FLAMINGO

European Seventh Framework Network of Excellence

http://www.fp7-flamingo.eu/

WP6 — Automated Configuration and Repair

Deliverable D6.1 — First Year Report on Automated Configuration and Repair

University of Twente, The Netherlands (UT)
Institut National de Recherche en Informatique et Automatique, France (INRIA)
University of Zurich, Switzerland (UZH)
Jacobs University Bremen, Germany (JUB)
Universität der Bundeswehr München, Germany (UniBwM)
University Politecnicà de Catalunia, Spain (UPC)
iMinds, Belgium (iMinds)
University College London, United Kingdom (UCL)

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number ICT-FP7 318488

Document Control

Title:	D6.1 — First Year Report on Automated Configuration and Repair
Type:	Public
Editor(s):	Gabi Dreo Rodosek
E-mail:	gabi.dreo@unibw.de
Doc ID:	D6.1
Delivery Date:	31.10.2013
Author(s):	Anthea Mayzaud, Anuj Sehgal, Björn Stelte, Gabi Dreo, Christos Tsiaras, Daniel Dönni, Daphne Tuncer, Abdelkader Lahmadi, Marinos Charalambides, Mario Flores, Jeroen Famaey, Mario Golling, Maxim Claeys, Niels Bouten, Nikolay Melnikov, Radhika Garg, Rashid Mijumbi, Ricardo Schmidt, Jair Santanna, Rick Hofstede, Sebastian Seeber, Steven Latré, Corinna Schmitt, Guilherme Sperb Machado, Jair Santanna

For more information, please contact:

Dr. Aiko Pras
Design and Analysis of Communication Systems
University of Twente
P.O. BOX 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893778
Fax: +31-53-4894524
E-mail: <a.pras@utwente.nl>

Legal Notices

Executive Summary

The management of the Future Internet poses new challenges to the management of billions connected devices. Management, similar to IT security, can not be an afterthought but needs to be part of the functionality of managed objects (i.e., *Management-by-Design*). Due to the enormous amount of devices, it is further necessary to perform management actions in an automated way. Deliverable D6.1 reports the conception, implementation and achievements conducted within WP6 in its first year of activity. Centered in the very challenging topic of network and service management automation, this work package aims at going beyond the state-of-the-art in the embedment of management principles at the design stage of networks with particular focus on the Future Internet needs.

With respect to the S.M.A.R.T. (Specific, Measurable, Achievable, Relevant, Timely) objectives (Section B.1.1.5 of the Description of Work) we claim the fully achievement of our first year targets for WP6 regarding: (i) the integration of PhD students and (ii) the production of scientific publications. In fact, at the date of this deliverable WP6 has more than two fully integrated PhD students (i.e., which means students jointly supervised by professors of at least two different participating institutions and financially paid by FLAMINGO). This is what was stated in the Description of Work (DoW) of the project. In addition, due to the tight collaboration of this work package with others like WP1 and WP5, we can claim to have much more students that, although not financially paid by FLAMINGO, are jointly supervised and are directly contributing to the work package specific objectives.

As a scientific output, we can report a total of 37 papers, jointly reported by WP5 and WP6, that have been published, as well as 7 submitted. This scientific output is exceeding the objective initially set in the DoW that was 20 papers. In that respect we have to mention that this outstanding results have been made possible due to the close collaboration between WP5 and WP6, and the involvement of further PhD collaborations not payed by FLAMINGO. We point to Deliverable D8.1 for details.

The work package specific objectives center around the following three tasks, namely (i) to develop innovative architectural approaches for automated configuration and repair (Task 6.1), (ii) to analyze and develop enablers for these new architectures (Task 6.2) and (iii) to analyze the applicability of the developed approaches to selected application domains (Task 6.3). Key achievements of WP6 of this year, as specified in the DoW and as documented in D6.1, are summarized below:

Task 6.1: Architectures. An *inventory* of architectures and approaches in that field has been set up. Furthermore, to have the possibility to identify the characteristics of each approach and compare approaches, a *taxonomy* (resp. classification scheme) has been developed, and the approaches identified so far were evaluated according to the scheme. With this, we have the possibility to build up the inventory in a systematic way and qualitatively compare selected as well as further architectures like any one conceived in this project.

Task 6.2: Enablers. A first blueprint of metrics for a quantitative evaluation of automation architectures and approaches has been specified in the first year. The goal of the metrics is to compare architectures in a quantitative way, and with get an indication of the necessary enablers. An example of such a metric is the learning index.

Task 6.3: Application Domains. A detailed description of three application areas, namely (i) wireless sensors networks, (ii) cloud-based services and (iii) content-aware routing was in the focus of the first year. The objective behind this effort is to identify the specifics of the domains and to validate the applicability of the developed automation approaches.

Furthermore, the deliverable presents a first draft of a FLAMINGO Inter-Domain Automation Architecture in the area of Intrusion Detection Systems although this is an objective for the second year.

To summarize, we claim that the work package specific objectives in the first year have been fully achieved.

Contents

1 Introduction

A primary objective of the Management of the Future Internet is the automation of tasks and processes, especially due to the trend towards a myriad of networked devices with a broad range of capabilities (from a simple dumb sensor to the smart space, from isolated clouds to Inter-Clouds) and the dramatically increased complexity of communication environments.

The aim of Deliverable D6.1 is to report on the achievements of WP6 in the first year in this research area by focusing on the S.M.A.R.T. (Specific, Measurable, Achievable, Relevant, Timely) as well as WP6-specific objectives.

The first S.M.A.R.T. objective is the *integration of PhD students*. In adherence to the Description of Work (DoW), at least two fully integrated PhD students are active in WP6. For a detailed list of the fully integrated PhD students, we refer to Deliverable D8.1. In addition, many PhD collaborations within the consortium have started during this first year with students not payed by FLAMINGO. Information regarding these topics can be found in Section 4.

The second S.M.A.R.T. objective concerns the *scientific output* of the project. In the first year, WP5 and WP6 have published 37 papers, both at major conferences and in journals. In addition, seven other papers are currently under review. We report this summarized number in both deliverables due to the tight research integration of WP5 and WP6 that is manifested also in the joint publications. For a detailed list of the FLAMINGO published and submitted papers, we refer to Deliverable D8.1.

The work package activities are structured around three focal points: (i) the development of architectures, (ii) the development of enablers such as knowledge description and discovery, learning techniques, algorithms, and (iii) application domains to validate the developed concepts.

Sections 3.3, 3.4 and 3.5 document the achieved research objectives of the first year:

1. *Overview of architectures and approaches for automation of configuration and repair actions.*

   Building an inventory of architectures and approaches for automation without a taxonomy, resp. classification criteria, is not very helpful. After having identified several architectures, we started with the specification of a taxonomy resp. evaluation criteria to be able to highlight characteristics of each architecture and to compare these approaches in a qualitative manner. The outcome of this activity is documented in Section 3.2. Several architectures have been identified and classified according to the proposed taxonomy, as reported in Section 3.3. With this, we have exceeded the planned objective, and have achieved a systematic basis for evaluating other architectures as well.

2. *Specification of a first blueprint of metrics for a quantitative comparison of approaches.*

   The outcome of this research activity is a first blueprint of metrics (e.g., learning index) for a quantitative comparison of architectures and approaches, as documented in Section 3.4. Furthermore, the metrics have been put in relation with the taxonomy.

3. *Description of application domains*

   The outcome of this task is a description of the selected application domains such as (i) wireless sensor networks, (i) cloud-based services and (iii) content-aware routing. These application domains, as specfied in the DoW, have been selected due to their diversity with respect to the automation requirements, and provide a basis to validate the developed approaches. The outcomes of this effort are documented in Section 3.5.

Since PhD collaborations form the basis of the research work, Section 4 highlights the PhD contributions, and visualizes the contirbutions with respect to the work package specific objectives.

Section 5 describes a first draft of a FLAMINGO Inter-Domain Automation Architecture in the area of Intrusion Detection Systems. This architecture extends the FLAMINGO Monitoring Architecture, as described in D5.1, with aspects of automation, and is planned according to the DoW for the second year.

Section 6 concludes the deliverable.

2 Objectives and Activities

This section presents an overview of the S.M.A.R.T. objectives for WP6. For each S.M.A.R.T. objective, we indicate how it has been achieved in the first year of the project. For the WP6-specific objectives, we summarize the activities that have taken place among the consortium members.

2.1 S.M.A.R.T. Objectives

To meet the S.M.A.R.T. objectives, WP6 has been active in the following aspects.

- **Integration of PhD students** – The DoW (Section B.1.1.5) states that "after 9 months each research WP will have identified at least two fully integrated PhD students, which means that these students will be jointly supervised and financially paid by FLAMINGO". Since the beginning of the project, seven PhD students have joined FLAMINGO as fully integrated PhD students. These students, their affiliation and the co-supervising institution are listed in D8.1. For the FLAMINGO project, collaboration are a basis of research. Therefore, PhD students are not working in isolation, but have extended collaborations with other institutions. For this reason, there is not a one-to-one match between a PhD student and a single WP. Details on the integration of PhD students, the PhD students active in the context of WP5 and WP6 and their collaborations within the consortium can be found in Section 4.
- **Scientific Output** – The Description of Work (Section B.1.1.5) states that "after 18 month at least 20 scientific papers will be submitted / published". In the first year, WP6 and WP5 have published 37 papers, both at major conferences and in journals. Furthermore, seven other papers are currently under review and seven additional publications like posters or presentations were published. A complete list of publications, submissions under review and other scientific output can be found in D8.1.

Tables 1 and 2 show the FLAMINGO publications in collaborations with other EU projects and institutions as well as publications with multiple FLAMINGO partners. Again, WP6 and WP5 report here the same output due to the tight collaboration.

Table 1: FLAMINGO publications in collaboration with other EU projects and institutions.

Authors	**Title**	**Venue**	**EU project/ institution**
I. Drago, E. Bocchi, M. Mellia, H. Slatman, A. Pras	Benchmarking Personal Cloud Storage [1]	ACM/SIGCOMM IMC 2013	mPlane[1]
M. Barrére, R. Badonnel, O. Festor	Vulnerability Assessment in Autonomic Networks and Services: A Survey [2]	IEEE Surveys & Tutorials	UniverSelf[2]
A. Lareida, T. Bocek, S. Golaszewski, C. Lüthold, M. Weber.	Box2Box – A P2P-based File-Sharing and Synchronization Application [3]	P2P 2013	SmartenIT[3]
G. Sperb Machado, T. Bocek, M. Ammann, B. Stiller	A Cloud Storage Overlay to Aggregate Heterogeneous Cloud Services [4]	LCN 2013	SmartenIT[3]
P. Poullie, B. Stiller	Fair Allocation of Multiple Resources Using a Non-monetary Allocation Mechanism [5]	AIMS 2013	SmartenIT[3]
C. Schmitt, B. Stiller, T. Kothmayr, W. Hu.	DTLS-based Security with two-way Authentication for IoT [6]	IETF	SmartenIT[3]
O. Festor, A. Lahmadi, R. Hofstede, A. Pras	Information Elements for IPFIX Metering Process Location (Internet Draft) [7]	IETF	EIT ICT Labs[4]
D. Tuncer, M. Charalambides, R. Landa, G. Pavlou	More Control Over Network Resources: an ISP Caching Perspective [8]	CNSM 2013	Fusion[5]

[1] http://www.ict-mplane.eu/
[2] http://www.univerself-project.eu/
[3] http://www.smartenit.eu/
[4] http://www.eitictlabs.eu/
[5] http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/fusion-factsheet.pdf

Table 2: Publications authored by multiple FLAMINGO partners.

Authors	**Title**	**Venue**	**FLAMINGO partners**
S. Seeber, A. Sehgal, B. Stelte, G. Dreo Rodosek, J. Schönwälder	Trust Computing Architecture for RPL in Cyber Physical Systems [9]	CNSM 2013	UniBwM, JUB
R. de O. Schmidt, N. Melnikov, R. Sadre, J. Schönwälder, A. Pras	Linking Network Usage Patterns to Traffic Gaussianity Fit [10]	PAM 2014 (under review)	UT, JUB
R. Mijumbi, J. Serrat, J.L. Gorricho, M. Claeys, F. de Turck, S. Latré	Design and Evaluation of Learning Algorithms for Dynamic Resource Management in Virtual Networks [11]	NOMS 2014 (under review)	UPC, iMinds
A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, J. Schönwälder	Mitigating DODAG Inconsistency Attacks in RPL Networks [12]	IPSN 2014 (under review)	INRIA, JUB

2.2 Work Package Specific Objectives

Inside FLAMINGO each work package has its own defined objectives. This section reports on the WP6 specific objectives and the achievements during the first year. Since the PhD collaborations mainly contribute to the work package specific objectives, we defined acronyms to refer to them (see also Section 4.1). Acronyms are build by combining the abbreviations of the involved institutions and the subject of the collaboration. For example, if there is a collaboration between University of Twente (UT) and Universität der Bundeswehr München (UniBwM) concerning the subject Intrusion Detection Systems (IDS) the acronym used is **UT-UniBwM-IDS**. An overview of the established collaboration is depicted in Figure 20.

2.2.1 Ongoing Objectives

OBJECTIVE 1 - To integrate European research in the area of automated configuration and repair: In cooperation with WP3, WP4 and WP5, WP6 has taken part in several activities in the area of automated configuration and repair. WP6 members have been actively involved in the organization of the Dagstuhl Seminar Global Measurement Framework[6]; in the tutorials for Large-scale Measurement Platforms (AIMS 2013) and Management of the Internet of Things (IM 2013); and finally in the organization of the 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2013) as well as the 9th Conference on Communications Networks Service Management (CNSM 2013). The most prominent demonstration of the integrated European Research, however, are the tight and extensive PhD collaborations betweeen institution.

OBJECTIVE 3 - To develop an inventory of approaches for automated configuration and repair: Section 3.3 reports on the inventory of approaches for automated configuration and repair, including the developed taxonomy. In addition, several collaborations developed application domain specific inventories which provide a further basis for the development of inter-domain automation architectures. The first collaboration activity **UT-UniBwM-IDS** in this objective provided an evaluation of the state-of-the-art IDS message exchange protocols [13]. The next collaboration **INRIA-JUB-RPL** developed an inventory of inconsistency attacks in RPL networks. Also with a security background the collaboration **UT-UZH-DoS** identified research aspects of adaptation mechanisms to prevent DDoS attacks. Furthermore, a cloud storage overlay to aggregate heterogeneous cloud services was investigated in [4].

OBJECTIVE 4 - To specify guidelines about the applicability of approaches for automated configuration and repair to specific application domains: The description in Section 3.5 gives an overview of the application domains which are investigated in FLAMINGO. The established collaborations lead to contributions to this objective as well. Cloud-based services were investigated during the collaboration **UT-UniBwM-IDS**. Furthermore, Inter-Clouds with the focus to the management of Inter-Clouds have been analyzed in detail in [14, 15]. The environment of content-aware-routing was addressed in the collaboration **UCL-iMinds-Cache** where requirements for the applicability of automated configuration and repair approaches in this application domain have been analyzed [16].

[6] http://www.dagstuhl.de/13472/

OBJECTIVE 5 - To develop new architectures for automated configuration and repair approaches across administrative boundaries: New architectures for automated configuration and repair have been addressed in the collaboration **UT-UniBwM-IDS** by focussing on the cloud-based services environment in conjunction with security issues [14, 15, 17]. In addition, in the collaboration **iMinds-UPC-NetVirt** new approaches concerning virtualization of networks for automated configuration and repair are investigated [18].

OBJECTIVE 6 - To develop information models, algorithms, learning techniques and knowledge description approaches as enablers for automated configuration and repair: The development of information models as an enabler for automated configuration and repair was approached in various collaborations. In the collaboration **UT-UniBwM-IDS** enablers (e.g., learning algorithms, knowledge representation techniques) with respect to IDS have been analyzed and developed [17]. The collaboration **UniBwM-JUB-RPL** also developed enablers related to the security area [19, 20]. In the collaboration **iMinds-UPC-NetVirt** the enablers are more focused on network virtualization concepts. The collaboration **iMinds-UT-QoS** analyzed enablers in the area of Quality of Experience and Quality of Service [21, 22, 23]. Mechanisms for estimating whether the traffic is Gaussian or not are investigated in the collaboration **JUB-UT-Pattern** [10].

OBJECTIVE 7 - To evaluate automated configuration and repair approaches as being part of the autonomic control loops: By extending previous work on adaptive resource management and enhancing the energy-awareness control loop the collaboration **UCL-UT-Man** contributes to this objective [16]. In addition, the collaboration **UCL-iMinds-Cache** analyzes cache management in Content Delivery Networks (CDN) where the management is implemented in a control loop [24].

OBJECTIVE 8 - To apply policy-based and semantic-based approaches for automated configuration and repair: Policy-based and semantic-based approaches are addressed by the collaboration **UCL-iMinds-Cache** where the ISPs in CDNs are analyzed with respect to approaches for a cache management strategy [24]. The cache management approach (UCL-iMinds-Cache) is investigating various strategies to decide on the placement of contents, which are based on the geographic location of requests and content popularity. According to these characteristics, relevant policies are used to guide the content placement decisions and efficiently realize the appropriate strategy.

OBJECTIVE 9 - To propose and study automated configuration and repair in the context of the management of clouds (especially Inter-Clouds): In respect to the collaboration of **UT-UniBwM-IDS** the automated configuration and repair aspect in clouds has been analzyed in an Inter-Cloud environment [14, 15]. VoIP security in the cloud is also a topic that has been addressed in [25]. Vulnerability management in clouds is planned to be analyze in the collaboration between INRIA and UniBwM in the next year.

OBJECTIVE 10 - To apply the developed approaches to several application domains such as of (i) wireless sensor networks (ii) cloud-based services and (iii) content-aware networking: An analysis of the application domain of wireless sensor networks has been investigated in the collaboration **UniBwM-JUB-RPL**. In addition, the collaboration **INRIA-JUB-RPL** is developing approaches for RPL that will be tested on the IEEE 802.15.4 + 6LoWPAN platform, which is expected to form the basis of the Internet of Things. Similarly, all the work out of this collaboration is

being currently tested on the TelosB network node, which is a resource constrained device, used commonly in the Internet of Things (IoT) and Wireless Sensor Networks (WSN) areas. The cloud-based services are addressed in the collaboration **UT-UniBwM-IDS** by the development of respective architectures [14, 15]. New cache management approaches, that allow an ISP to automatically allocate its caching capacity to CPs or CDNs, and to generate content placement configurations, are developed in the collaboration **UCL-iMinds-Cache** [24].

2.2.2 Open Objectives

OBJECTIVE 2 - To create and maintain articles within Wikipedia and other online systems in this area: WP6 is currently identifying a list of possible topics which are either currently not covered in Wikipedia pages or that can be improved considering the expertise of the project members. Therefore, relevant articles concerning these topics has to generated first. This aspect contributes to WP2 and will be addressed in more detail in the second year.

2.3 Tasks and Objectives Mapping

Table 3 summarizes the status of the S.M.A.R.T. objectives relative to WP6 (Section 2.1) and the WP6-specific objectives (Section 2.2). For each of the addressed objectives, Table 3 indicates if the objective has been achieved (S.M.A.R.T. objectives) or if there are WP activities that are contributing to the objective (WP6-specific objectives). For the WP6-specific objectives, Table 3 shows to which of the Tasks in the DoW the objective is contributing to. Finally, the table acts as a guide for the reader to locate the sections of this deliverable that provide additional information on a specific objective.

Table 3: Mapping of objectives and tasks.

Objective	Task 6.1	Task 6.2	Task 6.3	Status	Additional Material
S.M.A.R.T. Objective 1				Achieved	Section 4, D8.1
S.M.A.R.T. Objective 2				Achieved	D8.1
WP Objective 1				Ongoing	Section 4
WP Objective 2				Open	
WP Objective 3	X			Ongoing	Section 3.3
WP Objective 4			X	Ongoing	Section 3.5
WP Objective 5	X			Ongoing	Section 5
WP Objective 6		X		Ongoing	Section 3.4
WP Objective 7	X	X		Ongoing	Section 3.4
WP Objective 8	X		X	Ongoing	Section 3.5
WP Objective 9	X		X	Ongoing	Section 3.5
WP Objective 10			X	Ongoing	Section 3.5

3 Automated Configuration and Repair

Many questions on the Management of the Future Internet center around the automation possibilities of management actions. Billions of connected devices, ubiquitous environments, context- and location-based personalized services, hunderds of millions of users are parameters demanding for new and innovative approaches with respect to an inter-domain management and automation.

To face the topic of automated configuration and repair, we first discuss the terminology in Section 3.1, and clarify terms such as autonomic vs. self-management vs. automated. Afterwards, we describe the taxonomy (i.e. evaluation criteria) that is used for the comparison of architectures in Section 3.2. Section 3.3 provides an overview of architectures and approaches for automation, including their qualitative comparison. The first blueprint of metrics for the quantitative comparison is presented in Section 3.4. Section 3.5 investigates the targeted application domains in FLAMINGO, where each application domain is described with a common description and a collaboration activity part. Existing surveys (e.g. [26]) provide mostly only an overview of (single) approaches without a systematically comparison. The developed taxonomy enables to highlight the specifics of each approach, and a qualitative comparison. Futhermore, this allows us to develop guidelines when to use what approach. The outcome of this analysis forms the basis to develop an inter-domain architecture for automation with efficient semantic information models for enabling cross-boundary automated configuration and repair.

3.1 Terminology

The variety of used terms with respect to the automation architectures and approaches requires some clarifications. The main issue is certainly the difference between an automated and an autonomous process. An automated process simply replaces routine manual processes with software or hardware ones that follow a step-by-step sequence that may still include human participation. Autonomous processes, on the other hand, have the more ambitious goal of emulating human processes rather than simply replacing them. Several initiatives centered around this topic.

The goal of Autonomic Computing is to develop self-managing computing systems. These systems should manage themselves given high-level goals by the administrators [27]. The idea of such systems is inspired by the nervous system of the human body. For example, it checks and steers temperature without any conscious effort [28]. Some self-properties for autonomic systems are existing and defined by IBM (see Table 4). These are achieved through key properties of awareness, adaptivity, automaticity. They consist of reactive and proactive approaches to several areas of computing systems. The autonomic elements cooperate with each other to provide a convergent autonomic management. Such autonomic elements are composed of different components:

- Monitoring Component,
- Analyzing Component,
- Planning Component,
- Executing Component.

These components were first mentioned in the MAPE-K model and enable the self-adaptation according to environment changes. In 2001 IBM started an autonomic computing initiative, which has lead to a number of research initiatives that developed achievements and trends in different research areas, furthermore applicable to autonomic computing. The main building blocks of an

autonomic component are investigated in the surveys from [29, 30]. Taken this into account a more precise clarification of used terms is as follows:

- **Automaticity** / **Automated** describes the ability to perform one or more tasks without manual or external intervention. In this case there is no performance optimization issue included to reach better performance goals. A queue scheduling mechanism which carries out the execution automatically referring to the types of packets is an example [31].
- **Autonomicity** / **Autonomous** / **Self*** implements self-managing using a set of given high-level objectives from administrators [28]. These objectives define for a system the goals to be reached, and in what way. During this process, the system itself is able to monitor its own performance, analzye and adapt itself in the respective manner. Additionally, it optimizes its use of resources and overcomes current events.

The aforementioned self-properties in Table 4 go back to properties of software or hardware agents which were identified by Wooldridge and Jennings [32]. These are described below:

- **Autonomy:** Means the operation of agents without direct interventions of humans or others, and having some kind of control over the actions and the internal state.
- **Social Ability:** Agents interact between themselves and in some cases with humans via some kind of agent-communication language.
- **Reactivity:** Describes the recognition of the agents and the corresponding reaction in a timely manner to changes that occur in this environment.
- **Proactiveness:** The actions of the agents are not only actions in response to their environment. Additionally, they are able to exhibit goal-directed behavior by taking initiatives.

Self-adaptive systems contain some elements of these properties for a specific time mainly for self-optimization. First research in this space was done in streaming media systems where codecs of the stream change with network bandwidth deviations [33, 34]. However, the autonomic research is identifying a system as autonomic if it exhibits more than one of the self-management properties described [35].

Table 4: Four properties of self-management in autonomic computing

Concept	Autonomic Computing
Self-configuration	This property refers to the ability of the system to configure and reconfigure itself according to high-level policies. It covers the ability of a new component and the existing system to configure, install and integrate when a new component is introduced. The new component is able to incorporate itself and the existing system to adapt itself to the presence. The end system is then able modify the behavior of the component and can use this component.
Self-optimization	To enable efficient operation of the system even in unpredictable environments the self-optimization property is necessary. The autonomic system will seek opportunities to make itself more efficient in performance and cost. To achieve this the system has to be aware of its own ideal performance, measurements of its current performance and strategies for improvements.
Self-healing	To discover and repair potential problems and ensuring the smoothly run of the system the self-healing property is required. This property is achievable by the prediction of problems and implementation of proactive actions to prevent failures or reduce the impact of the failure.
Self-protection	The self-protection property denotes the capability of the system to protect itself from compromising the achieving of the goals. It additionally involves the protection from inadvertent failures, intrusion tentative or malicious attacks.

