POLITECNICO DI TORINO

Tesi di laurea magistrale

User-oriented Network Service on a Multi-domain Infrastructure

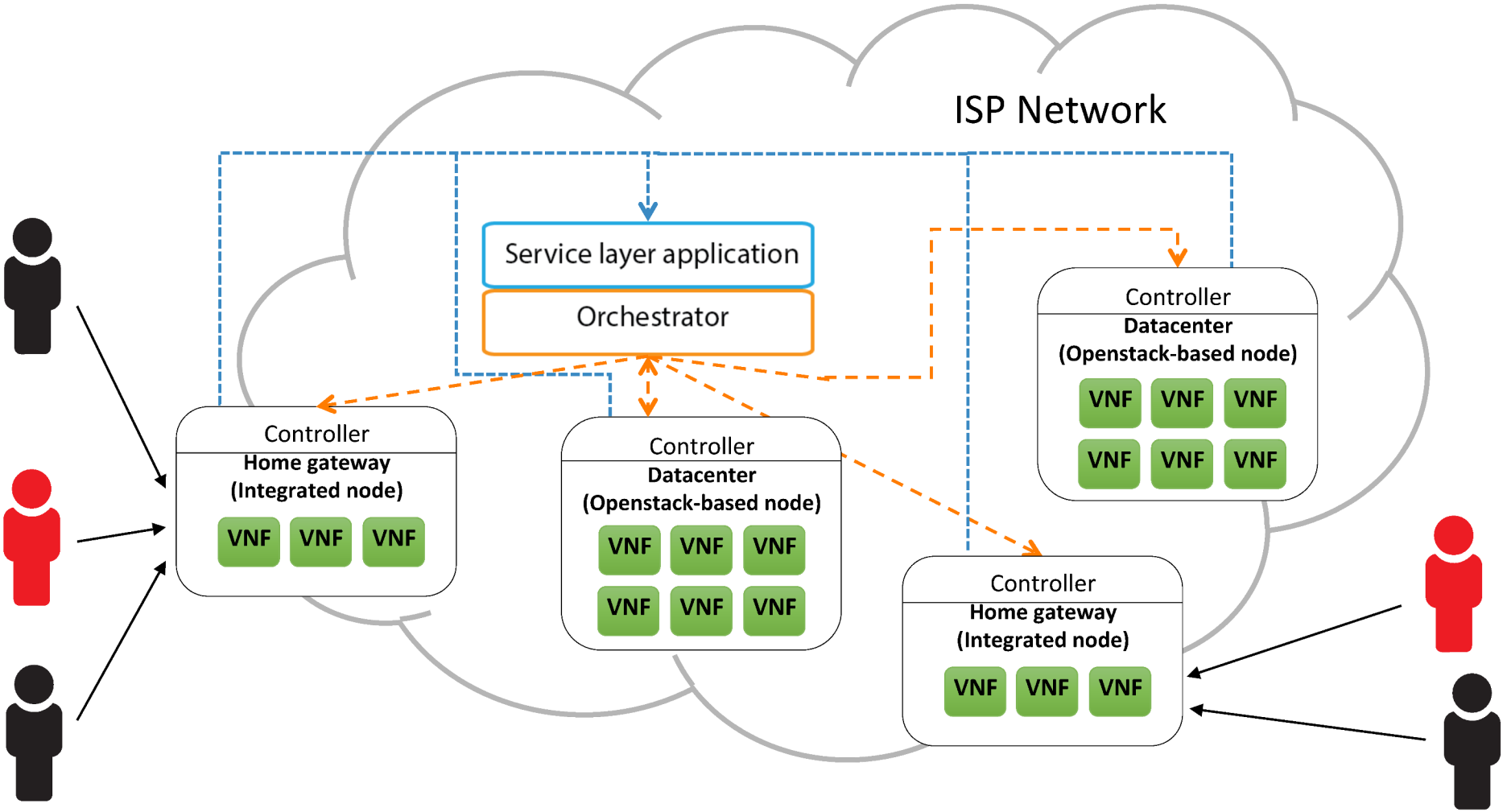
Supervisor
prof. Fulvio Risso

Candidate
Fabio Mignini

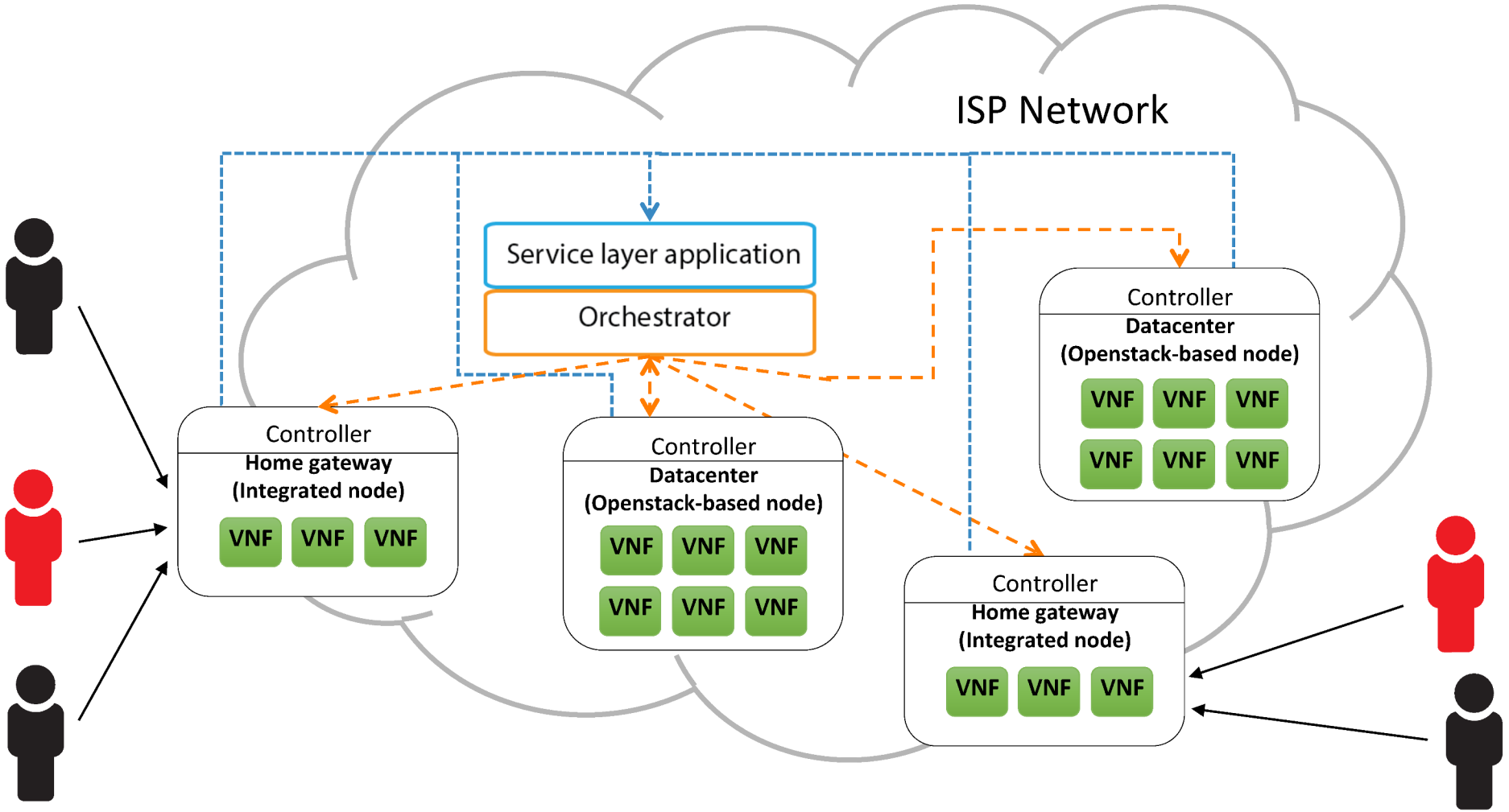
December 2014

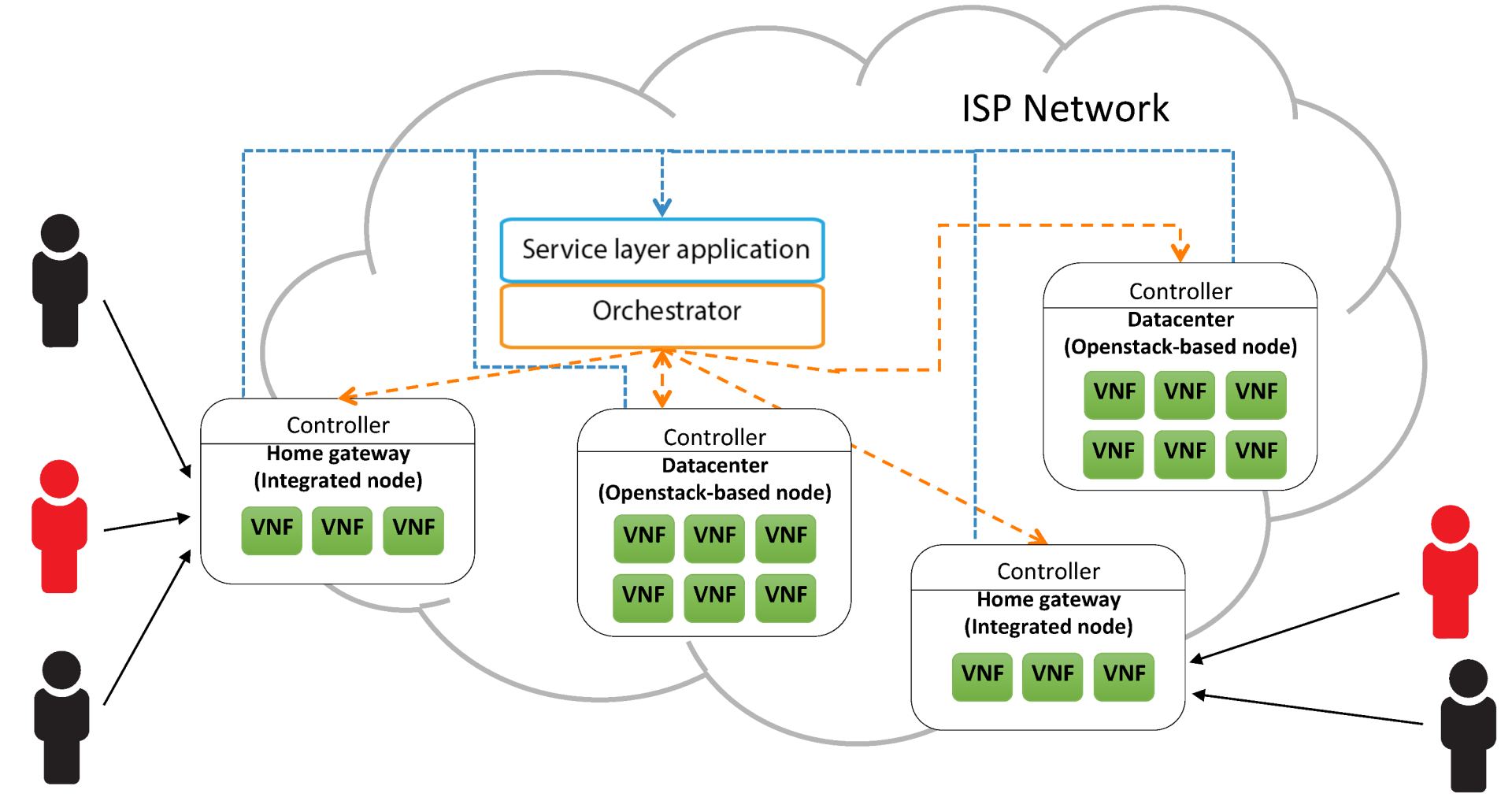
Objectives

VNFs deployment scenario

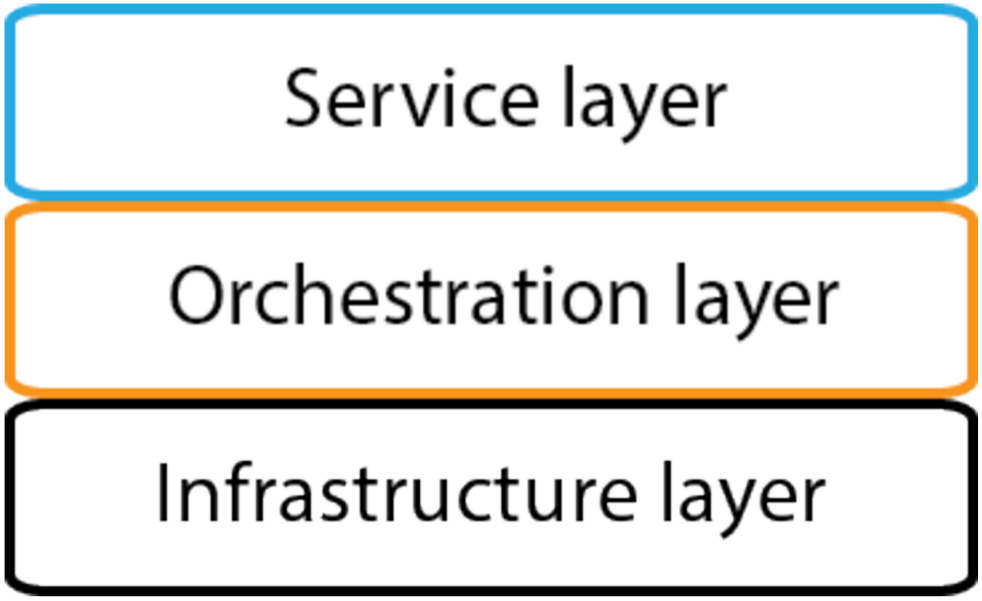


Scenario

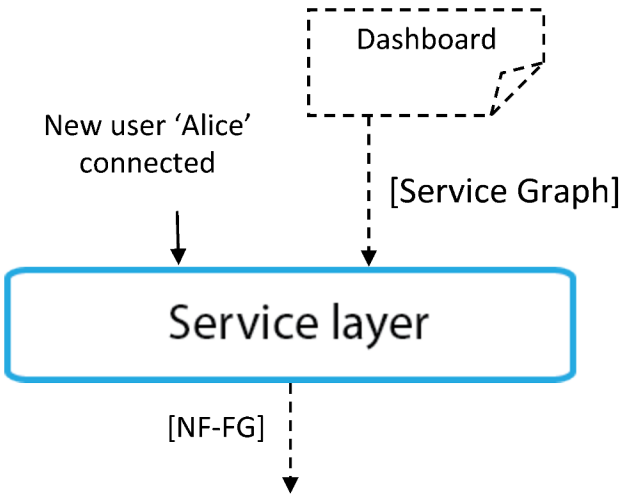

ISP Network
Service layer application
Orchestrator
Controller
Home gateway
(Integrated node)
VNF
VNF
VNF
Controller
Datacenter
