Objective FP7-ICT-2009-5-257448/D-2.4

Future Networks

Project 257448

“SAIL – Scalable and Adaptable Internet Solutions”

D-2.4
(D-A.4) Final Migration Description

Date of preparation: **13-02-11**
Start date of Project: **10-08-01**
Project Coordinator: **Thomas Edwall**
Ericsson AB

Revision: **1.0**
Duration: **13-02-28**

Document Properties:

Document Number:	FP7-ICT-2009-5-257448-SAIL/D2.4	
Document Title:	(D.A.4) Final Migration Description	
Document responsible:	*Jorge Carapinha (PTIN)*	
Author(s)/editor(s):	*Michael Soellner (ALUD)* *Pedro Aranda Gutiérrez (TID)* *Victor Souza (EAB)* *Asanga Udugama (UHB)* *Susana Pérez (Tecnalia)* *Lucian Suciu (FranceTelecom)* *Hannu Flinck (NSN)* *Börje Ohlman (EAB)* *Jean-François Peltier (FT)*	*Benoit Tremblay (EAB)* *Johan Myrberger (EAB)* *Azimeh Sefidcon (EAB)* *Avi Miron (Technion)* *Matthias Keller (UPB)* *Fabian Schneider (NEC)* *Ove Strandberg (NSN)* *Marco Marchisio (TI)*
Target Dissemination Level:	PU	
Status of the Document:	Initial Version	
Version	1.0	

Production Properties:

Reviewed by:	Hannu Flinck (NSN), Holger Carl (UPB)

Disclaimer:

This document has been produced in the context of the SAIL Project. The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2010-2013]) under grant agreement n° 257448.
All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.
For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Abstract:

This document proposes a migration roadmap for the three SAIL technology pillars, namely Network of Information (NetInf), Open Connectivity Services (OConS), and Cloud Networking (CloNe). It builds on the technical achievements of SAIL Work Packages A, B, C and D and outlines an overall migration path for the deployment of SAIL project results over time, starting out from existing architectures and protocols. In addition, the synergies that can be exploited from the combined deployment of these components to facilitate migration are also analysed.

Standardisation, as enabler of technological migration, is another major topic of this document. An overview of the technical fields addressed by SAIL are analysed from a standardisation point of view; in addition, the standardisation activities carried out in the framework of the SAIL project are described.

Keywords:

Migration, SAIL, NetInf, OConS, CloNe, Standardisation

Document History:

Revision	Date	Issued by	Description
1.0	11/02/2013	Jorge Carapinha	Initial revision

Executive Summary

This document is a public deliverable of the Scalable Adaptive Internet Solutions (SAIL) EU-FP7 [1] and addresses migration. It builds on the achievements of technical Work Packages B, C and D, respectively, Network of Information (NetInf), Open Connectivity Services (OConS) and Cloud Networking (CloNe), as well as results from Work Package A (Impact & Collaboration Enabling), dealing with business models and business case scenarios.

Migration is related to how new features, services or technologies can be introduced into an existing system. Multiple approaches can be followed, which can be categorized in three basic models: disruptive migration by parallel deployment, managed migration by parallel deployment, and seamless migration by incremental deployment.

Based on those models, a migration roadmap for the aforementioned SAIL technology pillars is proposed in this document. An overall migration path for the deployment of SAIL project results over time, starting out from existing architectures and protocols, is outlined.

In addition to the migrations roadmaps of NetInf, OConS and CloNe components, it is important to note that the combined deployment of these components enables the exploitation of synergies that simplify the migration process. An analysis of potential scenarios to exploit those synergies as well as the interactions with other systems is also provided in this document.

Standardisation is often a key enabler of technology migration. This document provides an overview of relevant Standards Development Organisations (SDOs), describes how SAIL technical achievements can be positioned in the framework of relevant standardisation bodies, and identifies the most relevant contributions from SAIL partners in terms of standardisation. In addition to the activities pursued in formal standardisation bodies, contributions to the open source community, which in many cases constitute *de facto* standardisation activities, are also noteworthy, which includes the generation of drafts and implementations of SAIL as open source software.

Contents

1 Introduction

1.1 Motivation and Objectives

Migration usually refers to the process of moving from one operating or technological environment to another, which, in most cases, is thought to be better or more efficient. Usually, in fields related with Information and Communications technology (ICT), migration is an important source of concerns and often represents an obstacle against the widespread deployment of new technical solutions. History has shown that the lack of a sound migration path for deployment of new ICT technologies can represent a hurdle against its widespread acceptance. Adopting innovative solutions is often hampered by not having a viable incremental deployment plan.

Incremental deployment of results has been identified as an ambition of the SAIL project from the very beginning. The definition of a sound migration strategy is an essential requirement to accomplish that ambition. SAIL identified as an objective to “set out an overall migration path for how the project results can be deployed over time, starting out from existing architectures and protocols” [7].

Incremental deployability is usually evaluated through a number of requirements:

- Technology should be backward-compatible and interoperable with legacy technologies, so that a massive upgrade or replacement of equipment will not be required. This implies two different sub-requirements:
    - SAIL systems can interwork with legacy systems
    - SAIL systems can interwork and take advantage of SAIL functionality (fully or partially) across legacy systems.
- A sound business model can be put in place: in order for a technology to be deployed on a significantly wide scope, there should be a clear benefit to all potentially involved players (e.g. end users, content providers, infrastructure providers, service brokers).
- There should be initial incentives (materialized in immediate benefits) for early adopters of the new technology:
    - Deployment provides benefits no matter how small the deployment is.
    - Benefits are incremental: larger deployment offers higher benefits.

Migration in networking technologies has been driven by the rapid pace of innovation and demand. Particularly in the last few years, the capability to cope with the evolution of technology has represented a challenge for the industry, particularly for network operators. Although technology and service migration have represented an important issue for network operators, not many studies have provided a fully inclusive analysis of the problems involved in this process. [16] provides an important contribution in this problem area, by proposing an approach to modelling and studying service migration and discussing the cost aspects and relationships between the variables of migration, in such a way that an operator would be able to plan its service migration process taking into account certain budget and duration constraints.

In the present deliverable, we focus specifically on SAIL technologies, from a migration perspective. A number of initial migration guidelines have been identified in SAIL Deliverable A.2 [2]. This document elaborates further on this topic, based on the latest results from Work Packages A, B, C and D and overviews migration for SAIL technologies, including functionalities, applications and network perspectives tackled by each of them.

Standards often represent a migration enabler and mitigate the risk that investments become prematurely obsolete. Therefore, an analysis of migration cannot be complete without a proper evaluation of standardisation in the relevant technological fields. In addition to formal standardisation efforts, open source software has emerged in several areas as an important source of *de facto* standards, which must not be overlooked. In summary, the objective of this document is basically to:

- Identify the main migration requirements and principles and associated challenges;
- Define a SAIL migration roadmap;
- Overview of standardisation bodies which are relevant to SAIL and standardisation activities conducted by SAIL partners.

This report is meant to be a stand-alone document; however, it assumes a basic understanding of the mechanisms and technical approaches put forward by SAIL WPs A, B, C and D, as described in other SAIL project deliverables, referenced in the text.

1.2 Structure of the Document

The remaining part of this document is structured as follows:

- Chapter 2 presents the fundamental migration concepts, from a technical as well as from a business perspective;
- Chapter 3 describes a migration roadmap for each of the SAIL technological building blocks (NetInf, OConS and CloNe) and discusses how synergies can be exploited from the combined use of these technologies;
- Chapter 4 provides a brief overview of the relevant standardisation groups and presents the main standardisation activities carried out in the framework of the SAIL project.
- Chapter 5 summarizes the main conclusions.

2 Migration Requirements and Challenges

2.1 Introduction

When deploying new technologies the overall deployment scenario, including migration aspects, needs to be considered. These cover both the technical as well as more business-oriented views.

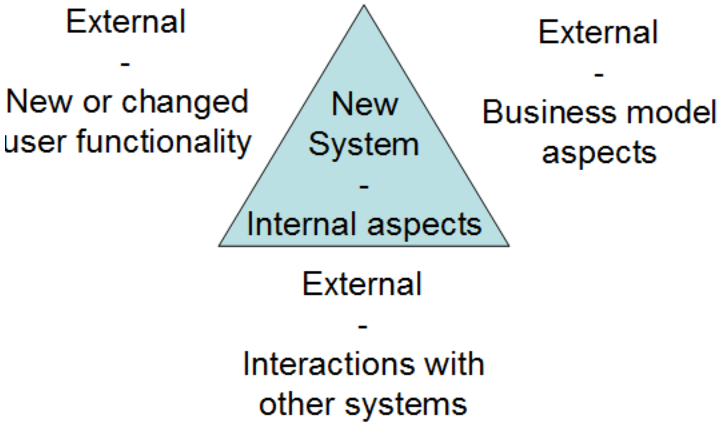

Figure 2-1 External vs internal migration aspects

In Figure 2-1 we broadly divide these aspects into an Internal and an External domain view. The borderline between internal and external is very much set by the deploying actor – the border is not necessarily inside or outside the deploying actor, but can also be within the context of the deploying actor. Two such examples from within a single actor include:

- When a New System is connected with another system at the same actor ("Interactions with other systems")
- When a New System interacts with the business aspects of the actor, e.g. by putting new requirements on the sales force ("Business model aspects")

In these two examples the internal/external borderline is within the same actor. In many, but far from all, cases the internal/external borderlines are also found outside a single actor.

The internal aspects are here meant to be very much how the actual deployment is carried out – is the system introduced as a "new, separate box", deployed either as a replacement of an existing system or as a parallel deployment, or is the system introduced more as a (software) upgrade of an existing system.

- **New or changed user functionality:** To what extent does the new system imply new or changed user behaviour?
- **Business model aspects:** To what extent does the deployment of the new system change the existing dynamics and setup around business models and B2B (business-to-business) interaction? This aspect is covered in two earlier deliverables D.A.7 [3] and D.A.8 [4].
- **Interactions with other systems:** To what extent does the new system require new types of interfaces and interworking with other systems?

2.2 Migration Concepts

We first discuss different general migration concepts from a system technology point of view that could be applied in the SAIL case.

Migration is about how to introduce new features, services or technologies into an existing system. That can be organized in different ways. We distinguish here three approaches:

- **Disruptive migration by parallel deployment (Figure 2-2):**
  Either the same or another operator deploys a new system with similar functionality completely in parallel to an existing system. The user has the choice (via the system's user interface or special devices) to select between the 'competitive' systems – migration is disruptive, mainly from the user perspective. If there is a dominant system over time, the less accepted system will be shut off.
  Example: introduction of 1st generation of digital mobile systems (GSM).
- **Managed migration by parallel deployment (Figure 2-3):**
  The operator running the existing system (System1) deploys a completely new system (System2) in parallel to the old one, with the same user interface. When the new system is available, it is the operator's choice to activate System1 or System2 for the user. The user will not be affected as long as he only requests features offered by System1. New features only offered by System2 will have to be routed to System2 exclusively.
  Example: introduction of UMTS on top of GSM with multimode mobile phones.
- **Seamless migration by incremental deployment (Figure 2-4):**
  The operator running the existing system (System1) does not deploy a completely new system in parallel to the old one, but adds or upgrades incrementally only parts of the system with new functionality or technology. This means the system will consist of components with different generation levels over a longer period. This poses the most challenging requirements on the behaviour of the new components, and needs also some technical preparation in the old system components to guarantee seamless functionality in 'mixed mode'. This issue of backward compatibility and interoperability for seamless migration is discussed in the following section.
  Example: replacement of ISDN telephony by VoIP.

