Design Principles for the Future Internet Architecture

Dimitri Papadimitriou[1], Theodore Zahariadis[2], Pedro Martinez-Julia[3], Ioanna Papafili[4], Vito Morreale[5], Francesco Torelli[5], Bernard Sales[1], and Piet Demeester[6]

[1] Alcatel-Lucent Bell Labs, Belgium
{dimitri.papadimitriou,bernard.sales}@alcatel-lucent.com
[2] Synelixis Solutions, Greece
zahariad@synelixis.com
[3] Univ. of Murcia, Spain
pedromj@um.es
[4] Athens University of Economics and Business (AUEB)
iopapafi@aueb.gr
[5] Engineering, Italy
{vito.morreale,francesco.torelli}@eng.it
[6] Ghent University (UGent), Belgium
piet.demeester@intec.ugent.be

Abstract. Design principles play a central role in the architecture of the Internet as driving most engineering decisions at conception level and operational level. This paper is based on the EC Future Internet Architecture (FIArch) Group results and identifies some of the design principles that we expect to govern the future architecture of the Internet. We believe that it may serve as a starting point and comparison for most research and development projects that target the so-called Future Internet Architecture.

1 Introduction

Design principles play a central role in the architecture of the Internet as driving most engineering decisions not only at conception level but also at operational level. Many ICT systems do not consider design principles and derive their model directly from requirements. However, when it comes to the design of the Internet, the formulation of design principles is a fundamental characteristic of the process that guides the design of its protocols. On the other hand, in searching for Internet architectural principles, we must remember that technical change is continuous in the information and communication technology industry. Indeed, as stated in RFC 1958 [1], "*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.

This paper, based on the work accomplished within the EC Future Internet Architecture (FIArch) group [2], identifies some of the design principles that we expect to govern the future architecture of the Internet. It may serve as a starting point

F. Álvarez et al. (Eds.): FIA 2012, LNCS 7281, pp. 55–67, 2012.

and comparison basis for all research and development projects that target the so-called Future Internet Architecture. This paper is structured as follows: Section 2 contains the definitions used in our analysis, and gives the needed background and our understanding of the current design principles of the Internet. Section 3 summarizes the Design Principles that we expect to remain or evolve towards the Future Internet and Section 4 gives some seeds of new design principles.

2 Definition and Background

2.1 Definitions

We define *architecture* the set of functions, states, and objects/information together with their behavior, structure, composition, relationships and spatio-temporal distribution. 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.).

Design principles refer to agreed structural and behavioral rules on how a designer/an architect can best structure the various architectural components and describe the fundamental and time invariant laws underlying an engineered artefact (i.e., an object formed/produced by engineering). By "*structural and behavioral 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 behavior 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.).

We use the term *data* to refer to any organized group of bits, e.g., packets, traffic, information, etc. and *service* to refer to any action or set of actions performed by a provider in fulfillment of a request, which occurs through the Internet (i.e., by exploiting data communication, as defined below) with the aim of creating and/or providing added value or benefits to the requester(s). "*Resource*" is any fundamental element (i.e., physical, logical or abstract) that can be identified.

This paper refers to *communication* as the exchange of data (including both control messages and data) between a physical or logical source and sink referred to as *communication end-points*; when end-points sit at the same physical or logical functional level, communication is qualified as "end-to-end".

Security is a process of taking into account all major constraints that encompasses *robustness*, *confidentiality* and *integrity*. *Robustness* is the degree to which a system operates correctly in the presence of exceptional inputs or stressful environmental conditions. *Confidentiality* is the property that ensures that information is accessible only to those authorized to have access and *integrity* includes both "*data integrity*" and "*system integrity*". The term *complexity* refers to the *architectural complexity* (i.e., proportional to the needed number of components and interactions among components), and *communication complexity* (i.e., proportional to the needed number of messages for proper operation). Finally, *scalability* refers to the ability of a computational system 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, volume or rate.

