Towards the Future Internet
G. Tselentis et al. (Eds.)
IOS Press, 2009
© 2009 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-60750-007-0-112

Management Architecture and Systems for Future Internet Networks

A.Galis[1], S. Denazis[2], A. Bassi[2], P. Giacomin[2], A. Berl[3], A. Fischer[3], H. de Meer[3], J. Srassner[4], S. Davy[4], D. Macedo[5], G. Pujolle[5], J. R. Loyola[6], J Serrat[6], L. Lefevre[7], A. Cheniour[7]

[1] University College London U.K. {a.galis@ee.ucl.ac.uk},
[2] Hitachi France {spyros.denazis, alessandro.bassi}@hitachi-eu.com, {yrz@anche.no}
[3] University of Passau Germany {andreas.berl, andreas.fischer, demeer}@uni-passau.de
[4] Waterford Institute of Technology Ireland {jstrassner, sdavy}@tssg.org
[5] Lip6 France {daniel.macedo@rp.lip6.fr, Guy.Pujolle@lip6.fr}
[6] UPC Spain {jrloyola,j.serrat}@tsc.upc.edu
[7]INRIA France {laurent.lefevre, abderhaman.cheniour}@ens-lyon.fr

Abstract — This paper presents a new autonomic management architectural model consisting of a number of distributed management systems running within the network, which are described with the help of five abstractions and distributed systems: Virtualisation, Management, Knowledge, Service Enablers and Orchestration Planes. The envisaged solution is applicable to the management design of Future Internet as a service and self-aware network, which guarantees built-in orchestrated reliability, robustness, context, access, security, service support and self-management of the communication resources and services.

Keywords — Service and Self-aware Network Management, Autonomicity, Virtualisation, Management Plane, Knowledge Plane, Service Enablers Plane, Orchestration Plane.

1. INTRODUCTION AND FRAMEWORK

Networks are becoming service-aware. Service awareness means not only that all content and service logic pertaining to a service are delivered but also that all business or other service characteristics (e.g. QoS, SLAs) offer are fulfilled and the network resources are optimally used in the service delivery. In addition, the network's design is moving towards a different level of automation, autonomicity and self-management. The envisaged solution for Future Internet is a service and self-aware network, which guarantees built-in orchestrated reliability, robustness, mobility, context, access, security, service support and self-management of the communication resources and services. We suggest a transition from a service agnostic Internet to service- and self-aware Internet managing resources by applying Autonomic principles as depicted in Figure 1. In order to achieve the objective of service-aware and self-aware networking resources and to overcome the ossification of the current Internet, the Autonomic Internet (AutoI) project [1][2][3][4][17] aims to develop a self-managing virtual resources that can span across heterogeneous networks and that supports service mobility, security, quality of service and reliability. In this overlay network, multiple virtual networks could co-exist on top of a shared substrate with uniform control. One of the main research challenges for designing a new service-aware network is the inclusion of capabilities such as self-awareness, self-network knowledge and self-service knowledge. These capabilities are used to facilitate continuous tuning of the networks, adaptation to unpredictable conditions, prevention and recovery from failures and provision of a dependable environment.

