UNIVERSITÄT
BERN

2015 Autumn Semester Seminar

ICN in the Cloud

André Gomes
Universität Bern
2015-11-16

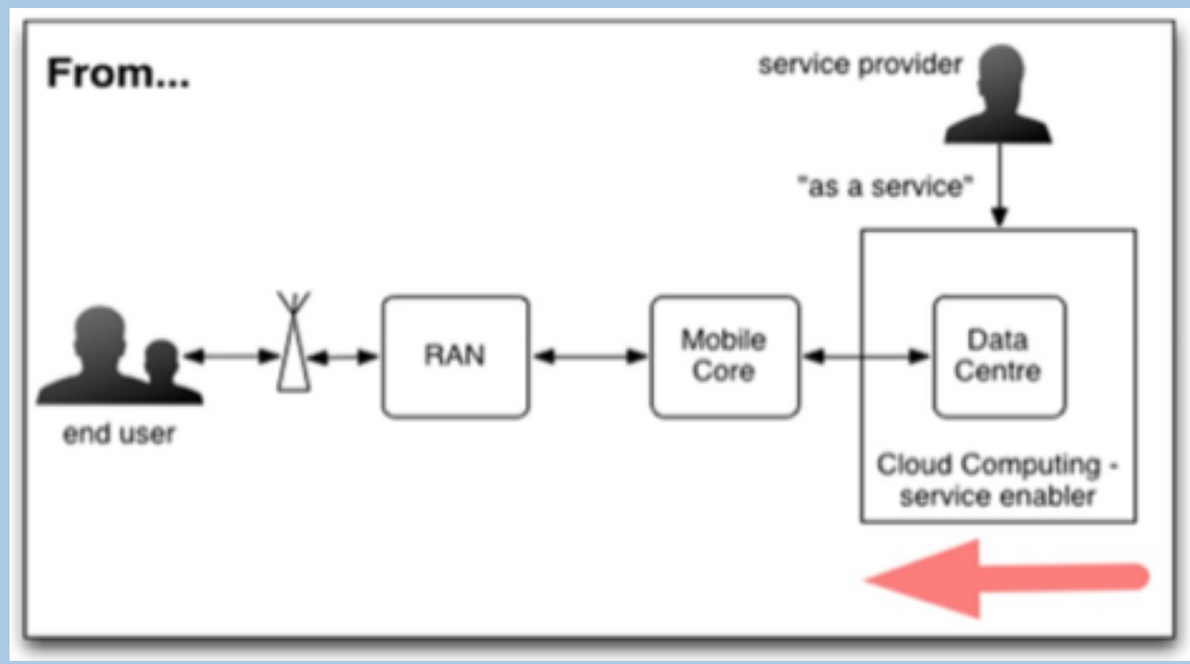
Outline

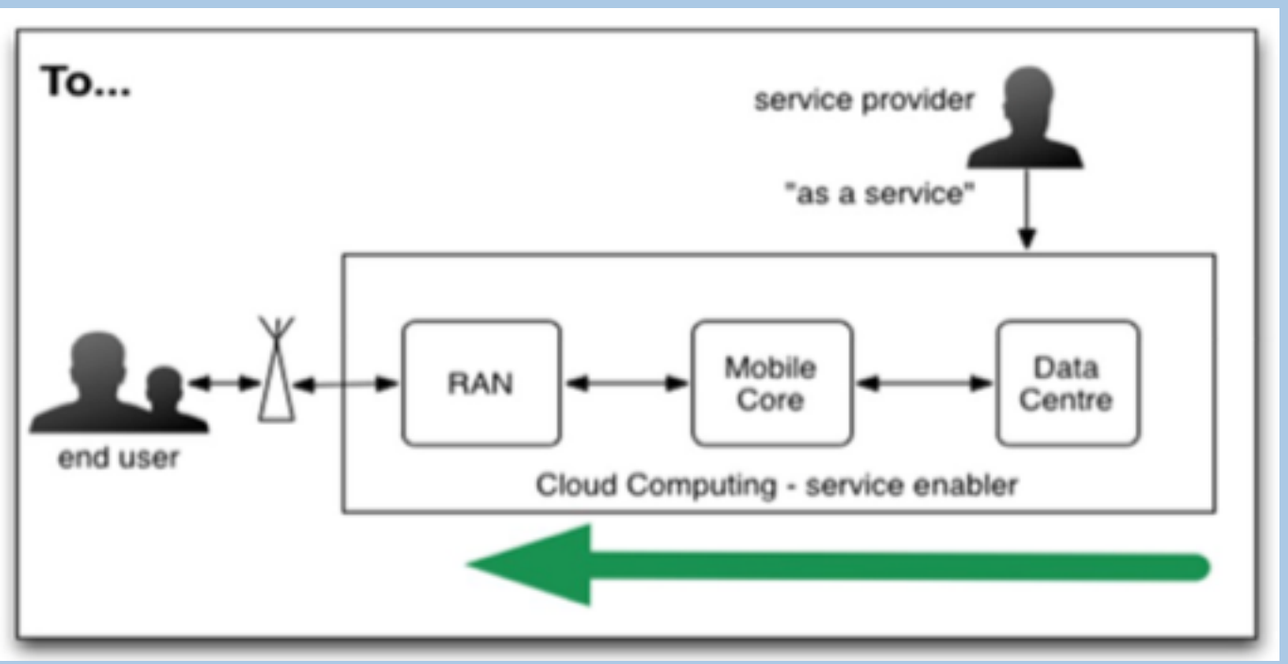
- Introduction and Motivation
- Requirements
- Architecture and Design
- Evaluation and Improvements
- Conclusions

