

WP2: System Requirements and Specifications

ADISORN LERTSINSRUBTAVEE
UMOBILE PROJECT MEETING
CAMBRIDGE, 21-22 SEPTEMBER 2016

Deliverables	Task	Task Leader	Status	Due on	Issues
D2.1: End-user requirements report	2,1	UCAM	Delivered	M5: May 2015	-
D2.2: System and network requirement specifications (1)	2,2	COPELABS	Delivered	M14: March 2016	Update any requirements prior to the review meeting
D2.3: System and network requirement specifications (2)	2,2	COPELABS	draft version 0.2 - [on hold]	M28: May 2017	Start discussion in M26 (March 2017)
D2.4: System and Network Deployability Design	2,3	TEKEVER	No draft yet	M30: July 2017	No update from task leader

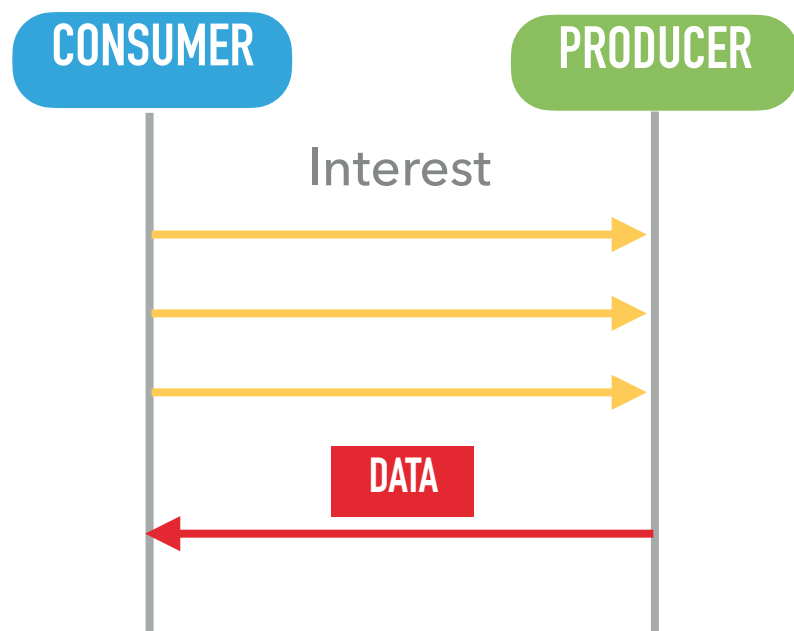
- ▶ Issues from Xanthi meeting
 - ▶ Update requirements
 - ▶ QoS requirements are still missing
 - ▶ Do the requirements fit to UMOBILE applications (i.e., Oi ,Persense)?
 - ▶ Should we revise D2.2 or update in D2.3?

WP3: Contributions from UCAM

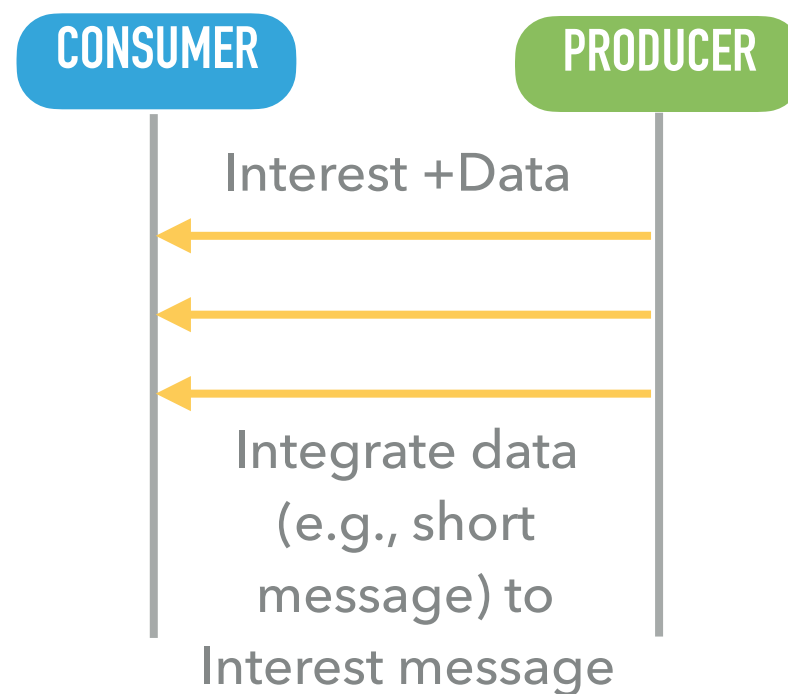
ADISORN LERTSINSRUBTAVEE
UMOBILE PROJECT MEETING
CAMBRIDGE, 21-22 SEPTEMBER 2016

- ▶ By default NDN is Pull based
- ▶ Push based communication - Producer takes control the content and intimately send Data to consumer
- ▶ UCAM proposed 3 approaches to implement the Push based communication models (D3.1)
- ▶ Implemented through PyNDN