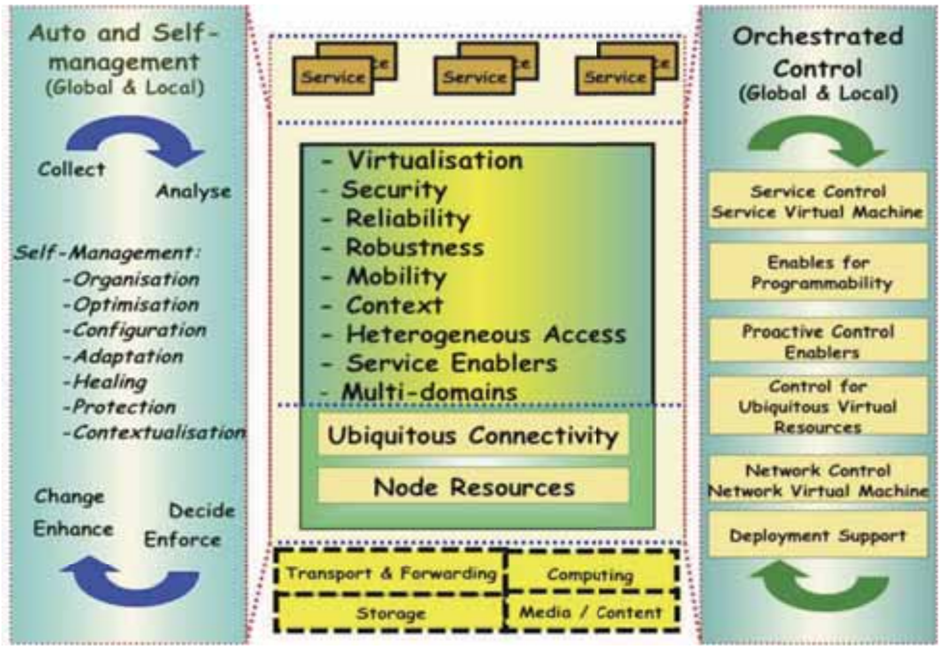

Fig. 1 **- Autonomic Internet**

To meet the above needs and to support the design of the self-managing virtual resources overlay that can span across heterogeneous networks where multiple virtual networks co-exist on top of a shared substrate with uniform control we proposed a new autonomic management architectural model. It consists of a number of distributed management systems within the network, which are described with the help of five abstractions and distributed systems - the OSKMV planes: Orchestration Plane (OP), Service Enablers Plane (SP), Knowledge Plane (KP), Management Plane (MP) and Virtualisation Plane (VP). Together these distributed systems form a software-driven control network infrastructure that will run on top of all current networks and service infrastructures.

Figure 2 depicts the network and management resources as distributed in the OSKMV planes. The OSKMV planes are new higher-level artefacts to make the Future Internet more intelligent with embedded management functionality, including self-knowledgeable, self-diagnosing and ultimately fully self-managing. At one level the OSKMV planes gather observations, constraints and assertions, and apply rules to these to generate service-aware observations and responses. They are embedded on network hosts, devices, attachments and servers.

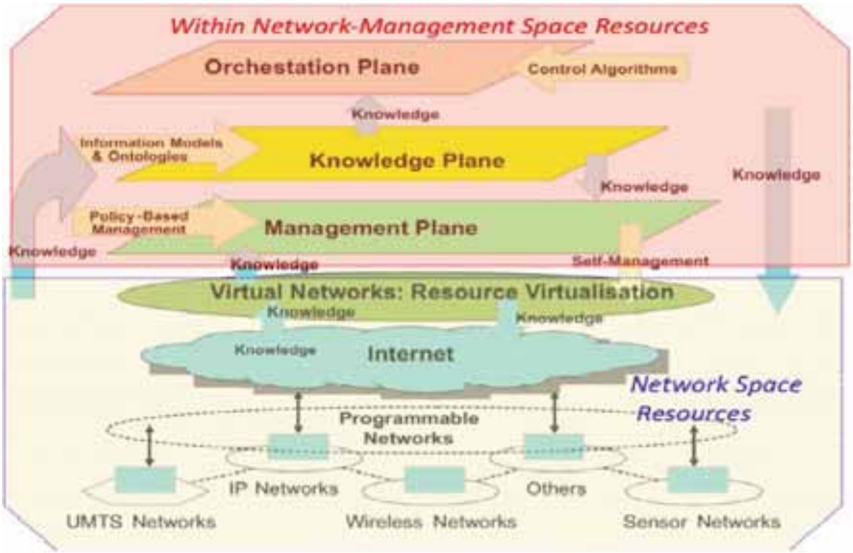

Fig. 2 **- AutoI Planes**

1.1 Architectural Model Overview

AutoI framework, as presented in the figure 3, consists of a number of distributed management systems described with the help of the OSKMV planes. Together these distributed systems form a software-driven network control infrastructure that will run on top of all current networks and application service physical infrastructures to provide an autonomic virtual resource overlay.

