# Unifying Cloud and Carrier Network

## EU FP7 Project UNIFY

András Császár*, Wolfgang John*, Mario Kind[$], Catalin Meirosu*,
Gergely Pongrácz*, Dimitri Staessens^, Attila Takács*, Fritz-Joachim Westphal[$]

* Ericsson Research [$] Deutsche Telekom ^ iMinds
Corresponding author email: *andras.csaszar@ericsson.com*

***Abstract*— Telecom providers struggle with low service flexibility, increasing complexity and related costs. Although "cloud" has been an active field of research, there is currently little integration between the vast networking assets and data centres of telecom providers. UNIFY considers the entire network, from home networks up to data centre, as a "unified production environment" supporting virtualization, programmability and automation and guarantee a high level of agility for network operations and for deploying new, secure and quality services, seamlessly instantiatable across the entire infrastructure. UNIFY focuses on the required enablers and will develop an automated, dynamic service creation platform, leveraging fine-granular service chaining. A service abstraction model and a proper service creation language and a global orchestrator, with novel optimization algorithms, will enable the automatic optimal placement of networking, computing and storage components across the infrastructure. New management technologies based on experience from DCs, called Service Provider DevOps, will be developed and integrated into the orchestration architecture to cope with the dynamicity of services. The applicability of a universal node based on commodity hardware will be evaluated in order to support both network functions and traditional data centre workloads, with an investigation of the need of hardware acceleration.**

***Keywords—UNIFY; SDN; NFV; service chaining***

## I. INTRODUCTION

The current networking and Internet service models present a challenging environment for network service provider (NSPs) as over-the-top players leverage the NSPs' connectivity infrastructure to provide services to end-customers without the NSPs gaining fair revenues. NSPs are pushed towards becoming plain bitpipe/connectivity providers. Due to the extreme high investments required by network operators and due to declining revenues the existing service models are likely not sustainable. It is crucial for NSPs to be able to offer and sell advanced services instead of just connectivity, and to be able to optimize network economics by improving operations. These could cover services for end-customers, as well as "upstream" offered services towards other operators or over-the-top providers (OTT). Moreover, it is essential to leverage the full flexibility and capabilities offered by networking equipment to introduce new services and improve existing ones towards better customer experience and reduced OpEx.

The business and platform operations of service providers are increasingly complex due to increased requirements:, specifically the following:

- Isolated and monolithic platform based operation of service creation architecture
- Accelerating demand for fixed-mobile convergence
- Sharing of infrastructure, packet forwarding and processing platforms with other operators
- Sharing of services and appliances among service creation platforms including edge devices in home networks and data centres
- Increased connections with other access network operators, e.g. FTTx deployed by municipalities
- Continuous evolution of service creation architectures and its impact on CPEs
- Migration and integration of specific services particularly applications like security (e.g. firewall) or delivery optimizations (e.g. TCP accelerators)

The aim of the consortium of the EU FP7 Integrated Project "UNIFY" is thus to provide sustainable and agile solutions for handling mentioned challenges and to provide the means for flexible service creation within the context of unified Cloud and Carrier Networks, especially focusing on telco functions. The UNIFY consortium consists of a subset of the partners which cooperated successfully in the past in the SPARC project [2]. The consortium consists of service providers (DT, TI, OTE), vendors (Ericsson, Intel), universities (BME, EHU, Polito, TUB) and research institutes (IMINDS, SICS, Acreo).

This paper describes the vision of the UNIFY consortium, the associated challenges, the expected benefits and the main goals of the project.

### A. Related Concepts

*Cloud services* and *cloud networking* are receiving enormous attraction from providers and customers. For customers, on one hand, cloud services mean reduced costs for obtaining and managing hardware, while enabling more flexible control of resources. This is enabled by *virtualising* (processing, storage and networking) resources of the provider. On the other hand, cloud also means the freedom of accessing data and the virtualised resources anywhere throughout the network. For providers, cloud services represent an opportunity to offer new services with highly efficiently utilised hardware leveraging *resource sharing* of virtualised appliances.

