# XIFI

**eXperimental Infrastructures for the Future Internet**

# Network Controller Overview

*24/06/2014, Madrid*
*Luis M. Contreras*
*Telefónica*

# www.fi-xifi.eu

FUTURE
INTERNET
PPP

SEVENTH FRAMEWORK
PROGRAMME

# Outline

- Objective and context for the Network Controller

- Component overview

- Status

- Next steps

# Outline

- Objective and context for the Network Controller

- Component overview

- Status
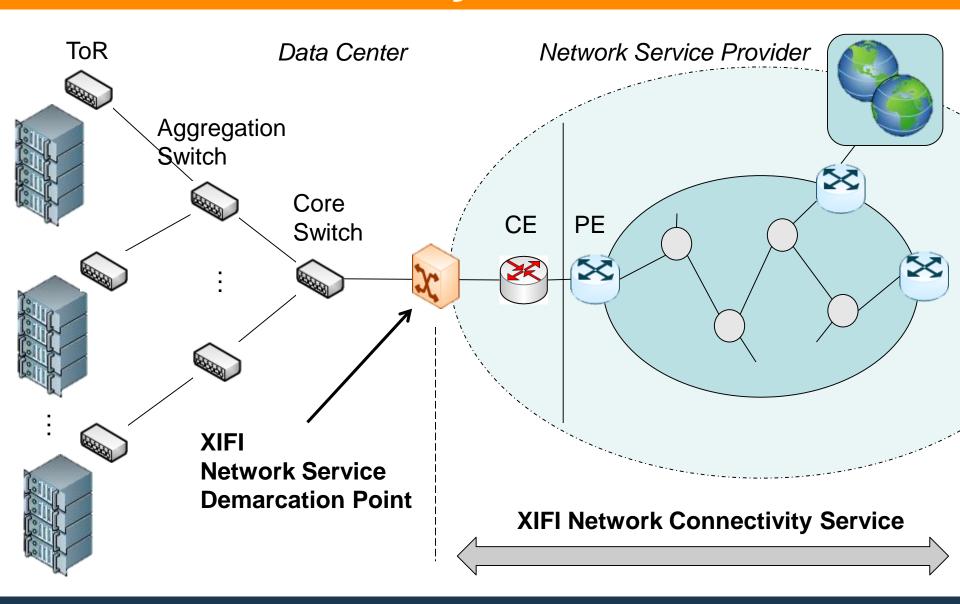
- Next steps

# Objective

- Infrastructure adaptation mechanisms to provide unified control and access to network components of the infrastructure

- Support of dynamic interconnection across distributed pilot sites to perform large-scale tests of FI applications
  - ensuring that network performance parameters are suitable to support access to services with a controlled quality level, extended seamlessly across infrastructures
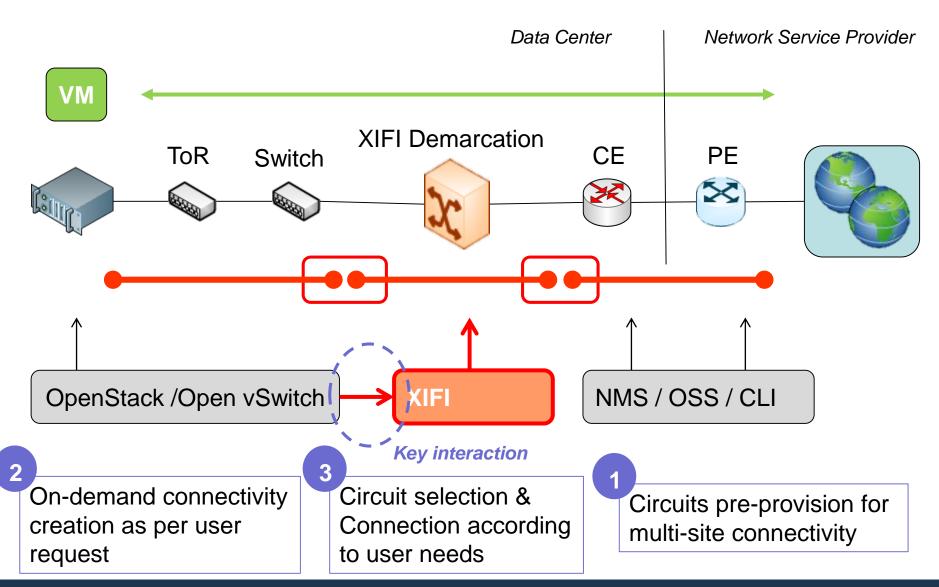  - end-to-end coverage to experimentation deployments

