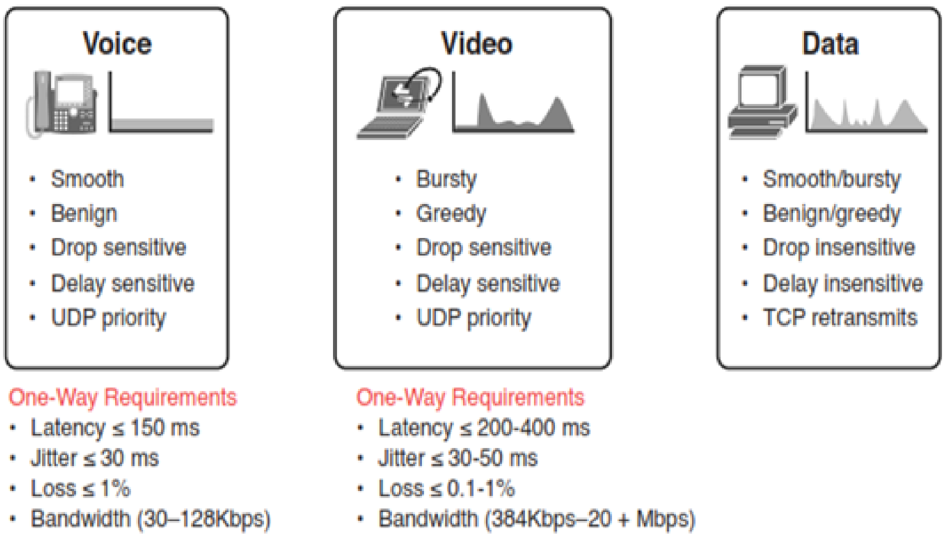


Figure 2.2 IP major traffic characteristics [Sziget et al. 2013]

If we go deeper into the video network scenarios, three cases can be differentiated: download, streaming of prerecorded content and live video transmission. For download, the entire prerecorded and encoded video is received and stored in the receiver before playback commences, so the user waits for a longer time until playing the video. For streaming, the video playback can begin as soon as possible, or may use a buffer that adds a delay to the start of the video, but tends to ensure continuous playback during periods of network congestion when the delivery of encoded video frames over the network slows down. In the case of live video transmission two subcategories can be defined whether it is an interactive video transmission (e.g. a video conference conversation), or not (e.g. streaming a live music concert).

Among all the mentioned scenarios, the live interactive video is the one that poses the most restringing requirements. In those cases, the one-way end-to-end delay, including the delays for video encoding, network transmission, and video decoding, should preferably be less than 150 ms to preserve the interactive conversational nature of the communication.

However, nowadays exist standards such as H.264/AVC and H.265/HEVC offer a wide range of encoding options to accommodate the timing and other constraints (e.g., computational capabilities) of the different video transmission scenarios. For instance, for live interactive video transmission, low-delay encoding options arrange the inter-frame dependencies to permit fast encoding of video frames from a live scene to avoid extensive delays due to waiting for the capture of future video frames.

Regarding streaming protocols, in the first decade of the century research attention has mostly focused on the design and implementation of new streaming protocols, such as the design of the Real-time Transport Protocol (RTP) [Schulzrinne et al., 2003] specifically for streaming media. RTP was proposed for end-to-end real-time transfer of stream data, trying to detect packet loss and compensate jitter during transmissions over an IP network. RTP was designed with two fundamental principles: application-layer framing (i.e., framing for video data should be performed properly by the application layer) and integrated layer processing (i.e., integrating multiple layers into one to allow efficient cooperation).

Internet Engineering Task Force (IETF) standardized the RTP/RTCP/RTSP protocol suite [Schulzrinne, 1998] [Schulzrinne et al., 2003] designed specifically for Internet video streaming. Based on the User Datagram Protocol (UDP), RTP defined a standardized packet format for delivering video over IP, and was designed for end-to-end real-time transfer of stream data. The RTP Control Protocol (RTCP), also based on UDP, was designed to monitor transmission statistics and QoS, and to achieve synchronization across multiple streams. RTSP is an example of session control protocols, similar to the Session Initiation Protocol (SIP) that was later designed [Rosenberg et al., 2002].

However, along the last 5 years after the start of the thesis, streaming over HTTP has been increasingly employed by major content providers, more and more video streaming systems have been built and deployed with HTTP streaming. HTTP video streaming can be regarded as downloading video segments progressively from web servers via the HTTP protocol, so that clients that support HTTP can seek to arbitrary positions in the media stream by performing byte range requests to the web server. Figure 2.3 shows an example of an scenario where recorded video is injected and stored in a media cloud and then served to end-users through HTTP streaming.

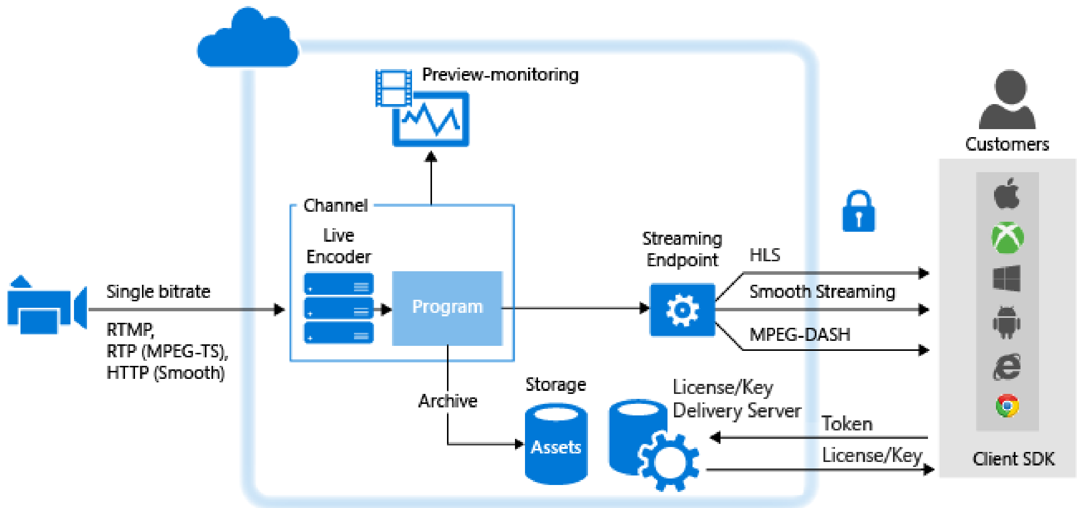


Figure 2.3 Example of a scenario with HTTP Streaming in a media cloud [Kornich, 2016]

A clear advantage of solutions such as MPEG-DASH, HLS or Smooth Streaming is that going through HTTP port 80 they are usually able to go across different networks and through firewalls. WebRTC [Bergkvist et al. 2012] seemed to appear later specially for trying to avoid those kind of streaming problems; however, due to its limited deployment by browsers, a solution to stream videos over HTTP has seen a substantial amount of industry support.

And last but not least, it is important to point out that the distributed flavor of the HTTP streaming file format (divided in chunks that can be classified for one layer or another) enables to easily provide adaptive quality to the end-users (Figure 2.4).

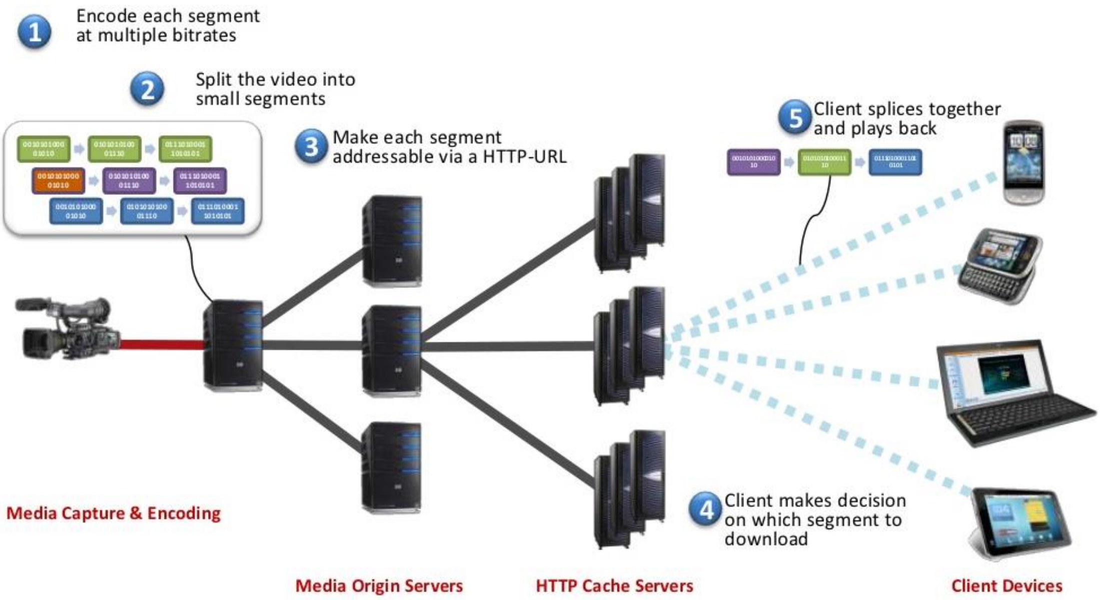


Figure 2.4 Adaptive HTTP streaming process [Tapper et al., 2013]

2.1.2 Media networks future challenges

In order to accommodate the amount of novel media services requirements and the increasing need of higher bandwidth, Future Media Internet Task Force of the European Commission remarked some years ago several needs [Azcorra et al., 2009] of the future media networks:

- *New traffic patterns;*
- *Content adaptation in the network and the terminals that enable the availability of media for a range of heterogeneous devices;*
- *New models of content distribution that consider not only the protocols involved but also the social characterization of the nodes and the data;*
- *New network management modules and mechanisms to provide and monitor Quality of Experience (QoE), trust and privacy.*

Further, they define a set of research challenges that are summarized in the following:

- <u>Content Distribution</u>:
    - New architectures and overlay networks for content distribution. The main research challenge related to new architecture is the (dynamic, autonomic and self-organized) creation of overlay network infrastructures to support the provisioning of media services to end-user communities. Some of the issues related to overlay networks have a wider impact and span, in fact, multiple areas. For instance, the specification and measurement of QoS parameters and other metrics that can be used to assess the underlying communication technologies achieve network friendliness (via local caches), suggest most suitable service instances for the end---user. A possible approach to deal with these cross-layer issues is to gather information from the underlying networks and combine it with higher-level quality assessment and requirements of applications to adjust the overlay networks.
    - Content-Centric networks. Content-aware real-time transmission of future media means that the relative importance of each packet towards increasing the end-to-end utility function is established. That is, the more important packets should be better protected (by allocating appropriately network resources) or should be transmitted first in a scheduling scenario.
    - Content/Information driven routing. Internet routing system shall be capable to consider associated routing information and metrics for path calculation such as the link quality, security level, energy consumption, priorities or location.

- <u>Content adaptation:</u>
    - Personalization. The highly heterogeneous environment in terms of diversity of end user devices, networks and user preferences will remain. To ensure a real seamless access to new media applications, it is desirable that the network itself and the services could automatically realize content adaptation and enrichment inside the network.

- In-network content enrichment. Novel methods for in-network content enrichment and cross-network adaptation will be needed to allow for optimal use of available resources and enriched QoE. By dynamically combining the inherited content scalability of the same resource transmitted from multiple sources (e.g. different servers or peers in case of P2P streaming) and/or received over diverse paths or networks (use the MDC features), on-the fly content adaptation, inherited resiliency and enriched QoE may be achieved. Reconstruction of the content segments may take place either within the network or at the edge of the network (at content aware edge routers) offering transparent streaming to low-end terminals or at the terminal side in case multi-network connectivity is available. Cross-network adaptation and in-network content enrichment especially in P2P overlay topologies, will offer traffic adaptation (load balancing to avoid network flooding), optimal use of available resources (bandwidth) and enriched QoE.

- Quality of Experience (QoE):
  - QoE evaluation. Quantification of QoE using objective and subjective measurements remains a challenging research problem. Furthermore, the relationship between network level QoS measures and overall QoE must be studied.
  - QoE monitoring. Tools and methods to acquire real-time feedback of the QoE received by the users will be needed. This information can be used by autonomic media adaption mechanisms in order to modify the service parameters accordingly to the response received from the user and trying to improve the perceived QoE. Figure 2.5 shows a basic example of a possible channel optimization using QoE monitoring.

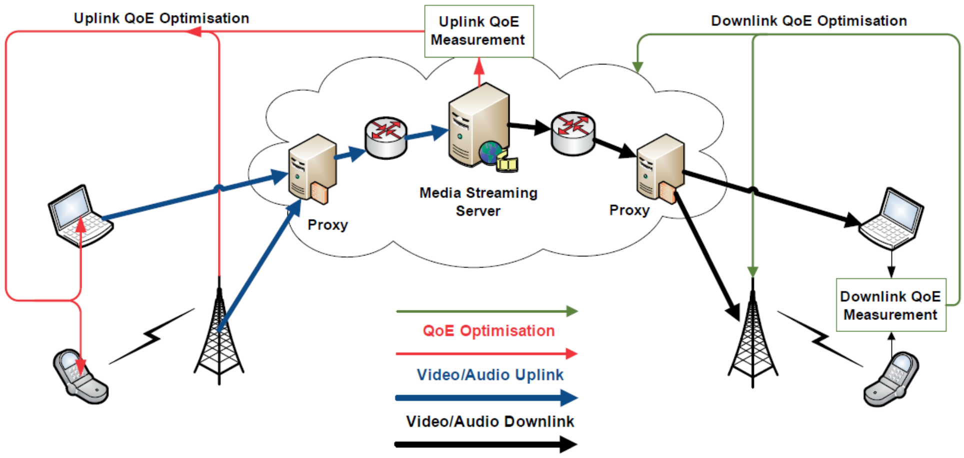


Figure 2.5 Proposal of network optimization using QoE monitoring [Azcorra et al., 2009]

- Identity, Trust, Privacy and Security:
  The content that is being produced and distributed is increasing rapidly. Users expect to be able to take advantage of the future widespread availability of multimedia content and access to virtual worlds. At the same time, they need to feel confident that their security and privacy are being protected. The increasing complexity and scale of future media systems make problems of Identity, Trust, Privacy and Security are harder to solve.

2.1.2.1 Thesis contributions

The contributions of this thesis are focused on the benefits that service composition approach can provide to each of these challenges. The increasing need of real-time media adaptation and higher-standard demands by end users on the level of QoE received, seek solutions that can cover the fine-grained personalization of content. Furthermore, the increasing need of flexibility is not only on the result of the service towards the user, but in the media distribution itself and the provision of new features not even available with current network architectures. From the dependency on Content Delivery Networks (CDNs) to the rise of Fog Computing, the increasing need of in-network services is a current reality. We need mechanisms to deploy these services in an efficient, flexible and controlled way in the network. The combination of service composition with FN architectures proposed in this dissertation represents a promising approach in this direction.

2.2 Smart Grids

This section will overview the Smart Grid ecosystem and environment. However, analogously to the previous section, the scope of this overview will be limited. In this case, it will try to summarize the basic concepts of the Smart Grid, but it will especially focus in detailing the communications and security areas, the more relevant aspects to the research later elaborated in Chapter 5.7.

Smart grids are electricity networks that use digital technology to co-ordinate the needs and capabilities of all generators, grid operators, end users and electricity market stakeholders in such a way that it can optimise asset utilisation and operation while maintaining system reliability, resilience and stability [Gungor et al., 2011] (Figure 2.6).

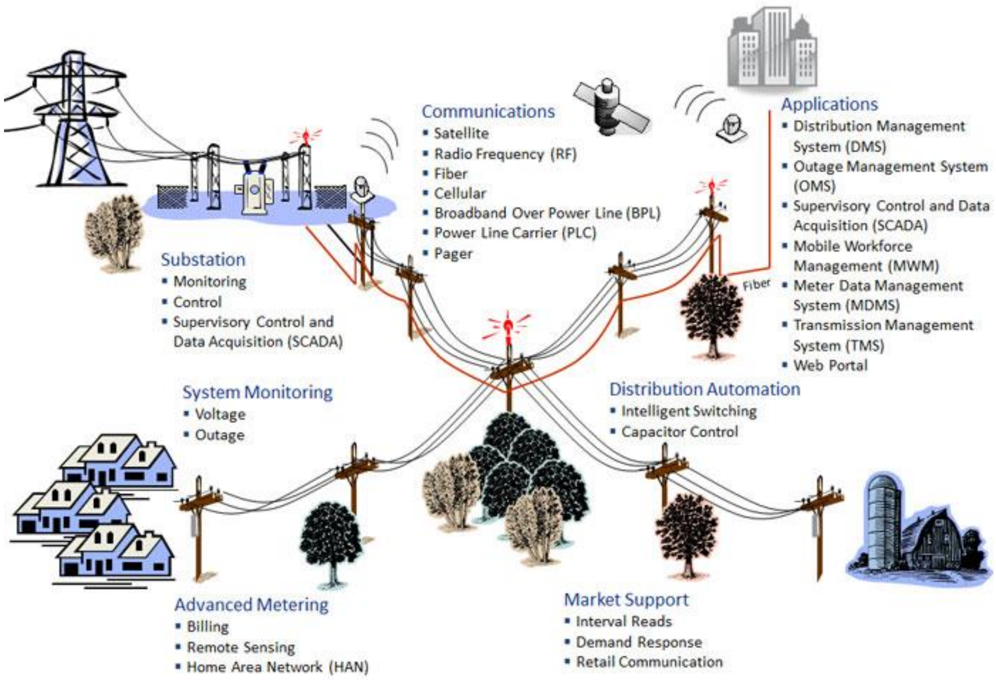


Figure 2.6 Examples of Smart Grid technologies [ONCOR]

The use of new communication and network technologies (wireless, mobile, PLC), facilitates access from/to information (i.e. smart metering, sensor networks) and development of cooperative (agent /multi-agent view) strategies are fundamental for the development of new monitoring systems (including SCADAs, communication and overall information system access). From power operation perspective, new techniques should be developed capable to deal with complexity (combinatorial explosion) and flexibility (continuously changing network) of smart grids, without neglecting power quality, adaptability to new topologies, and management of huge data flow issues.

First, a brief description of the three groups of the Smart Grid components (plus communications, which will be expanded later) (Figure 2.7):

- Advanced Meter Infrastructure (AMI): The AMI systems are the primary means of communication for companies to interact with service counters located in the band of customers. AMI refers to systems that measure, collect and analyze data on the use of energy from devices such as electricity meters, gas or water. This infrastructure includes hardware, software, communications, customer associated systems and management software for meter data (MDM Meter Data Management). The network between the measurement devices and enterprise systems allow the collection and distribution of information from/to customers, suppliers and service companies. This allows for participation and cooperation between the business units of different parts. Moreover, the fact of reporting information to customers allow that it can adapt its consumption based on peak usage or prices. Therefore, you can perform tracking functions, management of load peaks and sending signals in real time clients.

- Advanced Distribution Automation (ADA): ADA systems implement advanced infrastructures control methods in order to optimize their efficiency. ADA is concerned with complete automation of all the controllable equipment and functions in the distribution system to improve strategic system operation. An example of ADA developments can be the development of advanced tools to improve power network construction, trouble-shooting and repair.

- Distributed Energy Resources (DER): The change of paradigm that is happening in current-day electricity networks is that they are moving from a single point of generation to multiple distributed microgeneration power resources (generally renewable energy resources). DER components aim to reduce losses due to transport and distribution approaching generators to consumers. This allows diversify thus helping to mitigate the effect of intermittent generation sources by combining many heterogeneous sources of generation. Due to the increased generation of renewable energy, energy storage systems become also essential, for those moments in which the generated energy is not consumed.

The concept of Smart Grids has been considered in last years as the appropriate answer to address the new challenges in the energy domain: network reliability, energy efficiency, new distributed renewable energy sources and increased network complexity. However, multiple barriers appear in the road to realize these achievements. Proactive operation of the grid, efficient integration of demand into grid operation, optimal integration of renewable generation or the integration of Smart Grids with other energy networks still need additional resources to get applicable solutions.

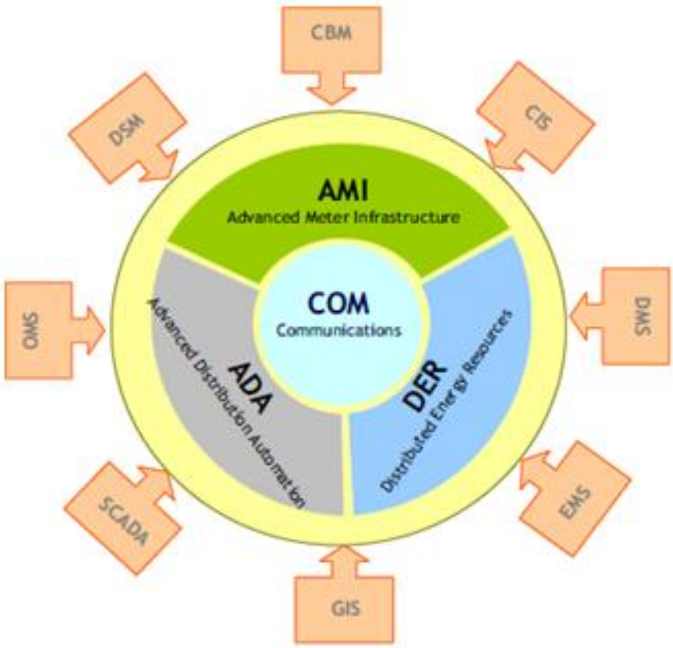


Figure 2.7 Smart Grid components [SMARTCITYMALAGA]

Indeed, not all the parts of what this trend involves are equally deployed [Gungor et al., 2011] [Gungor et al., 2013]. There are many times when they have not been integrated together in previous systems [Zaballos et al., 2011]. Partial solutions targeted only to specific aspects of the power system are no longer valid given the many services to be provided and the high cost of deployment of many specific systems [Selga et al., 2014]. It is clear that the energy sector still requires mechanisms to efficiently integrating smart grid components and communications.

2.2.1 Smart Grids requirements and challenges

There are still several challenges to be tackled on the Smart Grid ecosystem. For the limited scope of the thesis, this section will only focus on the communications, security and network operation and restoration. In the following, some of them are specified [Oualmakran et al, 2011]:

- Adaptation to new topologies: Actual distribution networks assume radiality from transformation substations, but moving towards looped or meshed structures is necessary since they enable power to be supplied from many points and improve the reliability and efficiency of the restoration. Islanding has also to be considered as an operational mode referring to a part of a network that remains energized while being isolated from the main network; however, it involves some risks: safety of troubleshooter, power quality for the consumer and damage for distributed generators. A multi-agent conception of the grid is necessary to deal with coordination and optimization requirements for an efficient and sure network.
- Novel monitoring strategies: It is needed to research on monitoring strategies based on decentralized architectures (service oriented, multi-agent), supported by real time middleware capable to deal with huge amounts of information at different time scale, process events (complex event processing) and discover sequences of them (sequence event discovery) working together with data mining solutions.

- Data flow, interoperability and security: Pervasive monitoring devices with communication capabilities increases security of the network as a whole but implies new cyber-security issues and makes optimal restoration and reconfiguration more complicated. Traditional methods were designed when little data was available and network operation simple whereas new monitoring capabilities implies accessibility (timely) and proper management of data and interoperability of applications.
- Energy data and privacy: They are a top priority for policymakers, ensuring that the energy industry delivers on its commitments to provide strong protections for both. It is expected that several states and countries will adopt smart meter data privacy and security policies, during the next years [Herzog, 2014].
- Decision support and artificial intelligence: Currently, heuristics, meta-heuristics and learning methods are mature and available. Restoration and reconfiguration techniques including auto healing will benefit from these reasoning engines.

The need for more flexibility in the network operation and the integration of new security techniques are one of the key aspects to be considered. If we take them from an architecture point of view, some multi-agent approaches, organic computing and SOAs and real time platforms have been proposed for the smart grid [FINSENY] [FINESCE]. The idea is to distribute the intelligence, implementing algorithms in services that can be hosted in nodes connected to the platform. It should be supported by extreme processing middleware capable of handling millions of events at real-time and near real-time speeds or CEP (Complex Event Processing) engines with the capacity to detect patterns in the huge flow of events that characterizes the Smart Grids. Large-storage technologies, that can hold the massive amounts of information generated by the millions of sensors implemented in the Smart Grids and that will make the information available in negligible times. Reactive multi-agent solutions to deal with outages and propose support to system reconfiguration proposed in the literature have to adapt communication protocols to those extended standards.

Notwithstanding, those proposed novel solutions have to deal with the stringent communications and security requirements of the Smart Grid. Figure 2.8 summarizes most of those requirements, while Figure 2.9 gives further details of the low latency requirements in such critical infrastructure that is the Smart Grid.

Network	Application	Data rate	Latency	Relia-bility	Security	Coverage range	Communication technologies: Wired				Communication technologies: Wireless						
							Fiber Optic	DSL	Coaxial Cable	PLC	ZigBee	WLAN	Z-Wave	Wireless Mesh	WiMAX	Cellular	Satellite
HAN/BAN/ IAN	Home/building automation	< 100 kbps	< minutes	High	High	Up to 100m				X	X	X	X				
NAN/ FAN	On-demand meter reading	>100 kbps	< 5 sec	High	High	Up to 10km		X	X	X				X	X	X	
	Multi-interval meter reading	>100 kbps	< several hours					X	X	X				X	X	X	
	Load management	>50 kbps	< 5 sec					X	X					X	X	X	
	Distribution automation	>18 kbps	< 1 sec					X	X					X	X	X	
WAN	Synchrophasor	> 2 Mbps	< 20 ms	Very High	Very High	100 km or more	X								X	X	
	Backhaul/core/metro networks	>10 Mbps	< 50 ms				X								X	X	X

Figure 2.8 Smart Grid communications general requirements [Kuzlu et al., 2013]

As it can be read in these figures, protection and monitoring information speed mean very low latency in order to avoid potential blackouts and millions loss.

In addition, the Smart Grid will generate many data from thousands of devices and energy consumers themselves [Cupp et al., 2008]. These data must be converted into information through a knowledge management cycle in which data from counters and applications substations and distribution systems are analyzed and integrated. Utilities can use all this generated data to estimate states (predictions based on historical data) allowing electric network management in real time.

Information Types	Internal to Substation	External to Substation
Protection Information, high-speed	1/4 cycle	8-12 ms
Monitoring and Control Information, medium-speed	16 ms	1 s
Operations and Maintenance Information, low-speed	1s	10 s
Text Strings	2s	10 s
Processed Data Files	10 s	30 s
Program Files	60 s	10 min
Image Files	10 s	60s
Audio and Video Data Streams	1 s	1 s

Figure 2.9 Smart Grid communications latency requirements [IEEE1646]

With this, it will be possible to identify trends and alert operators of possible impending failures. The first step in this effort is the retention of data in so-called data warehouse and ensure data security, action that will be vital. The fact collect data, store, analyze and visualize contribute to the development of a large number of new ideas that can greatly improve the quality of life.

Actually, if we go in depth in the communication requirements of the Smart Grid, it can be observed that the keypoints are speed, reliability and service continuity to maintain those latency-critical application of the Smart Grid always available. Hence, the following points describe how the Smart Grids and the communications networks that they rely on should be [Kuzlu et al., 2013] [Yan et al., 2013] [Yan et al., 2012] [Wang et al., 2011] [Li et al., 2010]:

- <u>Based on standards, flexible, scalable and future-projected</u>: Communications infrastructure must be based on standards at all levels to ensure the support of a good set of applications from utilities and to have a certain protection of the performed investment. It is applicable to radio communications protocols, networking interfaces and specifications of safety standards. It should also contribute to make the network and the management system more scalable, allowing to support the large associated developments at long term (usually the amortization of the electric infrastructure is 40 years long).

Taking into account the very long amortization time, the design of the Smart Grid must be flexible. The changes in the context situation of a given Smart Grid area must lead to differentiated or adaptable security policies depending on that context and requires a context-aware security design.

- High capacity IP-based network: For applications that must support the Smart Grid, there must be a network traffic to add these to be able to transport several Mbps (Figure 2.8). It should also be borne in mind that in the future will appear new applications and the network must be prepared to support them. The network must provide real-time communication capabilities to provide low latency, which is necessary for applications such as automation distribution, detection of blackouts or electrical protection systems (as shown on Figure 2.8 and Figure 2.9). Latency will be affected by both the means by which it performs communication such as routing protocols used.

  Beyond this, a full IP-based network provides the widest possible platform to offer a large number of applications, and can enable traffic prioritization (QoS) communications networks. It must be able to prioritize the delivery of critical applications and latency-sensitive, as might be the distribution automation against other no delay-critical traffic, such as meter reading data. Possible options would be the use of QoS standards of the IETF (DiffServ) and IEEE (802.11e 802.1p, 802.1Q) or mesh protocol extensions to support QoS functions.

- Reliable, resilient and secure: The fact that the Smart Grid is reliable[11] is key for businesses power can meet the requirements of customers. It should be resilient to follow operating after any unexpected alteration, such as cuts or falls circuit equipment. You must consider any alteration source, as they are the human error, extreme weather conditions, terrorist activities, etc. In order to meet the reliability requirements imposed by power companies, the network architecture should be invulnerable and able to remain operational even if there are failures. The Smart Grid should be designed to operate and maintain the continuity and throughput in a wide range of hostile conditions and temperature changes. Some solutions to these requirements are the ways of backup, mesh topologies, integrated battery backup systems protection interference, etc. The network should be operational and available time of 99.99% in a year. Moreover, some functions need availabilities 99.999% for achieve the global 99.99%.

  Moreover, the information passing through the Smart Grid is private and potentially sensitive. Therefore, the data stream for the Smart Grid should be safe to the level to ensure that this flow can be neither consulted nor altered by entities or unauthorized persons. We can say that this safety will be increased if it does not have dependence on

[11] The term "reliability" is accepted as a set of real indexes or expected that they are reported or predicted. Currently there are a lot of indexes used, such as loss of load expectation (LOLE, Loss Of Load Expectation) measured in hours/year, the expectation of energy loss (LOEE, Loss Of Energy Expectation) measured in MWh/year, the expectation of demand not provided (EDNS, Expected Demand Not Supplied) measured in MW / year, frequency load loss (FLOL, Frequency of Loss Of Load) measured in events / year and energy reliability index (EIR Energy Index of Reliability) [UPA, 2008].

third parties, as may be other operators. The main implementing measures are data encryption and password protection (authentication).

The significance of the cybersecurity in the Smart Grid is critical. The needed migration of the communications to IP and the stringent requirements on reliability, high availability and data privacy mean that the security layers of the Smart Grid should be carefully designed. Furthermore, security mechanisms are related to the operation of the network in many ways. Security procedures may introduce delays and cause the unavailability of some devices thus denying the service. When introducing cybersecurity techniques to maintain the same level of reliability into the new IP-based infrastructure, some additional considerations must be taken [Ericsson, 2010]:

- The stringent requirements in terms of latency and reliability defined in Figure 2.8 mean that security decisions are to be kept as close as possible to the affected devices. Therefore, security needs to be as decentralized as possible in the Smart Grid. The need to distribute some of the computations to be performed in the Smart Grid means that data has to be also protected when being used by the distributed applications.
- The very high reliability required for some functions implies that special care has to be taken to minimize Denial of Service (DoS) attacks to a minimum.
- Care needs to be taken regarding key management given the large amount of keying material to be handled. Special care is required regarding a common policy for the different sets of keys.
- The typical long duration of the investments in power network assets means that the horizon is about year 2056 and, since by that year it is expected that encryptions such as 3DES will be broken. It must be able to update gradually the encryption systems without stopping the Smart Grid operation.
- The high confidentiality needed for Meter Data means that, at least for these data, it is necessary to encrypt it while being transmitted and when being stored in intermediate systems.

As seen in this section, the introduction of Information and Communication Technologies (ICT) in the electric power system have facilitated the management, control and operation of the whole electrical grid. Although the energy sector is very reluctant to incorporate changes in their management models, novel communication technologies have been integrated in the monitoring and control of infrastructures along the history. The inevitable integration of IP communications and thus, the communication and security challenges abovementioned are a current-day reality in the Smart Grid. However, a lot of research should be done in the design of efficient ways to control the resources of the Smart Grid that tend to be increasingly distributed. It is important for the deployment of the full Smart Grid to evolve those solutions that allow a flexible (to make easier the introduction of new mechanisms and resources) but secure (to handle the stringent requirements of such critical infrastructure) management of the Smart Grid. The contributions of this thesis to Smart Grid field (detailed in Chapter 6) focus especially on these aspects, proposing a set of components that should simplify the deployment and integration of ICT technologies (Internet of Things, cloud computing and storage, cybersecurity mechanisms, etc.) towards a Software-Defined Utility (SDU).

3 Service Composition

Once reviewed the different fields of application, this chapter deepens on the foundations of the thesis hypothesis, which proposes to study the applicability of service composition methodology and mechanisms to media distribution and Smart Grid fields. Therefore, it explains the concept and rationale behind service composition process and its application. Besides this, it reviews different approaches that can be found in the literature and the fields of application in which it is already widespread. Finally, and taking into account that the service composition in communication networks is highly considered to be applied jointly with a migration to Future Network (FN) architectures, it also summarizes the FNs initiatives and projects that consider to provide a service-oriented perspective or service composition capabilities.

3.1 Introduction and approaches

In the last decade, distributed software systems have become more dynamic, allowing transparent distribution, self-reconfiguration, portability and migration of applications, etc. Consequently, new application development paradigms emerged such as those based on multiple services dispersed in the environment [Satyanarayanan, 2001].

These new paradigms, used in conjunction with a composable service model, provide more flexibility in the application development and execution. Considering services as self-contained, self-describing, modular applications that can be published, located, and invoked across the net [Rao et al, 2004], service composition process can be defined as the combination of those services required to create new processes and services.

This approach permits to create applications dynamically from existing services, possibly remote, enhancing the reuse of code and decreasing the development time, instead of creating monolithic applications resident in one single node. In literature [Peltz, 2003] [Ross-Talbot, 2005] [Quinton et al., 2009], two ways to combine services are distinguished: orchestration and choreography.

Orchestration refers to an executable process that can interact with both internal and external services to fulfill its task. The difference with choreography relies in the agent or agents that control the composition. Since Orchestration is done always from one party's perspective, it differs from choreography, which is more collaborative and addresses the interactions that implement the collaboration among services. Choreography tracks the message sequences among multiple agents rather than a specific process that a single party executes.

As stated in [Papazoglou et al., 2007] service compositions must understand and respect one another's policies, performance levels, security requirements, service-level agreement (SLA) stipulations, and so on. That exemplifies the close relationship between optimal compositions and context-awareness. In fact, one of the motivations of this research is to accomplish a task the most efficiently possible by assembling services from heterogeneous devices. Some points of context-awareness are also considered in the following revision of the state of the art and research proposal.

Taking this into account, the purpose of this research is to define a composition methodology and an orchestration agent that given a set of atomic services, controls the service composition not limited to application level, permitting to form a more complex application and network service. In order to do that, it will base its decision process in maximize a defined cost function that should reflect as many context parameters as possible to create dynamic, efficient and context-aware services. Context parameters considered mainly include user preferences, requirements on the requesting services, network constraints and service's Quality of Service (QoS). Notice also that, although the service composition will involve the interaction of other nodes that return their service's capabilities, it could not be defined as a choreography because the decision making process is done only by one agent in the requester node. Hence, it is considered out of the scope of this thesis, although it can be considered for future work.

In the resulting networked control system, different controllers and filters could also be seen as services that can be shared and updated on-line. Indeed, it should help to deploy distributed multimedia systems in which the different services correspond to filters and encoders/decoders possibly with different levels of QoS.

Notice that this research wants to take special relevance for FN architectures, where it can be used in the development of distributed real-time embedded systems, and permits the allocation of network functions/services according to each situation and not in a monolithic way. Thus, services must be allocated all along the route, executing just the desired service at each hop, section of hops or end-to-end. Hence, this research should help to pave the way to highly flexible networks, by efficiently applying service-oriented approach to networking, resources and services.

Furthermore, it should also permit a user empowerment in service choice and routing, so a service requester must be able to choose from matching service responders which specific service wants to consume.

However, there are many approaches or visions of doing service composition processes, but there are some points to take into account and some common problems in most of the techniques applied.

- **Fulfilling preconditions:** A service that can provide the desired effects exists; however, sometimes not all of this service's preconditions are met from the outset.
- **Generating multiple effects:** Sometimes a service query is translated to multiple effects that can be generated by different services.
- **Overcoming a lack of knowledge:** Additional knowledge gathering is necessary in order to fulfill correctly a request.

3.2 Service Oriented Architecture (SOA) paradigm

Service composition is a fundamental part of Service Oriented Computing [Huhns et al., 2005] and is a process that plays a key role in service-oriented architectures (SOA) [Papazoglou et al., 2007]. SOA paradigm proposes to organize and to use distributed capabilities operating under the control of different ownership domains. Services are the essential building blocks of SOA, and it is based on three premises:

1. Use of self-contained services and data
2. Use explicit descriptions instead of implicit assumptions
3. Use well defined interfaces between services

This paradigm can be related to Component Based Computing (CBC) design [Heineman et al., 2001] [Bramley et al., 2000], based on the decomposition of a problem into several pieces called components. Then, these components can be composed in order to fulfill specific problems and covering the requirements needed at specific times. This approach gives important benefits in terms of flexibility, scalability and adaptability, as the components are instantiated as needed.

In addition, SOA defines three basic elements: Service User, Service Broker and Service Provider. The latter two create a service oriented communication system, while the Service User access to the service by means of a User Application. To allow the integration and support of different protocols and formats, an additional element, such a Service Adapter can be placed in order to make the corresponding mappings [Zdun et al., 2006].

The benefits of this type of architectures are clear, permitting:

- Loose coupling between components
- Flexible configurability
- Task Distribution
- Granularity
- Implementation neutrality
- Persistence

As specified in [Huhns et al., 2005], a SOA system implies the process of discovering and composing the proper services to satisfy a specification, whether it is expressed in terms of a goal graph, a workflow, or some other hybrid model (Figure 3.1).

3.3 Services

Services are the basic or fundamental blocks or components where service composition process is established. Although they are defined in multiple languages and multiple representations [Mennie et al., 2000] [Lalanda et al., 2007] [Ankolekar et al., 2002] [Meyer et al., 2005], all of them maintain certain premises that services expressed by the following definition: *Services are self-contained, self-describing, modular applications that can be published, located, and invoked across the net.*

For example, in [Kumar et al., 2010] resources are abstracted as services to applications, presenting a framework (named ROSA –Relationship Oriented Service Architecture–) that works with a set of services (shown in Figure 3.2). ROSA tries to take a similar principle and framework to Web-Services, but it is applied to future heterogeneous communication networks in order to interconnect business borders and administrative domains. The set of services that it defines can inspire the definition of services for FN communication services.

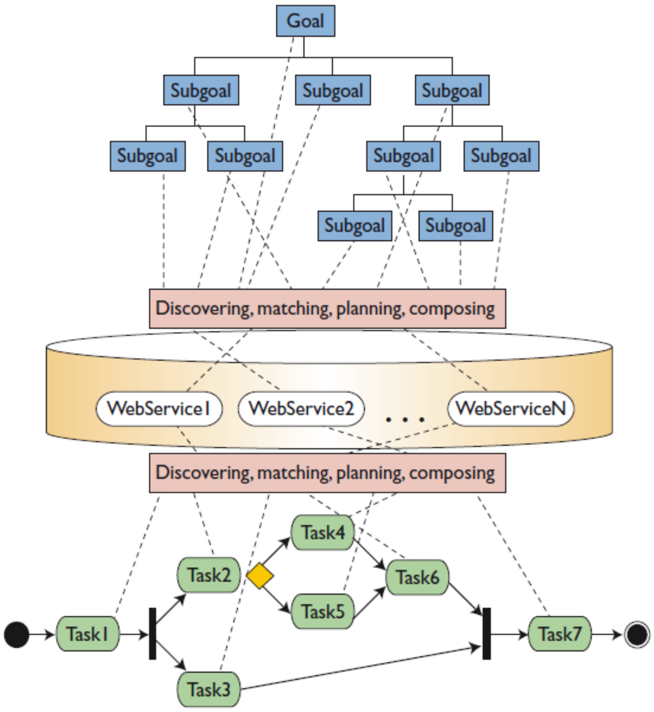


Figure 3.1. Top-down SOA service building model [Huhns et al., 2005]

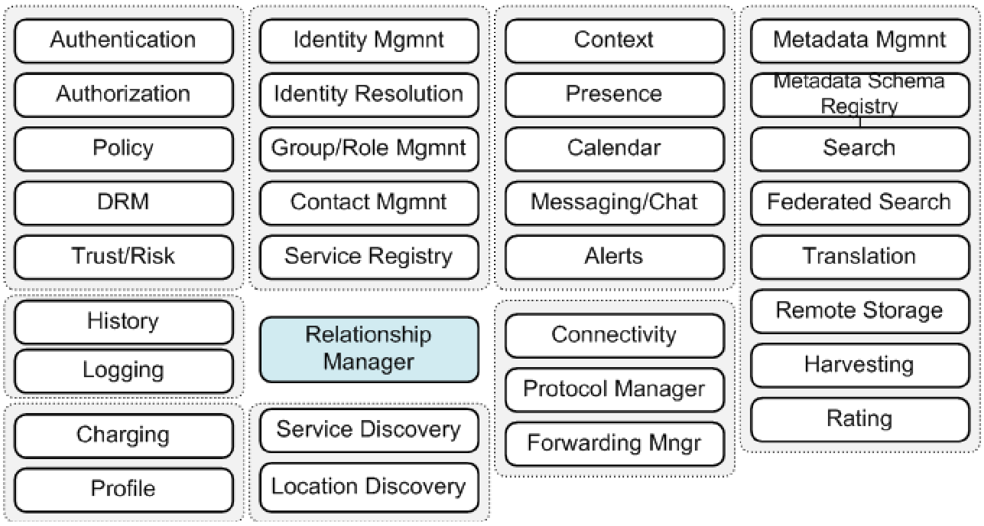


Figure 3.2 Set of services presented in ROSA [Kumar et al., 2010]

Alternatively, [Meyer et al., 2005] defines services as self-contained functions, which accept requests and return responses through a well-defined interface. They use logical language the F-Logic dialect Flora-2/XSB [Yang et al., 2003] to represent them, as shown on Figure 3.3. The language semantics and especially the structure in the definition of the services represents very elaborated example of how to define services and was taken as a basis for the services representation in the service composition methodology applied in the thesis (especially on the specification of preconditions, effects, properties, input and output parameters).

```
fpn:atomicService[
  spec     -> fpnSpec:semanticServiceSpecification[
              conditions ->> fpnCond:condition[
              precondR -> ${N:string, N:parameter, X:person, X[name->N]}:reification,
              precondS ->  "N:string, N:parameter, X:person, X[name->N]":string,
              posEffR  -> ${P:phoneNumber, P:parameter, X[phoneNumber->P]}:reification,
              posEffS  ->  "P:phoneNumber, P:parameter, X[phoneNumber->P]":string]],
  grounding -> fpnBridge:serviceGroundingSpecification[
                 serviceImplRef -> "fpnRef":string,
                 operationName  -> "doIt":string,
                 inParamSeq     ->> {_#:oSP[ord -> 1, str -> "N":string]},
                 outParamSeq    ->> {_#:oSP[ord -> 1, str -> "P":string]}],
  properties -> fpnProps::minServProps[
                 serviceName  *=> fpnSNType:enumeration[type -> string,
                                        values ->> {"findPhoneNumber":string}],
                 providerName *=> fpnPNType:enumeration[type -> string,
                                        values ->> {"HPI":string}],
                 cost         *=> zeroCostType,
                 payment      *=> noPaymentType]].
```

Figure 3.3 Service representation example using Flora-2/XSB [Meyer et al., 2005]

[Mennie et al., 2000] presents another approach trying to aboard dynamic software composition within a specific service domain, using existing technologies and without the need for a complex compositional language. In this framework, atomic services (Figure 3.4) collaborate to form new composite services. They define also these composite services (Figure 3.5), presenting a new common interface that must be constructed at runtime which allows other services to interact with this set of collaborating service components as if it was a single service. It simplifies the specification used by [Meyer et al., 2005] and provides a much lighter definition. It is more limited, but the XML syntax represents a way easier to integrate and much more deployable in communications. If we consider that, the information may be sent from one node to another the information should be the necessary, but minimized to not overload the network and speed up the service discovery and negotiation processes.

```
<?XML version="1.0" ?>
<SERVICE>
<DESCRIPTION>
      <NAME>Call Forward Unconditional</NAME>
      <VENDOR>Carleton University</VENDOR>
      <VERSION>1.3.2</VERSION>
      <PROTOCOL>H.323</PROTOCOL>
</DESCRIPTION>
<PROPERTIES>
      <COMPOSABLE>Yes</COMPOSABLE>
         <INPUTS>Caller, Call Agent</INPUTS>
         <OUTPUTS>Callee, Gatekeeper</OUTPUTS>
      <CHAINING_ORDER>First</CHAINING_ORDER>
      <COMPOSABLE_METHODS>Forward, Log Call Info</COMPOSABLE_METHODS>
</PROPERTIES>
</SERVICE>
```

Figure 3.4 Atomic Service Representation example [Mennie et al., 2000]

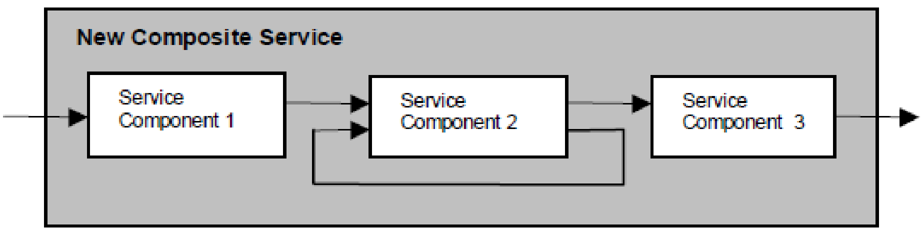


Figure 3.5 Composite Service Representation example [Mennie et al., 2000]

All the above-mentioned approaches represent a part of the first research on dynamic composition of network communications. However, service composition was especially developed in web-services environment and it is relevant to consider and analyze it in relation with the work presented in this thesis. Although web-services operate at the top of current Internet stack, they apply the core concepts that could be borrowed in this research. In order to limit the scope, it will be reviewed from the service representation and composition perspective, aspects that can be related to a generalized service composition.

3.3.1 Web services

One of the environments where Service Oriented Architecture (SOA) is widely deployed is in web services. Web Services (WS-*) specifications, standardization and its implementations represent the best-known SOA environment [Preist, 2004]. Indeed, Web Service composition is an extensive subject in the literature. Although they are often related to higher-level services, it is worth it to take a look at their methodology and mechanisms and how they could be generalized or inspire the composition process on dynamic network communication field.

Some requirements introduced by Web Services are the following ones:

- Representation of an abstract Web Process: To represent or specifying the abstract process in a proper form,
- Discovery and Interoperability of Services: To manually or automatically search for appropriate services and the discovered services should interoperate,
- Efficiency of a Composed Web Process: To compose processes which are efficient in terms of performance,
- Process Execution: To adopt a suitable technique for executing the composed concrete process,
- Process Monitoring: To use a monitoring technique for run time analysis of the Web process execution.

Web services are service entities that can be advertised, located and used through Internet. They are described using WSDL (Web Services Description Language) [Christensen, et al. 2001] and they operate jointly using mainly SOAP (Simple Object Access Protocol) [Box et al., 2000] and UDDI (Universal Description, Discovery, and Integration) [Bellwood et al., 2002] standards. They encapsulate application functionality and information resources, and make them available through programmatic interfaces, as opposed to the interfaces typically provided by traditional web applications, which are intended for manual interactions.

There are two different approaches for web-services composition: (1) Static Composition, in which the services to be composed are decided at design time, and (2) Dynamic Composition, in which are decided at run-time. It is also extensible to network communications services composition.

Although there is not a single standard, some languages for web service composition are already defined, such as BPEL4WS (Business Process Execution Language for Web Services), WSFL (Web Services Flow Language), XLANG (BizTalk), BPML (Business Process Modeling

Language) or ebXMl BPSS (Business Process Specification Schema) [Van der Aalst et al., 2003]. However, these languages are usually used to compose service manually at design time. Currently, Web services are usually described using WSDL descriptions, which provide operational information. WSDL descriptions do not contain (or at least explicate) semantic description; they do specify the structure of message components using XML schema constructs. For that reason, there are many projects (revised in next section) that extend this representation in elements and attributes, using DAML+OIL [Horrocks, 2002], OWL-S [Martin et al. 2007], or Web Service Modeling Ontology (WSMO) ontologies [De Brujin et al., 2005]. The use of ontologies permits to represent Web service descriptions in a machine-interpretable form like DAML-S [Ankolekar et al., 2002].

Moreover, composition approaches that have been seen in this field of literature, can show many available methods for composing services, such as scripting and coordination languages [Gelernter et al., 1992], rule-based systems [Ponnekanti et al., 2002], planning [Wu et al., 2006], situation calculus [Beradi et al., 2003], data view integration [Thakkar et al., 2003] and integer programming [Kumar et al., 2008], to name a few. They vary in their ability to represent and model non-functional properties of the service, to verify the correctness of the composite service, and to automate the process of service composition.

3.3.1.1 Composition approaches

In [Kumar et al., 2008] authors propose a classification system (TCS, Taxonomical Classification System) in the form of taxonomy for semantic web service composition approaches, that can be generalized to the global concept of service composition.

It provides a classification providing four levels (except workflow-based node) of hierarchy in classification taxonomy as shown in Figure 3.6. The focus of classification at first level is the amount of user involvement in the composition process. It provides three nodes Manual/User-Defined, Semi-Automatic, and Automatic.

The second level, which is based on the procedure or steps followed in the composition process, creates template and instance based categories under user-defined node while Workflow, Template/Instance, Declarative, AI Planning [Vukovic et al., 2007], Ontology based, and Hybrid categories under Automatic node [Alamri et al., 2006]. The third level of taxonomy is based on the amount of dynamicity involved in the composition process and the technology used to implement the procedure adopted in second level. It creates dynamic and static nodes under each of template based, instance based, and workflow based nodes of second level; chaining, logic and rule based under AI planning based and context and agent based under ontology based node of second level [Vukovic et al., 2007]. The fourth level of taxonomy produces different categories as shown in Figure 3.6 and is focused on some variations possible in the technology adopted in the third level. The fifth level of taxonomy in workflow-based node is based on the centralization level in the execution process.

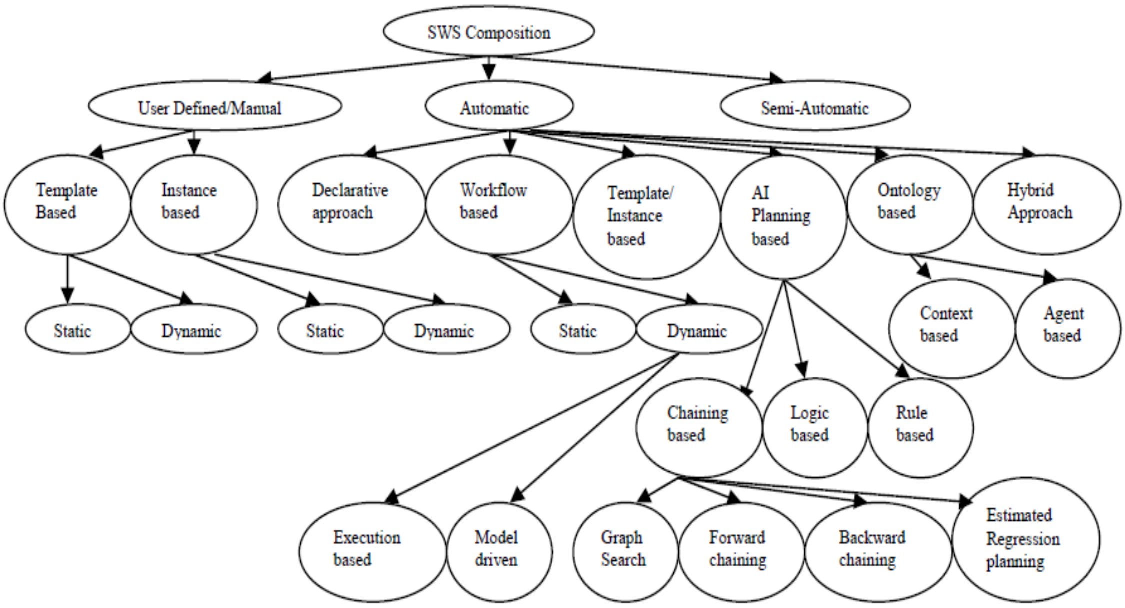


Figure 3.6 Service Composition Methods Taxonomy [Kumar et al., 2008]

The approach taken later in this thesis is a mix of automatic template-based and declarative approaches. It was designed considering the requirements and goal in which the service composition process was applied, especially trying to avoid complex and highly demanding compositions to speed up the communications establishment or the application deployment. However, other approaches would be more appropriated for no delay-critical compositions and is important to review them for other future use cases and contexts. The following deepen a little more on the different approaches that can be selected.

A. User Defined/Manual approaches

These approaches do not plan from scratch like other approaches. They are manually configured or described. The most used methods in this area are template or instance-based compositions, which use recommendations methods to choose a specific pre-designed composition among all available templates or instances. These approaches do not reach the flexibility level aimed in this research, only being relevant because they put the basis of other semi-automatic and automatic approaches (like workflow or template/instance ones) [Rao et al., 2004] [Bucchiarone et al., 2006].

B. Automatic approaches

The potential of automated services composition for the exploration of benefits of service oriented architectures is clear, although the term *automated* service composition is used differently by varying groups, making discussion and comparison not easy. Some of the different *automated* solutions are detailed in the following list:

Template-based and instance-based approaches

This approach usually takes outlines of permissible workflows (or templates) and instantiates one of them to obtain an executable workflow. This process can be based on a set of rules that covers policies, context events and constraints. Policies are specifications of behavior that the composition should have, usually described as Event-Condition-Action (ECA) formulations, while context parameters and constraints are used to describe conditions that affect these policies.

This method could be applied alone, where the initial template should be hand-coded or can work jointly with other automatic methods generalizing the workflows that these automatic approaches extract, in order to re-apply a known workflow for a new situation.

The policies can also guide plan adaptation to external events (e.g. changes in context constraints or the performance of the service) while a composition is being executed. The template-based approach works best for environments where compositions follow regular patterns (in general more static networks with infrastructure) and its adaptation is required to be along anticipated context.

Some approaches [Sivashanmugam et al., 2005] take this methodology along with templates of services' semantic descriptions, and propose a template-based searching mechanism to allow dynamic searching of business process partners at design or deployment time of a Web process.

Other methods [Su et al., 2003] introduce the notion of composition templates only to provide composability soundness; compositions are associated with each composite service and are used to compare values added by different compositions. To check whether a composition is sound or not, stored templates are used to store a composite service's template in the service repository. A service composition is considered as sound if its template is the subset of a stored template. They are used more in an evaluation process than really on the composition.

Declarative approach
In declarative composition, composite services are generated from a high-level declarative description. The technique uses composability rules to determine whether two services are composable [Dustdar et al., 2005]. Most of the time these rules act as constraints that must be satisfied in order to compose a service. The rules are used to generate composition plans that conform to a service requester's specifications. Techniques that fall under this classification usually tend to reach optimality of composition against some defined objectives (i.e., cost, time…etc.) as they are mathematically modeled. Mostly, the optimality can be achieved by mapping rules to constraints and trying to solve them using operation research methods [Channa et al., 2005]. Some examples of this methodology are explained next.

For example, FUSION is a software infrastructure system that provides the common infrastructure elements needed to support service portals [Van der Meer et al., 2003]. Given a user service specification, it automatically generates a correct and optimized execution plan, then executes this plan and verifies the result. The most important advantage of this system is its ability to generate an optimal execution plan automatically from the abstract requirements that a user may specify. In addition, this composition system verifies that the result of the execution plan meets the user's requirements, and if not, it immediately recovers this execution plan. However, this system creates a bundle of services and uses it to specify an execution plan by choosing services from this bundle. Web services are evolutionarily increasing and determining which subset of these web services to use will limit the choices for the user and will not guarantee the optimality of the result. Also, there will be a probability of using web services that no longer exist.

SELF-SERV [Benatallah et al., 2002] is a framework for dynamic and peer-to-peer provisioning of web services. In SELF-SERV, web services are declaratively composed, and the resulting composite services are executed in a decentralized way within a dynamic environment. The framework uses and adapts the state-charts as a visual declarative language. The significant advantage of SELF-SERV is the peer-to-peer service execution model, whereby the responsibility of coordinating the execution of a composite service is distributed across several peer software components called coordinators. Nevertheless, this system does not provide a method to create a composition at runtime for services. Also, it does not consider any semantics of web services during composition decisions. Moreover, this technique imposes some unrealistic requirements that should be implemented by service providers from the partial point-of-view. Therefore, although the declarative language proposal seems interesting to be investigated in future work, it only offers design-time composition not suitable for the context-aware services that we target.

In this sense, another significant work was performed in [Channa et al., 2005] to reduce the complexity and time needed to generate and execute a composition and improve its efficiency by selecting the optimal services at the current time. This research proposed an architecture of dynamic web service composition by runtime searching of registries to find services. Therefore, this technique does not use any service template. Moreover, it reduces the dynamic composition of the web services to a constraint satisfaction problem where any linear programming solver can be used to solve it. Besides, it assures the optimality of the web services selection based on domain specific QoS parameters identified by the user. However, this approach does not support user interactive participation in the composition process, which is sometimes important to ensure the user's satisfaction.

enTish [Ambroszkiewicz, 2003] is somewhat different from typical composition platforms. Services are typically created on the fly to realize client requests, which is also one of the objectives of this thesis in network communications service composition. Most frameworks are based on the assumption that first the business process has to be created. For enTish, a different architecture is needed, since client requests are expressed in a declarative way using formal languages. The declarative approach consists of two phases: the first phase takes an initial situation and the desired goal as starting point and constructs generic plans to reach the goal. The latter one chooses one generic plan, discovers appropriate services and builds a workflow out of them. This methodology partly inspired the overall composition process design proposed below in Chapter 5.

Ontology based

An ontology [Gruber, 1995] is a collection of services that share the same domain of interest. In contrast to manually describing service composition, there is also a lot of effort done in the direction of ontology based web service composition. This technique facilitates the semantic dynamic composition of web services. The ontological descriptions and relationships among web services are used to automatically (or semi-automatically) compose services. Ontology-driven techniques compose the services based on the goal-oriented inferring and planning [Küester et al., 2005]. The fact is that web service ontologies are going very used and extended, and they are too large to be used in a single application, and this has stimulated many researchers.

For example, DAML-S (DARPA Agent Markup Language for Web services) is a semantic markup language for describing web services and related ontologies [Ankolekar et al., 2002]. It presented an interesting classification of services, distinguishing between atomic, simple, and composite processes:

1. Atomic processes are directly invokable (by exchanging messages with the service), have no subprocesses, and execute in a single step, from the perspective of the service requester. That is, the requester sends a single message, and receives back a single message, in making use of the service. Atomic processes must provide a grounding that enables a service requester to construct an invocation message and interpret a response message.
2. Simple processes, on the other hand, are not directly invokable and are not associated with a grounding. Like atomic processes, they can be conceived as having single-step executions. Simple processes are used as elements of abstract processes; a simple process may be used either to provide a view of (a specialized way of using) some atomic process, or a simplified representation of some composite process (for purposes of planning and reasoning).
3. Composite processes are decomposable into other (non-composite or composite) processes. Their decompositions are specified using control constructs such as Sequence and If-Then-Else. Decompositions show, among other things, the control structure associated with a composition of processes and the input-output dataflow of the composition.

As well as DAML-S, WSOL [Tosic et al., 2002] dealt with the need for the requirements for ontologies. Both defined a list of some proposed ontologies needed for the management of Web services that can be summarized as:

- Ontology of QoS metrics
- Ontology of measurement units
- Ontology of currency units
- Ontology of measured properties
- Ontology of measurement methods

However, DAML-S and WSOL and other RDFs based languages are currently superseded by OWL-S [Martin et al. 2007], an ontology build on top of OWL (the Web Ontology Language) [McGuinness et al., 2004], that enable users and software agents to automatically discover, invoke, compose, and monitor web resources offering services, under specified constraints.

Most of the ontology-driven techniques mark-up web service descriptions with ontologies and develop algorithms to match and annotate WSDL files with relevant ontologies. The possible

compositions are obtained by checking the semantic similarities between interfaces of individual services (semantic matching) and considering the service quality (QoS matching). Then, these compositions are ranked and presented according to these two dimensions.
The composition is usually performed based on understanding the semantics of interactions/capabilities of the elementary services [Mrissa et al., 2005]. To achieve semantic composition, these techniques mostly require a domain-specific ontology design that defines explicit formal specifications of the concepts and relationships among the concepts. It might also require an extraction module that helps us in building ontologies from service profiles. Notwithstanding, these techniques would require a long-term research on the definition of a complete service ontology that is lightweight enough to be supported for any communication service composition and any device. Therefore, they were initially discarded, and not deeply analyzed for the purpose of this thesis.

However, regarding to context representation in service ontologies, COBRA-ONT [Chen et al., 2003] exposed an ontology for context-aware pervasive computing environments represented in OWL. CONON [Wang et al., 2004] also presents a similar approach, presenting an OWL encoded context ontology for pervasive computing environments. Especially remarkable in this field is GAS Ontology [Christopoulou et al., 2005], an ontology that was developed in order to describe the semantics of the basic concepts of a ubiquitous computing environment and define their inter-relations. All of them were reviewed in order to acquire the knowledge of how to represent the context in the network communications composition process, although it can be helpful to be adapted also for other future work, in the Smart Grid field, for example.

Artificial Intelligence (AI) Planning

Planning is a problem solving technique where knowledge about actions and their consequences is used to identify a sequence of actions, which when applied in a given initial state, satisfy a desired goal [Russell et al., 2003].

A planner can receive three inputs mainly:

- Initial state: describes the starting state of the application domain. It is normally called world.
- Goal state: describes the desired world state.
- Domain description: describes actions that, when invoked, transform the world states.

The output of the planning process is a plan, a sequence of actions that can be executed in order to achieve a desired goal state.

The composition engine may be implemented by a number of different composition methodologies, as mentioned in section 3.3. Services can be represented in terms of their non-functional and functional properties. Non-functional properties describe service provider details and Quality of Service parameters. Functional properties contain descriptions of service operations in terms of inputs, outputs, preconditions and effects, which make it easy to translate them into planning actions.

Planning systems need semantics for describing the available actions, thus to describe the formalized domain [McIlraith et al., 2002]. The simplest form of domain formalism is based on state-transition models. It defines states to describe the context at a certain point in time, such as an initial state or goal state, and actions, which perform transitions between states. An action can be described with a list of preconditions, states and post conditions [Wu et al., 2006]. Planning technologies differ in the complexity of the problems they can handle and the representations that they use, employing different search algorithms to synthesize plans and the constraints they observe [Wu et al., 2006] [Ghallab et al., 2004].

However, the computational requirements for applying AI Planning techniques can lead to high computational costs for composing real-time delay-critical services in commodity hardware, or even not feasible to run in IoT or other type of constrained devices.

3.4 Context-Awareness

Other relevant point to consider in this research is the context processing. According to the definition provided by Dey in [Dey, 2001], context can be defined: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

Considering the discussions in [Chen et al., 2003] [Ranganathan et al., 2003] [Wang et al., 2004] context can be classified according to the following:

- The user context can include the user characteristics, the user location, the user preference, and the environmental constraint of the user (e.g. public are where silence is required, working place, home, etc.).
- The device context can include the type and the capability of the device.
- The service context can include service availability, required QoS level, and service performance.
- The system resource context can include CPU, memory, processor, disk, I/O devices, and storage.
- The network context can include bandwidth, traffic, topology, and network performance.

An example of the potential of processing the context related to service, network or resource, is further explained in Chapter 5.7, detailing how CPU, memory, processor or available bandwidth metrics were used in SHECloud [SHECloud] or FINESCE [FINESCE] projects to develop a reasoner that selects the optimal storage platform (among public or private clouds) [Briones et al., 2016].