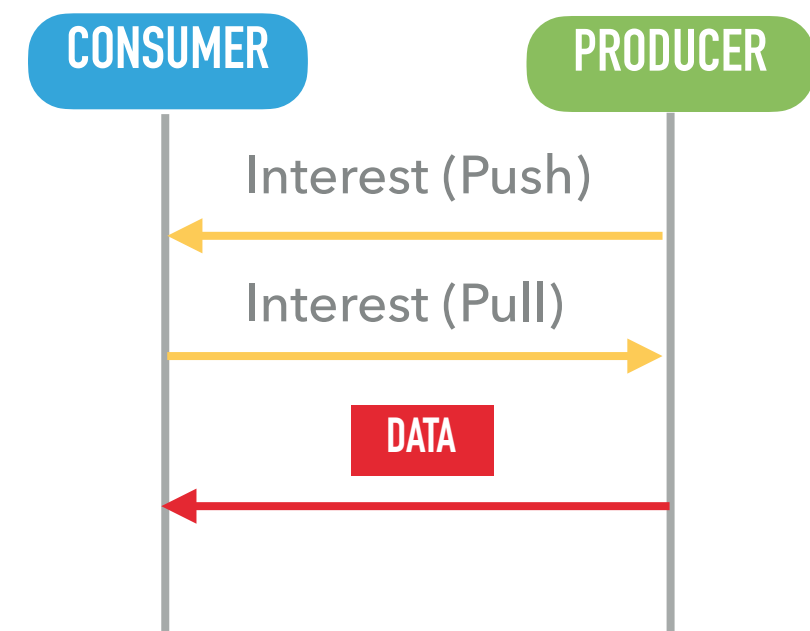
Interest polling



Interest notification

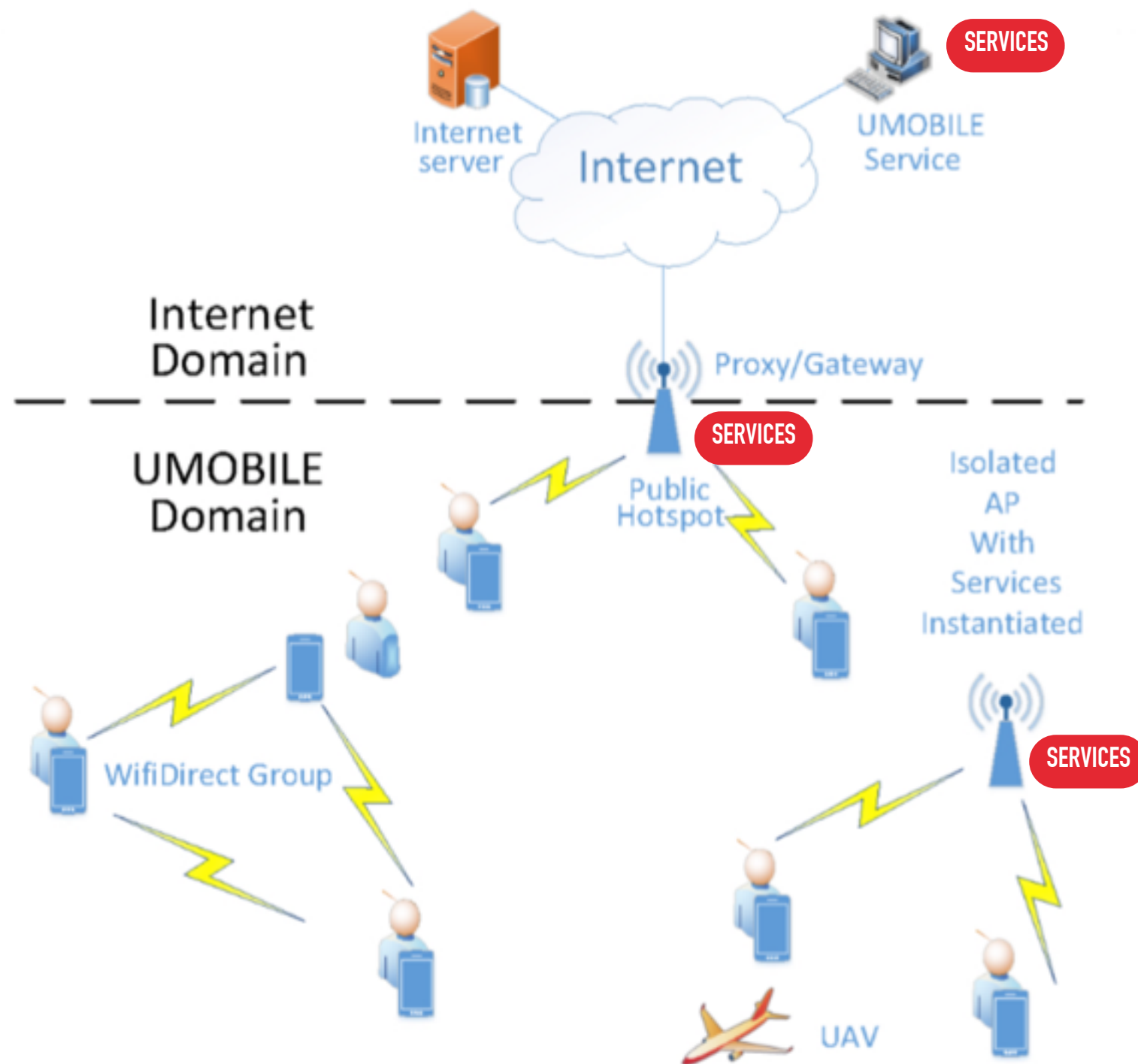


Publish data dissemination



Implemented and evaluated through the smart lighting system^[1]

Apply to service migration framework



- Migrate services to satisfy some QoS requirements (e.g., low latency, high availability). For example, the services are hosted close to the users

High level design of UMOBILE architecture (D3.1)

Decision Engine

- ▶ **Decide when and where to migrate/replicate the services**
 - ▶ Improve QoS (e.g., access latency, availability)
 - ▶ Minimise the cost of migration/replication (e.g., storage, migration traffic)
 - ▶ Provide different classes of QoS (D4.4)

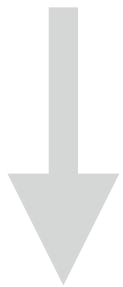
ICN-Based Data Dissemination

- ▶ **Service/Network monitoring**
 - ▶ Pull communication model
- ▶ **Service/Content Distribution (Migrate service to the edge)**
 - ▶ Benefit from in-network caching of NDN
 - ▶ Push communication model
- ▶ **Name based routing**
 - ▶ Decouple the location of producer and consumer
 - ▶ Multicast by name

Service Execution

- ▶ **Provide service at the edge**
 - ▶ Lightweight virtualisation (e.g., docker)
 - ▶ Improve the QoS (minimising the access latency)
 - ▶ Minimise traffic in the network

Decision Engine



ICN-Based
Data
Dissemination



Service
Execution

- ▶ Collaborated with AFA system to design the UMOBILE hotspot
- ▶ Microtik board (Access and Network function) + Raspberry Pi (Service Execution)
- ▶ Preliminary Benchmarking on Raspberry Pi
- ▶ Operating the lightweight services (e.g., Docker) on Raspberry Pi
- ▶ Understanding the scalability issues of lightweight OS virtualisation technology^[2]