2.2 Existing Design Principles

The Internet architecture is considered as progressively moving from the pure network connectivity functionality to a networking ecosystem, which integrates the network connectivity with the services combining network, computing, and storage. Yet, beforehand, it is mandatory to establish a common understanding of the main design principles that governs the Internet today and are mainly focused at the connectivity functionality.

- *Heterogeneity support principle* [1]: heterogeneity is inevitable and must be supported by design. This applies at many levels: devices and nodes, scheduling algorithms and queue management mechanisms, routing protocols, levels of multiplexing, protocol versions and implementations, underlying data link layers (e.g., Ethernet, WiFi, etc.) and physical layers (e.g., SDH, OTN, wireless/wireline access link), in the traffic mix and in the levels of congestion at different times and places.
- *Scalability and amplification principle*: Scalability [1] states that "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. The amplification principle [3] states that "there do exist non-linearities which do not occur at small to medium scale, but occur at large scale". As a result complexity can amplify small perturbations, and designers must ensure such perturbations are extremely rare.
- *Robustness and adaptability principle*: the robustness principle [4] also known as the Postel Law, warns that each protocol implementation must interoperate with others as 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*". This principle aims at maximizing interoperability between protocol implementations, particularly in the face of ambiguous or incomplete specifications. Moreover, adaptability [5] advises that "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..."; as result, protocols would improve their robustness.
- *Modularization/Layering principle*: decomposes the communication functionality into different modules with well-defined interfaces. Each of these modules corresponds to a functional assignment which offers various behavioral and structural advantages, such as complexity reduction, isolation, and reusability of modules. On the other hand, modularization hinders the overall system optimization, as each module/layer has to be optimized separately.
- *Unambiguous addressing principle*: following [1], the upper layer Internet protocols must be able to identify end-points unambiguously and must be independent of the hardware medium and hardware addressing to allow exploiting any new digital transmission technology, 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.

- *Loose Coupling principle*: Coupling is the degree to which each architectural module relies on each one of the other modules [6]. Loose coupling defines a method for interconnecting system components so that they depend on each other to the least extent practicable. 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. In today's Internet design, "Modularity is good. If you can keep things separate do so" [1]. The best example of loose coupling in the communication stack is the decoupling between applicative layers and the TCP/IP protocol. The loose coupling principle is further refined in [3] by stating that as things get larger, they often exhibit increased interdependence between components. Much of the non-linearity observed in large systems is largely due to coupling of horizontal and/or vertical components. Loose coupling minimizes unwanted interaction among system elements but can also give rise to difficulty in maintaining synchronization among diverse components when such interaction is desired.
- *Locality Principle*: in computer science, this principle guiding the design of robust replacement algorithms, compiler code generators, and thrashing-proof systems, is useful wherever there is an advantage in reducing the apparent distance from a process to the information or data it accesses. It has been used in virtual memory systems, processor caches, disk controller caches, storage hierarchies, network interfaces, etc. We distinguish the principle of *temporal locality* (recently accessed data and instructions are likely to be accessed in the near future) from the *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.
- *The "end-to-end" and minimum intervention principle*: End-to-end is one of the fundamental principle on which the Internet has been structured and built, as it guides the functional placement and the spatial distribution of functions across the layers of the communication stack [7]. Following this principle, a function 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) while the core of the network should provide a general connectivity service. The end-to-end principle has also important consequences in terms of protocol design that should not rely on the maintenance inside the network of state information. The application of this principle, together with the minimum intervention (i.e., where possible, payload should be transported as received without modification), results in a network that is transparent to the host application communication and provides for a general, application agnostic transport service.
- *Simplicity principle*: this common sense engineering principle also expressed as the KISS ("Keep it Simple, ... Stupid") or the "Occam's Razor" principle, states when facing doubts or multiple choices or ways in the design of, e.g., protocols and intermediate systems, choose the simplest solution [1]. Adding functionality or improving performance should not come at the detriment of increasing complexity.

