FEDERATED TEST-BEDS FOR LARGE-SCALE INFRASTRUCTURE EXPERIMENTS
FELIX EU-JP

Collaborative project co-funded by the European Commission within the Seventh Framework Programme

Grant agreement no: 608638
Project acronym: FELIX
Project full title: "Federated Test-beds for Large-scale Infrastructure eXperiments"
Project start date: 01/04/13
Project duration: 36 months

Deliverable D4.2
FELIX Experiments Report

Version 1

Due date: 31/08/2015

Submission date: 17/09/2015

Deliverable leader: iMinds

Author list: Vicent Borja Torres (iMinds), Brecht Vermeulen (iMinds), Carlos Bermudo (i2CAT), Carolina Fernandez (i2CAT), Bartosz Belter (PSNC), Łukasz Ogrodowczyk (PSNC), Damian Parniewicz (PSNC), Kostas Pentikousis (EICT), Umar Toseef (EICT), Gino Carrozzo (NXW), Roberto Monno (NXW), Atsuko Takefusa (AIST), Jason Haga (AIST), Tomohiro Kudoh (AIST), Takatoshi Ikeda (KDDI), Jin Tanaka (KDDI).

Dissemination Level

☒	**PU:**	Public
☐	**PP:**	Restricted to other programme participants (including the Commission Services)
☐	**RE:**	Restricted to a group specified by the consortium (including the Commission Services)
☐	**CO:**	Confidential, only for members of the consortium (including the Commission Services)

Abstract

This report contains information on integration steps and experimental results obtained from running some FELIX use cases in the FELIX infrastructure, as well as plans for action on the rest of the use cases portfolio defined in the project. Originally, the FELIX consortium defined 6 use cases in total, under two major categories: Data Domain and Infrastructure Domain use cases. Once specific commonalities have been identified among some of them, the portfolio of FELIX use-cases has been consolidated around 3 major representative experimental scenarios: high quality media transmission, infrastructure domain and data pre-processing. This document describes for each merged use case its background and storyline, goal of experiments, experimental conditions, KPIs, experimental scenarios and the physical infrastructure in use. This report will be updated by D4.3 in M36.

Executive Summary

Deliverable D4.2 details the status report on the WP4 experimentation phase of the FELIX project.

The deliverable presents and overview of the use cases described in detail on D2.1 and is aimed to explain how the use cases are mapped to use the FELIX testbed infrastructure already deployed to execute the experiments. The document is mainly structured with two chapters; Chapter 2 reports on the FELIX Use Cases in Progress and Chapter 3 FELIX Use Cases Planned.

This document also describes the experiments for each use case, goals, experimental scenario and architecture, physical infrastructure provisioned, experimental conditions and KPIs.

During its implementation activities executed in Y2, the consortium realized that the different use cases might have had commonalities and effort required to implement all of them individually would have been beyond the project capabilities and redundant to validate the project concept and objectives. A decision has been taken on merging some of the use cases, thus consolidating the FELIX evaluation around 3 fully representative scenarios. The result of this portfolio consolidation work is reported in this deliverable. Part of the final FELIX use cases have just been validated, part are in progress and the remaining part is planned for execution in the last period before the end of the project.

Final validation details on all the FELIX use cases will be presented in deliverable D4.3, due at M36, which will update this document with final data from experimentation.

Table of Contents

Figure Summary

Table Summary

1 Introduction

Experimentation is one of the most effective ways to perform validations where the whole development of the project is put to the test. Considering this fact, FELIX use cases have been designed in an attempt to find the most realistic approach to benchmark the project. This involves verifying the preset objectives through performing stress testing of all the components and features developed during the development phase.

Initially FELIX use cases were six as reported in D2.1 delivered in the first year of the project. After a full development loop in Year2 and the evaluation of integration efforts needed to experiment in one use case, the FELIX consortium decided to make the validation strategy more focused and merge use cases which were showing commonalities and/or complementarities and which could have been better tested and validated in merged scenarios.

Therefore, the previous six use case identified in D2.1 have been grouped under three new experimental scenarios:

- High Quality Media Transmission
- Infrastructure Domain
- Data Pre-processing on Demand

Each group is focused to validate certain features of the architecture such as robustness, dynamic network reconfiguration and steering of data flows using SDN and OpenFlow controllers, etc.

This document describes per use case the background and storyline, the success conditions, description, goals, scenarios and architecture for the experiments, physical infrastructure and the FELIX components used by the experiments.

In Chapter 2 we report on FELIX use cases in progress at the time of delivery of this document. Chapter 3 describes FELIX use cases whose execution is planned for the period between Q3-2015 and Q1-2016..

This document is to be intended as a report on the first phase of FELIX experimentation, Final conclusions and results will be presented in the next deliverable D4.3 to be produced by the end of the project (Mar-2016) with data and analyses on the execution of all the planned experiments.

2 FELIX Use Cases in Progress

2.1 HQ Media Transmission

2.1.1 UC.6 – High Quality Media Transmission over long-distance networks

SDN control and orchestration tools for automatic adjustment of connection paths and parameters could be powerful control engines for advanced multimedia testing purposes. High resolution visualization solutions, especially streaming technologies, could be tested in order to pinpoint sensitivity for network issues changes. Since they are very demanding and require high quality of transmission, streaming can act as very sensitive test engines for transmission problems. While FELIX delivers a federation of SDN testbeds interconnected through NSI-enabled domains, it enables opportunities for testing media streaming over existing network technologies in backbone networks in conjunction with OpenFlow.

With respect to the deployed technologies, this experiment would help to determine the behaviour of transmission mechanisms in case of streaming high resolution multimedia content over a very long distance. The key aspect of such kind of transmission is to take heed on sensitivity for negative effects. The experimenter may want to evaluate several aspects of the transmission such as round trip delay, jitter, network bandwidth and wrong frames sequences which cause frames losses and reduce frames per second (FPSs) decoded and properly displayed at the media consumer side.

The use case presented herein was also described in the previous publicly available document D2.1 “Experiment Use Cases and Requirements” [D2.1].

2.1.2 Storyline

A Service Provider is interested in running a new service between Europe and Japan. He/She plans to run streaming of the high-quality content from Europe to Japan and vice versa and would like to acquire connectivity services from commercial providers. He/she assumes multiple streams running in parallel on top of existing infrastructures so he/she plans to examine long distance network capabilities for media streaming. The Service Provider would also like to test an application for an intelligent network reconfiguration in case of problems with end-to-end performance over the network infrastructure when using multiple streams transmissions.

The risk of side effects may be reduced by transmission parameters that would be under use case research. Therefore long distance network connections would be the most factual test case for advanced content distribution systems. Performed tests could define the possibility of future development and usage of delivery systems regardless of the transmission distance. Many aspects of this sensitive communication could be identified and qualified in order to define a sufficient set of requirements for seamless streaming. Additionally, QoS as well as QoE (Quality of Experience) issues could be examined in order to find a correlation between network problems and their visual reflection.

Success Conditions

From the FELIX point of view, the success condition for this experiment is the correct setup, orchestration and control (via SDN controller) of the streaming environment over long distance, with automatic network re-configurations based on application performances.. The streaming will be effectively delivered to end-users when all the side effect are minimized or ruled out. It also determines the success scenario for end-user as well as for multimedia provider, which may be verified by Quality of Experience technicians.

From a FELIX experimenter perspective, a success condition for this use case is the availability of a solid experimental environment where to develop software for streaming control policies with repeatability and flow monitoring information to use to guarantee appropriate network behaviours when streaming high resolution media.

2.1.3 Experiment description and components

2.1.3.1 Goals of the Experiments

The experimenter runs at least two physical machines on two different testbeds federated over a long distance: a media source and a media consumer. The experimenter installs software for streaming, which may be one of the popular software or his/her original solution (during use case validation UltraGrid v 1.3 [UG] software was used).

When the experimenter installs his/her own software dedicated to media streaming, the FELIX federated testbeds allow to examine it over a long distance network. Having such a powerful environment the experimenter may test the software original solutions for all the streaming side effects supported by monitoring data and configurable network flows. During the tests the experimenter may monitor various metrics of the network connections setup for the streaming and do research on their impact on the quality of streamed media as well as on minimizing side effects. Playing with network flows allows the experimenter to find the best set of configuration parameters.

The goal of the experiment is to examine long distance network capabilities for media streaming. The network is configurable in terms of flow and measured using various metrics. The software used for streaming provides quality measurement capabilities that are used by the network streaming control application.

2.1.3.2 Experimental Scenario and Architecture

Resources provisioned for the experiment together with additional facilities needed for media transmission streaming, control and visualization formulates “experiments architecture” which is visualized in Figure 1. The Experiment GUI provides mechanisms for triggering SDN path configuration and media streaming control. Moreover, GUI visualizes the current scenario network topology and monitoring parameters. The GUI system can control all resources on the path from the media streamer to the media player.

To perform the scenario the following interfaces are used by the GUI:

- REST API to SDN controller allows OpenFlow path configuration over the network through proper flow modes installation (flow_modes add/delete)
- REST API to the Rate limiter allows configuration of the bandwidth on the OpenFlow path (bandwidth per list of destination IP addresses)
- Interface towards UltraGrid-streamer allows media streaming control (start/stop).
- Interface towards the Monitoring system allows collecting transmission and media parameters. During the experiment the following parameters are measured:
    - Bandwidth (in [Mbps]) – provides information about media player network interface statistics,

- RTT (Round Trip Time in [ms]) – provides information about round trip delay time between media player at PSNC and media streamer at AIST site,
- FPS (Frame Per Second) – number of frames that the media player could decode and visualize during one second time slot,
- Movie frames losses – percentage of frames that are not properly visualized during last 5 seconds due to reordering, jitter, too much delay, etc.

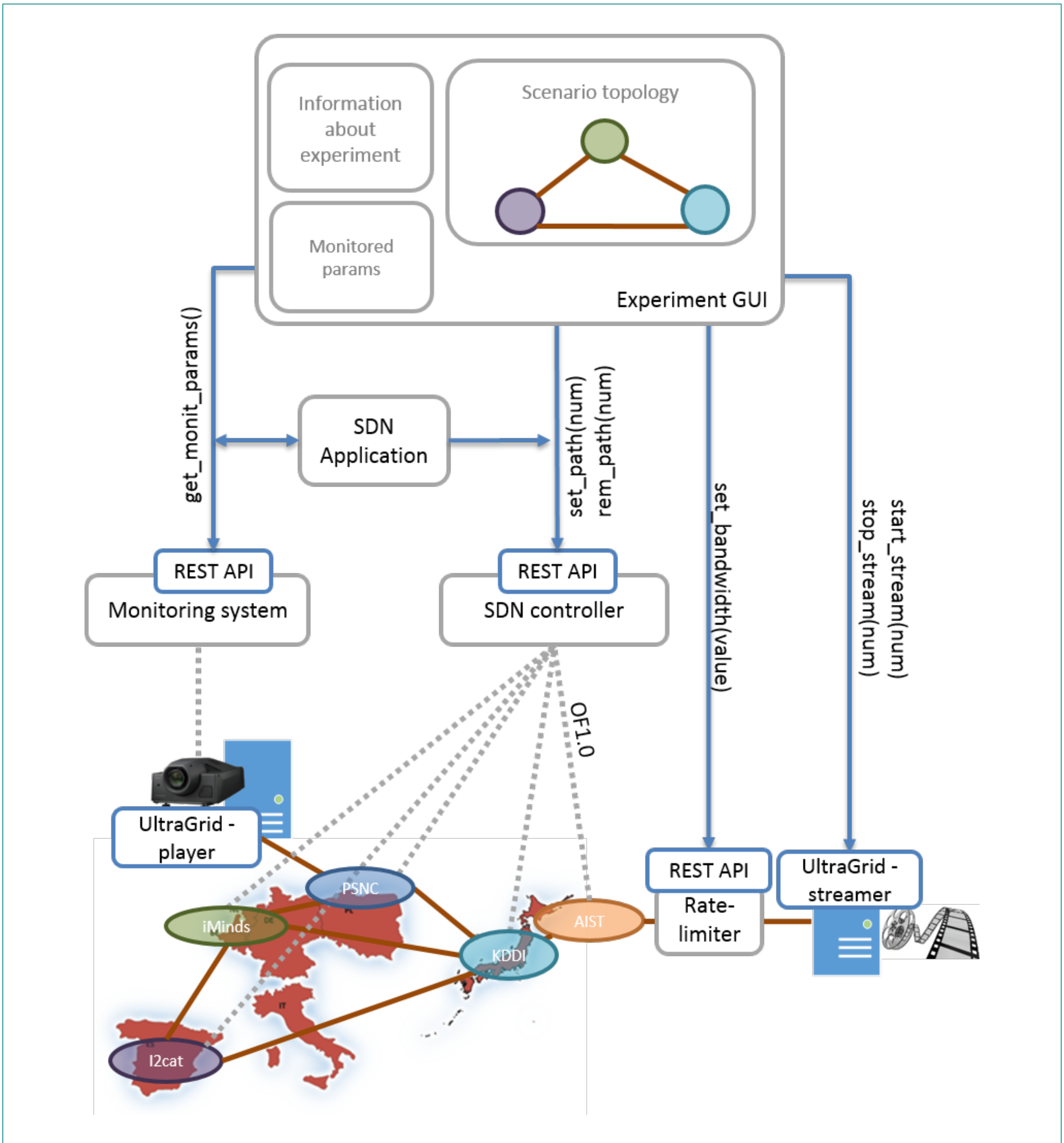

Figure 2.1: Experiment Architecture – Resources Provisioned for the Experiments.