3.2 Taxonomy

Architectures and approaches for automatic configuration and repair can be classified in many different ways (e.g. [36, 37, 38, 39, 40]). As Section 3.4 describes in detail, we have - on the basis of the available literature - pursued a taxonomy which takes into account requirements that have been analyzed with respect to the related work and the PhD collaborations, and adds also new aspects. Therefore, the architectures presented in Section 3.3 are classified according to this taxonomy:

- **Degree of Automation:** In order to be able to classify and compare different approaches and architectures with each other, for the sake of clarity, the **degree of automation** will be specified by the use of three values (i) *manual* refers to lack of automation, (ii) *automated*, as previously outlined, describes the ability to perform one or more tasks without manual or external intervention, and (iii) *autonomous/autonomic/self** addresses the four properties of self-management in autonomic computing as outlined in Table 4, and referred to as action in Figure 1.
- **Approach:** Proactivity or *proactive* behavior refers to anticipatory, change-oriented and self-initiated behavior in situations and involves acting in advance of a future situation, rather than just reacting. Unlike the *reactive* condition, it implies taking control and making things happen rather than just adjusting to a situation or waiting for something to happen.
- **Memory:** Approaches can also be classified based on the type of information stored and analyzed in order to make a decision. An approach is treated as *current state* when decisions are only made based upon information which is available at direct environment, whereas *recent developments* implies storing information for a certain period of time. In contrast to

this, an approach is classified as *history*, when extensive use of previously stored information is used, such as a long-term storage of data.

- **Degree of Hierarchy:** Decisions can either be taken on an equal base *none/peer* or with the use of superior/subordinate elements *Hierarchy*.

- **Time Horizon:** In contrast to *Memory*, Time Horizon does not take a look on how long information is stored. Instead, *Time Horizon* reflects the levels of control within an organization. *Strategic* management provides overall direction to the enterprise/an organization ("future vision of the business"), e.g. examining where the company is now, determining where it wants to go, and then determining how to get there; performing a situation analysis, self-evaluation and competitor analysis. Thus, *strategic* performs a broader future oriented view. The *tactical* level involves the actual steps needed to achieve that vision or strategy. Hence, the *tactical* level is centered around the idea of mid-term planing. Consequently, the *operational* layer performs the "day-to-day output" relative to schedules, specifications, and costs. Thus, *operational* considers short-term planing.

- **Fault Management, Configuration Management, Accounting Management, Performance Management, Security Management** (FCAPS): The classification is done by rating the degree of how much the FCAPS tasks, which are described in the following are achieved. *Fault Management* aims is to identify, isolate, resolve and log network faults that have been occurred. Here, isolating a fault means: In case of a trouble you have to find the common cause of the failure (root cause analysis). *Configuration Management* includes: (i) Collecting and storing configurations of network components, (ii) simplifying the configuration of a network component, (iii) detecting changes in the network configuration and (iv) configuring links or paths through the network or a part of the network (network sub). *Accounting Management* provides useful statistics on the use of network resources, so that costs can be settled or quotas can be controlled. *Performance Management* refers to collecting and analyzing performance data in order to monitor the stability of the network. Trends may indicate future problems with the capacity or the reliability of the network. *Security Management* is the identification of all types of vulnerabilities, weaknesses and risks. The aforementioned management tasks in the FCAPS model are not strictly separated. Nevertheless the FCAPS management tasks are the basis for further functions in upper layers. The FCAPS model is supported by various computer aided network management systems, but most of them include only a small part of it. There exist network management systems which include a couple of areas covered by FCAPS, but these are mainly specific to one single manufacturer.

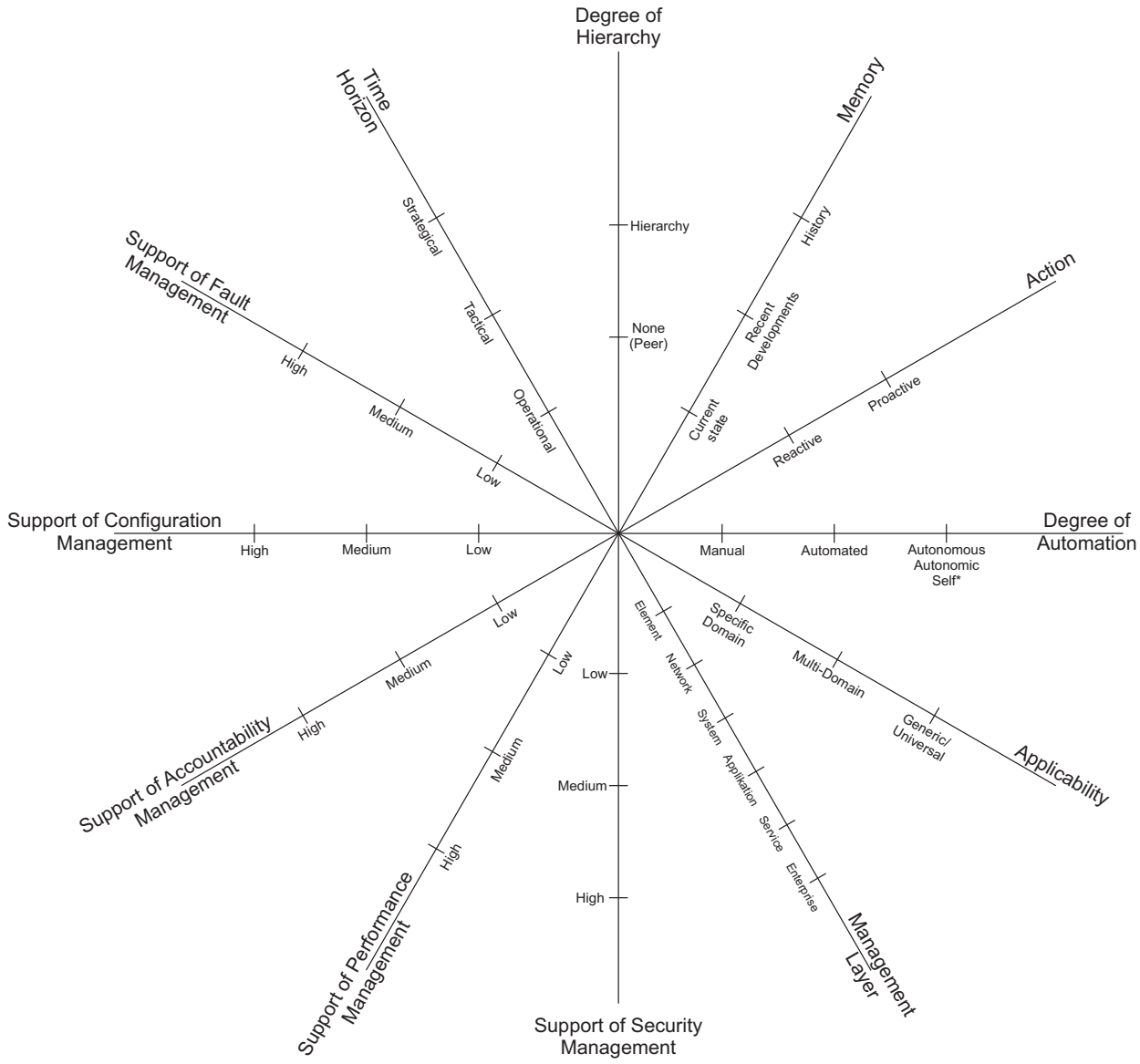


Figure 1: Taxonomy

3.3 Architectures on Automated Configuration and Repair

This section describes a subset of existing architectures in the scope of automated configuration and repair. The reason for chosing them was primarily in the diversity of the approaches. In the following these architectures are described in more detail and evaluated with respect to the introduced taxonomy (see Section 3.2).

3.3.1 ANA: Autonomic Network Architecture

The Autonomic Network Architecture (ANA) [41] is a framework and an execution environment for the development and testing of autonomic networks which is aimed at creating a more flexible and general network architecture. To this end, the authors introduce abstractions such as network compartments, which are used to model the network operation and allow for the coexistence of

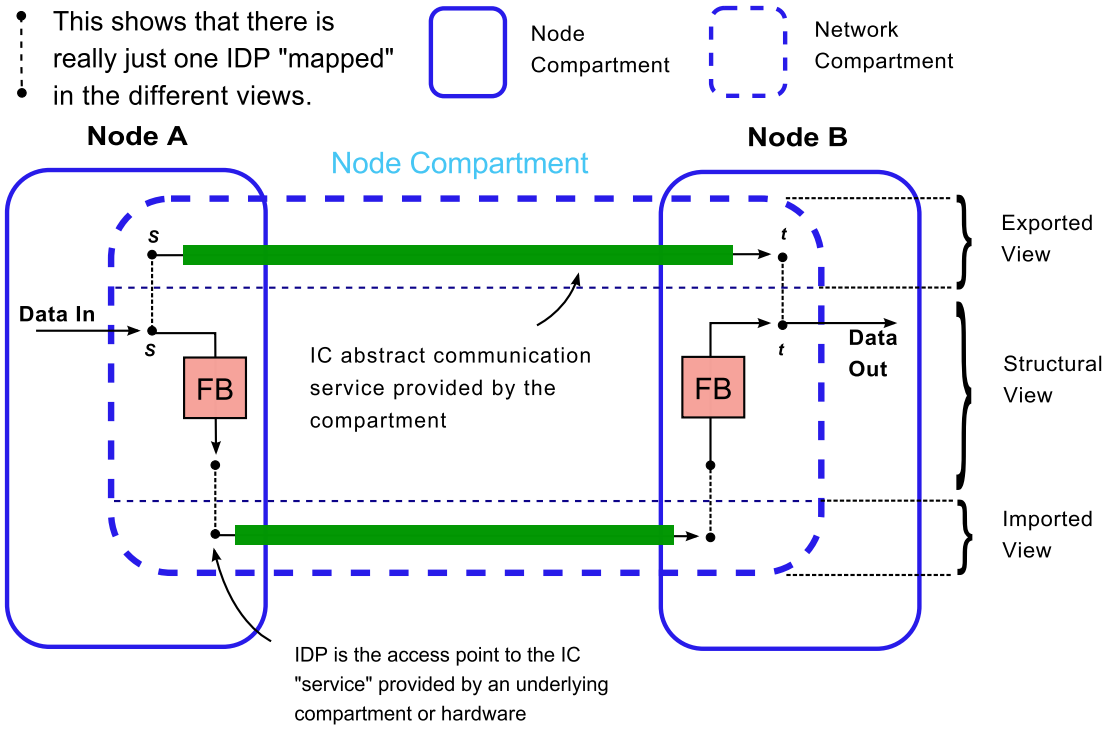


Figure 2: ANA architecture Abstractions [41]

multiple and diverse networking styles and protocols. For example, each network component, is allowed to use a different addressing, naming, routing mechanism, protocol, packet formats, etc.

Figure 2 shows the abstractions in ANA. Instead of routing packets in a hierarchical way as in the Internet, ANA proposes routing between network components. For this, the authors define a communication core that allows these different abstractions to communicate amongst each other. To offer a standard access to the communication services it provides, each compartment must support a generic “compartment API” that wraps its internal operation with generic constructs. In each network compartment, a distributed set of protocol entities collaborate in order to provide communication services to other compartments and applications. This communication service is abstracted by an information channel (IC), which is illustrated by Figure 2 where the network compartment exports a communication service abstracted by the IC from *s* to *t*.
When interacting with a compartment with the generic API, an entity actually does so with a (node-local) software component implementing the operation of the compartment, called the functional block (FB). The FB is an abstraction of any protocol entity generating, consuming, processing and forwarding information, such as an IP stack or a TCP module. In addition to ICs and FBs, a fundamental feature of ANA is that an IC or a FB is always accessed via an indirection system that forms the core of ANA. This is the information dispatch point (IDP). Basically inside an ANA node, a functional block (FB) is always accessed via one or multiple IDPs attached to it. This means that all interactions are carried out via an indirection level built in the network architecture. Finally, ANA has the additional particularity of pushing networking abstractions inside the network hosts. The architecture considers a networking node to be itself a network composed by the functional blocks running on the host. As a result, every ANA node is organized as a node compartment which operates like any other (network) compartment except that it does not provide information channels. This permits functional blocks to discover each other and interact inside the node compartment in the same manner as with any other network compartment. Figure 3 describes how the Autonomic Network Architecture map the evaluation criteria which we proposed in section 3.2. Node

compartments are illustrated in Figure 2 by solid-line rectangles with rounded corners. All these abstractions and interfaces are publicly available as a prototype software, which is developed in C for Linux platforms[7].

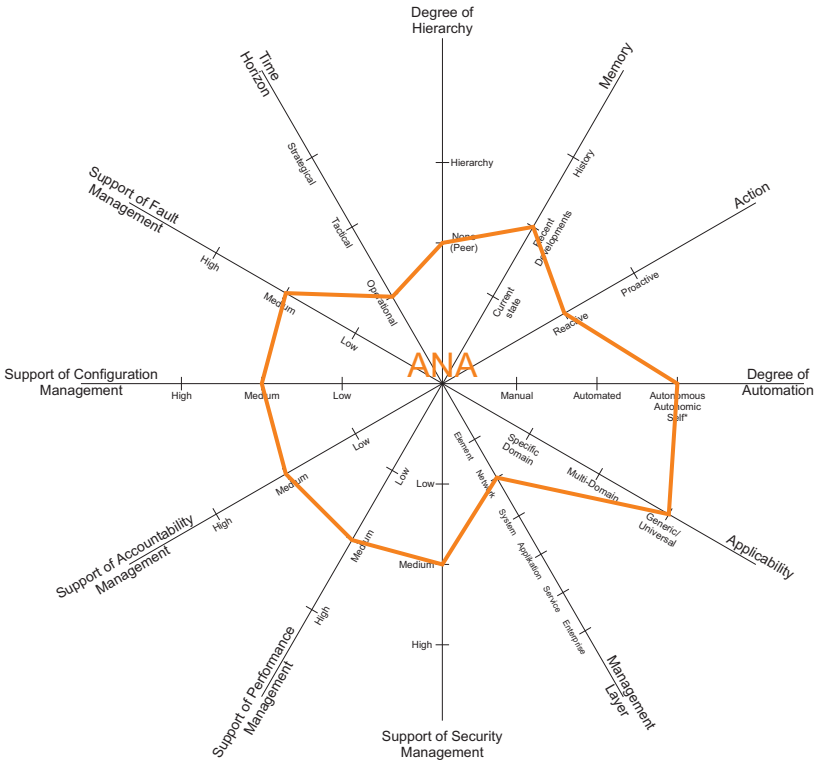


Figure 3: Classification of ANA

3.3.2 ANEMA: Autonomic network management architecture

The autonomic network management architecture (ANEMA) [42] uses policies and utility function theory to achieve autonomic behavior in IP-based network equipment. The architecture defines three types of policies: The utility function policies are used to capture the high-level objectives of the human administrators and the users while the goal policies describe the high-level management directives needed to guide the network to achieve the utility functions. Finally, behavioral policies describe the behaviors that should be followed by network equipment to react to changes in their context and to achieve the given goal policies. ANEMA organizes the operation of the architecture into two layers: the objective definition layer (ODL) and the objectives achievement layer (OAL). These layers are illustrated in Figure 4.

The major component of the ODL is the objectives definition point (ODP), which allows administrators and experts to introduce high-level requirements and management guidelines respectively. The high-level requirements are transformed into network utility functions (NUFs), while the management guidelines are transformed into abstract management strategies. The NUFs represent the policy rules that describe the network performance criteria from the human viewpoint. They are used to express the network functionalities in terms of optimization functions, whereas the management strategies describe the high-level guidelines needed to map the NUF to the specific management architecture that can be implemented within the target network infrastructure.

[7] http://sourceforge.net/projects/ana/

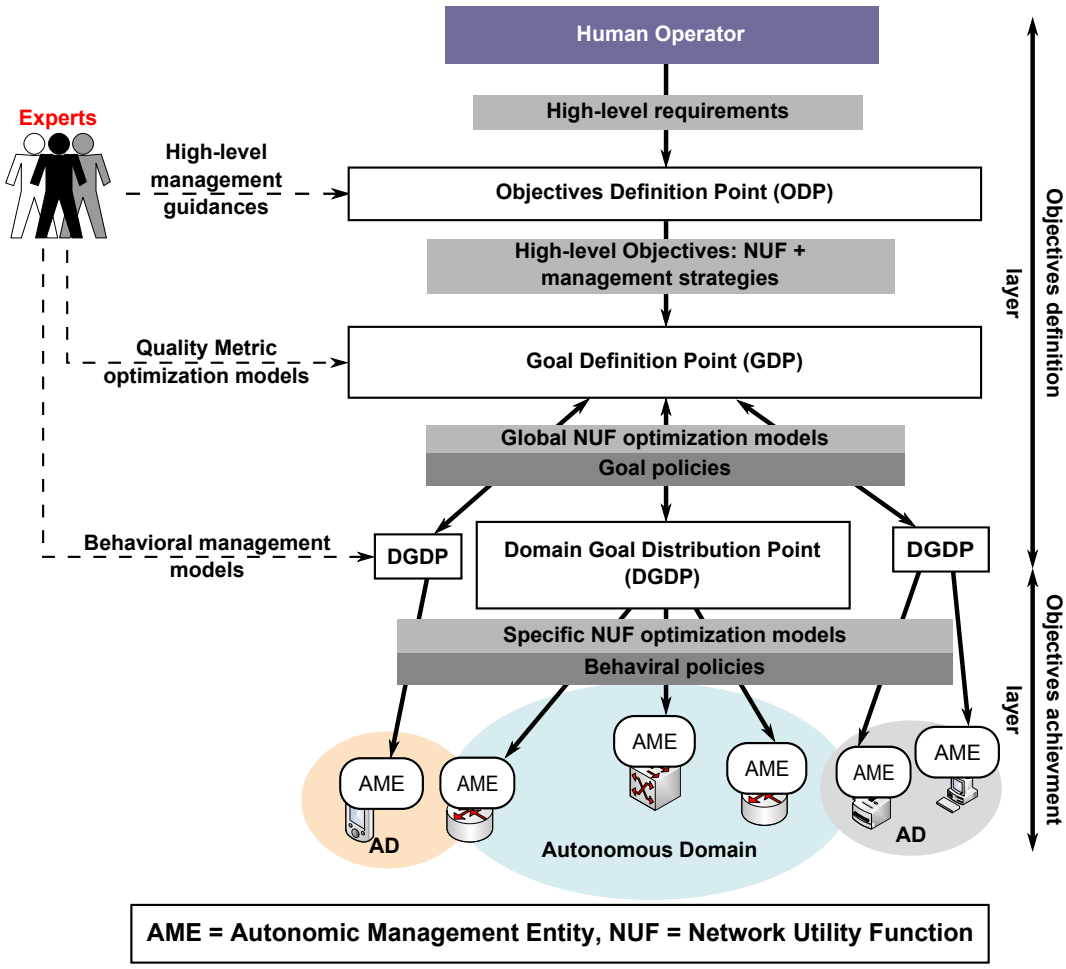


Figure 4: ANEMA architecture components [42]

The ODP forwards the formulated NUF and strategies to the goal definition point (GDP), which analyzes them and identifies the related goal policies. Specifically, the GDP analyzes the NUF and selects the appropriate global models from the quality metrics optimization models introduced by the experts. It also selects the appropriate management strategies to achieve the NUFs. The selected strategies represent the goal policies, which are defined as an aggregation of management strategies needed to achieve one or more quality metrics related to the target NUFs. To facilitate the goal policies distribution, the network infrastructure is divided into multiple and smaller domains, each of which is supervised by a coordinator called Domain Goal Distribution Point (DGDP). Therefore, the GDP forwards the goal specifications and global NUF optimization models to all DGDPs which analyze them to identify the appropriate behavioral policies to achieve the goal policies, by using abstract behavioral management models introduced by the experts.

The last form of policies is expressed in terms of behavioral management rules recognized by the target autonomic management entities (AMEs) – also called goal achievement points (GAPs) – in various management situations. The global NUF optimization models are transformed into specific models by applying the constraints deduced from the goals, after which the DGDPs send the behavioral policies and the specific NUF optimization models to all the AMEs under their control.

The OAL layer contains a set of GAPs. Each GAP is an entity that behaves autonomically while trying to achieve the target high-level requirements by considering the goal policies and the NUF optimization models. In fact, according to these informational elements, the GAP can make its own

decisions to achieve the target requirements by means of its elementary monitoring, analyzing, planning and executing (MAPE) capabilities. The GAP is also able to interact with its environment and communicate with other GAPs. In real networks, a GAP can be a router, switch, gateway, software, multimedia device, etc. The experts are only needed to introduce their knowledge in the knowledge base of the ANEMA components, but are not part of the management process. For this reason, the links between the experts and the management components are realized by dotted lines in Figure 4. Figure 5 describes how the autonomic network management architecture map the evaluation criteria which we proposed in Section 3.2.

The authors perform a set of simulations based on several scenarios in IP networks. However, they focus on just two (self-configuration and self-optimization) of the self-* properties of autonomic computing.

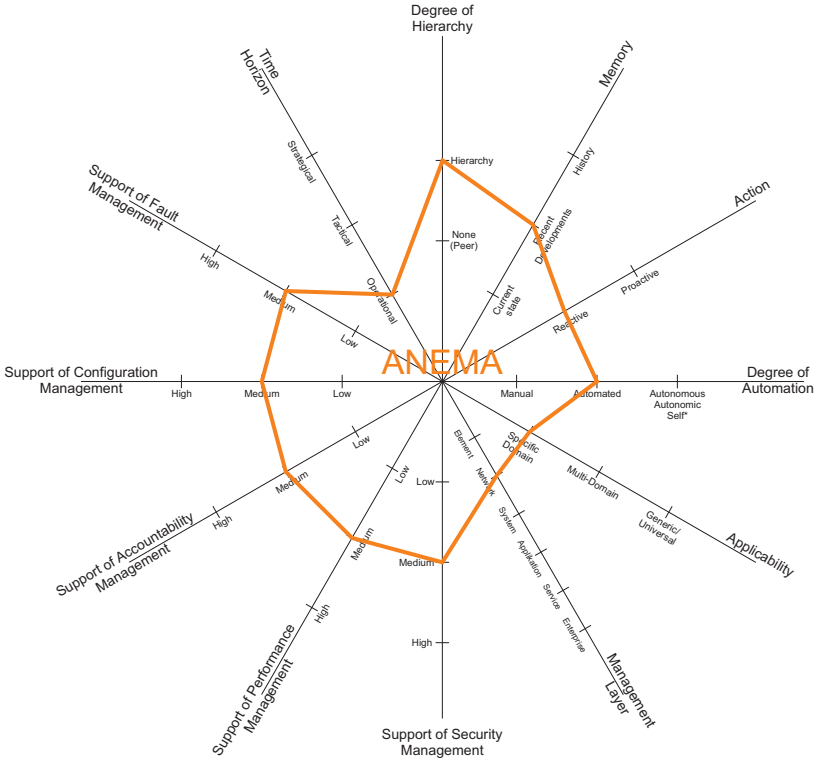


Figure 5: Classification of ANEMA

3.3.3 CASCADAS: Component-ware for Autonomic, Situation-aware Communications and Dynamically Adaptable Services

The CASCADAS framework [43, 44], acronym for 'Component-ware for Autonomic, Situation-aware Communications and Dynamically Adaptable Services', is the result of the eponymous project. CASCADAS aims at introducing "smart" services, capable of adapting their logic to the dynamically changing context they operate in without human intervention. To support this vision, four key enablers have been identified.

- **Self-similarity:** Self-similarity allows individual atomic components to self-organize and self-aggregate to realize any complex service. Eventually, the aggregated service appears as if it was atomic as well. Self-similarity helps to increase architectural scalability and is the key enabler for the composition of complex communication-intensive services. In the CASCADAS

framework, self-similarity is supported by imposing that autonomous components, formed as aggregation of other autonomous components, provide the same set of interfaces as the original ones.

- **Situation awareness:** The ability of refining decisions according to the specific contextual situations. Compared to context-awareness, where services are given access to isolated pieces of contextual data, in situation-awareness, services are given access to properly elaborated and organized information, representing comprehensive situation knowledge in a much more expressive way. It is achieved through the use of a Knowledge Network (KN) service, which is in charge of gathering and processing information to form a collection of Knowledge Atoms (KAs). These KAs structure information in a data model conceived around the consideration that any bit of contextual knowledge is produced as a consequence of an event occurring in the context. Therefore, all data is represented in the form of the 4-tuple (who, what, where, when), respectively representing the entity, activity, location and moment. The consistency of the information in the KN is verified by the Context Verificator. The architecture of the KN is shown in Figure 6.

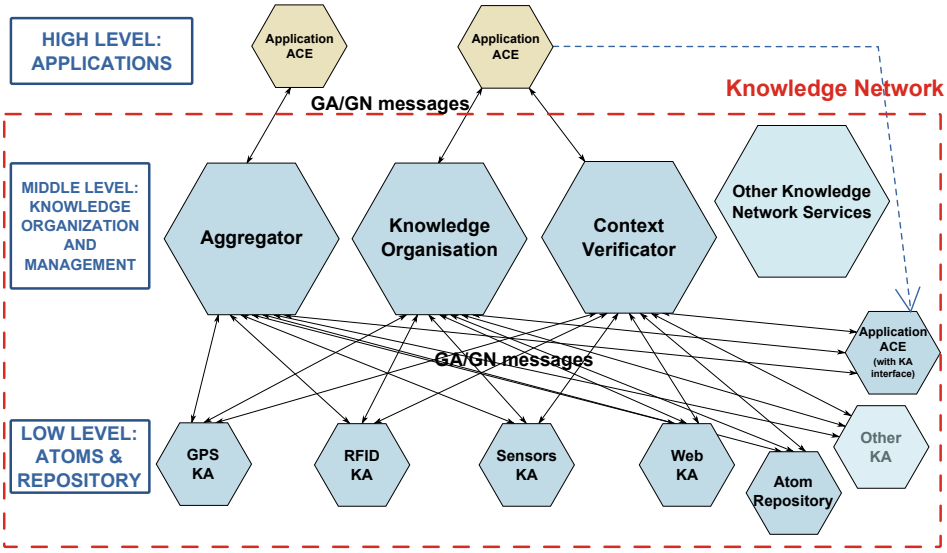


Figure 6: Architecture of the KN [44]

- **Semantic self-organization:** Self-organizing systems work bottom-up without any high-level representation. A large number of components interact according to simple local rules. The global adaptive behavior is the result from these local interactions. Self-organization enables service composition and aggregation. By enriching self-organizing components with more semantic, they can evaluate their own behavior and change it when they are not accomplishing what they intended or when better performance is possible. In the CASCADAS framework, semantic self-organization is supported through autonomous clustering, differentiation and synchronization.