Introduction and Motivation

UNIVERSITÄT BERN

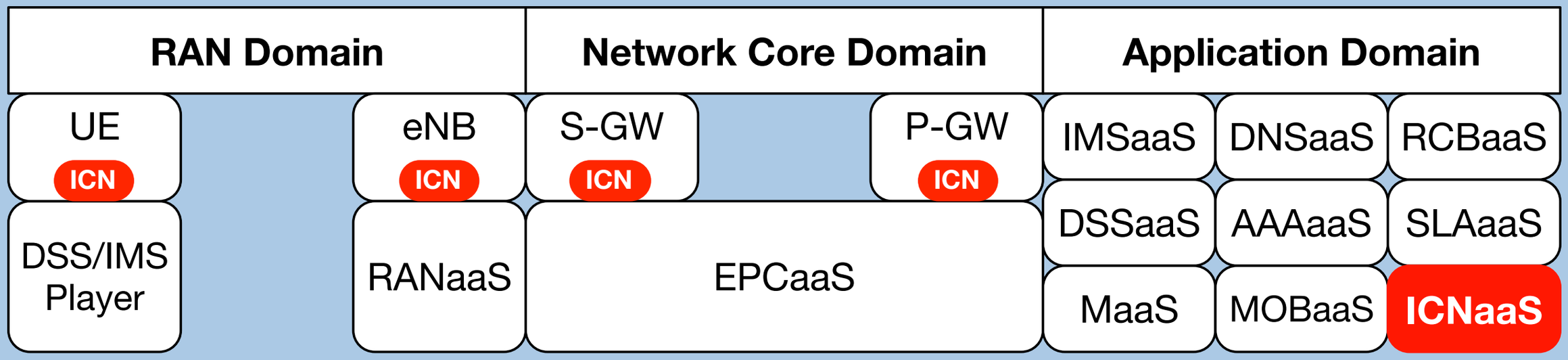
- Top-most motivations of Mobile Cloud Networking:
  - ➡ Extend the concept of Cloud Computing beyond data centres towards the mobile end-user.
  - ➡ Deliver and exploit the concept of an End-to-End Mobile Cloud for novel applications.