In order to manipulate context data, it must be in a format compatible with the models that will be used in context data processing such as reasoning and situation recognition. These models can be object oriented, ontology-based, rule based, logic based, semantic-based or based on fuzzy logic sets.

Indeed, context-awareness becomes a key feature when considering service composition [Baker et al., 2009]. Context-awareness refers to the capability of an application or services to be aware of its physical environment or situation and responding proactively or reactively and intelligently based on such context. Therefore, it is important to compose services and dynamically adapt them according to the context information and changes in order to provide personalized and customized services to users. This will allow improving the Quality of Experience (QoE) of users while optimizing the usage of network and computational resources. This aspect is especially interesting as today the ubiquity of mobile devices and the proliferation of wireless networks will allow everyone permanent access to the Internet at all times and all places. It can be said that the next step to an Internet of Services is an Internet of context-aware services.

In the work done in this thesis, context have been considered in many aspects in order to automatically adapt services or solutions functionalities. Network, device/node or user context (requirements, capabilities, preferences, etc.) are extracted using different methodologies (such as extracting services metadata, reading configuration files or including user preferences in the service request) and used as elements to consider in the reasoning process of the service composition.

3.5 Service Composition in Future Networks projects

The research of this thesis considered a strong implication of FN architectures development in of the application of SOA concepts for composing context-aware communications and processes. Therefore, in order to complement this chapter focused on the state of the art, this section points out some relevant FN projects and initiatives that presented service-oriented proposals or mechanisms.

In recent years, multiple initiatives have contemplated restructuring the current Internet architecture in order to cope with its limitations, such as mobility or security functionalities that were not considered during Internet inception. Many of these solving strategies introduce novel clean-slate architectures that do not take TCP/IP as their groundwork.

In that sense, the interest of the scientific community to propose new solutions to current Internet architecture is being recently driven by standardization bodies such as ISO/IEC JTC1/SC6/WG7, defining a clean-slate approach [ISO/IEC TR 29181] or ITU-T SG13, which adopts a more evolutionary perspective [Matsubara et al. 2013].

Some projects in the USA (NSF GENI/FIND [GENI] [FIND]), EU (4WARD [4WARD]) and Japan (AKARI Project [Harai, 2009]) have been issued to develop new network solutions from scratch. These clean-slate proposals share some common concepts, like micro-modularization[12] and virtualization[13] as a means to support multiple architectures simultaneously, in their design and objective; although differing in scope and development. Projects selected for the analysis

[12] To split the protocols and processes in their fundamental (atomic) functionalities.

[13] To abstract the real physical resources, processing them as a set of computation, storage, bandwidth capacity, etc.

in this section have focused on protocol functionality organization to try to solve current limitations. All of them share the goal of doing this organization dynamically whilst enabling the adaptation of communications to application requirements and network properties. However, communication services depend on limitations given by the infrastructure, physical available resources and specific policies within a domain. They must interact with policy enforcement points inside the network like firewalls and may use services that reside somewhere in the network to establish communications (e.g. lookup service used during session setup) or to adapt/enrich services (e.g. use of transcoding services to modify data-streams). In today's Internet, an edge does not interact and dialogue with all these entities inside the network (edges, users, providers, services, etc.) and, consequently, cannot provide communications adapted to specific requester and network demands and requirements at all levels.

Consequently, a communication service usually implies several additional mechanisms or processes whereby former proposals usually focus on configuring processes locally a global configuration of all the entities that participate in a specific communication. Former “Future Network” research projects consider “how to organize protocols”, and thus consider data processing but not these mechanisms or processes. These missing processes typically are: to negotiate services provided by the network under specific conditions (e.g. context and its changes), to negotiate with middleboxes and to call services in the network according to user needs and application goals and requirements.

Protocols are basically means of data exchange for a communication between nodes or systems. In order to understand each other, the concerned communication parties must agree on syntax and semantic of exchanged data units. Syntax means the data unit’s structure and the fields it contains. Semantic means the meaning of each field of the data unit. Syntax and semantic make one protocol different from another and create a variety of network protocols. Current data units are usually composed of payload and header. The header in turn is composed by fields carrying control information such as network address, data checksum or congestion notification.

The projects reviewed in this section change drastically this structure. They take in a way or another SOA paradigm as the base of their architectures; define the composition blocks as services [NETSERV], netlets [Völker et al., 2008] or functional/building blocks [Bouabene et al., 2010]. All of them try to tackle the composition of this blocks in more complex workflows in order to provide network and application services. For example, based on the header structure presented in Role-based Architecture (RBA) [Braden et al., 2003], they propose a method to design new protocol and dynamically build the protocol header by functionality composition.

In this context, the rest of this section outlines these approaches and their architecture similarities to TARIFA (The Atomic Redesign of the Internet Future Architecture) project approach [Sallent et al., 2010] (further explained in Chapter 4) in which the research of this thesis was initiated. It will help to understand how influenced the previous work on these projects to TARIFA architecture and to the subsequent work presented in this thesis.

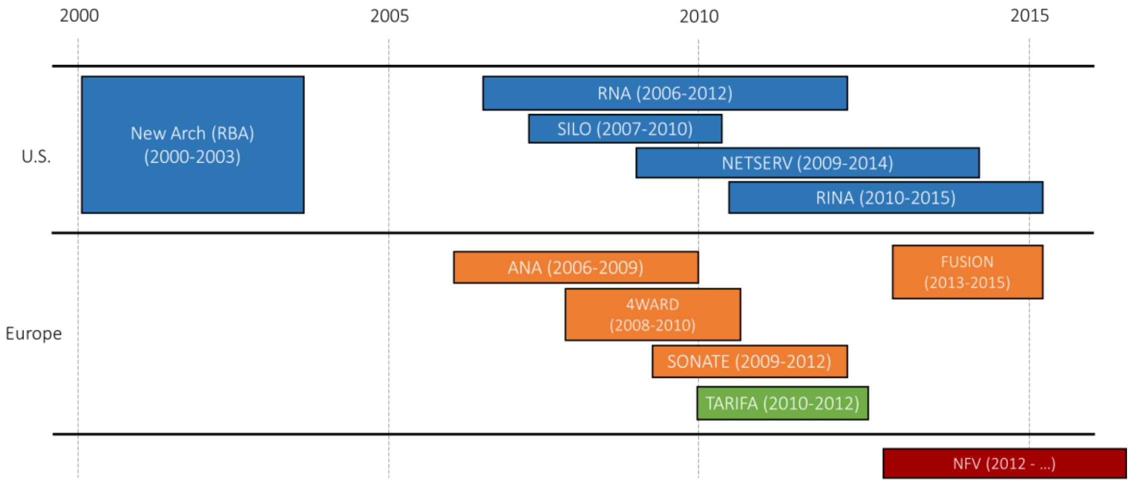
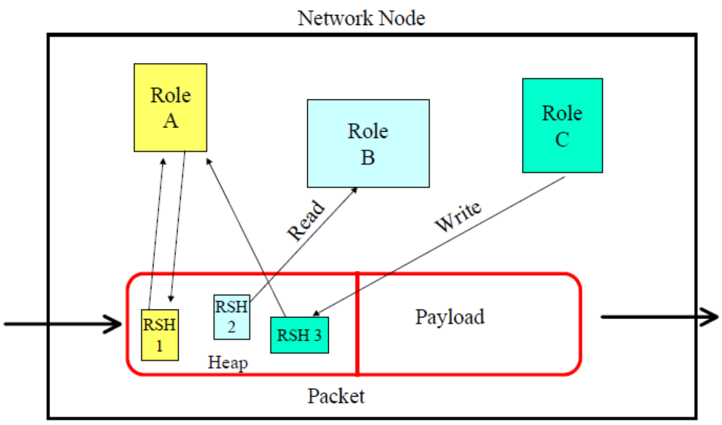


Figure 3.7. Chronology of the analyzed FN service-oriented projects and initiatives

3.5.1 RBA

Role-Based Architecture (RBA) [Braden et al., 2003] is an abstract approach to a non-layered architecture that was an output of the MIT's New Arch project [Clark et al., 2003]. The RBA organizes communication using functional units called roles. Every role has its own roleID, which represents its functionality and they are not organized hierarchically as traditional layer based communication system. Instead, the roles are organized in a heap that does not defines dependencies or restrictions, thus provides more flexibility to the interconnection of functionalities. Roles can reside at single or multiple nodes (i.e. distributed roles). A network packet contains metadata called role data, which is divided into chunks called role-specific headers (RSHs) (see Figure 3.8). To avoid strict order of functionality, RBA introduces dynamic order of role headers in a packet header. The RSH has information about inputs and outputs of a certain role. A role is addressed with its id and location such as roleID@nodeID, if role is distributed over multiple nodes then it is represented with notation as roleID@* so that every node at the path will process the role. To give further flexibility, along the path roles can be added, deleted or modified.



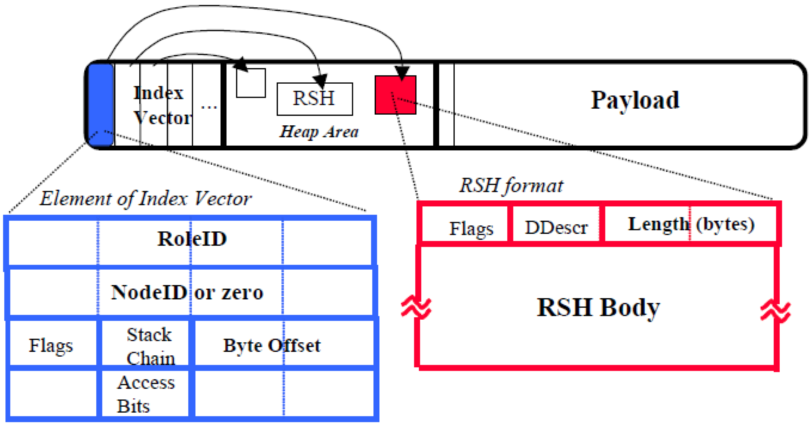


Figure 3.8. RBA protocol Heap Specification [Braden et al., 2003]

RBA defines standardized network services, which are called roles. The packet also carries the precedence information, as the node must decide in what order the roles must be executed. Roles can be composed into an aggregate role; these roles are directly bounded instead of using shared data in RSH. It is emphasized in the approach that RBA is a generic model, in the sense that it can be applied on top of any layer, whether all network functionality is split into roles or only the functionality above the Link, IP or even application layer is decomposed, depends on the realization.

It meant the first initiative that proposed the protocol micro-modularization approach, which inspired the rest of the proposals. It stated the grounds in which the other projects (and this thesis) were defined. However, its work was not deployed or implemented in a real scenario, remaining as a very inspirational academic work, but not furtherly explored. In any case, TARIFA architecture took RBA as an example in the way it modularized protocols into roles or services.

3.5.2 SILO

The NSF project SILO [SILO] provides a solution to separate control from data functions to enable transparent cross-layer interactions. Opting for a service-oriented/role-based approach, SILO uses basic re-usable fine-grained services to build custom protocol stacks (called silos) per traffic flow [Dutta et al., 2007] (Figure 3.9). This approach represents a middle ground between the strict protocol stack imposed by current architectures and the “heap” approach advocated by the RBA. Services are defined by describing its interfaces to other services, its function and tuning parameters (aka knobs) for adapting service execution to application requirements. Similar to other service-oriented solutions, each service may present different methods or implementation mechanisms to carry out the service execution. Regarding the composition of silos, the project defines a simple ontology for specifying dependences and constraints between services [Vellala et al., 2007], while a control agent performs the composition by executing a simplistic recursive algorithm based on pre-established ordering precedence of services.

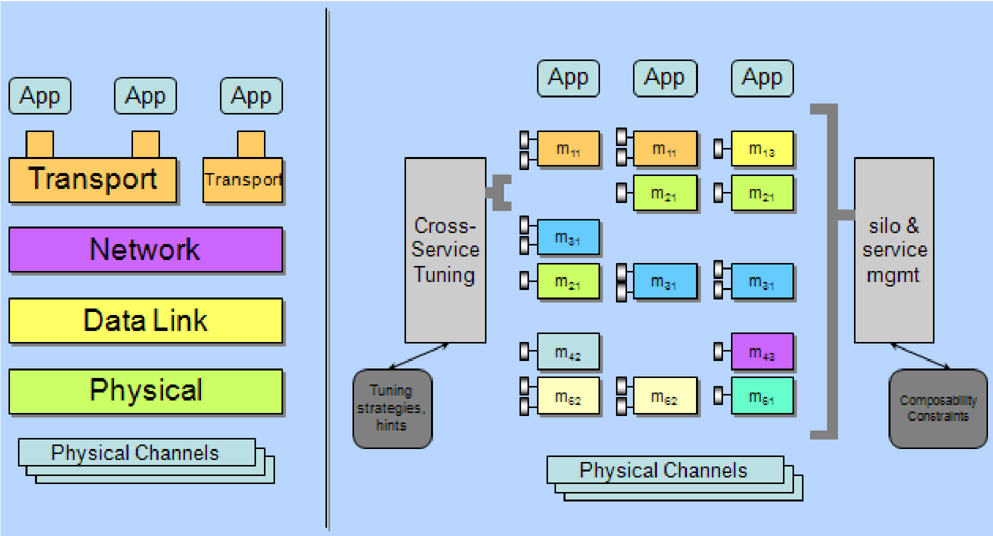


Figure 3.9. Example of protocol silos created in SILO project [Dutta et al., 2007]

SILO does not clearly define the set of services that should be available to create layers and silos, but assumes that they cover all those functions used by current protocol at the different layers. Besides, SILO does not cover context description, but actually silo composition should be based on context information, as such, mechanisms for describing and monitoring network context will always be a constructive addendum.

3.5.3 RNA

The NSF FIND project Recursive Network Architecture (RNA) [RNA] provides a solution to generalize layering solutions. RNA introduces a generic protocol meta-layer offering different basic services, tuned to support a concrete context. In RNA each protocol stack instantiates as many layers (each one with arbitrary different configurations) as required to fulfil its functional objectives and, hence, providing flexibility in the building of the stack [Touch et al., 2006]. This way, new functionalities can be added to the stack by just instantiating new layers and configuring them accordingly (Figure 3.10). In order to implement the required dynamism, flexibility and adaptability during stack composition, RNA provides a simple protocol for negotiating the stack instances amongst the different nodes in a given communication [Touch et al., 2008].

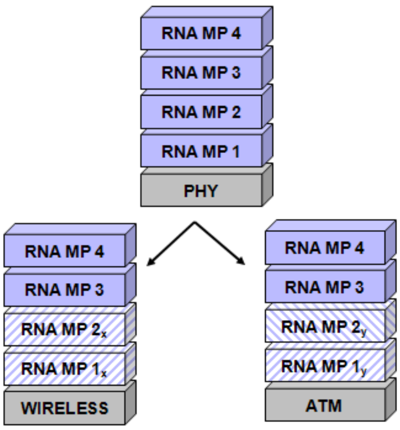


Figure 3.10. Customization of the RNA metaprotocol stack [Touch et al., 2006]

The approach of RNA is having a generic layer composed of instances of basic services. In this approach, role abstraction can be done at two different granularities, layer and basic service. Some of the types of basic services covered by RNA meta-layers are:

- handshake / state management.
- security policy (admission control, filtering).
- multiplexing and demultiplexing.
- retransmission.
- reordering.
- pacing / congestion control.
- switching / forwarding.

Although RNA does not cover context description, it is assumed that meta-layer instancing and configuration will be context-dependent; as such, mechanisms for describing and monitoring network context will always be a constructive addendum.

3.5.4 4WARD

One of the most influential FN visions was provided by 4WARD [4WARD]. It meant the main commitment to the future of the European Commission, investing 14.45 million Euros in this large FP7 project that does not define a unique approach for FN, but multiple approaches in order to tackle different network challenges.

One of the approaches was focused on the dynamical composition of communications. The microscopic point of view introduced in 4WARD tries to tackle a close problem of this research, composing basic functionalities or atomic services into more complex services. The goal of the composition is to build new protocols fulfilling the requirements of each communication.
Mainly, the composition of functionality proposed is based on chaining some Functional Blocks (FBs) which are implemented by specific Mechanisms. 4WARD is based on a Component Based Architecture (CBA), where the decomposition of the engineered systems into functional or logical blocks with well-defined interfaces used for communication across these components. Components are considered to be a higher-level abstraction than objects and as such they do not share state and communicate by exchanging messages carrying data.

In 4WARD, the dynamic behavior is provided by a set of policies specified by each FB, which are evaluated at run-time. An example of their appliance would be when it is necessary to choose a polynomial to calculate the CRC in the presence of an alarm or specific event. It organizes FBs in different strata (Horizontal, Vertical and Abstract). In addition, 4WARD was able to compose context-aware services thanks to the concatenation and execution of Building Blocks, which perform a specific action. They are contained and specified in a Netlet (see Figure 3.11). The project proposed to use a repository of well-known or best practices to effectively compose different types of functionalities, it is called Design Repository.

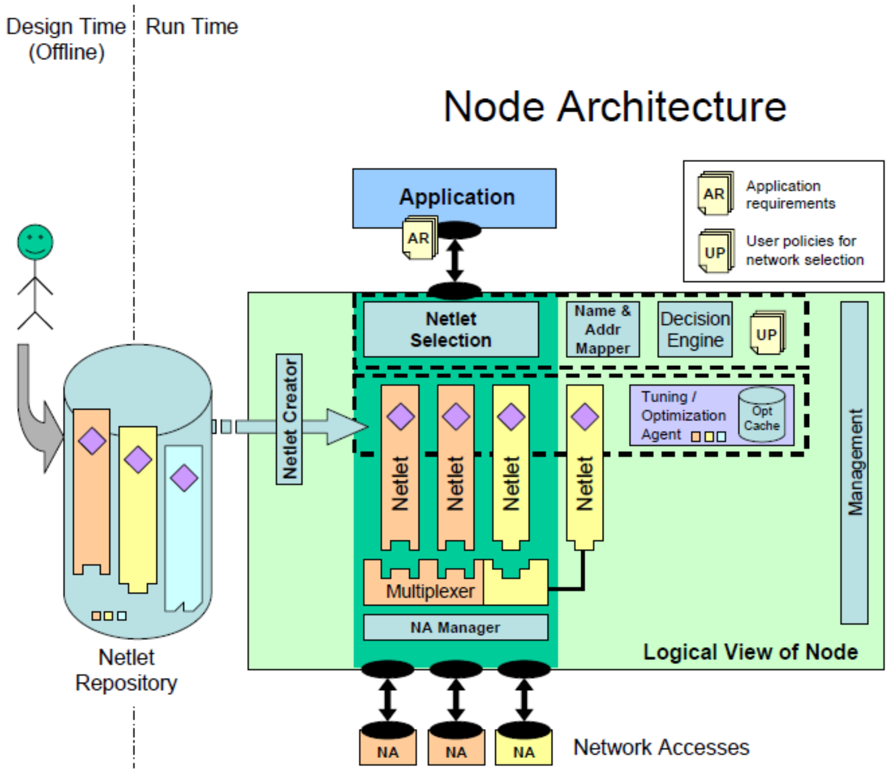


Figure 3.11. 4WARD (Netlet) Node Architecture [Völker et al., 2008]

In order to cope with the high complexity of communication systems, 4WARD provides two different views on network architectures: the macroscopic view and the microscopic view. The first one is more related to an overall structuring of the network architecture at a rather high level of abstraction in terms of strata. The second one deals with the functionalities needed within network architectures, their composition to so-called Netlets in order to fulfill desired requirements as well as the Node Architecture hosting various Netlets of the same or different families of network architectures.

In 4WARD, functionality is represented as a set of protocols and FBs that contain mechanisms and policies. Some of the network functionalities defined by the project were *Data Transfer, Traffic Control, QoS, Mobility, Security or Naming and Addressing*, each of them containing different functional blocks such as *forwarding, sending and receiving data, error control and retransmission, priority queuing, encryption,* etc. Defined that way, functional blocks can belong to different functionalities and each functionality may contain diverse functional blocks. The composition process is based on the data inputs and outputs of each functional block, building a chain of functional blocks that are giving some communications functionalities.

Following an approach inspired in RBA, a protocol in 4WARD implements a set of abstract functionalities and is composed by one or many FBs. The header is built based on the functionalities needed by composing the corresponding FBs, and finally associated with the corresponding payload to form the whole PDU of the protocol. The separation between the functionality and the functional block can provide the protocol design and header construction with a high flexibility similar to the concept of polymorphism that exists in software engineering. As an example, many addressing schemes may be used to implement the addressing functionality and each provides an address value, which can be transported by the header. When switching the address scheme, the address format and value in the header change accordingly. One could achieve this by implementing the addressing scheme as Functional Block Policy.

Regarding the communication related functionalities, different FBs can implement a needed functionality. The protocol specification and the header format will change accordingly to the functionality composition in order to get the required protocol.

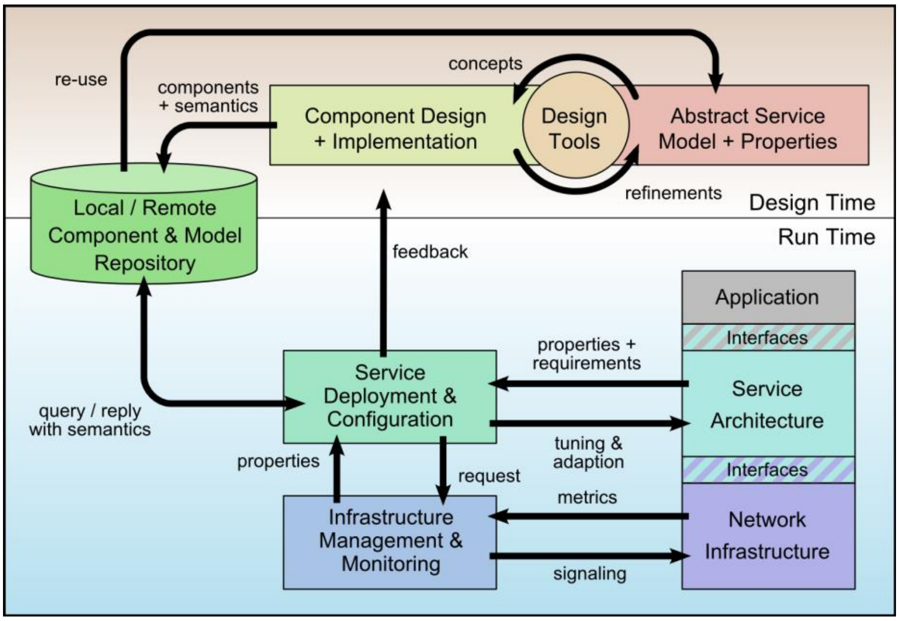


Figure 3.12. Life cycle for composed network services [Völker et al., 2009]

The selection algorithm of 4WARD compares different candidates and chooses the optimal one, based on user or application requirements (see Figure 3.12). A candidate is the combination of TCP/IP and one or more security protocols as wells as a configuration of that protocol. For the node architecture, these candidates are more flexible, they are Netlets that are possibly composed of functional blocks. When trying to find the optimum, it is important to find a trade-off between different properties of the candidates. Special care is needed for security; therefore, they have introduced a special security property called Effective Bit Strength (EBS), which will be also used for the new solution. The main idea behind EBS is that an encryption algorithm, for example, is only as strong as the easiest attack against it, even if that means attacking the key exchange. In the first step of the evaluation of candidates, they filter all candidates based on system policy or user/application requirements. This means candidates that cannot fit the requirements will not be considered later on.

Based on these properties, 4WARD uses a well-known decision-theoretic method, i.e., Multi-Attribute Utility Theory (MAUT) [Keeney et al., 1993], to aggregate the results of a candidate with respect to each single property into an overall ranking of each candidate. MAUT structures a complex decision process, which depends on multiple attributes, into a per-attribute utility evaluation.

3.5.5 SONATE

SONATE (Service Oriented Network Architecture) approach [Müller et al., 2007] is based on principles of service oriented architectures (SOA), foreseeing FN architecture as a large, and distributed (software) system, where a set of services communicate, cooperate, and inter-operate with each other. The goal of this service-oriented networking definition is to offer fine-grained functionalities to create applications.

It defined services/mechanisms as an abstraction of specific algorithms and data structures used to implement a functionality. Hence, each service provides a self-contained functionality and a well-defined interface that must not take assumptions about internals of other services in order to loose coupling among them.

According to their scope, the services are conditionally grouped into three service groups [Schwerdel et al., 2009]:

- ***Application Services.*** *The application services cloud offers services related to application functionality. Examples of such services are authentication, application notification service, application information exchange service, etc.*
- ***Mediation Services.*** *The scope of the mediation services cloud incorporates services related to network functionality. Typical example of mediation services are connection establishment and release, flow control and congestion control, etc. The presented approach focuses on this cloud.*
- ***Connection Services.*** *The connection services cloud encompasses services related to data transport. Such services are, for example, modifying network data, signaling, signaling error correction, etc.*

As a result of the cooperation and composition of these services, SONATE defines workflows, which provide more complex functionalities by selecting sets of services and defining their interaction. Thanks to the loose coupled definition of services and the abstraction of their functionalities SONATE tries to simplify the service composition and orchestration, permitting to add or remove services and change their implementation without affecting the whole workflow. For building a network based on principles of SOA, SONATE considers specific supporting techniques; since Web Services and XML are inappropriate to implement services on a network level, (other light-weighted semantics are needed). However, for primary implementations or proof-of-concept of the efficient behavior of their framework, they specify services and workflows using XML.

To build dynamically these micro-protocol stacks mechanism SONATE partners consider necessary to describe their capabilities and requirements in a way that an algorithm can predict the outcome of any combination. To achieve it they specify that each mechanism has a set of requirements, a set of provided effects, a description of the costs of the mechanism and a formula how these costs “add up” with the costs of other mechanisms. An algorithm selects mechanisms so that all requirements are met, all desired effects are provided and the combined costs are low.

One approach to do composition process is presented in [Schwerdel et al., 2009]. It bases the composition process on a network architecture that is described by a set of elements. The main groups are *Effects* (Figure 3.13) which are the desired functionalities needed in the communication, and the *Mechanisms* (working as FBs).

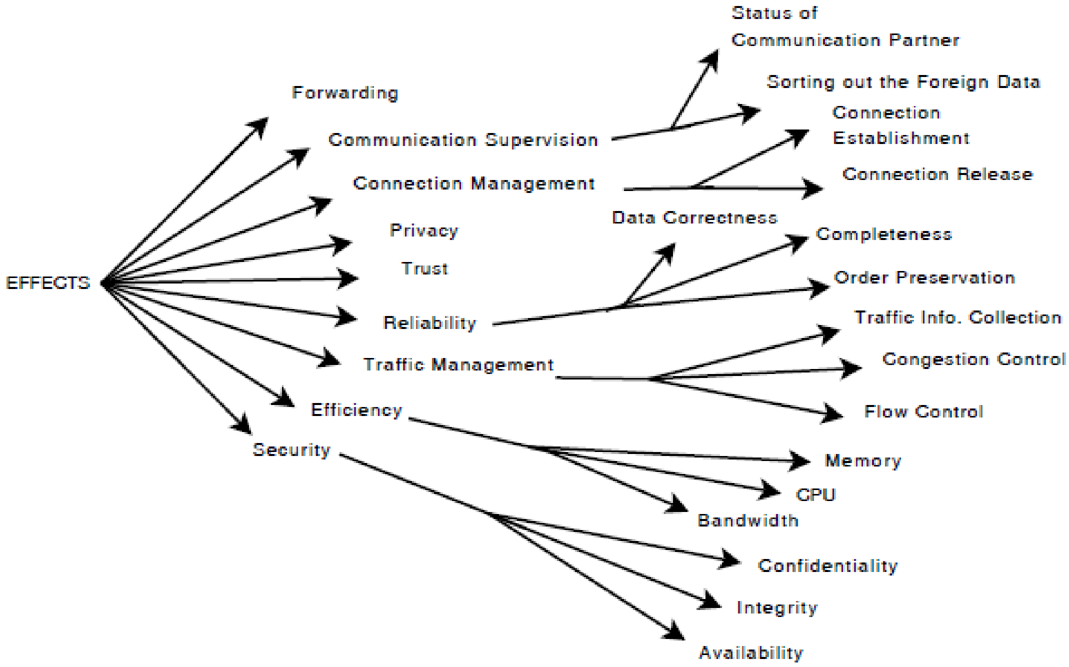


Figure 3.13. Effects defined in SONATE [Völker et al., 2009]

It also defines the subset of fixed *Mechamisms,* which represent those related to lower network layers with less flexibility. Two functions are also defined, *P*, a function that provides which effects provide each mechanism, and *R*, a function that specifies which other mechanisms a mechanism require. To compare different consistent sets of mechanisms a cost function *C*, is used, which combine the costs of a set of mechanisms. Since different types of costs "add up" differently (e.g. latencies sum up but loss ratios are inverse multiplicative) and the factors with which these types of costs are combined depend on the needs of the application layer.

In SONATE project, the approach presented for composition is a task of cost (*C*) optimization that entails the following process:

1. **Select mechanisms to provide the requested effects**. The base technology already provides some fixed mechanisms. This will yield a number of different premature solutions. At this step, the premature solutions are not checked for consistency, they provide the requested effects, but their requirements might be missing.
2. **Create consistent solutions and rate them**. Mechanisms are added to the premature solutions to make them consistent. After this step a number of consistent solutions is known and compared using the cost function.
3. **Improve solutions with additional mechanisms**. In this final step, the best solutions are improved by adding mechanisms to them.

The goal is to improve the cost of the solution without losing the consistency. Another possible solution is to have a pre-calculated list of solutions and to adapt them in a predefined way, as the Netlets in 4WARD project. Alternatively, maybe multiple heuristics can calculate solutions concurrently and then the best one is chosen.

3.5.6 ANA

The project “The Autonomic Network Architectures” (ANA) [Bouabene et al., 2010] has been funded by the EU commission, as a Future and Emerging Technologies (FET) project. In ANA, any protocol entity generating, consuming, processing or forwarding information is regarded as a functional block (FB). A FB does not have any given granularity it ranges from functionality like encryption till entire TCP/IP stack. Functional blocks are connected to Information Dispatch Points (IDP) (see the letter-named points in Figure 3.14).

IDPs are interfaces, which are used to communicate with a functional block and help a FB to hide its internal mechanism from outside world. ANA uses an abstraction called network compartments to encircle the nodes with same FBs. One node may be part of more than single network compartment, depends on existing functional blocks. Each compartment is independent of other compartment, so that it can have its own communication scheme such as addressing, forwarding. Functional composition is achieved by connecting FBs via different IDPs.

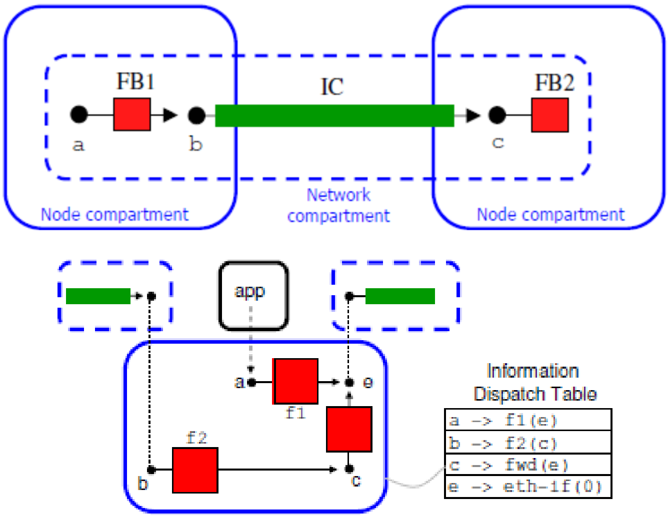


Figure 3.14. FB interconnection examples in ANA project [Bouabene et al., 2010]

ANA has also a focus on empirical testing thereby providing an implementation to building an experimental autonomic network architecture in order to demonstrate the feasibility of autonomic networking principles [Skevik et al., 2008] [Hossmann et al., 2008]. For the record, ANA released the first non-layered architecture prototype in 25th July 2008.

Currently ANA is one of the RBA/SOA architectures that have a more extensive definition and implementation of roles or FB, showing that there is a lot of work to do on this subject. As such, they provide FBs for features like:

- *Regular Functional Blocks*: Internet Protocol; IP to ANA adaptation layer; Store and Forward Transport Protocol; Functional Composition Framework Functional Block; Service Discovery; Network Sharing Compartment (netShare); Virtual Link; Nagios Report Brick.

- *Functional Blocks with Self-* Properties*: Adaptive Monitoring Framework; Namespace Routing System (NRS); Field-based service discovery and routing; Latency Measurement Service; Content Centered Routing Protocol.
- *Complex Functional Blocks and Applications*: MCIS: Multi-Compartment Information Sharing System; Failure Detection Engine; Streaming compartment; Address Agnostic Chat.

It must be noted that, inside each FB, the bricks that compose it are also described, although not exhaustively. Regarding context description and vocabularies, ANA does define *context* and *service* constraints for compartments as a way to focus and give scope to services, but defining the values and attributes described in this context is deemed out of scope of the project. Therefore, in order to describe new compartments, our project can help by providing extensive context description vocabularies that cover different scenarios and entities that could be translated as constraint values when designing new compartments and FBs

3.5.7 NETSERV

The NSF FIND NetServ [NETSERV] is based on the concept of service virtualization. In their words, *NetServ strives to break up the functions provided by Internet services and to make these functionalities available as modular building blocks for network services*. Instead of relying on overlay networks at application layer, they want to overcome the limitations of current Internet by making service virtualization available at network core, making router technology more extensible and programmable [Srinivasan, 2009]. Thereby they focus on validating their proposal by implementing a content distribution network enabled by extra services deployed on top of the Click modular router [Lee et al., 2011]. Departing from other proposals which service execution is driven by in-route packets headers and content, NetServ proposes invocating services by out-band signalling amongst routers. The services are defined and deployed by means of the OSGi (Open Service Gateway Initiative) [OSGI] framework and deployed by sending NSIS messages [Hancock et al., 2005] (Figure 3.15 briefly illustrates the NETSERV node architecture, communication and service instantiation process).

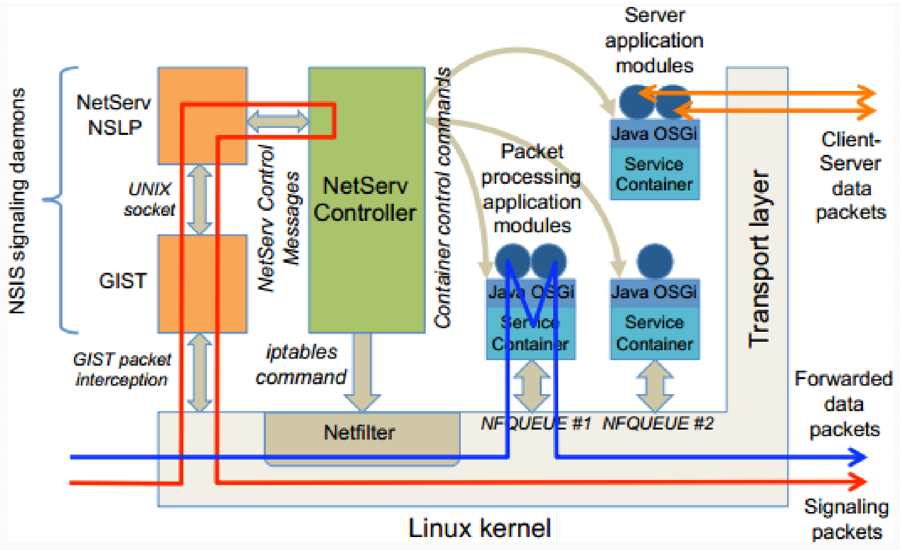


Figure 3.15. Netserv node architecture and communication process [NETSERV]

The project is focused on resolving the following research challenges in their proposed service architecture:

1. the definition of requirements for service-virtualized Internet architecture
2. the design of an architectural framework for modular, virtualized services
3. the identification of an initial set of key building blocks, which together can provide a foundation for common network services
4. the development of mechanisms and protocols for service discovery and service distribution
5. the design and implementation of a content distribution service based on our NetSerV architecture.

3.5.8 RINA

Pouzin society[14] proposes the Recursive Internet Network Architecture (RINA) [Wang et al., 2014] as an implementation design of the architecture described by John Day in [Day, 2007]. Their conceptual approach is porting the philosophy of OS layered architecture design to networking. Their premise of design is that "Networking is IPC[15] and only IPC" [Day et al., 2008]. Therefore, they postulate that every aspect of communication is a matter of communication between two remote processes, likewise two processes in the same host communicate via the OS facilities. Accordingly, a communication is composed of variable stacks of IPCs ports distributed across the communication. For executing out remote IPC communication, they define a protocol to carry application names and access control information and a protocol for error and flow control [Mattar et al., 2009]. Figure 3.16 shows an example of the RINA architecture. Vertical rectangles represent systems (machines), black solid lines physical interconnections, horizontal rectangles represent DIFs[16] (Distributed IPC Facilities) and circles represent processes. According to this view, networking in RINA is not a layered set of different functions but rather a single layer of distributed IPC that repeats over different scopes, i.e. providing the same functions/mechanisms but tuned under different policies to operate over different ranges of the performance space (e.g. capacity, delay, loss). Solid circles are IPC processes; each color is associated to the membership of different DIFs. In the figure, IPC processes can be easily identified as those building blocks or atomic services analogous to other projects.

[14] Pouzin Society. http://pouzin.pnanetworks.com/

[15] IPC: In computer science, Inter-Process Communication (IPC) refers specifically to the mechanisms an operating system provides to allow processes it manages to share data.

[16] RINA defines a DIF as an organizing structure, grouping together application processes that provide IPC services and are configured under the same policies.

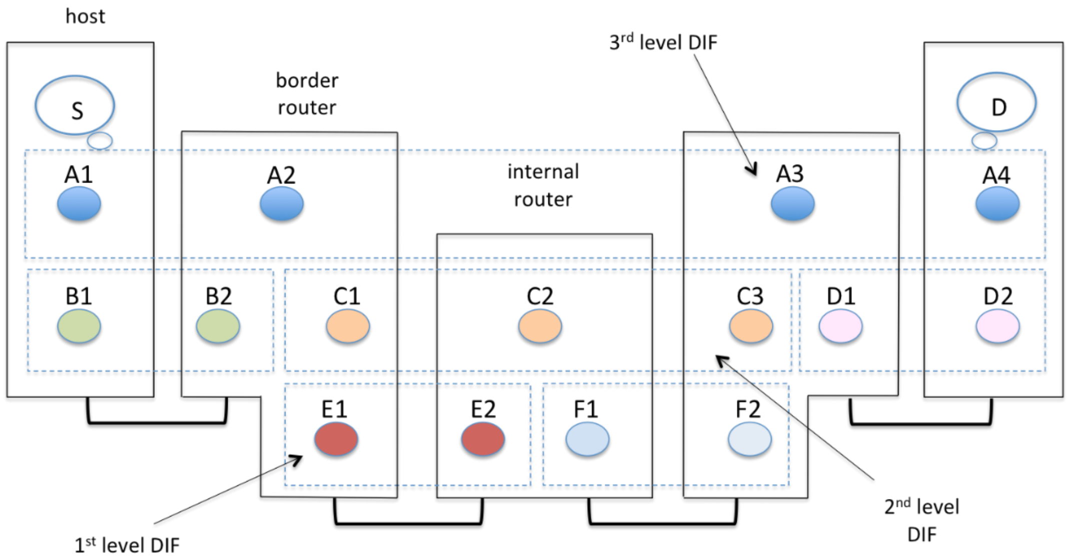


Figure 3.16. An example of the RINA architecture [Wang et al., 2014]

3.6 Summary and conclusions

As we have seen in this Chapter, Service Compostion discussion embraces a larger topic than Service Composition at network level proposed on some novel network architectures. Generic service composition approaches were analysed giving a picture of the most important service and context parameters that should be taken into account, all techniques and proposals are weighted.

In that sense, it can be derived that instance- and template-based approaches do not accomplish the dynamicity and flexibility demanded as a requirement for FNs. Therefore, they could be considered as a medium-term approach, taking into account that can simplify the composition process but their level of personalization and context-awareness is limited. Declarative approaches present very interesting near to optimum compositions respect a cost function. For example, FUSION [Van der Meer et al., 2003], although it is focused in web services, it presents a framework very similar to which is foreseen using networking services. It works with a close bundle of services that are previously defined. This fact delimits very well their compositions, as well as it is desired in this initial research. Moreover, an important feature of this composition system is that it is able to verify if the result of the execution plan that it prepared meets the user's requirements, and if not, it immediately remake an execution plan. However, it should be interesting to add additional features to empower the user selection, by providing a more accurate description of context and services.

The revision of ontology based approaches give some clues about how to declare the specific bundle of services that will be the fundamental blocks of this research. For example, DAML-S and its successor, OWL-S, present very complete and extendable ontologies for services, and a classification of them in atomic, simple and composite services. Regarding to context parameters, and taking into account all the groups defined by Dey in [Dey, 2001], the ontologies defined by WSOL should support all of them. At this point, it should require a more extensive analysis once services and context parameters are delimited. As an alternative, service and context ontologies could be defined taking into account only the parameters needed using logical language as done in [Meyer et al., 2005]. This representation can also include order restrictions in services, permitting to automate the composition process only considering knowledge parameters from service and context.

Contrarily, AI Planning comes at the cost of high computations and this is the reason that has not been selected. However, other hybrid techniques could be considered for optimizing composition processes in future work but most of them resulted out of scope for this thesis.

Regarding the discussion on network architecture design, protocol modularization and redesign of the Internet has been around since the nineties, but until the 2000-2003 lapse, no serious project with backing funding was issued.

The MIT's New Arch project [Clark et al., 2003] developed some inspirational work on new network architectures and protocols that lead to many other similar service-oriented approaches. From an architectural point of view, the most groundbreaking (and influential for our own research) outcome of this project was the RBA [5], a non-layered and modular architecture build around the concept of roles. A role represents a communication building block that performs some specific function relevant to forwarding or processing packets. Therefore, new relations amongst network components are established by defining a set of roles to be played by each component involved. This way, nodes implement the different roles required for a certain communication (i.e., "packet forwarding", "fragmentation", "flow rate control", "byte-stream packetization", "request web page", or "suppress caching"). In this architecture, the protocol stack is replaced by a protocol "heap" where each node's realm of execution in the protocol heap is restricted to the roles it exercises. Notwithstanding, no layering implies new rules for modularity, header structure and ordering for the processing of metadata, and encapsulation. This new rules and relationships were (and still are) not fully investigated and unexpected synergies and incompatibilities may appear in implementing RBA-based architectures. This kind of micro-modularization, breaking existing protocols into small modules with a definite function, is common practice in most of clean-slate proposals [Braden et al., 2003] [4WARD] [Dutta et al., 2007] [NETSERV] [Müller et al., 2007]. Although all of them aim to improve flexibility, the key difference is how this modularization is approached (in terms of scope, focus, purpose and level of granularity) and how network architecture and protocols are abstracted, defined and built on using these modules as basis; that is how the architecture and communications are composed with these modules.

For example, from the GENI/FIND program, the SILO proposal [SILO] uses micro-modularization similarly to RBA, but not to avoid layering, in fact, they generalize the layering approach to ease cross-layer interactions and avoid sub-layer proliferation. Thus, they advocate building a custom-made protocol stack (known as silo) for each connection made of fine-grain building blocks. This custom stack is the same all across the delivery context, so it is not adapted/tuned to the different requirements of each network section found in heterogenic environments. Furthermore, it is not clear how silos are negotiated between the edges.

Another take on protocol micro-modularization from the FIND project is the Recursive Network Architecture (RNA) [RNA]. In this case, similar to SILO, the modularization approach is not to avoid layering but to generalize layer design/implementation, avoid recapitulation and make it flexible by taking profit of the existing synergies amongst the different protocols layers. As such, some basic services are detected as being common in most protocols of all layers of the protocol stack and, therefore, a single meta-protocol with a unified API can be designed for instancing and tuning protocol layers and stacks. This way, multiple instances of different stacks, each one with their own set of “personalized” layers tuned to different purposes may run simultaneously in a node. This approach allows instancing any number of layers in any order, hence providing support for shim and virtual layers with ease. Making the RNA approach viable requires solving different challenges like: stack management and meta-protocol instantiation/tuning negotiation, interlayer coordination, context and performance awareness and adaptability, etc.

The 4WARD [4WARD] project built all the architecture on top of a virtualization layer. Their take on protocol design is focused on analyzing protocol invariants in order to build protocols from its most basic functional blocks (e.g., error control, encryption, etc.). Thus, similarly to Object Oriented Programming, they advocate virtualizing the different architectures and protocols using constructions called Netlets made of several functional blocks to meet specific requirements. In this way, using interpreted programming languages like Java, nodes instantiate and execute Netlets on top of a kind of protocol virtual machine.

From the G-Lab Initiative in Germany comes the SONATE proposal [Müller et al., 2007]. Their approach to protocol micro-modularization is designing a network architecture based on the Service-oriented Architecture (SoA) paradigm. Thus, opposed to layers, services within nodes are the building blocks of the architecture. Services provide self-contained functionality with a well-defined interface, hiding the internal mechanism and data structures and allowing for a loose coupling amongst services, easing service composition and decoupling from service interface definition and service implementation. The granularity proposed for services is similar to micro-protocols (like “ensure reliable transmission”) where several interacting services will be orchestrated in a single communication workflow in order to provide complex functionality. For service intercommunication, they propose interchange streams of messages consisting of TLV (“Type-Length-Value”) structures, which allow high flexibility in ordering messages.

ANA presented another European interesting approach, defining an architecture of composed Functional Blocks (FB) and a set of FB with different levels of complexity. It even defines some FB for IP adaptation. It must be also noted that, inside each FB, the bricks that compose it are also described, although not exhaustively. Regarding context description and vocabularies, ANA does define ‘context’ and ‘service’ constraints for compartments as a way to focus and give scope to services, but defining the values and attributes described in this context is deemed out of scope of the project. Therefore, in order to describe new compartments, it should be defined extensive context description vocabularies that cover different scenarios and entities that could be translated as constraint values when designing new compartments and FBs. In addition, autonomic composition process is not defined with the same clarity, focusing on the architecture and letting the composition process to a manual off-line work.

NetServ [NETSERV], divergently to the other proposals analyzed, represent the "more evolutionary" way of presenting a Service-Oriented network architecture. Laying over IP, and focusing on the service virtualization concept, it provides a Java-based framework to instantiate services on "NetServ nodes". Although firmly rooted in evolutionary path and linked with IP solutions, some of their work might be of interest for the definition of services/roles, since they want to enhance router functionality.

RINA project was included in the analysis, although it is not directly related to RBA/SOA solutions. It proposes a rethinking of the legacy Internet design principles, mainly layering, but their approach is interesting in the sense it defines the scope of communications, as paired services running in different nodes. This concept was considered when designing the architecture, as it could be incorporated in a form similar to network compartments in ANA or meta-layers in RNA.

Table 3.1 shows a summary of the different approaches analyzed, comparing how they tackle different stages of the service provisioning.

Table 3.1 Service provisioning comparison in FN architectures

SERVICE PROVISIONING STAGES					
	Definition	**Publication**	**Discovery**	**Composition**	**Adaptation**
R B A	Services as essential building blocks that provide self-contained functionality called roles.	Within the node.	Not considered.	Manual composition, described in the service specification which node realize which role.	Not considered.
S I L O	Middle ground definition of services between the strict protocol stack imposed by current architectures and the "heap" approach advocated by the RBA.	Previous compositions can be stored to be reused within a node.	Not considered.	Simple ontology for specifying dependences and constraints between services. Control agent executes a simplistic recursive algorithm based on pre-established ordering of services	Different implementations of the same service. Service chain dynamic adaptation is not considered.
R N A	Single protocol, (metaprotocol) with a set of basic services (configurable capabilities). Three types of elements: Data element, Control element and Template element.	Previous compositions can be stored to be reused within a node.	Not considered.	The metaprotocol is tuned (manual or automatic) depending on the context in which it is located.	Not considered.

4WARD	Object-oriented, defining Functional Blocks, Functional Block Mechanisms and Netlets as constructions of them.	Maintain a repository containing Building Blocks and Design patterns for guiding the composition process. Only NetInf[17] proposes an approach that uses DHTs to publish Information Objects.	NetInf proposes a dictionary based on DHTs.	Decomposition and recomposition. Chaining Functional Blocks into Netlets. Done at design time by a network or a service architect.	Dynamic behavior provided by a set of policies that tunes the building blocks at run-time In-network management approach considers building block adaptation.
SONATE	Services as essential building blocks that provide self-contained functionality, well-defined interfaces and loose coupled among them. Also defined by a set of effects.	Service Providers publish services at Service Broker.	Service Consumers ask for a service to Service Broker.	Service Broker is the entity in charge of selecting and composing the service matching to the requirements demanded by the Service Consumer. It relies this process on Analytical Hierarchy Process (AHP).	Not considered
ANA	Functional Blocks (FB) defined as any protocol entity generating, consuming, processing or forwarding information. They are connected through well-defined Information Dispatch Points (IDP) to create services that are more complex.	Not considered. Communications composite services are manually created and not published.	Service Discovery is such another FB, but it refers exclusively to find the service at the other end, not network capabilities or FBs in intermediate nodes.	Services are manually created. It only specifies how to combine FB in order to create custom protocol stack in the end systems.	FBs present some adaptation properties to be self-adaptive, but the composition does not have adaptation capabilities.

[17] Netinf is another initiative of 4WARD focused on defining an Information-Centric Architecture. Although they are part of the same project, the netlet node architecture and the NetInf node architecture are different and it does not define any detail that explains a possible interconnection between them.

NETSERV	Present a router architecture intended for dynamically deploying in-network services. Service modules are written in Java using OSGi framework, and deployed by sending NSIS signaling messages.	Services are defined as self-content functions by means of OSGi service framework, but they are not published outside the Netserv node.	Services are discovered and remotely invoked in a node through NSIS out-of-band signaling.	The framework does not present any functionality for autonomic composition. The composition should be manual before in order to guide the service invocation and deployment in each node.	Logic can be applied to the NetServ Controller module of each Self-adaptive service implementations. The logic of the Netserv Controller module in the Netserv node can be set to react to certain NSIS messages, but the grade of possible adaptations is not defined.
RINA	Services are defined by a set of primitives and arguments, and a set of rules. Associate each protocol to a service definition, but separate its behavior.	Define a Resource Information Exchange Protocol (RIEP) to populate a Resource Information Base (RIB) with application names, addresses, and performance capabilities.	Routing is executed creating the forwarding tables based on RIB information.	RINA bases its composition in an establishment of an Inter Process Communication (IPC) among nodes.	RIEP can be used in both, a request/response mode and a notify/confirm mode, using the same managed objects. This allows IPC processes to notify the other members when there is a significant change or to request information.
TARIFA[18]	Self-contained, self-describing, modular block offering a specific functionality that can be published, located, and invoked across the network. This self-* definition should permit loose-coupling among services	Distributed Plane (DP) used for maintaining services and context information. DP is divided into a Resource Plane (RP) and a Knowledge Plane (KP).	Propagation of semantic requests specifying the requester requirements. Nodes evaluate incoming requests taking into consideration context information.	Nodes can compose services and adapt them in runtime. Compositions are specified in the form of WFs.	Services can be self-adapted considering context variations.

Besides the reviewed projects[19], some other more recent initiatives were proposed with a service-oriented perspective or gathering some protocol or service micro-modularization concepts.

An example is FUSION “Future Service Oriented Networks” [FUSION], an European 7th Framework Programme project started in 2013. It looked for a new networking architecture

[18] TARIFA architecture is detailed below in Chapter 4.

[19] Abovementioned projects and initiatives are a summary of the most influential proposals for the work of the subsequent work of the thesis and were reviewed when building the state of the art of service-oriented FN architectures along the duration of TARIFA project (2010-2012). It is updated with later outputs of some of these projects and complemented with some new initiatives.

designed to natively support efficient provisioning, discovery and execution of service components distributed over the Internet. FUSION, similarly to TARIFA, also aims at combining service instantiation and network routing at a fine granularity, integrating service provisioning and service-centric networking, although it especially focused on the service-centric routing and network management protocols, mechanisms and algorithms. It did not deepen on the composition process and methodology, although it did present some approaches related to the multi-domain service resolution time after the finalization of TARIFA and not covered in the scope of this thesis.

Other proposals such as MAKI (Multi-Mechanism-Adaptation for the Future Internet) [Steinmetz et al., 2015], funded by the German Research Foundation (DFG) and Technische Universität Darmstadt followed a much more academic approach (compared to the industry-led NFV proposal), investigating on dynamically adaptive protocol stacks for FN communications. They define that future communication systems must dynamically switch between different mechanisms that serve the same purpose but are designed for different surrounding conditions. In doing so, applications, services and protocols are able to dynamically adapt to different settings in an optimal manner. Hence, they follow part of the same rationale that motivated this thesis. However, they are still working on some research questions such as: how to determine the level of (de)centralized decision making concerning transitions, how to capture and cope with the dependencies between different mechanisms, or how to exploit the information of a mechanism's current state in order to achieve a smooth transition to another state.

However, considering their impact and the attention it has taken from the industry, the most relevant initiative in the recent years is the Network Functions Virtualisation trend [Han et al., 2015] [NFV, 2015]. Fundamentally, Network functions virtualisation is about realising in-network processing, such as the functionality offered today by middleboxes, in software that can run on a range of industry standard server hardware and that can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment.

NFV proposes, to the extent possible, to replace network functionality that resides in specialised hardware running in the network (such as middleboxes), with equivalent functionality (virtual appliances), running in virtualised environments on commodity hardware. It intends to develop, deliver and deploy network functions as software components running on industry-standard commodity hardware platforms. Coordinated work on NFV started amongst the carriers with informal discussions at the Open Networking Summit of April 2012. In September 2012 it was decided to parent under ETSI as an "Industry Specification Group"[20], which was approved in November 2012 and held its first meeting in January 2013. Nowadays, the membership of ISG NFV has grown to over 290 individual companies including 38 of the world's major service providers as well as representatives from both telecoms and IT vendors.

[20] http://www.etsi.org/news-events/news/644-2013-01-isg-nfv-created

At the same time, NFV trend define the flexible composition of network functions as network service chaining (NSC), a generic service concept in which an abstract network service can be decomposed into more refined, inter-connected network functions. It is basically the same objective as this thesis but NSC focuses on ISP kind of network functions (e.g. firewalls, CDNs, WAN optimization, etc.) more than fundamental communication services (e.g. acknowledgements, retransmissions, enconding, etc.), but the composition processes and orchestration defined are similar [Sahhaf et al., 2015].

3.6.1 Thesis approach

All in all, the approach of this thesis on protocol micro-modularization aims at providing a flexible framework of reusable components in order to build context-aware services on top of the architecture. It adopted a similar approach to "RBA", "SILO" or "SONATE" in the service modularization, and the context-awareness focus is reflected in most of the design decisions made and devised features. For instance, one of the objectives of this service-oriented approach is provide a framework for building "custom" context-aware protocols able to react and adapt to changes in context (link conditions, device capabilities, energy constraints, etc.) at run-time. Therefore, services are allocated along the communication path and are composed into a single coordinated workflow for each communication flow.

It must be noted that, in contrast with other proposals [Braden et al., 2003] [Dutta et al., 2007] [Müller et al., 2007] where the same micro-protocols are allocated at each node, here it proposed to use different functions/modules/micro-protocols at different nodes in order to obtain the desired global behavior. Another distinct feature of our proposal is the semantic discovery of services, thus incorporating context-awareness in service, and consequently, route discovery. Another difference with other proposals is that it integrates a QoS-friendly protocol for service discovery and negotiation [Sanchez-Loro et al., 2009] into the architecture in order to discover services on demand and negotiate and configure (according to a certain SLA) the modules to be used by each node in a communication flow.

Finally, revised FN projects present similar approaches, in the sense that both use a similar set of services (mainly based on Role-Base Architectures), although they differ in the level of dynamicity and flexibility that they offer. For example, SONATE presents a promising approach trying to implement a more dynamic service composition defining a set of effects and meta-effects to have a close control of the iterative composition process, while 4WARD realizes it in a more statically way, not permitting a real automatic and dynamic composition at run-time. A high level of dynamicity and flexibility in service composition increments the complexity, but achieves always-tailored communications. However, it could derive on long connection establishments that decrement the Quality of Experience (QoE) of the user. Therefore, there is a challenge not yet resolved in finding an efficient trade-off with regards to this point, which enables acceptable connection establishment delay and accomplishes a high level of personalization and adaptation demanded by the end user.

4 Proposed semantic context-aware network architecture

This section summarizes the details of the semantic context-aware FN architecture model that is taken into account in the thesis for the service composition proposal over FN. The proposed architecture works inherently with protocol micro-modularization and services, and presents an approach for service description, discovery and composition extending previous proposals reviewed in Chapter 3. It is an output of the TARIFA project [TARIFA] [Sallent et al., 2010], in which I participated in the early stage of the PhD. The work done in this project was guided by a small group of researchers, so everyone could contribute on the final definition of the architecture principles and how the elements of the architecture were mapped into a node. In this sense, I especially worked on the service-orientation of the architecture, defining the requirements and the modules of the node to enable the service composition mechanisms. This work allowed the subsequent development of the service composition methodology and tools that are explained in the following chapters, and how they are applied as an approach for building context-aware services and applications in the media distribution and Smart Grid fields (with or without the FN architecture presented in this chapter).

4.1 Why change is necessary?

Considering that the Internet is facing an architectural crisis as the load and stress on the network increase with new users and applications, the architecture of the Internet is becoming more and more ossified and complex; with lots of patches aimed to amend different issues that have arisen during its almost 40 years of existence [Feldmann et al., 2007]. This patching has been done outside the core of the architecture, increasing its complexity and blurring -and sometimes even contradicting- the neat principles behind the original design of the Internet. Furthermore, new services and computing paradigms require new modes of interaction, new features (identification, context-awareness, seamless service discovery and composition, etc.) and clean solutions to known issues (mobility, security, QoS, flexibility, etc.); but it is not clear how the current Internet architecture will be able to cope with all these new requirements.

Current layered network architecture based on TCP/IP stack has enabled the global deployment of Internet, internetworking lots of different devices and services. However, it was neither designed nor optimized for the rising of new features, services and applications required for the FNs. As abovementioned in the previous chapters, current Internet deficiencies have been studied especially during the last decade, suggesting the necessity of a change. In summary, the studies, initiatives and projects that worked in this field define an optimal underlying network architecture that allows the flexibility, context-awareness and adaptability of the communication infrastructure required to ease the development of ubiquitous services and applications.

This rigid layering approach derives in monolithic network architectures, lacking flexibility. Network functions are executed regardless of the characteristics of the surrounding context, underlying network technologies and capabilities of the devices involved in a communication. So, for some situations, the same functions can be redundantly executed at different levels, degrading communication performance and wasting computing resources. Even worse, in certain environments, the execution of certain functions can be counterproductive for the correct operation of an application or network service (e.g. TCP's congestion control in wireless networks), obliging to modify existing protocols to adapt them to environments with restrictions.

In addition, inter-layer communication is strongly restricted. This has led to different cross-layering approaches to enhance protocol and application performance in wireless environments where traditional protocols show a poor performance. These cross-layering solutions, violating the layered structure of the stack, further complicate the situation, bending the model and its related standards. Although having their benefits in optimization terms, from an architectural point of view, these practices pose serious interoperability issues resulting in an increased complexity of the network architecture.

Another cause of strife is the existence of new sub-layers like MPLS at layer 2.5, IPsec at layer 3.5, and TLS at layer 4.5. These sub-layers are features not considered in the original design of the stack and they have been implemented as patches as a consequence of the lack of flexibility of the TCP/IP stack.

Besides it, middle-boxes (NATs, proxies, gateways, etc) erode the end-to-end model as new features and participants are placed in-network without control and knowledge of the edges. These now common solutions further complicate the situation, as they were not considered during the original design and development of the Internet (they did not exist then). Similar situations will surely arise during the following years as technology and applications evolve, and the TCP/IP stack lacks flexibility to clearly and easily deal with them.

Most of these issues derive from the fact that TCP/IP stack was designed with wired networks and mainframes in mind. The first users of these earlier networks were a trustful community of people with high technical skills, enforcing the end-to-end arguments. Protocols were tailored for the capabilities of the nodes and technologies of the time, whilst the applications were much simpler than today's, in fact most of them were simply replicas of the services of an operating system on a network (telnet, FTP and RJE). In the following decades, Internet applications and network services evolved, increasing its complexity and requirements, diverting them from a strict end-to-end philosophy. Quality of service (QoS) and security were introduced as critical issues. With the rising of wireless networks (mobile, ad hoc, mesh, sensor) and mobile devices, new applications and issues arise: mobility, hostile and time variant access media (interferences, lower bandwidth, higher error rates, higher latency, etc.), intermittent connections, energy restrictions, multi-modality issues, device capabilities issues, localization, nomadism, roaming, context-awareness, network and device heterogeneity, transcoding, etc.

Furthermore, with the rising of commercial Internet, the user community grew with millions of non-technical users and different stakeholders (operators, content providers, governments, etc.) became involved. This growth changed the Internet landscape forever, since some Internet players' interests are (and will always be) directly at odds with each other. Thus, in our brave new world, Internet players cannot blindly trust each other anymore, requiring new mechanisms to enforce security, privacy and provide users with more control over their communications (including which routes will be used). Another side effect of this growth causing deep concern for the Internet community is routing scalability, which involves both the size and dynamics of the global Internet routing table.

Now, there’s a myriad of different devices, applications and network technologies. These advances have allowed new computing paradigms like Pervasive and Ubiquitous Computing or the Internet of the Things, to name a few, but these paradigms shift from strict end-to-end arguments and pose very different requirements and philosophy than original TCP/IP applications -some don’t even use the IP protocol [Amadeo et al., 2016]. Information-centric networking for the internet of things: challenges and opportunities. IEEE Network, 30(2), pp.92-100.]. They require new modes of interaction between nodes and components of network services not fulfilled by current network architectures. So, network services should evolve on an architectural framework to become flexible, ubiquitous, composable, dependable, secure, context-aware and adaptable in execution time. Following this trend some discussion about “clean slate” re-design of the Internet has arose during the last years and, we believe it will intensify in the next years. Now, it is time to design the Network with the characteristics we would wish to take for granted in 10 years.

In the research community, there are two approaches to solve this crisis: evolutionary and disruptive. I agree with the opinion that it is hard to believe the evolutionary path will be able to solve efficiently all the issues and challenges facing the Internet [Anderson et al., 2005]. On the contrary, the revolutionary one presents a challenge itself, on providing a gradually deployable solution and a way to migrate from current architecture to completely new communication methodology [Rexford et al., 2010] [Fisher, 2014].

Considering the fact that most of the known issues to solve from current Internet derive from the basic architectural principles behind the TCP/IP stack and are difficult to change with an evolutionary perspective, part of the work done in this thesis relies on a revolutionary clean-slate architecture based on service-orientation concept, as reviewed in chapter 3. Hence, contributing to the FN architectures research topic, an alternative architecture is proposed, trying to define an approach that redesigns the network architecture with current requirements, building on all the knowledge on the field gathered during these years. Besides, various subsequent contributions (in the fields of context-aware media distribution or smart grid data management, detailed in following chapter of this thesis chapter 4 and chapter 5, respectively) are built upon the foundations of a clean-slate service-oriented network architecture.

4.2 Proposed principles for a service-oriented FN architecture

The following characteristics and features were discussed and agreed in TARIFA project as the main principles in which the proposed architecture should be based:

- <u>Flexible design and execution</u>. When designing a new network architecture, it is relevant to consider that it should be flexible and adaptable enough to perform reasonably well in all kind of environments, without making assumptions about execution environment, infrastructure support or a minimum set of device capabilities. It should be truly ubiquitous, providing tools for consuming network services anytime, anywhere, and anyhow (that is with any device, any platform); thus, integrating all kind of edge networks, platforms and devices [Raychaudhuri et al., 2005]. This is very important because most advances in the networking field come from the edges. Actually, even those very centralized paradigms such as cloud computing are moving to the edge, in the way of Fog/Edge Computing [Bonomi et al., 2012], pushing applications, data and computing power (services) away from centralized points to the logical extremes of a network. Therefore, the architecture should provide ways to distribute the intelligence (not only of the network itself, but also the logic of applications and services above), in order to support new trends on IoT computation, Smart Grid management, etc.

