Future Internet Design Principles

January 2012

Created by the “Future Internet Architecture (FIArch) Group”

Coordinated by eight FP7 Coordination and Support Action (CSA) projects supported by the DG Information Society and Media of the European Commission

European Commission
Information Society and Media

DISCLAIMER

The EC Future Internet Architecture (FIArch) Experts Reference Group (ERG) focuses on a few key architectural issues, contributing to an EC research roadmap towards a Future Internet Architecture. It is composed of representatives of the most relevant FP7 research projects in relation to Future Internet architectures and renowned experts from industry and academy covering in a complementarily way all areas related to the Future Internet Architecture.

The work of the group is coordinated by the FP7 CSA projects in the area of Future Internet:

- NextMedia
- IOT-I
- SOFI
- EFFECTS+
- EIFFEL
- Chorus+
- SESERV
- Paradiso 2

and supported by the EC Units

D1: Future Networks
D2: Networked Media Systems
D3: Software & Service Architectures & Infrastructures
D4: Networked Enterprise & Radio Frequency Identification (RFID)
F5: Trust and Security.

It was coordinated by Isidro Laso Ballesteros (European Commission). A list of contributors may be found in the Annex.

Disclaimer: This document does not necessarily reflect the view of the European Commission. It addresses the majority of the FIArch Group experts' views but do not necessarily coincide with each member's personal opinion.

Table of Contents

Future Internet Design Principles[1]

EC FIArch Group[2]

Release Date: *January 2012*

1 INTRODUCTION

The Internet is the most important information exchange means nowadays. It has become the core communication environment, not only for business relations, but also for social and human interaction. Yet, the immense success of Internet has created even higher hopes and expectations for new immersive and real-time applications and services, without guarantees that the Internet as we know it today will be able to support them.

The new demands can be addressed to a certain degree through incremental infrastructure investment (i.e. more and more bandwidth in wireline, wireless and mobile networks) combined with "over-dimensioning". However, analyses have shown [Jacobson09] [Zahariadis11] that increasing the bandwidth to peta-bps on the backbone network will not suffice due to new qualitative requirements in, for example, highly critical services such as e-health applications, clouds of services and clouds of sensors, new social network applications like collaborative 3D immersive environments, new commercial and transactional applications, new location-based services and so on.

The EC Future Internet Architecture (FIArch) group has already identified some of the fundamental limitations of current Internet architecture and some of the Design Objectives of the Future Internet [FIArch]. This is the next step, which contributes towards the specification of the Design Principles that will govern the Internet architecture..

1.1 Motivation

Design principles play a central role in the architecture of the Internet as driving most engineering decisions at conception level but also operational level of communication systems. Often cited as the corner stone of the Internet design compared to architectures that rely exclusively on modelling, they are not formally defined (using a closed mathematical formulation). Classical telecommunication systems (i.e. legacy voice communication) do not

---

[1] The views expressed are those of the authors and not necessarily those of the European Commission or any of its officials.

[2] The document addresses the majority of the FIArch Group experts' views but do not necessarily coincide with each member's personal opinion. Any feedback to the FIArchitecture Group is welcome at the email addresses: fiarch@future-internet.eu or infso-future-internet@ec.europa.eu

consider design principles and derive their model directly from requirements[3]. However, when it comes to the design of the Internet, the formulation of design principles is a fundamental characteristic of the Internet design process that guides the specification of the design model.

As already stated by B.Carpenter [RFC1958], in searching for Internet architectural principles, we must remember that technical change is continuous in the information and communication technology industry. Hence as stated in RFC 1958, "*in this environment, some architectural principles inevitably change. Principles that seemed inviolable a few years ago are deprecated today. Principles that seem sacred today will be deprecated tomorrow. The principle of constant change is perhaps the only principle of the Internet that should survive indefinitely*".

In this context, it is important to provide a detailed analysis of the application of known design principles and their potential evolution. Via long discussions, email communication, brainstorming as well as face-to-face meetings and workshops, the FIArch group has tried to analyse the design principles of today's Internet and foresee the design principles that will govern Future Internet. Detailed analysis has been performed to minimize as much as possible the subjective component. However, as the environment is changing rapidly, different experts of the group have different views and opinions on both the short term (evolutionary) and the longer term (either still evolutionary or clean-slate) design principles underlying the Internet Architecture , which we try to capture and document.

1.2 Scope and Purpose

The purpose of this current document is to identify and to reach some understanding of the different design principles that we expect to govern the architecture of the Future Internet. This may serve as a starting point and comparison basis for all research and development projects that target Future Internet Architecture.

This report targets not only the Research and Academic community, but also the European ICT industry and decision makers (including but not limited to telecom operators, ISPs, networking equipment manufacturers and providers of value-added networking services).

The rest of the document is structured as follows: Section 2 contains the necessary definitions used in this group so as to avoid misunderstandings due to the different background of the group's members. Section 3 gives a background and our understanding of current Internet design principles, while Section 4 summarizes the Design Principles that we expect to remain or evolve towards the Future Internet.

[3] It suffices to analyze the process followed in PSTN, GSM, and other voice application specific-networks or the OSI Reference Model (OSI RM) whose design principles were loosely defined and in this practice resulted in lot of protocol misconceptions -

2 DEFINITIONS

Before describing the approach that the FIArch Group has followed, it is important to explain some definitions that we have used in our work.

For convenience we replicate some of the definitions provided in [FIArch]. We use the term:

- "***data"*** to refer to any organized group of bits a.k.a. data packets, data traffic, information, content (audio, video, multimedia), etc.
- "***service***" to refer to any action or set of actions performed by a provider (person or system) in fulfilment of a request, which occurs through the Internet (i.e. by exploiting data communication, as defined below) with the ultimate aim of creating and/or providing *added value* or benefits to the requester(s). Note that this document refrains from taking position on the localization and distribution of these APIs.

We define as ***"architecture"*** a set of functions, states, and objects/information together with their behavior, structure, composition, relationships and spatio-temporal distribution[4,5]. The specification of the associated functional, object/informational and state models leads to an architectural model comprising a set of components (i.e. procedures, data structures, state machines) and the characterization of their interactions (i.e. messages, calls, events, etc.).

Please note that the canonical definition of architecture includes the principles and guidelines governing their design and evolution over time.

"Design principles" refers to agreed *structural & behavioural rules* on how a designer/an architect can best structure the various architectural components and describe the fundamental and *time invariant* laws underlying the working of an *engineered artefact*.

- By ***"structural & behavioural rules"*** *we* refer to the set of commonly accepted and agreed rules serving to guide, control, or regulate a proper and acceptable structure of a system at design time and a proper and acceptable behaviour of a system at running time.
- ***Time invariance*** refers to a system whose output does not depend explicitly on time (this time invariance is to be seen as within a given set of initial conditions due to the technological change and paradigms shifts, the economical constraints, etc.). Robustness and longevity over time is a consequence of this time invariance.
- ***Engineered artefact*** is an object formed/produced by engineering.

One of the critical points that we have faced many times during our analysis has been the term "***complexity***". There are multiple definitions of the complexity.[6]

---

[4] Many definitions of (system) architecture have been formulated over time. We borrow the terms of our definition from Dewayne E. Perry and Alexander L. Wolf. "Foundations for the Study of Software Architecture". ACM SIGSOFT Software Engineering Notes, 17:4, October 1992, Garlan and Perry, guest editorial to the IEEE Transactions on Software Engineering, April 1995, and Booch, Presentation at Software Developers Conference 1999

[5] The time dimension is often omitted we include it here to keep a generic nature of our definition.

[6] There are multiple definition of complexity:

- *Computational complexity* is a measure of the computational resources needed to solve computational problems. Computational resources are measured either in terms of time (i.e., number of elementary computational steps per second with respect to the input size) or in terms of space (i.e., memory storage size usually measured in bits or bytes) or some combination of the two.

