# Generic Network Service Components for Networks of the Future

Markus Fiedler
Blekinge Institute of Technology
Dept. of Computing
Karlskrona, Sweden
markus.fiedler@bth.se

## I. Introduction

The ability to dynamically compose services is growing in importance. Appreciable network connectivity has become a mandatory part of successful service provisioning. From the end user point-of-view, good experience of networked services is an obligation no matter which network is used.

There is an ongoing discussion on Networks of the Future including Future Internet, facing the fact that current Internet and its key protocol IP – located at the waist of the well-known hourglass model – are facing difficulties to cope with (a) the ever-increasing multitude and demands of applications from the top, and (b) the diversity of network access technologies at the bottom. The basic data transport and in particular the implicit control feedback facilities provided by a standard socket interface are by far insufficient for many services [1]. As an example, *seamless communication* – facing the need to hand over applications and their data traffic between devices and networks without breaks – needs to work around these limitations [2].

During the recent years, *Service Oriented Architectures* (SOA) arose to facilitate the composition of business processes. In their context, network connectivity may be seen as a service itself, provided by a *Network Service Component* (NSC) that implements a *communication function* with an *Application Programming Interface* (API). In order to make service supply chains work as expected such a network service component has to offer well-defined network transport facilities according to agreed-upon *Service Level Agreements* (SLA).

So far, the notion NSC typically appeared in product specifications and patent applications. In literature, Service Components (SC) are typically discussed in the context of (web) service composition, cf. amongst others [3], with network connectivity taken for granted.

In Section II, this paper proposes a generic structure of a network service component. In Section III, it enumerates some early approaches to NSC from the area of seamless communications. In Section IV, the use case of a NSC-based seamless video service is discussed. Section V concludes the paper.

## II. The Generic Network Service Component

A generic NSC should be usable in any kind of combination with other NSC, which includes the use of services offered by other, potentially simpler NSC in order to provide value-added services. For instance, a NSC that implements routing and/or handover facilities could make use of NSCs providing mobile and wireless access, which is further detailed in Section IV.

Fig. 1 sketches a generic structure of a NSC. It provides separate data and control interfaces towards other (N)SC. Its main features are as follows: (a) separate *data and control interfaces*, typically accessible via APIs, where the control interface allows for requests, responses and notifications even while the data interface is in use; (b) a *context information interface* in order to receive and share context data with the environment (not necessarily other components); (c) an internal control loop for implementing autonomic *self-* functionalities* (self-configuration, self-healing, self-optimization, self-protection); and (d) a *storage* for data to be used internally or to serve as context or control data for other components.

The provisioning is accompanied by exchange of quality, cost and security information [1, 2, 4] during setup and execution of the provided connectivity. While during setup, information about the matching of desired and supported quality, cost and security levels is precious, any (unforeseen) developments during the execution should need to be signaled towards the

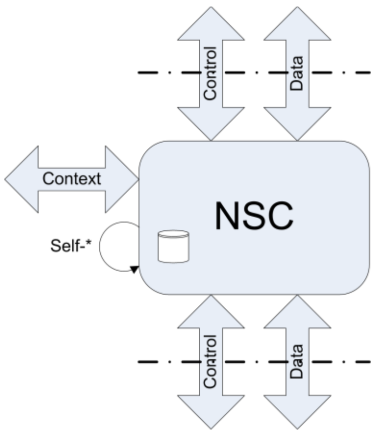


Figure 1. Generic sketch of a Network Service Component.

SC that placed the corresponding order, so that appropriate countermeasures can be taken before protocols and/or user patience time out. This feedback facility [1] takes advantage of the logical separation of data and control streams, which will be underlined by the example discussed in Section IV.

## III. NSC Approaches For Seamless Communications

This section lists a set of communication functions that (implicitly) follow the NSC paradigm, with particular focus on seamless communications.

### *A. PIITSA's Network Selection Box*

Within the Swedish VINNOVA-sponsored project PIITSA (Personal Information in Intelligent Transport Systems using Seamless communications and Autonomic decisions) a Network Selection Box (NSB) was designed and implemented [2]. While its first draft provided specific functions for data transfer, the final version offered a transparent data interface based on a virtual network interface. This means that the NSB was invisible to the data stream, which was diverted into the most suitable tunnel in order to make the application Always Best Connected (ABC). There was a separate control interface available for providing parameters for or to control the decision making inside the NSB. This decision making built amongst others on internal performance monitoring.