Network service providers are in the progress of building *data centres* (DCs) at some of their sites. Data centres are the enablers for cloud services and currently the only well programmable part of the infrastructure housing mostly generic purpose hardware. NSP DCs not only allow offering cloud services to customers (enterprise as well as residential) but also

provide an opportunity to concentrate existing and new telco functions, further increasing their utilisation, and decreasing costs compared to monolithic, vendor dependent and/or purpose oriented gear. So, "*Telco Cloud*" is also about virtualising networking and telecom functions and moving these to the NSP's cloud infrastructure. In addition, the "Telco Cloud" will be offered to other service providers allowing the deployment and advertisement of own services, leading to new revenues for NSPs. One could easily imagine billions of Euro/Dollar revenues for network integrated and assisted platforms for firewall, (cloud) anti-virus or energy optimising applications. This two-sided market principle is typically described as Platform, Infrastructure or Network as a Service (P/I/NaaS). To foster this transformation, especially in the mobile area, an industrial forum has been established within ETSI called *Network Functions Virtualisation* (NFV).

NSPs' cloud infrastructure, however, not only consists of big central data centres: there could be smaller but more distributed data centre locations, all capable of running virtual machines. Furthermore, the *distributed cloud* is not only implemented by deploying smaller generic purpose server clusters at various network locations, but also by many existing networking equipment supporting the installation of service blades that contain generic purpose hardware. These service blades implement existing telecom functions that do not require dedicated hardware. Seeing the opportunity, vendors and NSPs are in the progress of leasing these service blades to the overall cloud infrastructure. Such service blades may reside in various nodes such as BNG, 3GPP PGW or SGW, but may also reside in nodes implementing 3G radio network controller (RNC) functionality, or even some transport nodes lower in the aggregation network may support such cloud extensions. An important aspect of distributed cloud is the possibility to instantiate functions at various points in the network.

The appearance of *Software Defined Networking* (SDN) enables a new control architecture and operation practices with fine granular control over services. SDN gets rid of monolithic nodes and enables the introduction of new features and services by installing software on a (logically) centralised controller. The principle of *Service Chaining* is derived from operation of data centres, that already deal with virtualization of computing resources and automatic deployment of virtual machines and applications on top. The fine granular control of traffic flows,

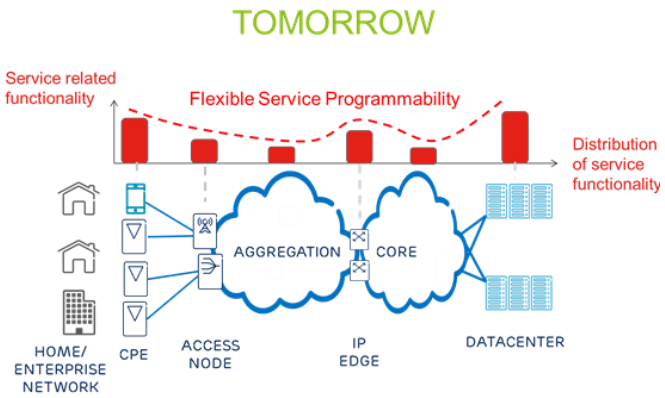


Figure 1: Unified programmability enables flexible placement of service functions

enabled by SDN, allows NSPs to offer several value added functions (such as firewalls, parental control, etc.) in addition to connectivity services. These value added functions may be offered in flexible bundles and may be personalised to subscribers. Technically, the traffic flows are passing a "service chain", i.e. a series of advanced service functions (ASFs). SDN allows fine granular control of traffic steering between ASFs, and, indeed, service chaining became one of the most promising applications of SDN.

While these existing technologies enable flexible placement of service components and fine granular traffic steering between these components, NSPs still do not possess means to optimally orchestrate the placement of service functions in their networks – this is a major target for UNIFY.

### *B. UNIFY Vision and Service Provider Benefits*

Although cloud computing and networking have been two active fields of research, there is currently little integration between the vast networking assets and data centres of telecom providers. UNIFY addresses this by considering the entire network, from home networks up to data centre, as a "unified production environment". While today advanced service related functions are concentrated in or near the IP Edge, and flexible programmability is mostly possible in the data centres only, UNIFY envisions that networking equipment, starting from the customer premises equipment (CPE), through access nodes and up to the data centre, will be more programmable, allowing the configuration and instantiation of fine granular service functions, as shown in Figure 1. The unified production environment opens up the potential for virtualization, programmability, and automation, which guarantee an unprecedented level of agility for network operations and for deploying new, secure and quality aware services with seamless instantiation across the entire infrastructure.