(Openstack-based node)
VNF
VNF
VNF
VNF
VNF
VNF
Controller
Datacenter
(Openstack-based node)
VNF
VNF
VNF
VNF
VNF
VNF
Controller
Home gateway
(Integrated node)
VNF
VNF
VNF


ISP Network
Service layer application
Orchestrator
Controller
Home gateway
(Integrated node)
VNF
VNF
VNF
Controller
Datacenter
(Openstack-based node)
VNF
VNF
VNF
VNF
VNF
VNF
Controller
Datacenter
(Openstack-based node)
VNF
VNF
VNF
VNF
VNF
VNF
Controller
Home gateway
(Integrated node)
VNF
VNF
VNF

General architecture



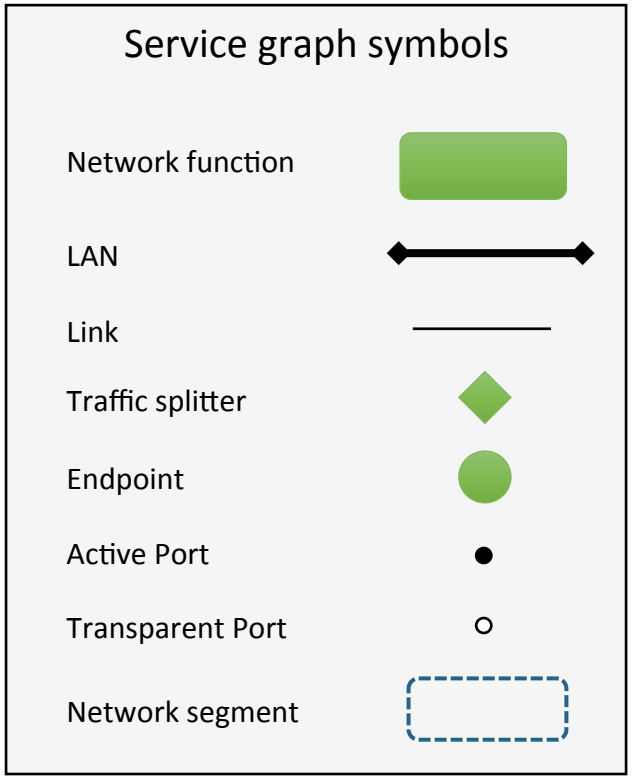
Service layer

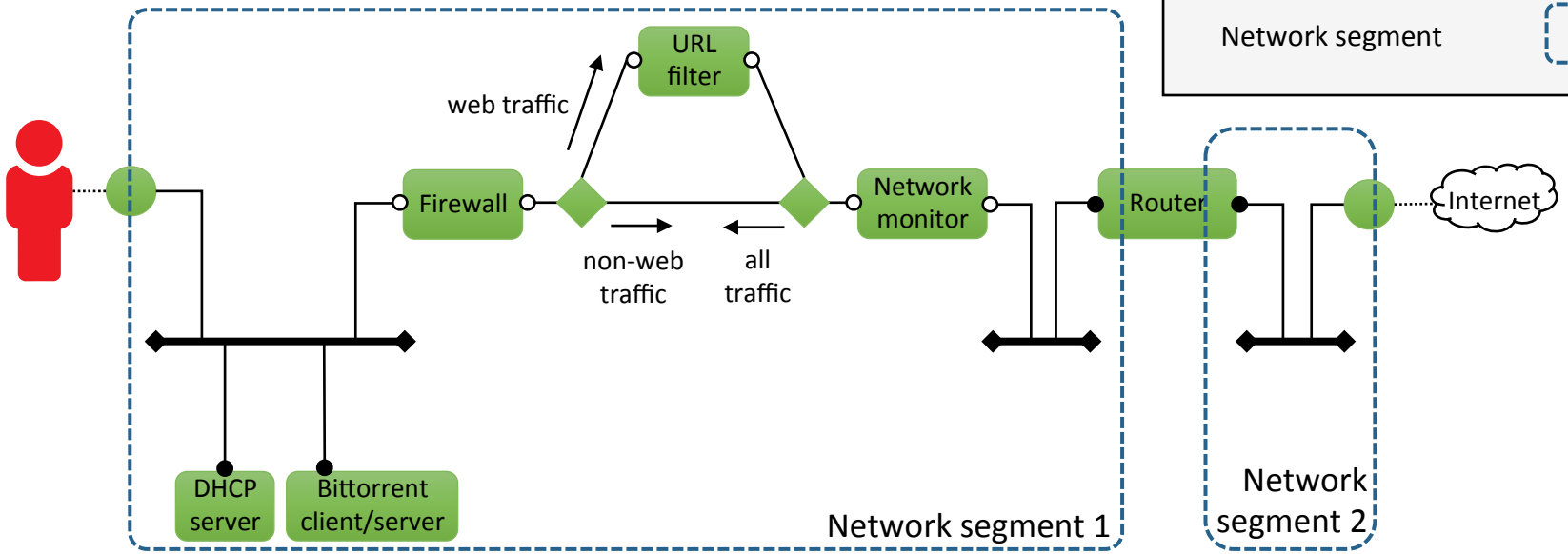


- The **service layer** represents the external interface of our system and allows the different actors that can potentially use our solution to define their own network services.
- The input of this architectural part is hence a per-actor service description, expressed in an high level formalism called *service graph (SG)*.
- The service layer should be able to translate the service graph specification into an orchestration-oriented formalism, namely the *forwarding graph (FG)*.
- It implement also the logic of connection of graphs of multiple user