2.1.3.3 Physical Infrastructure Provisioned for the Experiments

Figure 2 shows OpenFlow resources of that slice used during experiments together with a media streamer and media player facilities. SDN resources from 5 FELIX Islands are utilized during high quality media streaming: PSNC, AIST, KDDI, i2cat and iMinds.

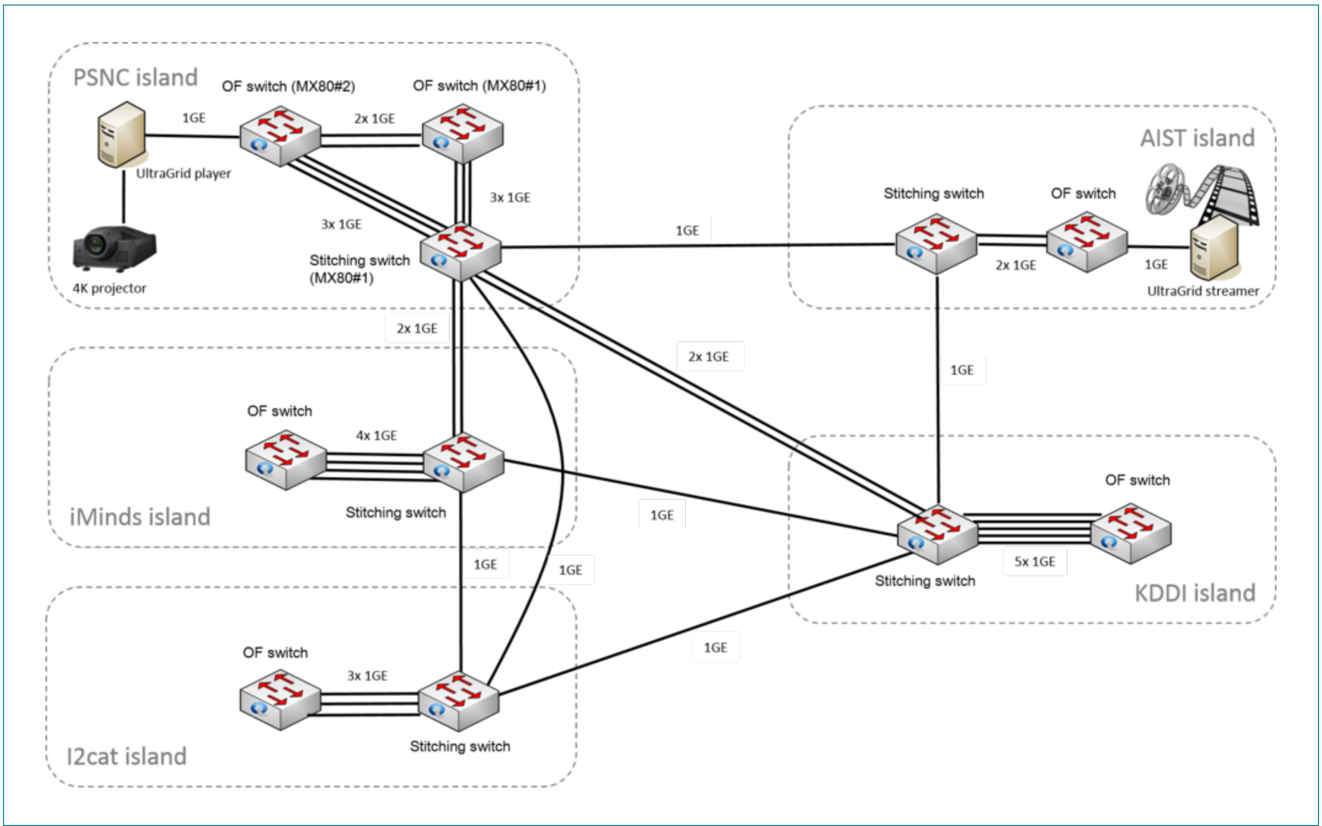

Figure 2.1: Physical Infrastructure Provisioned for the Experiments.

2.1.3.4 Experimental Conditions

Experimental Conditions	
Involved Islands	PSNC, AIST, KDDI, iMinds, i2cat
Actors	Media provider, testbed operator, media consumer (end user), experimenter.
Requirements for Use Case execution in FELIX test environment	▪ SDN controller together with a network application for path reconfiguration (addition/removal/modification of OpenFlow flows) based on QoE or monitoring parameters ▪ High capacity of streaming machines (e.g. Media workstation with Intel i7,

	12GB RAM, AMD Fire Pro 3D Graphics v8800) ▪ Basic monitoring system (traffic at the media player side, round trip delay, frames losses, FPS displayed)
FELIX framework components	▪ Expedient GUI ▪ Resource Orchestrator (RO) ▪ Resource Managers: SDN-RM, TN-RM, SE-RM ▪ NSI-controlled network domains ▪ Experiment Control and Management
Provisioned resources	▪ OpenFlow switches and stitching entities at PSNC, AIST, KDDI, i2cat, iMinds ▪ SDN network Controller (Ryu) compatible with OpenFlow v.1.0
Additional resources	▪ Media player at PSNC (workstation with UltraGrid v3 - software for media streaming/decoding) ▪ Media streamer at AIST (workstation with UltraGrid v3 - software for media streaming/decoding) ▪ Rate-limiter at AIST (software module for limitation of the bandwidth on the path) ▪ 4k display at PSNC to visualize parallel transmitted media streams ▪ Network application for streaming control ▪ GUI to visualize current network configuration and streaming parameters
Network capacity	▪ 1Gbps ▪ Rate-limiter at AIST provides software solutions of traffic throughput limitation for emulation of different bandwidths over paths in the SDN network
Media content	Four different movies encoded in H.264 with rates: 18, 20, 24, 28 Mbps
Lifecycle	Choose experiment scenario, initialize connections, run streaming, monitor performance (check network application behaviour), examine QoE, stop streaming, tear down circuits

Timeslot	About 30 minutes

Table 2.1: Experimental Conditions UC.6

2.1.3.5 *Experiment 1: Analyse Long Distance Network Capabilities for High Quality Media Streaming.*

During this experiment user experience (QoE) with high quality media transmission over a long distance network are evaluated. The network path could be reconfigured on demand through the SDN controller based on feedback from the users. Four different paths over NSI domains connect AIST and PSNC sites as depicted in Figure 3. Each path in the network is configured with different bandwidth (18, 20, 24, 28 Mbps) by a rate-limiter module at the AIST site. H.264@20Mbps media content is pre-prepared for the experiment.

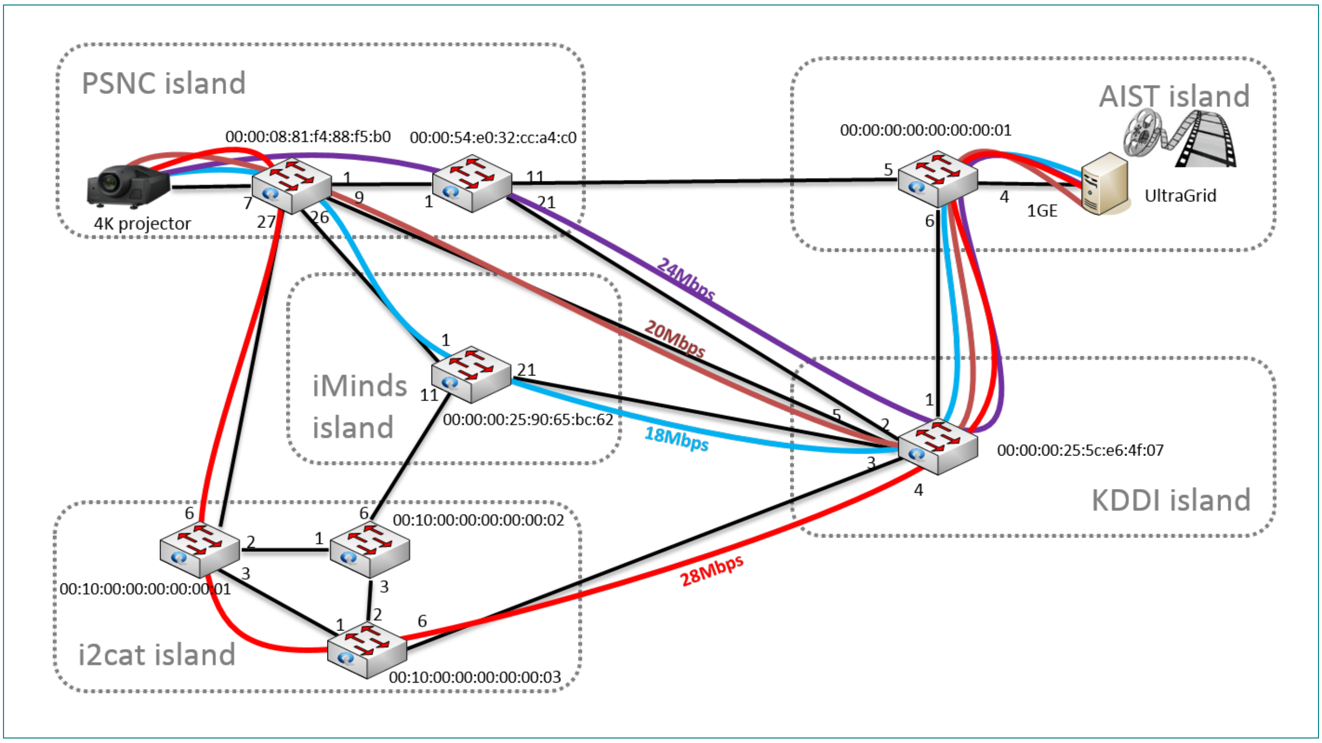

Figure 2.2: Experiment 1 – SDN Network Topology

Experiment scenario:

1. User starts the media player together with 4k projector at PSNC
2. User sets up rate-limiter at AIST side to 18Mbps
3. User configures the first OpenFlow path (18Mbps) over the provisioned slice and starts media streaming
4. Group of end users analyse QoE at the PSNC side
5. In case of insufficient streaming quality the user changes the path (repeat steps 2-4 for the following paths: 20/24/28Mbps)
6. User stops the movie and releases the path

The aim of the experiment is to check the boundary condition of the network capacity needed for transmission of the high quality media. The experiment should provide information about the minimum bandwidth required for media delivery with satisfying quality. Moreover, the monitoring system should inform in real time about possible stream degradation due to traffic congestions on the path.

2.1.3.6 *Experiment 2: Analyse New Intelligent Network Application for Control High Quality Media Streaming.*

During this experiment network application for controlling the media streaming over the SDN network is validated. Such application should provide smart solutions for automatic network reconfiguration and adjustments in case of media streaming parameters degradation. Four different paths over NSI domains connect AIST and PSNC sites as depicted in Figure 4. Each path in the network is configured with different bandwidth (20, 40, 60, 80 Mbps) by the rate-limiter module at the AIST site. Four different H.264@20Mbps media contents are pre-prepared for the experiment. During the experiment the user adds multiple streams, which decreases the media visualization quality and should automatically trigger path recalculation in the network to ensure the proper level of QoS.

