

Fog Computing Project

MockFog



Emulating a realistic Fog Computing environment
on virtualized infrastructure.

Sören Becker

Miro Conzelmann

Elias Grünewald

Sascha Huk

Michael Narodovitch

Meike Stoldt

Franz Tscharf

16th July, 2018



Motivation

Use case

Implementation

Walkthrough

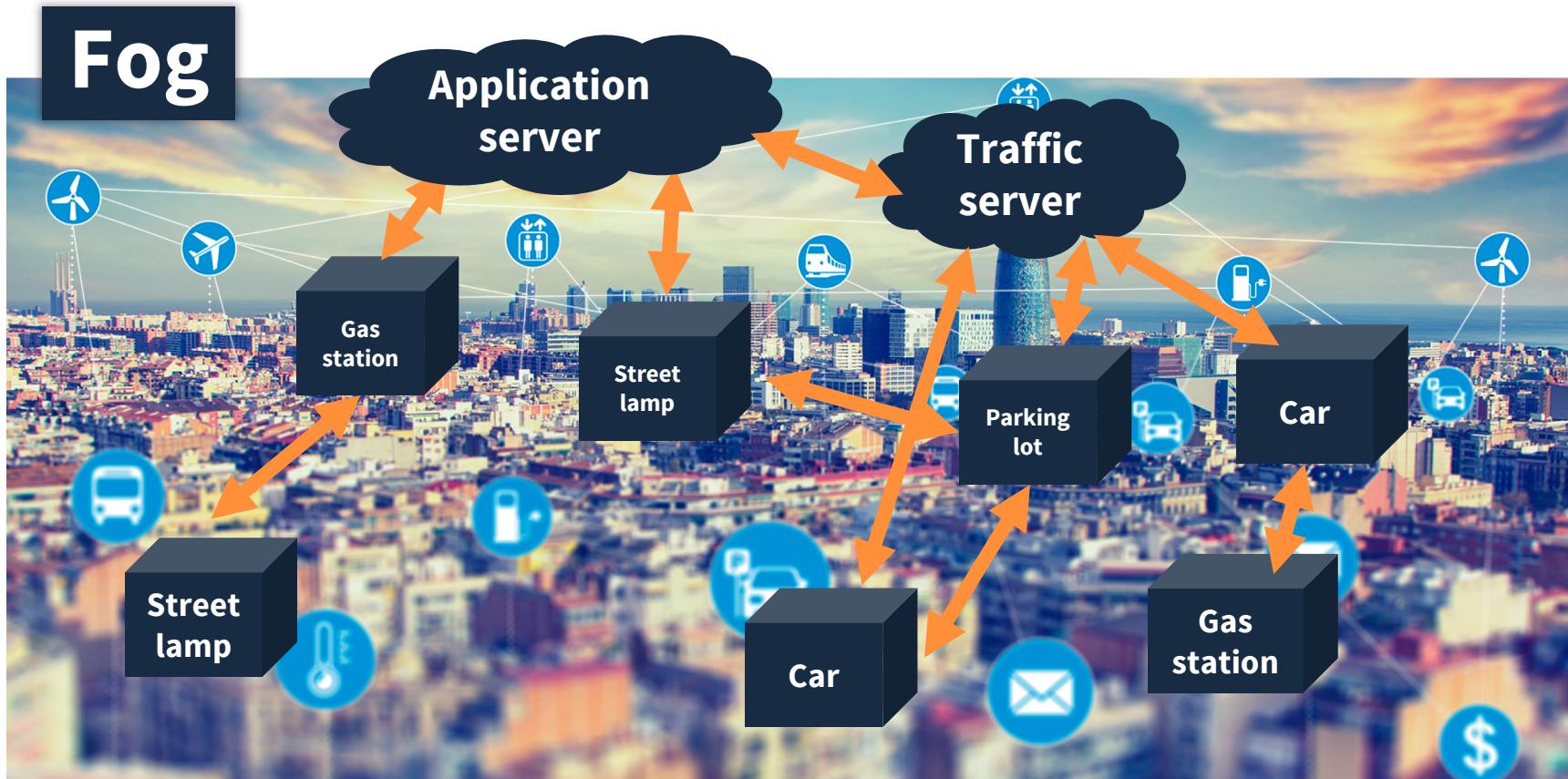
Conclusion



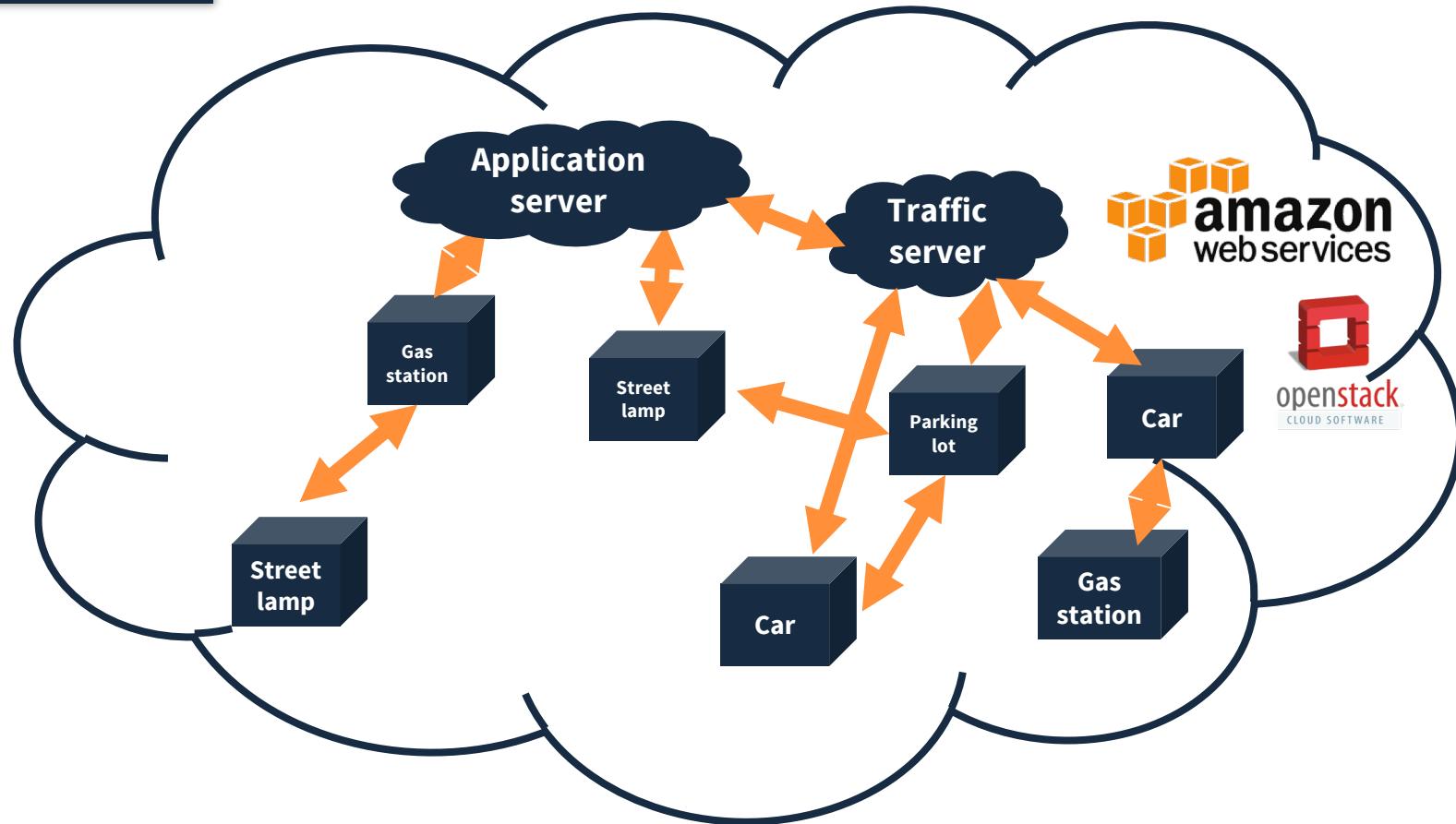
Barcelona 2018+



Use Case: Smart City Scenario

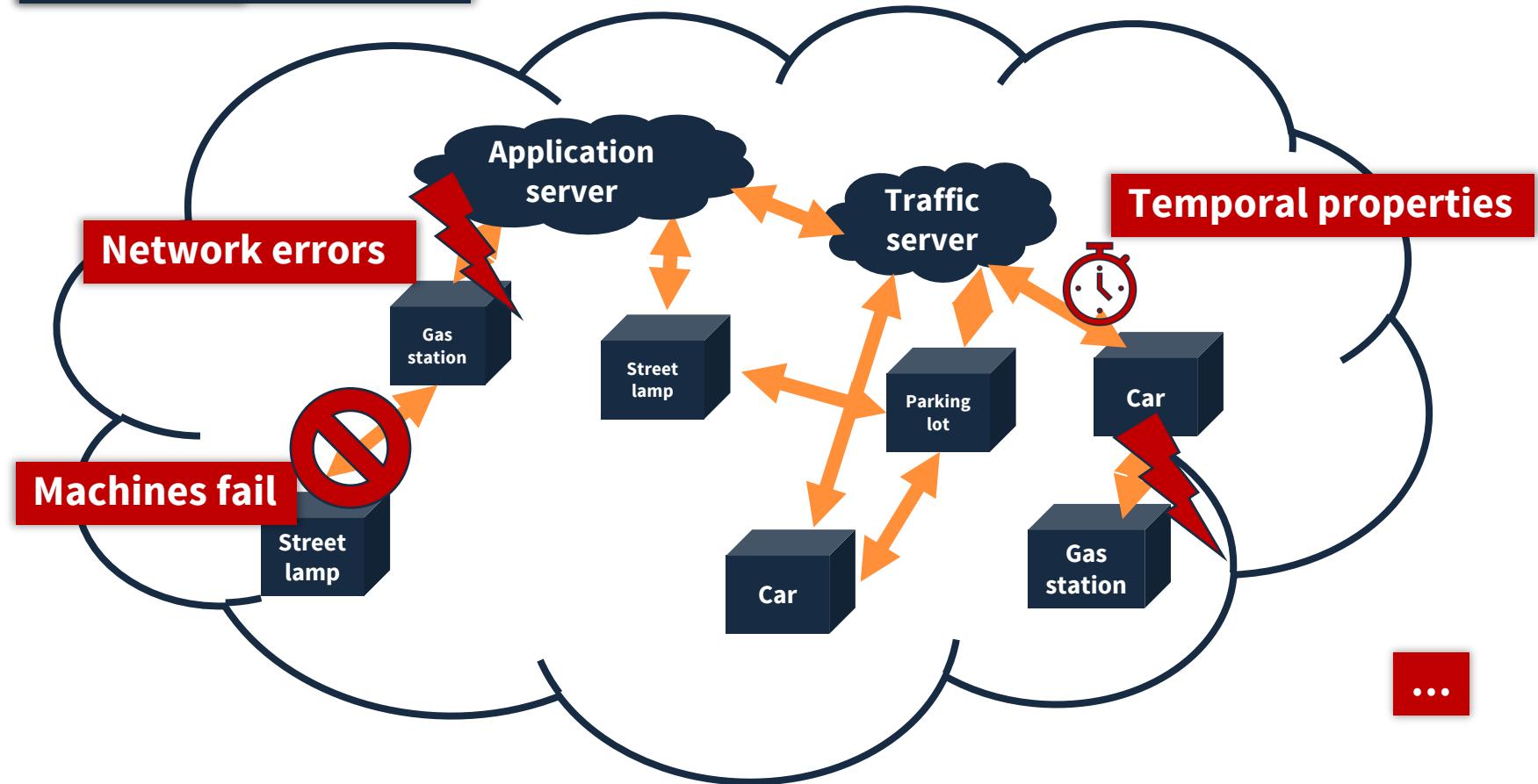


Idea



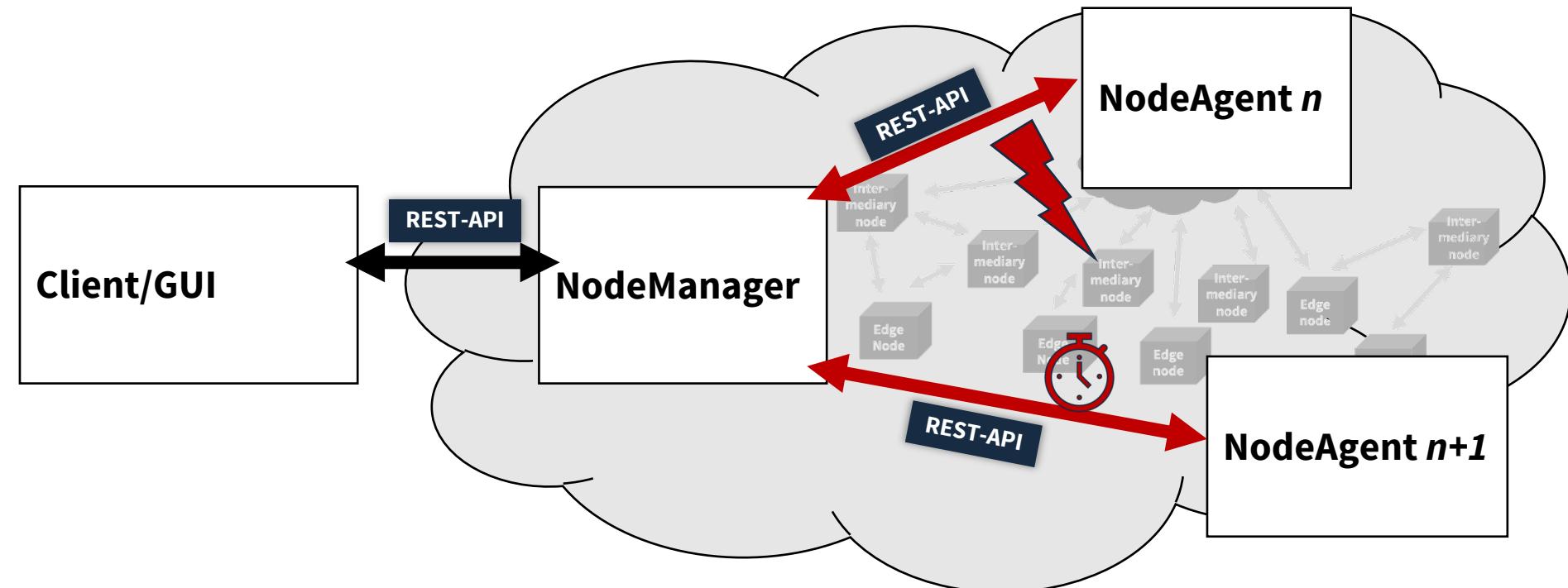
**Emulating a realistic Fog Computing environment
on virtualized infrastructure**

Challenge



How to assign these properties at runtime?