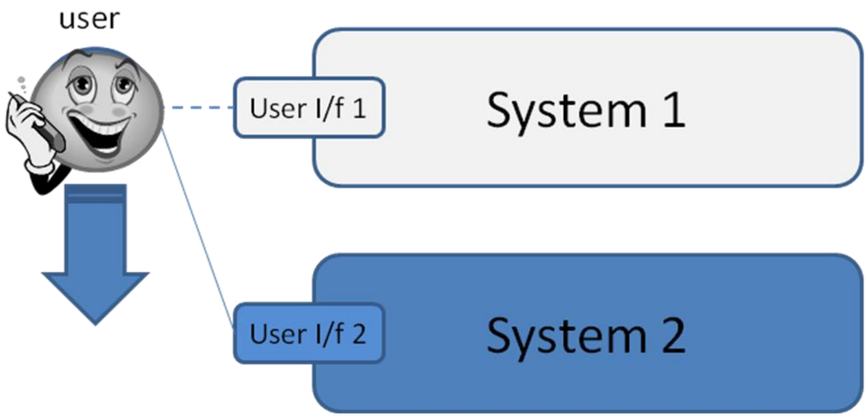

Figure 2-2 Disruptive migration by parallel deployment

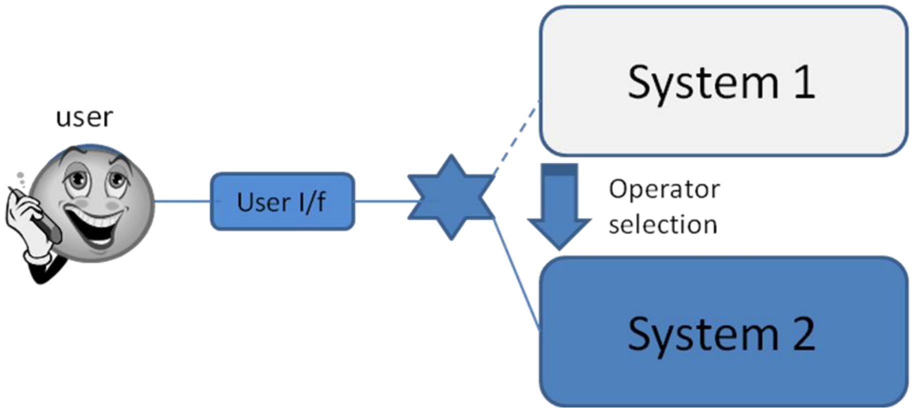

Figure 2-3 Managed migration by parallel deployment

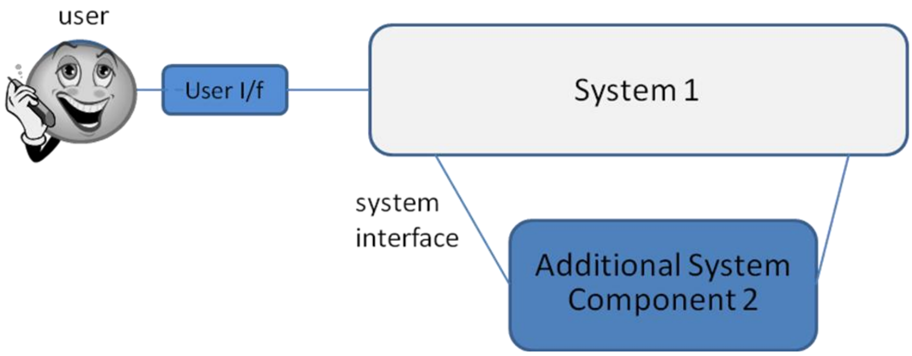

Figure 2-4 Seamless migration by incremental deployment

Technical means to achieve and mediate a seamless interoperation between components of old and new generations are:

- Introduction of a '**Convergence Layer (CL)'**:

  The convergence layer (CL) is the horizontal interface between the layers of different generations, typically when the requirements of a new generation upper layer to its lower layer are different from those lower layer services that are currently available or deployed.

  By means of a convergence layer, new upper-layer functionalities can already use new lower-layer service functions and interfaces without having the need to upgrade the lower layer immediately. The CL provides the necessary enhancements between the previous-generation lower layer and the needs of the next-generation upper layer. So the upgrade and migration schedule between upper layer and lower layers can be decoupled. Once the upper layer is completely migrated to a new functionality and original lower layer functionalities are no longer used, the CL plus old lower layer deployment can be migrated to a new lower-layer generation without changing the already migrated upper layers any more. It should be noted that a horizontal CL only involves adjacent layers; however, it affects every node in the network that needs to be upgraded with new upper-layer functionalities and therefore might be costly to deploy. Therefore, stateless CLs with local adaptation functions are preferred to stateful CLs across different network nodes.

- Introduction of an **‘Interworking Function (IWF)’**:

  The interworking function (IWF) is a method for interfacing different communication networks with similar functions, but different protocol stacks and coding standards. The IWF typically converts the data transmitted over one system into a format suitable for the other system and vice versa at a single (architecturally, not deployment-wise) interworking reference point (‘vertical dissection’). It can also adapt different protocol behaviours with or without keeping states of the involved connection. As the IWF acts as a gateway between two networks or systems, it has to decode, convert and encode the whole protocol stack up to a layer that is commonly used in both interfaced system, which might be computationally complex and costly. On the other hand, this is done in a single isolated node and does not need wide deployment efforts in the whole network. By introduction of an IWF, new system features can be provided based on the installed basis of an existing system.

2.3 Backward Compatibility and Design for Seamless Migration

Backward compatibility plays a prominent role in the discussion of migration and interoperability between system components of different technological maturity levels.

Best practices in system design might follow the guidelines and principles in **‘Design for Evolvability’** that we establish here as general requirements for SAIL (see also [5]).

Let us assume we have a Subsystem (C) that implements a functionality set of a ‘**C**urrent’, deployed generation. A Subsystem (N) now implements a functionality set of a follow-up ‘**N**ext’ generation.

In order to guarantee a smooth technical evolution of an overall system that at the same time (during the migration period) comprises components of the current generation (C) and the next generation (N), we set the following requirements for the interworking of Subsystem (C) with Subsystem (N):

1. The next-generation Subsystem (N) shall implement, at least, the functionalities of the current subsystem (C) as far as they are observable from outside (i.e., the backward compatibility).
2. The next-generation Subsystem (N) shall be able to detect whether an involved subsystem operates according to current or next-generation functionality or standards.
3. The next-generation Subsystem (N) shall interoperate with another next-generation Subsystem (N) via an interface compliant to next-generation standards.
4. The next-generation Subsystem (N) shall behave as a current-generation Subsystem (C) towards another current-generation Subsystem (C).
5. The next-generation Subsystem (N) shall exploit the new functionality only when inter-operating with another next-generation Subsystem (N) and while this does not interfere with (C) operations.

2.4 Business Models and Initial Incentives

In order for a technology to be widely deployed, there should be a clear benefit to all potential players involved (e.g. end users, content providers, infrastructure providers, service brokers). This implies the definition of one or more business models where the above condition is verified.

For SAIL technologies the business models are investigated and evaluated in two earlier deliverables: D.A.7 [3] and D.A.8 [4]. Within these documents the concept of Value Configuration Networks is applied. This allows the identification of existing and new business roles, which are of importance when a commercial actor decides how to approach a new opportunity and technology like SAIL. The potential monetary flow, on a qualitative level, and the regulatory aspects are also covered in these two deliverables.

Another essential condition for the successful commercial deployment of a new technology is that the benefit from that deployment should be clear for the involved players (mainly focusing on the end users and the return of the investment) no matter how small that deployment is. Usually, a major obstacle against the widespread deployment of a technology is the lack of incentives for early adopters or insufficient attractiveness to justify the transition.

The initial incentives apply to three groups of actors:

- The vendors: The actors that develop and resell the technology
- The service providers: The actors that implement the technology in their networks
- The users: The actors that uses the technology implemented by service providers

Particularly in the field of communications and networking, a low number of communicating users or devices often limits a technology’s utility and represents a major constraint against the technology uptake (according to Metcalfe’s Law, the value of a communications network is proportional to the square of the number of compatible communicating devices).

The need for global harmonisation in terms of regulation and standardisation can also represent a major obstacle. Several examples (e.g. the very slow uptake of IPv6) can be used to demonstrate this assertion.

3 SAIL Migration Strategy

The SAIL migration roadmap should guarantee that for each of the three fundamental SAIL technology building blocks (NetInf, OConS, CloNe) a viable migration path is defined. In addition, as the three components are expected to coexist and interwork with each other, it is also necessary to define a common overall migration roadmap. In particular, it is important to explore the potential synergies between the three technical areas to further foster further foster migration to the full instantiation of the SAIL architecture.

We notice that there are multiple migration paths for the adoption of SAIL technology building blocks. A certain migration path is motivated by its unique incentives but many issues and questions are shared among other foreseen migration paths. The migration path to choose will depend on some particular issues:

- Shared standards defining the products and features?
- What is the cost of retraining users? What is the likely cultural impact on the user community, how can it be controlled?
- What is the total cost of the migration steps and what benefits will they deliver?
- What is the cost of introducing the new products and features and what would be their payback time?

Adoption of a technology is likely to process multiple paths that strengthen each other or sometimes may even contradict. However, the cumulative benefits to the ecosystem will push the migration forward.

3.1 NetInf Migration

NetInf offers a rich tool set for various migration paths supporting different sets of incentives to various players in the ecosystem. These migration tools, which pertain to the Architecture Invariants and lead essentially to the notion of convergence layer and ni-naming scheme, are documented in D.3.3 [10]. Here we discuss migration from business case and incentive perspective relating them to key NetInf enablers. These technical enablers of migration are covered in D.3.3.

NetInf deployment will be driven by market decisions, depending on the incentives of the different actors such as Content Provider, Access Network Provider and Inter-Connectivity Provider (see D.A.8 [4] for a description of these roles).

We will discuss the previous questions in the earlier section for each identified migration path in the next sections from the perspective of NetInf. The different migration paths irrespective from their original starting points have common segments and merge latest when NetInf becomes visible across inter-operator interface.