Introduction and Motivation

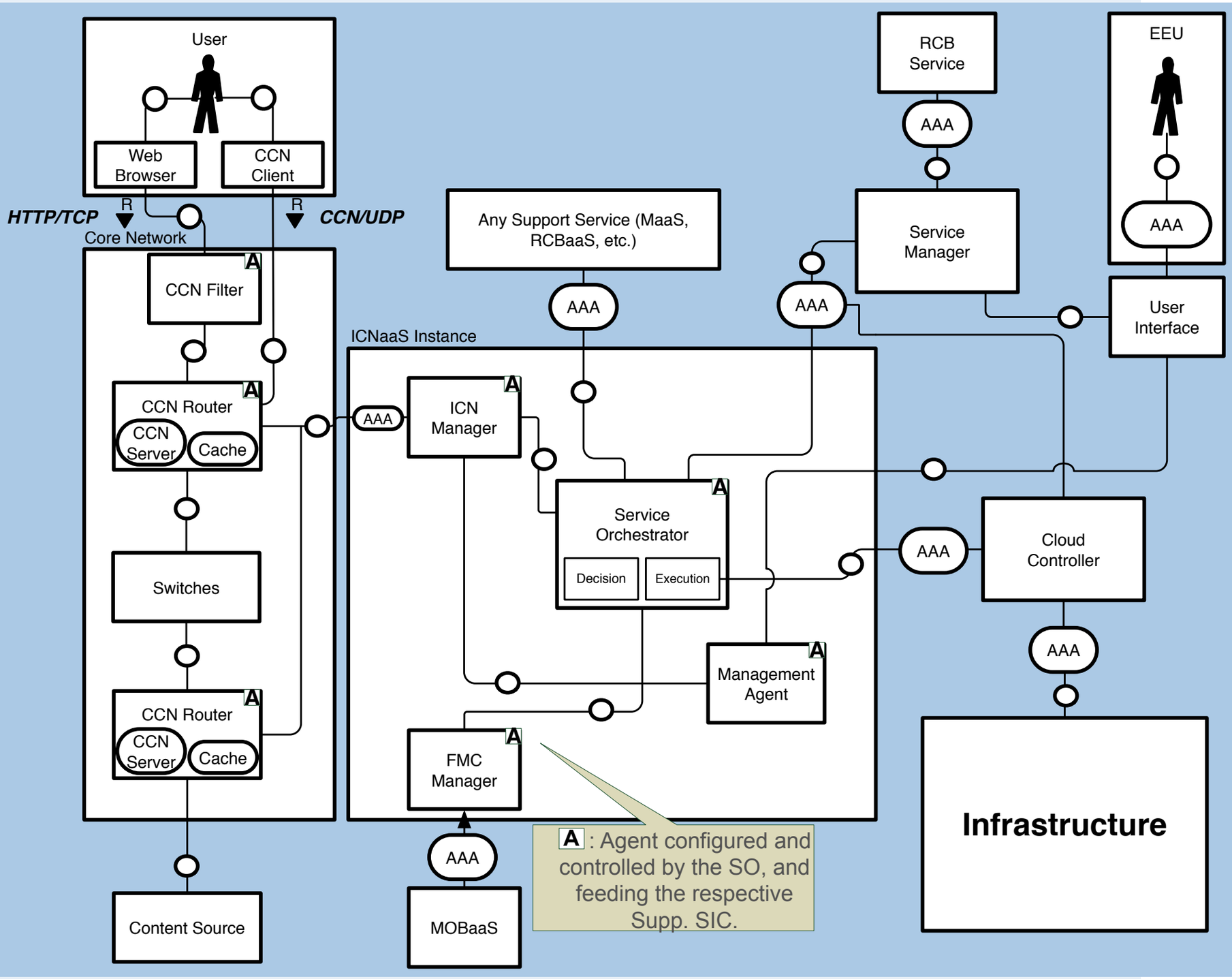
> Where does ICN fit?



Requirements

- Cloud Principles
  - ➡ Agility
  - ➡ On-demand instantiation
  - ➡ Multi-tenancy
  - ➡ Pay-As-You-Go
  - ➡ Elasticity
  - ➡ Reliability
  - ➡ Performance
- Specific Requirements
  - ➡ Integrated with network, delivering content at the edge.
  - ➡ Leverages multiple radio technologies.
  - ➡ Accounts for very dynamic user mobility.

Architecture



Design

- Platform
  - ➡ Infrastructure
    - OpenStack (Infrastructure as a Service)
    - Includes multiple modules, e.g. Nova, Neutron, Heat
    - Typically using Kernel Virtual Machine (KVM) as hypervisor
  - ➡ Cloud Controller
    - Abstracts interfaces to OpenStack modules
    - Manages stacks, coordinates inter-service interfaces and supports external modules
  - ➡ Service Manager
    - Provides a way of describing services for users, allowing them to select/configure the desired service.
    - Manages instances of services by deploying each of their Service Orchestrators and starting/ending lifecycles.