Service graph

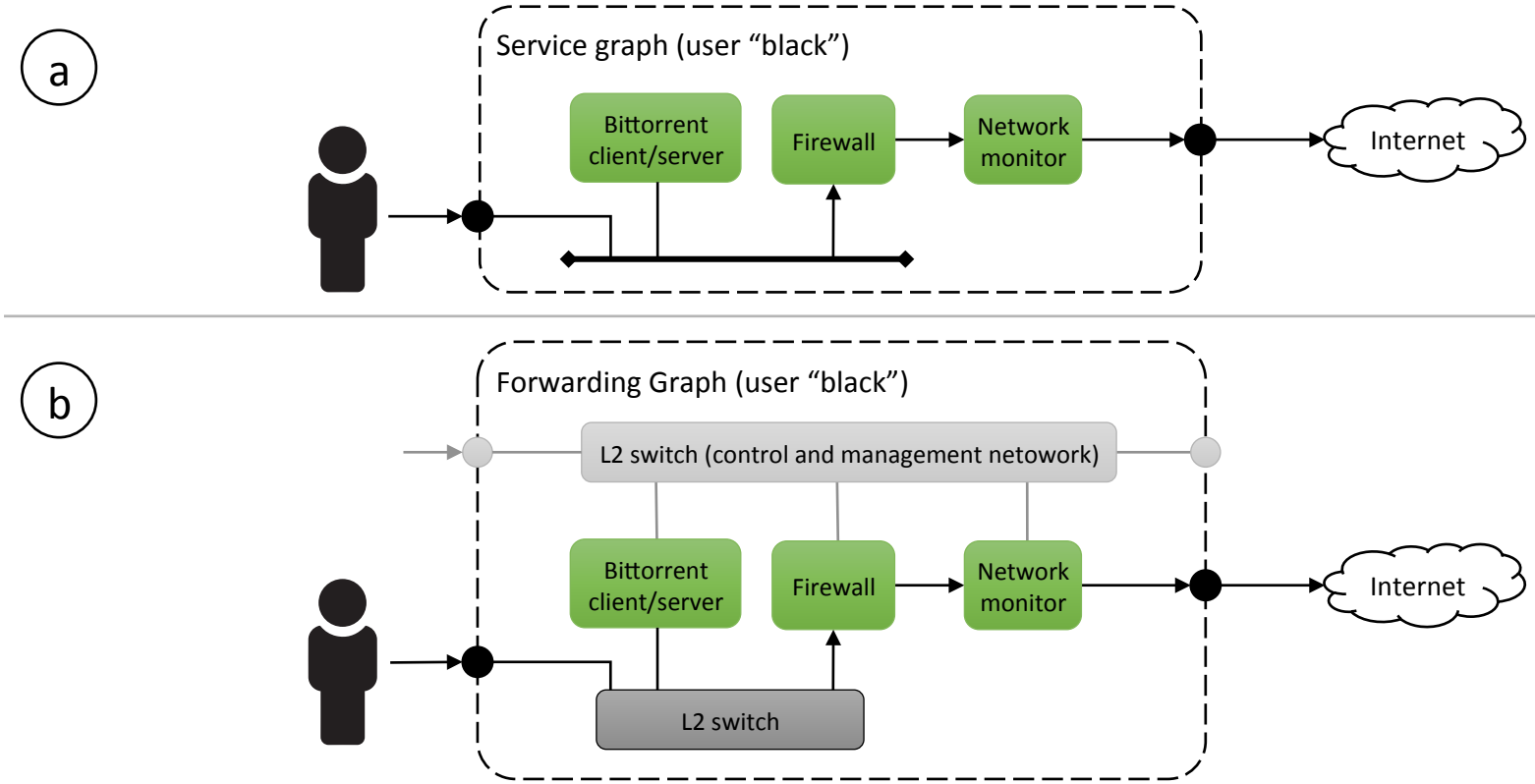
- The **service graph (SG)** is an high level representation of the service to be implemented on the network, and it includes both aspects related to the infrastructure and to the configuration of these network functions.





Forwarding graph

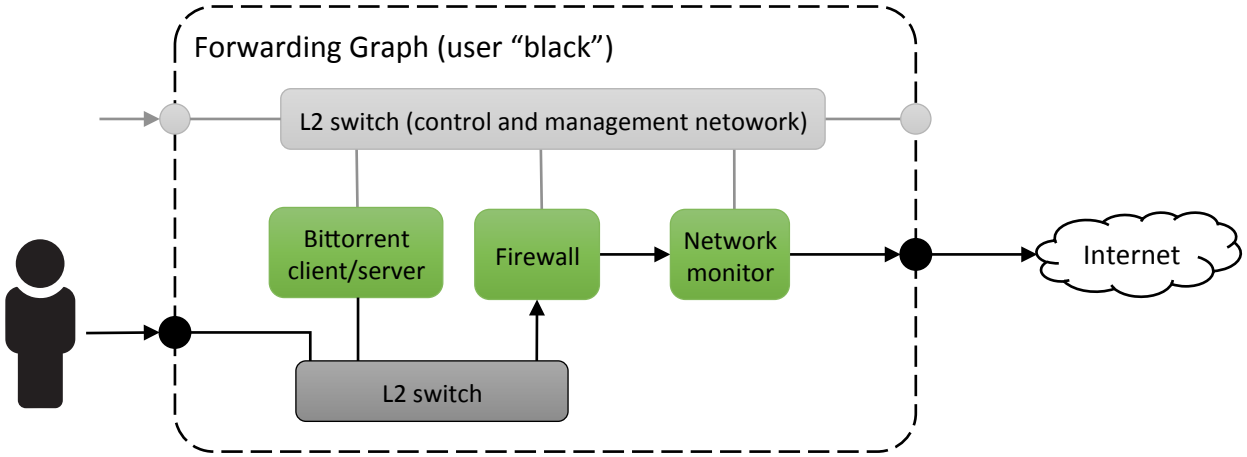
- The **forwarding graph (FG)** is a resource oriented representation obtained from the SG through the so called *lowering process*.