Within our analysis we have mainly focus on the *architectural* and *communication* complexity. Moreover, we define the following terms being used throughout this document:

- ***"Communication"*** the exchange of "data" (including both control messages and "data") between a source and a sink.
- ***"Communication end-point"*** the physical or logical source and sink of information. The communication end-point can be considered to be an application, a service, a process, a protocol, a node, a network.
- "***End-to-end communication"*** a communication that takes place between communication end-points of the same physical or logical functional level.
- ***"Module"*** is a unity that represents functions. It could be considered as a physical or logical unity.
- **"Security"** is a process of taking into account all major constraints. Security includes: Robustness, Confidentiality and Integrity.
- ***"Robustness"*** degree to which a system operates correctly in the presence of exceptional inputs or stressful environmental conditions. [IEEE-610]
- ***"Confidentiality"*** is the property of "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of information security [ISO-27002].
- ***"Integrity***" In literature there are multiple definitions of the integrity. Here we consider mainly the "***data integrity***"[7] and "***system integrity***"[8].

3 BACKGROUND - OUR UNDERSTANDING OF EXISTING DESIGN PRINCIPLES

Before we start our analysis of the Design Principles of the Internet, it is mandatory to establish a common ground and understanding of the cornerstone design principles that governs the Internet today.

3.1 Heterogeneity support principle

Since the early days of Internet, heterogeneity is one of its major characteristics. The Internet is characterized by heterogeneity on many levels: devices and nodes, router scheduling algorithms and queue management mechanisms, routing protocols, levels of multiplexing, protocol versions and implementations, underlying link layers (e.g., point-to-point, multi-

---

- *Kolmogorov complexity* (a.k.a. program-size complexity) is a measure of complexity that quantifies how complex a system is in terms of the length of the shortest computer program, or set of algorithms, need to completely describe the system. In other terms, this measure of complexity qualifies how small a model of a given system is necessary and sufficient to capture the essential patterns of that system.
- *Architectural complexity* is a measure of the complexity of an architecture proportionally to its number of components and interactions among components.
- *Communication complexity* is the rate *x* size of messages (input/output) exchanged between nodes and that are needed for function performing on these nodes to properly operate.

[7] Refers to the trustworthiness of system resources over their entire life cycle.

[8] That condition of a system wherein its mandated operational and technical parameters are within the prescribed limits

access links, wireless, FDDI, etc.)[9], in the traffic mix and in the levels of congestion at different times and places. Moreover, as the Internet is composed of autonomous organizations and internet service providers, each with their own separate policy concerns, there is a large heterogeneity of administrative domains and pricing structures.

As a result, heterogeneity principle is proposed in [RFC1958] to be supported by design.

3.2 Scalability & the Amplification Principle

Scalability refers to the ability of a computational system (hardware or software) to continue to function (without making changes to the system) under satisfactory and well specified bounds, i.e., without affecting its performance, when its input is changed in size or volume or in their respective rate of variation. Examples include increasing number of nodes/AS, increasing number of links, increasing number of hosts/prefixes.

From its introduction in [RFC1287], the need to ensure scalability is of increasing importance. Indeed, scalability is considered among the major general principles of Internet [RFC1958] which states "***All designs must scale readily to very many nodes per site and to many millions of sites***". This principle refers thus to the scale invariant that the global design should meet.

Moreover, [RFC3439] also defines the "*Amplification Principle*". The amplification principle states that "there do exist non-linearities which do not occur at small to medium scale, but occur at large scale". In other words, following the butterfly effect [Hilborn04], "in many large interconnected networks, even small things can and do cause huge events; even small perturbations on the input to a process can destabilize the system's output". As a result "***complexity can amplify small perturbations, and the design engineer must ensure such perturbations are extremely rare***." [RFC3439].

3.3 Robustness and Adaptability principles

The Internet *Robustness Principle* [RFC760] [RFC791] also known as the Postel Law was based on the condition that each protocol implementation must interoperate with other created by different individuals. As there may be different interpretations of the same protocol, each one should "*be liberal in what you accept, and conservative in what you send.*" The fundamental objective of this principle was to maximize interoperability between network protocol implementations, particularly in the face of ambiguous or incomplete specifications.

The robustness principle appears also in RFCs 793 [RFC793] and 1122 [RFC1122]. Focusing at the software part of the Internet, "*Software should be written to deal with every conceivable error, no matter how unlikely. In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect.*" This assumption leads to suitable protective design, although the most serious problems in the Internet have been caused by un-envisaged mechanisms triggered by low-probability events; mere human malice would never have taken so devious a course!

Moreover, RFC1122 also defines *adaptability* as a major design principle: "*Adaptability to change must be designed into all levels of Internet host software. As a simple example, consider a protocol specification that contains an enumeration of values for a particular header field -- e.g., a type field, a port number, or an error code; this enumeration must be*

[9] It is important to remember that IP has been designed as the common denominator among all data link layers.

assumed to be incomplete. Thus, if a protocol specification defines four possible error codes, the software must not break when a fifth code shows up." The second part of the principle is almost as important: software on other hosts may contain deficiencies that make it unwise to exploit legal but obscure protocol features. A corollary of this is "*watch out for misbehaving hosts*"; host software should be prepared, not just to survive other misbehaving hosts, but also to cooperate to limit the amount of disruption such hosts can cause to the shared communication facility.

It is important to emphasize that the RFC 1122 version of the Robustness Principle clarifies a common misconception about the Robustness Principle, in that its initial version might imply that receiving systems would become vulnerable to attack. Indeed, the principle as interpreted in RFC 1122 advises: "*In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect. This assumption will lead to suitable protective design..."* but emphasizes that as result protocols design would improve their robustness. Notice also that this version says that it applies at "every layer of the protocols"--including the application layer.

3.4 The "modularization" principle

Communication systems have been designed by applying the modularization principle [Dijkstra68] by decomposing the communication functionality into different modules with well-defined interfaces. Each of these modules corresponds to a functional assignment. The layering principle in network protocol design offers various conceptual and structuring advantages, such as reduction of complexity, isolation of functionality, and reusability of protocol modules. Each layer offers services to the next higher layer using the services offered by the layer below. The current Internet follows a 5-layer model, in contrast to the 7-layer model in ISO/OSI Reference Model (RM). The Internet, in narrow sense, often refers only to IP and TCP/UDP, the network layer and the transport layer respectively. The IP layer enables end-to-end communication by addressing hosts' interfaces (even if IP address name space is often used as "host ID"), computing routes, and delivering IP packets. The TCP/UDP layer offers reliable data transfer using flow and congestion control at the communication endpoints. Separation of IP layer from various underlying networking technologies allows IP to run on various networks. The transport layer provides byte stream via TCP or message service via UDP to diverse applications. Both the network and transport layers form the narrow waist of the hourglass model, and promote rapid adaptation of network techniques and applications.

However, in the data networking, the invariant and static binding between different modules or layers (fixed stack) implies that the functions of each layer are carried out completely before the protocol data unit is passed to the next layer. Following the modularization principle, this means that performance optimization ***of each layer has to be performed separately. Such ordering constraints may conflict with efficient implementation of data manipulation functions.***

In addition, some in-line boxes, such as firewall or NAT (network address translator), violate the layering principles in that they inspect or change the contents of packets that are outside the scope of their layers functionality. Also new changes that enhance the capabilities of the Internet, such as MPLS, IPsec, and recent ID layer, are awkwardly inserted into the existing layers. This violation comes from the fact that layering separates networking functionality only vertically within a system. Recently several approaches tried to escape from the rigid vertical layering principles; those are cross-layer [Feldman07], [Conti04], [Goldsmith02],

[Haapola05], layer-less [Johnsson03], role based model [Prasad08], and horizontal model [Braden03], [Hakala06].

3.5 Loose Coupling principle

In the architectural context, ***coupling*** is the degree to which each architectural module relies on each one of the other modules [Stevens74]. ***Loose coupling*** is a method of interconnecting architectural components of a system so that those components depend on each other to the least extent practicable. The best example of loose coupling in the communication stack of the Internet is the decoupling between applicative layers and the TCP/IP protocol. The extent of coupling in a system can be qualitatively measured by noting the maximum number of element changes that can occur without adverse effects. Examples of such changes are adding elements, removing elements, renaming elements, reconfiguring elements, modifying internal element characteristics and rearranging the way in which elements are interconnected.