Decision Engine



ICN-Based
Data
Dissemination

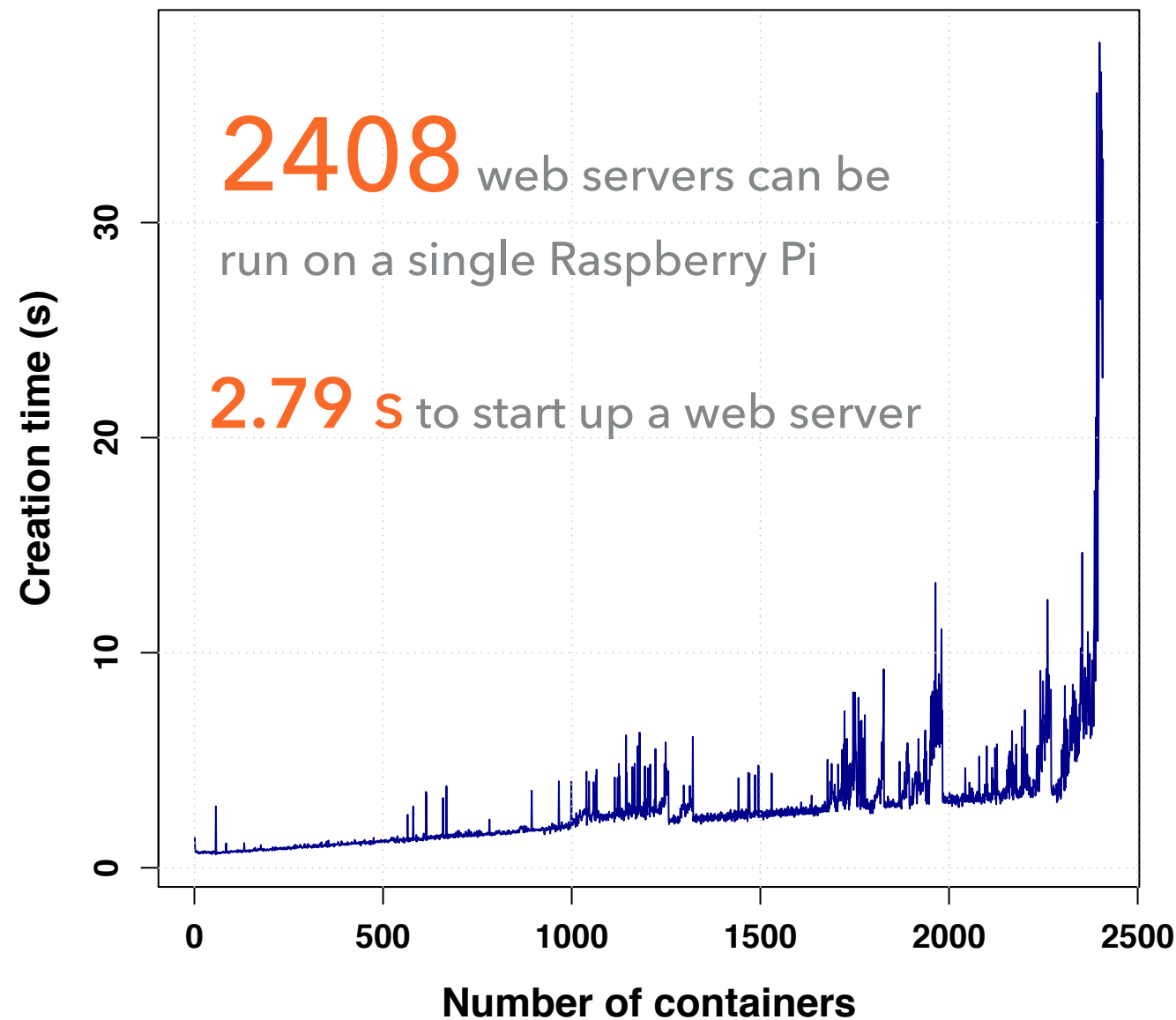


Service
Execution

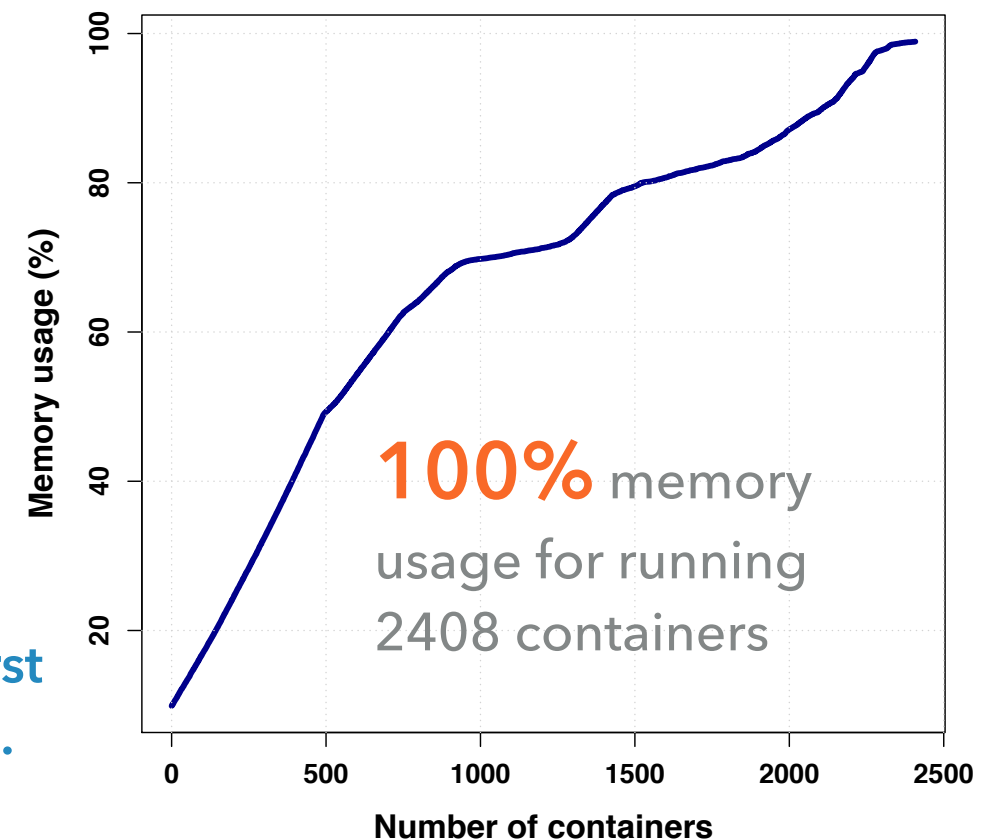
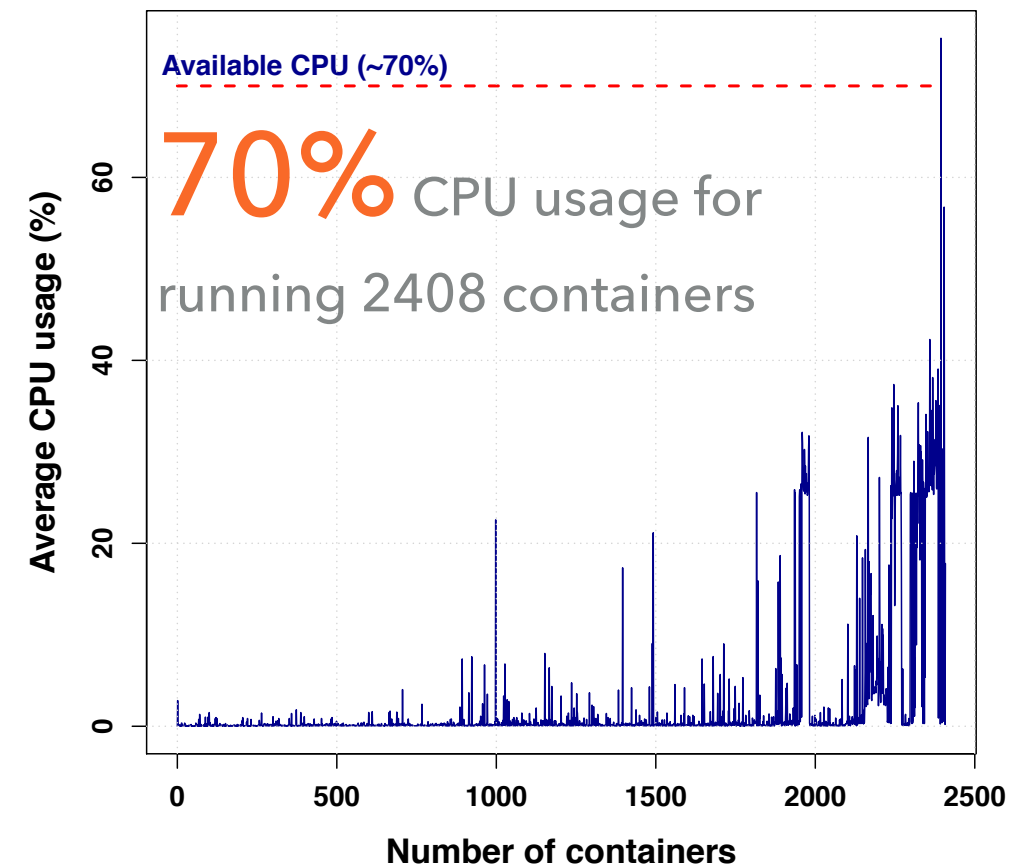
- ▶ Operating the lightweight services (e.g., Docker) on Raspberry Pi
- ▶ Benchmarking the service execution on Raspberry Pi
 - ▶ **Understanding the scalability issues** of lightweight OS virtualisation technology^[2]
 - ▶ **Identifying the critical constraints** of the system for deploying services
 - ▶ Exploring the **maximum limit** of the system
 - ▶ Maximum number of deployed containers
 - ▶ Maximum number of concurrent users