Main components



Log in to cloud

Define network topology

Bootstrap environment

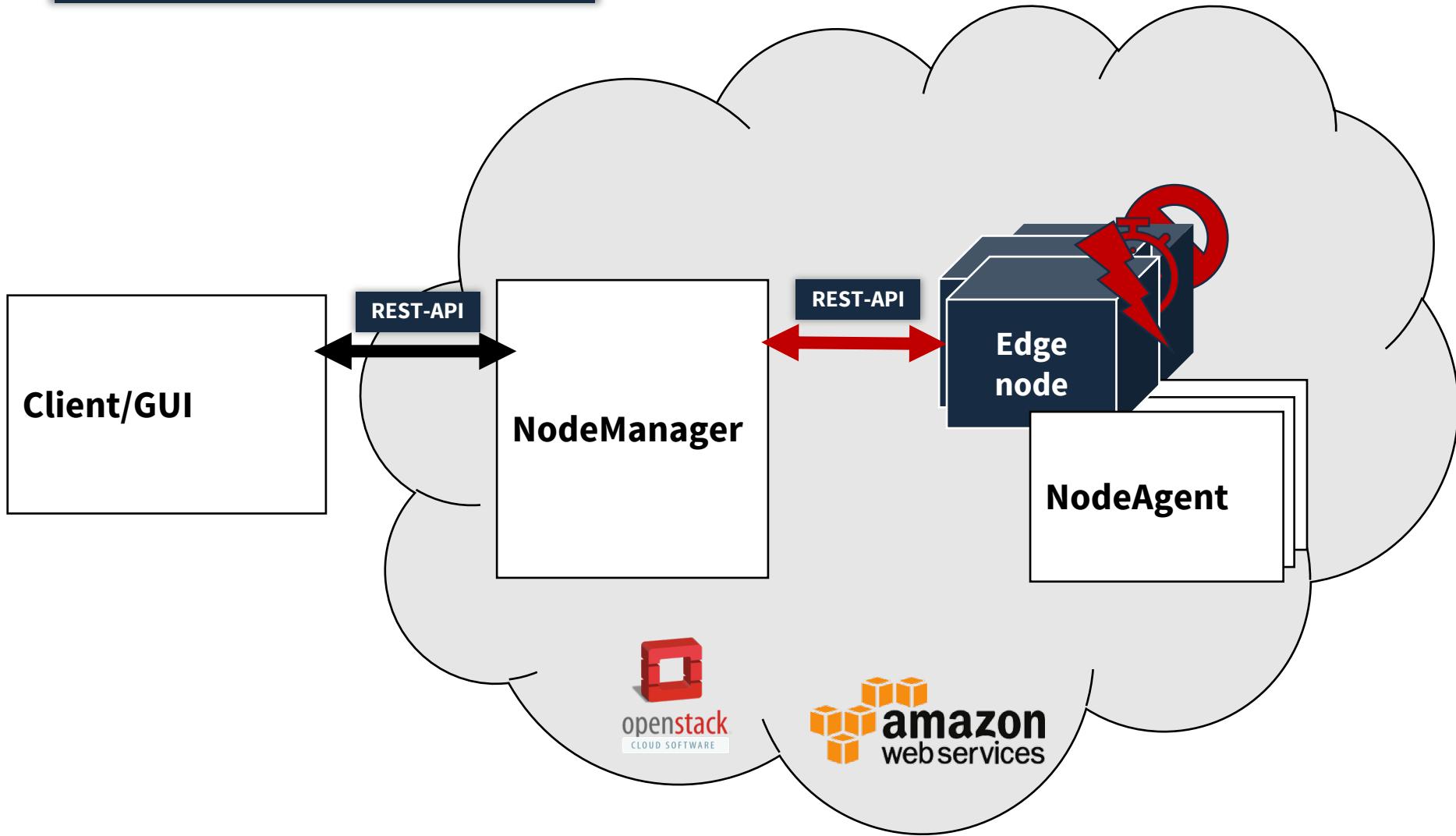
Assign properties

Destroy environment

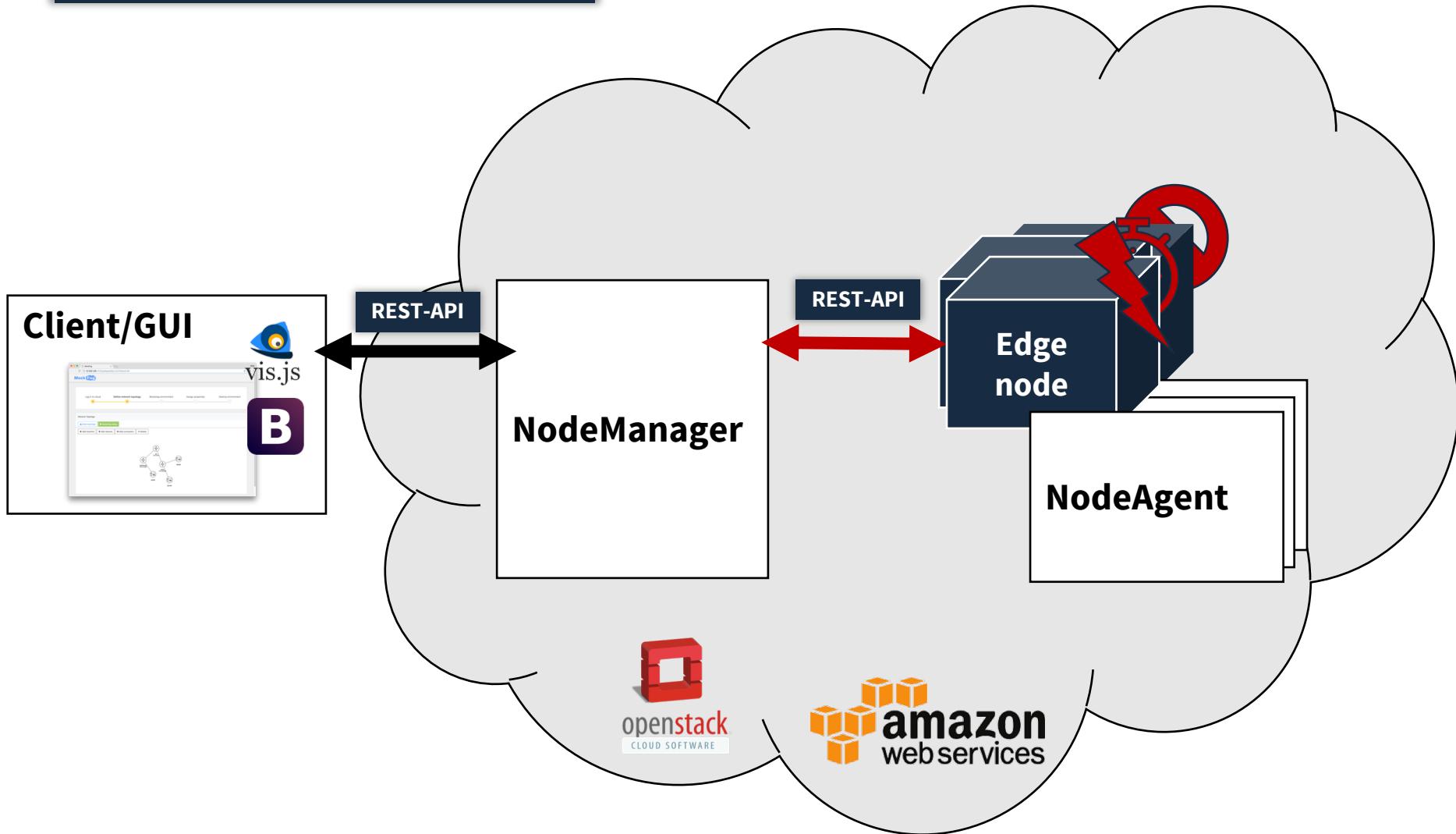
Demo

<https://youtu.be/Vkinxu6Wdx8>

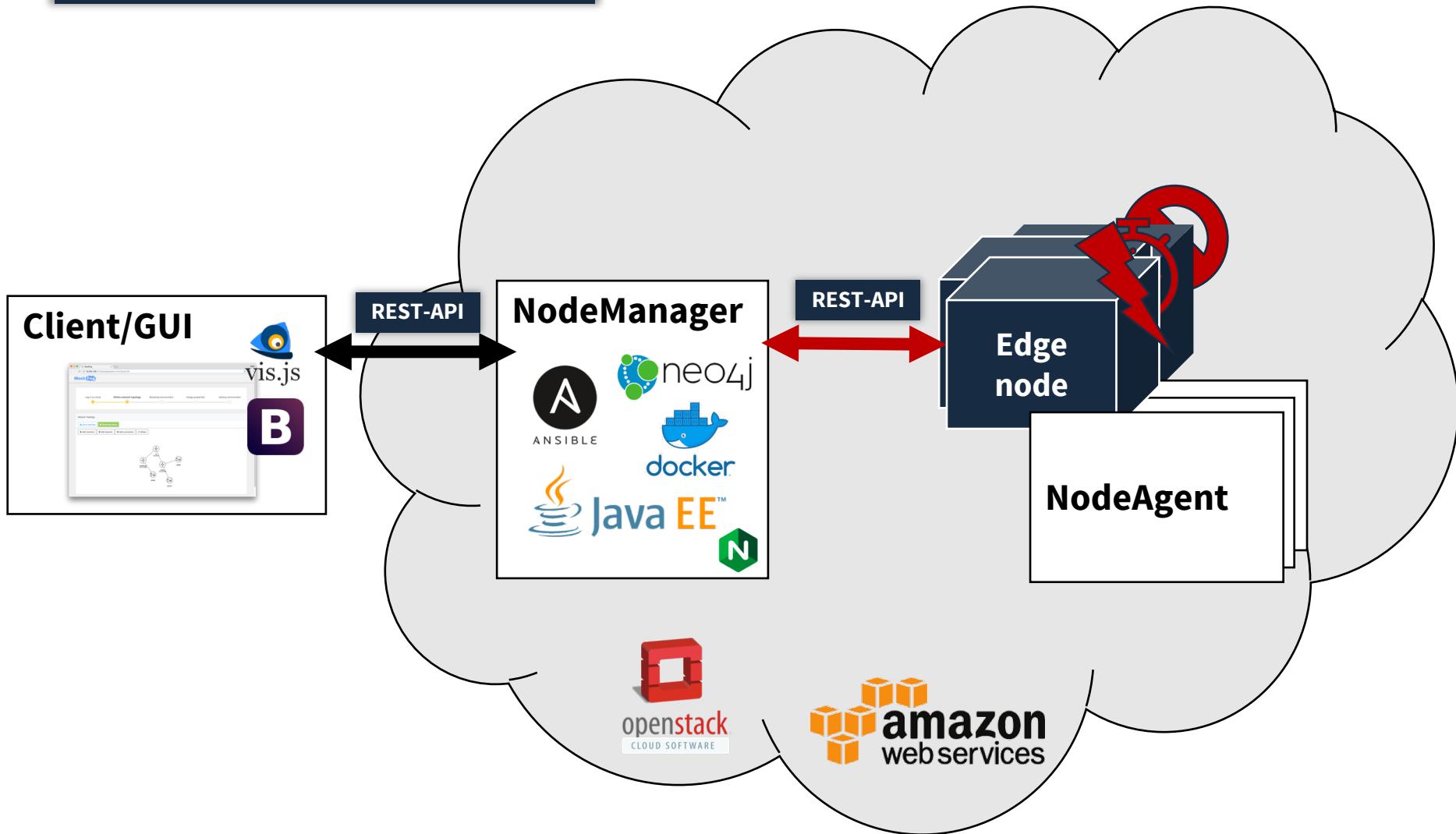
Technologies



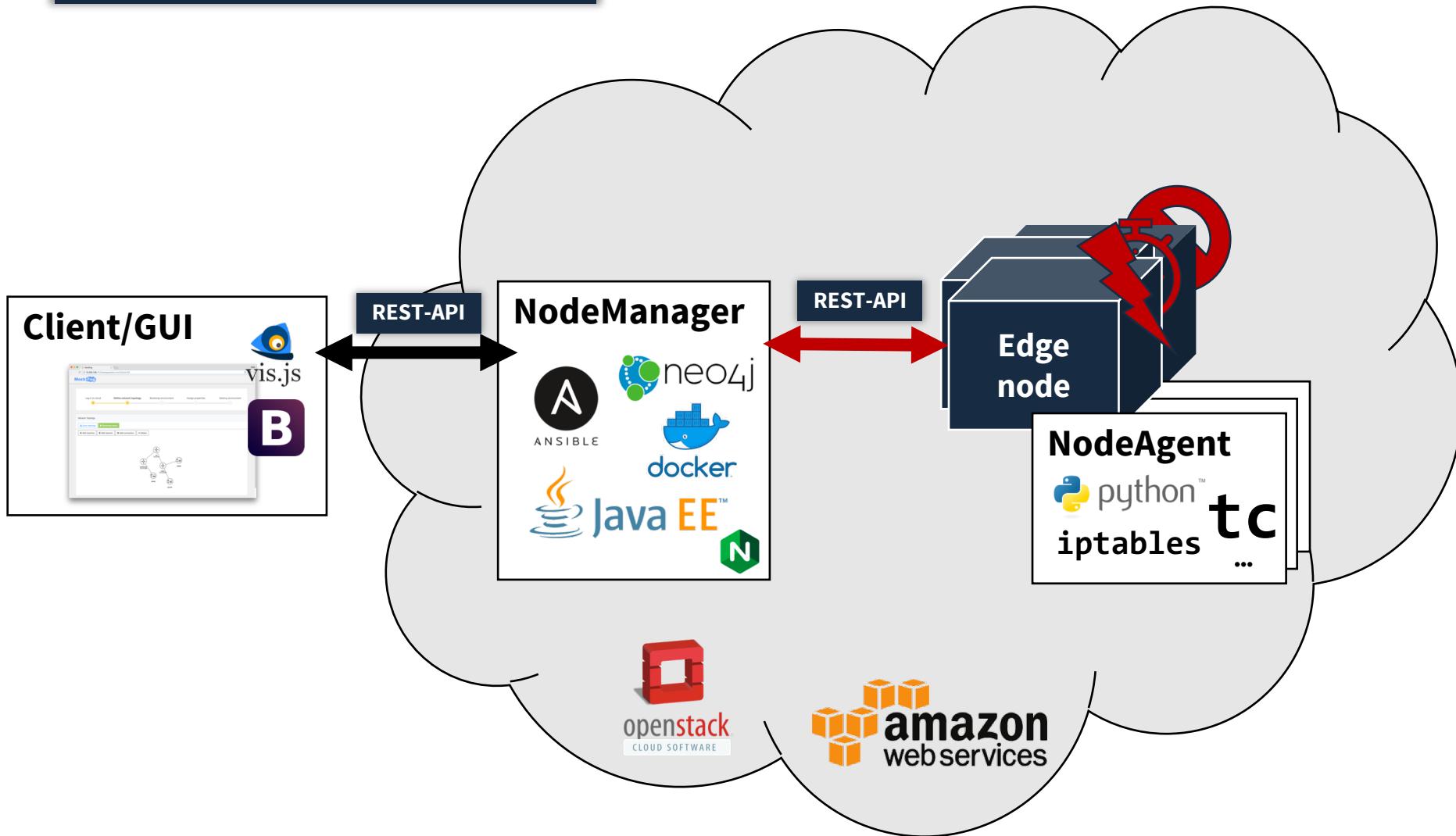
Technologies



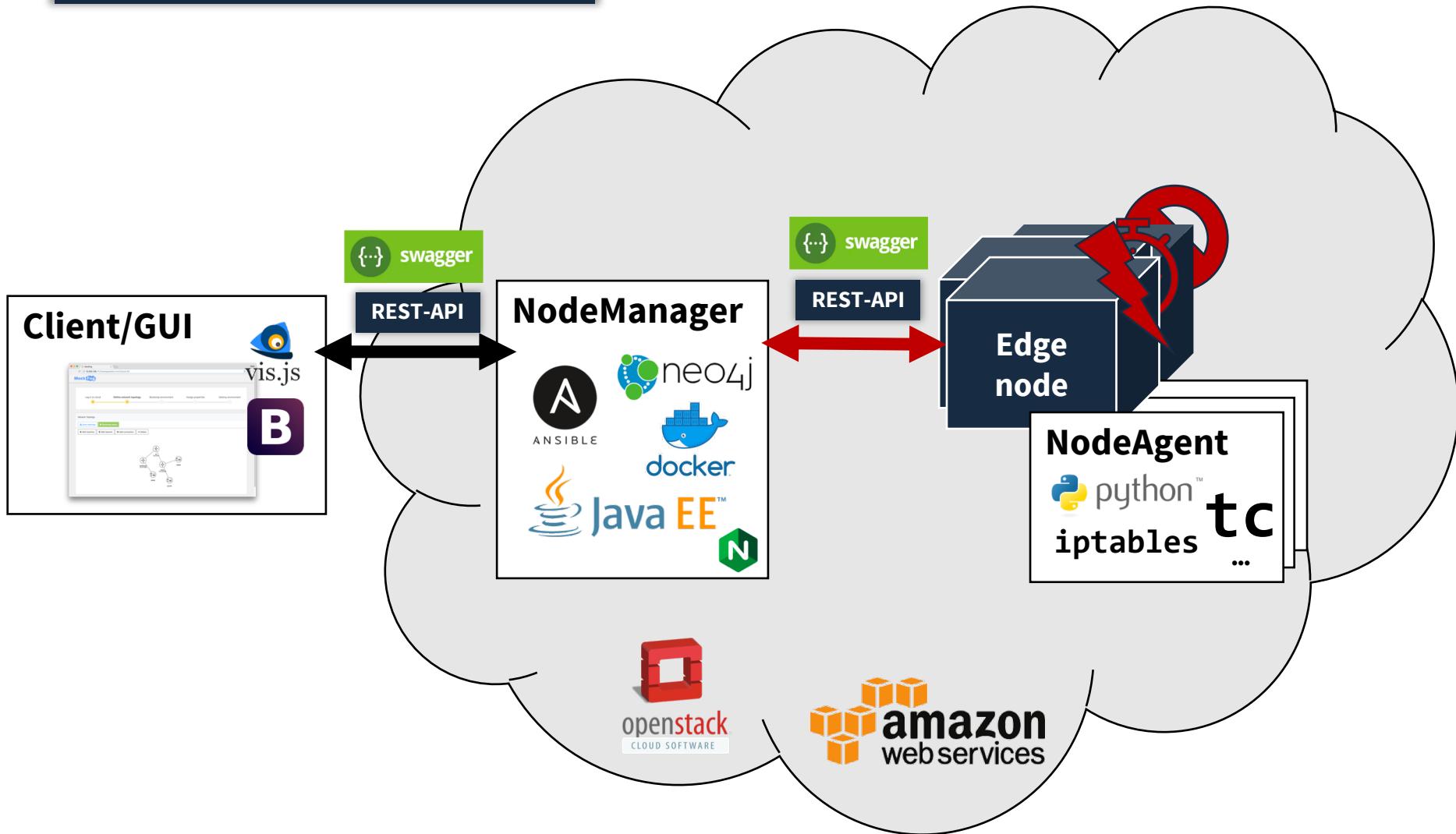
Technologies



Technologies



Technologies



Strengths

Abstraction of complex network architectures

Separation of components Independent deployment

Quick bootstrap and destruction

Automation of network generation

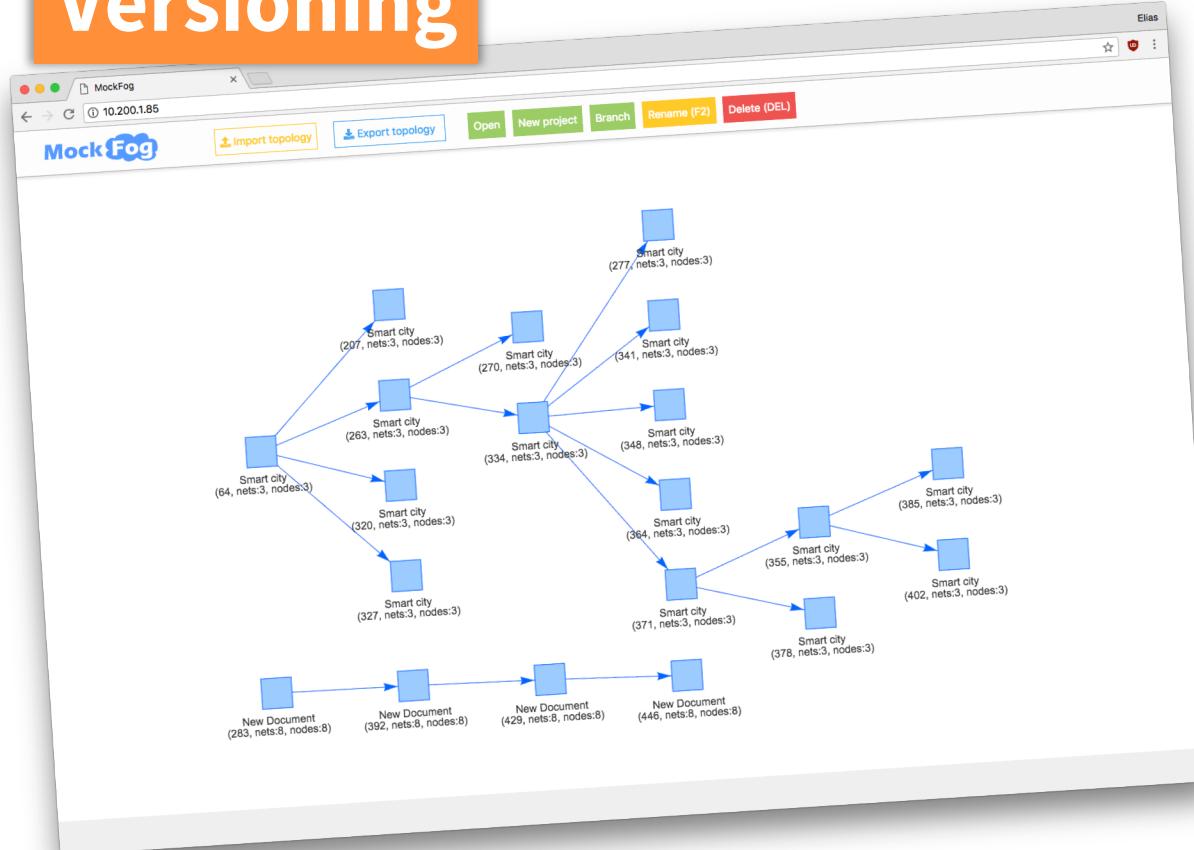
Automation of connection property creation

Dynamic injection of network anomalies

Easily extendable e.g. more properties

Strengths

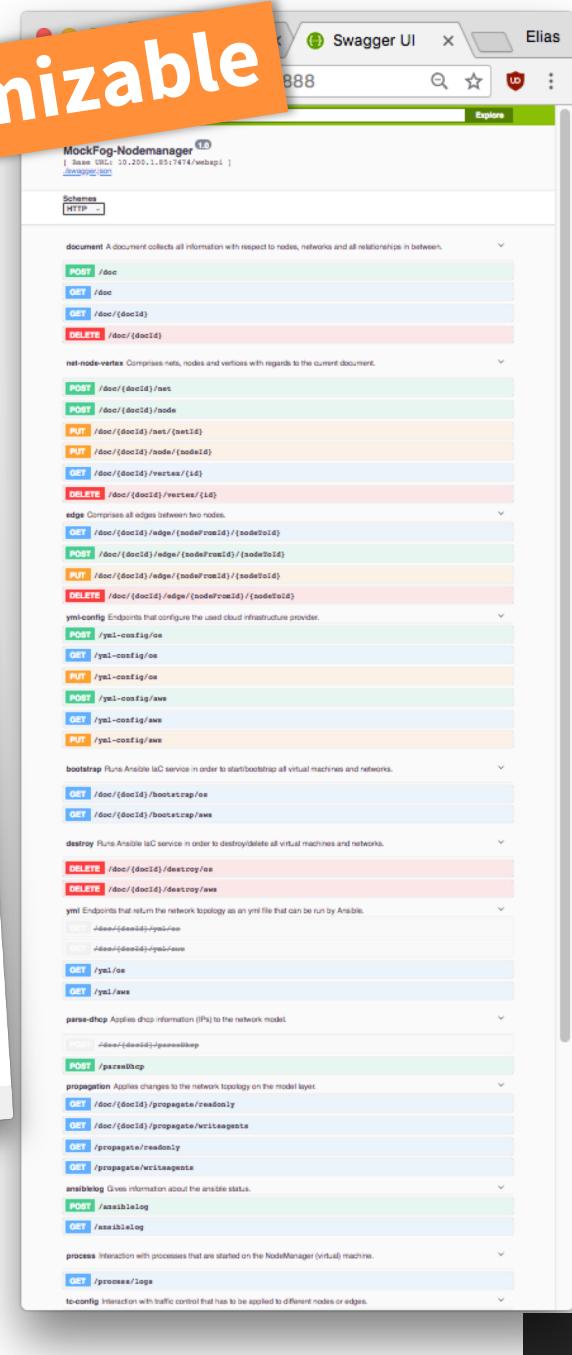
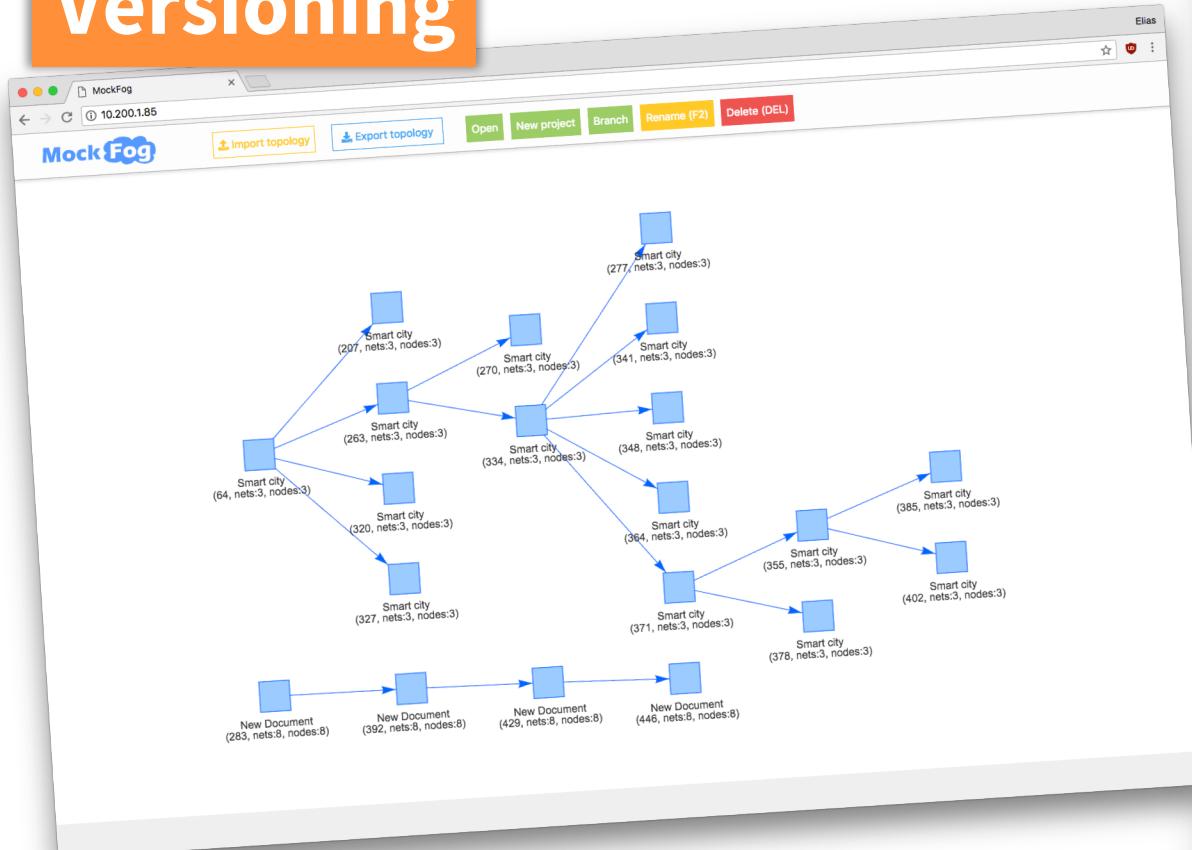
Versioning



Strengths

Customizable

Versioning



Outlook

Extend Agent properties

CPU/memory stress
