

A conceptual-driven survey on future internet requirements, technologies, and challenges

Antonio Marcos Alberti

Received: 12 April 2012 / Accepted: 5 February 2013 / Published online: 3 March 2013
© The Brazilian Computer Society 2013

Abstract Nowadays, research initiatives to redesign the Internet are popping up around the world. Each of these projects has a particular focus and its own set of design requirements and key technologies. The panorama shows a wide diversity of pre-requirements and arguments, some very specific and superficial, others more general and deep. Despite this diversity, a conceptual-driven analysis of such scenario could reveal the current state-of-the-art on future Internet research, the common aspects that permeate all initiatives, and the open issues to be addressed in future work. This paper provides a comprehensive overview of the contemporary research from an abstract point of view, covering the design requirements and ingredients that are being adopted, interrelating, and analyzing them.

Keywords Future internet · Future networks · New generation · Architecture · Requirements · Challenges · Technologies · Concepts

1 Introduction

Nowadays, research projects to redesign the Internet are popping up around the world [1]. This is happening as many people have begun to question the capability of the current Internet architecture to continue meeting the desires of our society [2]. In fact, we are trying to continue using an architecture that was designed in the 1970s to meet a different scenario of requirements and applications. The uncertainty behind this continuity elucidates why many research projects are aimed at redesigning the Internet: they are looking for

innovative architectures that can better achieve our information and communication demands. That was exactly what researchers at Stanford University were doing when they asked: What if we could redesign the Internet with the current technologies and resources? What would it look like? The movement arising from these questions became known as *clean-slate* design [3] and gave rise to the impressive wave of research that we are witnessing today.

In this context, what could people expect from a *clean-slate* Internet? Nowadays, people put on the Internet a significant portion of their expectations regarding the evolution of information and communication technologies (ICT). The Internet infrastructure is already a basic need for individuals, organizations, and even governments. Everything indicates that the role of the Internet will continue to be of much relevance in our society. The desires for a new Internet are very diverse and complex [4, 5], with aspects ranging from socio-economic, energetic, cultural, environmental, educational, and entertainment up to political, climatic, and health, among others. People expect a network that is more trustworthy, secure, efficient, resilient, pervasive, and open for innovative applications, with higher capacity, connectivity, mobility, autonomy, diversity, and quality.

It worths to mention that the *clean-slate* approach is not the only one possible. There are those that believe that significant changes can be obtained by evolving the current Internet protocols. The debate between evolutionary versus *clean-slate* approaches called a lot of attention on the community [6]. The fact is that Internet technology will continue evolving not only by means of incremental steps, but also by revolutionary ideas. Both paths are equally important.

Albeit quite inspiring, the current scenario is characterized by a wide variety of ideas, arguments, and visions. Virtually every future Internet proposal has its own list of requirements, challenges, and key technologies. Some are more

A.M. Alberti (✉)
Instituto Nacional de Telecomunicações–INATEL,
Santa Rita do Sapucaí, Minas Gerais 37540-000, Brazil
e-mail: alberti@inatel.br

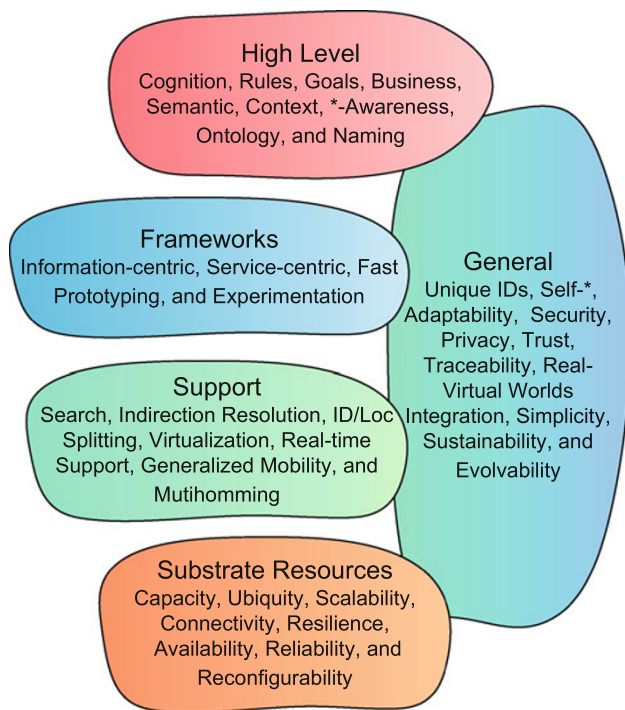


Fig. 1 Future internet requirements and concepts selected from the current proposals

specific and include very particular requirements. Others are more general, with pre-requirements spread over several areas. Such diversity creates the impression that a generalized understanding of the technical issues adopted in overall approaches is an impossible goal. This is exactly the focus of this paper: to summarize the core ideas, concepts, and technologies behind the current state-of-the-art on future Internet design. Compared to the previous surveys on future Internet research (which are driven mainly by geographical and funding issues [1, 7, 8]), this paper relies on a conceptual-driven approach. It focuses on relating ideas and concepts behind those approaches from an abstract point view.

1.1 Paper organization

This conceptual review started with the selection of the most representative requirements and indirectly the concepts behind contemporary proposals for the future Internet. The selection was made through a critical analysis based on the following criteria: (1) how frequently each prerequisite appeared in current approaches for the future Internet; (2) how deep each requirement contributes to achieve our societal aspirations; and (3) what the architectural gains are behind each requirement. Figure 1 presents the selected requirements. At the bottom are the substrate resources and their requirements. Above there are some support systems that hold broader frameworks. At the top are the high-level

aspects. On the right side are general requirements that cover all previous scopes.

In order to cover Fig. 1 landscape, the remainder of the paper was organized as follows: Sect. 2 concentrates on the requirements behind substrate resources (i.e., hardware), including capacity, ubiquity, interactivity, and traffic growth. Section 3 covers the integration of real and virtual worlds. Section 4 is focused on the virtualization paradigm, while the Sect. 5 covers the innovative paradigm of software-defined networking (SDN). Section 6 focuses on adaptability, autonomy, and manageability. Section 7 discusses service and application-level requirements, such as architecture neutrality, openness, diversity, extendibility, flexibility, compose-ability, and usability. Section 8 tackles the support for persistent information representation and innovative communication models. Section 9 concentrates on indirection resolution. Section 10 looks into the support for mobility and multihoming, including the identification (ID)/location (LOC) splitting paradigm. Section 11 delineates the security, privacy, trust, and transparency requirements. Section 12 focuses on designing for simplicity, evolvability, and sustainability.

2 Technology evolution and its impact on capacity, ubiquity, and traffic

For a long time, people have been talking about technological evolution and the rate at which it occurs. Very often, people cite Moore's Law [9] as an example of law being capable of predicting technological developments in computing power.¹ More recently, Kurzweil [10] presented a theory for technological evolution: the Law of Accelerated Returns. It states that two positive feedback loops occur during some technology evolution process. The result is that an evolutionary process starts to reduce faster and faster the time interval between important returns (outcomes).

In his book, Kurzweil [10] presents a series of figures that show exponential growth trends for memory capacity (DRAM in bits per dollar), microprocessor clock speed (Hz), transistors per chip, processor performance in million instructions per second (MIPS), and magnetic storage (bits per dollar). For example, the decrease in the ratio of cost per MIPS is about 8 million to 1 from 1967 up to 2004 [10]. In the same period, memory capacity improved approximately 2,000 times. Saracco [11] argued that technological developments in digital storage and processing have been consistent in recent years. The number of hosts on the Internet has also been progressing exponentially, at least for now. High-performance computing based on supercomputers (or computer clusters) has already achieved petaflops (10^{15}) floating

¹ The number of transistors in an integrated circuit doubles every 2 years.

point operations per second [12] and evolution proceeds to exaflops (10^{18}). Moreover, display technology has advanced enormously in recent years, allowing ameliorated quality and larger screens, substantially improving the quality of experience and allowing new forms of digital interactivity [11]. The advance of consumer electronics in the form of handsets, laptops, HDTVs, e-books, video games, GPS, etc., pleads for exponential growth in these technologies. Besides being interesting, these estimations are important to characterize how substrate technologies capacity will evolve in the next decades. In other words, they help to answer the question: which is the capacity available to the design of future ICT? To know the available capacity is important to any project.

However, the exponential growths are also valid for traffic [10]. The Minnesota Internet Traffic Studies (MINTS) [13] estimate that the annual Internet traffic growth rates were about 50–60% in 2008 and about 40–50% in 2009. With this number, we can contemplate a growth of roughly 30–100 times in the next decade. Moreover, the monthly Internet traffic estimated by MINTS was circa 7,500–12,000 petabytes, i.e., $7.5–12 \times 10^{18}$ bytes or exabytes. Another estimate made by researchers from the Japanese Akari project [2] indicates that traffic could increase 1.7 times per year in Japan in the next years. Therefore, network architects need to consider exponential growths in traffic.

Also, according to [14], North Americans spent approximately 60 h online every month in 2010. People are spending more and more time online. Many people would like to be always online, aware of their received e-mails, agenda, Skype, MSN, and social networks posts. Therefore, new architectures need to provide improved reliability. In addition, people would like to enjoy the augmenting diversity of applications that are becoming available on mobile devices, e.g., social networks, blogs, forums, etc., with very high quality of service and experience. We could expect exponential growths on interactivity as well.

2.1 Initiatives

More research is being undertaken to find ways to meet such capacity requirements in various portions of the network. At the mobile access, cognitive radio (CR) [15, 16] (Sect. 6) and other reconfigurable radio standards are candidate technologies. At fixed access, the candidate technology is fiber-to-the-home (FTTH) technology.

At the network core, state-of-the-art optical transmission and switching are promising technologies [2]; they include ultra-dense wavelength division multiplexing, optical time division multiplexing, and optical packet switching. The Akari project adopted such technology combination to move towards petabit per second capacity in 2015.

The scalable and adaptive Internet solutions (SAIL) consortium [17] approach is based on three tiers: open connec-

tivity services (OConS), cloud networking (CloNe) (Sect. 4), and network of information (NetInf) (Sect. 8). OConS adopted an open multi-layer transport perspective, where overspread allocation of integrated resources (computing and networking) takes place. Supported transport technologies include all-optical networking, optically switched networking (OTN), multiprotocol label switching (MPLS), WiFi, and third generation/long term evolution (3G/LTE). OConS also includes the support for cooperative spectrum sensing, a typical feature of CRs.

Besides high capacity, the core network must have adequate availability, resilience, robustness, and reliability. Akari project focuses on distributed control to improve scalability. Another related requirement is electrical power consumption [4]. Akari adopted the optical packet and circuit switching to reduce the consumption of electricity.

2.2 Analysis

We increasingly connect using mobile devices [18, 19] and, of course, would like to become online while moving. Thus, mobile devices require a higher bandwidth and degrees of connectivity. Failures and intermittence must be dealt with, traffic must be redirected, and availability must be improved. In fact, the technology evolution of available computing is changing the degrees of connectivity and interactivity. The deflation of computing price is leading us towards ubiquitous computing [20], leading to computing that is everywhere, every time. Therefore, each device will probably have more neighbors where connections can be maintained, thus improving connectivity. New designs need to take advantage of pervasive and ubiquitous computing to increase availability.

As a result of the exponential growths on quantity of devices, connectivity, interactivity, and traffic appears to be a huge scalability challenge. Inexpensive computing leads to more and more devices that are computationally capable. If they connect to the Internet (e.g., through clothing, buildings), they could become the majority of connected devices, as discussed in the next section. Smart environments and ambient intelligence could emerge to ameliorate the quality of our lives, but they could also produce more pressure on network scalability. More ubiquity leads to more scalability problems, mainly regarding identification, location, routing, mobility, multihoming, and other issues, as discussed in the other sections of this paper.

3 Integration of real and virtual worlds

The current abundance of storage, processing, and display virtually qualifies any device to be connected to the Internet

as a network-enabled device (NED) [21]. This trend has been coined as the Internet of things (IoT) [22, 23]. The IoT idea is frequently assigned to the Massachusetts Institute of Technology (MIT) Auto-ID center. Examples of things are household appliances, security equipment, sensors and actuators, bottles of wine, surveillance equipment, goods in a supermarket, etc.

Sensors and identification devices could provide real-time information to enrich virtual-world applications, allowing changes in real-world objects to be reflected in virtual environments. The reverse is also a significant requirement. Through actuators, changes made to virtual objects can become real. For instance, people could close a virtual garage door of a 3D virtual model of their home. This action reflects in a real-world event through an actuator. This scenario is also interesting to illustrate another RWI challenge: the contextualized/semantics-rich discovery of sensors and actuators [24]. For instance, in a fire, an application can query whether there are any anti-fire actuators or sensors that can measure the temperature of the residence rooms. In this scenario, sensors and actuators need to be precisely described to enable a contextualized search by emergency applications. Security, traceability, and privacy are also crucial.