Design

UNIVERSITÄT BERN

> Service Instance Components (SICs)

➡ Service Orchestrator

- Manages the entire instance lifecycle via OCCI interfaces to Service Manager and Cloud Controller.
- A decision module gathers processed metrics from the monitoring service (MaaS) and dimensions the service accordingly.

➡ ICN Manager

- Based on information received from the network topology, decides about placement of CCN routers.
- Using a REST API, allows the full control of the ICN topology. Namely: endpoints, prefixes management, automatic routes setting and load balancing policies.

Design

- ➡ CCN Routers
  - Run CCNx 0.8.2.
  - Modified code to include monitoring, Follow-Me Cloud (FMC) and legacy compatibility (HTTP proxying).
  - CCN Server to receive external commands (REST API) and provide monitoring information to FMC Manager.
  - Zabbix Agent to push gathered metrics to the monitoring service (MaaS).
- ➡ FMC Manager
  - Decide if content migration should occur, where to and what content should be transferred.
  - Inputs from mobility prediction (MOBaaS) and from metrics gathered at the CCN Routers.
- ➡ Management Agent
  - Provides a direct interface to the API of the ICN Manager to control the SICs.

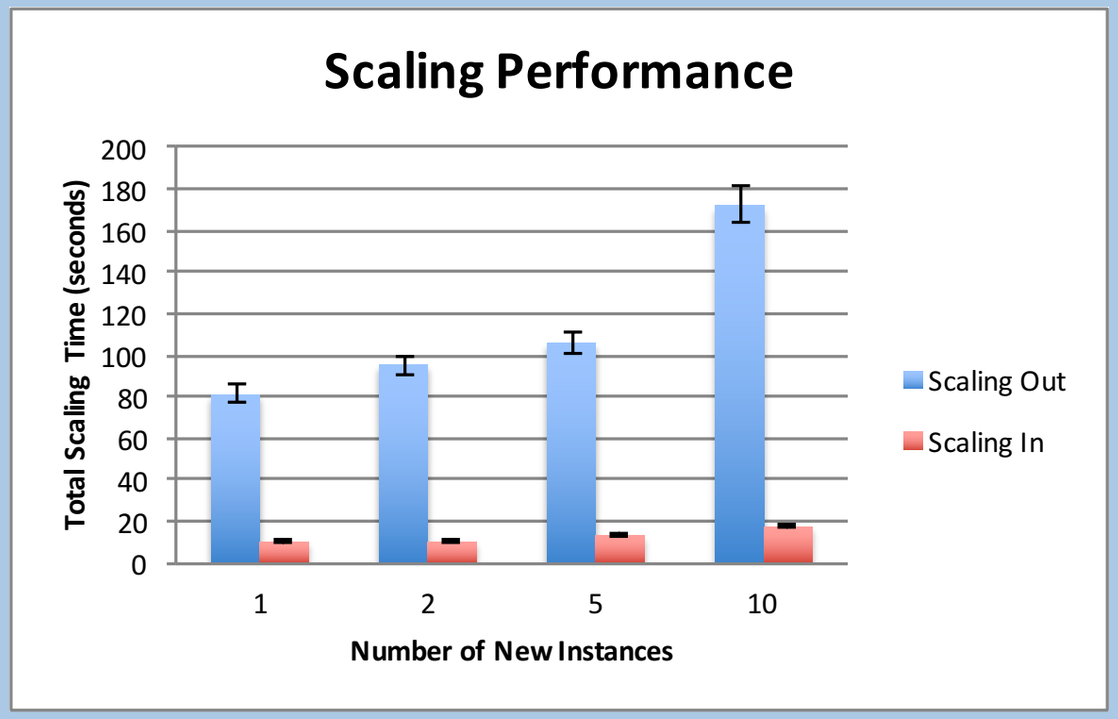
Standalone Evaluation

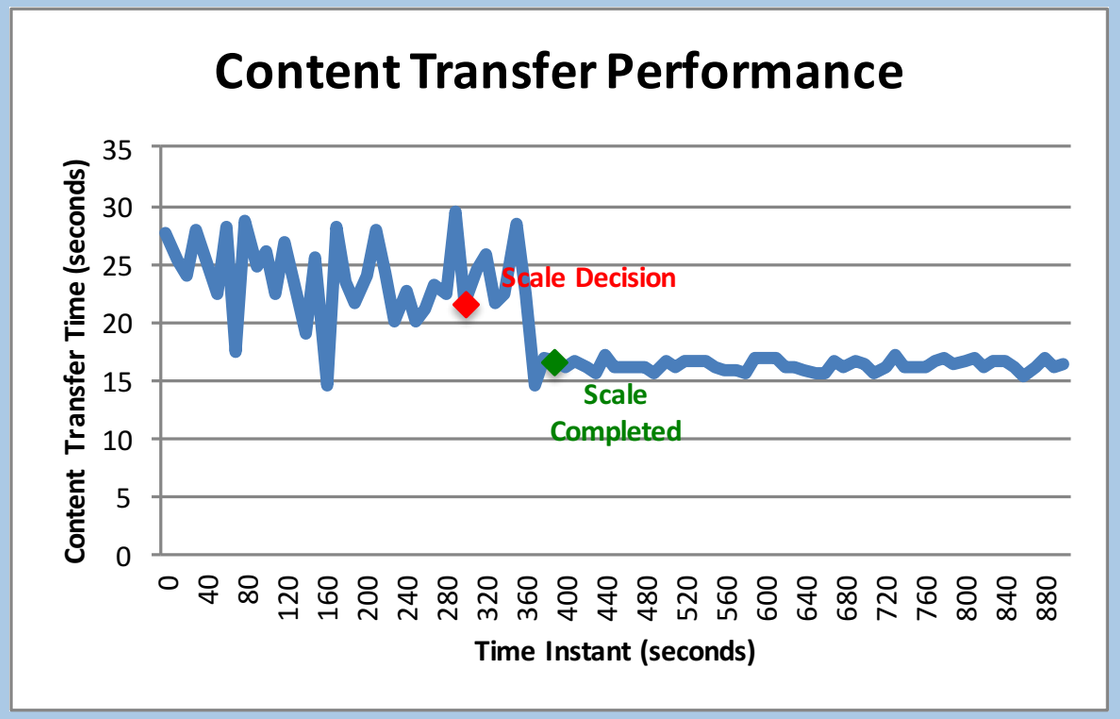
UNIVERSITÄT BERN

- Two types of evaluation:
  - ➡ Functional:
    - All the SICs are correctly deployed, integrate well between each other and ensure content is delivered when requested.