hardware failures

More IaaS providers

Google Cloud Platform

Other operating systems

centOS

Microsoft Azure

Other CPU architectures

ARM

Conclusion

MockFog facilitates Fog application development by reducing complexity of distributed system setups by the ability to easily emulate a realistic behavior.

Ease of use

Well-defined interfaces

Complex service composition

Abstraction

Automation

Extendability



GitHub

github.com/MockFog

Thank you!

Q&A

MockFog

Emulating a realistic Fog Computing environment
on virtualized infrastructure.

Sören Becker

Miro Conzelmann

Elias Grünewald

Sascha Huk

Michael Narodovitch

Meike Stoldt

Franz Tscharf

Addendum

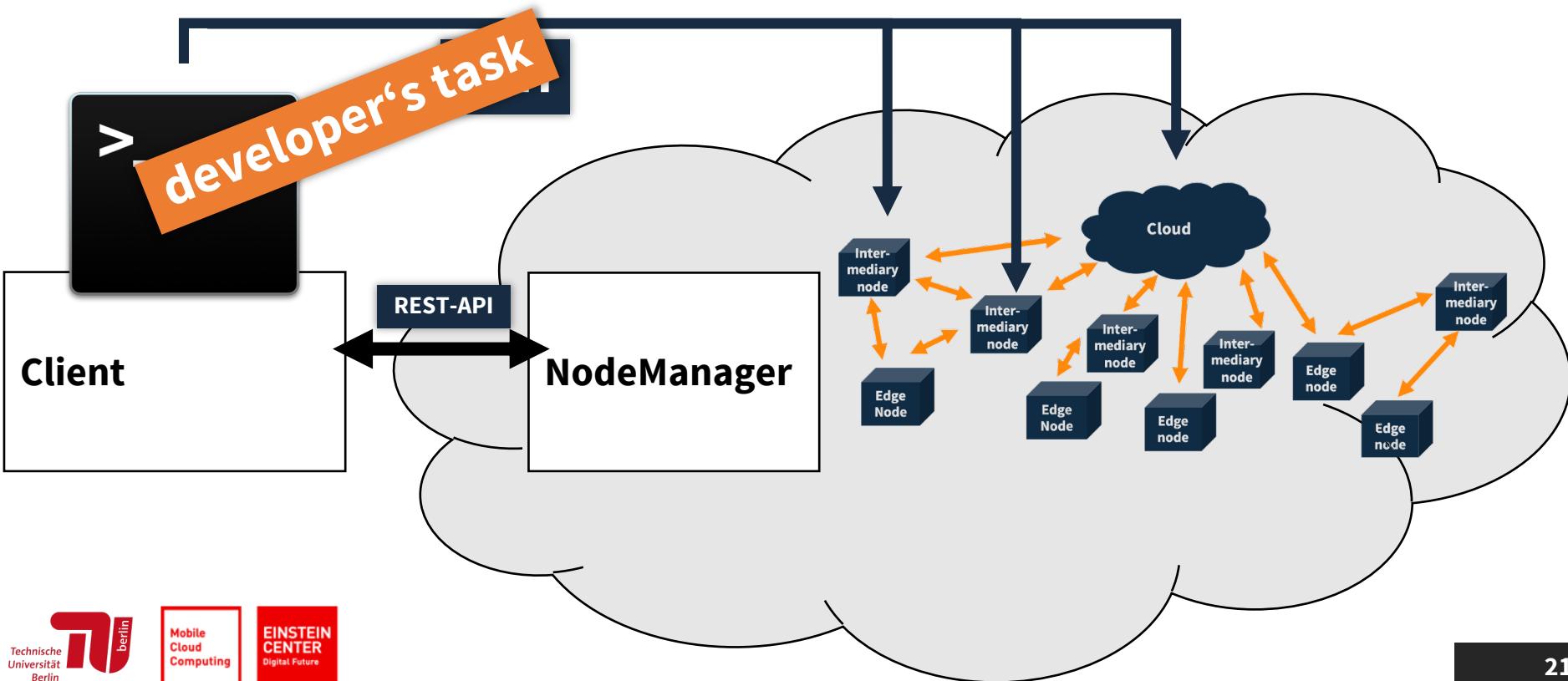
Application deployment

Stage #0: Start MockFog (NodeManager)

Stage #1: Define topology

Stage #2: Run environment

Application deployment



Assign properties

POST /assign/tc-config/doc/{docId}/node/{nodeId}

Live manipulation: set Tc-Configs of Node Agent by Id.

