Enos: a Holistic Framework for Conducting Scientific Evaluations of OpenStack

Adrien Lebre

The DISCOVERY Initiative

Cloud Resource Management Systems have significantly evolved to become complex ecosystems

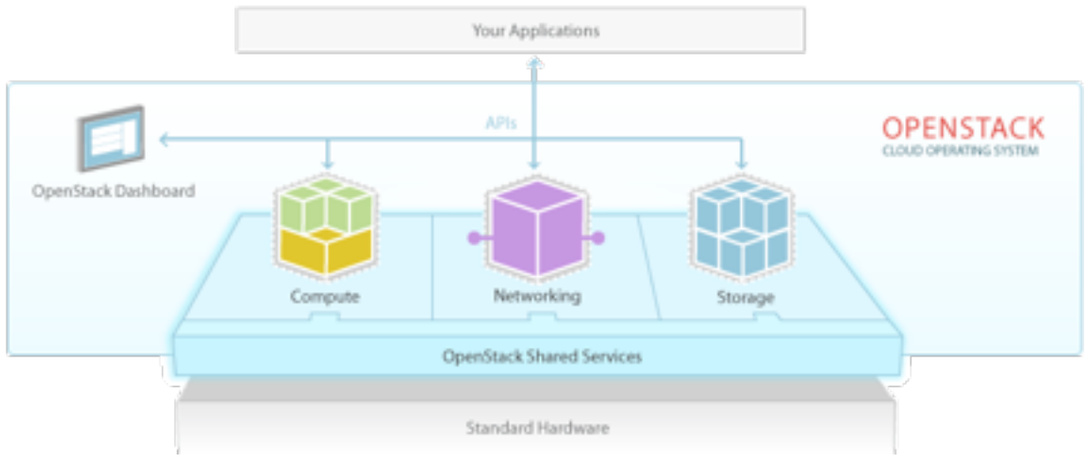
More and more features (bare-metals, VMs, containers… application life-cycle management, … *automation, control loops…*)

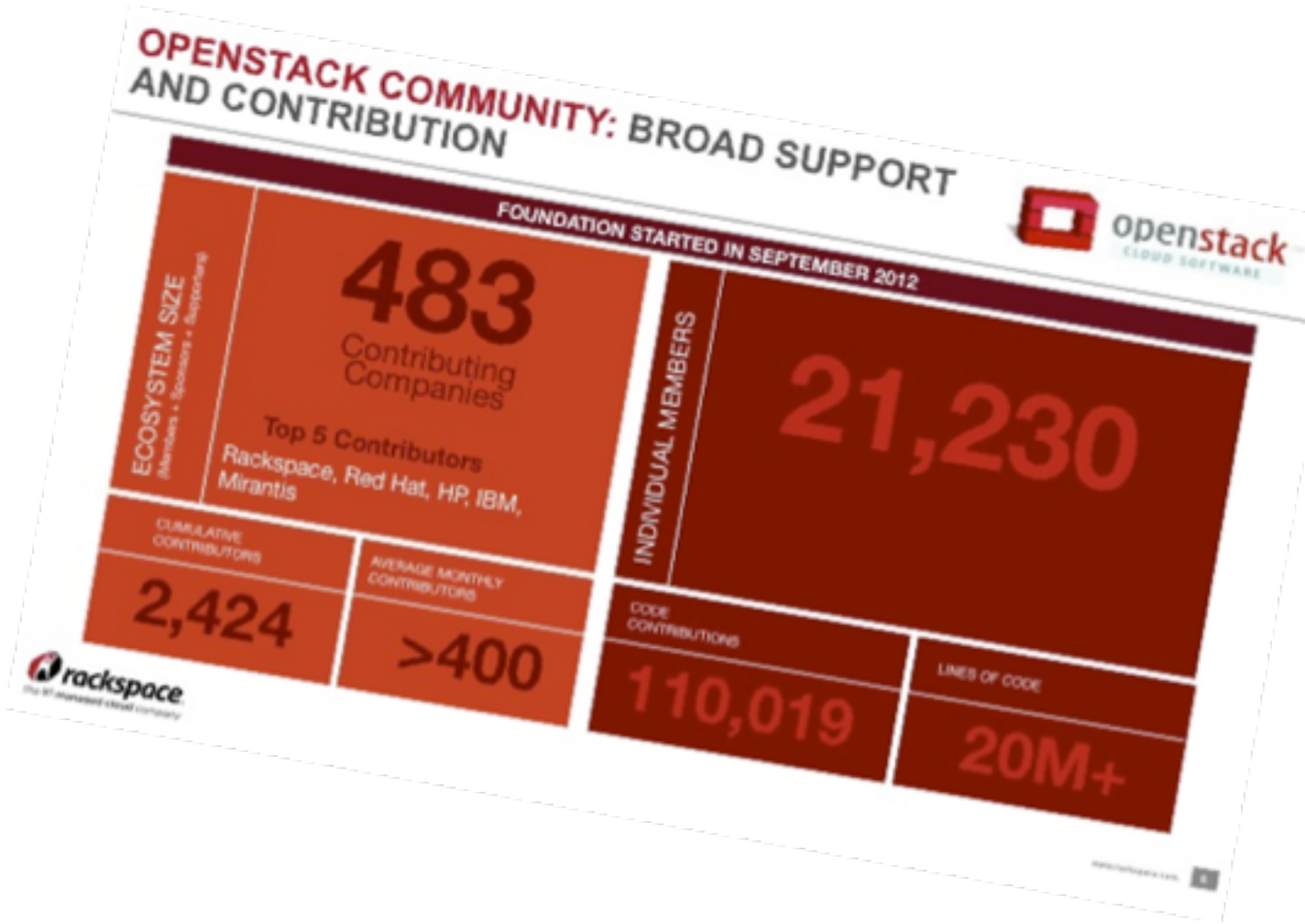
Research PoCs vs. Production Systems

A Huge Gap
that needs to be filled, but…

OpenStack

- Industry standard for creating public and private clouds

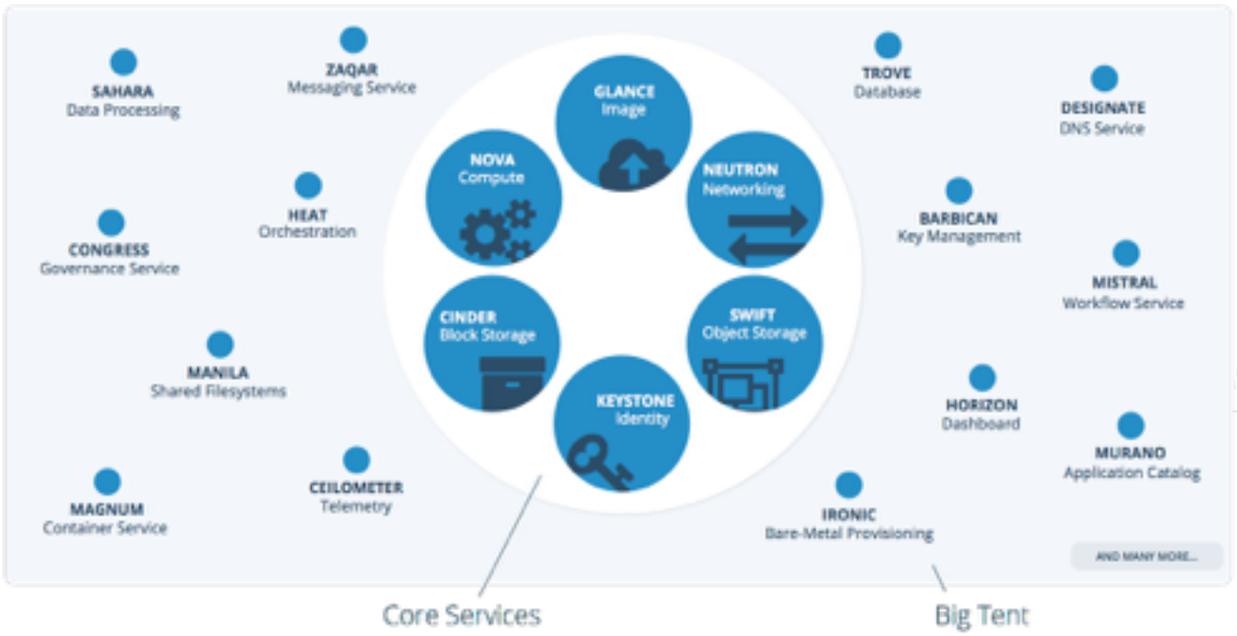


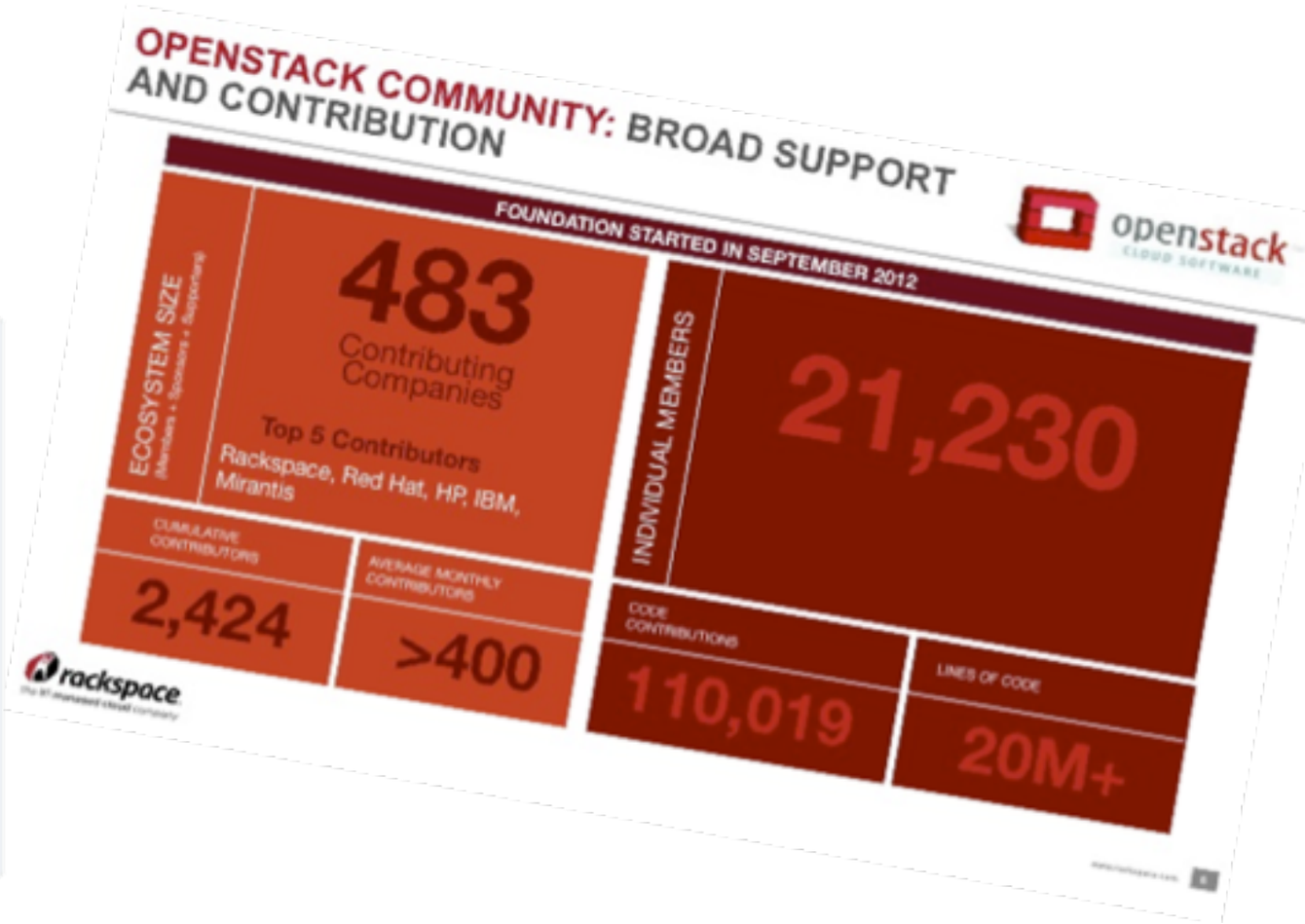


- A rich (and complex) ecosystem
  - **20 Millions of LoC**, 164 services, some services are composed of sub-services (e.g. nova-scheduler, nova-conductor, …)