    - All the APIs respond as supposed and no exceptions occur.
  - ➡ Non-Functional:
    - Service lifecycle operates under reasonable timings.
    - Clear benefits can be obtained from newly developed concepts (e.g. Follow-Me Cloud), edge caching, etc.
    - Service scaling keeps content access latencies low.

Standalone Evaluation

Avg. Deployment Time	Avg. Disposal Time	Avg. Scaling Decision Time
207.817 seconds	10.485 seconds	305 milliseconds





Issues and Improvements

> How realistic was the evaluation?

➡ Measurements were obtained with a static Service Orchestrator (SO). What happens when it will be deployed along the instance?

- With OpenShiftv2 to run SOs, initial deployment times rise by up to 5 minutes.
- OpenShiftv3 to the rescue, but it means SOs have to be adapted and now be based on a Docker container pulled from Docker Hub. SO deployment time: up to 30 seconds.

Issues and Improvements

➡ Could images be shrunk to improve deployment times?

- Docker container images: yes. Reduce the number of layers and use a small base distro. New SO deployment time: less than 5 seconds.
- Other images: not much of a difference, but deployment + provisioning phases could be optimized at the SO side.

➡ Automated and more accurate way to collect metrics?

- Yes. Graylog to log events inside Service Managers and Orchestrators. Zabbix (MaaS) to fetch service specific metrics and correlate.

Issues and Improvements

➡ How to scale?