3.1.1 NetInf Disruptive Migration by Parallel Deployment

This approach corresponds to the simplest one for initial deployment of “NetInf islands”. But we must make clear who the user is as there are multiple interfaces: it could be either the end user asking for NetInf content or the Content Provider publishing specific content for these NetInf island that are parallel to an existing system as defined in section 2.2. This excludes the network operator from the consideration, as there are no clear initial incentives for an operator to create a disruptive and parallel deployment for content delivery. Therefore we focus here only on end-user driven and content provider driven migration paths and leave network operator considerations to later sections.

In an end-user driven migration path, the starting point corresponds to how peer-to-peer applications were adopted to share content between the peers. Here the interest of content sharing is by the end users, the content provider and network provider are in the role of externalities. Minimally an end user needs an application or browser plug-in (e.g. OpenNetInf) that uses an information-centric naming scheme, most preferably ni-naming scheme and NetInf messaging. In fact, this case is part of the NetInf "Event with Large Scale" scenario that has been implemented [9] [10]. The cost of the migration path is related to the distribution of the needed application or plug-in as well as the setup of the NetInf infrastructure. Benefits are the same as with peer-to-peer applications but with more efficient and secure content delivery, thanks to inherent NetInf capability to support multicast and getting content from multiple sources. Moreover, from the network operator point of view, there are other benefits such as traffic engineering, reliable storage, name-data integrity, integration into topology, that are not typically derived from the peer-to-peer systems. This migration starting point is likely to be limited to infrastructure-less cases where all needed NetInf functionality (e.g. peer discovery and name resolution) is provided by the participating peers. The applicability of this migration path is limited to specific technology domains such as unlicensed radio technology like WLAN and Bluetooth or home networking to avoid tussles with the network operator (as has happened with peer to peer technology). Therefore, the business impact of this migration starting point is also limited. The next step in this migration would be to get infrastructure support for the name resolution, Convergence Layer, content routing and caching which widens the participating actors and the potential business impact. At this stage, standardisation of the NetInf functionality starts to be critical.

When the Content Provider triggers the migration, it corresponds to the case where no adaptation has been developed to ease content publishing but the CP has to use specific APIs and protocols to interact with NetInf, in the same way as what is used today for publishing towards a specific Content Delivery Network. A CP may try to build its own overlay for NetInf content delivery with name resolution and caches. However, it is likely that a third party similar to the current CDN operators will start to offer NetInf as a service sharing the cost among multiple users. This is because of the need for in-network caches to achieve the delivery efficiencies which requires also agreements with the network operators or a peer to peer tracker like sever network. The CP benefits from requiring NetInf support from a CDN operator are related to content publishing services, secure naming with associated metadata and to efficient content delivery (due to inherent multi-source, multi-cast nature of NetInf) to improve QoE for the end users. Secure naming and use of metadata provide a platform for efficient searching, authenticating, authorizing, monitoring and charging for delivered content. Specific management applications on top of NetInf need to be developed to leverage these advantages in addition to a client-side application (e.g. OpenNetInf plug-in) for commercial viability. The cost of this initial migration step would be comparable to any CDN system. A differentiating factor compared to CDNs is in the envisioned (management) applications that can take full advantage of ni-naming and the NetInf features (see also Network Management Key Performance Indicators in D-3.3 [10]).

No immediate need for a standardised Convergence Layer is seen to allow NetInf to use current transport and protocols in the above overlay scenario, because only a NetInf-dedicated transport would be sufficient as the communication takes place in a closed group of the CP and its clients and through a CDN provider. This is also evidenced by the minimal standardisation of the current CDN concepts and peer to peer systems that do not require special standardised convergence layers.

The GIN approach [10] of NetInf goes along the Content Provider-driven, disruptive migration similar to CCN/NDN [12]. Both of them require introduction of special content-based routers and caches adding a cost factor to this migration.

The above-mentioned migration starting points may be viable only as a short-term low cost initial deployment of NetInf. Having some interfaces allowing the user to transparently access NetInf content would be more desirable, as explained in the next section.

3.1.2 NetInf Managed Migration by Parallel Deployment

A likely starting point for this migration case is a local cost optimisation of Access Network Provider similar to the case of deploying transparent caches. A key motivator is the projected traffic growth dominated by the video content that consumes significant end to end network capacity. NetInf offers a full set of tools including backwards compatible/well-known URL in the ni-naming scheme [13,10] and capability of Niproxy [10] and NNRP [10] to translate HTTP requests into NetInf messages useful for this migration path.

A major question is if the Access Network Provider would reuse its existing name resolution infrastructure. There is a huge infrastructure dedicated to name resolution in the current Internet through the DNS system. In NetInf, the first point of connectivity for a node is a Name Resolution System (NRS) node. In the previous approach (disruptive migration) we do not interact with the DNS, as the name resolution for new content in NetInf uses its own Name Resolution that can grow as more ICN content is deployed in “NetInf islands”. Bypassing of DNS is done by giving a node a default route to the NRS, for example through DHCP.

If we want to take advantage of the DNS infrastructure, one way of accomplishing this would be to use an alias in the DNS to point to a nearby NRS as it is done today (for example through the use of views in Bind) within Content Delivery Networks. The initial NRS deployment can be simple, incremental and the cost controlled. For example, for NRS scalability, it can start with a few resolvers on the same level of MDHT (region) and then add resolvers and MDHT levels as the regions will be inter-connected as the number of network objects start to grow.

From a network perspective, having both systems in parallel means we can take advantage of legacy networks and protocols and use them to transport NetInf content: this is the way NetInf has been conceived with the Convergence Layer. The Convergence Layer is a mediation between NetInf and current (and future) internet (or other) protocols and allows the communication of two NetInf nodes independently of the lower layers. Convergence layers have been described for HTTP (unicast communication), UDP (multicast communication) and the DTN bundle protocol.

For any approach, it is necessary to maintain compatibility with legacy infrastructures, but also compatibility with legacy content. In a first step, the NetInf naming scheme allows for the use of URLs pointing to existing content. If name-data integrity is needed for legacy content handled in new applications, then the content should be republished using URIs and the NI naming scheme. The Access Network Provider can do this locally.

The migration that starts as local cost optimisation grows incrementally to an inter-operator solution with -name based routing and content peering (see also D.3.3 [10]). Inter-operator solution requires an update to the BGP routing scheme (as described in D.3.3 [10], routing hints) that could be handled with BGP to enable inter-domain routing.

In addition, new troubleshooting tools and methods need to be developed before any commercial deployment. It is relatively easy today to trace connectivity issues in the client/server architecture, but for NetInf, troubleshooting should go through Naming (is the right name/hash used), Name Resolution (are there locators for the content, is it still cached), Convergence Layers, content routing etc.

The success of this migration path – as any other one crossing the operator boundaries – requires thorough standardisation of naming schemes, name resolution, content routing, and the inter-operator interface that can be based on CDNi work in progress in the IETF, see standardisation in Chapter 4. Some of the needed standardisation has already started but potentially needs a few more years to reach sufficient maturity level (Chapter 4).

When considering the cost of this approach, we note that in addition to the cost of new network elements (caches, NRS, etc) there is also an associated cost related to mind set change to re-educate the operations personnel from the familiar client/server approach to an information-centric approach. Similarly, for the Content Provider, keeping control of the published data is different, for example unpublishing may be challenging or even impossible, as the sources for a popular content are numerous and there will not be a unique place from where to remove locators for the content. The cost of hardware for caches and NRS nodes is easy to balance with revenue and savings in operation the operational costs. However, the scaling of NRS is more difficult to evaluate as it is a common resource to all actors and the load depends mainly on the use by the end user. The inter-connection of the NRS of different ANPs (Access Network Providers) will need an upper layer in the domain of the Inter-Connectivity Provider that necessitates wide collaboration among the NetInf enabled operators.

3.1.3 NetInf Seamless Migration by Incremental Deployment

The previously discussed managed migration already contains aspects of incremental deployment of NetInf as the Convergence Layer is the primary means to interface to legacy networks and can be deployed incrementally. The transitioning of the transport layer towards a NetInf receiver-based transport can also be done incrementally, resulting in better performances, multi-path and caching (cp. Interest Control Protocol in D.3.3 [10]). Protocol translation functionality, as exemplified by Niproxy and NNRP implementations that translate HTTP requests into NetInf messages are needed to connect pure NetInf islands with legacy services. The translation between the URL and ni-naming scheme is accomplished through backwards-compatible, well-known existing URLs, which is part of the ni-naming scheme.

D.A.8 describes the evolutionary steps for business model adaptation strategy [4]:

1. Internal Network optimisation,
2. Transparent caching,
3. Telco CDN: Providing commercial CDN services to content providers,
4. Telco CDN with Content Delivery Network Interconnection (CDNI): Extending the footprint of the Telco CDN by interconnecting with other (Telco) CDNs.
5. Virtual CDN: Outsourcing the customer relationships with content providers to a virtual CDN provider,
6. Elastic NetInf deployment.

Clearly, the first two steps are local improvements and by their very nature must be seamless to existing business models and their supporting technologies, such as accounting. Step 3, where the service provider handles QoE and SLA management, can be implemented as disruptive or seamless depending on how much the end-user protocol stack will be updated. If a proxy at the access network is used to translate HTTP messages into NetInf messages with ni-naming scheme, the end-user protocol stack is not affected and this step can be implemented as managed seamless migration. Disruptive management with parallel deployment can be implemented with a client-side plug-in or an application where NetInf is implemented as an overlay, as explained earlier. Seamlessness is achieved by either translating proxies or Inter Working Function (IWF) even in this case as well. Seamless migration to NetInf requires that the protocol, its translation and/or convergence layer adaption must be standardised to avoid vendor lock-in and proprietary IWFs. These efforts are already in effect, see chapter 4 for details. Steps 4 and 5 require multilateral adoption between different actors of the value chain with a federation between NetInf domains and possibly between legacy CDNs. To reach this point some further work is still needed, especially for the accounting and charging, which will be based on a new value chain, with the content being priced instead of only the delivery. For example, the model for content level peering needs to be defined. See D.B.3 [10] for a description of this new peering model and how transit revenues will be impacted.

Finally, step 6 does not require collaboration with other ANP, but a customer relationship to a CloNe operator. It is then not solely a NetInf migration; see Section 3.4.2.

3.1.4 Conclusion of Migration to NetInf

We can use history to foresee what may come in the future: Internet in its infancy was first deployed over existing telephone lines designed for voice. As its value was realized and the technology was adopted, Internet got its own connection and resources. Now as Internet has matured, it is used to transport voice reversing the scheme we had originally.

In the same manner, this is likely to happen with NetInf and IP. First NetInf is deployed over IP and as the adoption and value of NetInf increases it will get its own network and eventually all the services currently deployed over IP will be migrated to (evolved) NetInf. For example all massive user generated content accessed nowadays from a well-known site, could be much more efficiently published, cached and delivered by NetInf taking advantage of multiple nearby locations with cached content.