Parameters

Try it out

Name	Description
docId * required integer(\$int64) (path)	Identifier of current Document.
nodeId * required integer(\$int64) (path)	Identifier of node.
body * required (body)	List of key-value pairs of properties.

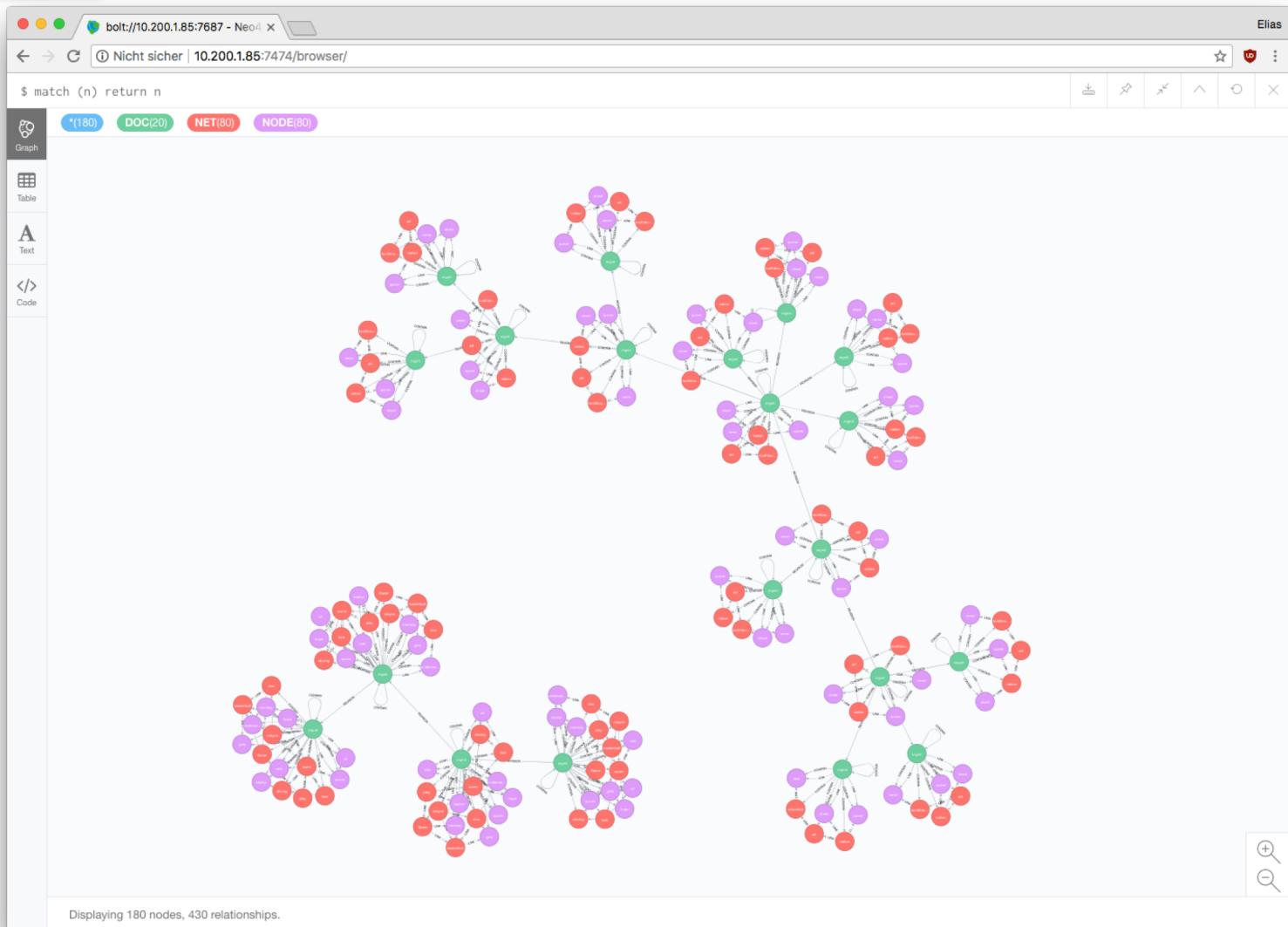
Example Value Model

```
{
  "out_rate": "1000000",
  "in_rate": "1000000",
  "rules": [
    {
      "dst_net": "string",
      "out_rate": "1000000",
      "delay": "10",
      "dispersion": "10",
      "loss": "0.05",
      "corrupt": "0.01",
      "duplicate": "0.05",
      "reorder": "0.1"
    }
  ]
}
```

Parameter content type

application/json

Neo4j



Postman

My Workspace ▾

http://10.200.2.38:7474

POST http://10.200.2.38:7474/webapi/doc/tree

Params Send Save

Cookies Code

History Collections

Open Government Data Service 3 requests

Postman Echo 37 requests

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "allDocs": [  
3     {  
4       "docName": "New Document",  
5       "name": "topology",  
6       "addr": "192.168.100.0/24",  
7       "id": 15,  
8       "allNets": {  
9         "16": {  
10           "name": "testnet5",  
11           "addr": "5.0.0.0/24",  
12           "edgesBack": {}  
13         }  
14       },  
15       "allNodes": {}  
16     },  
17   },  
18 }  
19 }
```

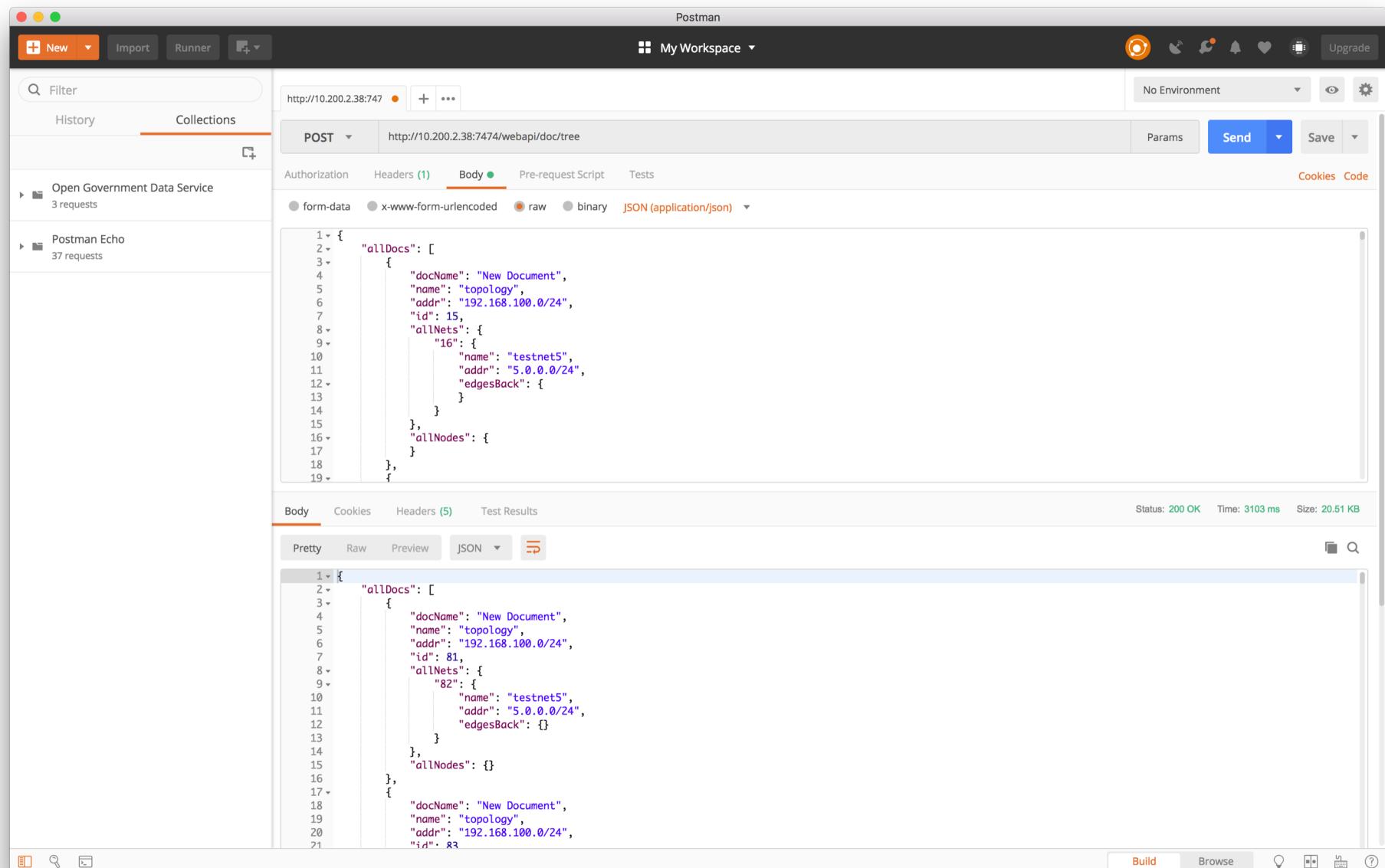
Status: 200 OK Time: 3103 ms Size: 20.51 KB

Body Cookies Headers (5) Test Results

Pretty Raw Preview JSON

```
1 {  
2   "allDocs": [  
3     {  
4       "docName": "New Document",  
5       "name": "topology",  
6       "addr": "192.168.100.0/24",  
7       "id": 81,  
8       "allNets": {  
9         "82": {  
10           "name": "testnet5",  
11           "addr": "5.0.0.0/24",  
12           "edgesBack": {}  
13         }  
14       },  
15       "allNodes": {}  
16     },  
17     {  
18       "docName": "New Document",  
19       "name": "topology",  
20       "addr": "192.168.100.0/24",  
21       "id": 83  
22     }  
23   }  
24 }
```

Build Browse



Login

MockFog

Log in to cloud

Define network topology

Bootstrap environment

Assign properties

Destroy environment

OpenStack



SSH key name:

mockfog

External network:

tu-internal

Authentication URL:

http://cloud.cit.tu-berlin.de:5000/v2.0

Project name:

MockFog

Username:

mockfog-user

Password:

→ Login/OpenStack

Add machine

MockFog

Log in to cloud

Destroy environment

Network Topology

Save topology Bootstrap setup

+ Add machine + Add network + A

Add machine

Configure your machine here.

Type:

- Raspberry Pi 2 Model B
- Raspberry Pi 3 Model B
- Raspberry Pi 3 Model B+**
- BeagleBone
- BeagleBone Black
- Banana Pi
- Banana Pi M3
- Arduino Tre
- Arduino Intel Galileo
- Nebbiolo fogNode Series

Close + Create