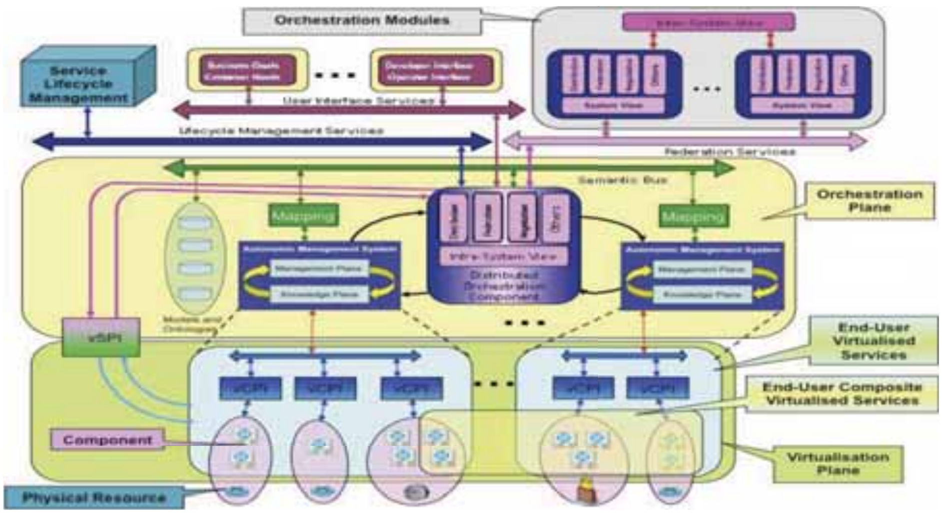

Fig. 3 **– AutoI Framework**

1.2 Orchestration Plane

The purpose of the Orchestration Plane is to govern and integrate the behaviours of the system in response to changing context [14] and in accordance with applicable business goals and policies. It supervises and it integrates all other planes' behaviour insuring integrity of the Future Internet management operations. The Orchestration Plane can be thought of as a control framework into which any number of components can be plugged into in order to achieve the required functionality. These components could have direct interworking with control algorithms, situated in the control plane of the Internet (i.e. to provide real time reaction), and interworking with other management functions (i.e. to provide near real time reaction).

The Orchestration Plane would supervise the optimisation and the distribution of knowledge within the Knowledge Plane to ensure that the required knowledge is available in the proper place at the proper time. This implies that the Orchestration Plane may use either very local knowledge to deserve a real time control as well as a more global knowledge to manage some long-term processes like planning. The Orchestration Plane would host one or more Autonomic Management Systems (AMSs) and it is made up of one or more Distributed Orchestration Components (DOCs), and a dynamic knowledge base consisting of a set of models and ontologies and appropriate mapping logic. Each AMS represents an administrative and/or organisational boundary that is responsible for managing a set of devices, subnetworks, or networks using a common set of policies and knowledge. The AMSs access a dynamically updateable knowledge base, which consists of a set of models and ontologies. A set of DOCs enable AMSs to communicate with each other. The Federation bus enable the Orchestration Plane to be composed with other Orchestration Planes.

The Orchestration Plane acts as control workflow for all AMSs ensuring bootstrapping, initialisation, dynamic reconfiguration, adaptation and contextualisation, optimisation, organisation, closing down of AMSs. The Orchestration Plane provides assistance for the Service Lifecycle Management, namely during the actual creation, deployment, activation, modification and in general, any operation related to the application services or management services.

Autonomic Management System

In the current Internet, the data, control, and management, planes are bound together and often use the same links. For example, TCP connection setup control messages and SNMP management messages use the same links as the data messages. This has at least three drawbacks: i. the data plane is limited to packet-oriented data; ii. the design of each of the three planes is unduly complicated, and iii. inherent security risk exist, since it is relatively easy to get at control and management data by simply attacking the data plane path. A key advantage of the AutoI architecture is that it can provide a programmable mix of isolation and sharing of network resources. Each AMS consists of a management plane and a knowledge plane, as well as interfaces to a dedicated set of models and ontologies and interfaces to one or more Distributed Orchestration Components. Mapping logic enables the data stored in models to be transformed into knowledge and combined with knowledge stored in ontologies to provide a context-sensitive assessment of the operation of one or more virtual resources. Another set of interfaces enables framework services, such as directory services, naming, federation, and others, to be used by the AMS.