OpenStack

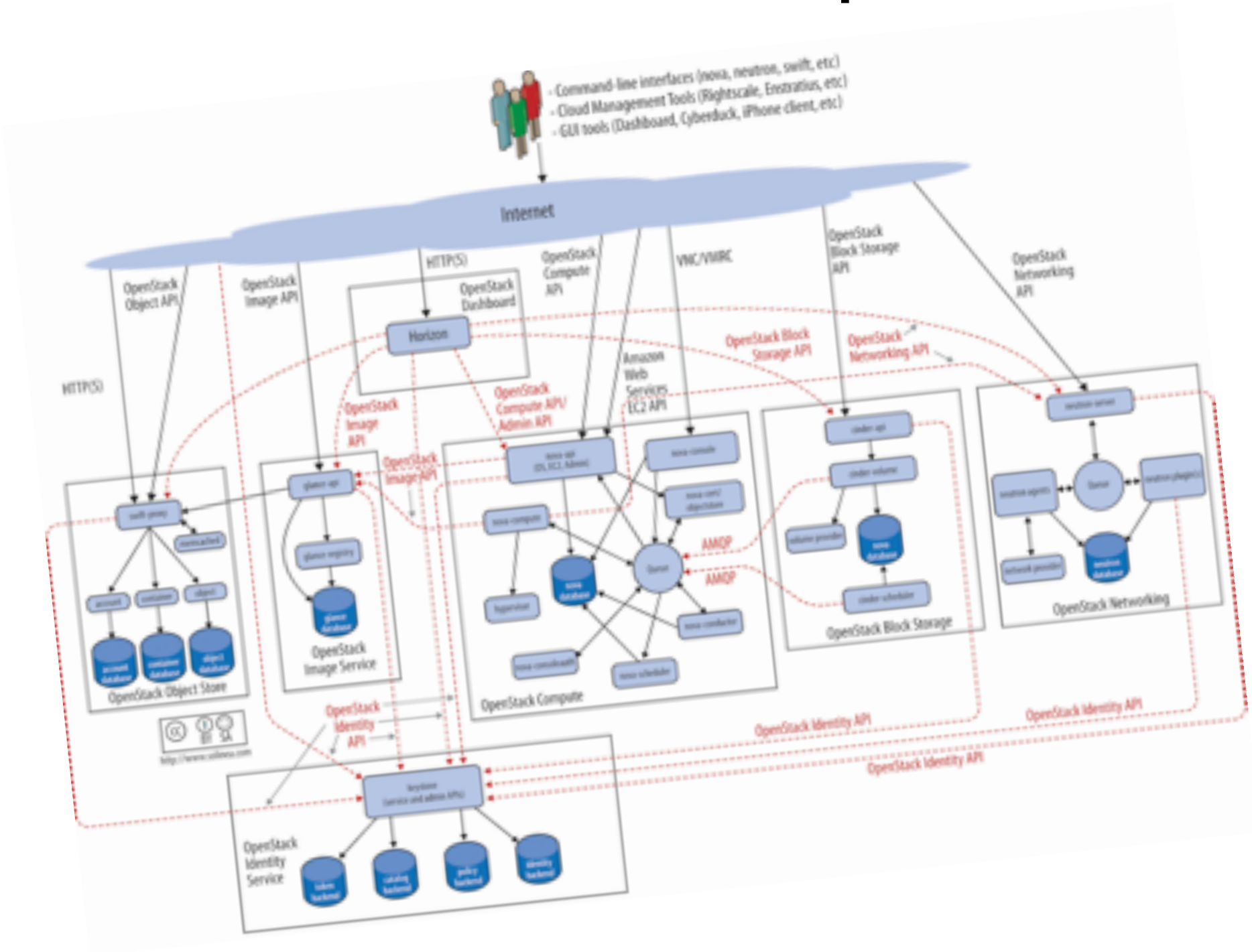
- Industry standard for creating public and private clouds



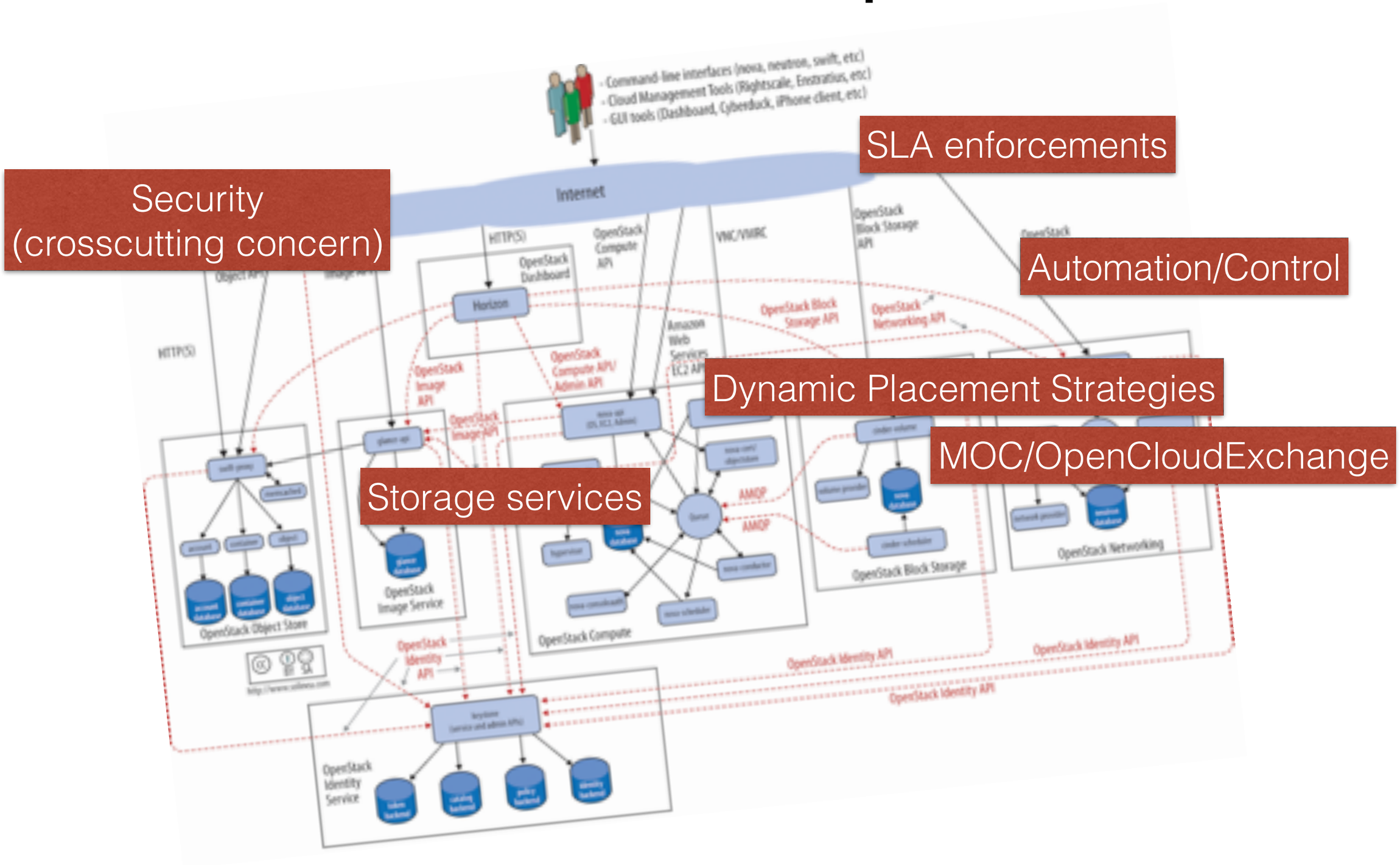


- A rich (and complex) ecosystem
  - **20 Millions of LoC**, 164 services, some services are composed of sub-services (e.g. nova-scheduler, nova-conductor, …)