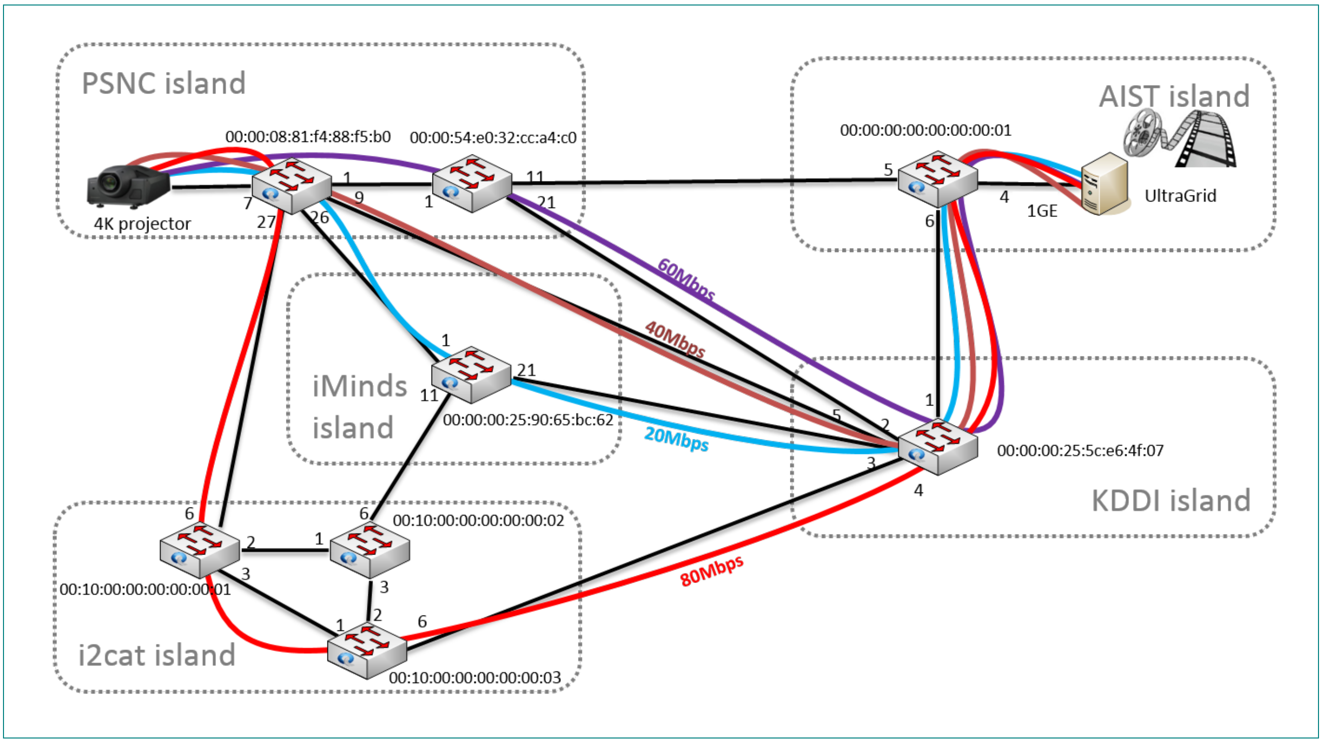

Figure 2.3: Experiment 2 – SDN Network Topology

Experiment scenario:

1. User starts the media player together with a 4k projector at PSNC. The projector is configured to visualize multiple streams simultaneously on the screen
2. User sets up a rate-limiter at the AIST side to 20Mbps
3. User configures the first OpenFlow path (20Mbps) over the provisioned slice and starts media streaming
4. User adds the next stream
5. Network application runs on top of the Monitoring System and SDN Controller checks media quality at the media player side

6. User repeats step 4 four times (for each pre-prepared stream). In case of insufficient streaming quality application triggers path recalculation. It also triggers rate-limiter reconfiguration (depending on the selected path)
7. User stops the movie and releases the path.

The aim of the experiment is to examine behaviour of the smart network application for automatic network configuration and adjustments in case of media streaming parameters degradation. Such solution should ensure short convergence time in case of any network rearrangements. The experiment should also provide information about influence of the network overloads caused by multiple streams on the ongoing media transmissions.

2.1.4 KPIs of the Experiments

The following table contains the list of KPIs we have identified in order to measure the degree of consecution of the Use Case implementation.

KPI	FELIX approach
Streaming is successfully provided by the media provider and received by the media consumer	The FELIX project provides testbed facilities and FELIX Control Framework for creation and management of the experimental slice and to run experiments over a long distance network. Additional hardware and software facility such as media streamers, players, rate-limiters can be added to the slice for media streaming through SDN-controlled network.
Network and streaming monitoring data are collected and affect decisions taken by the network application	The smart network application runs on top of the Monitoring System and SDN Controller and can take a decision about network reconfiguration for transmitted media characteristics improvements.
The network application for the streaming control can handle up to four parallel media streams	The network application for the streaming control can handle up to four parallel media streams
Interruption of media streaming due to network reconfiguration takes less than 10 seconds	Network rearrangement in the OpenFlow-controlled network is a matter of deleting old and adding new flow_modes to all OpenFlow switches on the path. During the FELIX experiment it is also time needed for rate-limiter module reconfiguration.
The provisioned slice provides up to 80Mbps capacity for streaming four parallel media contents each encoded with H.264 @20Mbps	1Gbps is limitation of the NSI-controlled network. However, this bandwidth is not guaranteed in the network and may be shared by other NSI interconnections. Network performance provisioned through NSI should be examined also during the FELIX experiments.

Table 2.2: KPIs UC.6

2.2 Infrastructure Domain Use Cases

In the previous publicly available document D2.1 “Experiment Use Cases and Requirements” [D2.1], it was proposed to have three different infrastructure domain use cases. These use cases would demonstrate and assess the capability of the FELIX system, leveraging on both the infrastructure resources and FELIX management stack. As part of the consolidation of the FELIX use case portfolio, we have combined the three Infrastructure Domain use cases as they all share common success conditions and goals with respect to the FELIX project. This will allow partners to focus on other aspects of the project as recommended by the reviewers. Below is a summary of the three use-cases and how they are similar (the same indexing from D2.1 is used for easier reference and readability).

2.2.1 Original Use Case Descriptions

A brief summary of the three infrastructure domain use cases is provided below to highlight the rationale behind their merging. This will also help to identify the main common points of each use case.

2.2.1.1 UC.3- Data Mobility Service by SDN Technologies (Inter-Cloud use case)

In order to apply the cloud system to mission-critical areas, such as electronic administration, medical care, and finance, it will be essential to meet demands for end-to-end guaranteed service quality, reliability of compliance, and power saving of cloud systems. However, single-cloud systems have limitations in the availability of resources and will have difficulty in providing satisfactory quality of service. Thus, there is a need to provide a mechanism for flexibly reassigning resources among cloud systems by means of an inter-cloud system. For example, a user of a service provided by a cloud system moves to a remote location (e.g., on a business trip). Because of the longer physical distance (causing a longer network delay) from the home site where the service is provided, the user experiences degradation in response performance. The cloud system detects the performance degradation and leases some resources from a cloud system located closer to the user location, and relocates the application, service, and/or data using SDN infrastructure to this cloud system. As a result, the user will experience home site performance levels and quality of service by accessing services from a site near his or her current location. When applying inter-cloud to cloud systems on the market, it is required to satisfy demands for guaranteed end-to-end service quality among SDN enabled cloud networking across multiple data-centers.

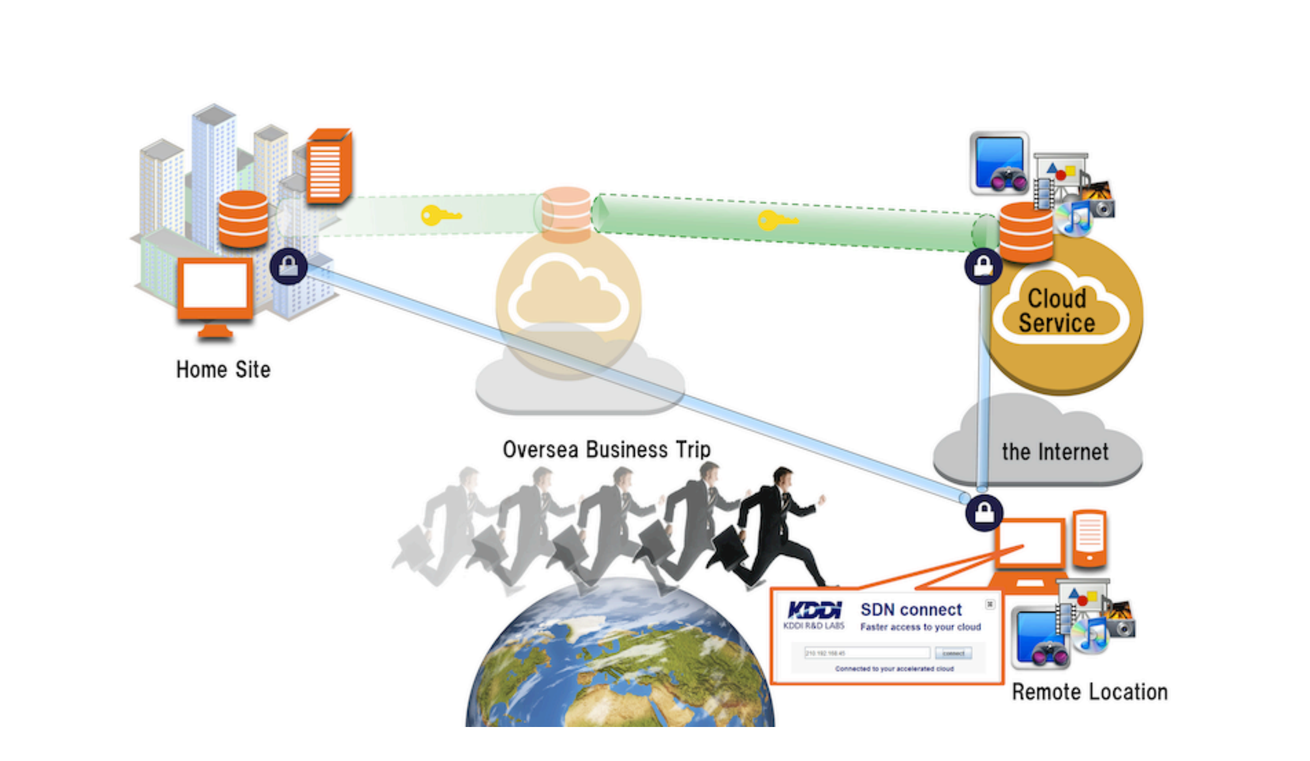

Figure 2.4: Diagram Illustrating UC.3

Goals for UC.3

The goal is to verify the end-to-end tenant and network connectivity across SDN testbeds that are created by the interworking of cloud agent's SDN controller and FELIX management system. The user's data (user's ID information, application) are temporarily moved to the cloud system near a user's location among SDN islands interconnected by NSI based transport network. Use case failure means user can't get the same level of network performance as before.

Success Conditions for UC.3

The cloud system detects the degradation in response performance (trigger).

The cloud system leases some resources from a cloud system located close to the user visiting place (provision of new resources).

The user's ID, application, and data are migrated to this cloud system so that the user can access the service provided from a site near his or her current location (migration).

User can detect no change in performance during his temporary access to his service (no degradation in QoS).

2.2.1.2 UC.4 - Follow-the-sun / follow-the-moon principles

The original idea, which served as inspiration for this use case is described in detail in [Qureshi, Asfandyar et al., 2009]. Therein, the authors state that Internet usage curves are following a similar daily pattern everywhere in the world, and that there is a natural potential to shift the load of data centres to places in the world where it is currently night ("follow the sun/moon"). Moreover, the prices of renewable energy strongly depend on the availability of wind and solar energy. As cooling is reported to comprise of up to 50% of a data centres total energy bill, recently, large data centre providers have started to place infrastructure in places such as Iceland and Finland, where the centres can leverage the low environmental temperatures for cooling. Similar projects may be considered in desert areas, where the amount of solar energy available may even make up for the additional cooling required.

In order to leverage these monetary and environmental benefits, the load of one data centre needs to shift to another. This shift can be done following two approaches. First, the workload is moved to the more efficient data centre entirely and the consumers' traffic is rerouted. Alternatively, a less drastic option is to handle the consumers' requests at the less efficient centre and delegate the actual processing to the more efficient data centre. This way, the hosts in the first data centre can act as proxies and workload load balancers outsourcing part of their compute/storage/networking effort. These kinds of proxies naturally have lower resource consumption, so the less efficient centre can reduce its load. The second option requires the service provided by the data centre to enable delegation. This means that the algorithm for processing must enable distributing the computation load.

Both scenarios - complete migration and delegation – have a strong need for establishing dynamic, on-demand end-to-end connections between the data centres, which can be seen and implemented in practice as federations. These federations would require the use of SDN mechanisms for (re-)routing of the consumers' traffic, the traffic within the data centres and between them.

When the workload is moved from one data centre to another, compute resources need to be provisioned, as discussed above. The more efficient data centre gets notice to allocate more resources and once they are available the traffic migration can be triggered.