Leveraging this, service providers can offer functionality previously provided either by over-the-top providers (e.g. content and caching), or by CPEs (firewall, parental control) as value-added services on top of connectivity services, or even provide an integrated platform open for such over-the-top providers. Moreover, these service functions can be flexibly placed in the network at the service provider's discretion enabling optimisation taking into account quality of experience (e.g., delay) or operational economics aspects (e.g., load, energy efficiency, etc.). The provisioning of services as well as the operation and optimization of the infrastructure will be conducted in the most automated manner. Therefore, the architecture will contribute to a flexible and optimized use of resources; thereby reduce costs and energy consumption while also open up new options for fast and easy service creation. This in turn will help to improve the economical production of existing services, expedite the delivery of new services and create new business opportunities and possibly new business models. UNIFY envisions an architecture that achieves economics of scale easier by leveraging resource sharing, and in this way it also accelerates broadband rollout. Moreover, todays services evolve very rapidly. UNIFY aims to enable dynamic scaling of resource according to application needs.

### *C. Key Aspects to Investigate*

The first aspect to realize this vision is to identify key network services of a converged fixed-mobile network, and

then to identify the minimal set of components which will provide more flexibility for service chaining. Instead of instantiating a DPI or BRAS/BNG function we aim to divide these functions into smaller components. DPI, e.g., may be subdivided to a complex traffic classification component that identifies the traffic type and a flow cache component that keeps track of already identified traffic and applies preconfigured policies. In the case of BRAS/BNG, identified components are Layer-2 termination functions, QoS and policy enforcement as well as IP routing.

The second aspect towards the UNIFY vision is a service abstraction model and an associated domain specific service creation language to cope with the network flexibility required for establishing and programming service chains. It should be possible to program services with an interface supporting automated, policy-driven placement, optimisation and orchestration of service functions and components. Service chains are programmed into the network based on a combination of information elements from the different layers (e.g. L2-L4 and possibly higher). Based on operator policies, various services can be applied to traffic flows in the network.

A third aspect is the feasibility of a universal node architecture boosting the flexibility in instantiating various types of service functions throughout the network. The universal node may be based on standard x86 components, potentially paving the way for future hybrid devices handling traditional network oriented functions like BNG or EPC in parallel with upcoming network functions like customer oriented security or measures to improve energy efficiency.

A fourth aspect is a new network management and operation paradigm, including tools and workflow to cope with increased network/service agility and to handle services end-to-end in the unified production environment. The concept of the UNIFY Service Provider DevOPs will bring closer the previously separated roles of developers and infrastructure operators. The verification and activation of complex new services needs to be eased while at the same time novel network and service observability, diagnostics and troubleshooting methods are needed to ensure proper operation of the agile and unified Cloud and carrier network.

## II. Overarching Architecture

A main architectural target is to lower the technical entry barriers for sharing resources and services of the access and aggregation domain. The Overarching Orchestration, Management and Control (OOMC), see Figure 2, is ensuring the optimal distribution of service components and functions on the network- / data centre infrastructure.

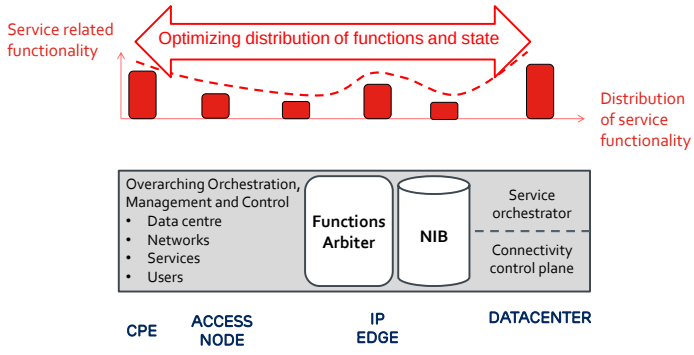


Figure 2: Overarching Architecture, logical view

Within the OOMC the optimisation of function placement is provided by the Functions Arbiter, which is responsible for dynamically orchestrating network functions and resources (e.g., instantiating and moving VMs) and to steer network traffic flows in a reliable, efficient, and high performance manner. The Functions Arbiter works with another component integrated in the OOMC that maintains historical and real-time network data. This component can be called a Network Information Base (NIB). The NIB is a map of network and processing resources as well as their current state. Network and function performance is monitored at continuously and the data is fed into the NIB for immediate response to faults and degradations. The NIB forms the basis for the creation of service maps used to direct end-users' traffic to the processing resources capable of executing required functions and services, allowing a differentiating service experience for customers.