```
graph TD; art((art 0.0.0.0)) --> sheet((sheet)); art --> unlabeled(( ));
```

Add network

MockFog

Log in to cloud Destroy environment

Network Topology

Save topology Bootstrap setup

+ Add machine + Add network + Add connection - Delete

Add network

Configure your network here. For example:

A Virtual Pseudo-Network (IP/Mask default to 0.0.0.0/0) is only available for easily structuring complex network topologies.

IP/Mask:

10.0.2.0/24

Close Create

```
graph LR; art((art 0.0.0.0/0)) --- host(( ));
```

Assign property

The screenshot shows the MockFog web-based configuration tool. The top navigation bar has three tabs: 'MockFog' (active), 'MockFog', and 'MockFog'. The URL in the address bar is `10.200.1.85:7474/webapi/editor.html?docId=64`. The title bar says 'Mock Fog'. On the right, there's a user name 'Elias'.

The main interface is divided into several sections:

- A horizontal progress bar at the top with five steps: 'Log in to cloud', 'Define network topology' (highlighted in yellow), 'Bootstrap environment', 'Assign properties', and 'Destroy environment'.
- A 'Network Topology' section below the progress bar. It includes buttons for 'Save topology' (blue), 'Bootstrap setup' (green), and actions: '+ Add machine', '+ Add network', '+ Add connection', and '- Delete'.
- A network diagram showing two nodes connected by a line. One node is labeled 'art 0.0.0.0' and the other is 'toothbrush 10.0.4.0/24'. Arrows indicate bidirectional connections between them.
- A modal dialog box titled 'Edge' for the connection between the two nodes. It contains the following settings:

Edge	Done
Delay:	10 ms
In Rate:	100000 kbps
Out Rate:	100000 kbps
Dispersion:	5 %
Package lost:	0 %
Corrupt:	0 %
Reorder:	0 %
Duplicate:	0 %

At the bottom of the dialog is a 'Delete' button.

Recap: Main stages

Stage #0: Start MockFog (NodeManager)

Stage #1: Define topology

Stage #2: Run environment assign properties dynamically

Stage #3: Destroy environment