How many containers can be supported by a single PI ? 10

- ▶ Testing with simple web server image (html + a small jpg)
- ▶ Container size is about 90 KB



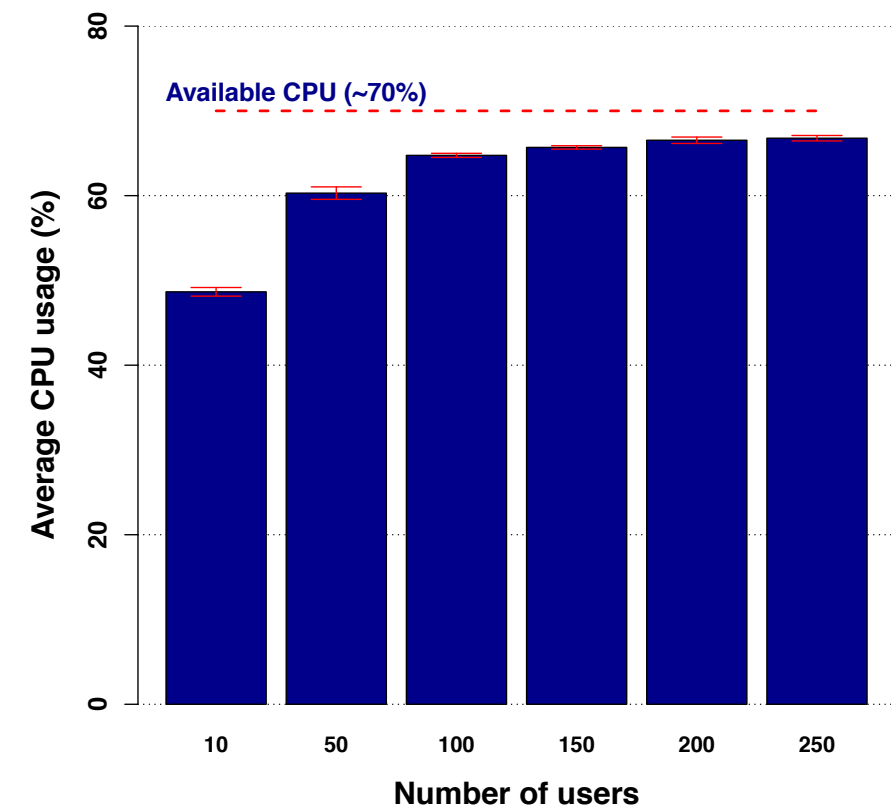
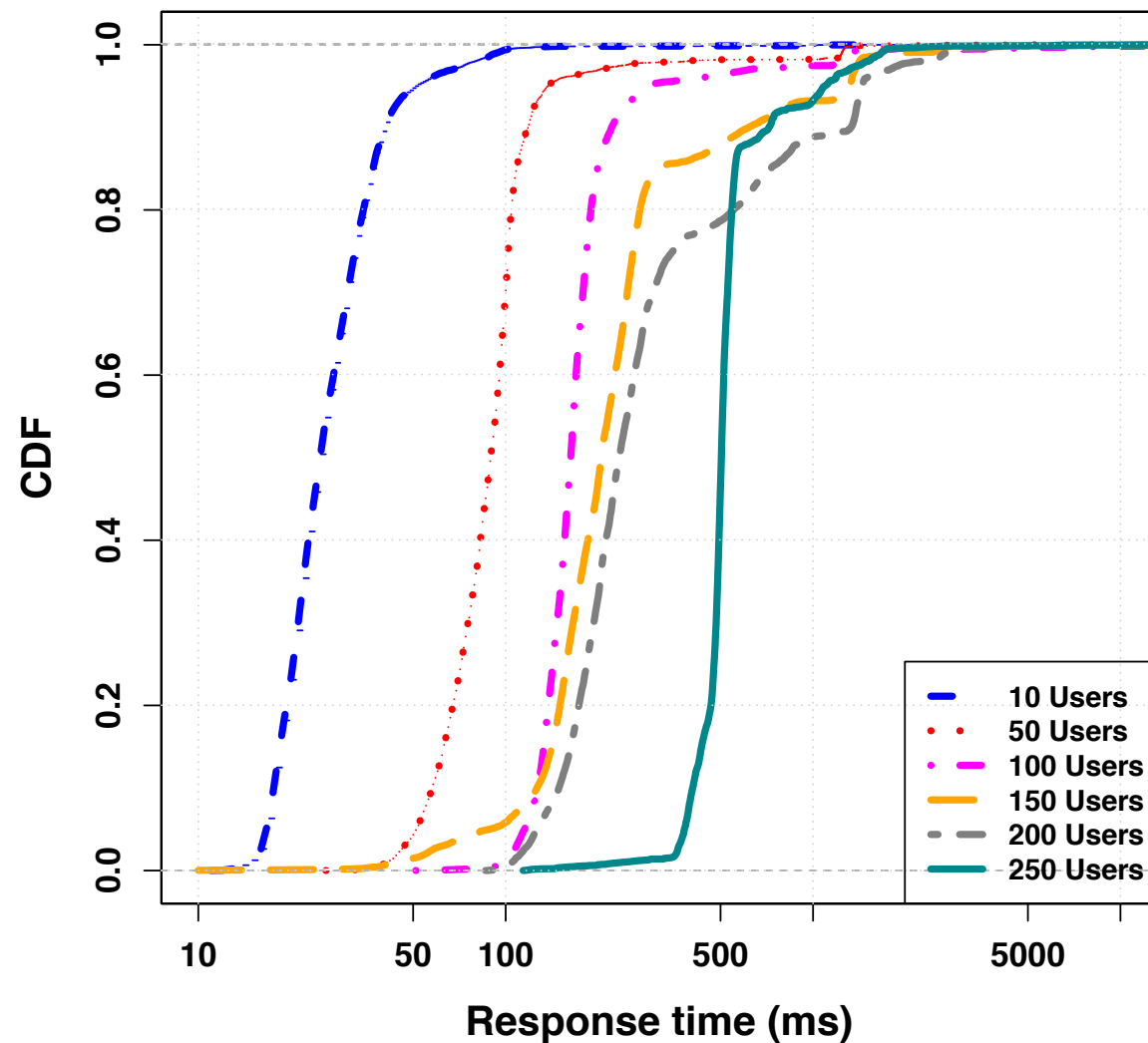
The initial memory usage before creating the first container was about 98 KB (a PI has 1GB RAM).