In today's Internet design, "*Modularity is good. If you can keep things separate do so*" [RFC1958]. Moreover, [RFC3439] defines the "***Coupling Principle***" by stating that as things get larger, they often exhibit increased interdependence between components. This principle is also expressed as "unforeseen feature interaction" [Willinger02] and means that if more events occur simultaneously, the likelihood that two or more will interact is increased. Much of the non-linearity observed in large systems is largely due to coupling. Coupling has both horizontal and vertical components. In the context of networking, horizontal coupling is exhibited between the same protocol layer, while vertical coupling occurs between layers. *Loosely coupled systems are said to have more flexibility in time constraints, sequencing, and environmental assumptions than do tightly coupled systems.*

Loose coupling simplifies testing, maintenance and troubleshooting procedures because problems are easy to isolate and unlikely to spread or propagate. Loose coupling appears thus a necessary condition for a well-structured computer system and a good design. When combined with high cohesion, supports the general goals of high readability and maintainability. Loose coupling minimizes unwanted interaction among system elements. However, loose coupling can also give rise to difficulty in maintaining synchronization among diverse components within a system when such interaction is desired. In some systems, a high degree of element interdependence is necessary for proper functionality.

In addition, systems with complex interactions and tight coupling are likely to have unforeseen failure states (of course, complex interactions permit more complications to develop and make the system hard to understand and predict); this behaviour is also described in [Willinger2002]. Tight coupling also means that the system has less flexibility in recovering from failure states.

3.6 Locality Principle

Locality principle has multiple roots. In physics it refers to the principle that local cause(s) shall result in local effects [Einstein48]. In computer science, the locality principle guides the design of thrashing-proof, self-regulating, and robust logical systems. The principles of locality are facts about data and instruction accesses that arise from patterns that are frequently used for storing and accessing information. This principle is useful wherever there is an advantage in reducing the apparent distance from a process to the information or data it accesses. Locality is among the oldest systems principles in computer science guiding the design of robust replacement algorithms, compiler code generators, and thrashing-proof systems. It was proposed in 1967 by Denning [Denning67] during efforts to make early virtual memory systems work well (working set memory management). The locality principle

found application well beyond virtual memory. Today it directly influences the design of processor caches, disk controller caches, storage hierarchies, network interfaces, database systems, graphics display systems, human-computer interfaces, individual application programs, search engines, etc. In the future, this principle is expected to help overcoming problems with brittle, unforgiving, unreliable, and unfriendly software.

One shall distinguish the principle of Temporal Locality (recently accessed data and instructions are likely to be accessed in the near future) from the principle of Spatial Locality (Data and instructions close to recently accessed data and instructions are likely to be accessed in the near future) leading to a combined principle of Locality where recently accessed data and instructions and nearby data and instructions are likely to be accessed in the near future. From this perspective cache-based systems will heavily rely on the application of the locality principle.

In distributed systems, mechanisms such as reachability information aggregation or segmentation of routing systems in areas are means to keep local changes with local effects (by hiding details to remote processes that can still run without that knowledge) and thus closer to the physics interpretation. Protocols that didn't respect this principle are today under heavy pressure to cost-effectively sustain the dynamics of the Internet. For instance, we think here of Border Gateway Protocol (BGP) which explores all transient states during re-convergence process and distribute the corresponding information to its neighbours before converging to the next stable state.

3.7 The “End-to-end” and the “minimum intervention” principles

Introduced by [Saltzer84], this principle is one of the fundamental principle around which the Internet architecture has been structured and built. This principle ensures applications to survive partial network failures (note: survivability is one of the main design objectives of the Internet architecture). The end-to-end principle deals with functional placement and the spatial distribution of functions across the layers of the communication stack by stating that "*The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement).*" It implies that a mechanism should not be placed in the network if it can be placed at the end node (provided it can be implemented "completely and correctly" in the end nodes except for performance enhancement[10]) and that the core of the network should provide a general connectivity service, not one that is tailored to a specific application.

The end-to-end principle has important consequences in terms of protocol design that should not rely on the maintenance inside the network of state information, i.e., information about the state of the end-to-end communication. The application of the end-to-end arguments results in a network that is transparent to the host application communication (abstraction principle) and provides for a general transport service capable of supporting many different applications. So, contrary to the telephony networks (PSTN), the Internet is not tailored for any single application but is designed for generality and (expectedly) evolvability. On the other hand, the end-to-end principle (as such) is not specific to the Internet design, e.g., X.25 also runs end-to-end. It is the association with the "fate sharing" and the "minimum intervention" of the end-to-end principle that gives to the Internet its distinguishing property.

[10] The best example is congestion control whose primal component resides at TCP level and dual component at IP level.

The end-to-end principle is also guiding placement and distribution of functionality inside the network rather than at host/end-systems if all applications need it, or if a large number of applications benefit from an increase in performance while keeping the ratio cost/performance acceptable. The fate sharing part is intrinsically related to user-stateless nature of the network, i.e., the network does not maintain any state information about the flows that traverses the network; this ensures survivability of the application in case state information would be lost due to partial network failures. Such state should be maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks (from this perspective the connectionless packet switching principle can be seen as enabling fate sharing).

3.8 The "simplicity" principle

One of the major design principles of current Internet is the simplicity principle. It has also been expressed as the KISS ("Keep it Simple, Stupid") or the "Occam's Razor"[11] principle. It states that, when in doubt during design, choose the simplest solution [RFC1958].

The fundamental motivation of this design principle has been formulated by J.Doyle [Doyle02] "*The evolution of protocols can lead to a robustness-complexity-fragility spiral where complexity added for robustness also adds new fragilities, which in turn leads to new and thus spiralling complexities*". In other terms, the addition of functionality or improvement of performance should not be performed at the detriment of increasing complexity.

3.9 Connectionless packet switching and distributed adaptive routing

The most fundamental service of the Internet relies on packet switched network, which provides unreliable, best-effort, connectionless packet delivery. The service is called "connectionless" since packets can be delivered without any prior end-to-end connection setup phase[12]. The forwarding decision is taken per-packet, independently at each node (referred to as "hop"): upon receiving packets, nodes lookup their routing tables to determine the outgoing interface for that packet. The routing mechanism is called as "***proactive routing***" since all the entries, including default entry, in the routing table must be setup before packet delivery [Baran62] [Baran64]. Any packet can use the full link bandwidth on any link but may have to wait in a queue if other packets are already using the link. If a datagram traverse a hop with a full queue it is simply dropped, which corresponds to the best effort service principle. The delivery service is thus unreliable because packets may be lost, duplicated, delayed, or delivered out of sequence. The service is also called best-effort since delivery is not guaranteed and does not discard packets capriciously. This switching mode also implies that it is possible to use a stateless forwarding system at the network layer, which does not require per connection state. This ensures scalability and contributes to the cost effectiveness of the communication system and its communicating entities.

The seminal work of P.Baran aims to design a robust distributed communication system that could operate even if many of its links and switching nodes were destroyed so as to overcome the resiliency limitation of (de)centralized communication systems. For this purpose, Baran envisioned a network of unmanned nodes (applying the principle of minimum intervention at the control level) acting as switches that forward datagrams from one node to another to their

[11] The "Occam's Razor" is a principle that generally recommends, when faced with competing hypotheses that are equal in other respects, selecting the one that makes the fewest new assumptions.

[12] Connection and local provisioning/configuration actions are not synonyms.

final destinations. Nodes would use a scheme P.Baran called "hot-potato routing", foundation of ***adaptive routing***.

3.10 Network of collaborating networks - interconnection via gateways

The Internet is often called “network of networks” since it is composed of subnets with heterogeneous data link layer techniques and autonomous systems with independent operation domains. Following the principle developed by Pouzin [Pouzin71], routers provide for the inter-connection of network devices of the Internet infrastructure that is sub-divided into a collection of autonomous systems (AS) managed by an Internet Service Provider (ISP). Within an AS, routing is determined by interior gateway protocols such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Inter-domain routing between ASs is controlled by the Border Gateway Protocol (BGP). The latter is the policy-based path-vector routing protocol of the Internet. This protocol distributes routing information between routers belonging to different AS. Following Baran's approach, this design of the routing system ensures survivability and allows for distributed management as long as ISPs are (at least partially) collaborative. Routing decisions are made on per-IP-networks basis; for example, ARP in subnet, OSPF and IS-IS within an AS, and BGP among ASs. The Internet can be viewed as a flat architecture in the sense that every communicating host has assigned IP addresses and shares the common IP packet format; but in the other sense its control structure is organized in ISP tiers.