- *Connectionless packet switching and distributed adaptive routing*: provides unreliable, best-effort, connectionless packet delivery. The service is "connectionless" since packets can be delivered without any prior end-to-end connection setup phase. Forwarding decision is taken per-packet, independently at each node: upon receiving packets, nodes lookup their routing tables to determine the outgoing interface for that packet. The routing mechanism is called "proactive routing" since all routing entries in the routing table must be setup before packet delivery. 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 and best-effort since delivery is not guaranteed. 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 to ensure scalability and cost effectiveness of the communication system and its entities.
- *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. 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). This design of the routing system ensures survivability and allows for distributed management as long as ISPs are (at least partially) collaborative.

3 Evolution of Existing Design Principles

3.1 Principles That Should Be Preserved

In this section, we detail the design principles that should be preserved and applied to the future architecture of the Internet. Other should be adapted or augmented.

- *Heterogeneity support principle*: In the future, the heterogeneity is expected to be much higher than today. Multiple types of terminals/hosts, network nodes, protocols, and applications will co-exist. Hence, the capability to support heterogeneity should remain (and even enforced).
- *Scalability and amplification principle*: the number of devices with Internet access (e.g., computers, mobile devices), communication nodes (e.g., home, access, edge and core routers), autonomous systems, and 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. As a result, scalability is among the design principles that should govern Future Internet, and the amplification principle would definitely remain.
- *Robustness principle*: the Internet is expected to increasingly handle mission and time critical applications, related to, e.g., health, energy, and transport. As a result, for what concerns the minimization of malfunction, uninterrupted operation and

interoperability, the robustness principle remains unchanged. Yet, as explained in Section 3.2, this principle should be extended to cover security issues.

- *Loose coupling principle*: defines a necessary condition for a well-structured and well-designed system as i) it simplifies testing and troubleshooting 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 components. In addition, tightly coupled systems are likely to experience 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 principle shall be preserved and even reinforced as a result of the increasing importance of the availability objective. Nevertheless, loose coupling may also increase difficulty in maintaining synchronization among system components 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.
- *Locality principle*: 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 locality principle should 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.

3.2 Principles That Should Be Adapted (Modification of Existing Description)

In this section we highlight design principles that apply to the current Internet architecture but should be adapted to address the design objectives of the Internet [11].

- *Simplicity principle*: Complex systems are generally less reliable and flexible. Architectural complexity dictates that in order to increase the reliability it is mandatory to minimize the number of components in a service delivery path (being a protocol, a software, or a physical path). However, this principle has already been challenged as 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 complexity 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 A.Einstein 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 3.1, scalability and simplicity should be handled as strongly interconnected first priority design principles.

- *Minimum Intervention principle*: is critical to maintain and preserve data integrity and to avoid useless intermediate information message or packet processing. However, in some cases, it may conflict with the simplicity principle; e.g., in sensor networks where communication gateways and actuators 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.
- *Robustness principle*: in order to increase robustness and system reliability, some have advocated transforming 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, adopting this approach would result in dropping a significant level of interoperability between protocol implementations. 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. 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 however leads to relatively weak security. As stated in [1]: “*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*”. 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. Moreover, even if individual subsystems can be simple, the overall system resulting from 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".
- *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. After 30 years of evolution, communication stacks are characterized nowadays by i) the repetition of functionality across multiple layers, such as monitoring modules repeated over multiple layers and security components

each associated to a specific protocol sitting at a given layer (which result into inconsistent response to attacks), which emphasizes the need to define common functional modules; ii) the proliferation of protocol variants (as part of the same layer) all derived from a kernel of common functions/primitives; which emphasizes the need to define generic modules; iii) the limited or even absence of capability for communication stacks to cope with the increasing variability and uncertainty characterizing external events (resulting from increasing heterogeneity where communication systems proliferate); this observation emphasizes that the functional and even performance objectives to be met by communication systems could vary over time (thus messages would be processed by variable sequence of functions determined at running time); iv) the inability to operate under increasingly variable running conditions resulting from the increasing heterogeneity of substrate on top of which communications stacks are actually performing. These observations lead to reformulate the modularization principle so as to i) consider functional modules connected by realization relationships that supply their behavioral specification, ii) distinguish between general and specialized modules, and iii) enable dynamic and variable binding between the different modules such that the sequence of functions performed is specified at running time. In turn, the application of the adapted principle allows designing systems with a larger autonomy in diagnosing internal/external stimuli but also in their decision and execution.