The OOMC provides interfaces to the NSP and also provides hooks enabling end-user and service provider applications to create and update service functions or parameters in an automated and programmable way. Advanced service functions should be configurable at different granularities such as per-subscriber, per-connectivity-service or per-application-micro-flow. OOMC is likely organised into two layers, an upper service orchestration layer, and a lower connectivity or infrastructure controller that provides an abstract network view and the connectivity services therein.

The OOMC has to meet the expected flexibility requirements so both the service orchestration layer and the connectivity controller will show a logically centralised view of itself to upper applications and the network operator. At the same time, the scalability and redundancy requirements likely require that the control plane components (processing functions, data stores) are actually distributed on multiple machines located both in central and in distributed locations.

The OOMC also offers a virtualisation interface that provides access to foreign operators which could also include access for over-the-top service providers. A foreign operator leveraging the views provided by the native operator's control plane along with an ASF virtualisation interface can instantiate and control ASFs in the physical operator domain. The foreign operator can use its existing service orchestration platform to control these virtualised ASFs. The hardware owner (the native operator) may rent virtual instances to several other foreign service providers, thereby sharing its infrastructure and utilizing hardware resources more efficiently.

## III. Service Orchestration and Programming

UNIFY has a vision of service chains composed of micro functional blocks, covering traditional network and data centre virtualized entities. These blocks are orchestrated dynamically to form advanced services that are highly resilient and economic to implement and operate. The fine-granular service chaining architecture will be based on a unified view on the infrastructure, which makes flexible placement and dynamic adjustment of service components in the network possible. This

will allow optimizing the use of network resources as well as the allocation of L4–L7 network functions.

Network virtualization essentially decouples the services from the constraints of the underlying infrastructure, introducing a new degree of flexibility on how, where, and when services and service chains can be allocated. Further optimization depends on the service: sometimes a service should be embedded so that the resource cost is minimized and that the infrastructure is optimally shared among multiple services in a secure way; in other scenarios, the maximal network load or congestion should be minimized; or the service should be migrated in order to deactivate parts of the physical network for energy efficiency purposes. Finally, also fault-tolerance aspects may be considered here.

Current programmability techniques work on different abstraction levels:

- Functionality abstraction: Description languages for services, resources, NFV
- Transport/Forwarding level: OpenFlow, OF-Config, NetConf, ForCES
- Controller level specification of forwarding functions: Frenetic, Pyretic, NetCore, Nettle
- Service level: Orchestration software for services
- Full Service Deployment: adds assignment of functionality to nodes and network service monitoring

These techniques are considered standalone elements and it is not clear how to combine them into a flexible but scalable solution based on such components. A novel programmability framework, extending current standards where appropriate, needs to be defined in order to enable the rapid development and deployment of micro functional blocks and service chains. UNIFY will derive a generic programming and optimization framework which supports a variety of services and service chains, infrastructures, and objective functions. A major objective of this framework is the reduction of the complexity (e.g., need for manual configurations), rendering the management of the deployment less labour-intensive.

Initially, the following elements will be considered:

- A hardware platform for performing the main data plane functions. It will consist of a forwarding engine and additional (custom or off-the-shelf) hardware for other network functionality leveraging NFV concepts.

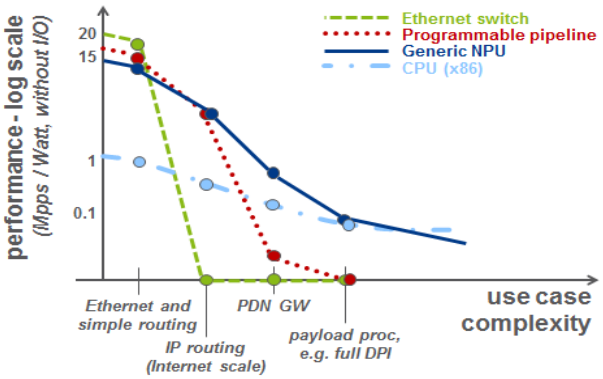


Figure 3: Chip efficiency vs. use case complexity [1]