A good example of misuse of this principle is illustrated by the following example: One important aspect of this autonomy is the address space control; in this sense, a network separated by Network Address Translation (NAT) [RFC1631] can be viewed as a typical example of independent domains - noticing that it violates the end-to-end principle. Originally NAT has been proposed for overcoming the shortage of the available IPv4 addresses by reusing the address space. They decouple routing in the private network from routing in the public Internet. NAT enables the use of a smaller pool of IPv4 addresses (address amplification) and independence of providers address allocation (mimics provider-independent allocated space). This comes with a list of serious drawbacks including i) NAT prevents or impairs end-to-end diagnosability, troubleshooting, backtracking, etc., ii) the combination of IPSEC with NAT poses a series of problems as explained in [RFC 3715], iii) split DNS (local vs. global address space), iv) performance deterioration, e.g., delay induced by re-computing header checksum, scaling in terms of #active sessions/states), v) direct access to CE/Home gateways is complex to operate without support of the ISP, and vi) NAT gives the "impression" of securing the local address space (security by obscurity).

4 EVOLUTION OF EXISTING PRINCIPLES

4.1 Principles that should be preserved

In this section we highlight design principles that apply to current Internet and we believe that they should be preserved also in the future architecture of the Internet. Other principles should be adapted or augmented.

4.1.1 *Heterogeneity support principle*

Heterogeneity at the applications, services, terminals, network topologies and characteristics significantly complicates Internet operation. As a result, one of the major high-level objectives of the Internet is to "leverage and evolve information and communication technologies as well as capabilities and services to fulfil increased quantity and quality of Internet use considering the requirements from an increasingly heterogeneous set of applications" [FIArch]

In the future, the heterogeneity is expected to be much higher than today. Multiple types of terminals/hosts, multiple network nodes, multiple protocols, and multiple applications will exist. Hence, the capability to support **heterogeneity should remain as one of the main design principles.**

4.1.2 Scalability & the Amplification Principle

The number of devices with Internet access (e.g., PCs, laptops, mobile phones/smart phones, PDAs), communication nodes (e.g., home, access, edge and core routers), autonomous systems, services, applications in the Future Internet is expected to significantly increase. Moreover, the direct interconnection of the sensor networks with the legacy Internet will exponentially increase the number of Internet nodes. If one sees the Internet currently comprising three level of tiers, extension of the Internet at its periphery could expectedly lead to a fourth tier which would have a fundamental impact on the properties of the routing system.

As a result, we believe that ***scalability is among the major design principles that should govern Future Internet, while the amplification principle would definitely remain.***

4.1.3 Robustness principle

In the future, the Internet is expected to handle mission and time critical applications, related with health, energy, transport and financial transactions. On the other hand, the Internet progressively replaces existing application specific networks (e.g. broadcast networks) and other communication medium from newspapers to postal mails -- uniformitarian of the communication medium -- it becomes critical to ensure its robustness (otherwise the decrease of diversity in communication medium may amplify the impact of bug occurrence). As a result, part of the robustness principle that covers issues related to minimizing the malfunction, uninterrupted operation and interoperability, remains unchanged.

Yet, as it will be explained in Section 4.2.3, we believe that ***the robustness principle should be extended/ adapted to cover security issues***.

4.1.4 Loose Coupling principle

As explained in Section 3.4, loose coupling appears to be a necessary condition for a well-structured system and a good design as: i) it simplifies testing and troubleshooting procedures because problems are easy to isolate and unlikely to spread or propagate, ii) combined with high cohesion, it supports the general goals of high readability and maintainability, and iii) it minimizes unwanted interaction among system elements. In addition, tightly coupled systems are likely to have unforeseen failure states (as complex interactions permit more complex systems to develop and make the system hard to understand and predict) and implies that the system has less flexibility in recovering from failure states. For these reasons this design principle shall be preserved in Future Internet and even reinforced as a result of the increasing importance of the availability objective (see [FIArch]).

Nevertheless, recent evolution shows that loose coupling can also increase difficulty in maintaining synchronization among diverse components within a system when a higher degree of element interdependence is necessary. Hence, it would be appropriate to consider that under stress conditions, higher cohesion should be possible for proper functionality. Grading coupling level at running time could be considered as a mean to circumvent this problem.

In addition, the proliferation of various multi-agent distributed systems with complex interactions raises a couple of issues that were not initially foreseen. Note that in such systems, the individual subsystems can be simple as well as their basic rules, the overall system resulting from these complex interactions becomes sophisticated and elaborated. Therefore, these systems are prone to the emergence of nonlinearity that results from the coupling between components, i.e., the positive feedback (amplification) loops among and between subsystems and unending oscillations from one state to another. It is possible to prevent the known amplification loops and unstable conditions to occur but still impossible to anticipate and proactively set the means to prevent all their possible occurrences. In these conditions, ***it is fundamental to prevent propagation and that each system keeps its own choice as last resort decision, and become "conservative to what each system accepts and adopts".***

4.1.5 Locality Principle

The locality principle has played a very important role in computer design, programming and the Internet the last decades. Following the principles of spatial and temporal locality, M.Wilkes introduced in 1965 [Wilkes65], cache memory fulfil the speed gap between the CPU and main memory. Based on this concept, recent advances in computer systems engineering have pushed cache memory to higher levels in the computer systems but the essence remains the same: reflect the chosen methods for using the principles of spatial and temporal locality. In this context, the principles of spatial and temporal locality will have to be extended to distributed computing systems and to the higher layers space of distributed application architectures.

On the other hand, locality will play a fundamental role in self-stabilizing distributed systems by ensure sub-linear stabilization with respect to the number of local system components and interactions among components.

As a result, we believe that the ***locality principle is important and should be preserved, while its scope should be extended to cover additional roles in distributed systems and distributed application architectures.***

4.2 Principles that should be adapted (modification of existing description)

In this section we highlight design principles that have been described and apply to the current Internet architecture. Yet, we challenge that they should be adapted in order to address the design objectives of the Internet.

4.2.1 Keep it simple, but not "stupid" principle

As already explained, one of the main design principles of the Internet was the approach to keep things simple (the term things refers here in particular to protocols and intermediate systems). If there were many ways to do the same thing, one should choose the simplest one [KISS]. This common sense engineering principle continued requirement to make the usage of network functionality simple and robust, but more processing logic is needed in order to achieve the expected functionality.

In current Internet design, the complexity belongs at the edges, and the IP layer of the Internet remains as simple as possible. Complex systems are generally less reliable and flexible. Architectural complexity (defined in Section 2) implies that in order to increase the reliability it is mandatory to minimize the number of components in a service delivery path, where the service delivery path can be a protocol path, a software path, or a physical path.

However, this principle has already been challenged. Complex problems sometimes require more elaborated solutions and multidimensional problems such as the Internet architecture will be providing non-trivial functionality in many respects. The general problem can be seen as follows: determine the placement and distribution of functionality that would globally minimize the architectural complexity. In that respect, arbitrary lowering complexity (over space) might result in local minimum that may be globally detrimental. Thus, when designing the Internet, the famous quote attributed to Albert Einstein[13] may be adopted: "***Everything should be made as simple as possible, but not simpler***".

Though we have to recognize that this principle is still weakly applied, together with the conclusion of Section 4.1.2, we support that ***scalability and simplicity should be handled as strongly interconnected first priority principles of the Future Internet.***

4.2.2 Relaxing the "Minimum Intervention" Principle

As stated in [RFC3439]: "*Interworking function approaches have been problematic at large scale"* while the principle of Minimum Intervention states that: *"To minimize the scope of information, and to improve the efficiency of data flow through the Encapsulation Layer, the payload should, where possible, be transported as received without modification*".

