https://rainbow-h2020.eu/

@RainbowH2020

Horizon 2020

RAINBOW

Fogify: A Fog Computing Emulation Framework

Moysis Symeonides

msymeo03@cs.ucy.ac.cy

H-CLOUD community event 29/9/2021

Horizon 2020

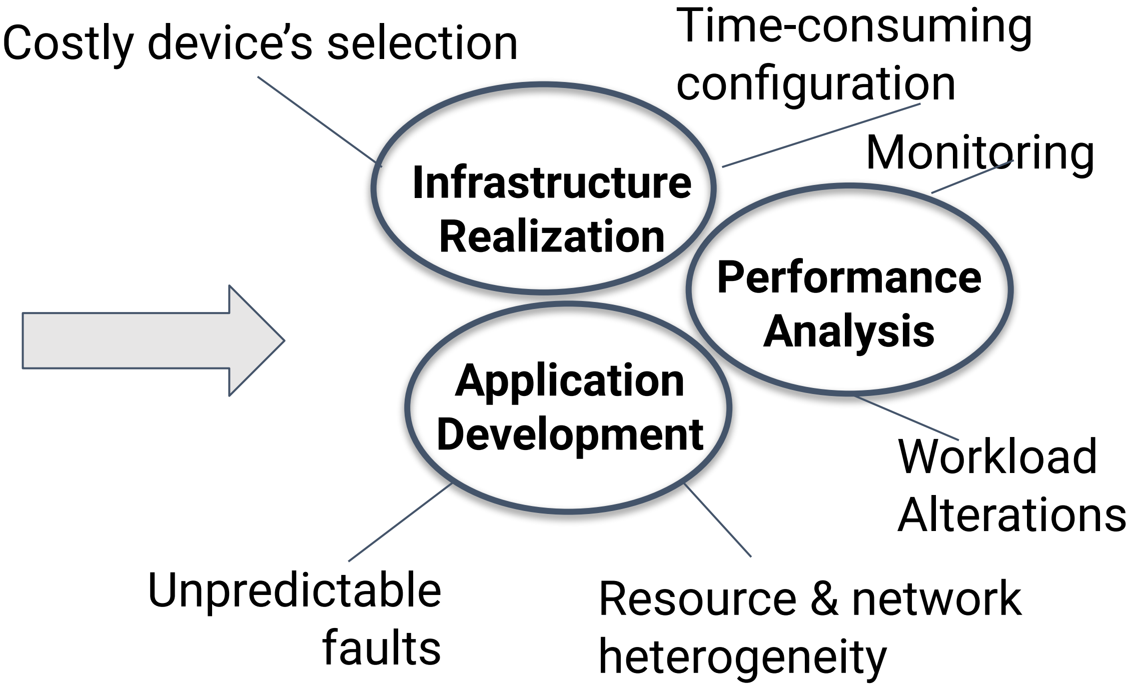
Fog Design and Deployment Challenges

Fog Application Use-cases

UC1. Human-Robot Collaboration in Industrial Ecosystems

UC2. Power Line Surveillance via Swarm of Drones

UC3. Digital Transformation of Urban Mobility



Infrastructure Realization

Performance Analysis

Application Development

Costly device's selection

Time-consuming configuration

Monitoring

Workload Alterations

Unpredictable faults

Resource & network heterogeneity

UBITECH ubiquitous solutions · TU WIEN TECHNISCHE UNIVERSITÄT WIEN · POLITECNICO DI TORINO · University of Cyprus · DTU · ARISTOTLE UNIVERSITY OF THESSALONIKI · LINKS ISMB SiTI · infineon · INTRASOFT INTERNATIONAL · Suite5 We Deliver Intelligence · K3Y · uni.systems · CRF · BIBA Bremer Institut für Produktion und Logistik · MSP

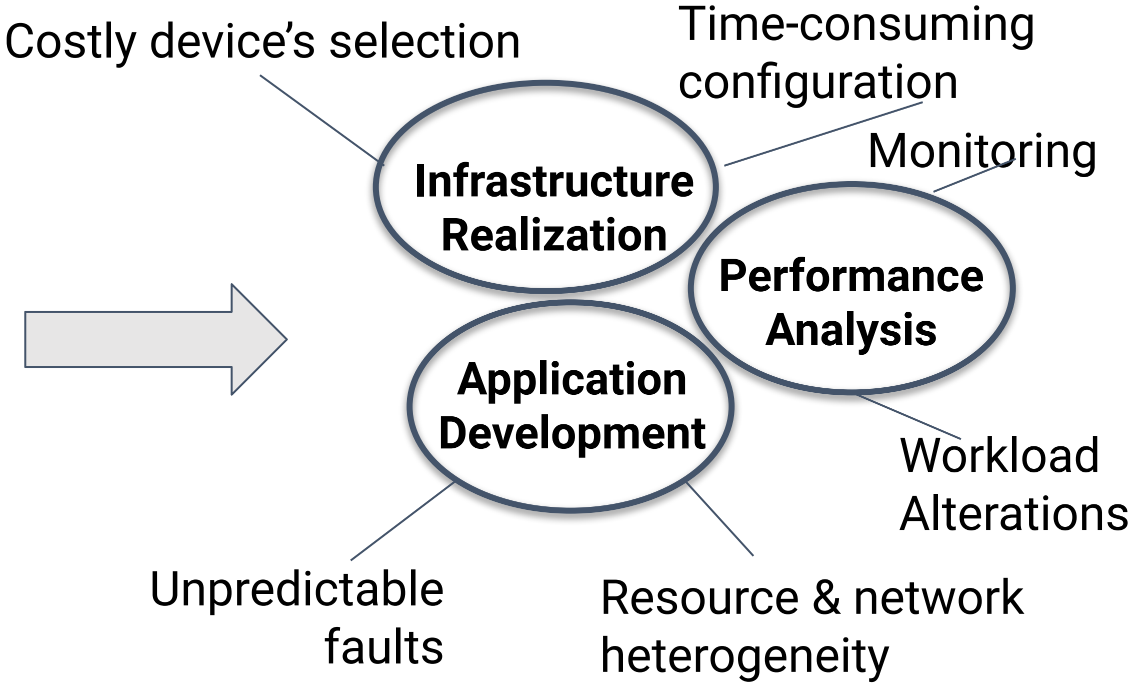
Fog Design and Deployment Challenges

Fog Application Use-cases

UC1. Human-Robot Collaboration in Industrial Ecosystems

UC2. Power Line Surveillance via Swarm of Drones

UC3. Digital Transformation of Urban Mobility



Costly device's selection

Time-consuming configuration

Monitoring

Infrastructure Realization

Performance Analysis

Application Development

Workload Alterations

Unpredictable faults

Resource & network heterogeneity

To this end, the users seek solutions to model and analyze the behavior of Fog infrastructure and IoT services…

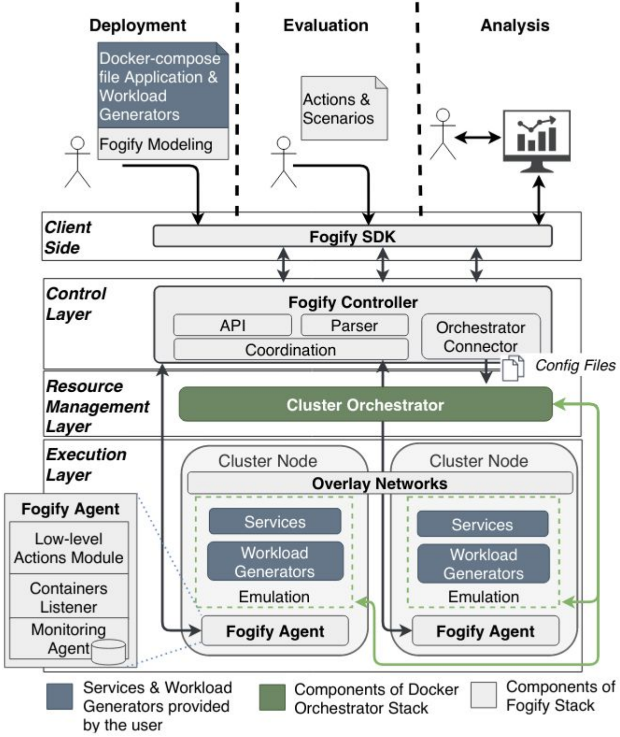
Why do we Need a Fog Emulator?