### *B. The Cognitive Network Stack*

The vision document [4] provided by the European FP6 Network of Excellence (NoE) Euro-NGI discussed a Cognitive Network Stack (CGS), which has its roots in above-mentioned NSB augmented by (a) mediation and (overlay) routing functions; (b) self-organisation features; and (c) lightweight monitoring capabilities. Thus, the CGS had already most of the features as described in Section II, with exception of the context interface and ability to freely combine such entities.

### *C. ROMA*

ROMA denotes a "middleware framework for RObust Mobile Applications" [5], and allows for abstraction of the complexity of connectivity including routing and network choice. Its predecessor is a middleware for routing in overlay networks (ROVER), developed within a homonymous Euro-NGI Specific Joint Research Project and subsequent projects founded by the Swedish Internet Infrastructure Foundation ".se". The step from ROVER to ROMA was made by adding seamless handover facilities. Interfaces towards applications are both communication functions and a legacy stack.

## IV. Use Case: Seamless Video Conferencing

Consider the case that a video conference lasts longer than expected, and one of the delegates needs to leave the office and take the session with her on a mobile phone. In a classical case, she would have to disconnect, switch the network and reconnect. With help of Mobile IP, the change of the IP address could be automated, but still, one expects some handover delay due to the need of "break (the old)-before-make (the new)" connection. True seamless experience is reached through "make-before-break" as outlined in [2]. In the sequel, it is described how NSCs enable such a transition.

Assume we have a seamless video network service component (SV-NSC), a WLAN (W-NSC) and a mobile network service component (M-NSC) at hand. When starting the video conference, the SV-NSC uses the W-NSC, whose control interface is used to exchange all relevant setup information. When the user needs to leave the office, the W-NSC discovers a slowly decreasing link quality and alerts the SV-NSC through the control interface. The latter establishes a second connection via the M-NSC's control interface. All this happens while the data stream of the video conference still uses the W-NSCs data interface. Once the M-NSC's data connection is stable, the SV-NSC switches over the data traffic in a seamless manner. If the user would return to the office and the video conference would still be running, a seamless switch back to the W-NSC would be handled in a similar way, triggered by the discovery of the WLAN. The emergence of new network types is rather simple to handle, as long as they can be encapsulated in a corresponding NSC. On the other hand, the SV-NSC could also use another NSC with handover functionality.

## V. Conclusions

This paper has illustrated a couple of added values obtained from thinking modular in terms of network connectivity. Network Service Components providing transport functions enable new ways of keeping Service Level Agreements, e.g. through advanced monitoring and feedback facilities as well as controlled network switches following the "make-before-break" principle. Future work will need to deal with key issues regarding the interfaces for data and control. In general, revolutionary approaches (API) have to be traded against evolutionary approaches (building on legacy technology). The control interface brings up issues such as component discovery, initiation, execution, termination and feedback, which need to be addressed with SOA in mind.

## Acknowledgements

Thanks to the colleagues from the FP7 NoE Euro-NF for fruitful discussions on the vision of the Network of the Future.

## References

[1] M. Fiedler, "Quality feedback flows in future networks," To appear in *e&i Elektrotechnik und Informationstechnik*, 7/8/2009, Springer, Wien.

[2] S. Chevul et al., "Network Selection Box: An Implementation of Seamless Communication," Third EuroNGI IA.8.3 Workshop on Wireless and Mobility, Sitges, Spain, June 2006. Springer Verlag, *LNCS 4396: Wireless systems and Mobility in Next Generation Internet*.

[3] J. Zhu, J. Wang and B. Li, "On dynamic and concurrent model of web service components," *11$^{th}$ Int. Conf. on Computer Supported Cpperative Work in Design* (CSCWD 2007), April 2007, pp 568–572.

[4] D. Kofman, ed., "Euro-NGI Deliverable D.WP.SEA.10.1.1: A view on Future Communications," May 2006.

[5] A. Popescu et al., "ROMA: A middleware framework for seamless handover," *Second Euro-NF IA.7.1 Workshop on Future Internet Architecture and New trends in Network and Services Architectures*, Santander, Spain, June 2009.