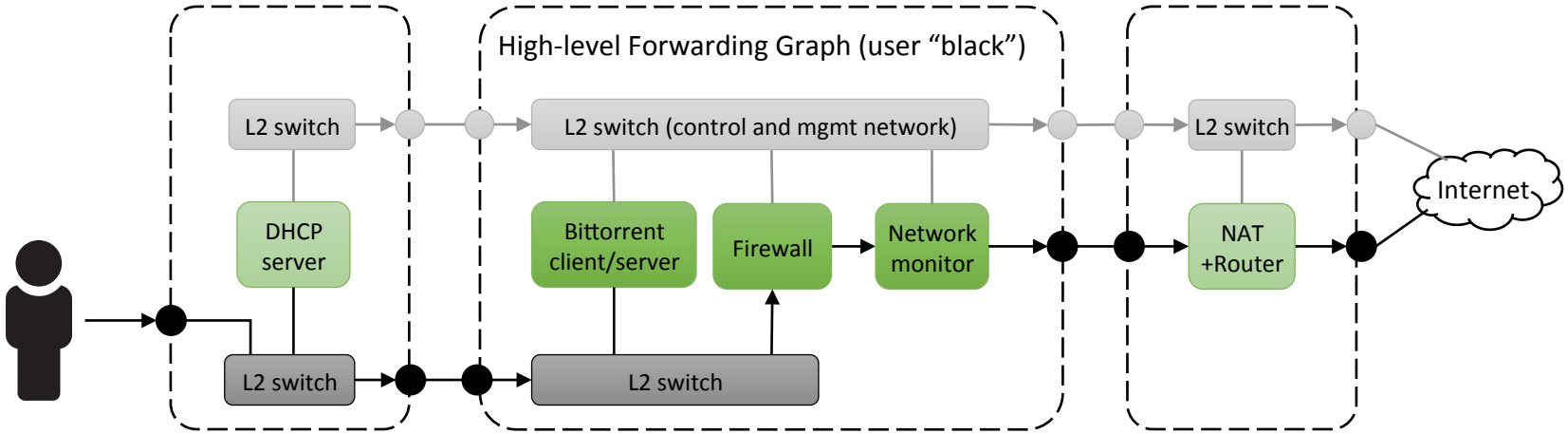
VNF template

Forwarding graph – lowering process

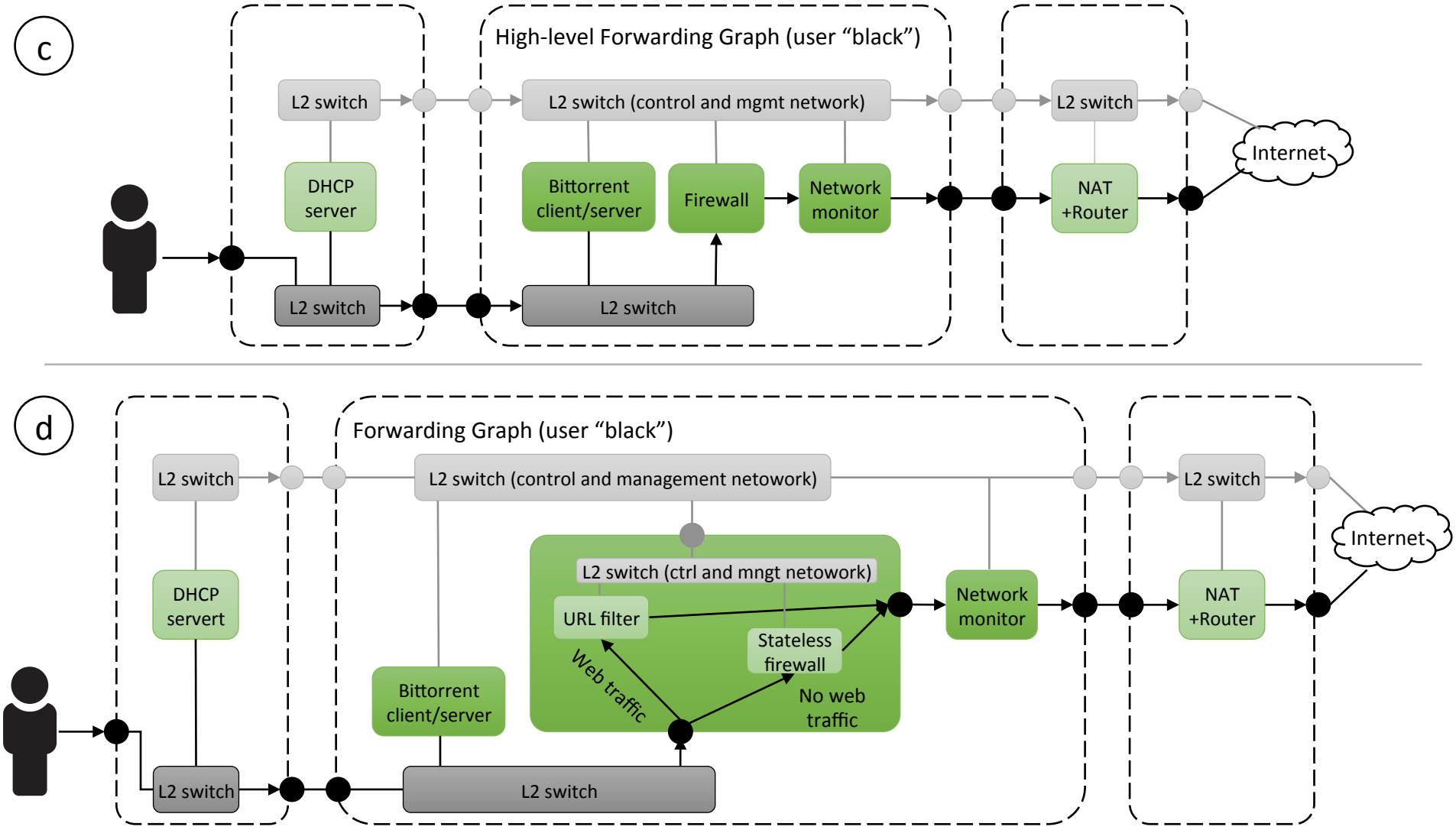
b



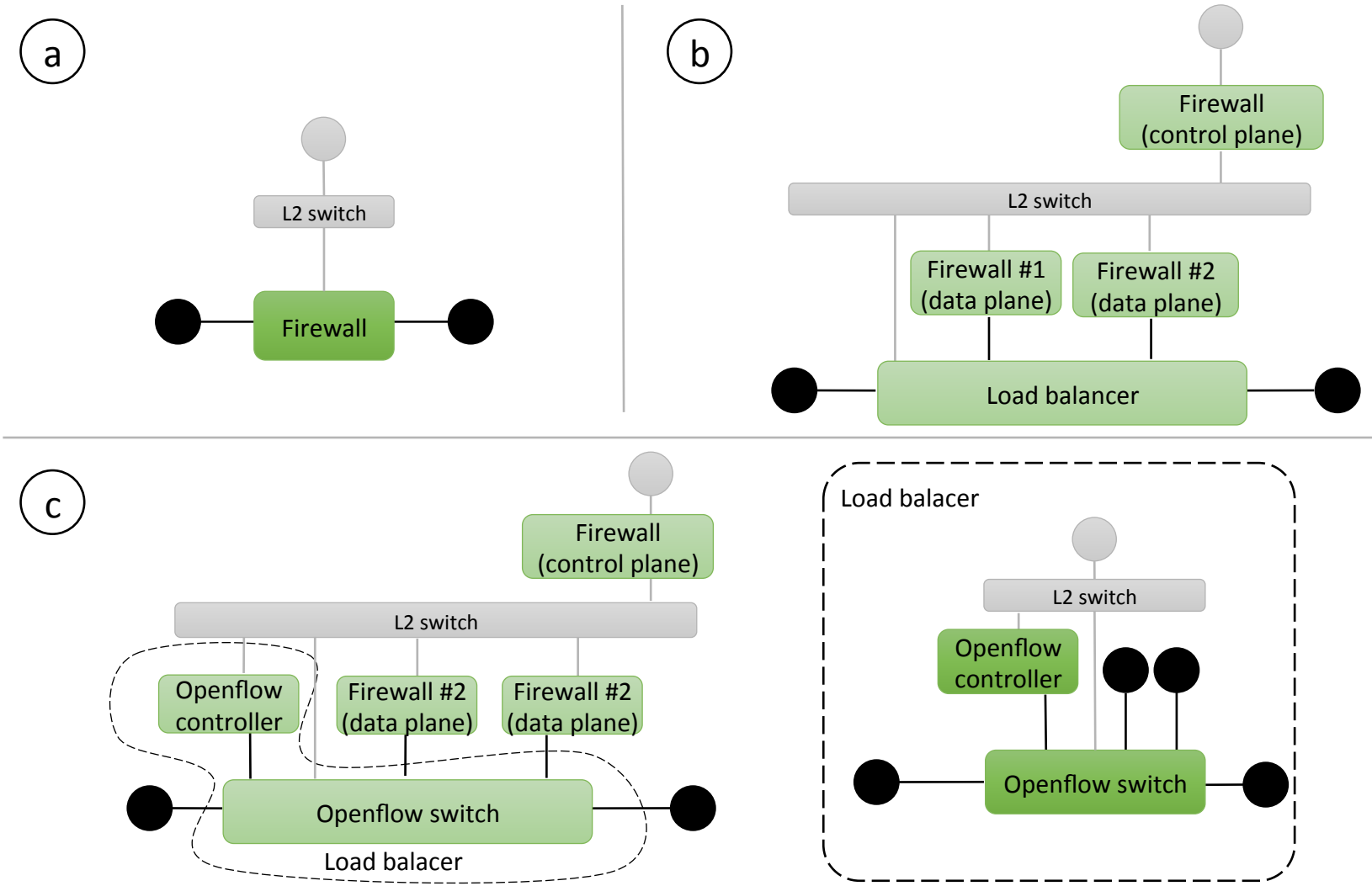