3.1 Initiatives

Some technologies that are already used in IoT are radio frequency identification [25], near field communications [26], and wireless sensor and actuator networks (WSAN) [27]. Examples of WSAN applications are the monitoring of industrial plants, manufacturing processes, houses, cars, environment, climate, and healthcare.

According to the European Union Future Internet Assembly activities, the real world will become increasingly integrated with the virtual one, making it possible to greatly increase the interaction between them. Such approach is called real-world Internet (RWI) [21, 24]. There are several initiatives under the RWI umbrella. SENSEI [24] aims at realizing the RWI vision by integrating real world information and devices with virtual world services and applications. The idea is to integrate heterogeneous WSANs by means of a global common framework, exposing obtained information via universal interfaces. It aims at creating a secure, trustable, real-time, open, sensorial information market, where contextualized information is disseminated, managed, and reasoned. The SENSEI architecture employs service oriented architecture (SOA) (Sect. 7) concepts to provide resource discovery and composition.

In the Akari proposal, the real-world information is collected by a sensorial network called “NerveNet”, which supports a diversity of NEDs.

3.2 Analysis

The application of the Law of Accelerated Returns in the scope of NEDs could impact current ICT in several ways [28]. First, a tremendous growth in the number of sensors collecting real-world information could generate a veritable flood of traffic on the network. According to Akari [4] and Cross-ETP [5], the amount of NEDs plugged into the Internet could reach billions.

Another point is how to make this information securely available to services, applications, and other entities. Sensor networks carry information that is sensible for user privacy, such as identity, location, and other contextualized information. Traceability to a particular sensor node will require greater transparency from the network, without compromising the security and privacy of individuals. Other security aspects are trust relations among nodes, and data integrity and confidentiality. Trust and reputation mechanisms are interesting for NEDs, since traditional security could expend too much energy [29].

In addition, there is the challenge of how to address billions of new nodes. How do we locate them geographically? The collected information needs to be successfully contextualized to allow delivery to the right destination, at the right time (information freshness). Data description based on ontology is necessary to facilitate classification and contextualization. As summarized by Presser et al. [21], NEDs need to be uniquely identified, and they should be capable of collaborating with each other, of semantically and contextually interpreting information, of establishing trusted relations, and of exchanging ascertained knowledge. Furthermore, the mobility of NEDs needs to be supported, since some sensor or actuators could be mobile, e.g., cars and clothes.

This flurry of sensitive information will push network scalability to new limits, especially regarding security, traceability, privacy, location, addressing, identification, semantics, and context. We cannot imagine thousands of networked sensors and actuators that rely on human intervention to perform their tasks. Therefore, these devices need to be originally designed to self-organize, self-configure, self-optimize, self-protect, and self-heal. On the other side, sensing is relevant to collect real-world information needed for autonomic and cognitive decision-making processes, while acting is important to reflect high-level decisions in the real world.

In addition, the sensors and actuators have to deal with internal limitations, i.e., limited energy, transmission power, receiver sensitivity, transmission rates, etc. Therefore, network devices require **-awareness*² properties, c.f. self-awareness, energy-awareness, and service-awareness (Sect. 6). They also need to be aware of their environment

² The asterisk is used to represent awareness to several aspects of the architecture.

(c.f. situation-awareness). In summary, in a simplistic analogy with humans, the RWI role on future Internet is analogous to our sensory or somatic nervous systems, i.e., they provide raw data to be further contextualized for autonomic functionalities.

4 Virtualization

The exponential growth in ICT created a diffuse substrate of digital technologies composed mainly of processing, storage, display, and communication resources. Today, most pieces of telecommunications equipment are turning into computers, with CPUs, operating systems, etc. The same is occurring to our network terminals. There is a perceptible convergence towards multi-core generic hardware. In this context, some questions arise: how to make this diffuse substrate of hardware resources transparently and uniformly available to software? How can these substrate resources be exposed and shared by virtual versions of entities that perform as if they are real? What is necessary for these purposes?

Interestingly, in the late 1970s, a technology was proposed for this aim: computer virtualization [30–32]. It was a heated topic thereat, after IBM set up the virtual machine monitor (VMM) in the 1960s. The goal was to make a mainframe that could perform more than one operating system (OS). Then arose the virtual machine (VM), which is software made machine. A single mainframe could run multiple VMs, each with its own OS. In this context, to virtualize means to create an abstraction layer between the hardware and the OS. This abstraction layer “hides” and “homogenizes” hardware resources, allowing any OS to run concurrently in the same physical machine. In a broader sense, to virtualize can be defined as the act of creating the necessary conditions to support virtual versions of entities that perform as the original ones. According to this definition, the VMM creates a virtual machine abstraction, which performs like the real one from the operating system point of view.

Nowadays, computer virtualization is back in the spotlight with the so-called cloud computing [33,34]. Cloud computing uses virtualization paradigm to share physical machines, creating and migrating virtual machines according to customers’ need. The idea is to reduce costs by sharing existing physical infrastructures. The virtualization is not restricted to processing facilities. It can be applied to storage resources (virtual disks), software frameworks [e.g. virtual enterprise resource planning (ERP)], and even for individual applications (e.g. virtual text editor).

In the context of communication networks, virtualization started with the advent of the virtual private networks (VPNs) and virtual local area networks (VLANs). However, Peterson et al. [35] proposed to create virtualized networks to overcome traditional Internet protocol (IP) testbed limitations.

This led to the emergence of the overlay networks and PlanetLab [36,37], which is a virtual testbed network over traditional IP networks. Since then, virtualization has been used as a tool to create distributed experimentation environments. The idea is to design by experimenting, or to do experimentally driven research [38–42].

Virtualization of wireless networks is also possible [43–45]. However, wireless networks have characteristics that make such networks harder to virtualize, e.g., interference, shadowing, multipaths, multiple access, and other aspects of the propagation environment. These difficulties do not mean that radio resources are an exception. Virtualization can expose radio resources to software. Wireless virtualization is also explored on the context of the software-defined radio (SDR) [46]. SDR is a radio where hardware resources are exposed and reconfigured by software. Almost all signal processing can be done in software, creating the so-called virtual radio [43] or software radio [46]. One can expect this kind of mobile terminals on the future Internet.

In summary, network virtualization [47] allows multiple virtual networks (VN) to share the same substrate network (SN). A VN has several virtual nodes connected by physical and/or virtual links, thus forming a virtual topology. Link virtualization enables the network to share one physical link among two or more virtual links. Therefore, the SN can be sliced to enable simultaneous VNs.

More recently, the role of virtualization on future Internet has been expanded. Now, it is being considered as a key ingredient in the core of many approaches. Peterson et al. [35] contended that virtualization can be an elementary aspect of the architecture itself, instead of just a way to test new architectures. Many efforts to reengineer the Internet consider virtualization as the solution to offer generality, isolation, transparency, and programmability of substrate resources.

4.1 Initiatives

PlanetLab had a significant multiplier effect, catalyzing regional PlanetLab-based initiatives worldwide, such as the global environment for network innovations (GENI) [39] in the United States; future Internet research and experimentation (FIRE) [48] and OneLab2 in Europe [8,41,49]; CoreLab [50,51] in Japan; G-Lab [52,53] in Germany; and future Internet testbeds/experimentation between Brazil and Europe (FIBRE) [54], among others. The argument of such initiatives is that designers need prototyping environments to experiment with new ideas, to make proof of concepts, and to test such prototypes under controllable but real conditions.

4WARD [55,56] provides a unified virtualization framework (VNet) aimed to enable physical resources discovery and provisioning, as well as control and management of virtualized resources. VNet considers virtualization of wired and wireless resources.

Another initiative called generalised architecture for dynamic infrastructure services (GEYSERS) [57] aims at extending 4WARD VNet scope to consider optical networking and information technology (IT) infrastructure resources. It deals with the placement of VMs.

The resources and services virtualization without barriers (RESERVOIR) project [58] adopted a virtual entity lifecycle management approach based on service-oriented architecture (SOA). The project proposes to interwork clouds of partner providers to better use available virtualized resources. All the technologies required for collaborating clouds are in the scope of the project. The proposal covers service level agreement (SLA) and lifecycle management.

Another initiative is called SLA at service-oriented infrastructure (SLA@SOI) [59]. It is focused on lifecycle management considering a SLA-based approach. Contracts cover relationships among clients, real and virtual infrastructure providers, and service and software providers.

The NEBULA project [60] proposes a new architecture for the Internet that is based on the high capacity, reliable, and policy-driven interconnection of clouds.

The SAIL and Akari adopt virtualization as core architectural ingredient. SAIL proposes a cloud networking (CloNe) paradigm. The idea is to create dynamic slices of networking that are shared by providers to connect data centers. Network virtualization seems as a tool to facilitate migration and co-existence of legacy and future technologies. Akari provides support for recursive network virtualization.

The Autonomic Network Architecture (ANA) [61] provides recursive support for virtualization based on some generic core ingredients. It enables the federation of virtualized resources, e.g. virtual links.

4.2 Analysis

To discuss the requirements and challenges behind virtualization, let's assume an hypothetical scenario where computational substrate resources are shared among VMs customized for a certain client needs, let's say a video server. A first step on the VMs lifecycle would be to discover and select the appropriate real-world resources that can properly host the VMs. It is desirable that the search for a suitable substrate involves automated solutions, since we are experiencing exponential growths in the number of substrate resources. A possible approach to overcome this challenge is to expose the hardware functionalities, attributes, descriptors, states, configurations, etc., to software. This can be done by means of substrate resources descriptors.

After the selection of an appropriate substrate, it is necessary to check if the selected physical resources can admit or not the new virtual entities. Some negotiation could take place to facilitate admission. If accepted, the VM needs to

be installed and configured. Physical resources need to be reserved. Scheduling and management of processing, storage, and transportation resources should be done according to established contracts.

During operation, the virtual entities demand isolation, security, privacy, and stability. Such requirements give rise to some additional issues: how to balance quality and utilization of real world resources for concurrent virtual entities? How to secure slices from threats coming from other slices? How to make software controls private and secure? How to avoid stability problems caused by other virtual entities? How to isolate effects of software bugs?

Virtual entities also need to be policed, managed, modified, and optimized. Virtual entities can be policed to avoid misbehavior and unfairness. Also, the quality of service (QoS) provided needs to be monitored. Fault situations could be transparently managed. Virtual entities can be optimized to reflect client needs, i.e., service awareness. Finally, the reserved physical resources need to be released when they are no longer needed, such as in the case of the virtual entity movement.

An additional concern is related to the real-time support for virtual entities. It is important to enable deployment of time-sensitive functionalities on software, such as advocated by SDR and CR, i.e. some physical and link layer functionalities (e.g., signal processing, frame delineation, timeout control, and retransmission) require real-time support. This approach uses real-time hypervisors that enable the creation of real-time VMs. Another approach is to use a real-time operating system (RTOS) to implement network functionalities as real-time processes.

Ideally, virtualization of resources should be as broad as possible, covering every type of resources, i.e. not only from the network but also from the cloud. 4WARD covers the virtualization of wired and wireless resources. GEYSERS extend the support for IT and optical networking. SAIL does the same with the CloNe approach. NEBULA and Akari provide a similar scope. However, it is unclear at this time how deep is the support for real-time virtual entities on these approaches.

The virtual entities lifecycle is implemented differently in each of the initiatives presented. RESERVOIR considers a service-centric approach. SAIL combines NetInf (information-centric) and CloNe. Some projects adopt SDN-based solutions. However, in this case virtualization is limited to network resources. Horizon project adopts an autonomic, multi-agent solution. A future direction could be to generalize the lifecycle management to any type of resource, through combining autonomic, virtualization, software-controlled, service- and information-centric approaches.

In summary, virtualization can provide generality in the use/exposure of substrate resources. Virtualization decou-

ples evolution of overlay information networks from substrate resources. It homogenizes, generalizes, and exposes hardware resources to accommodate in software the accelerated technological evolution. However, the abstraction layer created by substrate resource virtualization brings new challenging tasks in terms of scalability, efficiency, virtual entity life-cycling, interoperability, real-time support, isolation, and security.