# Context

- No possibility of directly configuring NRENs production nodes / networks

- No possibility of configuring Data Center's CE routers (natural Border Routers)

- Then a new element (or group of elements) is deployed in between allowing for network control and management capabilities

- Such a node will be a XIFI demarcation point for connectivity service
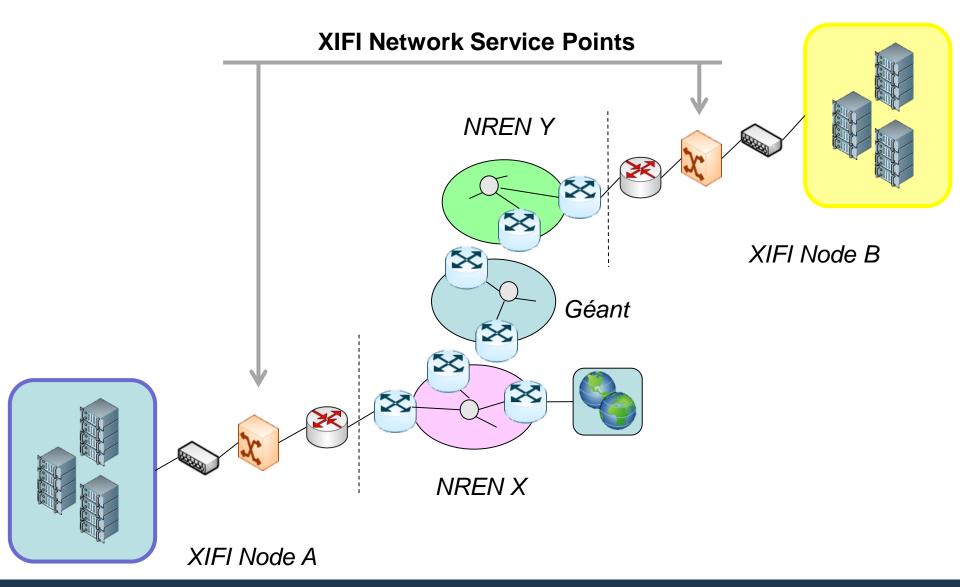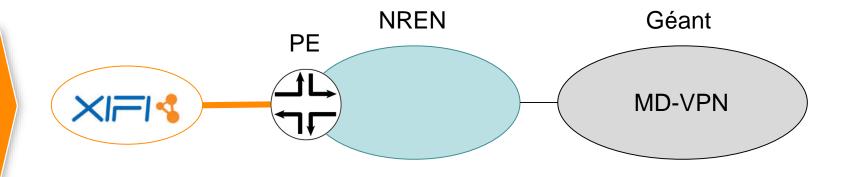
# XIFI Connectivity Service