Table 3-1 Summary of NetInf migration paths

NetInf adopter	Initial Incentives	Benefits at the end	Category (ref. sec 2.)
End user community	Content sharing, similar to p2p	New domain specific application, efficient delivery, content security	Disruptive migration by parallel deployment
Content Providers	Efficient content delivery similar to CDNs, better control over content	Potential for content federation between other CPs, content level peering	Disruptive migration by parallel deployment
Network Operators	OPEX, CAPEX savings, similar to transparent caching, local content delivery services	New business models, content level peering, CDN functionality integral part of infrastructure	Managed migration and seamless migration by incremental deployment

3.2 OConS Migration

OConS provides an innovative way to define connectivity services in the Internet. One of the main prerequisites of the SAIL project is that any solution follows an evolutionary, non-disruptive path to speed up its adaption and deployment. In this spirit, OConS foresees an evolution of the underlying network technologies.

OConS provides a variety of mechanisms in three levels: link, flow and network. These mechanisms are designed according to the OConS common architecture and open interfaces, so that migration can be reached either by including one of them at a specific level, or by a combination of them using Orchestration functionality. The benefit that Orchestration can bring to the take-up of these (and other) mechanisms can be summarized as follows:

- It defines a comprehensive system ranging from bootstrapping to network operation monitoring;
- It registers multiple available mechanisms (legacy or not), their functionality and combination rules for them in order to facilitate and simplify the migration;
- It is aware of entities and their capabilities (services/mechanisms available);

- It monitors activity in order to take/suggest network decisions for a better performance or QoE;
- It can coexist with legacy deployments (not OConS aware) without affecting inner domain performance.

3.2.1 OConS Disruptive Migration by Parallel Deployment

A disruptive deployment of the OConS architecture would create a network domain where only OConS mechanisms are applied. Thus, it would not be aware of alternative legacy mechanisms outside its domain. Orchestration could neither monitor network state changes if legacy mechanisms are operating, nor compare alternatives between legacy and innovative solutions when it decides how to react to changes that have been detected.

An example of this situation could be the following:

- A Delay Tolerant Network (DTN) where people use their smartphone to share real-time content (i.e. photos or videos recorded on site) is established.
- The user application for content sharing is running on top of an open connectivity service developed within OConS (e.g. making use of OConS DTN routing based on the HURRY protocol, use network coding in network realms where reasonable to use it, etc.), which is registered with its specific capabilities and possibly associated with certain functional rules and/or under certain network conditions.
- A new user with an OConS-enabled smartphone (i.e. a smartphone that implements DTN routing) joins the network:
    - The user application can register and share content with other users. OConS orchestration would allow it to run the same application over different connectivity services available depending on the terminal and the network conditions.
- A new person with an OConS-unaware smartphone tries to join the network:

Since the smartphone does not implement any OConS routing mechanism, the user application would not work properly. It would not be possible for that specific user to share content with others. OConS Orchestration would not be able to allow the user application use different connectivity services.

3.2.2 OConS Managed Migration by Parallel Deployment

If migration is designed as an inclusive path, the key objective is to embrace the existing connectivity mechanisms to be mapped onto the OConS architecture as a totally functional alternative. Each mechanism would need to implement a minimum set of actions to perform the registration process and to accept and execute the decisions taken by Orchestration during runtime operation of a service. Any service should be developed so as to be running in a seamless way regardless the connectivity protocols actually running underneath.

As opposed to the example described in the previous section, an inclusive approach for the same situation would result in:

- The new user application would try to run using a connectivity link operated by a legacy mechanism. This request would be received in an OConS-aware node (operating OConS DTN routing, implementing network coding, etc.), which would trigger a query towards the Orchestration.
- Orchestration could analyse this request and evaluate the connectivity options available, in the form of a cost function, for example:
    - Maximise the probability of success to establish a connection and exchange routing information

- Maximise network resilience using network coding Orchestration could enforce the use of a legacy connectivity protocol in both edges of the communication link (users' smartphones involved in the request), since it is the minimum common suite available to reach success in the connection establishment

If a connection change occurs, and no legacy edge is longer required, Orchestration could enforce going back to OConS specific mechanism, if its benefit is still applicable for the user application/service.

3.2.3 OConS Seamless Migration by Incremental Deployment

OConS can also be seen as an enabler for network evolution in a more general sense. Current technologies in the wireless and mobility fields can be enriched by OConS mechanisms like DTN, DMM, network coding and routing technologies that can be enhanced with network-wide orchestration. This can be exemplified by the migration of the core network to embrace OConS technologies, especially with network control in mind:

Today's ecosystem in the core network consists of a mixture of the following technologies:

- IP as convergence layer (both IPv4 and IPv6)
- MPLS as a way to provide a unified control plane for advanced services over evolved IP infrastructures. Services provided using MPLS include, for example:
    - L3VPN and Internet access
    - L2-VPN

Looking at the network control layer, technologies like GMPLS and the Path Computation Element (PCE) [8] can be found. These technologies allow network operators to integrate the management of the different layers of their network (e.g. light-path management) implement different Traffic Engineering (TE) technologies that facilitate their coordinated use (e.g. IP traffic offloading) [15].

In the medium term, OpenFlow shows promising features that make it an adequate candidate for the access, resulting in a scenario where the core network is based on MPLS/GMPLS and the access network is based on an enhanced OpenFlow.

In this scenario, orchestrating both network realms provides a feature-rich and integrated network. The orchestration mechanisms provided by OConS can play a crucial role in providing this integration and the interworking between both network realms.

A phased approach to the deployment in the core network would include following phases:

Phase 1: Deployment of different core network technologies in islands. Interworking between the islands is implemented in a static or semi-static manner. The OpenFlow-enabled island in the network core is configured statically with a limited set of rules that allow interworking with the 'legacy' IP/MPLS core. Network elements implement the different components of the OConS architecture. However, these elements are currently configured statically. Time wise, some elements will be ready to be deployed soon.

Phase 2: Deployment of OConS orchestration within the different technological islands. The OConS orchestrators deployed operate elements within their technological island. Traffic rules at the borders are honoured; however, the orchestrator allows the use of different OConS mechanisms that optimise resource usage, traffic flow, etc. within the different islands. This phase should start once a critical mass of OConS-able nodes is deployed and the first orchestrators are implemented. According to current estimations, this phase would start between the beginnings of 2014 and the end of 2015.

Phase 3: Interconnection of the different technological islands. The final phase would imply the activation of interworking mechanisms between the different islands, allowing a progressive end-to-end control within an operator domain. First trials should be possible after Phase 2.

3.3 CloNe Migration

CloNe proposes the deployment of distributed services across multiple providers in an automated fashion. Specifically, cloud networking is about integrating the on-demand provisioning of network resources to data centre resources, allocating both computing and networking resources at once. This allocation and configuration of resources across different providers is automatically performed by the system. In that way, existing data centre-based IaaS can be interconnected through the wide-area network using a network service model.

Since CloNe's vision relies on a multi-provider deployment, migration and coordination aspects have to be carefully understood. As such, the deployment of CloNe services should not demand all operators to deploy the same services at once (no flag day). Early adopters should be able to sell their services before a worldwide deployment has happened.

Referring to the migration approaches described in chapter 2, Clone migration is feasible in parallel deployment. That means the DC and network providers will be running their existing system (System1) while deploying a completely new system (System2) in parallel to the old one, with an evolved user interface that includes all the necessary fields for the same user interface. When the CloNe-enabled system is available, based on the type of demand that the provider receives, it can choose to activate System1 or System2 for the user. The user will not be affected as long as he only requests features offered by System1.

In the following sections, different aspects of CloNe migration are described. CloNe can follow a seamless migration by a stepwise incremental migration strategy. The migration can be broken down to multiple phases which are undertaken by different providers based on their individual plans. Providers can continue to use their old system in parallel while migrating to the new system. This approach allows them to integrate new business models without any loss on previous system.

3.3.1 Seamless Migration by Incremental Deployment

To simplify prototyping and ensure an easier migration path, CloNe has built on and extended existing accepted cloud management tools, interfaces, networking protocols and libraries when those were available. Moreover, legacy (e.g., VPNs creation using the Operator Network Management system) and upcoming technologies (e.g., OCons or OpenFlow for control over network elements) are seamlessly integrated in the architecture through a network abstraction layer. Thus, the most natural migration path for CloNe technologies is through *incremental deployment*. Some general principles that CloNe has adopted throughout its development to ease migration were:

- CloNe does not rule out specific business models (the Distributed Infrastructure Service can be implemented by a third party or by existing players – IaaS and network providers)
- CloNe envisions loose coupling between different service providers. CloNe is agnostic to how infrastructure is implemented internally in a data centre or operator network. An example is the use of different cloud management systems (OpenStack and OpenNebula).
- CloNe is both future-proof and compatible with the wide range of currently deployed network technologies. This is essential to enable technology migration and facilitate interoperability.
- The CloNe eco-system is enabled through service interfaces (OCCI/OCNI) and protocols (DCP protocol suite). For CloNe to succeed those interfaces need to be standardised or become a *de facto* standard (e.g., through open source)

Based on the above stated principles and technology developed in CloNe, a stepwise incremental migration strategy was devised where early adopters can start selling their services regardless of others. The full vision of distributed infrastructure services will be enabled when the last phase of migration has been introduced. Overall, the proposed phases of migration are:

- Phase 1: Operator networks offering flash network slices. This phase alone can transform rigid and lengthy virtual network provisioning to a dynamic, on demand and flexible network provisioning.
- Phase 2: Data centre providers enhance their services allowing for more advanced on-demand networking in data centres as today there is no network as a service offered inside the Data centres.
- Phase 3: Operator networks and data centre providers implement DCP (Distributed Control Plane) to connect both services above, providing an end-to-end solution.
- Phase 4: Distributed Infrastructure Services can be implemented amongst service providers who have implemented phases 1 to 3 of migration.

The entry criteria and expected deployment time frame for each one of the phases are indicated below. The indicated dates should be taken as a rough indication, for illustrative purposes only:

- Phase 1: The operator must own a network and have full administrative and operational rights over it. That network shall support traffic separation and provisioning of traffic engineering for guaranteed services. It is expected that this phase of migration would start between 2013-2014.
- Phase 2: The entry criterion is for the data centre provider to possess an extensible and open cloud management system where network services can be easily added. This migration phase has already started in 2012 (e.g., Quantum network services becoming a core project in OpenStack).
- Phase 3: Operators willing to cooperate must have implemented phases 1 and 2 of the migration strategy. This phase is expected to start between 2014-2015.
- Phase 4: Dependent on phase 3. This phase is expected to start in 2015-2016.

3.3.2 Providing Flash Network Slices (Phase 1)

In the first stage of migration, network operators should start offering network connectivity as a service. CloNe provides that through flash network slices, which are a guaranteed network resource that can be deployed or reconfigured in the same timeframe as the creation of computing resources in existing datacentre-based clouds.