- **Autonomic componentware:** An autonomic component model should provide both a robust and dynamic modular conceptual framework for building autonomic, self-organizing, semantic services. Furthermore, this model should act as an abstract and generic reference model for the production of a new generation of programmable communication elements that can be reused at different stack layers. The autonomic component model has to provide proper abstractions and tools to support both self-similarity, self-organization and situation awareness. Therefore, the autonomic components have to be explicitly situated in a world of situational

knowledge, provided with mechanisms for semantic self-aggregation and composition and designed to support self-similarity independently of scale.

As illustrated in Figure 7, the Autonomic Communication Element (ACE) abstraction forms the cornerstone of the CASCADAS component model. ACEs act as entities that can implement communication services, typically in a distributed way. They act and are perceived as service access points. A central concept within the ACE model is the organ. An organ is an internal ACE component, able to harmonize its own behavior to the execution of other interacting organs and to the context in which the ACE is operating. Inside an ACE, the component is observed and corrective measures are issued upon detection of hazardous situations. This principle is called pervasive supervision. Another pertinent issue in the component model is that of security and self-preservation. As the system lacks any soft of centralized authority, a-priori trust relationships between ACEs belonging to different administrative domains cannot be assumed. Figure 8 describes how CASCADAS map the evaluation criteria which we proposed in Section 3.2. The CASCADAS framework therefore exploits aggregation functionality and the GN/GA (goal needed / goal achieved) communication protocol to provide security in an adaptive and flexible way.

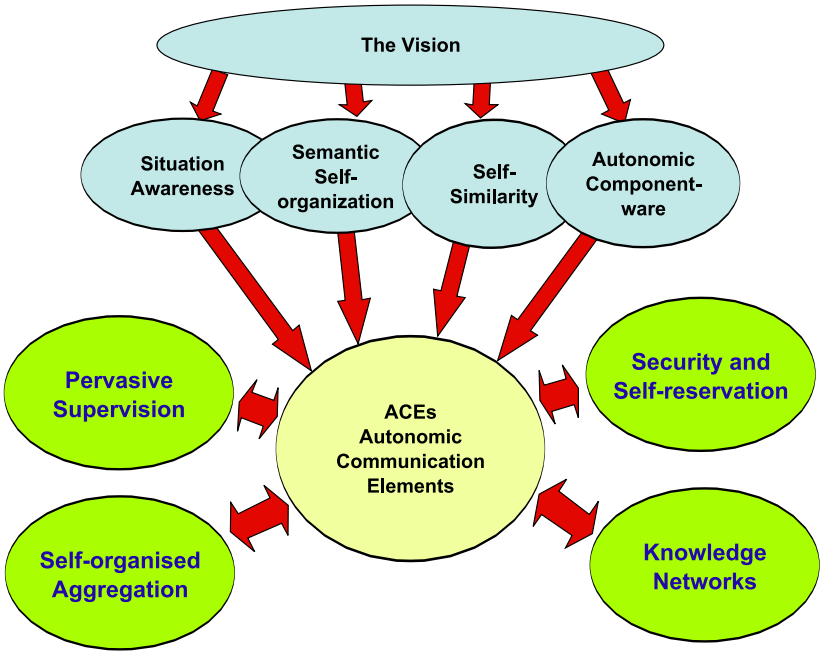


Figure 7: ACEs as the central abstraction of the CASCADAS component model [43]

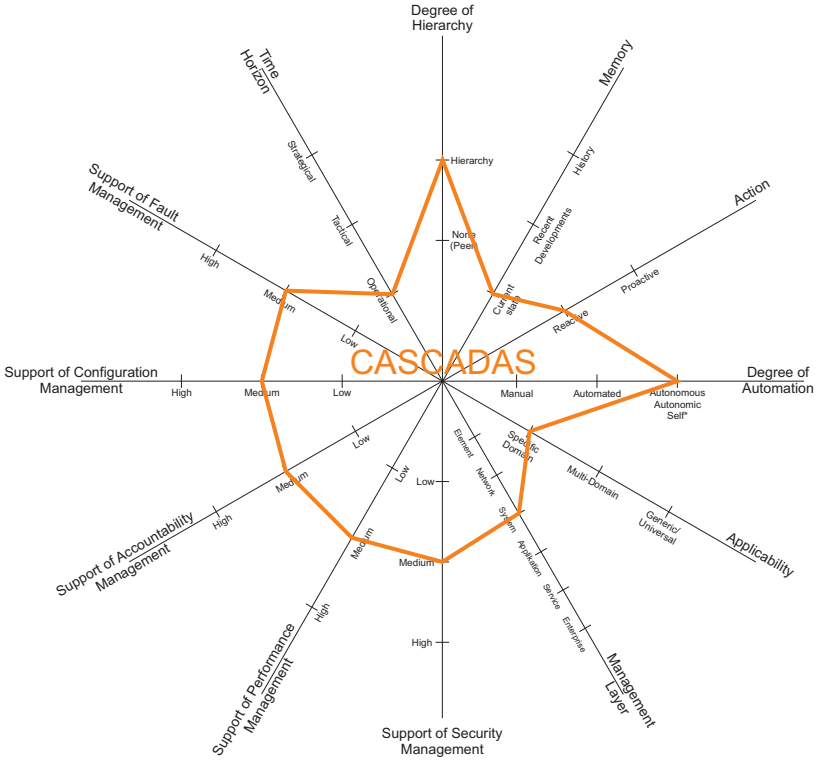


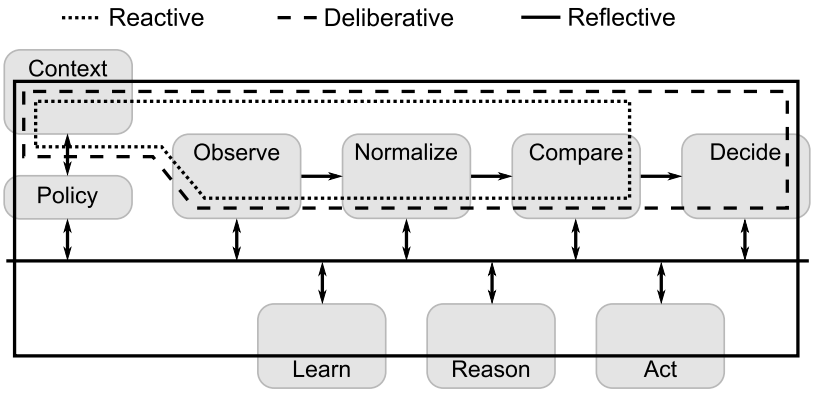
Figure 8: Classification of CASCADAS

3.3.4 FOCALE: Autonomic Network Management Architecture

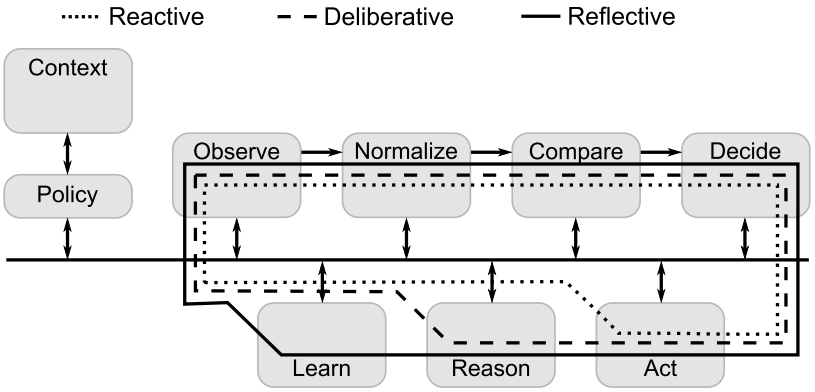
FOCALE stands for Foundation, Observation, Comparison, Action, Learning, rEason, which are the six key principles required to support autonomic networking. The FOCALE autonomic architecture is built around the FOCALE control loop, on which we will elaborate further on. FOCALE [45, 46] uses an internal representation mapping vendor specific functionality to a common technology-independent form (DEN-ng). This internal representation is augmented with ontologies to attach meaning and semantics to it (DENON-ng), which allows discovering semantically similar functionality offered by heterogeneous devices. Not only information and messaged are modeled, FOCALE also allows modeling context to support context-aware policy management. Multiple control loops use this context representation to provide adaptive control to govern resources and services.

FOCALE control loops are based on advances in cognitive psychology and provide two sets of control loops [46]: a set of outer control loops (Figure 9a) that affect the computations performed in a corresponding set of inner control loops (Figure 9b). The outer control loops are responsible for large-scale adjustments of functionality by adapting to context changes, while the inner control loops make more granular adjustments within a particular context. Both the outer and inner control loops use reactive, deliberative and reflective loops. The reactive loop is used to react to context changes that have previously been analyzed. In such case, no complex reasoning is required and we can perform a previously inferred behavior change. The deliberative control loop is used when a context change has occurred, but its details are not sufficiently well understood. The reflective loops is used to better understand how the context changes affect the goals of the autonomic element.

Next to a set of control loops, FOCALE also provides the notion of enhanced Autonomic Elements (AE). It is an abstraction that allows FOCALE to provide distributed functions such as commu-



(a) FOCALE Outer Control Loops



(b) FOCALE Inner Control Loops

Figure 9: FOCALE Control Loops [47]

nication, learning, reasoning and management. Each AE provides a set of services to perform knowledge management, composition, business-enabling and orchestration. Additionally, AEs can cooperate and collaborate in communities by sharing functionality and information as shown in Figure 10. In addition Figure 11 describes how the FOCALE approach map the evaluation criteria which we proposed in Section 3.2.

FOCALE is centered around the enhanced AE, which is made up of a set of distributable components connected through an enterprise context bus (ECB). This event-driven message broker is an extension to the enterprise service bus (ESB) which handles the exchange of messages as well as content and allows meaning-based routing. Furthermore the ECB supports different types of knowledge acquisition (e.g. push, pull and scheduled) and performs processing (e.g. semantic annotation, filtering and storage). This enables AEs to register interest in knowledge in a more precise fashion, reducing the messaging overhead. The Autonomic Manager uses the ECB to orchestrate behavior of the managed entities. Components can register interest in specific types of knowledge, which reduces messaging overhead.

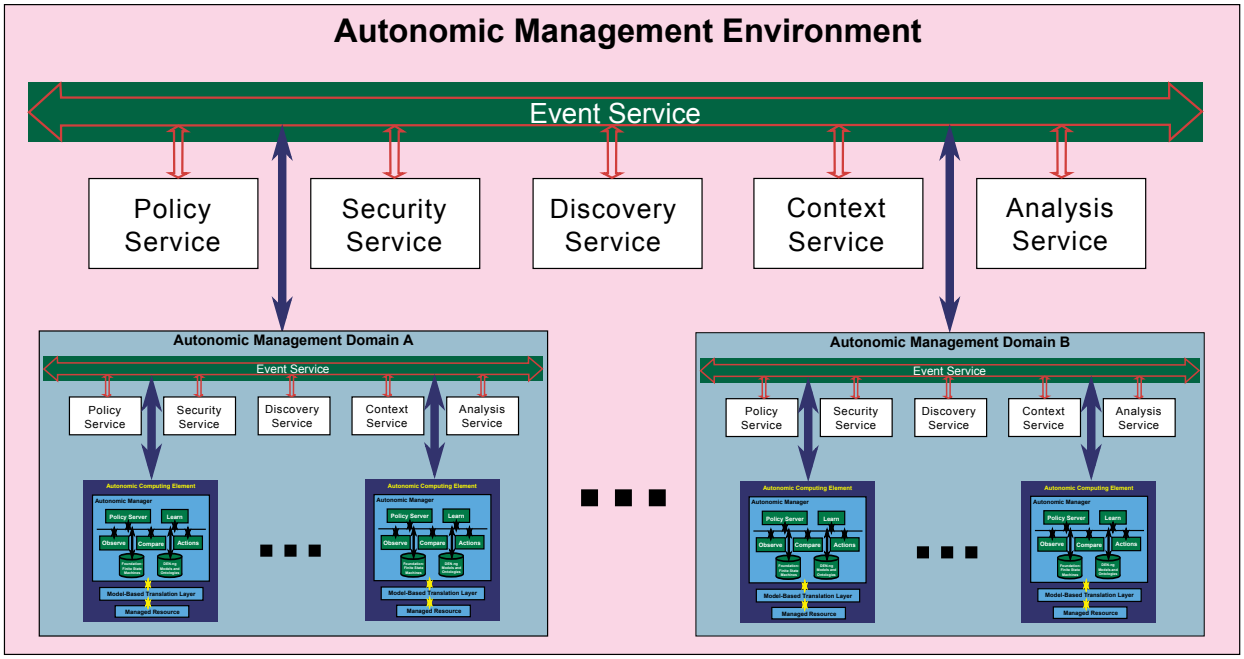


Figure 10: FOCALE Autonomic Management Environment [45]

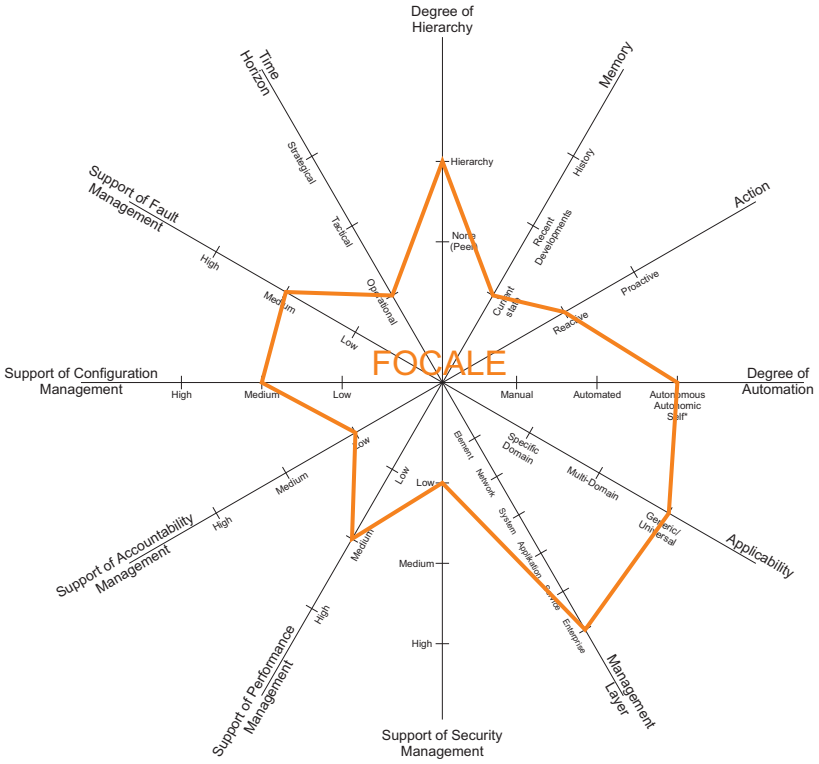


Figure 11: Classification of FOCALE

3.3.5 The SELFMAN Framework

This framework was proposed in the context of the SELFMAN specific targeted research project (STREP) in the Information Society Technologies (IST) Strategic Objective 2.5.5 part of the sixth European Framework Program. The project introduces the foundations of a self-management framework designed for large-scale distributed networks [48, 49]. The framework, depicted on Figure 12, is mainly based on two concepts: a structured overlay network developed over peer-to-peer systems whose goal is to provide efficient routing services and robust communications, and an advanced component model whose objective is to easily extend the self-managing properties of the overlay network by rebuilding it using components. On the one side, as most structured overlay networks, the framework relies on a ring structure, where all the nodes are connected and periodically join or leave the network. On the other side, the components must provide introspection, reflection and dynamic reconfiguration abilities to those nodes. For that purpose, the self-management activities of nodes and programs are implemented based on feedback structures. Each feedback structure is composed of one or more feedback loops and aims at maintaining one global system property. In a well-designed system, no part should exist outside of a feedback structure.

Combining these two key concepts, the SELFMAN framework has been designed to tackle several specific challenges:

- A first challenge was to handle the network partitioning issue. During the lifetime of the ring, the structured overlay network may be split into several isolated rings after the failure or the leave of given nodes. A ring merge service has therefore been introduced in order to properly repair individual rings during a split, and in order to support the merging of rings in an efficient manner once the rings are closer and not isolated anymore. The algorithm has been refined to improve speed and ensure termination, neither cluttering the network with many messages nor adding significant overhead information amount. Furthermore, the algorithm has been designed in a flexible way since the trade-off between message complexity and time complexity can be adjusted by a dedicated parameter.

- A second challenge was to implement an efficient decentralized transaction service over the structured overlay network - prone to churn (frequent node joins, leaves, and failures) and Internet-style failures - in order to support self-management applications. This service relies on a distributed hash table (DHT) capable of storing keys and their values with strong data consistency and atomic transactions. Strong data consistency among replicas as well as atomic transactions have been introduced in order to allow a wide range of additional application domains to benefit from the inherent scalability and fault-tolerance of DHTs. The solution uses symmetric key replication to ensure data availability in the face of churn, and enforces the data consistency by using a lightweight Paxos commit protocol that exploits information from the replica distribution in the DHT table.

- A third challenge was to introduce a load balancing service in the structured overlay network. The objective was to maintain an equitable distribution of work amongst nodes when management operations are performed, typically when queries are performed over the distributed hash table. The solution exploits local techniques complemented with gossiping techniques to obtain estimates of global information with low overhead, in particular the average load and the standard deviation of load among the network nodes. Such information can then be added to existing load balancing algorithms that can use the additional knowledge to improve their performance. In particular, the experiments have shown the linear scalability of the approach as well as a benefit of up to 30% in terms of load balancing when exploiting this knowledge.

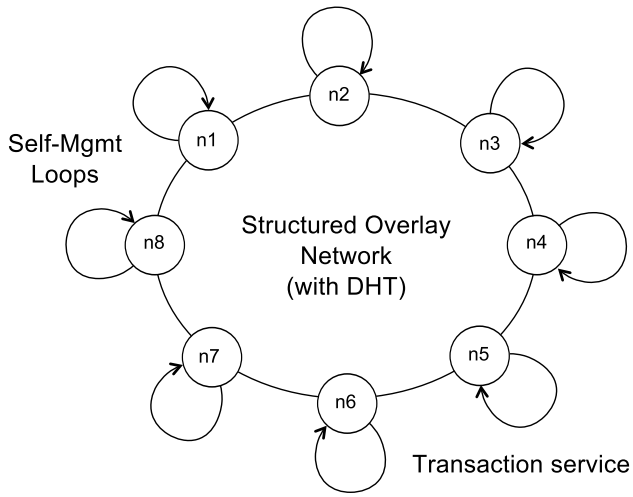


Figure 12: SELFMAN Framework Overview

Two additional contributions have resulted from the SELFMAN project: a simulation environment for peer-to-peer systems and guidelines for securing peer-to-peer networks. The first contribution has consisted in the design of a simulation environment for peer-to-peer systems that implements the Kompics component model in order to provide strong modularity. The Kompics model has also been developed in the context of the SELFMAN project and offers an advanced component model for building reconfigurable distributed systems from event-driven components. The second contribution has focused on security aspects. The project has analyzed the use of small-world network (SWN) topologies in order to make peer-to-peer network more secure. The trust and identity relationships defined in these topologies allow to reduce the probability of attacks. Figure 13 describes how the SELFMAN approach map the evaluation criteria which we proposed in Section 3.2.

Three main demonstrators have been developed to evaluate the SELFMAN framework: an on-demand video streaming application (called Peer TV) with dynamical reconfiguration in order to optimize quality of service, a distributed wiki application (called Distributed Wikipedia) with versioning and replication functionalities, and a decentralized drawing application (called DeTransDraw) for mobile phones capable of self-recovering after failures and supporting transactional operations. The SELFMAN project has also developed an open-source library, named Scalaris, that provides a scalable storage service. Scalaris is built on top of the structured peer-to-peer network and includes most of the features that the SELFMAN project has focused on, including failure tolerance, strong data consistency, and load balancing.

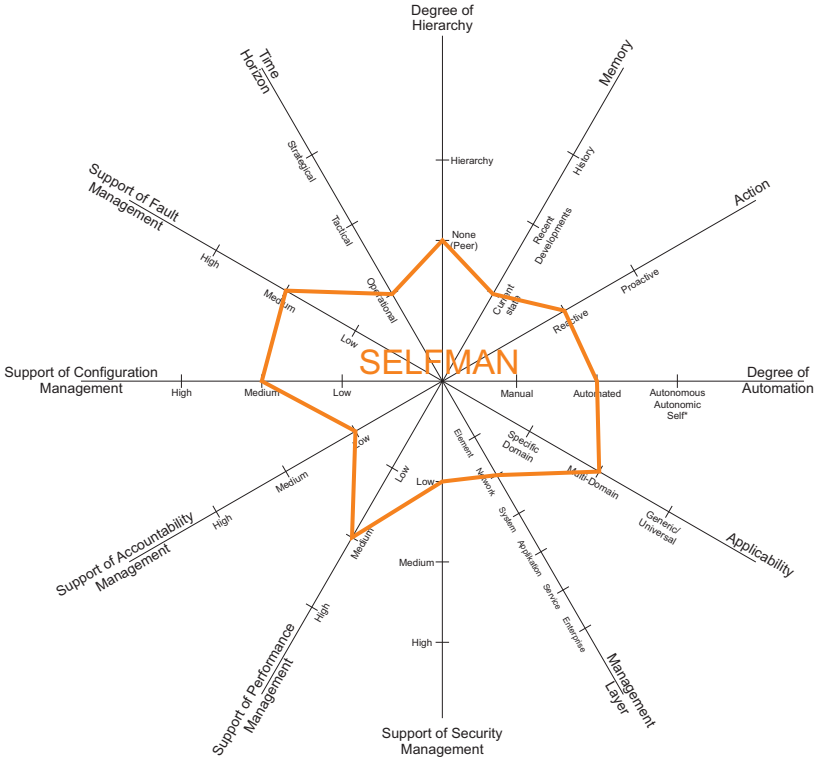


Figure 13: Classification of SELFMAN

3.3.6 SCAP-oriented architecture

The Security Content Automation Protocol (SCAP) is a suite of specifications that standardize security information and its related processes. In particular, it provides a support for automating the detection and repair of vulnerable configurations [50]. Its development is supported by the National Institute of Standards and Technology (NIST) [51]. It includes the Open Vulnerability and Assessment Language (OVAL) [52] and the eXtensible Configuration Checklist Description Format (XCCDF) [53] that allow us to specify and identify configuration vulnerabilities, but also to bring a device into compliance through the remediation of identified vulnerabilities or misconfiguration. These two languages play an important role with respect to automated configuration and repair.

On the one hand, the OVAL language permits to express vulnerability descriptions, configuration information and assessment results. It standardizes the process by which the state of a computer device can be analyzed and reported. Briefly, the OVAL specification is supported by XML schemas which serve as both, the framework and vocabulary for the language. These schemas specify what content is valid within an OVAL document and what is not. OVAL is organized in three main XML schemas, namely, (i) the OVAL Definition Schema that expresses a specific machine state; (ii) the OVAL Characteristics Schema that stores configuration information gathered from a system; and (iii) the OVAL Results Schema that presents the output from a comparison of an OVAL Definition against an OVAL System Characteristics instance. Valid XML instances typically represent specific machine states such as vulnerable states, configuration settings and patch states. Real analysis however is performed by OVAL interpreters such as Ovaldi [17] and XOvaldi [18]. Usually, a vulnerability is considered as a logical combination of conditions that if observed on a target system, the security problem described by such vulnerability is present on that system. OVAL follows the same idea by considering a vulnerability description as an OVAL definition. An OVAL definition specifies a criterion that logically combines a set of OVAL tests. Each OVAL test in turn represents

the process by which a specific condition or property is assessed on the target system. Each OVAL test examines an OVAL object looking for a specific OVAL state. Components found in the system matching the OVAL object description are called OVAL items. These items are compared against the specified OVAL state in order to build the OVAL test result. The overall result for the criterion specified in the OVAL definition will be built using the results of each referenced OVAL test.

On the other hand, XCCDF is an XML-based specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target systems. Each XCCDF document holds exactly one Benchmark object. An XCCDF benchmark is the main container of the checklist and it holds three types of items, namely, (1) groups, that hold related items into a common structure, (2) rules, that hold check references and remediation information, and (3) values, that provide named data values that can be substituted into other items properties or into checks. An XCCDF benchmark can also contain additional data of different types, such as *Profile* objects (a collection of attributed references to Rule, Group and Value objects) and *TestResult* objects (that hold the results of performing some compliance tests against a single target device or system). XCCDF rules are of particular interest because they allow to specify remediation information that can be used by automated systems to perform corrective actions when specific states are detected. These specific states can be specified by languages such as OVAL, while the actions to take when vulnerable states are found can be expressed with XCCDF.