The minimum intervention principle is critical to maintain and preserve data integrity and to avoid useless intermediate information message or packet processing. Deep Packet Inspection (DPI) and network coding provide two good examples of detrimental intermediate in-band processing of packet flows. Moreover, in some cases, it may conflict with the simplicity principle; e.g. in sensor networks and the Internet of Things where communication gateways and actuators meant to enable communication between networks by offloading capabilities that would be costly to support on sensors.

As a result, we propose to ***relax the minimum intervention principle as a design principle.***

[13] See discussion on <http://quoteinvestigator.com/2011/05/13/einstein-simple/>.

4.2.3 Robustness principle

Over recent years, in order to increase robustness and system reliability, some have advocated to transform this fundamental principle from “be liberal in what you accept, and conservative in what you send" into "be conservative in what you send and be even more conservative in what you accept from others". However, adoption of this approach would result into dropping a significant level of interoperability between protocol implementation. Indeed, being liberal in what you accept is the fundamental part that allows the Internet protocol to be extended.

With the anticipated architectural evolution of the Internet, another aspect of interoperability will play a critical role: "how to change the engine of plane while flying". Moreover, we shall account that the new engine can be of completely different nature than the one it replaces (the new engine being not necessarily driven by the same principles that were used to construct the old one). There is no universal operational principle telling how such transition should best be performed nevertheless it is possible to provide the minimal conditions the new system has to support in order to facilitate this transition.

This principle shall thus be maintained but at the condition that it doesn't come at the detriment of reliability (is it suitable for receiver to accept bogus information?) and security by being too liberal (a system that spend 100% of its CPU cycle to respond to bogus messages becomes unusable - its utility cycles drop to 0). Remember here that is general much better to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect as this assumption will lead to suitable protective design.

Nevertheless with respect to security, this design principle leads to weak security architecture thus requiring adaptation. Indeed, the RFC 1122 version of the Robustness Principle advises: "*In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect. This assumption will lead to suitable protective design..."* Notice also that this version says that it applies at "*every layer of the protocols*" including the application layer. However, it does not provide any guideline on which "suitable protective" component shall be part of a common architectural baseline.

Moreover, as stated in [RFC1958] “*It is highly desirable that Internet carriers protect the privacy and authenticity of all traffic, but this is not a requirement of the architecture. Confidentiality and authentication are the responsibility of end users and must be implemented in the protocols used by the end users*” [RFC1958]. Henceforth, we argue that ***the principle should be adapted to incorporate self-protection structural principle (coordination of the local responses to external intrusions and attacks including traffic, data and services traceback that would enforce in turn accountability) as well as confidentiality, integrity and authentication should be inherently offered to information applications and services.***

4.2.4 Modularity Principle

Current communication systems are designed as a stack of modules structured by static and invariant binding between layers (modules) that are specified at design time. Indeed, when they were developed CPU and memory were scarce resources and the expected uniformity of their utilisation (computing machine interconnection) lead to a design optimizing the cost/performance ratio at design time.

Nowadays, looking at current evolution with i) repetition of functionality across multiple layers, e.g., overlays that allow carrying TDM over IP over Ethernet over IP/MPLS, emphasize the need to define common patterns, monitoring modules repeated over multiple layers (which then requires to recombine information in order to be semantically interpretable) as well as security components each associated to a specific protocol sitting at a

given layer (which result into inconsistent response to attacks), ii) as part of the same layer, the proliferation of protocol variants all derived from a kernel of common functions/ primitives, iii) the variability of external and internal events that communication systems have to cope with emphasize that the cost/performance objective to be met by communication systems can vary over time (thus messages would be processed by variable sequence of functions determined at running time), iv) the increasing heterogeneity of environments where communication systems are involved emphasize that some of these functions may be more or less elaborated, and v) Increasing heterogeneity of running conditions as well as increasing occurrence of unexpected events leads to i) consider modules connected by means of realization relationships that supply their behavioural specification, ii) distinguish between general and specialized modules (inheritance) and iii) enable dynamic and variable binding between the different modules such that the sequence of functions performed is specified at running time.

This being said, in the current architecture, the transport module is not complete and thus not modular. Indeed the transport address depends on the IP address and more generally its usage relationship does exclusively depend on the existence of other modules in the stack: one can't replace or use a TCP stack without knowledge of how it will impact operations of other modules.

4.3 Principles that should be augmented (addition to the existing description)

In this section we highlight design principles that have been described and apply to current Internet but we challenge that they should be augmented or extended.

4.3.1 *Polymorphism principle (as extension to the modularity principle)*

Polymorphism (ability to take on different forms) in computer science/programming space applies to data (generalized data type from which a specialization is made) or functions (function that can evaluate to or be applied to values of different types). It enables to manipulate objects of various classes, and invoke methods on an object without knowing that object's type.

The introduction of polymorphism principle is driven by the motivation to make use of this fact to make our architecture simpler. In many cases, the modularity and layering principles have been the driving principles for both communication protocols and software implementations. This principle has led to faster deployments, but suboptimal solutions; as such these principles have been challenged in many cases, especially in environments where functions of each layer needs to be carried out completely before the protocol data unit is passed to the next layer.

In this context, polymorphism enables to manage and operate first class objects belonging to different kinds of classes (which within a hierarchy often share common methods and attributes) while providing the ability for a super-class to contain different objects of a subclass type at different points in time. In turn, this allows i) for objects of different classes to respond differently to the same function call thus results in different functionality being executed for the same method call, and ii) for run-time (dynamic) instead of compile-time (static) binding. Henceforth, the introduction of ***polymorphism would enable the same abstract and autonomous loosely coupled components/objects to have different functional and non-functional behaviour under different environments or circumstances.*** The question remains open though as how to parameterize these environmental variables and

whether this parametrization could be performed through distant exchanges (remotely) which would turn this principle close to the concept envisaged by active networks in the late 90's.

4.3.2 Unambiguous naming and addressing principle

As stated in RFC 1958, the Internet level protocol are and must independent of the hardware medium and hardware addressing. This approach allows the Internet to exploit any new digital transmission technology of any kind, and to decouple its addressing mechanisms from the hardware. It allows the Internet to be the easy way to interconnect fundamentally different transmission media, and to offer a single platform for a wide variety of Information Infrastructure applications and services.

Concerning name and addressing, the following augmentations are considered (using RFC 1958 as starting point):

- Avoid any design that requires addresses to be hard coded or stored on non-volatile storage. When this address this is an essential requirement as in a name server or configuration server a discovery process is recommended. In general, user applications should use names rather than addresses. In that respect the transport layer address should be decoupled from any locator and use space invariant identifiers associated to the communication end-point. In turn, this would facilitate dynamic multi-homing, TCP connection continuity (required for mobility) and more generally misuse of IP addresses.
- A single and common naming structure should be used.
- LOC/ID separation (a.k.a. LOC/ID - initially proposed in NIMROD): resulting from the overload of IP address usage, upper layer protocols must be able to determine end-point identifiers (ID) unambiguously, and make use of locators (IP addresses) strictly for end-to-end routing (processing at intermediate nodes) must be the same at start and finish of transmission. This separation involves the need to provide the capability for mapping (or resolving) identifiers to locators at the end-points. Both identifiers and Locators must be unambiguous and be unique within any scope where they may appear.

In the future, it is foreseen that not only the end-points (ID) and their attachment points (LOC) need to be unambiguous and unique within the scope in which they appear and are used, but the data and the services, as well. At the end, in most cases, the user is not willing to access a specific server, but the content or the services that this server hosts or offers. If exactly the same data (e.g. content, type, quality, security, …) and/or service (i.e. functional and not functional matching) can be provided in another way (e.g. from another server or method), it is also acceptable and in many cases even preferable if the actual quality is better (or cost lower).

Moreover, the current ID/LOC approach only deal with hosts and can not provide a method to ensure that an entity is the one claiming to be or, even worse, they disclose a fixed identifier that can be easily traced by any other network element to know the operations that an entity performs, thus violating its privacy.