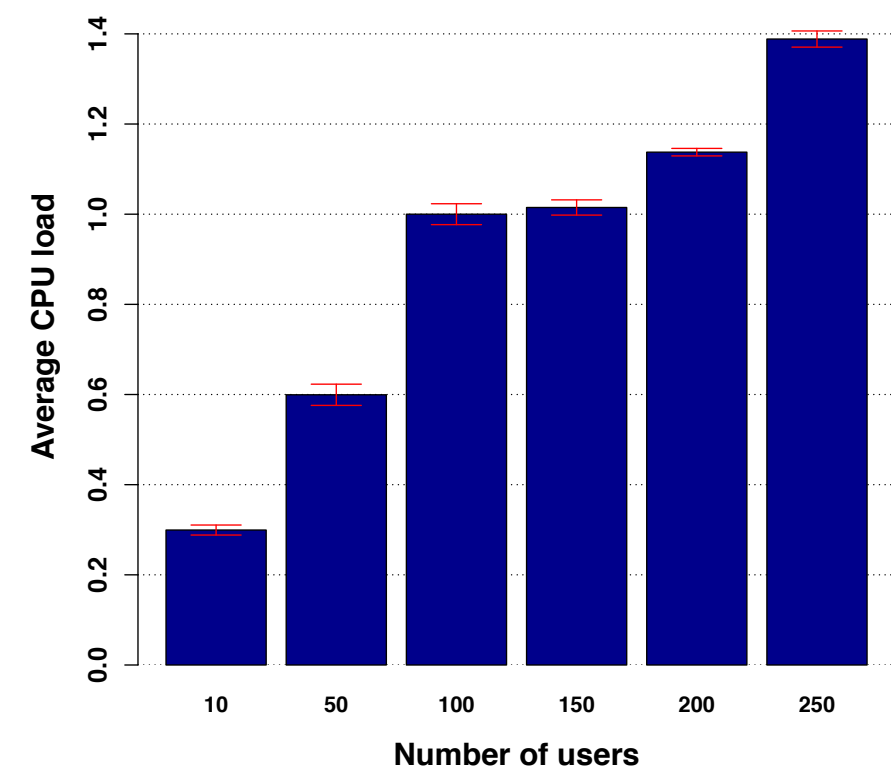
How many user requests can be supported by a single container?

11

- Scaling the number of concurrent users from 10 to 250
- 10,000 transactions were set per experiment



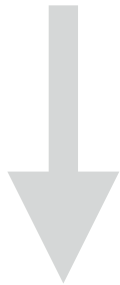
CPU load is increased up to 90%



- High response time when number of users is scaled up
- The amount of computational work that CPU needs to process (CPU load) is increased.