- Metrics component by component, scale components individually.
- Metrics aggregated by layer, scale layers as a whole 1 at a time.
- Metrics aggregated by layer, scale layers as a whole calculating how many more components are needed.
- Huge differences in performance for the 3 methods.

End-to-End Evaluation

UNIVERSITÄT BERN

> Also two types of evaluation:

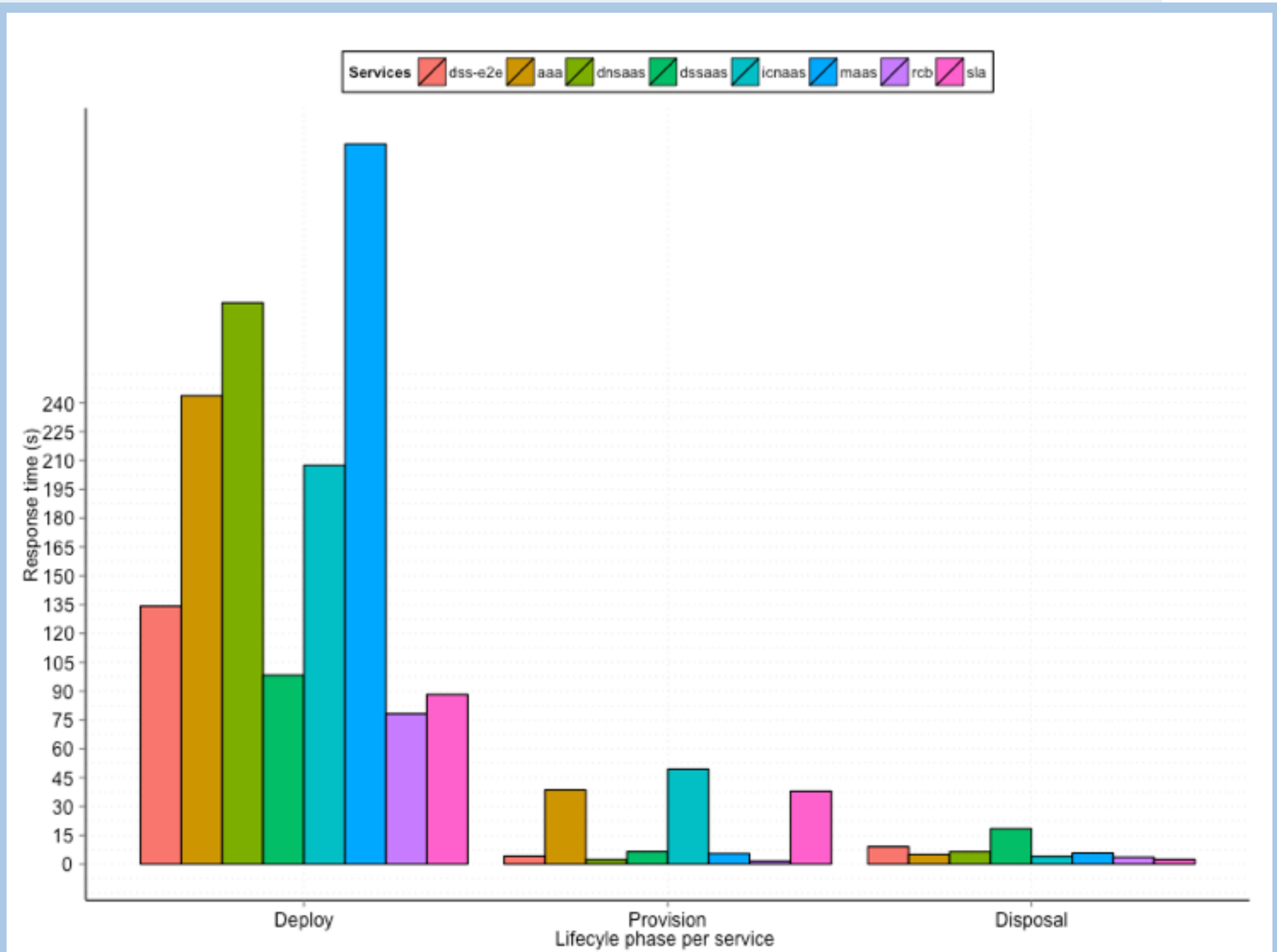
➡ Functional:

- Inter-service communication is working as expected. No exceptions/errors in the multiple APIs.
- After deployment and provisioning, service functionalities work well and leverage the usage of other services.