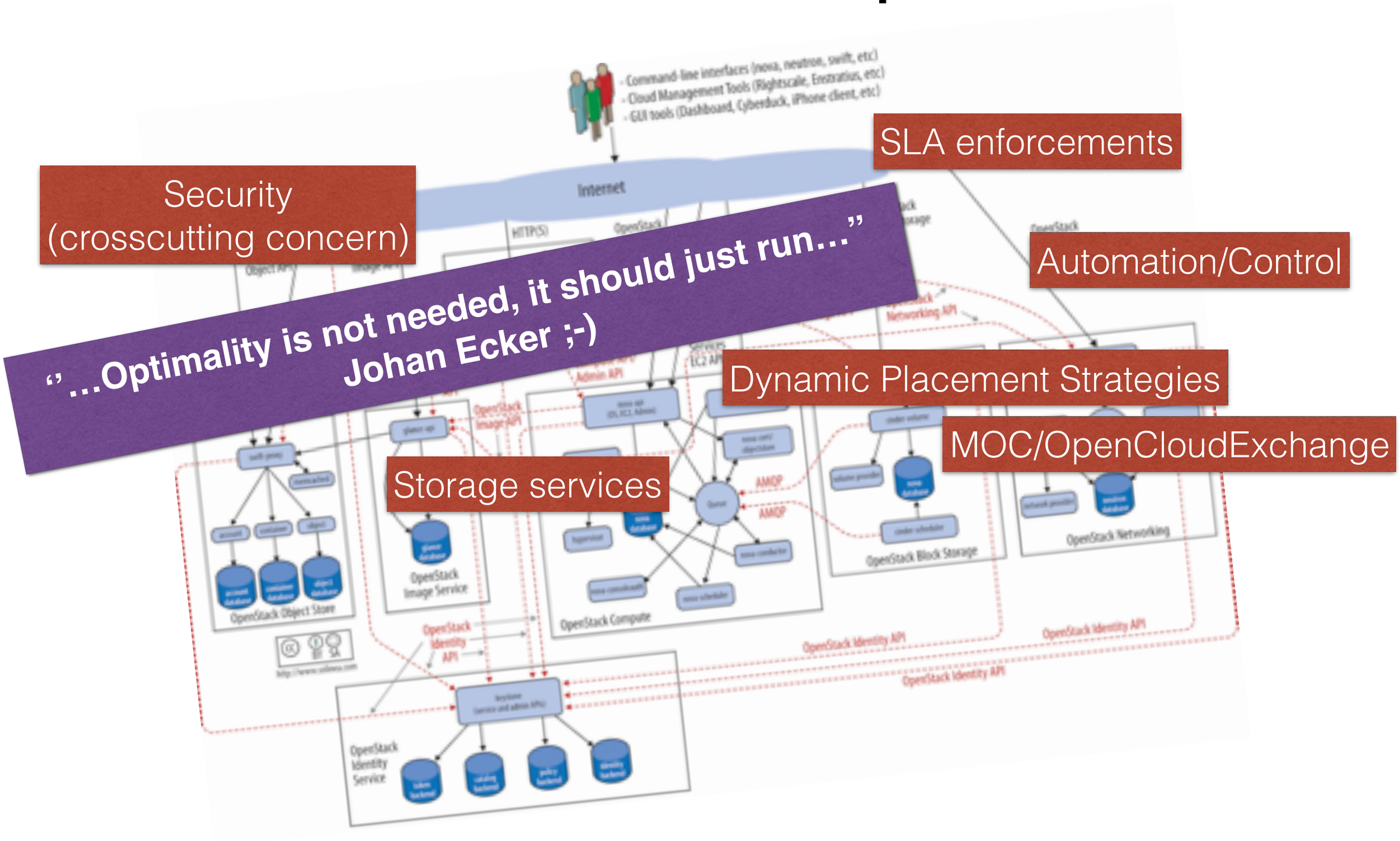
Research vs. OpenStack



Research vs. OpenStack



Research vs. OpenStack



Security
(crosscutting concern)

**"…Optimality is not needed, it should just run…"
Johan Ecker ;-)**

SLA enforcements

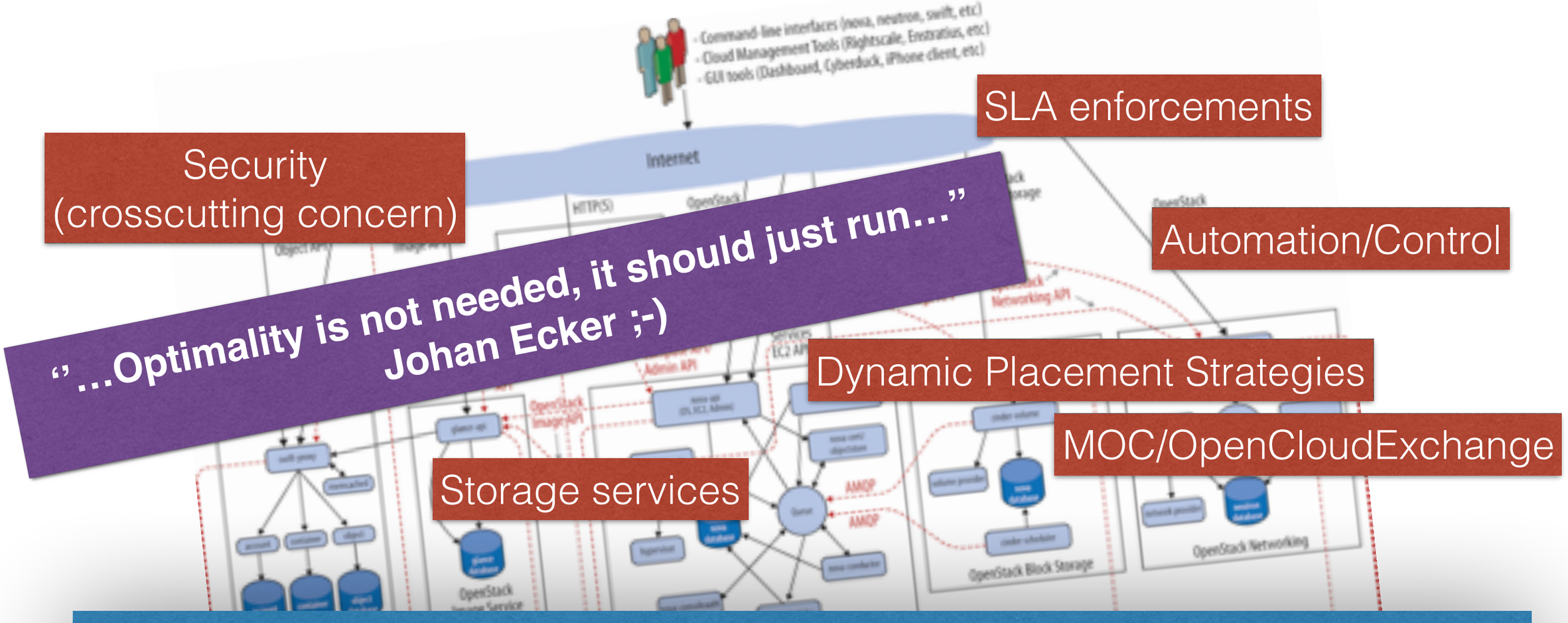
Automation/Control

Dynamic Placement Strategies

MOC/OpenCloudExchange

Storage services

Research vs. OpenStack



SLA enforcements

Security
(crosscutting concern)

"…Optimality is not needed, it should just run…"
Johan Ecker ;-)

Automation/Control

Dynamic Placement Strategies

MOC/OpenCloudExchange

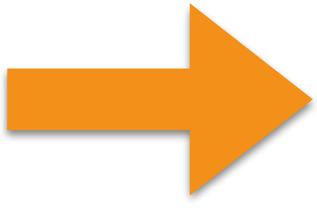
Storage services

How could we favour the scientific community to be involved in OpenStack, like it has been once done for Linux.

EnOS
Experimental Environment for OpenStack

- Motivation: help developers/researchers
  - Understand how OpenStack runs and identify possible improvements
  - Make new proposals and evaluate them under multiple setups/topologies
- **Conducting performance analyses in a scientific and reproducible manner**
    - Automation
      At small and large-scale
      Under different network topologies (traffic shaping)
      Between different releases
      With any kind of benchmarks
- Workflow

  1. Book and provision servers
  2. Deploy OpenStack
  3. Perform benchmarks
  4. Collect and save metrics
  5. Visualize & share



```
$ enos deploy
$ enos bench
$ enos backup
```

EnOS deploy Resource/Topology Description

```
$ cat ./basic.yml
resources:
  clusterA:
    control: 1
    network: 1
  clusterB:
    compute: 50

$ enos deploy -f basic.yml
```

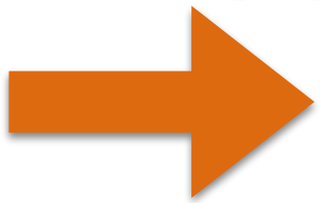
```
$ cat ./advanced.yml
resources:
  clusterA:
    control: 1
    network: 1
    nova-conductor: 5
  clusterB:
    compute: 50

$ enos deploy -f advanced.yml
```

```
$ cat ./network-topo.yml
resources:
grp1:
    clusterA:
      control: 1
      network: 1
      nova-conductor: 5
grp2:
    clusterB:
      compute: 50
network_constraints:
  src: grp1
    dst: grp2
    delay: 100ms
    rate: 10Gbit
    loss: 0%
    symetric: true

$ enos deploy -f network-topo.yml
```

EnOS deploy - Under the Hood

```
resources:
  grp1:
    clusterA:
      control: 1
      network: 1
  grp2:
    clusterB:
compute: 50
network_constraints:
  delay: 100ms
  rate: 10Gbit
  loss: 0%
```

```
$ enos deploy
```



1. Provider gets 2 nodes on clusterA, 50 nodes on clusterB and returns node's IP addresses
2. EnOS provisions nodes with Docker daemons
3. EnOS installs OpenStack using Kolla
4. EnOS sets up bare necessities (flavors, cirros image, router, ...)
5. EnOS applies network constraints between grp1 and grp2 using tc

---

- Provider to get testbed resources
  - Resources: anything running a Docker daemon and EnOS can SSH to + some IPs
  - Existing Provider: Vagrant (VBox/Libvirt), **Grid'5000, Chameleon, OpenStack** ~500 LoC each
- Kolla to deploy OpenStack over testbed resources (containerised based deployment)
- TC to apply network constraints (Linux Netem)

EnOS bench

- Benchmarks description

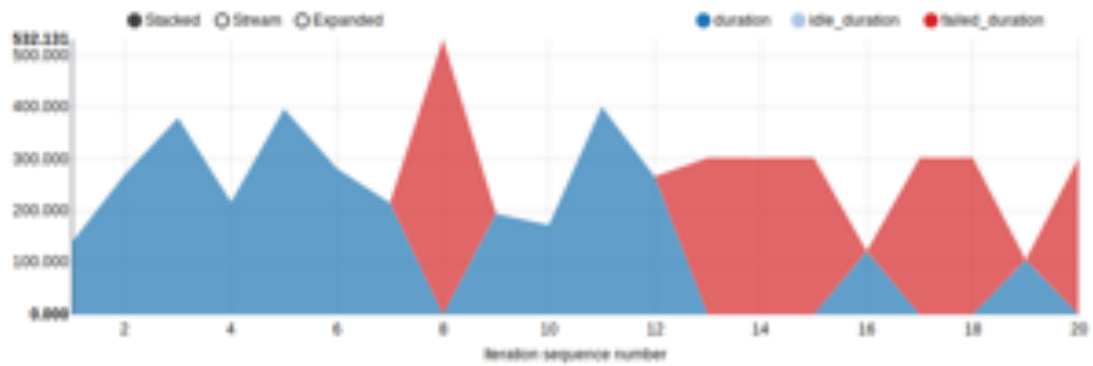
```
$ cat ./run.yml
rally:
  args:
    concurrency: 5
    times: 100
  scenarios:
    - name: boot and list servers
      file: nova-boot-list-cc.yml
      osprofiler: true
    -...
shaker: ...

$ enos bench --workload=run.yml
```

- Under the hood
  - Rally: control plane benchmark
  - Shaker: data plane benchmark
  - OSProfiler: code profiling
  - Monitoring stack: cAdvisor/Collectd to collect CPU/RAM/Network consumption per service/node/cluster

EnOS backup

- EnOS backup produces a tarball with
  - Rally/Shaker reports
  - InfluxDB database with cAdvisor/collectd measures
  - OpenStack logs
  - OS Profiler traces