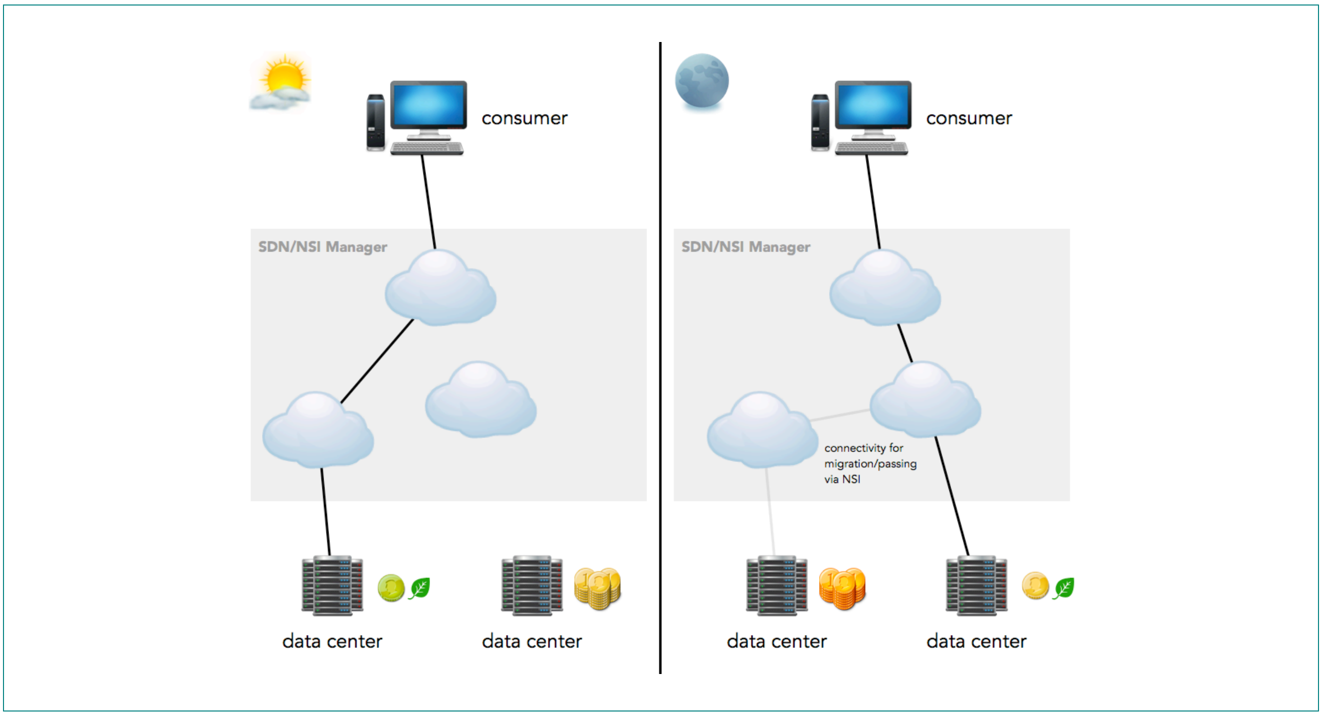

Figure 2.5: Diagram Illustrating UC.4

Goals for UC.4

The overall goal is to minimize the total cost of operation for a virtual data centre while maintaining an acceptable end-user experience. This cost is mainly determined by the energy costs, but may also include cost for CPU/RAM usage and bandwidth cost. Another goal is to regulate the usage of resources in a data centre. This may be used to shape the consumption according to the availability (time and place) of renewable energy sources, for example.

Success Conditions for UC.4

The operating cost of one data centre is increasing or is in a timezone where daytime is beginning (trigger).

The system established new inter-provider connections via NSI for the migration and/or the customer connectivity (provision of new resources).

The consumers' data/processing load was successfully migrated from one data centre to another (migration).

The consumer did not experience synchronization problems. Ideally the migration process is transparent to the user (no degradation of QoS).

2.2.1.3 UC.5 - Disaster recovery by migrating IaaS to a remote data center

Business Continuity Planning (BCP) of IT services is one of the important issues for Cloud users and providers especially after the Great East Japan Earthquake in 2011. Infrastructure as a Service (IaaS) providers prepare physical resources, such as computers, storage and network in a data center, and install IaaS management software, such as OpenStack [OpenStack] and CloudStack [CloudStack], on the resources. Then IaaS providers feature isolated tenants, which consist of virtual machines, isolated storage volumes and isolated network, with multiple users, who create and provide business or entertainment application services on the provided tenants, respectively. Users of such IaaS expect the IaaS environment to be stable and fault free, however it is difficult to continue to provide such services when a serious disaster, e.g., earthquake, happens because of disruption of the electric power supply to the physical resources. In order to continue a variety of mission critical services on an IaaS, all of the IaaS resources must be migrated to another undamaged site, while short-term electric power sources such as UPS and simple power generators are available. Once the IaaS environment is migrated to a safe site, the users can continue to access services during a disaster.

This use case will realize such IaaS migration between data centers in Europe and Japan. Hardware as a Service (HaaS) is a key idea to realize this IaaS migration. HaaS is a service, which dynamically configures and provides virtual resources, on which IaaS can run, by using nested virtualization technologies, such as KVM and OpenFlow. The virtual resources running on the HaaS layer of a data center can be migrated on the HaaS layer of another data center. In addition, high bandwidth SDN internetworking technology will enable this IaaS migration to be performed in a rapid manner within a limited time span.

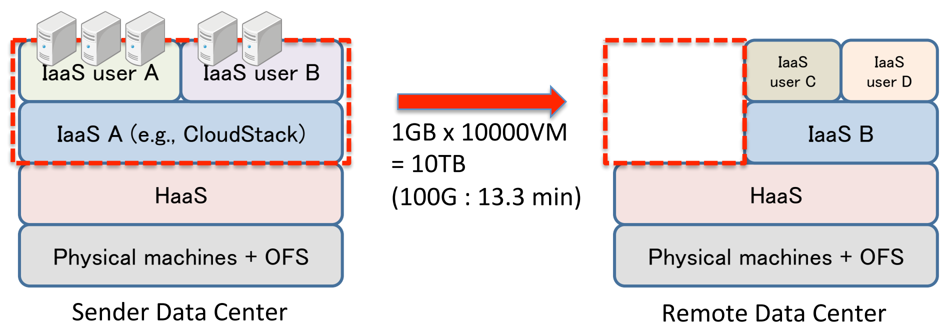

Figure 2.6: Diagram Illustrating UC.6.

Goals for UC.5

Migration of a working IaaS on which multiple tenants are being executed from a data center to another data center. The FELIX middleware will enable the migration of the whole IaaS environment, which consists of a cluster of virtual machines including IaaS management node, to a remote data center to continue/maintain business operations.

Success Conditions for UC.5

System detects/is alerted to natural disaster and requests migration (trigger).

System requests new resources and extends a single L2 plane (provision of new resources).

A working IaaS and tenants on the IaaS are migrated from a data center to another data center, according to a trigger event (migration).

Tenants on the migrated IaaS continue operation (no degradation of QoS).

IaaS image is cached at the destination data center and only the updated portion of the image is transferred at the time of migration, so as to realize quick migration of processing.

2.2.2 Updated Storyline

Generally, the following storylines demonstrate the abilities of the FELIX infrastructure. In particular, FELIX allows the business/operators to test migration of applications and data and how to optimize their performance. They can also optimize various parameters that affect the migration, such as size of the data/applications to be migrated, network bandwidth required for the migration, re-establishment of services, overall time required to migrate and resume service, etc.

2.2.2.1 UC.3 - Data Mobility Service by SDN Technologies (Inter-Cloud use case)

A business person is traveling from Japan to Europe. His/her work requires connection to a variety of applications and data that are integral to business operations. The distance makes access to these applications difficult and the performance suffers as a result of the distance between the hotel and business located in Japan. Thus, local cloud resources are leased and the applications are moved to the local infrastructure allowing for seamless interactions with the applications and data as well as performance that is equivalent to his office in Japan.

2.2.2.2 UC.4 - Follow-the-sun / follow-the-moon principles

Data center administrators in Europe would like to minimize operating costs through the use of infrastructures that are operating at night in Asia. Thus, it is desirable to move a European center at the beginning of the day to a location in Asia, where evening is beginning. Concurrently, customers' traffic to this new data center is rerouted. As this location (Asia) approaches daytime, the data center and traffic are moved again. The lower operational cost is sustained by maintaining the center in lower usage time periods around the globe.

2.2.2.3 UC.5 - Disaster recovery by migrating IaaS to a remote data center

After a large natural disaster strikes Japan, there is loss of power at a computer data center. Operators of this center have a limited time to migrate all applications, services, and data to a safe destination located in a geographically safe place. This time is generally determined by the UPS (uninterruptible power supplies) and backup generators that are available on-site and are usually of very limited duration (up to 24hrs). Once the migration is complete, this will help to maintain services, especially the flow of information that may be critical for disaster management officials to use during the disaster.

The FELIX infrastructure allows the business/operators to test migration of applications and data and how to optimize their performance. They can also optimize various parameters that affect the migration, such as size of the data/applications to be migrated, network bandwidth required for the migration, re-establishment of services, overall time required to migrate and resume service, etc.

2.2.3 Experiment description and components

For each of these use cases, there are three different triggers: (1) degradation of application performance in UC.3, (2) increase in cost of operations/beginning of daytime in UC.4, and (3) natural disaster event causing loss of power in UC.5. Although the triggering events are all different, generally all three use cases leverage on the same FELIX resources and management stack:

1. API between experimenter's SDN controllers
2. FI experimental facilities
3. NSI-controlled network domains
4. SFA-based federation
5. SDN Manager
6. Resource Management
7. Experiment Control and Management
8. Monitoring Framework
9. Inter-data center network provider software
10. Inter-data center resource management software
11. Data center resource management software including SDN managers and computer and storage resource managers

Importantly, a minimum of two HaaS-enabled data centers/SDN islands must be available for these infrastructure domain use cases. One center acts as the sender and the destination center acts as the receiver. IaaS must be running at the sender data center on the HaaS. The destination data center must have sufficient physical resources as well as the ability to emulate the physical resource environment of the migrated IaaS in the HaaS. Furthermore, most data centers will be running multiple, nested virtualization technologies and the destination data center will need to be capable of supporting these technologies. Regardless of their role, both data centres must be able to dynamically provision compute resources and both must trust each other in terms of authentication and authorization.

2.2.3.1 Goals of Experiments

There are three mains goals common to all these use cases:

1. Provision new resources and reroute traffic
2. Migrate applications, data, and/or services
3. Provide no noticeable degradation in QoS for the end-user

Because of this, one of these use cases can demonstrate and assess the capability of the system in the following ways:

1. Provision of end-to-end network connectivity via an on-demand NSI-based multi-domain transit network.
2. Usage of SDN capabilities to provide/reconfigure the routes within the SDN islands (data centres).
3. Dynamic allocation of network, compute, and storage resources.
4. Migration of applications/data/services to newly provisioned resources.
5. Demonstrate high-levels of performance such that the end-user does not experience degradation in QoS.
6. Flexibility of the infrastructure to accommodate different storylines for broader applicability to different experimental requirements.

2.2.3.2 Experimental scenario and architecture

We are selecting the “disaster recovery by migrating IaaS to a remote data center (UC.5)” as the representative use-case for the infrastructure domain. The overall scenario is depicted in Figure 2.8 from the FELIX Space. First the IaaS manager makes a request for migration to the HaaS coordination system called HaaS coordinator. Next, the HaaS coordinator then requests HaaS-enabled resources. Finally a slice is provisioned and created. From the User Space, as shown in Figure 2.9, first the HaaS coordinator configures provisioned VMs and then notifies the IaaS administrator. The IaaS administrator then extends the IaaS environment over the remote date center and issues a request for migration. The HaaS coordinator releases the resources at the sender site after the migration is performed and confirmed.

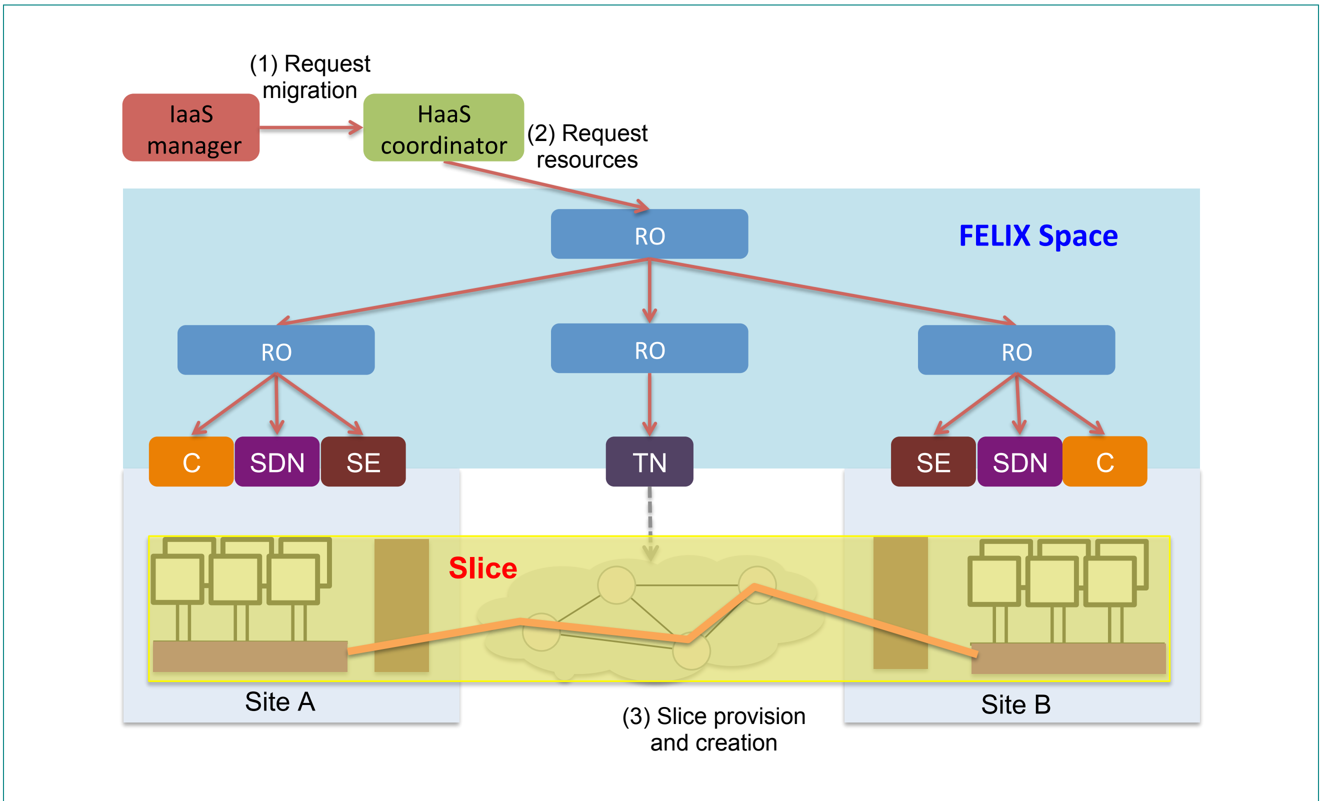

Figure 2.7: Diagram showing the steps involved in the disaster migration scenario from the FELIX Space.