- A (set of) transport SDN controller(s) serving transport programs (e.g., tunnelling and OAM functions)
- A (set of) service SDN controller(s) where services are defined and instantiated
- A virtualization system
- A network management system

At least the following aspects are required to specify all functions and relations in the framework:

- description languages (function descriptions, resource descriptions and service descriptions)
- configuration languages
- functional mappings between description languages (for instance, mapping service description to resource description when service instantiation takes place)
- management functions for placing components and allocating resources
- optimization blocks, possibly requiring feedback loops
- APIs allowing full flexibility of functionality, resources and services

Automated optimization algorithms have the capability to adapt the chain to varying customer demands and changes to business policies. Connectivity between service blocks is considered essential and has similar properties with respect to on-demand activation and mobility. UNIFY will identify possible performance bottlenecks relating to service creation and define methods for implementing flexible service creation quickly and at carrier scale. Furthermore a complete set of primitives (functions) is required for assigning functionality to NFV nodes, migration of virtual nodes and service endpoints over the network in order to deal with, e.g., failures, equipment upgrades, or different optimization targets like energy efficiency, QoS, latency.

The ultimate goal of the programmability framework is to reduce the operational costs associated with service orchestration, accelerate innovation and deployment of new services, tailored to maximize customer experiences on any device connected to the network.

## IV. UNIVERSAL NODE

In a recent paper [1] we argue, in contrast to the common view, that the cost of programmability in the data plane is relatively small. DP programmability alone does not incur high performance penalty. Today's Ethernet chips are already at a level of complexity where the additional overhead of programmability is relatively low. This realization may have serious impact to the unfolding of SDN. If the price of programmability is indeed low, more programmable chips may prove to be a cost-effective solution for a wide-range of task eventually paving the way towards SDN.

Today the above is valid for network processing units (NPUs) as one can see on Figure 3. Note that use case complexity (axis X) is not a continuous axis, while the efficiency axis (axis Y) uses logarithmic scale.

It is visible that

- As the complexity of the use case grows, naturally the performance of all solutions decreases

- Purpose-built hardware solutions have around 20-30% advantage over more generic, but packet processing oriented NPUs
- Purpose-built chipsets are limited to simpler scenarios: when moving towards more complex tasks it is simply not possible to use them
- The programmable pipelines studied so far did not show significant advantage over other NPUs – but we assume that a well-designed pipeline is between Ethernet switches and NPUs for simple tasks, while it will fall back when the use-case gets more complicated
- Generic, non-optimized CPUs, such as Intel x86 have clear disadvantage in simpler, more packet processing specific use-cases
- On the other hand generic CPUs can outperform NPUs in really complicated use cases, such as metrics based packet classification, where complex instructions and floating point arithmetic is used

Using the values above, we could come to two completely different solutions:

Option 1: The right hardware for SDN is the combination of Ethernet switches and generic CPUs, like Intel's x86 – this way the system would be split to a slow and a fast path, hopefully in a way where the bulk of the traffic could be handled directly in FP. If the FP and SP can be merged into one hardware unit (card or box) that is even better.

Option 2: The right hardware for SDN is NPU-based where we don't have to define a strict boundary between slow and fast path: the behaviour and the performance of the selected data plane hardware will depend on the program that runs on it.

Option 1 would lead to a more complex system, but if we could build the network in such a way that we mostly use only switching functionality it could be cheaper and performance-wise more predictable than Option 2. Option 2, however, is more flexible and as the performance-difference is not huge between switch hardware and NPUs it would be almost as powerful even in use cases where most traffic can be handled by the fast path of Option 1. In some intermediate use cases where most of the traffic passes through the CPUs in Option 1 the NPU-based Option 2 would be far more powerful.

However in the future the gap seems to be closing even between CPUs and NPUs: as CPUs get more and more cores and the power requirements are more and more important in that segment eventually the gap could diminish. A good example for these possible "gap-bridges" is Intel's Xeon Phi product which contains 60 low-power cores. UNIFY will aim to study the current hardware market and answer the question whether one solution could fit all.