5 Software-defined networking

The software-defined networking is a new paradigm to redesign communication networks considering a software engineering point of view. The argument is that current networks are essentially designed to “master the complexity” behind existing technologies, rather than to “extract simplicity” from the learned lessons [62]. Shenker defends the idea that abstractions play a big role on computer science, shielding high level software from the complexity existing in the lower levels. Thus, why not to define good abstractions for networking? In this context, SDN means to rethink network architectures considering the important role of abstractions [62]. It is important to notice that in the context of SDR and CR, software-defined (or software-controlled) means that some functionality is defined by software, i.e. it works accordingly to some controlling software. Thus, in SDR and CR context, SDN means to establishment networks where equipment functionalities are controlled by software. This definition comprehends any equipment that could be used to do networking as well as any network functionality. To conclude, both definitions share the software-controllability aspect, since Shenker et al. proposal is also based on software-controlled equipment.

The Shenker et al. SDN paradigm [62] is based on the premise that we have never developed the right abstractions for networking. Thus, SDN proposes four abstractions to simplify network control: (1) forwarding; (2) state distribution; (3) configuration; and (4) specification. The forwarding abstraction encompasses a flexible, software-controlled, frame forwarding model. The state distribution abstraction comprehends a centralized control program that operates over a summarized network view. It avoids the complicated distributed states approach used today in many networks. The output of the control program is a networking configuration map. To create the required network view, a network operating system (NOS) is used. The NOS communicates with the forwarding equipment to obtain state information, as well as to send controls—the realization of the configuration abstraction. The specification abstraction enables the generation of abstract configurations for network devices. Such abstract configurations need to be mapped to the physical ones.

5.1 Initiatives

OpenFlow is probably the best well-known SDN initiative [63]. It is a standard that covers SDN forwarding and configuration abstractions. More specifically, it covers the structure of an OpenFlow switch as well as the protocol used by the control program (controller) to generate the network view and to configure forwarding tables. A diversity of controllers can be used together with OpenFlow: NOX [64], HyperFlow [65], DevoFlow [66], and Onix [67].

A special controller called FlowVisor [68] enables the creation of isolated slices of resources through the orchestration of OpenFlow switches and controllers. In Brazil, a CPqD initiative called RouteFlow [69] enables the creation of virtual IP networks over OpenFlow switches. OpenRoute enables the interoperability with IP networks without the need to required protocols in a centralized way. On wireless media, OpenRadio [70] applies the SDN paradigm to create a wireless network operating system that controls forwarding in an heterogeneous radio environment.

And, if the software-defined networks are programmable, why not to develop a programming language for them? This is the objective of the Frenetic language [71]. Rexford recently presented a discussion on programming languages for SDN [72]. Other recent projects include SDN compilers [73] and debuggers [74].

5.2 Analysis

Virtually all abstractions of Shenker et al. SDN paradigm are currently being investigated. More specifically, the scalability of the centralized-control model is being evaluated [75,76]. New controllers are being proposed to overcome emerging limitations [75]. Controller placement problem is also being explored [77]. The OpenFlow forwarding model has received some criticisms regarding its generality, i.e. limited framing support. Nonetheless, new versions are addressing this limitation. New controller applications enable virtual networks orchestration, but integrated ICT resources orchestration is still missing. The just emerged networking programming languages (and tools) are an exciting new frontier for research. In summary, the current OpenFlow versions can be seen as a first implementation of the SDN idea. Future approaches could emerge considering alternative abstractions and implementations. The SDN paradigm is more deep and broad than what we have implemented today.

6 Autonomicity and manageability

Digital technological development, especially in computing, communications, and data storage technologies, has augmented largely the diversity, quantity, and complexity of

computational systems. Moreover, the pace of technological progress is requiring more adaptable systems, capable of self-adapting to the environment in which they operate. To implement, integrate, install, configure, and maintain large software systems manually is an eminently stressful job and often brings to us a deep sense of failure averse to the problem.

Concerned with this scenario, some IBM researchers published a manifesto in 2001 that gave rise to the so-called autonomic computing. The IBM researchers claimed that the complexity of computing systems is approaching the limits of our capacity to deal with such complexity. Like the human autonomic nervous system, which governs various functions without our awareness, the IBM researchers proposed that computational systems should manage themselves according to high-level objectives outlined by human operators. The idea was to reduce human interference in the system's operation, administration, and maintenance, keeping the system's complexity tractable, reducing operational expenditures (OPEX), and allowing the information technology industry to continue evolving. Therefore, autonomic computing can be seen as a technology to manage complexity, as defended by Strassner [78]. Autonomic technologies frequently emerge as a candidate solution to deal with the increasing complexity on future networks.

The autonomic computing defined by Kephart and Chess [79] has four autonomic properties: self-configuration, self-optimization, self-healing, and self-protection. They are collectively denoted as self-*.³ Autonomous managers implement these properties and interact with each other and with human operators to obtain the expected behavior for the system: the so-called self-emergent or "social" behavior.

Autonomous elements use communication resources to exchange obtained knowledge. Dobson et al. [80] cite this as one of the most notable omissions from the original vision of Kephart and Chess: How do autonomous elements communicate with each other? In the same vein, Clark et al. [81] have proposed incorporating more autonomy in communication networks, creating the so-called network knowledge plane. This idea influenced Fraunhofer FOKUS researcher Smirnov to propose the idea of autonomic communications in 2004 [82,83]. Both approaches make evident the convergence of ICT. The two things are very close: information and its transfer to enable communication.

Besides the four original autonomic properties, Sterrit and Bustard [84] argued that to achieve the goal of autonomicity, the system must be aware of its internal state (self-awareness) and the conditions of the external environment (self-situation); it should also be able to automatically detect changes in circumstances (self-monitoring) and to adapt

appropriately to them (self-adjustment). The autonomous system must then be aware of its skills, available/unavailable resources, internal and external status, communication procedures, and status, as well as its rules, goals, and other high-level information necessary to operate [84]. Context-aware policies and ontologies can be used to deal with high-level objectives [85]. Rules and goals must reflect exactly what infrastructure owners want to obtain. Adjustments are necessary to improve efficacy and avoid instability, modifying existing feedback control loops.

The relationship between autonomic properties and context/situation is captured in the form of *-aware properties. *-aware can be seen as a generalization of contextualized actions in ICT. Take, for example, a service enablement platform as will be discussed in Sect. 7. It is said to be network-aware if it considers the network condition (situation) in its actions.

6.1 Initiatives

In 2003, Clark et al. [81] addressed the need for a new network research objective towards more autonomicity and proposed the concept of network knowledge plane. The idea is to create a self-organizing network that follows high-level goals, reorganizes itself to adapt to changes, self-monitors to detect problems, and self-heals to fix such problems (or explains why it was unable to fix them).

In the following year, the Fraunhofer FOKUS institute established a research initiative in autonomic communications aimed to develop self-* properties for communication networks [86]. Smirnov [83] presented the vision of situated and autonomic communications (SAC) as a paradigm shift towards self-* and *-aware networks to deal with the growing complexity and demands of our information society.

In 2007, the Foundation, Observation, Comparison, Action, and Learning Environment (FOCALE) [87] proposed an entire framework to perform autonomic networking management. The authors of FOCALE proposed two control loops: a maintenance control loop and an adjustment control loop. The approach uses the former loop if policy adjustments are required, while the architecture activates the maintenance loop when it does not detect an abnormal behavior.

The Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services (CASCADAS) is an autonomic, self-similar, pub/sub service orchestration framework [88]. Its fundamental abstraction is called autonomic communication element (ACE). CASCADAS provides a comprehensive autonomic architecture. It supports a diversity of interrelated self-* functionalities. Services can synchronize actions to provide the desired self-emergent behavior. The ACEs can experiment and evaluate new plans, i.e. sequences of actions. The services social behavior is guided by high level objectives. Applications

³ The asterisk is used to generalize all self-management and self-control properties of autonomic and cognitive systems.

can self-adapt to context changes. Some CASCADAS functionalities are related to the RWI, such as knowledge networks, information contextualization, context handling, self-monitoring, self-adjustment, and self-management.

ANA is a self-organizing approach for network elements and its functionalities [61, 89]. Functional blocks can cooperate each other to achieve high level tasks. ANA provides: (1) autonomic monitoring and healing; (2) the dynamic evolution and adaptation of functionalities; (3) an event notification system to provide network- and context-awareness. Importantly, ANA architecture was implemented and tested.

The Akari project employs an automatic locator numbering as well as the optical self-organized control. Self-emergent property is one of the design pillars of the project. Akari aims to create energy-aware solutions.

SAIL is focused on the cross-layer coordination across multiple domains. The aim is to provide service-, network-, cache-, flow-, and energy-awareness. However, it is not clear to which extent it employs autonomic technologies.

Horizon [90] aims to create an automatic management system capable to learn and adapt networking protocols accordingly to network conditions. The project encompasses an “autopilot plane” aimed to help network entities in achieving improved levels of situation-awareness. Thus, entities can better decide on how they will support client virtual overlay networks. It is an autonomic solution to the problem of allocating virtualized resources. Multi agent systems (MAS) were adopted to implement the project’s “autopilot plane”.

Autonomic technologies are also being used in wireless networks, especially in the context of the so-called cognitive radio (CR). Cognitive radio [15, 16] is a flexible wireless communication platform that is aware of its environment (situation awareness), capabilities, and status (self-awareness), and is capable of sensing, analyzing, learning, planning, acting, experimenting, and self-adapting to its environment, spectrum opportunities, and user requirements, according to desired goals, rules, regulations, and policies. Cognitive radio networks (CRN) are self-organizing, cooperative, competitive, dynamic, and efficient spectrum utilization networks that make use of autonomic and cognitive technologies.

6.2 Analysis

Distributed, self-*, and *-aware architectures and frameworks have already been demonstrated and evaluated. Clear examples are ANA and CASCADAS. While the autonomic design is being adopted to a multitude of specific problems, designing a new Internet deeply employing autonomic ideas is a quite challenging task.

Autonomic control and management requires a clear vision not only of the internal states of the managed entities but also from the environment where they are inserted.

Therefore, there is a strong relationship with the real-virtual world integration aspect discussed on the Sect. 3. Autonomic functionalities can be seen as a client of the rich and contextualized information obtained at the real world.

Autonomic technology is a candidate approach to manage the exponential growths on future Internet numbers. The “autopilot” is already a reality in many knowledge areas. Therefore, it is an important candidate to deal with the scale and complexity we can expect on future networks. The role of autonomicity as an antidote for complexity is discussed on [91].

7 Services and applications: extendibility, flexibility, compose-ability, and usability

The end-to-end principle is one of the central principles of the current Internet. It states that no application level functionality can be placed at the network layer. IP design was minimal: the “dumb network” with “smart hosts” model. This model favored the neutrality, innovation potential, openness, diversity, extendibility, and flexibility of network applications. The Akari project [4] defends the world wide web (WWW) as perhaps the most important outcome of this principle.

Such history can repeat itself, but in another scale! Nobody knows for sure what will be the most successful applications in a few decades. Therefore, a generic (usage-independent) information processing and exchanging infrastructure is required, where the coexistence of evolvable, extendible, flexible service/application frameworks over such generic information network is achieved.

Software design is changing from component-based to service-oriented design, giving rise to what has been called service-centrism, e.g., service-oriented architecture (SOA) [92]. The idea is that applications can be flexibly and dynamically constructed by the composition of distributed software services or utilities.

The life cycling of service/applications is dynamic, distributed, and cross-domain. It starts when a new service-based application is invoked. It also involves the search, discovery, and selection of candidate services. Third-party software, which would not be under the control of developers, can be used.

To facilitate compose-ability, it is necessary to seamlessly describe services, publish, discover, and negotiate. In an analogy with nature, where the colors of flowers help to attract pollinators, the descriptors of services will be important to facilitate the selection of the most appropriate services to compose a given application. Examples of attributes are availability, security, quality of service, cost, and usability [93]. Such service information could be published in services responsible for promoting other services.

Besides service selection, negotiation will be necessary to establish a service-level agreement (SLA) or a service binding. During the negotiation phase, an admission control should be performed to verify if resources are available to attach the desired service to one more application. If yes, admission installation proceeds to configure the service.

To assure that the desired quality is met, service monitoring is necessary, as are logging and exception handling. Thus, a lot of service management functionality is required, e.g., to deal with failures, accountability, quality, availability, resilience, etc. Therefore, autonomic service management is indicated to reduce human intervention and OPEX.

Finally, changes in the application behavior can be reflected in the inclusion or elimination of participating services, as well as in SLA changes. When the application turns off, used resources must be released. Notice how this life cycle looks like the cycle of virtual entities in Sect. 4.

7.1 Initiatives

For many people, the service-centrism paradigm will be dominant in the upper portion of a new Internet. The main reason is that above some level of abstraction, any functionality can be viewed as a service, leading to the concept of the Internet of services (IoS) [94–96]. The IoS is of great importance to the service sector of the economy. In Europe, for example, approximately 70% of gross domestic product is related to services [97].