In the general case, architectures using the SCAP protocol exhibits an OVAL repository containing known vulnerability descriptions and an XCCDF repository involving vulnerability treatments to be performed when these specific states are detected. Figure 14 illustrates the use of these two standards, OVAL and XCCDF, for automatically detecting security vulnerabilities and performing corrective actions. The vulnerability manager plays a central role and it is in charge of orchestrating the whole activity in an automated manner. First, OVAL descriptions are consumed and analyzed over those devices under control. To achieve this point, agents are deployed in the network in order to manage target network devices. These agents receive specific operations from the vulnerability manager component in order to execute concrete activities. For those devices found to be vulnerable, the manager will search and consume available treatments (XCCDF descriptions) for eradicating these vulnerabilities. Once again, specific operations are sent to agents in the network in order to bring controlled devices to safe configuration states. Depending on the policies that govern the system, the vulnerability manager can also take advantage of the Common Vulnerability Scoring System (CVSS) [54] in order to decide what configuration security issues must be considered of high priority. Figure 15 describes how the SCAP-oriented architecture approach map the evaluation criteria which we proposed in Section 3.2.

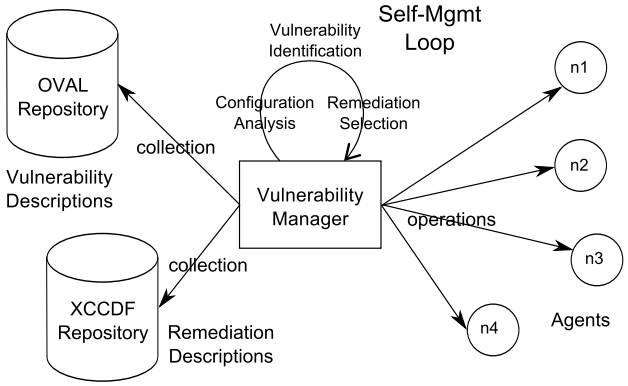


Figure 14: SCAP-Oriented Architecture

In addition, SCAP includes the CVSS providing a framework for quantifying the impact of vulnerabilities. The normalized score computed for each vulnerability is based on several types of metrics. It provides a strong support for risk assessment techniques and enables the prioritization of actions to be taken, when changes are performed. In addition, SCAP considers enumeration languages such as the Common Vulnerabilities and Exposures language (CVE), a standard for the enumeration of known information security vulnerabilities; the Common Platform Enumeration (CPE), a nomenclature and dictionary of hardware, operating systems, and applications; and the Common Configuration Enumeration (CCE), a nomenclature and dictionary of security software configurations.

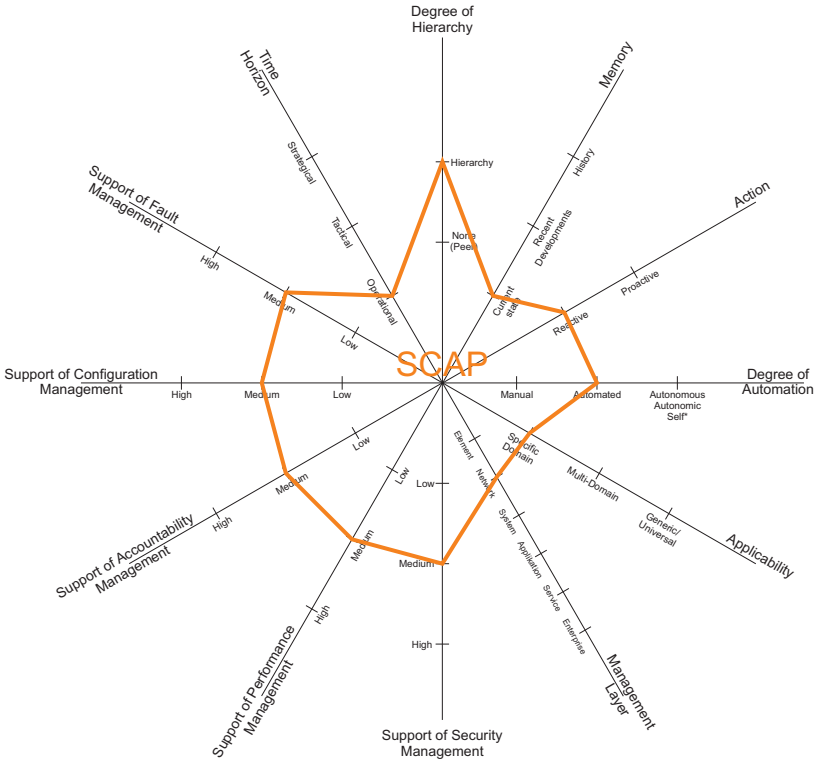


Figure 15: Classification of SCAP

3.3.7 DACoRM: Decentralized and Adaptive Network Resource Management Framework

Aimed at overcoming the limitations of centralized offline solutions, DACoRM [55] is a decentralized in-network management framework for the reconfiguration of network resources. According to the proposed framework, the decision making process is distributed across the edge nodes of the network, which are embedded with a level of intelligence that allows them to react to network conditions in an adaptive fashion based on local feedback regarding the state of the network. In contrast to centralized solutions where reconfigurations are decided by a centralised management system that has a global knowledge about the network, reconfiguration decisions are directly taken by the network edge nodes that coordinate among themselves in order to decide upon the best sequence of actions to perform to satisfy a common objective. These can for instance consist in adjustments of routing parameters for load balancing purposes. In order to support this decentralized decision making process, the network edge nodes are organized into a *management substrate* (MS), which is a logical structure used to facilitate the exchange of information between decision making entities [56]. The management substrate is used by the edge nodes for coordination purposes (i.e. signaling) providing the means through which nodes can communicate.

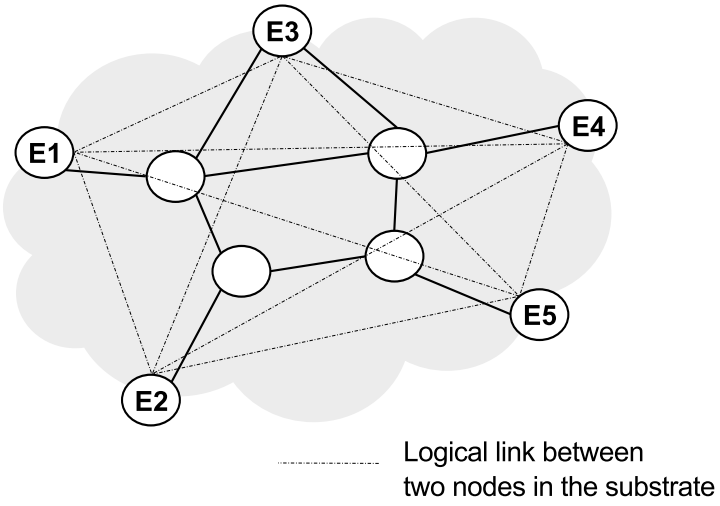
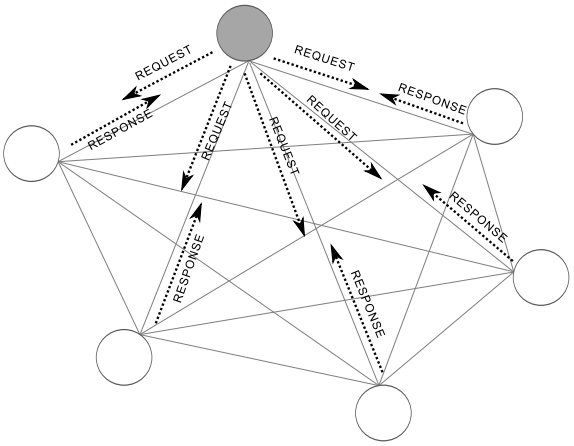
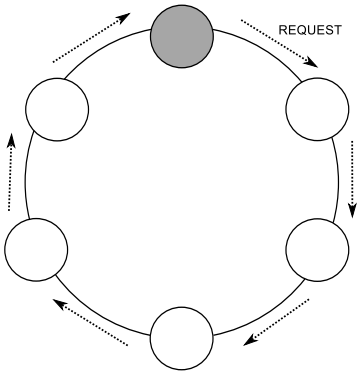


Figure 16: In-network management substrate

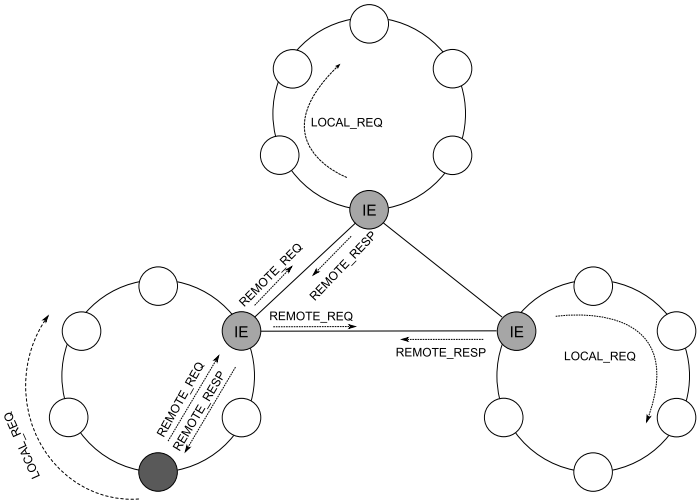
A management substrate structure example is depicted in Figure 16, where each network edge node E is logically connected to a set of other network edge nodes (neighbor nodes in the MS). Any MS node can directly communicate exclusively with its neighbors, which are defined by the topological structure used. The choice of the substrate topology can be driven by different parameters related to the physical network, such as its topology, the number of edge nodes, but also by the constraints of the coordination mechanism between the nodes and the associated communication protocol. The overhead incurred by the communication protocol in terms of delay and number of messages exchanged, for example, is a key factor that can influence the choice of the MS topology. DACoRM supports three different substrate topology structures: full-mesh, ring and hybrid.How the evaluation criteria proposed in Section 3.2 map DACoRM is shown in Figure 18.



(a) Full-mesh topology



(b) Ring topology



(c) Hybrid topology

Figure 17: Management substrate topology structures

- **Full-mesh Topology**: As shown in Figure 17a, this is a flat structure where each node is logically connected to every other node in the substrate. In this model, each MS node has a global view about other MS nodes since they can all communicate directly with REQUEST-RESPONSE messages. This provides a greater flexibility in the choice of neighbors with which to communicate since all MS nodes belong to the set of neighbors. Each MS node, however, needs to maintain locally information about every other MS node, which may raise some limitations as the number of nodes in the substrate increases.
- **Ring Topology**: According to Figure 17b, this is a flat structure where each node is logically connected to two other nodes only. The view of each MS node is limited to its two direct neighbors. Communication is unidirectional, which means that a node can only pass information to its immediate neighbor in the ring. To communicate with any other node, a message (REQUEST) needs to be sent over the ring until it reaches its destination. To minimize the communication cost (i.e. delay), a heuristic is used to construct the ring.

- **Hybrid Topology**: As shown in Figure 17c, this consists of a set of rings inter-connected in a fully-meshed fashion through nodes, called the Intermediate Entities (IE), so that there exists exactly one IE in each ring. More specifically, MS nodes are partitioned into at least two clusters, so that nodes in each of the clusters are connected according to a ring topology. A distinct node from each cluster is selected as the IE, i.e. to act as the interface to other clusters. The partitioning of MS nodes into clusters and the IE selection are based on algorithms that aim to minimize the communication cost. Communication between nodes in the same cluster is achieved with local messages, whereas remote messages through interface nodes are used to reach other clusters.

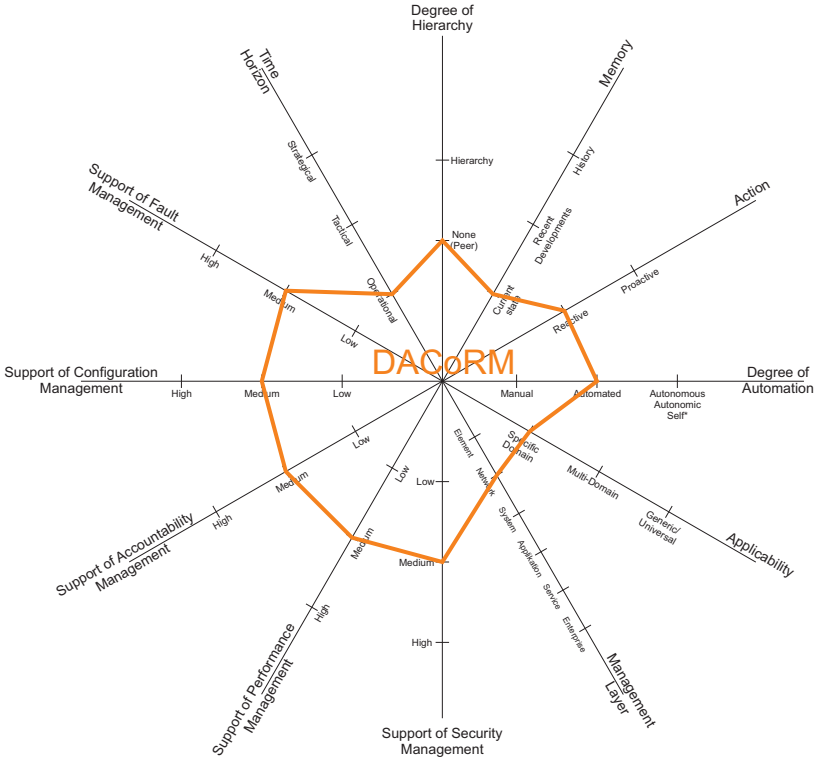


Figure 18: Classification of DACoRM

3.4 A First Blueprint of Metrics

The developed taxonomy allows us to compare architectures and approaches according to some qualitative criteria. However, with respect to the development of enablers such as learning algorithms, a quantitative comparison in terms of metrics is needed as well. In the following a first blueprint of metrics is described. Although metrics address mainly quantitative aspects, some qualitative viewpoints can be addressed as well.

3.4.1 Qualitative Metrics

An extensive number of metrics have been identified in the paper ”A survey of autonomic network architectures and evaluation criteria”, published in the IEEE Communications Surveys Tutorials 2012 [36], and building the basis of this section.

Table 5 provides an overview of these evaluation criteria.

Table 5: Evaluation Metrics

METRIC	DEFINITION	REFERENCES
DEGREE OF AWARENESS	The degree of self-awareness and environment-awareness of the architecture.	[40, 36]
PERFORMANCE OF AUTONOMIC RESPONSE (QOS)	A mean to conclude the degree to which the system is reaching its primary goal to improve some aspect of a service.	[37, 38, 36]
COST OF AUTONOMY	A mean to conclude the degree to which the system is reaching its primary goal to improve some aspect of a service.	[37, 38, 39, 36]
GRANULARITY/ FLEXIBILITY	The degree of decentralization of the architecture as a trade-off between the failure avoidance and overhead.	[37, 38, 36]
FAILURE AVOIDANCE	The ability to cope with predicted and unpredicted failures.	[37, 36]
DEGREE OF SELF-OPERATION	In which extend the architecture (or each MAPE-K components) performs in itself without administration involvement.	[37, 38, 39, 40, 36]
SPEED OF AUTONOMIC RESPONSE	How fast the system adapts itself to a change. It includes the total time to adapt, the reaction time and the stabilization time.	[37, 38, 36]
SENSITIVITY TO CHANGE	The reaction time, taken between an event is occurred until executing an adaptation response to cope with the event. Ideally, a reasonable observation time, not too short neither too long, is needed to avoid oscillations.	[37, 38, 36]
MEMORY STRENGTH	The ability of the system to maintain current state, behavior trend or historic of past actions in order to utilize them for management decisions.	[40, 36]
DEGREE OF ACTIVITY	The reactive or proactive behavior of the system to satisfy the self-management properties. Proactivity is critical in computing systems, where consequences of slight performance degradation or sudden failures are at higher costs than that of computing future states. Obviously, a favorable ANM architecture is highly proactive rather than reactive.	[40, 36]
ABILITY TO LEARN	The ability of the architecture to continuously evolve and enhance their applied adaptation strategies.	[40, 36]

GRANULARITY OF INTELLIGENCE	The degree of distribution of the intelligence in the network, i.e. the granularity of learning mechanisms.	[40, 36]
BENCHMARKING	A mean to bring all above-mentioned metrics together and observe their effective impact on the smooth operation of the real system.	[37, 36]
COMMON CRITERIA	The Common Criteria (CC) was developed to facilitate consistent evaluations of security products and systems. It is an international effort to define an IT Security evaluation methodology, which would receive mutual recognition between customers and vendors throughout the global economy. The CC does so by providing a common set of requirements for the security functionality of IT products and for assurance measures applied to these IT products during a security evaluation. These IT products may be implemented in hardware, firmware or software.	[57, 58]
ISO 27001	The ISO 27001 is the specification for an ISMS, an Information Security Management System. The objective is to ”provide requirements for establishing, implementing, maintaining and continuously improving an Information Security Management System (ISMS)”. Further, ”The design and implementation of an organization’s information security management system is influenced by the organization’s needs and objectives, security requirements, the organizational processes used and the size and structure of the organization”.	[59]
ITIL	The IT infrastructure library (ITIL) best practice framework enables managers to document, audit, and improve their IT service management processes.	[60, 61, 62]
REDUNDANCY	Redundancy is the additional presence of functionally identical or similar resources of a technical system, when they are not needed for a stable operation in a normal case.	[62]
ADAPTABILITY	As other software architectures a system cold be something between close-adaptive and open-adaptive.	[63]
STANDARDS	A standard is a comparatively uniform or unified, widely recognized and usually applied (or at least intended) way to produce or perform, which has prevailed over other ways. With regard to IT, the use of standards is generally considered to be the best way of assuring adaptability and compatibility.	[64, 65]

As depicted in Figure 19, the classification used within this deliverable covers all metrics identified in the literature (especially those identified in [36]), and is visualized in relation with the taxonomy (green color is used for qualitative metrics, orange color is used for quantitative metrics).

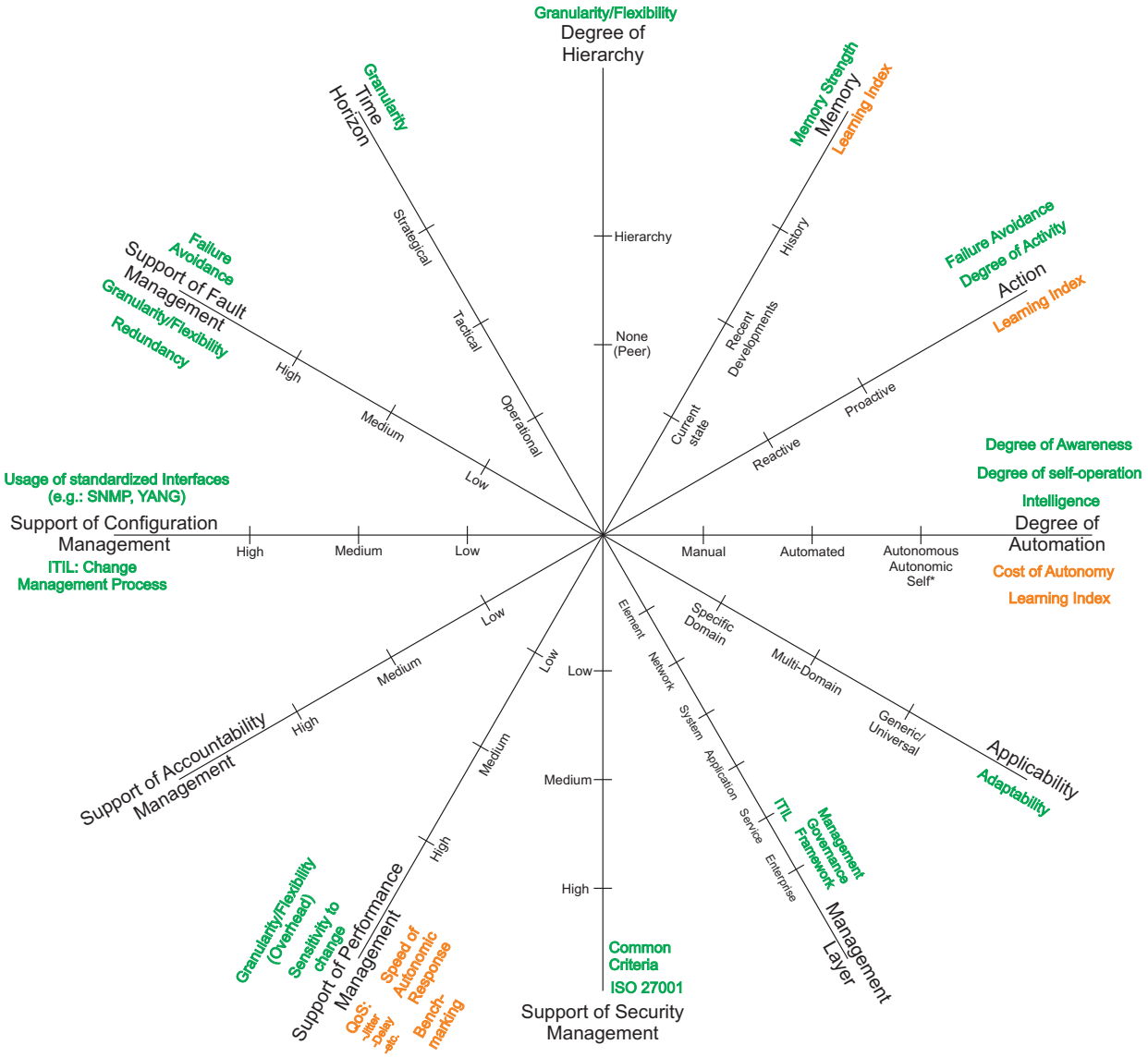


Figure 19: Comparison of qualitative metrics with our taxonomy

3.4.2 Quantitative Metrics

While Section 3.4.1 mainly focused on qualitative evaluation metrics, this section highlights important quantitative evaluation metrics. The quantitative metrics are especially important with respect to the analysis of enablers. For example, the learning index is important for the selection of learning algorithms.

Performance of Autonomic Response (QoS)
The performance of the autonomic systems describes the degree in which the system is reaching its primary goal to improve some aspects of the service. Autonomic networking can be seen as a

way to increase the overall performance of the network in terms of quality of service [36]. QoS can be used as a means to measure the efficiency of the autonomic solution, based on quantitative measurements of domain-specific metrics, more usually packet delivery ratio or throughput, loss rate, end-to-end delay, jitter, etc [66]. However, it is important to note that the desired QoS metrics may be different regarding to the objectives and applications of the target network [36]. Table 6 and 7 provide two examples of QoS parameters.

Table 6: Delay Guideline for VoIP [36]

END TO END DELAY	EFFECT ON PERCEIVED QUALITY
0 - 150 ms	Delay not noticeable /by humans)
150 - 250 ms	Still acceptable quality
250 - . . . ms	Unacceptable delay

Table 7: Jitter Guideline for VoIP [36]

END TO END DELAY	EFFECT ON PERCEIVED QUALITY
0 - 40 ms	Jitter not noticeable /by humans)
40 - 75 ms	Good quality, but occasional delay or jumble noticeable
75 - . . . ms	Too much

Cost of Autonomy
In autonomic networking, the network consumes additional resources to support autonomic control loops. This in turn, helps the network to reduce the amount of control traffic generated to support network services which is referred to as the cost of services [36].

[36] distinguishes two types of cost:

- Cost of autonomicity: The cost of autonomicity is defined as the overhead of extra autonomic activity which includes internal cost (e.g. CPU usage, storage, etc.) as well as external cost (e.g. overhead of knowledge monitoring, distributed intelligent adaptation, etc.) [38, 39].
- Cost of services: The cost of services is defined as the amount of control traffic generated to support prerequisite requirements of network services. As well, the cost of services can be divided into internal and external cost. For the case of a routing protocol service, the internal cost is the CPU usage and storage memory used to support the protocol. The external cost is the amount of control traffic generated to assist the well-performance of the protocol.

Subsequently [36] defines a cost index as the average of the costs consumed for autonomicity and services assistance. E.g., a solution which provides a perfect degree of autonomicity with a higher cost may be globally better than an architecture providing a poor autonomicity with lower cost. The overall internal cost of autonomicity and services, $Cost_{int}$, is computed by the average amount of internal resources (percentage) consumed for the purpose of autonomicity and services. The external cost are obtained by the relative total amount of extra traffic overhead (for autonomicity and services), $E_{Overhead}$, compared to the total useful network traffic, E_U.

The cost index, denoted as $Index_{Cost}$, is the average of internal and external cost:

$$Index_{Cost} = \frac{Cost_{int} + min\{\frac{E_{Overhead}}{E_U}, 1\}}{2}$$

Ability to Learn/Learning Index

An extreme potential of autonomic networking is its ability to learn from past experiences to enhance future operations [36]. This is mandatory to converge to an optimal adaptation.

For instance, a (traditional) policy-based solution without learning capability may consider any exceed of delay up to a predefined threshold as a sign of congestion in its policies description. Accordingly, each time this threshold is exceeded, the manager decides to drop some non-priority packets to reduce the delay for QoS packets [36].

In contrast, a learning-based solution would rethink the accuracy of previous congestion perception by analyzing the success of past experiences [36]. If learning mechanisms are used, the system could change this threshold if it learns that an increase or decrease of that value fits more with the reality of the underlying network. As well, the system could adjust its reaction once it learns that the further reaction provides better result [36].