In Future Internet, naming and addressing as a design principle should be ***extended to unambiguous identify hosts, resources, data, services.***

4.3.3 *Extending the end-to-end principle*

Historically, the "end-to-end principle" has been one of the most controversial issues in the Internet innovation. Many experts in the area insist that the "end-to-end" principle is still valid as it applies as the communication is divided at autonomous legs. However, the clear definition of communication end-points becomes more and more complex to delimit, as middle boxes and application layer gateways are deployed at the edges of networks[14].

Another challenge concerning this principle is that IP overlay applications such as IP multicast and mobile IP (MIP), require support from intermediate nodes (RP in ASM, and Home Agent in MIP). It is important to notice though that some of these supports are purely driven by arbitrary choices, e.g. PMIP for mobility management or delayed migrations, e.g., NAT instead of rolling out IPv6. Another challenge comes from the Internet of Things/Smart objects communication, where the end-to-end communication may be significantly modified by intermediate gateways and sensor networks sink nodes.

It is also well perceived that for many modern applications (e.g. mobile applications, distributed searching, certain aspects of collaborative computing) maintaining state information within the network may now be desirable for efficiency if not overall performance effectiveness. To argue today that the only stateful elements that may be active in the Internet environment should be located at the edges of the Internet is to ignore the evolution of software and other technologies to provide a host of services throughout the Internet [WGIG04].

Finally, as stated in the [FIArch] and further analyzed in [RFC6077], support of congestion control cannot be realized as a pure end-to-end function: congestion is an inherent network phenomenon that in order to be resolved efficiently require some level of cooperation between end-systems and the shared communication infrastructure. Instead of placing specific functions in specific positions—either in end systems or routers in the network core—services and functions must be allowed to be deployed anywhere they are needed [RFC3234].

As a result, we believe that ***motivations to "update" or augment this principle increase; however even if this principle is challenged, due to the heavy consequence in terms of scalability, survivability and robustness on the Internet at large departing from this principle remains open.***

[14] Please note the definition of end-to-end communications that we have already given.

5 SEEDS FOR NEW DESIGN PRINCIPLES

So far we have presented design principles that should be preserved, adapted or even augmented. Yet, we believe that the Internet Architecture will evolve from a network architecture to a complete communications' operational ecosystem, which will be extended to capture resources of any type (any type of network nodes and their subcomponents and supporting services, any type of network enabled services and any type of content or information) and be extended to even socio-economic dimensions. To realise such Internet Architecture ecosystem requests design principles that go well beyond the networking and primitive services aspects. In this section, we introduce seeds for completely new design principles that may apply to the evolution of the Internet Architecture.

A seed for a new design principle refer to a concept or a notion at the inception of a well formulated design principle. The term seed acknowledges that i) formulating principles is a complex exercise, ii) research is still ongoing in proving their value and utility (some of our analysis and exploitation of research results may not be mature enough) but also impact, and iii) the proposed seeds may not be flourishing (a lot of proposal came in and very few will materialize).

5.1.1 Resources Awareness

Taking into consideration that resources refer to different types (i.e. service components and infrastructure resources), abstractions of these resources will contribute towards specific situations. In this context, FI architecture should include **resources as first order abstractions**, in order to facilitate resource and situation awareness. Services are the means for users (organizations, public bodies, companies and people) to get *controlled* access to the available information and functionalities provided through the Internet. While in general this principle could be applied both to service components and to infrastructure resources, within this description emphasis is put on services (in terms of service components).

Including resources as first order abstraction requires the enhancement of current data-oriented management approaches with solutions more oriented to services, **even at the layers below application**. The latter is of major importance considering the predicted growth of media traffic[15], which raises specific requirements towards FI infrastructures. While current offerings are based on service-unaware approaches, the increased growth of both data and user-generated services poses the need for delivery schemes (allowing media coding and rich service characterization) to overcome limitations with regard to efficiency[16]. Thus, it is necessary to provide the application level with more complex or meaningful abstractions than the ones currently adopted for services, which are currently seen just as URIs and/or APIs to be invoked over the FI infrastructures. It is however also necessary to guarantee that **the service components network as a whole is able to manage these abstractions**. This means that the abstractions should influence the behaviour of the operations on the infrastructure layer based on those lower levels.

Addressing the aforementioned challenges requires (establishing design principles that support) the definition of suitable abstractions and mechanisms for allowing the cooperation across all FI infrastructure levels.

[15] Cisco Visual Networking Index (VN): Forecast and Methodology, 2010-2015

[16] Content Networking: Architecture, Protocols, and Practice", by Markus Hofmann and Leland Beaumont, Morgan Kaufmann, February 2005

Each layer of the FI architecture should be a self-aware and self-managed set of services that works in isolation to provide specific functionalities along with quality guarantees, but at the same time it must cooperate with the others for enabling a holistic service provision. Cooperation modalities should be defined in order to:

1. Guarantee that each level is aware, at a certain level of abstraction, of the effects on the levels above of achieving or not its guarantee, and
2. Allow each level to negotiate and agree certain guarantees with the levels below.

This service awareness of the infrastructure as a whole is expected to benefit service delivery and allow for higher levels of interactivity. Moreover, network operations based on the lower levels (e.g. routing) will benefit from being able to understand services as first order abstractions, and to optimize their behaviour according to them.

This principle is strongly related to "**modularization** by layering", and should complement it by specifying requirements on the functionalities that each module exposes for supporting cross-layer cooperation.

Furthermore, applying this principle in combination with the "**loose** coupling" one, will allow for understanding and evaluating the effects of cross-layer awareness and cooperation, in order to avoid or minimize unwanted interactions and non-linear effects.

Another principle that needs to be considered refers to "**locality**". Given the need to reduce the distance from a process (i.e. service) to the corresponding data, service awareness will contribute by allowing the development of delivery models (applied both to services and data), being enabled through self-management and cross-layer cooperation approaches.

5.1.2 Dependability Logic

In the current Internet there is a lack of methods and means for reliable, accountable, and verifiable processing and handling of network and systems infrastructure with respect to the services they host. This is particularly important in many critical environments, such as health care, transportation and manufacturing where compliance with legal regulations, performance requirements, etc. are of critical importance and must be guaranteed and verifiable.

In future, large-scale Internet-based service deployments, the current and contemporary issues beneath it encountered by consumers will and must be tackled. Indeed, with the current design of the Internet:

- Services are not a "cure-all" and are not cognisant of end-user expectations and needs, especially for enterprises and mission critical applications. Moreover, if existing, grade of service are often static, lack of flexibility and not negotiable. Often it is left up to the users/clients to implement their own systems to ensure the service performs as expected.
- Services operate on a "best-effort" basis. Indeed, in certain parts of a service stack there is little or no comprehension of performance expectations. Moreover, services are often not accountable towards the end-user.
- Services are modelled prior to their deployment in any environment and according to the aforementioned modelling scalability rules and policies are enforced during runtime. Nevertheless and given that infrastructures are application-unaware, the enforced scalability rules and policies are not always adequate to meet the application requirements in terms of efficiency (e.g., in cases of multi-tenancy), performance (e.g., scaling after a specific level doesn't lead to better performance), etc.

- Dynamic distributed environments ask for control policies able to deal intelligently and autonomously with problems, emergent situations, tasks, and other circumstances not necessarily envisaged at the design time.

To address this, ***the design of the Internet including its services must be imbued with the principle of dependability (reliability - accountability – verifiability) including self-adaptation and self-learning capability to cope and learn from changes in the operating conditions.***

However, enabling such capability (reliability - accountability – verifiability feedback loop) shall not result into monopolistic or a monolithic-proprietary designed architecture. In that respect this principle ought to provide means to avoid vertical integration with proprietary components. This critical element is part of the open research questions remaining unaddressed since so far/at the time of writing.

5.1.3 Allow the exchange of information between end-points of different type

The Internet has evolved over the years from a research-oriented network of networks to a playground for many different ***stakeholders*** such as Internet Service Providers (ISPs), Content Distribution Network providers (CDNs), Content Providers (CPs), end-users, etc. The set of stakeholders can be divided either vertically in *players, e.g. two ISPs*, or horizontally in *layers, e.g. the Transport layer*; all three terms will be used below equivalently.