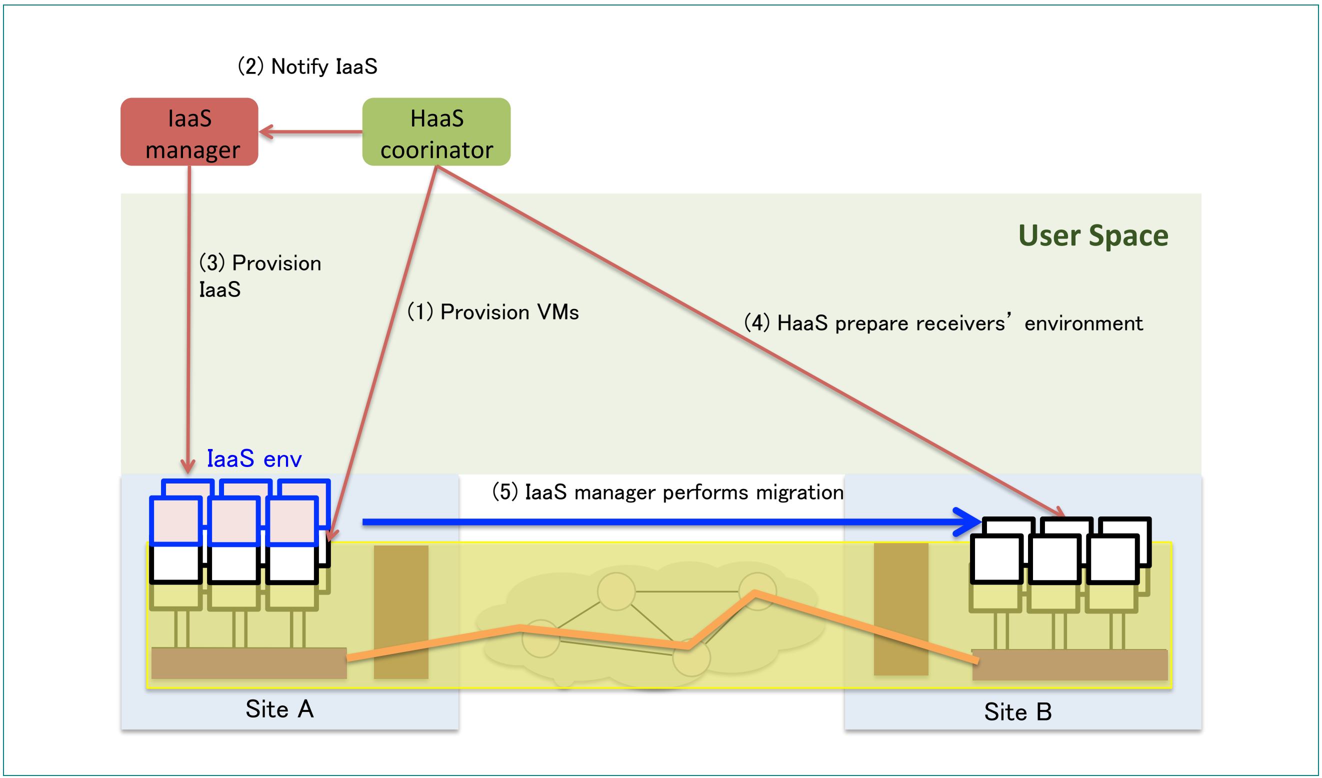

Figure 2.8: Diagram showing the steps involved in the disaster migration scenario from the User Space.

2.2.3.3 Physical infrastructure provisioned for the experiments

Figure 2.10 shows the physical compute, SDN, SE and TN resources to be used in experiments of the infrastructure domain use cases. There are two islands in the AIST domain that are identical in hardware and software specifications. The resources are provided by AIST, KDDI and PSNC.

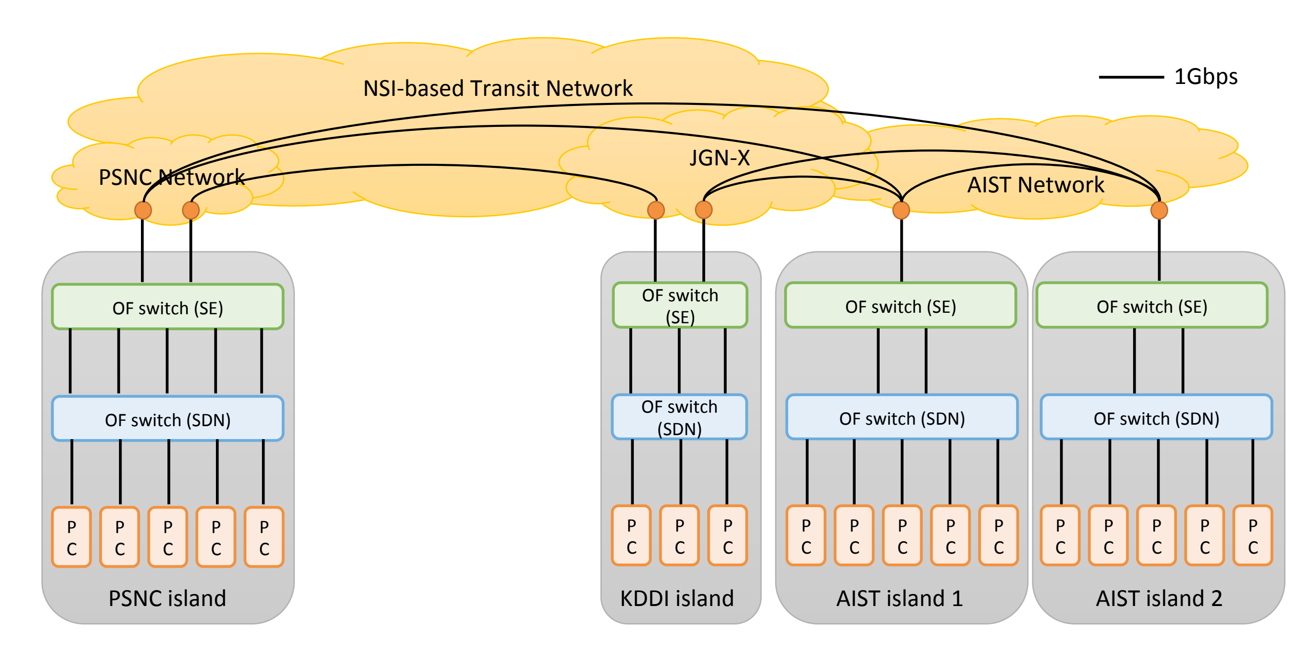

Figure 2.9: Physical infrastructure provisioned for the experiments.

2.2.3.4 Experimental Conditions

Experimental Conditions	
Involved Islands	PSNC, AIST, KDDI
Actors	A sender, which runs IaaS software on a HaaS layer at a data center. A receiver (e.g. infrastructure provider) that provides HaaS in a data center. Testbed operator and data center operator.
Requirements for use case execution in FELIX test environment	MUST requirements: Two HaaS-enabled data centers. One is a sender, which runs an IaaS software on a HaaS-enabled data center. The second is a receiver (e.g. infrastructure provider) that provides HaaS in a data center. Network, which provide a dynamic network circuit between the data centers from and to which an IaaS is migrated. The destination data center has sufficient physical resources and can emulate a physical resource environment of the migrated IaaS. Resources of the destination data center support nested virtualization technologies. DESIRED requirements: The network bandwidth between the source and destination data centers can be assured. Software acceleration, such as high performance data transfer under a high latency environment, and high-speed VM migration technologies are available.

System components	Resource Orchestrator (RO) Resource Managers: SDNRM, TNRM, SERM, KVM-CRM (HaaS-enabled) NSI-controlled network domains Experiment Control and Management Inter-data center network provider software Inter-data center resource management software HaaS coordination software
Provisioned resources	OpenFlow switches and stitching entities at PSNC, AIST, KDDI
Additional resources	IaaS manager as an end user and HaaS coordinator Traffic re-routing protocols/software.
Network capacity	1Gbps
Migration content	1-20GB VMs (depends on the VM usage)
Lifecycle	Select data centers, provision resources, initialize connections, migrate IaaS, monitor performance (check progress/behaviour), reroute traffic and examine continuation of service/QoS, shutdown experiment.
Time	1-6 hours

Table 2.3**:** Experimental Conditions Infrastructure Domain Use Cases.

2.2.3.5 *Experiment 1 – Migrating IaaS between different islands in the AIST domain.*

In this experiment, a set of VMs consisting of IaaS (IaaS VMs) managed by common IaaS management software is migrated to a remote island. First, the IaaS VMs including a management server and user VMs are deployed over VMs and network provisioned by FELIX KVM-CRM and SDNRM in one island. So the VMs provisioned by FELIX are called Layer 1 VMs and the IaaS VMs are called Layer 2 VMs.

In this scenario, after a disaster has occurred, the compute and SDN resources in a remote island and SE and TN links between the current and remote islands are allocated by using the FELIX software stack, and the L2 data plane of the IaaS is extended to the remote island. Then, the IaaS VMs are migrated to layer 1 VMs in the remote islands as shown in Figure 2.11.

We use AIST island 1 as the sender island and AIST island 2 as the remote island.

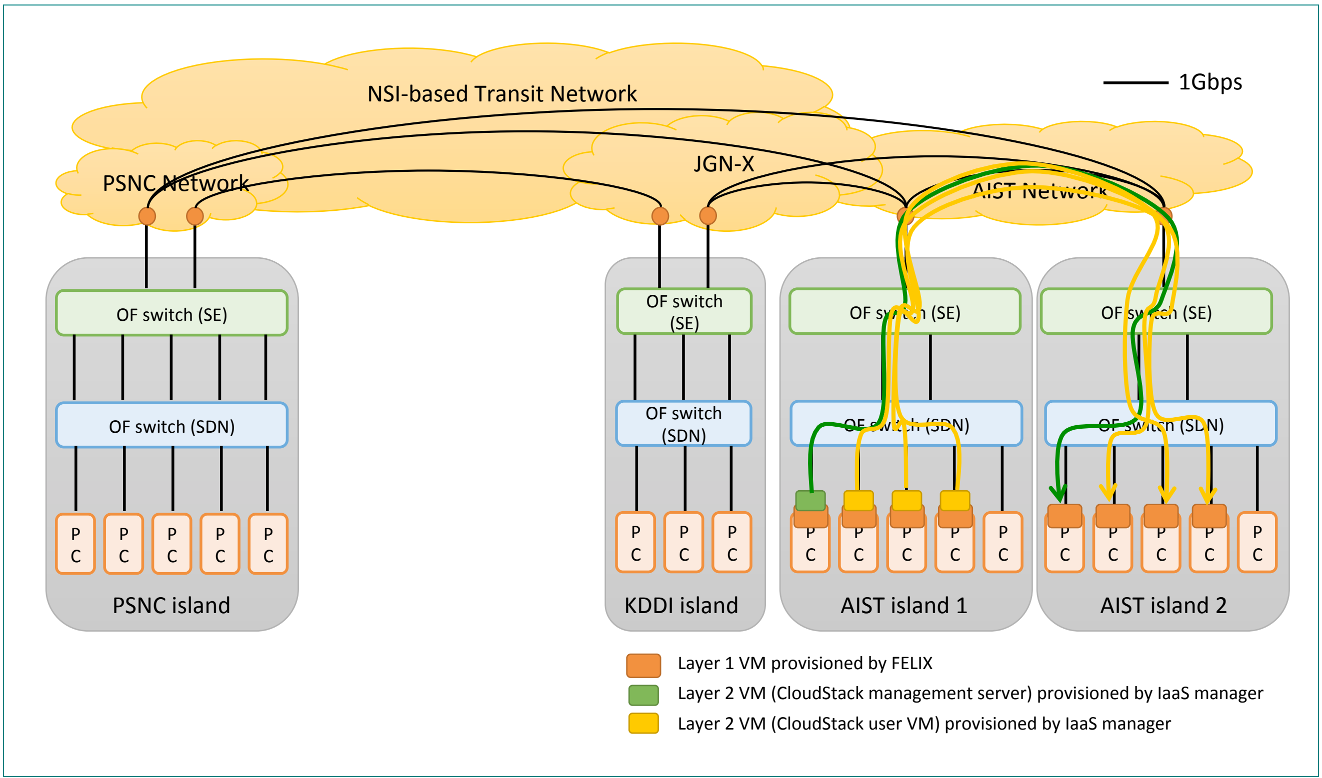

Figure 2.10: Physical infrastructure provisioned for Experiment 1.