The IoS is an umbrella to the related concepts of everything as a service (XaaS) [98], e.g., from cloud computing; software as a service (SaaS) [99], which delivers cloud applications as a service; platform as a service (PaaS) [98], which delivers a platform and/or solution stack as a service; and infrastructure as a service (IaaS) [100], which delivers computer infrastructure (cloud of virtualized resources) as a service.

The networked European software and services initiative (NESSI) [101] is committed to realize the IoS paradigm by implementing an open service framework (OSF). The OSF relies on the combination of SOA, virtualization, and autonomous software [5]. In a nutshell, the objectives are: (1) to enable the dynamic composition of context-aware services and XaaS; (2) to decouple software components from the substrate resources; (3) to facilitate deployment and reduce human interference; (4) to deal with cross domain (seamless) services and infrastructures; (5) to enable service self-adaptation according to context and semantics; (6) to enable context-aware personalized experience for users; (7) to improve security, trust, and dependability; (8) to manage service-lifecycle.

The first step in implementing the OSF was under the responsibility of the NESSI open framework reference architecture (NEXOF-RA) project [102]. This project speci-

fied a reference architecture to be used on NESSI compliant designs. Other NESSI projects are: SOA4ALL [103], RESERVOIR [58], and SLA@SOI [59].

The SOA4ALL approach integrates context-aware technologies, semantic web, and web 2.0 with SOA in order to create a framework and software infrastructure for seamless and transparent service delivery [103]. The idea is to create a scalable “service web”—a global service delivery platform, where a huge number of stakeholders can expose and consume services. SOA4ALL Studio encompasses services life-cycling management, user friendly service composition, provisioning, and analysis. The idea is to support the lightweight process modeling (LPM) paradigm to enable ease-to-use business process modeling, search for appropriated services, and automatic composition.

CASCADAS offers a versatile service-framework focused on the self-organization and self-adaptation of services. The framework was designed considering a set of fundamental ingredients: situation-awareness, semantics self-organization, and self-similar modular design. Services can form topic-specific clusters through dynamic SLAs, giving rise to service-based, self-assembly applications. In other words, services can synchronize actions to provide the desired self-emergent behavior. Services can adapt themselves to changes in context. Through this feature, services can become specialized in a certain topic. They can also self-monitor, self-heal, and self-protect against threats. Whenever they are instantiated, services can establish connections with other services and initialize self-* algorithms.

Akari project provides the support for service overlays and context-aware services fed by sensor networks. ANA facilitates service searching, placement, and advertisement towards self-organizing applications. SAIL offers support for generic service frameworks that will run over the cloud networking (CloNe) with network-awareness support.

Finally, there is the semantic web effort. It is an idea advocated by Berners-Lee [104]. It aims to define the meaning for information and how to treat it, so that the web can “understand” what people and machines want. The idea is to make an autonomous knowledge web, which includes context-aware applications and service compose-ability. The proposal argues that to enable customized experience, the web needs to evolve to the semantic level.

7.2 Analysis

Substrate resources, such as communication, processing, storage, and others, need to be properly exposed for overlying frameworks to allow the compose-ability and orchestration of services and applications, as well as management of their life cycles. Services and applications are information treatment processes, dynamically instantiated, from available descriptors/metadata. Virtualized resources could be

customized to adequately support service needs, creating resource-aware services and applications. Every information processing functionality is seen as a service, including network functionalities. The integrated and synergistic resource, service, and application orchestration creates a bottom-up environment better aligned with the overall needs of our information society. Considering this vision, one can say that NESSI, CASCADAS, and SAIL initiatives are aligned to it. However, many other proposals presented in this paper are misaligned with this vision—posing an insurmountable barrier between services and network. For many people, it is difficult to accept the vision that some networking features can be implemented as any other service, therefore subject to search, discovery, negotiation, contracting, etc. Moreover, many software solutions are superficial to address the required support for this purpose. Support for real-time and high performance computing are some of the concerns. Thus, it is still unclear today how the joint orchestration of physical and virtual resources will take place in a future Internet.

From a user's perspective, the benefits of the IoS are very promising: (1) self-servicing capabilities—users can themselves configure exactly what they want; (2) improved usability—personalization and contextualization (context awareness) can be achieved in applications, varying features according to user preferences; (3) semantic invocation—services are invoked, managed, and adapted depending on semantic information; (4) user-designed applications—users will be able to create their own applications and export them to their friends; (5) diversity—some research projects expect a huge diversity of services, e.g., billions of customized services and applications [105]; (6) better resource usage, reducing energy fingerprints.

In summary, the idea of dynamic service compose-ability could be used to integrate business processes with applications, services and exposed substrate resources, creating what is being called digital business ecosystems (DBE) [106]. The DBEs could evolve to a “digital savannah”, where a diversity of services, applications, business processes, operators, users, enterprises, stakeholders, and other entities will “artificially live”, compete, collaborate, exchange information, “die”, and evolve together.

8 Information

The current Internet was designed in an era in which technological development was completely different from that of today. Memory, processing, and communication resources were very limited when compared with present resources [107].

In this scenario, the principles selected to guide the design of the Internet focused on inter-terminal connectivity through routers (host-centrism); designing of a simple (but robust)

forwarding network, in which more complex features were left to the upper layers at the terminals (end-to-end principle); and on end-to-end communications among computer applications.

The openness, flexibility, neutrality, diversity, and extensibility of applications generated by such principles led to the emergence of the WWW and the popularization of the Internet. This movement led to the transformation of the Internet as the main infrastructure for information exchange, and it is considerably changing the way we interact with content [108]. As Internet designer Van Jacobson recently declared [109], the web changed the way people communicate forever—what matters is the content, not how to take it or where it is.

While the endpoint-centrism has produced tremendous success in the last decades, many researchers believe it is now time to put the spotlight on the information, originating the so called information-centric approach [110]. The idea is to consider information as a key ingredient in design, since information is everywhere, i.e., contracts, location, police, IDs, descriptors, naming, etc. The motivation is to overcome Internet limitations to support content distribution and exchange in a coherent way.

8.1 Initiatives

Since 2006, many info-centric approaches have emerged to overcome the limitations and distortions caused by host-centrism. Some examples are the network of information (NetInf) [17, 56, 111, 112], publish/subscribe Internet routing paradigm (PSIRP) [113] and content-centric networking (CCN) [114].

The main idea behind such blueprints is to make information the center of design [115], representing it persistently and consistently. Such representation could be done indirectly by means of information objects (IO) [111, 113] or directly by means of immutable names as done in CCN design [114]. IOs could contain information metadata such as digital signatures, checksums, access rights, formats, and ontology. CCN names have three portions: a flat portion that contains the data itself or a checksum, a versioning and segmentation portion, and a hierarchical portion contemplating the domain name where the information is, i.e., provenance information. On the other hand, NetInf and PSIRP use flat names. Flat names are typically opaque (non-legible). The core idea behind such naming schemes is to allow access to information independently to where it is located as well as to adequately manage content with different versions and encodings. The matter of content copies is also a concern. Therefore, at some point, the mapping between the information name and the location of their copies will need to be resolved.

The opportunity to rethink Internet design from the point of view of information is also being used to propose alter-

natives to the current “receiver-accepts-all” communication model. PSIRP [115] proposes a publish/subscribe (pub/sub) paradigm. The goal is to enable efficient anycast and multipath routing of previously located information, to improve multicast and to efficiently distribute content, customizing, and improving quality of experience (QoE). Anycast support can be achieved by locating the nearest copy of a published content. NetInf also adopts the pub/sub paradigm. IOs are published/subscribed based on global unique IDs. On the other hand, CNN adopts an approach that resembles the hypertext transfer protocol (HTTP). However, it is implemented at the network layer. Interest packets disseminate the desire for some specific content (resembles the “get” method in http). The response packet contains the desired data (similar to the http “response” method). The packets do not contain addresses, but names. The CCN protocol delivers content, instead of connecting hosts. Packets are cached at CCN routers. Thus, future interest packets can be answered by the closest cached packets.

CASCADAS, Akari, and ANA can support overlaid content networks, but they are not information-centric designs. The CASCADAS service discovery mechanism is based on a publish/subscribe protocol—it employs self-description mechanisms to enable semantics orchestration of services.

8.2 Analysis

NetInf, PSIRP, CCN, among other information-centric initiatives are proposing solutions to many of the challenges behind this paradigm shift. A highly abbreviated list would be the following: (1) to temporarily store information in the network, c.f. caching; (2) to allow semantics-rich and context-based information search and manipulation; (3) to deal with locality, provenance, ontology, and coherence of information; (4) to rethink security from the point of view of information, securing information per se as a means to improve information reliability, integrity, and traceability; (5) to provide secure rendezvous among information producers and consumers, using trust relations; (6) to verify publisher privacy before content publishing as well as to authenticate and authorize subscribers during rendezvous; (7) to solve indirections dynamically, efficiently, and robustly; (8) to deal with content on multilevel, multidomain environments; (9) to deal with the scalability of all these mechanisms; (10) to enable anycast and multipath routing of previously located content; (11) to uniquely identify information; (12) to explore self-certifying names, i.e., names that contain the result of a cryptographic hash function over the binary data.; and (13) to provide autonomic manipulation of content based on high-level policies, goals, privacy, objectives, etc. [114, 116].

Note that these information-centric approaches require dozens of information-related software functionalities. From the service-centric point of view, such functionalities are no

longer different from any other service. On the other hand, the service-centric approaches require dozens of specific information, such as descriptors, identifiers, names, contracts, goals, etc. It is clear that both approaches are complementary. Thus, what could we expect from the current panorama of research? More synergy among them! However, this is not the case. Apparently, there is very little interaction between the two proposals, leading to establishment of non-optimal solutions. Every side is reinventing the wheel when covering aspects from the other side.

Recently, a convergent paradigm was proposed to merge both approaches: the Internet of information and services (IoIS) [117]. In fact, this paper is the first step of a broader work that aims to create a new architecture integrating service- and information-centric approaches. This research project is called NovaGenesis and started in 2008 at Inatel, Brazil.

9 Indirection resolution

Computer scientists have long discussed the role of indirection in software development. In 1972, Butler Lampson contended that all problems in computing can be addressed by including another level of indirection. The problem is that the number of levels can be very large and hence efficiency drops. Another impact may be in scalability, since the greater the number of levels, the greater the need to store and to resolve indirections.

9.1 Initiatives

Indirection appears in the current Internet as well as in several proposals to design new network architectures. Siekkinen et al. [118] argue that indirection is another point where the current Internet lacks adequate support. The information-centric approaches of NetInf [111] and PSIRP [113] apply indirection to decouple information objects from their storage sites. Autonomic network architecture (ANA) [61] uses generic indirection systems—information dispatch points (IDP)—to store a lot of information, including bindings between functional entities to create an evolving protocol stack. Internet indirection infrastructure (i3) [119] uses indirection principle to support mobility and multihoming in the current Internet. The ANA IDP has some resemblance to i3, but the main focus is on a clean-slate approach. Host identification protocol [120] creates an indirection layer between host IDs and locators. In fact, indirection resolution systems are used on the majority of ID/Loc splitting, name resolution, and information-centric approaches.

9.2 Analysis

Indirection resolution is present everywhere on ICT architectures. It is present on domain name service mappings, on open systems interconnection layers' service access points, on different technologies' address mappings, on input–output port mappings, etc. It is also present on approaches to the future Internet: (1) when decoupling the hosts' identification from location; (2) on information-centric approaches to resolve information ID/Loc mappings; (3) on substrate resource virtualization to store mappings among real-virtual entities; and (4) on service-based applications to map among participants entities. Therefore, designs need to consider indirection resolution in a more comprehensive way, creating mechanisms to generically store mappings among entities, services, identities, locators, etc. Indirection resolution is also important to facilitate the search and discovery of functionalities, entities, and information. By resolving indirections and analyzing mappings, architectural functionalities can search for resources, applications, and event content.

10 Identification, location, mobility, and multihoming

Another important point today is that not only hosts are identified and located based on IP addresses, but also that information is, since uniform resource locators (URL) have domain names, which in essence lead to IP addresses. The reason is that current Internet design defined a dual functionality for IP addresses: to identify hosts, nodes interfaces or content servers, and to locate them into the network. The original Internet design does not support mobile hosts.

The shortage of valid IPv4 addresses in the Internet give origin to the network address translator (NAT) as a solution to enable the reuse of IP addresses. As a consequence, IP addresses are frequently changed, generating some sort of “identity loss” that ultimately may lead to inconsistencies in information and host localization. User, information, and host traceability is greatly affected by this situation, since the relation between user profiles and IP addresses is difficult to obtain. In addition, such frequent changes limit mobility, localization, roaming, and multihoming support on the Internet [121].