- Evaluation on real infrastructure is extremely costly and time-consuming (configuration, deployment, etc).

- Why do not use a simulator?
  - Must have models for every piece of infrastructure, application behavior and their interactions… application not actually run…

Emulators mimic production-end environment and application executes in real-time.

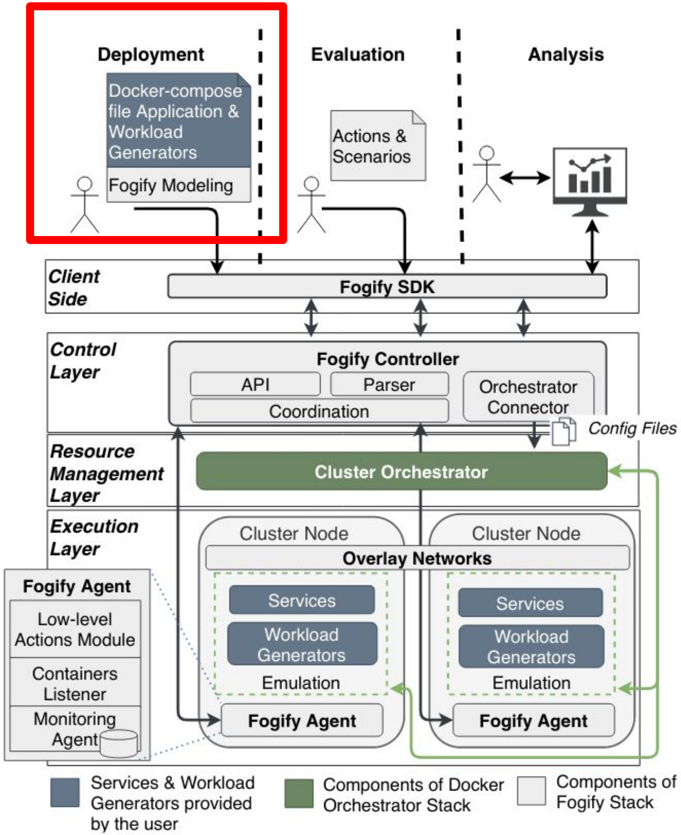
Fogify: Fog Emulation Framework



- Resource Heterogeneity
- Network Link Heterogeneity
- Controllable Faults and Alterations
- Any-scale Experimentation
- Monitoring Capabilities
- Rapid Application Deployment

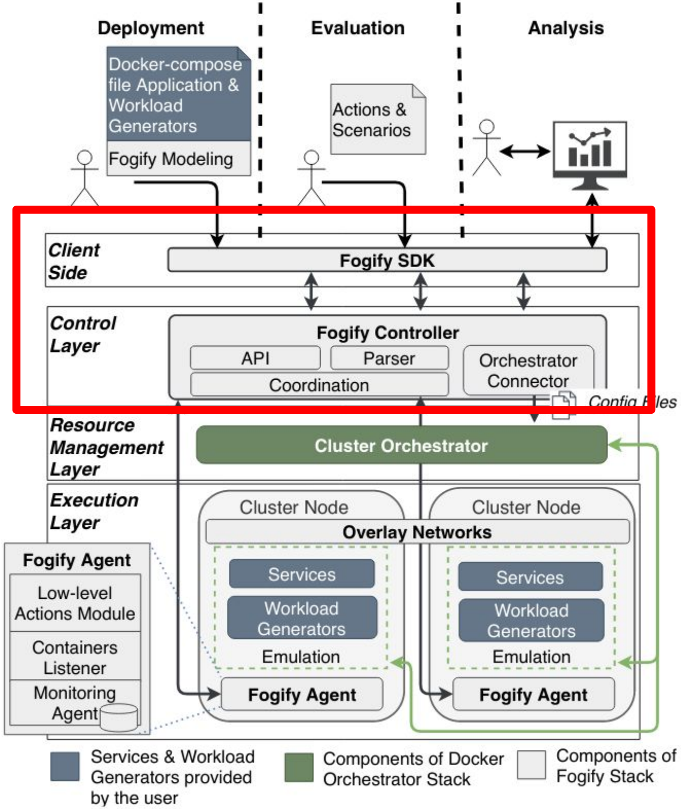
Fogify: A Fog Computing Emulation Framework, M. Symeonides, Z. Georgiou, D. Trihinas, G. Pallis & M. D. Dikaiakos,*EEE/ACM Symposium on Edge Computing (SEC)*, 2020
Demo: Emulating Geo-Distributed Fog Services, M. Symeonides, Z. Georgiou, D. Trihinas, G. Pallis & M. D. Dikaiakos, *IEEE/ACM Symposium on Edge Computing (SEC)*, 2020