[36] introduces a unified index called the learning index characterizing the ability to learn. These factors are as follows:

- The ratio of:
  - L the number of learnable management objects (policies, weights, parameters), L, versus
  - M the total number of network management parameters
- The coefficient of learning ability c_l, which is the relative number of well-performing learning-based decisions. This factor can be computed based on the analysis of the evolution of learnable parameters:
  - E_D the average number of total correct learning-based decisions,
  - $E_{D'}$ the average number of total incorrect learning-based decisions,
  - E_{PD} the total number of correct decisions of "Target Achievers" (learning objects with positive evolution towards the goal), and
  - E_{ND} the total number of incorrect decisions of "Target Damagers" (learning objects with negative evolution against the goal).

  Hence, the total equation is as follows:

$$c_l = \frac{max\{1 - \frac{E_{D'}}{E_{ND}}, 0\} + min\{\frac{E_D}{E_{PD}}, 1\}}{2}$$

- The efficiency of learning e, which is the overall degree of well-performance of the learning. This factor can be calculated based on the amount of progress towards the target:

$$e_l = \frac{K_{PD}}{K_{PD} + K_{ND'}}$$

  where K_{PD} is the average of total percent of learning object's enhancement of "Target Achievers". K_{ND} is the average of total percent of learning object's declination of "Target Damagers".

The learning index $Index_l$ and is computed by correlating the three contributing factors:

$$Index_l = \frac{L}{M} * c_l * e_l$$

The learning index is a number between 0.0 and 1.0 which indicates the overall learning ability of the architecture. The minimum value of 0.0 corresponds to a closed-adaptive system where L is zero. A complete open-adaptive system gives the value 1.0 for the learning index. If a system learns from the past experiences to continuously evolve and enhance their applied adaptation strategies it has a higher learning index as a system that uses a set of predefined adaptation strategies which are applied without any enhancement in adaptation schemes in runtime (see Figure 19)

The presented metrics present a first blueprint of metrics, and will be extended with other relevant metrics during the analysis and development of the enablers which is planned for the second year.

3.5 Description of Application Domains

The description of application domains investigated in FLAMINGO are in the focus of this section. Since the application domains are the basis for the validation of the developed approaches, a detailed description of the domain, and with this of the domain-specific requirements with respect to automation, is essential. To establish the link between the different application domains and the aforementioned PhD collaborations, each description is extended with a short paragraph concerning the activities within the collaborations in the respective application domain.

3.5.1 Wireless Sensor Networks

Wireless Sensor Networks Characteristics
Nowadays, sensor networks are widely deployed around the world. Typical sensors provide data for healthcare, energy management, environmental monitoring, etc. In the future sensors will become a part of critical infrastructures. In such scenarios the network operator has to monitor the integrity of the network devices, otherwise the trustworthiness of the whole system is questionable. Especially, when system security is based only on cryptography, inside attacks (e.g. insider) are a problem. Three general requirements were identified to be mandatory to thwart inside attacks on Wireless Sensor Networks (WSN), namely data authenticity, integrity, and freshness. Autonomic management, such as self-configuration and self-healing, are only feasible if nodes trust each other. Especially, for network routing protocols, such as the RPL protocol, trustworthy information is mandatory to protect the network and enable autonomic management functionalities. Even if wireless sensor networks permit reducing costs in terms of deployment and maintenance thanks to self-organization, they present several constraints compared with wired networks:

- **Limited Bandwidth:** The wireless physical medium presents a limited bandwidth which must be shared between devices from the same neighborhood. The available bandwidth depends on both the number of devices in the neighborhood and the amount of traffic to be forwarded, and this regardless of potential physical disruptions.
- **Energy Constraint:** The mobile devices are strongly constrained by the limited lifetime of their battery. Despite the battery improvements and increasingly energy efficient technologies, the nodes are particularly solicited especially when they forward traffic from other nodes.

- **Dynamic Topology:** Wireless sensor networks are spontaneously formed by mobile nodes without using fixed infrastructure. A WSN node is able to join or leave the network at any time. The topology may thus be highly changeable over the course of moving nodes and their operating state. This impacts the routing protocol that has to ensure regularly path maintenance.
- **Nodes Heterogeneity:** The WSN nodes may correspond to a large amount of devices: from laptop to smart sensor including mobile phone. These devices do not have the same physical and software properties but they have to establish a common interoperable network.
- **Limited Security:** The physical medium nature as well as the lack of central coordination make the wireless sensor networks more vulnerable than fixed infrastructures. Wireless transmissions can be captured without difficulty by a node in the neighborhood. Deny of service attack may be easily performed by a malicious node which can appropriate bandwidth or overload a neighbor node with a large amount of traffic to be routed .

WSN are required to operate through adapting to the environmental changes that the sensors monitor. Thus, a sensor network should be self-learning. Reliability is the ability to maintain the sensor network functionalities without any interruption due to sensor node failures. Sensor nodes may fail due to lack of energy, physical damage, communications problem, inactivity, or environmental interference. Therefore, a network should be able to detect the failure of a node and organize itself, reconfigure and recover from node failures without loosing any information. If sensor nodes are moving dynamically, routing protocols must be adaptive to these changes and should have self-healing and self-configuring functionality. Information should be persistent in spite of changes in network nodes. Low processing capacity of a node creates many challenges for routing packets throughout the neighboring nodes intelligently. Routing algorithms should be intelligent to choose minimum hop and minimum distance paths for data transfer. Thus, WSN self-management principles comply with the concept of IBM's autonomic computing, such as self-configuration, self-healing, self-protection, and self-optimization.

Routing Protocol for Low power and Lossy Networks (RPL)
The IETF ROLL (Routing Over Low power and Lossy networks) working group has proposed a new standard, RPL (Routing Protocol for Low power and Lossy Networks) based on IPv6 and specifically designed for strongly constrained environment. The devices are interconnected according to a topology structure which combines mesh and tree topologies: the Destination Oriented Directed Acyclic Graphs (DODAG). A DODAG graph is built from a root node which is the data sink of the graph. A network is composed of one or several DODAGs grouped into RPL instances which are associated to an objective function. An objective function computes the best path for a set of metrics or constraints. For instance, this function may minimize energy consumption or may simply compute the shortest path. These multiple instances enable the protocol to perform different optimizations, such as quality-of-service ones. A rank is associated to each node and corresponds to its location in the graph with respect to the root. The node rank is always increasing in the downward direction. The RPL packets may be forwarded according to three traffic patterns: point-to-point (P2P), multipoint-to-point (MP2P) and point-to-multipoint (P2MP). A set of new ICMPv6 control messages is defined to exchange RPL routing information. These routing information are exchanged according to timers based on the trickle algorithm. This algorithm optimizes the transmission frequency of control messages depending on the network state. It consists in increasing the frequency of messages when an inconsistency is detected. This allows faster recovery. On the other hand, the frequency of messages may be reduced when the network shows stability. The RPL protocol define loop avoidance and detection mechanisms as well as repair processes to ensure path maintenance.

Cyber-Physical Systems
A Cyber Physical System (CPS) is typically designed as a network, in which sensors, actuators, and embedded devices are networked to sense, monitor, and control the physical world. CPS applications use physical information collected by WSN and are able to react on environment changes directly. Thus, a CPS always needs a kind of autonomic management system. The use of WSN for a CPS consists mostly of self-organized sensor nodes, which are small devices that react on changes in the monitored environment. Each device is equipped with sensors for monitoring physical or environmental conditions. Today, CPS are used to cooperatively monitor large environments, such as critical infrastructures in gas and oil industries or even for battlefield surveillance in military scenarios or national infrastructures such as Smart Rescue System, Intelligent Building System, Smart Grid, Smart Highway and etc. Requirements for CPS are scalability, cost-effectiveness, and security. Mostly such systems have to operate in real-time and must be dependable and safe. To satisfy the robust and reliable characteristics of CPS, autonomic computing technologies are used.

Autonomic WSN Management Architectures
The basic IBM Autonomic Computing model only provides the conceptual guidance on designing self-managed systems and needs to be mapped to an implementable management and control architecture for Autonomic Networks. Service-oriented architecture (SOA) is an approach to build distributed systems that deliver application functionality as services to end-user applications or to build other services. SOA is mostly dependent on Web Services with standardized web technologies such as Web Services Description Language (WSDL), Open Grid Services Architecture (OGSA). As a result, it is not directly applicable to resource-constrained sensor nodes. Some initial ideas of using the concept of service semantics from SOA is presented in the Management Architecture for WSN called MANNA. In MANNA, all the management function units sit at the lowest level of management architecture. Different services can share the same functions, but still concern each individual given aspect based on the polices and network state obtained from WSN models. SOA can specially deal with WSN unique aspects such heterogeneity, mobility and adaptation, and offers seamless management integration in the wireless environments. Although the special features of SOA are marvelous, there is still a large amount of research challenges needed to address before the concepts of SOA can be appropriately applied into WSN. Policy-based management has presented its robust ability to support designing of self-adaptive decentralized management service in WSNs. Davy S. et al. [67] proposed an autonomic communications architecture that manages complexity through policy-based management by incorporating a shared information model integrated with knowledge-based reasoning mechanisms to provide self-governing behavior. Similarly, in MANNA, policies describe a set of desired behaviors of management components (e.g. manager and agent) for indicating the real-time operations. Based on polices, managers and agents can interact with each other in a cooperative fashion to achieve a desired overall management goal such as form groups of nodes, control network density, and keep the coverage of the WSN area.

Collaboration Activities
The collaborations **UniBwM-JUB-RPL** and **INRIA-JUB-RPL** focus on wireless sensor networks. The first collaboration investigates the possibility to secure the connections in an RPL network by using a standard Trusted Platform Module (TPM) [9]. In addition, the trustworthiness of sensor data in the context of cyber-physical systems is investigated in [19]. The second collaboration investigated the vulnerability of RPL networks by investigating the attacks against the network and identifying the key parameters which are required to detect the aforementioned attacks [68]. Additionally, they develop a mitigation strategy to reduce such attacks and to enable nodes to detect malicious parent nodes.

3.5.2 Cloud-based Services

Private Cloud, Community Cloud, Public Cloud, Hybrid Cloud
There exist several deployment models, differing in ownership, hosting location, management responsibilities, etc. which can be categorized as follows [69]:

Private Cloud: This infrastructure is typically provisioned for exclusive use by a single organization and could be hosted internally or externally and managed by the organization, a third party or a combination of those. Since these clouds are tailored to the needs of a single organization they require a significant effort to set up.

Community Cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers that have shared concerns (e.g., security requirements, policy and compliance considerations). It can be hosted internally or externally by one or more organizations in the community or a third party.

Public Cloud: The cloud infrastructure is provisioned for open use by the general public. The infrastructure is owned, managed and operated by a business, academic or government organization and are accessible through the Internet. Examples of public cloud service providers are Amazon AWS, Microsoft and Google.

Hybrid Cloud: This cloud infrastructure is a combination of two or more distinct cloud infrastructures (private, community or public) that remain unique entities, but are bound together through standardized or proprietary technology that enables data and application portability. This for example allows organizations to use public cloud computing resources to meet temporary needs.

Inter-Cloud
The Inter-Cloud is an interconnected global *cloud of cloud* and an extension of the Internet *network of networks* [70] on which it is based [71, 72]. The key concept of Inter-Clouds is that every single cloud has finite physical resources and that when the computational and storage demands exceed the available resources, the cloud is unable to satisfy all requests. The Inter-Cloud scenario addresses such situations by considering the individual clouds as an interconnected cloud by analogy with the Internet where a service provider to which an endpoint is attached, will access or deliver traffic from/to source/destination addresses outside of its service area by using Internet routing protocols with other service providers with whom it has a pre-arranged exchange or peering relationship. This allows the different clouds to share computational, storage or any kind of resource within the infrastructure of other clouds. The Inter-Cloud would be a mesh of cloud computing resources owned by multiple parties and interconnected and shared via open standards. This enables cloud service providers to maximize profit by serving more customers using existing infrastructure, while meeting SLA's by balancing load across other service providers during demand spikes. Furthermore, new revenue from other service providers can be generated by renting out unused capacity on an as-needed basis.

Collaboration Activities
The cloud based services are addressed by the collaboration **UT-UZH-DoS** which investigates solutions that detect and stop the malicious traffic before reaching the target. DDoS Network Protections (DNP) are examples of these solutions that redirect and filter the malicious traffic during attacks. An example of DNP is the online service CloudFlare. The planned approach for this research is to monitor targets, which are protected by a DNP solution, especially during DDoS

attacks. This can be achieved by long term monitoring of services known to be using DNP, or in a lab environment, where DDoS can be generated on-demand. The collaboration **UT-UniBwM-IDS** investigated architectures in the cloud-based environments [14], [15]. Furthermore, it is planned to approach to use clouds for managing vulnerability incidents (planned as a collaboration between INRIA and UniBwM in the second year).

3.5.3 Content-aware Routing

Content placement
Content Delivery Networks (CDNs) have been the prevalent method for the efficient delivery of rich content across the Internet. In order to meet the growing demand for content, CDN providers deploy massively distributed storage infrastructures that host content copies of contracting content providers and maintain business relationships with ISPs. Surrogate servers are strategically placed and connected to ISP network edges [73] so that content can be closer to clients, thus reducing both access latency and the consumption of network bandwidth for content delivery.

Current content delivery services operated by large CDN providers like Akamai [74] and Limelight [75] can exert enormous strain on ISP networks. This is mainly attributed to the fact that CDN providers control both the placement of content in surrogate servers spanning different geographic locations, as well as the decision on where to serve client requests from (i.e. server selection) [76]. These decisions are taken without knowledge of the precise network topology and state in terms of traffic load and may result in network performance degradation.

Previous work investigated a cache management approach with which ISPs can have more control over their network resources. Exploiting the decreasing cost of storage modules, the proposed approach involves operating a limited capacity CDN service within ISP networks by deploying caches at the network edges. These can be external storage modules attached to routers or, with the advent of flash drive technology, integrated within routers. Such a service can cache popular contents, specific to an ISP, and serve most client requests from within the network instead of fetching content items from surrogate/origin servers. Empowering ISPs with caching capabilities can allow them to implement their own content placement and server selection strategies which will result in better utilisation of network resources. In addition, there are economic incentives for an ISP to adopt this approach given that traffic on inter-domain links can decrease significantly. In order to deploy such an approach, new interaction models between ISP and CDN providers may need to be defined. These would address issues relating to the resolution of content requests and the exchange of necessary information, e.g. content items to cache, their popularity and size.

While the utilisation of network resources is affected by both content placement and server selection operations, previous work focused on the former. Commercial CDNs have been traditionally using centralised models for managing the placement of content in distributed surrogate servers. Complex algorithms are executed in an offline fashion for determining the optimal placement of content copies for the next configuration period (typically in the order of days). With the objective of keeping the cost and the complexity of the approach low, previous work investigated simple strategies that can be used by the ISP to decide on the placement of content copies in the various caching points according to user demand characteristics. Such strategies do not incur significant processing and communication overhead among distributed decision points and their functionality can thus be realised by commodity hardware components.

Routing in ICN

Information-Centric Networking (ICN) is a general term for emerging future Internet architectures that, in contrast to current host-centric architectures, focus on content as first class citizens to provide a network infrastructure service that is better suited to today's use. The main abstraction in all ICN architectures is the Named Data Object (NDO), which keeps its name regardless of its location and of how it is copied, stored, or communicated [77]. Shifting from host-based to content-based networking heavily impacts how data is discovered and routed.

To route an NDO request to a node that holds a copy of it, two general approaches can be distinguished in ICN. In the first approach, a Name Resolution Service (NRS), storing lower-layer storing locations, is queried. Using these locators, the request can then be forwarded using a topology-based routing protocol like IP. Using the second general approach, the request is directly forwarded in the network, without having to resolve the object name into lower-layer locators. This approach is also referred to as name-based routing. The latter approach is most frequently used in existing ICN architectures. It is important to note that name-based routing can also be applied in some steps of the name resolution process, as is the case when using DHT-based name resolution systems.

Within the name-based routing techniques in ICN, two general classes can be identified [78, 79]. A first group of techniques, analogous to routing in IP-based networks, do not rely on specific structures to construct and maintain the routing tables. This approach is, amongst others, applied in the Content Centric Networking (CCN) architecture [80]. Since the distribution of routing information is often based on flooding, this approach introduces a considerable amount of overhead. To limit the amount of overhead, a second group of techniques construct and maintain the routing tables based on a specific structure. For example, in the Data-Oriented Network Architecture (DONA) [81], the routers compose a hierarchical tree in which every router contains information about the data stored in all underlying nodes in the tree. Even though the amount of management overhead is limited using this approach, the root node has to contain information about all data in the entire network. The Publish-Subscribe Internet Routing Paradigm (PSIRP) architecture applies the structure of hierarchical distributed hash tables (DHTs) [82]. Using DHTs, the size of the routing tables is limited for every network node. However, this comes at the cost of longer paths when compared to tree-based approaches.

When comparing name-based routing to NRS-based approaches, it is clear that name resolution-based approaches allow more flexibility in terms of storage and caching location. The NRS can manage and provide locators for any available object copy, including nearby but off-path caches. On the other hand, name-based routing can reduce the overall latency and simplify the overall process by eliminating the name resolution step. Furthermore, the number of NDOs is considerably larger than the number of hosts in current host-based networks. This increases the complexity of ICN name resolution and routing, compared to today's IP routing and DNS name resolution. Therefore, efficient routing information aggregation and scalable name resolution are crucial for ICN to be benificial. This however remains an open research topic.

Collaboration Activities

The collaboration **UCL-iMinds-Cache** is focusing on a scenario where a large-scale ISP [24] leases caching capacity to multiple content or CDN providers. The objective of this work is to develop a new cache management approach that can allow an ISP to automatically allocate its caching capacity to CPs or CDNs, and to generate content placement configurations. The performance of the approach will be evaluated over a number of network parameters using real traces.

4 PhD Collaborations

One of the S.M.A.R.T. objectives for this WP in FLAMINGO is the integration of PhD students. In Section 4.1 we draft the collaborations within the FLAMINGO consortium to describe them in detail in Section 4.2. An overview of all fully integrated PhD students can be found in D8.1, including the co-supervisors and their affiliation.

4.1 PhD Student Collaborations

Joint research in FLAMINGO was made possible through the integration of PhDs into the project. Therefore a bottom-up approach was used to include also PhDs which are not necessarily paid by FLAMINGO, but imply research experience and up to date research approaches to the project. In the course of the first year of the project, the FLAMINGO PhD students and other partners in the consortium have been invited to develop collaborations. These have been identified during the FLAMINGO meeting that took place in February 2013. Additional collaborations were established during the FLAMINGO meeting in October 2013. All are graphically described in Figure 20.

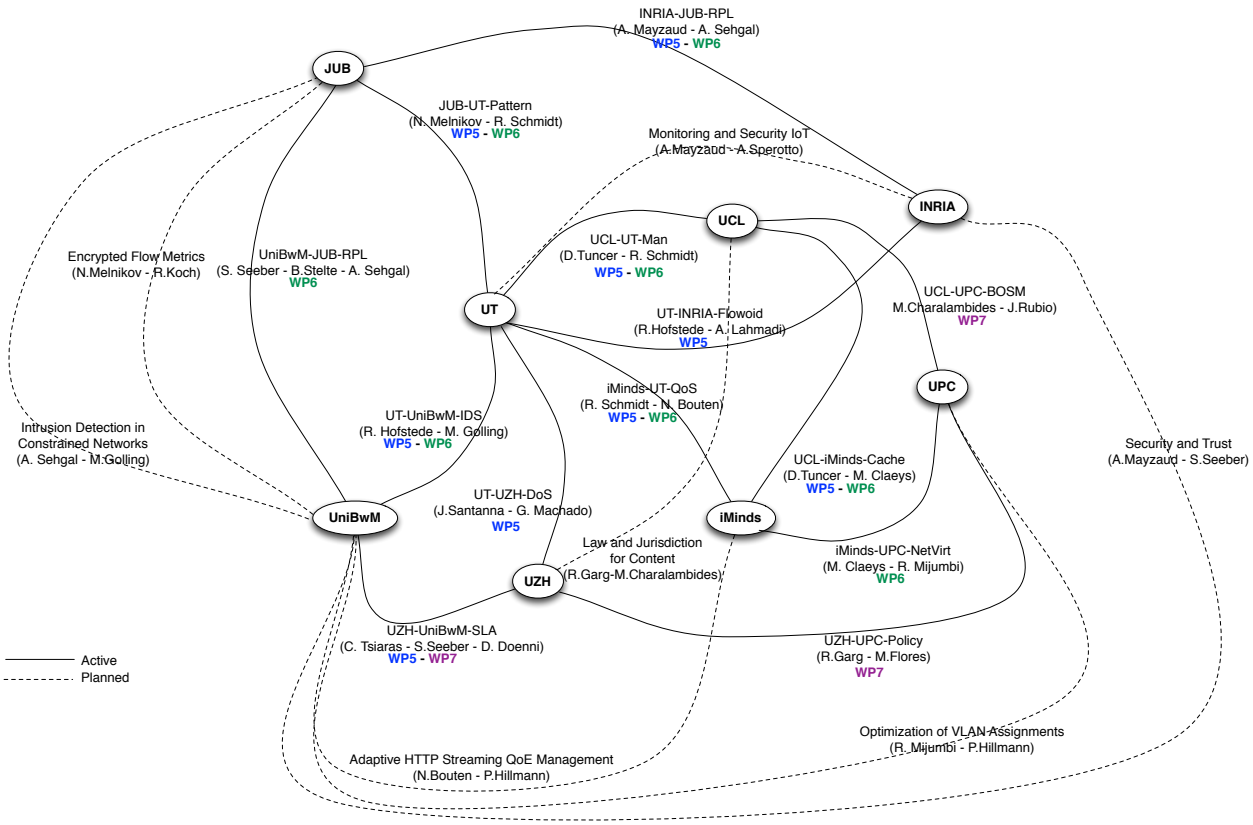


Figure 20: PhD collaborations map

Due to the fact that WP6 and WP5, *automated configuration and repair* and *network and service monitoring* are strongly connected Section 4.2 provides a description of the collaborations in both WPs with their corresponding acronyms used in Figure 20. In addition Table 8 gives an overview of the PhD students active in WP6 and it lists the collaborations that have been started among members of the consortium. These are also used in the further sections to establish the link to the collaborations if needed. WP7 specific collaborations also shown in Figure 20 are described in deliverable D7.1.

Table 8: PhD students active in WP6, and collaborations with consortium partners

Name	**Affiliation**	**Collaborations**	**Acronym**
Anthéa Mayzaud	INRIA	JUB	INRIA-JUB-RPL
Nikolay Melnikov	JUB	UT	JUB-UT-Pattern
Rick Hofstede	UT	UniBwM, INRIA	UT-INRIA-Flowid UT-UniBwM-IDS
José Jair C. de Santanna	UT	UZH	UT-UZH-DoS
Anuj Sehgal	JUB	UniBwM, INRIA	INRIA-JUB-RPL UniBwM-JUB-RPL
Vaibahv Bajpai	JUB	iMinds	
Mario Golling	UniBwM	UT	UT-UniBwM-IDS
Christos Tsiaras	UZH	UniBwM	UZH-UniBwM-SLA
Guilherme Sperb Machado	UZH	UT	UT-UZH-DoS
Daphne Tuncer	UCL	iMinds, UT	UCL-iMinds-Cache UCL-UT-Man
Maxim Claeys	iMinds	UCL, UPC	iMinds-UPC-NetVirt UCL-iMinds-Cache
Niels Bouten	iMinds	UT	iMinds-UT-QoS
Sebastian Seeber	UniBwM	JUB, UZH	UZH-UniBwM-SLA UniBwM-JUB-RPL
Daniel Dönni	UZH	UniBwM	UZH-UniBwM-SLA
Ricardo Schmidt	UT	JUB, UCL, iMinds	iMinds-UT-QoS UCL-UT-Man JUB-UT-Pattern
Björn Stelte	UniBwM	JUB	UniBwM-JUB-RPL
Rashid Mijumbi	UPC	iMinds	iMinds-UPC-NetVirt

4.2 Description of the collaborations

During the first year of the FLAMINGO project several PhD collaborations have been established. This section describes these more in detail. Because these collaborations involve elements from WP5 (*Network and Service Monitoring*) and WP6 (*Automatic Configuration and Repair*), this section appears in deliverable D6.1 and D5.1.

4.2.1 Linking Network Usage Patterns to Traffic Gaussianity Fit (JUB-UT-Pattern)

The Gaussianity of traffic aggregates is a desirable characteristic in the domain of network traffic modelling due to the wide adoption of Gaussian models. Past works have been extensively researched the Gaussian property of traffic aggregates, its advantages for proposing traffic models and how this can be disturbed by traffic bursts. To the best of our knowledge, however, never a work has tried to connect traffic Gaussianity, or lack thereof, to network usage patterns. This knowledge would be valuable in understanding the limitations of current traffic models in presence of certain traffic. Therefore, the aim of this collaboration is to find out the potential connections between traffic bursts and poor Gaussianity , and also to point out what sort of host and/or application traffic pattern creates such disruptions. Preliminary results show that Gaussianity fit can be directly connected to presence or absence of extreme traffic bursts. The results also show that even in a more homogeneous network (i.e., hosts with similar access rates) we can identify extreme traffic bursts that might ultimately compromise Gaussianity fit. Figure 21 describes how the approach out of the collaboration map the evaluation criteria which we proposed in Section 3.2. A paper on the topic of this collaboration is currently under review for PAM 2014 [10].