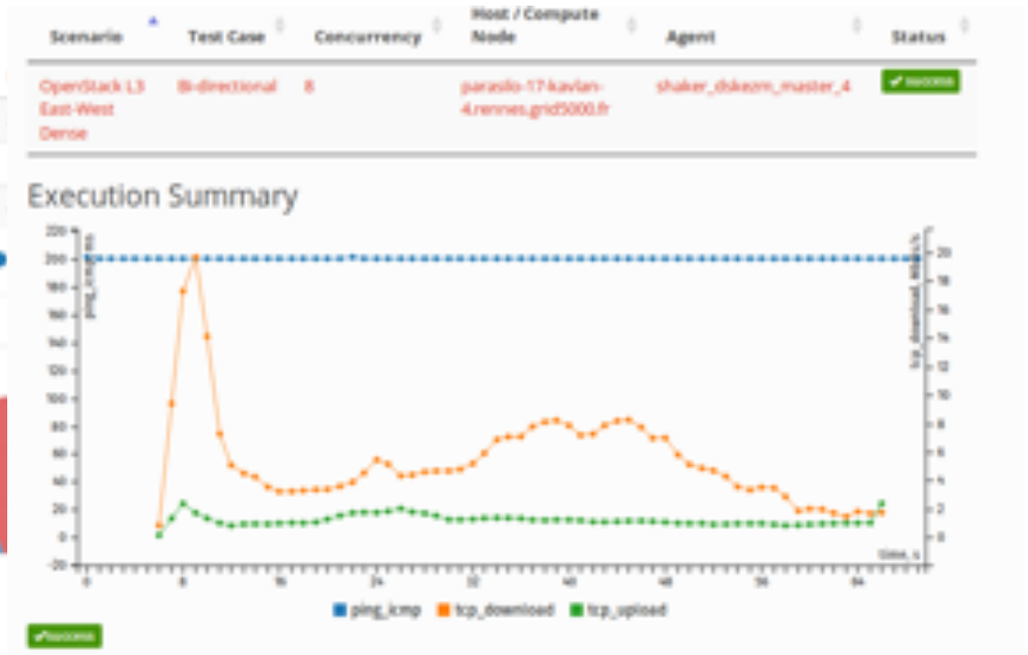
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.boot_server	55.826	205.736	263.908	271.632	279.61	181.196	65.0%	20
nova.delete_server	10.778	40.468	135.245	148.816	162.719	63.336	92.9%	14
total	104.694	217.548	394.025	399.115	401.527	243.532	65.0%	20



Further information: http://enos.readthedocs.io

EnOS backup

- EnOS backup produces a tarball with
  - Rally/Shaker reports
  - InfluxDB database with cAdvisor/collectd measures
  - OpenStack logs
  - OS Profiler traces



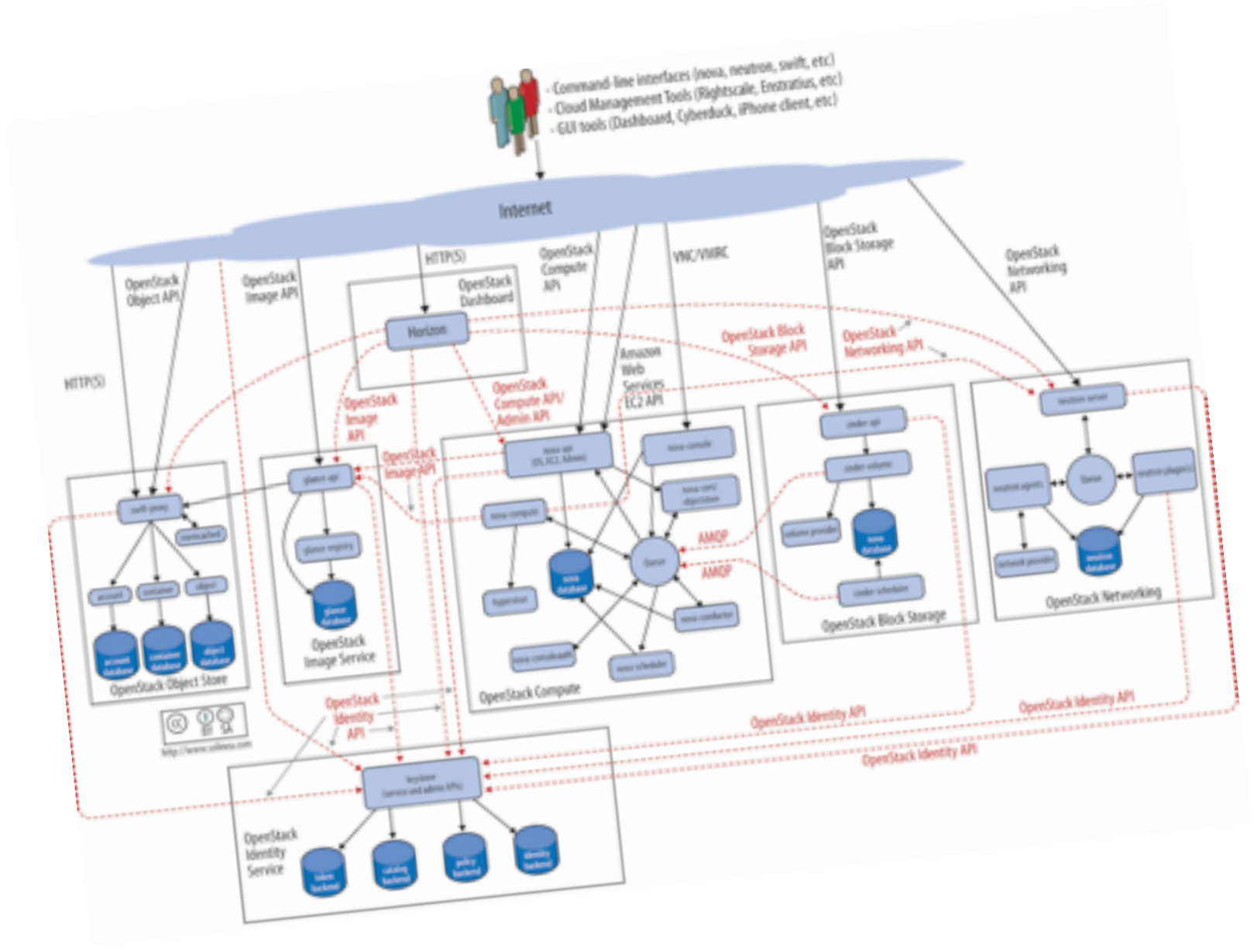
Further information: http://enos.readthedocs.io

EnOS backup

- EnOS backup produces a tarball with
  - Rally/Shaker reports
  - InfluxDB database with cAdvisor/collectd measures
  - OpenStack logs
  - OS Profiler traces