3.3 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.

- *Polymorphism principle* (as extension to the modularity principle): in computer science/programming, polymorphism applies to data or functions. 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, 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. Introducing polymorphism would enable the same abstract and autonomous loosely coupled components to benefit from different

functional and/or non-functional behavior under different environments or circumstances. The question remains open though as how to parameterize these environmental variables and whether this could be efficiently performed through distant exchanges (remotely).

- *Unambiguous naming and addressing principle*: in order to cope with the evolution of the use of name and address spaces, the following augmentations are considered (using [1] as starting point): i) avoid any design that requires addresses to be hard coded or stored on non-volatile storage (when an address is an essential requirement a discovery process is recommended); ii) A single and common naming structure should be used. iii) Locators (LOC) and Identifiers (ID) should be separated. 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 also the data and the services. Moreover, the current ID/LOC approach only deals 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.
- *Extending the end-to-end principle*: many experts insist that the "end-to-end" principle is still valid, even though middle boxes and application layer gateways are deployed at the edges of networks, as communication is divided at autonomous legs. Another challenge concerning this principle is that IP overlay applications such as IP multicast and mobile IP (MIP), require support from intermediate nodes (e.g., 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, 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 [8]. Finally, 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 [9]. 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. As a result, we believe that motivations to "update" or augment this principle increase; however even if this principle is challenged, it remains due to heavy consequence in terms of scalability, survivability and robustness at large departing from this principle.

4 Seeds for New Design Principles

The Internet will evolve from a connectivity inter-network to a service ecosystem, able to offer resources of any type (e.g., any type of network, computation, storage

and content). Realizing such Internet Architecture requires 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).

4.1 Resources Awareness

Taking into consideration that resources (associated to service components) refer to different types (e.g., data, infrastructure resources) including resources as first order abstraction in the Internet architecture would facilitate situation awareness. While current service offerings are based on resource-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 and quality in general. In turn, positioning as first order abstraction processing, storage, and transmission resources (or their combination) as well as data provides the required mechanism for the adoption of richer service-oriented models that extend current models adapted/tailored only for higher-order logic level(s). However, as it is also necessary for the service components network as a whole to manage these abstractions, implies providing means to influence their behavior.

Addressing the aforementioned challenges requires (establishing design principles that support) the definition of suitable abstractions and mechanisms for allowing the cooperation across all resource abstraction levels (e.g., for monitoring, negotiation). This principle is strongly related to the "modularization principle", and should complement it by specifying the functional details each module exposes for supporting crossed cooperation. Furthermore, applying this principle in combination with the "loose coupling" principle, will allow for evaluating the effects of cross-module awareness and cooperation, in order to avoid or minimize unwanted interactions and non-linear effects. Another principle that needs to be considered is the "locality" principle to which resource awareness will contribute by allowing the development of service-delivery models enabled through self-management and cross-module cooperation approaches.

4.2 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. Indeed, with the current design of the Internet:

i) Services are not cognizant of end-user expectations and needs, especially for mission critical applications. Services are often static, lack of flexibility and they are not negotiable. Often it is left up to the users/clients to implement their own systems to ensure the service performs as expected
ii) Services operate on a "best-effort" basis. Moreover, services are often not accountable towards the end-user;
iii) Services are modeled prior to their deployment in any environment and according to the aforementioned modeling 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, performance, etc.; and
iv) Distributed dynamic 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.

The design of the Future Internet must be imbued with the principle of dependability (reliability–accountability–verifiability feedback loop) including self-adaptation and self-learning capability to cope and learn from changes in the operating conditions. However, enabling such capability 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.