As discussed in Sect. 2, who would not want to move and take all services and information with no loss of quality of experience? The rapid growth in the number of mobile devices puts support mobility as one of key issues in future Internet design. The requirement is to comprehensively support user, terminal, service, application, virtual network, information, and other real and virtual entity mobility. The challenge behind this idea justifies why unique identifiers are needed: it is because we want to move real or virtual entities as well as information, without loss of identity, and we want

to continue finding them during and/or after movement. People want to move without loss of identity and functionality. Services and applications need to follow users as they move.

Moreover, it is necessary to support redundant connectivity for fixed and mobile devices, i.e. to provide multihoming support. Multihoming is limited on the current Internet and is based on four pillars: physical link redundancy, switching/routing redundancy, routing path redundancy, and host functionality duplication. In future Internet, multihoming needs to be rethought to more generally accommodate redundancy on network access.

10.1 Initiatives

In the last years, several initiatives appeared to support host mobility on the Internet [122]. Mobile IP approach [123] proposes two IP addresses for the hosts: a home-address, which is static and works as an identifier (ID) for the application layer; and a care-of-address, which is dynamic, dependent on node's location. Mobile IP requires two components: a home-agent, which allocates the home-address and maintains a mapping to current location; and a foreign-agent, which allocates the care-of-address, informing the home-agent in case of mobility. Instead of using just one IP address to identify and locate the host on the network, Mobile IP decouples identification from location functionality, using one address to identify the host (home-address), and another one to locate it (care-of-address). This functionality decoupling is noted as ID/Loc splitting. Another approach, host identity protocol (HIP) [124] creates a new namespace between Internet network and transport layers. The identifier is a flat public key attributed to the host. The locator is an IP address.

There is also the locator ID separation protocol (LISP) [125]. It is based on an address mapping system between edge and core IP networks. Edge datagrams are encapsulated on UDP messages and further accommodated on core network datagrams. Two IP addresses are used: endpoint identifiers (EIDs), which are persistent and used as an ID for the hosts; and routing locators (RLOCs), which are used to locate the edge routers. Ingress tunnel routers maps EIDs on RLOCs, while egress tunnel routers maps RLOCs on EIDs.

The Mobile IP and LISP approaches can be classified as patches to the current Internet, since they are intrinsically tied to its design. However, in the recent years, new approaches appeared to rethink the mobility problem more deeply, no longer being necessarily tied to TCP/IP stack. In fact, some of them can be used with IP addresses, non-IP, or post-IP. Nonetheless, identifier-based solutions are being preferred to support mobility in future Internet [126].

Akari [4] adopted an ID/Loc splitting solution to support mobility and multihoming of real world equipment. It creates a new namespace between network and transport layers: the ID layer. Host identification can be performed in two ways:

by a readable, unique local name; or by an identifier obtained as the result of a hash function. The names can be local or global. Local names are unique on the local network and are used for host identification and network management. Local names are generated by the combination of representative host-related words, i.e. their function in context, owner, serial number or date and time of installation. Global names include the local names, as well as additional hierarchical topology information. The overall solution is based on three mapping systems: identity management server (IMS), name mapping server (NMS), and location management server (LMS). The IMS holds the local dynamic mapping among names, IDs, and locators. The NMS deals with global static information mapping, like in the current DNS. More specifically, the NMS provides the mapping between the global part of the nodes names to the locators of domain specific identity management servers (IMSS). The LMS maps IDs to global locators.

MobilityFirst aims to redesign the Internet through the comprehensive support for mobility, trust, robustness, usability, and manageability [1, 127, 128]. The approach adopts a global name resolution service called direct mapping (DMap) [129]. Entities are supposed to have a flat ID that is bound to one or more locators. The DMap service maps an ID to a list of network addresses where the ID/Loc bindings are stored. Thus, every autonomous system (AS) will have ID/Loc bindings of other ASs, sharing the load of the bindings hosting.

The mobile-oriented future Internet (MOFI) proposes mobility as a central aspect on future Internet design. It encompasses the integration of ID/Loc splitting, host ID-based communications, dynamic and distributed mapping system (DDMS), and a query-first data delivery (QFDD) [130]. Hosts are identified by 128-bits IDs, while layer 2 or 3 addresses can be used as locators. End-to-end communication is based on host IDs (HIDs). Locators only need to be unique in the local network. The DDMS has two-levels: intra-domain and inter-domain. If communication is restricted to a domain, the search for the appropriate locator is done among edge routers (ERs). On the contrary, searching is performed by domain gateway ERs. On initialization or in the case of movement, every host should inform its location to the DDMS. Thus, when data delivery takes place, the sending host queries the DDMS about the destination host before sending any data.

SAIL adopts an ubiquitous, seamless, and transparent mobility approach. Mobility support is not restricted to hosts, it covers content (based on NetInf) and virtual entities (based on CloNe), e.g. processes and VMs. NetInf employs unique identification and decoupling from locators. Names are flat, persistent, non-legible, authenticated, and self-certifiable. At the open connectivity services (OConS) level, SAIL employs a dynamic, distributed mobility management solution. The solution is flow-based and includes information collection, network selection, path selection, handover decision making,

execution, enforcement, and optimization. More specifically, a modified dynamic mobility anchoring technique is adopted [131].

ANA proposes a global identifier management framework called iMark, which enables decoupling of resources identifiers from locators.

10.2 Analysis

When identifiers are decoupled from locators, it is possible to move things without “loss of identification”. Thus, when a terminal moves from a geographic region A to B, locators change, but identifiers remain the same, allowing all the other functions to work properly [132]. This idea could be extended to uniquely identify every logical or physical entity in the network as well as information, so they can be moved, searched, and located without changing their identifiers. This means that identifier-based mobility solutions demand ID/Loc splitting as well as indirection resolution, i.e. dynamic mapping between identifiers and locators.

The benefits of ID/Loc splitting are many: IDs become persistent, enhancing accountability; traceability based on persistent IDs discourage network misuse; unique IDs enhance digital credentials; IDs help to authenticate and authorize entities; persistent IDs enable autonomic security mechanisms; granted trust relations could be established among entities based on IDs; and mobility support becomes ubiquitous.

However, ID/Loc splitting brings important challenges: (1) how to generate unique digital identifiers for real or virtual entities; (2) how to comprehensively manage ID/Loc bindings in order to provide mobility for real or virtual entities; (3) how to manage the large number of IDs, their relationships, and life cycles; and (4) how to manage credentials and their relations to IDs, including how to discover the IDs of real or virtual entities.

Many of these challenges are being addressed by the cited projects. Identifier generation is frequently related to naming, since legible and hashed names can be used as identifiers if they are unique in some scope [133]. In addition, self-certifying identifiers are preferred due to their intrinsic security properties. Akari, MobilityFirst, MOFI, and SAIL adopt this approach. However, the ID/Loc splitting should cover all inhabiting entities. In this regard, many approaches are limited. Akari, MobilityFirst, and MOFI limit the ID/Loc splitting support to the physical devices. ANA covers virtual network entities. But, only SAIL covers physical devices, virtual networking, services, and information ID/Loc support in the same architecture. SOA4ALL and CASCADAS does not support ID/Loc splitting.

The global scalability problem behind the ID/Loc bindings storage/recovery is also addressed by the mentioned projects. In MOFI, a distributed global domain-based map-

ping system is adopted for inter-domain ID/Loc binding storage/recovery. SAIL considers the idea of hierarchical name resolution systems and provides a specific implementation for the problem: the multi-level distributed hash table (MDHT) system. Akari divides the problem in three distributed systems: IMS, NMS, and LMS. They provide dynamic local mappings among names, IDs, and locators (IMS); static global mapping between names and IMS locators (NMS); and identifier to global locators mapping (LMS). MobilityFirst DMap's approach shares ID/Loc bindings storage responsibility among autonomous systems using current network addresses. The global scalability of these solutions still demands to be demonstrated in practice.

Regarding multihoming, new designs need to cover it comprehensively. Functionality redundancy can be achieved by means of virtualization, i.e., virtual links and nodes. In this case, virtual nodes need to be duplicated to allow hot swapping. Link layer connectivity needs to be improved, taking advantage of pervasive and ubiquitous computing. SAIL, Akari, MOFI, MobilityFirst, among others support improved physical connectivity. Cognitive radio is also interesting as a means of establishing multiple physical links to other radios. SAIL and Akari consider cognitive radio support. Multipath, concast, and anycast routing are also necessary. Concast routing enables information transfer from multiple sources, through multiple paths, improving user experiences. The anycast routing allows transfer of the desired content from the nearest source, reducing delay. SAIL provides a multipath/point/protocol (Multi-p*) routing.

11 Security, privacy, trust, and transparency

It is nothing new to claim that the current Internet has critical deficiencies with respect to security, privacy, transparency, and accountability. In its early days, the Internet was controlled by a small group of trusted institutions with restricted access. The institutions and their computers were considered reliable. Today, the Internet has reached an unprecedented scale and the computers that are part of this network are mostly untrustworthy. The list of vulnerabilities, weaknesses, and attacks is long: computer viruses, worms, trojans, spyware, dishonest adware, phishing, spam, exploits, spoofing, code injection, fraud, etc. Everyone is a victim of such threats, despite existent protection tools. There is a growing sense of powerlessness in the face of so many problems. Few know what to do when the operating system displays a window indicating a problem—often people are unaware of the risks. Attackers have already affected large networks, creating unavailability, loss of revenue, and fines.

The challenge behind security and privacy support in future networks is getting worse because people are increasingly exposing themselves through social networks, videos, photos, instant messaging, blogs, etc. Moreover, there is the

increasing monitoring of the real world as expected on the IoT and RWI. Real- and virtual-world information is being massively cached and used without permission on the Internet, leading to loss of privacy, freedom, and security. Imagine what could happen if the attackers began to exploit RWI vulnerabilities—damages could affect real-world appliances.

11.1 Initiatives

Security issues are considered a key aspect for all future Internet initiatives. Therefore, all projects somehow cover the subject. Let's consider first the Akari project. It defines a general framework for network security, which is divided in three levels: device, infrastructure, and service security. Akari's framework adopts host ID-based security associations to improve traceability. Transport security includes confidentiality and integrity of packets, as well as sender authentication. Trust formation and enforcement is also supported. Cooperative mechanisms are embraced to deal with distributed threats.

On the other hand, ANA provides public and private (authenticated) access to information dispatch points. It supports autonomic security, i.e. allows the dynamic composable of security mechanisms. The iMark framework provides ID-based traceability among physical and virtual network entities.

However, SAIL considers security and privacy as a "theme" that permeates the entire architecture. Security objectives are expressed by security services taking into account that security and privacy can impose contradicting issues. Security services encompass security policies (goals, action methods), security modeling (entities and their relationships), architectural security principles, and security mechanisms. The initiative also tries to avoid overdimensioning the security support. OConS allocates substrate resources among clouds in a seamless, isolated, and secure way; while NetInf secures information per se.

CASCADAS embraces an autonomic social security as a service paradigm. It offers mechanisms for trust, reputation, self-preservation, self-healing, intrusion detection, self-monitoring, and self-protection.

Finally, there are the projects focused exclusively on security problematic. Some of them are related to the management of identity, privacy, trust, dependability, and risk. Others are focused on validation, monitoring, enforcement, and auditing of security policies. Some examples are: privacy and identity management for community services (PICOS) [134,135]; PrimeLife [136,137]; Think-Trust [138]; managing assurance, security and trust for services (MASTER) [139]; trusted architecture for securely shared services (TAS3) [140,141]; automated validation of trust and security of service-oriented architecture (AVANTSSAR) [142,143]; and trusted embedded computing (TECOM) [144].

11.2 Analysis

Considering the outlined initiatives, how do we improve security in a new Internet? The first thing to do is to consider security, privacy, trust, and transparency from the beginning of the design—they must be built-in or inherent. The fact is that one or more of these issues are generally left to a further stage of the design or to other approaches, making it much more difficult to meet pre-requirements and to eliminate inconsistencies. For example, CASCADAS, NEBULA, OpenFlow, SOA4ALL, among others do not provide self-certifying unique identifiers for entities. Akari and MobilityFirst provides it just for physical devices. ANA provides generic identification for physical and virtual entities. In SAIL, the info-centric approach adopted in NetInf provides information objects traceability based on self-certifying IDs. According to Pan [8], “self-certifying and hash-based addresses are effective tools for security”. Therefore, self-certifying identifiers should be considered at all architectural levels, preferably without relying on external proposals or frameworks.