## V. SERVICE-PROVIDER DEVOPS

Today's NSPs network management was developed as add-on to particular technologies, leading to significant integration efforts for launching new services. Both centralized and distributed algorithms have been developed to solve specific problems related to fault or performance management, considering networks that are static from a functional perspective. Introducing new functionality in the network is complex and requires significant efforts for verification and integration. However, assuming multiple, parallel service providers and demanding openness in future networks, efficient operations is a major differentiator compared to current relatively cohesive service provider environments

Recent concepts such as SDN and NFV are evolving networks into a dynamic environment with fast-paced innovation of network functions and services. SPARC [2] provided first considerations on managing an SDN carrier network by mapping traditional management functions onto a hierarchical control architecture. However, this approach falls short on supporting dynamic services chains as envisioned by UNIFY. Furthermore, current management techniques, based on simple counters, physical network taps and active probing, provide only a small part of the observability required in such a dynamic environment and scale poorly.

The DevOps movement [3] addresses the gap between developers and operational teams in enterprise networks by borrowing techniques from agile programming practices, building tools that automate well-known manual steps. Scaling DevOps to software-defined carrier networks is, in a sense, like scaling agile development to large projects in multi-national software companies.

Major challenges include:

- inherent geographical distribution of the physical network nodes, imposing non-negligible delay and high cost (both time-resource-wise) for any intervention that needs physical contact with the equipment
- network and service state visibility limited to administrative domains, yet quality of experience needs to be managed end-to-end
- difficulties to define and extract the relevant subset of the network state for troubleshooting particular service conditions by the developers
- high velocity of new releases, despite a substantial need for regression testing and validating that new requirements are met by the elastic infrastructure

While first debugging and troubleshooting techniques for SDN have been proposed (e.g. [4][5]), integrating them into operational network management systems is challenging since most tools are limited to particular well-defined problems. In [6], it is shown how recently-developed troubleshooting tools fit into a coherent debugging framework for a typical SDN control hierarchy. However, this systematic approach also highlights gaps in the debugging tool-chain as well as opens questions, e.g. how to extend network programming with common software engineering tools. Furthermore, existing SDN troubleshooting tools [5] show a lack of programmatic interfaces that would enable composition, automation and easy integration into network operations workflows.

For fast service creation from dynamic service components, automation is essential and there is a need to bring closer the previously separated roles of developers and operators of future telecommunication infrastructure. Compared to DevOps,"Service Provider" DevOps (SP-DevOps) teams need to possess a broader skill set for troubleshooting (layer 1-7, instead of layer 3 as in the data center) and have an increased focus on high availability because of the expectations from a

carrier network. Also, infrastructure is deployed in much more locations, with less and less redundancy closer to the access air interface, meaning that costs associated to service failures are likely higher and situations are potentially more difficult to fix than in the data center.

The concept of the UNIFY SP-DevOps is depicted in Figure 4: The members of an agile SP-DevOps team are in the centre of the process, empowered with a set of automated tools that increase efficiency many times. In UNIFY, we will work on tools, their interfaces and workflows that allow aggregating individual troubleshooting and debugging tools into powerful mash-ups that adapt dynamically to complex service chains.

Programmable and consistent observability features will enable the operations personnel to gather detailed views on the runtime of a newly introduced service and exchange the knowledge with designer and developers. Among the tools, we envision a network debugger that uses flexible observability points (extensions of the fixed NDB-introduced breakpoints [4] and measurement sketches [7]) along with methods that allow configuring them in a consistent and efficient manner in a virtualized network. Consistency is important because low-level access to node programming capabilities create increasing opportunities for introducing errors. While this has been addressed for flow configuration management [8], we argue that DevOps teams need a similar assurance of consistency for the observability processes they dynamically deploy across multiple network domains.

The amount of traffic expected in future networks along with the high speed of service introduction make it simply uneconomic to direct all traffic towards troubleshooting processes. It is also impractical to access all counters supported by nodes, or continuously execute frequent active measurements at large scale. Approaches that improve probabilistic algorithms [9] are needed in order to reduce the traffic load on the network and management systems while increasing the knowledge inferred about the network state. They should deliver intelligent ways of sampling, along with dynamic models that adapt to network conditions and are able to maintain a high accuracy of state estimation.