An AMS collects appropriate monitoring information from the virtual and non-virtual devices and services that it is managing, and makes appropriate decisions for the resources and services that it governs, either by itself or in collaboration with other AMSs, as explained in the next section. The DOC provides a set of framework network services [15]. Framework services provide a common infrastructure that enables all components in the system managed by the Orchestration Plane to have plug-and-play and unplug-and-play behaviour. Applications compliant with these framework services share common security, metadata, administration, and management services. The DOC enables the following functions across the orchestration plane:

- Federation: each AMS is responsible for its own set of virtual and non-virtual resources and services that it governs as a domain. Federation enables a set of domains to be combined into a larger domain.
- Negotiation: each AMS advertises a set of capabilities (i.e., services and/or resources) that it offers for use by other components in the Orchestration Plane.
- Distribution: the DOC provides communication and control services that enable tasks to be split into parts that run concurrently on multiple AMSs within an Orchestration Plane, or even across multiple Orchestration Planes.
- Governance: each AMS can operate in an individual, distributed, or collaborative mode. It collects appropriate monitoring data in order to determine if the virtual and non-virtual resources and services that it governs need to be reconfigured. Business goals, service requirements, context, capabilities and constraints are all considered as part of the decision making process.
- Autonomicity: AMSs acting, as individual entities are responsible for managing their resources and services, and send status messages to other AMSs
- Views: i. Intra - Future Internet System View provides an overall, composite view of the system as seen by the components within a given Orchestration Plane; ii. Inter - Future Internet System View provides an overall, composite view of Orchestration Planes that are collaborating, as in a multiple domain system.

1.3 Service Enablers Plane – Life Cycle Management

The AutoI architecture defines a common mechanism for the identification and specification of a business problem, along with the specification and development of a deployable solution, which enables services to be built and deployed. This is necessary due to the changing conditions in which a service is provisioned; hence, the service and in particular management service must be lifecycle managed as it evolves and responds to these changes. The Service Enablers Plane (SP) consists of functions for the automatic (re) deployment of new management services, protocols as well as resource – facing (i.e. QoS functions) and end-user facing service. It includes the enablers to allow code to be executed on the network entities. The safe and controlled deployment of new code enables new services to be activated on demand. This approach has the following characteristics: i. Service (re)deployment is taking place automatically and allows a significant number of new services to be offered on demand; ii. Special management functions and services can be easily enabled locally for testing purposes before they are automatically deployed network-wide; iii. Eases the deployment of network-wide protocol stacks and management services; iv. To enable secure but controlled execution environments; v. An automatic decision making infrastructure that guides the deployment of new tested network services; vi. Optimised resource utilization of the new services and the system.

1.4 Knowledge Plane

AutoI introduces a focused functionality knowledge plane, consisting of models and ontologies, to provide increased analysis and inference capabilities; its purpose is to provide knowledge and expertise to enable the network to be self-monitoring, self-analyzing, self-diagnosing, and self- maintaining or -improving. AutoI's KP brings together widely distributed data collection, wide availability of that data, and sophisticated and adaptive processing or KP functions, within a unifying structure that brings order, meets the policy [8], scaling and functional requirements of a global network, and, ideally, creates synergy and exploits commonality of design patterns between the many possible KP functions. The main KP components are an information and context service plus models and ontologies, which enable the analysis and inferencing capabilities. Knowledge extracted from information/data models forms facts. Knowledge extracted from ontologies is used to augment the facts, so that they can be reasoned about. The information and context service provides: i. information-life cycle management (storage, aggregation, transformations, updates, distribution) all information and context in the network and addresses the size and scope of the Internet; ii. responsiveness to requests made by the AMSs; iii. triggers for the purpose of contextualisation of AMSs (supported by the context model of the information model); iv. support for robustness enabling the KP to continue to function as best possible, even under incorrect or incomplete behaviour of the network itself; v. support of virtual networks and virtual system resources in their needs for privacy and other forms of local control, while enabling them to cooperate for mutual benefit in more effective network management.

1.5 Management Plane

The Management Plane consists of AMSs, which are designed to meet the following design objectives and functionality: i. Embedded (Inside) Network functions: The majority of management functionality should be embedded in the network and it is abstracted from the human activities. As such the AMSs will run on execution environments on top of virtual networks and systems, which run on top of all current network and service physical infrastructures.; ii. Aware and Self-aware functions: It monitors the network and operational context as well as internal operational network state in order to assess if the network current behaviour serve its service purposes.; iii. Adaptive and Self-adaptive functions: It triggers changes in network operations (state, configurations, functions) function as a result of the changes in network and service context.; iv. Automatic self-functions: It enables self-control (i.e. self-FCAPS, self-*) of its internal network operations, functions and state. It also bootstraps itself and it operates without manual external intervention. Only manual/external input is provided in the setting-up of the business goals; v. Extensibility functions: It adds new functions without disturbing the rest of the system ((Un)Plug_and_Play/ Dynamic programmability of management functions & services); vi. Simple cost functions: Minimise life-cycle network operations' costs and minimise energy footprint.