Therefore, a shift from strict end-to-end arguments is needed, building a network architecture that provides more intelligence to the network-side whilst still leaving decision-making processes to the end-points [Clark et al., 2005] [Anderson et al., 2005] [Raychaudhuri et al., 2005].

In the proposed architecture, network functions are allocated according to each situation and not in a monolithic way. Thus, functions are allocated all along the route, executing just the desired functions at each hop, section of hops and end-to-end and applying them just to the desired transmission unit (bit, frame, packet, etc.). The architecture provides flexible support for different semantic schemes or vocabularies for identifying services, resources and nodes and to describe their capabilities and schemes for specifying desired/requested and provided QoS, in order to adapt it to different contexts, from media distribution to IoT and Smart Grid communications.

- <u>Oriented to service/resource interconnection and not machine/interface interconnection</u>. If we want that FN architectures are able to deploy truly ubiquitous and heterogeneous networks, the architecture should be designed to avoid hierarchical layering. It should not depend on the current layering constraints that lead to intrinsic efficiencies (e.g. congestion control in wireless communications [Xylomenos et al., 2001]). Instead, a service-oriented approach for a flow-oriented context-aware network architecture is proposed, where communications are composed in situ (using reusable components) according to the needs and requirements of the consumed service. Thus, the network is defined as a set of services and nodes.

  On the one hand, nodes are those physical (device) or virtual entities (cluster) that possessing networking capabilities are able to consume and to provide services to themselves and other entities and service. They provide the necessary environment for service composition and execution.

  And on the other hand, services, which are classified into atomic and composed services. Atomic services are those individual functions commonly used in networking protocols (i.e. acknowledgments, sequence numbers, flow control, etc). These are well-defined and self-contained functions, used exclusively to establish communications for consuming composed services. Composed services are network applications with a wider scope than just establishing communications (e.g. printer service, directory service, file transfer, instant messaging, presence, etc.). Each composed service or application imply consuming different atomic and, sometimes, other composed services; appearing possible dependences between them. Also, they can involve one or more nodes, depending on the complexity of the service.

- Context-awareness and dynamic adaptability during execution time. The flexibility design abovementioned should be complemented by dynamicity of the architecture, able to understand the context in which it is being deployed and adapting their requirements and behavior accordingly at service execution time. In other words, the architecture should be able to provide enough context information (about the network, the nodes and the communication or service requirements) to the services that are running over it, and enable them to adapt to the execution environment. For instance, in current-day networks we are used to integrate different kind of communication networks with different requirements and capabilities. It is the usual scenario of the Smart Grid, in which we combine IoT networks connecting sensors, IP over Ethernet, 4G or PLC to connect to Smart Meters, Fiber Optics for tele-protection services, etc. In these so heterogeneous networks it is easy to find nodes that may require very different network functions to communicate with other nodes in order to consume the same or a different composed service21. If we take a look at the heterogeneous network example such as the one in Figure 4.1, on the one hand, there may be network segments with reliable communication environments neither requiring error correction nor error recovery, or perhaps very basic functions like a small CRC computation. This could be the case for nodes connected with wired and reliable links (e.g. segment E). On the other hand, other network segments could require strong error detection and recovery mechanisms in order to accomplish reliable data delivery in front of high error rates in unreliable links (e.g. segment A, B, F). Function allocation not only depends on network and link state but also on device capabilities. In this way, there may be nodes with strict restrictions in battery, memory and CPU preventing them to perform intensive operations like message sequencing, data transcoding or executing complex timed state-machines (e.g. segment B).

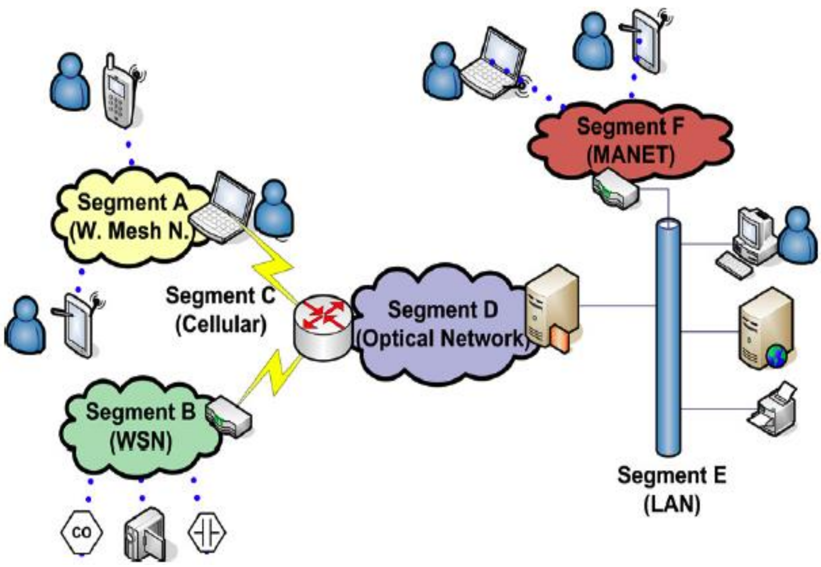


Figure 4.1 Example of heterogeous networks

[21] Examples of a composed service in the context of Smart Grid could be teleprotection, Smart Meter reading, check Electric Vehicle (EV) charging point status, etc.

The proposed architecture takes into account the capabilities of the nodes, services and network links to establish new routes and to manage existing ones. So, mechanisms to interchange context information between entities in ad hoc and structured ways must be supported. In this model, in order to obtain the desired behavior, functionality and QoS constraints, communications are established concatenating atomic services into a workflow for consuming a certain composed service. So, atomic services are the building blocks used to establish communications and to deliver data in a self-adaptable, self-configurable and context-aware way.

In order to provide and consume (composed) networked services in a truly ubiquitous fashion (according to desired functionality, behavior and QoS constraints), atomic services must be suitably allocated along the communication path where they are actually needed and tuned/configured accordingly to the context in order to accomplish the application requirements. They are allocated amongst involved nodes, as required by conditions of temporal context and service requirements. In this way, all functions are used only when and where they are required, so it is assured that there is no function overlapping or usage of counterproductive functions. Besides, atomic services can be executed in a per-hop, per-section (between two non-adjacent nodes) and/or end-to-end basis (section ranging the entire route) and they can be executed with different levels of granularity, depending on desired functionality (bit flow, frame, packet, object, byte flow). For each level, a different unit of information is processed/affected.

- Semantic identification and addressing of nodes, resources and services. Nowadays TCP/IP architecture still has to deal with inherent mobility problems [Mazoni et al., 1995] that has led to intensive research on naming and addressing, looking for a locator and identifier separation [Quoitin et al., 2007]. Instead, addressing schemes should be designed to be dependent on the location of entities in a network, but route independent [Day, 2007].

  The architecture also promotes a shift on the focus on network addresses to a service/data-centric approach that allows the semantic discovery of services and resources (including data objects), easing the interaction with network facilities. This proposal defines service discovery and, hence, routing is based on the semantic description of the desired service. This way, mandatory explicit addressing (and naming) is avoided. Besides, existing addresses (locators and identifiers) are treated as another characteristic of the service/node/resource. Services and nodes are identified semantically through attributes; avoiding using just addresses to identify a node's interfaces (addresses receive same treatment as the rest of attributes). Hence, arbitrary interface addressing is substituted by semantic description of the desired service. According to this description, requests are routed searching for its final destination: a service entry-point able to provide the requested service. When a node provides or participates in a service that matches this description, it responds. Furthermore, this semantic context-aware service-derived route discovery approach provides intrinsic support for mobility, multihoming and nomadism. Macro-mobility (inter-domain) and nomadism are supported by the semantic routing scheme. However, mechanisms for route management and reallocation are needed to cope with route degradation during micro-mobility (intra-domain). Also, multihoming is supported when required (addresses are just attributes), but more important, multihoming can be dodged by semantic identification as we search for a service or node, but not for a certain network interface address.

- QoS integration and user empowerment in service choice and routing. Discovery, establishment and management of routes should be based on requested QoS and resource availability [Yang et al., 2005]. In addition, with the diversification and popularity of the Internet and the maliciousness of some of their players [Clark et al., 2005], users should be provided with mechanisms that allow them more control over their communications.

  In the architecture, this control is reflected in flexible routing and service selection. So, a service requester chooses from matching service responders which specific service wants to consume. Also, if desired, source can specify preferred and trusted carriers/domains and blacklist distrusted or malicious domains, this way end-points have a certain degree of control over which routes their communications follow.

- Security functions must be fully integrated in architecture design. Security must not be an addendum. Internet was developed and deployed by researchers that trust each other and security issues were not considered from the beginning [Bellovin, 1989] [Sicari et al., 2015]. Internet nowadays has become the most critical tool for most of the business, and its social and economic impact is massive now. With multiple interests of current users – many times lead by economic tussles– Cybersecurity issues have provoked lot of concern nowadays.

  Thus, TARIFA architecture proposes service discovery/consumption taking into account available security features. Other points to assure are: data integrity and confidentiality, plus user privacy and confidentiality. Also, another important characteristic is traceability, i.e., finding the path to the traffic source for a specific communication. As each connection is established for service consumption, traceability can be achieved by univocally tagging each single traffic flow. Therefore, a unique tag (Session Identifier) is used to identify and route each traffic flow.

4.3 The new architecture definition

In order to give a brief summary of the TARIFA FN architecture, this section describes first the elements that form it.

4.3.1 Services

A service is a set of network-related functions executed with a common purpose at the request of a consumer (other services, node or user). Services are also described with attributes. Attributes are used to help with the identification of services and to express parameters or "knobs" for service configuration. Services are classified into atomic and composed services.

Atomic services are those individual functions or roles commonly used in networking protocols (i.e. acknowledgments, sequence numbers, flow control, etc). These are well-defined and self-contained functions, used exclusively to establish communications for consuming composed services. It must be noted that atomic services can model any role of legacy stacks, even functionalities of legacy application protocols. Thus, a composed service can use its own protocols as data on top of the workflow of atomic services; but, with the introduction of the suitable atomic services, a composed service may just send data without control messages, since the introduced atomic services can be configured to control the behavior and operation of the composed service.

Each atomic service provides one concrete and well-defined networking function (along with the reverse function, if any). Different algorithms and implementations of an atomic service could exist (i.e. different congestion control algorithms), and co-exist in the same node, using attributes to both describe the different possibilities and to tune/configure the atomic service in order to use it to fulfill specific workflows needs.

Atomic services can be executed with different levels of granularity, depending on desired functionality. For each level, a different unit of information is processed/affected. We define 5 levels of granularity for atomic services:

1. Bit flow. At bit flow level, atomic services are executed affecting the whole communication flow at bit level paying no attention to logical abstractions, akin to circuit-oriented communications[22].
2. Frame. At frame level, services are executed on per-frame basis[23]. So, all transformations, headers and responses use physical frames as basic I/O unit and interfaces. The frame level has its own sublevels of granularity:
    a) Default: service execution affects the entire frame.
    b) Payload: service execution only affects frame payload.
    c) Overhead: service execution only affects frame overhead.
3. Chunk. Session data is fragmented in chunk for its transmission according to the requirement of the transmission medium. At chunk level, logical data chunks are the basic I/O unit. Identically to the frame level, chunks level granularity has the default, payload and overhead granularity sublevels.
4. Object. In object level, services are executed on a per-object basis. An object is an encapsulated/structured data resource and may contain further objects inside. In order to feasibly implement this level of granularity, support for the object flow atomic service must be provided in all the nodes executing services with object-level granularity.
5. Byte flow. In byte flow level, services are executed affecting all the session's data flow (without overhead) and treating it as a raw stream of bytes.

Composed services are network services or applications with a wider scope than just establishing communications (e.g. printer service, directory service, instant messaging,

[22] In this case, services are negotiated as usual, agreeing on a *Encoding & Signaling* service implementation that allows working on circuit mode (allocating a dedicated frequency, TDM scheme, etc); giving support for a continuous traffic flow and just tagging the traffic flow at the start of the session with a session id

[23] This is for backward compatibility and practical deployment of the solution, since programmable and adaptable physical technologies are not widespread and most deployments may probably be on top of existing physical or virtualized layers. For programmable and adaptable physical mediums, defined atomic services will just work with chunks or bit flow according desired granularity.

presence, etc.) which imply consuming different atomic and, sometimes, other composed services in a single workflow; appearing possible dependences between them (i.e. much like the current WWW's dependence on the DNS service. Also, they can involve one or more nodes, depending on the complexity of the service. Besides, service execution may depend on certain node capabilities (hardware, software, network, etc), thus becoming inoperative or changing its operation mode based on changes on node capabilities. These dependencies can also be expressed as profile attributes. In order to ease composed service invocation, a univocal identifier and different locator attributes may also be provided. Thus, services could be found via semantic attribute matching or searching for locator attributes (addresses, URLs, interfaces, etc.). This way, different approaches could be applied to service composition, allowing flexible service coupling.

4.3.2 Nodes

A node is a physical (device) or virtual (e.g. cluster of nodes) entity with communication capabilities able to consume and provide network services to itself and 3rd-party entities and/or services. It provides a run-time environment for service execution and composition. Node and service capabilities, including possible addresses and locators, are described with attributes

The architecture is defined as a set of services and nodes.

4.3.2.1 Node architecture

The proposed architecture can be represented from the node's point of view as illustrated in Figure 4.2. It contains the following entities:

- Application. Final data sender and receiver. Session initiation and other control messages are relayed to controller, whilst data transfer messages are relayed directly to the service manager.

- Controller. Signaling and control protocols manager. It manages the transmission and reception of service discovery and negotiation protocol packets, plus transmission and reception of context exchange and other management protocols (if any). It solely manages the transmission and reception of the service discovery and negotiation protocol packets, plus transmission and reception of context exchange and other management protocols (if any). It can initiate the discovery and context exchange procedures at request of other entities (see interactions).

- Service Manager. It manages the execution of composed services, that is, it manages the reception and transference of data packets and reacts to any non-fulfillment of SLA[24] signaled by the Service Monitor, signaling to the Controller at its own discretion which actions should be performed according to the policies specified by the Service Composer during service composition. This component is in charge of managing the proper execution of the different atomic services required in a communication, which

[24] SLA concept basically remains the same as legacy networks, but its application is more fine-grained: we have providers and consumer agreeing on a certain level of service assurance and behaviour. Tussles amongst providers will probably remain the same as today, hence the same problems to establish long-standing contracts and agreements between different providers will arise. However, contracts established among the different stakeholders will allow to the appearance of novel and innovative business models.

are defined by a workflow, and form a composed service. Notice that the execution of the desired services generates the data transferences (Service Plane) among the involved nodes. Furthermore, this component can decide if modifications in the operation of a service according to management reports are required. In addition, if modifications are required, the Service Manager can ask the controller to execute the negotiation protocol, which will allocate the required services involved in a communication.

The Service Plane carries data generated by those services in execution. Data includes useful data exchanged among atomic services and the associated and required signaling by each service. Then, the Service Plane transports all protocol services' messages.

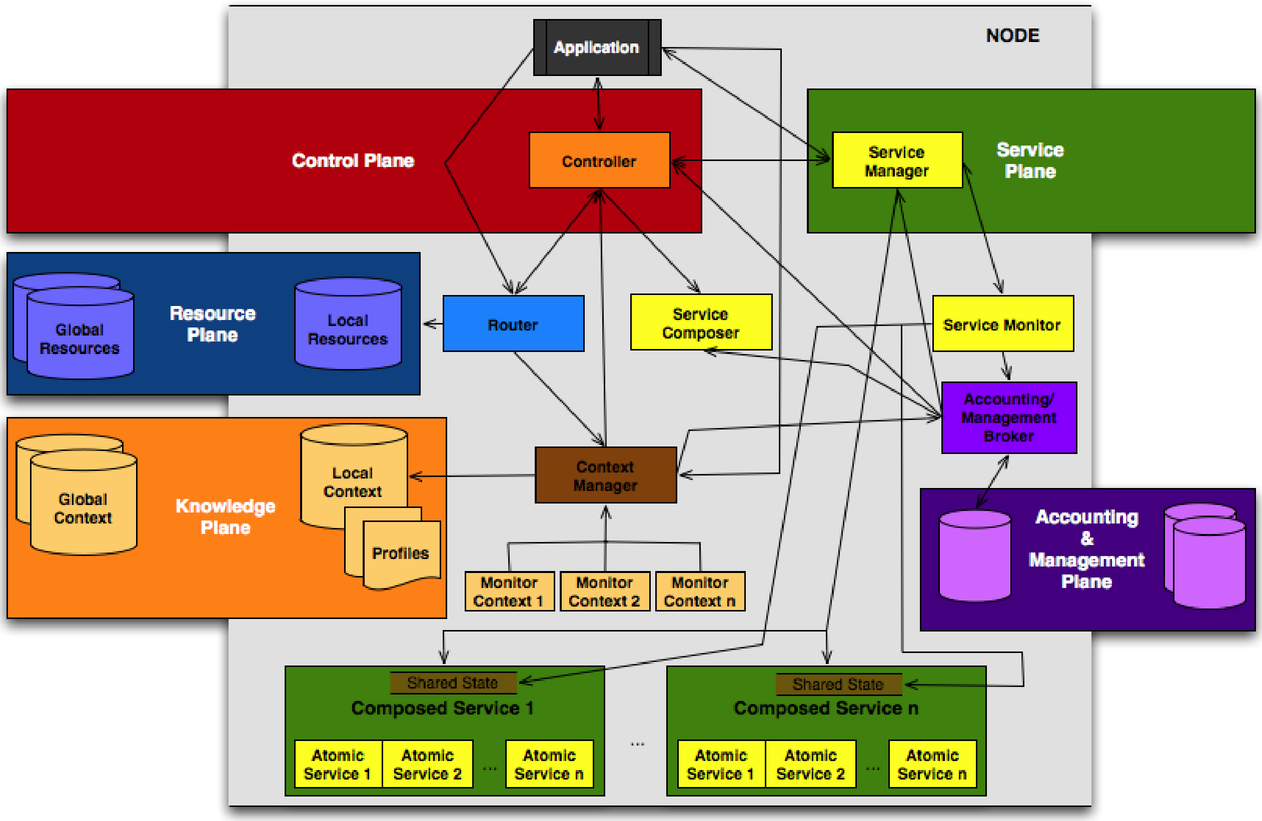


Figure 4.2 Node Architecture

- Service Monitor. It monitors composed services behavior and reacts to any non-fulfillment of SLA, signaling any SLA violating event to the Service Manager. This component is focused on the service delivery state monitoring of the atomic services required in a communication; the state of the executed atomic service is monitored directly by looking-up the shared state database. The monitoring process must provide information allowing to the Service Monitor to know state events generated during service delivery and inform about these events to the Service Manager.

- Service Composer. It manages the composition, allocation and orchestration of services (both atomic and composed) from the control data obtained from the controller via the discovery and negotiation protocol. It dictates the orchestration policies (establishing relations and interactions between ASs and, possibly, between different composed services) to compose the service workflow. This component is responsible of defining a workflow for a composed service. This workflow will be formed by instances of different atomic services. This component receives a call from Controller component (Control Plane), requesting for a new composed service from Service Composer. In the composition process, Service Composer takes into account the requirements from application and may take into account other network policies received from A&M broker the services workflow. It defines atomic services' configuration and order, for a correct communication between them.

As a result, a composed service is obtained; defined by a succession of well-defined atomic services and other composed services. It also defines their order, and atomic services' interfaces, to a correct communication between them. Service Composer is also in charge of orchestrating already composed service, in order to obtain other more complex composed services. The Service composer decides if it has to create a new workflow or it can return a predefined template. Data requested and generated by those services in execution are carried by the Service Plane.

- Router. It manages the routing of discovery packets: where to forward packets based on the packets' semantic query. Besides, it also acts as a semantic resolver, concretizing source semantic queries at controller's request. Semantic resolving may need communication with the context manager both for local knowledge and for external knowledge (access to knowledge plane). The goal of this component is to lookup the possible paths in an end-to-end communication, taking into account semantic characteristics during the search. This component receives control messages from both the Controller (Control Plane) and the Application. These messages contain all the information needed to establish the end-to-end communication. If this component does not have the information needed to solve the semantic search, resends the request to another entity capable of resolving it. In addition, in a scenario where the lookup constraints are too soft, this component could suggest both the Application and the Controller, a more restrictive search. In turn, this component is the front-end to gather information related to the network's nodes and links. In summary, the router is in charge of the data-forwarding plane, only, while the controller takes care of the control plane, as an analogy to the SDN concept.

- Accounting & Management Broker (A&M Broker). The broker collects monitoring data with the purpose of accounting and network resource management. Hence, it directly collaborates with the accounting and management plane to distribute and manage collected data. This component is responsible of accounting the usage of the resources and Services based on the context states information provided by Service Monitor and Context Manager. This information is filtered and distributed to the Accounting and Management Plane with purposes of billing and managing.

  From this information, the A&M Broker can collaborate with global management services through the events or executed polices that allow to modify the service. Hence, this information may be also used for global network management. In this later case, management services are built on top of the architecture like any other control or data service, using the basic protocol for establishing connections and the service plane for transferring data, using its own service protocol on top of this connection to interchange orders and commands and to report results.

- Context Manager. It manages the context data (register of profiles and different context monitoring data) in itself plus its interchange with the knowledge plane (including elemental knowledge plane formed with its neighbors). It is in charge of managing context information: coordinate the gathering of raw context state information, analyze and store them (locally or globally). This information is classified in a local context with identifiers and related profiles. The context classification is divided in: node states, link states and user states. The objective of this classification is to know the nodes and links states and the user preferences. The characteristics of resources (links and nodes) can be dynamic or static, for example, the ratio of occupied resources is dynamic while the memory capacity is static. This state information is provided to the management components in the resources, control and accounting planes. So that these components can self-configure their services.

  Therefore, this component is responsible to manage autonomously the context information of the internal state of each node in order to be analyzed and classified, every node collects its own context information based on its available monitors context. Similarly, the application provides information about preferences and requirement of the service requested by user and also inform on its physical capabilities.

- Context Monitors. Different unrelated context monitors, i.e. environmental monitors, network, device, etc.

- Composed Services. Collection of atomic services orchestrated and composed in a single workflow to provide the required functionality for this composed service, according to policies dictated by the service composer.

- Shared States. Database collecting the state of each of the different atomic services in use for a certain composed service.

4.3.3 Planes

In addition to those elements, the architecture proposes 5 different planes, each one with its own domain of operation and scope. There is a Control plane in charge of managing the basic control protocol, plus a Service plane for managing service operation. Besides these, there are planes that are abstractions of different distributed information spaces or clouds. These are: the Knowledge plane, the Resource planes and the Accounting and Management plane.

4.3.4 Control Plane

The control plane is in charge of managing the execution of the basic protocol for discovering and negotiating services. In this sense, the control plane manages all the signaling information used to establish connections and negotiate service operation. Other planes may use the control plane to establish connections to the services in charge of managing the publication, subscription and distribution of its own managed data.

The management of the control plane is completely autonomic and flat-distributed as each node self-manages entirely its signaling via its controller component. Hence, there are no superior entities like super-controllers with global managing priorities or other types of high-level signaling. It must be noted that the control plane does not make decisions. The controller just signals and collects control data and then delegates decisions to other entities.

The control plane is in charge of the following procedures:

- Service discovery. Transport of semantic requests and discovery of available services.
- Route discovery. Available routes fulfilling the QoS SLA for discovered services.
  - Collection of available atomic services for this route.
  - Collection of available resources (QoS SLA fulfillment) for this route.
  - Reallocation of routes. This is done upon demand of the Service Manager.
- Service allocation. Signaling of the allocation of atomic services for the chosen service and route. The processes for the composition of services, and thus, the allocation decision are done by the Service Composer component.

4.3.5 Service Plane

The Service plane is in charge of managing the execution of composed services and the transference of data packets. So, all the data transfer passes through the Service Manager and then through the Service plane. Therefore, the result of execution of the different atomic services on data packets is managed here.

The Service plane, via the Service Manager and related entities, is in charge of the following procedures:

- Delivery of data packets. Transport and framing of data packets.
  - Execution of atomic services: obtains the transformations and headers resulting of the execution of the atomic services on a data packet (transferred and received).
- Request for new routes. The Service Manager requests new routes at its own discretion based on Service Monitor reports.
  - Route degradation (reported by the Service Monitor).
  - SLA violation (reported by Service Monitor).
- Service selection and composition. The Service Composer decides which service and route are the most suitable and then computes the best composition for that certain service and connection. This procedure is done at Controller's request.
- Service Monitoring. The Service Monitor monitors the execution of composed services through monitoring the state of their related atomic services.

4.3.6 Resource Plane

This plane stores the information required to discover resources during semantic routing. As explained above, this is the information that defines and identifies an accessible and localizable resource (service, content, user). So the Resource plane stores information on resources like: identification attributes of any nature: type, providers, locators, identifiers, names, etc.

It must be noted that resources like content and users are accessed via the chosen service acting as the interface to that certain resource.

4.3.7 Knowledge Plane

The Knowledge plane stores the information on network resources state, that is the information required to characterize the status of a resource; but not to define or identify the entity in itself (this is done in the Resource plane). This distinction is important because status information helps to characterize the situation of a resource and how it is best accessed whilst identity information is used to identify and locate resources. Therefore, the knowledge plane stores context status information like device capabilities, node characteristics, link conditions, user preferences, environmental conditions, etc.

4.3.8 Accounting and Management Plane

This plane stores reports on network resources utilization, service usage and fulfilment of SLA for purposes of billing and accounting. This information is obtained from the Context Manager and the Service Monitor and classified, filtered and distributed by the A&M Broker. Besides, this information may be also used for global network management in order to change nodes and services execution for optimize operation and respond to faults in the system. In this later case, management services are built on top of the architecture like any other control or data service, using the basic protocol for establishing connections and the service plane for transferring data, using its own service protocol on top of this connection to interchange orders and commands and to report results.

4.4 Conclusions

FN design should address most of the shortcomings of current Internet architecture. Specially, the proposed architecture should address the lack of ubiquity, pervasivity and context-awareness in the TCP/IP stack. From a service-oriented paradigm, this architecture design could solve/mitigate some of the issues and shortcomings of the current Internet architecture providing features like: context-awareness and dynamic adaptability during execution time; flexible allocation and execution of network functions; QoS, security and service discovery supported by the core of the architecture; routing based on semantic description of services (non-mandatory use of addresses, support for mobility, nomadism and multi-homing); QoS and resource availability integrated into routing and service discovery; enhanced user control; etc.

Some critical issues to consider are network architecture implantation and its impact on the sector's business models. This proposal can support with ease the traditional business models implanted in today's Internet, as functions may be allocated only at the end-nodes and the network can still remain as a "transparent pipe" where injected bits at source arrive unmodified to destination. But, also the business model may change drastically as services are placed in-network, diverting from the end-to-end principles, making the network a service provider and a third player in service provisioning (see section 5.5). Given the "operator-friendly" nature of our proposal, as it diverts from the end-to-end vision, it may provide operators with new business opportunities as they provide enhanced, differentiated and personalized services; but new modes of coordination and cooperation amongst operators will be required, since services may be placed between different domains, the billing, monitoring and assuring QoS parameters will become a more complex matter, requiring more coordination in peering relations.

Although many parts of the proposed architecture are still not covered and specified (such as a further exploration on syntax, rules and semantics, or mechanisms to address communications and service discovery across domains in a scalable way) the proposed architecture means a basic specification that enables to continue defining how services can work over it and how composite end services can be offered, accessed by users and delivered. Next chapter will explain the work done in this sense, one of the main contributions of the research exposed in this thesis.

5 Framework and mechanisms design for context-aware service composition over a Future Network architecture

As seen in previous chapters, in recent years multiple initiatives have contemplated restructuring the current Internet architecture in order to cope with the limitations of current Internet. Many of these solving strategies introduce novel clean-slate architectures that do not take TCP/IP as their groundwork.

As an example of them, the architecture presented in chapter 3 follows the principles on which SOA and SOC [Papazoglou et al., 2007] are grounded and offer some design guidelines that can pave the way to implement a new flexible architecture for the Future Networks (FN) where services could be easily deployed, discovered and reused. SOA promotes the usage of services to support the development of rapid, interoperable, evolvable, and massively distributed applications.

It is partially inspired in Role Based Architecture (RBA) [Braden et al., 2003], which presented a very disruptive idea on how to establish network communications. Their proposal was to avoid the existing layered structure of the TCP/IP stack and in so doing extract the roles or network functionalities which as a result could be interconnected without current layer restrictions. Decomposition of current protocol stack allows a certain granularity of network functionalities and enables their selection only as required. This is the first step to provide inherent cross-layering functions to a FN architecture. Such an architecture and mechanisms to build services on demand would allow avoiding numerous specific and complex cross-layer solutions. These solutions, although they provide some improvements (especially done in wireless networks), are too specific and cannot be reused for different purposes in varied situations [Srivastava et al., 2005]. An architecture that does not have rigid layers (and thus is not handicapped by the hierarchical restrictions on their interconnection) can solve the same problems without this drawback as functionalities can be requested just as it is required by a specific communication. In order to move service-oriented paradigms to the network level, the work done in this thesis proposes and validates the decomposition (RBA-based) and re-composition (SOA-based) of network functionalities so as to enable a native cross-layering solution whilst avoiding functional duplicities. These functionalities can be seen as in-network services and as such should be discovered and combined to supply seamless communications. This approach aims at improving the satisfaction of users' expectations by matching offered service characteristics with requirements and preferences previously determined by them.

5.1 Service composition challenges for Future Networks

As shown in section 3.5, several service-oriented approaches have been presented during last few years trying to solve current Internet deficiencies. However, these proposals must face several challenges [Rexford et al., 2014] to design an overall solution overcoming the current TCP/IP stack functional and performance deficiencies.

Furthermore, proposed service-oriented architectures for the FN consider that service communications are dynamically composed and located around the network according to requirements of the requested service and underlying network conditions.

The service composition model and methodology exposed in this chapter represents another of the main research outputs of this thesis. At the time it was published, it did not exist any publication presenting such research in the field of FN. Considering the network scenario abovementioned and focusing on the service-orientation, a set of requirements that FN must fulfill are identified, in order to support current and future services and technologies efficiently. In order to define those service composition mechanisms complementary that work over a FN service-oriented architecture, it must be designed to deal with the following challenges:

- **Dynamic Composition.** Network functions should be dynamically composed. Functions must be executed where and when needed, in order to guarantee efficient network operation. In addition, dynamic composition provides flexibility for allocating network functions along the nodes involved in service provisioning, according to network context information. This is the main challenge around which the others revolve. The following sections on this chapter detail the mechanisms and methodology approach to provide dynamic network services composition over a service-oriented FN architecture.
- **Network heterogeneity and attribute acquisition.** Heterogeneity of nodes and links add another level of complexity to the service composition process. If instances of a service are executed in nodes with different capabilities and network access links, every service instance should be evaluated individually. Attributes of a specific node may not be applied to another node. The composition process should be based on the attributes of services. However, extracting the complete and updated information of a service is extremely difficult. It sometimes requires a preliminary empiric process to extract information about how the inclusion of one service or another affects the final result in terms of QoS parameters (delay, error rate, etc.) that are relevant for a complete solution. IoT is a clear example of how this current-day heterogeneity is complex to manage. Mechanisms to monitor and manage highly heterogeneous networks of smart objects, handhelds, wearables, vehicles, etc. and provide real-time information of their different and dynamic capabilities and services will also be necessary. Research on a scalable web platform, a Web of Things (WoT) is also derived from this challenge and presented in Chapter 5.7.
- **Requester empowerment and costing model.** Requesters should have more control over the services that they want to consume. Empowering requester service selection must reflect this control, while offering all network available services matching the initial service requirements. This challenge is especially relevant for providing services as much adapted to the end users as possible. Linked to this challenge, an additional output was extracted sketching the basis for a cost framework in a service-oriented network (see section 5.5). Internet brings opportunities to create new business models according to novel services, applications and capabilities demanded by users. Hence, it is necessary to introduce innovative models for costing and pricing. FN architectures should also provide a transparent framework that permits service consumers and providers to interact, establish agreements and satisfy their goals.

5.2 Services framework

Services should not be fixed but dynamically composed where and when necessary, with respect to service requirements, network transfer capabilities and surrounding context in the user and the network environments. This proposal presents a service-oriented framework able to deal with functionalities at all levels (connectivity, transport, application) by considering the provided service and not the technology behind the functionality. All these network functionalities can be seen as services by means of suitable service-oriented abstractions. Herein, existing functionalities and protocols can be included –as well as linked or enriched–, and new functionalities could be easily introduced.

Subsets of services will be provided by nodes in the network and will be composed in order to create efficient end-to-end communications according to the requirements of the communication requesters. Hence, this task involves a context-aware service composition process. Depending on the type of requester, requirements may change. Typically, the basic requirements of a communication are expressed in terms of QoS parameters. However, requirements can also be other desired, or even mandatory, attributes in the communications such as: energy consumption, geographical location, price, etc.

Our work proposes a constrained-based routing (CBR) [Kuipers et al, 2002] performed hop-by-hop during the service discovery process. It establishes end-to-end virtual circuits between the requester and the provider of a demanded end service in a local domain.

Furthermore, this framework allows to empower the choices of requesters as it gives them the capacity to choose between different communications. Although it adds certain level of complexity, it will meet the socio-economical requirements of the current commercial Internet, where all participating stakeholders have different, and sometimes conflicting, interests. For instance, network providers would like to minimize the consumption of their network resources, while service providers would prefer to maximize the quality and competitive pricing of their services over the network [Clark et al., 2005].

5.2.1 Services definition

In order to follow the service-orientation and context-awareness principles proposed in section 4.2 and trying to tackle the challenges defined in section 5.1, a service should be represented as a self-contained, self-described modular block offering a specific functionality that can be published, located, and invoked across the network. This self-* definition should permit loose-coupling among services. Thus, services can be reused and chained to compose other services, hence offering more complex functionalities.

The proposed solution considers three basic components: Atomic Services (AS), Atomic Mechanisms (AM) and Composed Services (CS). ASs are individual functions or roles commonly used in networking protocols (e.g. acknowledgment, sequencing, flow control, etc.). These are well-defined and self-contained functions, used to establish communications to create CSs. AMs are specific implementations for each AS, which provide the desired atomic mechanism functionality. An AS can be implemented by different AMs (an extended example is shown later in section 5.2.2, Table 5.12). Finally, CSs are a combination of Atomic Services (AS) that work together to provide a more complex service. CS logics need to be specified in a Workflow (WF) (as can be seen in the example of Figure 5.1) to describe the composition and execution process of functionalities or ASs that could be offered by different implementations (AMs).

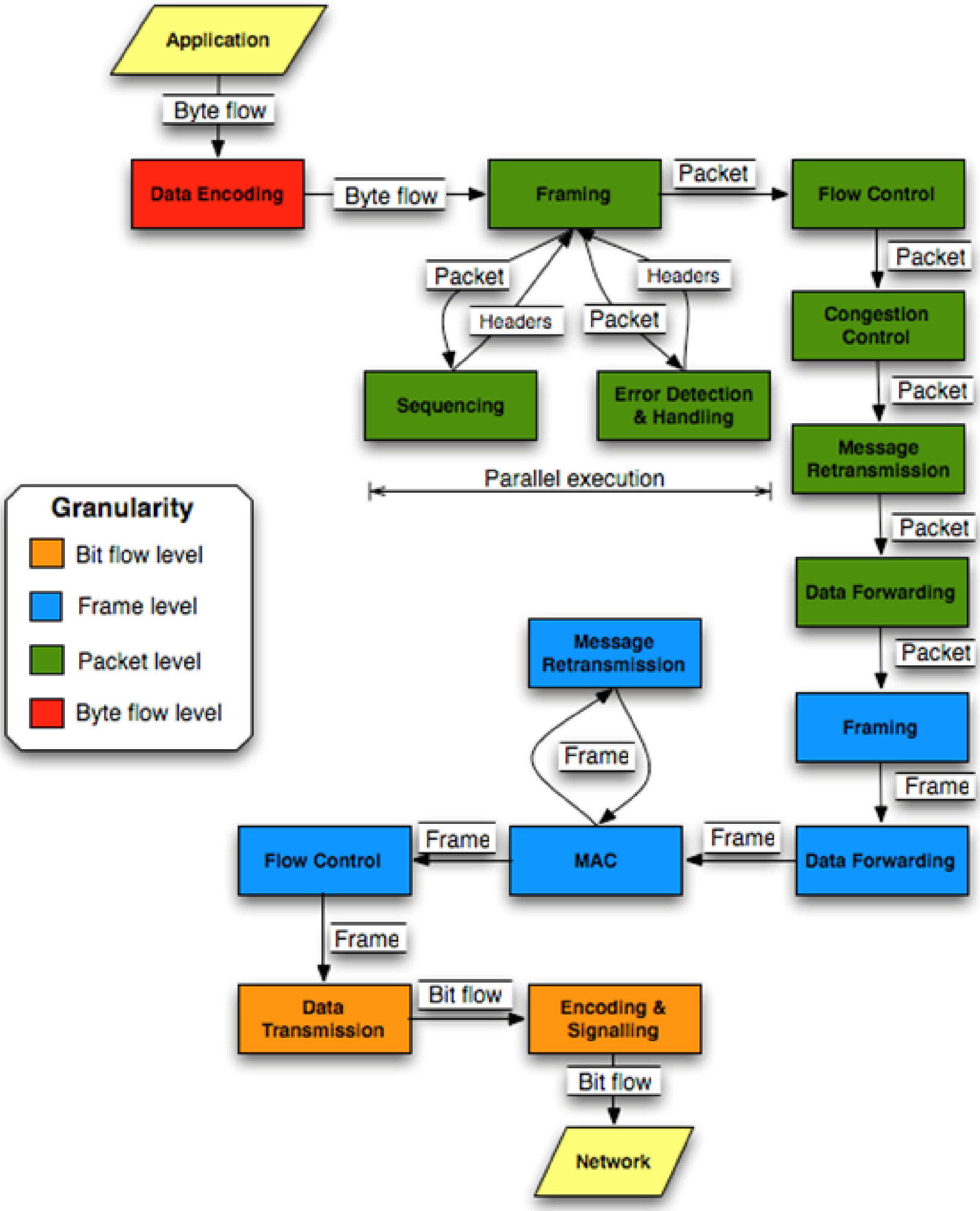


Figure 5.1 Workflow (WF) example

Based on previous service definitions seen in chapter 3 and the requirements and principles of service-oriented FN defined in chapter 4, the following definition for generic AS is considered containing – at least – the properties shown in Table 5.1.

On the one hand, proposed definition includes a set of semantic properties that specify the main description parameters of the AS. It defines name, description and properties, which can be used to semantically search or identify the AS. It also includes the requirements (possible QoS restrictions in general) and effects (to which the AS contributes). On the other hand, it includes a set of technical properties, such as the supported AMs, events, dependencies or incompatibilities on other AS, a cost (which can be included by a fixed number or an equation depending on some QoS parameters), and a locator (to help to uniquely identify the service).

Table 5.1 Atomic Service (AS) properties definition

Semantic Properties	
Name	Common name of the AS.
Description	Human-readable description of the AS.
Attributes	List of public attributes of a service. They can also include default values as a list of preferences and best practices of the AS.
Requirements	List of requirements. They differ from *Attributes* by adding some restrictive value that must be fulfilled.
Effects	List of effects [6] that an AS provides to ease the selection and composition processes.
Technical Properties	
Supported Events	List of events to which the AS should react. This list indicates which events should be handled, since they can change the service behavior or life-cycle.
Supported AMs	List of AMs (implementations) that are supported by the AS.
Commands	List of commands that an AS handles, specifying the operation name, its required input parameters and the returned data. This list of commands is specified in the service interface to interact with an application or other services.
Dependencies	List of other ASs to be executed and required preconditions.
Incompatibilities	List of those ASs not permitted to be used with this AS.
Cost	This parameter indicates the cost of the service in QoS terms, it should reflect the effect that the use of this service has for a QoS parameter (e.g. delay, %CPU used, etc.). However, these efforts may also be aggregated and mapped to monetary terms, in order to contribute with this metric to the service selection process.
Locator	Optionally, each service can include a locator field, usually derived from a node locator. This parameter should help to uniquely identify the service among the network and facilitate its discovery process.

Atomic mechanism (AM) definition (Table 5.2), analogously to the AS definition, includes name, description, attributes and requirements. It extends the requirements of the AS with the ones specific of the AM, overriding them in case the AM ones are more restrictive. Besides, it also may include a cost function as a technical parameter. It also inherits from the AS cost function, and overrides it.

Table 5.2 Atomic Mechanism (AM) properties definition

Semantic Properties	
Name	Common name of the AM.
Description	Human-readable description of the AM.
Attributes	List of specific public attributes of the mechanism.
Requirements	List of requirements. They differ from *Attributes* by adding some restrictive value that must be fulfilled.
Technical Properties	
Cost	This parameter indicates the cost of the specific implementation in QoS terms; it should overwrite cost reflected by the AS that it is included.

The definition of CS (see Table 5.3) separates semantic and technical properties too. Semantic include only the name, human-readable description and the end service name, while technical properties include dependencies on ASs (and where should they be placed), desired or optional ones, and supported events to which the CS should react.

In addition, it is considered that these service structures should not be linked to a unique specific modeling language. Thus, different languages may be used according to the device capabilities. For instance, high capacity devices can consider using more expressive languages such as those based on XML (e.g. WSDL, OWL-S, WSMO, etc.), while low capacity devices, such as sensors can use compressed formats like EXI (e.g. service discovery solutions such as Salutation, etc.).

Table 5.3 Composed Service (CS) properties definition

Semantic Properties	
Name	Common name of the CS.
Description	Human-readable description of the CS.
End Service	Target name of the CS (e.g. reliable video streaming service).
Technical Properties	
Required ASs	List of mandatory ASs for completing the CS. This list should include AMs (implementations) restrictions, to help identify where they must be placed within the WF. Additionally, each AS should also have a parameter indicating if it is required in all nodes from requester to end service provider, at least in a number of nodes, or only in end nodes.
Optional ASs	List of optional ASs that can be placed between some required ASs. They should indicate AMs restrictions as well.
Supported Events	List of events to which the CS should react.

Having these main pieces to work with, the composition process aims to select the best possible combination of ASs that enable to receive the end service. Therefore, the composition process consists in discovering, selecting, combining and allocating those ASs to be executed along the path from the node that request the service –Requester Node (RN)– to the node that provides the requested composed service –End Service Node (ESN)– going through different Intermediate Nodes (IN). In this context, a composition process orchestrated by the RN is proposed with the aim to empower the requester's control over the communication establishment. The RN will therefore be able to decide which discovered services best meet its requirements and preferences by centralizing the process of service selection, composition and allocation. Figure 5.2 exemplifies the composition process in which the RN demands for a composed end-service and three options are discovered providing different features along the path to the ESN. Those options are specified using different sets of services, which in turn can be implemented or deployed by different mechanisms.

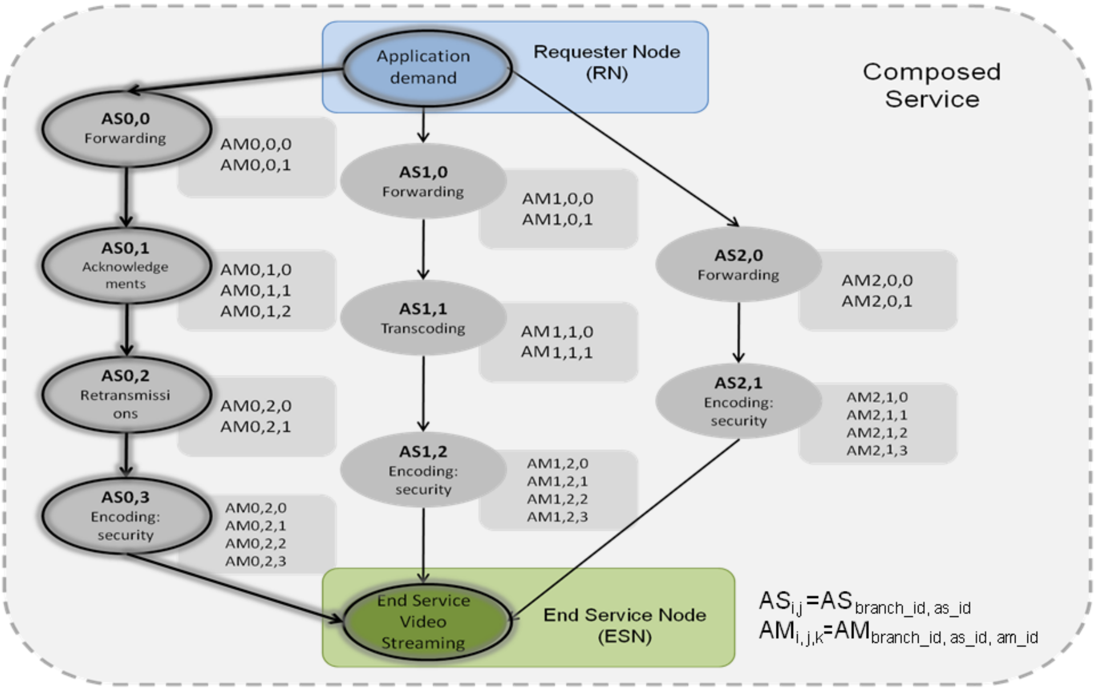


Figure 5.2 Service Composition example

5.2.2 Service discovery

This section presents a proposal and discusses the different aspects of the service discovery, a required process previously to the composition, which enables to identify and locate all the available atomic services and end services.

5.2.2.1 Semantic service identification

A service-oriented network paradigm should rely on the idea that users should be able to consume network services anytime, anywhere and anyhow (with any device and platform). This implies mechanisms to create, discover, negotiate and consume composed services in a flexible and context-aware way. Service consumers (users and other entities) may not know which device provides the desired composed service, they may even not know the name or the identifier/locator of the service or its provider; but they do know the characteristics of the service they want to consume. Thus, consumers must be able to describe the desired capabilities of the requested service and the network must be able to resolve if there is any service matching this description and provide them with a reachable locator. In this way, service consumers describe the desired service using semantic constructions like "I want a color printer close to building X with toner and paper" and probe travels the network with this semantic query. As the probe travel the network, traversing nodes match against their profiles if any of their services comply with the semantic service description of the probe. Since each node knows its own capabilities and which composed services it provides and their characteristics, which are described in node and service profile instances, they can match against the attributes of their service profiles if any of their composed services complies with the desired functionality. This process is called semantic service identification. It must be noted that attribute resolution in the semantic identification process is done in-route, during packet travelling and not as a previous phase before actually start sending packets.

In order to be feasibly implemented, network nodes must share a common knowledge base or ontology; that is a common attribute semantics and syntax. Although different ontologies may be supported, all nodes must support the minimum identification ontology. This basic ontology is designed to be minimalistic, in order to be supported by all kind of devices and platforms with enough ease (in terms of memory, computing power and energy), but still providing enough level of expressiveness and completeness when building semantic constructions. Hence, the nature of the relations expressed by each attribute (i.e., "is a", "has a", "belongs to", etc.) is inferred from its own definition. Attributes are defined for describing node capabilities (identifiers, CPU, memory, network interfaces, battery, etc.), temporal context characteristics (location, domain), atomic services characteristics (type, supported granularity, dependences, configuration parameters, etc.) and composed services characteristics (identifiers, I/O behavior, negotiation scheme, description, provider, etc.) Besides, in order to minimize the amount of information transferred, attribute syntax is dictionary-based. The following operators can be applied to attributes when constructing semantic descriptions:

- Logical operators: AND, OR, NOT
- Operators <, >, = for comparing attribute values
- Regular expressions

For example, a RN could demand for an IoT service in which it demands for sensitive Smart Metering data can specify a communication to the smart meter asking for "electric measurement from smart meter id=147 AND *end-to-end signaling AND (encryption=AES-256 OR AES-128) AND data integrity AND latency<10 seconds".* RN can specify the desired end-services, the atomic services required or desired, specific features, even if a specific implementation is required (identifying the atomic mechanism, such as *AES-256* for providing *encryption* AS).

5.2.2.2 Clustering and scalability

With such architecture definition, scalability would be limited in ad hoc environments as semantic flooding [Sanchez-Loro et al., 2009] and restricts network domain size, although with the benefit of not requiring any infrastructure support. Hence, semantic identification as described will not scale well outside ad hoc environments unless some infrastructure support is provided. Besides, clustering strategies are needed to make the semantic routing approach scalable. Therefore, domain-related attributes are used as labels or identifiers to distinguish between administrative domains and networks.

Although this approach does not redefine the domain concept of today's Internet it does pose some differences since the information used to define and label domains is not the same. Current Internet works with network prefixes and addresses, (and related names) which are tightly coupled with routing protocols and route binding; whilst, as stated in section 4.2, this proposal works with semantic attributes, which are not so constrained in their form and meaning as IP addresses and domain names and they are not coupled with the architecture in any form. Hence, any type of attribute (including addresses) is supported. For instance, in case of domain labelling, these domain identifiers could take any form: it could possibly be anything that allows the identification and/or location of a domain, therefore it could be a geographical (or even virtual) coordinate bounding the domain area, an arbitrary label, IP or other legacy addressing schemes without needing changes in architecture. However, using certain type(s) of attribute or another(s) may imply different routing casuistries (e.g. geographical routing for geographical coordinate).
In order to obtain high scalability in structured environments, service discovery cannot rely on semantic flooding at Internet-scale, but on special entities providing the required infrastructure/signaling services. Inspired in the Information-Centric Networks [Bari et al., 2012], if it has some infrastructure support, it can rely on global service/name resolution services (based on Multilevel Distributed Hashtable, for example [D'Ambrosio et al., 2011]) instead on the request flooding that will be by default.

Another important factor is to limit network domain size is the length of the session id tag, as it imposes a limit on the number of available concurrent connections in a domain. So, the following measures must be taken:

1. No semantic flooding, but semantic resolver. A semantic resolver resolves semantic descriptions and maps them to identifiers/locators. These resolution services should be designed to concretize semantic identification descriptions into more concise and directly routable identifiers and/or locators making extensive use of domain related attributes.
2. The range of session id flow tag is restricted to domain boundaries.
3. Tunnelling of session ids. Establishing domain-dependent hierarchies of session ids.
4. Well-known unique generic attributes for infrastructure services.

5.2.2.3 End Service discovery and negotiation process

The proposed end service discovery is the process of identifying the nodes that can provide the desired service to be accomplished. It includes as well as the ASs that may be required in the nodes of the communication path ranging from the RN to the ESN. This phase is divided into three steps as well. In the first step, requester requirements are mapped out to a service request. Secondly, the nodes that receive the query evaluate if they are able to provide the demanded service. Finally, context information is consulted to guarantee that the service can be provided under the required QoS parameters. Traditional web service discovery mechanisms are focused on enterprise communications and use heavy formats based on XML. Pervasive Computing Service Discovery Protocols (SDP) do not provide a unique and integrated solution able to work in a global heterogeneous network [Zhu et al., 2005]. Hereby it proposed to use an innovative negotiation protocol introduced in [Sanchez-Loro et al., 2011].

This negotiation protocol allows discovering a service in the network taking into account specific requirements established by the requester. In addition, it is simple enough to be able to work even in small and constrained environments such as ad-hoc sensor networks without infrastructure support. This semantic negotiation protocol integrates service discovery and service allocation. The service discovery process consists of searching in the network for services under certain conditions, by means of a Communication Request message (named Creq) (see Figure 5.3). In order to specify the criteria that will guide the search, this message should specify requester's service requirements in terms of[25]: (a) network performance parameters; (b) additional constraints (e.g. geographic, domain restrictions or specific attributes definition for certain services); and (c) required functionalities. Moreover, service requirements may be differentiated in two types of parameters: restrictive and non-restrictive. Restrictive parameters are those that are completely necessary to establish a communication. Non-restrictive ones represent non-mandatory parameters which allow optimizing the communication. Considering the inclusion of service requirements in the request, the following generic definition of a `Creq` message is proposed (1) where, `QoS_Requirements[i]`, `Context[j], and AS[k]`[26] correspond to lists of QoS requirements, context parameters and AS attributes respectively. QoS parameter values are divided into minimum and, optionally, optimum values. They specify desired metric values (flow QoS) and network functionalities (functional QoS). Furthermore, the `Creq` is tagged with a session id field to univocally distinguish that flow. Additionally, effects can be specified as desired high-level features for the communication, like security or reliability. Resources of a service such as a film provided by a streaming service can be specified as another extra parameter.

```
Creq = session_ID, End_Service_Name,
     QoS_Requirements[i](min,max),
     Context[j](constraints,preferences),
     AS[k](mandatory/optional)[Effects][Resources]
```
(1)

Using this type of requests, information about the capabilities of nodes is discovered through the network (see a representation of the service discovery process scenario in Figure 5.3 and the negotiation protocol dialog in Figure 5.5). The default operation performed in a node when receiving a request will be to evaluate if it can provide the service. If that is the case, the node answers with a Communication Response message (`Cresp`) that will be transferred through the reverse path.

One important issue when propagating a service request is to limit the scope of the request so as not to flood the entire network or propagate it indefinitely. Depending on the kind of network, several approaches can be adopted. In a network without infrastructure support like a sensor ad-hoc network, a Time To Live (TTL) counter can be set. In the event, the node that is not able to provide the end service will propagate the request to its neighbors until the requested service is found.

[25] In the first version of the negotiation protocol the attributes defined in section 5.2.2.1 were limited to the following subset.

[26] Where i, j and k specify the number of elements for each set of each set (QoS_Requirements, Context, and AS respectively).

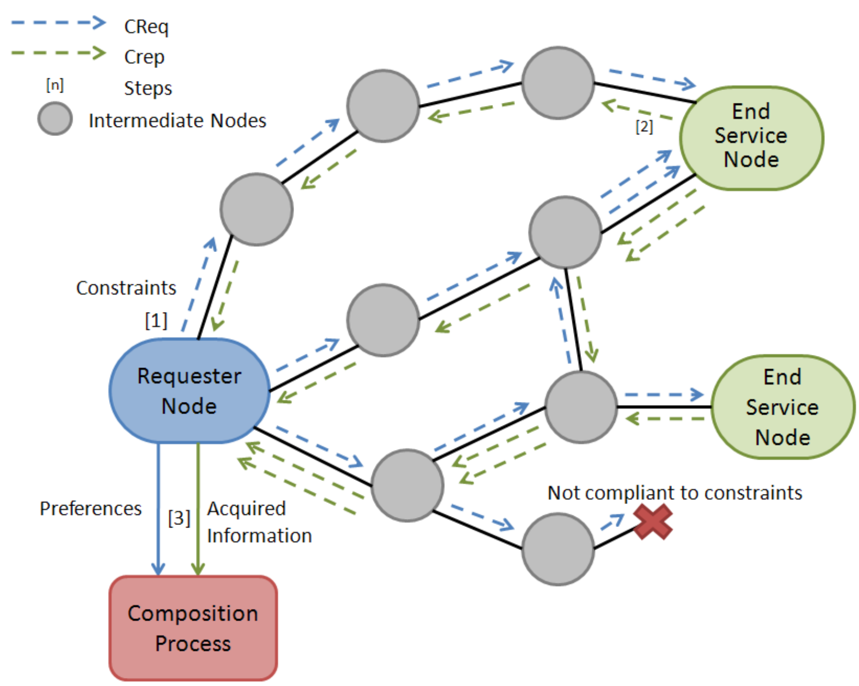


Figure 5.3 Service Discovery process

A clear benefit of performing this operation by each node is that it allows facing dynamic and frequent context changes. Moreover, neighbors' information could be exchanged by means of a Context Exchange Protocol (CEP) [Korpipää et al., 2004] [Kanter et al., 2009] able to notify the nodes in a network its presence, status and known nodes. This approach has the benefit of not requiring any infrastructure support, but, at the moment, scalability is limited to small environments due to the flooding methodology used in our experiments. Oppositely, if the network has infrastructure support, dedicated entities (e.g. distributed directory nodes) can provide the required infrastructure/signaling services within a domain -considering a domain as a set of nodes interconnected according to any criteria, such as autonomous systems, administrative domain, geography, topology scope, etc.- Furthermore, in order to obtain high scalability in structured environments, the measures proposed in [Sanchez-Loro et al., 2010] could be applied.

The scope of the work presented in this thesis was limited to the operation of the framework in a single domain. To verify the scalability of the proposed solution in a wider scale is a challenging issue that may be explored in future work. A possible implementation would be to use domain manager nodes in charge of relaying messages between different administrative domains and networks. These domain managers could be in charge of calculating the optimal path whilst considering different constraints. The main drawback is the complexity of this problem, that it is NP complete [Song et al., 2000]. However, a near optimal path can be computed taking into account the topology, capacity and context constraints by means of specific heuristics. For example, specifying a maximum admitted latency, end-to-end encryption capabilities or specific geolocation of the end service the scope of the search is limited and the composition time is also reduced. Hop-by-hop each IN validates that these conditions can be achieved, and truncate the search possibilities if not. This approach is similar to compact routing approaches like stretch-3 [Mooney, 2010] [Cowen, 2001].

In addition to this, the proposed solution is agnostic to the underlying technology, including networks and devices, thanks to the methodology explained previously, and well-defined service abstraction (Figure 5.4). Furthermore, the heterogeneity of the network can be faced by consulting the available context information (e.g. nodes capabilities, services or resources available).

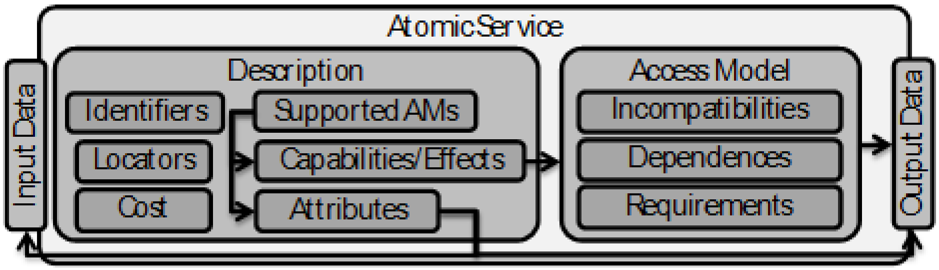


Figure 5.4 Service Generic Interface

Typically, QoS requirements specified in the request depend on dynamic conditions. Dynamism can be faced by means of reservation messages before establishing the communication or through admission control mechanisms. Then, what does it happen if a node fails? Resource reservation is a key feature of this approach at the moment of establishing the service, but it is not clear which is the optimal model for service adaptation. The storage of alternative compositions can be a methodology in order to reduce the adaptation time. A first generic model is proposed (later in this chapter and chapter 5.7, specific adaptations of this methodology are deployed into different application fields). It defines four possible reactions that can be taken depending on the level of adaptation required:

1. **Change an AM:** Identify the mechanism that provokes the failure and look up for a mechanism that can fix the service behavior and solve the problem.

2. **Change the WF within a node:** When a node is unable to find a possible swap of AMs that fix the behavior of the service, it should find a change in its workflow of ASs that fulfill the requirements once again. It must assure that its outbound and inbound interfaces with other nodes will not change, to avoid the reconfiguration of other nodes.

3. **Delegate a WF to another node:** When a node cannot find any possible workflow to provide the service achieving the demanded requirements. It should demand other nodes if they are capable of offering its workflow or equivalent one (a workflow that provide the same functionalities but with better behavior).

4. **Recalculate the composition:** As a last option, if any none of the previous solutions can fix the problem, RN is informed, and it initiates a new composition from scratch taking into account the updated context.

For example, considering the case of a real-time video streaming service over a noisy wireless channel. In this case, we do not mind data integrity, but we are looking forward the best QoE to the user. End-to-end latency, but specially jitter and packet losses are relevant in order to reconstruct the video image at the receiver side. We are assuming that the Composed Service is established, H.264 is agreed as the video format to be used at the sender and the receiver side and the video is running correctly at the receiver side, handling some losses with little impact in the visualization of the video. However, at some point, the losses of the channel increase that much that the receiver cannot reconstruct the 5%, 10% or 20% (depending how exigent do you define the user) of the video frames and they cannot be correctly displayed. How would the system react? It will look for a solution following the steps above specified. First, it will analyze if a potential change of an AM can provide a better performance of the whole CS. An example would be, the change of the video codification. Another codec, such a scalable video codec as H.264-SVC may be more robust to the channel losses. A change of AM is proposed the at the receiver, RN (who always leads the composition process), negotiated (in this case with the ESN) and done on-the-fly, trying to minimize the impact in the continuity of the service. After that change, if the problem is not solved (still a high percent of main video frames lost), we can look forward to add another AM or swap them in order to mitigate the losses. However, in some cases neither of those two steps would be possible for the receiver to support that video codec or additional AMs to be more robust can increase the latency too much. In that case, RN can try to delegate the video transcoding to a node network along the path and reconfigure its workflow accordingly. Finally, if none of these solutions is feasible to be executed dynamically without stopping the service, the RN can try to reconfigure the whole CS, looking for a new composition with better performance capabilities.

Following the service discovery process, a generic definition of a `Cresp` (2) is also proposed. The requester will receive *N* response messages that specify *N* candidate paths. Each of them contains the identifiers of the *m* nodes of the path, with the QoS capabilities, ASs and AMs that each node can offer. Moreover, in order to facilitate routing and session management, the destination includes a valid locator (by default) or a univocal identifier in the reply (when source explicitly requested it).

```
Cresp = Session_ID, Node[m](node_ID,
      QoS_Capabilities[j], AS[k], AM[l]),
      Locator, Identifier
```
(2)

, where `Node[m]` is a list of nodes in the end-to-end selected path, `node_ID` the identifier for each node, and `QoS_Capabilities[j], AS[k] and AM[l]` are lists of the corresponding parameters.

Finally, the last message considered in the negotiation protocol is the communication allocation message (`CAll`) (3), used to specify to each node which ASs and AMs must be executed. For this purpose, WF-based representations are used.

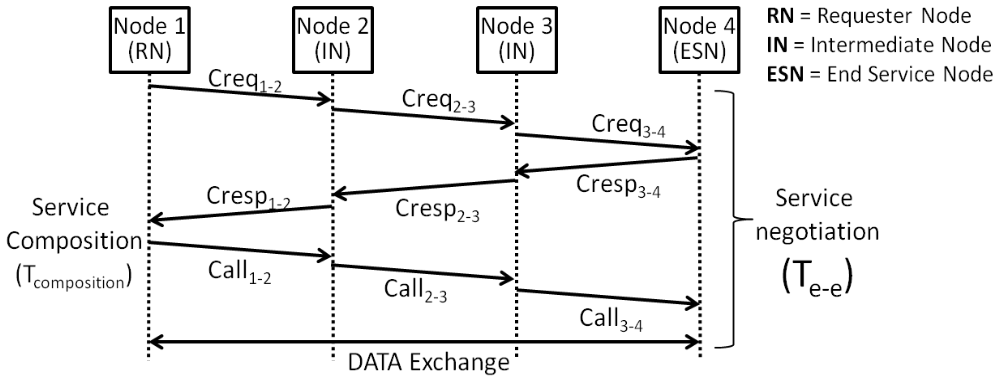
```
CAll = Session_ID, Node[m](node_ID, WF)
```
(3)

, `Node[m]` is the list of nodes in the path (the same as `Cresp`), `node_ID` the identifier and `WF` for the defined workflow for each node.

Figure 5.5 shows the whole 3-way handshake negotiation process. It can be seen how the `Creq` message goes from RN to the ESN searching for the desired composed service through two INs.

They also report the aggregate metrics that the route can assure. These include node capabilities like CPU, memory (buffering), battery and link characteristics like error probability and available bandwidth.

After receiving various `Cresp`, `Creq` originator starts the service composition process (explained in section 5.3) in order to construct the optimal service workflow and path that will be used to establish the service session. Once finished, it sends a unicast Communication services allocation (`CAll`) message indicating which functions (hop and segment) must perform each node in the service path. As `CAll` message is forwarded, each intermediate node erases its own orders, concealing them from following nodes. At the end, the destination node receives its instructions for end-to-end functions. At this point, service session is established and data can be transferred. Data forwarding is based on Session Identifier look-up to determine next hop, output interface and functions to be executed. Scheduling of the data transfer to achieve required QoS metrics is managed by the data forwarding atomic service.