In a dynamic service chain, service validation needs to keep pace with changes in the location of service components, including their connectivity. The validation needs to ensure that new functionality introduced into the chain is compatible, that allocated resources are adequate and correctly configured. The difficulty increases when dynamic load balancing may be introduced anywhere within a chain. Significant advances are needed in the area of automated formal verification and correctness. Again, probabilistic algorithms may help in reducing the search space and speeding up the verification.

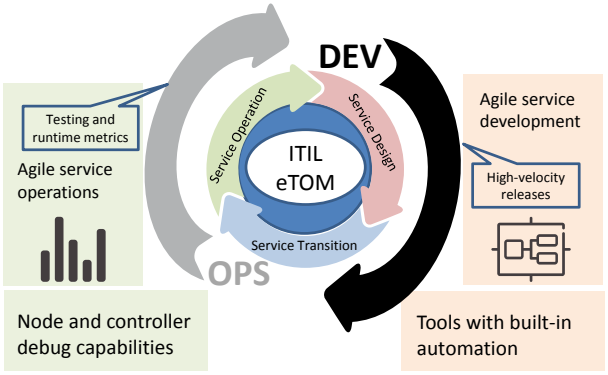


Figure 4. The concept of SP DevOps

We conclude that the standard stack-based network management model is no longer adequate when network functions are virtualized and can be migrated between locations. Thus, with SP-DevOps, we set out to bring carrier-grade network management into the era of SDN.

## VI. Summary

UNIFY targets the problem of slow and rigid networking where flexibility of service creation is limited. UNIFY envisions and strives to create an architecture where the whole network from the homes up to the data centres form a Unified Production Environment, where it is up to the operator and automated orchestration engines to distribute functions and state anywhere in the network. A key enabler to this is virtualisation (including NFV) and service chaining, while UNIFY will concentrate on the missing pieces of orchestration and network function placement optimisation. This potentially includes decomposition of traditional network functions to more fine granular components.

We expect UNIFY to (*i*) derive carrier requirements and describe use-cases with supporting techno-economic studies; (*ii*) describe the UNIFY service chaining architecture enabling flexible placement and dynamic adjustment of service components, and automation and optimisation of operations; (*iii*) describe and evaluate novel capabilities and algorithms for debugging and troubleshooting dynamic service chains (i.e. SP-DevOps); (*iv*) characterise performance of standard hardware and accelerators where needed; (*v*) develop an integrated prototype.

## Acknowledgment

Thanks to all partners of the UNIFY consortium, who contributed to the UNIFY technical work description.

## References

[1] G. Pongrácz, Z. Turányi, L. Molnár and Z. L. Kis, “Cheap Silicon: Myth or Reality? Picking the Right Data Plane Hardware for Software Defined Networking”, to appear in HotSDN 2013

[2] The SPARC consortium, “SPARC Deliverable D3.3: Split Architecture for Large-Scale Wide-Area Networks”, 2012. [Online] Available at: http://www.fp7-sparc.eu/assets/deliverables/

[3] B. Rockwood, The DevOps Transformation, keynote at the USENIX LISA’11 conference, https://www.usenix.org/conference/lisa11/devops-transformation, retrieved Aug 1, 2013

[4] B. Heller, N. Handigol, V. Jeyakumar, N. McKeown,and D. Mazières. “Where is the debugger for my Software-Defined Network?”,in HotSDN, 2012.

[5] M. Canini, D. Venzano, P. Peresini, D. Kostic, and J. Rexford. “A NICE Way to Test OpenFlow Applications”, in NSDI, 2012.

[6] B. Heller, C. Scott, N. McKeowny, S. Shenker, A. Wundsam,_ H. Zeng, S.Whitlock, V. Jeyakumar, N. Handigol, J. McCauley, K. Zarifis, and P. Kazemian, “Leveraging SDN Layering to Systematically Troubleshoot Networks”, to appear in HotSDN 2013

[7] Y. Minlan, J. Lavanya and R. Miao, “Software defined traffic measurement with OpenSketch”, in NSDI 2013.

[8] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, “Composing software-defined networks”, in NSDI 2013.

[9] A.G. Prieto, D. Gillblad, R. Steinert, and A. Miron, "Toward decentralized probabilistic management," Communications Magazine, IEEE , vol.49, no.7, pp.80,86, 2011