```
3. elias@wlan-141-23-185-252: ~/Desktop/MockFog-IaC/created (zsh)
elias@wlan-141-23-185-252 ~/Desktop/MockFog-IaC develop ? cd created
ls
elias@wlan-141-23-185-252 ~/Desktop/MockFog-IaC/created develop ? ls
mockfog  mockfog.pub
elias@wlan-141-23-185-252 ~/Desktop/MockFog-IaC/created develop ? cat mockfog
-----BEGIN RSA PRIVATE KEY-----
MIIJKAIAKCAgEAqDG81CCU1D38qnGL20VsAFXwz0kfAIcvfyjkxpTr8sN+Hp
prfYPzfXlXGXf6KzMC7cli07Xsd2ooWe79JreMz5HG4NEaj7LdQ1Q3RShw+KSpCW
CyBlqnv358Ma4UDpCUMYNzft1Xdz00fCsZVyardb0ACtCondwiS9rgAcLWE+b
+cnw3JdLpgMNNGy2SQGLsdWdrCmlP9PKTr5p4V1LkyPA/srz3B7+kv8tGb0vBVI
Ue+DFKyZJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJFkYzJ
bJp6nd3u0wIIhwgc0w+2EmWbYWIq+HbQ+s4Xt28nc5D3VwUk4J/XOAWM8TiLxn
UmyhvZv6eSrdWbNk8EduLr8Qw6VELEGUnKuf7vs304wdAiPVPK6IA77Kxdib
1EgtzLzrJEoGWi+NAVjjAOJNjThfq+t+sUFTFLhlRRkiRp3QGdJrau5wgcJYjdjx
Qnky2TOeKfUBCm5ZSOClAucfLRwN1+1xo+G00iUX0VRIx45iBcua2awAQYa7tcyx
L6kmbH33o2t7sMpbHmVN2FiQgycrSAa8ATYy8Bwmz11lwu2DSxTD60Hk6Ty/8Z0c
jV7eGeGctM0F8jK771RPxM4benbTdnMRK1bmZbfQ6sSfjlmuEulkAST2qUCAwEA
AQKCAGBv0LUwx0opAcUxxtBDVNRezMSJLMoP6BSAJ67J4FZT6UEIw7GMu0+Zi+zZ
u4EAu0vctWDT0VY5yKI/4DW3S20fQw4KqwyheN4U9J00upxJDRhbNb-EK0yszeCn
t/Lyu76Ts+NMyy153E6XYDzxVt8vLzd8aTy56jtYBoG0v1NMG1x5TkxL6CwM9VD1
vGbjIHf73ka8iSllddWliz2TrGiE2kM1HHCIIfazf2X9T9HyzCw6JqV3k8FjS0amYv
yV/nFP0GJ002iSYqDeT7D+c8Yuwml3a1UhUPc6u7EjHYDQ0wA86K/3tzfsVP2RCz
IkkZY1eoBz/U8rRTKe6WGIq0Y7oFKFhMDwp6duFkr16dv5a2p3Cxtt2Ca6J60kBE
lk+Qt1m7F0yTrznPjI3WwIrr0Pz47qaTnuh0e51yBSIDZviSK6oQ5jiJLz+qfo09I
diyPjloWBjyH8bolalo6JNpkWucSusRYYES1tAEr78C0dSNFgw+B4ru9suuNyDk
hxJ50X3/43grdAdFFbD9ugNiph2LDp0CFHVCJfLjhMQRD/zERFPGH05HHGApo31S
g50ff0SPwwJv/djLC+E1AjjIYev/Ppzckm8Vpqwdgj7SNu136Fy419WVpKr+KK4e
yT/sSgbs0W3eFbfIwrwyCX1voDy3gvE3Liivxe2K/U3Y0RjDEQKCAQEAE3gyISSt
3YiRCX1tizBqLq3EFk6mrKzUHJevJF5T2Q0L7mYCBbqulE9Vky8Dt5QGNvZd
ydF02xAiuHcu09SvMBK0czEl9rjDcDYHgkXTAGbcm5aceFzCZIGhainnCGaTv7h1
svm/NvAwpwWcTHFoeC40s06ePyINK5inrz7JAyt81/kHOUDIB/KgPJbdHJfac5xI
IV1se7LMqHBa5IFFXapqwFSArXd0YsijJzxb4FpVzTv+TPTENo/r3dzVgrB9KLGx
D9/LgV0uEH2V41hkiLnX8v+01remJxzuUM66x00+LNP6n30Z1/Prl3jBnMq1/r6t
nmMgkj9lc2rRhWKAQEwek4H9iEWaAnZj2f1Ewda0/S1JiQtiifbEf0kMzyFjZ
g8eF6T1cKzT98Dnzrb8bX4vvxBvjCBfaFxHfJxj0+CZ5+PTxkIZUUPyJIGHI9fIG