Protocol Messages/ Node i- Node j	1-2	2-3	3-4
$Creq_{i\text{-}j}$	SessionID, EndServiceName, $QoS_{e\text{-}e}$	SessionID, EndServiceName, $QoS_{e\text{-}e}$-$QoS_{1\text{-}2}$	SessionID, EndServiceName, $QoS_{e\text{-}e}$-$QoS_{1\text{-}2}$-$QoS_{2\text{-}3}$
$Cresp_{i\text{-}j}$	SessionID, Locator Node4, Identifier Node4, QoS_4,QoS_3,QoS_2	SessionID, Locator Node4, Identifier Node4, QoS_4,QoS_3	SessionID, Locator Node4, Identifier Node4,QoS_4
$Call_{i\text{-}j}$	SessionID, $WF_2(AS_2[],AM_2[])$, $WF_3(AS_3[],AM_3[])$, $WF_4(AS_4[],AM_4[])$	SessionID, $WF_3(AS_3[],AM_3[])$, $WF_4(AS_4[],AM_4[])$	SessionID, $WF_4(AS_4[],AM_4[])$

Figure 5.5 Negotiation Process

5.3 Service composition process

In order to empower requesters' control over the communication, the RN will orchestrate the services. The requester will always decide which services will be chosen according to the discovered ones. This selection will be done by means of a selection of the ASs that the requester desires to receive considering requirements are met. To do that, it is important to have an expressive negotiation protocol, as described in previous section, to allow matching the requested services with the services available in each node until the end service.

The information discovered in the network is organized in graph structures where the nodes of the graph are the ASs or AMs of each node. However, the `Cresp` obtained from the discovery process can be directly mapped into a tree of disjoint branches like the structure shown in Figure 5.6. This way, the work divides the composition process to create a CS into four phases. To solve the service composition process, the process is divided into the following main sub-processes: Filtering, AM Scoring, AS Composition and Path Selection. The filtering phase provides a set of AS and AM that are scored in order to get the AM with better capabilities. After that, possible combinations of AS (and the selected AM for each AS) are built forming a tree graph. The best combination for each node is then scored by means of a graph search algorithm. The following subsections detail this process.

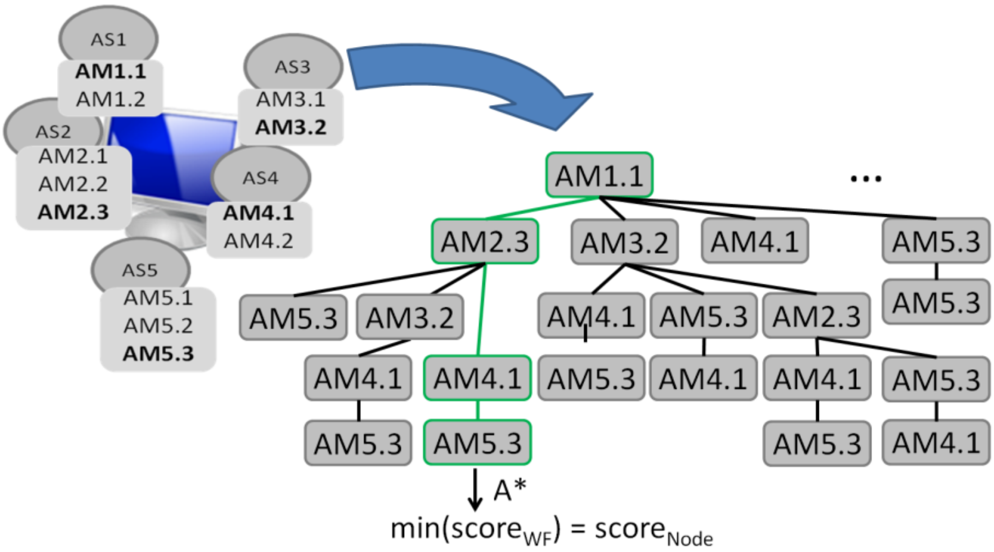


Figure 5.6 AS composition process scheme

5.3.1 Filtering

This phase consists in filtering all received `Cresp` messages according to the requirements specified by the RN. A range of possible costs acceptable by the user can be set up when specifying a constraint or a preference in the `Creq`. These filters are represented by specific rules, which can be solved by means of a Constraints Satisfaction Problem (CSP) method [Kumar, 1992]. The filtering process is inherent in service discovery. However, a validation phase is applied in the requester side, once all the communication responses are received, to assess that all QoS requirements along the end-to-end path are achieved.

5.3.2 AM Scoring

During this phase, the AM that implements each AS is selected according to specific scoring functions. This score considers different specific attributes related to the AS, for instance the QoS parameters that they can provide and the priorities of the RN. For each AS, a set of possible AMs are scored and the best one is selected. In its preliminary implementation (shown in section 5.6.1), the AM scoring considers each AM as an isolated process, regardless AMs interconnection relationships. I propose to use a generic weighting function (eq.5.1) to score the AMs, where weights may vary depending on the preferences introduced by requesters. Herein, it is possible to define trade-offs between different parameters such as the quality provided by the network, requirements and the price to pay for a service. However, scoring functions can be defined for each AS in order to consider specific requirements as shown in [Gonzalez et al., 2011], where a score metric for audiovisual contents is proposed.

$$Score_{AM} = \sum_{i=0}^{n} Parameter_i \cdot Weight_i \qquad \text{(eq.5.1)}$$

5.3.3 AS Composition

Usually, an operation can be offered by different combinations of ASs. For instance, a reliable service can be provided by means of acknowledgment, error detection and retransmission functions, or by applying forward error correction. Depending on the combinations, the provided QoS may change. Thus, those best suited to satisfy requested priorities will be chosen. Note that the RN composes the services per each node in the path (RN, INs and ESN), and generates the corresponding WFs. Considering the described architecture, once all services are discovered, RN should be able to create a tree graph (Figure 5.6) with all discovered services at each node. Our solution evaluates first the different dependencies at each hop as well as the input and output attributes among ASs, and concatenates those that could be executed within a node in order to satisfy a communication goal. Then, the best branch of services is selected and the final CS is generated for each node. Depending on the level of granularity in service definition and the grade of accuracy when composing a service, service composition can be based on selecting predefined services compositions (e.g. manually configured or calculated once and stored in specific repositories) or fully calculated considering all the possible combinations and, thus, obtaining the best possible solution available. Ideally, it would be necessary to find a trade-off in order to find a feasible solution in a reasonable time. In completely predefined ones the path should be known beforehand.

Additionally, composition could be predefined only at some levels. For example, if a link used previously does not exist any more but an equivalent link is found and it does not change the characteristics of the communication, the same workflow can be reused, especially in the case of edge nodes. Section 5.6.3 describes a first proof of concept implementation where used predefined sets of compositions to enable fast service invocation. Then, different graph traversal and path search algorithms were considered (such as Depth first [Tarjan, 1972], Breadth first [Zhou et al., 2006], Greedy search [Galinier et al., 2006], Hill climbing [Johnson et al., 2002], etc.) to observe the behavior of the system for more accurate compositions. The implementation of the service composer component was finally undertaken by the A* algorithm [Hart et al., 1968] for searching the best solution taking into account the nodes with best score. A* was finally deployed due that it offered a trade-off of simplicity (it was considered the easiest to be adapted for the composition process, because it was easy to include the composition filtering) and performance (it required less computational time to resolve compositions compared to other greedy solutions). However, it has some scalability inefficiences both in terms of computational time and memory space. Alternative algorithms implying trade-offs can be analyzed in the future work. For example, variations of A* such as

IDA* (Iterative Deepening A*) were not taken into account at the moment of implementation (because of time constraints in TARIFA project development) and they can reduce computational costs while maintaining performance in some cases [Korf et al., 2001].

5.3.4 Path Selection

Paths are considered as sequences of nodes containing different ASs, which are capable of reaching the demanded end-service from RN. Hence, they are defined from RN to ESN and can be represented as a tree graph structure (Figure 5.7).

When each node is scored, the path is selected using graph theory search strategies to determine the most cost-effective cost path, or branch in this case. Depending on the preferences of the requester, the selected path can offer, for example, the best trade-off between different parameters. For instance, selection could be done according to the lowest delay path, lowest cost per transmitted bit, or one considering a trade-off between both criteria by means of a weighted scoring function. In the case of Figure 5.7, different paths are represented; the score of the different nodes of the path are specified and the left branch is selected because it is the branch that sums the lowest score (we are targeting the branch that minimizes the scoring function).

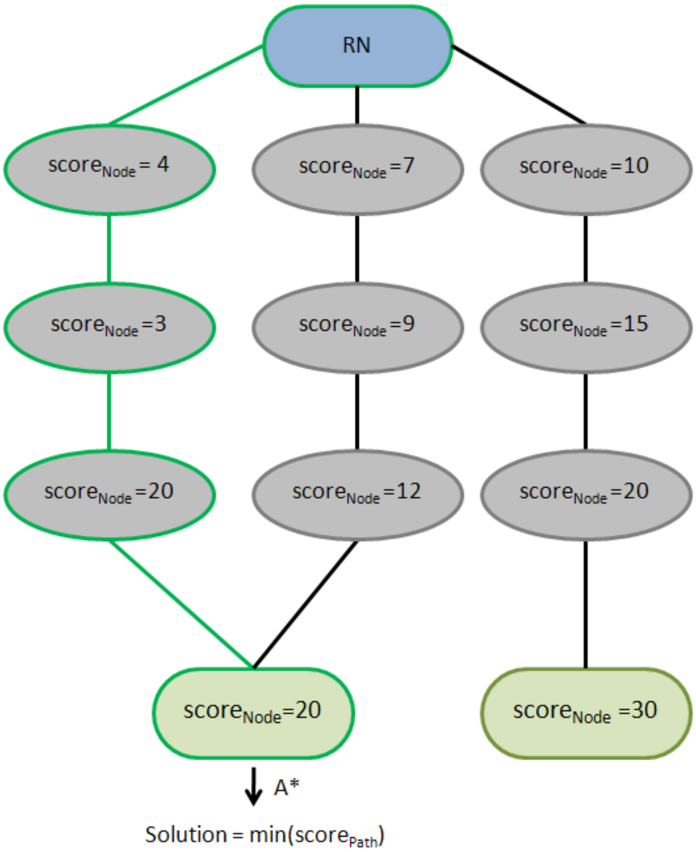


Figure 5.7 Path selection process

Additionally, if a cost restriction specified by the RN is at stake, it would be necessary to discard those paths with a higher cost than the demanded restriction (eq. 5.2).

$$\sum_{i=0}^{n}(W_N)_i \leq MaxCost \qquad \text{(eq. 5.2)}$$

This will discard those paths with a cost higher than the expected. When each node is scored ($Score_{WF}$ = $Score_{node}$), a path is selected using, for instance, search strategies used in graph theory. As an example, the RN will apply A* algorithm to find the network path that will provide the CS.

5.4 Service composition over FN architecture

The process described in previous section 5.3 was part of the whole operation of the service-oriented FN architecture defined in chapter 4. Figure 5.8 shows the whole process from the user semantically demanding for a service to the execution of the whole services (coordinated ASs) to enable the provision of the end service with the restrictions or preferences demanded by the user.

In the figure, step 1 describes the semantic request reception from a user (or another smart object, for example). It is translated to specific QoS attributes and requirements, when possible. Others are included as well as restrictions (e. g. geographical). Step 2 does that job, creating the `Creq` message and sending it or looking for the end service in a service name resolution system (if available). The Router is the module of the TARIFA node that will send this message through the network, gathering all the information of the nodes and starting the composition process described in this chapter. After that the Controler entity of the TARIFA node sends a Call message to allocate all the WFs in each node of the path (from RN to ESN). Finally, they start to execute, at the same time that each node activates their Service Monitor and reports to the A&M Broker. Together, with context information gathered and managed by the Context Manager, reconfigurations of the Service Composition can be considered.

5.5 Costing Framework

Future Internet Socio Economics (FISE) [Courcoubetis et al., 2009] consider a number of research approaches such as: *business innovation, connectivity, confidentiality, costing, drivers, governance, information delivery, emerging communication paradigms (e.g. Internet of Things, Internet of Services, etc.), infrastructure (e.g. network virtualization), new business, overlay applications, security, Quality of Interconnection (QoI), regulation, service provisioning, social welfare or Internet usage* [SMOOTHIT] [Burness et al., 2009] [Stiller, 2009] [Sallent et al., 2010]. All these research lines converge in the need to meet requester needs, based on its preferences according to requested data, with a desired level of QoS, and the amount of money a user wants to pay for a concrete service or content. This approach gives to users enough freedom to choose among different services and contents in order to fulfill specific technical and non-technical goals while maximizing the satisfaction of users. Service Composition requires a number of services in order to accomplish the requester requirements. Besides technical aspects, it is considered that every AS in a CS has a cost, which also entails non-technical arrangements derived from Socio Economical aspects (e.g. price, value, an hypothesis or business models). Thus, the service provisioning process carries a socioeconomic implication to be addressed between providers and requesters.

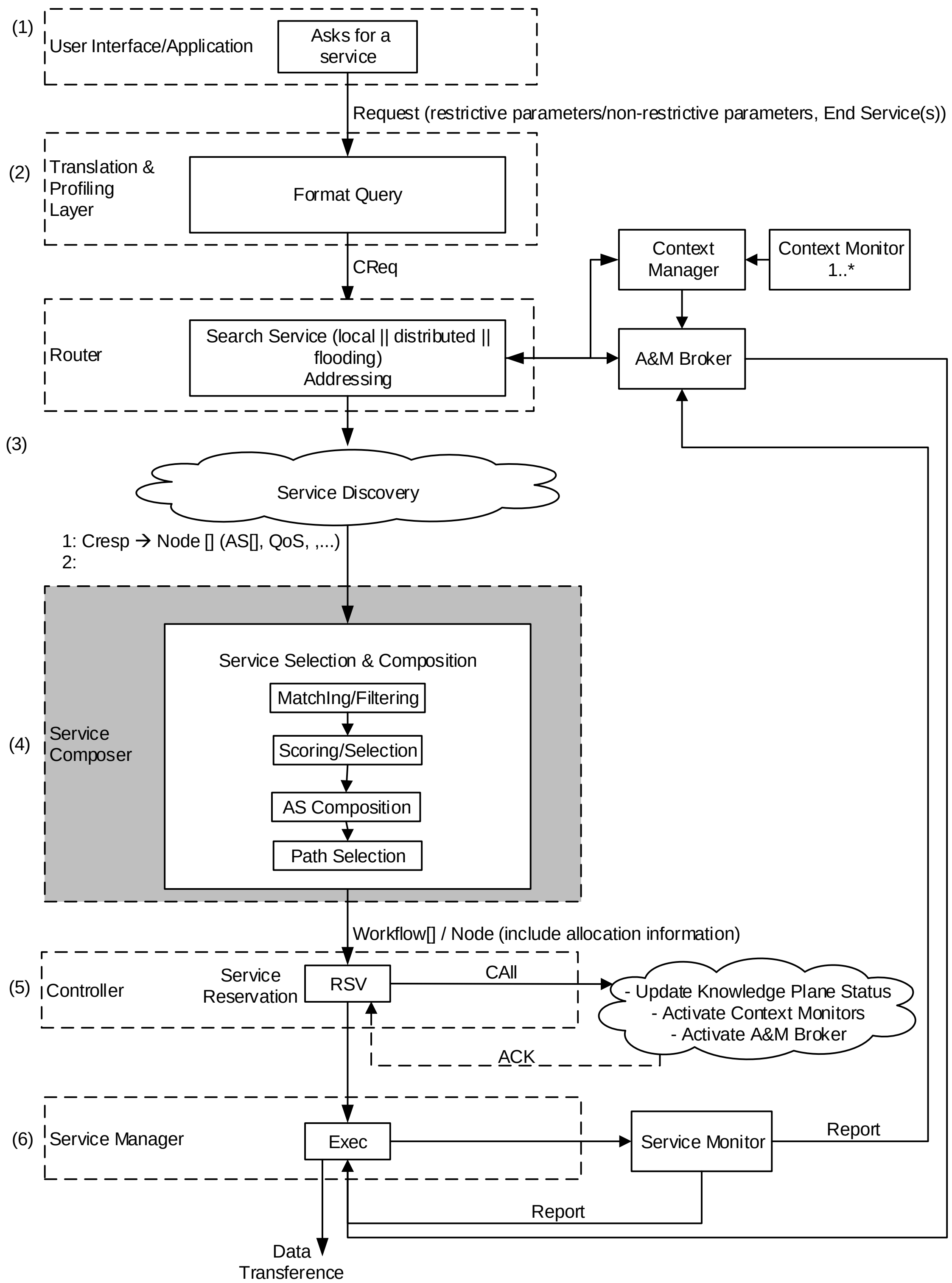


Figure 5.8 Service composer module in TARIFA FN architecture

Costs of services in the network are related to expenses which can be derived from production, infrastructure amortization, energy consumption, research, management, etc. All these expenses have a result that affects the level of Quality of Service (QoS), which is considered as a function of functional and non-functional service quality attributes, such as service metering and cost, performance metrics, security attributes, reliability, scalability, and availability [Papazoglou et al., 2007]. This has consequences in QoS parameters such as dedicated bandwidth, controlled jitter and latency, and improved loss characteristics [Guérin et al., 1999].

The price of services also depends on the set of assumptions made by providers in order to obtain a balance between cost and benefit. That is the Business Model (BM) that takes into account a series of components like price, convenience, commodity-plus, experience, organization planning, distribution channels, intermediaries, trust chains, and innovation [Linder, 2000]. Depending on provider interests in social and/or economic terms, it will make a hypothesis to establish a value factor in order to set a price on each of the components.

Many assumptions can be considered to make a hypothesis. Some examples are: level of use of the service and network, agreements between networks (e.g. two or more networks may be considered as a unique network), implementation costs of each network, management and maintenance costs of the network, regulation, etc. All these criteria are used to define a dynamic function to calculate a price for each composed service.

To explain the operation of the proposed framework, an Internet model where End Services can be offered by different providers in different domains and can be reached by different paths is used. This model (Figure 5.9) is different from current Internet hierarchical environment.

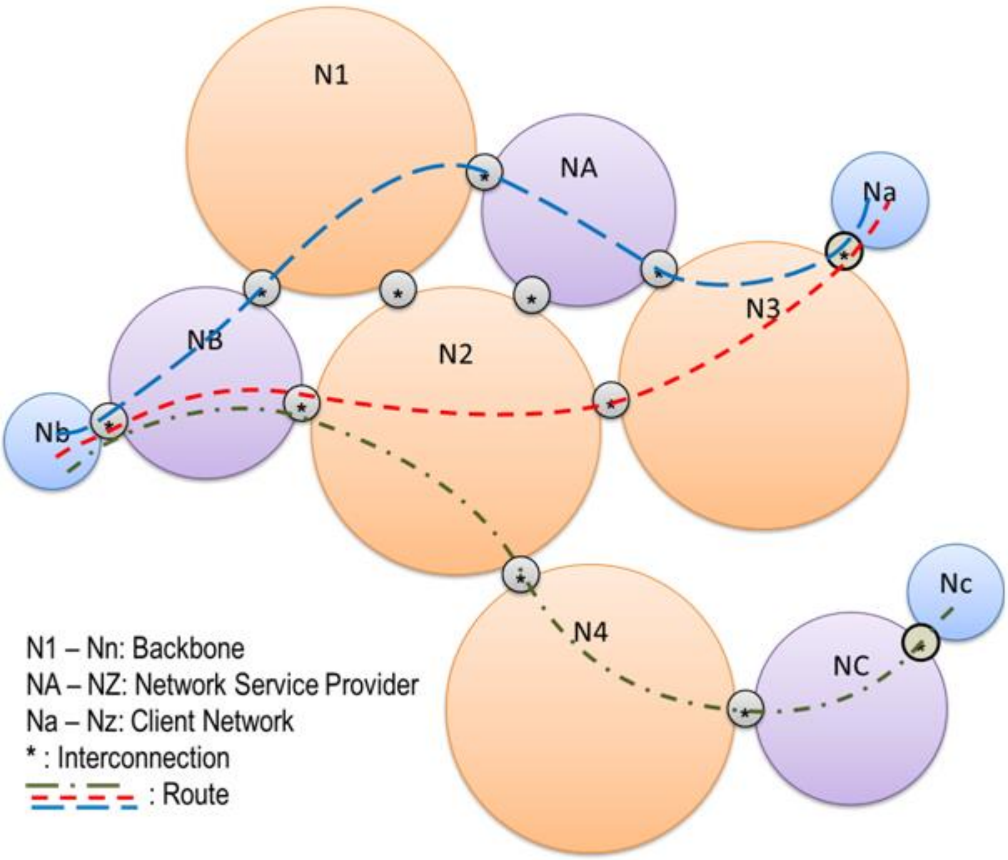


Figure 5.9 End Service Nodes (ESN) can be reached via different routes

However, different models can coexist in the future. Thus, FN architectures should guarantee compatibility among them and also backward compatibility with current Internet.

In this scenario, the requester is capable of searching an end service over the network and choosing a combination of available ASs along the path. Service discovery process for an end service will return as a result a set of routes compliant with requester requirements. Each route has different characteristics and, consequently, different costs (Figure 5.9)

It is necessary to define a socioeconomic model suitable for the proposed scenario. Therefore, in this work, a framework able to calculate the cost for a composed service is introduced. Based on service composition mechanics and granularity, this costing framework is divided into three main components: Atomic Cost, Node Cost, and Composed Service Cost (Figure 5.10).

- **Atomic Cost** (W_{AS}) refers to the cost of an Atomic Service (eq.5.3). It is an objective metric, which can be calculated according to the resources consumed by an AS into a node. The cost directly depends on the effect of the AS into the Quality of Service (QoS) parameters associated to its execution.

$$W_{AS} = K_R \qquad \text{(eq. 5.3)}$$

, where K_R is a constant value relative to each AS.

- **Node Cost** (W_N) depends on the cost of the set of ASs (eq.5.4) executed in a node. In addition, it depends on a subjective parameter (S_Q) which adds value to the execution of a specific AS. S_Q is a value obtained from a specific function for each AS and depends on the desired subjective metrics, for instance, the relative cost of demanding a specific resource at a precise time. The product of W_{AS} and S_Q for each AS is called ASCost.

$$W_N = \sum_{i=1}^{n} \left[(w_{AS})_i \times (s_Q)_i \right] = \sum_{i=1}^{n} [ASCost_i] \qquad \text{(eq 5.4)}$$

- **Composed Service Cost (W_{CS})** is the aggregated cost of each contributing node (W_N) in a route (eq. 5.5). From the network service provider point of view, it should be interesting to identify the cost of its network (W_{CS_Nx}) in providing a CS. It could be calculated as the aggregation of costs related to its internal nodes. However, RN, when demanding a service, will obtain W_{CS} as the addition of all WN participating in the CS. It is the same as the sum of the W_{CS_Nw} of each provider in a route from RN to ESN.

$$w_{cs} = \sum_{i=1}^{n} (w_N)_i \qquad \text{(eq. 5.5)}$$

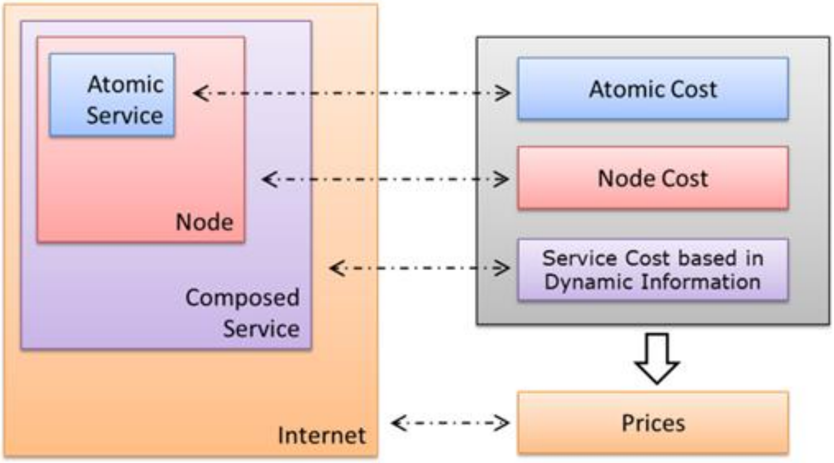


Figure 5.10. Costing framework

Note that the cost framework defined is focused on atomic services. This fine granularity is useful for creating a dynamic protocol stack, consequence of selecting those required services to establish a communication. However, it is also possible to define a coarser granularity to apply the cost parameter to sets of services or CSs, which are used very often together. Hence, avoiding to associate a cost to tiny individual atomic services that in a practical system are very difficult to quantify in pricing terms. For example, for a node performing forwarding of packets in a network (e.g. a router), the cost parameter can be applied to the combination of the services involved in this operation: encoding of data signals to transmit over a network, data framing and data forwarding.

5.6 Framework tests

A Java-based prototype was developed first in order to validate as fast as possible the basic selection and composition algorithm proposed. In addition, this development included a GUI based on an applet in order to graphically represent the steps performed by the algorithm until the final solution is provided. The main goal of this tool was to simulate a composition algorithm proof of concept separately to the FN architecture proposed in Chapter 4. It enabled to visually analyze and verify its behavior in the selection and combination of services without taking into account the discovery protocol or other required mechanisms to complete the service composition. It allowed to test the composition methodology and algorithms, and guide the selection of which components are needed to be integrated into the architecture and the whole platform of TARIFA project [TARIFA] [Sallent et al., 2010].

Once the proposed composition algorithm was tested and validated independently through this prototype, another one was generated based in C, which corresponded to the Service Composer module for the TARIFA core platform. This other module and its evaluation will be seen below in Section 5.6.2.

Different tests were performed to evaluate the correctness of the different implementations. They allowed to observe the performance of the developed components and the decision making process which generates the final composition of ASs, selection of AMs and the service allocation in different nodes. This section describes those prototypes and the validation results gathered from the performed tests.

5.6.1 Composition algorithm prototype

5.6.1.1 Description

In order to perform a quick validation of the proposed composition process, an algorithm was designed taking as a basis the implementation of the A* algorithm. The adaptation was called A* TARIFA algorithm, in reference to the project that funded this research. To test it, a simple Java-based applet was firstly used (Figure 5.11) to speed up the development of a first approach that enables to assess graph search algorithms for the composition process.

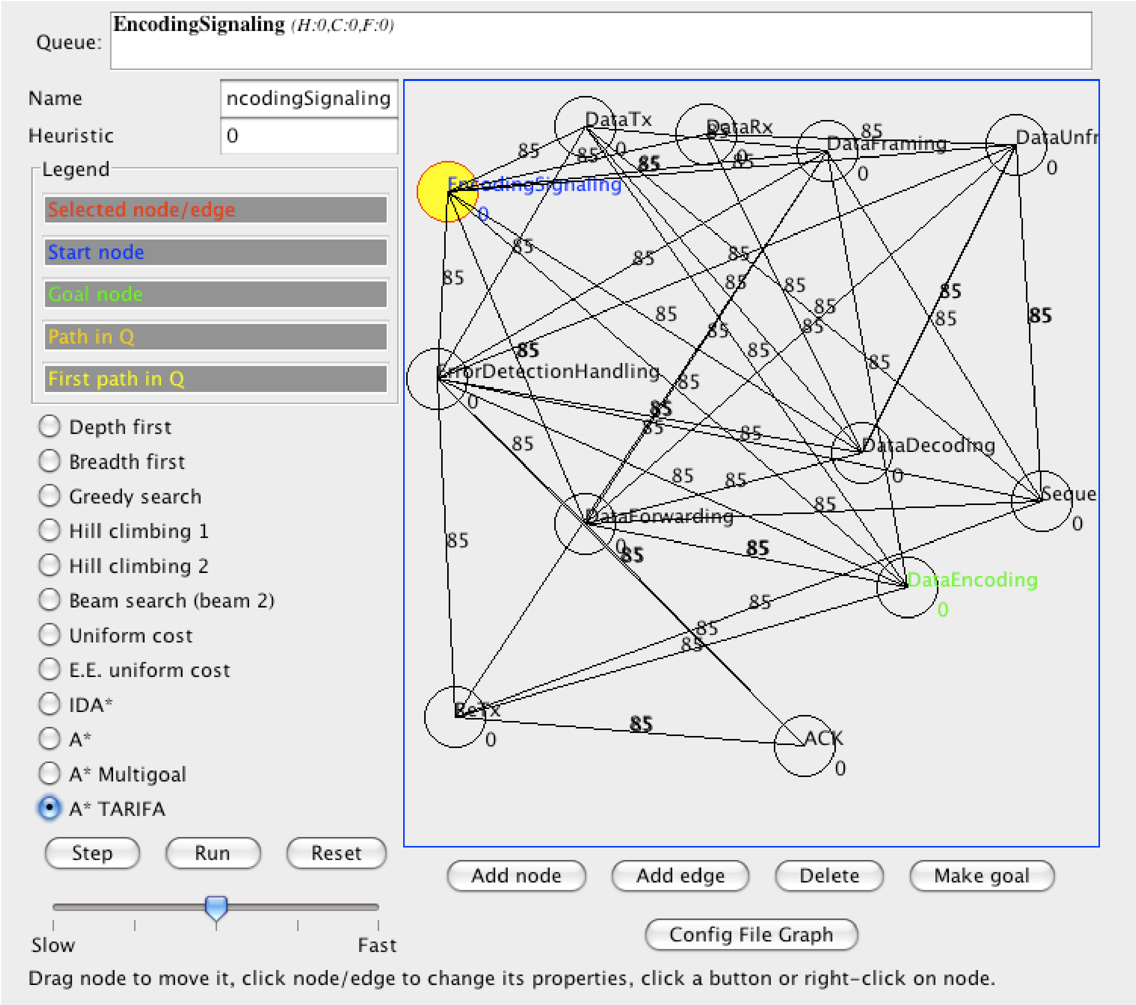


Figure 5.11 Java Applet GUI

It was mainly done with the objective of getting knowledge about graph and path search algorithms applied to the service composition case. Although the computational cost and the time required by it to orchestrate the composed service and offer the different solutions are high (due to the usage of Java), it offered an open, platform-independent and powerful platform that can graphically represent the composition process, and enabled a slight but fast and visual comparison with alternative search algorithms such as depth-first, breadth-first, greedy search, hill climbing, A*.

This tool also helped to take the decision of using A* as a basis of the composition algorithm, empirically testing the modification of other algorithms to adapt to the use case of network service composition and assessing that the time required was relatively higher than A.

Figure 5.11 shows the graphical interface of the Java applet used. It sequentially loads the different node, AS and AM properties provided in a set of folders and files structures defined below (Figure 5.13). Afterwards, it applies the graph search algorithm selected from the list at the right, and starts showing (through the graph render and the AS queue specification) how the algorithm build the chain of AS per each node.

The A* algorithm code was taken as a basis and adapted in A*TARIFA to support the input parameters and filters required for the service composition case. In addition, the scoring function required was added in order to calculate cost and heuristic parameters. Finally, an extra module was created to save the results of the simulated composition process, and extracting the statistics of each composition process demanded.

5.6.1.2 Prototype specifications

This section specifies the Java-based prototype development, including the hardware and software used for the simulation tests. All simulations were done using two different machines with the following configurations:

Table 5.4 Deployment machine A

CPU	Intel Core 2 Duo
Processor speed	2.4 GHz
Number of processors	1
Total number of cores	2
L2 Cache	3MB
RAM Memory	2GB
FSB Speed	800MHz
Operating System	MAC OS X 10.5.8
JAVA runtime version	SE, build 1.5.0_28-b04-382-9M3326

Table 5.5 Deployment machine B

CPU	Core 2 Quad Q6600
Processor speed	2.4 GHz
Number of processors	1
Total number of cores	4
L2 Cache	4MB
RAM Memory	2GiB
Operating System	Linux Ubuntu 10.10 maverick (2.6.35-28-generic)
JAVA runtime version	SE, v1.5

The prototype was coded using an integrated development environment, Eclipse. A Java project was created with the following structure (see Figure 5.12):

- *i2cat.tarifa.algorithms.old*: this package contains the implementation of different graph search algorithms that were implemented in the applet and are no longer used in the TARIFA generated prototypes. Some of them are: depth first, breadth first, greedy search, hill climbing, beam search, IDA*, A*.
- *i2cat.tarifa.log*: contains a *Logger.java* class to create log files (monitoring the composition process).
- *i2cat.tarifa.model*: contains the object representation of an AS, an AM and Constants used in the generated prototype. Some classes of this package are *AtomicService.java* or *AtomicMechanism.java*.
- *i2cat.tarifa.properties*: Formed by the *PropertiesConfigurator.java* class, which contains the necessary tools to process properties files, used for configuring the prototype (getting the properties names, values and weights from associated property files).
- *i2cat.tarifa.statistics*: contains the necessary classes to generate measurements in the application. These measurements are used to generate statistics (e.g. processing delay for composing services into the RN).
- *i2cat.tarifa.utils.costfunction*: implementation of the scoring function based on weights. In this package a single generic cost function is built. However, it is prepared for the addition of classes derived from this one that implement more complex cost functions.
- *i2cat.tarifa.validation*: implementation of the A* TARIFA algorithm.
- *i2cat.tarifa.validation.reducedFramework*: A reduced set of classes of the validation package, just maintaining the Java classes strictly necessary to implement the A* TARIFA algorithm[27].

[27] Graphic interface is disabled executing the program with this reduced framework, but the composition process was performed faster. It was done also thinking on the subsequent implementation explained in section 5.6.2.

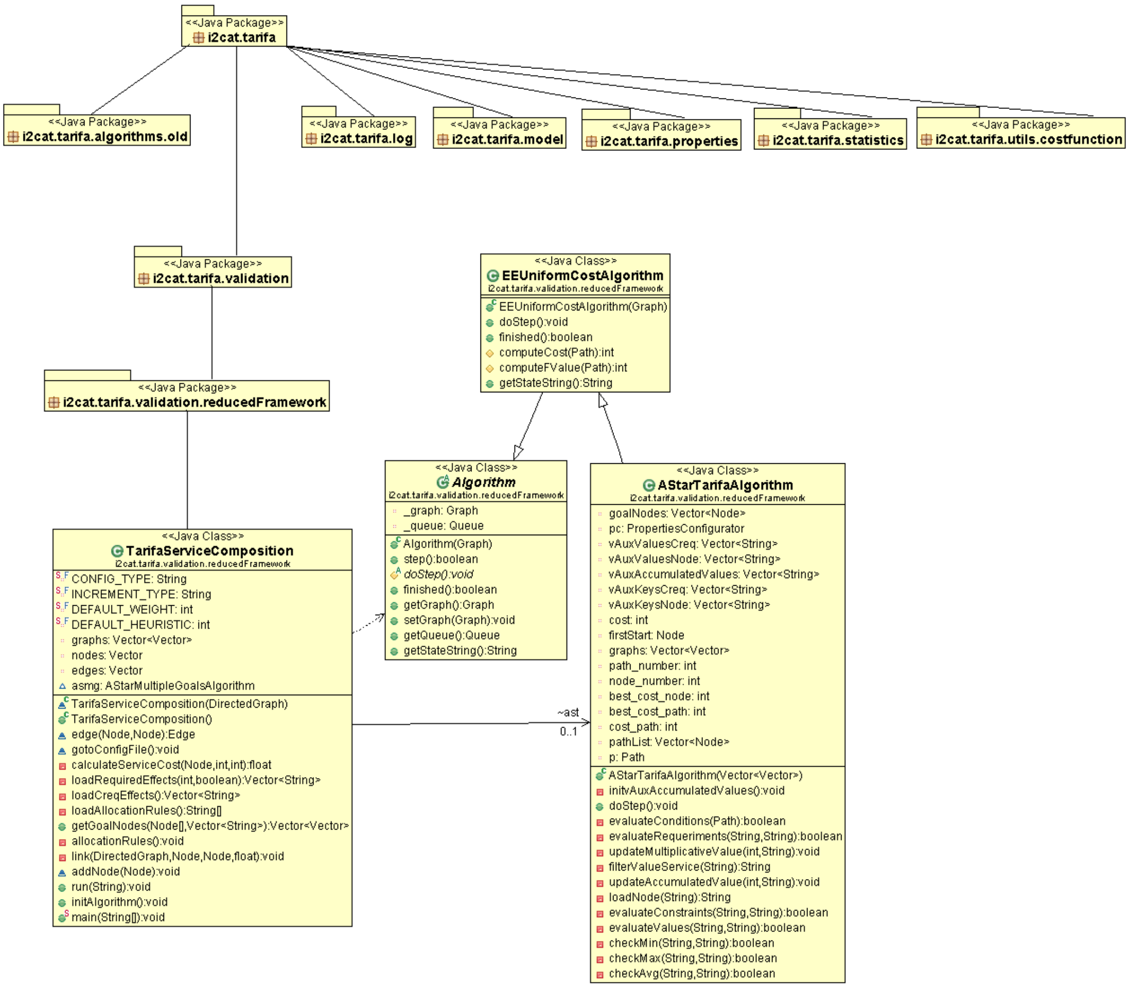


Figure 5.12 UML class diagram of the Java code

Some brief details of the main classes developed in this package are shown in the following tables[28].

Table 5.6 TarifaServiceComposition (Java class description)

TarifaServiceComposition.java	
This is the main class of the project. It starts the stand alone execution of the A* TARIFA algorithm and initializes the Java-based prototype.	
Functions and procedures	**Details**
TarifaServiceComposition (DirectedGraph g)	Constructor of the class. It initializes the graph to be evaluated, *g*.
void gotoConfigFile()	Loads from the use case configuration file the properties of the ASs that are in the graph. This includes loading graph nodes and edges properties.
public void initAlgorithm()	Initializes the algorithm variables.
public Vector <Vector> getGoalNodes(Node[] AS,	Loads graph goal nodes from property files. Concretely,

[28] The emphasis is given to the classes of this package because these ones are the ones strictly related to the composition process. Other packages are only necessary to prepare the framework (e.g. model or properties) or to provide feedback (e.g. log, statistics).

Vector<String> reqEffects)	goal nodes are a set of ASs defined by the effects. In addition, this class calculates and combines all the combinations of ASs according to the required effects. Finally, it prunes repeated services from each solution and repeated solutions, to avoid redundancies.
private float calculateServiceCost(Node AS, int path, int node)	Calculates the score of an AS implemented by a specific AM.
private Vector<String> loadRequiredEffects(int n, boolean endServiceNode)	Loads “required effects” from the configuration file.
private String[] loadAllocationRules()	Loads allocation rules depending on the kind of node (requester, intermediate or end service node). In this version, the allocation rules consist on a set of specific effects that should be provided by a concrete node. This part is not fully implemented, but it is expected to be further developed in future work.

Table 5.7 Algorithm (Java class description)

Algorithm.java	
This is an abstract class that is used to set up the graph in which the algorithm will be applied. It also initializes the queue of nodes with the start node of the graph. This class is also in charge of controlling the process (step loop and end clauses) of the algorithm.	
Functions and procedures	**Details**
public Algorithm(Graph graph)	Constructor of the class. It receives the graph on which the algorithm will be applied.
public boolean step()	Executes one loop of the algorithm.
protected abstract void doStep()	Abstract method which calls the step implementation of the search algorithm (e.g. A* TARIFA).

Table 5.8 EEUniformCostAlgorithm (Java class description)

EEUniformCostAlgorithm.java	
This class is used to evaluate the cost of a path.	
Functions and procedures	**Details**
public EEUniformCostAlgorithm(Graph graph)	Constructor of the class.
protected int computeCost(Path p)	Computes the cost of a given path.
protected int computeFValue(Path p)	Calculates the cost of the final path, considering the heuristic introduced by the end node of the given path.
public void doStep()	Calculates, at each step, all the paths which can be constructed by adding a child node to the last node of a given path.

Table 5.9 AStarTarifaAlgorithm (Java class description)

AStarTarifaAlgorithm.java	
This class contains the implementation of the A* TARIFA algorithm.	
Functions and procedures	**Details**
public AStarTarifaAlgorithm(Vector <Vector> _graphs)	Constructor of the class. It receives a vector of directed graphs to be evaluated.
public void doStep()	Implements the main operations performed in a step of the implemented algorithm.

private boolean evaluateConditions(Path p)	Evaluates the QoS requirements conditions of a path in the directed graph which is going to be evaluated. Conditions are divided into:
	`boolean constraints = evaluateConstraints(nodeId);`
	`boolean requirements= evaluateRequeriments(nodeId);`
	`boolean preferences= evaluatePreferences(nodeId);`
	The current version of the Java-based prototype only implements the `evaluateRequirements()` method. However, the code is prepared to add the other methods when needed.

private boolean evaluateRequeriments (String nodeId, String serviceId)	Evaluates the QoS parameters of a network node of the path which is being evaluated by the algorithm. This method also considers the accumulated values introduced by the nodes in a path.

In addition to the code, some configuration files are needed in order to run the tests. They should be placed in the project workspace, into the folder `/config`. The parameters specified in these files include, for example, which nodes the composition process should consider, the existing connectivity between the nodes until the ESN, the services that each of them offer, etc. All this information provided through these files emulate the information that would be gathered through the service discovery process. This way we isolate the composition problem from the other associated needed processes (e.g. network and service discovery).

- `/config/ProtocolMessages/UseCase`: This folder contains the configurations per each node in terms of supported ASs, AMs and QoS parameters associated to each node. Each Use Case specifies the properties of the nodes that are going to be considered. This way, a specific context for each node within a path to the ESN can be simulated (Figure 5.13).

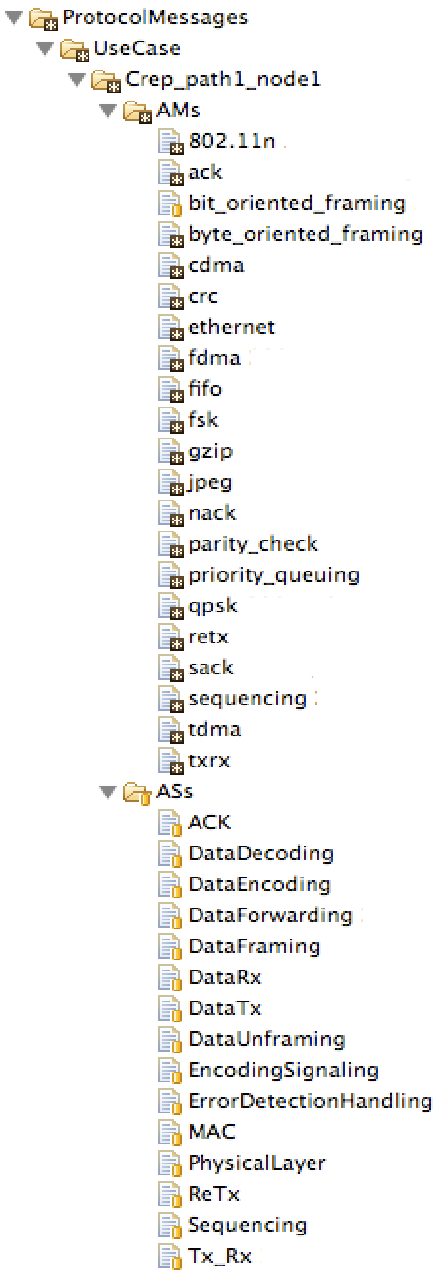


Figure 5.13 Config folder

- `/config/AtomicServices.csv`: This file contains the connectivity matrix of the supported AS, specifying which AS can be directly chained to another AS (Figure 5.14).

EncodingSigr	DataTx	DataRx	DataFraming	DataUnframi	ErrorDetectic	DataForward	DataEncodin	DataDecodin	Sequencing	ReTx	ACK
0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	1	1	1
0	0	1	0	0	0	1	0	0	1	1	1
0	1	1	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	1	1	1	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	1	1	0

Figure 5.14 AS connectivity matrix example

- `/config/Effects/Effects`: The effects are considered as requirements or desired characteristics of the communication that a user can demand. This file specifies the combination of ASs required to produce each effect (see Figure 5.15). Each effect can be offered by means of different combinations of ASs. When a requested service specifies several effects, the services implementing them are combined in order to find the best combination.

```
#Orden: bottom-up
encoding_data=DataEncoding
decoding_data=DataDecoding
cypher=DataEncoding
decypher=DataDecoding
encrypt=DataEncoding
decrypt=DataDecoding
compress=DataEncoding
uncompress=DataDecoding
transcoding=DataEncoding
security=DataEncoding:DataDecoding:DataFraming,Sequencing,DataTx:DataFraming,ACK,ReTx
chunck_tagging=Sequencing
numbering=Sequencing
ordering=Sequencing
chunck_duplication_detection=Sequencing
chunk_loss_detection=Sequencing
reliability=ACK,ReTx:ErrorDetectionHandling,ReTx:ErrorDetectionHandling,ACK,ReTx
integrity=ACK:ErrorDetectionHandling
retransmission=ReTx
scheduling_messages=DataForwarding
queuing=DataForwarding
error_detection=ErrorDetectionHandling
error_correction=ErrorDetectionHandling
chunk_correction_detection=ErrorDetectionHandling
encapsulate=DataFraming
decapsulate=DataUnframing
framing=DataFraming
packetization=DataFraming
depacketization=DataUnframing
create_chuncks=DataFraming
data_transmission=DataTx
data_reception=DataRx
signal_modulation=EncodingSignaling
channel_allocation=EncodingSignaling
channel_aggregation=EncodingSignaling
forwarding=DataForwarding
```

Figure 5.15 Effects configuration file

- `/config/Creq`: This file contains the *Communication Request* requirements, which define the requirements specified by the RN. An example of the file format file is shown in Figure 5.16.

```
# Generic format of a Communication Request (Creq)
#Creq = End Service Name & QoSRequeriments[N](min/max) & Context[N](constraints, preferences)
& AS[N](mandatory/optional) [& Effects]

end_service_name=reliable_chat

#QoS parameters=min,max,avg
#it can take a positive value or a * (means, indifference)
qos_node_cpu_consumption=0,5000,*
#[%]
qos_node_memory_consumption=0,5000,*
#[MB]
qos_node_disc_space_consumption=0,10000,*
#[MB]
qos_network_bandwidth=0,256,*
#[Kbps]
qos_network_delay=0,200,*
#[ms]
qos_network_jitter=0,1,*
#[]
qos_network_packet_loss_rate=0,2,*
#[%]
qos_as_cost=0,1,*
#[cost_units]

#Context
context_constrains=qos_as_cost,max&0

#AS

as_mandatory=ACK
as_optional=DataEncoding

#Effects

effects=

#Resources

resources=
```

Figure 5.16 `Creq` format

5.6.1.3 Tests and performance evaluation

As stated previously in section 5.6.1.1, the prototype was used to make a functional validation of the proposed composition process. For that reason, and taking into account that there was no other dynamic composition methodology proposed at that time, the comparison with other solutions was unable.

Therefore, the tests were focused only on the functional analysis proposing a scenario where a RN asks for a service that it is only available in one ESN and has two possible homogeneous paths to reach the ESN. The service composition algorithm must calculate all the possible combinations of ASs and select which is the path with less cost. The homogeneous case supposes a worst case where both paths are valid. Tests were carried out considering different numbers of intermediate nodes in order to analyze the increase of the composition time with regards to the increase of INs (that in turn, can give a hint about what will be the response against an increase of the network complexity). Concretely, all the tests were done increasing the number of intermediate nodes from 4 to 8, 16 and 32 Intermediate nodes per path, and all the tests were performed 100 times.

At the same time, we identified that the increment of complexity in service compositions and indeed the raise of the composition time, is also led by the type and quantity of desired effects that are demanded. For example, a communication that demands for data integrity, reliability (error detection and handling), flow/congestion control, security (encryption encoding) will require much more AS than service compositions that are only demanding basic communications. To achieve those service desired properties, the service composition will evaluate much more composition options and combinations falling into a much higher composition time.

Therefore, each scenario is configured by specifying the number of involved nodes and the desired effects. An example of the basic scenario proposed can be seen in Figure 5.17 showing a directed graph for representing the network connectivity. It shows a graph where the RN is connected to the ESN via two different paths containing two INs each of them.

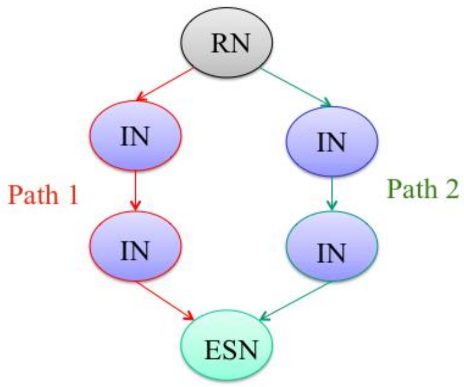


Figure 5.17 Validation Scenario

Configuration files where prepared in order to determine the QoS parameters offered by each AM and the weights introduced by each node for each QoS parameter. These parameters were set in a homogeneous manner. In addition, the presented results were obtained for the case where all the paths can find a solution to the ESN.

The desired effects were changed in each scenario in order to evaluate the cost of introducing more possible combinations of ASs, as different chained effects had to be obtained (see Table 5.10). The composition process delay implies performing the following tasks:

1. To load graph (of nodes).
2. To load service graph (service connectivity) per node.
3. To load ASs information.
4. To score AMs per AS.
5. To select best AMs per AS.
6. To apply search algorithm (A* based) to get the concatenation of services per node.
7. To score WF of ASs per node considering the selected ASs and their AMs.
8. To score all the nodes of all the discovered paths.
9. To apply search algorithm (A* based) to get best path till the End Service Node.

During steps 5 and 8, QoS filtering is applied hop by hop to avoid calculating all the possibilities. Once a path in a graph doesn't accomplish QoS requirements (according to `Creq`), it removes the path.

Note that the composition process time is being evaluated, considering that all the available services in the network are already discovered. This means that the RN has received all the possible `Cresp` to the ESN. In the considered scenarios, two possible paths to the ESN are defined. Thus, two `Cresp` are received in the RN. It is worth remembering that as A* search algorithm traverses the graph; it follows a path of the lowest known cost, keeping a sorted priority queue of alternate path segments along the way. If, at any point, a segment of the path being traversed has a higher cost than another encountered path segment, it abandons the higher-cost path segment and traverses the lower-cost path segment instead. This process continues repeatedly until the goal is reached.

5.6.1.4 Testing Scenarios and results

The following details the different scenarios that were used to test the composition process abovementioned in previous subsection, and the results obtained considering the composition time for each scenario and test. It represents a first comparative evaluation between the different composite service options that the process selects. It also provides a first look at the communication establishment delay that the composition process could imply. However, shown in this document are based on the Java-based prototype. The algorithm presents the same response form for the C based implementation (shown in section 5.6.2), but values are lower (depending on the capabilities of the machine its response is faster or slower).

First tested scenario was formed using two selected paths contain identical IN and cost between nodes. The configured desired effects were the minimal effects required in each node RN, INs or ESN. As commented, the other scenarios were gradually more complex combining security and reliability. These effects are selected among others of the defined set (Figure 5.15) because they include more combination options and therefore may need longer time to evaluate the service composition solution. Scenario 2 defines also security desired effects in RN and ESN. Scenario 3 specifies reliability (end-to-end as well). Scenario 4 combines both desired effects demanding for security and reliability in RN and ESN (see Table 5.10).

Table 5.10 Desired effects defined for each scenario

Scenario	Node	Desired effect
Scenario 1	RN	`data_transmission`
	INs	`forwarding`
	ESN	`data_reception`
Scenario 2	RN	`security, data_transmission`
	INs	`forwarding`
	ESN	`security, data_reception`
Scenario 3	RN	`reliability, data_transmission`
	INs	`forwarding`
	ESN	`reliability, data_reception`
Scenario 4	RN	`security, reliability, data_transmission`
	INs	`forwarding`
	ESN	`security, reliability, data_reception`

Figure 5.18 (simplest scenario –scenario 1–) and Figure 5.19 (scenario 4, including end-to-end reliability and security) show examples of the composition process output with the workflows obtained for each node and the best path cost:

```
Path 1 Node 1 WF 1: EncodingSignaling DataFraming DataTx
Path 1 Node 2 WF 1: EncodingSignaling DataUnframing DataForwarding
Path 1 Node 3 WF 1: EncodingSignaling DataUnframing DataForwarding
Path 1 Node 4 WF 1: EncodingSignaling DataUnframing DataRx
Best Cost Path: 32

Path 2 Node 1 WF 1: EncodingSignaling DataFraming DataTx
Path 2 Node 2 WF 1: EncodingSignaling DataUnframing DataForwarding
Path 2 Node 3 WF 1: EncodingSignaling DataUnframing DataForwarding
Path 2 Node 4 WF 1: EncodingSignaling DataUnframing DataRx
Best Cost Path: 32
```

Figure 5.18 Resulting workflows of services and the best path cost obtained for Scenario 1

```
Path 1 Node 1
WF 1: EncodingSignaling DataUnframing ACK ReTx Sequencing DataEncoding DataTx
Path 1 Node 2
WF 1: EncodingSignaling DataUnframing DataForwarding
Path 1 Node 3
WF 1: EncodingSignaling DataUnframing DataForwarding
Path 1 Node 4
WF 1: EncodingSignaling DataUnframing ACK ReTx Sequencing DataDecoding DataRx
Best Cost Path: 48

Path 2 Node 1
WF 1: EncodingSignaling DataUnframing ACK ReTx Sequencing DataEncoding DataTx
Path 2 Node 2
WF 1: EncodingSignaling DataUnframing DataForwarding
Path 2 Node 3
WF 1: EncodingSignaling DataUnframing DataForwarding
Path 2 Node 4
WF 1: EncodingSignaling DataUnframing ACK ReTx Sequencing DataDecoding DataRx
Best Cost Path: 40
```

Figure 5.19 Resulting workflows of services and the best path cost obtained for Scenario 4

The obtained results considering composition time show a polynomial response of the algorithm as it was expected (see Table 5.11 and Figure 5.20).

Table 5.11 Composition delay time for the different testing configurations

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
4 nodes	29.93ms	329.58ms	332.36ms	1741.56ms
8 nodes	52.56ms	405.71ms	398.33ms	1618.68ms
16 nodes	127.73ms	450.11ms	463.32ms	1745.89ms
32 nodes	197.24ms	690.33ms	666.26ms	2061.24ms

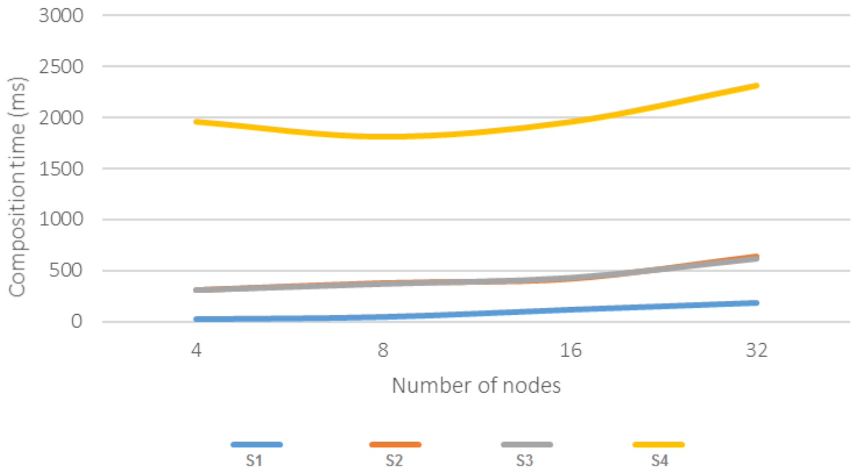


Figure 5.20 Delay response of the composition algorithm

5.6.1.5 Conclusions

A* is an extension of Edsger Dijkstra's algorithm. A* dedicates less time to resolve the composition with respect to time by using heuristics and limiting the number of composition options. In this evaluation, they were expressed through the limitations of the service set and the desired effects (no QoS parameters limitations were defined). However, when the heuristics are consistent, it is actually equivalent to running Dijkstra.

Regarding to the complexity of the algorithm, the time complexity of A* depends on the heuristic. In the worst case, the number of nodes expanded is exponential in the length of the solution (the shortest path), that is, $O(r^p)$ where r is the ramification factor and p is the tree solution level. However, it is polynomial when the search space is a tree (our case), there is a single goal state, and the heuristic function h meets the following condition [Korf et al., 1998]:

$$|h(x) - h^*(x)| = O\,(\log h^*(x)) \quad \text{(eq. 5.6)}$$

, where h^* is the optimal heuristic, the exact cost to get from x to the goal. In other words, the error of h will not grow faster than the logarithm of the "perfect estimation" $h^*(x)$ that returns the true distance from x to the goal.

As another output extracted in the research of this thesis, the A* algorithm has been adapted to the context and necessities of service composition over a FN architecture. Concretely, the main modifications realized are:

We can observe that the composition delay of scenario S1 (Figure 5.20) is much smaller than in the other compositions and that security or reliability, when demanded isolated, generate similar service composition delays. However, when the two effects are combined, the result is not linear, requiring almost double computational time than the sum of the S2 and S3 for all the tests.

- Introducing filtering rules for guaranteeing that the QoS parameters are provided for each path of ASs and network nodes. Each QoS parameter is considered in a separated way according to its effect on the global path. For instance, delay and jitter are additive metrics. Bandwidth is a concave metric, while packet loss is a multiplicative metric. This filtering allows speeding the algorithm up as it discards unsatisfactory nodes earlier.

- Weighted scoring function for each AM, although each AM can specify its own scoring function. The score for each AM is calculated every time an AS is selected. This scoring calculation introduces an extra constant delay in the results, as nodes are homogeneous.

Obviously, the obtained results can be outperformed (in comparison to the Java-based prototype used) by native operating systems implementations. However, a first quality analysis could be extracted, showing that the response of the algorithm is polynomial.

In summary, in this first approach a Java implementation of a generic composition algorithm was developed, allowing to visually debug errors on the algorithm behavior. The generated Java tool is useful as platform for testing different composition algorithms. A* search algorithm was used to select the best concatenation of services and the best path towards the ESN. AMs were scored considering basic generic weighted function, and different allocation rules were specified depending on the type of node (RN, IN or ESN) and through the configured effects for each scenario.

However, it is clear that future work has to be done. Some of the remaining tasks to be considered are:

- Add specific rules for optimizing service selection and allocation within a node.
- Add specific rules for optimizing service selection and allocation within a path.
- Verify WF composition applying satisfaction rules.

5.6.2 Enabling Service Provisioning in Future Networks

The Java-based prototype was developed in order to make a quick functional validation of the algorithm before coding the final service composition component in the TARIFA architecture.

In this second phase, the main goal is to validate the complete solution taking into account the defined FN architecture and the service composition process, showing how to enable FN service provisioning that meets QoS (e.g. bandwidth, delay, jitter) and QoE requirements. Thus, the use case scenario will consider requester needs and context information, including network, device and user features.

To show the main features and benefits of this approach we will describe a challenging multimedia use case showing how to provide inherent adapted communications (an additional uses case focusing in this field is shown in section 5.6.3).

Service and content adaptation is an extremely important issue for multimedia communications, especially when it comes to distribution of audiovisual contents in heterogeneous and dynamic networks due to the strong requirements they present in terms of bandwidth, delay, losses, device capabilities, etc. To provide the best QoE to users, QoS needs to be assured, while systems must react against dynamical changes in the network. However, this framework is not only designed to meet QoS and QoE (e.g. preceived video quality in a streaming communication) requirements in the provisioning of advanced multimedia services. It also aims at doing it in an efficient and transparent manner.

Imagine a user (U_a) who wants to watch a film (F) online from the sofa. U_a accesses the network using a tablet device supporting the following video codecs: MPEG4, MPEG2 and WMV. At home, U_a uses WLAN 802.11g technology to access the Internet. Then, it is subscribed to an xDSL line (25Mbps DL and 10Mbps UL). This would be a basic specification of the context of U_a. In the network, there are four different streaming services available: service A (S_a), service B (S_b), service C (S_c) and service D (S_d). We assume that these services are placed in different End Service Nodes (ESN), named N_a, N_b, N_c and N_d respectively. These are candidate service providers for U_a as they can offer the service that the user is asking for. Table 5.12 summarizes the available ASs and AMs on these nodes.

Table 5.12 ASs and corresponding AMs supported by each ESN providing a streaming service

<table>
<tr><th colspan="8">Na, Nb, Nc, Nd</th></tr>
<tr><th colspan="4">AS</th><th colspan="4">AM</th></tr>
<tr><td colspan="4">Data_Tx</td><td colspan="4">tx_rx</td></tr>
<tr><td colspan="4">Data_Rx</td><td colspan="4">tx_rx</td></tr>
<tr><td colspan="4">Data_Fwd</td><td colspan="4">fifo, priority</td></tr>
<tr><td colspan="4">Seq</td><td colspan="4">Incremental, temporal</td></tr>
<tr><td colspan="4">ACK</td><td colspan="4">Ack, sack</td></tr>
<tr><td colspan="4">Rext</td><td colspan="4">retx</td></tr>
<tr><td colspan="4">Framing</td><td colspan="4">bit_oriented, byte_oriented</td></tr>
<tr><td colspan="4">MAC</td><td colspan="4">csma/cd</td></tr>
<tr><td colspan="4">Error_Detection & Handling</td><td colspan="4">Parity, crc</td></tr>
<tr><th colspan="2">Na</th><th colspan="2">Nb</th><th colspan="2">Nc</th><th colspan="2">Nd</th></tr>
<tr><th>AS</th><th>AM</th><th>AS</th><th>AM</th><th>AS</th><th>AM</th><th>AS</th><th>AM</th></tr>
<tr><td>Video Coding</td><td>MPEG-1
MPEG-2
MPEG-4
VCx
WMV
FLV</td><td>Video Coding</td><td>WMV
3GP
MOV
FLV</td><td>Video Coding</td><td>MPEG-4
FLV</td><td>Video Coding</td><td>MPEG-2
MPEG-4</td></tr>
<tr><td>Audio Coding</td><td>AAC
MP3</td><td>Audio Coding</td><td>AAC
MP3</td><td>Audio Coding</td><td>WAV
AAC
MP3</td><td>Audio Coding</td><td>WMA
MP3</td></tr>
</table>

It can be observed that all the nodes N_a, N_b, N_c and N_d have the same set of AS, including: Data Transmission, Data Reception, Data Forwarding, Sequencing, Error detection and handling, Acknowledge and Retransmission management (sending and handling their reception), Framing/Unframing, Medium Access Control, Audio Coding and Video Coding. Some implementations of these ASs are provided by means of different AM options. However, to give an example of how the system works and identify that it selects one or another ESN depending on the context and RN preferences, they differ in the AM implementations provided for Video Coding and Audio Coding AS. For the sake of clarity, in this use case, it is assumed that each node (including INs) knows which services can provide. Thus, each node has a local repository with this information.

As a first proof of concept, a scenario without infrastructure support is considered[29]. This elementary scenario is defined in order to illustrate how the service composition process and the service-oriented context-aware FN architecture proposed will work without external support to store information. Note that we are composing, not only services at application level, but also at network level in a clean-slate manner, avoiding layered rigidities and asking network level services on demand. However, in future work, an approach with a distributed global directory will be undertaken. This approach will require the support of specific nodes, in order to improve service and context information searches, but compatible with the negotiation protocol to assure interconnectivity between heterogeneous networks.

To get the film, U_a sends a `Creq` to its neighbors, which is propagated hop by hop. Each node evaluates if it can provide the services requested under the desired conditions, that is, in this case, the QoS parameters that the user requires. S_a cannot be reached because the INs in path 1 (P1) make the path unsuitable for the communication as they introduce too much delay. S_b, S_c and S_d can be reached through path 2 (P2), path 3 (P3) and path 4 (P4) respectively. Nodes from P2, P3 and P4 build a `Cresp` that goes back to the requester through the reverse path form S_b, S_c and S_d. Once this stage has been accomplished, RN evaluates each received `Cresp`. This is done by applying the service composition process proposed in section 5.3. Remember that this algorithm allows to play with all the possible combinations of available services.

In this case, U_a can select S_b if he wants a service offered with lowest delay. S_c is the best considering coding compression. However, S_d is the best considering a trade-off between energy consumption (it uses a less demanding codec) and audiovisual quality (measured using objective –PSNR– and subjective –MOS– metrics). Finally, U_a decides to opt for S_d because it meets his visualization preferences and makes a better use of the life of the battery in comparison with the previously tracked down services. As an example, MPEG-2, which is the video codec (AM) available in S_d, requires eight times less the processing power for encoding and three times less the processing power for decoding in comparison with H264/AVC [Haivision, 2011] (Figure 5.21a).

[29] As described in section 1005.2.2, it is not relying on any global service/name resolution service, but semantic identification process is done by doing in-route attribute resolution.