Experiment scenario:

1. Based on the IaaS manager's request, HaaS coordinator provisions IaaS resources, which consists of four Layer 1 VMs provisioned on four different physical nodes in the AIST island 1. IaaS manager deploys IaaS environment using CloudStack, one of the common IaaS software stacks. The IaaS environment consists of one management server VM and three user VMs.
2. When the IaaS manager requests IaaS migration, HaaS coordinator configures four VMs and SDN links in the remote island, and SE and TN 1Gbps links between the current and remote islands, and extends the IaaS L2 data plane.
3. IaaS manager migrates the management server VM to a Layer 1 VM in the remote island.
4. IaaS manager makes the management server manage the three other Layer 1 VMs in the remote island.
5. IaaS manager makes the management server shutdown the three Layer 1 VMs running user VMs in the sender island, and then three user VMs are migrated to the remote island.

In order to perform rerouting of incoming traffic to the newly established data centre, the HaaS coordinator manages a set of global IP addresses for each island. The IaaS manager configures and reconfigures a user VM global IP address provided by HaaS coordinator from the FELIX user space, which then facilitates the traffic rerouting to the new data centre. The aim of the experiment is to check the feasibility of IaaS migration consisted of multiple VMs between the different islands in a stable and low latency environment. We will check the feasibility of IaaS migration over long distance network between AIST and PSNC in the next step.

2.2.3.6 *Experiment 2 – Data Mobility Service by SDN Technologies.*

As a result of merging three use cases together, KDDI will be performing their use case over the same infrastructure used for the IaaS migration experiment. In this experiment, a user's VM is migrated from a source island to a remote island

based on the response performance. At first, the management system, i.e. VPN server, performance measurement server and SDN controller on the IaaS environment is deployed over VMs and network provisioned by FELIX resource managers (CRM, SDNRM, SERM and TNRM) in one island within the FELIX infrastructure used for this experiment.
In this scenario, when the end user moves to another location and the SDN controller discovers a more suitable island for providing services to the user based on the response performance, the user VM in Layer2 VM is migrated to the remote island and the SDN controller reroutes the user's request to the remote island. Specifically, we will use the KDDI island as the source island and AIST or PSNC island as the remote island.

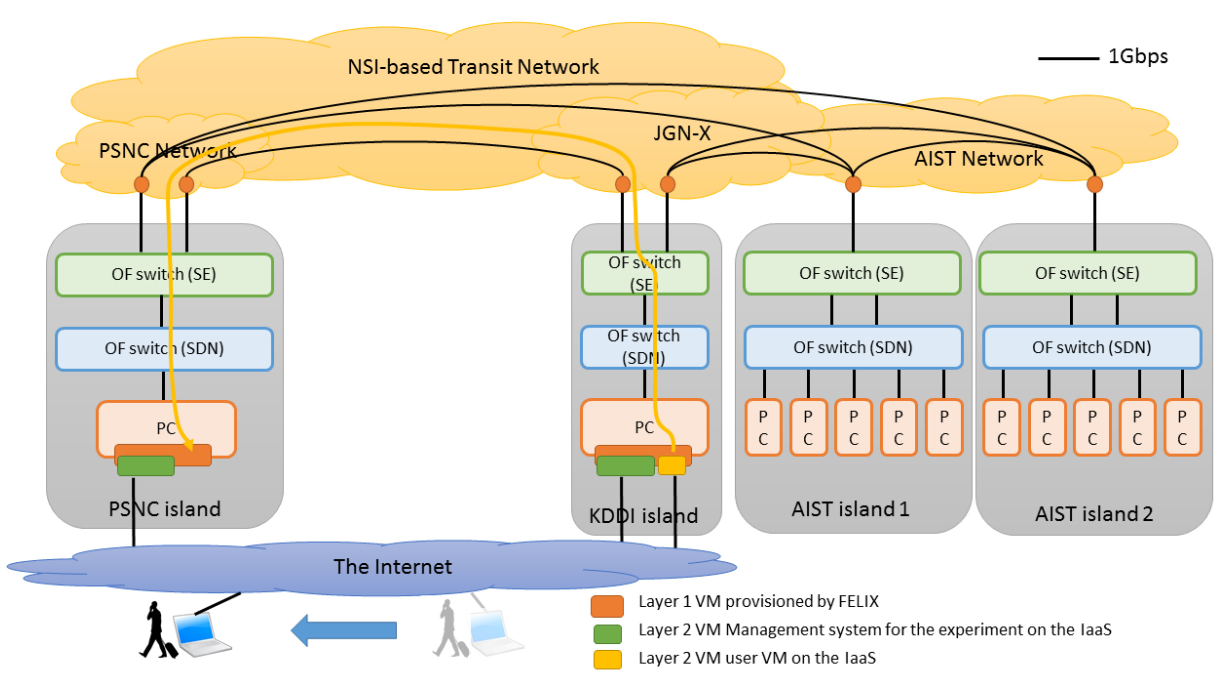

Figure 2.11: Physical infrastructure provisioned for Experiment 2.

Experiment scenario:

1. The user (i.e. service provider) deploys an IaaS environment, which consists of one Layer 1 VMs with the Internet access in the KDDI and remote island. The IaaS environment consists of four Layer2 VMs for VPN, performance measurement, internal routing and SDN controller for the management and one VM for the end user.
2. The measurement server of each island measures the response performance from the island to the end user.
3. The SDN controller checks the performance and makes a request to migrate the user VM to the remote island.
4. The user VM is migrated through the FELIX infrastructure
5. Following the completion of the VM migration, the SDN controller reroutes the end user traffic to the migrated user VM in the remote island.
6. The end user can then access their VM with no change in the performance.

The aim of the experiment is to validate the inter-cloud use case as a proof-of-concept. We will check the feasibility of user VM migration based on the user location to improve the quality of service experienced by the user.

2.2.3.7 *Experiment 3 – Follow-the-sun / follow-the-moon principles*

The third use case will also leverage on the infrastructure setup for IaaS migration (Experiment 1). As explained earlier in this experiment, active VMs are migrated from one location to another following the working hours in different time zones. First, a data center comprising three VMs is setup over the FELIX infrastructure at one location. These three VMs are Layer 2 VMs, which run different example services, i.e., a data software repository, a digital library, and a general-purpose server which runs scientific simulations. Access to these VMs is managed through a proxy server, which acts as a request forwarder to the VMs of the data center at its current location. As soon as the trigger for migration occurs the VMs are migrated to another FELIX island across NSI-enabled transit network. We plan to use PSNC's island as first location and AIST's island as the second location of the data center for migration. The VMs will be migrated from one location to the other on every trigger. Moreover, the proxy server will be realized at EICT's island.

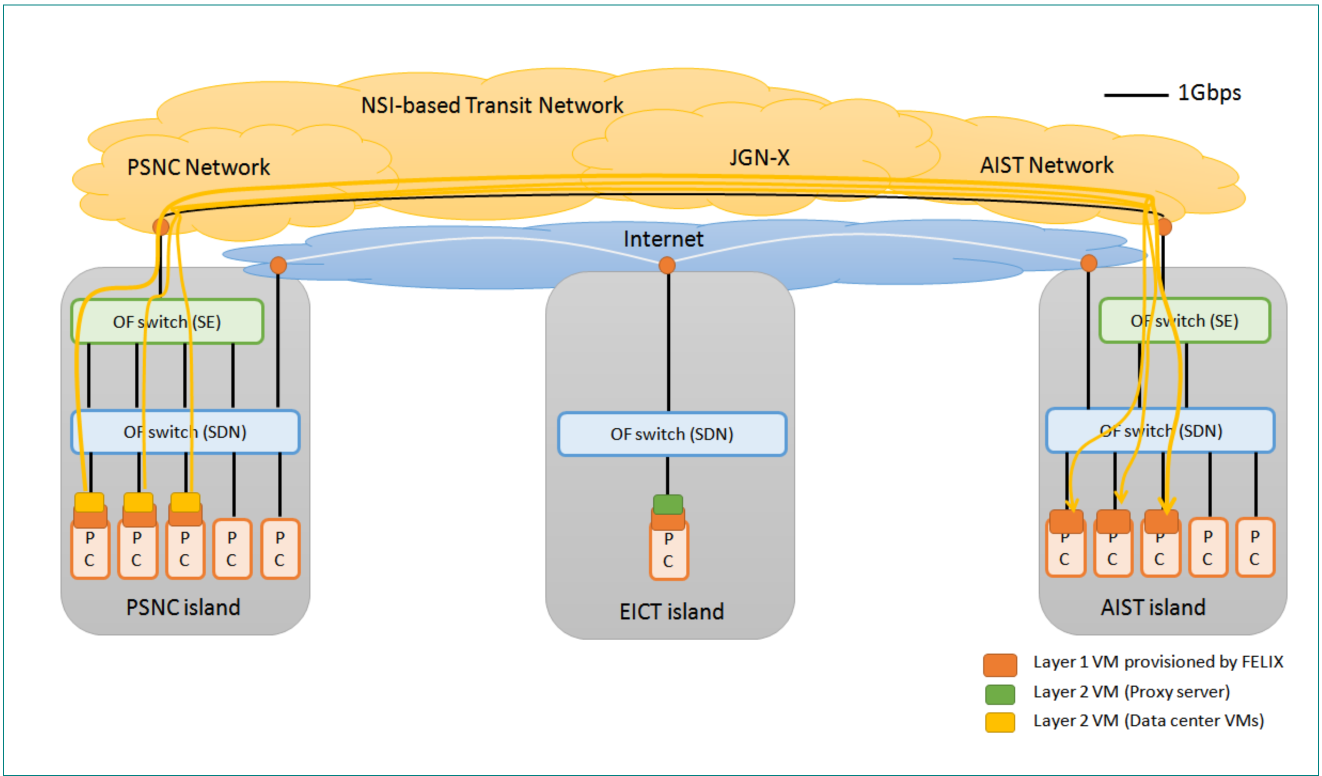

Figure 2.12: Physical infrastructure provisioned for Experiment 3

Experiment Scenario:

1. IaaS manager requests HaaS coordinator to provision three VMs (Layer 1 VMs) at PSNC's island. The needed SDN resources to network the VMs are also provisioned at this point.
2. IaaS manager deploys three data center VMs (Layer 2 VMs) on the provisioned Layer 1 VMs.
3. Proxy server is setup at EICT's island that forwards all user requests to Layer 2 VMs in PSNC's island.
4. Now the system waits for the trigger to start the migration. On this trigger, HaaS coordinator configures three VMs and SDN links in the AIST's island as well as SE and TN links between the PSNC's and AIST's islands.
5. To make migration transparent to the users, we employ snapshots of VMs that save the running states of VMs including their virtual disk data contents. After Layer 1 VMs are ready at AIST's island, the snapshots of

three Layer 2 VMs running at PSNC's island are created and along with other configuration information transferred to AIST's island over the transit network. The transferred data is used to create VMs at AIST's island and restore their running state to the point in time when snapshots were created.

6. The proxy server is now configured to forward all future user requests to the data center now running at AIST's island.
7. The resources reserved at PSNC's island are released.
8. On next trigger above steps are repeated to migrate data center from AIST's island to PSNC's island.

2.2.4 KPIs of experiments

The following table contains the list of KPIs we have identified in order to measure the degree of consecution of the Use Case implementation.

KPI	FELIX Approach
Resource composition	The FELIX The slice creation and provision should be directly managed by the FELIX management framework
Slice creation / modification / destruction time	The FELIX framework allows to automate and reduce time to set up and tear down resources over the different islands
Interconnectivity between islands using NSI	FELIX framework will be able to define links over combined SDN and NSI networks.
The FMS maintains the global view of the topology the FELIX infrastructure	FELIX framework will be able to discover/receive the complete topology in order to set up the testing environment for an experimenter.
Improve the migration efficiency and the QoS the user experiences after migration of services.	FELIX framework will provide the resources that are at the optimum position within the global infrastructure.
Determine additional data center to support the migration in a specific amount of time if resources are available.	FELIX framework will be able to automatically provide compute and network resources that are globally dispersed.
Perform data migration between two sites within a specific amount of time depending on size of data and infrastructure requirements.	FELIX framework will incorporate NSI transit networks with SDN islands to support stable, rapid migration.
User request redirection transparent to the user (i.e., user does not notice changes about serving data center)	FELIX will redirect traffic between the old and new serving data centers without requiring any configuration changes at the user end.