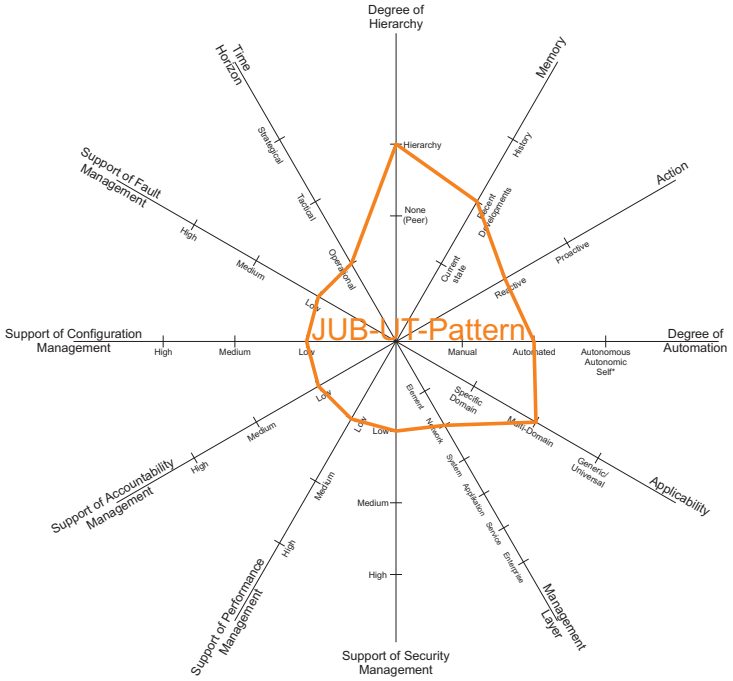


Figure 21: Evaluation of JUB-UT-Pattern

This work contributes to WP5 since it is based on the collection of packet-level traffic traces from many different locations around the globe.

The collaboration also contributes to WP6, since it aims at establishing a “rule-of-thumb” to estimate whether traffic is Gaussian or not, based on conclusions from data analysis such as hosts behavior and applications usage..

4.2.2 Energy-aware Traffic Management (UCL-UT-Man)

UCL has developed an adaptive resource management approach, which can reduce the energy consumption of core IP networks [16]. The approach is based on the reconfiguration of traffic splitting ratios at the edges of the network so that traffic is distributed over a subset of router line cards, while unused ones enter sleep mode. UT has been working on methods to estimate the required bandwidth of network links based on flow-level traffic measurements [83].

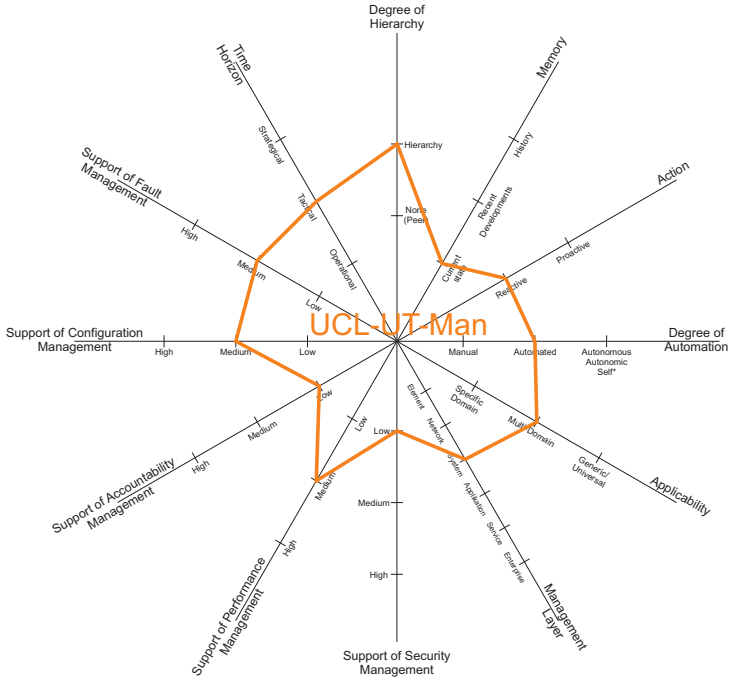


Figure 22: Evaluation of UCL-UT-Man

The goal of the collaboration between UCL and UT is to propose a decentralized system for traffic management supported by a link dimensioning approach for better reallocation of link resources. The system takes decisions on traffic splitting within a backbone network focusing on energy efficiency by reducing the number of required line cards. Currently, the system assumes that traffic averages represent the required capacity to be allocated per flow. Link dimensioning would allow the management system to have better estimations of required capacity, e.g., considering quality of service metrics, which would ultimately be used on the reconfiguration of traffic splitting.

Figure 22 describes how the presented approach map the evaluation criteria which we proposed in Section 3.2.

This work is being carried out in the scope of WP5 and WP6, addressing the following aspects: The collaboration contribute to the collection of anonymised monitoring data. This data comprises packet-level traffic captures used to validate the link dimensioning procedure and also to produce synthetic traffic to be used in simulations of the management system. The need for synthetic traffic is due to difficulties in acquiring traffic captures of an entire backbone network. This aspect contributes to WP5. This activity extend previous work on adaptive resource management and enhance the energy-awareness control loop, by providing better link load estimates. These can allow the control loop to make more energy-friendly reconfiguration decisions. The performance of the approach will be evaluated in terms of the number of line cards that can enter sleep mode, as well as link utilization. This aspect contributes to WP6.

4.2.3 Intrusion Detection Systems (UT-UniBwM-IDS)

This joint research activity is a collaboration between UT and UniBwM. Intrusion detection is nowadays commonly performed in an automated fashion by IDS [84]. Several classifications for IDSs are common. One of these classifications focuses on the kind of data that is used for performing intrusion detection. The first class of IDSs mainly uses packet headers (flows) for intrusion detection. While these flow-based IDSs have a high-performance and are usually little privacy-intrusive, they are typically affected by a high number of undetected attacks (false negatives; see Figure 23).

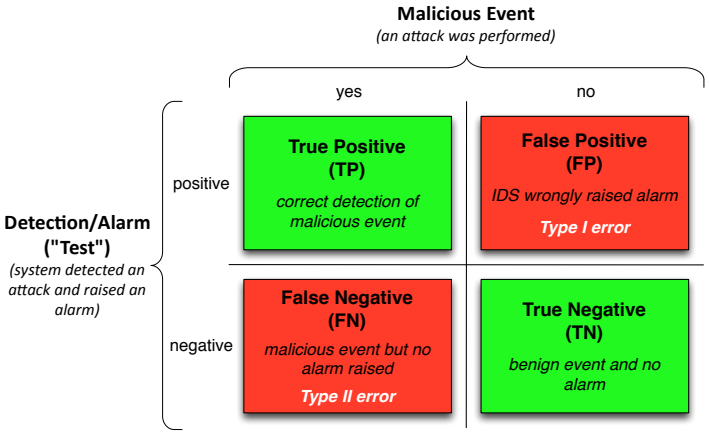


Figure 23: Categories of Alarms ("Confusion Matrix")

In contrast to flow-based IDSs, payload-based IDSs are capable of performing extensive layer-7-detection (and, therefore, have a lower false negative rate), but at the expense of a much higher system requirements as well as a violation of privacy [85]. Given these observations, performing intrusion detection in high-speed networks is a challenging task. While many payload-based IDSs are working well at the backend of service provider networks, the backbone is often characterized by communication links with high-speed connections and thus requires well equipped IDS in order to be capable of handling 100 Gbps or more, for example [86]. Within this collaboration, it is planned to create a framework for distributed intrusion detection in high-speed networks by combining especially flow-based and payload-based intrusion detection. As already stated, in addition to monetary aspects, legal issues in general and privacy issues in particular are also important reasons, why payload-based IDS are rarely deployed in high-speed networks today [85].

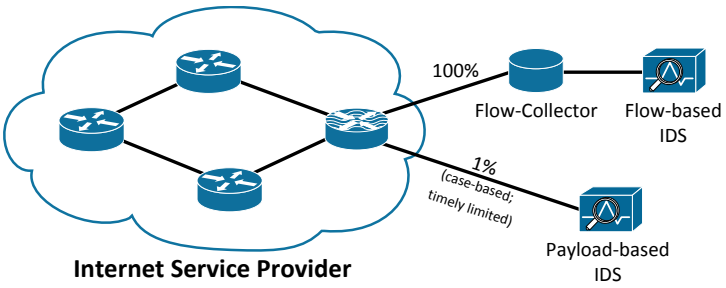


Figure 24: Simplified Scenario

In order to overcome these disadvantages, this collaboration tries to make use of both approaches (both flow-based and payload-based intrusion detection) in a multi-layered approach. As depicted in Figure 24, the approach is centered around the ideas that (i) the first layer comprises flow-based intrusion detection, which performs detection based on the entire packet stream (100%) and that (ii) depending on the result of the flow-based detection, the payload-based IDS is activated for a certain period of time to investigate the anomaly of the flow-based IDS in more detail (1%) - in order to verify or falsify the result of the flow-based IDS. As network attacks can last shortly and a switch has to be made from flow-based to packet-based detection, detection has to be performed in real-time.

This work is being carried out in the scope of WP5 and WP6, addressing the following aspects: The collaborative work contributes mainly to WP5: Network and Service Monitoring by performing multi-layered IDS. Especially novel solutions for IDS is addressed with this collaboration. As the long-term goal of this collaboration is also to link different managers with each other. Regarding this objective, in [13] an Evaluation of State of the Art IDS-Message Exchange Protocols was already performed.

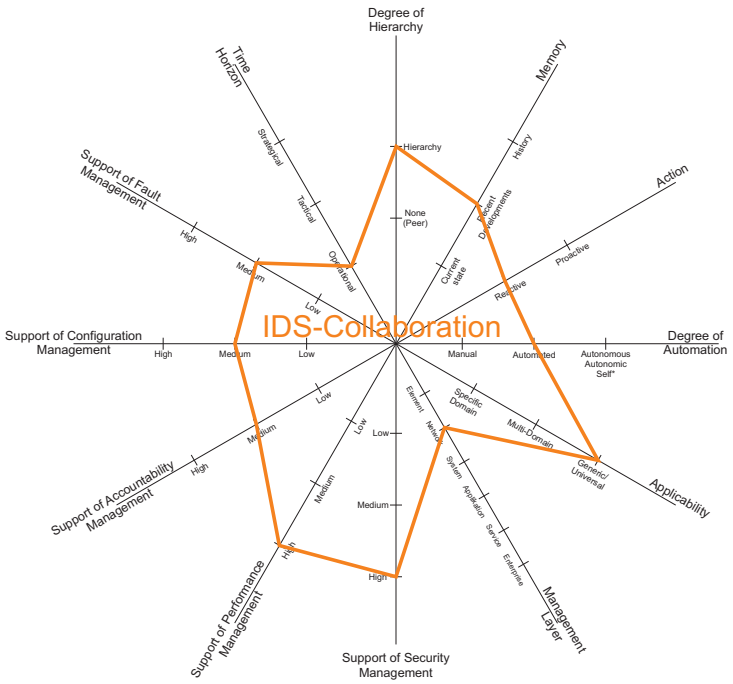


Figure 25: Evaluation of UT-UniBwM-IDS

For WP6, [13] also contributed by developing an inventory of IDS-Message Exchange Protocols. By using cloud-based solutions (as described in [14, 15]), requirements for cloud-based Services have been investigated and architectural approaches specific for this application domain have already been partially developed. As it is planned to develop an ”automated” architecture the collaboration addresses an additional objective. The approach is evaluated in Figure 25 by the evaluation criteria which we proposed in Section 3.2.

4.2.4 Towards A Trust Comupting Architecture for RPL in Cyber Physical Systems (UniBwM-JUB-RPL)

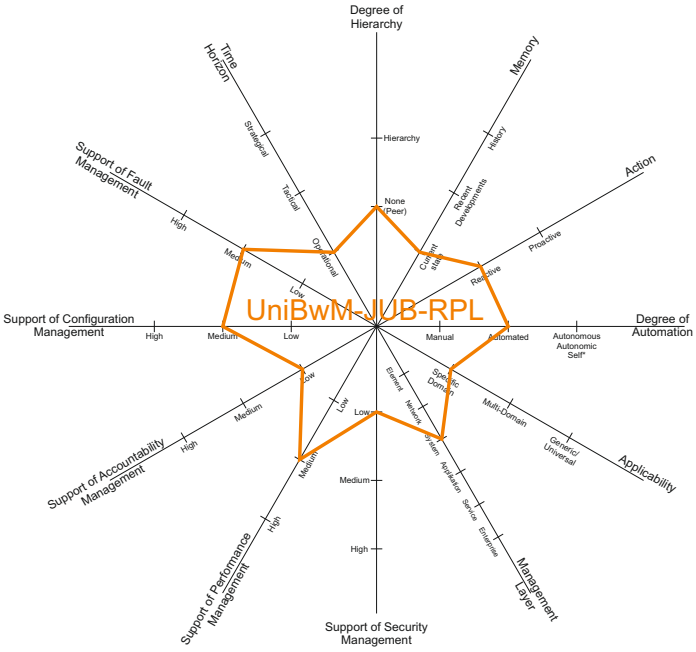


Figure 26: Evaluation of UniBwM-JUB-RPL

Cyber Physical Systems (CPSs) are widely expected to be formed of networked resource constrained devices. In order to suit the constraints of such networks, the Internet Engineering Task Force (IETF) developed the Routing Protocol for Low power and Lossy Networks (RPL) and Low-power and Lossy Networks (LLNs). Security in CPSs is important for maintaining the integrity and privacy of data, while also improving network resiliency to attacks. Even though RPL provides support for integrity and confidentiality of messages, details regarding key management and signatures are not covered. Since complexity and size is a core concern in LLNs, off-loading the security features to a Trusted Platform Module (TPM) makes it possible to include sophisticated security provisions in an RPL implementation.

This collaboration develops mechanisms to use the security mechanisms of a TPM in order to secure the communication in an RPL network. The design of a trust establishment and key exchange mechanism around the implied trust of a TPM to provide keys for secure RPL nodes, is a main task of this research. With this approach, the usage of a TPM on Resource Constrained Devices reduces the processing load on the main processor. The goal of this examination is the prevention of the dissemination of misleading routing information, which can affect the availability of the whole network. Figure 26 describes how the presented approach map the evaluation criteria which we proposed in Section 3.2. As a next step, the previous developed idea will be deployed on real hardware devices to evaluate the solution in comparison to other approaches. This is necessary to prove the existing simulation results.

The collaboration fits within WP6, since it develops a mechanism, specifically, targeted to RPL networks, to secure the communication. This approach is applicable in RPL networks which are used in wireless sensor networks.

4.2.5 Flowoid: a NetFlow/IPFIX probe for Android-based devices (UT-INRIA-Flowoid)

Analysis of the network behaviour of applications running on a smartphone device resor networks. quires the collection of information about data leaving the device and where it is sent. Cisco's NetFlow and the more recent IPFIX are flow export technologies that have seen a rapid adoption and widespread integration in many campus, enterprise and backbone networks. To be able to export and analyse mobile device characteristics (such as its location at the moment of certain network activity), the NetFlow and IPFIX protocols have to be extended. The flow exporter, flow collector and analysis application need to be aware of these extensions as well. The work in this collaboration has been divided between INRIA and UT. On the one hand, INRIA is responsible for developing a flow exporter tailored to Android devices. On the other hand, UT is responsible for the flow collector and analysis application. The approach is evaluated in Figure 27 by the evaluation criteria which we proposed in Section 3.2. The major achievements of the collaboration are:

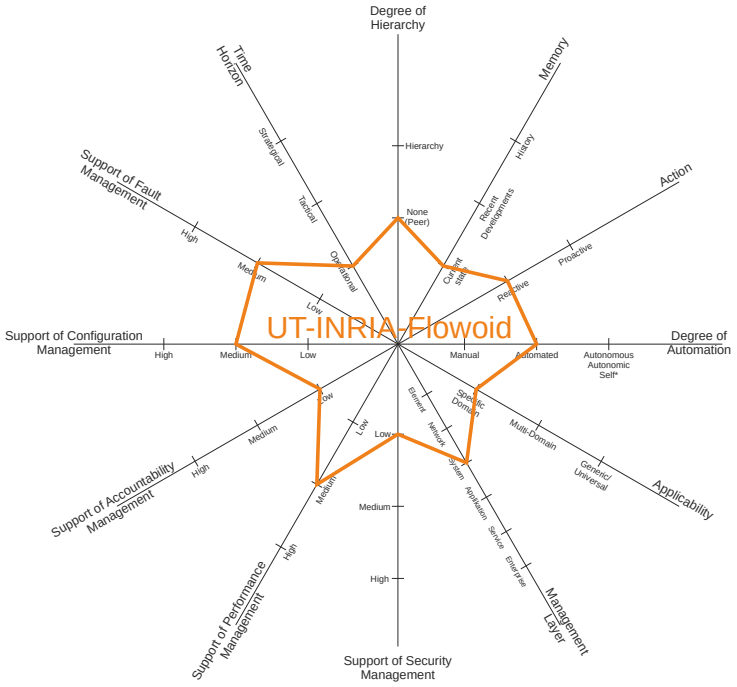


Figure 27: Evaluation of UT-INRIA-Flowoid

- Development of a NetFlow and IPFIX metering process for Android devices;
- Extension of nfdump/Nfsen and SURFmap with location support;
- IETF draft describing a set of information elements for IPFIX metering process location [7].

The following aspects of this collaboration fall within WP5. In this work, INRIA has developed Flowoid, a NetFlow and IPFIX metering process tailored to Android devices. The probe associates

geolocation data with each observed network flow, consisting of the GPS coordinates of the mobile device, among others. This information is exported together with the traditional fields defined in the NetFlow and IFPIX: IP version, source and destination addresses, the number of exchanged bytes, the type of protocol, the number of exchanged packets, the source and destination ports, and the duration of a flow. In addition, it contains 7 additional fields that denote the identifier of the device: the identifier of the localization method, a timestamp, the integer part of the latitude, the decimal part of the latitude, the integer part of the longitude, and the decimal part of the longitude. UT has extended the state-of-the-art flow collector nfdump/NfSen with location support, allowing us to analyse the flows exported by the Flowoid probe. In a previous work UT has developed a network monitoring tool based on the Google Maps API, named SURFmap [87], which adds a geographical dimension to flow data and displays the data on a map. Since SURFmap [87] already supports network traffic geolocation (i.e. adding physical locations of hosts to network data), this tool has been extended to visualize the locations of devices on a map. This will allow us to visualize network traffic of mobile device with respect to devices locations.

The following aspects of this collaboration fall within WP6. Relating the location information of a device to its network traffic can be beneficial to many network management and measurement applications, including traffic profiling, anomaly detection and provider-independent measurements. The developed approach allows us to better understand Android apps regarding their network flows and usage and it proves to be promising for the automated configuration of mobile networks. When Metering Processes are running on devices with a (frequently) changing physical location, data analysis applications may need to be aware of these movements since they are likely to affect the behavior of the network in terms of routing, throughput, etc. Thus, configuration policies and actions could be applied to adapt the network according to the observed locations and maintain an acceptable quality of service of running applications on the user's devices. For example, knowing the location of a device at a moment of certain network activities could be used to dynamically reroute its traffic to closer data sources.

4.2.6 Flow-based Traffic Measurements for In-Network Video Quality Adaptation (iMinds-UT-QoS)

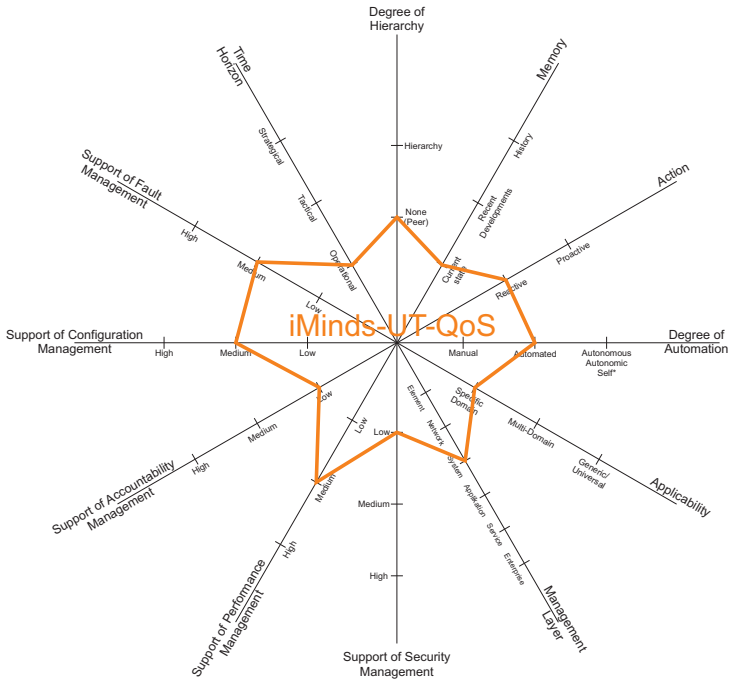


Figure 28: Evaluation of iMinds-UT-QoS

HTTP Adaptive Streaming (HAS) services allow the quality of streaming video to be automatically adapted by the client application in face of network and device dynamics. A major obstacle for deploying HAS in managed networks, is the purely client-driven design of current HAS approaches, which leads to excessive quality oscillations, globally suboptimal behavior, and the inability to enforce management policies. These challenges can be tackled by steering the quality selection from within the network. iMinds already deployed a distributed in-network management heuristic, which is able to reduce the number of switches with a factor 5 and increase the quality up to 30%. One of the shortcomings of the cur-

rent deployment, however, is the assumption of a static bandwidth for each link and the absence of cross-traffic. To overcome this issue, this collaboration will use flow-based measurements to measure and predict the per-link throughput, which is available for HAS traffic. Using these predictions the in-network video quality adaptation can divide the resources among the different HAS clients based on a provider's policy.

This work contributes to WP5 because real world traffic traces will be collected and used to validate the link dimensioning approach.

This work contributes to WP6 in the following: The goal of this collaboration is to develop a distributed algorithm/heuristic, which is able to divide the resource among the various HAS clients subject to a provider's policy. Using the measurements provided by WP5, each agent is able to perform a local optimization based on the throughput predictions. The different distributed agents share this network information and their local decisions with each other to be able to automatically react to changes in the network environment.

Using the measurements and predictions on the current and future cross-traffic, the agents can make estimations on the residual bandwidth that can be shared among the different HAS sessions. These local estimations and the shared network information serve as input to the algorithm which limits the quality of the HAS sessions crossing the managed resources. This leads to a more stable quality selection at the client, since oscillations due to changed network environments are avoided. Figure 28 describes how the presented approach map the evaluation criteria which we proposed in Section 3.2.

4.2.7 Study of DODAG Inconsistency Attacks in RPL Networks (INRIA-JUB-RPL)

The growing interest for the IoT has resulted in the large-scale deployment of Low power and Lossy Networks, such as wireless sensor networks and home automation systems. A new routing protocol called RPL for IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) has been specifically designed by the IETF Routing Over Low Power Lossy Networks (ROLL) working group to deal with these requirements.

RPL forms a tree like topology for routing packets, called a Destination Oriented Directed Acyclic Graph (DODAG). In order to detect possible loops, also referred to as DODAG inconsistencies, RPL uses IPv6 header options to track the direction of the packet and any rank errors. Specifically, the O-Bit option is used to track direction of the packet, i.e. upwards or downwards in a tree. If an upwards packet is received from a node with a rank lower than the current node, an inconsistency is detected. As such, the node will set the R-bit option, used to track rank errors, and forward the packet. If the next receiving node also detects an inconsistency in the direction of the packet, and the R-bit option is also set, this node will drop the packet and reset the trickle timers used by RPL.

Such a reset of trickle timers leads to an increase in the number and frequency of control packets being sent and received in the DODAG, thereby also impacting the already low available energy. A malicious node can create artificial DODAG inconsistencies by manipulating these IPv6 header options, thereby leading to increased overhead, denial of service and even black-hole attacks that are hard to detect.

The objective of this joint scenario is (1) to establish a state-of-the-art overview about security attacks against RPL networks, (2) to identify the key parameters that are required to detect these attacks, (3) develop a mitigation strategy to reduce the effect of such attacks, (4) develop an approach for children nodes to detect when a parent node might be malicious and (5) to experiment and evaluate the developed solutions.

A review of the state-of-the-art in RPL security has been conducted and this has led to the identification of DODAG inconsistency attacks that can lead to increased overhead and energy consumption, denial of service and even black-hole attacks.

Based on this, scenarios were constructed to study the performance of the RPL network when such attacks are carried out. Via an implementation in Contiki, it was identified that the mitigation strategy proposed by RPL, which involves ignoring packets with the appropriate IPv6 header after a fixed threshold is reached, uses an arbitrary value for the threshold. A new function that dynamically scales this threshold was developed to improve performance of the network while under attack. A comparitive study between the (1) no threshold, (2) fixed threshold and (3) dynamic threshold scenarios has been performed. An approach to counter the black-hole attack has also been developed. This approach allows nodes to periodically enter a promiscuous mode in order to verify that the parent is not modifying packets in malicious ways, i.e. setting incorrect IPv6 header options. When such manipulation is detected, it is countered by blacklisting the parent, informing the neighborhood and triggering a rebuild of the DODAG. Figure 29 describes how the described approach map the evaluation criteria which we proposed in Section 3.2. An implementation of this algorithm in Contiki is underway, following which an evaluation will be performed. A paper on the topic of this collaboration is currently under review for IPSN2014 [12]. This research contributes to WP5 and WP6: The collaboration is developing methods to identify malicious nodes that might attempt to infiltrate and negatively impact an RPL based network. Since there are many possible attack vectors, there are different approaches being designed to identify and subsequently mitigate the effect of these attacks. Upon successful identification, we are also working on an approach to blacklist such malicious nodes from being able to rejoin the network. These aspects of the collaboration fits within WP5.