Fogify workflow

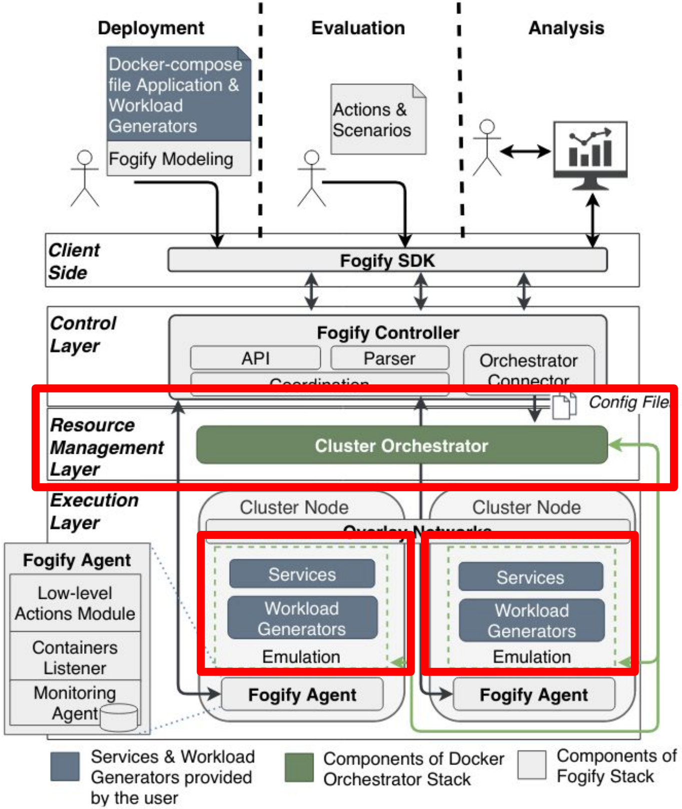


Horizon 2020

Fogify workflow

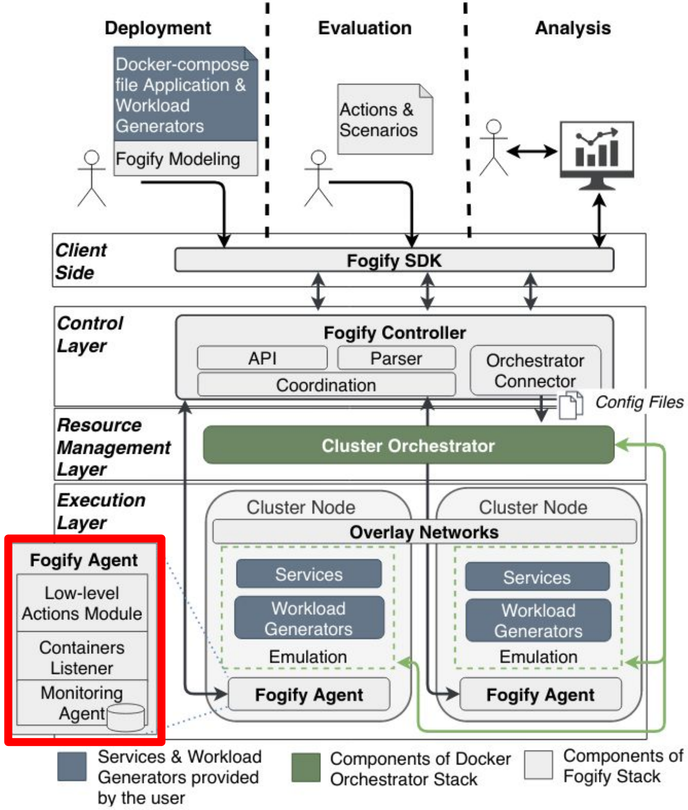


Fogify workflow

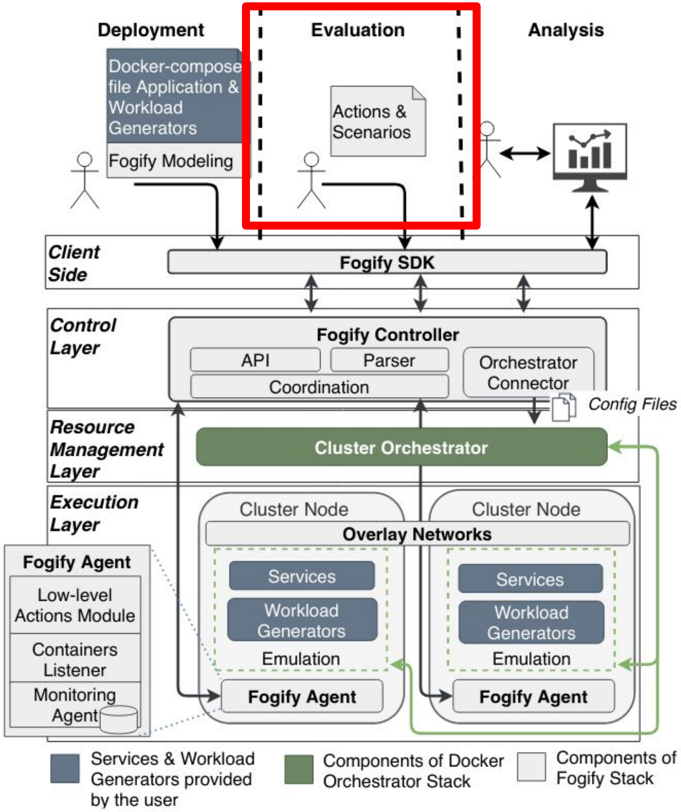


Horizon 2020

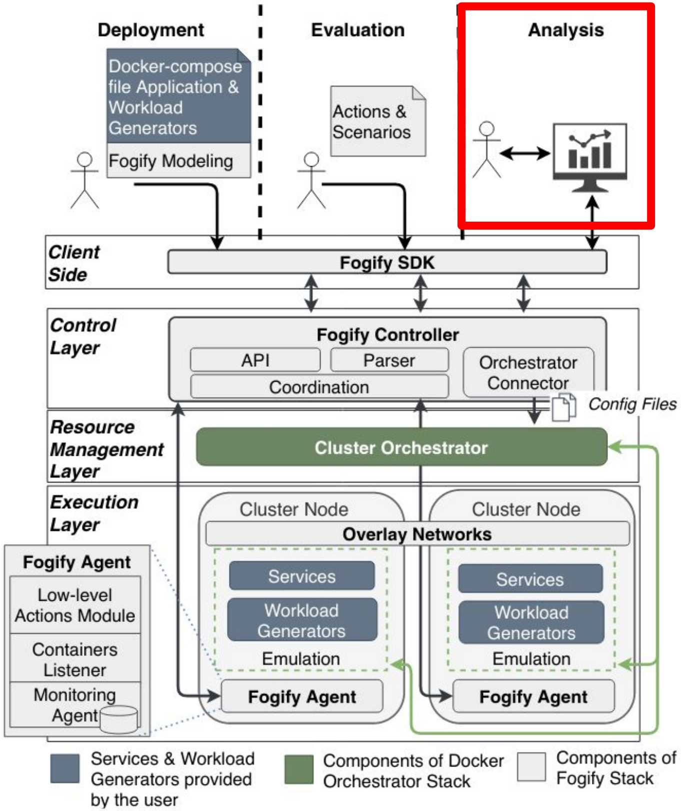
Fogify workflow



Fogify workflow



Fogify workflow



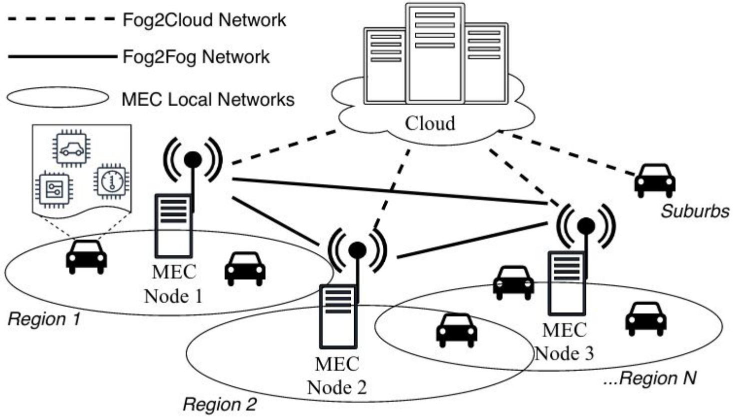
Modeling

Motivating Example

Use-case: A taxi company wants to analyse **region-based data** from its fleet.

In a **real testbed**, the company should:

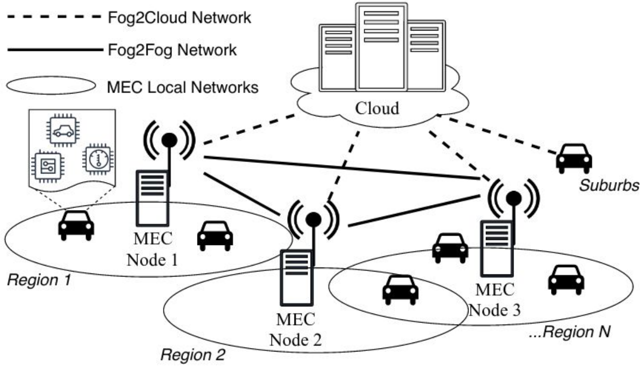
- **Purchase** MECs and taxis sensors
- **Configure** the MECs and **place** them at their physical location
- Setup the **network components**
- **Monitor** the "health" of the infrastructure



At the end, the developers will **not be sure if they tackle every single obstacle of network changes and device's failures**.

A similar application to RAINBOW's UC3 (Digital Transformation of Urban Mobility)

Use-case Modeling in Fogify



```
services: ......
x-fogify:
   nodes: ......
   networks: ......
    topology:
       - label: mec-node-1
         service: mec-svc
         node: mec-node
         networks:
         - name: mec-net-1
           links:
             car-node-at-mec-1:
                 downlink: {
                    latency: 50ms}
         - name: mec-2-cloud-net
         - name: mec-2-mec-net
         replicas: 1
       - label: mec-node-2
         service: .....
```

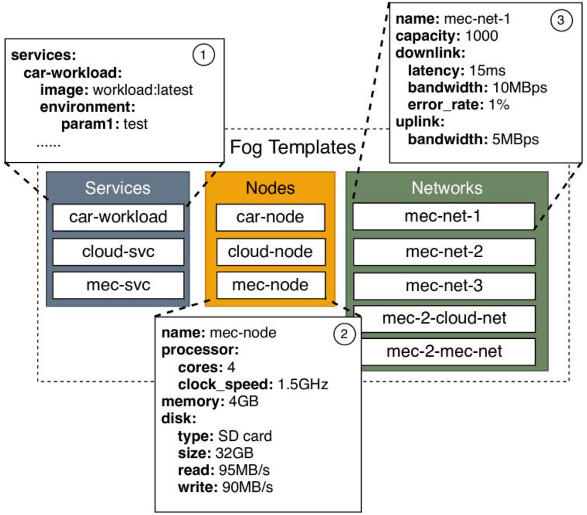
For the Fog Nodes:

- **5 MECs** (4 cores@1.4GHz, 4GB RAM) are placed in 5 different regions (*region-{1-5}*)
- **Taxis/car-node** (1 core@700MHz, 256MB RAM) sending sensed data to nearby MEC
- **Cloud server** (8 cores@2.4GHz, 8GB RAM) computes the final results.