Table 2.4: KPIs Infrastructure Domain Use Cases

3 Planned FELIX Use Cases

3.1 Data Pre-processing on Demand

In this section we introduce a new experiment, resulting from merging the two remaining Data Domain Use Cases, i.e. Data on Demand and the Data pre-processing. The motivation for merging them into one is to focus validation efforts, as these use case share common functionalities that can be tested into a single storyline and environment.

A basic requirement for the merge is to fulfil the original objectives of both use cases. For this reason, we have dedicated the first part of the section to analyse and obtain an overview of the storyline and goals of both scenarios. This complements and extends the contents in deliverable D2.1 [D2.1], where we highlighted the specific conditions that underlie the design of the use cases.

After the analysis of the common objectives, we present the merged use case, i.e. the Data pre-processing on Demand. The first key point is the comparison of the similarities among them in terms of preconditions and proposed achievements. Simultaneously, the peculiarities of each use case will be pointed out as a starting point for the development activities. The architecture and workflow proposed for the Data pre-processing on Demand use case are both introduced.

At the end, we list the physical infrastructure and software expected for the deployment of the Use Case and also a number of KPIs to evaluate the degree of consecution of the deployment of the UC.

3.1.1 Original Use Case Descriptions

3.1.1.1 UC.1 – Data on Demand

This Use Case, as well as others belonging to the Data Domain, focuses on the dynamic interconnection of dispersed Software-Defined Networking (SDN) enabled Data Storages or Data Centres. The virtual infrastructure to be used for interconnecting the domains consists of the SDN-enabled infrastructure per domain, which shall be interconnected through dynamic circuit-switched networks.

The aforementioned connectivity is typically used for collaborative investigations (e.g. astronomical observations), generating large amounts of data throughout distributed sites and usually storing data in the same site where these were generated. Anyone interested in processing that raw data shall retrieve it first from each storage site and then apply their algorithm (Data Processor). However, retrieving the whole data at once may not be neither possible nor efficient; instead a possibility is to retrieve only the required amount of data at each moment.

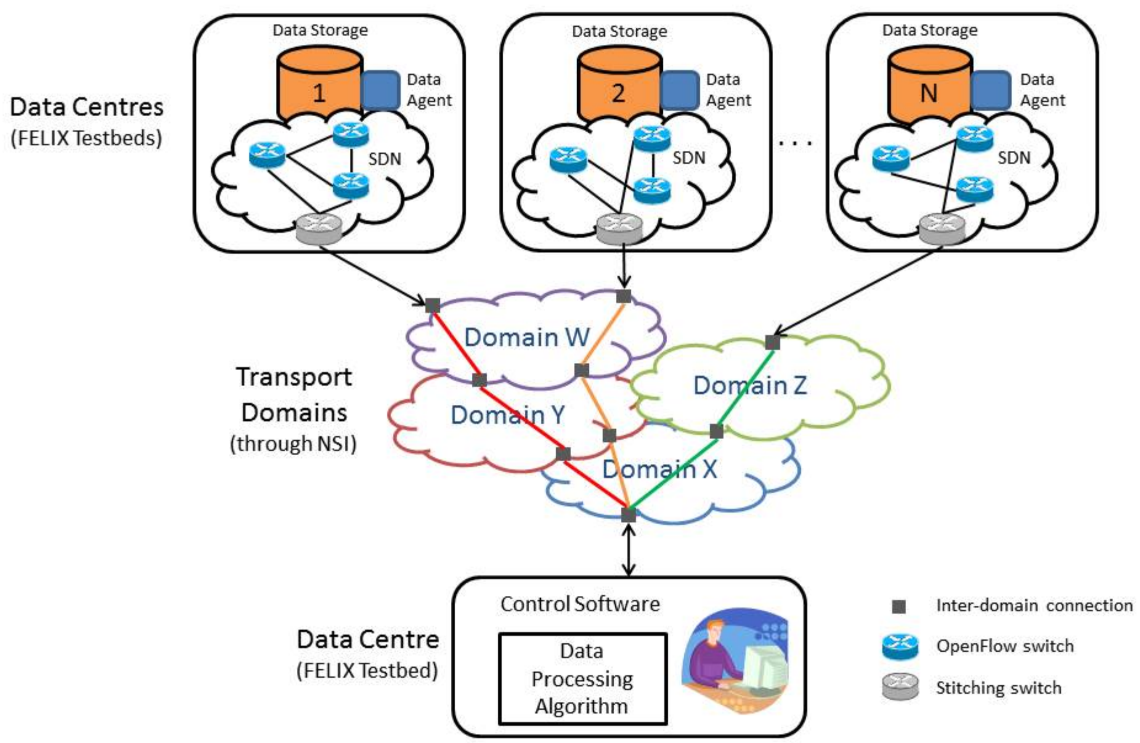

Figure 3.13: Data on Demand Overview.

When the experiment starts, the Data Processing algorithm (running in the Processing Site) requests some initial data. An SDN controller identifies at every time the appropriate link between the Data Processing site and any given Storage Site, so the Data Agent (within the Processing Site) provides the requested data through it. In parallel with the processing, the Controller will enable specific paths when further data can be handled. Figure 3.1 presents an overview of the scenario. In order to set that up, the FELIX stack shall enable fast communication between infrastructures and the SDN Controller within the Data Processor must request and set appropriate flows of data in order to obtain the expected data section at every moment of the processing procedure.

Goals for UC.1

The main goal of this use case is to validate the dynamic reconfiguration of the network connections provided through NSI and SDN. Other capabilities offered by the FELIX testbed are also necessary and will be used and validated as well, such as the creation of computing and networking slivers, its approval, the contextualisation of credentials in the machines, etc.

3.1.1.2 UC.2 – Data Pre-processing

As already introduced in deliverable D2.1, the Data pre-processing use cases demonstrates the SDN capabilities to interconnect testbeds distributed across different continents dynamically. In the following section, the storyline and the objectives of this use case are briefly recalled.

Here, the transfer of the nearly real-time data, from the sources where they are produced to the distant destinations in which they will be consumed, is mostly affected by the Round Trip Delay (RTD). Indeed, the RTD value is significantly higher in the data transmissions over the public Internet, which is largely based on the best-effort principle. In order to reduce the network degradation, the FELIX system can be used to allocate computing, caching and networking resources both in the transit domain and in the islands based on the SDN paradigm.

This use case shows the satellite images which are generated by the ESA Earth portal. The huge flow of data can be processed in a data centre closed to the producer and then compressed with proper algorithms to reduce the size of the sources to be transmitted (see Fig. R2). In other words, the traffic load that the transit network should support has enormously decreased; thereby improving the performance of the whole system. In this context, the FELIX framework can reserve the computing resources at the source and destinations islands and can schedule the configuration of the network paths on demand allowing the experimenters to choose the minimum required bandwidth or the maximum guaranteed delay.

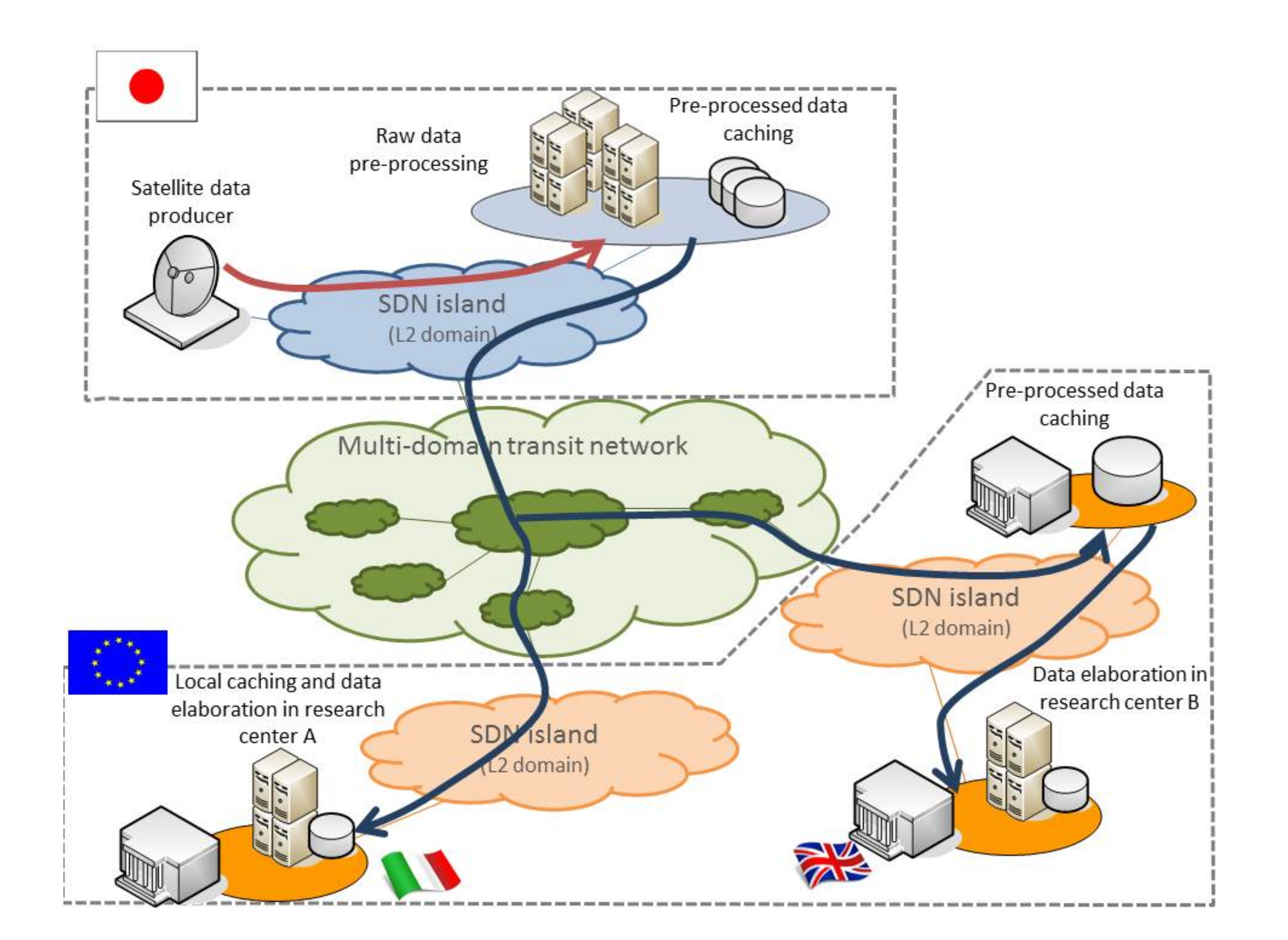

Figure 3.14: Data Pre-processing High Level Overview.

Goals for UC.2

The goal of the Data pre-processing use case is to validate the coordinated control of computing and network resources, that is, allocating and provisioning different kinds of resources (i.e., computing, SDN, stitching or transport resources) and providing an on-demand and application-driven end-to-end network connectivity service interconnected across domains.

3.1.2 Updated Storyline

This section summarizes previous description and goals in order to converge into a common storyline per each scenario.

3.1.2.1 UC.1 – Data on Demand

An astronomer is testing an improved algorithm to detect gravitational collapses of stars within the Milky Way from astronomical observation sources taken by several telescopes and astronomical observatories. In order to validate it, she constructs a test set out of images obtained from different observatories located in Asia and Europe. Provided she will be testing in her own computer, storing the whole set of images in a single place is not affordable. Instead, she decides to perform sequential processing of the data set and work on a limited set at each time. For this, she constructs a small framework to coordinate the consecutive request and sequential processing of the data set.

3.1.2.2 UC.2 – Data Pre-processing

A computer vision engineer is working together with an environmental scientist to conduct a study on the desertification suffered in Eurasia during the few last decades and attempt to infer future trends. For that matter, they collaborate on the development of algorithms to automatically identify specific traits that are associated with land where water is scarce and is near to become a desert. The ESA Earth Observation portal is the source they use for retrieving satellite data. To finalise the study, they need to retrieve a large number of images to feed their application. The two experimenters in this scenario can find a number of servers located in Asia storing pre-processed, compressed images which they plan to use to feed the algorithm. To fetch those, they develop some application and controller to request the needed network and computing resources to respectively set up the paths to the servers and run their algorithm on the images, once retrieved.

3.1.3 Experiment description and components

From the storylines of each Use Case, it can be observed that the Data Domain Use Cases operate on a fairly similar manner. Their main goal is to validate the dynamic establishment of the connections, and for that, both require a coordinated usage of multiple, heterogeneous resources; namely computing nodes for large data processing and network resources to forward traffic according to the rules provided by the experimenter, or implicitly within the facility. The storyline and workflow of execution naturally differs between the two of them; regarding, for instance, the location and number of endpoints that request or store large amounts of data. Yet, both demonstrate the dynamic configuration and set-up of network connections in the FELIX testbed.

We consider it is appropriate to merge the previous use cases into a single one. To achieve this, the goals of both use cases shall be fulfilled; but a different workflow and architecture will be provided to encompass the two of them.