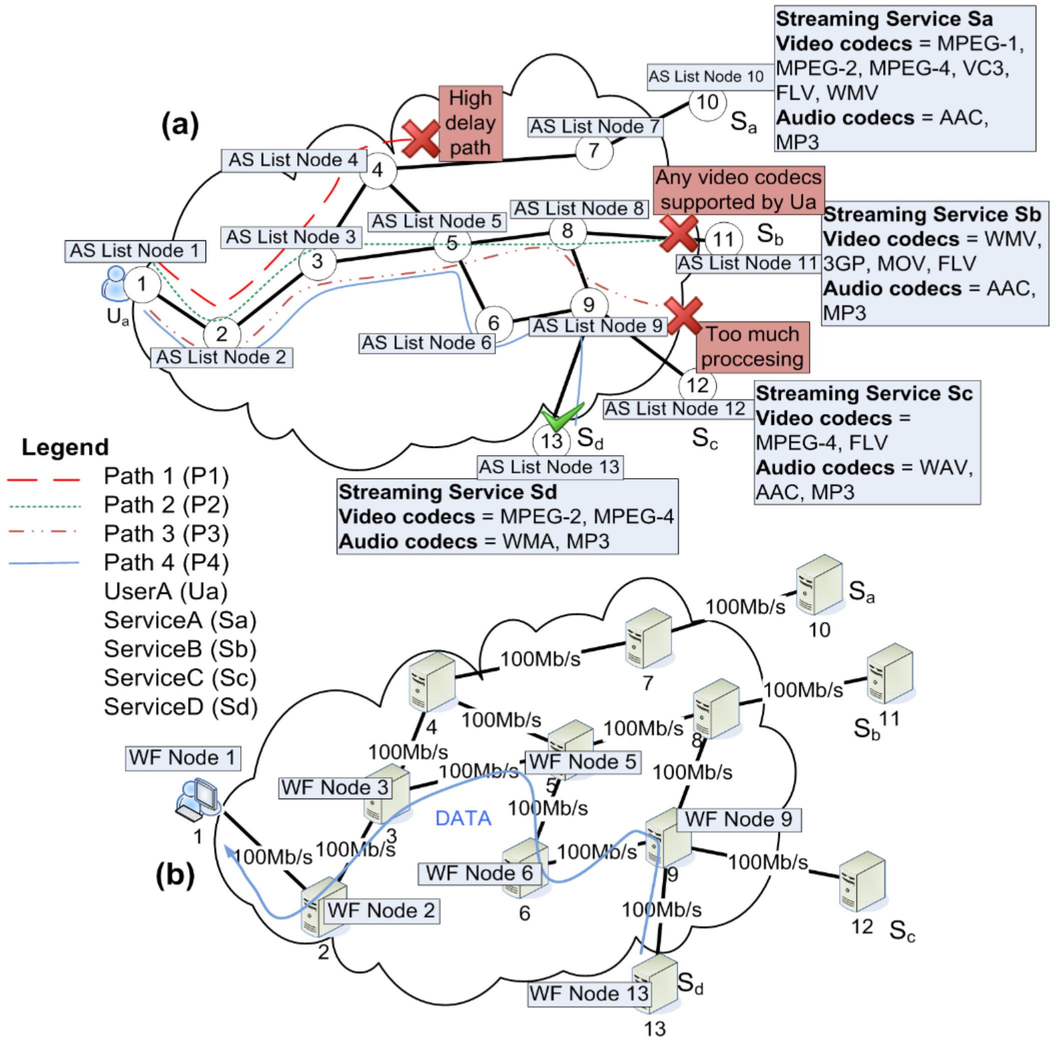


Figure 5.21 a) Adapted multimedia communication use case and b) Implemented testbed

Once services are selected for each node and WFs are created, U_a sends a `CAll` through the selected path. This message is the last message defined in the basic negotiation protocol and its main goal is to allocate the services. The total time to consume a service can be expressed as follows (eq. 5.7).

$$T_{e-e} = T_{comp} + (N-1)\cdot(3\cdot T_{prop} + T_{fwd} + T_{resp}) \qquad \text{(eq. 5.7)}$$

Where, T_{e-e} represents the total end-to-end time required from demanding a service to the start in data reception. T_{comp} is the time required for AS selection, composition, path selection and, consequently, per node workflow creation. T_{prop} is the propagation delay between two consecutive nodes (it is multiplied by 3 because of the 3-way handshake used). In the depicted scenario, we assume a constant time for each link. T_{fwd} is the time required for deciding if a node can provide the demanded end service or not. T_{resp} is the time needed to write the `Cresp` parameters offered by each node. N represents the total number of nodes in the scenario.

As already mentioned, the proposed protocol (section 5.2.2.3) allows discovering services whilst evaluating context conditions hop by hop to guarantee the required QoS. To achieve this, the discovery process includes the routing to the ESN. It is done on a per-hop CBR basis during the establishment of the end-to-end path. Thus, the routing is undertaken considering the context of the network and available services.

This use case shows a network with homogeneous INs which perform the same operations. However, the network could be composed of different network nodes with different capabilities and different services. Service composition and allocation specify which services should be placed and executed at each node in order to obtain the best possible communication. Consequently, a node with WiFi and wired (e.g. copper providing xDSL access) interfaces will be able to use different ASs depending on the context. An example would be to use congestion control functionalities in the wired interface whilst avoiding them in the WiFi interface. This is possible thanks to the RBA decomposition of functionalities and SOA-based composition. It allows to modularize segments of the network and then to place services when and where needed.

5.6.2.1 Deployment

A proof-of-concept implementation of the proposed solution was undertaken within the framework of TARIFA project [TARIFA] [Sallent et al., 2010] for the validation of the adapted media distribution use case. Some results were obtained performing a context-aware service search into a local network.

Code was adapted to run in a Linux-based desktop computer to test the proposed solution in a LAN network. The final code was formed by the modules shown in Table 5.13, and the whole development required a total of memory space of 1.5 MB and requires 229KB of RAM. This is a very low requirement for the core architecture and it would be able to run in very small devices (e.g. sensors). Actually an adaptation was also deployed in a System-on-Chip (SoC) CC2430 [Texas Instruments, 2011] from Texas Instruments platform to test it on ad-hoc constrained networks [Ferrer et al., 2012].

Table 5.13 Detail of the size of the generated code

Name	Storage size (bytes)	RAM (bytes)
Base code	1194204	116180
Service composition & Allocation	217341	61896
Search service engine	58838	18913
Constraint-based routing	75681	32697
Total	**1546064**	**229686**

A total number of 13 nodes were used (1 RN, 4 ESN, 8 IN) in the testbed (see Figure 5.21b). All of them were Intel Pentium 4 (3,2GHz, 1024KB L2 Cache) with 512MB RAM, and run OS Ubuntu 11.04 32b. All were connected using several network interfaces configured in full-duplex 100Mb/s Ethernet mode.

5.6.2.2 Results and conclusions

The time required to establish an end-to-end communication (T_{e-e}) and the resource consumption of the process were measured. Concretely, different communication requests, asking for different requirements and network functionalities were tested. Table 5.14 specifies the high-level communication goals for each test[30]. In practice, these goals are associated to different combinations of ASs, which in turn can be implemented by different AMs.

Table 5.14 Desired effects configured in the different use case scenarios

Goals	Scenario 1 (S1)	Scenario 2 (S2)	Scenario 3 (S3)	Scenario 4 (S4)
data_transmission	RN	RN	RN	RN
data_forwarding	IN	IN	IN	IN
data_reception	ESN	ESN	ESN	ESN
decoding	RN	RN	RN	RN
encoding	ESN	ESN	ESN	ESN
security		RN,ESN		RN,ESN
reliability			RN,ESN	RN,ESN

As an example, imagine that an encoding goal is defined, in which can be provided by *video_coding* and *audio_coding* ASs. Each AS can be then provided by AMs like: MPEG-1, MPEG-2, MP3, WMA, etc.

Regarding performance parameters, the average total consumption during the process of negotiation was: RN 13% CPU and 224.3KB RAM, INs 5% CPU and 113.4KB RAM and, finally, ESN 9% CPU and 192.3KB RAM. Although it can be considered high, especially in the INs, it has to be taken into account that it was run over commodity hardware with a medium profile. Doing the same example with TCP/IP and configuring those INs nodes forwarding IP the CPU achieved also peaks of 3% (but providing fixed features, with the whole connection pre-configured).

Measurements of T_{e-e} and T_{comp} are shown in Figure 5.22. Concretely, we show the results for the longest path (P4) of the testbed. We specify the time required for starting a communication using two different approaches for service composition: a) using A* as exhaustive service composition algorithm and b) using predefined templates specifying the services offered by a node.

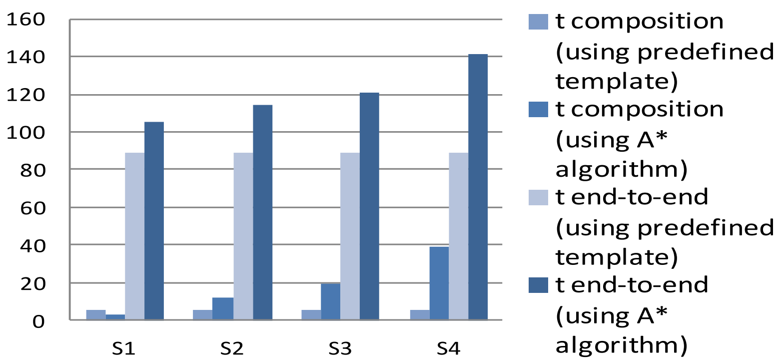


Figure 5.22 End-to-End time and composition time in milliseconds

[30] They are analogous to the ones that were defined for the Java prototype in section 5.6.1.3.

It can be observed that both, T_{comp} and $T_{e\text{-}e}$ are consistently higher applying A* algorithm with regards to a predefined template option. This is a drawback that was assumed since the beginning, but under specific circumstances, applying A* instead of predefined templates might come with the advantage that the selection of the AS, AM and nodes might be different from those selected with the predefined templates. With this deployment of this scenario it can be observed that really dynamic context-aware services can be offered achieving a $T_{e\text{-}e}$ below 150ms in networks of such that size.

Note that in our tests, T_{prop} is almost negligible as we use dedicated links in a local testbed. T_{fwd} is also constant as all the involved nodes perform a lookup into its local databases of ASs, and all of them have the same size. Regarding T_{resp}, it is slightly different in the ESN than in the INs because it has to insert more data into the `Cresp` (each node inserts information about their AS, AMs and QoS capabilities).

The gathered results are preliminary for the different scenarios introduced in Table 5.14. The most representative value is $T_{e\text{-}e}$ because it measures the delay time since the user demands a service until it starts receiving it, and the most influential parameter is T_{comp}. Composition can be a very demanding process if full flexibility and the best possible solutions required. Mostly, its value depends on the number of goals to successfully target and the number of ASs and AMs supported by a node. The proposed composition algorithm must calculate all the possible combinations between ASs in order to select the best one. The more services, the more combinations must be calculated. For example, S3 is higher than S2 because the complexity of the rules that we use to calculate the reliability goal is higher than for security.

It has to be noticed that this validation is only for this specific use case of adapted media distribution. We acknowledge that it only represents a proof-of-concept, and that the scope is limited. However, it also allowed to test the composition framework together with the whole TARIFA architecture in a controlled, but real, scenario.

In future work, it is expected that some techniques that improve this process will be studied. Even though, once a composition is performed, the resulting workflow of services could be stored for future reuse so as to avoid calculating again all the combinations of services. Finally, note that in the presented prototype, monitoring functions are not implemented. An efficient monitoring system that provides context information is especially important for the development of this solution. In the future, specific monitoring mechanisms to get real context data is expected to be used

5.6.3 Context-aware multimedia service composition using quality assessment

An additional use case was developed focusing only in the application of the defined service composition methodology for media distribution adaptation. Actually, with the proliferation of multimedia capable devices, media services have to deal with heterogeneous environments where very different types of terminals wish to receive content anywhere and anytime. This situation motivates the appearance of multimedia services that adapt contents to the specific context of users. However, as stated in previous chapters, current Internet architecture is based on a rigid layered model, which makes difficult to introduce new functionalities efficiently. To solve this, this use case introduces how Service Composition paradigm can be applied not only to define flexible service communications, but also to adapt the media for context-aware multimedia communications.

A scoring function for selecting different service implementations is presented and particularized for a case of selecting transcoding functions taking into account different quality assessment metrics. Hence, quality assessment offers a good metric to be used for measuring the quality of multimedia communications. The goal is to select the best AM for each communication, and best means that can be provided for obtaining the highest possible perceptual quality. This use case proposes to use quality metrics for deciding which AM is the best to use when an AS of the type transcoding is used. A scoring function is defined using the measured objective quality and the compression ratio provided by different codecs. However, other parameters could be added, such as performance ones (e.g. CPU, energy consumption) in order to generalize it for the selection of service in a FN multimedia service-provisioning scenario (as it was exemplified in section 5.6.2).

5.6.3.1 Multimedia quality analysis and service scoring

For this specific use case a multimedia quality analyzer was utilized (Figure 5.23), which calculates a score for each codec supported by the multimedia transcoding service. In this case, each codec corresponds to an AM, implementations of the AS named transcoding. A Full Reference [Carnec et al., 2003] system is used that can use the metrics defined in Table 5.15 for determining the obtained quality.

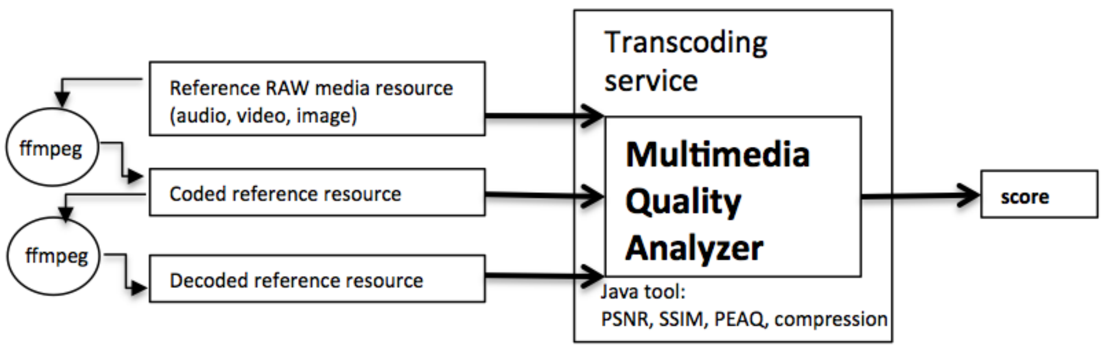


Figure 5.23 Quality Analyzer module

Table 5.15 Full Reference metrics

Media		Metrics	
Image	PSNR	SSIM	-
Video	PSNR	SSIM	-
Audio	PSNR	SSIM	PEAQ

PSNR is an objective quality metric used to calculate the ratio between the maximum possible power of a signal (in this case an audio or video stream) and the power of the corrupting noise. It is commonly used to calculate the effect of losses in a video or audio signal. SSIM is a metric for calculating the similarity between two images, which lies in the assumption that human visual perception is highly adapted for extracting structural information from a scene. Its application to audio measurement is still being studied. Finally, PEAQ is a standardized algorithm for objectively measuring the perceived audio quality.

The inputs of the quality analyzer module are: (a) a media resource (image, video or audio) in raw format, (b) the same media resource coded with a supported codec, (c) the same media resource decoded to raw format.

Inputs (a) and (b) are used to evaluate the compression ratio and to obtain the file with losses due to the effect of coding. Input (c) is used to compare the resulting resource with the original one in raw format (input of the multimedia analyzer) and to measure the differences and impairments. In our system, this process is performed offline, when the system starts or a new codec is added to the system. Then, the system performs all the analyses and stores the results (scores) into a table, which is looked up when necessary by the decision-making algorithm.

The use of lossy codecs allows the compression of the resource size. But, intrinsically, it also reduces the user quality perception. So, it must be found a trade-off of the compression ratio and the perceptual quality.

A way to decide which codec is better than another is to consider the perceptual quality and the compression ratio of a coded media resource. Thus, it can be said that a codec is better than other if this presents a better perceptual quality and compression ratio relationship. This can be expressed according to:

$$score = A * perceptual\_quality + (1 - A) * compression\_ratio, \quad where\ 0 \leq A \leq 1 \qquad \text{(eq. 5.8)}$$

A is a weight, which determines the relevance of each parameter considered in the scoring function. Hence, the relevance of each parameter can be changed. The weight that specifies an equitable relationship between perceptual quality and compression ratio is obtained for A = 0.5.

The score parameter is defined in the R set and can take values from -1 to 1:

$$score \in \mathbb{R}, -1 \leq score \leq 1 \qquad \text{(eq. 5.9)}$$

, where -1 and 1 indicates respectively the worst and the best perceptual quality and compression ratio relation.

The compression ratio parameter is defined in the R set and it can also take values between -1 and 1:

$$compression\_ratio \in \mathbb{R},\quad -1 \leq compression\_ratio \leq 1 \qquad \text{(eq. 5.10)}$$

, where -1 indicate that there is no compression between the original and the coded resource, but there has been an increment in the total number of bits, and a positive value (less than 1) indicates a reduction of the total number of bits.

The compression ratio parameter mathematical expression is defined in eq. 5.11.

$$compression\_ratio = \frac{(original\_resource\_num\_of\_bits - coded\_resource\_num\_of\_bits)}{(original\_resource\_number\_of\_bits)} \qquad \text{(eq. 5.11)}$$

The perceptual quality parameter can take values between 0 and 1 and it is also defined in the R set:

$$quality \in \mathbb{R},\quad 0 \leq quality \leq 1 \qquad \text{(eq. 5.12)}$$

, where 0 indicates, in perceptual quality terms defined by ITU-R in [ITU-R, 2001], very annoying perceptual quality, and 1 indicates no difference between the original and coded resource.

Some of the considered quality metrics do not take values between the defined ranges. Furthermore, they must be normalized. The quality metrics to be normalized are:

$$0 \leq PSNR \leq \infty,\quad -4 \leq PEAQ \leq 0 \qquad \text{(eq. 5.13)}$$

It is not necessary to normalize the SSIM quality metric as its output range fits into the perceptual quality parameter range. More details are shown in [Solé, 2009].

Moreover, the scoring of audiovisual contents should take into account the relationship between audio and video, not only considering their individual scores in an independent manner. The goal is to avoid bad combinations of audio and video profiles, for instance when obtaining combined profiles with very good audio and very poor video (or viceversa).

$$score\,(audioQ, videoQ) = (score_A + score_V) - \frac{|\,(A * score_V) - (B * score_A)|}{\sqrt{A^2 + B^2}} \qquad \text{(eq. 5.14)}$$

In eq. 5.14 a combined score was defined as the sum of individual qualities and the substraction of each point (videoQ and audioQ) to the line defined by the optimal quality (Ax-By=0). A and B coefficients are defined by the R parameter (eq. 5.15). This parameter can be introduced by default (system administrator) or by the user as a preference.

$$A = 0.5 * B = R, R \in [0, 0.5]$$
$$A = (1 - R) * B = 0.5, R \in [0.5, 1] \qquad \text{(eq. 5.15)}$$
$$R \in [0,1]$$

Figure 5.24 shows how the quality function looks like for an example value of R=0.66, which corresponds to a 4:3 relation between audio and video. Thus, we give a little bit more priority to video than to audio. However, this relation can be tweaked according to user preferences.

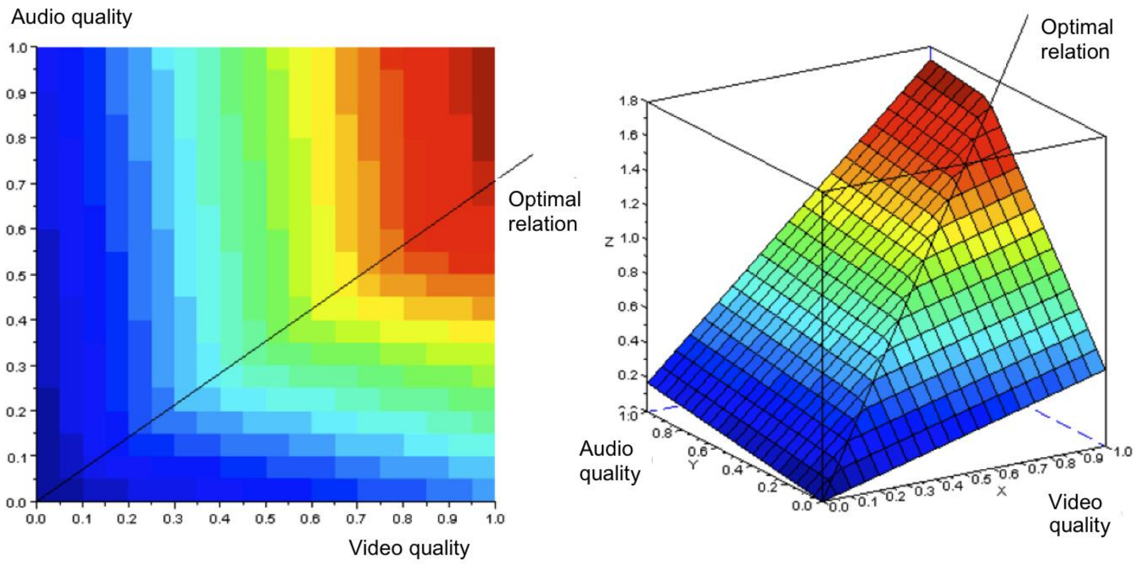


Figure 5.24 Score combination

5.6.3.2 Deployment

The testbed scenario is composed of three basic elements. Firstly, a streaming media server; secondly, a streaming client where the media resource is analyzed by means of a quality media analyzer; and thirdly a controlled network over which losses are introduced.

The media resource server acts as a video streaming server. It has been used the FFMPEG transcoding software (libavcodec 52.10.0)[31]. This transcoder allows transcoding multimedia resources with a wide range of supported codecs. FFMPEG also supports streaming over a network interface. In our case, we used UDP/RTP to stream the content over the network and to notice the loss effect. In order to remotely control this transcoder, a Web Service interface that publishes the transcoding service has been deployed.

The streaming client is a Java application that realizes requests to the transcoding Web Service and receives the streaming sent by the server. Once it receives the coded video resource, it decodes the video and analyzes its perceptual quality. The decoding process is done using the FFMPEG framework too. It is mandatory that the client gets the original resource in raw format to allow the analyzer module to perform the resource analysis. The controlled network consists on a PC running the DummyNet[32] network emulator, which permits to emulate networks with a specific bandwidth and Packet Loss Rate (PLR). The analyzed codecs, configuration and input resources are shown in Table 5.16, Table 5.17 and Table 5.18 respectively. It has been chosen these multimedia resources because they are those used in typical quality assessment studies. The packet loss rates applied in the video and audio tests were: 1%, 3%, 5% and 10%. Image analysis considers that if there is a loss in the transmission, the entire image is lost.

[31] "FFMPEG transcoder," http://ffmpeg.org

[32] "Dummynet," http://info.iet.unipi.it/ luigi/dummynet/

Table 5.16 Tested codecs (AMs)

Image	JPEG, GIF, PNG
Video	MPEG-1 video, MPEG-2 video, MPEG-4 part 2, H.263, H.264, WMV1, WMV2
Audio	MP3, AAC, AC3, Vorbis

Table 5.17 Configuration parameters

Video	Audio
Bitrate: 1024kbps	Bitrate: 128kbps
Frame rate: 25fps	Sampling frequency: 44100Hz
GoP size: 12	Bits/sample: 16bits
Quantification scale variation:	Coding quality parameter: default codec

Table 5.18 Tested resources

Image	Lena
Video	Foreman
Audio	Vocal quartet, Instrument flute

5.6.3.3 Results and conclusions

Some results shown in Figure 5.25. They focus on audio because is the feature that can be measured with all the mentioned metrics of Table 5.15, and therefore the complete analysis. From the PEAQ scoring results it can be concluded that the best audio codec in terms of quality is AC3, followed by AAC. However, in terms of compression ratio the best one is Vorbis, although it is the worst in terms of quality. Thus, if the score parameter is considered, the best scored one is the AC3 codec. The scoring function allows to order different implementations of a specific function, this case a transcoding service implemented by different codecs, in order to determine which one is the best. Moreover, the use of SSIM for audio quality assessment is still being studied [Kandadai, 2008].

In summary, the study presented in this subsection introduces a way to enable context-aware adaptation of content by applying the presented service composition methodology. Indeed, it can be combined with proposed FN architecture, for enabling, not only to compose dynamic and suited communications, but also for placing in-line media adaptation services on demand. Through the context-aware media distribution use case, it exemplified that, thanks to the proposed specification of FN architecture and service composition methodology, new functionalities can also can be added in an easy and flexible way, allowing the proliferation of new in-line applications while adapting architectures to past, present and newcoming requirements.

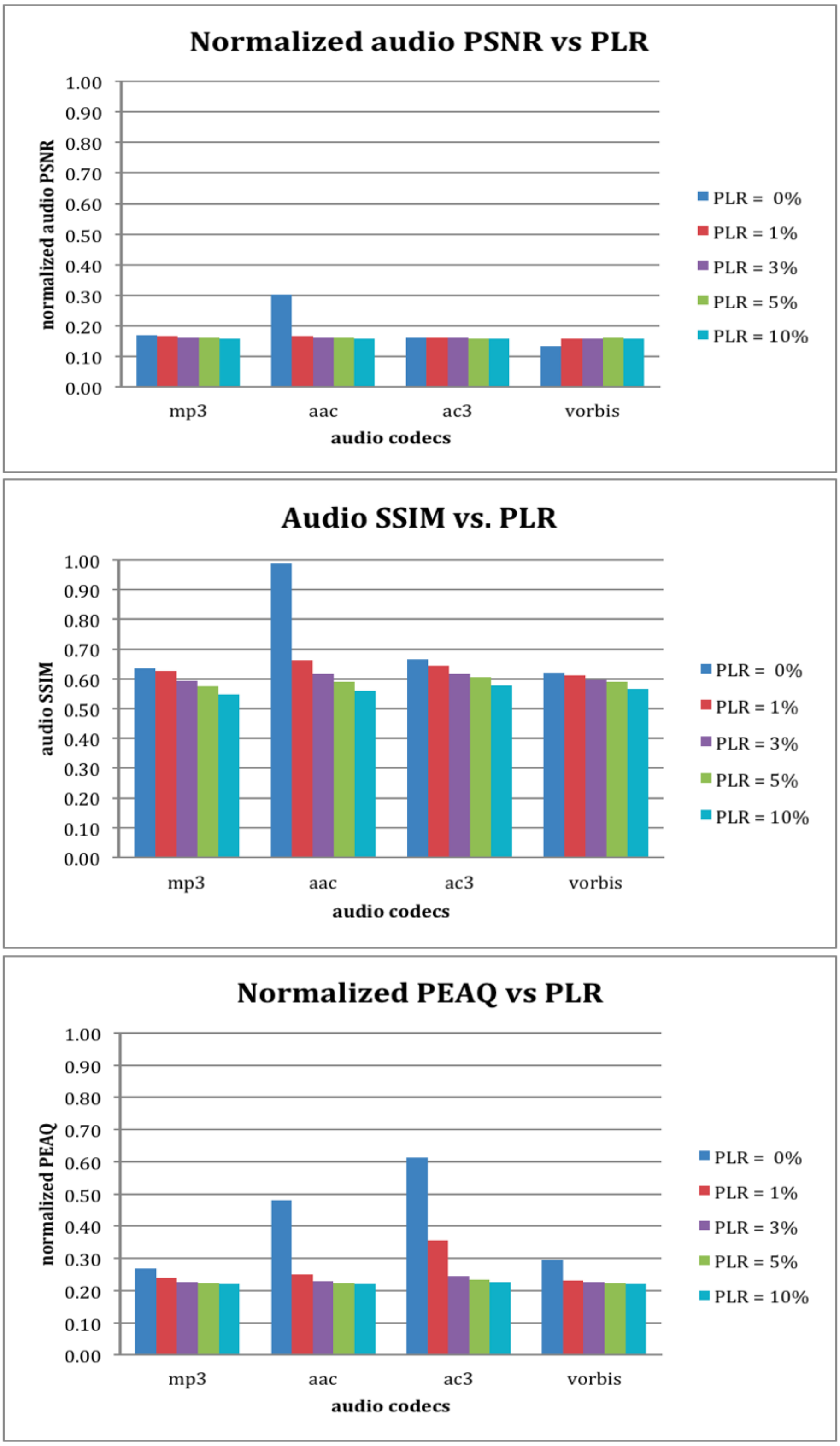


Figure 5.25 Testbed results

5.7 Summary and conclusions

This chapter presented the proposed service composition mechanisms and methodology and the tests and proof-of-concept scenarios deployed in the field of media distribution. It reviewed the specific challenges of service composition in FN (section 5.1), defined the different services types and processes (section 5.2 and 5.3) and described the structure of the service composition modules and how they are mapped (section 5.4) in a FN architecture such as the one proposed in chapter 4.

It included a description of the Atomic Services and Composite Services, considering the requirements of the FN scenario and inspired on the service description used in web-services composition and service-oriented FN proposals such as SONATE. It detailed a chain of processes to complete service composition of on-demand context-aware communications or services, developing each of these processes (service discovery, filtering, scoring, composition, and deploying). This methodology and the mechanisms defined to undertake these processes were validated in a proof-of-concept FN scenario of a context-aware media distribution service. Although the scope of those tests is limited, it was a first real implementation of such service-oriented FN architecture and also allowed to test the composition framework in a real scenario.

Taking into account a potential change of paradigm in the way the services are accounted and charged, a generic costing framework methodology is also proposed as complementary work (section 5.5). It considers the empowerment of end-user not only in the selection of the services, but also in the distribution channel and intermediate in-network services. Although the proposed FN architecture and the composition mechanisms in this thesis are not yet mature enough, this scenario, unavailable in current-day networks, could be enabled by a massive deployment of these elements in the future.

Finally, an additional use case of the service composition framework was also presented, proposing it to not only be used together with the FN architecture, but as a methodology to modularize media streaming services and compose them on demand. It was tested for providing adaptive video and audio streaming, selecting the most appropriate audio or video codec depending on the context characteristics (user and network). This additional case showed how the framework could also be used as a flexible way for allowing new functionalities and in-line applications while adapting also current-day architecture to newcoming requirements.

6 Proof-of-concept of service composition on Smart Grid: from context-aware network management to flexible application deployment

During the research on the application of Service Composition techniques in FN architectures, some characteristics were identified as potentially beneficious to other fields, such as media delivery (as shown in previous chapter) or the Smart Grids. While media delivery and media adaptation can be directly associated to Service Composition and its evolution in the future can be easily linked to FN architectures, Smart Grid environment may seem at first glance a difficult scenario for the deployment of the concepts proposed previously in this thesis.

Notwithstanding, Smart Grid can be seen as a hybrid system composed by many systems. At large scale point of view, it merges the electric power system itself and a heterogeneous Information and Communication Technology (ICT) infrastructure. At the same time, those systems are composed by many building blocks that are hard to fully integrate [Zaballos et al., 2011] between each other and, thus, some partial solutions have been proposed so far [Selga et al., 2014]. However, these partial solutions are no longer feasible given the broad range of services the Smart Grids are due to support and the high cost of deploying specific solutions [Selga et al., 2014].

New models of networks presented in this thesis such as Service Composition [Gonzalez et al., 2013] [Khondoker et al., 2010], but also in conjunction with Software Defined Networks/Anything (SDN/SDx) [Monsanto et al., 2013] have arisen as powerful tools and methodologies for managing the FN architectures that propose modularity, adaptability and centralized management of the communication system. The characteristics provided by these approaches match perfectly with the requirements stated by the Smart Grid and more concretely against cybersecurity. In the case of the Smart Grid and more concretely in the electricity distribution network a huge amount of data collected is processed continuously. Nowadays it is treated usually by dedicated and highly expensive devices. Relying on the experiences arisen and the knowledge gathered from the partners during the development of the FP7 European projects INTEGRIS [INTEGRIS] and FINESCE [FINESCE] project, we advocated for a Software Defined Utility (SDU) concept for the management of the Smart Grid and its security [Martin de Pozuelo et al., 2016]. Following this proposal, many of those functions that nowadays are undertaken by expensive dedicated devices, will rely on programmable commodity hardware, low-cost sensors, and high-speed and reliable IP-based communications underneath. In this scenario, different research topics were derived and they are detailed in this chapter.

As it is seen in chapter 2, main challenges of Smart Grid revolve around (1) the adaptation of the electrical network infrastructure to novel topologies that improve the resilience and self-healing capacity of the grid; (2) novel monitoring strategies, based on decentralized (service-oriented) architectures that can deal with the huge amount of information gathered within the Smart Grid ecosystem; (3) the security of the whole system and the associated concerns about the collected energy data and its privacy.

SDU proposes a new way of managing the Smart Grid following a software-based approach that enable a much more flexible operation of the power infrastructure. Indeed, we would want that the Smart Grid provides autonomic behavior reacting against possible isolation of some parts from the main network. Therefore, self-adapted and context-aware applications are needed to efficiently provide Smart Grid services in this heterogeneous and highly dynamic environment. As it has been shown in chapter 5, a service-oriented architecture and Service Composition techniques can provide a solution for flexible operation and deployment at different fields and levels.

At the same time, Smart Grid demands for monitoring strategies that adapt to the heterogeneity of the Smart Grid. Thus, a flexible solution should also be presented for gathering and process real-time Smart Grid data. Those novel monitoring strategies can also benefit from service-oriented decentralized architectures that configure each specific IoT device and its communication services separately.

And finally, it also needs to handle the variable privacy requirements and the security thread arisen from the interconnection of the power grid to IP-based communication network. In this case, a solution based on service composition can also present several benefits, providing security functionalities on-demand and only if necessary, adapting their operation to each specific area and function of the Smart Grid.

6.1 A flexible and context-aware Smart Grid infrastructure

All of the functionalities proposed by Smart Grid require a higher level of control of the network that increases the complexity in its management. The boost on distributing and virtualizing electrical resources requires new methodologies that simplify the tasks of the administrators of the electrical distribution networks [FINESCE]. In this sense, the advances on computer network architectures and management could give a hint about how it could be done. Especially, there are two new trends that present capabilities completely aligned with the specific needs of the Smart Grid.

On the one hand, Software-Defined Networks/Anything (SDN/SDx) present a way to manage highly distributed resources in a more autonomous and centralized way, programming certain type of resources to be adapted on the fly by themselves (self-configuration) and providing an easier and centralized way to manage the resources remotely. On the other hand, the generalization of the concept of service oriented computing and Service Composition for Smart Grids could give also many advantages [Gungor et al., 2013] [Spano et al., 2015] [Zaballos et al., 2011]. This methodology widely spread in the world of web-services, presents a way of modularizing the functionalities as small independent services that could be published, placed, invoked and combined together with other services, to run them remotely and on demand.

SDN/SDx aim to decouple the network control plane and the data plane [Giorgetti et al., 2013], which allows to (1) manage highly distributed resources in an autonomous and centralized way, (2) program certain type of resources to be adapted on the fly by themselves (self-configuration), and (3) provide an easier and centralized interface to access the resources remotely.

Indeed, one of the main advantages of SDN/SDx is the abstraction of the network to unify and centralize the management of the network. Abstraction gives network administrators a global vision of the whole. Also, it allows system architects to achieve greater scalability and an easy integration with different middleware, even on demand (e.g., using a cognitive system able to learn, react, and configure the network according to certain parameters). This will make the network to act like a living organism that adapts itself to certain situations boosting its versatility and ease of operation.

Taking into account the critic low latency required in Smart Grid communications and that the resources of the distribution electrical grid such as generators, storage devices or actuators are increasingly becoming more spread, the authors consider that the combinations of both technologies could make a great impact in the development of the Smart Grids in the following years. Actually, the necessity of totally protection against failures in Smart Grids leads to research different solutions to solve this issue. Moreover, the use of an orchestrator for handling redundancy in different type of networks can efficiently tackle the stringent requirements of recovering services from a failure in the system, even though an overload is introduced on it [Sancho-Asensio et al., 2014] [Navarro et al., 2013]. The performance depends on the status and characteristics of the underlying network.

As an example, in the scope of the INTEGRIS [INTEGRIS] project, a context-aware broker for the Smart Grid (Figure 6.1) was designed to actuate according to the gathered information and operate efficiently some aspects of the Smart Grid (such as communications QoS, security, or data replication among the Smart Grid infrastructure). It was based on a semi-supervised cognitive system that decides, among other directives, which security services are deployed to protect which part of the electrical network. A concrete example of the dynamic and flexible Smart Grid service configuration is further detailed in section 6.1.3.3, focused on securing Smart Metering services.

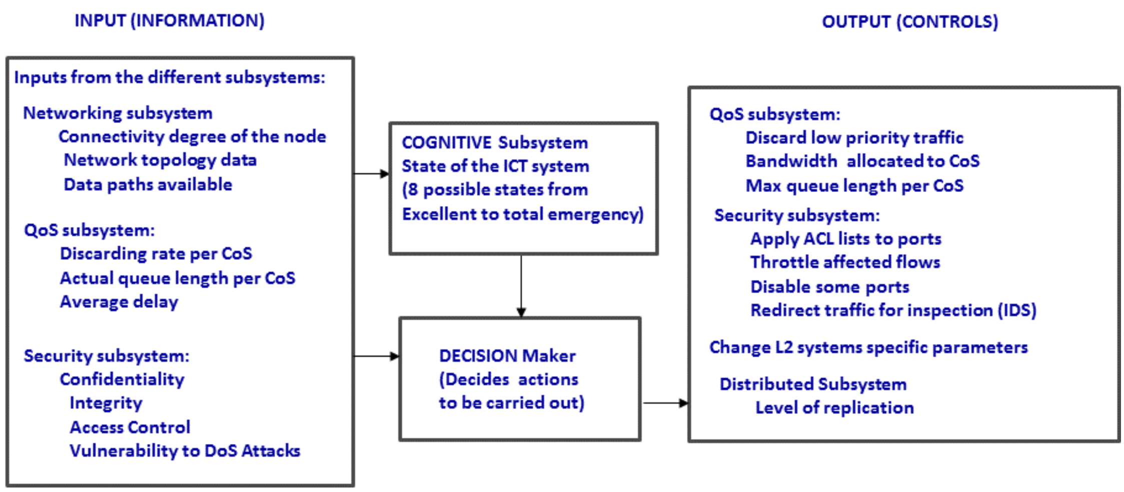


Figure 6.1 Context-aware Smart Grid management broker

Moreover, the rest of this chapter revolves around the forthcoming Smart Grid transformation towards an intelligent programmable network. It is possible by translating the philosophy, concepts and technologies from the SDN/SDx and Service Composition in order to implement a Software Defined Utility (SDU).

Therefore, the different subsequent sections will go through three different pieces of this SDU concept that has been studied in the scope of this thesis:

- A flexible data management system for the Smart Grid. A distributed storage system based on a dynamic configuration of the nodes that collect and store the data gathered of the Smart Grid at distribution level. From our real-world experiences collected during the INTEGRIS and FINESCE projects, we have found that dividing the Smart Grid into these logical layers poses some critical difficulties arisen from the fact that typically, IEDs (Intelligent Ellectronic Devices) are closed devices that do not allow implementing custom developments (e.g., security or information-exchange protocols)—as novel experimental devices do. Therefore, we proposed a new device coined as FIDEV [Martin de Pozuelo et al., 2016] [INTEGRIS][FINESCE] that behaves as a frontier between these two layers and implements (1) a communications subsystem that allows heterogeneous network coexistence, (2) a security subsystem that provides a reliable and secure low layer communications infrastructure, (3) and distributed storage subsystem that smartly stores all data generated by IEDs. In addition, a specific challenge has been studied on this topic, seeking an orchestrator for the hybrid cloud in Smart Grid environments, as a system that decides in which cloud the resources and data collected by the Smart Grid should be placed.
- A Web of Energy (WoE) management system: An IoT-based infrastructure that enables machine-to-machine interactions between small and resource-constrained devices on the Smart Grid domain based on HTTP protocol. It extends the IoT concept by providing a bidirectional human-to-machine interface, inspired by the Web of Things (WoT), which results in a ubiquitous energy control and management system (i.e., uniform access to all devices of the Smart Grid) coined as Web of Energy (WoE). Its main objective is to carry out a proof-of-concept of an open web-based interface that isolates the electricity grid domain from its utility functions, relying on the abovementioned distributed storage layer to support the massive amount of data generated by the grid.
- A framework for flexible security services deployment: Definition of a framework to flexible deploy Smart Grid services. The research has focus especially on the Smart Grid security requirements, analyzing the different level of security that each Smart Grid function need, and apply different policies (that translate into different services and different service compositions). A basic use case is defined for the Smart Metering security infrastructure, in order to exemplify the potential of the proposed solution and compare it with current operation.

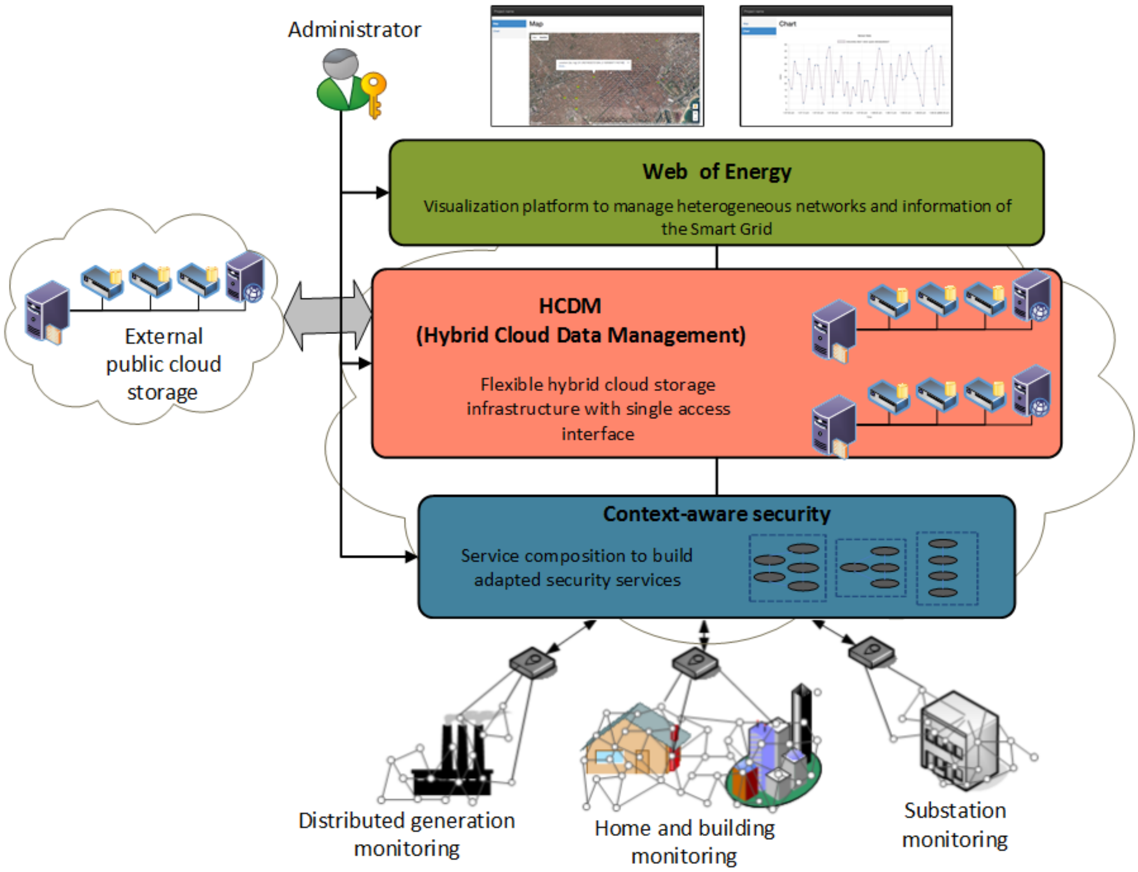


Figure 6.2 Developed components of the Software defined Utility

6.1.1 The Smart Grid data management

In our work in FINESCE, we were focused on validating and evaluating the novel SDU concept, which advocates the migration of the utility infrastructure to software systems as much as possible instead of relying on complex and rigid hardware-based systems.

In that sense, a first step proposed was establishing a distributed storage system that provides high-availability and reduces the latency in acquiring data from the local sites of the utility while offering a secure solution to share data information with external stakeholders.

The scenario considered by FINESCE partners stressed on this flexible data management considering a novel ICT infrastructure for Smart Distribution Grids that allow to flexibly move Smart Grid data and applications from local systems to an external cloud and protect them by using security mechanisms that can be added on demand.

There can be several reasons for the mobility of applications and information from the public cloud to local and vice versa. They range from application latency improvement (placing apps closer to data when necessary) to the confidentiality of the data (when the data is too sensitive to be stored in the public cloud), through the low capacity of local resources (and using the public cloud when more storage resources –and more flexible and dynamic ones– are required). However, it will make Distribution System Operator (DSO) infrastructure ready to interact with the cloud in a very gradual incorporation of the novel functionalities.

FP7 INTEGRIS [INTEGRIS] project in which I was previously involved, developed an integrated ICT environment able to efficiently encompass the communications requirements of the Distribution Smart Grid. Within this scope, it provided a distributed architecture capable of integrating the different elements needed for the Smart Grid: Remote Terminal Units (RTUs), Smart Meters, IEDs, sensors, Smart Grid Applications, communications systems which allow adequate performance to the Smart Grid (SG) functions and adaptation to stringent requirements and specific situations [Zaballos et al., 2011].

The outcome of the INTEGRIS project was the design and field testing of a single, yet distributable, device (called I-Dev) which integrates the needed functions for it:

- Integrated communications management.
- Support for Smart Grid functions.
- RTUs and Smart Meters data collectors.
- Special functions to improve latency, reliability and QoS.

During FINESCE project, we proposed to integrate an upgraded version of these devices, that we called FIDEV (FINESCE Devices) by adapting the concepts developed in INTEGRIS into the FIWARE [FIWARE] eco-system of Generic Enablers[33] (GEs) and cloud computing, with the objective of providing a data storage that is adapted to the dynamics of the Smart Grid. It should provide flexibility in order to configure the specific allocation of the information due to the sensitivity of some collected data, and the possible exchange of other information (such as smart metering or electrical vehicle charging point status) with third parties.

The stringent communication latency, security and privacy requirements lead to a hybrid solution in which the Smart Grid manager can configure where is the best place to store the information collected at different points of the network (e.g. substation information, smart metering, etc.). The system offers two possibilities, to store it locally in the own infrastructure of the utility company or by means of external public cloud services. Therefore, a flexible distributed storage system allocated all over the utility infrastructure was designed. It relies on a set of FIDEV devices placed at different points of the electrical power distribution network (e.g. secondary substations), and interconnected in order to replicate the information gathered and make it accessible all over the Smart Grid when necessary.

Three main priorities were demanded by FINESCE partners in its design:

- To provide a scalable distributed storage solution that handles the large amount of data that could be generated in the distribution grid, and indeed, be the basis of a SDU.
- To provide tools that can be adopted easily by the DSO administrators. Graphical interfaces must offer simplicity and usability.
- To assure that the solution provides the level of security required for managing the communications and data of the critical infrastructure for what is designed.

[33] A library of components with reference implementations provided by FIWARE framework projects that allow developers to put into effect functionalities such as the connection to the Internet of Things, Identity Management or Big Data analysis, making programming much easier (http://catalogue.fiware.org/enablers).

We proposed a Hybrid Cloud Data Management (HCDM) module (Figure 6.3) that works over the FIDEV machines. It includes a REST service which provides users with transparent access to the Hybrid Cloud (distributed local storage or cloud storage system) infrastructure for the data management of the SDU, combining and integrating functionalities from object storage management of local instances (FIDEVs) and object storage in a public cloud service; identity management and additional encryption functionalities.

This module is understood to be a data management tool for utilities (to upload public information on Electric Vehicle charging points, Smart Metering, costs of transmission system and power plants, energy costs, etc.). It could be useful for any retailer that wants a straight-forward but secure tool to manage data and share it with other stakeholders.

HCDM allows the DSO managers (or external stakeholders, depending on their access permissions to the stored data) to manage the information offering different functions: list all the public data objects stored in a specific directory of the storage platform, upload or download a data object to/from the public or private data storages, create new data containers to classify the objects and delete stored containers or objects.

Notice also that the definition of the HCDM was done very modular, thinking on service composition premises, in order to dynamically plug security services, or change the replication policies on demand.

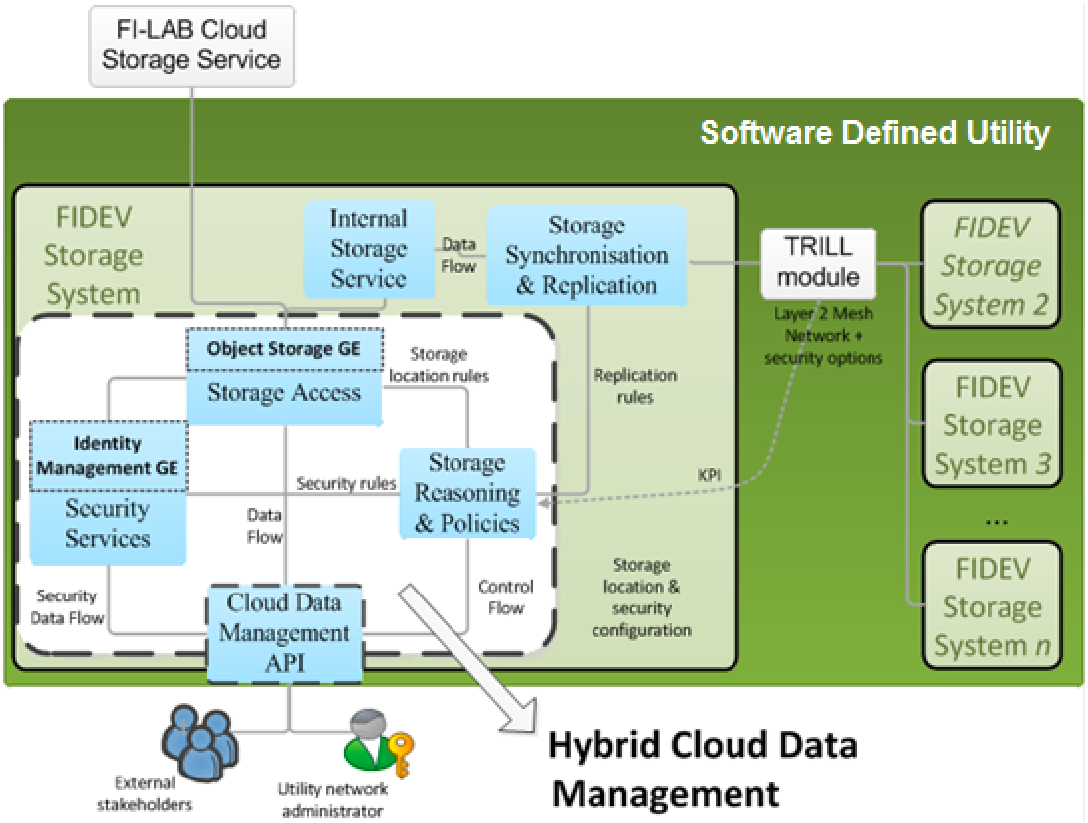


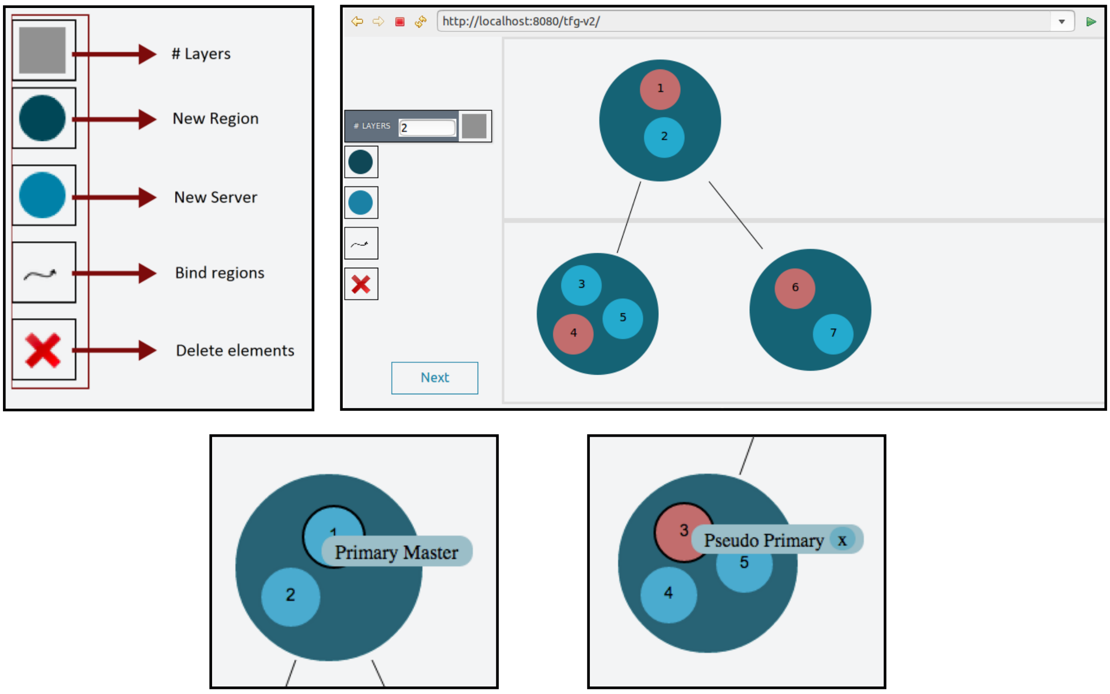
Figure 6.3 Hybrid Cloud Data Management (HCDM) functional blocks diagram

6.1.1.1 Distributed storage deployment and configuration tool

In addition, a graphical interface has been developed in the context of the distributed storage system that enables to easily configure and deploy it over the utility facilities.

Figure 6.4 shows some screenshots of the tool developed to easily configure the distributed storage system. It offers a toolbar that a network manager can use in order to define the number of layers, regions and servers that will be taken into account for the information replication logic, and afterwards send the configuration to all the servers.

It is important to point out that the servers considered in the application are the HCDM modules of the FIDEV machines placed inside the utility infrastructure. They are collecting data from different energy resources and are in charge of replicating the data among them in order to provide data redundancy and spread the information over the Smart Grid, so it can be rapidly accessible at different points of the network. Layers define the different levels of replication, being the servers on layer above the ones that are updated more frequently [Navarro et al., 2013]. Inside any layer we can define the different regions, which determine the scope of the data replication. Each server could be then located inside any of the regions, and we can also bind the different regions from the same layer or not. Therefore, depending on the layer that the region is placed, the time of refresh of the data can be different.



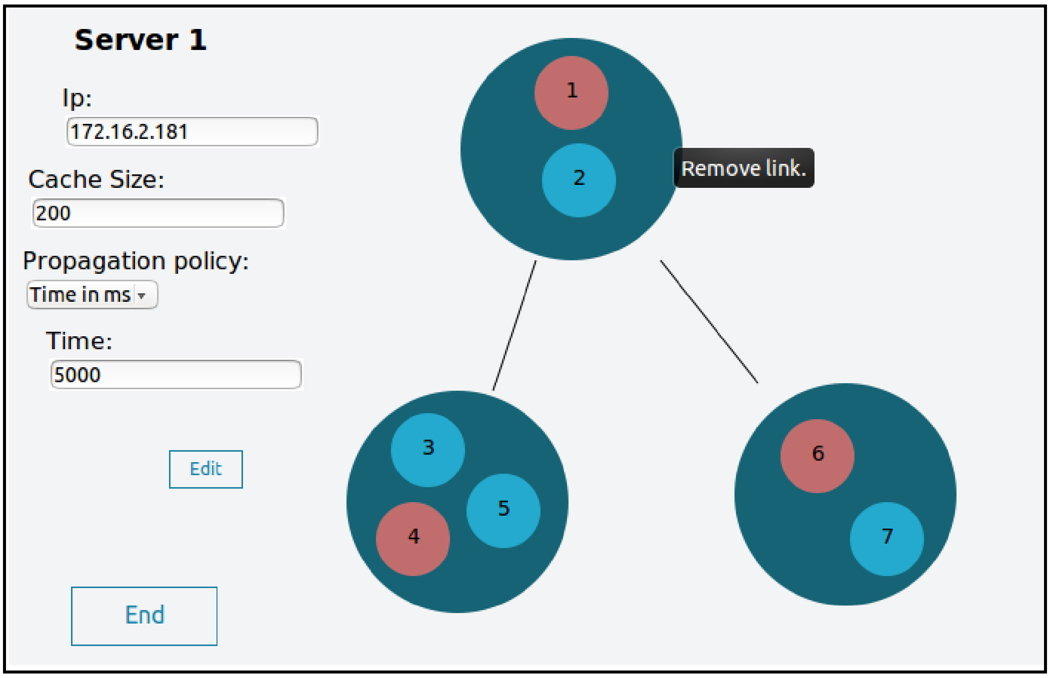


Figure 6.4 Deployment configurator tool

For example, if we create 2 layers and place one region on layer 1, and two regions on layer 2. Then we can bind them using the links, and define the main server in each of the regions, that will be the one that contains the original data that will be replicated among the servers of the region, and afterwards among the servers of other linked regions [Navarro et al., 2013].

Finally, it allows to define the network interfaces of the different servers, and when finished, it starts configuring the distributed system in the network, and deploying the data replication logic among the different servers and specified regions.

Once all the distributed system is deployed and operative, other mechanisms or tools are needed to interact with it and to manage the data stored. A RESTful API can be used to interact with the HCDM modules and store the data in the closest FIDEV. That can be directly configured to work with some sensors or basic software systems in order to store sequentially any piece of information collected. In addition, a java-based front-end application (Figure 6.5) is also offered to upload new datasets to the system, access and download them, or easily migrate them to the a public cloud when they want to offer them to external stakeholders. It provides three basic windows: (1) a login window will allow the user to identify itself[34]; (2) a preferences window that allows users to configure the FIDEV to which it will be associated, the remote access to the public cloud[35] and SSL functionalities; (3) the file manager window that enable users to create private or public (in the cloud) new folders, delete them, upload or download objects from both sites, migrate them from private to public storage and vice versa, or delete objects.

[34] In the version deployed in FINESCE project, the authentication is provided by means of the Identity Management service of the FIWARE Lab, although it can be easily changed to other authentication mechanisms if necessary.

[35] In the version deployed in FINESCE project, it was provided by means of the FIWARE Lab Cloud infrastructure.

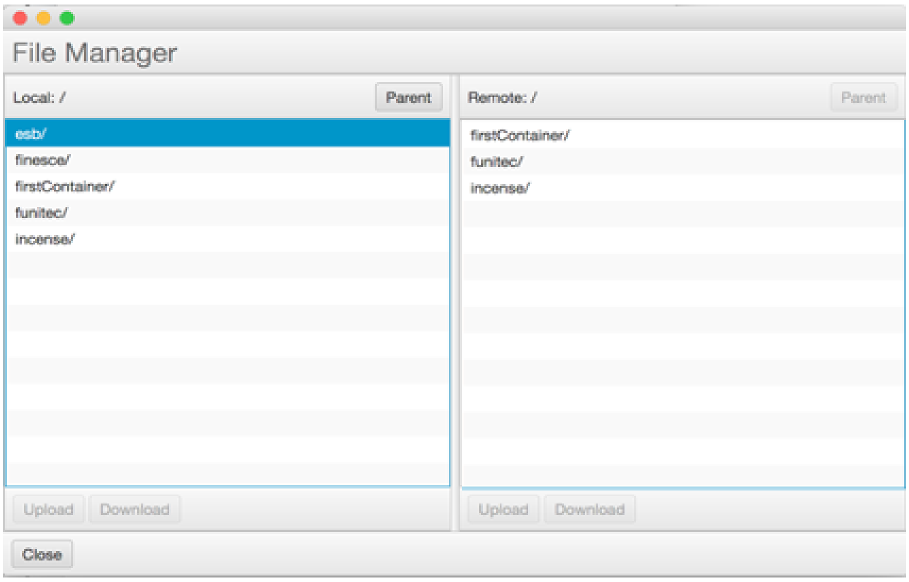


Figure 6.5 HCDM front-end application

At the end, from our experience in FINESCE and INTEGRIS, we identified that DSO network administrators seek low-cost and simple tools to manage their infrastructure, with a "software" approach that enables to collect and manage data in a straight-forward way, and considered (together with the involved industrial partners) that the system and tools presented can be highly useful for the management of all the data generated by future electrical power networks.

6.1.1.2 Hybrid cloud allocation reasoner

One of the objectives of this research was also to investigate in the efficient usage of this hybrid cloud for managing the data generated by the Smart Distribution Grid environment. A private cloud was considered to store recent gathered data generated (smart meter data, electric vehicle charging stations, etc.), but there is not always enough storage capacity to keep all the historical data. Then, a public cloud is used to respond less restrictive queries or to handle peak demands, using an outsourcing burst. In the case of outsourcing burst, there is only an additional expense on demand when the private cloud cannot provide all the services, being the necessary additional resources provided by the public cloud [Genez et al., 2013].

Concretely, Software Define Utility trial (SDU) specifies data gathering from utility and data replication between nodes located in two different environments, in public and private clouds, creating a hybrid cloud. These nodes have the ability to replicate information through them and aim to store information in several allocations to have access from anywhere, regarding the cyber-security aspects, and allowing the users with the corresponding permissions to access the system.

However, the crucial factor for economic savings in IT by using a hybrid cloud is the optimal allocation of resources. If we imagine a scenario with different services to be allocated in more than one cloud, the distribution of these services is not trivial. Besides, a high time response is a detriment of the Quality of Service (QoS) offered by the cloud [Li et al., 2014]. The fact of choosing a particular location without a defined strategy may entail not the best choice for a resource distribution to fulfil the defined requirements or can represent a cost much higher than the optimum cost [Chu et al., 2013]. It is necessary to design a set of rules that mark preferences, priorities and limits of cost, time, etc. in order to obtain the best possible location for services or data in a particular scenario [Shifrin et al., 2013]. In order to analyze the behaviour of the cloud and help to determine the best place to store the Smart Grid data, a table (Table 6.1) was created to characterize the cloud depending on the service provided and a set of objective metrics [Briones et al., 2015]. This table was created thanks to the joint feedback of experts from different perspectives (Academia, DSOs, ICT industries, etc.) working in FINESCE project.

In the vertical axis, the table provides a set of metrics classified in different categories: Computation, Storage and Network. The Miscellanea category is added to complement this classification, including economic, elasticity, scalability or security aspects. So, the importance of several metrics on a cloud is qualitatively defined depending on which action is being carried out, with a weight of 1 (low importance), 2 (medium importance) or 3 (high importance). It is important to emphasize that the assigned colors, red (1), yellow (2) or green (3), were defined through a process of cross opinion between different stakeholders involved in the project. The boxes in blue were subsequently modified in a second iteration, analyzing the contributions and opinions from other partners and cloud service providers.

The horizontal axis defines a set of processes and operations that are performed in a cloud, in order to characterize the behavior of the cloud. Depending on the type of services mostly used, and therefore the type of cloud demanded by the user, this table can determine which metrics should be selected if we want to evaluate and grade the available clouds where the user can allocate resources. If we want to use it in a more fine-grained view, before doing one of the specific operations, Table 6.1 can be consulted in order to select which metrics should be taken into account to evaluate in which cloud to allocate a resource.

After reviewing several similar studies [Mazhelis et al., 2011], it was found that there is a tendency to follow a specific strategy to deploy services in the most suitable cloud. This strategy is based on using metrics to evaluate the location of services in one cloud or another. Some studies [Malawski et al., 2013] [Zhang et al., 2014] stand on the premise that the placement of resources will always be cheaper in the private cloud than in public clouds. This is because it is supposed to have available resources in the private cloud (the investment to deploy the infrastructure has already been performed previously). Thereby, the proposal is to place all the services in the private cloud, which shall not assume any additional cost unless the operation itself, relying on a threshold value. Beyond this threshold, the resources should be placed in the public cloud due to peak loads and their associated cost on demand. To set this threshold, the use of metrics that help to mark the boundary of the private cloud is necessary.