4.3 Allow Exchange of Information between End-Points of Different Type

The Internet has evolved to a playground for different stakeholders such as Internet Service Providers (ISPs), Content Distribution Network (CDN) providers, end-users, etc. and each stakeholder tries to optimize its 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. The so-called information asymmetry between different stakeholders leads often the ecosystem to a suboptimal performance. Addressing the information asymmetry problem may allow stakeholders to make alternative decisions that would lead them collectively to a more beneficial state. Furthermore, the emerging *Design for Choice principle* seed suggests that Internet technologies should be designed so that they allow variation in outcome, rather than imposing a particular outcome [10]. The rationale behind is that the Internet is a rather unpredictable system and it is very difficult to assess if a particular outcome will remain desirable in the future. 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: i) the exposure of information to a stakeholder, ii) the abstraction/aggregation of information to be exchanged, iii) the collection of information by a stakeholder, iv) the assessment of information by a stakeholder, and iv) the decision making.

4.4 Sustain the Resources and Brain Investment

“Coopetition” refers to 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" [10]. Instead, Internet could be designed so as to lead to a global positive return, the so-called “all-win” situation for the society at large.

Moreover, 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.

5 Conclusion

New functionality as well as performance expectation from the 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 together with system upgrades will not suffice anymore due to new qualitative and quantitative requirements, resulting from, e.g., highly critical services such as e-health applications, clouds of services and clouds of sensors, new social network applications like collaborative immersive environments, new commercial and transactional applications, new location-based services as well as the natural expansion and growth of the Internet. Hence, a deeper architectural evolution is required at the behavioral and structural level to sustain these new demands that are confronted to the objective limits of the current Internet.

As design principles have played and will play a central role in the architecture of the Internet as driving most of its engineering decisions at the conception level but also the operational level, this document investigates their potential evolution (adaptation and/or augmentation which arguably cover already a significant part of their evolution). Acknowledging that new principles are emerging, this document also explores a non-exhaustive set of new "seeds" translating current architecture research work being realized. Altogether, the result of this investigation by the FIArch group has lead to the identification of the design principles that will expectedly govern the architecture of the Future Internet if corroborated by further proofs and experimental

evidences. Consequently, we believe that this work may serve as a starting point and comparison basis for many research and development projects that target the Future Internet Architecture. The result of these projects would in turn enable to refine the formulation of these principles that will govern the design of the foundation of a common architecture.

Acknowledgment. The authors acknowledge the FIArch group as this paper is a summary of the design principles document, which is available at [2]. Part of this work has been funded under the EC projects FP7 ICT COAST, EULER, REVERIE, SESERV and SOFI.

References

1. Carpenter, B.: Architectural Principles of the Internet. RFC 1958 (June 1996)
2. European Commission, Future Internet Reference Architecture Group, http://ec.europa.eu/information_society/activities/foi/docs/fiarch designprinciples-v1.pdf
3. Bush, R., Meyer, D.: Internet Architectural Guidelines. IETF, RFC 3439 (updates RFC 1958) (December 2002)
4. RFC793, Transmission Control Protocol (1981)
5. Braden, R.: Requirements for Internet Hosts-Communication Layers. RFC1122 (October 1989)
6. Stevens, W., Myers, G., Constantine, L.: Structured Design. IBM Systems Journal 13(2), 115–139 (1974)
7. Saltzer, J.H., Reed, D.P., Clark, D.D.: End-To-End Arguments in System Design. ACM Transactions on Computer Systems 2(4), 277–288 (1984)
8. Lyons, P.A.: The End-End Principle and the Definition of Internet. Working Group on Internet Governance (WGIG) (November 10, 2004)
9. Papadimitriou, D., Welzl, M., Scharf, M., Briscoe, B.: Open Research Issues in Internet Congestion Control. RFC6077 (February 2011)
10. Clark, D.D., Wroclawski, J., Sollins, K.R., Braden, R.: Tussle in Cyberspace: Defining Tomorrow's Internet. IEEE/ ACM Trans. Networking 13(3), 462–475 (2005)
11. Zahariadis, T., et al.: Towards a Future Internet Architecture. In: Domingue, J., et al. (eds.) FIA 2011. LNCS, vol. 6656, pp. 7–18. Springer, Heidelberg (2011)