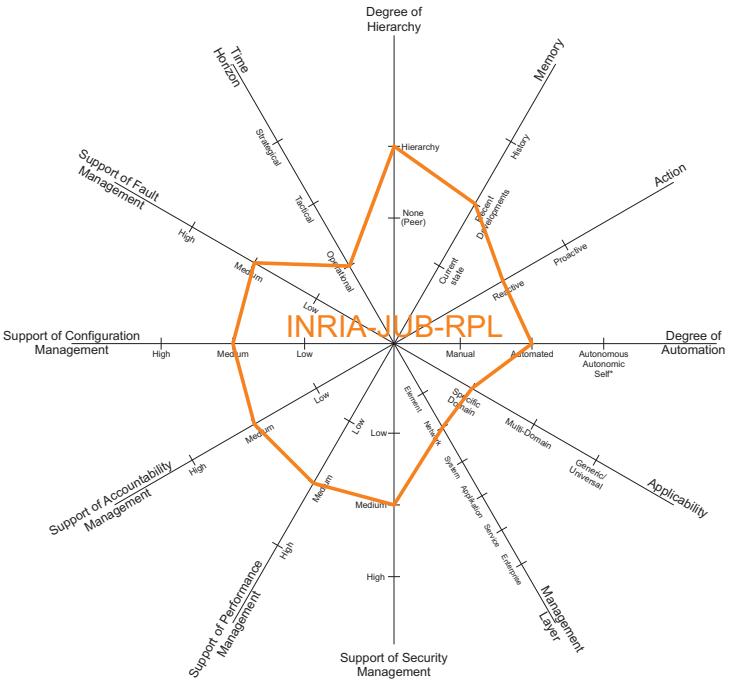


Figure 29: Evaluation of INRIA-JUB-RPL

The collaboration approaches for RPL are being developed and tested on the IEEE 802.15.4 + 6LoWPAN platform, which is expected to form the basis of the Internet of Things. Similarly, all the work is being currently tested on the TelosB network node, which is a resource constrained device used commonly in the IoT and WSN areas. In the future, the collaboration is also expected to investigate resource constrained devices. Since resource constrained devices need targeted solutions for reconfiguration, this collaboration also contributes to WP6.

4.2.8 SLA Fulfillment Mechanism (UZH-UniBwM-SLA)

One research goal of this research activity is the definition of a mechanism that detects if an Service Level Agreements (SLA) is violated in the context of a voice service over mobile networks. The research is motivated by the need for an SLA violation detection mechanism to identify that a QoS-guarantee is fulfilled or not. The need for this mechanism derives from the Auction-based Charging User-centric System (AbaCUS) [88]. Furthermore, this joint research activity aims to determine suited actions in respect to charging when a violation is detected. Facilitating the first goal on the level of traditional circuit-switched mobile phone calls would demand insight in a Mobile Network Operator's (MNO) infrastructure. Since this is currently not possible, the decision to focus on Voice-over-IP (VoIP) services over mobile networks has been taken. To our knowledge, nowadays there is not a QoS related metric for VoIP services over mobile networks. Thus, this work of SLA Fulfillment Mechanism aims to provide the respective metric as well as a prototype of the evaluation mechanism, which highly contributes to creating a blueprint of metrics in WP6 (see also Figure 30)

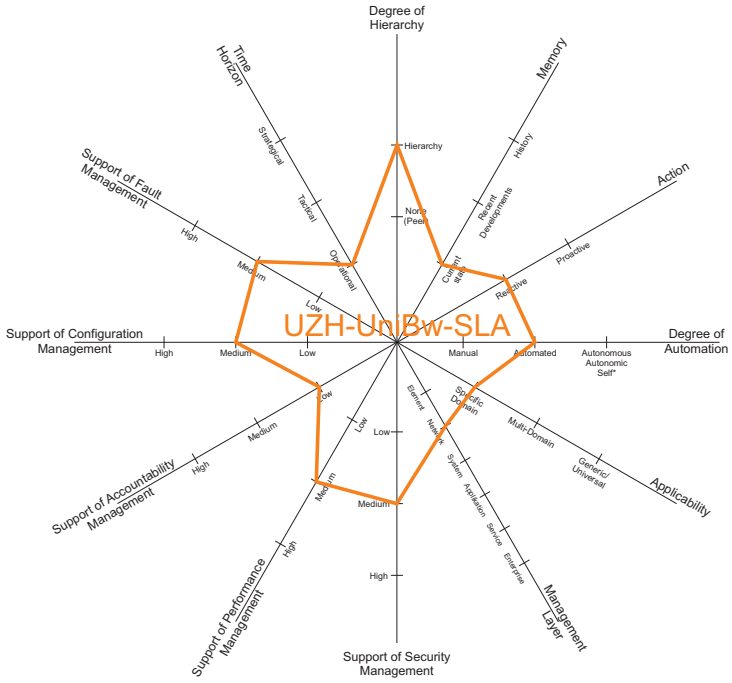


Figure 30: Evaluation of UZH-UniBwM-SLA

In respect to the WP5, the joint research activity aims to monitor the environment of the participants of an VoIP call which establishes the connections via a mobile network. Additionally the statistics of a VoIP call are monitored if accessible. This data will be used to identify if an SLA violation has taken place during the last VoIP call. In respect to the used end device the mobile signal-strength and the battery level will be monitored. These monitored values will help to identify possible SLA violations.

Furthermore, for the needs of the SLA Fulfillment Mechanism joined research project and the Value of Service (VoS) PhD project (Daniel Dönni), a measurement application will be designed and implemented. The application will be developed for the Android platform and allow the end-user to determine the connectivity, one-way delay and loss, round-trip delay and loss, delay variation, loss patterns, packet reordering, bulk transport capacity, link bandwidth capacity, and packet duplication metrics. The measurement application will use UDP packets to determine the above metrics, except for the bulk transport capacity which relies on TCP. Furthermore, novel metrics will be developed to capture price and price-performance aspects of IP networks. Details are subject to current research. The measurement application will be equipped with a graphical user interface which allows the end-user to specify measurement parameters. In particular, it will be possible for the end-user to define a measurement schedule which contains a list of metrics that shall be measured. For each metric it will be possible to specify the following:

- Whether a single measurement or a series of measurements shall be conducted
- What destination address shall be used
- What payload size shall be used

Depending on the metric, additional configuration options might be provided.

For the measurement application to function correctly, two additional components will be implemented. The first one is the measurement server which acts as a counterpart to the measurement application. Its main responsibility is to accept and reflect packets submitted by the measurement application. The second one is the measurement database which is responsible for storing the measurement data. The database will also contain a suitable schema for storing prices offered by network operators to capture pricing aspects as well.

All three components of the measurement platform will be implemented adhering to best-practice design and coding standards. This includes but is not limited to the definition of suitable application components, the implementation of unit tests using JUnit, as well as a proper documentation using JavaDoc. Furthermore, the measurement application will be made available in Google play, such that it can be easily downloaded and used by the research community.

The collaboration between the members of the SLA Fulfillment Mechanism and the Value-of-Service (VoS) member also contributes to WP7 by proposing business and regulation actions in case of SLA violation identification and price-performance metrics for IP networks. Additional details can be found at D7.1, Section 4.7 and Section 4.9.

4.2.9 Cache Management (UCL-iMinds-Cache)

Current content delivery services operated by large Content Delivery Networks (CDN) providers can exert enormous strain on Internet Service Provider (ISP) networks. This is mainly attributed to the fact that CDN providers control both the placement of content in surrogate servers spanning different geographic locations, as well as the decision on where to serve client requests from (i.e. server selection). These decisions are usually taken by using only limited information about the carrier networks, and this can adversely affect network usage.

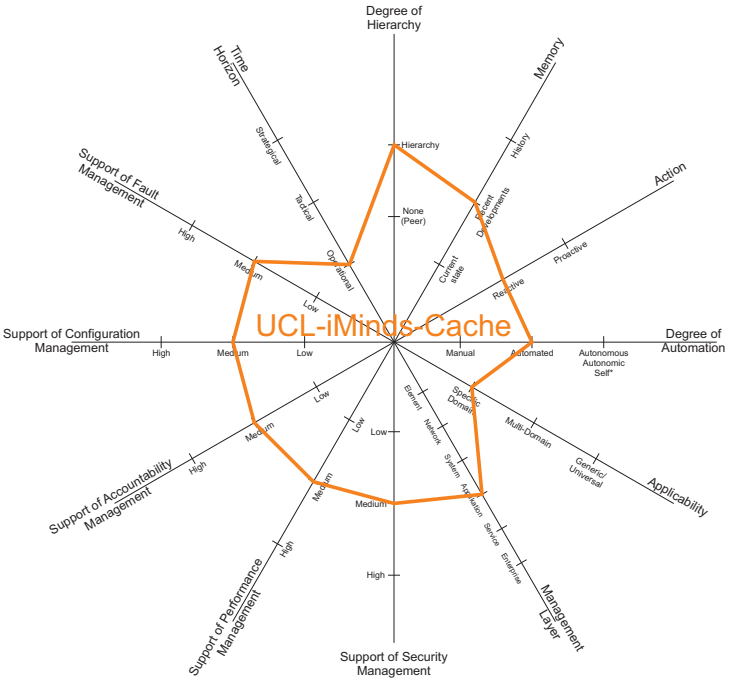


Figure 31: Evaluation of UCL-iMinds-Cache

UCL has developed an approach by which ISPs can have more control over their resources [24]. This involves operating a limited capacity CDN service within ISP networks by deploying caching points in the network. Empowering ISPs with caching capabilities can allow them to implement their own content placement and server selection strategies which will result in better utilization of network resources. The work has investigated content placement strategies that can be used by the ISPs to manage the placement of content items in the various network caching locations according to user demand characteristics.

In this joint research activity, UCL and iMinds are extending the scenario previously considered by focusing on the case where a large-scale ISP leases caching capacity to multiple content providers

(CP). The objective of this work is to develop a new cache management strategy that can be used by the ISP to decide on the cache capacity allocation and content placement configuration that can minimise its network resource usage given the capacity requirements of each CP and the characteristics of the user demand. Figure 31 describes how the described approach map the evaluation criteria which we proposed in Section 3.2.

Initial work has mainly focused on formally modelling the problem as an Integer Linear Program (ILP) problem, forming a common basis for the development of more simple and lightweight heuristic algorithms. The performance of the different heuristics will be evaluated based on realistic traffic traces provided by iMinds and will be compared according to several parameters.

This work is being carried out mostly in the scope of WP6. In the context of the proposed scenario, the cache management approach to be developed will implement a control loop that automates the configuration of in-network caching points. The performance of the approach will be evaluated over a number of parameters using real traffic traces. The collection of such traces contributes to WP5.

4.2.10 Management of Virtualized Networks (iMinds-UPC-NetVirt)

This joint research activity is a collaboration between iMinds and UPC. The work focuses on virtual network embedding. In virtual network embedding, a Virtual Network Provider (VNP) acts as a mediator between service providers (SPs) and infrastructure providers (InPs). Virtual network requests are launched by the service providers and requests contain requirements on node and link capacities.

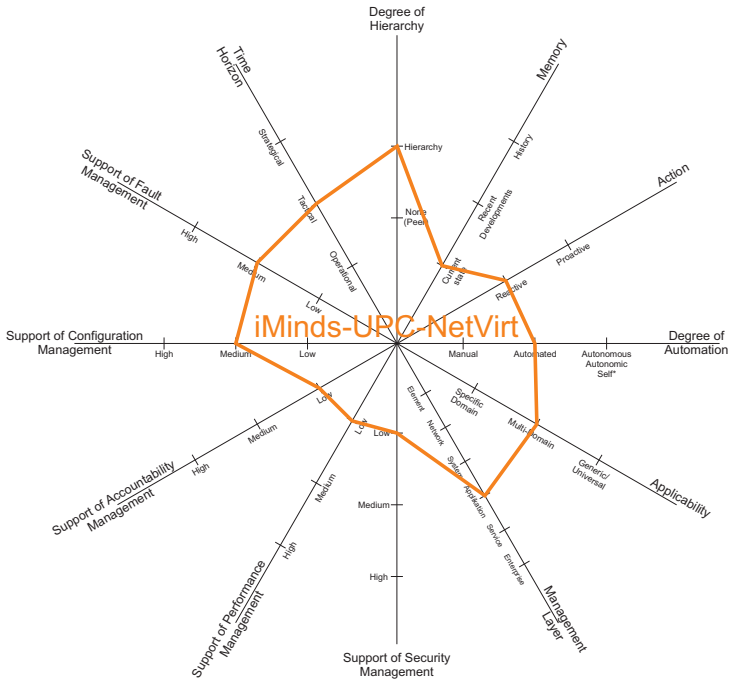


Figure 32: Evaluation of iMinds-UPC-NetVirt

Service providers target to receive a virtual network, fulfilling the request, that minimizes the embedding costs. The Virtual Network Provider reserves substrate resources from the infrastructure provider to be able to embed the virtual networks. Embedding solutions are based on the resource capacities requested by the service providers. Actual load will however vary over time, leading to situations where a lot of substrate resources remain unused.

The first goal of this collaboration therefore is to develop a dynamic embedding algorithm, able to dynamically adapt the embedding solution to the actual demands, perceived by network monitoring. This will optimize substrate resource usage and increase the acceptance rate of virtual network requests. For this purpose, the researchers will apply a multi-agent reinforcement learning approach. This research has resulted in a paper, currently under review for NOMS 2014 [11].

For the next steps of this research, the collaborations will take a broader look at the network virtualization problem by considering a service provider point of view. Instead of embedding a virtual

network topology onto a substrate network, the researchers will consider an additional step where the virtual network topology is constructed based on the service requirements. Figure 32 describes how the presented approach map the evaluation criteria which we proposed in Section 3.2. As a first step, a catalog of service enablers and requirements that can be considered for embedding is being compiled. Based on this catalog, the researchers will narrow down the specific problem definition.

The goal and approach of this collaboration lies within WP6 (see Figure 32). This is clearly visible when considering the dynamic embedding algorithms for automatically identifying the optimal network configuration that the collaboration is developing.

4.2.11 TraceMan-based Monitoring of DoS attacks (UT-UZH-DoS)

Distributed Denial of Service attack (DDoS) is one of the major threats to the Internet. In general, this kind of attack aims to deny the ability of a host (target) to respond to legitimate network traffic. Especially, when a DDoS is based on exhausting network resources, it is very difficult, and often ineffective, to mitigate with an on-premise solution (in which a target itself stops the attack), since the effect of the attack will also affect all the services and users in the same network.

In order to attempt to mitigate this kind of attacks, solutions that detect and stop the malicious traffic before reaching the target are considered. DDoS Network Protections (DNP) are examples of these solutions that redirect and filter the malicious traffic during attacks. An example of DNP is the online service CloudFlare[8].

DNP solutions have been adopted by a wide range of companies, and it seems also by companies that exchange a considerable amount of private data with customers, such as banks. However, few knowledge is available on the practical use of DNP. Questions that raise are, for example, if these DNP solutions redirect malicious traffic to third-party domains, and, if so, where the traffic is redirected to, and what is – if any – the impact of employing DNP. Therefore our goal is to understand how DNP solutions work in practice, if they are reliable and what it means for the end users when their are handled by a DNP.

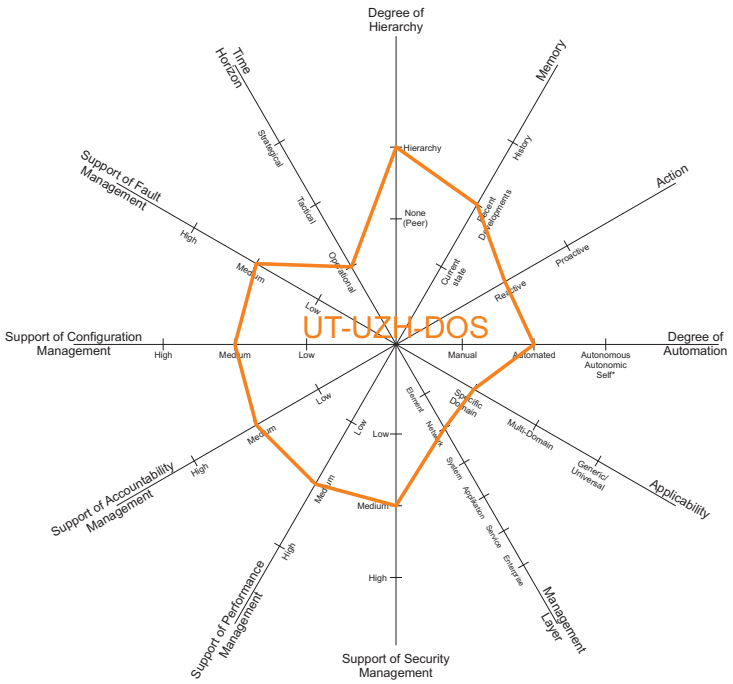


Figure 33: Evaluation of UT-UZH-DoS

The planned approach for this research is to monitor targets, which are protected by a DNP solution, especially during DDoS attacks. This can be achieved by long term monitoring of services known to be using DNP, or in a lab environment, where DDoS can be generated on-demand. The monitoring should encapsulate several dimensions (e.g., round-trip-time, volume of traffic generated) , but

[8] https://www.cloudflare.com/

being focused on where the malicious traffic is going through, for example by means of one of the open source tools developed in this project, TraceMan (see D1.1).

At this stage, the collaboration and its goals have been identified, but the research work is planned to start later in the project. Given its strong measurement focus, this collaboration fits in WP5. Since it is also planned to perform automated mitigation of attacks, this collaboration also falls within WP6 (see also Figure 33).

4.3 Activities

In this section we list the activities that have taken place in relation to the aforementioned collaborations. Also this part appears in deliverable D6.1 and D5.1 because the collaborations involve elements from WP6 and WP5.

- Face-to-face FLAMINGO meetings have taken place during the IM 2013 conference in Gent, in particular:
  - **UT-UZH**: The goal of the meeting was to investigate possible collaboration between the UZH PhD student Guilherme Sperb Machado and the researchers from UT. The outcome of this meeting has later concretised in the collaboration **UT-UZH-DoS**
  - **UT-INRIA**: the goal of the meeting was to investigate possible collaboration on the topic of network security and modeling between the PhD student Anthéa Mayzaud (INRIA) and the UT.
  - **UCL-iMinds**: The goal of the meeting was to define the work plan for the collaboration **UCL-iMinds-Cache**. During the meeting the research topics under investigation were discussed in more detail including cache management algorithms and mechanisms to dynamically scale the cache capacity.
  - **UT-iMinds**: R. Schmidt (UT) and N. Bouten (iMinds) discussed the different approaches for traffic measurements (sFlow, packet-based) and how they can be integrated within the ns-3 based HTTP Adaptive Streaming simulation framework (**iMinds- UT-QoS**). This meeting was at the basis for the collaboration **iMinds-UT-QoS** which progressed in the next months.
- Nikolay Melnikov (JUB), has visited the UT for one week in June 2013 in the scope of the PhD collaboration with Ricardo Schmidt. The topic of the collaboration **JUB-UT-Pattern** is establishing the contribution of individual hosts to the Gaussianity properties of the traffic aggregate. This research is carried out in collaboration with Dr. R. Sadre of the University of Aalborg. Regular phone calls are taking place.
- Jair Santanna (UT) has visited the Jacobs University Bremen for a face-to-face meeting with the co-supervisor J. Schönwälder. The goal of the meeting was to discuss the Jair Santanna's PhD topic and his current research.
- In the context of the collaboration on **UZH-UniBwM-SLA**, bi-weekly telcos are taking place. In the mid of November one week in Zurich is planned to evaluate mobile measurement environments and to get possible inputs for conference papers. Also a visit from Christos Tsiaras and Daniel Dönni is planned in Munich in the early 2014.
- Rashid Mijumbi (UCL) has visited iMinds for two weeks in the scope of the collaboration **iMinds-UPC-NetVirt**, with the goal of defining research directions for a collaboration between the two institutions, on the topic of virtual network embedding. The outcomes are

detailed discussions to clear definitions of the collaboration timeline including public targets. Additionally a paper to NOMS has been submitted [11].

- UT and iMinds are currently collaborating **iMinds-UT-QoS** on combining flow-based traffic measurement with In-Network Video Quality Adaptation. Researchers from the two institutions have regular phone calls to discuss the progress of the collaboration.
- Rick Hofstede (UT) has visited UniBwM two days in February 2013 in the scope of Intrusion Detection Systems in the collaboration **UT-UniBwM-IDS** with Mario Golling. The outcomes are further collaboration plans and ideas for joint papers. In the future more shorter visits in Enschede and Munich, depending on paper deadlines, are planned.
- Radhika Garg (UZH) is planning a visit at UPC in the scope of the collaboration **UZH-UPC-Legal** in the context of legal regulations.
- Daphne Tuncer (UCL) is planning to visit at iMinds to prepare a joint paper in the scope of the collaboration **UCL-iMinds-Cache**.
- Anthea Mayzaud (INRIA) visited JUB for one week in June 2013 in the scope of the collaboration **INRIA-JUB-RPL**. During this visit they defined the exact collaboration aspects. They discussed a topology for simulation study with multiple scenarios that take into account different attack and data-delivery rates. Additionally they modified the Contiki RPL implementation to enable the evaluation of possible DAG inconsistency attacks and the structure for a full paper was defined.

5 First Draft of a FLAMINGO Automation Architecture in the Area of Intrusion Detection Systems

Monitoring data is the basis to build a data base, from which a knowledge base as well as information models can be derived. These can be used as an input for correlation methods, learning techniques and knowledge description, what in turn builds the basis for the corresponding automated actions.

Since the goal of WP6 is to develop an inter-domain architecture for automated configuration and repair, we decided to address this by starting with a bottom-up approach analyzing the problem area on specific research issues/domains. Regarding the general cooperation between WP5/WP6 and following the interdependence between monitoring and configuration and repair, the main ideas is as follows. While WP5 develops the FLAMINGO Monitoring Architecture, WP6 extends this architecture with aspects related to automated configuration and repair. Due to the tight cooperation between WP5 and WP6 in the area of IDSs as well as the concentration of several collaborations on this topic, the area of Intrusion Detection was selected as the most appropriate one to begin with.

Monitoring data, as done with the TraceMan tool, provided by UZH and also used by the collaboration **UT-UZH-DoS**, is one of the basic modules of our architecture. The monitoring includes several elements (e.g., round trip time, volume of traffic generated), and is focused on the paths where the malicious traffic gets routed through. To support mobile devices as well, the architecture uses data from Flowoid, a NetFlow/IPFIX probe for Android-based devices, which is developed and further investigated by the collaboration **UT-INRIA-Flowoid**.

In the scope of WP6, we developed a first draft of an inter-domain automation architecture based on the collaboration **UT-UniBwM-IDS** (see Figure 34).

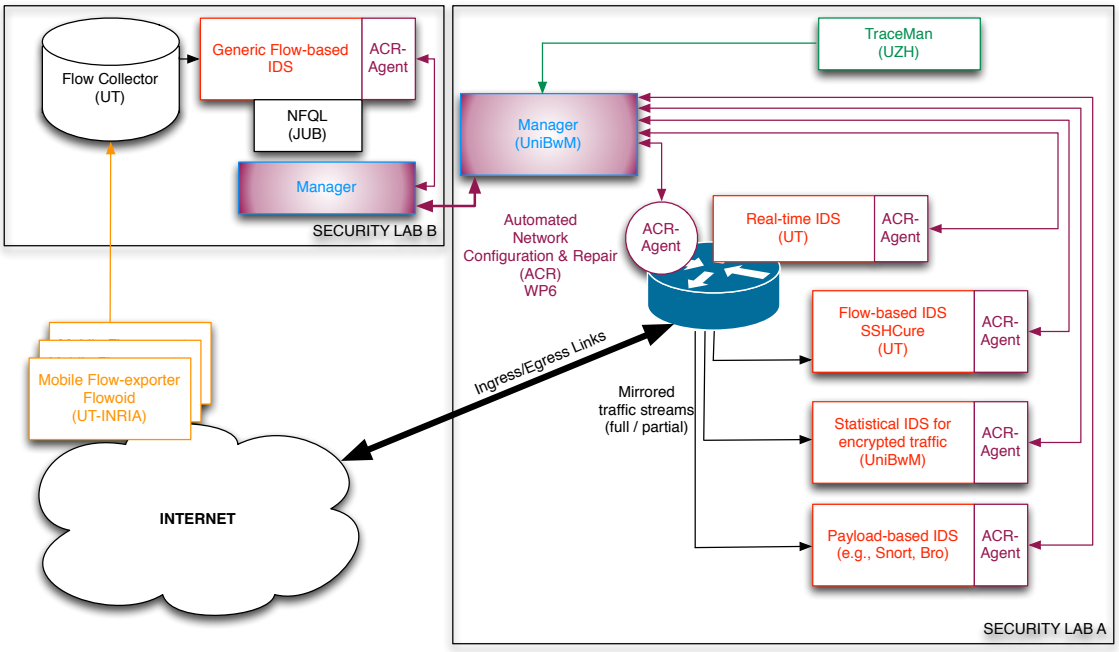


Figure 34: First draft of own architecture

The first approach is a Real-Time IDS which is capable of real-time Intrusion Detection using a set of easily measurable metrics, for example multiple source-IP-addresses accessing a single

destination-IP-address typically indicating DDoS attacks. In comparison with traditional, ”heavy-weight” sensors, this Real-Time IDS is a comparatively ”light-weight” sensor, since we are not dealing with a dedicated device, but rather elude resources on the router (hampering him to fulfill his ”core business”).