Status of Service Migration: ICN-based Data Dissemination

Decision Engine



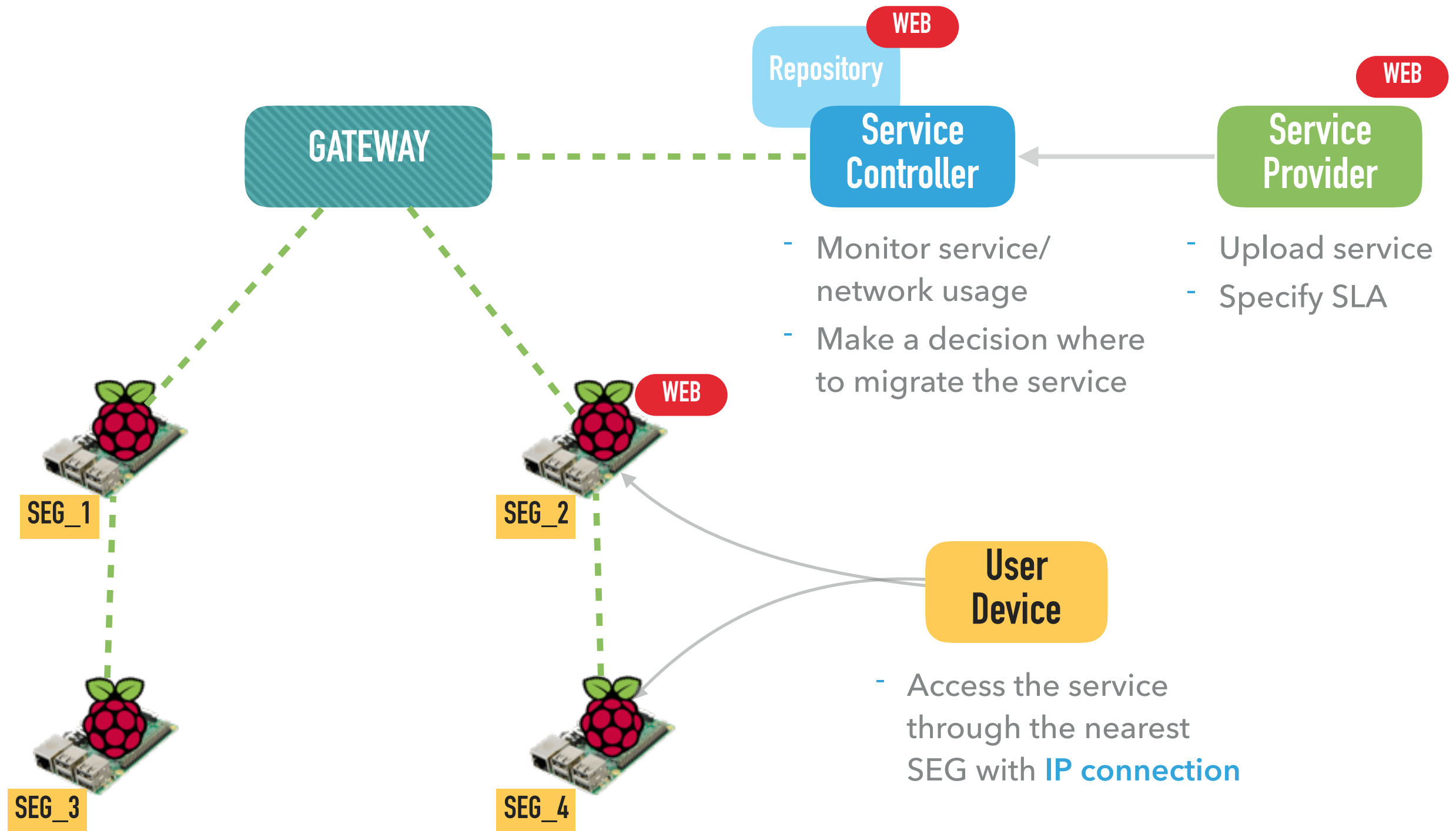
ICN-based
Data
Dissemination



Service
Execution

- ▶ Implemented the **service migration frame work**
- ▶ **Network/Service monitoring** using pull based communication
- ▶ Migrate and execute Docker container over NDN (**Docker and NDN integration**)
- ▶ Multicast communication through **named based routing**
- ▶ Optimising the traffic through **in-network caching**