0ddJVRdYuumTsFmbwA6CVin0mY2jppveKCRJqaDl0TUg0mrXXTKNpxRebgmMLsk
N/nEUGrri3y09/08e6fxD/jjQMwea0NHNOauqlelQsv5MgiJmu71JuhEJfRBYzt
5rtFfZv/YM2tFeY3dXdDPmvC+3qaf21F4132mJsZeb8PCSVkbX7ulW4A1H0afnYD
rP3dzlTlHV6hJXVo8c587jQyU+f3RK7XEGXkEeUtwKCAQEao5p60+cJ6W75MrRm
pNpcMd50DmAltaTyIce7B46tlULRRqEdVd6k+NGuVaoKdJ8u6L93TsyqU0BnP7D
```

this one won't work anymore...

Complex environments

Network Topology

[Save topology](#)

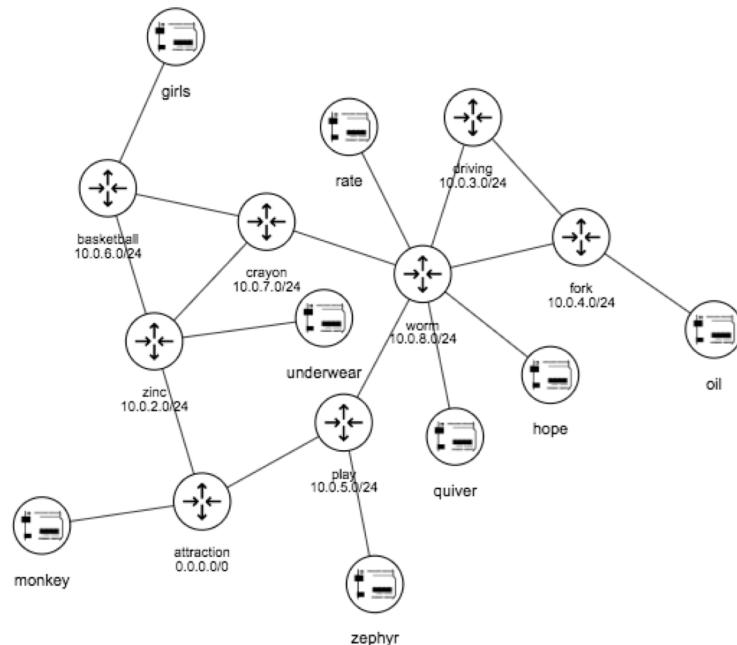
[Bootstrap setup](#)

+ Add machine

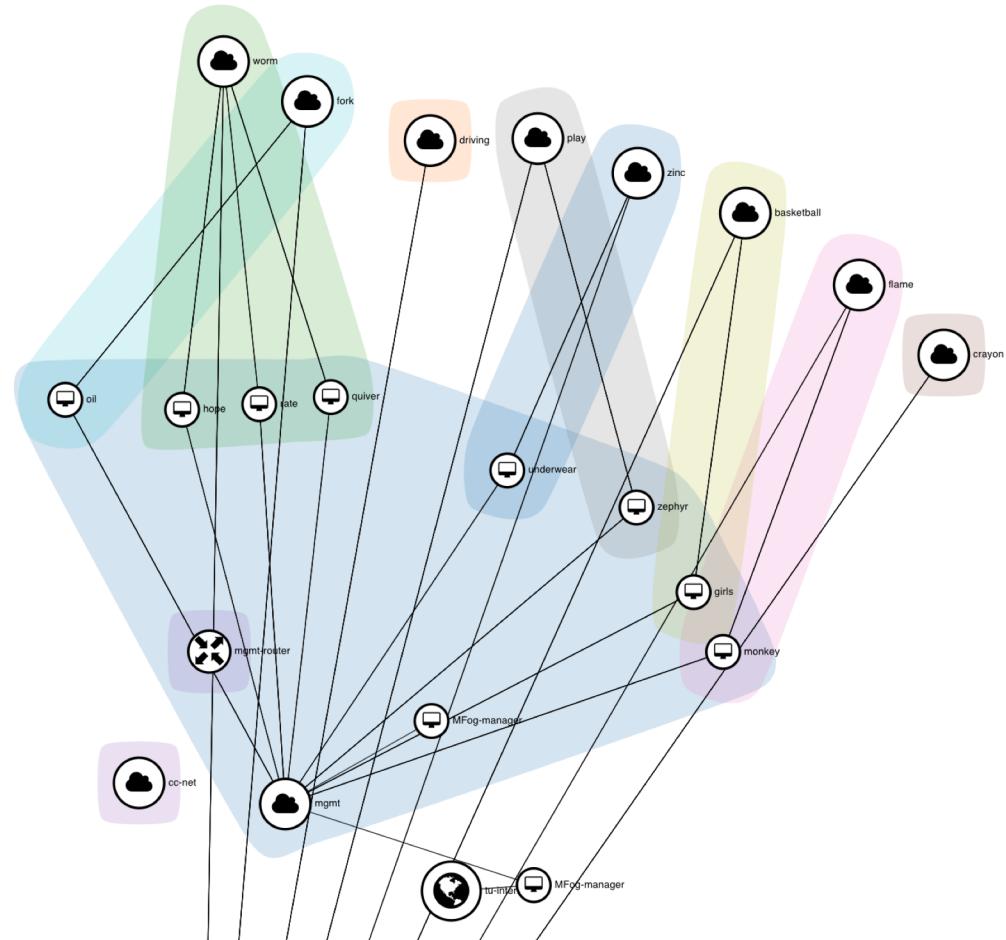
+ Add network

+ Add connection

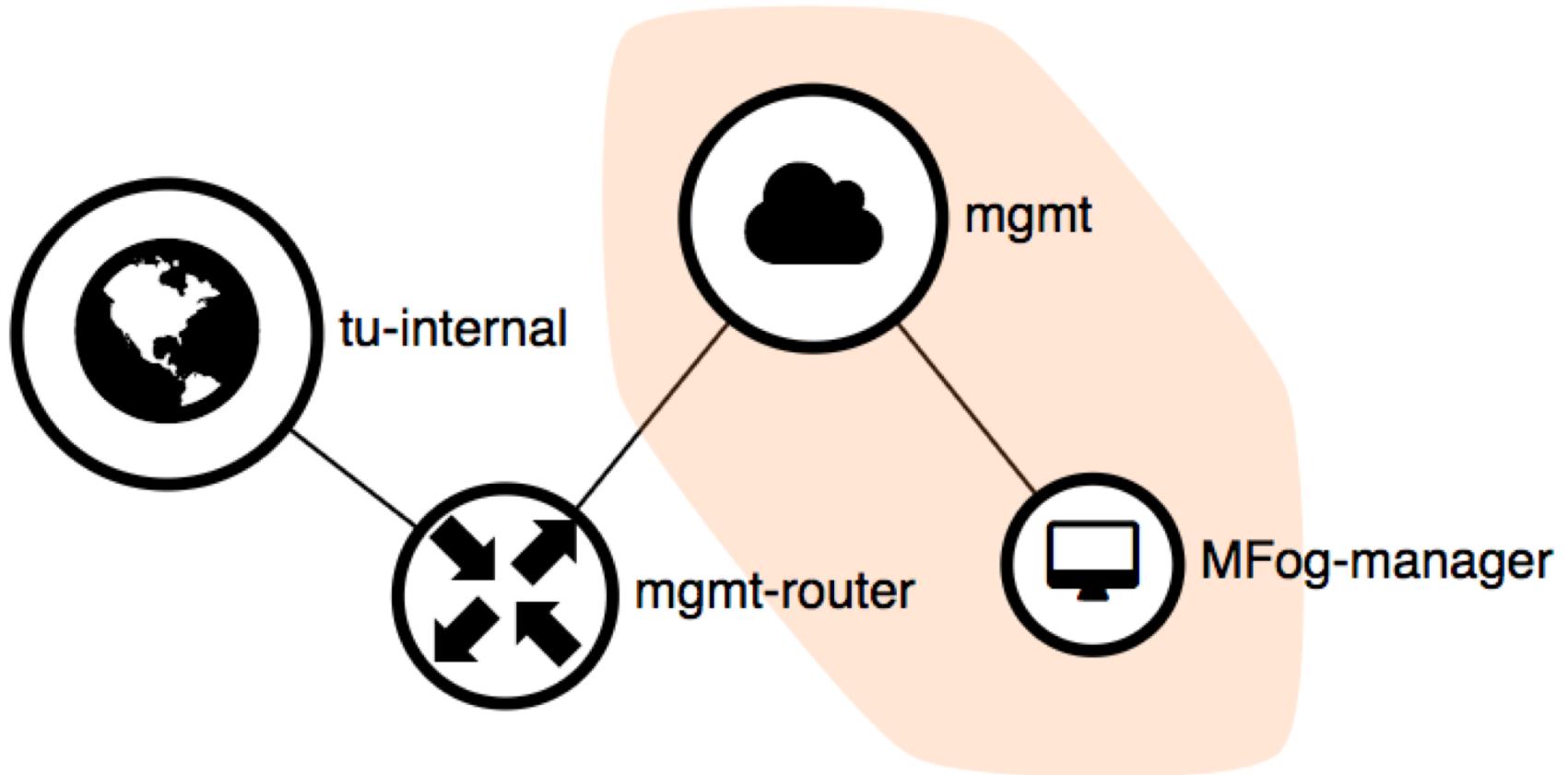
- Delete

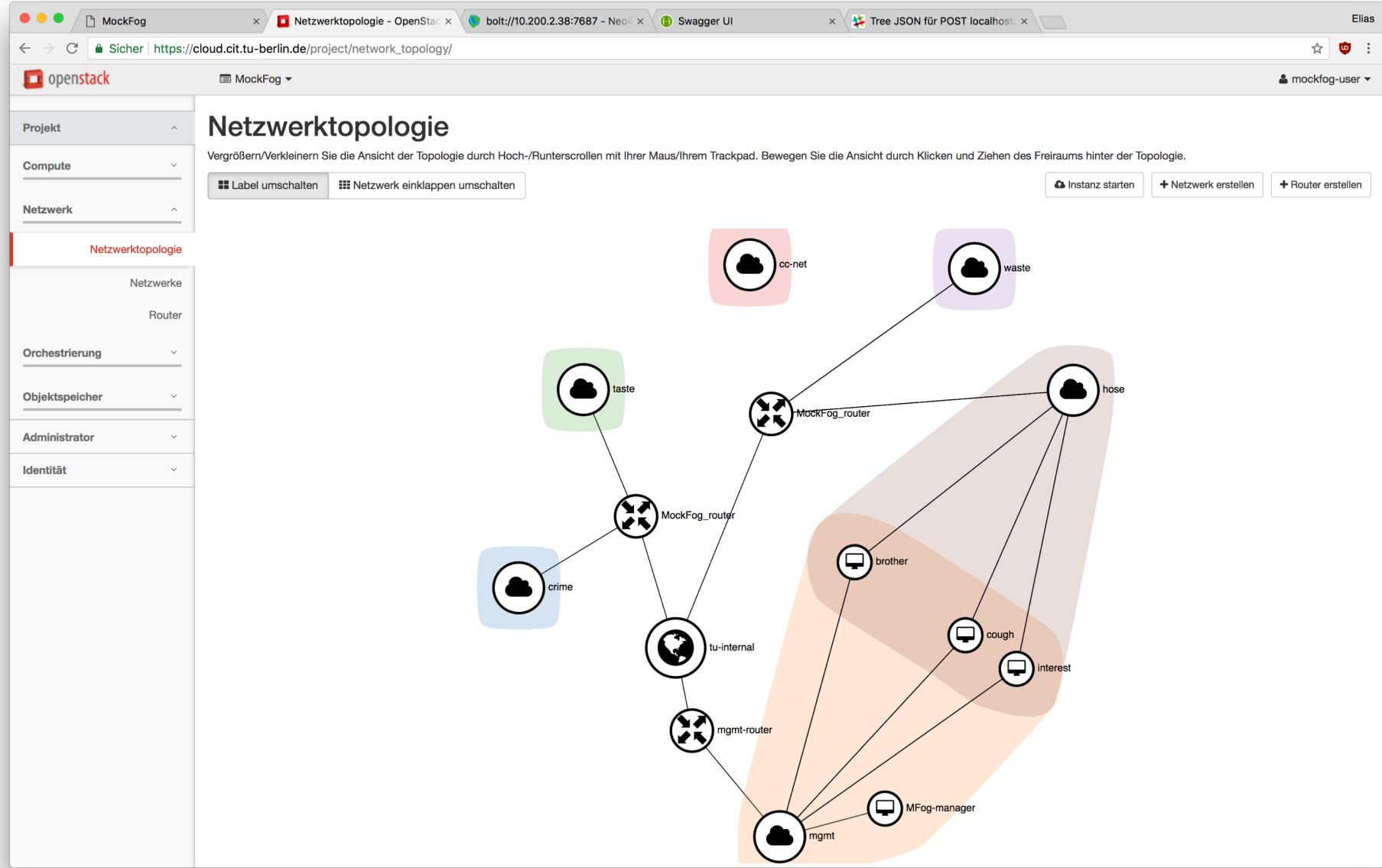


Complex environments



Management interface





Projekt
Compute
Übersicht
Instanzen
Datenträger
Abbilder
Zugriff & Sicherheit
Netzwerk
Orchestrierung
Objektspeicher
Administrator
Identität

Übersicht

Limit Summary



Instanzen

Used 1 of 12



vCPUs

Used 2 of 12



RAM

Used 4.096 of 12.288



Floating IPs

Used 5 of 12



Sicherheitsgruppen

Used 1 of 10



Datenträger

Used 0 of 0



Datenträger-Speicher

Used 0 of 0

Usage Summary

Select a period of time to query its usage:

From: 2018-07-01

To: 2018-07-04

Submit

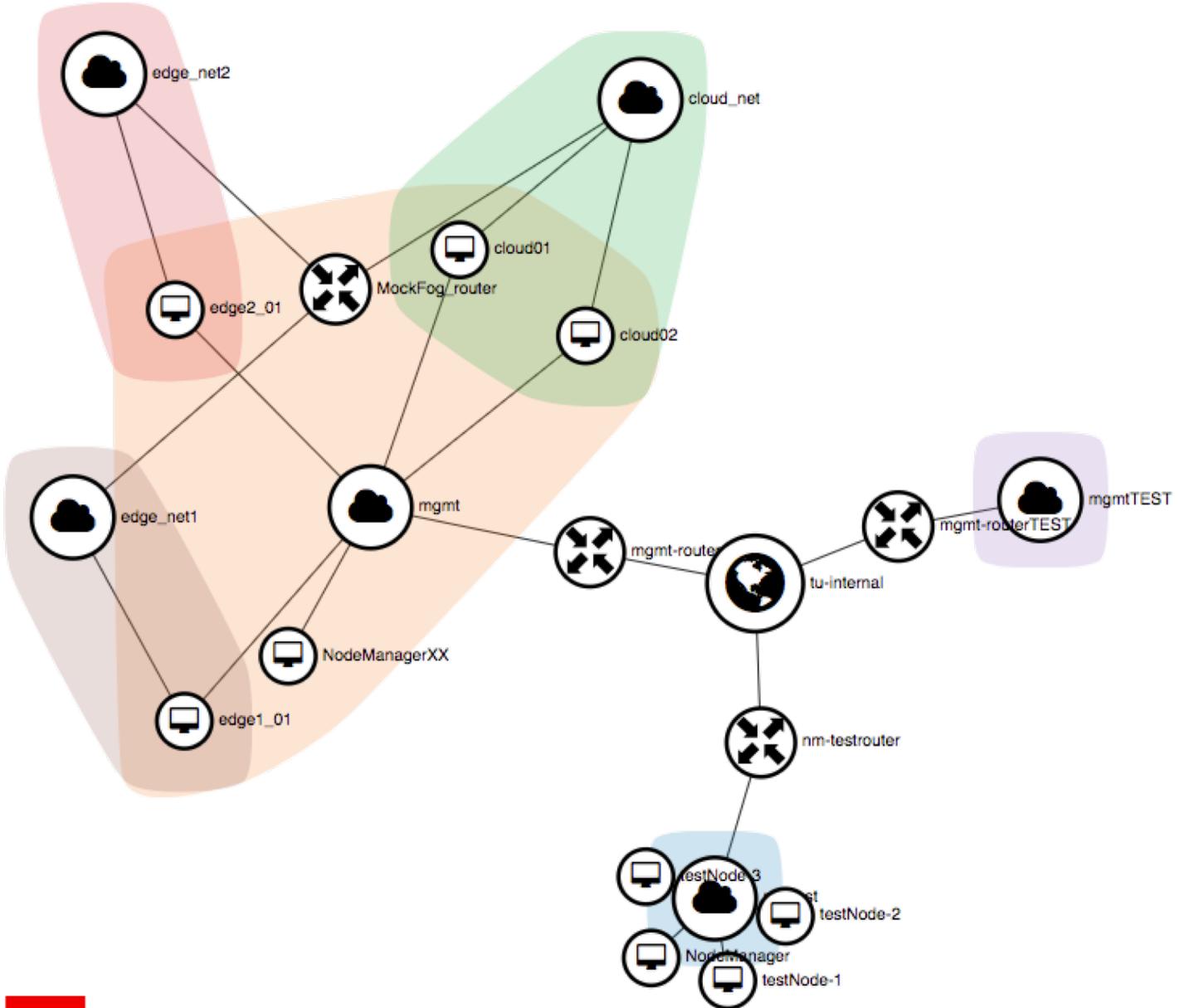
The date should be in YYYY-mm-dd format.

Aktive Instanzen: 1 Active RAM: 4GB This Period's VCPU-Hours: 359,91 This Period's GB-Hours: 6817,16 This Period's RAM-Hours: 531579,98

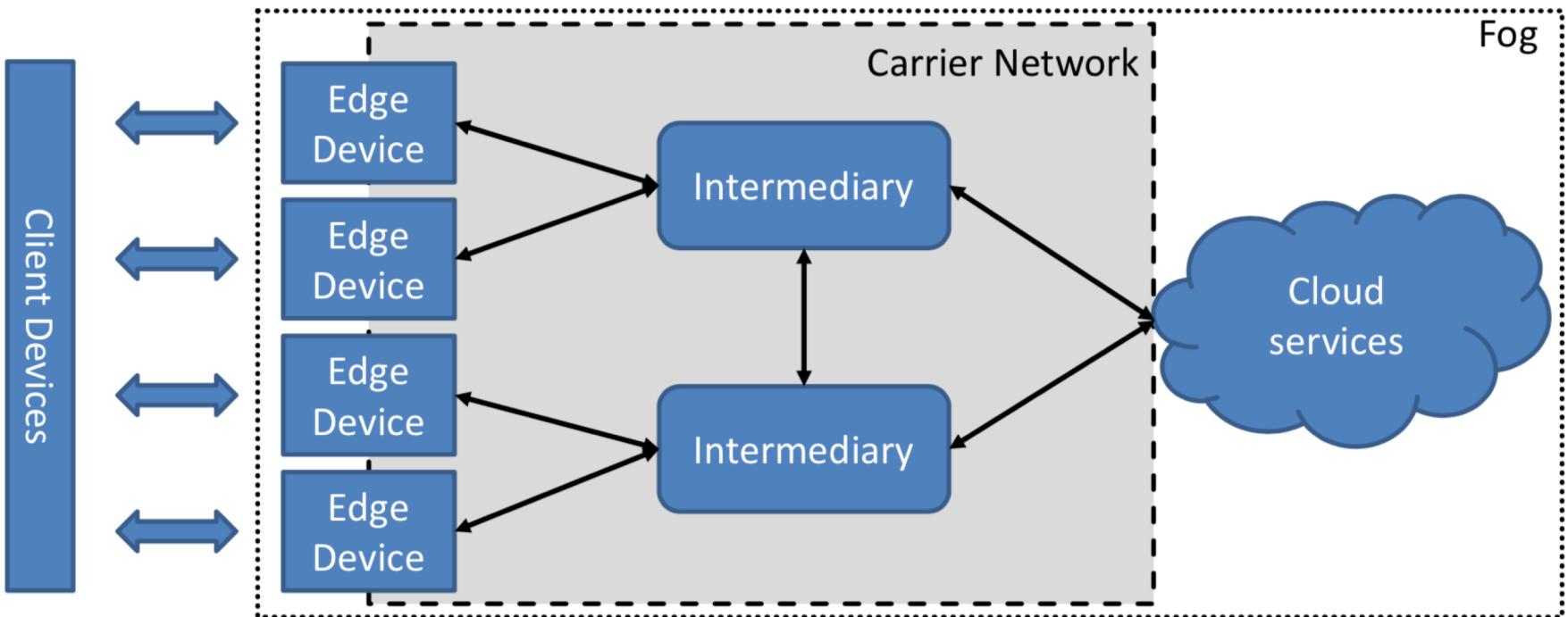
Verwendung

Instanzname	VCPUs	Festplatte	RAM	Zeit seit Erzeugung
MFog-manager	2	10GB	4GB	15 Stunden, 12 Minuten

Displaying 1 item

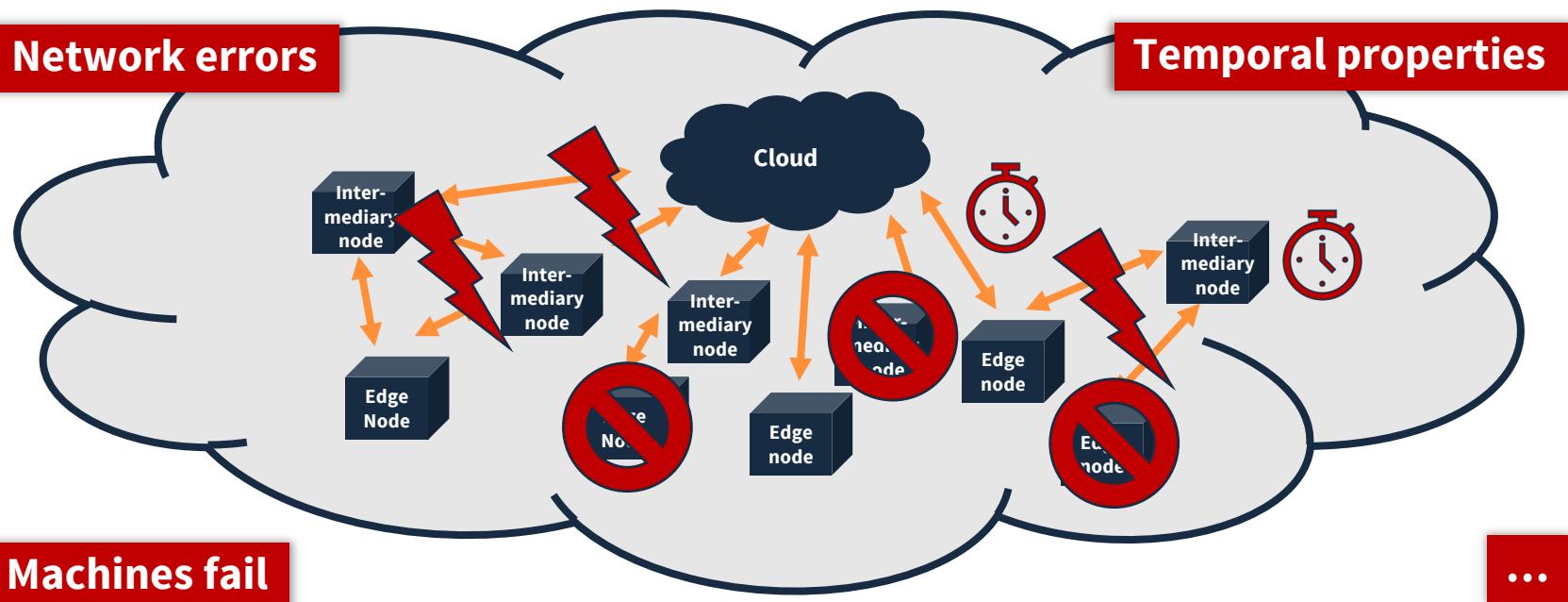


Terminology



Bermbach, David, Frank Pallas, David Pérez, Pierluigi Plebani, Maya Anderson, Ronen I. Kat and Stefan Tai. 2017. "A Research Perspective on Fog Computing."

Constraints



Different types of nodes in terms of storage, computing, network

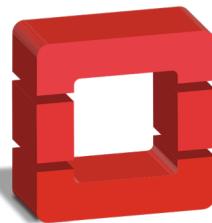
Nodes are in different networks emulating realistic environment

Properties are constantly changing latencies, failures, losses...

Infrastructure as Code



ANSIBLE



openstack®
CLOUD SOFTWARE

62 lines (58 sloc) | 1.28 KB