ToR

*Data Center*

*Network Service Provider*

Aggregation Switch

Core Switch

CE   PE

**XIFI Network Service Demarcation Point**

**XIFI Network Connectivity Service**

# Stitching

**XIFI**

Data Center | Network Service Provider

**VM**

ToR    Switch    XIFI Demarcation    CE    PE

OpenStack /Open vSwitch → **XIFI**    NMS / OSS / CLI

*Key interaction*

**2** On-demand connectivity creation as per user request

**3** Circuit selection & Connection according to user needs

**1** Circuits pre-provision for multi-site connectivity

# XIFI Connectivity Service



**XIFI Network Service Points**

NREN Y

Géant

NREN X

XIFI Node A

XIFI Node B

# Scenarios of Integration



**Case A**

PE — NREN — Géant

XIFI ——— [PE] ——— NREN ——— MD-VPN

**Case B**

NREN — PE — Géant

XIFI — [PE] ——— MD-VPN

**Case C**

Internet — PE — Géant

XIFI ——— Internet ——— [PE] — MD-VPN
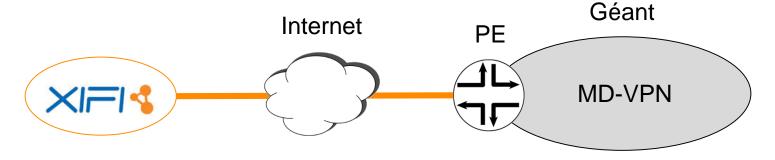
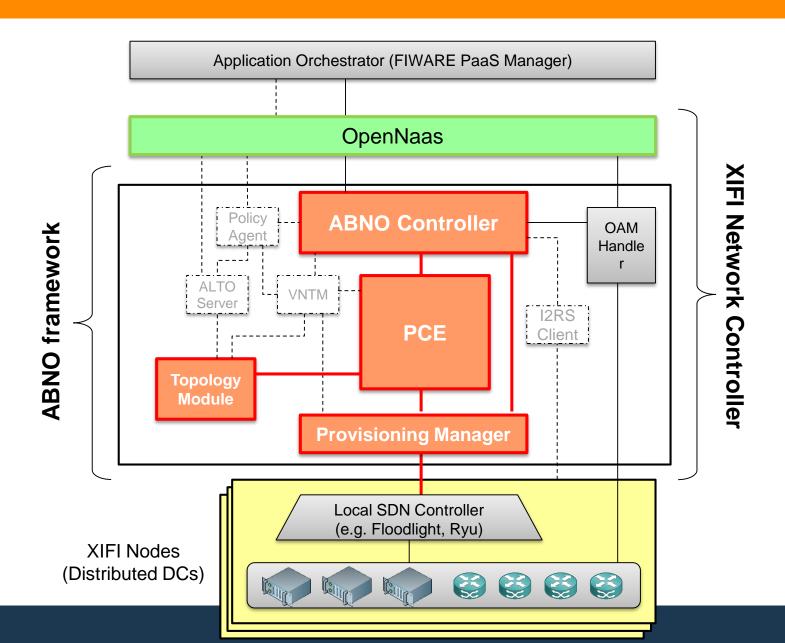# Outline

- Objective and context for the Network Controller

- Component overview

- Status

- Next steps

# ABNO Framework

- ABNO stands for Application-Based Network Operations

- Proposal being defined in the IETF
  - \<draft-farrkingel-pce-abno-architecture-07\>

- Framework based on existing protocols

- Facilitates on-demand and application-specific reservation of network connectivity, reliability, and resources
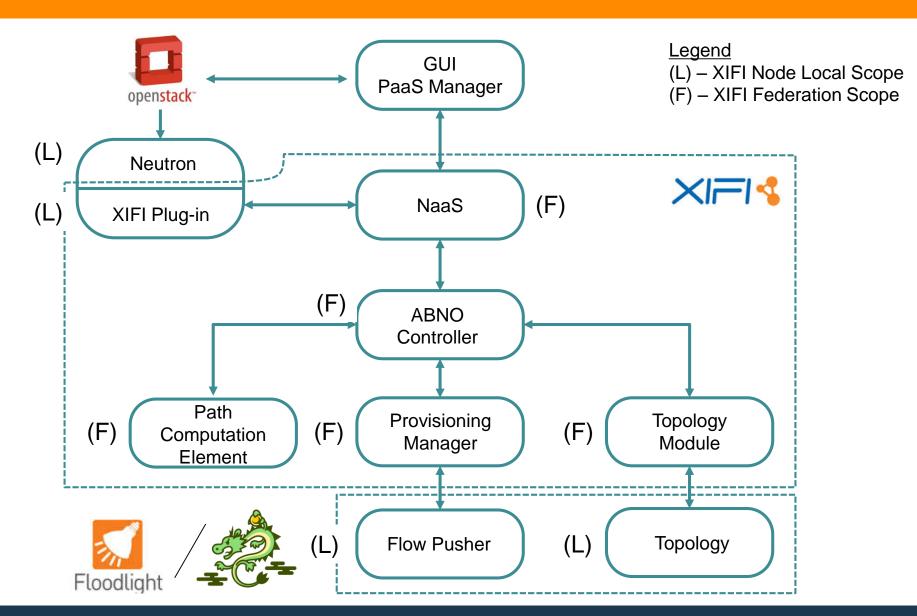
- OpenNaaS to provide the <u>service awareness</u> of the XIFI Network Controller

  - Internal DB structures for storing service information pointing to local deployments in each DC