Further information: http://enos.readthedocs.io

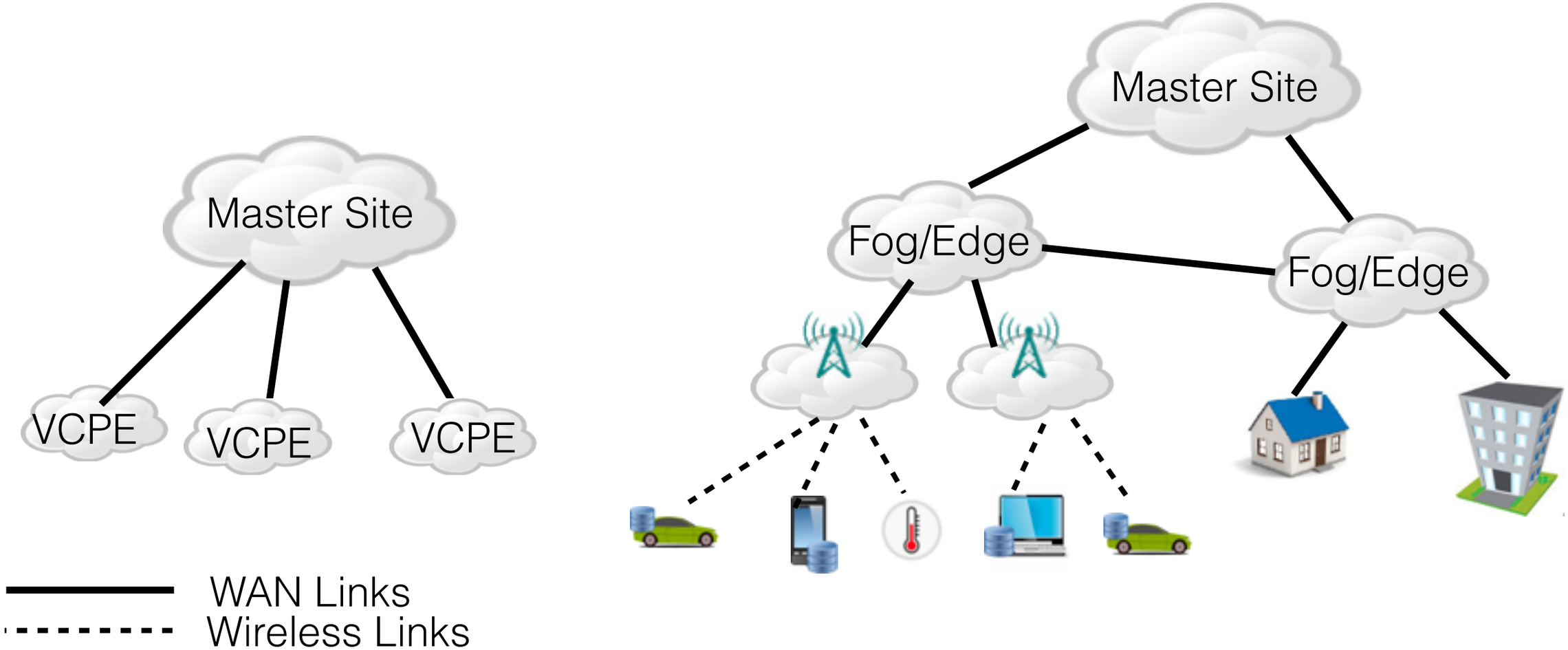
Let’s see an example…



OpenStack WANWide

NFV/Fog/Edge
WANWide OpenStack Cloud Infrastructures

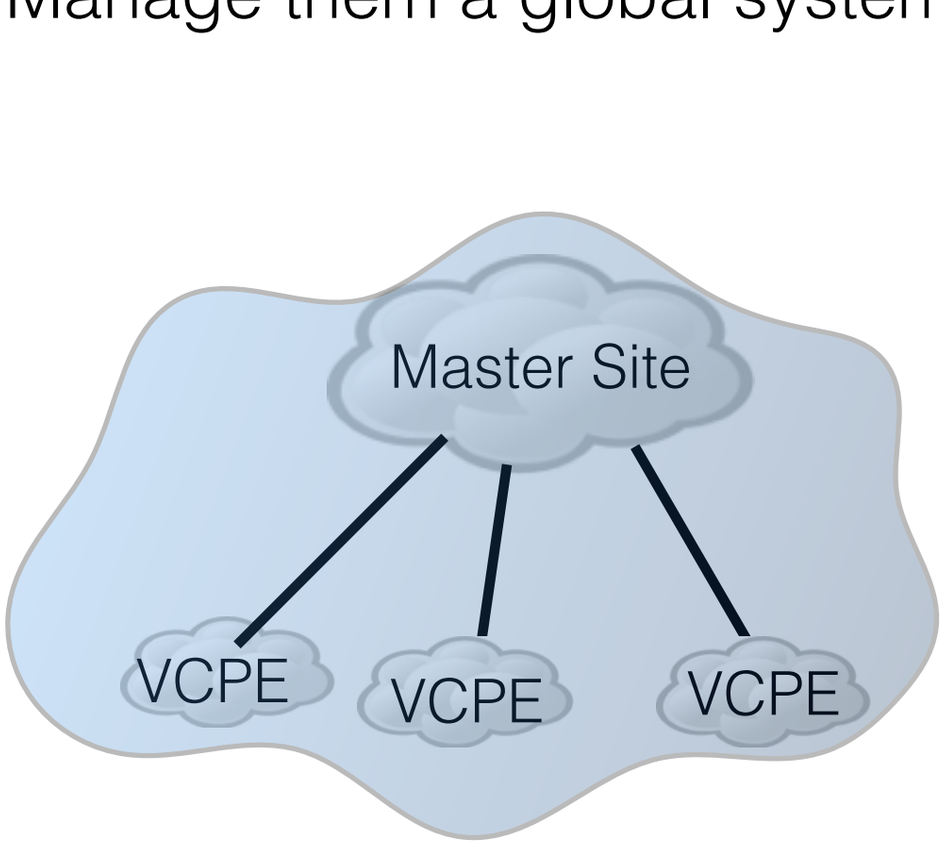
- Telcos viewpoint: Computation/Storage resources are deployed in network Points of Presence (aka. PoP)

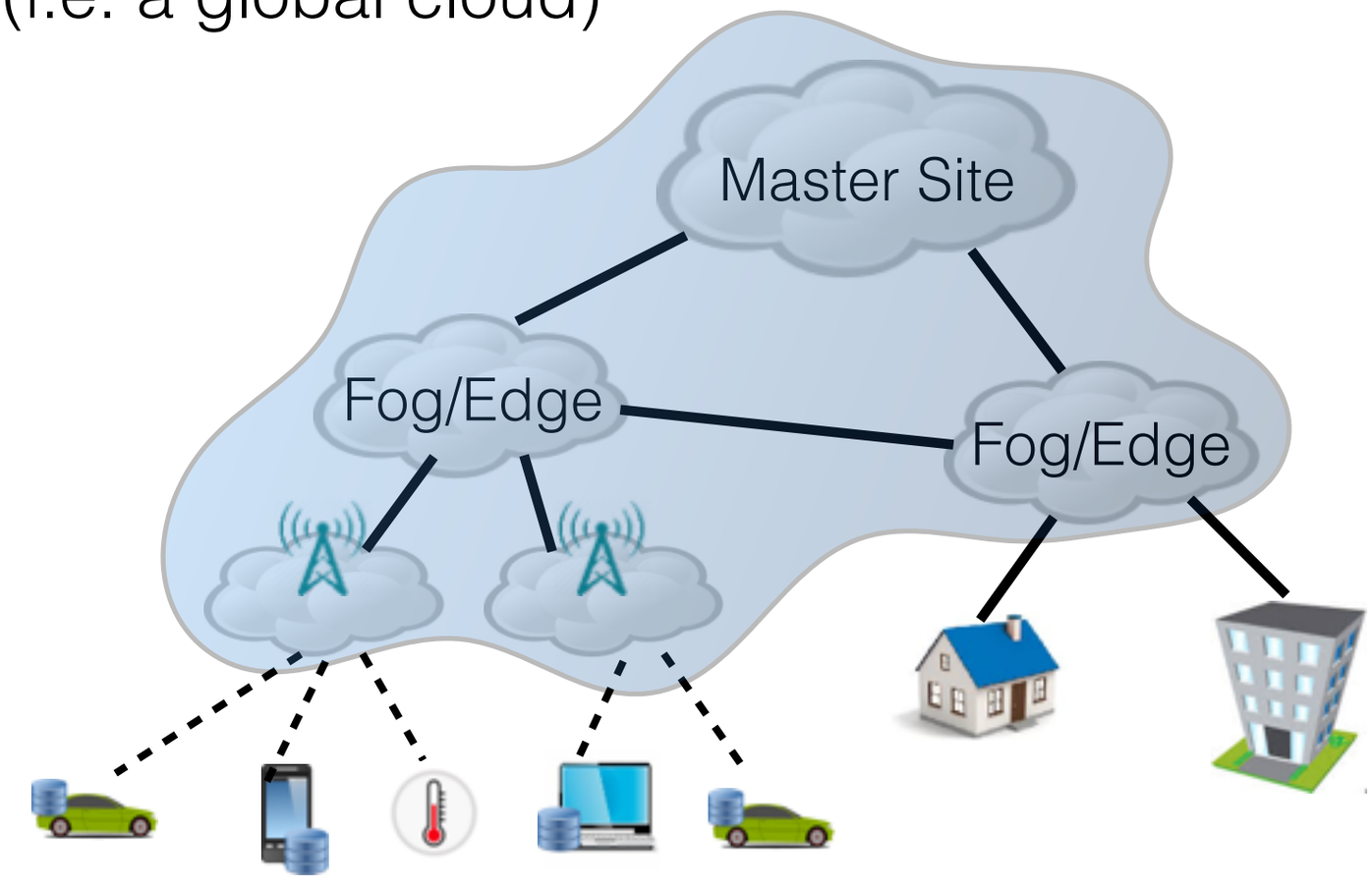


NFV/Fog/Edge
WANWide OpenStack Cloud Infrastructures

- Telcos viewpoint: Computation/Storage resources are deployed in network Points of Presence (aka. PoP)

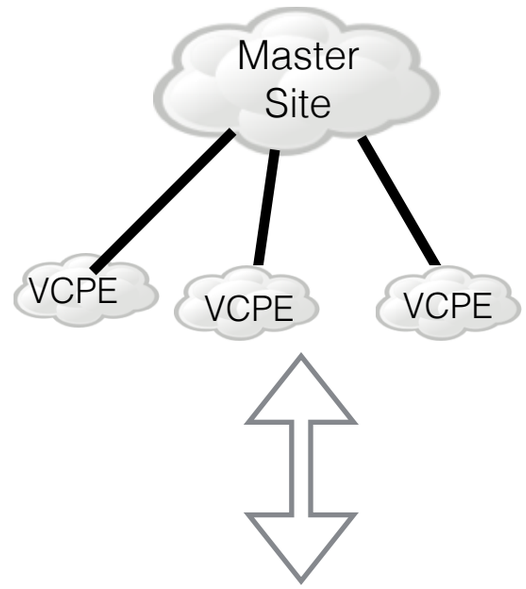
- Manage them a global system (i.e. a global cloud)



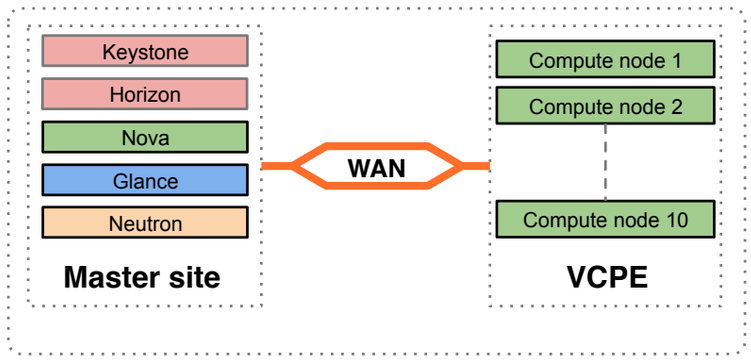


OpenStack WANWide

- A Single OpenStack to operate remote compute resources deployed at the edge
  - All control services are deployed into the master.
  - The communication bus is deployed across all locations (i.e., through each server composing the infrastructure)

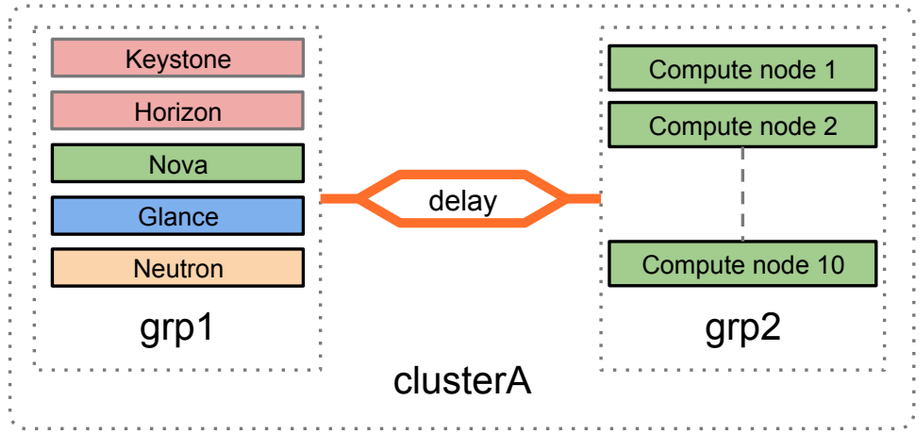


- Pros: simple
- Cons:
  - security management for RPC message and port, Single Point of Failure...
  - Scalability
  - **Network latency/throughput impacts on functional behavior and performance degradations.**