According to the **Network QoS**:

- **Regional Network:** *15ms latency* and up to *10Mbps bandwidth*
- **Edge-Cloud connection:** *100ms latency* and up to *5Mbps bandwidth*

Fog Topology



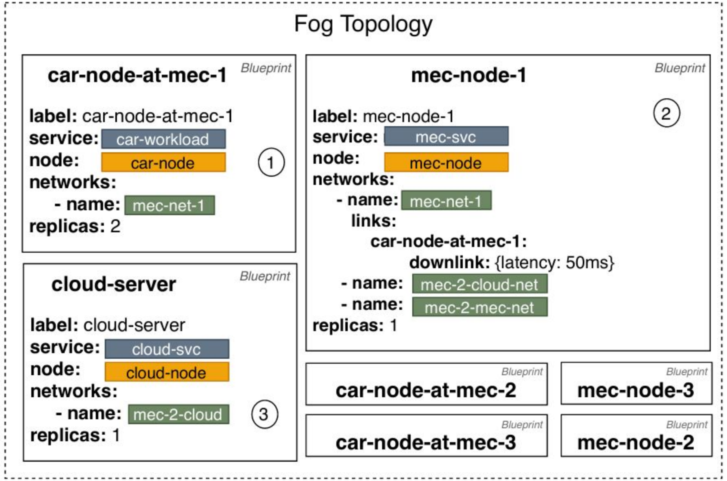
```
services:
  car-workload:
    image: workload:latest
    environment:
      param1: test
  ......
```

```
name: mec-net-1
capacity: 1000
downlink:
  latency: 15ms
  bandwidth: 10MBps
  error_rate: 1%
uplink:
  bandwidth: 5MBps
```

```
name: mec-node
processor:
  cores: 4
  clock_speed: 1.5GHz
memory: 4GB
disk:
  type: SD card
  size: 32GB
  read: 95MB/s
  write: 90MB/s
```

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



```
car-node-at-mec-1
label: car-node-at-mec-1
service: car-workload
node: car-node
networks:
  - name: mec-net-1
replicas: 2
```

```
mec-node-1
label: mec-node-1
service: mec-svc
node: mec-node
networks:
  - name: mec-net-1
    links:
      car-node-at-mec-1:
        downlink: {latency: 50ms}
  - name: mec-2-cloud-net
  - name: mec-2-mec-net
replicas: 1
```

```
cloud-server
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
  - name: mec-2-cloud
replicas: 1
```

A **Fog Topology** consists of Blueprints.

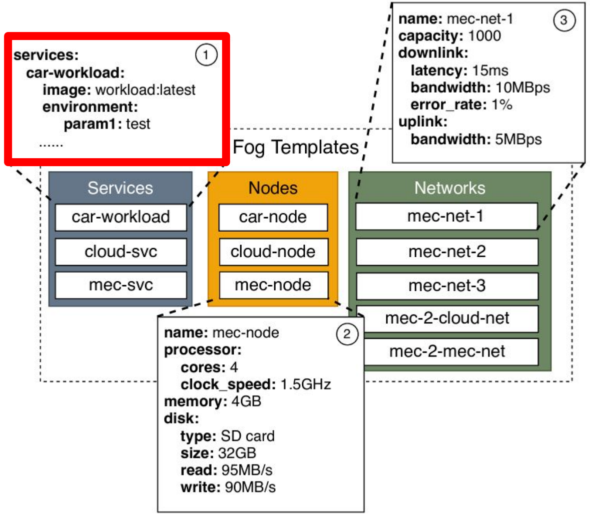
- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*

Blueprints support the ***overriding*** *of* ***network-level QoS*** *and* ***specific links*** *between Fog Nodes.*

Fog Topology

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



Fog Templates

Services	Nodes	Networks
car-workload	car-node	mec-net-1
cloud-svc	cloud-node	mec-net-2
mec-svc	mec-node	mec-net-3
		mec-2-cloud-net
		mec-2-mec-net

1

```
services:
  car-workload:
    image: workload:latest
    environment:
        param1: test
  ......
```

2

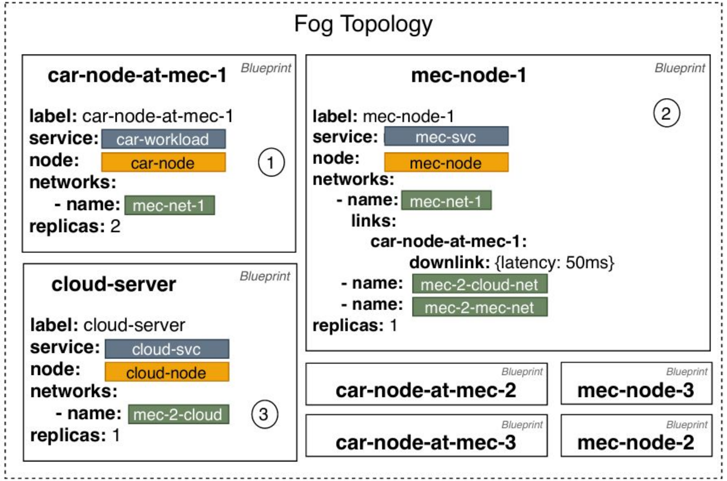
```
name: mec-node
processor:
   cores: 4
   clock_speed: 1.5GHz
memory: 4GB
disk:
   type: SD card
   size: 32GB
   read: 95MB/s
   write: 90MB/s
```

3

```
name: mec-net-1
capacity: 1000
downlink:
   latency: 15ms
   bandwidth: 10MBps
   error_rate: 1%
uplink:
   bandwidth: 5MBps
```

A **Fog Topology** consists of Blueprints.

- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*



Fog Topology

car-node-at-mec-1 — Blueprint — 1

```
label: car-node-at-mec-1
service: car-workload
node: car-node
networks:
   - name: mec-net-1
replicas: 2
```

mec-node-1 — Blueprint — 2

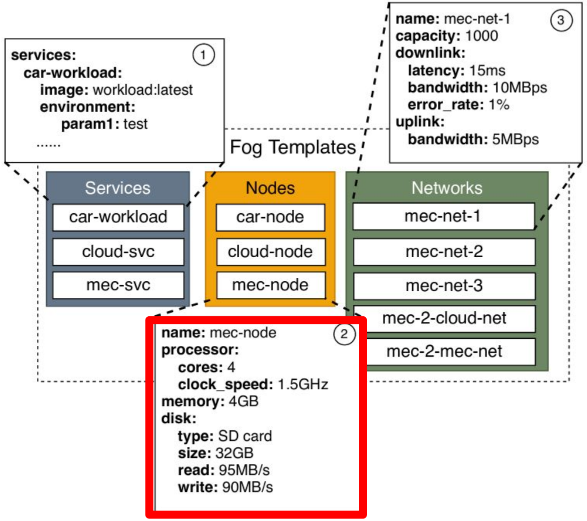
```
label: mec-node-1
service: mec-svc
node: mec-node
networks:
   - name: mec-net-1
     links:
        car-node-at-mec-1:
           downlink: {latency: 50ms}
   - name: mec-2-cloud-net
   - name: mec-2-mec-net
replicas: 1
```

cloud-server — Blueprint — 3

```
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
   - name: mec-2-cloud
replicas: 1
```

*Blueprints support the **overriding** of **network-level QoS** and **specific links** between Fog Nodes.*

Fog Topology



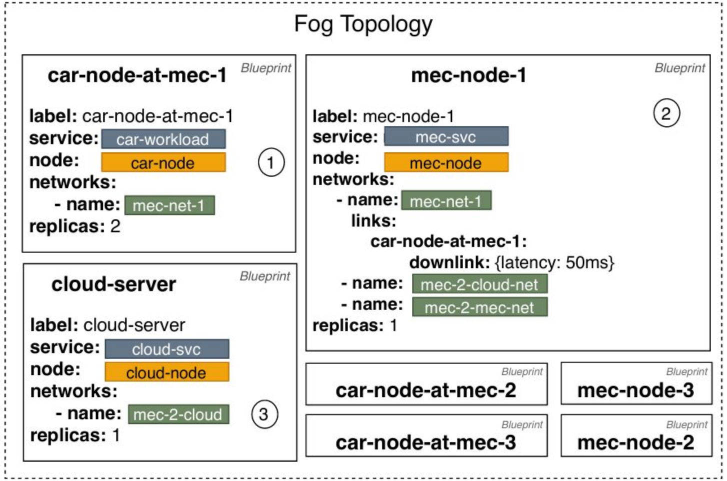
```
services:
  car-workload:
    image: workload:latest
    environment:
      param1: test
    ......
```

```
name: mec-net-1
capacity: 1000
downlink:
  latency: 15ms
  bandwidth: 10MBps
  error_rate: 1%
uplink:
  bandwidth: 5MBps
```

```
name: mec-node
processor:
  cores: 4
  clock_speed: 1.5GHz
memory: 4GB
disk:
  type: SD card
  size: 32GB
  read: 95MB/s
  write: 90MB/s
```