An example of Service Migration scenario



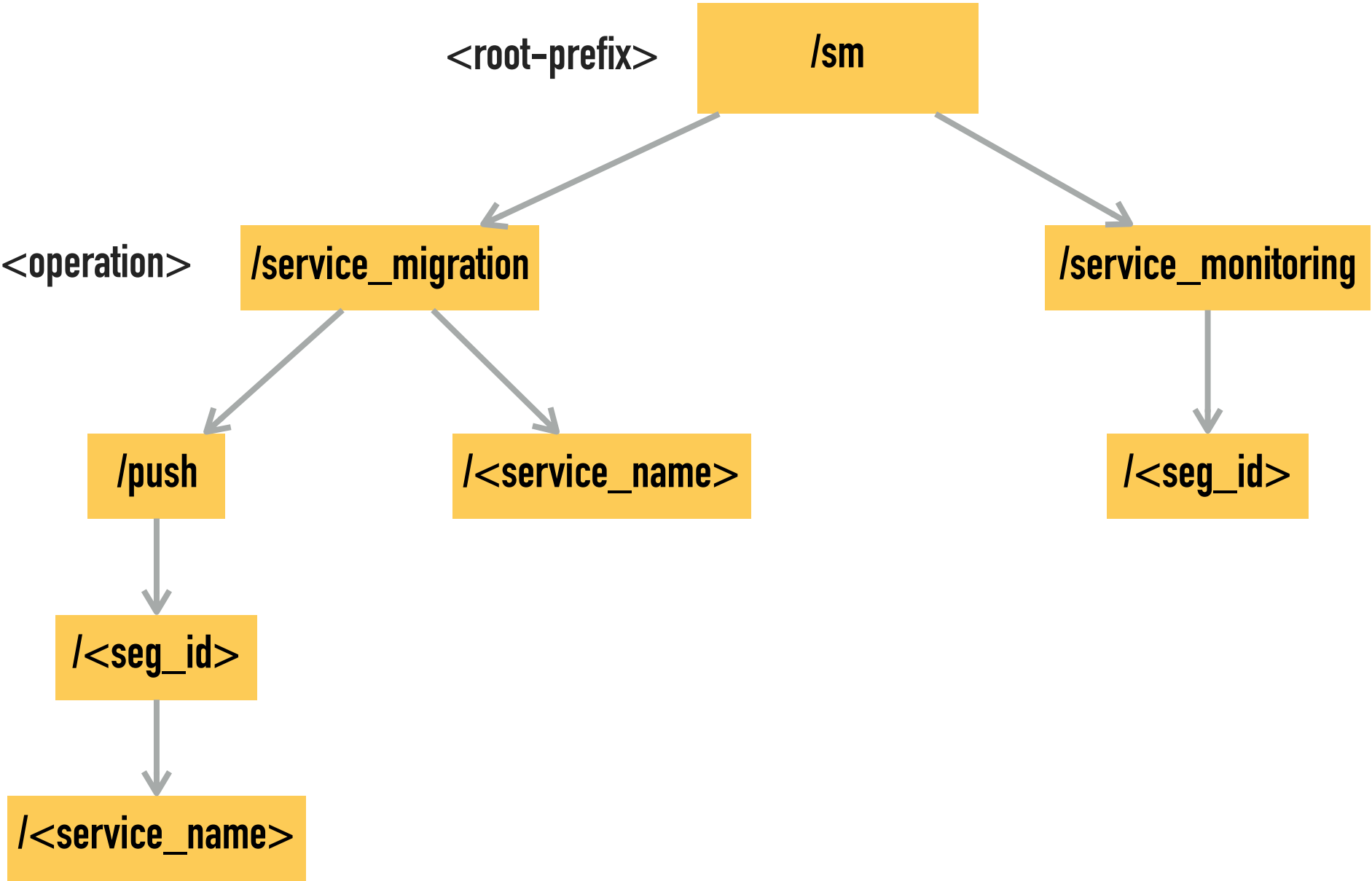
NDN connection
- - - - -

IP connection
—————→

SEG = Service Execution Gateway

Communication Model and Naming Scheme

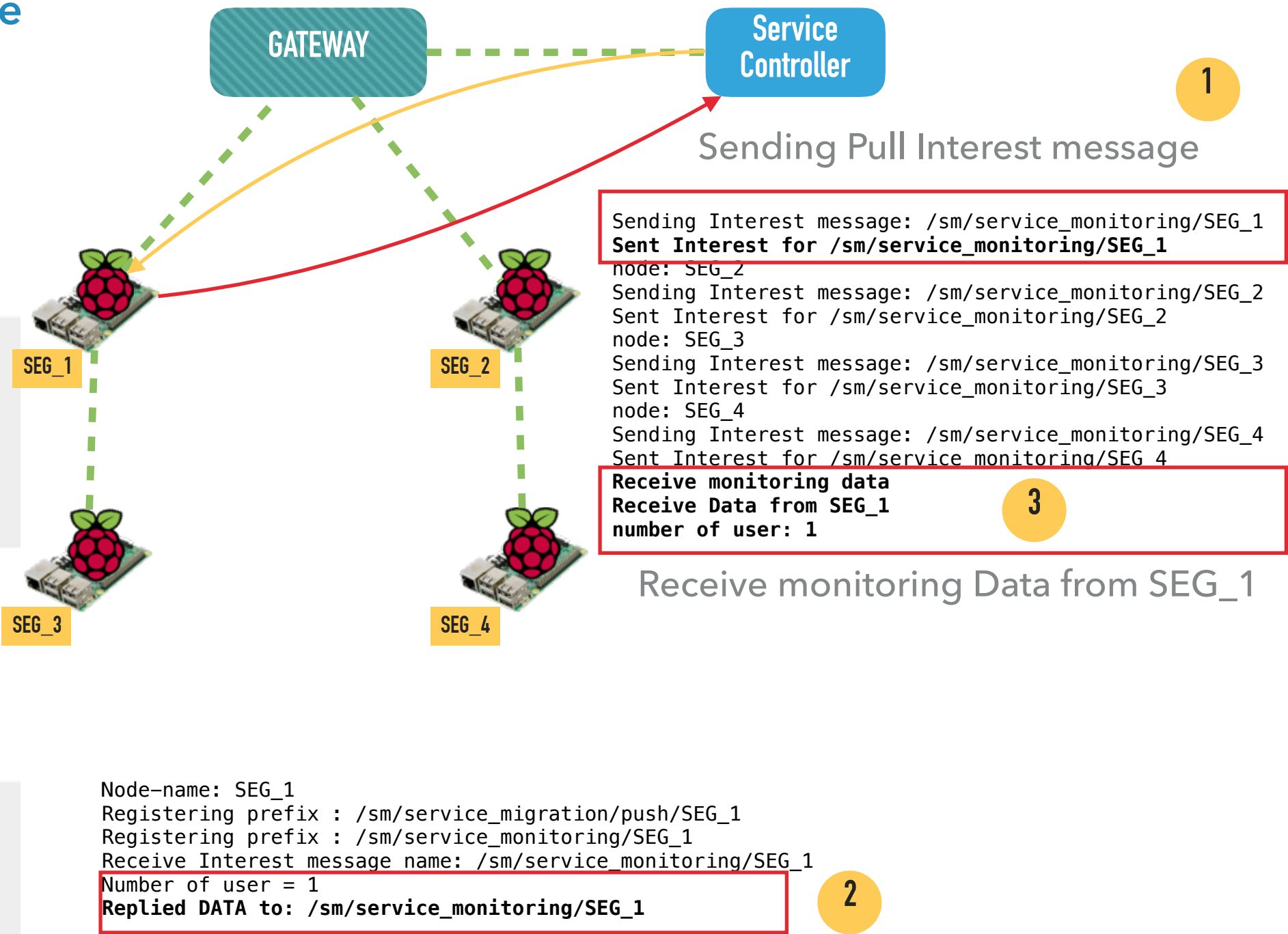
Operation	Model	Nature	Producer	Consumer
Monitoring	Pull based	Many to One	All SEGs	Service Controller
Migrating Services	Push and Pull based	One to Many	Service Controller	SEGs