Another paradigm that could help to improve security is the establishment of trusted networks among entities. Trust networks would not be restricted to services or network devices, but comprehensively include as much as possible entities. Akari calls this concept of trustworthy network [4]. CASCADAS supports services social control based on trust and reputation mechanisms. NetInf information objects allow content trust assertions. NEBULA policy-driven design concerns to transport path trustworthiness. Due to the benefits of establishing comprehensive trust networks, it is highly desired that architectures have explicit support for this purpose. Open issues are multidomain and multilevel trust support [5], as well as to help users protect and preserve their privacy by controlling the establishment of trust relationships.

It is also relevant for new architectures to manage identities, credentials, and reputation. The authors of [5] point to the protection of user credentials and ID management as issues to be addressed in a new Internet. People must trust not only in the network but also in its entities. In this sense, some questions emerge: (1) How do we supply anonymity for users that desire it, while maintaining accountability when legal issues are entitled to investigation? (2) How do we evaluate trust and reputation? (3) How do we determine entities’ dependability on trusted parties? (iv) How do we create intuitive risk announcements? (5) How do we monitor and police postures to determine the trustworthiness of entities? (6) How do we identify, assess, monitor, analyze, and sort risks, vulnerabilities, and threats? To answer this questions, the precise relationships among identities, identifiers, trust, credentials, and reputation need to be determined.

Finally, there is the increasing complexity behind the security support. Autonomicity could be considered, like

proposed on CASCADAS or Akari. The network must deal proactively with distributed massive attacks and unpredicted vulnerabilities and threats. Self-emergent solutions based on entities social control are promising approaches.

12 Simplicity, evolvability, and sustainability

Last, but not least are the issues concerning design simplicity, evolvability, and sustainability.

To call attention to how difficult it is to design with simplicity, one can quote Leonardo Da Vinci: “Simplicity is the ultimate sophistication.” Or Einstein: “Make everything as simple as possible, but no simpler.” These quotes illustrate that the architects of a new Internet should simplify their designs to its essence. This is perhaps the greatest requirement/challenge in designing the future Internet, since to simplify it is necessary to have a comprehensive view of all aspects of the future Internet and their complex interrelationships. Simplification of integrated technologies is one of the concerns in the Akari [4] project.

Two other relevant requirements are evolvability and sustainability. Evolvability is related to biological systems. According to Rowe and Leaney [145], evolvability is the ability to self-adapt to changes in the environment, pre-requirements, or technology changes. Accordingly, this definition has a close relationship with the autonomic technologies previously discussed.

Sustainability can be defined as the property of maintaining a certain level/situation in the course of time. In fact, to have a sustainable design, it is necessary to accommodate technological change, as discussed in Sect. 2. Akari also aims to project a sustainable network, capable of evolving and supporting the requirements of the information society in the next decades.

13 Conclusions

Despite the diversity of ideas, requirements, and challenges behind the future Internet design initiatives, from a conceptual point of view there are many similarities among them. This paper addressed this common aspects in order to summarize the current state-of-the-art from an abstract, but coherent point of view. This generalized understanding can guide the future Internet architects towards more broad and comprehensive blueprints, capable of achieving global architectural gains—instead of local ones. A future work is to determine how each of these selected technologies could be combined to meet overall project pre-requirements, how they interrelate with each other creating dependencies, and how every technology can take deep advantage of the others. This is a possible path to simplify the future Internet architecture design to its essence, while maintaining the scope as general as required by our information society.

Acknowledgments The author thanks INATEL for the research support.

References

- Pan J, Paul S, Jain R (2011) A survey of the research on future internet architectures. *IEEE Commun Mag* 49(7):26–36. doi:10.1109/MCOM.2011.5936152
- Aoyama T (2009) A new generation network: beyond the internet and ngn. *IEEE Commun Mag* 47(5):82–87
- Luo J, Pettit J, Casado M, Lockwood J, McKeown N (2007) Prototyping fast, simple, secure switches for etha. In: High-performance interconnects. *HOTI 2007. 15th annual IEEE symposium on*, pp 73–82. doi:10.1109/HOTI.2007.7
- Harai H, Inoue M, Otsuki H, Kuri T, Nakauchi K, Furukawa H, Miyazawa T, Xu S, Kafle VP, Umezawa T, Ohnishi M, Fujikawa K, Li R, Andre M, Decugis S, Peng C, Yagi H, Kamiya M, Murata M, Teraoka F, Hiroyuki, Morikawa, Nakao A, Ohta M, Imaizumi H, Hosokawa M, Aoyama T (2010) New generation network architecture akari conceptual design, Tech. Rep. v1, NICT
- Abramowicz H, Arbanowski S, Alvarez F, Andrikopoulos I, Bassi A, Bisson P, Bourse D, Boutroux V, Cardinal B, Cavanillas J, Cisneros G, Clarke J, Danet P-Y, de La Iglesia F, Fairhurst G, Gidoïn D, Gil G, Gerteis W, Hierro J, Hussmann E, Huth H-P, Jenkins P, Jensen T, Jimenez J, Kennedy D, Liolis K, Martinez R, Mascolo S, Menduina E, Meunier J-D, Mohr W, Oliphant A, de Panfilis S, Papadimitriou D, Point J-C, Posada J, Riedl J, Salo J, Smirnov M, Williams F, Zseby T (2009) Future internet: the cross-etc vision document, Tech. rep., Cross-ETP
- Rexford J, Dovrolis C (2010) Future internet architecture: clean-slate versus evolutionary research. *Commun ACM* 53(9):36–40. doi:10.1145/1810891.1810906
- Stuckmann P, Zimmermann R (2009) European research on future internet design. *IEEE Wirel Commun* 16(5):14–22. doi:10.1109/MWC.2009.5300298
- Paul S, Pan J, Jain R (2011) Architectures for the future networks and the next generation internet: a survey. *Comput Commun* 34: 2–42
- Chatterjee P, Doering R (1998) The future of microelectronics. *Proc IEEE* 86(1):176–183. doi:10.1109/5.658769
- Kurzweil R (2006) *The singularity is near: when humans transcend biology*, A Penguin Book Science, Penguin
- Saracco R (2009) Telecommunications evolution: the fabric of ecosystems. *Revista Telecomunicaç oes Inatel* 12 (2):36–45
- T. 500, Japan reclaims top ranking on latest top500 list of world's supercomputers (2011) [cited June 2011]. www.top500.org/lists/2011/06/press-release
- University of Minnesota, Minnesota internet traffic studies(2009) [cited May 2011]. <http://www.dtc.umn.edu/mints/>
- Huffington, internet usage statistics (2010) [cited June 2011]. www.huffingtonpost.com
- Mitola J, Maguire G (1999) Cognitive radio: making software radios more personal. *IEEE Pers Commun* 6(4):13–18. doi:10.1109/98.788210
- Haykin S (2005) Cognitive radio: brain-empowered wireless communications. *IEEE J Sel Areas Commun* 23(2):201–220. doi:10.1109/JSAC.2004.839380
- Ahlgren B, Aranda P, Chemouil P, Oueslati S, Correia L, Karl H, Sullner M, Welin A (2011) Content, connectivity, and cloud: ingredients for the network of the future. *IEEE Commun Mag* 49(7):62–70. doi:10.1109/MCOM.2011.5936156
- Commission E (February 2011) Fp8 expert group: services in the future internet, workshop report, European Commission
- Zahariadis T, Papadimitriou D, Tschofenig H, Haller S, Daras P, Stamoulis GD, Hauswirth M (2011) Towards a future internet architecture
- Weiser M (1993) Hot topics: ubiquitous computing. *Computer* 26(10):71–72
- Presser M, Daras P, Baker N, Karnouskos S, Gluhak A, Krco S, Diaz C, Verbauwhede I, Naqvi S, Alvarez F, Fernandez-Cuesta AA (2008) Real world internet. Tech. rep, Future Internet Assembly
- Conti JP (2006) The internet of things. *Communications Engineer* vol 4
- Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. *Comput. Netw.* 54:2787–2805
- Presser M, Barnaghi P, Eurich M, Villalonga C (2009) The sensei project: integrating the physical world with the digital world of the network of the future. *IEEE Commun Mag* 47(4):1–4. doi:10.1109/MCOM.2009.4907403
- Kaiser U, Steinhagen W (1995) A low-power transponder ic for high-performance identification systems. *IEEE J Solid-State Circuits* 30(3):306–310. doi:10.1109/4.364446
- Walko J (2005) A ticket to ride [near field communications]. *Commun Eng* 3(1):11–14
- Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. *IEEE Commun Mag* 40(8): 102–114. doi:10.1109/MCOM.2002.1024422
- Nikolaidis I (2008) Sensor networks and the law of accelerating returns [editor's note]. *IEEE Netw* 22(4):2–3. doi:10.1109/MNET.2008.4579763
- Trakadas P, Zahariadis T, Leligou H, Voliotis S, Papadopoulos K (2008) Awissenet: setting up a secure wireless sensor network. In: *ELMAR, 2008 50th international symposium*. 2:519–522
- McLaughlin GT, Liu LY, DeGroff DJ, Fleck KW (2008) Ibm power systems platform: advancements in the state of the art in its availability. *IBM Syst J* 47(4):519–533. doi:10.1147/SJ.2008.5386517
- Naghshineh M, Ratnaparkhi R, Dillenberger D, Doran JR, Dorai C, Anderson L, Pacifici G, Snowdon JL, Azagury A, VanderWiele M, Wolfsthal Y (2009) Ibm research division cloud computing initiative. *IBM J Res Dev* 53(4):1–10. doi:10.1147/JRD.2009.5429055
- Figueiredo R, Dinda P, Fortes J (2005) Guest editors' introduction: resource virtualization renaissance. *Computer* 38(5):28–31. doi:10.1109/MC.2005.159
- Lucky R (2009) Cloud computing. *IEEE Spectrum* 46(5):27. doi:10.1109/MSPEC.2009.4907382
- Narasimhan B, Nichols R (2011) State of cloud applications and platforms: the cloud adopters' view. *Computer* 44(3):24–28. doi:10.1109/MC.2011.66
- Anderson T, Peterson L, Shenker S, Turner J (2005) Overcoming the internet impasse through virtualization. *Computer* 38(4): 34–41. doi:10.1109/MC.2005.136
- Bavier A, Huang M, Peterson L (2005) An overlay data plane for planetlab. In: *Telecommunications, 2005. Advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop. aict/sapir/elete 2005. proceedings*, pp 8–14. doi:10.1109/AICT.2005.24
- Roscoe T (2005) The planetlab platform. In: Steinmetz R, Wehrle K (eds), *Peer-to-Peer systems and applications*, vol. 3485 of Lecture Notes in Computer Science, Springer, Heidelberg, pp 567–581
- Elliott C (2008) Geni—global environment for network innovations. In: *Local computer networks. LCN 2008. 33rd IEEE conference on*, p 8. doi:10.1109/LCN.2008.4664143
- group G (2006) Geni design principles. *Computer* 39(9):102–105. doi:10.1109/MC.2006.307
- Lemke M (2009) The european fire future internet research and experimentation initiative. In: *Testbeds and research infrastructures for the development of networks communities and*