Therefore, the next step includes the involvement of more ”heavy-weight” sensors. A first example of such an IDS is based on SSHCure, a flow-based IDS specific for SSH attacks, which implements an algorithm for near real-time detection of ongoing attacks and allows the identification of compromised attack targets. In case the flow-based IDS reports an alert, an additional IDS is used to verify/falsify the results of the flow-based IDS, for example, the well-known IDS snort[9] in addition to Bro[10].

This approach is used because in our architecture (see Figure 34) we focus on expensive IDSs mostly placed in the backbone of the network rather than on IDSs which are commonly used and are much cheaper. Therefore, the flow-based IDS inspects the entire stream, and the additional IDSs analyze the packets of the stream which are flagged as suspicious. As a consequence, the entire stream is analyzed by the flow-based IDS, and the partial streams are analyzed by packet-based IDSs.

These IDSs have to be managed to allow an adaptation of a single IDS based on detections from other IDSs, supported by a couple of knowledge description approaches, learning techniques and algorithms. This functionality is consolidated in a centralized manager component per site, which evaluates the output of the IDSs to define an overall goal for the intrusion detection process at a glance. The definition of this goal results in piecewise configuration actions for the IDSs. To establish this functionality, the centralized manager should communicate via standardized protocols like IDMEF, YANG, SNMP with the detection systems. Where YANG describes a data modeling language which is used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF) with its functions. The communication acts, in contrast to WP5, in a bidirectional way, where the detection systems provide the manager the necessary data to configure the whole combined detection system with a global goal. As a consequence, the manager derives the corresponding configuration actions for the specific IDSs. This configuration information is sent back to the detection systems. Therefore, an Automated Configuration and Repair (ACR) agent on each detection system is needed. These agents translate the configuration information into detection system specific configuration commands, which are direct interpretable by the detection systems. Furthermore, the manager can influence the bypassing of the traffic from the router to the detection systems in a way that every IDS gets the corresponding traffic for the actual configuration.

By establishing a “rule-of-thumb” to estimate whether traffic is Gaussian or not, based on conclusions from data analysis such as hosts behavior and applications usage, the collaboration **JUB-UT-Pattern** also supports our manager. The Network Flow Query Language (NFQL), developed by JUB, acts as a possible part of the architecture by providing functions that allow pattern querying in a flow message stream.

To allow a fast reconfiguration, which is necessary to analyze bypassing streams using different approaches, the network virtualization approach, emerged out of the collaboration **iMinds-UPC-NetVirt**, supports our architecture. The approach is based on service requirements, which can also be derived out of a configuration action provided by the client. In addition, these reconfigurations could be supported in an energy-aware context, which is based on the collaboration activities from **UCL-UT-Man**.

[9] http://www.snort.org/
[10] http://www.bro.org/

A fast reconfiguration of network backbone links could implicate incidents in the context of Service Level Agreements. For example, if a provider reconfigures its backbone link the connected consumers could detect a decreasing bandwidth and therefore assume a violation of their contract with the network provider. We could probably cope with these issues by adjusting the research out of the collaboration **UZH-UniBwM-SLA**.

In this first draft our architecture exclusively addresses the environment of a single security lab to be expanded with the functionality of automated configuration and repair through the application of a centralized manager component and an agent component being part of the detection systems. In further steps throughout the project lifetime this simple and strong hierarchical approach can be improved by introducing manager-of-manager components and dissemination of manager components within the detection systems and agents thus introducing peer-to-peer concepts in automated network management systems. While a single instance of a manager-of-manager may define the overall goal by examining the results derived from other manager components. These manager components process their local outputs from the detection systems and adapt them considering the goal definitions from the manager-of-manager component implementing a cyclic configuration process for both components. In further development the manager components may support other manager components via a peer-to-peer communication, which dissolves the hierarchical structure between the managers and enables autonomic configuration and repair mechanisms. To combine several security labs for the purpose of mutual gain, it is necessary that a manager component can take this functionality. However, a peer-to-peer approach is envisioned, and will be investigated.

During the lifetime of the project, the active PhD collaborations described in Section 4 and the planned collaborations shown in Figure 20 will contribute to the core architecture. Due to the fact that WP6 requires the monitoring data from WP5, we are currently in an early step of our architecture. Within 2014 we will further focus on the specification whether and how existing research approaches, particularly with respect to knowledge description, model building and learning techniques, can be used for our architecture, resp. needed to be extended. Similar, research questions will center around protocols for the information and configuration exchange.

6 Conclusions and Outlook

Deliverable 6.1 describes the achievements of WP6 in the first year with respect to the S.M.A.R.T. as well as work package specific objectives. This deliverable documents also the full achievement of these objectives. For more details about the achievement of the S.M.A.R.T. objectives, the reader is referred to Deliverable D8.1; the achievements of work package specific objectives in the first year have been reported in this deliverable.

The work package specific objectives in the first year have centered around (i) building an inventory of architectures and approaches for automated configuration and repair, including the development of a taxonomy, (ii) a first blueprint of metrics, and (iii) a description of selected application domains. In addition, D6.1 reports on the first draft of a FLAMINGO inter-domain automation architectures, based on the monitoring architecture, as presented in D5.1 but extended with automation aspects. All WP6 specific objectives (except one) are in the status of ongoing research.

A strong integration of PhD collaborations (both, fully payed by FLAMINGO as well as not payed by FLAMINGO) is the basis for the immense and excellent scientific output, as well as the achieved results. The extreme success of PhD collaborations in the first year, especially between WP5 and WP6, is a guarantee to obtain excellent research results in the next years as well, and to foster and extend the joint PhD collaborations.

Acknowledgments

This deliverable is based on input from the WP6 Partners of the FLAMINGO consortium. A particular acknowledgment goes to all the PhD students that have not only provided textual input, but that are working on a daily basis on the challenging research topics that we report.

Abbreviations

$6LoWPAN$	IPv6 over Low Power, Wireless Networks
ACE	Autonomic Communication Element
ACR	Automated Configuration and Repair
AE	Autonomic Element
API	Application Programming Interface2
AME	Autonomic Management Entities
ANA	Autonomic Network Architecture
$ANEMA$	Autonomic Network Management Architecture
ANM	Autonomic Network Management
$CASCADAS$	Component-ware for Autonomic, Situation-aware Communications and Dynamically Adaptable Services
CC	Common Criteria
CCE	Common Configuration Enumeration
CCN	Content Centric Networking
CDN	Content Delivery Network
CP	Content Provider
CPS	Cyber Physical Systems
CVE	Common Vulnerabilities and Exposures language
$CVSS$	Common Vulnerability Scoring System
$DACoRM$	Decentralised and Adaptive Network Resource Management Framework
$DDoS$	Distributed Denial of Service attack
DNP	DDoS Network Protections
$DGDP$	Domain Goal Distribution Point
DHT	Distributed Hash Table
$DODAG$	Destination Oriented Directed Acyclic Graph

$DONA$	Data-Oriented Network Architecture
DoW	Description of Work
DSL	Digital Subscriber Line
EU	European Union
ECB	Enterprise Context Bus
ESB	Enterprise Service Bus
FB	Functional Block
$FCAPS$	Fault Management, Configuration Management, Accounting Management, Performance Management, Security Management
FI	Future Internet
FN	False Negative
$FOCALE$	Foundation, Observation, Comparison, Action, Learning, rEason
FP	False Positive
$FTTH$	Fiber to the Home
GA	Goal Achievable
GAP	Goal Achievement Points
GDP	Goal Definition Point
GN	Goal Needed
GPS	Global Positioning System
HAS	HTTP Adaptive Streaming
$HSPA$	High Speed Packet Access
$HTTP$	Hyper-text Transfer Protocol
$ICMP$	Internet Control Message Protocol
ICN	Information-Centric Networking
IDP	Information Dispatch Point
IDS	Intrusion Detection System
$IDMEF$	Intrusion Detection Message Exchange Format
$IETF$	Internet Engineering Task Force
IFP	Infrastructure Providers
ILP	Integer Linear Program
InP	Infrastructure Provider
$INRIA$	Institut National de Recherche en Informatique et Automatique

IoT	Internet of Things
IP	Internet Protocol
$IPFIX$	Internet Protocol Flow Information Export
ISP	Internet Service Provider
IST	Information Society Technologies
$ITIL$	IT infrastructure library
$ITU-T$	International Telecommunications Union - Telecommunications Standardization Sector
JUB	Jacobs University Bremen
KA	Knowledge Atoms
KN	Knowledge Network
LLN	Low-power and Lossy Networks
LTE	Long Term Evolution
$MAPE$	Monitoring, Analyzing, Planning, Executing
MNO	Mobile Network Operator
MOS	Mean Opinion Score
$MP2P$	Multipoint-to-Point
NDO	Named Data Object
$NETCONF$	Network Configuration Protocol
$NFQL$	Network Flow Query Language
$NIST$	National Institute of Standards and Technology
NRS	Name Resolution Service
NUF	Network Utility Function
OAL	Object Achievement Layer
ODL	objective Definition Layer
ODP	Object Definition Point
$OGSA$	Open Grid Services Architecture
$OVAL$	Open Vulnerability and Assessment Language
$PSIRP$	Publish-Subscribe Internet Routing Paradigm
QoE	Quality-of-Experience
QoS	Quality-of-Service
$ROLL$	Routing Over Low Power Lossy networks

RPL	Routing Protocol for Low power and Lossy Networks
$SCAP$	Security Content Automation Protocol
SLA	Service Level Agreement
$S.M.A.R.T.$	Specific Measurable Achievable Relevant Timely
SN	Substrate Network
$SNMP$	Simple Network Management Protocol
SOA	Service-oriented architecture
SP	Service Provider
SSH	Secure Shell
SWM	Small-World Network
$P2P$	Peer-to-Peer
$P2MP$	Point-to-Multipoint
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
TPM	Trusted Platform Module
ToS	Type of Service
$UniBwM$	Universität der Bundeswehr München
UCL	University College London
UDP	User Datagram Protocol
$UMTS$	Universal Mobile Telecommunications System
UPC	Universitat Politecnica de Catalunya
UT	University of Twente
UZH	University of Zürich
$VoIP$	Voice-over-IP
VoS	Value of Service
VNP	Virtual Network Provider
$WLAN$	Wireless Local Area Network
WP	Work Package
WSN	Wireless Sensor Network
$WSDN$	Web Services Description Language

XML	Extensible Markup Language
$XCCDF$	eXtensible Configuration Checklist Description Format

References

[1] I. Drago, E. Bocchi, M. Mellia, H. Slatman, and A. Pras. Benchmarking personal cloud storage. In *ACM/SIGCOMM Internet Measurements Conference 2013 (IMC 2013)*, 2013.

[2] M. Barrere, R. Badonnel, and O. Festor. Vulnerability assessment in autonomic networks and services: A survey. *Communications Surveys & Tutorials, IEEE*, 2013.

[3] A. Lareida, T. Bocek, M. Waldburger, and B. Stiller. Rb-tracker: A fully distributed, replicating, network-, and topology-aware p2p cdn. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1199–1202, 2013.

[4] G. Sperb Machado, T. Bocek, M. Ammann, and B. Stiller. A cloud storage overlay to aggregate heterogeneous cloud services. In *Proc. of the 38th IEEE Conference on Local Computer Networks (LCN 2013),*, Oct. 2013.

[5] P. Poullie and B. Stiller. Fair allocation of multiple resources using a non-monetary allocation mechanism. In *Proc. 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2013)*, pages 45–48. 2013.

[6] C. Schmitt, B. Stiller, T. Kothmayr, and W. Hu. DTLS-based security with two-way authentication for IoT (Internet Draft). http://tools.ietf.org/html/draft-schmitt-two-way-authentication-for-iot-01, October 2013.

[7] O. Festor, A. Lahmadi, R. Hofstede, and A. Pras. Information elements for ipfix metering process location (internet draft). http://tools.ietf.org/html/draft-festor-ipfix-metering-process-location-01, July 2013.

[8] Daphne Tuncer, Marinos Charalambides, Raul Landa, and George Pavlou. More control over network resources: an isp caching perspective. In *IFIP/IEEE International Conference on Network and Service Management (CNSM 2013)*, 2013.

[9] S. Seeber, A. Sehgal, B. Stelte, G. D. Rodosek, and J. Schönwälder. A trust computing architecture for rpl in cyber physical systems. In *IFIP/IEEE International Conference on Network and Service Management (CNSM 2013)*, 2013.

[10] R. Schmidt, N. Melnikov, R. Sadre, J. Schönwälder, and A. Pras. Linking network usage patterns to traffic gaussianity fit. In *submitted to PAM 2014*, 2014.

[11] R. Mijumbi, J. Serrat, J.L. Gorricho, M. Claeys, F. De Turck, and S. Latré. Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In *Submitted to NOMS 2014*, 2014.

[12] A. Sehgal and A. Mayzaud. Mitigating dodag inconsistency attacks in rpl networks. In *submitted to ACM/IEEE IPSN 2014*, 2014.

[13] R. Koch, M. Golling, and G.D. Rodosek. Evaluation of state of the art IDS message exchange protocols. In *Communication and Network Security, 2013 International Conference on*, 2013.

[14] G.D. Rodosek, M. Golling, W. Hommel, and F. Tietze. Iceman: An architecture for secure federated inter-cloud identity management. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1207–1210, 2013.

[15] G.D. Rodosek, M. Golling, and W. Hommel. Music: An it security architecture for inter-community clouds. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 812–815, 2013.

[16] M. Charalambides, D. Tuncer, L. Mamatas, and G. Pavlou. Energy-aware adaptive network resource management. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 369–377, 2013.

[17] R. Koch and M. Golling. Architecture for evaluating and correlating NIDS in real - world networks. In *Cyber Conflict (CyCon), 2013 5th International Conference on*, pages 1–20, 2013.

[18] R. Mujumbi, J.-L. Gorricho, J. Serrat, M. Claeys, S. Latre, and F. De Turck. Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In *Network Operations and Management Symposium Workshop 2014*, 2014.

[19] B. Stelte and G.D. Rodosek. Assuring trustworthiness of sensor data for cyber-physical systems. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 395–402, 2013.

[20] B. Stelte and G.D. Rodosek. Thwarting attacks on zigbee removal of the killerbee stinger. In *IFIP/IEEE International Conference on Network and Service Management (CNSM 2013)*, 2013.

[21] N. Bouten, S. Latré, and F. De Turck. Qoe-centric management of multimedia networks through cooperative control loops. In G Doyen, M Waldburger, P Čeleda, A Sperotto, and B Stiller, editors, *Emerging management mechanisms for the future internet*, volume 7943 of *Lecture Notes in Computer Science*, pages 96–99. Springer Berlin Heidelberg, 2013.

[22] N. Bouten, S. Latre, J. Famaey, F. De Turck, and W. Van Leekwijck. Minimizing the impact of delay on live svc-based http adaptive streaming services. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1399–1404, 2013.

[23] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck. Design of a q-learning-based client quality selection algorithm for http adaptive video streaming. In *Adaptive and Learning Agents Workshop, part of AAMAS2013, Proceedings*, pages 30–37, 2013.

[24] D. Tuncer, M. Charalambides, R. Landa, and G. Pavlou. More control over network resources: an isp caching perspective. In *Network and Service Management (CNSM 2013), 2013 IFIP/IEEE International Symposium on*, 2013.

[25] O. Dabbedi, R. Badonnel, and O. Festor. Leveraging countermeasures as a service for voip security in the cloud. *International Journal of Network Management*. submitted, currently under review.

[26] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A aurvey of autonomic network architectures and evaluation criteria. *Communications Surveys Tutorials, IEEE*, 14(2):464–490, 2012.

[27] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[28] IBM Corp. *An architectural blueprint for autonomic computing*. IBM Corp., USA, October 2004.

[29] N. Samaan and A. Karmouch. Towards autonomic network management: an analysis of current and future research directions. *Communications Surveys Tutorials, IEEE*, 11(3):22–36, 2009.

[30] M. C. Huebscher and J. A. McCann. A survey of autonomic computing—degrees, models, and applications. *ACM Comput. Surv.*, 40(3):7:1–7:28, August 2008.

[31] S. Schmid, M. Sifalakis, and D. Hutchison. Towards autonomic networks. In *Proceedings of the First IFIP TC6 international conference on Autonomic Networking*, AN'06, pages 1–11, Berlin, Heidelberg, 2006. Springer-Verlag.

[32] M. Wooldridge and N. R. Jennings. Intelligent agents theory and practice. *Knowledge Engineering Review*, 10:115–152, 1995.

[33] J. A. Mccann and J.S. Crane. Kendra: internet distribution delivery system. In *In Proceedings of SCS Euromedia*, pages 134–140. IEEE, 1998.

[34] A. Lippman. Video coding for multiple target audiences. In K. Aizawa, R. L. Stevenson, and Y.-Q. Zhang, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3653 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 780–782, December 1998.

[35] A. Ganek and R. J. Friedrich. The road ahead—achieving wide-scale deployment of autonomic technologies. In *In Proceedings of the 3rd IEEE International Conference on Autonomic Computing*, 2006.

[36] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A survey of autonomic network architectures and evaluation criteria. *Communications Surveys Tutorials, IEEE*, 14(2):464–490, 2012.

[37] J. McCann and M. Huebscher. Evaluation issues in autonomic computing. In *Grid and Cooperative Computing-GCC 2004 Workshops*, pages 597–608. Springer, 2004.

[38] T. De Wolf and T. Holvoet. Evaluation and comparison of decentralised autonomic computing systems. *CW Reports*, page 10, 2006.

[39] A. Brown, J. Hellerstein, M. Hogstrom, T. Lau, S. Lightstone, P. Shum, and M. Yost. Benchmarking autonomic capabilities: Promises and pitfalls. In *Autonomic Computing, 2004. Proceedings. International Conference on*, pages 266–267. IEEE, 2004.

[40] N. Samaan and A. Karmouch. Towards autonomic network management: an analysis of current and future research directions. *Communications Surveys & Tutorials, IEEE*, 11(3):22–36, 2009.

[41] G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, and M. May. The autonomic network architecture (ana). *Selected Areas in Communications, IEEE Journal on*, 28(1):4–14, 2010.

[42] H. Derbel, N. Agoulmine, and M. Salaün. Anema: Autonomic network management architecture to support self-configuration and self-optimization in $\{\mathrm{IP}\}$ networks. *Computer Networks*, 53(3):418 – 430, 2009.

[43] A. Manzalini and F. Zambonelli. Towards autonomic and situation-aware communication services: the cascadas vision. In *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on*, pages 383–388, 2006.

[44] L. Baresi, A. Ferdin, A. Manzalini, L. Baresi, A. Ferdinando, A. Manzalini, and F. Zambonelli. The cascadas framework for autonomic communications. In *Autonomic Communication (Springer, Heidelberg and*, 2009.

[45] J. Strassner, N. Agoulmine, and E. Lehtihet. Focale: a novel autonomic networking architecture. 2006.

[46] N. Agoulmine. *Autonomic network management principles: from concepts to applications*. Access Online via Elsevier, 2010.

[47] J. Famaey, S. Latré, J. Strassner, and F. De Turck. A hierarchical approach to autonomic network management. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 225–232. IEEE, 2010.

[48] P. Roy, S. Haridi, A. Reinefeld, J. Stefani, R. Yap, and T. Coupaye. Self management for large-scale distributed systems: an overview of the selfman project. In F. Boer, M. Bonsangue, S. Graf, and W. Roever, editors, *Formal methods for components and objects*, volume 5382 of *Lecture Notes in Computer Science*, pages 153–178. Springer Berlin Heidelberg, 2008.

[49] Peter Van Roy et al. Publishable final activity report of the SELFMAN project. Technical report, European Sixth Framework Programme, Information Society Technologies, November 2009.

[50] J. Banghart and C. Johnson. The technical specification for the Security Content Automation Protocol (SCAP). `http://scap.nist.gov/revision/`, 2011. Cited September 2013.

[51] NIST, National Institute of Standards and Technology. `http://www.nist.gov/`. Cited September 2013.

[52] The OVAL Language. `http://oval.mitre.org/`. Cited September 2013.

[53] N. Ziring and S. D. Quinn. Specification for the Extensible Configuration Checklist Description Format (XCCDF). NIST (National Institute of Standards and Technology). `http://scap.nist.gov/specifications/xccdf/`. Cited September 2013.

[54] CVSS, Common Vulnerability Scoring System. `http://www.first.org/cvss/`. Cited September 2013.

[55] D. Tuncer, G. Charalambides, M.and Pavlou, and N. Wang. Dacorm: A coordinated, decentralized and adaptive network resource management scheme. In *In Proceedings of IEEE NOMS '12*, pages 417–425, April Hawaii, USA, 2012.

[56] M. Charalambides, G. Pavlou, P. Flegkas, N. Wang, and D. Tuncer. Managing the future internet through intelligent in-network substrates. *IEEE Network*, 25(6):34–40, November 2011.

[57] A. Aizuddin. The common criteria iso/iec 15408–the insight, some thoughts, questions and issues, 2002.

[58] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, September 2012. `http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4_marked_changes.pdf`.

[59] An Introduction To ISO 27001 (ISO27001). `http://www.27000.org/iso-27001.htm`.

[60] A. Cater-Steel and W. Tan. Implementation of it infrastructure library (itil) in australia: progress and success factors. In *2005 IT Governance International Conference*, pages 39–52. Auckland University of Technology, 2005.

[61] Official ITIL Website. `http://www.itil-officialsite.com/`.

[62] M. Johnson, A. Hately, B. Miller, and R. Orr. Evolving standards for it service management. *IBM Systems Journal*, 46(3):583–597, 2007.

[63] M. Salehie and L. Tahvildari. Self-adaptive software: landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(2):14, 2009.

[64] J. Schönwälder, M. Björklund, and P. Shafer. Network configuration management using netconf and yang. *Communications Magazine, IEEE*, 48(9):166–173, 2010.

[65] J. Case, M. Fedor, M. Schoffstall, and C. Davin. *A simple network management protocol (SNMP)*. Network Information Center, SRI International, 1989.

[66] Y. Mansour and B. Patt-Shamir. Jitter control in qos networks. *IEEE/ACM Transactions on Networking (TON)*, 9(4):492–502, 2001.

[67] S. Davy, K. Barrett, S. Balasubramaniam, S. Van der Meer, B. Jennings, and J. Strassner. Policy-based architecture to enable autonomic communications - a position paper. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 590–594, 2006.

[68] A. Mayzaud, R. Badonnel, and I. Chrisment. Monitoring and security for the internet of things. In *Proc. 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2013)*, June 2013.

[69] Peter Mell and Timothy Grance. The nist definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[70] E. Krol and E. Hoffman. Fyi on what is the internet? RFC 1462, may 1993.

[71] Wetzel R. Cloud federation primer: the coming intercloud. `http://searchcloudprovider.techtarget.com/feature/Cloud-federation-primer-The-coming-Intercloud/`.

[72] Bernstein D. Virtualization, cloud computing, the next internet. In *The Open Group 2009 Enterprise Cloud Computing Conference, February 2-6, 2009, San Diego, California*, 2009.

[73] A. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante. Drafting behind akamai (travelocity-based detouring). *SIGCOMM Comput. Commun. Rev.*, 36(4):435–446, August 2006.

[74] Akamai CDN. `http://www.akamai.com`.

[75] Limelight Networks CDN. `http://www.limelight.com`.

[76] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber. Pushing CDN-ISP collaboration to the limit. *SIGCOMM Comput. Commun. Rev.*, 43(3):34–44, July 2013.

[77] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26–36, 2012.

[78] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi. A survey on content-oriented networking for efficient content delivery. *Communications Magazine, IEEE*, 49(3):121–127, 2011.

[79] E. Borcoci and D. Negru. Content oriented routing and forwarding. In *NexComm 2012 Conference*, 2012.

[80] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, 2009.

[81] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, pages 181–192, 2007.

[82] M. Ain and D. Trossen. Architecture definition, component descriptions, and requirements. Deliverable D2.3, PSIRP project, 2009.

[83] R. Schmidt, R. Sadre, A. Sperotto, and A. Pras. Lightweight link dimensioning using sflow sampling. 2013.

[84] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (IDPS). *NIST Special Publication*, 800(2007):94, 2007.

[85] M. Golling and B. Stelte. Requirements for a future EWS-cyber defence in the internet of the future. In *Cyber Conflict (ICCC), 2011 3rd International Conference on*, pages 1–16. IEEE, 2011.

[86] Géant. Breakthrough GÉANT network marks ten years of success: high bandwidth pan-european research network continues advances with 100 Gbps plans . `TenYearsOfSuccess`, November 2010.

[87] R. Hofstede and T. Fioreze. SURFmap: A network monitoring tool based on the google maps API. In *2009 IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), Long Island, New York, USA*, pages 676–690, June 2009.

[88] C. Tsiaras and B. Stiller. Challenging the monopoly of mobile termination charges with an auction-based charging and user-centric system (AbaCUS). *NetSys 2013 - Networked Systems*, GERMANY, March 11-15, 2013.