Since CloNe should be future-proof and compatible with currently deployed network technologies, CloNe has to be agnostic to specific characteristics of the underlying network infrastructure. This implies a clear separation between technology-independent and technology-dependent functions, which has been introduced in D-5.4 (D-D.3) Refined Architecture [11].

The idea behind the separation of functions is that different technologies can be incorporated in the architecture simply by creating modules (or drivers) adapted to the specific network technologies. The separation is done through a network abstraction layer that hides network complexity from the customer of the network service. The separation has important advantages: CloNe is able to make use of a high number of existing network technologies, without the need to create specific ““ad hoc” solutions; new technology can be accommodated simply by building the corresponding module; finally, by using common reference points, deployment of new technology can be backward-compatible and interoperable with legacy technologies, i.e., a massive upgrade or replacement of equipment is not required (no flag day).

One instantiation of the network abstraction layer that was implemented in the prototyping activity was libNetVirt. If more than one type of FNS (Flash Network Slice) is to be provided (e.g., L2 and L3) and a unique network management system that abstracts the different types of network is not in place, the operator may deploy libNetVirt to simplify the management of its own networks and hide away network complexity from the user. That network abstraction layer supports the separation of technology-dependent (network drivers) and technology-independent (the northbound library interface) functions.

In order for the operators to offer network services, a suitable network service interface needs to be implemented and deployed. For that purpose, CloNe has devised and implemented OCNI (Open Cloud Networking Interface). Through OCNI, a RESTful interface is exposed to customers for the creation and management of network resources. Thus, the OCNI interface should be implemented and connected to the internal implementation of the FNS (libNetVirt enabled or not).

3.3.3 Providing On-demand Advanced Datacentre Networking (Phase 2)

When examining the current state of technology of data centre networks, one will notice that so far, only limited networking options have been provided (e.g., flat layer 2 networks with DHCP or VLANs). If complex virtual infrastructures are to be deployed in data centres, other types of network services must be supported. In fact, as of the writing of this document, OpenStack is moving in that direction as well through the Quantum API. CloNe proposes the deployment of complex application topologies (e.g., 3-tier applications) supported by the proposed architecture. In the architecture, existing IaaS interfaces are enhanced with more advanced networking capabilities.

The provisioning of networking within the datacentres also happens on-demand. Within the datacentre, different network solutions may be used and that is left to the provider to decide. In the prototyping activity, CloNe utilized different solutions, e.g., VNET, OpenFlow through libNetVirt, and routed network services (e.g., Quagga) providing the same type of service through different technologies. This is optimal for migration since it doesn't impose a restriction on the type of technology data centre providers shall use.

Nevertheless, the interface utilized for deployment of those network services integrated with computing services needs to be agreed upon. In this context, CloNe proposed the use of OCNI to be implemented by data centre providers. Another alternative is the use of VXDL for the description of a complete complex virtual infrastructure that can be deployed by the datacentre provider. As both of them are functionally equivalent, either could be used.

3.3.4 Implementing a Distributed Control Plane (Phase 3)

Once phases 1 and 2 of migration have been performed, both datacentre and network providers are ready to integrate their services. That integration will allow on-demand creation of WAN resources, connecting datacentre infrastructures. To enable this, CloNe developed the Distributed Control Plane (DCP) protocol suite.

Initially, a framework for exchange of messages among providers needs to be put in place. Due to the asynchronous nature of this communication, a brokerless message bus mechanism was designed, the Cloud Message Brokering Service (CMBS). CMBS allows inter-provider communication to happen in a structured way, based on topics of interest or a selected group of receivers. CloNe has designed CMBS to carry other DCP protocols (e.g., the Link Negotiation Protocol).

The Link Negotiation Protocol (LNP) should be implemented to negotiate L2 and L3 parameters for inter-provider links. It is implemented on CMBS. Providers can start using LNP for agreeing on links, encapsulation methods, routing and addressing schemes, and more. A

Reference Resolution protocol may be deployed, easing late binding of resources across different providers. Please refer to D-5.4 [11] for details.

In order to go through this phase, probably no worldwide deployment is feasible, nor is it necessary. It is in fact possible by incremental alliances between DC providers and network providers, between DC providers for the purpose of delegation, and between network providers for extending the network coverage to a greater distance. Obviously, more coverage provides better service to the users and creates attraction for extending alliances.

However, this remains an important phase in the migration process, as it will enable cross-provider deployments, which are at the heart of CloNe.

3.3.5 Enabling Distributed Infrastructure Services (Phase 4)

Last, in order for different providers to automatically find each other's services and to announce capabilities, a Discovery Protocol [11] must be implemented. The discovery mechanism is part of the DCP suite, as well. If not implemented, manual agreements can be made across providers that are willing to cooperate and share the deployed infrastructure.

When full integration of data centre and network services has happened (Phase 3), a virtual infrastructure specification language can be used for the customer to specify its infrastructure requirements. CloNe proposes the VXDL language to allow clients to specify distributed infrastructure requests with support for elasticity both in the network and in the computing/storage levels.

Each provider that is part of the CloNe ecosystem shall implement the Decomposer function. That function breaks down resource requests onto smaller pieces of infrastructure, which can then be passed to other providers for deployment. That is done via the Delegation mechanism, whereby the Distributed Infrastructure Provider uses interfaces implemented in Phases 1-3 to request resources across multiple providers.

In that context, a delegation-aware authorisation service must be implemented and deployed to allow customers and providers to authenticate each other.

Although this phase is described last, it can start as soon as there are resources that are distributed over multiple domains, multiple providers, or even the same provider that for technical requirement has distributed its infrastructure.

3.3.6 Business Considerations

CloNe proposes an incremental migration strategy towards fully distributed infrastructure services. From a network operator perspective, the dynamic network services proposed by CloNe should enable new businesses and revenue income. Those revenues will be available to early adopters (Phase 1) before the full migration roadmap has been completed; those network services can be used to interconnect geographically distributed applications running in different datacentres (by calls made to the exposed network service interface, e.g., OCNI).

There is little impact on the businesses of existing data centre providers. The relationship with tenants is not changed, nor the pay-per-use business model currently applied by most of them. However, new business relationships need to be established to enable multi-provider distributed infrastructure services. Those relationships shall be established with other datacentre and network providers. Charging models that fit the delegation concept should be designed for that purpose.

Another important aspect for deployment of new technology is the return on investment (ROI). We believe that the deployment of CloNe has high potential ROI: a.) due to larger utilisation of networks through virtualisation and new services provided; b.) due to new business agreements and delegation where a provider may charge for being a broker; c.) due to low hardware investment costs since large part of it is already deployed (datacentres and networks); d.) due to low software investment costs since a large part of it can be obtained as open source (e.g., OpenStack).

3.4 NetInf/OConS/CloNe Migration Synergies

NetInf, OConS and CloNe coexist and interwork with each other; therefore, it is natural that synergies can be exploited to make the migration path more attractive still. **Error! Reference source not found.** shows potential scenarios for interworking between the three main technical areas. For example, CloNe can rely on legacy technologies, while it is also able to make use of enhanced network functionalities provided by OConS. In a similar fashion, NetInf can be supported by legacy technologies and also take advantage of functionalities provided by CloNe and OConS, or both simultaneously.

An analysis of migration synergies should focus both at system level aspects (synergies between different systems used to facilitate/accelerate migration) and at migration process level aspects (synergies between different migration processes when performed together). The following subsections identify a number of synergies to facilitate migration from the combined use of NetInf/OConS, NetInf/CloNe and CloNe/OConS when more than one technology is deployed at the same time.

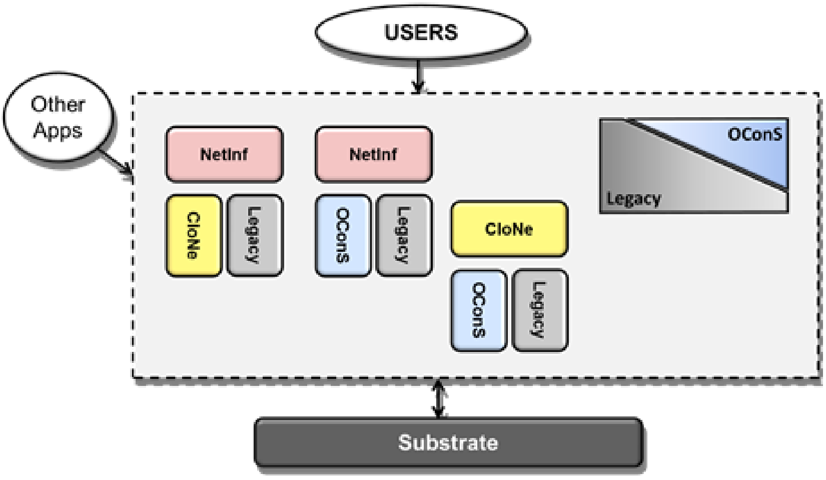

Figure 3-1 SAIL overall migration scenarios

3.4.1 NetInf/OConS

Ideally, NetInf/OConS synergies are mainly focused on three specific OConS mechanisms that could help the adoption of NetInf and improve or extend its functionalities over heterogeneous network topologies in the near future:

- Multipath extensions for Information Centric Networks: there are a number of multi-path strategies that a NetInf node can adopt based on the information supplied by the OConS framework. These strategies are defined as rules. These rules are evaluated continually to select the appropriate multi-path strategy following a content request and delivery over the multiple attachments
- DTN routing based on social routines: the combination of Bundle Protocol Query (BPQ) extensions to the DTN suite with the OConS routing strategy for DTN using the history of social interaction among individuals would benefit the adoption of NetInf concepts over an opportunistic network topology. The BPQ extension block introduces the NetInf publish/get model into the DTN common operation. The relevance of social routines, here applied to opportunistic routing, helps the integration of the user activity

in legacy social networks, or legacy communications, with the ad-hoc multi-hop basis of a DTN path establishment.

- Enhanced access selection in heterogeneous networks: the main focus is done at a flow level (a decision is taken upon a service request – by NetInf in this case – a connectivity service for a flow); network level can be also considered when connectivity is required even without ongoing services. This enhanced access selection guarantees improved QoS/QoE, considering current context and service requirements, possibility to interact with other OConS (or legacy) mechanisms, which might be considered by the OConS Orchestration if need be.

The OConS extensions for NetInf come in the form of new forwarding functionalities included in the NetInf architecture. The OConS-based NetInf convergence layer carries the majority of these functionalities. The migration path in this context consists of a phased approach to adopting the new OConS functionality.

Without OConS, the NetInf architecture depends on legacy transport mechanisms to request and retrieve the content over the networks. OConS functionality improves user experience for NetInf users. An example is the use of multi-path content retrieval. In NetInf-enabled networks, multiple copies of content may exist in the network for a given content. Furthermore, NetInf nodes are equipped with multiple network attachments. When OConS is deployed in such an environment, it is able to assist NetInf to retrieve the content in the best possible manner by using the OConS framework components to decide the best possible multiple paths and NetInf caches to obtain the content.