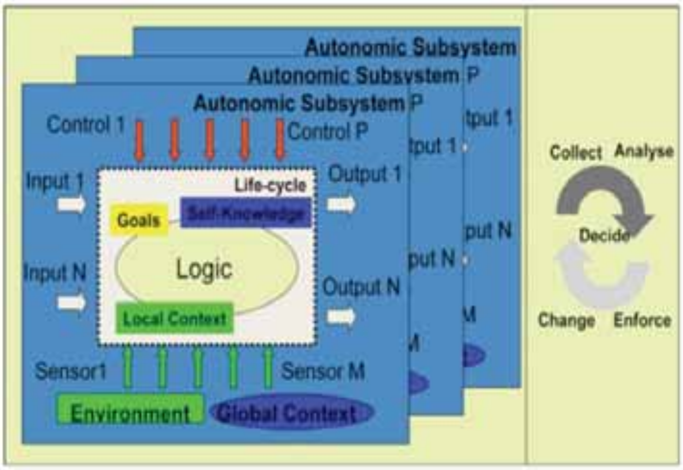

Fig. 4 **– Control Loops**

In addition the Management Plane, as it governs all virtual resources, is responsible for the optimal placement and continuous migration of virtual nodes and virtual servers into hosts (i.e. physical nodes and servers) subject to constraints determined by the Orchestration Plane. The AMSs are design to follow the autonomic control loops depicted in the figure 4.

1.6 Virtualization Plane

One of the key requirements that differentiate AutoI from other efforts is its emphasis on virtualisation (i.e., the abstraction) of resources and services, which are controlled by the MP. Virtualisation hides the physical characteristics of the computing resources being used from its applications and users. AutoI uses platform virtualisation to provide virtual services and resources. Platform virtualisation separates an operating system from its underlying platform resources; resource virtualisation abstracts physical resources into manageable units of functionality (e.g., the concept of a virtual router, where a single physical router can support multiple independent routing processes by assigning different internal resources to each routing process).

The virtualisation plane consists of software mechanisms to treat selected physical resources as a programmable pool of virtual resources that can be organised by the Orchestration Plane into appropriate sets of virtual resources to form components (e.g., increased storage or memory), devices (e.g., a switch with more ports), or even networks. The organisation is done in order to realise a certain business goal or service requirement. Two special interfaces, as shown in Figure 5, called the Virtualisation System Programming Interface (vSPI) and the Virtualisation Component Programming Interface (vCPI) are under development.

The vSPI is used to enable the Orchestration Plane to govern virtual resources, and to construct virtual services and networks that meet stated business goals having specified service requirements. The vSPI contains the system-view of the virtual resources that a particular Orchestration Plane governs, and is responsible for orchestrating groups of virtual resources in response to changing user needs, business requirements, and environmental conditions.

The vSPI is responsible for determining what portion of a component (i.e., set of virtual resources) are allocated to a given task. This means that all or part of a virtual resource can be used for each task, providing an optimised partitioning of virtual resources according to business need, priority, and other requirements. Composite virtual services can thus be constructed using all or part of the virtual resources provided by each physical resource, as shown in Figure 5. The vSPI monitors system-level status of the virtual resources that it governs. This is different than the vCPI, which monitors "micro-level" status of the virtual resources that it configures. For example, the vSPI is responsible for informing the AMS that a particular virtual resource is ready for use, whereas the vCPI is responsible for informing the AMS that a particular virtual resource has been successfully reconfigured.

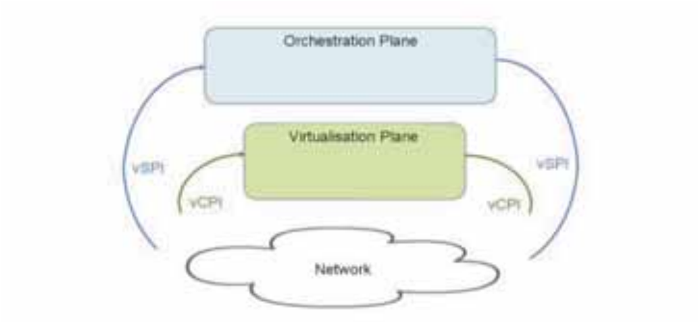

Fig. 5 **– Virtualisation Interfaces**

Each physical resource has an associated and distinct vCPI. This enables the AMS to manage the physical resource, and to request virtual resources to be constructed from that physical resource by the vCPI of the Virtualisation Plane. The AMS sends device-independent commands via the vCPI, which are translated into device-specific commands that reconfigure the physical resource. The vCPI also provides monitoring information from the virtual resources back to the AMS that controls that physical resource. It is responsible for providing dynamic management data to its governing AMS that states how many virtual resources are currently instantiated, and how many additional virtual resources of what type can be supported.