These stakeholders try to optimize own utilities (or more generally benefits) e.g. ISPs to reduce inter-domain costs, CDNs to improve content routing, users to benefit from different choices (e.g. by making choices of application providers or ISPs, or of application parameters, etc.), each one on the basis of the incomplete information available thereto. The so-called ***Information Asymmetry*** between different stakeholders of the Internet leads often the ecosystem to a suboptimal performance; e.g. see [Liu05]. Addressing the information asymmetry problem may allow stakeholders to make alternative ***decisions*** that would lead them collectively to a more beneficial state.

Furthermore, Clark et al. [Clark05] proposed the *Design for Choice* principle that suggests that Internet technologies should be designed so that they allow variation in outcome, rather than imposing a particular outcome. The rationale behind this is that the Internet is a rather unpredictable system and it is very difficult to assess whether a particular outcome will remain desirable in the future.

In order to both enable the *Design for Choice* principle and address the *Information Asymmetry* problem, we introduce the ***Allow the exchange of information between layers and players principle***, which suggests that ***different stakeholders should be able to provide to others information on possible choices and their preferences***. In this way, stakeholders that are provided this information are able to express their interests, to coordinate their objectives and to align their incentives, if these are indeed compatible; as well as to appreciate what the effect of their choices to others will be. Incentive compatibility between players applies when one player's selfish action implies the improvement not only of his own objective but also of those of the other players. This information diffusion can possibly lead to the so-called *"all-win"* situation, whereby all existing players are better off, at least temporarily. In the long term and if new stakeholders also enter/exit the ecosystem then action is anticipated by the remaining ones.

In practice, the application of the proposed principle implies the design and deployment of more “open” systems and interfaces for the interaction/communications between different stakeholders anticipating also users’ reactions in cases of unsatisfactory quality of experience. Therefore, all stakeholders, including users, will have the ability to react, by means of making new choices, in cases of unsatisfactory benefit (for users: quality of experience or value for money).

The exchange of information between stakeholders implies a flow of information from one stakeholder to another, and the “processing” by each stakeholder; therefore the *constituent capabilities* of this principle include:

- the exposure of information to a stakeholder,
- the abstraction/aggregation of information to be exchanged.
- the collection of information by a stakeholder,
- the assessment of information by a stakeholder, and
- the decision making.

The exposure of information addresses the Information Asymmetry problem but should be restricted to the necessary level, so that no “sensitive” information is exposed that could damage its original owner/producer. This is taken care of by the second capability, which is very important and in fact which also provides efficiency in the information exchange procedure. The idea behind this capability is that critical details must be hidden to avoid being exposed to competitors, while required information should be exchanged in a way so that no repurposing of this information can take place. The implementation of suitable interfaces is required for this to be attained.

The remaining three capabilities are incentive compatible, since the stakeholder that each time collects, assesses the information, or makes a decision based on that information will have available more (if not full) information to optimize his own utility.
Two open research questions remain to be further explored:

- how to ensure the application of the principle doesn't partition the shared common infrastructure between islands where certain information gets exchanged that becomes at the end detrimental for the end-user (lock-in)?
- how to ensure fairness of gain among participants/stakeholders (how to prevent that the "rich gets richer") meaning that exchanges of information does not progressively falls into hands of a minority of highly connected hubs?

The aforementioned issues are need to be address by the use of more “open” systems and interfaces that will allow for creative solutions by small players to flourish (expressed as new, revolutionary technologies), thus leading to a more fair distribution of the gains among them, despite the inherent differences among them w.r.t. their ability to invest.

5.1.4 Sustain the Resources and Brain Investment

“Competition” is the result of competing antagonistic actions due to conflicting interests between parties implicitly cooperating in technological terms, but resulting into negative global return - this technical term has its associated and overused buzzword: "*tussle*" as introduced by D. Clark in [3]. Investigating candidate tussles in possible Internet evolution scenarios is a way to understand what the socio-economic incentives of the different stakeholders are (leaving the open question of who represents people/Internet users in such studies) [Kalogiros11]. One possible tool for this is to employ tussles analysis in possible Internet evolution scenarios. Indeed, addressing the inevitable tussles "correctly" (by giving adequate control to actors to influence/negotiate the outcomes that make sense from a

technology point-of-view, e.g., not per packet) should reduce the global negative return. On the other hand, this does not mean that the Internet should be designed to sustain conflicting interests or steer them to unfair outcomes (i.e., not just be *Designed for Tussle* [3]) but instead be designed so as to lead to a global positive return for the all of its users (as individuals but also as member of various communities), the so-called "*all-win*" situation but also for the society at large.

Instead, it is important that the Internet is designed to ***sustain brain investment, innovation investment and resource investment toward a global positive return***. For this purpose, it is fundamental to first recognize here the capability of the Internet to accommodate since so far *new applications communicating over a commonly shared infrastructure* (and it basically because the architecture was not designed with the idea to privilege one class of actor against another). It is thus essential to keep the entry barrier as low as possible and structure the design of the Internet so as to allow various communities and people's involvement by, e.g., steer open applications development but without impeding the genericity, evolutivity, openness, and accessibility design objectives. Over time, the Internet shall thus cultivate the opportunity for new players to take benefit of the infrastructure foundation without sacrificing on its global architectural objectives and design principles. Moreover, the Internet architecture should be able to accommodate and sustain its actors and stakeholders' needs in terms of fundamental capabilities, e.g. forwarding and processing capacity.

However, it is not technically possible neither operationally feasible to homogenize user satisfaction, utility functions and individual interests across the entire Internet. Nevertheless, it should be made possible for Internet communities (e.g. users, developers, enterprises, operational communities) to reward (with positive feedback) architectural modules/components (together with the interactions among them) that deliver positive returns; in turn, leading to positively weighted modules (i.e., strong modules as one could refer here to the "strength" of a module as a measure of its reward by the community) and to progressively deprecate modules/components with negative return (i.e., weak modules).

6 CONCLUSION

The new demands of Internet can be addressed to a certain degree through incremental infrastructure investment combined with "over-dimensioning". However, analyses have shown that increasing the bandwidth to peta-bps on the backbone network will not suffice due to new qualitative requirements in, for example, highly critical services such as e-health applications, clouds of services and clouds of sensors, new social network applications like collaborative 3D immersive environments, new commercial and transactional applications, new location-based services and so on.

Design principles have played a central role in the architecture of the Internet as driving most engineering decisions at conception level but also operational level of communication systems. In this document, the FIArch groups has identified the design principles that we expect to govern the architecture of the Future Internet.

This may serve as a starting point and comparison basis for all research and development projects that target Future Internet Architecture.

7 REFERENCES

[RFC3439] R. Bush, D. Meyer, “Internet Architectural Guidelines,” December 2002

[RFC1958] B. Carpenter, “Architectural Principles of the Internet,” June 1996

[RFC3234] B. Carpenter, “Middleboxes: Taxonomy and Issues,” February 2002

[RFC1631] K. Egevang, P. Francis, “The IP Network Address Translator (NAT),” IETF RFC 1631

[Braden03] R. Braden, T. Faber, M. Handley, From Protocol Stack to Protocol Heap – Role-Based Architecture, ACM SIGCOMM Computer Communications Review, Vol. 33, No. 1, Jan. 2003, pp. 17-22

[Clark88] D.Clark, The design philosophy of the DARPA internet protocols, ACM SIGCOMM Computer Communication Review, Vol.18, No.4, pp.106-114, August 1988.

[Conti04] M. Conti, G. Maselli, G. Turi, and S. Giordano, “Cross-Layering in Mobile Ad Hoc Network Design,” IEEE Computer, special issue on Ad Hoc Networks, February 2004.

[Feldman07] A. Feldman “Internet Clean-Slate Design: What and Why?”ACM SIGCOMM Computer Communications Review, Vol. 37, No. 3, July 2007, pp. 59-64

[FIArch11] EC FIArch Group, “Fundamental Limitations of current Internet and the path to Future Internet,” March 2011,

[Goldsmith02] A.J. Goldsmith, S.B. Wicker, “Design challenges for energy-constrained ad hoc wireless networks,” IEEE Wireless Communications Magazine, vol. 9: 4, 2002, 8 – 27

[Haapola05] J. Haapola, Z. Shelby, C. Pomalaza-Ráez, P. Mähönen, “Cross-layer energy analysis of multi-hop wireless sensor networks,” in: 2nd European Workshop on Wireless Sensor Networks (EWSN), Istanbul, Turkey, 2005.