There are a number of different convergence layers that NetInf currently uses which work over legacy technologies (e.g. HTTP CL). When migrating NetInf to use an OConS based CL, the first phase involves the co-existent approach where different types of CLs can operate along side each other. By doing this, NetInf will continue to have the benefits of both of the worlds (legacy and OConS). For example, with the OConS multi-path extensions, NetInf can be made to use multi-path functionality for selected content or networks, leaving the rest for the legacy CLs to retrieve content. Once an understanding is built on the suitability of the different CLs, in subsequent phases, NetInf can be configured to use these different CLs (including the OConS CL) based on the experience gained.

3.4.2 NetInf/CloNe

In this section we outline our vision for a global NetInf deployment. We start out explaining the benefit of using cloud resources for hosting NetInf overlay nodes. While this is a first step, several issues remain, which are addressed by CloNe. In the second part we discuss the benefits of using CloNe's approach to managing cloud resources when deploying a global NetInf overlay.

Cloud Computing brings NetInf closer to the network core.

A "pure" NetInf approach[1] requires many functions that are not provided by the architecture and infrastructure of the current Internet. Missing functions include support for information object routing and addressing (instead of end-device addressing) as well as in-network storage and computation. Ideally NetInf would be supported not only by the end-user devices, but also – and even more importantly – by all network devices connecting the former. Internet routers would have storage and use the NetInf protocol to direct requests, process responses and maybe cache information objects.

[1] We consider a "pure" NetInf approach one where all network elements in the network are NetInf enabled, that is, they are able to process, maybe cache, aggregate, and forward NetInf requests and objects.

The routing and addressing issue can be addressed by forming a NetInf overlay on top of the current Internet. Routing and addressing overlays are well-established mechanisms to enable new networking paradigms. For example, P2P systems implement their feature by forming a P2P-protocol aware overlay of all the participating nodes in the network on top of the Internet. Using one or several overlays, they provide means to search and transfer content.

However, the in-network capabilities are harder to address. Today most network overlays, which are typically introducing new routing approaches, are composed of the set of end-devices of currently active users. Those are typically at home or at work and may be switched off at any time, which translates into an overlay composed of many non-reliable end-devices attached to the Internet via low-bandwidth, high delay network connections. This contradicts the "in-network" concept of NetInf, i.e. it does not help to cache content on nodes that might be unavailable the next second. Cloud computing in general, but in particular the multi-provider/domain and resource allocation features from CloNe, can remedy these shortcoming to a certain degree. Moreover, compute and storage cloud resources, i.e. data centres, are often placed close to the core of the network in order to provide good accessibility and high performance (bandwidth, latency).

Although cloud resources are not available on every network device, they are more conveniently placed than an end-user-only overlay and thus enable to leverage the potential of NetInf. It is much more likely that a network operator will deploy several well connected data centres, which it might use also for non-NetInf service, rather than upgrade all its networking equipment with NetInf functionality. As a first step towards a global NetInf deployment, we propose to include cloud/data centre resources in a NetInf overlay. Cloud resources can provide always-on, well-connected, close-to-the-network core NetInf nodes.

CloNe provides even better support for NetInf

Using today's cloud offerings, it not possible to specify service requirements like "delay and bandwidth towards a certain end-host" or "place this node in Bristol or Munich". Also, building a global network of cloud resources requires managing deployments across multiple cloud providers. Even harder, some cloud providers require you to micro-manage deployments across multiple data centres. However, to take full advantage of NetInf, utilizing data centres across several network operators, and even better across the whole globe, is desirable.

Since CloNe provides the ability to provision cloud resources across different providers and across multiple locations, the resulting overlay for NetInf on top of CloNe can create a global NetInf overlay of many, well-connected, geographically wide-spread nodes that offer both computing and storage.

The other benefit of using CloNe to provide resources to NetInf is CloNe's adaptive resource provisioning (adding/removing/migrating resources). It allows changing the NetInf deployment according to actual usage and environmental conditions. For example, when the active NetInf users change and the new set of users is requesting content at a different location, CloNe can detect this demand change and provision additional NetInf caches close to the requesting users. Likewise, the now vacant NetInf nodes can be stopped. This leads to another advantage of CloNe: the resource allocation algorithm can be tailored to the special requirements of NetInf, e.g. an old but unused cache can be kept alive until all its content has been replicated to a newly spawned cache. Without this feature the cost for deploying a global Network of Information (NetInf) will not scale with its usage, especially in the beginning at low load levels. Thus without the need to invest too much money upfront the NetInf approach can grow to attract users and raise operators acceptance.

These benefits are demonstrated in the Elastic NetInf deployment SAIL cross-WP use-case (refer to D.A.9 [17] for details) where we dynamically scale the NetInf deployment over multiple CloNe-controlled testbed data centres while varying the request load to the NetInf system. Using a dedicated component (called ADT in the use-case) we tailor the resource allocation to the KPIs measures by the NetInf nodes, highlighting the customizability of CloNe toward specific services running on top of it.

3.4.3 CloNe/OConS

OConS and CloNe can jointly provide several building blocks in the migration path towards the next generation network services proposed by SAIL. The Data Centre Interconnection Use Case exemplifies the synergies that can be mobilised between OConS and CloNe.

- CloNe mostly manages virtual infrastructure (storage, processing and networking resources) within an administrative domain and provides a framework to interact with OpenFlow-enabled networks with OCNI (pyOCNI and libnetvirt) providing back-ends to interact with OpenFlow controllers.
- OConS mainly manages connectivity services between end-points over data centres deploying physical paths and routes across network domain boundaries and provides advanced orchestration features that allow heterogeneous network domains to interact.

Both OConS and CloNe propose OpenFlow as a network transport mechanism. Here the two work-packages come together and the full potential of a coordinated approach could be best exploited in the near future.

Using the Data-centre interconnect use-case, CloNe demonstrates that virtual infrastructures can be built in different data centres managed by different infrastructure service providers and connected through wide area networks operated by third-party infrastructure providers, while making it appear as a single virtual infrastructure. With respect to OConS, this use case demonstrates that OConS can efficiently manage connectivity between data centres.

One conclusion extracted from this study is that OpenFlow does not cover all the needs of this use-case, specifically at the WAN part. Therefore, the integration provided by Open Connectivity Services is needed to improve the scenario.

Maintaining the functional separation between Virtual Machine (VM) and connectivity management in an integrated environment allows OConS/Clone interaction.

Currently the data centre virtualisation and the Network virtualisation are separated. In order to perform a migration using OConS/CloNe synergy, network functionalities and Virtual Machines are moved to the virtualised environment. CloNe allows different technical solutions on network virtualisation to coexist, such as the use of layer2/layer3 techniques (i.e. MPLS or QinQ), whereas OConS proposes the use of OpenFlow capabilities adding the Open Connectivity Services integration to obtain complete next generation services.

The use case also provides insight on a migration path towards a network with integrated OConS and CloNe services. This migration path can be extended to cover other use cases. It involves several steps:

1. The data-centres migrate to implement their internal data plane with OpenFlow.
2. The network operators along the interconnection path between the OpenFlow-enabled data centres implement an OCNI interface to allow interaction with the data centre operators. Initially, this OCNI interface is able to control the technology that has been deployed by the network operator.
3. A common orchestrator for network and data-centre operators allows interaction over the OCNI interface
4. The orchestrator integrates OConS mechanisms to allow a policy-based interaction between the data-centres and the network.

Once this point is achieved, the network operator can decide on use-case-by-use-case base whether or not to evolve his network infrastructure to natively provide OpenFlow based services. This might not be always the case and the integration through the OCNI interface could well allow the stakeholders to benefit from OConS functionalities over their current networks in a much shorter period of time.

4 Standardisation

4.1 General Overview

Standardisation is a key enabler of migration. This section identifies the relevant standardisation bodies in the areas of interest to SAIL and how SAIL is related to them, identifying potential gaps and mismatches. In addition, the SAIL contributions to standardisation are described.

Figure 4-1 provides an overview of potentially relevant standardisation bodies from the point of view of SAIL WPs B, C and D (respectively, NetInf, OConS, CloNe). The following sections include a brief description in those bodies that have a direct relationship to SAIL.

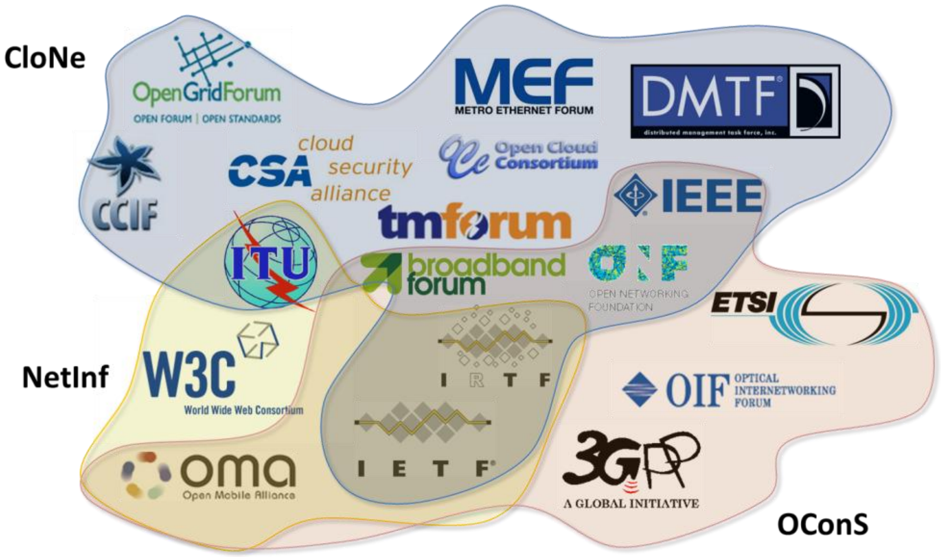

Figure 4-1 General overview of the relevant standardisation bodies

4.2 NetInf Standardisation

NetInf is the SAIL version of Information-centric Networking (ICN). In this section we describe the current state of the ICN standardisation that is applicable to Netinf.

4.2.1 Overview/Relevant Bodies

ICN is still at an early research stage and standardisation efforts are just being started.

The IRTF Information-centric Networking Research Group (ICNRG) was established in the spring of 2012 on an initiative of the SAIL partners EAB, NEC and NSN. The group is chaired by Börje Ohlman (Ericsson) and Dirk Kutscher from (NEC) together with Dave Oran (Cisco). For more info see: http://irtf.org/icnrg

Table 4-1 shows a list of standardisation bodies and how they are related to the fundamental NetInf architecture components.