- workshops, 2009. TridentCom 2009. 5th international conference on, pp 2–3. doi:[10.1109/TRIDENTCOM.2009.4976186](https://doi.org/10.1109/TRIDENTCOM.2009.4976186)
41. Fdida S, Friedman T, Parmentelat T (2010) OneLab: an open federated facility for experimentally driven future internet research, vol. 297 of studies in computational intelligence, pp 141–152
 42. Zahemszky A, Gajic B, Rothenberg CE, Reason C, Trossen D, Lagutin D, Tuononen J, Katsaros K (2011) Experimentally-driven research in publish/subscribe information-centric inter-networking
 43. Perez S, Cabero J, Miguel E (2009) Virtualization of the wireless medium: a simulation-based study. In: Vehicular technology conference. VTC Spring 2009. IEEE 69th, 2009, pp 1–5. doi:[10.1109/VETECS.2009.5073908](https://doi.org/10.1109/VETECS.2009.5073908)
 44. Mahindra R, Bhanage G, Hadjichristofi G, Seskar I, Raychaudhuri D, Zhang Y (2008) Space versus time separation for wireless virtualization on an indoor grid. In: Next generation internet networks. NGI, pp 215–222. doi:[10.1109/NGI.2008.36](https://doi.org/10.1109/NGI.2008.36)
 45. Singhal S, Hadjichristofi G, Seskar I, Raychaudhuri D (2008) Evaluation of uml based wireless network virtualization. In: Next generation internet networks. NGI 2008:223–230. doi:[10.1109/NGI.2008.37](https://doi.org/10.1109/NGI.2008.37)
 46. Mitola J (1993) Software radios: survey, critical evaluation and future directions. *IEEE Aerasp Electron Syst Mag* 8(4):25–36
 47. Chowdhury N, Boutaba R (2009) Network virtualization: state of the art and research challenges. *IEEE Commun Mag* 47(7):20–26. doi:[10.1109/MCOM.2009.5183468](https://doi.org/10.1109/MCOM.2009.5183468)
 48. Gavras A, Karila A, Fdida S, May M, Potts M (2007) Future internet research and experimentation: the fire initiative. *SIGCOMM Comput Commun Rev* 37(3):89–92. doi:[10.1145/1273445.1273460](https://doi.org/10.1145/1273445.1273460)
 49. Fdida S, Friedman T, MacKeith S (2010) OneLab: developing future internet testbeds, vol. 6481 of Lecture Notes in Computer Science, pp 199–200
 50. Nakao A, Ozaki R, Nishida Y (2008) Corelab: an emerging network testbed employing hosted virtual machine monitor. In: Proceedings of the 2008 ACM CoNEXT conference, CoNEXT '08, ACM, New York, pp 73:1–73:6. doi:[10.1145/1544012.1544085](https://doi.org/10.1145/1544012.1544085)
 51. Fischer V, Gerbaud L (2005) Corelab: a component-based integrated sizing environment. *COMPEL Int J Comput Maths Electr Electron Eng* 24(3):753–766. doi:[10.1108/03321640510598111](https://doi.org/10.1108/03321640510598111)
 52. Schwerdel D, Günther D, Henjes R, Reuther B, Müller P (2010) German-lab experimental facility. In: Proceedings of the third future internet conference on future internet, FIS'10, Springer, Berlin, pp 1–10
 53. Schwerdel D, Günther D, Henjes R, Reuther B, Müller P (2010) German-lab experimental facility, vol. 6369 of Lecture Notes in Computer Science
 54. Farias F, Salvatti J, Cerqueira E, Abelem A (2012) A proposal management of the legacy network environment using openflow control plane. In: Network operations and management symposium (NOMS). IEEE pp 1143–1150. doi:[10.1109/NOMS.2012.6212041](https://doi.org/10.1109/NOMS.2012.6212041)
 55. Correia LM, Abramowicz H, Johnsson M, Wnstel K (2011) Architecture and design for the future internet: 4WARD project, 1st edn. Springer Publishing Company, Incorporated
 56. Niebert N, Baucke S, El-Khayat I, Johnsson M, Ohlman B, Abramowicz H, Wuenstel K, Woesner H, Quittek J, Correia L (2008) The way 4ward to the creation of a future internet. In: Personal, indoor and mobile radio communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, pp 1–5. doi:[10.1109/PIMRC.2008.4699967](https://doi.org/10.1109/PIMRC.2008.4699967)
 57. Antonescu A-F, Robinson P, Contreras-Murillo LM, Aznar J, Soudan S, Anhalt F, Garcia-Espin JA (2012) Towards cross stratum sla management with the geysers architecture. In: Proceedings of the (2012) IEEE 10th international symposium on parallel and distributed processing with applications, ISPA '12. IEEE Computer Society, Washington, DC, USA, pp 527–533. doi:[10.1109/ISPA.2012.78](https://doi.org/10.1109/ISPA.2012.78)
 58. Rochwerger B, Breitgand D, Levy E, Galis A, Nagin K, Llorente IM, Montero R, Wolfsthal Y, Elmroth E, Caceres J, Ben-Yehuda M, Emmerich W, Galan F (2009) The reservoir model and architecture for open federated cloud computing. *IBM J Res Dev* 53(4):4:1–4:11. doi:[10.1147/JRD.2009.5429058](https://doi.org/10.1147/JRD.2009.5429058)
 59. Comuzzi M, Kotsokalis C, Spanoudakis G, Yahyapour R (2009) Establishing and monitoring slas in complex service based systems. In: Web services, 2009. ICWS 2009. IEEE International Conference on, pp 783–790. doi:[10.1109/ICWS.2009.47](https://doi.org/10.1109/ICWS.2009.47)
 60. Weissman JB, Sundararajan P, Gupta A, Ryden M, Nair R, Chandra A (2011) Early experience with the distributed nebula cloud. In: Proceedings of the fourth international workshop on data-intensive distributed computing, DIDC '11, ACM, New York, NY, USA, pp 17–26. doi:[10.1145/1996014.1996019](https://doi.org/10.1145/1996014.1996019)
 61. Bouabene G, Jelger C, Tschudin C, Schmid S, Keller A, May M (2010) The autonomic network architecture (ana). *IEEE J Sel Areas Commun* 28(1):4–14. doi:[10.1109/JSAC.2010.100102](https://doi.org/10.1109/JSAC.2010.100102)
 62. ONE, Open networking foundation (2012). www.opennetworking.org
 63. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008) Openflow: enabling innovation in campus networks. *SIGCOMM Comput Commun Rev* 38(2):69–74. doi:[10.1145/1355734.1355746](https://doi.org/10.1145/1355734.1355746)
 64. Gude N, Koponen T, Pettit J, Pfaff B, Casado M, McKeown N, Shenker S (2008) Nox: towards an operating system for networks. *SIGCOMM Comput Commun Rev* 38:105–110. doi:[10.1145/1384609.1384625](https://doi.org/10.1145/1384609.1384625)
 65. Tootoonchian A, Ganjali Y (2010) Hyperflow: a distributed control plane for openflow. In: Proceedings of the 2010 internet network management conference on research on enterprise networking, INM/WREN'10. USENIX Association, Berkeley, CA, USA, p 3
 66. Curtis AR, Mogul JC, Tourrilhes J, Yalagandula P, Sharma P, Banerjee S (2011) Devoflow: scaling flow management for high-performance networks. *SIGCOMM Comput Commun Rev* 41(4):254–265. doi:[10.1145/2043164.2018466](https://doi.org/10.1145/2043164.2018466)
 67. Koponen T, Casado M, Gude N, Stribling J, Poutievski L, Zhu M, Ramanathan R, Iwata Y, Inoue H, Hama T, Shenker S (2010) Onix: a distributed control platform for large-scale production networks. In: Proceedings of the 9th USENIX conference on operating systems design and implementation, OSDI'10, USENIX Association, Berkeley, CA, USA, pp 1–6
 68. Sherwood R, Chan M, Covington A, Gibb G, Flajslik M, Handigol N, Huang T-Y, Kazemian P, Kobayashi M, Naous J, Seetharaman S, Underhill D, Yabe T, Yap K-K, Yiakoumis Y, Zeng H, Appenzeller G, Johari R, McKeown N, Parulkar G (2010) Carving research slices out of your production networks with openflow. *SIGCOMM Comput Commun Rev* 40(1):129–130. doi:[10.1145/1672308.1672333](https://doi.org/10.1145/1672308.1672333)
 69. Nascimento MR, Rothenberg CE, Salvador MR, Corrêa CNA, de Lucena SC, Magalhães MF (2011) Virtual routers as a service: the routeflow approach leveraging software-defined networks. In: Proceedings of the 6th international conference on future internet technologies, CFI '11, ACM, New York, pp 34–37. doi:[10.1145/2002396.2002405](https://doi.org/10.1145/2002396.2002405)
 70. Bansal M, Mehlman J, Katti S, Levis P (2012) Openradio: a programmable wireless dataplane. In: Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, ACM, New York, pp 109–114. doi:[10.1145/2342441.2342464](https://doi.org/10.1145/2342441.2342464)

71. Foster N, Harrison R, Freedman MJ, Monsanto C, Rexford J, Story A, Walker D (2011) Frenetic: a network programming language. *SIGPLAN Not* 46:279–291. doi:[10.1145/2034574.2034812](https://doi.org/10.1145/2034574.2034812)
72. Rexford J (2012) Programming languages for programmable networks. *SIGPLAN Not* 47(1):215–216. doi:[10.1145/2103621.2103683](https://doi.org/10.1145/2103621.2103683)
73. Monsanto C, Foster N, Harrison R, Walker D (2012) A compiler and run-time system for network programming languages. *SIGPLAN Not* 47(1):217–230. doi:[10.1145/2103621.2103685](https://doi.org/10.1145/2103621.2103685)
74. Handigol N, Heller B, Jeyakumar V, Mazières D, McKeown N (2012) Where is the debugger for my software-defined network? In: *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*, ACM, New York, NY, USA, pp 55–60. doi:[10.1145/2342441.2342453](https://doi.org/10.1145/2342441.2342453)
75. Lin P, Bi J, Hu H (2012) Asic: an architecture for scalable intradomain control in openflow. In: *Proceedings of the 7th international conference on future internet technologies, CFI '12*, ACM, New York, NY, USA, pp 21–26. doi:[10.1145/2377310.2377317](https://doi.org/10.1145/2377310.2377317)
76. Levin D, Wundsam A, Heller B, Handigol N, Feldmann A (2012) Logically centralized? State distribution trade-offs in software defined networks. In: *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*, ACM, New York, NY, USA, pp 1–6. doi:[10.1145/2342441.2342443](https://doi.org/10.1145/2342441.2342443)
77. Heller B, Sherwood R, McKeown N (2012) The controller placement problem. In: *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*, ACM, New York, NY, USA, pp 7–12. doi:[10.1145/2342441.2342444](https://doi.org/10.1145/2342441.2342444)
78. Strassner J (2007) *The role of autonomic networking in cognitive networks*. Wiley, London, pp 23–52
79. Kephart J, Chess D (2003) The vision of autonomic computing. *Computer* 36(1):41–50. doi:[10.1109/MC.2003.1160055](https://doi.org/10.1109/MC.2003.1160055)
80. Dobson S, Denazis S, Fernández A, Gañi D, Gelenbe E, Massacci F, Nixon P, Saffre F, Schmidt N, Zambonelli F (2006) A survey of autonomic communications. *ACM Trans Auton Adapt Syst* 1:223–259. doi:[10.1145/1186778.1186782](https://doi.org/10.1145/1186778.1186782)
81. Clark DD, Partridge C, Ramming JC, Wroclawski JT (2003) A knowledge plane for the internet. In: *Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03*. ACM, New York, NY, USA, pp 3–10. doi:[10.1145/863955.863957](https://doi.org/10.1145/863955.863957)
82. Dobson S, Sterritt R, Nixon P, Hinchey M (2010) Fulfilling the vision of autonomic computing. *Computer* 43:35–41
83. Smirnov M (2004) *Autonomic communication: research agenda for a new communications paradigm*. Tech. rep, Fraunhofer FOKUS
84. Sterritt R, Bustard D (2003) Autonomic computing—a means of achieving dependability? In: *Engineering of computer-based systems, 2003*. In: *Proceedings 10th IEEE international conference and workshop on the*, pp 247–251. doi:[10.1109/ECBS.2003.1194805](https://doi.org/10.1109/ECBS.2003.1194805)
85. Strassner J, Meer S, O'Sullivan D, Dobson S (2009) The use of context-aware policies and ontologies to facilitate business-aware network management. *J Netw Syst Manag* 17:255–284. doi:[10.1007/s10922-009-9126-4](https://doi.org/10.1007/s10922-009-9126-4)
86. Zseby T, Hirsch T, Kleis M, Popescu-Zeletin R (2009) *Towards the future internet*, IOS Press, Ch. *Towards a future internet node collaboration for autonomic communication*
87. Jennings B, van der Meer S, Balasubramaniam S, Botvich D, Foghlu MO, Donnelly W, Strassner J (2007) *Towards autonomic management of communications networks*. *IEEE Commun Mag* 45(10):112–121. doi:[10.1109/MCOM.2007.4342833](https://doi.org/10.1109/MCOM.2007.4342833)
88. Baresi L, Ferdinando AD, Manzalini A, Zambonelli F (2009) The cascadas framework for autonomic communications. In: *Vasilakos AV, Parashar M, Karnouskos S, Pedrycz W (eds) Springer US, Autonomic Communication*, pp 147–168
89. Jelger C, Tschudin C, Schmid S, Leduc G (2007) Basic abstractions for an autonomic network architecture. In: *World of wireless, mobile and multimedia networks, 2007. WoWMoM 2007. IEEE international symposium on a*, pp 1–6. doi:[10.1109/WOWMOM.2007.4351692](https://doi.org/10.1109/WOWMOM.2007.4351692)
90. Horizon project (2012). <http://www.gta.ufrj.br/horizon>
91. Sterritt R, Hinchey M (2005) Autonomicity—an antidote for complexity?. In: *Computational systems bioinformatics conference, 2005. Workshops and poster abstracts. IEEE*, pp 283–291. doi:[10.1109/CSBW.2005.28](https://doi.org/10.1109/CSBW.2005.28)
92. Papazoglou M, Traverso P, Dustdar S, Leymann F (2007) *Service-oriented computing: state of the art and research challenges*. *Computer* 40(11):38–45. doi:[10.1109/MC.2007.400](https://doi.org/10.1109/MC.2007.400)
93. Galis A, Abramowicz H, Brunner M, Raz D, Chemouil P, Butler J, Polychronopoulos C, Clayman S, de Meer H, Coupaye T, Pras A, Sabnani K, Massonet P, Naqvi S (2009) *Management and service-aware networking architectures for future internet x2014; position paper: system functions, capabilities and requirements*. In: *Communications and networking in China, 2009. ChinaCOM 2009. Fourth International Conference on*, pp 1–13
94. Schroth C, Janner T (2007) Web 2.0 and soa: converging concepts enabling the internet of services. *IT Professional* 9(3):36–41. doi:[10.1109/MITP.2007.60](https://doi.org/10.1109/MITP.2007.60)
95. Buxmann P, Hess T, Ruggaber R, *Internet of services*. *Bus Inf Syst Eng* 1(5):341–342
96. Cardoso J, Winkler M, Voigt K (2011) Berthold, *IoS-based services, platform services, SLA and models for the internet of services*, vol. 50 of *communications in computer and information science*, pp 3–17
97. Villasante J (2009) *Internet of services, the 1st European conference on software services and service oriented knowledge utilities technologies*
98. Banerjee P, Friedrich R, Bash C, Goldsack P, Huberman B, Manley J, Patel C, Ranganathan P, Veitch A (2011) *Everything as a service: powering the new information economy*. *Computer* 44(3):36–43. doi:[10.1109/MC.2011.67](https://doi.org/10.1109/MC.2011.67)
99. Turner M, Budgen D, Brereton P (2003) Turning software into a service. *Computer* 36(10):38–44. doi:[10.1109/MC.2003.1236470](https://doi.org/10.1109/MC.2003.1236470)
100. Prodan R, Ostermann S (2009) *A survey and taxonomy of infrastructure as a service and web hosting cloud providers*. In: *Grid computing, 2009 10th IEEE/ACM international conference on*, pp 17–25. doi:[10.1109/GRID.2009.5353074](https://doi.org/10.1109/GRID.2009.5353074)
101. NESSI, Nessi: networked European software and services. <http://www.nessi-europe.eu/>
102. Stricker V, Heuer A, Zaha JM, Pohl K, de Panfilis S (2009) *Towards the future internet*, IOS Press, Ch. *Agreeing Upon SOA Terminology—Lessons Learned*
103. Krummenacher R, Norton B, Simperl E, Pedrinaci C (2009) *Soa4all: enabling web-scale service economies*. In: *Semantic computing, 2009. ICSC '09. IEEE International Conference on* pp 535–542. doi:[10.1109/ICSC.2009.46](https://doi.org/10.1109/ICSC.2009.46)
104. Shadbolt N, Hall W, Berners-Lee T (2006) *The semantic web revisited*. *IEEE Intell Syst* 21(3):96–101. doi:[10.1109/MIS.2006.62](https://doi.org/10.1109/MIS.2006.62)
105. Fensel PD (2007) *Serviceweb 3.0*. In: *Intelligent agent technology. IAT '07. IEEE/WIC/ACM International Conference on*, p xxii. doi:[10.1109/IAT.2007.103](https://doi.org/10.1109/IAT.2007.103)
106. Kennedy J, Vidal M, Masuch C (2007) *Digital business ecosystems (db)*. In: *Digital EcoSystems and technologies conference, 2007 DEST '07. Inaugural IEEE-IES*, p 1. doi:[10.1109/DEST.2007.371942](https://doi.org/10.1109/DEST.2007.371942)
107. Leiner BM, Cerf VG, Clark DD, Kahn RE, Kleinrock L, Lynch DC, Postel J, Roberts LG, Wolff S (2009) *A brief history of the internet*. *SIGCOMM Comput Commun Rev* 39: 22–31. doi:[10.1145/1629607.1629613](https://doi.org/10.1145/1629607.1629613)