Monitoring number of potential users

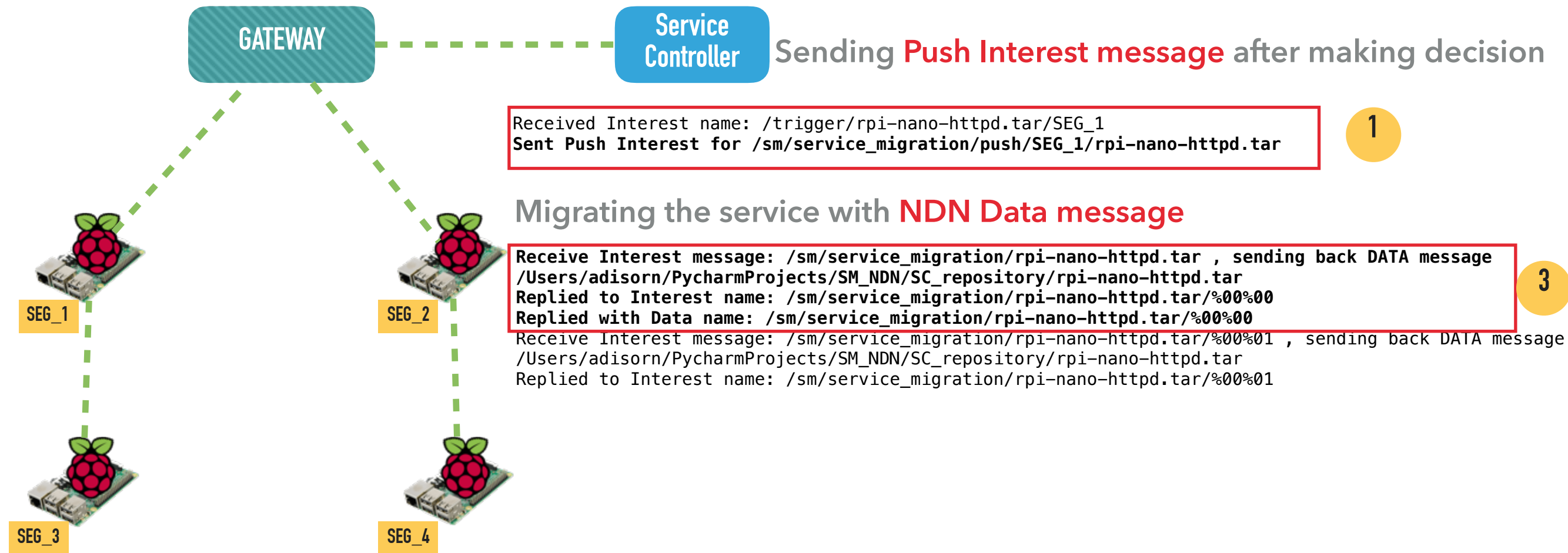
Service Controller's FIB table

/sm/service_monitoring/SEG_4	<table><tr><td>Faceld</td><td>267</td></tr><tr><td>Origin</td><td>255</td></tr><tr><td>Cost</td><td>0</td></tr><tr><td>ChildInherit</td><td>Y</td></tr><tr><td>RibCapture</td><td></td></tr><tr><td>Expires in</td><td>Never</td></tr></table>	Faceld	267	Origin	255	Cost	0	ChildInherit	Y	RibCapture		Expires in	Never
Faceld	267												
Origin	255												
Cost	0												
ChildInherit	Y												
RibCapture													
Expires in	Never												
/sm/service_monitoring/SEG_3	<table><tr><td>Faceld</td><td>264</td></tr><tr><td>Origin</td><td>255</td></tr><tr><td>Cost</td><td>0</td></tr><tr><td>ChildInherit</td><td>Y</td></tr><tr><td>RibCapture</td><td></td></tr><tr><td>Expires in</td><td>Never</td></tr></table>	Faceld	264	Origin	255	Cost	0	ChildInherit	Y	RibCapture		Expires in	Never
Faceld	264												
Origin	255												
Cost	0												
ChildInherit	Y												
RibCapture													
Expires in	Never												
/sm/service_monitoring/SEG_2	<table><tr><td>Faceld</td><td>267</td></tr><tr><td>Origin</td><td>255</td></tr><tr><td>Cost</td><td>0</td></tr><tr><td>ChildInherit</td><td>Y</td></tr><tr><td>RibCapture</td><td></td></tr><tr><td>Expires in</td><td>Never</td></tr></table>	Faceld	267	Origin	255	Cost	0	ChildInherit	Y	RibCapture		Expires in	Never
Faceld	267												
Origin	255												
Cost	0												
ChildInherit	Y												
RibCapture													
Expires in	Never												
/sm/service_monitoring/SEG_1	<table><tr><td>Faceld</td><td>264</td></tr><tr><td>Origin</td><td>255</td></tr><tr><td>Cost</td><td>0</td></tr><tr><td>ChildInherit</td><td>Y</td></tr><tr><td>RibCapture</td><td></td></tr><tr><td>Expires in</td><td>Never</td></tr></table>	Faceld	264	Origin	255	Cost	0	ChildInherit	Y	RibCapture		Expires in	Never
Faceld	264												
Origin	255												
Cost	0												
ChildInherit	Y												
RibCapture													
Expires in	Never												



Name based routing

Migrating the service

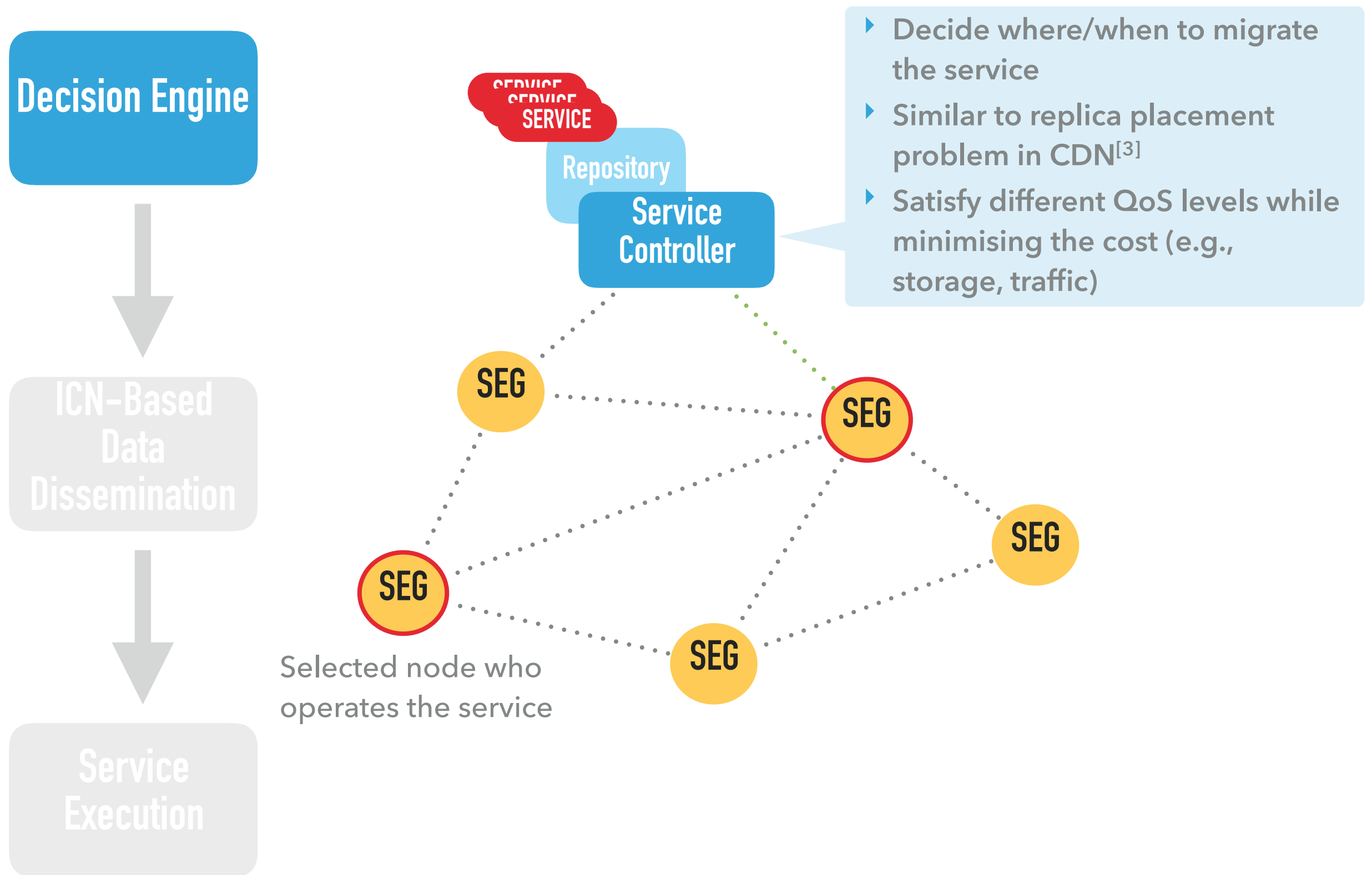


Fetching the service with NDN

2

Receive Interest message name: /sm/service_migration/push/SEG_1/rpi-nano-httpd.tar
Sending Interest message: /sm/service_migration/rpi-nano-httpd.tar
Sent Interest for /sm/service_migration/rpi-nano-httpd.tar
Received data name: /sm/service_migration/rpi-nano-httpd.tar/%00%00
dataSegmentNum0
lastSegmentNum23
Sent Interest for /sm/service_migration/rpi-nano-httpd.tar/%00%01
Received data name: /sm/service_migration/rpi-nano-httpd.tar/%00%01
dataSegmentNum1
lastSegmentNum23

Status of Service Migration: Decision Engine



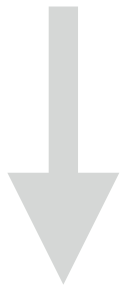
[3] Xueyan Tang and Jianliang Xu, "QoS-aware replica placement for content distribution," in IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 10, pp. 921-932, Oct. 2005.

Status of Service Migration: Decision Engine

Decision Engine



ICN-Based Data Dissemination



Service Execution

- ▶ Identify and measure **critical constraints** of the system
 - ▶ These parameters include **CPU load, Memory, number of users, storage** (from service execution benchmarking)
- ▶ Identify the **QoS requirements**
- ▶ Develop **heuristic algorithms** for decision engine
- ▶ This task is recently started in month 18 (D4.4)

Milestones and Achievements