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



```
car-node-at-mec-1
label: car-node-at-mec-1
service: car-workload
node: car-node
networks:
  - name: mec-net-1
replicas: 2
```

```
mec-node-1
label: mec-node-1
service: mec-svc
node: mec-node
networks:
  - name: mec-net-1
    links:
      car-node-at-mec-1:
        downlink: {latency: 50ms}
  - name: mec-2-cloud-net
  - name: mec-2-mec-net
replicas: 1
```

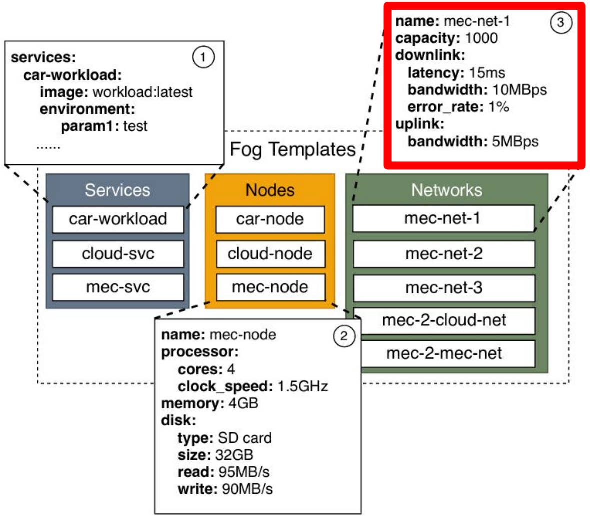
```
cloud-server
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
  - name: mec-2-cloud
replicas: 1
```

A **Fog Topology** consists of Blueprints.

- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*

*Blueprints support the **overriding** of **network-level QoS** and **specific links** between Fog Nodes.*

Fog Topology



```
services:                 ①
  car-workload:
    image: workload:latest
    environment:
      param1: test
    ......
```

Fog Templates

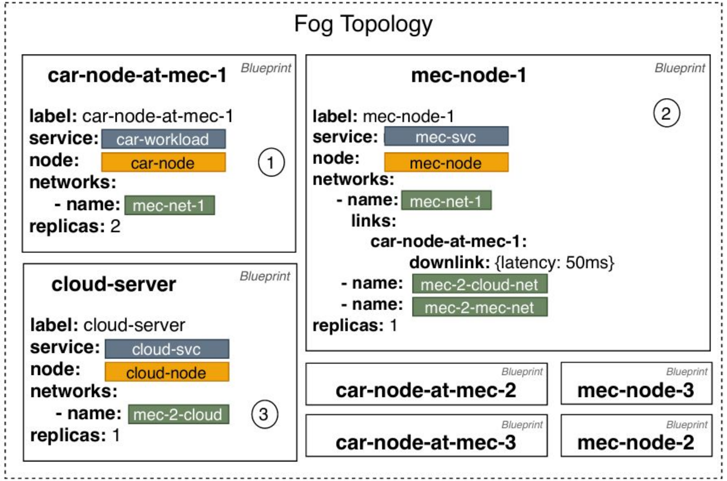
Services	Nodes	Networks
car-workload	car-node	mec-net-1
cloud-svc	cloud-node	mec-net-2
mec-svc	mec-node	mec-net-3
		mec-2-cloud-net
		mec-2-mec-net

```
name: mec-net-1           ③
capacity: 1000
downlink:
  latency: 15ms
  bandwidth: 10MBps
  error_rate: 1%
uplink:
  bandwidth: 5MBps
```

```
name: mec-node            ②
processor:
  cores: 4
  clock_speed: 1.5GHz
memory: 4GB
disk:
  type: SD card
  size: 32GB
  read: 95MB/s
  write: 90MB/s
```

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



Fog Topology

```
car-node-at-mec-1                    Blueprint
label: car-node-at-mec-1
service: car-workload                ①
node: car-node
networks:
  - name: mec-net-1
replicas: 2
```

```
mec-node-1                           Blueprint
label: mec-node-1                    ②
service: mec-svc
node: mec-node
networks:
  - name: mec-net-1
    links:
      car-node-at-mec-1:
        downlink: {latency: 50ms}
  - name: mec-2-cloud-net
  - name: mec-2-mec-net
replicas: 1
```

```
cloud-server                         Blueprint
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
  - name: mec-2-cloud                ③
replicas: 1
```

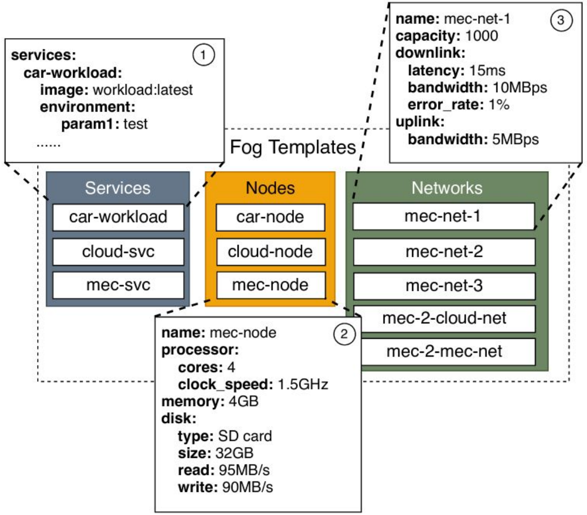
Blueprint car-node-at-mec-2	Blueprint mec-node-3
Blueprint car-node-at-mec-3	Blueprint mec-node-2

A **Fog Topology** consists of Blueprints.

- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*

*Blueprints support the **overriding** of **network-level QoS** and **specific links** between Fog Nodes.*

Fog Topology



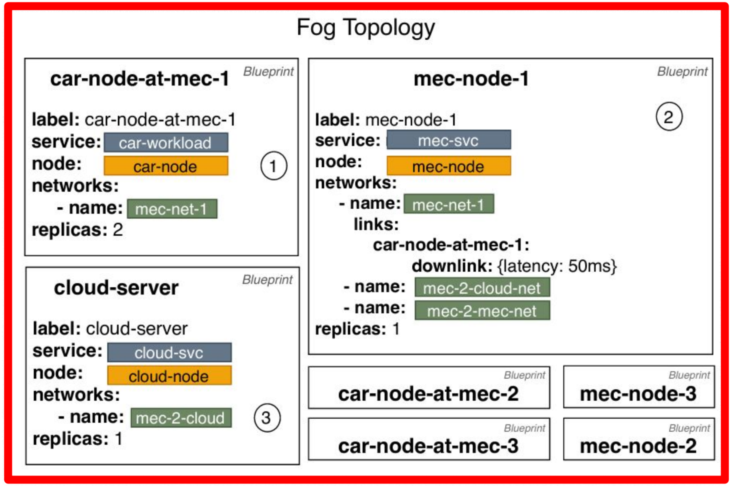
```
services:
  car-workload:
    image: workload:latest
    environment:
      param1: test
  ......
```

```
name: mec-net-1
capacity: 1000
downlink:
  latency: 15ms
  bandwidth: 10MBps
  error_rate: 1%
uplink:
  bandwidth: 5MBps
```

```
name: mec-node
processor:
  cores: 4
  clock_speed: 1.5GHz
memory: 4GB
disk:
  type: SD card
  size: 32GB
  read: 95MB/s
  write: 90MB/s
```

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



```
car-node-at-mec-1
label: car-node-at-mec-1
service: car-workload
node: car-node
networks:
  - name: mec-net-1
replicas: 2
```

```
mec-node-1
label: mec-node-1
service: mec-svc
node: mec-node
networks:
  - name: mec-net-1
    links:
      car-node-at-mec-1:
        downlink: {latency: 50ms}
  - name: mec-2-cloud-net
  - name: mec-2-mec-net
replicas: 1
```

```
cloud-server
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
  - name: mec-2-cloud
replicas: 1
```

A **Fog Topology** consists of Blueprints.