Latency impact

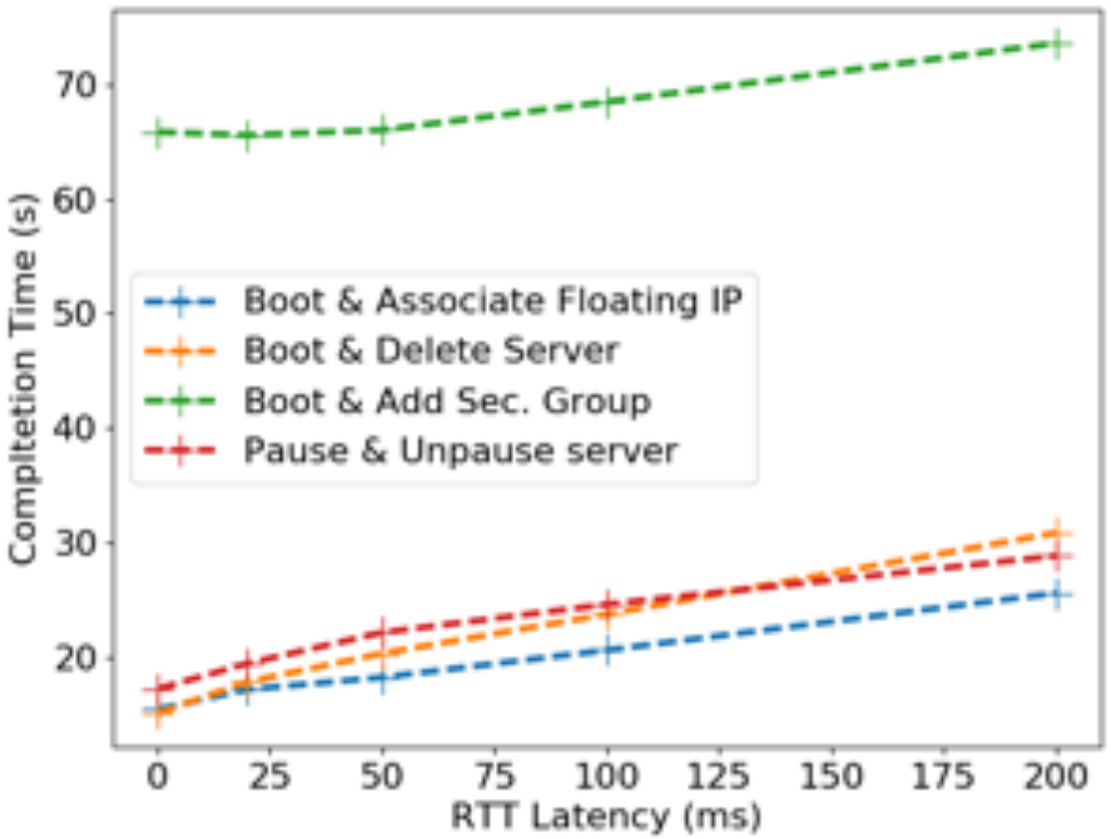
```
$ cat ./wan-exp1.yml
resources:
 grp1:
   clusterA:
       control: 1
 grp2:
    clusterA:
     compute: 10

network_constraints:
  delay: 0ms # 10ms, 25ms, 50ms,
  100ms loss: 0%
  rate: 10Gbit
  src: grp1
  dest: grp2
  symetric: true

$ enos deploy -f wan-exp1.yml
```



Experiments performed
on top of Grid'5000 and Chameleon

Latency Impact – Control Plane (Rally Metrics)

```
$ cat ./run.yml
rally:
 args:
  concurrency: 1
  times: 20
 scenarios:
  file:nova-boot-and-associate-floating-ip.yml
  file:nova-boot-and-delete.yml
  file:nova-boot-and-add-secgroup.yml
  file:nova-pause-and-unpause.yml

shaker: ...

$ enos bench --workload=run.yml
```



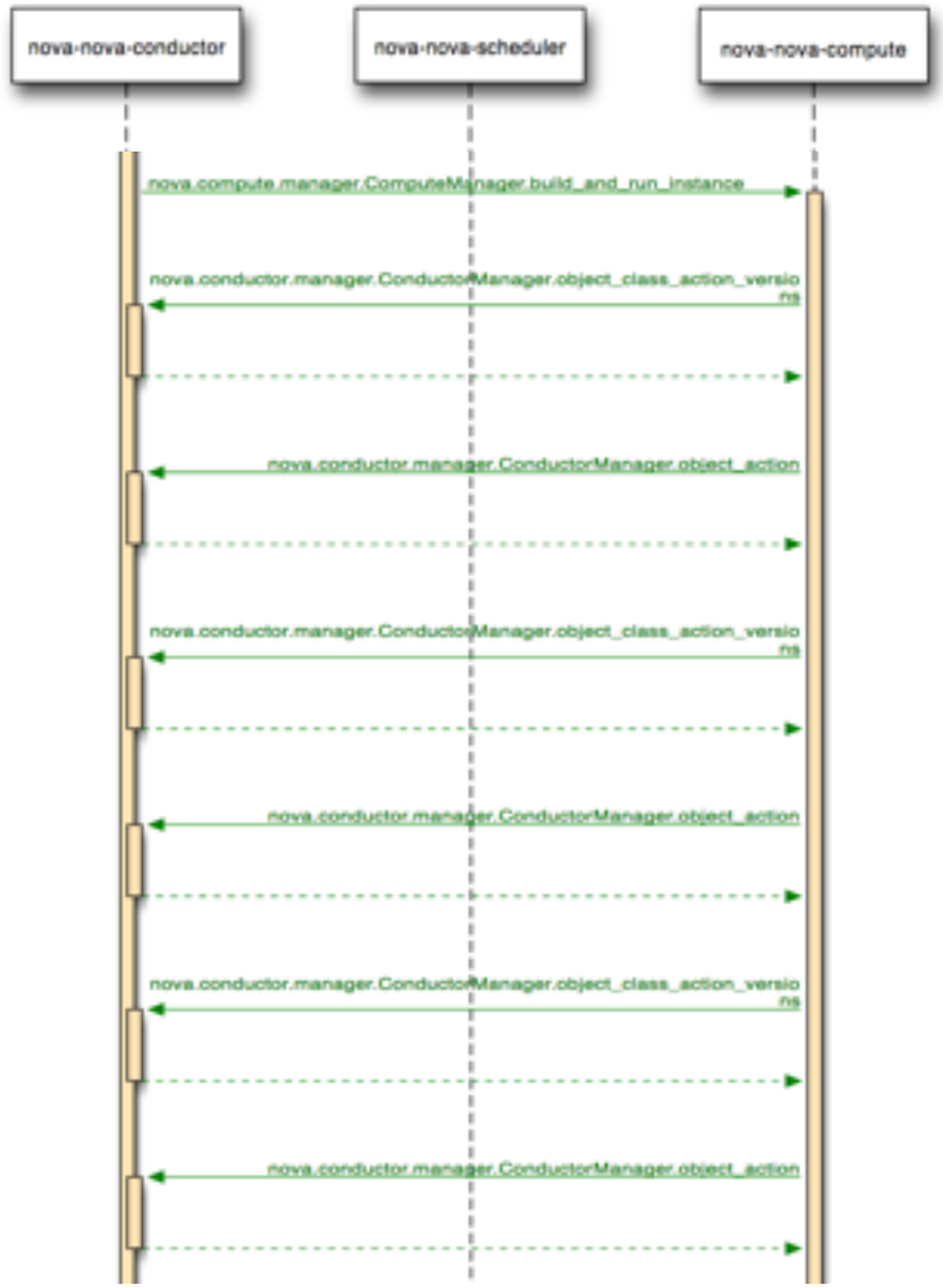
Latency Impact – Control Plane (OSProfiler Metrics)

HTML View

− 4	51 ms	rpc	nova	nova-conductor
5	2 ms	db	nova	nova-conductor
5	2 ms	db	nova	nova-conductor
− 5	5 ms	rpc	nova	nova-compute
+ 6	52 ms	rpc	nova	nova-conductor
+ 6	196 ms	rpc	nova	nova-conductor
+ 6	14 ms	rpc	nova	nova-conductor
+ 6	208 ms	rpc	nova	nova-conductor
+ 6	20 ms	rpc	nova	nova-conductor
+ 6	55 ms	rpc	nova	nova-conductor
+ 6	145 ms	rpc	nova	nova-conductor
+ 6	46 ms	rpc	nova	nova-conductor
6	1 ms	neutron_api	nova	nova-compute

. . .

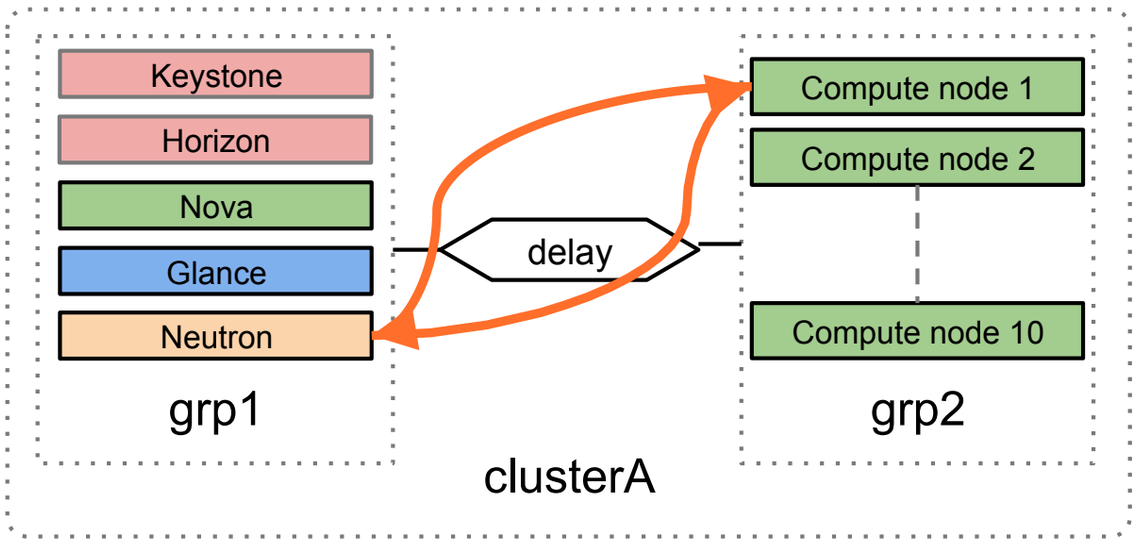
Sequence Diagram

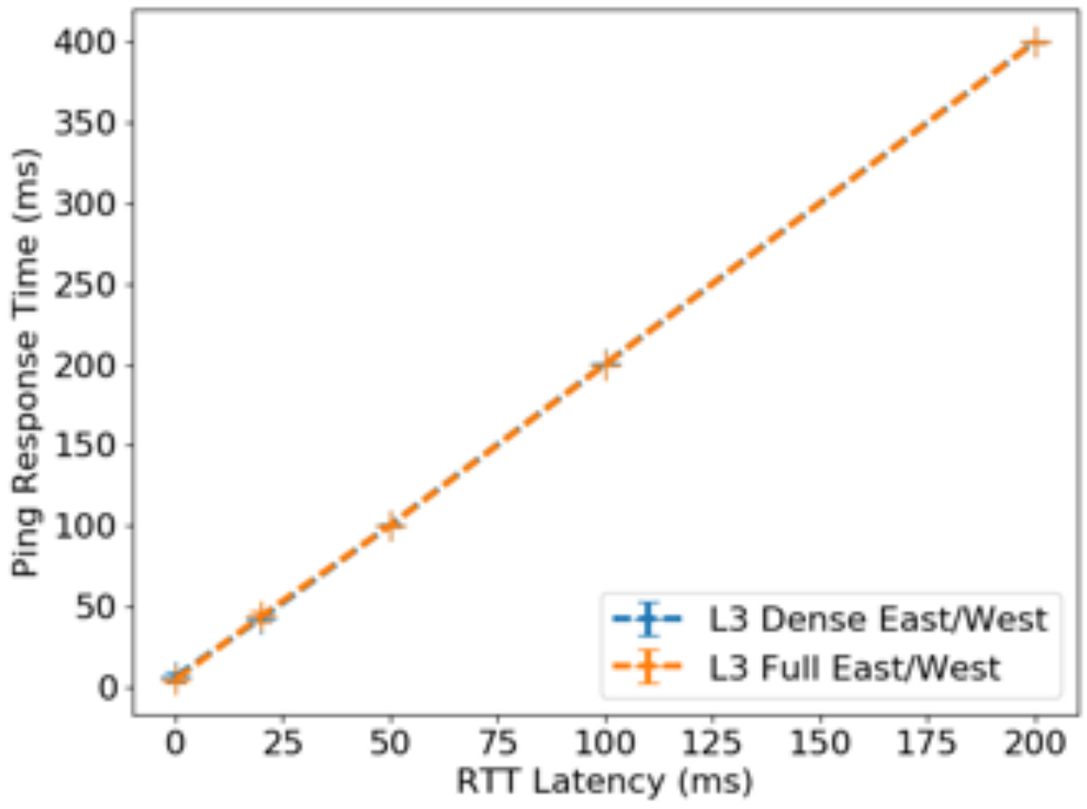


. . .