c



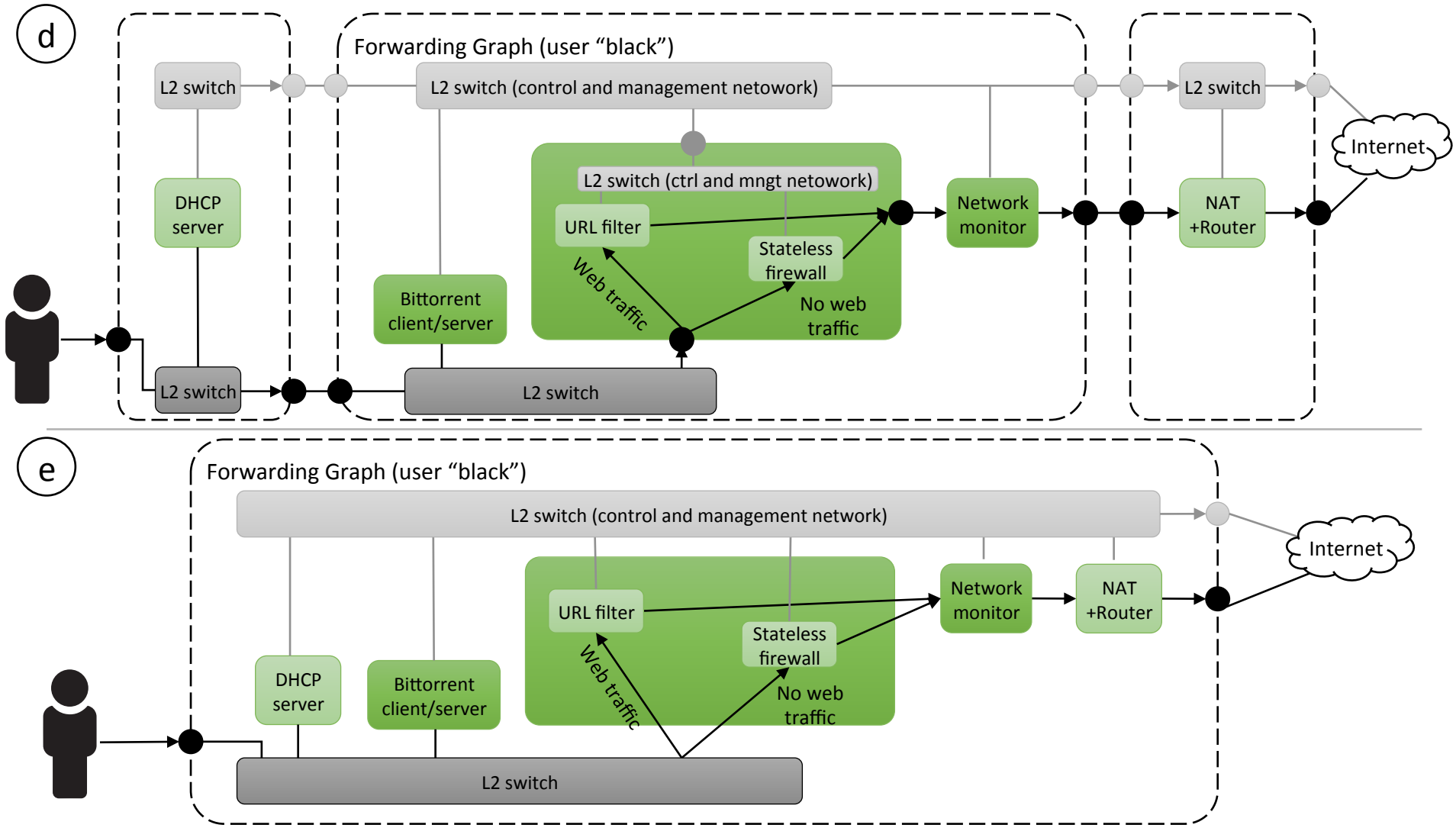
Forwarding graph – lowering process



Graph expansion

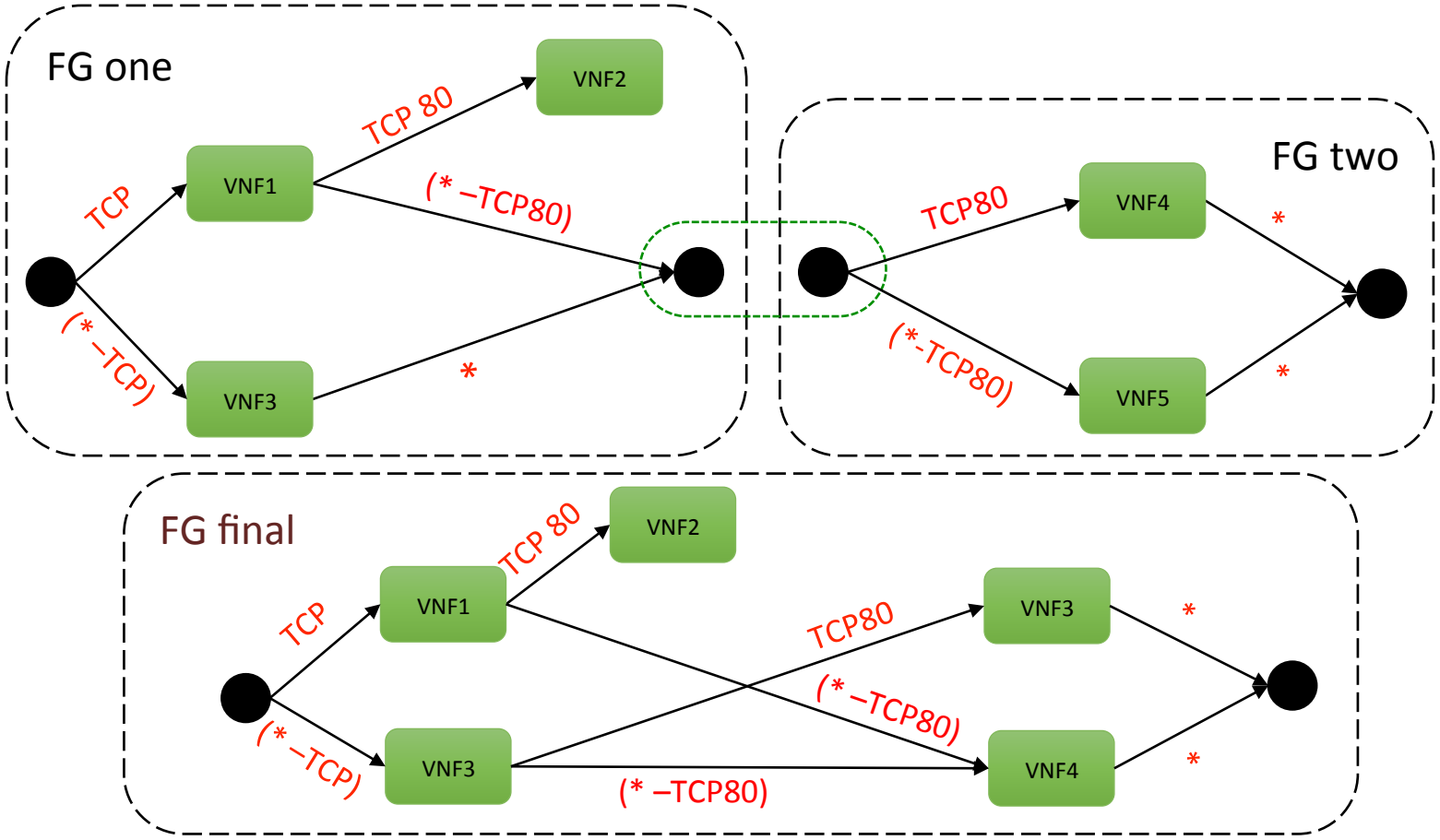


Forwarding graph – lowering process



Graphs connection

- The graphs connection requires **Cartesian product** between the rules on ports involved in the connection.