- Keep the association between local resources in the DCs

  - Net_IDs, MAC addresses, ports, OVS_IDs, IP addresses

  - Region_ID to be considered in the final API

- Facilitate the modification and deletion of the service

- Future developments could integrate this information with monitoring capabilities to offer a view of service(/customer) affection
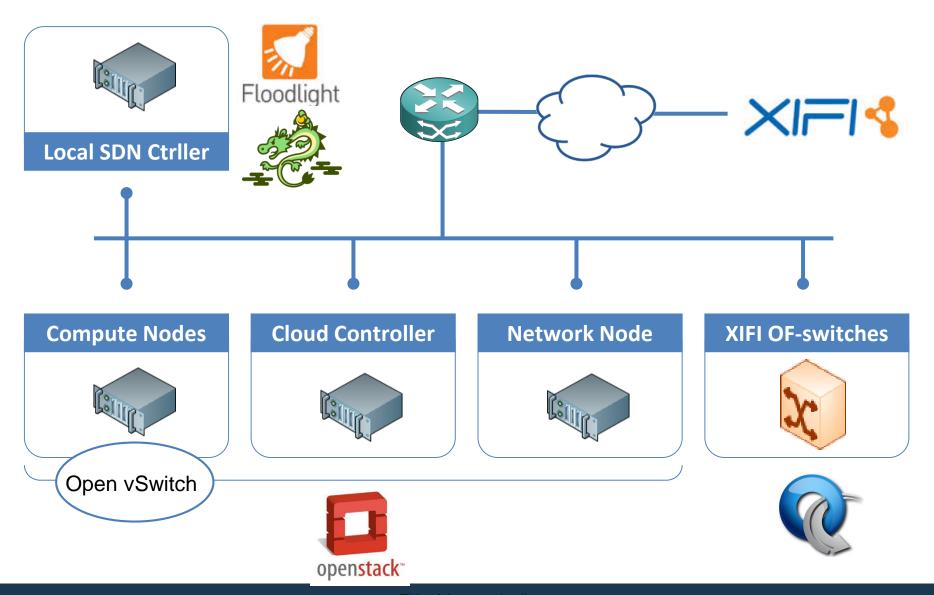
# Implementation



**Legend**
(L) – XIFI Node Local Scope
(F) – XIFI Federation Scope

GUI
PaaS Manager

openstack™

(L) Neutron

(L) XIFI Plug-in

NaaS (F)

(F) ABNO Controller

(F) Path Computation Element

(F) Provisioning Manager

(F) Topology Module

Floodlight /

(L) Flow Pusher

(L) Topology

# Data Center Components



Local SDN Ctrller

Floodlight

XIFI

Compute Nodes

Cloud Controller

Network Node

XIFI OF-switches

Open vSwitch

openstack

# Outline

- Objective and context for the Network Controller

- Component overview

- Status

- Next steps

# Where we are now

- Integration of the Network Controller with Havana

  – Adaptation of the Neutron ML2 plug-in

- Preliminary integration with RYU controller

  – RYU Manager 3.8

  – Adaptation of the Provisioning Manager component to work with RYU

  – Some problems found with message types and sizes

    - Some messages received by RYU controller from the OF nodes triggered exceptions producing malfunction

    - Infinite while loop entered when processing TLVs
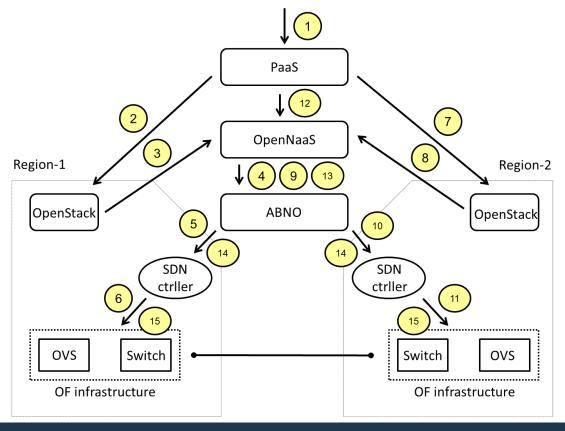
# Where we are now

- On-going integration with OpenNaaS and PaaS
  - Integration between ABNO part and  PaaS was intended for the M12 demo
    - Misalignments were raised during this integration
    - PaaS requirements are not fully covered by the controller at this time (e.g., access to metadata)
  - New strategy: joint integration with OpenNaaS and PaaS to ensure alignment on requirements and parameters among the components
  - An stable mock-up will be setup at Telefónica labs premises for ensuring this integration

# Where we are now

- ## On-going integration with OpenNaaS and PaaS
  - Review of the workflows to ensure consistency: Creation / Modification / Deletion

# Additional work

- First experience with OpenDayLight playing the role of local SDN controller

  - Creation of a specific ODL dispatcher to retrieve the information about the topology, including nodes and links information.

  - First stages in the development of a parser to push flows to the switches using the ODL controller.