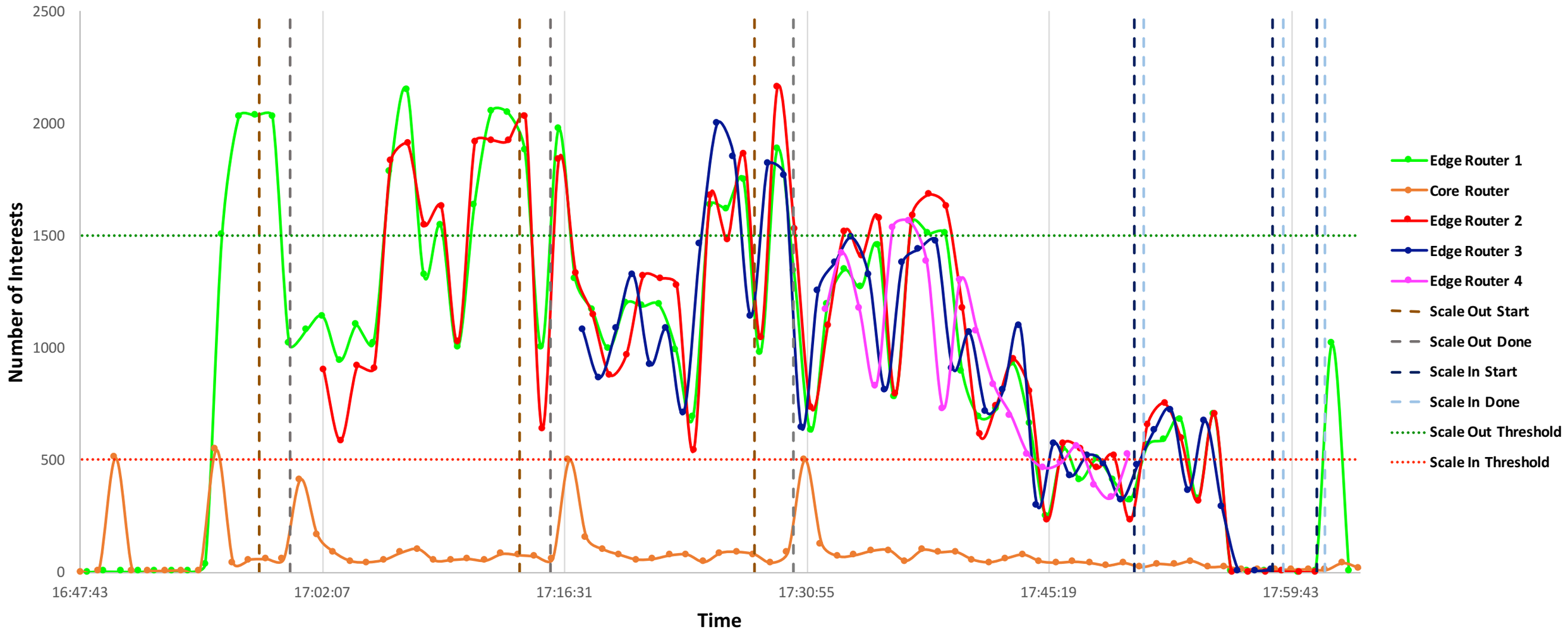
➡ Non-Functional:

- Services' lifecycle operate under reasonable timings.
- Performance improvements and other benefits can be gathered by leveraging integration of cloud services.
- Services scale according to load and are able to stay within the pre-defined thresholds.

End-to-End Evaluation



End-to-End Evaluation



Conclusions

- Mobile Cloud Networking brought an innovative and complete platform for cloud services.
- ICN was brought to the cloud and specifically developed and researched to provide numerous benefits to mobile networks and their users.
- Future directions already point to even higher granularity of data centers (fog computing) and more flexible SDN-based mobile networks.
- All the software is open-source and is readily available:

https://github.com/MobileCloudNetworking

http://git.io/v4Z5j

UNIVERSITÄT
BERN

Q&A - Discussion