Latency Impact – Data Plane (Shaker Metrics)

```
$ cat ./run.yml

rally: ...
shaker:
  file: openstack/dense_l3_east_west.yml
  file: openstack/full_l3_east_west.yml

$ enos bench --workload=run.yml
```



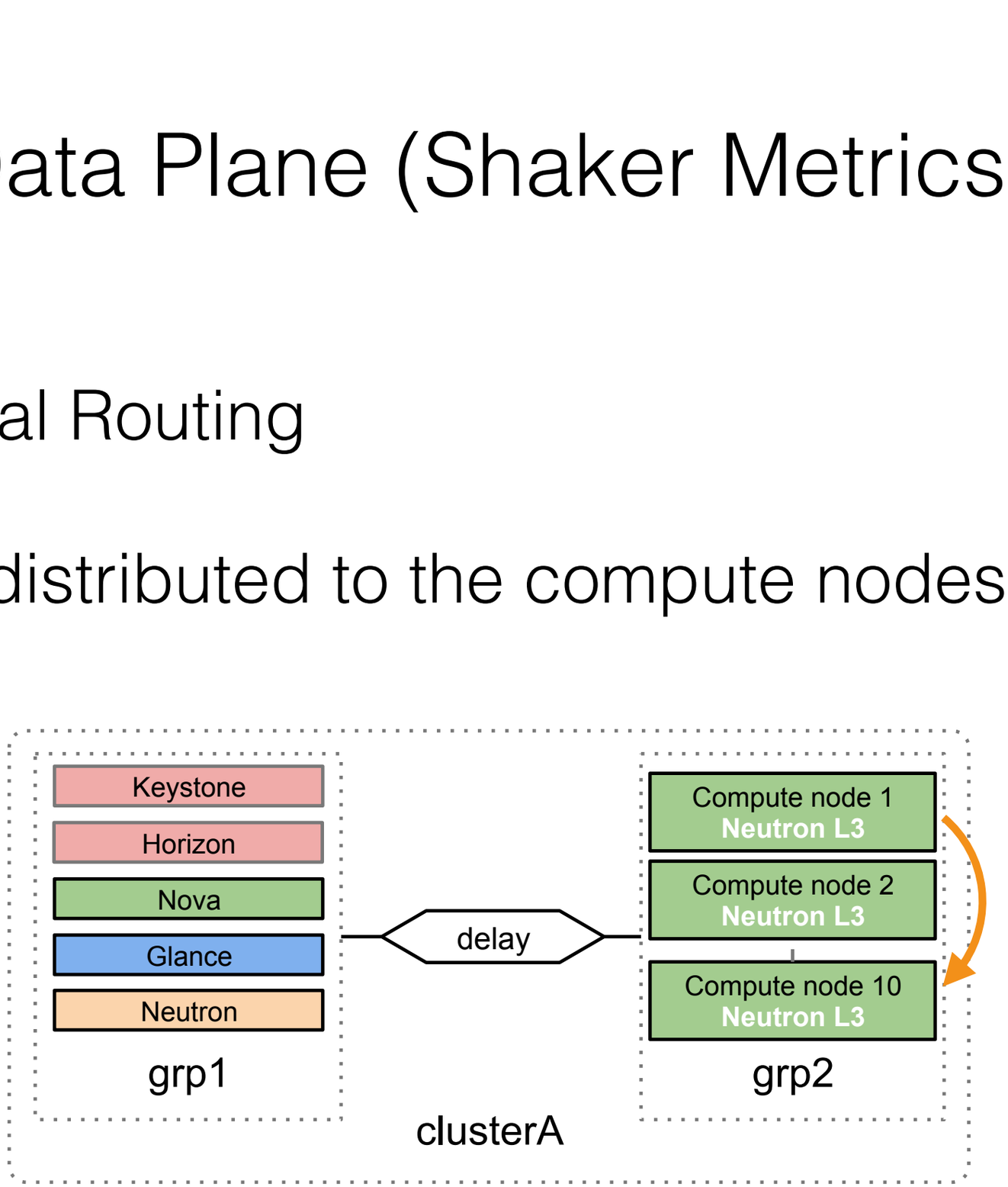


Ping response time is twice the RTT (which corresponds to the normal workflow)

Latency Impact – Data Plane (Shaker Metrics)

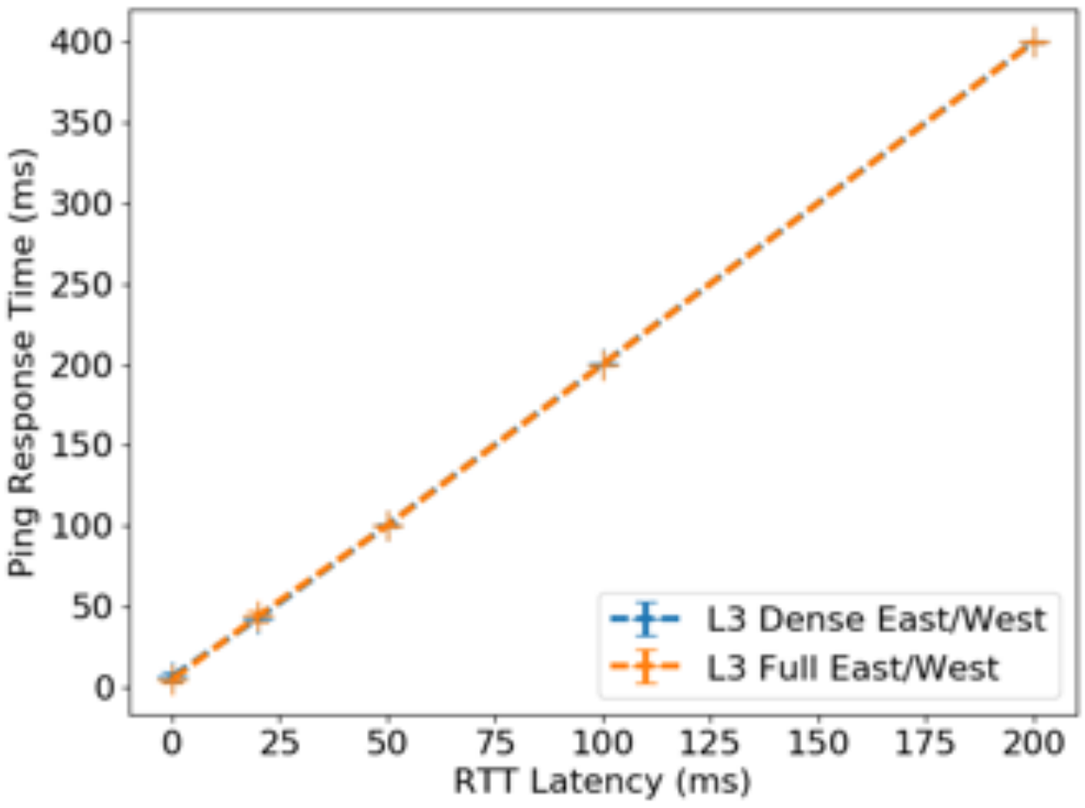
- DVR: Distributed Virtual Routing
  - L3 forwarding/NAT distributed to the compute nodes

```
$ cat ./wan-exp2.yml

resources: ...
network_constraints: ...
kolla:
  enable_neutron_dvr: true

$ enos deploy -f wan-exp2.yml
```

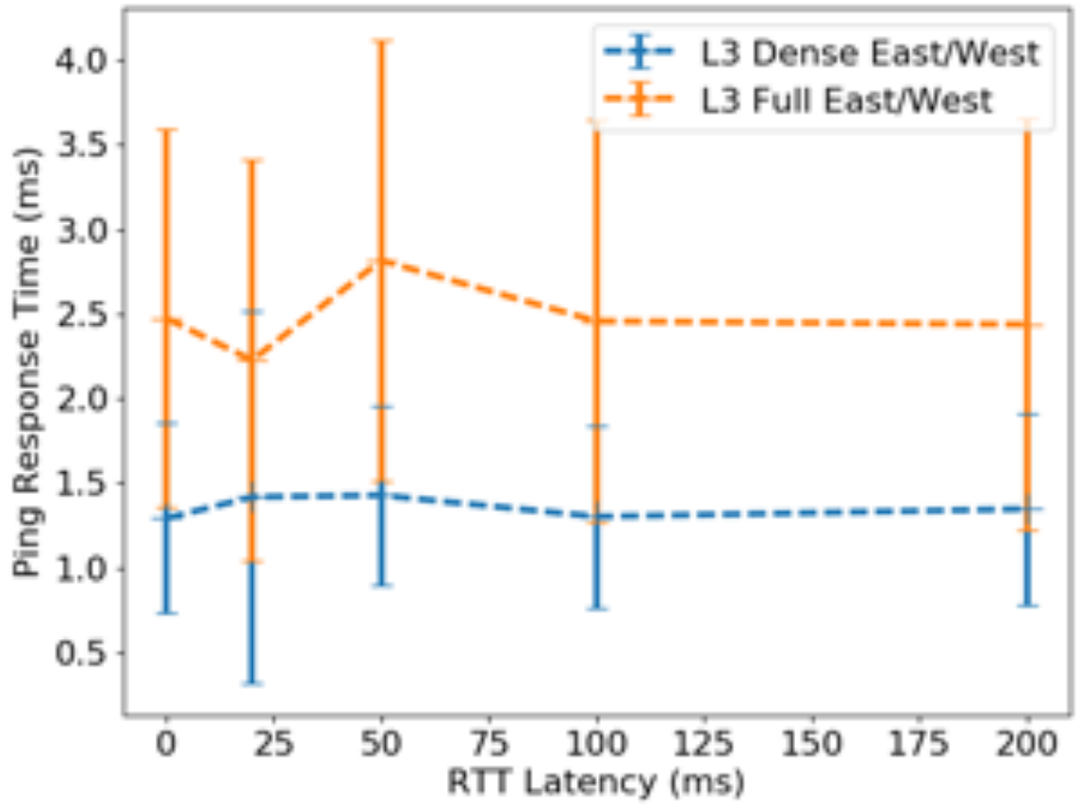


Latency Impact – Data Plane (Shaker Metrics)

Without DVR (2*RTT)