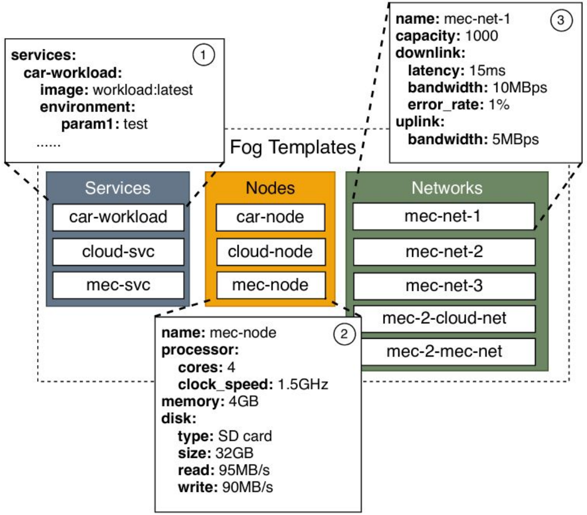
- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*

*Blueprints support the **overriding** of **network-level QoS** and **specific links** between Fog Nodes.*

Fog Topology

The initial **Fog Templates** of Fogify consist of:

- a set of *Services*,
- a set of *Nodes*,
- a set of *Networks*



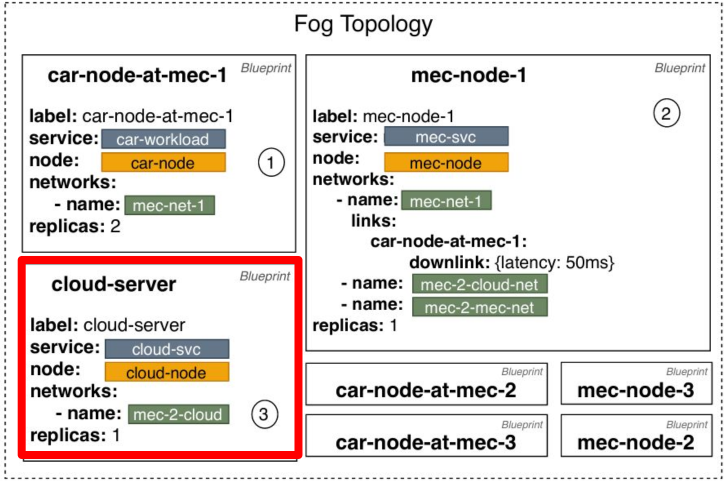
```
services:
  car-workload:
    image: workload:latest
    environment:
      param1: test
    ......
```

```
name: mec-net-1
capacity: 1000
downlink:
  latency: 15ms
  bandwidth: 10MBps
  error_rate: 1%
uplink:
  bandwidth: 5MBps
```

```
name: mec-node
processor:
  cores: 4
  clock_speed: 1.5GHz
memory: 4GB
disk:
  type: SD card
  size: 32GB
  read: 95MB/s
  write: 90MB/s
```

A **Fog Topology** consists of Blueprints.

- A **Blueprint** is a combination of a *Node*, *Service*, set of *Networks*, *replicas* and a *label*



```
car-node-at-mec-1
label: car-node-at-mec-1
service: car-workload
node: car-node
networks:
  - name: mec-net-1
replicas: 2
```

```
mec-node-1
label: mec-node-1
service: mec-svc
node: mec-node
networks:
  - name: mec-net-1
    links:
      car-node-at-mec-1:
        downlink: {latency: 50ms}
  - name: mec-2-cloud-net
  - name: mec-2-mec-net
replicas: 1
```

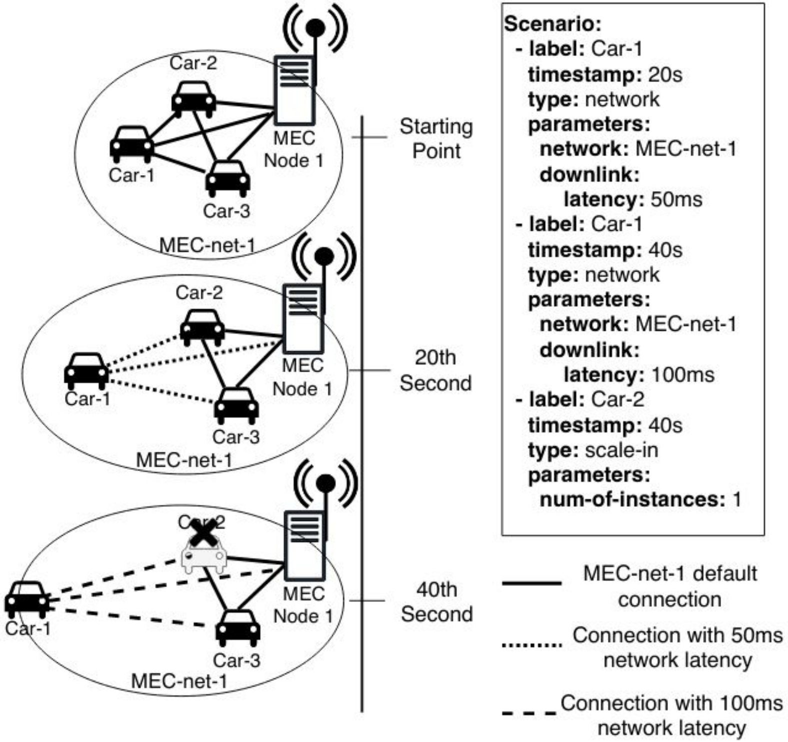
```
cloud-server
label: cloud-server
service: cloud-svc
node: cloud-node
networks:
  - name: mec-2-cloud
replicas: 1
```

*Blueprints support the **overriding** of **network-level QoS** and **specific links** between Fog Nodes.*

Runtime Evaluation Model

The evaluation model consists of:

- **Actions** that change properties of a running Fog Topology. Actions can be:
  - *Scaling Actions* (horizontal or vertical)
  - *Network Actions*
  - *Stress Actions*
- **Scenario** is a *sequence of time scheduled actions* that Fogify will execute to emulate more complex user-driven experiments.

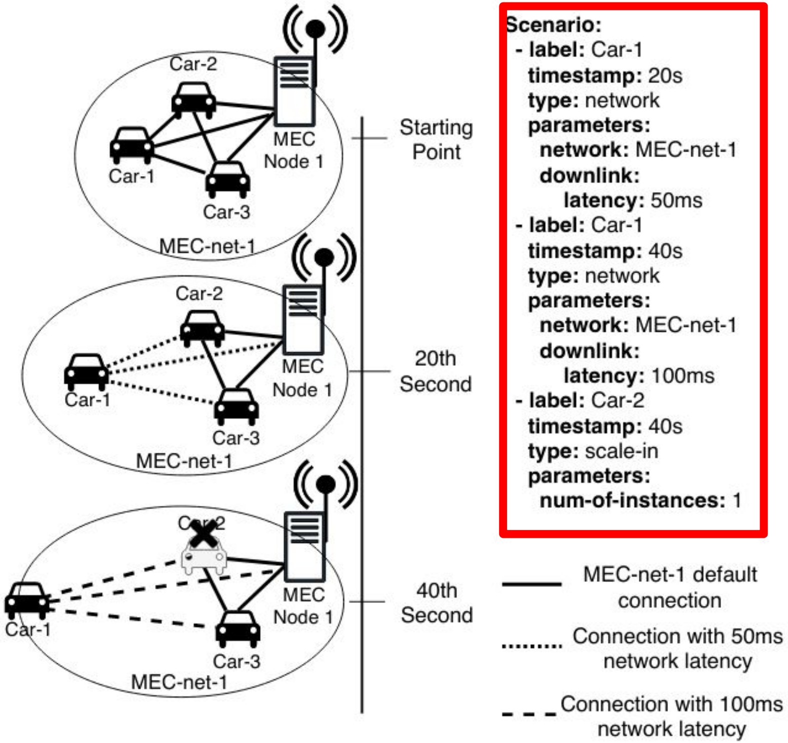


```
Scenario:
 - label: Car-1
   timestamp: 20s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
        latency: 50ms
 - label: Car-1
   timestamp: 40s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
        latency: 100ms
 - label: Car-2
   timestamp: 40s
   type: scale-in
   parameters:
     num-of-instances: 1
```

Runtime Evaluation Model

The evaluation model consists of:

- **Actions** that change properties of a running Fog Topology. Actions can be:
  - ○ *Scaling Actions* (horizontal or vertical)
  - ○ *Network Actions*
  - ○ *Stress Actions*
- **Scenario** is a *sequence of time scheduled actions* that Fogify will execute to emulate more complex user-driven experiments.