  - ODL seen as a potential evolution for the controller

    - XIFI outcomes could be contributed to ODL project in the future

# Outline

- Objective and context for the Network Controller

- Component overview

- Status

- Next steps

TM2.3-Connectivity solutions

# Next steps

- Work on Neutron L3 agent for Havana

- Integration with the Geant MD-VPN overlay connecting the XIFI DCs

  - Functional definition, service requirements and PoC proposal

    - Pending to start from last meeting → URGENT ⚠️

- Integration with the OFERTIE use case

  - Exposition to OFERTIE components of the QoS capabilities of the local SDN controllers

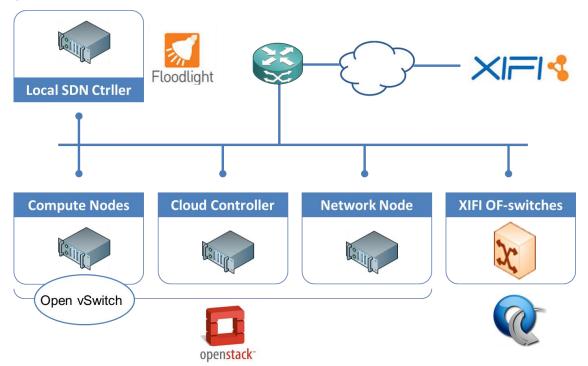- Transition to Ice House

# Next steps

- Resilience / reliability
  - Define resilience methods against network events
    - Nodes outage (either OVS or physical nodes)
    - Local SDN Controller outage
    - XIFI Network Controller outage
  - Disaster recovery
  - Workflow required for reliability?
    - To check potential failure cases
    - To decide how to implement / include the necessary logic to recover the network connectivity

# Open points

- Deployment plan
  - Master node for the XIFI Network Controller
  - Integration on the Administration Network linking all the components
    - OpenStack instances, local SDN controllers, OF-switches, etc

# Guidelines to be released

- Operational guidelines necessary to introduce XIFI network control
  - Introduction of the XIFI Network Controller in the federation nodes
    - How to make the Network Controller work
    - Master node(s) and normal nodes
    - Link to external connectivity
    - Roll-back
  - Introduction of new nodes into the federation
    - Declaration of new nodes to be controlled

# Evolution of the controller XIFI

- Distributed architecture

  - Deployment in each DC of ABNO and OpenNaaS components for local connectivity

  - Integration with central entity for e2e services

  - Reliable design for prevent controller outages

- Consideration of new functionalities offered by OpenStack (Networking API v2.0)

  - VPNaaS, experimental for Havana release

  - Heat component (orchestration capabilities)
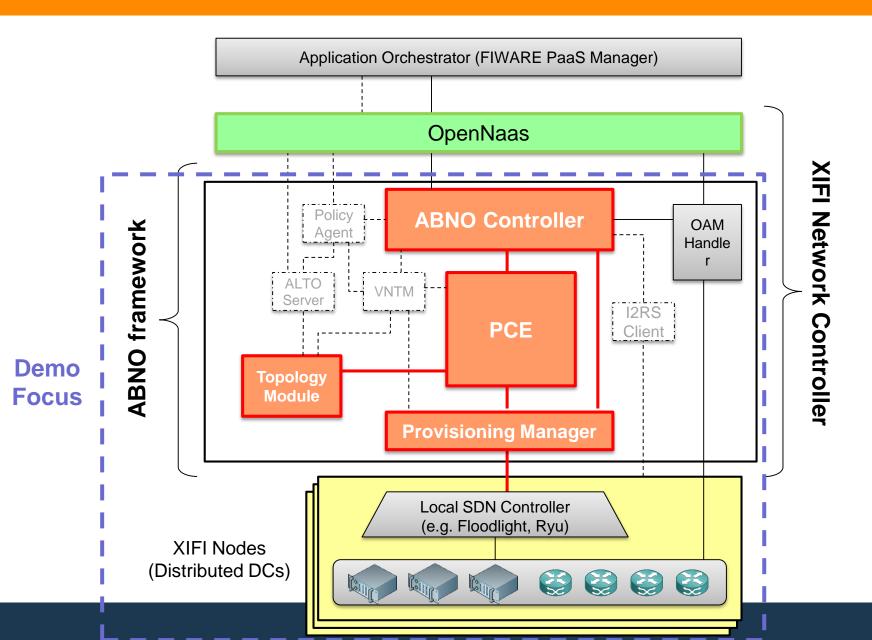
- OpenDayLight

# Thank you for your attention!

# XIFI Network Controller

# Physical set-up

XIFI DC1 | XIFI DC3 | XIFI DC2

Plug-in
Nwk Ctrller

Plug-in

Plug-in

*mordecai* | *one2* | *rigby*

eth1 | eth1 | eth1

A19 | A19 | A19

350 → | ← 351 | 348 → | ← 349

A20 | A20 | A21

*xifi1* 1G | *xifi3* 1G | *xifi2* 10G

349 | 347

local | local

international
(via Paris & Frankfurt)

Acknowledgment to Telefónica
Global Solutions for providing
the L2VPN international circuit