2. RELATED WORK

Autonomic Computing and Communications - A comprehensive state of the art review in Autonomic computing and communications is provided in [16]. The purpose of autonomic computing and networking [5][6] is to manage complexity. By transferring more manual functions to involuntary control, additional resources (human and otherwise) are made available to manage higher-level processes. One difference between autonomic computing and autonomic networking is that the latter must cope with and coordinate multiple control mechanisms, such as those used by different types of networks, which the former usually does not consider.

Knowledge Plane – The Knowledge Plane [9] was proposed as a research objective to build "a fundamentally different sort of network that can assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why it cannot do so.” The Knowledge Plane approach organised functionality into two planes. The data plane was responsible for packet forwarding; the Knowledge plane was a new construct “that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network.”

Foundation Observation Comparison Action Learn Reason (FOCALE) - This architecture [7] is equally appropriate for legacy devices and applications as well as for next generation and cognitive networks.

Inference Plane - This approach [10] was created to solve some of the problems with the Knowledge Plane [9]. The Inference Plane is a coordinated set of intelligent decision-making components that represent the capabilities of the computing elements being controlled, the constraints placed upon using different functions, and the context in which they are being used.

Ambient Networks - AN is an EU sponsored project (www.ambient-networks.org), which envisaged the development of a software-driven network control infrastructure for wireless and mobile networks. The concept of the Ambient Control Space (ACS) is introduced to encompass all control functions in a certain domain, which can be used as a control plane overlay to integrate and interoperate seamlessly any existing networks [11] [12].

Autonomic Network Architecture - ANA is an EU sponsored project (www.ana-project.org), which aims at exploring novel ways of organizing and using networks beyond legacy Internet technology. ANA should be regarded as an architecture for autonomic devices.

4WARD is an EU sponsored project (www. 4ward-project.eu), which aims to make the development of networks and networked applications faster and easier, including an in-network management approach.

Programmable Network Management - Programmable networks techniques allow software components or programs to be activated or injected into network components, which run customised tasks on the data passing through these components. They are especially attractive in the realisation of real-time activation, adaptations and thus, the implementation of real-time capable control loops. Full review of the state of the art in programmable networks and their application to management of networks is in [13].

FIND (Future Internet Design) is a new long-term research program of the National Science Foundation - www.nets-find.net). The philosophy of the program is to help conceive the future by momentarily letting go of the present - freeing our collective minds from the constraints of the current state.

GENI (Global Environment for Network Innovation Program - www.geni.net) is a research program addressing serious problems facing to-day Internet: inadequate security, reliability, manage-ability and evolvability.

3. Conclusion

This paper presents the needs for a self-managing virtual resource overlay that can span across heterogeneous networks that can support service mobility, security, quality of service and reliability as part of Future Internet. In this overlay network, multiple virtual networks co-exist on top of a shared substrate with uniform control. In support of the design of such overlay a new management architectural and system model for Future Internet, which is under development in the AutoI [1] project, is presented. It consists of a number of distributed management systems within the network, which are described with the help of five abstractions and distributed systems - the OSKMV planes: Virtualisation Plane (VP), Management Plane (MP), Knowledge Plane (KP), Service Enablers Plane (SP) and Orchestration Plane (OP). Together these distributed systems form a software-driven control network infrastructure that run on top of all current network and service infrastructures.

Acknowledgment

This work was undertaken in the context of the Autonomic Internet FP7 project [1][2][3][4], which is partially financed by the European Union. We are acknowledging the constructive comments and discussions with Lefteris Mamatas, Meriem Abid, Zohra Boudjemil, Jean-Patrick Gelas, Odysseas Koufopavlou, Zeinab Movahedi, Martín Serrano, and Hubert Zimmermann.