The information exposed in this table can be used in the future to develop a hybrid cloud reasoner that decides autonomously which is the best allocation for a resource or a service, guided by the context of it and the evaluation made with the knowledge of the experts from industry and academia[36]. It can be used to extract a set of guidelines that automatically, or assisted by the data manager, can indicate which will be the best location to realize the different cloud operations defined in the horizontal axis.

Table 6.1 Cloud metrics and Smart Grid evaluation

Process/Operation	Desploy an application/service/instance	Migrate an application/service/instance	Power off an application/service/instance	Copy an application/service/instance: Running	Power off an application/service/instance	Pause an application/service/instance	Increase resources: Add physical disk	Increase resources: Increase storage	Increase resources: CPU	Increase resources: RAM	Increase resources: Network	Decrease resources	Autoscaling	Data migration	Data copy	Backup: Machine (100%)	Backup: Snapshot (changes)	Backup: Data	DRS (Disaster Recovery System)	Drop physical machine	Manteinance mode of a physical machine
Economic costs	3	2	2	2	2	2	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2
Percentage of apps that meet SLA	3	2	1	1	1	1	1	2	2	2	2	1	3	2	2	2	2	2	3	1	1
Total-Cost-of-Ownership	3	2	2	2	2	2	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2
Infrastructure cost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Power Cost	3	2	2	3	2	2	2	3	3	3	3	1	3	2	2	2	2	2	3	3	2
Server Cost	3	2	1	1	2	2	2	3	3	3	3	1	3	2	2	2	2	2	3	1	1
Networking Cost	3	3	1	1	3	3	1	1	1	1	2	1	1	3	2	2	2	2	3	1	1
Maintenance Cost	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Cold Spares estimation	2	2	1	1	1	1	3	3	3	3	2	1	3	2	2	2	2	2	2	1	1
Hot Spares estimation	2	2	1	1	1	1	3	3	3	3	2	1	3	2	2	2	2	2	2	1	1
Generic costs	3	2	2	2	2	2	3	3	3	3	3	3	3	2	2	2	2	2	3	3	2
EUE (kWh it) Energy Use Efficiency	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	1	1	1	2	3	2
EUE (kWh) Energy Use Efficiency	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	1	1	1	2	3	2
EUE (CPU) Energy Use Efficiency	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	1	1	1	2	3	2
Power Usage Effectiveness (PUE)	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	1	1	1	2	3	2
Data Center Compute Efficiency (DCcE)	2	3	3	3	1	1	2	2	2	2	2	3	2	3	1	1	1	1	2	3	3
Service Compute Efficiency (ScE)	2	3	3	3	1	1	2	2	2	2	2	3	3	3	1	1	1	1	3	3	3
Digital Service Efficiency	3	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	2	1	1
Scheduling based on fairness	3	3	2	2	3	2	2	2	2	2	2	2	2	2	2	2	1	1	3	1	1
Scheduling/Metric based on execution cost	3	3	2	2	3	2	2	2	2	2	2	2	2	2	2	2	1	1	3	1	1
Spot price dynamics	3	3	1	1	3	2	3	3	3	3	3	2	2	3	2	2	2	2	3	1	1
Resiliency (security/isolation)	3	3	2	2	3	2	2	2	2	2	2	2	3	3	3	2	2	2	3	1	2
Realiability	3	3	2	2	3	3	2	2	2	2	2	1	3	3	3	2	2	2	3	1	2
Availability	3	3	2	2	3	3	2	2	2	2	2	1	3	3	3	2	2	2	3	1	2
Jitter	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Latency	3	3	2	1	3	2	2	2	2	2	2	2	3	2	2	1	1	1	3	1	1
Speed	2	3	1	1	3	2	1	1	1	1	3	1	3	3	3	2	2	2	3	1	1
Migration Cost	1	3	1	2	3	2	1	1	1	1	1	1	1	1	1	3	3	1	3	2	2
Hardware reliability	2	2	1	1	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	1	1
Data throughput	1	3	1	1	3	2	2	2	1	1	3	1	2	3	3	3	3	3	3	1	1
Average delivery time of new products or services	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1
Average time to deploy an application	3	3	1	1	3	2	1	1	1	1	1	1	3	1	1	1	1	1	3	1	2
Time to Genesis	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Recovery time	3	3	1	1	3	3	1	1	1	1	1	1	3	3	3	2	2	2	3	1	1
Average time to provision a node	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1
Average Weighted Response Time (AWRT)	3	3	1	1	3	3	3	1	3	3	3	1	3	3	3	2	2	2	3	1	1
Instance Efficiency (% CPU Peak)	3	3	1	1	3	2	2	2	2	2	2	1	3	3	3	2	2	2	3	1	1
ECU Ratio (Gflops/ECU)	3	3	1	1	2	2	1	1	3	1	1	1	3	2	2	2	2	2	3	1	2
Maximum performance loss	3	3	1	1	3	2	2	2	2	2	2	1	3	3	3	2	2	2	3	1	1
Degradation time	3	3	1	1	3	2	1	1	1	1	1	1	3	3	3	2	2	2	3	1	1
Slowdown time/fairness	3	3	1	1	2	2	2	2	2	2	2	1	2	3	2	2	2	2	3	1	1
Elasticity	3	3	2	1	3	1	3	3	3	3	3	2	3	1	1	1	1	1	3	2	2
Density	2	2	1	1	2	1	1	1	1	1	1	1	3	2	2	1	1	1	3	1	1
Time to deploy	3	3	1	1		1	1	1	1	1	1	1	2	1	1	1	1	1	3	1	1
Runtime performance	3	3	1	1	3	2	2	2	2	2	2	1	3	3	3	2	2	2	3	1	1
Workload	3	3	1	1	3	2	2	2	2	2	2	1	3	3	3	3	3	3	3	1	1

[36] It is based on the knowledge of FINESCE partners and some external cloud service providers. However, in the future this table is expected to be further developed with the consensus from additional partners.

6.1.2 The Smart Grid as an IoT

Another piece of the SDU involves the graphical remote management from a single interface [Vernet et al., 2015]. Recent advances on Smart Grids have explored the feasibility of considering the power electrical distribution network as a particular case of the IoT [Guinard et al. 2011] [Navarro et al., 2014] [Vernet et al., 2015]. Certainly, this specific domain poses appealing challenges in terms of integration, since several distinct smart devices (also referred to as Intelligent Electronic Devices or IEDs) from different vendors — often using proprietary protocols and running at different layers — must interact to effectively deliver energy and provide a set of enhanced services and features (also referred to as smart functions) to both consumers and producers (prosumers) such as network self-healing, real-time consumption monitoring and asset management [Zaballos et al., 2011]. Although the latest developments on the IoT field have definitely contributed to the physical connection of such an overwhelming amount of smart devices [Bo et al., 2014], several issues have arisen when attempting to provide a common management and monitoring interface for the whole Smart Grid [Aman et al., 2013] [Navarro et al., 2013].

Indeed, integrating the heterogeneous data generated by every device on the Smart Grid (e.g., wired and wireless sensors, smart meters, distributed generators, dispersed loads, synchrophasors, wind turbines, solar panels and communication network devices) into a single interface has emerged as a hot research topic [Zaballos et al., 2011]. So far, some experimental proposals [Navarro et al., 2014] have been presented to face this issue by using the Web of Things (WoT) concept to access a mashup of smart devices and directly retrieve their information using reasonably thin protocols (e.g., HTTP, SOAP) [Guinard et al., 2011] [Vernet et al., 2015]. However, the specific application of these approaches into real-world environments is fairly dubious because (1) they may open new security breaches [Bou-Harb et al., 2013] (i.e., end-users could gain access to critical equipments), (2) there are no mature electric devices implementing WoT-compliant standards available in the market [Navarro et al., 2014] and (3) industry is averse to include foreign modules (i.e., web servers) on their historically tested and established — but poorly evolved —proprietary systems [Gungor et al., 2011].

Therefore, a new way to overcome these issues was explored through the European projects INTEGRIS [INTEGRIS] and FINESCE [FINESCE]. This work provides a management interface for the Smart Grid inspired by the WoT. Continuing the work done in these projects, the aim is to implement an ICT infrastructure—based on the IoT paradigm—to handle the Smart Grid storage [Navarro et al., 2013] and communications requirements [Zaballos et al., 2011] to manage the whole Smart Grid and link it with end-users using a WoT-based approach, which results in a new bridge between the IoT and WoT. This proposal, which takes the pioneering new form of the WoT, is targeted at providing a context-aware and uniform web-based novel environment to effectively manage, monitor, and configure the whole Smart Grid. Moreover, conducted developments prove the feasibility and reliability of our approach and encourage practitioners to further research in this direction and to envisage new business models [Bari et al., 2014] [Rodríguez-Molina et al., 2014].

The open IoT-based infrastructure presented in our Web of Energy proposal will provide new tools to manage energy infrastructures at different levels, from IoT-based infrastructure enabled machine-to-machine interactions between small and resource-constrained devices on the Smart Grid domain. Thus, the IoT concept has been extended by providing a bidirectional human-to-machine interface— inspired by the WoT—that results in a ubiquitous energy control and management system coined as Web of Energy [Navarro et al., 2014] [Vernet et al., 2015]. This proposal will combine the web-based visualization and tracking tools with the Internet protocols, which enables a uniform access to all devices of the Smart Grid. In order to provide such an effective and reliable management interface to address the heterogeneous nature of devices residing on the grid, we will continue the deployment of an intelligent subsystem devoted to (1) learn from the real-world events, (2) predict future situations, and (3) assist in the decision making process.

The tools developed can be provided in an open format, available to anyone in the research community, and able to contribute and enhance the platform building new modules for managing other resources. As a first step, the platform will be demonstrated through the management of Smart Metering resources, based on the formats of DLMS COSEM and IEC 61850, but the whole system will be designed for a much wider scope where every utility, enterprise, public administration, or any organization or single prosumer can build their own sensor application on top of it. Figure 6.6 represents a conceptualization of this system, in which different actors (TSO, DSO, PROSUMERS, etc.) interact wich a single interface based on a public-subscription model, and in interact with heterogeneous data ontologies and management achitectures (represented in the figure by AMI, Virtual Substation proposed in FINESCE [FINESCE], Electrical Vehicle or data network).

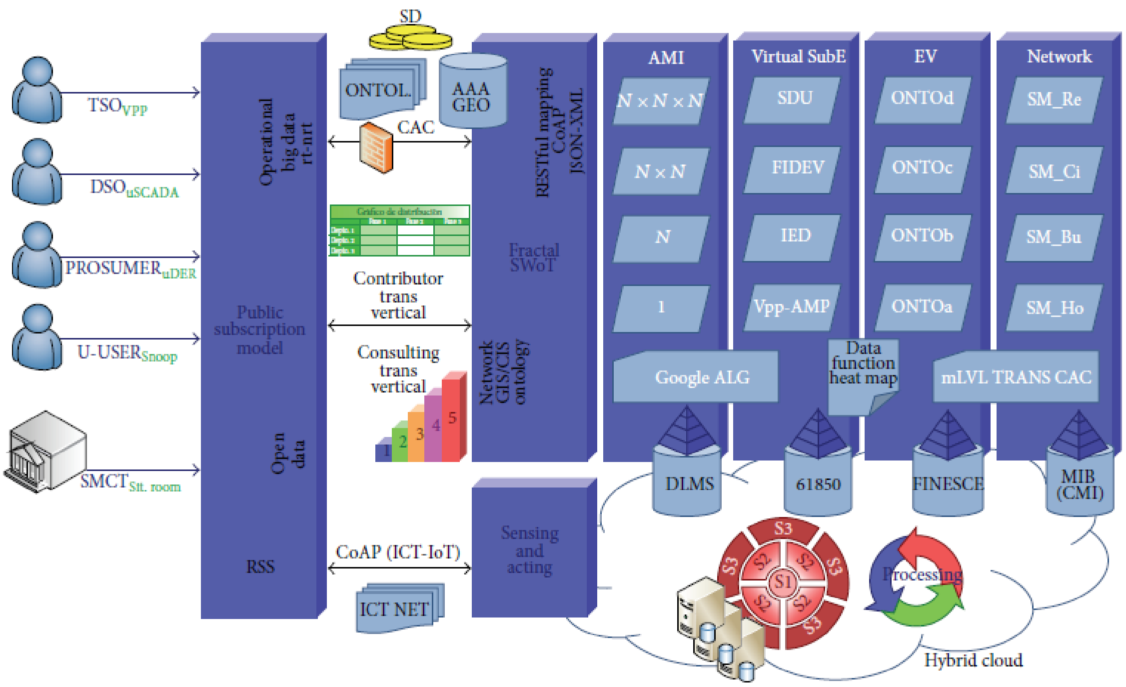


Figure 6.6 WoE interactions

6.1.2.1 WoE design and development

The concept of Smart Grids has been considered in recent years as an appropriate answer to address new challenges in the energy domain. However, additional resources are still needed in challenges such as the proactive operation of the grid, efficient integration of demand into grid operation, integration of renewable generation or maximum network reliability to obtain applicable solutions. The WoE will combine energy domains and ICT technologies with the objective of building an interoperable platform for the coordinated planning, operation and settlement of future distribution and access to networks by integrating demonstrable solutions in real user environments based on web of things technologies.

The current control of electricity distribution networks, including the Advanced Distribution Automation (ADA) and Demand Side Management (DSM), is (1) centralized, (2) silo oriented (fragmented by applications), (3) using specific communication technologies for each purpose but there is a lack of integration among them and (4) DSO owned ICT systems tend to be used [Selga et al., 2014]. So, in practice the situation is still centralized and fragmented but with existing solutions and trials allowing the integrated management and the distribution of selected granular functions. This contrasts with the distributed and fractal nature of the future Smart Grids, which can only be based on standardization, flexibility, distributed systems and communication among all the actors [Gungor et al., 2013] [FINESCE].

Furthermore, the increased use of renewable and distributed generation means the operation and management of the electric power system must change radically. Increased levels of automation, distributed intelligence and on-line data mining and management are required to deliver the network control functions, reducing reconfiguration and the restoration times. Reconfiguration of smart grids addresses new challenges during normal operation and also for restoration and management of crisis situations [Oualmakran et al., 2011]. The connection of end-users (prosumers) to the energy market will facilitate the installation and connection of devices that offer grid services that will help to mitigate capital and operational costs of the grid modernization required for energy transition, and to minimize environmental impact, thus ensuring lower electricity prices for everything involved. New benefits will be generated and shared in a fair way between all actors, from aggregators to industrial end-users and citizens.

A coordinated vision of the grid will provide mechanisms to tackle the challenges mentioned above showcasing them through real and modern applications. WoE will provide synergies at different levels: Smart Grids and other smart networks, individuals and communities as a first approach to a Software Defining Utility (SDU). The potential benefits of the WoE are framed by: a) innovative communications, acquisition and processing platform based on the extensive use of the “Real Time Services” concept providing open and interoperable access; b) metering integration platform based on multiplatform web technologies; c) Advanced Medium/Low voltage control centre integrating real-time grid information coming from devices to provide a clear view of the current and the near-future status of the grid thanks to a high performance environment; d) an energy services market platform. The performance quantification of that WoE concept will be the key to accelerate the implementation of new policies, market rules, and emerging smart grid programs [Bari et al., 2014] [Rodríguez-Molina et al., 2014].

WoE outlines some of the challenges in improving the resiliency of the electrical grid and proposes an approach that makes use of advanced sensor technology (advanced sensors are needed to improve the knowledge of state), analytics, and agile control in a Smart Grid [Miller et al., 2014]. Furthermore, WoE proposes a smart grid supervision infrastructure, which can deliver real-time and high-performance notifications on a global scale for transferring measurements from different distributed sensors and take actions over the grid via different communication protocols, informing the different stakeholders (e.g. producers, consumers, aggregators, system operators, etc.) within the adequate time frame [Zaballos et al., 2013] [Selga et al., 2014]. We propose to use a distributed infrastructure based on web of things and implement a novel service platform for facilitating distributed control, auto healing and power grid control.

In addition, WoE technology will tackle the implementation of smart real time distributed monitoring platforms enabling the data fusion and knowledge extraction for the different faults detection and prevention schemes. Intelligent HTTP based sensors will provide a new source of relevant distribution status information, including loadings, voltage profiles, harmonics and outage conditions which, combined with equipment condition data, such power frequency interference signatures, will provide predictive perspectives of potential equipment failures. This platform requires large storage technologies that can hold the massive amounts of information that will be generated by the millions of sensors implemented in the Smart Grids and that will make the information available in negligible times to the system or systems demanding it (Figure 6.7).

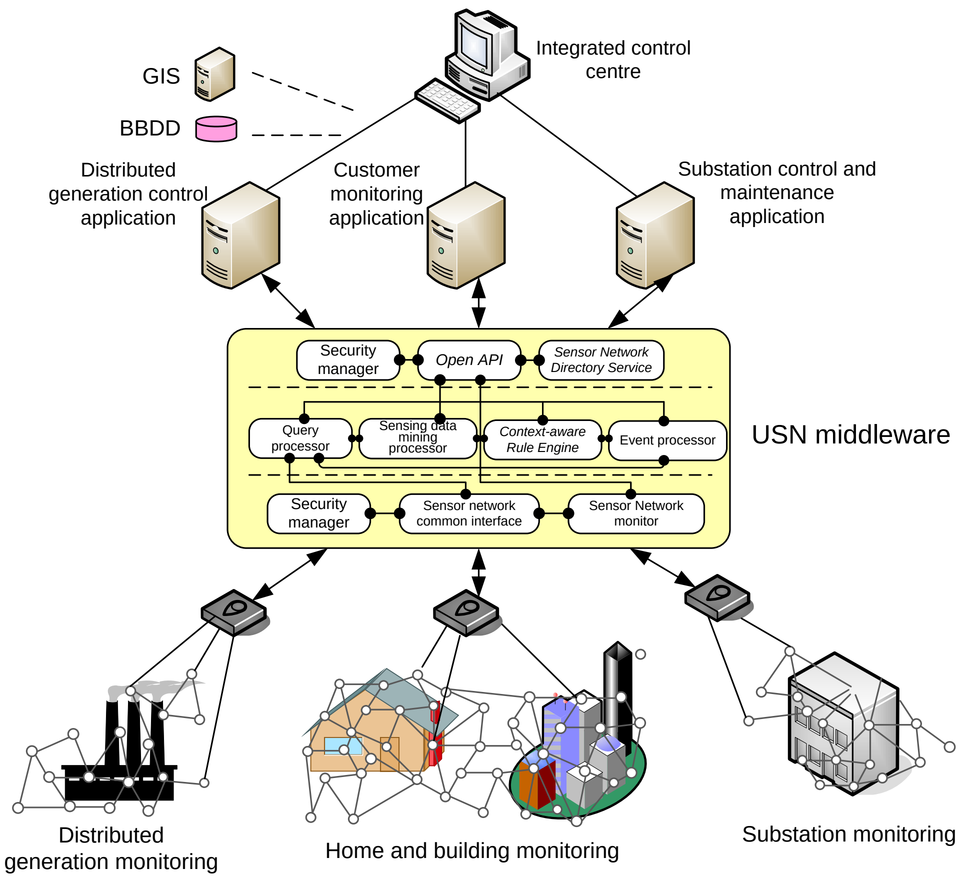


Figure 6.7 Open API architecture

Smart grids need monitoring strategies based on decentralized and uncoupled architectures (service oriented, multi-agent), supported by real time middlewares (DDS – Data Distribution Service) capable of dealing with huge amounts of information at different time scales, process events (complex event processing) and discover sequences of them (sequence event discovery) working together with OLAP (Online Analytical Processing) and Data Mining solutions. A multi-agent conception of the grid is necessary to deal with coordination and optimization requirements for an efficient and safe network. Pervasive web monitoring devices with communication capabilities increase security of the network as a whole but imply new cyber-security issues and make optimal restoration and reconfiguration more complicated [Sancho-Asensio et al., 2014]. Currently, heuristics, meta-heuristics and learning methods are mature and available. Restoration and reconfiguration techniques including auto healing will benefit from these reasoning engines.

6.1.2.2 The Web of Things

Day by day, the number of connected Things is growing exponentially. The latest data shared by Gartner finds that 4.9 Billion of Things will be accessible during 2015 and this figure will increase to 25 Billion in 2020 [Gartner, 2013]. The way to access these devices from a single platform is undoubtedly one of the biggest headaches for researchers. In this regard, standardized solutions provided by the rapid evolution of the Internet have laid the foundation of what we call the Web of Things. Therefore, WoT architectures aim to integrate everyday objects with web technologies. Those devices should be able to communicate with each other using existing web standards. Prerequisites for those Things are minimal processing and communication capabilities. WoT researchers try to define and delimit concepts (e.g. what is a Thing?) implied on those envisioned architectures and solve some problems that arise when every Thing may sense or actuate on every Thing.

The architecture presented by Dominique Guinard in his thesis [Guinard et al., 2011] proposes a good basis to start other WoT designs. Guinard defines WoT general prerequisites, he also defines what a Thing is and a Virtual Object is in a WoT architecture and finds solutions to problems like how to discover and find Things and how those Things can connect and push information to a server. In [Guinard et al., 2011] a layered architecture (see Figure 6.8) that consists of four layers that address four main problems is proposed:

- Device Accessibility Layer: how to enable consistent access to all kinds of connected objects?
- Findability Layer: how do we find their services to integrate them into composite applications?
- Sharing Layer: how we preserve privacy?
- Composition Layer: how do we get closer to end-users?

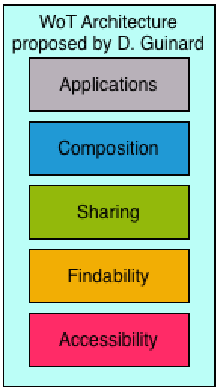


Figure 6.8 Layered WoT architecture [Vernet et al., 2015]

One of the main problems of the WoT architecture is to standardize communication protocols between different Things. Indeed, there is still no clear standard defined for this purpose and there are different options available. In the next section, the best placed and supported protocols by researchers are presented, discussing also which protocols are also valid/eligible alternatives.

6.1.2.3 WoT protocols

It would be easy for WoT to use only one protocol, but the heterogeneity of devices composing the WoT makes it unfeasible. Thus, different communication protocols should be considered. The use of each one depends on the final proposed solution. Preferred candidates by the community are MQTT and CoAP [Al-Fuqaha et al., 2015], although some other alternatives are also presented.

A. MQTT

As stated in its specification [Locke, 2010], MQTT stands for Message Queue Telemetry Transport. It is a publish/subscribe (pub/sub), extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst attempting to ensure reliability and some degree of assurance of delivery. These principles also contribute in the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium. MQTT was introduced by IBM and Eurotech companies.

MQTT is a protocol that uses a pub/sub model, connecting publishers and subscribers via a broker (server). Its headers are small and therefore their overhead is minimum. MQTT can also work over SSL for security reasons, but SSL adds an extra overhead to the communication. As publishers and subscribers connect via a broker, the use of a centralized server leads to a SPF (Single Point of Failure).

B. CoAP

CoAP [Bormann et al., 2012] was specified and standardized by the CoRE (Constrained RESTful Environments) group in IETF, the Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks, such as those that will form the Web of Things. This protocol shares several similarities with HTTP like its REST architectural style but instead of using TCP it uses UDP to achieve its goals.

As it is a request/response protocol like HTTP, both WoT servers and constrained devices or gateways should act as servers and clients at the same time to ensure bidirectional communication at any time. For example, a constrained device using this protocol may fire an event to the WoT server and the WoT server may request something to the constrained device. Proxies between HTTP and CoAP will achieve interoperability between HTTP and CoAP clients. Translation between CoAP and HTTP protocols is easy and straightforward as equivalences of response codes, options and methods are present in both protocols. Security is achievable using DTLS and a variety of key management methods.

C. Other Candidates

DDS (Data Distributed Service) [Pardo-Castellote, 2003] is an API specification and an interoperability wire-protocol that defines a data-centric publish-subscribe architecture for connecting anonymous information providers with information consumers". DDS follows a decentralized pub/sub model. It differs from MQTT model in the following two key points:

- DDS protocol starts to operate on top of the link level layer of the OSI model creating a Common Data Bus where every device can connect in a decentralized manner. This protocol also defines several QoS options.
- As a decentralized protocol, it does not have a SPF like the broker in MQTT.

DDS has only implementations for C, C++ and Java and has a higher learning curve compared to MQTT. On the other hand, MQTT clients are implemented for several languages.

XMPP [Saint-Andre, 2011] (originally named Jabber) is a protocol for person-to-person communication based on XML. Its main use is for chat communication but since the growth of the IoT concept, the XMPP Standards Foundation is working on defining extensions (XEP) for use in the IoT. These extensions aim to specify standards for a wide variety of communication types between IoT devices such as Control, Discovery, Multicast, Pub/Sub message types, or many others. They use EXI[37] (compressed XML) to reduce the size of messages, as XML is known to produce larger file/message sizes than other text based formats. Even though XEPs for the IoT are not as much as popular as MQTT or CoAP, it is worth to keep track of them as they are growing fast and may be a basis of the format WoT messages are defined.

AMQP [Vinoski, 2006] is a message-centric binary wire protocol that uses a centralized broker. AMQP is built on top of the TCP layer (at least, it is assumed to work on top of TCP). Authentication and encryption is made available through SASL and TLS respectively. As AMQP was created by businesses-to-businesses, it provides transactional modes of operation that allow it to take part in a multi-phase commit sequence. The key feature of AMQP is that it was designed for interoperability between vendors. It mandates the behaviour of the messaging provider and client to the extent that implementations from different vendors are interoperable. Third party implementations of AMQP clients exist for several languages.

This section is going to review some WoT implementations of this architecture. Guinard joined the EVRYTHNG platform and engine [Guinard et al., 2011]. This engine allows Things (they call them "things"), to be connected to this platform through a RESTful API. They describe two types of things:

- Unconnected/Tagged: They are encoded in a 1D/2D bar code or a NFC/RFID tag and users can interact with them by scanning the tag.
- Connected: Those things can interact with the RESTful API of the EVRYTHING engine, they can be sensed and/or actuated.

This engine offers the creation of applications that represent remote client applications (like those used in social networks like twitter or facebook). Thanks to his THNG-Push technology

[37] Efficient XML Interchange Working Group. http://www.w3.org/XML/EXI/

(currently in beta), the engine provides a publish/subscribe MQTT M2M broker where WebSockets wrapping MQTT are used to allow communication with browsers. They are also working on adding CoAP support to this technology. Node.js and javascript libraries are available to facilitate the use of their API.

In [Tracey et al., 2013], the authors propose and implement a holistic web architecture for the Internet of Things. They point out key features and capabilities of a holistic architecture and they use a layered model with an abstraction layer for communication among devices.

Node-RED [Node-RED] is a visual tool for wiring hardware devices, APIs and services. From the Node-RED front page: "Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range nodes in the palette. Flows can be then deployed to the runtime in a single-click." Also in beta, this tool aims to provide users with a visual manner connecting things. Node-RED is built on Node.js and customized functions between nodes can be created within the editor using Javascript. There is an EVRYTHNG Node-RED integration library to add some functionality of the EVRYTHING platform to this tool.

Octoblu [OCTOBLU] is an open-source cloud platform (public, private or hybrid) built to connect people, devices and systems through a great variety of protocols like MQTT, CoAP, HTTP(S) and WebSockets using a RESTful API. It also offers a very powerful visual tool for connecting things (nodes). This tool also allows developers to program their own nodes as Javascript functions. Node.js and Javascript libraries are also available to facilitate the use of their API.

Neura [NEURA] and TempoIQ [TEMPOIQ] are platforms for collecting sensor data through a RESTful API. While Neura is more person-oriented, TempoIQ is a general-purpose data collector. TempoIQ (former TempoDB) also offers tools for monitoring and analyze this sensor data.

Finally, in [Bovet et al., 2013] the concept of storage registration is introduced for the WoT. In this storage approach a web client announces its interest of storing some sensor data to the server. The server will store the data until the web client requests removal or an expiration time is reached. This prevents the server from reaching its storage limit.

6.1.2.4 Proposed architecture

The purpose of this section is to present the architecture and announce the key parameters used to link the layers. From our real-world experiences collected during the INTEGRIS and FINESCE projects, we have found that dividing the Smart Grid into these logical layers poses some critical difficulties arisen from the fact that typically, IEDs are closed devices that do not allow implementing custom developments (e.g., security or information-exchange protocols)—as novel experimental devices do. Therefore, it is based on the flexible data management system detailed in section 6.1.1.

A. Specifications

In this proposal, a Thing can be understood as every device that has minimum communication, processing and storage capabilities so it is able to be sensed or actuated and can communicate with another device to send or receive data. The proposed architecture has those goals to satisfy:

- As standard Web protocols such as HTTP or WebSockets use resources that may not be available on some constrained Things, this architecture will connect them using a variety of protocols (but ideally one) like CoAP or MQTT that are more suitable for constrained devices.
- Every Thing must be using an open standard protocol and must be understandable by the architecture to connect to the WoT. Gateways serving as a proxy for those Things that use proprietary or non-understandable protocols should be used.
- Things that cannot connect to the WoT infrastructure directly should connect to a gateway instead and let that gateway do the connection for them.
- This architecture will allow developers to interact with Things without knowing the protocol they are really using to communicate; hence there is a need to provide developers with a communication abstraction layer.
- It must allow on-demand deployment due to the fact that the number of Things will increase and therefore more resources will be needed to connect them.
- This architecture should be able to balance the load on its servers. Load balancers will prevent or minimize bottle-necks.
- Things will become virtual objects for the architecture and those virtual objects could be aggregated and linked to form other virtual objects.
- Eventually, every virtual object will be accessible by a RESTful URI.
- The architecture must have authorization of Things to interact with them to preserve owner's privacy.
- Third-party applications must have the architecture authorization to interact with Things.
- Every Thing must have an owner.
- Things can be queried and discovered.

In Figure 6.9 we can see how Things are connected to servers directly or via a gateway/aggregator being part of the WoT.

B. Protocols

While the use of one protocol like HTTP for all the Web of Things would be desirable, it is obvious that it is not achievable due to the heterogeneity of devices. So, a WoT architecture would need interoperability with a variety of protocols to interconnect devices. As the Web uses HTTP to communicate, is mandatory for the architecture to support this protocol. It will also provide a RESTful communication using URIs to identify resources and methods to actuate or sense on them. With IPv4 still in use and incapable of addressing all internet connected devices, internet connection to personal computers, for example, is available through port translation mechanisms. This translation makes them incapable of receiving data asynchronously from a server, hence the use of some sort of mechanism to allow them to receive data in an asynchronous manner is required. In addition, WebSockets protocol can overcome this constraint by providing a mechanism for browser-based applications that need two-way communication with servers.

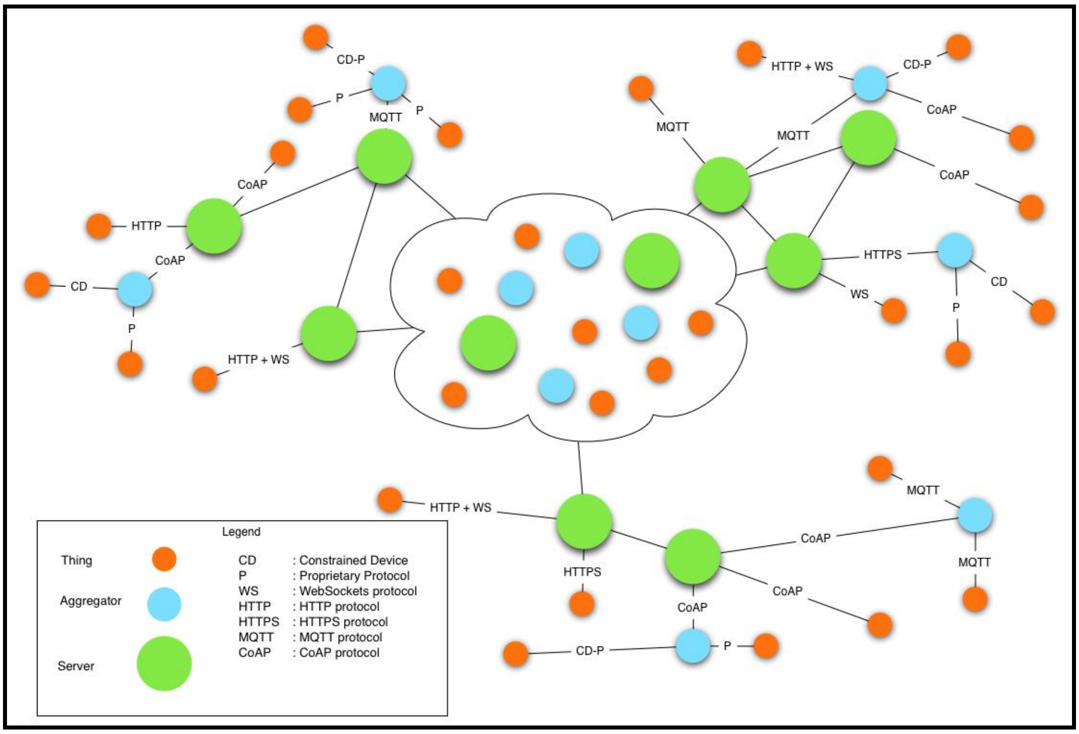


Figure 6.9 Proposed WoT architecture

As described in the previous section, MQTT and CoAP seem to be very suitable protocols for the WoT world. We propose to use these protocols but the architecture should be capable of understanding more protocols (e.g. DDS).

C. Layered and modular design

To be able to seamlessly intercommunicate these protocols and help developers implement and provide different implementations of functional units, (e.g., how a protocol is handled, adding a new protocol to the framework, how a VO is accessed, etc.) the architecture relies on a layered and modular design. It is composed of five main layers (Figure 6.10), they are:

- WoT Protocol Abstraction Layer: The purpose of this layer is to provide an abstraction mechanism for developers to interact with Things.
- REST to VO/Query: The goal of this layer is to map dynamically generated URIs to virtual objects (VO). It is an interface to sense or actuate on VOs, that is, Things. It is also an endpoint for querying Things like the temperature on a specific location.
- Auth: This layer is responsible for requesting and granting access between Things. As mentioned in section IV, every Thing must have an owner, therefore, WoT servers must request access to Things and VOs must request access to other VOs.
- Virtual Object Space: This is the space for virtual objects. Things will become virtual objects in the virtual world. Those VOs could be then sensed, actuated and aggregated.
- Proxy Layer: This abstraction layer will serve as a proxy for servers to communicate. A protocol for this communication must be defined for the architecture.

Those layers will be backed up by the flexible data storage and the context-aware broker reasoner modules. As the WoT architecture will generate URIs to address Things, a distributed database will be suitable for routing request based on their URI. Given a query, servers can first search in this database if this resource has been discovered beforehand. If this were the case, no discovery protocols would be needed; otherwise, a discovery protocol would start a search to find the resource and once discovered, the resource URI would be stored in the distributed database. Servers may also have local storage to implement the storage registration mechanism.

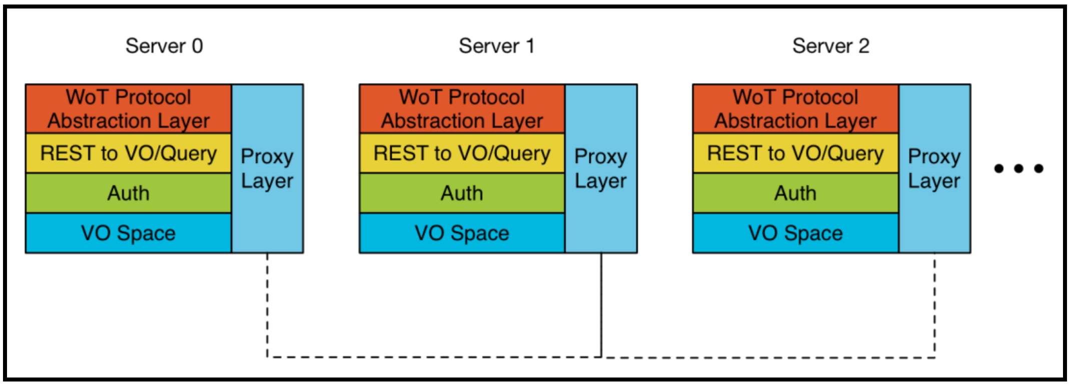


Figure 6.10 Layered and modular design

The WoT architecture presented in this section is based on the architecture proposed by D. Guinard. This section is focused on the architecture but we will highlight main similarities between two architectures.

On the other hand, the definition of each of the layers of the proposed architecture and its correspondence with the Guinard's proposal would be:

- Virtual Object Space: The layer that shares more similarities in Guinard's architecture is the Composition Layer, but in his approach this layer is a Physical Mashup where web services and enabled smart devices can create composite applications. In the proposed system and as in [Oriwoh et al., 2015], this layer serves the purpose of digitally enabling smart devices for their use by the architecture and by WoT users. Then, applications built on the upper layer of the architecture (e.g. Physical Mashups), could use these digitally enabled smart devices.
- Auth: This layer shares the same purpose as the Sharing layer exposed in Guinard's thesis. The goal is to preserve the privacy of each Thing, allowing Things to have owners that share the capabilities of their Things. A great approach to this authentication and authorization layer would be the same as the one proposed in [Guinard, 2011].
- REST to VO / Query: The goal of this layer is to dynamically generate meaningful, RESTful URIs for VOs as they are queried for the first time. Once the device has a meaningful base URI, its capabilities can be exposed through expanding its URI and the common HTTP verbs. Querying Things will involve complex semantic processing and a process such as the described in EVERYTHING [EVERYTHING].
- WoT Protocol Abstraction Layer and Proxy Layer: As several protocols can join the WoT architecture, there is a need to unify those protocols at the entrance of the proposed

architecture (WoT Protocol Abstraction Layer), translating those protocols to an internal language. This will allow the development of the inner layers of the architecture regardless of the protocol Things are using to connect to the architecture. This layer would accomplish the functions of the Accessibility Layer proposed by Guinard. In Figure 6.11, the correspondence between the layers defined by Guinard and the layers of the proposal is presented.

As the inner architecture should be agnostic from the outside protocols (those used by Things and the architecture to exchange information), there is a need to develop an internal format and mechanism (Proxy Layer) to pass messages between different nodes of the architecture. The next section shows a proof of concept of this idea. Firstly, the used dataset for testing is described and, secondly, the discovery algorithm is depicted. Finally, two implementation approaches are presented and the results obtained are discussed.

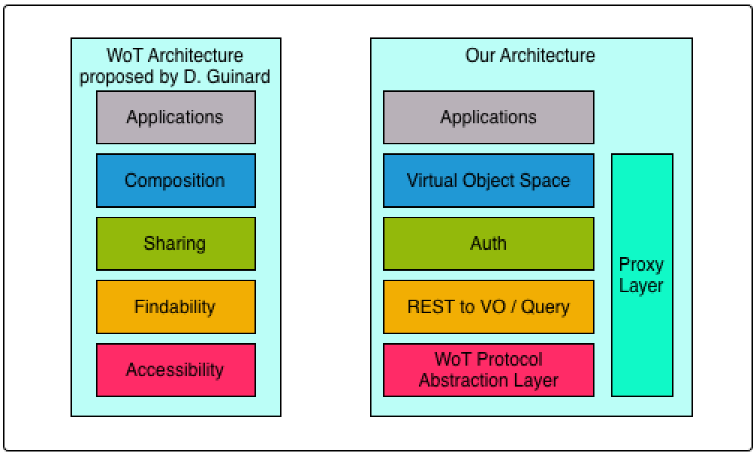


Figure 6.11 Comparison between the proposed architecture and Guinard's one

Taking into account the heterogeneous scenario of the IoT and WoT, this proposal is inspired in the service composition research seen and the service modularization and Future Networks requirements seen in previous chapters. Thus, it is designed to be flexible and context-aware, allowing an easily adaptation to new mechanisms and protocols.

6.1.2.5 Proof-of-concept

A. Dataset

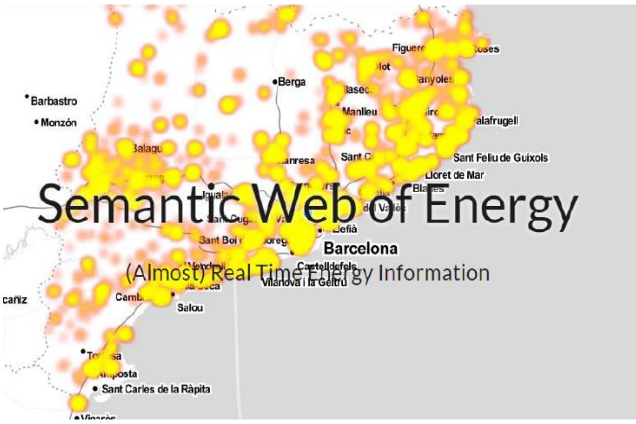
Two different datasets are used for performing the tests of this proof of concept. First one come from public data from the National Statistics Institute of Spain (http://www.ine.es) concerning cities/towns and inhabitants of Spain's municipality. Specifically, all the inhabitants of the Catalonia region have been accounted (about 7.5 millions), taking into account the following two assumptions:

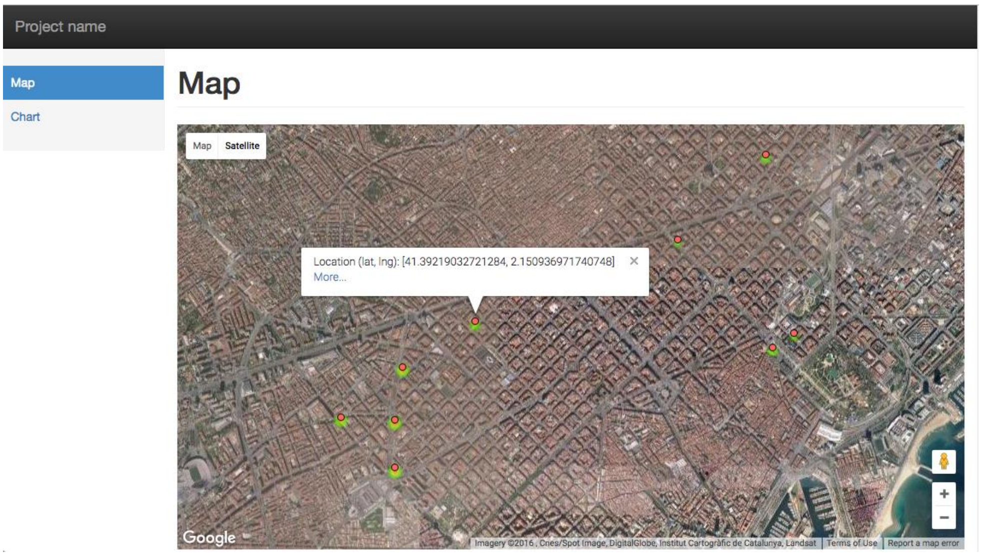
- In each house live an average of three people.
- Only 1% of households have some functional and Internet-controllable device.

Each house has been assigned to a city, and then a device is randomly assigned to each house. Different types of sensors are chosen, including temperature sensors, electricity meters, etc.

On the other hand, it was tested also with the *World's Largest Energy Data Resource* provided by Pecan Street Research Institute [PECANSTREET]. Pecan Street has a subscriber base, where participants may opt in to have their residence's electricity and power usage data monitored in high-resolution, including one-minute interval data measured at the home-, circuit-. Electric data can be also viewed on any web-enabled device and it is integrated in the WoE data. Displays are updated every second, giving immediate feedback on any load or generation changes. Finally, data is pushed to an external server via XML API where it is maintained, managed, and owned by Pecan Street, Inc. While Pecan Street has expanded nationally, the source of data collection for this study is solely 722 single-family homes in Austin, Texas, which are generally homes of newer construction, and ones in which there is already a high penetration of both solar PV installations and electric vehicle charging loads.

However, for each city it has been possible to obtain geolocation data (longitude and latitude) and it has been geographical represented on the map. Figure 6.12 shows the snapshots from the created application, called Semantic Web of Energy. The control of these devices is displayed, indicating in color the answer to a request for consultation (Discovery of Thing), and it allows to zoom in a specific zone and obtain further metadata of the sensors and historical graphic of the energy values captured by each sensor.





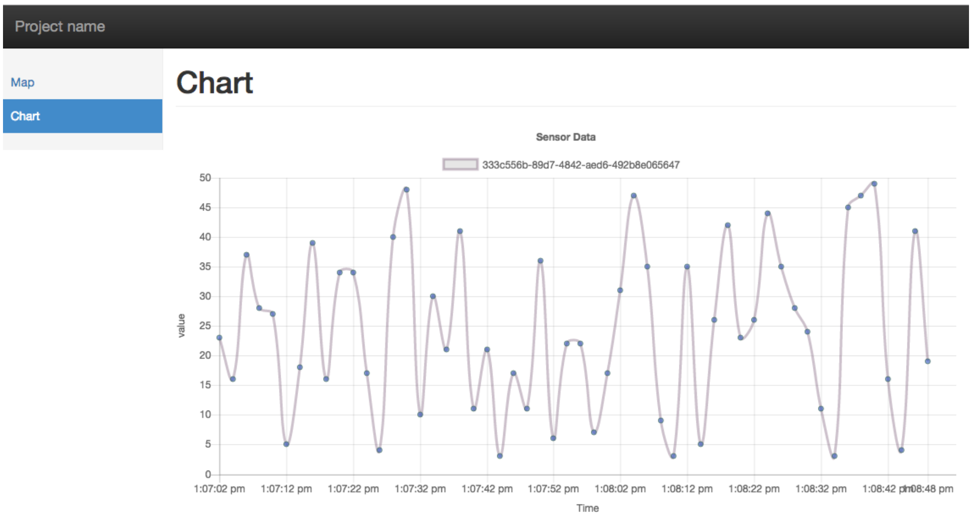


Figure 6.12 Snapshots of the representation of the devices response

B. Algorithm

Based on the assumptions outlined above a first version of the architecture presented in this section is implemented. In this sense we have implemented the first "Discovery of Things" functionality. It is simulated by the example of how to access a temperature sensor using the steps described in the following algorithm:

1. User A sends a request R1 to its local dispatcher node D1 (located in Madrid).

```
P1 = {"action":"GET", "what": "temperature", "loc": "Carrer de Sants"}
```

2. D1 processes the address ("Carrer de Sants"), searches for the Barcelona IP address and sends a response to user A.

```
R1 = {"request":{"what": "ws-conn" "to" "@IP-BarcelonaNode"}}
```

3. User A sends a request P2 to the node with the IP address @IP-Barcelona.

```
P2 = {"request":{"what": "ws-conn"}}
```

4. Barcelona's D2 dispatcher receives P2 and sends a response R2.

```
R2 = {"uri":{"method": "GET", "uri": "/ws/id/1"}}
```

5. User A connects to @IP-Barcelona/ws/id/1 via WebSockets (WS1 connection) and resends P1 through this connection. This connection will be managed by N1 node (server).

6. N1 processes the requests and searches for an address to forward the request and receive the temperature of Carrer de Sants. N1 has the private address of an aggregator/gateway (A1) that can provide the result of the query. N1 sends a CoAP request P2 to A1.

```
P2 = {"action":"GET", "what": "temperature"}
```

7. A1 processes the request and sends a response R3 to N1.

```
R3 = {"uri":{"method":"GET", "uri":"/temperature"}, "data":{"value":
15, "unit":"celsius" }}
```

A1 also saves an URI map to its local database: `saveUriMap:`

```
"(GET)/temperature" -> (function to get data)
```

8. N1 receives R3 response and saves to its local database:

```
saveUriMap:"/carrer-de-sants/" -> (GET)@IP-A1/temperature.
```

N1 also saves a query map to the distributed database:

```
saveQueryMap:"Carrer de Sants, temperature" -> "(GET)@IP-N1/carrer-de-sants/temperature".
```

N1 sends a R4 response to user A.

```
R3 = {"uri": {"method": "GET", "uri": "/carrer-de-sants/temperature"}, "data": {"value": 15, "unit": "celsius"}}.
```

Moreover, if N1 has users that need the same data, sends R4 to these users too.

9. User A receives the requested data and a URI to identify the resource and made future requests to it.

If the same user A wants to request the same data again, he will send an URI request ((GET)@IP-N1/carrer-de-sants/temperature) directly to N1 and as query and uri maps are now present, the response will return faster as less processing time will be needed.

If user B sends the same request P1 to D1, D1 will respond saying that user B must request a WebSocket connection to N1 and send a request to this URI "(GET)@IP-N1/carrer-de-sants/temperature".

As shown, the framework is capable of handling the discovery of the new object and further processing and storage in the system for future reference.

C. Implementation

In this section two implementation approaches are compared.

First approach. As a lot of services found in the Web 2.0 are implemented using the PHP language[38] and in order to ease Web of Things applications at every layer for experienced PHP developers, we have developed a prototype implementing the algorithm presented in this section using the PHP language. Although we have succeeded in developing basic functionalities using PHP, several problems have arisen during the implementation.

PHP was born to serve CGIs, e.g. serve HTTP requests with dynamic content. The main workflow where PHP has been used consists in three steps: load, execute and die. For this reason, few efforts have been made to solve problems such as memory leaks or the fact that executing one PHP statement requires more low-level instructions than the actually needed ones.

[38] http://trends.builtwith.com/framework

In recent years, PHP has undergone some changes in its usages and performance:

- With the arrival of Node.js [NodeJS], some PHP developers started to implement libraries with the objective of allowing PHP users to create servers in this language as in React PHP [REACTPHP], instead of relying on web servers like Apache HTTP.

- Compiled PHP frameworks such as the Phalcon Framework [PHALCONPHP] with a high-performance boost on execution time have motivated the release of PHP 7, solving memory leaks and decreasing the number of low-level instructions needed to execute a PHP statement.

In order to build the prototype, we have used the Phalcon Framework and React PHP to boost performance. The Phalcon Framework presents high speed in performing operations and React PHP presents a novel manner for building PHP applications using the reactor pattern and asynchronous programming. However, the immaturity of React PHP and the lack of asynchronous libraries in PHP present an obstacle for developers to use even basic technologies such as MongoDB.

In this way, PHP has succeeded in building and fast-prototyping Web applications thanks to its low learning curve and its dynamic typing. However, its use at low-level/core layers in a WoT/IoT architecture where program correctness is crucial, facilitates the appearance of execution time errors.

Second approach. After evaluating other solutions in the market, the Scala [Odersky et al., 2004] language has been selected for a re-implementation of the prototype. Although this language usage is not wider than PHP, Scala is experiencing an adoption growth[39].

Moreover, this language presents key characteristics that make it suitable for building a future WoT prototype architecture. Scala has also been chosen due to its research community. In fact, Scala was born at the EPFL (École Polytechnique Fédérale de Lausanne) thanks to Martin Odersky. Therefore, continuous investigation is being made in order to optimize speed and provide better API's.

As a result of the studied approaches, a comparison table between PHP and Scala is presented below (Table 6.2). Data presented in this comparison table have been extracted from the experimentation. Speed has been extracted from benchmarks done in [Fulgham et al., 2009].

6.1.2.6 Conclusions

A new framework approach and a proof of concept have been presented in this section. It has been shown that the proposed architecture is feasible and that the implementation of successive parts can be made using this design. Regarding its design, the proposed WoT architecture follows the service modularization perspective seen in previous chapters, and elaborates a design highly adaptable to different contexts and new requirements that Future Networks can bring. It was designed to dynamically change a protocol or mechanism to adapt the connection to the requirements of the specific communication.

[39] http://www.indeed.com/jobtrends?q=scala&l=

Table 6.2 Benchmarking PHP vs Scala languages

Feature	PHP	Scala
Natural workflow	Load, execute, die	Always running on a JVM
Speed	1	28 times faster
Backed by research	No	Yes
Libraries	PHP, C extensions	JVM compliant libraries
Distributed libraries	As C extensions	Akka and other JVM libraries
Async libraries/constructs	ReactPHP	Akka and Futures
Typing	Dynamic	Static

Focusing on the research of its implemtation, although the results exposed are promising, we have realized that PHP lacks libraries and implementation for the most relevant WoT protocols. If there is a valid implementation, it only covers basic features of the protocol.

Moreover, the reactor approach was performed to prototype the architecture using ReactPHP and it was found that there are not libraries to connect the most commonly used databases in distributed systems like Redis or MongoDB, or the WoT protocols.

For the reasons exposed, it seems that PHP is not mature enough for the purpose, that is, to develop a holistic architecture for the Web of Things. A reimplementation of the architecture using the Scala language has been made, speeding up its performance and opening up the possibility to take advantage of robust libraries and frameworks built on top of JVM compliant languages.

Heterogeneity, parallelization and distribution as explained in [Odersky et al., 2004] are also a key characteristic of a WoT architecture. More work has to be done to fully achieve these characteristics. The Actor Model [Miorandi et al., 2012] seems well suited to build an architecture with such characteristics as asynchronous messaging, location transparency, distribution and concurrency as its core principles.

6.1.3 The Smart Grid context-aware security

Following the discussion started on section 2.2.1, this section extends the analysis of the security threads for the Smart Grid and presents a concrete example of service composition in the field of Smart Grid focused on the dynamic configuration of the security mechanisms in the acquisition of Smart Metering data.

The main novelty of the Smart Grid is the addition of a telecommunication network to the electrical infrastructure in order to transport information such as the state of the grid, real-time power consumption, service fault locations, etc. Smart Grid brings Power network technology has been exceptionally stable for long time, which is in contrast with the fast evolution of ICT systems. In other terms, the objective of the Smart Grid is to ensure that the grid is economically efficient, sustainable and provides higher standards of power quality thanks to a lower level of losses and enhanced power management and security [Navarro et al., 2013].

However, as described before in the Smart Grid general requirements and challenges (section 2.2.1), there are still some concerning issues on the integration of telecommunication and electric power networks. Smart Grid evolution presents many operational problems that cannot be solved by current systems and technologies especially if they are used isolated. Fortunately, the evolution and current maturity of ICT systems makes it possible to cope with the mentioned problems, especially over the distribution grid where today ICT systems are scarcely deployed.

Regarding cyber security, the main parameters to consider are integrity and confidentiality, but also reliability and latency should also be taken into account. As it is also highlighted in section 2.2.1, it is important to underline the low latency and very high reliability needed for Smart Grid Active Protection Functions (APF) (see Table 6.3).

Table 6.3 Functional classes and requirements [Selga et al., 2014]

Function	Latency	Reliability	Integrity	Confidentiality
Active Protection Functions	<20ms	Very High (99,999%)	High	Low
Command and Regulations	<2s	High (99,99%)	High	Low
Monitoring and Analysis	<2s	High (99,99%)	High	Low
Advanced Meter and Supply	<5min <10s	Low (99%)	High	High
Demand Response	<5mim <5s	Medium (99,9%)	High	Low

This is always difficult to achieve with current technologies, but also the high reliability needed for Commands and Monitoring is not easy to achieve in practice in a distribution grid environment. This very high reliability required implies that special care has to be taken to minimize DoS attacks to a minimum [Selga et al., 2014]. Before going into deep on other security issues of the Smart Grid, the following section presents a brief review made about the high level of reliability needed in the Smart Grid and possible mechanisms and protocols to achieve it.

6.1.3.1 Redundancy and high availability

Security in Smart Grids is essential for the survival and feasibility of the global electricity distribution concept [IEC 62351] but for its achievement it is necessary to phase out the big challenges posed by the vulnerabilities inherited from Internet plus the new ones coming from the different applications, requirements and actors interacting together in a Smart Grid. Therefore, the Smart Grid has its own specificities concerning security that need to be considered, the strongest requirement being the need to continue securely operating even upon temporary communication disconnections due to communication network partitions. That forces the distribution of security servers and repositories to avoid a single point of failure.

Smart Grids control systems have high QoS requirements due to the accurate precision required to take measures and control an energy transformation plant in a secure and feasible way. Some industrial machines require real-time synchronization between the master node and slaves, limiting the response time in the order of a few milliseconds and even microseconds.

Security systems and its critical traffic set a number of key factors to follow for a good network performance: (1) Deterministic behavior in real-time and (2) High-availability. If the grace period is the maximum amount of time that an industrial installation tolerates a failure in the system, the recovery time must be lower than the grace period to let the Smart Grid operate in case that a problem appears in the network. The International Electrotechnical Comission (IEC) proposes a set of values of grace period depending on the application [IEC 62439] [US, 2010] [Zaballos et al., 2013] (see Table 6.4).

Table 6.4 Grace Time Classification

Application	Grace Time
Management Systems	20s
Automated management systems	2s
General Automation (Energy plants)	200ms
Critical Time (Synch. systems)	20ms

There are some communication protocols capable to cope with different types of redundancy by providing different recovery times. Depending on the chosen protocol, several resources will be consumed and the performance of the network would vary, affecting directly to the service. In fact, current ICT technologies unfortunately have big difficulties in meeting the latency requirements either at link layer [Selga et al., 2014] [Yan et al., 2012] or at routing layer [Zaballos et al., 2013] [Li et al., 2010]. Some link layer protocols have been evaluated [INTEGRIS] [Selga et al., 2013] as summarized in the qualitative benchmarking of Table 6.5.

Main protocols considered in the analysis are briefly described in what follows:

- Spanning Tree Protocol (STP) [Wodjak, 2003]: STP is standardized by the IEEE 802.1d as a cold standby protocol that does not use blocked-redundant links. This fact affects directly to the recovery time which varies between 30 and 60 seconds plus the overload of the root node. There are improvements such as Rapid Spanning Tree Protocol (RSTP - IEEE 802.1w), in order to reduce the recovery time from 1 to 3 seconds, depending on the network topology.
- TRansparent Interconnection of Lots of Links (TRILL) [Touch et al., 2009]: TRILL was designed to update STP looking for a better usage of redundancy and performance. This protocol is based on the IS-IS routing protocol and allows the management of the loops,

maximizing the use of the links. The recovery time is approximately less than a second with a limit of 700 devices.

Table 6.5 Smart Grid redundancy protocols benchmarking

Metric	Meaning
5	Totally Applies
4	Most Cases
3	Half/Partial Cases
2	Some Cases
1	Few Cases
0	Does not apply

	PRP	HSR	DRP	CRP	Fast MRP	MRP	TRILL	RSTP
Simultaneous Use of Paths	**5**	**5**	**3**	**3**	**0**	**0**	**0**	**0**
Redundant Paths	**5**	**3**	**3**	**5**	**3**	**3**	**5**	**5**
Disjoint Paths	**5**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
Active (5), Hot Standby (3), Cold Standby (0)	**5**	**5**	**3**	**3**	**3**	**3**	**0**	**0**
Centralized (5), Distributed (0)	**0**	**0**	**0**	**0**	**5**	**5**	**0**	**0**
Scalability (>50 devices)	**5**	**5**	**3**	**5**	**5**	**5**	**5**	**1**
Scalability (>100 devices)	**5**	**5**	**1**	**5**	**2**	**2**	**5**	**0**
Scalability (>250 devices)	**5**	**3**	**0**	**5**	**0**	**0**	**5**	**0**
Link Protection	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
Node Protection	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
Path Protection	**5**	**5**	**0**	**0**	**0**	**0**	**0**	**0**
End-devices mechanism	**5**	**0**	**0**	**5**	**0**	**0**	**0**	**0**
Switches mechanism	**0**	**5**	**5**	**0**	**5**	**5**	**5**	**5**
Routers mechanism	**0**	**0**	**0**	**0**	**0**	**0**	**5**	**0**
Ring Topology	**5**	**5**	**5**	**0**	**5**	**5**	**5**	**5**
Double Ring Topology	**0**	**5**	**5**	**0**	**0**	**0**	**0**	**0**
Mesh Topology	**5**	**0**	**0**	**5**	**0**	**0**	**5**	**5**
Tree Topology	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Star Topology	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Internode-aware mechanism	**0**	**5**	**5**	**0**	**5**	**5**	**5**	**5**
Interoperability mechanism	**3**	**3**	**0**	**3**	**0**	**0**	**3**	**0**
Synchronism mechanism	**0**	**5**	**5**	**0**	**5**	**5**	**0**	**0**
Recovery Time (<= 0s)	**5**	**5**	**0**	**0**	**1**	**0**	**0**	**0**
Recovery Time (< 50ms)	**5**	**5**	**2**	**2**	**3**	**0**	**0**	**0**
Recovery Time (< 100ms)	**5**	**5**	**4**	**3**	**5**	**2**	**1**	**1**
Recovery Time (< 500ms)	**5**	**5**	**4**	**4**	**5**	**4**	**4**	**3**
Recovery Time (< 1000ms)	**5**	**5**	**5**	**4**	**5**	**5**	**5**	**4**
Recovery Time (< 3s)	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**4**
Recovery Time (< 60s)	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**

- Media Redundancy Protocol (MRP) [Giorgetti et al., 2013]: MRP is standardized in IEC 62439-2 and manages the redundancy of ring topology networks. The recovery time specified by the IEC varies between 500ms and 200ms, depending on the profile of the network. MPR works for more than 50 nodes. Fast MRP can reach recovery times from 5 to 20 milliseconds.
- Cross-network Redundancy Protocol (CRP) [Antonova et al., 2011]: IEC 62439-4 protocol is installed in the end devices operated in a distributed way. Each node must be connected

to two different network devices, sending periodically a diagnosis frame from each of its interfaces. The recovery time for 6 switches with 100Mbps of bandwidth is around 800ms.

- Distributed Redundancy Protocol (DRP) [Kirrmann et al., 2012], IEC 62439-6, is used in ring networks. Its recovery time is deterministic with a period of time called macrocycles. DRP synchronizes all the nodes from the network and, after a macrocyle, one node will have the turn to send a Ring Check through its two interfaces every macrocycle. Moreover, every node in the network will send a Link Check to its neighbors to detect a possible failure. In case that some Link Check miss, the node modify the state of the link to BLOCKING, sending a multicast message, Link Alarm Frame, to the other network nodes. Approximately, in a 100 Mbps network with 50 nodes the recovery time is 85ms.
- Parallel Redundancy Protocol (PRP) [Kirrmann et al., 2007]: IEC 62439-3 protocol bases its operation on a complete duplication of the protected network. The main and the secondary paths are totally disjoint. The main objective is to achieve a recovery time of 0 seconds in case of failure, being an ideal protocol to protect real-time applications that have a grace time of few microseconds.
- High-Availability Seamless Redundancy Protocol (HSR) [Kirrmann et al, 2009]: IEC 62439-3 protocol is developed to overcome the deficiencies of PRP by reducing the cost of the deployment. It works only inside ring topology networks.

All these protocols are standardized and available to be deployed in a Smart Grid environment. In Table 6.5, the different characteristics of those protocols are graded (from 5, when it totally applies to 0, when it does not apply), depending on the level of support to the evaluated feature. The criteria and grades proposed were agreed by a set of other experts involved in INTEGRIS project. The selection of one or another depends on the different requirements demanded by each Smart Grid function (e.g. Active ProtectionFunctions require the highest level of reliability and the lowest latency possible). Therefore, these performance and security requirements, besides other context of the deployed use case (e.g. topology of the network), should be matched with the defined characteristics in Table 6.5 in order to select one or another technology. This selection must rely on an intelligent management system in charge of selecting the most appropriate protocol. That system could be provided by the context-aware broker abovementioned at the beginning in section 6.1 (Figure 6.1). It will be an even more powerful management tool if it can be used with real-time information collected by the IoT infrastructure of the Smart Grid, as the one presented in previous section 0.

6.1.3.2 The security thread

Besides all the benefits previously said, the evolution on the remote control of the electrical distribution grids could give back undesirable vulnerabilities if the systems are not correctly secured. Smart grid network control and monitoring are very important features in order to provide distributed energy generation and storage, quality of service (QoS) and security. Smart Grids link many distinct types of devices – also referred to as intelligent electronic devices (IEDs) – demanding very different QoS levels over different physical media. Indeed, this kind of data network is not exempt from the growing needs of cyber security. In addition, availability and secured communications are also crucial for the proper network operation [Yan et al., 2012] what drives practitioners to consider Active Network Management (ANM) techniques to coordinate the whole communication network.

In addition to this, the Smart Grid relies on sensors, actuators and a management network, usually controlled by Supervisory Control and Data Acquisition systems (SCADA), which are used to control and supervise industrial processes from a computer. That is to say, SCADA systems control items in the physical world through computer systems. This is one of the points in which the main security concern of Smart Grids relies. Some recent cases have demonstrated the critical relevance of it.