108. Esteve C, Verdi FL, Magalhães MF (2008) Towards a new generation of information-oriented internetworking architectures. In: Proceedings of the 2008 ACM CoNEXT conference, CoNEXT '08, ACM, New York, NY, USA, pp 65:1–65:6. doi:[10.1145/1544012.1544077](https://doi.org/10.1145/1544012.1544077)
109. Jacobson V (2011) Ccn routing and forwarding. Tech. rep, Stanford NetSeminar
110. Trossen D, Sarela M, Sollins K (2010) Arguments for an information-centric internetworking architecture. SIGCOMM Comput Commun Rev 40:26–33. doi:[10.1145/1764873.1764878](https://doi.org/10.1145/1764873.1764878)
111. Dannewitz C (2009) Augmented internet: An information-centric approach for real-world/internet integration. In: Communications workshops, 2009. ICC workshops 2009. IEEE international conference on, pp 1–6. doi:[10.1109/ICCW.2009.5207982](https://doi.org/10.1109/ICCW.2009.5207982)
112. Dannewitz C (2009) Netinf: an information-centric design for the future internet. In: Proceedings 3rd GI ITG KuVS workshop on the future internet
113. Dimitrov V, Koptchev V (2010) Psirp project—publish-subscribe internet routing paradigm: new ideas for future internet. In: Proceedings of the 11th international conference on computer systems and technologies and workshop for PhD students in computing on international conference on computer systems and technologies, CompSysTech '10, ACM, New York, NY, USA, pp 167–171. doi:[10.1145/1839379.1839409](https://doi.org/10.1145/1839379.1839409)
114. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th international conference on emerging networking experiments and technologies, CoNEXT '09, ACM, New York, NY, USA, pp 1–12. doi:[10.1145/1658939.1658941](https://doi.org/10.1145/1658939.1658941)
115. Fensel D (2009) The publish/subscribe internet routing paradigm (PSIRP): designing the future internet architecture, IOS Press
116. Daras P, Williams D, Guerrero C, Kegel I, Laso I, Bouwen J, Meunier J-D, Niebert N, Zahariadis T (2009) Why do we need a content-centric future internet. Proposals towards content-centric internet architectures, Tech. rep., Future Content Networks Group—Future Internet, Assembly
117. Alberti AM, Vaz A, Brandão R, Martins B (2012) Internet of information and services (iois): a conceptual integrative architecture for the future internet. In: Proceedings of the 7th international conference on future internet technologies, CFI '12, ACM, New York, pp 45–45. doi:[10.1145/2377310.2377325](https://doi.org/10.1145/2377310.2377325)
118. Siekkinen M, Goebel V, Plagemann T, Skevik K-A, Banfield M, Brusci I (2007) Beyond the future internet-requirements of autonomous networking architectures to address long term future networking challenges. In: Future trends of distributed computing systems, 2007. FTDCS '07. 11th IEEE international workshop on, pp 89–98. doi:[10.1109/FTDCS.2007.14](https://doi.org/10.1109/FTDCS.2007.14)
119. Stoica I, Adkins D, Zhuang S, Shenker S, Surana S (2004) Internet indirection infrastructure. IEEE/ACM Trans Netw 12(2): 205–218. doi:[10.1109/TNET.2004.826279](https://doi.org/10.1109/TNET.2004.826279)
120. Nikander P, Gurtov A, Henderson T (2010) Host identity protocol (hip): connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks. IEEE Commun Surv Tutor 12(2):186–204. doi:[10.1109/SURV.2010.021110.00070](https://doi.org/10.1109/SURV.2010.021110.00070)
121. Harai H (2009) Designing new-generation network—overview of akari architecture design. In: Communications and photonics conference and exhibition (ACP). Asia pp 1–2
122. Martins BM, Alberti AM (2011) Host identification and location decoupling: a comparison of approaches. In: Proceedings of the international workshop on telecommunications
123. Perkins C (Aug. 2002) IP mobility support for IPv4, RFC 3344 (Proposed Standard), obsoleted by RFC 5944, updated by RFC 4721. <http://www.ietf.org/rfc/rfc3344.txt>
124. Moskowitz R, Nikander P (May 2006) Host identity protocol (HIP) architecture, RFC 4423 (Informational). <http://www.ietf.org/rfc/rfc4423.txt>
125. Farinacci D, Fuller V, Meyer D, Lewis D (Apr. 2010) Locator/ID separation protocol (LISP), work in progress (draft-ietf-lisp-07). <http://tools.ietf.org/html/draft-ietf-lisp-07>
126. Wang Y, Bi J, Jiang X (2012) Mobility support in the internet using identifiers. In: Proceedings of the 7th international conference on future internet technologies, CFI '12, ACM, New York, pp 37–42. doi:[10.1145/2377310.2377322](https://doi.org/10.1145/2377310.2377322)
127. Bhanage G, Chanda A, Li J, Raychaudhuri D (2011) Storage-aware routing protocol for the mobilityfirst network architecture, wireless conference 2011—sustainable wireless technologies (European Wireless), 11th European, pp 1–8
128. Seskar I, Nagaraja K, Nelson S, Raychaudhuri D (2011) Mobilityfirst future internet architecture project. In: Proceedings of the 7th asian internet engineering conference, AINTEC '11, ACM, New York, pp 1–3. doi:[10.1145/2089016.2089017](https://doi.org/10.1145/2089016.2089017)
129. Vu T, Baid A, Zhang Y, Nguyen TD, Fukuyama J, Martin RP, Raychaudhuri D (2012) Dmap: a shared hosting scheme for dynamic identifier to locator mappings in the global internet. In: Proceedings of the 2012 IEEE 32nd international conference on distributed computing systems, ICDCS '12. IEEE computer society, Washington, DC, USA, pp 698–707. doi:[10.1109/ICDCS.2012.50](https://doi.org/10.1109/ICDCS.2012.50)
130. Jung H, Koh S (2012) A new internetworking architecture for mobile oriented internet environment. In: Cunningham P, Cunningham M (eds), Future network and MobileSummit 2012 conference proceedings
131. Louin P, Bertin P (2011) Network and host based distributed mobility. In: Wireless personal multimedia communications (WPMC), 2011 14th international symposium on, pp 1–5
132. Alberti A (2010) Future network architectures: technological challenges and trends. In: Tronco T (ed), New network architectures, vol. 297 of studies in computational intelligence, Springer, Heidelberg, pp 79–120
133. Chun W, Lee T-H, Choi T (2011) Yanail: yet another definition on names, addresses, identifiers, and locators. In: Proceedings of the 6th international conference on future internet technologies, CFI '11, ACM, New York, NY, USA, pp 8–12. doi:[10.1145/2002396.2002399](https://doi.org/10.1145/2002396.2002399)
134. Privacy and identity management for community services (2012). <http://www.picos-project.eu>
135. Vivas JL, Agudo I, Lopez J (2011) A methodology for security assurance-driven system development. Requir Eng 16(1):55–73. doi:[10.1007/s00766-010-0114-8](https://doi.org/10.1007/s00766-010-0114-8)
136. Primelife. <http://primelife.ercim.eu>
137. Trabelsi S, Sendor J, Reinicke S (2011) Ppl: primelife privacy policy engine. In: Policies for distributed systems and networks (POLICY). IEEE international symposium on, pp 184–185. doi:[10.1109/POLICY.2011.24](https://doi.org/10.1109/POLICY.2011.24)
138. Think-trust (2012) <http://www.think-trust.eu>
139. Master: Managing assurance, security and trust for services (2012) <http://www.project-master.eu>
140. Trusted architecture for securely shared services. <http://www.tas3.eu>
141. Bertolino A, De Angelis G, Polini A (2009) On-line validation of service oriented systems in the european project tas3. In: Principles of engineering service oriented systems, 2009 PESOS 2009. ICSE workshop on, pp 107–110. doi:[10.1109/PESOS.2009.5068830](https://doi.org/10.1109/PESOS.2009.5068830)
142. Automated validation of trust and security of service-oriented architecture. <http://www.avantssar.eu>
143. Viganò L (2012) Automated validation of trust and security of service-oriented architectures with the avantssar platform. In: High performance computing and simulation (HPCS). International conference on, pp 444–447. doi:[10.1109/HPCSim.2012.6266956](https://doi.org/10.1109/HPCSim.2012.6266956)

144. Trusted embedded computing (tecom). <http://www.tecom-project.eu>
145. Rowe D, Leaney J (1997) Evaluating evolvability of computer based systems architecturesan ontological approach. In: Engineering of computer-based systems. Proceedings international conference and workshop on, pp 360–367. doi:[10.1109/ECBS.1997.581903](https://doi.org/10.1109/ECBS.1997.581903)