References

[1] Autonomic Internet Project http://ist-autoi.eu/autoi/

[2] Cheng, L., Galis, A., Mathieu, B., Jean, K., Ocampo, R., Mamatas, L, Loyola, Serrat, J. R., Berl, A., de Meer, H., Davy, S., Movahedi, Z., Lefevre, L., - "Self-organising Management Overlays for Future Internet Services"- IEEE Manweek 2008 /MACE 2008; 22-26 Sept. 2008, Samos, Greece/ http://www.manweek.org/2008/index.php?lang=en

[3] Berl, A., Fischer, A., de Meer, H., Galis, A., Loyola, J.R.-" Management of Virtual Networks"- IEEE Manweek 2008/ EVGM 2008; 22-26 Sept. 08, Samos Greece/ http://www.manweek.org/2008/index.php?lang=en

[4] Bassi, A., Denazis, S., Galis, A., Fahy, C., Serrano, M., Serrat, J., -"Autonomic Internet: A Perspective for Future Internet Services Based on Autonomic Principles" - ManWeek 07/MACE 1–2 Nov. 07, San José, California, USA; http://magellan.tssg.org/2007/mace/mace.php

[5] Kephart, J.O., Chess, D.M."The Vision of Autonomic Computing", IEEE Computer", January 03 research.ibm.com/autonomic/research/papers/

[6] IBM, "An Architectural Blueprint for Autonomic Computing" June05, ibm.com/developerworks/autonomic/library/ac-summary/ac-blue.html

[7] Strassner, J., Agoulmine, N., Lehtihet, E., - "FOCALE – A Novel Autonomic Networking Architecture", ITSSA Journal, Vol. 3, No 1, May 2007, pp 64-79, ISSN 1751-1461

[8] Davy, S., Jennings, B. Strassner, J., -"Application Domain Independent Policy Conflict Analysis Using Information Models", NOMS 2008, Salvador Bahia, Brasil, 7-11 April 2008

[9] Clark, D., Partridge, C., Ramming, J., and Wroclawski, J.,-"A Knowledge Plane for the Internet", Proceedings ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Karlsruhe, Germany, pp. 3-10

[10] Strassner, J., Foghlú, M. Ó, Donnelly, W., Agoulmine, N.,- "Beyond the Knowledge Plane: An Inference Plane to support the Next Generation Internet", Proceedings of First International Global Information Infrastructure Symposium (GIIS), 2-6 July 2007, Marrakech, Morocco.
[11] Mathieu, B., Song, M., Galis, A., Cheng, L., Jean, K., Ocampo, R., et all "Autonomic Management of Context-Aware Ambient Overlay Networks" - IEEE ChinaCom, 15-17 August 07, Shanghai, www.chinacom.org/2007
[12] Mathieu, B., Song, M., Galis, A., Cheng, L., Jean, K., Ocampo, R., Lai, Z., Brunner, M., Stiemerling, M., Cassini, M. –"Self-Management of Context-Aware Overlay Ambient Networks - 10th IFIP/IEEE IM 2007; 21-25 May 2007, Munich, Germany; http://www.im2007.org
[13] Galis, A., Denazis, S., Brou, C., Klein, C.–"Programmable Networks for IP Service Deployment" ISBN 1-58053-745-6; pp450, June 2004; Artech House Books
[14] Raz, D., Juhola, A., Serrat, J., Galis, A., "Fast and Efficient Context-Aware Services" ISBN 0-470-01668-X; pp250, March 2006; John Wiley & Sons, Ltd.
[15] Lefèvre, L., "Heavy and lightweight dynamic network services"- The 7th International Symposium on Autonomous Decentralized Systems, Chengdu, Jiuzhaigou, China, April 05.
[16] Curran, K., Mulvenna, M., Galis, A., Nugent, C. –"Challenges and Research Directions in Autonomic Communications" - International Journal of Internet Protocol Technology (IJIPT) - Vol. 2 No. 1; Jan 07; SSN (Online): 1743-8217- ISSN (Print): 1743-8209;
www.inderscience.com/browse/index.php?journalID=144&year=2007&vol=2&issue=1
[17] Davy, S., Fahy, C., Griffin, L., Boudjemil, Z., Berl, A., Fischer, A., de Meer, H., Strassner, J. – "Towards a Policy-based Autonomic Virtual Network to support Differentiated Security Services" — International Conference on Telecommunications and Multimedia (TEMU) 2008; July 16 – 18, 2008; Ierapetra, Crete, Greece. http://www.temu.gr/2008/