- One of the most famous cases in this matter is Stuxnet [Virvilis et al., 2013], a very complex worm and Trojan discovered in June 2010 that attacked the Iranian nuclear enrichment program. Its code used 7 different mechanisms to expand itself, mainly exploiting 0-day vulnerabilities. It achieved the destruction of about a thousand nuclear centrifuges by changing the behavior of the actuators while telling the sensors that everything was good.
- A year later, in September 2011, a new Trojan called DuQu was discovered presenting a very similar behavior to Stuxnet so it is believed that the two worms were related [He, 2014].
- Stuxnet was purportedly used again in 2012 against a nuclear power plant in southern Iran but the damage could be avoided by taking timely measures with the cooperation of skilled hackers.
- In 2013, Iran hacked US Energy Companies (oil, gas and power) and was able to gain access to control-system software and was also accused of launching DDoS (Distributed Denial of Service) to US banks.

Cyberspace is defined as "an operational domain whose distinctive and unique character is framed by the use of electronics and the electromagnetic spectrum to create, store, modify, exchange and exploit information via interconnected information communication technology (ICT) based systems and their associated infrastructures" [Parks, 2011]. Thus, cyber warfare is the kind of war that happens in that space in contrast with the traditional kinetic warfare where physical weapons are used. Smart Grids have become a clear potential objective of cyber warfare considering that nowadays almost everything runs on electrical power and therefore potentially causing outages or, even worse, causing damage especially in some kinds of power plants (e.g. hydroelectric, nuclear, etc.). As brushed up in this section, this kind of attacks are becoming a reality and, recently, the information leakage of the US government that has brought to light by several initiatives like PRISM [Barret, 2013], has also revealed that USA has drawn up an overseas target list for cyber-attacks.

Regarding cyber security standards, many of the existing ones are to be taken into account in the Smart Grid as is highlighted in NISTIR 7628 [NISTIR7628, 2014], where they are listed and commented. A relevant one among them is ISO-IEC62351-6 [IEC 62351] because it is the cyber security standard of reference for IEC 61850 and, thus, for the Smart Grid. NISTIR 7628 gives guidelines for cyber security implementation in the Smart Grid and provides a logical security architecture of general nature. Significantly, it contains interesting considerations regarding the use of authentication certificates and secret keys management.

Table 6.6 was developed jointly with Smart Grid experts from industry and academia, presenting a table with a set of the most important security issues that can affect the proposed infrastructure for the FINESCE's Smart Grid [FINESCE]. The main goal is to establish an order of implementation priorities regarding the security aspects. Authors gathered this information from several utilities in order to establish these priorities by numbering them with numbers from 1 to 8 (1 being the highest priority and 8 the lowest). Utilities have to provide the impact level for every problem if the system is crashed down. In the Reason column of Table 6.6 is presented a brief explanation of the rationale behind the order and decisions of which aspects are more critical than others.

Table 6.6 Smart Grid Security Issues

Security Issue		Problem Description	Priority	Reason	Impact
Data Security	Data Leakage	Data is stolen and delivered without permission of the proprietary.	5	If a malicious user can access the system, user stored data could be compromised. This fact could derive in legal problems.	Very High
	Data Forgery	Data is modified by a malicious user and not detected.	6	Once the access is accomplished, if notifications of changes are not considered, a malicious user could modify user stored data.	High
	Data Lost	Data is erased by a malicious user or a human error.	7	If a backup system is maintained, this could be an important but not critical problem since data could be restored	Med.
Network Security	Data Transaction	Data is delivered through the network and could be visible to malicious users if it is not encrypted. It depends on the sensibility of the data transmitted that this issue becomes more critical.	1	It is not necessary to access the system to obtain data under these circumstances. Therefore, it is considered that the most important aspect is that data transactions (data in transit) are encrypted.	Very High
	Commands execution	Many applications that can reside in IEDs could be sensitive to latency. A DoS attack to the network resources could affect its performance. It affects availability of the services.	8	Network resources have to be controlled because the access to data stored and applications in IEDs depends on them. It is considered that network will be designed to detect DoS attacks and avoid latency problems.	High
Authentication		Access to IEDs and data storage has to be controlled and tracked to avoid wrong usage. It affects confidentiality, integrity and availability if a malicious user gets a user with rights granted.	2	It is very important to maintain control over the users that access data stored in IEDs and track the actions they perform to avoid problems with data stored and IEDs functionality. If a wrong usage is detected and users are authenticated, the system can isolate the problematic user to avoid damage.	Very High
Authorization		Not all users have the same authorization policies to different zones, resources or stored data. Admin users, privileged users, guest users and third party users must be catalogued with different authorization rules.	3	It is important to maintain isolated rights to access resources because the system could have third-party users, guests/clients, administrators, etc and not all should have complete access. The system could be modified by users without complete knowledge or by malicious users if a good authorization policy is not applied.	High
Identity Management (IdM)		Keep a good connection between users and authorization rules by implementing a robust IdM.	4	Necessary to map users with their respective authorization rules and to maintain control over granted access to the system.	Very High

The state of the art regarding security in the Smart Grid is in fact defined in the mentioned IEC62351-6 standard which basically applies security at transport layer (TLS1.0 [RFC 2246] with some restrictions) and upper layer communication protocols. It could be argued that protecting the transport layer could be enough since this may provide confidentiality, integrity and device authentication for user data and because many commercial systems rely on protecting systems just like this. However, protecting the Smart Grid only at the transport layer leaves the network and its links open to cyber security attacks that may produce DoS, eavesdropping of network management messages and, thus, prohibiting the users from accessing the service. This fact is not aligned with the high reliability feature that is required in the Smart Grid [Ghafoor, 2014]. For this reason, the Smart Grid really urges multilevel security, even above the transport layer [Navarro et al., 2013] [Selga et al., 2014].

To face this challenge and secure the Smart Grid, FINESCE proposed a security system in a way that it is really deployable and operative and that (1) balances the many and sometimes conflicting security goals of the different actors and subsystems and (2) accommodates a large and dynamic set of security mechanisms. This is done by creating an entity in which to concentrate the distributed agents that provide service to the Smart Grid among which the security server and repository.

As it can be seen, all the concerns revolves around the efficient adaptation of the security mechanisms to the specific requirements of the Smart Grid function that is wanted to be deployed. Therefore, we envisage those entities including a context-aware Smart Grid management broker placed in different locations of the SDU and working coordinately in order to provide the adapted security mechanisms when and where needed. As an example of these dynamic security configuration, a use case focused on flexible smart metering security is shown in the next section.

6.1.3.3 Securing Smart Metering through service composition

This section summarizes the extension of Service Composition techniques in order to manage flexible Smart Grid applications. More concretely it develops the methodology around a proof-of-concept use case that focuses on the securing the Advanced Metering Infrastructure of the Smart Grid. However, the obtained results can be extended to the whole Smart Grid and are showcased in real and modern applications.

The designed use cases are focused on collecting encrypted smart meter and Remote Terminal Units (RTU) data and saving it encrypted in some place (the Cloud, a concentrator, the IEDs, etc.) in a way that the authorized actors can work with the data without having access to user specific data, preserving their customers' anonymity and protecting them from malicious attacks.

As stated before, smart metering represents only a set of the Smart Grid solutions, but it is the part that has already been more regulated, deployed and tested around the world. Advanced Metering Infrastructure (AMI) consists of smart meters, data management, communication network and applications. AMI is one of the three main anchors of Smart Grids along with Distributed Energy Resources (DER) and Advanced Distributed Automation (ADA). Smart metering is usually implemented using automatic meter reading (AMR), a technology that automatically gathers data from energy, gas and water metering devices and transfers it to the central office in order to analyze it for billing or demand side management purposes. Data is read remotely, without the need to physically access the meter. AMR systems are made up of three basic components to be secured: the meter, the Central Office and the communication systems. AMR includes mobile technologies, based on radio frequency, transmission over the electric cables (power line), or telephonic platforms (wired or wireless) [Zaballos et al., 2011] [Selga et al., 2013].

First of all, in order to determine the security requirements of the smart metering function, it has been of great importance the work developed by NIST (National Institute of Standards and Technology) called NISTIR 7628 [NISTIR7628, 2014], because of its high detailed description of requirements and elements that must be taken into account when deploying a Smart Grid. Since this research targets to secure smart metering as a first approach, a limited set of requirements have been selected from among over 200 entries in [NISTIR7628, 2014], considering those that affect directly or indirectly to the smart metering. Once the requirements have been set up, it has been developed a chart with those requirements (Table 6.7) on one axis and the technologies that can be used to meet the requirements on the other axis. After studying in depth the selected requirements and technologies, the Table 6.7 specifies which secured-ICT technology can meet more accurately the requirements.

Technologies and techniques associated to each requirement have been selected based on the authors' experience developed during the INTEGRIS and FINESCE European projects and on some new state of the art techniques such as homomorphism that allows the information to be encrypted at all times, even when having to handle it, in contrast with other traditional techniques that require decryption to be performed before the information is treated and then encrypted again [Yukun et al., 2013]. Analogously to Table 6.3, the different technologies are graded (from 5, when it totally applies to 0, when it does not apply), depending on the level of support to the requirement. They were also selected based on the experience from INTEGRIS and FINESCE projects, besides an exhaustive review of the literature about Smart Grid security.

In addition, some other conclusions extracted from the experience acquired in those projects guided to define the following rules in the design of cybersecurity solutions for Smart Grid:

- To rely as much as possible on proven existing standards, only complementing them when strictly necessary. This comes from the evidence that the first versions of most standards contained serious vulnerabilities.
- From these standards, to choose the right options for the Smart Grid (see Table 6.7).
- To place cyber security services as close as needed to the sensing and actuation points to improve latency and reliability of applications. In fact, this is done based on Service Composition paradigm by placing them in the cyber security server and repository contained in the IEDs.
- To use a common coordinated cyber security data repository for all the involved technologies.
- To distribute this repository, either as a whole or partially, in the Cloud, although having also a central repository located elsewhere. The central cyber security repository is replicated so that, in case of disconnection, the system continues to work for some time even allowing the inclusion of new devices and functions.
- To define cyber security metrics to feed the context-aware system to enable improved system management.

- To adhere to the principle of the Trusted Computing Group (TCG) of using Trusted Platform Modules (TPM) [Shen et al., 2010] to protect in-built software and hardware as well as storage of data, including the basic keying material.
- To use, whenever feasible, authentication based on Certificates.

Table 6.7 Smart metering and smart grid services and features analysis

Metric	Meaning
5	Totally Applies
4	Applies a lot
3	Mostly Applies
2	Applies
1	Somewhat applies
0	Does not apply

	PKI	Encryption + Decryption (AES)	NAC	Checksum SHA	DoS Defense System	ACL (Different Layers)	IDS	IPS	NMS	Supervised Cognitive System	Unsupervised Cognitive System	Logging	Segmentation (VLAN, VRF,MPLS)	SSH	QoS	Format Check	Homomorphism
SG.SC-3 Security Function Isolation	**4**	**3**	**4**	**0**	**0**	**2**	**1**	**1**	**0**	**0**	**0**	**0**	**5**	**0**	**0**	**0**	**0**
SG.SC-4 Information remnants	**5**	**0**	**5**	**0**	**0**	**0**	**4**	**4**	**3**	**0**	**0**	**2**	**1**	**0**	**0**	**0**	**0**
SG.SC-5 DoS Protection	**0**	**0**	**0**	**0**	**5**	**3**	**4**	**4**	**0**	**0**	**2**	**2**	**0**	**0**	**0**	**0**	**0**
SG.SC-6 Resource Priority	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**0**	**1**	**0**	**5**	**0**	**0**
SG.SC-7 Boundary Protection	**5**	**1**	**4**	**0**	**0**	**4**	**3**	**3**	**1**	**0**	**0**	**1**	**5**	**0**	**0**	**0**	**0**
SG.SC-8 Communication Integrity	**5**	**3**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**5**	**4**
SG.SC-9 Communication confidentiality	**5**	**5**	**2**	**0**	**0**	**2**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**5**
SG.SC-10 Trusted Path	**5**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**4**	**1**	**3**	**0**	**0**	**0**	**0**	**0**	**1**
SG.SC-11 Crypto Key Establishment	**5**	**5**	**4**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**5**
SG.SC-12 Use of Validated Cryptography	**5**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**5**
SG.SC-15 PKI certificates	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SG.SC-19 Security Roles	**5**	**0**	**4**	**0**	**0**	**3**	**1**	**1**	**1**	**0**	**0**	**1**	**2**	**0**	**0**	**0**	**0**
SG.SC-20 Message Authenticity	**5**	**4**	**1**	**4**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**5**	**3**
SG.SC-26 Confidentiality at rest	**0**	**5**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**4**
SG.SC-29 Application Partitioning	**5**	**0**	**5**	**0**	**0**	**2**	**1**	**1**	**0**	**0**	**0**	**0**	**3**	**0**	**0**	**0**	**0**
SG.SI-2 Flaw Remediation	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**4**	**5**	**5**	**2**	**0**	**0**	**0**	**0**	**0**
SG.SI-3 Malicious Code and Spam protection	**0**	**2**	**0**	**0**	**0**	**5**	**5**	**5**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**5**
SG.SI-4 Information System Monitoring	**0**	**0**	**0**	**0**	**0**	**0**	**4**	**4**	**5**	**2**	**3**	**4**	**0**	**0**	**0**	**0**	**0**
SG.SI-7 Software and info integrity	**5**	**0**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**5**
SG.SI-8 Information Input Validation	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**5**	**0**
SG.AC-3 Account Management	**5**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SG.AC-8 Unsuccessful Login Attempts	**5**	**0**	**5**	**0**	**0**	**0**	**3**	**3**	**4**	**2**	**4**	**4**	**0**	**0**	**0**	**0**	**0**
SG.AC-11 Concurrent Session Control	**0**	**0**	**5**	**0**	**0**	**0**	**1**	**1**	**5**	**2**	**4**	**3**	**0**	**0**	**0**	**0**	**0**
SG.AC-13 Remote Session Termination	**5**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**
SG.AC-16 Wireless Access Restrictions	**5**	**4**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
SG.AC-17 Access control for portable and mobile devices	**5**	**0**	**5**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**
SG.AU-X Auditability	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**5**	**0**	**0**	**5**	**5**	**0**	**3**	**2**	**0**	**0**
SG.AU-16 Non repudiation	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**4**	**0**	**5**	**0**	**0**	**0**	**0**	**0**	**0**
SG.CM-x Configuration changes	**5**	**0**	**5**	**4**	**0**	**1**	**1**	**1**	**3**	**3**	**0**	**5**	**0**	**4**	**2**	**5**	**4**
SG.IA-5 Device Identification and Auth.	**5**	**0**	**5**	**0**	**0**	**0**	**3**	**3**	**2**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**
SG.MA-x Remote Maintenance	**5**	**4**	**5**	**1**	**0**	**2**	**2**	**0**	**3**	**3**	**2**	**4**	**1**	**5**	**2**	**1**	**1**

Many examples of Smart Metering use cases that should fulfill the security requirements of Table 6.7 can be defined [FINESCE] [INTEGRIS], such as the installation process of a smart meter, reading the power consumption, firmware updating, system monitoring, maintenance processes, etc. Aiming at just giving a proof-of-concept demonstration of some of this benefits that Service Composition and SDN could bring into Smart Grid, a basic use case was designed on securing the smart metering in a Software Defined Utility environment [FINESCE].

Combining SDN and Service Composition becomes especially powerful for the secure self-maintenance of networks, which represents a very important characteristic for the development of Smart Grids. They could be applied at different segments of the electrical distribution grid implementing easy and fast-to-deploy intelligent solutions.

The focus on the Service Composition interoperable standalone modules, which can be invoked or dropped on demand, leads to considerable cheap solutions in the field of Smart Grids and presents a solution that could be integrated incrementally. Moreover, it allows system architects to create flexible solutions that could be modified and evolved according to eventual new needs. Furthermore, the modularization of Smart Grid functionalities and encapsulation into self-contained services facilitate the distribution of the Smart Grid intelligence, approaching the reasoning and decision process and helping to handle its critical constraints of latency on fault reaction.

In this regard, the work conducted in [Gonzalez et al., 2013] is extended and presented a new taxonomy of services (Figure 6.13) for AMI security. This taxonomy classifies services into 6 different generic parameters: *Granularity, Execution, Scope* (Application/Network), *Purpose*, *Usage* (Mandatory/Optional) and *Order* (Dependent/Independent). All these criteria can be applied generically to any Smart Grid service, although the options shown in Figure 6.13 for *Purpose* criteria are limited to smart metering. Afterwards, the selected modules for smart metering are classified according to this taxonomy (Table 6.8).

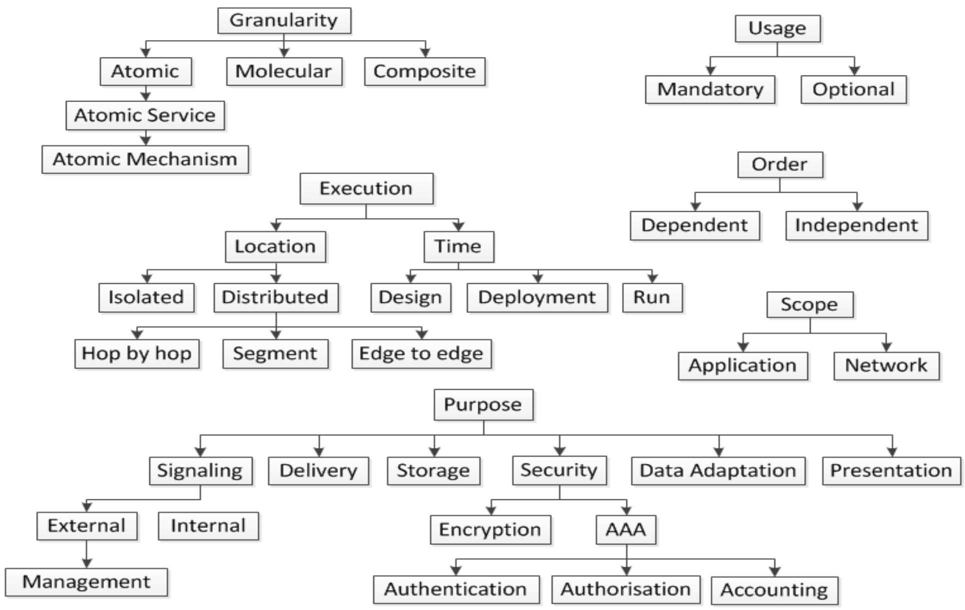


Figure 6.13 Extended taxonomy for secure smart metering

Table 6.8 Secure Smart Metering composition modules classification

Parameter	Consumption Increasing	Consumption Decreasing	Consumption Checking	Consumption Initializing	USB Authentication	User & Password Validation	User & Password Login	USB ID Login	Digital Certificate Login	Cypher key obtaining	Digital Certificate	Data Encryption	Data Decryption
Granularity	Atomic												
Scope	Application												
Execution location	Isolated / Segment / E2E			Segment / E2E								Isolated	
Execution Time	Run		Deployment	Run									
Purpose	Signaling - External Management			Signaling - Internal Management	AAA					Delivery		Encryption /Decryption	

The one by one definition and classification of the modules can guide the Service Composition design process done of composite services and the placement of the different modules in specific physical or logical locations of the Smart Grid. For example, if we consider the description of some security modules such as AAA module, its execution location could be in a specific segment or end-to-end, while encryption or decryption modules are isolated modules that could be placed in a specific location of the network. This fact can help the reasoning of the administrator person or the specification of automation processes for building and deploying composite services automatically. However, some characteristics are intrinsically related to the functionality offered by the composite service to the end-user, such as the atomic service usage (optional or mandatory) or the order of them inside the workflow (dependent or independent) and can only be completely defined when building the composition.

A. Modules definition

The first and most important task that should be accomplished in the definition of the use cases was which functionalities are required to achieve the goal of the use case and which service modules are necessary to cover each of these functionalities. After the in-depth analysis of the security requirements and the technologies available on the Smart Grid (see Table 6.7), it is time to figure out which functionalities are required and which modules could be useful to accomplish the objective of the final workflow. A correct modularization of the process must present services as loose-coupled as possible in order to (1) help to reuse the services in different use cases avoiding their reimplementation, and (2) facilitate the adaptability of the workflow to context changes (e.g., changes on the security level may allow some services to be added or removed at will to improve the process performance). This happens because the Smart Grid needs to manage many security schemes—that need to be coordinated—that in turn have different native key management schemes and policy enforcement methods which apply to different places or hierarchies. In the following, some examples of these most important service modules defined for these use cases are presented.

- USB Keychain authentication: This service encapsulates the functionality of using an USB token dongle for authentication purposes. It contains a unique ID, which converts it to more than a common password since the USB device cannot be easily replicated.

- USB ID Validation: In order to carry out the USB dongle verification and to assess whether it is valid or not, a module has been defined (Figure 6.14). It checks a list (AES encrypted) of revoked IDs to accept or deny the device.

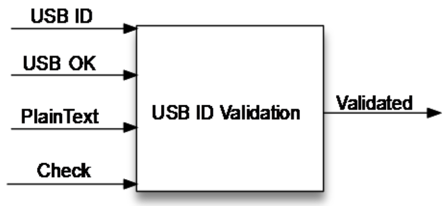


Figure 6.14 USB ID Validation Service module

- AES Decryption: This module (Figure 6.15) will perform the AES decryption (Advanced Encryption Standard) needed for the ID Validation module. Another different service will be in charge of generating and providing the required key.

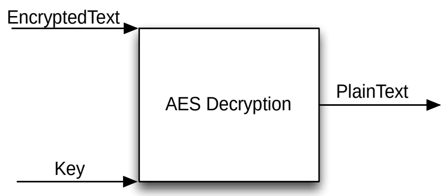


Figure 6.15 AES Decryption Service module

- Key-Distribution: This service provides the key needed to encrypt/decrypt using a symmetrical algorithm. Usually it uses either a pre-shared key or a Public Key Infrastructure system. Although both modules can be used, for the sake of this example the pre-shared key system is selected.

- Certificate download with User+Password: It allows downloading the asymmetrical certificates that will be used for the final enrollment of the smart meter to the Smart Grid system (Figure 6.16). This can be done in several ways but, in this case, it has been selected a module that allows to do so by entering the username and password of the technician. If both are correct and the USB ID has been validated, the certificate will be downloaded.

Figure 6.16 Certificate Download Service module

- Login with certificate: This service (Figure 6.17) is required to use the certificate previously downloaded and checking a Certificate Revocation List. Finally, the smart meter will be enrolled to the Smart Grid if everything is correct.

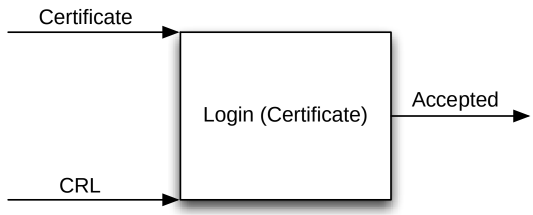


Figure 6.17 Login (with certificate) Service module

B. Interfaces definition

Usually, services modules like the ones presented in this chapter are expressed by means of well-defined interfaces that hide their implementation (in order to maintain them loose-coupled in the service composition process) that could be coded in different logic or development languages. In this case, the option of using Web Services Description Language (WSDL) [Chinnici et al., 2007] was selected to exemplify and deploy this use case. WSDL is the most extended language used by web-services description, and offered a good solution for an easy and fast definition of the services' interfaces, besides bringing a standardized and well-known language easy to integrate with other service frameworks like OSGi [Gu et al., 2004]. Definition of services used in Chapter 5 was adapted for network services. However, it was defined generic to be adapted to application cases as well, as this one. Therefore, this or other specific or more complex definitions can be used in the future, to completely define and deploy the services on demand, placing and invoking or dropping them on one device or another.

- *Types*: In this field the variables are defined using a simple name and type nomenclature. Interface name: Contains the name of the interface.
- *Fault_name*: Name of the attribute generated when an error appears.
- *Operation_name*: Indicates the name of the operation. It must be unique per interface. It also contains the Pattern that usually is "in-out" or "in-only" and defines how the data is exchanged and Style (non-mandatory).
- *In_msgLabel*: Defines the name and format of inputs.
- *Out_msgLabel*: Defines the name and format of outputs.
- *Out_fault*: Associates the output error with the operation.

At the end, the number and quality of the resulting workflows is limited by the compatibility and exchangeability of the individual service modules. Therefore, the adequate planning of the interfaces is highly relevant since it will define how the modules can communicate between each other.

C. Workflow example

Finally, the modules have to be joined to create the whole workflow. As it has been described before, this process allows the smart meter function to be enrolled into the Smart Grid system.

In order to do so, the whole process is being carried out in some steps (Figure 6.18). A request to the USB device is made, then both the signal saying that the USB is still connected and the ID of the USB itself is sent to the next module that checks whether the ID is valid or not, using the information in plain text received from the AES Decryption module. At the same time, this module receives the decryption key from another module. Then, if this step is completed successfully the technician must insert his username and password. Then, the certificate is downloaded and once checked with the Certificate Revocation List, the smart meter can successfully login to the Smart Grid network.

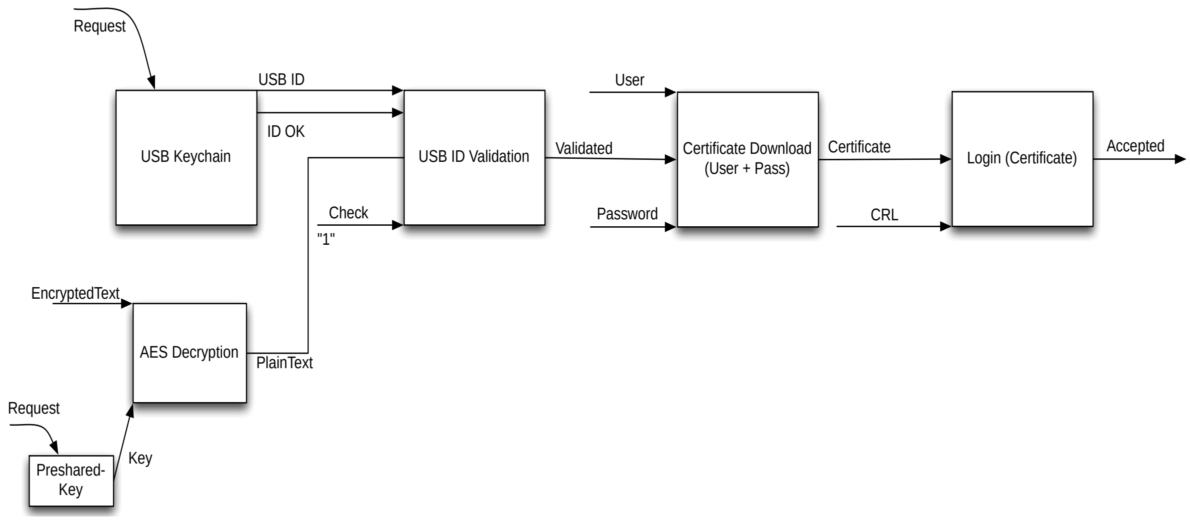


Figure 6.18 Complete workflow approach for "sign up with a non-validated USB" service

However, the importance of this process is not the process itself but its modularized design, development and deployment. That is to say, the main importance is the great flexibility that this way of working enables. It is very different to current-day straightforward deployments by procuring to the system architect a reusable design, a function virtualization and a chance to cloud computing deployment. If this process does not suit the utility's needs it can be easily changed, modules can be quickly swapped for others or even removed to simplify the process. And the greatness comes when this technique is combined with some kind of intelligent middleware that set the policies and build the workflows autonomously (orchestrator) [Goransson et al., 2014]. The orchestrator could even learn and make decisions based on the status of the network [Navarro et al., 2012]. Another interesting characteristic is that it does not depend on how the modules are implemented. As long as the input and output interfaces are well-defined the module interoperability is fixed. So it should also help to avoid any vendor lock-in and to foster the interoperability and reusability of the systems.

6.1.3.4 Conclusions

This proposal on context-aware security for the SDU presented a review of the relevance of the security aspects in a critical infrastructure such a Smart Grid. However, the diversity of contexts and security requirements within the Smart Grid makes it difficult to decide which is the appropriate level of security needed in each case. It usually means an overload of cybersecurity mechanisms or unprotected elements of the electric power network. Therefore, seeking a way to offer adapted security solutions on demand, the way service composition methodology and mechanisms can be used to provide them is analyzed. First of all, a large set of smart grid services are analyzed and its level of security is evaluated, matching them with different protection mechanisms. To exemplify the way to undertake this methodology in a specific Smart Grid use case, this section also presented a Smart Metering context-aware security selection, providing a set of basic services and how they can be combined to offer different levels of cybersecurity. It represents only a first proof-of-concept of this context-aware design for the SDU security, adapted to the stringent cybersecurity requirements of the Smart Grid, but open to newcoming mechanisms, protocols and architectures.

7 Conclusions, research outputs and future work

7.1 Summary and conclusions

This PhD. thesis reports my research done along the last years. It is mainly focused on how to apply service composition concepts to different fields. As stated in the main research hyphotesis, *"Service composition methodology and mechanisms can be applied to media distribution and Smart Grid fields in order to efficiently orchestrate flexible and context-aware communications and processes."*, it started from a media perspective (when I was involved in the Media Technologies Research Group in La Salle) and it embraced later Internet of Things and Smart Grids fields, in alignment to my participation in INTEGRIS and FINESCE projects (once I moved to the Internet Technologies and Storage Group). The research includes an in-depth review of the service-oriented Future Network (FN) architectures, adapted media distribution and Smart Grids communications and security management, and proposes different methodologies and mechanisms to enhance the performance and provide flexible services in both fields. Therefore, this chapter summarizes the main findings of the dissertation, elaborates the obtained conclusions and discusses some future research directions.

In order to provide the context of those different fields of application I was mainly concerned to the reader, first of all, this thesis starts explaining the state of the art and the requirements of the media distribution and smart grids fields (with special emphasis on the security and communication parts of the latter).

In this thesis it was investigated how service composition arguments can be used in order to offer more flexibility in different fields, especially focusing on how to build context-aware communication services, adapted to user or scenario requirements. *Chapter 3: Service Composition* explains the service-oriented paradigm in which those service composition premises are based, and presents, first, the state of the art of service composition techniques. Following the steps of some of the Internet architects, a radical change must be done in the way of managing Internet as a whole, and how we establish communication services in particular. Therefore, it also reflects the state of the art continues analyzing the different approaches of FN architectures that deal with service composition.

After acquiring the knowledge of the proposals considered in this complete review, the dissertation continues in Chapter 4 exposing the design principles that an alternative network architecture focused on services. It summarizes also the FN architecture design in which I was involved during the TARIFA project, aiming to define an architecture that address most of the shortcomings of current Internet architecture. Specially, the proposed architecture addresses the lack of flexibility, ubiquity, and context-awareness in the TCP/IP stack. During TARIFA project, service-oriented paradigm was identified as a potential methodology to solve/mitigate some of the issues and shortcomings of the current Internet architecture providing features like: context-awareness and dynamic adaptability during execution time; flexible allocation and execution of network functions; inherent QoS, security and service discovery; routing based on the semantic description of services (non-mandatory use of addresses, support for mobility, nomadism and multi-homing); QoS and resource availability integrated into routing and service discovery; enhanced user control; etc.

Besides the grounding architecture, this thesis also presents a service composition framework (methodology and mechanisms) to work over this service-oriented network architecture with that focus on services combination and adaptation to context conditions by means of service discovery and service composition. It defines the different type of services, Atomic (AS) and Composed (CS), and how them are described, discovered and composed on demand when a communication service is demanded. It also defines how the composition modules interact with other elements already defined in the architecture. Also, a general weighted scoring function for each AS is proposed, and the methodology of how to apply the search algorithm in order to score the ASs of a node.

It was identified that the proposed approach and the defined processes enable FN service provisioning in an adapted manner, satisfying the specific QoS/QoE requirements demanded by users. Unlike other solutions reviewed in Chapter 3, the service composition process, presented approach defines a preliminary discovery process that finds those services to be composed all along the end-to-end path considering the requirements specified by the requester and assuring a certain level of QoS.

To achieve this, the proposed approach also considered that routing functions are integrated into a service discovery and negotiation protocol and it evaluates context conditions hop-by-hop when a communication is requested. This protocol enables to find and to compose services that meet requesters' requirements efficiently. The negotiation time of this protocol was evaluated in comparison with another template-based approach in section 5.6.2. In the results shown it can be seen that template-based offered only slightly shorter composition times, and putting at risk the flexibility of the composed services.

Main details of first implementations of the proposed solution and discussion the preliminary gathered results are also provided in sections 5.6.1 and 5.6.2. Two initial prototypes were built, one based in Java and another in C integrated with the development of other modules within the TARIFA project framework.

On the one hand, Java-based implementation presented a first approach of the generic composition algorithm that allowed to test different search algorithms and visually identify algorithm behavior problems. Thanks to empiric analysis done using this first implementation, A* algorithm was validated and selected as basis for the composition algorithm.

The usage of A* graph search algorithm was then considered as a potential algorithm for building the composition of services, because it offered a better performance than Edsger Dijkstra's algorithm (with respect to time) and a solution that enables an easy adaption to the context and necessities of service composition over a FN architecture. It allowed to easily integrate the constraints of the different nodes and preferences of the users by means of heuristics. These filtering rules have been added to the algorithm for speeding the algorithm up as it discards unsatisfactory nodes earlier while guaranteeing that a certain QoS level is provided for each path of network nodes that is analyzed. Therefore, A* reduced the number of composition possibilities and limits the time of the composition process with regards to Dijkstra, depending on the requirements provided by the user and other service constraints defined (e.g. service establishment latency).

Note that the composition process defined can be seen as complex in computational terms, as it lays on the complexity of the proposed graph search algorithm, A* (eq. 5.6). However, the complexity of the algorithm for the service composition case was observed as polynomial, what gives certain confidence for scaling up the algorithm increasing the set of ASs available and the number of nodes of the network. In any case, during the negotiation, the CPU consumption in the network nodes comparing the service composition and establishing standard TCP/IP services were the same order of magnitude, as stated in the results shown in section 5.6.2.

The composition time using this solution was evaluated in section 5.6.1 and 5.6.2, obtaining that the time to build adapted communications could be excessive if the composition is completely applied at execution time for some delay-critical services. However, it offers other additional capabilities such as flexibility, context-awareness, adaptability and personalization. Besides this, in order to avoid the calculation of the combinations and scores every time a service is requested, past compositions can be stored and reused. Thus, in static environments only new or rare services will be created and template-based approaches could be applied. On the contrary, very dynamic environments will make this method computationally heavy and not suitable for nodes with low processing capacities.

On the other hand, the other prototype allowed to show a real networking example running over the FN architecture proposed in Chapter 4, developing the architecture basics and the composition framework in C code. It showed a more realistic end-to-end negotiation and communication establishment time, which was in all tests under 140 ms. As stated before, it can be considered an acceptable service establishment time for non-critical to latency services, and it could be reduced by means of hybrid template-based solutions on cases with more stringent latency requirements (although providing a more coarse-grained flexibility). It is important to notice that the use case considered and evaluated with the C-prototype was only set with a very limited set of services and this time is expected to increase with more complex networks and a large number of services available. Therefore, the time and efficiency of the composition can be longer than exposed. However, it has also to be considered that it was tested using commodity hardware and a first version implementation, and it is also expected to acknowledge the complexity increase using hardware with higher specifications and improving the prototype implementation.

Nevertheless, the adoption of the proposed the service-oriented approach enable to introduce the proposed architecture, framework and mechanisms gradually over current infrastructures (like SaaS or IaaS approaches) starting on local area and dedicated networks while trying to extend the paradigm to lower layers in the future to achieve a real disruptive approach at a wider level. Composition of basic network-level services calls for a clean-slate approach to the Internet, while composition of higher level (transport and application) services prompts for an evolutionary approach. Nevertheless, composition of communication services manifests itself as a revolutionary way of building communication systems.

It is also relevant to point out that this architecture would be the first clean-slate deployment completely aligned with the current work being done by the ISO/IEC JTC1/SC6/WG7 Future Networks group which is in development of the ISO/IEC vision of how a clean-slate FN architecture should be. Actually, some of the work of this thesis is exposed in the Technical Reports TR 29181-1 [ISO/IEC TR 29181-1] and TR 29181-7 [ISO/IEC TR 29181-7] accepted as International Standards in 2012 and 2013, respectively.

All in all, some critical issues to consider are network architecture implantation and its impact on the sector's business models. On the one hand, a migration plan is not yet defined by research community (neither for this, nor for most of FN) and I am aware that it is a really hard task to move it forward at Internet-scale. Considering the competition of other proposals that have heavier institutional and commercial support, a part of the current model that actually works (although with those deficiencies analyzed during this thesis dissertation) it looks really difficult that we can see a FN architecture working at high-scale in the next 5-10 years. Notwithstanding, the proposed architecture building blocks and framework can be borrowed by other FN proposals in order to gradually develop better designs for new network architectures that can definitively retire the way that networks work nowadays. Hence, most of the part of this dissertation should be analyzed in Internet science terms and the need of research and analysis on how the networks work and how could they work in the future without taking into account the difficulties that the Internet inertia can cause in a massive communications change.

On the other hand, the proposed architecture is designed to support with ease the traditional business models implanted in today's Internet, as functions may be allocated only at the end-nodes and the network can still remain as a "transparent pipe" where injected bits at source arrive unmodified to destination. But, also the business model may change drastically as services are placed in-network, diverting from the end-to-end principles, making the network a service provider and a third player in service provisioning. Given the "operator-friendly" nature of the proposal, as it diverts from the end-to-end vision, it may provide operators with new business opportunities as they provide enhanced, differentiated and personalized services; but new modes of coordination and cooperation amongst operators will be required, since services may be placed between different domains, the billing, monitoring and assuring QoS parameters will become a more complex matter, requiring more coordination in peering relations. In this context, it is fundamental to create a suitable cost model for dealing with services and to create a powerful environment to develop new business models. A novel, simple and easy-to-apply framework for costing over service-oriented architectures is another contribution of this thesis. Thanks to the proposed costing framework, services would be able to be combined according to the requester needs while allowing the providers to specify how services are charged according to different pricing models. It is important to remark that the costing framework should maintain its simplicity in order to be feasible and implementable. Thus, being able to run in real-time nodes.

Moreover, other proof-of-concept developments were undertaken during the research presented in this dissertation, testing the framework performance, not only at network level, but also how it can be adapted to offer on demand context-aware services in other fields, such as the two fields of application that were studied during the elaboration of this thesis.

First, following the work done in TARIFA and the special interest of the Media Technologies Research Group in which I was involved at the first stage of the thesis, an example was presented in chapter 5 (section 5.6.3), showing the usage of the service composition framework to offer adapted media services. This research wanted to take special relevance for FN architectures, in which services must be allocated all along the route, executing just the desired service at each hop, section of hops or end-to-end. Hence, in that type of FN architectures, media adaptation services can be placed as in-network services blindly to the end users, and that is the final goal of the proposed architecture and service composition framework. However, in order to test the framework in a real current scenario, an adaptation of the framework has been proposed for an adaptive video streaming service. Concretely, it defines a set of multimedia codecs as Atomic Mechanisms (AMs) and selects the optimal multimedia codec depending on the context of the end users. It particularized a general expression for scoring services in the case of selecting a multimedia codec taking into consideration quality assessment metrics. However, it is important to notice that other parameters can be added to the scoring function. Additionally, each AS can propose a specific scoring function in order to select the best AM that is able to provide it. The scoring of an AMs can be done by each node in the network or, if the profile information for all the nodes is available in the network, it can be done by specific external nodes that can carry out this task. Current services using these techniques will be able to adapt content taking into account the context of users intrinsically. In summary, the study presented introduces a way of enabling context-aware media services in the context of FNs, evidencing that new functionalities can be added in an easy and flexible way, allowing the proliferation of new applications while adapting architectures to past, present and newcoming requirements.

After that, and consequently with my rising concerns on other fields such as IoT, the participation on Smart Grid related projects (INTEGRIS and FINESCE) and the shift to Research Group on Internet Technologies and Storage, Chapter 5.7 presents different studies and developments analyzing how to build services efficiently adapted to the requirements of the Smart Grid. Actually, chapter 6 concentrated on the solutions developed in the context of those European research projects that focus on the protection of data while being transmitted, stored and used in the context of the distribution Smart Grid, with the objective of proposing a Software Defined Utility (SDU) that meets the data cyber security requirements of Smart Grid. In the project several issues were tackled such as access control, key management and context-aware security design in the case of the electrical distribution Smart Grid in the cloud. More concretely, three different research lines were studied and subsequent research outputs were given towards the development of the SDU concept, which advocates the migration of the utility infrastructure to software systems instead of relying on complex and rigid hardware based systems.

First, an in-depth analysis of a flexible hybrid-cloud storage systems for the Smart Grid (section 6.1.1), studying the security concerns of using external storage systems and agreeing with Telco operators and DSOs about the importance of cybersecurity in order to cover the stringent privacy requirements of Smart Grids. It was concluded that securing Data Transaction and Authentication are those more critical issues and should be the first to actions be protected. Reliability of the system was also a key point and some protocols to rely on were presented in order to offer redundancy and high availability in a Smart Grid environment. A table analyzing them and suggesting the best solution was given (Table 6.5), depending on different Smart Grid context (e.g. number of devices, topology, recovery time, etc.). It covers also an analysis of where is the best place to allocate resources or information (summarized in Table 6.1). Besides it, it presented the different deployment and management tools that were developed in the context of FINESCE project and enable the fast deployment of distributed storage services for a SDU data storage infrastructure.

Secondly, a unified and web-based management system for the SDU is also proposed. The Web of Energy (WoE) is proposed as an IoT-based infrastructure that enables machine-to-machine interactions between small and resource-constrained devices on the Smart Grid domain based on HTTP protocol. A framework approach, communications architecture and proof-of-concept is presented in section 0. It has been shown that the proposed architecture is feasible and that the implementation of successive parts can be made using this design. Different web development options have been analyzed (section 6.1.2.5) and the proof-of-concept tested assessing the feasibility of the architecture in order to manage the unified information scheme for smart objects.

Finally (in section 6.1.3), the proposed service composition methodology and framework was adapted and tested for providing a flexible and context-aware solution in the selection of the security mechanisms within the Smart Grid. In order to show the potential of those techniques in this other field, it was exemplified in the specific use case of the smart metering. It exalts how the characteristics of Service Composition suited perfectly with the needs of the Smart Grid field and how their applicability could represent a cheaper and more dynamic deployment for the Smart Grids in the following year, anticipating the SDU of the future.

The Smart Grid is at the same time a part of the Internet of Things and an example of a cyber-physical system where the physical power grid is surrounded by many intelligent and communication devices that enable an enhanced management of it. Despite its benefits, this system of systems may bring critical risks in terms of cyber security since it opens the power system to at least the same threats faced by the Internet. In fact, considering the novel, heterogeneous and distributed nature of the Smart Grid, it is reasonable to think that the vulnerabilities will be even larger. Furthermore, cyber security in Smart Grids is essential for the survival and feasibility of this electricity concept. Hence, the Smart Grid concept changes radically the way traditional energy grid works as it becomes dynamic, versatile and autonomous.

SDNs and especially Service Composition techniques can help to fulfill these requirements. They do not only apply for the management of the network but also for its security, one of the most important parts of the grid since it affects all of its stakeholders and can cause even international security problems, as some recent cyberwar attacks demonstrated. To sum up, these new paradigms on network computing architectures present a modular solution based to address the required functionalities and their deployment on demand –only when and where required– avoiding redundancies and fostering reusability for the development and deployment of the SDU.

At the end, it is also important to mention the evolution of myself within my current research group from my involvement. Starting from a follower in the group that brought a background of media, I started implementing and contributing to technical tasks, and I gradually moved to analytic and design tasks, becoming the scientific supervisor in the last projects we participated. I also take over the business development responsabilities of the group, bring me in not only in the design of the solutions that we were committed in the projects, but also envisaging the new proposals and fixing new research lines of the group in which other teammates are now working on.

In order to summarize the main research contributions of the thesis, section 7.2 enumerates and details briefly the research outputs obtained.

7.2 Research Outputs

- **Output 1: State of the art**. State of the art review of Service Composition methodologies, service-oriented approaches proposed in FN projects, and the context and requirements of the two fields of application that are covered in this thesis, presented in chapters 2 and 3. It was developed at the beginning of the thesis, with the aim of analyzing the different service composition possibilities. It review the Service Composition paradigm from a generic point of view, detailing the experience acquired from Web-service composition approaches, and also summarizes the in-depth literature research done of the alternative solutions presented by FN projects. It was updated along the years of the elaboration of the thesis with the proposals of new projects undertaken. The fields of application, Smart Grids and Media Distribution are also briefly covered, synthetizing the requirements, context and challenges of this fields and how they can benefit from service composition.
- **Output 2: Proposal and implementation.** Service-oriented FN architecture, presented in chapter 4 briefly summarizes the work done in TARIFA in which I contributed especially in the definition of the service-orientation vision of the architecture, and how the TARIFA nodes should work with the services (how they are demanded, discovered, invoked and managed). Detailed node architecture is given in Figure 4.2 and the complete life cycle of the service establishment in TARIFA architecture is shown in Figure 5.8.
- **Output 3: Design, definition and implementation.** Service composition methodology and framework are detailed in chapter 5, complementing the core definition of the TARIFA FN architecture. It provides the whole process definition, including design proposals for the different type of services, and mechanisms, their interfaces and the different modules that interact with them in order to create on demand context-aware communications in a service-oriented FN environment.

- **Output 4: Analysis and proof-of-concept implementation.** Service composition application to context-aware media distribution exposed in section 5.6.3 is shown as a proof-of-concept of the application of the proposed service composition framework in other fields beyond a generic context-aware communication use case. It extends this use case, generalizing the framework for its usage in the selection of the most appropriate video or audio formats in a streaming service. The context-aware streaming problem is modularized by means of defining the different formats as different on-demand services and selecting the solution following the service composition methodology presented in the first part of chapter 5.

- **Output 5: Analysis, conceptualization, design and research lead.** Service composition for Smart Grids and derived work towards the definition and deployment of the SDU, detailed in chapter 6. A set of modules is presented as different components of the SDU that have been elaborated during the duration of the thesis. The specific challenges and threads of the Smart Grid deployment are covered, and the Hybrid Cloud Data Management (section 6.1.1), Web of Energy (section 0), and context-aware security (section 6.1.3) components are detailed. They provide different functionalities in order to cover the full data chain of the Smart Grid, from the secure extraction of data from the consumer and the flexible management of Smart Grid information by DSO managers, to the homogenized management of the whole SDU.

7.2.1 List of publications

7.2.1.1 International Journals:

- Gonzalez, A.J., Martin de Pozuelo, R., German, M., Alcober, J. and Pinyol, F., 2013. New framework and mechanisms of context-aware service composition in the future internet. ETRI journal, 35(1), pp.7-17. ***IF: 0.945, Q3*. *[Objective 1].***

- Briones, A., Martin de Pozuelo, R., Navarro, J. and Zaballos, A., 2016. Resource Allocation on a Hybrid Cloud for Smart Grids. Network Protocols and Algorithms, 8(1), pp.7-25. ***[Objective 3].***

- Sánchez, J., Corral, G., Martin de Pozuelo, R. and Zaballos, A., 2016. Security issues and threats that may affect the hybrid cloud of FINESCE. Network Protocols and Algorithms, 8(1), pp.26-57. ***[Objective 3].***

- Vernet, D., Zaballos, A., Martin de Pozuelo, R. and Caballero, V., 2015. High Performance Web of Things Architecture for the Smart Grid Domain. International Journal of Distributed Sensor Networks, vol. 2015, Article ID 347413, 13 pages. doi:10.1155/2015/347413. ***IF: 0.906, Q3*. *[Objective 3].***

- Selga, J.M., Corral, G., Zaballos, A. and Martin de Pozuelo, R., 2014. Smart grid ICT research lines out of the European project INTEGRIS. Network Protocols and Algorithms, 6(2), pp.93-122. ***[Objective 3].***

7.2.1.2 International conferences:

- Gonzalez, A.J., Martin de Pozuelo, R., German, M., Alcober, J., Pinyol, F. and Ghafoor, K.Z., 2012, September. Enabling SCI-FI: service-oriented context-aware and intelligent future internet. In Proceedings of the 7th International Conference on Future Internet Technologies (pp. 52-57). ACM. ***[Objective 1]***.

- Gonzalez, A.J., Martin de Pozuelo, R., Alcober, J., Pinyol, F., Gutierrez, A. and Monguet, J.M., 2011, June. Costing framework for service-oriented future internet architectures: empowering requester's choice. In Proceedings of the 6th International Conference on Future Internet Technologies (pp. 84-90). ACM. ***[Objective 1].***

- Gonzalez, A.J., Martin de Pozuelo, R., Pinyol, F. and Alcober, J., 2011. In-Network Service Selection and Composition Based on User and Network Context. In Euro-NF International Workshop on Traffic and Congestion Control for the Future Internet (pp. 3-4). ***[Objective 1].***

- Sanchez-Loro, X., Gonzalez, A.J., Martin de Pozuelo, R., 2010, June. A semantic context-aware network architecture. In 2010 Future Network & Mobile Summit (pp. 1-9). IEEE. ***[Objective 1].***

- Gonzalez, A.J., Martin de Pozuelo, R., Alcober, J., Pinyol, F. and Ghafoor, K.Z., 2011, July. Context-aware multimedia service composition using quality assessment. In 2011 IEEE International Conference on Multimedia and Expo (pp. 1-6). IEEE. ***(ISI – CORE B conference)***. ***[Objective 2].***

- Martin de Pozuelo, R., Ponce de Leon, M., Howard, J., Briones, A., Horgan, J. and Sanchez, J. 2016. Software Defined Utility: a step towards a flexible, reliable and low-cost Smart Grid. In ICSGS'16, Proceedings of the 5th International Conference on Smart Grid Systems. ***[Objective 3].***

7.2.1.3 National conferences:

- Mallorquí, A., Zaballos, A., Navarro, J., Martin de Pozuelo, R. and Briones, A.2015. Modelo para la ubicación de recursos en Cloud Híbrido. ***[Objective 3].***

7.2.2 Standardization activities

- Member of the ISO/IEC JTC 1/SC 6 Telecommunications and information exchange between systems.

- Editor of the International Standard ISO/IEC TR 29181-7. Information technology -- Future Network -- Problem statement and requirements -- Part 7: Service composition. (Publication date: 2013-04-09).

- Contributor to International Standards ISO/IEC TR 29181-1, ISO/IEC TR 29181-3, ISO/IEC TR 29181-6, and draft ISO/IEC TR 29181-8.

- Member of the AENOR group AEN/CTN 071/SC 06 "TELECOMUNICACIONES E INTERCAMBIO DE INFORMACION ENTRE SISTEMAS".

- La Salle – Universitat Ramon Llull representative in City Protocol Society. Contributor and Sate of the Art editor in the City Protocol Agreement “Open Sensor Platform”.

7.2.3 Projects

7.2.3.1 TARIFA

TARIFA [TARIFA], from 2010-02-01 to 2012-02-31. Total cost: 354,088 €. This project is a clean slate approach to a Future Internet architectural redesign based on a role-based paradigm consisting of non-divisible, or atomic, functions. The project was developed having in mind two main use cases: an evolutionary use case which will demonstrate that the new design of the Future Internet architecture is backwards compatible with the current Internet by building with our approach a service or functionality similar to existing solutions; and a revolutionary use case which will demonstrate that the project is actually solving those problems that the current Internet cannot in an efficient manner.

Funded and coordinated by i2cat Foundation, it was developed by 7 different research groups:

- The Wireless Network Group (WNG) at Universitat Politècnica de Catalunya (UPC), BarcelonaTech, and led by Prof. Josep Paradells.
- The Media Team (MEDIAENTEL) of the Design and Evaluation of Broadband Networks and Service (BAMPLA) research group at UPC and led by Dr. Jesus Alcober.
- The Optical Networks Group (GXO) of BAMPLA at UPC, led by Dr. Cristina Cervelló.
- The Collaborative Design Research Group (COLS) at UPC, led by Dr. Josep M. Monguet.
- The Advanced Network Architectures (ANA) research group at UPC, led by Dr. Xavier Masip.
- The Multimedia Technologies Group (GTM) group at La Salle - Universitat Ramon Llull (URL), led by Dr. Francesc Pinyol.
- The Network Technologies and Strategies (NeTS) group at Universitat Pompeu Fabra (UPF), led by Dr. Boris Bellalta.

The technical core developments of the project was used to validate the improvements of this new approach. TARIFA contributed with new concepts for future derived projects with potential industrial and/or societal benefits basically in the following areas:

- Context-aware service composition: to study and propose solutions enabling the provisioning of Future Internet services through context-aware service composition. This allows providing adapted and personalized services, dealing with high dynamic and heterogeneous environments, in particular on Future Media Internet and Internet of Things environments.
- Autonomic network management: to define and design a reliable, scalable and robust network infrastructure based on automatic management principles that allow reacting properly to the network conditions in different scenarios and guaranteeing the end-to-end QoS goals of data flows.

- New routing & addressing paradigms for the Future Internet: to give solutions achieving better routing scalability taking into account the separation of IP addresses into endpoint Locators and routing Locators and fulfilling the challenges that this solution presents.
- Socio-economic impact: to design novel business models that can be exploited in the context of the Internet of the Future.

The project considered solutions thought to improve three important lacks of the current Internet: energy efficiency, heterogeneity of information and coexistence of networks. As a final goal, the project evaluated and validated the results in an integrated environment.

More concretely, my work in the project was focused in the definition, development and trial evaluation and validation of the context-aware service composition framework of the architecture, leading WP3 Service Composition together with Alberto J. González of MediaEntel group. However, I was involved and contributed in the overall architecture design as well as in the autonomic network management modules and the socio-economic framework definition.

7.2.3.2 INTEGRIS

INTEGRIS. European Union's 7th Framework Program (FP7) Project "INTelligent Electrical Grid Sensor communications". ICT-Energy-2009 call (number 247938), from 2010-02-01 to 2012-12-31. Total Cost: 5,789,200€.

INTEGRIS project proposes the development of a novel and flexible ICT infrastructure based on a hybrid Power Line Communication-wireless integrated communications system able to completely and efficiently fulfill the communications requirements foreseen for the Smart Electricity Networks of the future. This includes encompassing applications such as monitoring, operation, customer integration, demand side management, voltage control, quality of service control, control of Distributed Energy Resources and asset management and can enable a variety of improved power system operations, some of which are to be implemented in field trials that must proof the validity of the developed ICT infrastructure. Focus is on interoperability of the PLC, Wireless Sensor Network and Radio Frequency Identification, technologies that together are able to achieve the indicated goal with reasonable cost. The system will require an adequate management system that is also an objective of the project. Such system will be based on beyond the state-of-the-art cognitive techniques to provide the system with the adequate flexibility, scalability, availability, security, enhanced system lifetime and self-healing properties as is necessary in complex and dynamic systems. A further objective is to research on the limits and benefits of distributing smart grid applications in the newly designed INTEGRIS system. This will have an impact on the availability of those applications and influence the developed devices and platforms since they will require a certain level of storage and computing capabilities. The final aim of the INTEGRIS project is to provide an ICT system that enables the improvement of the performance of the electricity distribution grid in agreement with the impact foreseen in the work program. The INTEGRIS project is a cross-thematic research approach integrating knowledge and partners from ICT and Energy fields and aims to create and consolidate such a cross-thematic team.

The project was coordinated by Enel Energy Europe, and the participants included: Current Technologies International, Marvel Hispania, Indra Sistemas, La Salle – Universitat Ramon Llull, Technical University of Tampere, Schneider Electric, TemSec Service, A2A Reti Elettriche.

With INTEGRIS project, I started the knowledge acquisition about Smart Grid complexity, requirements and challenges. My work was especially focused in the definition of the IDEVs platforms, its different modules integration (Distributed Storage service, security, QoS, etc.) and their adaptation to the services context (e.g. context-aware management broker).

7.2.3.3 FINESCE

FINESCE [FINESCE], European Union's 7th Framework Program (Future Internet PPP) Project "Future INtErnet for Smart Utility ServiCEs". FI.ICT-2011 Grant 604677. From 2013-03-01 to 2015-09-30. Total cost: 19,259,850€. FINESCE is the smart energy use case project of the 2nd phase of Future Internet Public Private Partnership Programme (FI-PPP) funded by the European Union within FP7.

From 2013 until 2015, FINESCE contributed to the development of an open IT-infrastructure to be used to develop and offer new app-based solutions in all fields of the Future Internet related to the energy sector. The project will organize and run a series of field trials at trial sites in 7 European countries.

FINESCE builds on and extends the results of the FI-PPP FINSENY project to realise sustainable real time smart energy services. The consortium includes leading energy and ICT operators, manufacturers and service providers as well as research organisations and SMEs from all over Europe. Together, they contribute directly to strongly focused trials and business innovation. The project features a scale and scope ensuring the FINESCE results will drive the FI-WARE and Future Internet success and long-term exploitation internationally.

It was coordinated by Ericsson, and the consortium was built by 29 energy and ICT companies, R&D centers and universities from 13 European countries representing the big Smart Grid players as well as SMEs and young know-how.

In FINESCE, I lead the technical and scientific work of La Salle, in which we presented an approach designed to work in WP5 Stream II (Future Internet in Smart Grid Communications) in Ireland. It aimed at providing a solution to run more flexible communications and network management in the Smart Distribution Grids, going in the direction of a "Software Defined Utility. Our main role was the integration of FIDEVs in the Irish trial. A FIDEV was defined as an upgrade of IDEVs devices, integrating a set of Generic Enablers (GEs). IDEV devices were initially developed in FP7 INTEGRIS project, as a set of subsystems (distributed storage system, TRILL protocol, NMS, QoS) for helping in the management of the Smart Distribution Grid. The inclusion of GEs provided a secure interface to the distributed storage system, which can be used for network monitoring, and provides functionalities for a more flexible management (in this case, for a network manager from a DSO). Among these new functionalities, integrated GEs provide a homogeneous Cloud Data Management Interface (CDMI) that enables the seamless interaction between the distributed storage system in the FIDEVs and a Public Cloud deployed on FIWARE Lab through the Object Storage GE. In the trial, a set of separated FIDEV devices placed in Ireland trial site and La Salle lab in Barcelona were deployed, constituting a private cloud, plus the deployment and interconnection with a public cloud infrastructure based on FIWARE Lab. Data can easily be migrated between the public and private clouds with a manager front-end application according to the decision of their owners (e.g. utility companies).

The work done together between ESB and La Salle and the results obtained from the trial have been used by ESB to evaluate a novel "Software Defined Utility" approach, which consists on high-speed physical communications and flexible software infrastructure over them. FIDEVs would be only elements of this wider approach, focusing the trial on the demonstration of a secure and distributed storage system that can easily migrate data from private infrastructure of the utility/DSO, to public cloud, in order to easily sell or offer this data to external stakeholders. This also provided a platform to manage distributed data among different substations, automatically replicating it in the different locations, which can help to evaluate the substitution of some very expensive electrical network devices by software platforms such as FIDEVs, low-cost sensors and high-speed communications underneath.

7.2.3.4 ENVISERA

ENVISERA, "Optimized HF transmission for Near Vertical Incidence Skywave (NVIS) links for remote sensors in Antarctica", is a project funded by the "Dirección General de Investigación Científica y Técnica Subdirección General de Proyectos de Investigación" of the Spanish National Government. It started in March 2016 and it is in development until 2019, with a total amount of funding of 187,000€. It is completely developed by La Salle – Ramon Llull University researchers. The project continues the activities done in previous projects by La Salle researchers in Antarctica. Up to now, most of the sensors installed by the Spanish scientific community in Antarctica have been placed close to the stations, because that makes data collection and data transmission easier. This project is a step forward for the communication capabilities of Spanish scientists in Antarctica. The main goal is the design of a sensors network able to transmit their data from very remote places (even in the continent) to the Antarctic Station Juan Carlos I. This network will use the Near Vertical Incidence Skywave (NVIS) that allows the communication between nodes up to 200 Km away using vertical reflection in the ionosphere in the HF band (3-30 MHz). As both the transmitted and received signal come from the upper part of the atmosphere, no line-of-sight is needed and you can have any obstacle without any loss of the signal. Then, every single node can behave as a repeater if necessary, so large areas can be covered, provided that the distance between two contiguous nodes is less than 200 Km. The transmitted power for each node is low, and that improves the autonomy of the system. The sensor network will also allow bit-rates up to tens of kbps, good enough for most of the sensors for the transmission of medium/high quality digital voice. The proposal also includes the installation of a complete prototype of sensors network with repeaters, for both data and digital voice links.

For the design of the sensors network, first the NVIS channel has to be sounded via pattern sequences. These soundings will be compared with the data obtained by the ionosonde installed at the Antarctic Station Juan Carlos I. The main parameters of the channel will be studied and a new model for the NVIS channel will be proposed.

Then, we will design the best transmission system in terms of bandwidth and modulation that fits the NVIS channel. Once the physical layer is defined, a new link and network protocol will be designed in order to have a mesh network where the nodes can behave as repeaters when necessary and guarantee the transmission through long distances.

In this project, my task is to define, deploy and evaluate the sensor network architecture, by considering the application of communication paradigms and network architectures such as service-oriented communication, information-centric networking (ICN) or Delay-tolerant Networks, as promising approaches for tackling the challenges in communication, energy efficiency and security for IoT.

7.2.3.5 SHECloud

SHECloud, "Smart Hybrid Enterprise Cloud", is a project selected in 2013 call of "Acción Estratégica Economía y Sociedad Digital" (AEESD), from the Spanish Ministery of Industry, Energy and Tourism, which started at the beginning of 2014 and last until March 2015. In the project, La Salle participated in conjunction with the enterprises Abiquo, Claranet y MediaCloud. After analyzing the state of the art hybrid cloud solutions (a combination public and private cloud strategies to offer the best features of both types), SHECloud pretended to create an intelligent orchestrator for the interoperability of public and private clouds, automatizing the migration of workloads (virtual machines and other resources) among clouds.

In this sense, SHECloud offered a faster and more efficient deployment of hybrid clouds, providing SMEs a tool that will give them more flexibility in their business scalability, without needing large investments on infrastructure and optimizing the cost-efficiency of their cloud operations.

My task in the project was the coordination La Salle research, in which we were in charge of the cloud metrics analysis and the decision process of the orchestrator. By means of scalability and criticalness metrics previously defined by the user, and a characterization process of the different types of clouds, this system selects in which cloud is more efficient to place one resource or another.

7.2.3.6 VSNoIPv6, MBTAP and OMBTAP:

This trilogy of projects in which La Salle has been subcontracted by "VSN (Video Stream Networks) – Innovation and Media Solutions", in order to build an efficient transport layer protocol for the distribution of high volumes of information. The first project started at the beginning of 2013 and its duration was for one year, completely funded by VSN, while MBTAP (Media Bus Transfer Protocol) and OMBTAP (Optimization of MBTAP), were partially funded by the Spanish Ministery of Industry, Energy and Tourism in the in 2014 and 2015 calls of AEESD respectively. It presented the design and implementation guidelines of a reliable and secure transport protocol, optimized for large data transfers; especially over networks that present high bandwidth and delay product. MBTAP project was developed along the whole year 2015 and finished in February 2016, and its output was the basic definition of the protocol, detailing the required headers, communication dialogues, machine-state diagram and congestion control of the protocol for the different phases of connection establishment, channel bandwidth estimation, data exchange and connection shutdown. OMBTAP is still ongoing until from March 2016 until February 2017 and its objective is the optimization of the protocol for working on heterogeneous wired-wireless networks, by means of differentiating segments lost caused by congestion and those losses due to channel fluctuation or unavailability.

In this set of projects, I was involved as a scientific coordinator, contributing in the definition of the whole protocol specification and guiding the implementation, validation and evaluation tasks.

7.2.4 Other remarks

- Awarded with a 4-year EPIF grant (aid for training and development programs of research personnel) by La Salle – Ramon Llull University.
- 45-days stay at the center *ICSY - Integrated Communication Systems*[40] of the *Technische Universität Kaiserslautern* (TUKL), led by Professor Paul Müller, from August to October 2013. The stay has been focused on the exchange of ideas about network architectures and services based on the formal definition of services and features minimal network. Points in common were identified on service-oriented network research, focusing on a correct definition of service and simplifying the exchange between services in order to establish communications in a dynamic way. To delve deeper into these topics, a collaboration in both directions. On one side, regular meetings were held during the stay with Dr. Paul Müller, Dr. Bernd Reuther and other partners to discuss a consensual solution that meets the definition of a network service. Meanwhile, work on the code was made during the project TUKL during SONATE project in order to incorporate new algorithms composition of services and compare their behavior with respect to those already implemented.