Table 4-1 Summary of relevant standardisation bodies to NetInf

ID	***Description***	***Relation to NetInf components***
IRTF ICNRG	The initial work in ICNRG is focused on producing three documents intended to form the base for the ICN research field: • **ICN Survey document**, a document that provides a survey of different approaches and techniques • **ICN Research Challenges**, a document that describes the ICN problem statement, the main concepts and research challenges in depth. • **ICN Baseline Scenarios**, a document that defines reference baseline scenarios to enable performance comparisons between different approaches.	Overall NetInf architecture
IETF DECADE WG	DECADE set out to provide network storage for applications, e.g. P2P. This work shares its roots with those of ICN. We contributed the ni-naming scheme to this group. The group was recently closed, reason unclear.	Could provide a basic storage component for migration scenarios towards ICN
IETF CORE WG	Is defining the CoAP protocol which will use the ni naming scheme.	First user of the ni naming scheme.
IETF PPSP WG	PPSP is defining a p2p streaming protocol. It has a need to name NDOs; we will try to harmonize their naming with the ni naming.	Potential user of the ni naming scheme.
IETF CDNI WG	Is trying to make CDNs cooperate in a way that could be seen as a first version of the content distribution functionality that ICN will provide.	Not very relevant as the approach is to tweak existing protocols instead of using an ICN approach.
ITU-T SG 13	ITU-T draft Recommendation Y.FNDAN (Framework of Data Aware Networking for Future Networks), is in progress in Q21 of ITU-T Study Group 13. This document aims to prescribe overview and design goals of DAN (Data Aware Networking) which incorporates multiple aspects of ICN (Information-Centric Networking). It will provide standardisation from the telecom operator and manufacturer perspective, as it is the primary field of interest of ITU-T.	Low activity, so far no relevance to NetInf.

4.2.2 Contributions by Partners

EAB, NEC, UPB and TCD are co-authors on the by SAIL proposed ICN naming scheme 'ni', which standardises how to use hashes to name information objects. It has been approved to become an IETF Standards track RFC. For details please see: http://tools.ietf.org/html/draft-farrell-decade-ni [13].

The NetInf Protocol has been released as an internet draft draft-kutscher-icnrg-netinf-proto [14]. It contains a description of some underlying concepts, a specification of a message-based protocol and a specification of the HTTP and UDP convergence layers for the protocol. It is available at: http://tools.ietf.org/html/draft-kutscher-icnrg-netinf-proto-00

4.2.3 Contribution to Open Source

NetInf software from SAIL partners is available as open source software. The software provides implementations of the NI naming scheme (draft-farrell-decade-ni) and other NetInf features (such as convergence layers, forwarding, caching) in different languages, including C, Clojure, Java, PHP, Python, and Ruby. It also contains additional tools such as patches to curl and wget and shells scripts for web server support. It is available at: http://sourceforge.net/projects/netinf/

An OpenNetInf implementation has been developed and shared by University of Paderborn http://www.netinf.org.

The DTN2 Reference Implementation of the Bundle Protocol (RFC 5050) has been improved and extended by TCD. The Delay Tolerant Networking Research group is part of IRTF and can be accessed at http://www.dtnrg.org/wiki, the source code is at http://dtn.hg.sourceforge.net/hgweb/dtn.

4.2.4 Future Work

We will continue to release new and improved software for NetInf. We will also contribute to and continue to drive the ICN standards.

4.3 OConS Standardisation

4.3.1 Overview/Relevant Bodies

OConS is linked to several standardisation activities, as summarized in Table 4-2. Broadly speaking, these activities pertain either to the overall OConS framework (i.e., in relation with the OConS architectural concepts, orchestration, or information model), or they apply to individual OConS mechanisms adoption into standards (or plans for standardisation).

Table 4-2 Summary of relevant standardisation bodies to OConS

ID	*Description*	*Relation to OConS*
ONF Framework and Architecture WG	ONF aims at specifying a more programmatic Management, Control and Data planes (i.e., usually called Software-Defined Networking - SDN). The Framework will define the broad set of problems that the SDN Architecture needs to address. The Architecture part will define the functional breakdown into subcomponents that together address the broad set of problems outlined in the Framework. www.opennetworking.org	Overall OConS Framework. Including the Northbound API/OSAP.
IRTF SDNRG - Software Defined Networking Research Group	SDNRG can be seen as a forum for researchers to investigate key and interesting problems in the Software Defined Networking field. These should include the "hybrid" approaches in which control and data plane programmability works in concert with existing and future distributed control planes, i.e., not only having a "classical" SDN approach where the whole network control plane is logically centralized into a "controller". http://trac.tools.ietf.org/group/irtf/trac/wiki/sdnrg	SDN possible deployment models. Orchestration procedures and protocols.
ETSI/ISG/AFI (Autonomic network engineering for the self-managing Future Internet)	It provides a Generic Autonomic Network Architecture (GANA) Reference Model for Autonomic Network Engineering, Cognition and Self-Management. As a conceptual model, its purpose is to serve as a "blueprint model" that prescribes design and operational principles of "autonomic decision-making manager components /elements" responsible for performing "autonomic" and "cognitive" management and adaptive control of resources. http://portal.etsi.org/portal/server.pt/community/AFI/344	Management architecture for OConS. Models of Decision Elements and Managed Entities.
ETSI ISG on Network Functions Virtualisation	Upcoming ETSI Industry Specification Group (ISG) to be launched soon. The Network Functions Virtualisation aims to transform the way that network operators architect networks by evolving standard IT virtualisation technology to consolidate many network equipments. Please see the "Network Functions Virtualisation - Introductory White Paper", presented at the "SDN and OpenFlow World Congress", Darmstadt-Germany, Oct. 2012.	Applying the virtualisation technology to come up with appropriate deployment models for the OConS.
IETF/DMM (Distributed Mobility Management) WG	The objectives of DMM (Distributed Mobility Management) working group specifies IP mobility, access network and routing solutions, which allow for setting up IP networks so that traffic is distributed in an optimal way and does not rely on centrally deployed anchors to manage IP mobility sessions. http://datatracker.ietf.org/wg/dmm/charter/	Distributed Mobility Management (i.e., a specific OConS mechanism).

4.3.2 Contributions by Partners

FT/Orange has submitted an IETF draft on DMM-DMA (Distributed Mobility Anchoring) which can be found here: http://tools.ietf.org/html/draft-seite-dmm-dma-05. One of its authors is also a contributor to OConS Distributed Mobility Management mechanism.

Telefonica has contributed to an IETF draft on SDNi, which proposes a protocol for interconnecting Software Defined Networking (SDN) domains, specifically to exchange information between the domain SDN Controllers. The draft can be found here: http://tools.ietf.org/html/draft-yin-sdn-sdni-00.

4.3.3 Future Work

To capitalize on the insights gained from OConS related work, several WPC partners will pursue the standardisation activities related to the Software Defined Networking (SDN) and to the Network Functions Virtualisation (NfV). Specifically, FT/Orange and Telefónica will monitor and contribute at the ONF on the Architecture and Framework WG. Telefónica and FT/Orange are also among the initiators and the supporters of the new ETSI ISG on NfV.

4.4 CloNe standardisation

4.4.1 Overview/Relevant Bodies

Since cloud computing is a technology that has gained rapid traction and has quickly been deployed, the number of standardisation bodies focusing on cloud computing has increased. As a result, the scope and responsibility of each one of them is sometimes overlapping and/or limited. An example of that is the conclusion of the Cloud Computing Focus Group in ITU-T [18]:

- Almost all the Forums dealing with cloud computing have developed their own architecture for their own purposes. Unfortunately, they are not identical.
- There is no SDO or forum that shows the total picture of cloud computing standardisation.

This unfortunately creates a scattered standardisation picture that will most likely improve as the technology matures.

Due to that fact, CloNe has decided to focus its efforts on a few standardisation bodies that have more relevance to our work. Table 4-3 shows a list of standardisation bodies and how they are related to the fundamental CloNe architecture components.

Table 4-3 Summary of relevant standardisation bodies to CloNe

ID	***Description***	***Relation to CloNe components***
Open Grid Forum (OGF) OCCI WG	Defined OCCI. In contrast to Amazon EC2 [9], Apache Deltacloud [29], OpenStack and OpenNebula interfaces, OCCI [7] is an open standard developed by Open Grid Forum (OGF) providing both a rich data model and a concise set of operations facilitating the capture of the state and the management of the network, compute and storage resources offered by any cloud provider, respectively.	OCNI, a network extension of OCCI
Metro Ethernet Forum	The objective is to augment standardised MEF services to meet on-demand requirements of cloud services. Still in an early phase, defining use-cases.	Flash network slices, network connectivity as a service supporting the cloud
IETF L2VPN WG	The L2VPN working group is responsible for defining and specifying solutions for supporting provider-provisioned Layer-2 Virtual Private Networks (L2VPNs), addressing requirements driven by cloud computing services and data centres as they apply to Layer-2 VPN services.	Link Negotiation Protocol
VXDL Forum	A forum to develop insights and standards for virtual infrastructures modelling for Cloud and Network industry and to lead the VXDL language development. Lyatiss (a member of the workpackage) leads the Forum.	The VXDL language
Distributed Management Task Force (DMTF)	The CLOUD working group focuses on cloud interoperability aspects at management layer across multiple providers.	Management components and IaaS interfaces (OCCI/OCNI)
Cloud Security Alliance	Mission is to promote the use of best practices for providing security assurance within Cloud Computing and provide education on the uses of cloud computing.	Authorisation service, intrusion detection service
ITU-T CC Focus Group	Introduction to the cloud ecosystem, creation of reference architecture, including management gap analysis and security. Cloud computing benefits from Telecommunication/ICT perspectives	The CloNe architecture
Broadband Forum	WT-302 provides a framework to support cloud services in multi-service Broadband Networks. It contains use cases of Cloud Services as a new kind of networking function in the context of Multiservice Broadband Networks. The Cloud Service use cases are analysed to determine implications in areas of broadband multi-service architecture and network functionalities, including interfaces, service model, security, billing and operations.	The CloNe architecture

4.4.2 Contributions by Partners

CloNe has proposed OCNI as an extension to OCCI. OCCI is an open standard developed by Open Grid Forum providing both a rich data model and a concise set of operations facilitating the capture of the state and the management of the network, compute and storage resources offered by any cloud provider. Its generic data model makes it fairly easy to extend OCCI with new and more specific features; as is the case of OCNI, which extends the basic OCCI model not only to enhance the current intra-datacentre network offering, but also to include inter-datacentre features.

OCNI has been proposed as a contribution to OGF. Its public implementation is linked from OCCI WG website [6]. As OCCI is currently focused on a few functions in the OCCI Core framework, that contribution will be processed in the future.

The Link Negotiation Protocol is currently under evaluation for contribution to IETF.

4.4.3 Contribution to Open Source

CloNe has been active in the open source community as this is recognized as de facto standardisation of cloud computing and cloud networking-related concepts. For example, Citrixes' CloudStack was released as open source under the Apache Software License. However, the most prominent open source cloud management system is OpenStack that in a very short time managed to create a large community of contributors and strong industry support. Support for advanced networking is slowly progressing in OpenStack. As of October 2012, the latest development is the acceptance of Quantum, a layer 3 network service for datacentres, as a core project in OpenStack.