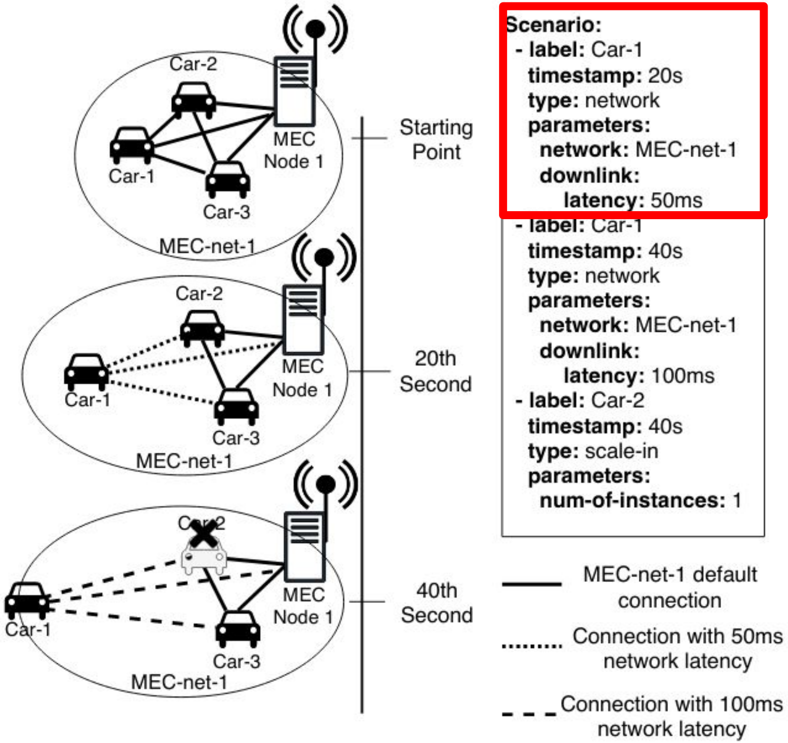

Car-2
MEC Node 1
Car-1
Car-3
MEC-net-1
Starting Point
20th Second
40th Second
MEC-net-1 default connection
Connection with 50ms network latency
Connection with 100ms network latency

```
Scenario:
 - label: Car-1
   timestamp: 20s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
       latency: 50ms
 - label: Car-1
   timestamp: 40s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
       latency: 100ms
 - label: Car-2
   timestamp: 40s
   type: scale-in
   parameters:
     num-of-instances: 1
```

Runtime Evaluation Model

The evaluation model consists of:

- **Actions** that change properties of a running Fog Topology. Actions can be:
  - *Scaling Actions* (horizontal or vertical)
  - *Network Actions*
  - *Stress Actions*
- **Scenario** is a *sequence of time scheduled actions* that Fogify will execute to emulate more complex user-driven experiments.

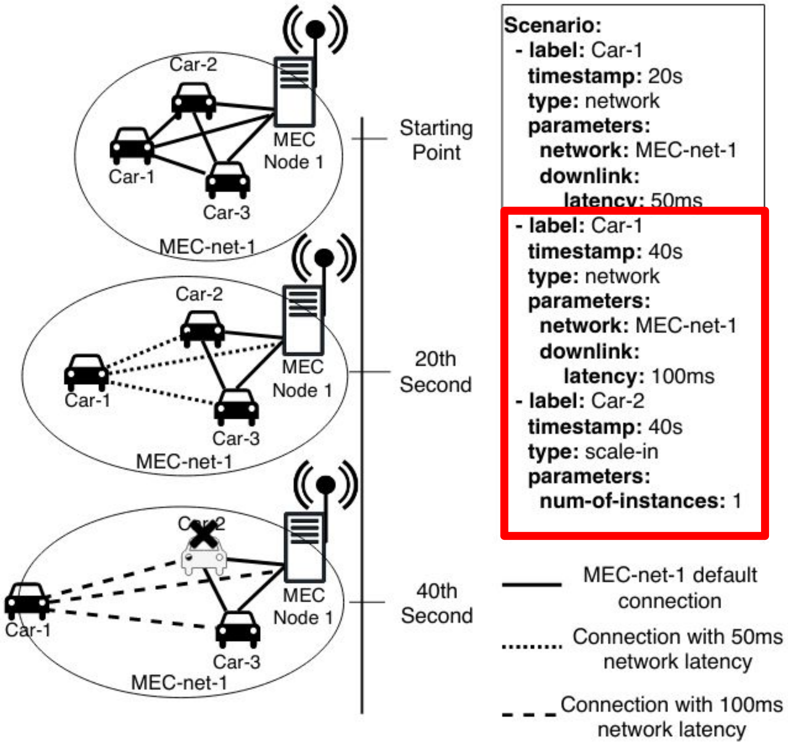

Car-2
Car-1
Car-3
MEC Node 1
MEC-net-1
Starting Point
20th Second
40th Second
MEC-net-1 default connection
Connection with 50ms network latency
Connection with 100ms network latency

```
Scenario:
 - label: Car-1
   timestamp: 20s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
       latency: 50ms
 - label: Car-1
   timestamp: 40s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
       latency: 100ms
 - label: Car-2
   timestamp: 40s
   type: scale-in
   parameters:
     num-of-instances: 1
```

Runtime Evaluation Model

The evaluation model consists of:

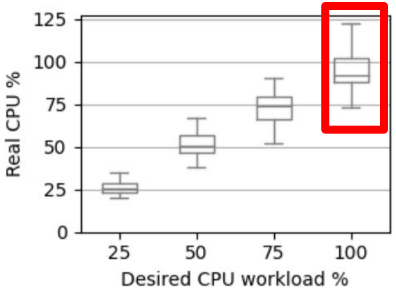
- **Actions** that change properties of a running Fog Topology. Actions can be:
  - *Scaling Actions* (horizontal or vertical)
  - *Network Actions*
  - *Stress Actions*
- **Scenario** is a *sequence of time scheduled actions* that Fogify will execute to emulate more complex user-driven experiments.