7.3 Future work

The work done proposed a service composition framework and mechanisms to provide composite in-network services and context-aware communications and demonstrated their applicability with benefits in the definition of adapted services to different fields such as media distribution and smart grids management. It defined an easy to use and flexible framework and methodology for dynamically deploy in-network functionalities for adapting media streaming services. The work also derived in different tools for a more flexible and context-aware management of the Smart Grid, that enable an easier management (single data management and visualization tool, hybrid cloud data storage infrastructure, dynamic and context-aware security access mechanisms) of the huge amount of data generated in the Smart Grid infrastructure. However, it is a large research ambit to cover and many research tasks are still remaining with regards to many different aspects of the proposals made in.

Regarding the definition of a FN architecture that supports service composition, some future work includes addressing different issues. It is necessary to define mechanisms to address

[40] http://www.icsy.de/

communications and service discovery across domains in a scalable way. The proposal defines how the services can be discovered in a single domain by means of a controlled flooding option. However, the combination with scalable name resolution services (such as MDHT [D'Ambrosio et al., 2011]) should be defined if the proposal is intended to be deployed at a wider scale. Atomic services optimal allocation strategies should also be explored in order to work over networks with high heterogeneity of nodes, which complicate the allocation decisions. Further exploration on syntax, rules and semantics of the different vocabularies for QoS, resource and service description is also needed. The proposal presented in this thesis is just a first step in this sense. It take as example FN projects and service composition methodologies reviewed in chapter 3, but it lacks the usage of a standardized ontology and vocabulary. Standardization on Network Functions Virtualization [NFV, 2015] will be also monitored in order to see the evolution of this trend and the synergies with the work done in this thesis about networks services definition and composition.

About the composition process, all composition steps and operations are performed following a centralized approach, carried out by RN. Future extensions of the framework should allow composing services in a distributed manner in order to reduce the number of calculations made by the requester. It should simplify the overall service discovery and composition process, reducing considerably the communication establishment time, and allowing also other latency-critical end services to be deployed this way. This would imply giving more intelligence to the network, demanding for a fog-computing alike methodology in which the nodes self-autonomously demand to their neighbors about their characteristics. It could allow to achieve a faster optimal composition, not only within a node, but an overall optimal one. However, the empowerment of the end user in the construction of the demanded service is another of the objectives, and benefits of the presented proposal. To apply this "choreography" instead a requester-centralized orchestration should not prevent that the requester/consumer, as the initiator of the service, takes the final decision in the composition.

In addition, a first exploration was done taking A* algorithm as a basis for the graph search in the selection of services and the end-to-end communication path. An adaptation of A* was validated as an option for calculating the best composite service, scoring and selecting the AMs and ASs, as well as selecting the less cost networking. It was empirically compared with a set of other search algorithms (Dijkstra, Depth first, Breadth first, Hill climbing, etc.) and adapted to the usage over the FN architecture and the service composition environment proposed. However, further analysis on search algorithms alternatives or other techniques (such as AI Planning methods) are considered as future optimization of the proposed generic composition algorithm able to run in smarter devices. In future stages, exhaustive benchmarks and comparisons of different composition algorithms and techniques are expected, thus determining which ones are the best under specific conditions and environments.

Moreover, the composition algorithm will be enhanced at run time thanks to the storage of predefined compositions in the form of templates saving common used workflows, allowing their reuse based on their past use.

The main concern to be tackled by all emerging FN architectures is the validation of the proposed solutions. It is also the main challenge of the future of this research, demonstrating how the presented proposals on FN architectures, service composition and the associated costing framework can scale up on an Internet environment.

Therefore, in future work we (myself and my current research group are focused current research group are focused on analyzing and testing scalability issues from two major perspectives. Firstly, to provide realistic and larger scenarios and, secondly, to find composition mechanisms that present satisfactory tradeoffs between performance and response time in different environments. Messages format is also an open issue that needs to be explored to reduce the overhead of the discovery protocol. It also remains as future work the study of reservation mechanisms and their implications when services are unavailable during the allocation phase.

And finally, to continue validating the usage of Service Composition methodology for other fields and use cases, especially targeting Smart Grid applications and the development of the SDU concept. Components shown in Chapter 5.7 are just a proof-of-concepts that are still in development not only by the author. They opened different research lines that are being continued by other members of the GRITS research group. The developments of INTEGRIS and FINESCE have been continued internally by the group for providing a complete SDU solution that enables the deployment of the different developments at large scale, resulting on a specific new research line on scalable high-performance web infrastructure for managing all the data of the SDU. The evolution of the Smart Grid concept in the recent years and the decrease of reluctance of electric utilities to bet for new technologies, a great potential was identified in the application of Service Composition and SDN technologies for providing flexible Smart Grid services at the edge. By means of already standardized technologies such as WSDL (for service interfaces definition) and OSGi (for service implementation), it will be interesting in the following years to define and deploy other Smart Grid use cases such as fault detection services, electrical vehicle charger points management or self-healing substations.

It is important to mention the current research lines that we are working on with my research group, in which we are focusing on providing flexible and context-aware communication architectures that enable inherent on-demand deployment of new services (fog computing services, enhanced security protocol mechanisms and tools, specific cross-layer solutions for QoS improvement, or solutions for maximizing bandwidth in high-demanding media transmissions). Software Defined Networks (SDN) and Future Network Architectures trends, such as Service-Oriented Networks or Information Centric Networks (ICNs), are still in the scope of our research topics for providing those intelligent network solutions adapted to specific context requirements of the networks, the data and the end users. The group is studying all of these topics separately or together with the other research groups of La Salle, and our immediate objective is the preparation of H2020 proposals for different 2017 calls that help to fund these research lines. First proposal is a Coordination and Support Action (CSA) together with La Salle Technova technology incubator park, in order to foster the innovation in the Smart Cities and help the research and development tasks of SMEs. The other one is a Research and Innovation Action on the Smart Cities environment data monitoring, proposing a service-oriented Web of Things architecture for managing the collected data, designed in collaboration by different research groups of La Salle. At a medium term, it is also important to point out that La Salle institution aims at a higher integration of its research groups and it is funneling it through a new entity called Business Institute of Technology La Salle (BITLaSalle). This entity will promote the integration of the research done by GRITS and the other La Salle research groups and promote their technology and knowledge transfer to the industry.

8 References

[4WARD] 4WARD. 2009. European Union's 7th Framework Program (FP7). Project website. [ONLINE] Available at: http://www.4ward-project.eu/. [Accessed 31st August 2016].

[Agrawal et al., 1998] Agrawal, R., Gunopulos, D. and Leymann, F., 1998, March. Mining process models from workflow logs. In International Conference on Extending Database Technology (pp. 467-483). Springer Berlin Heidelberg.

[Ahlgren et al., 2012] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. and Ohlman, B., 2012. A survey of information-centric networking. IEEE Communications Magazine, 50(7), pp.26-36.

[Alamri et al., 2006] Alamri, A., Eid, M. and El Saddik, A., 2006. Classification of the state-of-the-art dynamic web services composition techniques. International Journal of Web and Grid Services, 2(2), pp.148-166.

[Al-Fuqaha et al., 2015] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., 2015. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), pp.2347-2376.

[Amadeo et al., 2016] Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R.L. and Vasilakos, A.V., 2016.

[Aman et al., 2013] Aman, S., Simmhan, Y. and Prasanna, V.K., 2013. Energy management systems: state of the art and emerging trends. IEEE Communications Magazine, 51(1), pp.114-119.

[Ambroszkiewicz, 2003] Ambroszkiewicz, S., 2003. Entish: An approach to service description and composition. Orientation and Web Services, p.10.

[Anderson et al., 2005] Anderson, T., Peterson, L., Shenker, S. and Turner, J., 2005. Overcoming barriers to disruptive innovation in networking. In Report of NSF Workshop, 2005.

[Ankolekar et al., 2002] Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T. and Sycara, K., 2002. DAML-S: Web service description for the semantic web. In The Semantic Web—ISWC 2002 (pp. 348-363). Springer Berlin Heidelberg.

[Antonova et al., 2011] Antonova, G., Frisk, L. and Tournier, J.C., 2011, April. Communication redundancy for substation automation. In Protective Relay Engineers, 2011 64th Annual Conference for (pp. 344-355). IEEE.

[Azcorra et al., 2009] Azcorra, A., Laso-Ballesteros, I., Daras, P. and Guerrero, C., 2009., European Commission.

[Baker et al., 2009] Baker, N., Zafar, M., Moltchanov, B. and Knappmeyer, M., 2009, April. Context-Aware Systems and Implications for Future Internet. In Future Internet Assembly (pp. 335-344).

[Bari et al., 2012] Bari, M.F., Chowdhury, S.R., Ahmed, R., Boutaba, R. and Mathieu, B., 2012. A survey of naming and routing in information-centric networks. IEEE Communications Magazine, 50(12), pp.44-53.

[Bari et al., 2014] Bari, A., Jiang, J., Saad, W. and Jaekel, A., 2014. Challenges in the smart grid applications: an overview. International Journal of Distributed Sensor Networks, 2014.

[Barret, 2013] Barrett, B., 2013. "What is PRISM?" [ONLINE]. Available: http://gizmodo.com/what-is-prism-511875267. [Accessed 31st August 2016].

[Bellovin, 1989] Bellovin, S.M., 1989. Security problems in the TCP/IP protocol suite. ACM SIGCOMM Computer Communication Review, 19(2), pp.32-48.

[Bellwood et al., 2002] Bellwood, T., Clément, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y.L., Januszewski, K., Lee, S., McKee, B., Munter, J. and von Riegen, C., 2002. UDDI Version 3.0. Published specification, Oasis, 5, pp.16-18.

[Benatallah et al., 2002] Benatallah, B., Dumas, M., Sheng, Q.Z. and Ngu, A.H., 2002. Declarative composition and peer-to-peer provisioning of dynamic web services. In Data Engineering, 2002. Proceedings. 18th International Conference on (pp. 297-308). IEEE.

[Beradi et al., 2003] Berardi, D., Calvanese, D., De Giacomo, G. and Mecella, M., 2003, June. Reasoning about actions for e-service composition. In Proceedings of ICAPS (Vol. 3).

[Bergkvist et al., 2012] Bergkvist, A., Burnett, D. and Jennings, C., 2012. A. Narayanan," WebRTC 1.0: Real-time Communication Between Browsers. World Wide Web Consortium WD WD-webrtc-20120821.

[Bo et al., 2014] Bo, C., Xin, C., Zhongyi, Z., Chengwen, Z. and Junliang, C., 2014. Web of things-based remote monitoring system for coal mine safety using wireless sensor network. International Journal of Distributed Sensor Networks, 2014.

[Bormann et a., 2012] Bormann, C., Castellani, A.P. and Shelby, Z., 2012. Coap: An application protocol for billions of tiny internet nodes. IEEE Internet Computing, 16(2), p.62.

[Bouabene et al., 2010] Bouabene, G., Jelger, C., Tschudin, C., Schmid, S., Keller, A. and May, M., 2010. The autonomic network architecture (ANA). IEEE Journal on Selected Areas in Communications, 28(1), pp.4-14.

[Bou-Harb et al., 2013] Bou-Harb, E., Fachkha, C., Pourzandi, M., Debbabi, M. and Assi, C., 2013. Communication security for smart grid distribution networks. IEEE Communications Magazine, 51(1), pp.42-49.

[Bonomi et al., 2012] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S., 2012, August. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13-16). ACM.

[Box et al., 2000] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S. and Winer, D., 2000. Simple object access protocol (SOAP) 1.1.

[Bovet et al., 2013] Bovet, G. and Hennebert, J., 2013, June. A web-of-things gateway for knx networks. In Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on (pp. 1-8). VDE.

[Braden et al., 2003] Braden, R., Faber, T. and Handley, M., 2003. From protocol stack to protocol heap: role-based architecture. ACM SIGCOMM Computer Communication Review, 33(1), pp.17-22.

[Bramley et al., 2000] Bramley, R., Chiu, K., Diwan, S., Gannon, D., Govindaraju, M., Mukhi, N., Temko, B. and Yechuri, M., 2000. A component based services architecture for building distributed applications. In High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on (pp. 51-59). IEEE.

[Briones et al., 2016] Briones, A., Martin de Pozuelo, R., Navarro, J. and Zaballos, A., 2016. Resource Allocation on a Hybrid Cloud for Smart Grids. Network Protocols and Algorithms, 8(1), pp.7-25.

[Brooks et al. 2010] Brooks, P. and Hestnes, B., 2010. User measures of quality of experience: why being objective and quantitative is important. IEEE network, 24(2), pp.8-13.

[Buchiarone et al., 2006] Buchiarone A. and Gnesi, S., 2006, May. A survey on services composition languages and models. In International Workshop on Web Services–Modeling and Testing (WS-MaTe 2006) (p. 51).

[Burness et al., 2009] Burness, L., Eardley, P. Y Hancock, R. 2009. The Trilogy Architecture for the Future Internet. Towards the Future Internet. (2009), 79.

[Carnec et al., 2003] Carnec, M., Le Callet, P. and Barba, D., 2003, July. Full reference and reduced reference metrics for image quality assessment. In Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on (Vol. 1, pp. 477-480). IEEE.

[Chandrashekar et al., 2003] Chandrashekar, J., Zhang, Z.L., Duan, Z. and Hou, Y.T., 2003, December. Service oriented internet. In International Conference on Service-Oriented Computing (pp. 543-558). Springer Berlin Heidelberg.

[Channa et al., 2005] Channa, N., Li, S., Shaikh, A.W. and Fu, X., 2005, August. Constraint satisfaction in dynamic web service composition. In 16th International Workshop on Database and Expert Systems Applications (DEXA'05) (pp. 658-664). IEEE.

[Chen et al., 2003] Chen, H., Finin, T. and Joshi, A., 2003. An ontology for context-aware pervasive computing environments. The knowledge engineering review, 18(03), pp.197-207.

[Chinnici et al., 2007] Chinnici, R., Moreau, J. J., Ryman, A., & Weerawarana, S. (2007). Web services description language (wsdl) version 2.0 part 1: Core language. W3C recommendation, 26, 19

[Christensen, et al. 2001] Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S., 2001. Web services description language (WSDL) 1.1.

[Christopoulou et al., 2005] Christopoulou, E. and Kameas, A., 2005. GAS ontology: an ontology for collaboration among ubiquitous computing devices. International Journal of Human-Computer Studies, 62(5), pp.664-685.

[Chu et al., 2013] Chu, H.Y. and Simmhan, Y., 2013. Resource allocation strategies on hybrid cloud for resilient jobs. Small, 1005(1), pp.0-065.

[CISCO, 2016] CISCO. 2016. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper. [ONLINE] Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html [Accessed 31st August 2016].

[Clark et al., 2003] Clark, D.D., Sollins, K. and Wroclawski, J., 2003. Newarch project: Future-generation internet architecture.

[Clark et al., 2005] D. Clark, C. Partridge, R. T. Braden, B. Davie, S. Floyd, Van Jacobson, D. Katabi, G. Minshall, K.K. Ramakrishnan, T. Roscoe, I. Stoica, J. Wroclawski and L. Zhang. “Making the World (of Communications) a Different Place”. ACM SIGCOMM Computer Communication Review. Volume 35, Number 2, July 2005.

[Clark et al., 2007] Clark, D.D., Faratin, P., Bauer, S., Lehr, W., Gilmore, P.W. and Berger, A., 2008. The growing complexity of Internet interconnection. Communications & strategies, (72), p.51.

[Cook et al., 1999] Cook, J.E. and Wolf, A.L., 1999. Software process validation: quantitatively measuring the correspondence of a process to a model. ACM Transactions on Software Engineering and Methodology (TOSEM), 8(2), pp.147-176.

[Courcoubetis et al., 2009] Courcoubetis, C. and Man-Sze, L.I., 2009. Future Internet socio-economics–challenges and perspectives. Towards the Future Internet: A European Research Perspective, p.1.

[Cowen, 2001] Cowen, L.J., 2001. Compact routing with minimum stretch. Journal of Algorithms, 38(1), pp.170-183.

[Cupp et al., 2008] Cupp, J., Beehler, M. (2008). “Implementing Smart Grid Communications” TechBriefs Burns & McDonnell.

[D’Ambrosio et al., 2011] D'Ambrosio, M., Dannewitz, C., Karl, H. and Vercellone, V., 2011, August. MDHT: a hierarchical name resolution service for information-centric networks. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking (pp. 7-12). ACM.

[Day, 2007] Day, J., 2007. Patterns in network architecture: a return to fundamentals. Pearson Education.

[Day et al., 2008] Day, J., Matta, I. and Mattar, K., 2008, December. Networking is IPC: a guiding principle to a better internet. In Proceedings of the 2008 ACM CoNEXT Conference (p. 67). ACM.

[De Brujin et al., 2005] De Bruijn, J., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Kifer, M., König-Ries, B., Kopecky, J., Lara, R., Oren, E. and Polleres, A., 2005. Web service modeling ontology (wsmo). Interface, 5, p.1.

[Dey, 2001] Dey, A.K., 2001. Understanding and using context. Personal and ubiquitous computing, 5(1), pp.4-7.

[Domingo et al., 2008] Domingo, V., Enrich, F., Martin de Pozuelo, R., Pinyol, F. and Fernández, G., 2008, April. Delivery of H264 SVC/MDC streams over Wimax and DVB-T networks. In 2008 IEEE International Symposium on Consumer Electronics.

[Dustdar et al., 2005] Dustdar, S. and Schreiner, W., 2005. A survey on web services composition. International journal of web and grid services, 1(1), pp.1-30.

[Dutta et al., 2007] R. Dutta, G. Rouskas, I. Baldine, A. Bragg, and D. Stevenson, "The silo architecture for services integration, control, and optimization for the future internet," in Communications, 2007. ICC '07. IEEE International Conference on, June 2007, pp. 1899–1904.

[Enrich et al., 2008] Enrich, F., Domingo, V., Martin de Pozuelo, R., Pinyol, F. and Fernández, G., 2008, March. Intelligent playout for the optimal IP distribution of H264 SVC/MDC services over broadband and broadcast platforms. In 2008 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting.

[Ericsson, 2010] Ericsson, G.N., 2010. Cyber security and power system communication—essential parts of a smart grid infrastructure. IEEE Transactions on Power Delivery, 25(3), pp.1501-1507.

[EVERYTHING] EVRYTHNG—Every Thing Connected, April 2015, [Online] Available at: https://evrythng.com [Accessed 10th October 2016]

[Feldmann et al., 2007] Feldmann, A., 2007. Internet clean-slate design: what and why?. ACM SIGCOMM Computer Communication Review, 37(3), pp.59-64.

[Ferrer et al., 2012] Ferrer, J.L., Sanchez-Loro, X., Calveras, A. and Paradells, J., 2012, September. Implementation of context-aware network architecture for smart objects based on functional composition. In International Conference on Mobile Networks and Management (pp. 207-216). Springer Berlin Heidelberg.

[FIND] NSF NeTS FIND Initiative. 2006. Project Website. [ONLINE] Available at: http://www.nets-find.net/index.php. [Accessed 31st August 2016].

[FINESCE] FINESCE. 2015. European Union's 7th Framework Program (Future Internet PPP) Project "Future INtErnet for Smart Utility ServiCEs". FI.ICT-2011 Grant 604677. Project website. [ONLINE]. Available: http://www.finesce.eu. [Accessed 31st August 2016].

[FINSENY] FINSENY. 2012. European Union's 7th Framework Program (Future Internet PPP) Project "Future INtErnet for Smart Energy". Project website. [ONLINE]. Available: http://www.fi-ppp-finseny.eu. [Accessed 31st August 2016].

[Fisher, 2014] Fisher, D., 2014. A look behind the future internet architectures efforts. ACM SIGCOMM Computer Communication Review, 44(3), pp.45-49.

[FIWARE] FIWARE. 2014. European Union's 7th Framework Program (Future Internet PPP) Project FIWARE. Project website. [ONLINE]. Available: https://www.fiware.org/. [Accessed 31st August 2016].

[Fulgham et al., 2009] Fulgham, B. and Gouy, I., 2009. The computer language benchmarks game. [ONLINE] Available at: http://shootout.alioth.debian.org. [Accessed 31st August 2016].

[FUSION] FUSION. 2013. European Union's 7th Framework Program (FP7). Project website. [ONLINE] Available at: http://www.fusion-project.eu/ [Accessed 31st August 2016]

[Galinier et al., 2006] Galinier, P. and Hertz, A., 2006. A survey of local search methods for graph coloring. Computers & Operations Research, 33(9), pp.2547-2562.

[Gartner, 2013] Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020, Stamford, Conn, USA, December 2013. [Online]. Available at: http://www.gartner.com/newsroom/id/2636073 [Last Access: 9th October 2016].

[Gelernter et al., 1992] Gelernter, D. and Carriero, N., 1992. Coordination languages and their significance. Communications of the ACM, 35(2), p.96.

[Genez et al., 2013] Genez, T.A., Bittencourt, L.F. and Madeira, E.R., 2013, December. On the performance-cost tradeoff for workflow scheduling in hybrid clouds. In Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (pp. 411-416). IEEE Computer Society.

[GENI] GENI.net Global Environment for Network Innovations. Project website. 2008. [ONLINE] Available at: http://www.geni.net/. [Accessed 31st August 2016].

[Ghafoor, 2014] Ghafoor, I., Jattala, I., Durrani, S. and Tahir, C.M., 2014, December. Analysis of OpenSSL heartbleed vulnerability for embedded systems. In Multi-Topic Conference (INMIC), 2014 IEEE 17th International (pp. 314-319). IEEE.

[Ghallab et al., 2004] Ghallab, M., Nau, D. and Traverso, P., 2004. Automated planning: theory & practice. Elsevier.

[Ghodsi et al., 2011] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B. and Wilcox, J., 2011, November. Information-centric networking: seeing the forest for the trees. In Proceedings of the 10th ACM Workshop on Hot Topics in Networks (p. 1). ACM.

[Giorgetti et al., 2013] Giorgetti, A., Cugini, F., Paolucci, F., Valcarenghi, L., Pistone, A. and Castoldi, P., 2013. Performance analysis of media redundancy protocol (MRP). IEEE Transactions on Industrial Informatics, 9(1), pp.218-227

[Gold, 1978] Gold, E.M., 1978. Complexity of automaton identification from given data. Information and control, 37(3), pp.302-320.

[Gonzalez et al., 2011] Gonzalez, A.J., Martin de Pozuelo, R., Alcober, J., Pinyol, F. and Ghafoor, K.Z., 2011, July. Context-aware multimedia service composition using quality assessment. In 2011 IEEE International Conference on Multimedia and Expo (pp. 1-6). IEEE.

[Gonzalez et al., 2013] Gonzalez, A.J., Martin de Pozuelo, R., German, M., Alcober, J. and Pinyol, F., 2013. New framework and mechanisms of context-aware service composition in the future internet. ETRI journal, 35(1), pp.7-17.

[Goransson et al., 2014] Goransson, P. and Black, C., 2014. Software Defined Networks: A Comprehensive Approach. Elsevier.

[Greco et al., 2004] Greco, G., Guzzo, A., Pontieri, L. and Sacca, D., 2004, May. Mining expressive process models by clustering workflow traces. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 52-62). Springer Berlin Heidelberg.

[Gruber, 1995] Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, 43(5), pp.907-928.

[Gu et al., 2004] Gu, T., Pung, H.K. and Zhang, D.Q., 2004. Toward an OSGi-based infrastructure for context-aware applications. IEEE Pervasive Computing, 3(4), pp.66-74.

[Guérin et al., 1999] Guérin, R. and Peris, V., 1999. Quality-of-service in packet networks: basic mechanisms and directions. Computer Networks, 31(3), pp.169-189.

[Guinard et al., 2011] Guinard, D., Trifa, V., Mattern, F. and Wilde, E., 2011. From the internet of things to the web of things: Resource-oriented architecture and best practices. In Architecting the Internet of Things (pp. 97-129). Springer Berlin Heidelberg.

[Gungor et al., 2013] Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C. and Hancke, G.P., 2013. A survey on smart grid potential applications and communication requirements. IEEE Transactions on Industrial Informatics, 9(1), pp.28-42.

[Gungor et al., 2011] Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C. and Hancke, G.P., 2011. Smart grid technologies: communication technologies and standards. IEEE transactions on Industrial informatics, 7(4), pp.529-539.

[GWA, 2014] Department of Energy's Office of Electricity Delivery and Energy Reliability (OE) and the GridWise Alliance (GWA) "Future of the Grid: Evolving to Meet America's Needs" pp.1-40 (2014).

[HaiVision, 2011] HaiVision. 2011. MPEG-4 AVC (H.264) and why, only now, it can save 60% of the network video bandwidth and storage requirements. [ONLINE] Available at: http://www.infocomm.org/cps/rde/xbcr/infocomm/VideoCompressionMPEG4AVC%28H.264%29.pdf. [Accessed 31st August 2016].

[Han et al., 2015] Han, B., Gopalakrishnan, V., Ji, L. and Lee, S., 2015. Network function virtualization: Challenges and opportunities for innovations. IEEE Communications Magazine, 53(2), pp.90-97.

[Hancock et al., 2005] Hancock, R., Loughney, J., Karagiannis, G. and Bosch, S.V.D., 2005. Next steps in signaling (NSIS): Framework.

[Harai, 2009] Harai, H., 2009, November. Designing new-generation network: Overview of akari architecture design. In Asia Communications and Photonics Conference and Exhibition (p. FL2). Optical Society of America.

[Hart et al., 1968] Hart, P.E., Nilsson, N.J. and Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics, 4(2), pp.100-107.

[He, 2014] He, D., Chan, S., Zhang, Y., Guizani, M., Chen, C. and Bu, J., 2014. An enhanced public key infrastructure to secure smart grid wireless communication networks. IEEE Network, 28(1), pp.10-16.

[Heineman et al., 2001] Heineman, G.T. and Councill, W.T., 2001. Component-based software engineering. Putting the pieces together, addison-westley, p.5.

[Herzog, 2014] Herzog, C. 2014. 5 smart grid and smart city predictions for 2020. GreenBiz website. [ONLINE] Available at: https://www.greenbiz.com/blog/2014/01/08/5-smart-grid-smart-city-predictions-2020. [Accessed 23rd September 2016]

[Horrocks, 2002] Horrocks, I., 2002. DAML+OIL: A Description Logic for the Semantic Web.IEEE Data Eng. Bull., 25(1), pp.4-9.

[Hossmann et al., 2008] Hossmann, T., Keller, A., Dudler, S. and May, M., 2008, June. Implementing the Future Internet: A Case Study of Service Discovery using Pub/Sub in the ANA Framework. In Proc. International Conference on Future Internet Technologies (CFI'08).

[Huhns et al., 2005] Huhns, M.N. and Singh, M.P., 2005. Service-oriented computing: Key concepts and principles. Internet Computing, IEEE, 9(1), pp.75-81.

[i3MEDIA] i3Media. 2008. Project Website. [ONLINE] Available at: https://i3media.barcelonamedia.org/. [Accessed 31st August 2016].

[IEC 61850] International Electrotechnical Commission IEC 61850:2016 SER Series. Communication networks and systems for power utility automation - ALL PARTS, 2016.

[IEC 62351] International Electrotechnical Commission IEC 62351, Power systems management and associated information exchange - Data and communications security-Part 6: Security for IEC 61850, (2009).

[IEC 62439] International Electrotechnical Comission, "High Availability Automation Networks", IEC 62439 (2008).

[IEC 62439-6] IEC 62439-6: "Industrial Communication Networks: High Availability Automation Netwroks" - Part 6: Distributed Redundancy Protocol (DRP)

[IEEE1646] IEEE, Institute of Electrical and Electronics Engineers 1646-2004, "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation".

[INTEGRIS] INTEGRIS. 2013. European Union's 7th Framework Program (FP7) Project "INTelligent Electrical Grid Sensor communications". ICT-Energy-2009 call (number 247938). Project website. [ONLINE]. Available at: http://fp7integris.eu. [Accessed 31st August 2016].

[ISO/IEC TR 29181] ISO/IEC JTC1/SC6/WG7, "ISO/IEC TR 29181-1 Information technology -- Future Network -- Problem statement and requirements -- Part 1: Overall aspects," 2012.

[ONLINE]. Available at: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber= 57480. [Accessed 31st August 2016].

[ISO/IEC TR 29181-7] ISO/IEC JTC1/SC6/WG7, "ISO/IEC TR 29181-7 Information technology -- Future Network -- Problem statement and requirements -- Part 7: Service composition," 2013. [ONLINE]. Available at: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57489. [Accessed 31st August 2016].

[ITU-R, 2001] ITU-R, "Method for objective measurements of per-ceived audio quality," Recommendation BS.1387-1, International Telecommunication Union, 2001.

[Johnson et al., 2002] Johnson, A.W. and Jacobson, S.H., 2002. On the convergence of generalized hill climbing algorithms. Discrete Applied Mathematics, 119(1), pp.37-57.

[Kandadai, 2008] Kandadai, S., Hardin, J. and Creusere, C.D., 2008, March. Audio quality assessment using the mean structural similarity measure. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 221-224). IEEE.

[Kanter et al., 2009] Kanter, T., Pettersson, S., Forsström, S., Kardeby, V., Norling, R., Walters, J. and Österberg, P., 2009, August. Distributed context support for ubiquitous mobile awareness services. In Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on (pp. 1-5). IEEE.

[Katabi et al., 2002] Katabi, D., Handley, M. and Rohrs, C., 2002. Congestion control for high bandwidth-delay product networks. ACM SIGCOMM computer communication review, 32(4), pp.89-102.

[Keeney et al., 1993] Keeney, R.L. and Raiffa, H., 1993. Decisions with multiple objectives: preferences and value trade-offs. Cambridge university press.

[Khan et al., 2013] Khan, R.H. and Khan, J.Y., 2013. A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network. Computer Networks, 57(3), pp.825-845.

[Khondoker et al., 2010] Khondoker, R., Reuther, B., Schwerdel, D., Siddiqui, A. and Müller, P., 2010, December. Describing and selecting communication services in a service oriented network architecture. In 2010 ITU-T Kaleidoscope: Beyond the Internet? Innovations for Future Networks and Services (pp. 1-8). IEEE.

[Kirrmann et al, 2007] Kirrmann, H., Hansson, M. and Muri, P., 2007, September. IEC 62439 PRP: Bumpless recovery for highly available, hard real-time industrial networks. In 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007) (pp. 1396-1399). IEEE.

[Kirrmann et al, 2009] Kirrmann, H., Weber, K., Kleineberg, O. and Weibel, H., 2009, September. HSR: Zero recovery time and low-cost redundancy for Industrial Ethernet (High availability seamless redundancy, IEC 62439-3). In Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation (pp. 203-206). IEEE Press.

[Kirrmann et al, 2012] Kirrmann, H., 2012. Highly available automation networks standard redundancy methods; rationales behind the IEC 62439 standard suite.

[Korf et al., 1998] Korf, R.E. and Reid, M., 1998, July. Complexity analysis of admissible heuristic search. In AAAI/IAAI (pp. 305-310).

[Korf et al., 2001] Korf, R.E., Reid, M. and Edelkamp, S., 2001. Time complexity of iterative-deepening-A∗. Artificial Intelligence, 129(1), pp.199-218.

[Kornich, 2016] Kornich, J. 2016. Azure Media Services overview and common scenarios. Microsoft Azure website. [ONLINE] Available at: https://azure.microsoft.com/en-us/documentation/articles/ media-services-overview/ [Accessed 22nd September 2016].

[Korpipää et al., 2004] Korpipää, P., Häkkilä, J., Kela, J., Ronkainen, S. and Känsälä, I., 2004, October. Utilising context ontology in mobile device application personalisation. In Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia (pp. 133-140). ACM.

[Kuipers et al, 2002] Kuipers, F., Van Mieghem, P., Korkmaz, T. and Krunz, M., 2002. An overview of constraint-based path selection algorithms for QoS routing. IEEE Communications Magazine, 40 (12).

[Kumar, 1992] Kumar, V., 1992. Algorithms for constraint-satisfaction problems: A survey. AI magazine, 13(1), p.32.

[Kumar et al., 2008] Kumar, S. and Mishra, R.B., 2008. Trs: System for recommending semantic web service composition approaches. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2(11), pp.3679-3682.

[Kumar et al., 2010] Kumar, R., Haber, A., Yazidi, A. and Reichert, F., 2010, January. Towards a relation oriented service architecture. In Communication Systems and Networks (COMSNETS), 2010 Second International Conference on (pp. 1-8). IEEE.

[Küester et al., 2005] Küster, U., Stern, M. and König-Ries, B., 2005, December. A classification of issues and approaches in automatic service composition. In Intl Workshop WESC (Vol. 5, pp. 25-35).

[Kuzlu et al., 2013] Kuzlu, M. and Pipattanasomporn, M., 2013, February. Assessment of communication technologies and network requirements for different smart grid applications. In Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES (pp. 1-6). IEEE.

[Lalanda et al., 2007] Lalanda, P. and Marin, C., 2007. A domain-configurable development environment for service-oriented applications. Software, IEEE, 24(6), pp.31-38.

[Lee et al., 2011] Lee, J.W., Francescangeli, R., Song, W., Janak, J., Srinivasan, S., Kester, M., Baset, S.A., Liu, E., Schulzrinne, H., Hilt, V. and Despotovic, Z., 2011. Netserv framework design and implementation 1.0. Technical Report cucs-016-11, Columbia University.

[Li et al., 2010] Li, H. and Zhang, W., 2010, December. QoS routing in smart grid. In Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE (pp. 1-6). IEEE.

[Li et al., 2013] Li, B., Wang, Z., Liu, J. and Zhu, W., 2013. Two decades of internet video streaming: A retrospective view. ACM transactions on multimedia computing, communications, and applications (TOMM), 9(1s), p.33.

[Li et al., 2014] Li, S., Zhou, Y., Jiao, L., Yan, X., Wang, X. and Lyu, M.R., 2014, June. Delay-aware cost optimization for dynamic resource provisioning in hybrid clouds. In Web Services (ICWS), 2014 IEEE International Conference on (pp. 169-176). IEEE.

[Linder, 2000] Linder, J. and Cantrell, S., Changing Business Models: Surveying the Landscape. 2000. accenture Institute for Strategic Change.

[Locke, 2010] Locke, D., 2010. Mq telemetry transport (mqtt) v3. 1 protocol specification. IBM developer Works Technical Library. [Online] Available at http://www.ibm.com/ developerworks/webservices/library/ws-mqtt/index. [Accessed 9th October 2016].

[Malawski et al., 2013] Malawski, M., Figiela, K. and Nabrzyski, J., 2013. Cost minimization for computational applications on hybrid cloud infrastructures. Future Generation Computer Systems, 29(7), pp.1786-1794.

[Mazhelis et al., 2011] Mazhelis, O. and Tyrvainen, P., 2011, August. Role of data communications in hybrid cloud costs. In 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (pp. 138-145). IEEE.

[Martin et al. 2007] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T. and Sirin, E., 2004. OWL-S: Semantic markup for web services. W3C member submission, 22, pp.2007-04.

[Martín de Pozuelo et al., 2010] Martin de Pozuelo, R., Enrich, F. and Fernández, G. 2010. DVB-PCF to Flash Lite Transcoder. In EuroITV '10, Proceedings of the 8th international interactive conference on Interactive TV & Video.

[Martín de Pozuelo et al., 2016] Martin de Pozuelo, R., Ponce de Leon, M., Howard, J., Briones, A., Horgan, J. 2016. Software Defined Utility: a step towards a flexible, reliable and low-cost Smart Grid. In ICSGS'16, Proceedings of the 5th International Conference on Smart Grid Systems.

[Mathis et al., 2006] Mathis, M. and Reddy, R., 2006. Enabling high performance data transfers.

[Matsubara et al. 2013] Matsubara, D., Egawa, T., Nishinaga, N., Kafle, V.P., Shin, M.K. and Galis, A., 2013. Toward future networks: a viewpoint from ITU-T. IEEE Communications Magazine, 51(3), pp.112-118.

[Mattar et al., 2009] Mattar, K., Matta, I., Day, J., Ishakian, V. and Gursun, G., 2009. Declarative transport: a customizable transport service for the future internet. In in Proceedings of the 5th International Workshop on Networking Meets Databases (NetDB 2009), co-located with SOSP 2009, Big Sky, MT, October 2009.

[Manzoni et al., 1995] Manzoni, P., Ghosal, D. and Serazzi, G., 1995. Impact of mobility on TCP/IP: an integrated performance study. IEEE Journal on Selected Areas in Communications, 13(5), pp.858-867.

[McGuinness et al., 2004] McGuinness, D.L. and Van Harmelen, F., 2004. OWL web ontology language overview. W3C recommendation, 10(10), p.2004.

[McIlraith et al., 2002] McIlraith, S.A. and Fadel, R., 2002, April. Planning with complex actions. In NMR (pp. 356-364).

[Mennie et al., 2000] Mennie, D. and Pagurek, B., 2000. An architecture to support dynamic composition of service components. Systems and Computer Engineering. Carleton University, Canada.

[Meyer et al., 2005] Meyer, H. and Kuropka, D., 2005. Requirements for service composition.

[Miller et al., 2014] Miller, C., Martin, M., Pinney, D. and Walker, G., 2014. Achieving a Resilient and Agile Grid. National Rural Electric Cooperative Association (NRECA): Arlington, VA, USA.

[Miorandi et al., 2012] Miorandi, D., Sicari, S., De Pellegrini, F. and Chlamtac, I., 2012. Internet of things: Vision, applications and research challenges. Ad Hoc Networks, 10(7), pp.1497-1516.

[Monsanto et al., 2013] Monsanto, C., Reich, J., Foster, N., Rexford, J. and Walker, D., 2013. Composing software defined networks. In Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13) (pp. 1-13).

[Mooney, 2010] Mooney, G., 2010. Evaluating Compact Routing Algorithms on Real-World Networks. project dissertation, University of Glasgow, Glasgow, Scotland, UK.

[Moreno-Vozmediano et al., 2013] Moreno-Vozmediano, R., Montero, R.S. and Llorente, I.M., 2013. Key challenges in cloud computing: Enabling the future internet of services. IEEE Internet Computing, 17(4), pp.18-25.

[Mrissa et al., 2005] Mrissa, M., Benslimane, D., Maamar, Z. and Ghedira, C., 2005. Towards a semantic-and context-based approach for composing web services. International Journal of Web and Grid Services, 1(3-4), pp.268-286.

[Müller et al., 2007] Müller, P., Reuther, B. and Hillenbrand, M., 2007. Future Internet: A Service-Oriented Approach-SONATE.

[Navarro et al., 2013] Navarro, J., Zaballos, A., Sancho-Asensio, A., Ravera, G. and Armendáriz-Iñigo, J.E., 2013. The information system of INTEGRIS: Intelligent electrical grid sensor communications. IEEE Transactions on Industrial Informatics, 9(3), pp.1548-1560.

[Navarro et al., 2014] Navarro, J., Sancho-Asensio, A., Zaballos, A., Jiménez-Ruano, V., Vernet, D. and Armendáriz-Iñigo, J.E., 2014. The Management System of INTEGRIS-Extending the Smart Grid to the Web of Energy. In CLOSER (pp. 329-336).

[NETSERV] Netserv 2011. Project Website. [ONLINE] Available at: http://www.cs.columbia.edu/ irt/project/netserv/?page=about [Accessed 31st August 2016].

[NISTIR7628, 2014] National Institute of Standards and Technology, "NISTIR 7628, Revision 1, Guidelines for Smart Grid Cyber security, Volume 1, Smart Grid Cybersecurity Strategy, Architecture, and High Level Requirements", The Smart Grid Interoperability Panel - Smart Grid Cybersecurity Committee, pp:1-668, (2014).

[NEURA] Neura. 2015. Official Website. [ONLINE] Available at: http://www.theneura.com/. [Accessed 31st August 2016].

[NFV, 2015] ETSI, G., 2015. 001: Network Functions Virtualization (NFV); Infrastructure Overview v1. 1.1. ETSI, tech. rep.

[NodeJS] Node.js. Official Website. June 2015. [ONLINE] Available at: https://nodejs.org/. [Accessed 31st August 2016].

[Node-RED] Node-RED. 2015. Official Website. [ONLINE] Available at: http://nodered.org/. [Accessed 31st August 2016].

[OCTOBLU] Octoblu, "Wemake all APIs, Platforms and Devices talk to each other. Easily," April 2015. [ONLINE] Available at: http://octoblu.com/ [Accessed 31st August 2016].

[Odersky et al., 2004] Odersky, M., Altherr, P., Cremet, V., Emir, B., Maneth, S., Micheloud, S., Mihaylov, N., Schinz, M., Stenman, E. and Zenger, M., 2004. An overview of the Scala programming language (No. LAMP-REPORT-2004-006).

[ONCOR] Oncor Smart Grid Technology. 2012. [ONLINE] Available at: http://www.oncor.com/ EN/Pages/Smart-Grid-Technology.aspx. [Accessed 02nd November 2016].

[Oriwoh et al., 2015] Oriwoh, E. and Conrad, M., 2015. 'Things' in the Internet of Things: Towards a Definition. International Journal of Internet of Things, 4(1), pp.1-5.

[OSGI] OSGI Alliance, 2007. Osgi service platform, core specification, release 4, version 4.1. OSGi Specification.

[Oualmakran et al., 2011] Oualmakran, Y., Meléndez, J., Herraiz, S., López-Perea, M. and González, E., 2011. Survey on knowledge based methods to assist fault restoration in power distribution networks. In International Conference on Renewable Energies and Power Quality (ICREPQ'11). Las Palmas (Spain).

[PECANSTREET] Dataport. Pecan Street Research Institute. [ONLINE] Available at: https://dataport.pecanstreet.org/. [Accessed 21st September 2016].

[Papazoglou et al., 2007] Papazoglou, M.P., Traverso, P., Dustdar, S. and Leymann, F., 2007. Service-oriented computing: State of the art and research challenges. Computer, (11), pp.38-45.

[Peltz, 2003] Peltz, C., 2003. Web services orchestration and choreography. Computer, (10), pp.46-52.

[Peng et al., 2010] Peng, C., Vuorimaa, P., Naranen, P., Quico, C., Harboe, G., and Lugmayr, A. 2010. Adjunct Proceedings - EuroITV 2010 (web.sharing.tv.content), number ISBN 978-952-15-2355-7, Tampere University of Technology (TUT), Tampere, Finland.

[Pahl, 2005] Pahl, C., 2005. A conceptual architecture for semantic web services development and deployment. International Journal of Web and Grid Services, 1(3-4), pp.287-304.

[Pardo-Castellote, 2003] Pardo-Castellote, G., 2003, May. Omg data-distribution service: Architectural overview. In Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on (pp. 200-206). IEEE.

[Parks, 2011] Parks, R.C. and Duggan, D.P., 2011. Principles of cyberwarfare. IEEE Security & Privacy Magazine, 9(5), pp.30-35.

[PHALCONPHP] A Full-stack PHP Framework Delivered as a C-extension, Phalcon PHP, June 2015, [ONLINE] Available at: https://phalconphp.com/. [Accessed 31st August 2016].

[Preist, 2004] Preist, C., 2004. A conceptual architecture for semantic web services. In The Semantic Web–ISWC 2004 (pp. 395-409). Springer Berlin Heidelberg.

[Ponnekanti et al., 2002] Ponnekanti, S.R. and Fox, A., 2002, May. Sword: A developer toolkit for web service composition. In Proc. of the Eleventh International World Wide Web Conference, Honolulu, HI (Vol. 45).

[Quinton et al., 2009] Quinton, S., Ben-Hafaiedh, I. and Graf, S., 2009, September. From orchestration to choreography: Memoryless and distributed orchestrators. In FLACOS'09 Third Workshop on Formal Languages and Analysis of Contract-Oriented Software (p. 35).

[Quoitin et al., 2007] Quoitin, B., Iannone, L., De Launois, C. and Bonaventure, O., 2007, August. Evaluating the benefits of the locator/identifier separation. In Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture (p. 5). ACM.

[Ranganathan et al., 2003] Ranganathan, A., McGrath, R.E., Campbell, R.H. and Mickunas, M.D., 2003. Use of ontologies in a pervasive computing environment. The Knowledge Engineering Review, 18(03), pp.209-220.

[Rao et al., 2004] Rao, J. and Su, X., 2004. A survey of automated web service composition methods. In Semantic Web Services and Web Process Composition (pp. 43-54). Springer Berlin Heidelberg.

[Rao et al., 2011] Rao, A., Legout, A., Lim, Y.S., Towsley, D., Barakat, C. and Dabbous, W., 2011, December. Network characteristics of video streaming traffic. In Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies (p. 25). ACM.

[Raychaudhuri et al., 2005] Raychaudhuri, D. and Gerla, M., 2005, August. New architectures and disruptive technologies for the future internet: The wireless, mobile and sensor network perspective. In Report of NSF Wireless Mobile Planning Group (WMPG) Workshop.

[REACTPHP] React PHP. 2015. Official Website. [ONLINE] Available at: http://reactphp.org/. [Accessed 31st August 2016].

[Rexford et al., 2010] Rexford, J. and Dovrolis, C., 2010. Future Internet architecture: clean-slate versus evolutionary research. Communications of the ACM, 53(9), pp.36-40.

[RFC 2246] Internet Engineering Task Force, "RFC2246–The TLS Protocol" (1999).

[RNA] RNA: Recursive Network Architecture. 2008. Project website. [ONLINE] Available at: http://www.isi.edu/rna/. [Accessed 31st August 2016].

[Rodríguez-Molina et al., 2014] Rodríguez-Molina, J., Martínez-Núñez, M., Martínez, J.F. and Pérez-Aguiar, W., 2014. Business models in the smart grid: challenges, opportunities and proposals for prosumer profitability. Energies, 7(9), pp.6142-6171.

[Rosenberg et al., 2002] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E., 2002. SIP: session initiation protocol (No. RFC 3261).

[Ross-Talbot, 2005] Ross-Talbot, S., 2005, October. Orchestration and choreography: Standards, tools and technologies for distributed workflows. In NETTAB Workshop-Workflows management: new abilities for the biological information overflow, Naples, Italy (Vol. 1, p. 8).

[Russell et al., 2003] Russell, S.J., Norvig, P., Canny, J.F., Malik, J.M. and Edwards, D.D., 2003. Artificial intelligence: a modern approach (Vol. 2). Upper Saddle River: Prentice hall.

[Sahhaf et al., 2015] Sahhaf, S., Tavernier, W., Rost, M., Schmid, S., Colle, D., Pickavet, M. and Demeester, P., 2015. Network service chaining with optimized network function embedding supporting service decompositions. Computer Networks, 93, pp.492-505.

[Saint-Andre, 2011] Saint-Andre, P., 2011. Extensible messaging and presence protocol (XMPP): Core.

[Sallent et al., 2010] Sallent, S., Vidal, A., Madriles, O., Paradells, J., Sánchez, X., Alcober, J., González, A.J., Martin de Pozuelo, R., Pinyol, F., Cervelló, C., Jiménez, Y.A., Monguet, J.M. and Gutiérrez, A., 2010. The Atomic Redesign of the Internet Future Architecture.

[Sanchez-Loro et al., 2009] Sanchez-Loro, X., Casademont, J., Paradells, J., Ferrer, J.L. and Vidal, A., 2009, October. Proposal of a clean slate network architecture for ubiquitous services provisioning. In Future Information Networks, 2009. ICFIN 2009. First International Conference on (pp. 54-60). IEEE.

[Sanchez-Loro et al., 2010] Sanchez-Loro, X., Gonzalez, A.J., Martin de Pozuelo, R., 2010, June. A semantic context-aware network architecture. In 2010 Future Network & Mobile Summit (pp. 1-9). IEEE.

[Sanchez-Loro et al., 2011] Sanchez-Loro, X., Ferrer, J.L., Gomez, C., Casademont, J. and Paradells, J., 2011. Can Future Internet be based on constrained networks design principles?. Computer Networks, 55(4), pp.893-909.

[Sancho-Asensio et al., 2014] Sancho-Asensio, A., Navarro, J., Arrieta-Salinas, I., Armendáriz-Íñigo, J.E., Jiménez-Ruano, V., Zaballos, A. and Golobardes, E., 2014. Improving data partition schemes in Smart Grids via clustering data streams. Expert Systems with Applications, 41(13), pp.5832-5842.

[Satyanarayanan, 2001] M., 2001. Pervasive computing: Vision and challenges. Personal Communications, IEEE, 8(4), pp.10-17.

[Schulzrinne et al., 2003] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V., 2003. RTP: A transport protocol for real-time applications (No. RFC 3550).

[Schulzrinne, 1998] Schulzrinne, H., 1998. Real time streaming protocol (RTSP). (No. RFC 2326).

[Schwerdel et al., 2009] Schwerdel, D., Siddiqui, A., Reuther, B. and Müller, P., 2009, August. Composition of self-descriptive protocols for future network architectures. In 2009 35th Euromicro Conference on Software Engineering and Advanced Applications (pp. 571-577). IEEE.

[Seeling et al., 2014] Seeling, P. and Reisslein, M., 2014. Video traffic characteristics of modern encoding standards: H. 264/AVC with SVC and MVC extensions and H. 265/HEVC. The Scientific World Journal, 2014.

[Selga et al., 2013] Selga, J.M., Zaballos, A. and Navarro, J., 2013. Solutions to the computer networking challenges of the distribution smart grid. IEEE communications letters, 17(3), pp.588-591.

[Selga et al., 2014] Selga, J.M., Corral, G., Zaballos, A., Martin de Pozuelo, R., 2014. Smart grid ICT research lines out of the European project INTEGRIS. Network Protocols and Algorithms, 6(2), pp.93-122.

[Shen et al., 2010] Shen, C., Zhang, H., Wang, H., Wang, J., Zhao, B., Yan, F., Yu, F., Zhang, L. and Xu, M., 2010. Research on trusted computing and its development. Science China Information Sciences, 53(3), pp.405-433.

[Shifrin et al., 2013] Shifrin, M., Atar, R. and Cidon, I., 2013, May. Optimal scheduling in the hybrid-cloud. In 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013) (pp. 51-59). IEEE.

[Sicari et al., 2015] Sicari, S., Rizzardi, A., Grieco, L.A. and Coen-Porisini, A., 2015. Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, pp.146-164.

[Siddiqui et al., 2012] Siddiqui, F., Zeadally, S., Alcaraz, C. and Galvao, S., 2012, July. Smart grid privacy: Issues and solutions. In 2012 21st International Conference on Computer Communications and Networks (ICCCN) (pp. 1-5). IEEE.

[SILO] Service Integration Control and Optimization (SILO). 2008. Project website. [ONLINE] Available at: http://www.nets-find.net/Funded/Silo.php. [Accessed 31st August 2016].

[Silva et al., 2005] Silva, R., Zhang, J. and Shanahan, J.G., 2005, August. Probabilistic workflow mining. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (pp. 275-284). ACM.

[Sirin et al., 2003] Sirin, E., Hendler, J. and Parsia, B., 2003, April. Semi-automatic composition of web services using semantic descriptions. In 1st Workshop on Web Services: Modeling, Architecture and Infrastructure (pp. 17-24).

[Sivashanmugam et al., 2005] Sivashanmugam, K., Miller, J.A., Sheth, A.P. and Verma, K., 2005. Framework for semantic web process composition. International Journal of Electronic Commerce, 9(2), pp.71-106.

[Skevik et al., 2008] Skevik, K.A., Siekkinen, M., Goebel, V. and Plagemann, T., 2008, September. Experiences with Application Development for Autonomic Networks. In IEEE

International Workshop on Modelling Autonomic Communications Environments (pp. 1-13). Springer Berlin Heidelberg.

[SMARTCITYMALAGA] Smart City Malaga project. 2012. A model of sustainable energy management for cities of the future. White Paper. [ONLINE] Available at: http://www.endesa.com/en/saladeprensa/centrodocumental/Publicaciones/Smartcity-Malaga_ENG.pdf [Accessed 31st August 2016].

[SMOOTHIT] SmoothIT. 2008. European Union's 7th Framework Program (FP7). Project website. [ONLINE] Available at: http://www.smoothit.org/. [Accessed 31st August 2016].

[Solé, 2009] Solé, O., 2009. Multimedia quality assessment. Universitat Politecnica de Catalunya. MSc. Thesis.

[Song et al., 2000] Song, J., Pung, H.K. and Jacob, L., 2000. A multi-constrained distributed QoS routing algorithm. In Networks, (ICON 2000). Proceedings. IEEE International Conference on (pp. 165-171). IEEE.

[Spano et al., 2015] Spanò, E., Niccolini, L., Di Pascoli, S. and Iannacconeluca, G., 2015. Last-meter smart grid embedded in an Internet-of-Things platform. IEEE Transactions on smart grid, 6(1), pp.468-476.

[Srinivasan, 2009] Srinivasan, S.R., Lee, J.W., Liu, E., Kester, M., Schulzrinne, H., Hilt, V., Seetharaman, S. and Khan, A., 2009, December. NetServ: dynamically deploying in-network services. In Proceedings of the 2009 workshop on Re-architecting the internet (pp. 37-42). ACM.

[Srivastava, 2005] Srivastava, V. and Motani, M., 2005. Cross-layer design: a survey and the road ahead. IEEE Communications Magazine, 43(12), pp.112-119.

[Steinmetz et al., 2015] Steinmetz, R., Holloway, M., Koldehofe, B., Richerzhagen, B. and Richerzhagen, N., 2015. Towards Future Internet Communications–Role of Scalable Adaptive Mechanisms. ACADEMIA EUROPAEA, p.59.

[Stiller, 2009] Stiller, B., 2009. Telecommunication economics–overview of the field, recommendations, and perspectives. Computer Science-Research and Development, 23(1), pp.35-43.

[Su et al., 2003] Su, S.Y., Meng, J., Krithivasan, R., Degwekar, S. and Helal, S., 2003. Dynamic inter-enterprise workflow management in a constraint-based e-service infrastructure. Electronic commerce research, 3(1-2), pp.9-24.

[SUIT] SUIT: Scalable, Ultra-fast and Interoperable Interactive Television. 2006. European Union's 6th Framework Program (FP6). Project website. [ONLINE] Available at: http://suit.av.it.pt/. [Accessed 31st August 2016].

[Srivastava et al., 2005] Srivastava, V. and Motani, M., 2005. Cross-layer design: a survey and the road ahead. IEEE Communications Magazine, 43(12), pp.112-119.

[Sziget et al. 2013] Szigeti, T., Hattingh, C., Barton, R. and Briley Jr, K., 2013. End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks. Cisco Press.

[Tapper et al., 2013] Tapper, J., Labriola, M. 2013. Streaming video over HTML. Digital Primates website. [ONLINE] Available at: http://www.digitalprimates.net/ [Accessed 22nd September 2016].

[TARIFA] TARIFA – "The Atomic Redesign of the Internet Future Architecture". 2010. Project Website. [ONLINE] Available at: http://www.i2cat.net/en/projects/tarifa. [Accessed 31st August 2016].

[Tarjan, 1972] Tarjan, R., 1972. Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2), pp.146-160.

[TEMPOIQ] TempoIQ. Official Website. [ONLINE] Available at: https://www.tempoiq.com. [Accessed 31st August 2016].

[Texas Instruments, 2011] Texas Instruments. 2011. True System-on-Chip solution for 2.4 GHz IEEE 802.15. 4/ZigBee, [ONLINE] Available at: http://www.ti.com/lit/ds/symlink/ cc2430.pdf. [Accessed 31st August 2016].

[Tosic et al., 2002] Tosic, V., Patel, K. and Pagurek, B., 2002. WSOL—web service offerings language. In Web services, e-business, and the semantic web (pp. 57-67). Springer Berlin Heidelberg.

[Touch et al., 2006] Touch, J., Wang, Y.S. and Pingali, V., 2006. A recursive network architecture. ISI, Tech. Rep, (2006-626).

[Touch et al., 2008] Touch, J.D. and Pingali, V.K., 2008, August. The RNA metaprotocol. In 2008 Proceedings of 17th International Conference on Computer Communications and Networks (pp. 1-6). IEEE.

[Touch et al., 2009] Touch, J. and Perlman, R., 2009. Transparent interconnection of lots of links (TRILL): Problem and applicability statement (No. RFC 5556).

[Tracey et al., 2013] Tracey, D. and Sreenan, C., 2013, May. A holistic architecture for the internet of things, sensing services and big data. In Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on (pp. 546-553). IEEE.

[Trossen et al., 2012] Trossen, D. and Parisis, G., 2012. Designing and realizing an information-centric internet. IEEE Communications Magazine, 50(7), pp.60-67.

[UPA, 2008] UPA (2008). UPA Smartgrid Market Requirements. [ONLINE] Available at: http://www.upaplc.org/page_viewer.asp?page=Current+Specifications&pid=19 [Accessed 31st August 2016].

[US, 2010] U.S Department of Energy. 2010. Communications requirements of smart grid technologies. [ONLINE] Available at: http://energy.gov/gc/downloads/communications-requirements-smart-grid-technologies. [Accessed 31st August 2016].

[Van der Aalst et al., 2003] Van der Aalst, W.M., Dumas, M. and ter Hofstede, A.H., 2003, September. Web service composition languages: old wine in New bottles?. In Euromicro Conference, 2003. Proceedings. 29th (pp. 298-305). IEEE.

[Van der Meer et al., 2003] Van der Meer, D., Navathe, S.B., Datta, A., Dutta, K., Thomas, H. and Ramamritham, K., 2003, June. FUSION: A system allowing dynamic web service composition and automatic execution. In E-Commerce, 2003. CEC 2003. IEEE International Conference on (pp. 399-404). IEEE.

[Vellala et al., 2007] M. Vellala, A. Wang, G. Rouskas, R. D. I. Baldine, and D. Stevenson,"A composition algorithm for the silo cross-layer optimization service architecture," Mumbai, India, December 2007.

[Vernet et al., 2015] Vernet, D., Zaballos, A., Martin de Pozuelo, R. and Caballero, V., 2015. High Performance Web of Things Architecture for the Smart Grid Domain. International Journal of Distributed Sensor Networks, vol. 2015, Article ID 347413, 13 pages. doi:10.1155/2015/347413.

[Vinoski, 2006] Vinoski, S., 2006. Advanced message queuing protocol. IEEE Internet Computing, 10(6), p.87.

[Virvilis et al., 2013] Virvilis, N. and Gritzalis, D., 2013, September. The big four-what we did wrong in advanced persistent threat detection?. In Availability, Reliability and Security (ARES), 2013 Eighth International Conference on (pp. 248-254). IEEE.

[Vogel et al., 2009] Vogel, J., Xu, L., Schnabel, F., Krummenacher, R., Abels, S., Gonzalez, R., Mos A., Mehandjiev, N., Cekov, L., Ripa, G. and others, "State of the Art Report and Requirements for Service Construction. D6. 1.1, SOA4All EU project FP7-ICT-2008-215219," 2009.

[Völker et al., 2008] Völker, L., Martin, D., El Khayat, I., Werle, C. and Zitterbart, M., 2008, November. An architecture for concurrent future networks. In 2nd GI/ITG KuVS Workshop on The Future Internet. Karlsruhe, Germany: GI/ITG Kommunikation und Verteilte Systeme.

[Völker et al., 2009] Völker, L., Martin, D., Rohrberg, T., Backhaus, H., Baumung, P., Wippel, H. and Zitterbart, M., 2009, May. Design process and development tools for concurrent future networks. In 3rd GI/ITG KuVS Workshop on The Future Internet, GI/ITG Kommunikation und VerteilteSysteme, Munich, Germany.

[Vukovic et al., 2007] Vukovic, M. and Robinson, P., 2007. Context aware service composition. University of Cambridge, Cambridge, UK, PhD Thesis.

[Wang et al., 2004] Wang, X.H., Zhang, D.Q., Gu, T. and Pung, H.K., 2004, March. Ontology based context modeling and reasoning using OWL. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on (pp. 18-22). Ieee.

[Wang et al., 2011] Wang, W., Xu, Y. and Khanna, M., 2011. A survey on the communication architectures in smart grid. Computer Networks, 55(15), pp.3604-3629.

[Wang et al., 2014] Wang, Y., Matta, I., Esposito, F. and Day, J., 2014. Introducing ProtoRINA: a prototype for programming recursive-networking policies. ACM SIGCOMM Computer Communication Review, 44(3), pp.129-131.

[Wodjak, 2003] Wojdak, W., 2003. Rapid Spanning Tree Protocol: A new solution from an old technology. Reprinted from CompactPCI Systems.

[Wu et al., 2006] Wu, D., Sirin, E., Hendler, J., Nau, D. and Parsia, B., 2006. Automatic web services composition using shop2. MARYLAND UNIV COLLEGE PARK DEPT OF COMPUTER SCIENCE.

[Xue et al., 2011] Xue, M., Song, F., Qiu, F., Wan, M., Zhang, S. and Zhang, H., 2011, September. Redesigning Transport Layer Architecture for Future Internet. In Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on (pp. 1-5). IEEE.

[Xylomenos et al., 2001] Xylomenos, G., Polyzos, G.C., Mahonen, P. and Saaranen, M., 2001. TCP performance issues over wireless links. IEEE communications magazine, 39(4), pp.52-58.

[Yan et al., 2012] Yan, Y., Qian, Y., Sharif, H. and Tipper, D., 2012. A survey on cyber security for smart grid communications. IEEE Communications Surveys & Tutorials, 14(4), pp.998-1010.

[Yan et al., 2013] Yan, Y., Qian, Y., Sharif, H. and Tipper, D., 2013. A survey on smart grid communication infrastructures: Motivations, requirements and challenges. IEEE communications surveys & tutorials, 15(1), pp.5-20.

[Yang et al., 2003] Yang, G., Kifer, M. and Zhao, C., 2003. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE (pp. 671-688). Springer Berlin Heidelberg.

[Yang et al., 2005] Yang, S.W. and Shi, M.L., 2005. Model for web service discovery with QoS constraints. *Jisuanji Xuebao(Chin. J. Comput.)*, *28*(4), pp.589-594.

[Yukun et al., 2013] Yukun, N., Xiaobin, T., Shi, C., Haifeng, W., Kai, Y. and Zhiyong, B., 2013, July. A security privacy protection scheme for data collection of smart meters based on homomorphic encryption. In EUROCON, 2013 IEEE (pp. 1401-1405). IEEE.

[Yusta et al., 2011] Yusta, J.M., Correa, G.J. and Lacal-Arántegui, R., 2011. Methodologies and applications for critical infrastructure protection: State-of-the-art. Energy Policy, 39(10), pp.6100-6119.

[Zaballos et al., 2011] Zaballos, A., Vallejo, A. and Selga, J.M., 2011. Heterogeneous communication architecture for the smart grid. IEEE Network, 25(5), pp.30-37.

[Zaballos et al., 2013] Zaballos, A., Vernet, D. and Selga, J.M., 2013. A genetic QoS-aware routing protocol for the smart electricity networks. International Journal of Distributed Sensor Networks, 2013.

[Zdun et al., 2006] Zdun, U., Hentrich, C. and Van Der Aalst, W.M., 2006. A survey of patterns for service-oriented architectures. International journal of Internet protocol technology, 1(3), pp.132-143.].

[Zhang et al., 2014] Zhang, H., Jiang, G., Yoshihira, K. and Chen, H., 2014. Proactive workload management in hybrid cloud computing. IEEE Transactions on Network and Service Management, 11(1), pp.90-100.

[Zhou et al., 2006] Zhou, R. and Hansen, E.A., 2006. Breadth-first heuristic search. Artificial Intelligence, 170(4), pp.385-408.

[Zhu et al., 2005] Zhu, F., Mutka, M.W. and Ni, L.M., 2005. Service discovery in pervasive computing environments. IEEE Pervasive computing, 4(4), pp.81-90.

[Zhu et al., 2011] Zhu, W., Luo, C., Wang, J. and Li, S., 2011. Multimedia cloud computing. IEEE Signal Processing Magazine, 28(3), pp.59-69.

UNIVERSITAT RAMON LLULL

Esta Tesis Doctoral ha sido defendida el día ____ de________________ de ____ en el

Centro ____

de la Universitat Ramon Llull

delante del Tribunal formado por los Doctores abajo firmantes, habiendo obtenido la calificación:

Presidente/a

____

Vocal

____

Secretario/aria

____

Doctorando

____