```
1  ---
2
3  networks:
4    - name: "{{ mgmt_network_name }}"
5      subnet_name: mgmt_sub
6      subnet: 192.168.100.0/24
7    - name: cloud_net
8      subnet_name: cloud_net_sub
9      subnet: 192.168.10.0/24
10   - name: edge_net1
11     subnet_name: edge_net1_sub
12     subnet: 10.0.0.0/24
13   - name: edge_net2
14     subnet_name: edge_net2_sub
15     subnet: 10.11.11.0/24
16
17 routers:
18   - name: mgmt-router
19     interfaces:
20       - net: mgmt
21         subnet: mgmt_sub
22         portip: 192.168.100.1
```

MockFog Nodemanager Service 1.0

[Base URL: localhost:7474/webapi]

[/swagger.json](#)

Schemes

HTTP

default

POST /doc

GET /doc

GET /doc/{docId}

POST /doc/{docId}/net

NodeManager

PUT /doc/{docId}/node/{nodeId}

GET /doc/{docId}/vertex/{id}

DELETE /doc/{docId}/vertex/{id}

POST /doc/{docId}/edge/{nodeFromId}/{nodeToId}

PUT /doc/{docId}/edge/{nodeFromId}/{nodeToId}

DELETE /doc/{docId}/edge/{nodeFromId}/{nodeToId}

GET /doc/{docId}/yml

MockFog Agent API 0.1.0

[Base URL: /api]

[http://10.0.2.15:8889/api/swagger.json](#)

default Default namespace

properties Properties on the host

GET /properties/ Returns all started properties

tc-config The local TC configuration.

DELETE /tc-config/ TC shut down

PUT /tc-config/ Update the TC configuration

POST /tc-config/ Start TC

GET /tc-config/ Retrieve the current TC configuration

ping Run ping tests

POST /ping

GET /ping Returns dict of latency

GET /ping/{hostname}/

NodeAgent

Models

```
tc-parameters <-
  out_rate    string
              example: 100mbps
              Output bandwidth
  in_rate     string
              example: 100mbps
              Input bandwidth
  rules       <-
    <rule> <-
      dst_net   string
                  example: 1.1.1.1/24
                  Destination subnet or IP
      delay     string
                  example: 10ms
                  Additional latency to dst_net
      dispersion string
                  example: 10ms
                  Additional latency dispersion to dst_net
      loss      string
                  example: 0.1
                  Additional packet loss to dst_net
      corrupt   string
                  example: 0.1
                  Additional packet corruption to dst_net
      duplicate string
                  example: 0.1
                  Additional packet duplication to dst_net
      reorder   string
                  example: 0.1
                  Additional packet reordering to dst_net
```

(both shortened)



swagger

Agile development

Sören
Miro
Elias
Sascha
Michael
Meike
Franz