[Hakala06] I. Hakala, M. Tikkakoski, “From vertical to horizontal architecture – a cross-layer implementation in a sensor network node,” InterSense ’06 Proc. Of 1st International Conference on Integrated Internet Adhoc and Sensor Networks, Nice France, May 2006

[Hilborn04] R. Hilborn, “Sea gulls, butterflies, and grasshoppers: A brief history of the butterfly effect in nonlinear dynamics". American Journal of Physics 72 (4): 425–427. Bibcode 2004, doi:10.1119/1.1636492.

[ISO-27002] International Organization for Standardization (ISO), “Information technology. Code of practice for information security management,” ISO 27002,2005 (replaced ISO-17799)

[IEEE-610] IEEE Std 610.12.1990 - IEEE Standard Glossary of Software Engineering Terminology

[Jacobson09] Jacobson V, Smetters D., Thornton J., Plass M., Briggs N., Braynard R., “Networking Named Content,” Proceeding of ACM CoNEXT 2009. Rome, Italy, December 2009

[Johnsson03] K.B. Johnsson, D.C. Cox, “An adaptive cross-layer scheduler for improved QoS support of mixed data traffic on wireless data systems,” in: Vehicular Technology Conference, 6-9 October 2003, 1618 – 1622

[Kalogiros11] C. Kalogiros, C. Courcoubetis, G.D. Stamoulis, A SESERV methodology for tussle analysis in Future Internet technologies, White Paper, Sept. 2011

[Kempf04] J.Kempf, Ed., R.Austein, Ed., The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture, IETF, RFC 3724, March 2004.

[KISS] "Keep it Simple Stupid". The Jargon File, version 4.4.7.

[Prasad08] R. Prasad “A Perspective of Layerless Communications,” Wireless Personal Communications 2008, pp 95-100

[RFC 793] "Transmission Control Protocol" (1981)

[RFC 1122] "Requirements for Internet Hosts -- Communication Layers" (1989).

[RFC1631] K. Egevang, P. Francis, “The IP Network Address Translator (NAT),” IETF RFC 1631

[RFC1925] R.Callon, “The Twelve Networking Truths,” IETF, RFC 1925, April 1996.December 2002

[RFC1958] B. Carpenter, “Architectural Principles of the Internet,” June 1996

[RFC2775] B. Carpenter, “Internet Transparency”, IETF, RFC 2775, February 2000.

[RFC3234] B. Carpenter, “Middleboxes: Taxonomy and Issues,” IETF, RFC 3234, February 2002

[RFC3439] R. Bush, D. Meyer, “Internet Architectural Guidelines,” IETF, RFC 3439 (updates RFC 1958), December 2002

[Saltzer84] J.H. Saltzer, D.P. Reed, and D.D. Clark, “End-To-End Arguments in System Design", ACM Transactions on Computer SystemsTOCS, Vol. 2, Number 4, November 1984, pp 277-288.

[WGIG04] “The End-End Principle and the Definition of Internet”, Working Group on Internet Governance (WGIG) Contribution of Corporation for National Research Initiatives. Prepared by: Patrice A. Lyons (November 10, 2004)

[Willinger02] Walter Willinger and John Doyle, “Robustness and the Internet: Design and evolution", 2002

[Zahariadis11] Th. Zahariadis, D. Papadimitriou, H. Tschofenig, S. Haller, P. Daras, G. Stamoulis, M. Hauswirth, "Towards a Future Internet Architecture," Chapter of the book, “Future Internet Assembly," LNCS 6656, Springer, Copyright 2011, pp. 7–18, 2011

[Dening05] Peter J. Denning, The locality principle, *Communication of the ACM*, Vol. 48, No. 7, July 2005, pp. 19-24

[Doyle02] J.C.Doyle et. al. Robustness and the Internet: Theoretical Foundations, Work in Progress, 2002.

[Einstein48] A.Einstein, Quanten-Mechanik und Wirklichkeit, Dialectica vol.2, pp.320-324, 1948.

[Fogel66] L.J.Fogel, A.J.Owens, and M.J.Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.

[Heisenberg27] W.Heisenberg, Über den anschulichen Inhalt der quantentheoretischen Kinematik und Mechanik, Zeitschrift für Physik, vol..43, no.3-4, pp.172-198, 1927.

[Saltzer84] J.H.Saltzer,, D.P.Reed, D.D.Clark, End-To-End Arguments in System Design, ACM Transactions on Computer Systems (TOCS), vol 2, no.4, November 1984, pp 277-288.

[Stevens74] W.Stevens, G.Myers, L.Constantine, Structured Design, IBM Systems Journal, vol.13, no.2, pp.115-139, 1974.

[WGIG 04] "The End-End Principle and the Definition of Internet" Preparatory Process: Working Group on Internet Governance (WGIG), Contribution of Corporation for National Research Initiatives, Prepared by: Patrice A. Lyons (November 10, 2004)

8 REFERENCED PROJECTS

[COAST] http://www.fp7-coast.eu/

[COMET] http://www.comet-project.org/

[EIFFEL] http://www.fp7-eiffel.eu/

[EULER] http://www.euler-project.eu/

[nextMedia] http://www.fi-nextmedia.eu/

[OPTIMIX] http://www.ict-optimix.eu/

[SelfNet] https://www.ict-selfnet.eu/

[SESERV] http://www.seserv.org/

[UniverSelf] www.univerself-project.eu

9 LIST OF EDITORS

Papadimitriou Dimitri	Alcatel Lucent
Zahariadis Theodore	Synelixis Solutions

10 LIST OF CONTRIBUTIONS

Alonso Ignacio González	University of Oviedo, Spain
Alvarez Federico	UPM, Spain
De Leon Miguel Ponce	TSSG, Ireland
George Stamoulis	AUEB, Greece
Gomez-Skarmeta Antonio	University of Murcia, Spain
Grasa Eduard	i2CAT Foundation, Spain
Haring Günter	University of Vienna, Austria
Howker Keith	TSSG, Ireland
Kalogiros Costas	AUEB, Greece
Kantola Raimo	Aalto University, Finland
Kostopoulou Eleni	Synelixis Solutions, Greece
Krummenacher Reto	STI2, Austria
Kyriazis Dimosthenis	NTUA, Greece
Luciano Baresi	PoliMi, Italy
Manfred Hauswirth	DERI, Ireland
Markus Fiedler	Blekinge Institute of Technology, Sweden
Martinez-Julia Pedro	University of Murcia, Spain
Melideo Matteo	Engineering, Italy
Missiri Ioanna	Synelixis Solutions, Greece
Morreale Vitto	Engineering, Italy
Müller Paul	TU Kaiserslautern, Germany
Papadimitriou Dimitri (Editor)	Alcatel Lucent, Belgium
Papafili Ioanna	AUEB, Greece
Phelan Patrick	TSSG, Ireland
Pistore M.	SAYSERVICE, Italy

Soursos Sergios	Intracom Telecom, Greece
Stiller Burkhard	University of Zurich, Switzerland
Trouva Eleni	i2CAT Foundation, Spain
Tutschku Kurt	University of Vienna, Austria
Wainwright Nick	HP Labs, UK
Wajda Krzysztof	AGH University of Science and Technology, Poland
Zahariadis Theodore (Editor)	Synelixis Solutions, Greece

EC Commission officials as caretakers of the FIArchitecture Group (alphabetical order)

Name of the official	Directorate Information Society and Media (DG INFSO) - Unit
BABOT Jacques	DG INFSO – F4
DE SOUSA Paulo	DG INFSO – D1
FRIESS Peter	DG INFSO – D4
LASO BALLESTEROS Isidro (coordinator)	DG INFSO – D2
SCILLIA Mario	DG INFSO – F5
SASSEN Anne-marie	DG INFSO – D3