As a result, CloNe has released key parts of its work as open source: libnetvirt, pyOCNI, and CMBS. A short description of those pieces of software is here added for completeness.

LibNetVirt is an API that creates a network abstraction layer to allow for the management of different network technologies in a programmable way. It allows the management of network resources in uniform manner across a number of networking and virtualisation technologies. As of its current implementation, libNetVirt can control both OpenFlow and MPLS based networks. LibNetVirt is composed of a common interface and a set of drivers to enable for technology dependent and technology independent functions (as described in Section 3.3), easing migration.

pyOCNI is the implementation of the Open Cloud Networking Interface (OCNI). This is an interface that was defined in CloNe for the control of networking resources both in data centres and in operator networks. The OCNI interface allows customers to specify the needed network service through a high level interface where connectivity details can be later agreed upon between involved domains (e.g, using LNP). OCNI is an extension of OCCI as defined by OGF, where two additions were made. The first is a cloud networking extension to the OCCI Core model to add support for networking technologies other than VLANs alone. The second element consists of a number of specialized network *mixins* (a dynamic data type) to support different types of flash network slices.

CMBS is a service for exchanging messages between the domains/providers in CloNe. CMBS implements a brokerless message bus that allows for inter-provider communication to happen in a structure way, based on topics of interest or a selected group of receivers. CMBS is based on a message exchange pattern that has five layers: a discovery space, a broadcast space (message to all members), a specific destination space (messages to a given provider), a topic specific space (messages about a given topic) and specific destination and topic.

4.4.4 Future Work

CloNe anticipated standardisation initiatives in the areas of Cloud Networking in several ways. Cloud Networking combines two technical domains, Information Technologies and Communication Technologies, which traditionally have followed separate standardisation tracks, by distinct communities. This gap has not facilitated a swift uptake of standardisation initiatives in this area. This means that the standardisation of several components of the CloNe architecture is likely to take place after the completion of the SAIL project.

A relevant activity that is currently in the early stages of development and is expected to outlive the SAIL project is the recently established Metro Ethernet Forum (MEF) Technical Committee CE4Cloud, to define the dynamic Ethernet service requirements that match the on-demand requirements of cloud services. Several areas that have been researched by Clone are in the roadmap of the group activity, such as dynamic control of bandwidth capacity to an existing Ethernet connection, or service management orchestration between the Cloud Service Provider data centre resources and the Ethernet Cloud Carrier's network resources. This activity is now in the early stages of development and opportunities for cooperation will be exploited, if and when appropriate. Another forum where SAIL results may be presented is the IETF, namely through the submission of the Link Negotiation Protocol to IETF. This possibility is currently under evaluation, but a contribution is unlikely to take place before the conclusion of the project.

Another recently established initiative that is relevant for CloNe is the ETSI NfV (Network Functions Virtualisation). This effort will be closely followed by several SAIL partners that have been active in the CloNe Work Package.

Finally, SAIL partners' input to standardisation bodies that are already taking place (e.g. OCNI submission in OGF) will be continued and further developed after the completion of the project.

5 Conclusions

From the very beginning, SAIL has identified the definition of a sound migration roadmap as a key objective and a requirement to be fulfilled.

From a technical point of view, multiple migration strategies can be defined, supporting different sets of incentives to various players in the service ecosystem. In this deliverable, three basic approaches for technology migration have been outlined:

- disruptive migration by parallel deployment,
- managed migration by parallel deployment,
- seamless migration by incremental deployment.

Interoperability between legacy and novel technologies is a fundamental migration requirement. Two basic approaches to accomplish that requirement are the introduction of a Convergence Layer and an Interworking Function. The former consists of a horizontal interface between layers of different technology generations and can be used when the requirements of a new generation upper layer to its lower layer are different from those lower layer services that are currently available or deployed. The latter enables interfacing between different communication networks with similar functions, but different protocol stacks or coding standards.

For each of the three SAIL architectural pillars (NetInf, OConS, CloNe), a migration roadmap has been outlined in this deliverable, based on one or more of the aforementioned migration approaches.

NetInf migration roadmaps can be defined, following each of those three approaches, in different settings and service scenarios. The NetInf migration path can be thought of as a process similar to the Internet evolution: just like the Internet was initially deployed over existing telephone lines and gradually evolved to use its own resources, ultimately becoming the main global infrastructure, NetInf deployment will start by using IP but, as the value of NetInf increases, it will get its own network and eventually the services currently deployed over IP could be migrated to NetInf. For example, all massive, user-generated content accessed nowadays from well-known sites could be published in a much more efficient manner, cached and delivered by NetInf taking advantage of multiple nearby locations with cached content.

OConS provides innovative ways to establish Internet connectivity services, following an evolutionary, non-disruptive path by defining mechanisms at flow, network and link levels and by combining them using orchestration. On the other hand, OConS can be seen as a potential enabler of technology migration in general. In particular, several OConS mechanisms can be used as facilitators of NetInf and Clone migration.

With regard to CloNe, a 4-phase evolutionary migration path has been defined. In phases 1 and 2, network operators and data centre providers deploy flash network slices and on-demand networking services, respectively, as two distinct service offerings. Phase 3 keeps two essentially dissimilar technology domains but the implementation of DCP enables the creation of an end-to-end Cloud Networking solution through the integration of services offered by data centres and network providers. The fourth and final stage corresponds to a scenario where network and Cloud resources become a seamless pool of resources, specified by the VXDL description language.

Ensuring that migration is possible from a technical point of view is usually not sufficient – it also necessary to guarantee that a sound business model can be put in place and that clear benefits to all potentially involved players (e.g. end users, content providers, infrastructure providers, service brokers) can be gained from the new technology since the very early stages of deployment.

Last but not least, standardisation is a key ingredient of migration. This deliverable has provided an overview of standardisation in the technical areas related to NetInf, OConS and CloNe and identified the most relevant contributions from SAIL. WP-B has produced a significant output in terms of standardisation contributions, mainly in the IETF, and has supported the establishment of new Research Group in IRTF. In other cases (namely, CloNe, OConS), the fragmented picture of the standardisation ecosystem has limited the effectiveness of possible contributions in the respective area. However, several results achieved in these WPs have been proposed in several fora or are obvious candidates for contributions in the future.

List of Figures

List of Tables

List of Acronyms

ANP	Access Network Provider
API	Application Programming Interface
B2B	Business-to-Business
BPQ	Bundle Protocol Query
CCN	Content Centric Networking
CDN	Content Delivery Network
CDNI	Content Delivery Network Interconnection
CL	Convergence Layer
CloNe	Cloud Networking
CMBS	Cloud Message Brokering Service
CP	Content Provider
DC	Data Centre
DCP	Distributed Control Plane
DHCP	Dynamic Host Configuration Protocol
DMM	Distributed Mobility Management
DMTF	Distributed Management Task Force
DTN	Delay Tolerant Network
FNS	Flash Network Slice
GMPLS	Generalized Multiprotocol Label Switching
GSM	Global System for Mobile communications
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
ICN	Information Centric Networking
ICT	Information and Communications Technology
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
ITU-T	ITU Telecommunication Standardisation Sector
IWF	Interworking Function
KPI	Key Performance Indicator
L2VPN	Layer 2 Virtual Private Network
L3VPN	Layer 3 Virtual Private Network
LNP	Link Negotiation Protocol

MDHT	Multi-level Distributed Hash Table
MEF	Metro Ethernet Forum
MPLS	Multiprotocol Label Switching
NetInf	Network of Information
NI	Named Information
NNRP	NetInf Router Platform
NRS	Name Resolution System
OCCI	Open Cloud Computing Interface
OCNI	Open Cloud Networking Interface
OConS	Open Connectivity Services
OGF	Open Grid Forum
PCE	Path Computation Element
QoE	Quality of Experience
ROI	Return On Investment
SAIL	Scalable and Adaptive Internet Solutions
SDO	Standards Development Organisation
SLA	Service Level Agreement
SOP	Service Orchestration Process
TCP	Transmission Control Protocol
TE	Traffic Engineering
UMTS	Universal Mobile Telecommunications System
UNI	User Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VoIP	Voice over IP
VPN	Virtual Private Network
VXDL	Virtual private eXecution infrastructure Description Language
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WP	Work Package

References

[1] The SAIL project web site. http://www.sail-project.eu/

[2] SAIL Deliverable D-2.2 (D-A.2) Draft Architectural Guidelines and Principles, 31/07/2011, available at http://www.sail-project.eu/deliverables/

[3] SAIL Deliverable D-2.7 (D-A.7): New Business Models and Business Dynamics of the Future Networks, 29/07/2011, available at http://www.sail-project.eu/deliverables/

[4] SAIL Deliverable D-2.8 (D-A.8): Evaluation of Business Models, October 2012, available at http://www.sail-project.eu/deliverables/

[5] SAIL Deliverable D-4.1 (D-C.1): Architectural Concepts of Connectivity Services, July 2011, available at http://www.sail-project.eu/deliverables/

[6] OCCI WG, http://occi-wg.org, last accessed 30th Oct 2012

[7] SAIL Description of Work Grant agreement for Collaborative project, Annex I - "Description of Work"

[8] Path Computation Element (pce), http://datatracker.ietf.org/wg/pce/charter/

[9] SAIL Deliverable D-3.2 (D-B.2), “NetInf Content Delivery and Operations”, May 2012, available at http://www.sail-project.eu/deliverables/

[10] SAIL Deliverable D-3.3 (D-B.3): Final NetInf Architecture, 30/11/2012, available at http://www.sail-project.eu/deliverables/

[11] SAIL Deliverable D-5.4 (D-D.3): Refined architecture, 31/10/2012, available at http://www.sail-project.eu/deliverables/

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R L. Braynard. Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT '09, pages 1-12, New York, NY, USA, 2009. ACM

[13] S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, and Phillip Hallam-Baker. The Named Information (ni) URI Scheme: Core Syntax. Internet Draft draft-farrell-decade-ni-00, Work in progress, October, 2011

[14] D. Kutscher, S. Farrell, and E. Davies, The NetInf Protocol, draft-kutscher-icnrg-netinf-proto-00, October 2012

[15] Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO) http://www.3gpp.org/ftp/Specs/html-info/23829.htm

[16] D. Shayani, C. Machuca, and M. Jäger, A Techno-Economic Approach to Telecommunications: The Case of Service Migration, IEEE Transactions on Network and Service Management, Vol. 7, No. 2, June 2010

[17] SAIL Deliverable D-2.9 (D-A.9): Description of Overall Prototyping Use Cases, Scenarios and Integration Points, 11/06/2012, available at http://www.sail-project.eu/deliverables/

[18] ITU-T FG Cloud TR Focus Group on Cloud Computing Technical Report, Part 6: Overview of SDOs involved in cloud computing