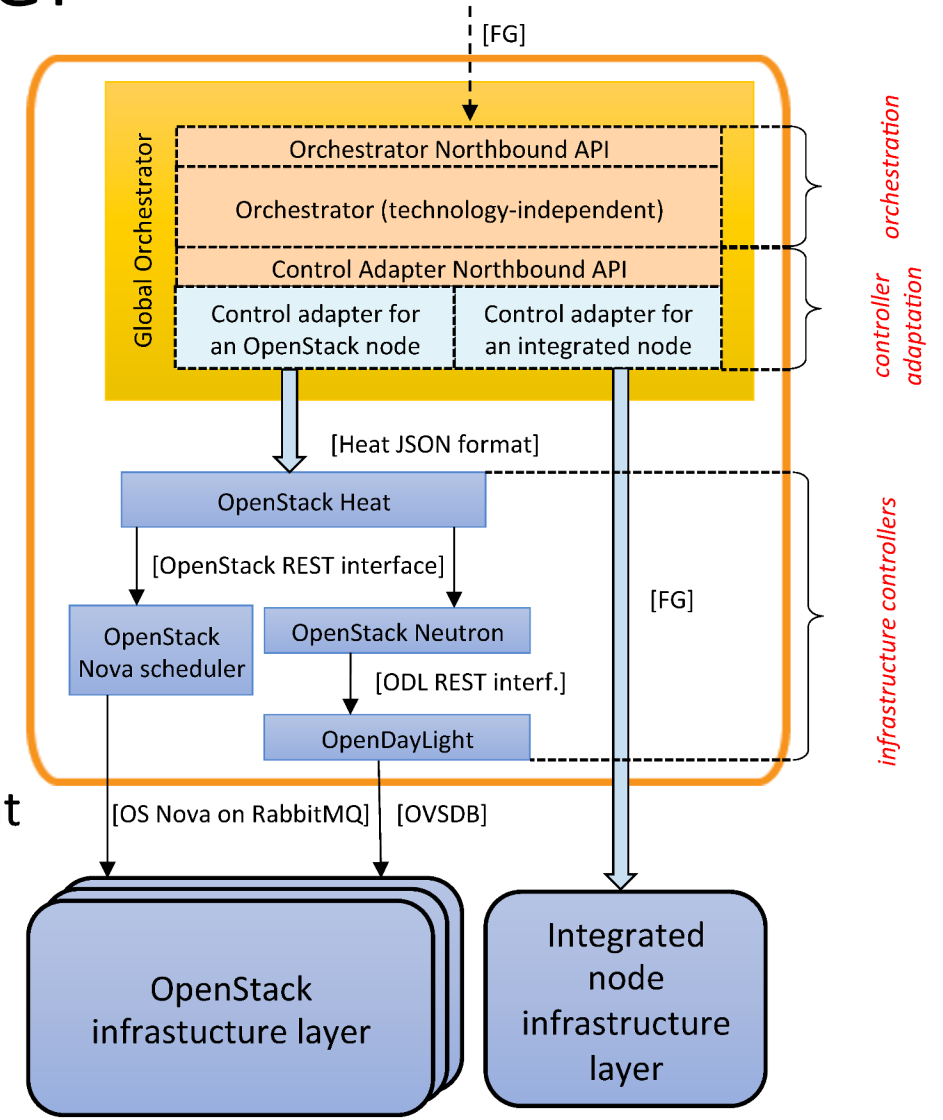


Graphs connection

Orchestration layer

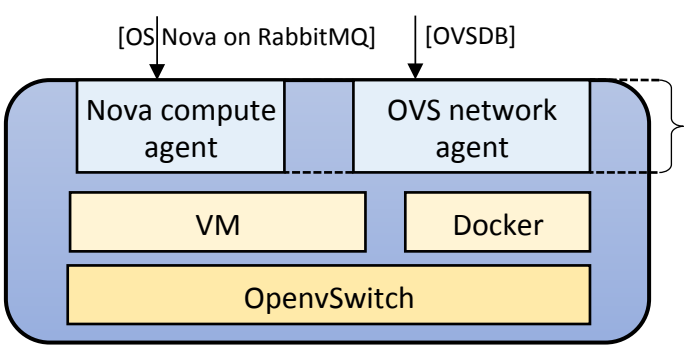
- The orchestration layer functions:
  - It manipulates the forwarding graph in order to allow its deployment on the infrastructure
  - It implements the scheduler that is in charge of deciding where to instantiate the requested service.

- The orchestration layer structure:
  - it is composed of three different logical sub-layers
    - *orchestration*
    - *controller adaptation*
    - *infrastructure controllers*



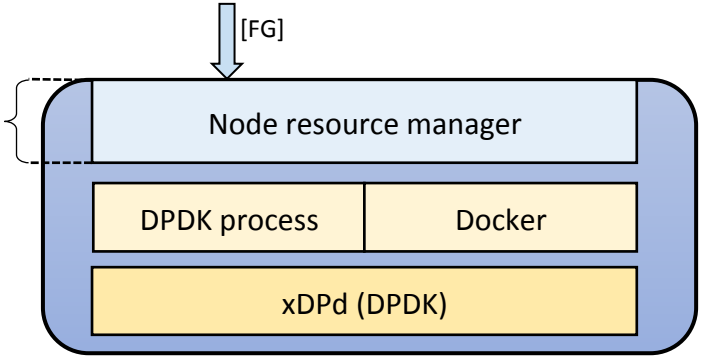
Infrastructure layer

OpenStack-based node



Local resource manager (network + compute + storage)

Integrated node

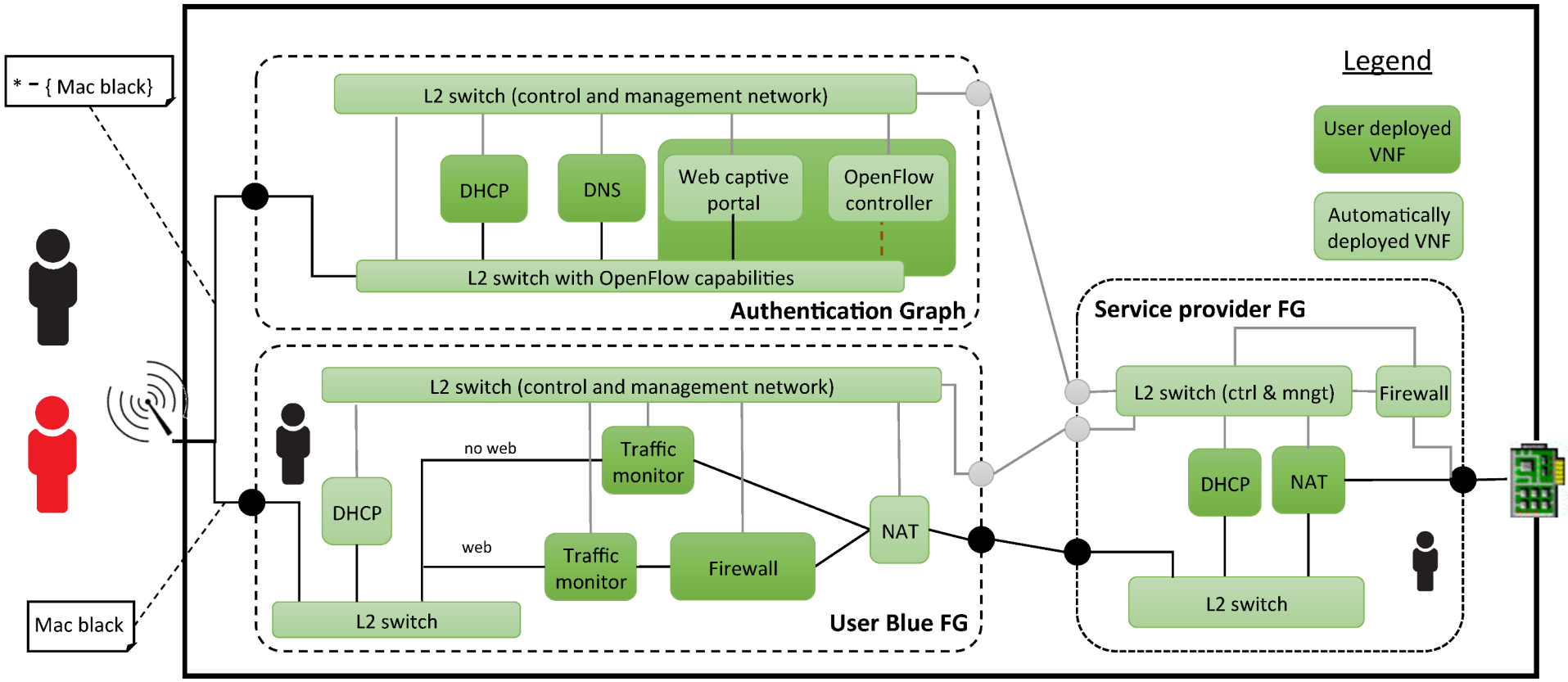


- Compatible with existing cloud enviroment
- Supports a lot of hypervisor (e.g. XEN, KVM, QEMU) and container (e.g. LXC, and partially support for Docker container)
- Lack of some essentially feature, as support to traffing steering that has been implemented as core of another thesis
- To allow the complete control of the FG the legacy network service

- Custom solution, no compatibility in cloud enviroment
- Supports Docker container and DPDK process
- Suitable to be implemented in nodes with limited resources

Infrastructure node – Memory usage

Prototype validation



Performance evaluation

Conclusions and future works

Questions