Car-2
Car-1
Car-3
MEC Node 1
MEC-net-1
Starting Point
20th Second
40th Second
MEC-net-1 default connection
Connection with 50ms network latency
Connection with 100ms network latency

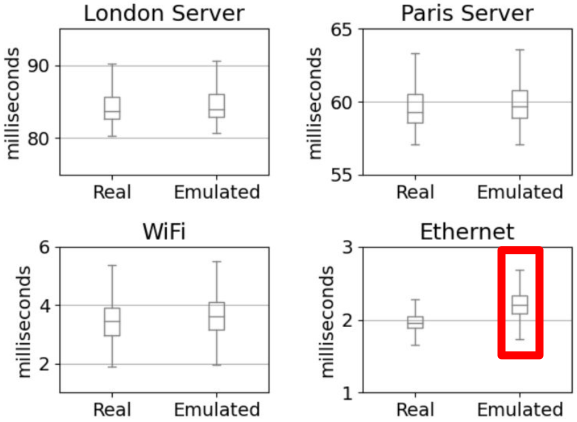
```
Scenario:
 - label: Car-1
   timestamp: 20s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
         latency: 50ms
 - label: Car-1
   timestamp: 40s
   type: network
   parameters:
     network: MEC-net-1
     downlink:
         latency: 100ms
 - label: Car-2
   timestamp: 40s
   type: scale-in
   parameters:
     num-of-instances: 1
```

Evaluation

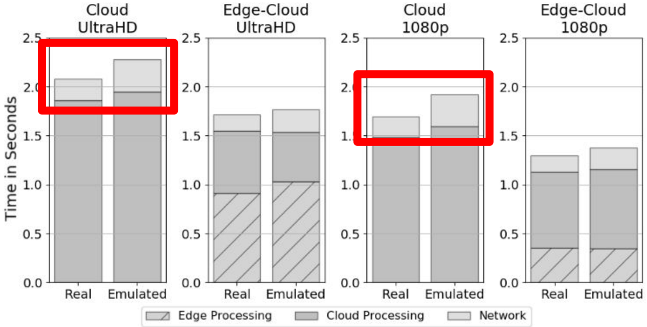
Emulation Accuracy



The **emulated computing resources** has only a *small performance degradation* for workloads approaching 100% CPU usage



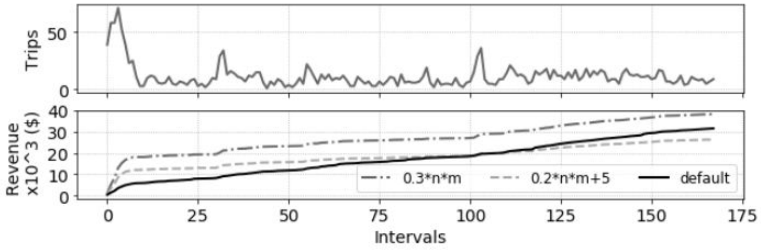
Fogify achieves near to **real-world network link capabilities**, with *only outliers* not captured and a *slight overhead in low-latency connections*

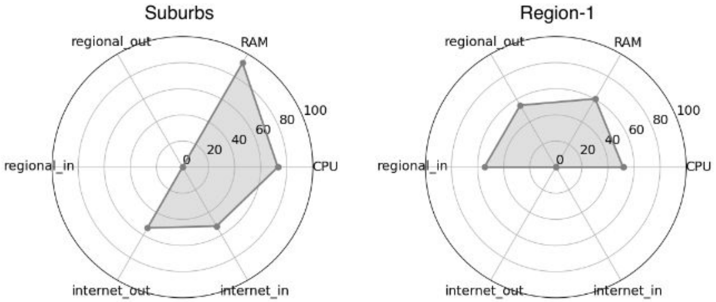


The emulation results **closely follow the real measurements** with a *5%-8% deviation* of the overall experiment time.

Scenarios Evaluation

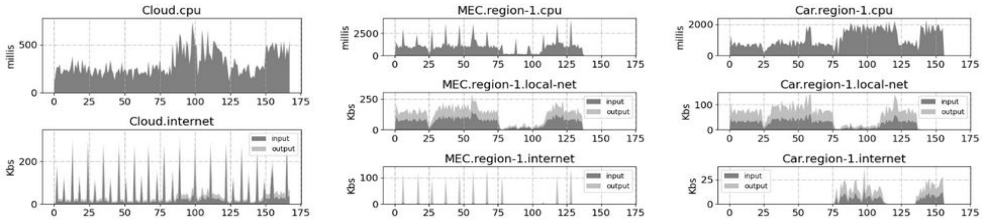
Application-level metrics: Operators can employ Fogify to produce and evaluate analytic insights, implementing adequate app-level metrics.





Node profiling: insights are highly beneficial to engineers for capacity planning, optimizing service and resource placement.

Scaling Actions, Network Alterations & Workload Changes: released insights about service performance and resources utilization.



How do we use Fogify in the RAINBOW project?

Performance evaluation of WP4’s components

- Evaluation of scheduling algorithms for RAINBOW’s analytic layer (Apache Storm)
- Benchmarking and performance analysis of RAINBOW’s storage layer (Apache Ignite vs Redis)

Quantitative experiments of Human-Robot Collaboration in Industrial Ecosystems (UC1)

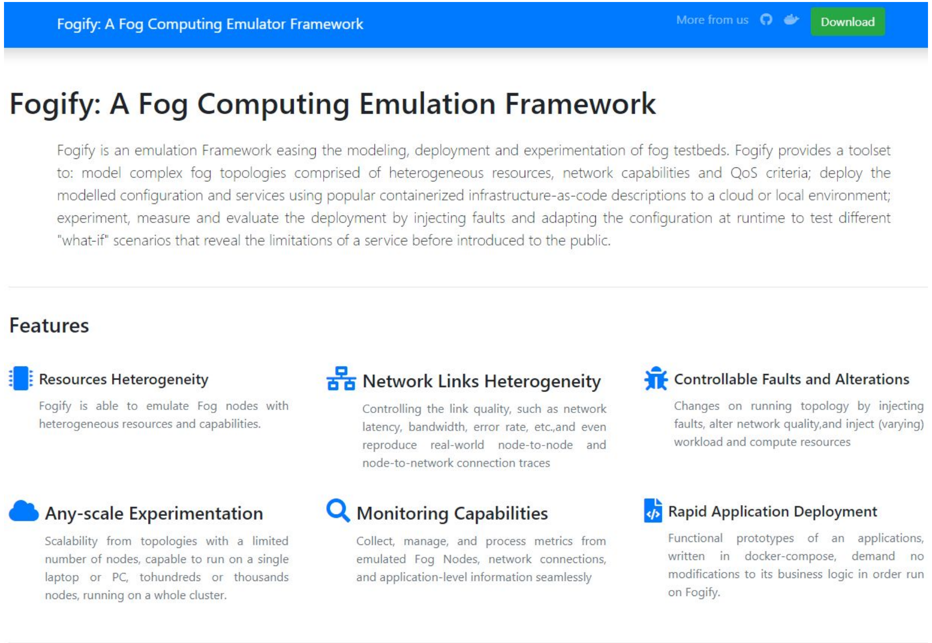
- Microservice application with more than 8 containerized services including queues (MQTT), in-memory databases (Redis), workloads, etc.

In creation of **Fog service placement algorithms** for K8s clusters (work in progress)

- Currently, we have integrated kind (k8s in docker) with Fogify

Horizon 2020

This work is partially supported by the EU Commission through RAINBOW 871403 (ICT-15-2019-2020) project and by the Cyprus Research and Innovation Foundation through COMPLEMENTARY/0916/0916/0171 project.



Fogify: A Fog Computing Emulation Framework

Fogify is an emulation Framework easing the modeling, deployment and experimentation of fog testbeds. Fogify provides a toolset to: model complex fog topologies comprised of heterogeneous resources, network capabilities and QoS criteria; deploy the modelled configuration and services using popular containerized infrastructure-as-code descriptions to a cloud or local environment; experiment, measure and evaluate the deployment by injecting faults and adapting the configuration at runtime to test different "what-if" scenarios that reveal the limitations of a service before introduced to the public.

Features

Resources Heterogeneity

Fogify is able to emulate Fog nodes with heterogeneous resources and capabilities.

Network Links Heterogeneity

Controlling the link quality, such as network latency, bandwidth, error rate, etc.,and even reproduce real-world node-to-node and node-to-network connection traces

Controllable Faults and Alterations

Changes on running topology by injecting faults, alter network quality,and inject (varying) workload and compute resources

Any-scale Experimentation

Scalability from topologies with a limited number of nodes, capable to run on a single laptop or PC, tohundreds or thousands nodes, running on a whole cluster.

Monitoring Capabilities

Collect, manage, and process metrics from emulated Fog Nodes, network connections, and application-level information seamlessly

Rapid Application Deployment

Functional prototypes of an applications, written in docker-compose, demand no modifications to its business logic in order run on Fogify.

Give it a try!

Moysis Symeonides
msymeo03@cs.ucy.ac.cy

Github: https://github.com/UCY-LINC-LAB/fogify

Demo: https://github.com/UCY-LINC-LAB/fogify-demo

Documentation: https://ucy-linc-lab.github.io/fogify/

Video: https://www.youtube.com/watch?v=PthMM6rC89o