3.1.3.1 Goals of Experiments

There are a number of goals that, whether directly sought or indirectly required, are common to the two Data Domain Use Cases:

1. Transfer data through SDN and transit network domains
2. Combined management of computing and network resources
3. Dynamic reconfiguration of network paths

Besides the common goals, the two original Use Cases have similarities on some of the required components, such as the need of data agents or post-processing algorithms.

To achieve these goals, the merged Use Case will use modules, components or tools to provision end-to-end network connectivity on transit networks and for dynamic set-up of network paths through SDN-enabled equipment; as well as the FELIX stack itself to provide coordinated management on different resources, and any other data processing algorithm, agent and controller will be developed.

3.1.3.2 Experimental scenario and architecture

The environment to support the merged Use Case shall consist of several applications or components, namely (i) satellite data producer, (ii) raw data pre-processing agent and caching, (iii) computing and networking management system, (iv) data post-processing agent, (v) graphical interface (GUI), and (vi) network controllers.

The satellite data producer is the entity responsible for generating nearly real-time data. The producer will contain images from the ESA Earth portal (https://earth.esa.int/web/guest/home, where different kinds of images can be browsed, such as products by ESA or Third-Party missions and instruments by topic, typology, processing level, etc.). We consider the satellite data producer as a dedicated data centre that stores large sets of images, where each of its servers keeps independent subsets of images in an unstructured manner.

The different sets of images can be processed and compressed in order to be efficiently transmitted in the transport network. This is the goal of the raw data pre-processing agent, which uses a dedicated data pre-processing algorithm to reduce the size of the data. The agent also provides the produced output so that it can be stored (cached) into a proper repository or storage node. The network management system is responsible of requesting and setting up the paths in both the SDN domain and the transit network. To realise its purpose, the set up can be performed within an application of the SDN controller (e.g. a POX controller) that manages the slice. So it can easily contact the SERM and the TNRM modules to set-up the inter-islands links using e.g. preconfigured RSPECs of the GENIv3 interface.

When the SDN paths and end-to-end connection are set up, the pre-processed images could be moved from the source to the destination that has requested the images for experimenting. In the destination, the incoming data can be finally post-processed to e.g. produce images for the final experimenters, or the experiments can directly work on them for further analysis or post-processing steps.

Figure 3.16 shows the overall scenario of the Data pre-processing on Demand use case.

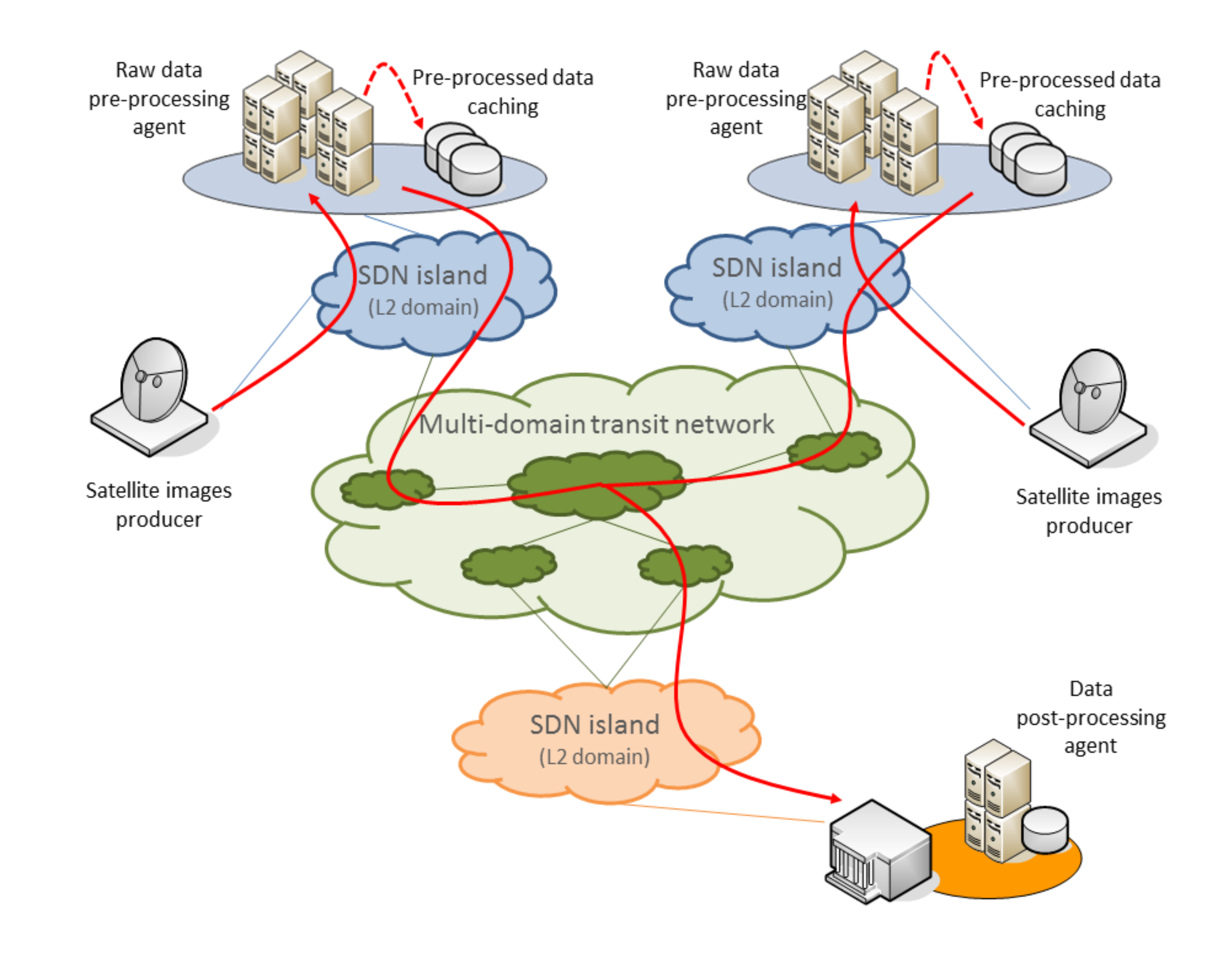

Figure 3.15: Data pre-processing on Demand scenario.

The figure above does not show most of the system components, such as the FELIX framework, exact storage, caching or processing nodes or the user interface; but these are listed in the table under the "Experimental conditions" section.

In the specific case of the interface, it will be developed to facilitate the conduction of the Use Case. It is devised to be a simple interface to allow sending commands to the different entities, to track the events following the workflow and, at the end, to highlight the results.

3.1.3.3 Physical infrastructure provisioned for the experiments

The deployment of this Use Case foresees the setup of a multi-domain slice composed of computing resources and SDN, stitching and transport network resources. This slice includes several devices located at different testbeds of the FELIX infrastructure, namely at PSNC, i2CAT and AIST islands. We plan to reserve around 4 or 6 virtual machines running on the physical servers and between 2 and 7 OpenFlow-enabled switches. At least two virtual links should be preconfigured through the SERM and TNRM modules in order to establish the connectivity between the satellite data centres (or islands) and the research data centre through the transport network domain. Moreover, between one and three SDN controllers (e.g. a standard POX) can be installed in specific running VMs to inject the flow entries in the physical equipments and to realise the scenario on-demand.

As a fundamental prerequisite, some software components of the FELIX stack must be up and running in order to manage the physical infrastructure properly. Also, some of the resources provisioned by the management stack may be required to be running prior to starting the Use Case. As per the software managing the stack, we expect to use at least one Master Resource Orchestrator (MRO) to act as a single and central point of contact to forward the requests for the slice management, i.e the creation, configuration, start-up, etc. Also, each island is equipped with one Resource Orchestrator (RO) to coordinate the flow of the requests for the Resource Managers. Indeed, the SDN, Stitching Entity (SE) and Computing (C) resource managers must be available per island to handle the different kind of resources of the FELIX system. Finally, the Transport Network Resource Manger (TNRM) is deployed within the AIST facility and provides a standard (i.e. GENIv3) interface towards the NSI domain

3.1.3.4 Experimental Conditions

Experimental Conditions	
Involved Islands	PSNC, i2CAT, AIST
Actors	The research data centre, where the experimenter's algorithm will be run. Two far-away satellite data centres, where both raw and post-processed data is stored.
Requirements for use case execution in FELIX test environment	MUST requirements: Three IaaS-enabled data centres. Two of them store raw and post-processed satellite images (data sources), and the third retrieves the images and runs the algorithm to elaborate the final data. Dynamic network circuits must be enabled between the satellite and research data centres. The research data centre must have enough physical resources to operate on the transmitted data and run the required machines to elaborate on such data. DESIRED requirements: The network bandwidth between the source and destination data centres can be assured.
System components	Resource Orchestrator (RO) Resource Managers: SDNRM, TNRM, SERM, XEN-CRM/KVM-CRM NSI-controlled network domains Experiment Control and Management Data agents within storage sites at data centres Intra-satellite data centre data pre-processing algorithm Inter-data centre data elaboration algorithm
Provisioned resources	Virtual machines and flowspaces for OpenFlow switches and stitching entities at PSNC, i2CAT, AIST
Additional resources	IaaS manager as an end user Traffic re-routing protocols/software
Network capacity	1Gbps

Media content	2GB for the whole set of ESA images (less upon pre-processing)
Lifecycle	Obtain minimum RTT, select satellite data centre, initialise connections, start resources, transfer and post-process data in satellite data centre, monitor status of process, transfer and analyse post-processed data in research data centre, shut down experiment.
Time	Around 15-30 minutes

Table 3.5: Experimental Conditions for the Data Pre-processing on Demand Use Case.

3.1.4 KPIs of experiments

The following table contains the list of KPIs we have identified in order to measure the degree of consecution of the Use Case implementation.

Specific Goal / Criteria	FELIX Approach
Resource composition	The FELIX slice allocation, provisioning or perform operational action over the resources should be directly managed by the FELIX management framework.
Slice creation / modification / destruction time	The FELIX framework allows to automating and reducing time to set up and tear down resources over the different islands.
Interconnectivity between islands using NSI	FELIX framework will be able to define links over combined SDN and NSI networks.
Provide the user with a global view of the topology the FELIX infrastructure	FELIX framework can discover the complete topology and provide the user with it, so as to set up the environment.
Determine minimum delay between satellite and research DCs	The FELIX framework will provide easy operations to determine availability of the ROs and RMs in order for the user to identify the delay.
Perform data transfers inside a satellite DC	FELIX framework will allow dynamically setting up the SDN paths to transfer the experimenting data between machines located within a single domain (satellite DC).
Allow traffic between heterogeneous domains	The FELIX framework will take care of identifying and forwarding traffic from SDN domain to the transit network domain and vice-versa.

Perform data transfers between dispersed domains, according to size of data and infrastructure requirements	The FELIX framework allows requesting links through different types of transit networks (NSI, etc.) to interconnect dispersed domains (satellite-to-research DCs).

Table 3.6: KPIs for the Data Pre-processing on Demand Use Case.

4 Conclusions

Firsts use case (HQ Media Transmission) implementation and experiments execution have been successfully and showed the potential of the FELIX framework for research and testing.

The chosen technologies and the distributed architecture of the components deployed and federated over several SDN islands demonstrated how flexible is and how different use cases is it possible to implement on. This way through experimentation the architecture of the project is validated.

During the execution of the experiments we are still learning lessons and collecting a feedback to continue improving and developing the different components. Next deliverable D4.3 will include the final results and lessons learned during the full execution of the WP4.

5 References

[D2.1] http://www.ict-felix.eu/wp-content/uploads/2014/04/FELIX-D2.1.pdf, online, available 28.04.2015.

[Qureshi, Asfandyar et al., 2009] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. Cutting the electric bill for internet-scale systems. *SIGCOMM Comput. Commun. Rev.* 39, 4 (August 2009), 123-134. DOI=10.1145/1594977.1592584.

[OpenStack] https://www.openstack.org/.

[CloudStack] https://cloudstack.apache.org/.

6 Acronyms

BCP	**Business Continuity Planning**
IaaS	**Infrastructure as a Service**
UPS	**Uninterruptible Power Supply**
FPS	**Frames Per Second**
GUI	**Graphical User Interface**
RTT	**Round Trip Time**
RTD	**Round Trip Delay**
HaaS	**Hardware as a Service**
VM	**Virtual Machine**
QoS	**Quality of Service**
QoE	**Quality of Experience**
KPI	**Key Performance Index**
GENI	**Global Environment for Network Innovations**
SDN	**Software Defined Networks**
RO	**Resource Orquestrator**
MRO	**Master Resource Orquestrator**
RM	**Resource Manager**
SDNRM	**Software Defined Network Resource Manager**
TNRM	**Transit Network Resource Manager**
SERM	**Stitching Entity Resource Manager**
KVM-CRM	**Kernel Virtual Machine Computation Resource Manager**
NSI	**Network Service Interface**
DC	**Data Center**

The scientific/academic work is financed from financial resources for science in the years 2013 - 2016 granted for the realization of the international project co-financed by Polish Ministry of Science and Higher Education.