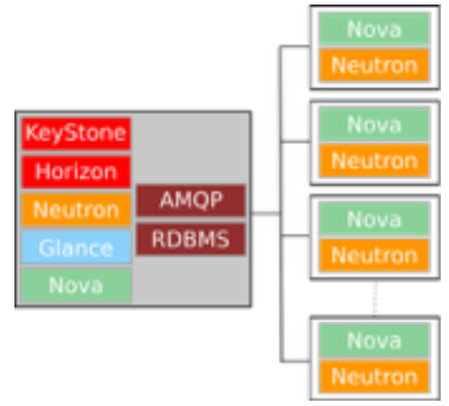
With DVR (LAN RTT)



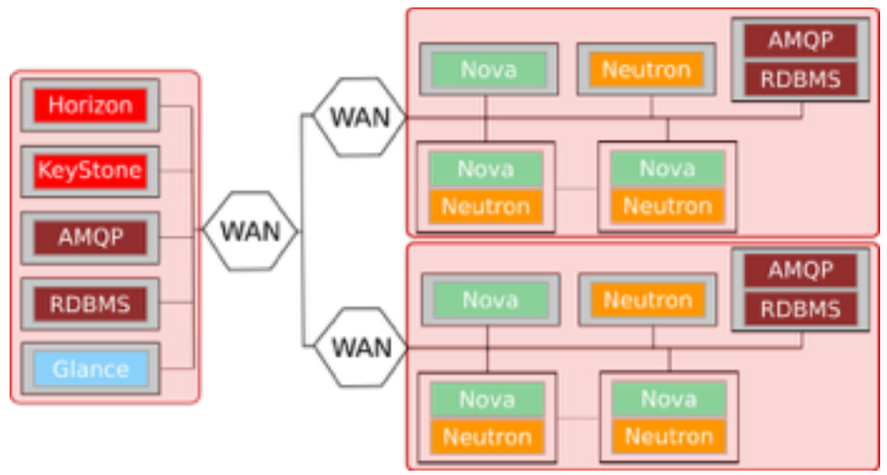
Proposed for Scalability issues but critical for WAN deployments

OpenStack Performance Characterisation (still an on-going action)

- Scalability: OpenStack to manage 1000 nodes and more…
- WANWide deployment
  - Latency impact
    Network degradations (loss impact)
  - More advanced deployment strategies
- On-going actions
  - short term: understanding gathered results
  - mid-term: automated Performance Regression Testing



Single controller,
multiple compute nodes



multiple controller,
multiple compute nodes,
multiple regions

Takeaway message 1

- **EnOS, A tool for diving into OpenStack and performing scientific investigations**

  - First reproducible results regarding OpenStack performance (Scalability: OpenStack Summit Barcelona 2016, Latency impact: OpenStack Summit Boston 2017, EnOS: general overview CCGRID 2017)

- **OpenStack is no more a monolithic system** but rather an ecosystem of building blocks to operate virtualised infrastructures.

- Should enable the Scientific community to make scientific contributions while favouring transfers to OpenStack

  - We have been doing that for several software stacks (Linux, hadoop…)

  - **There is room for improvement!**

Takeaway message 2

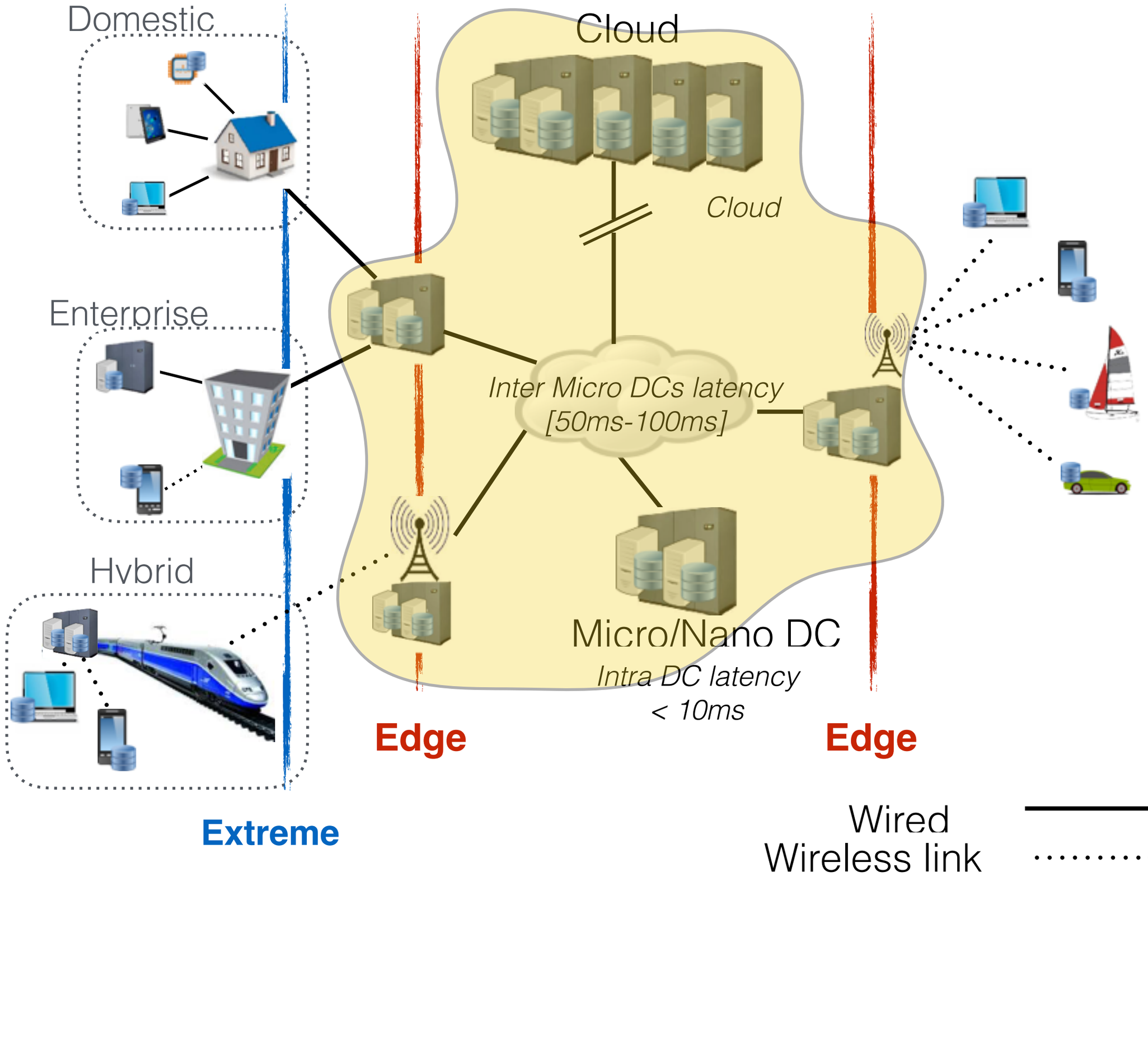
- **OpenStack for operating massively distributed infrastructures**
  - Just the premisses but we believe it should enable our community to move forward faster while preventing us to reinvent the wheel !

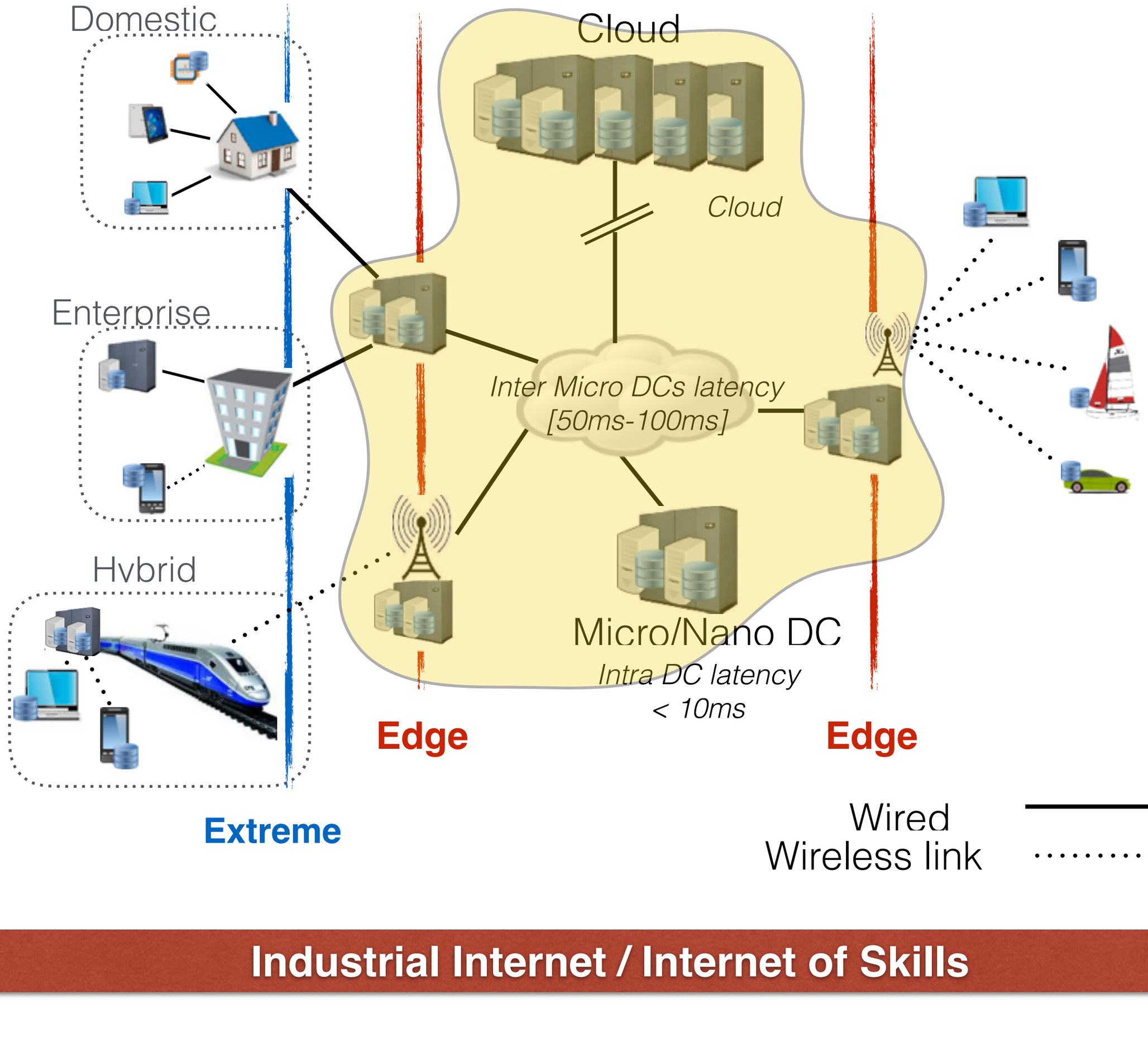
- **Strong interests** from telcos and other initiatives (ETSI MEC, OpenEdgeComputing initiative…)

- Fog/Edge Massively Distributed Cloud WG: debate and investigate how OpenStack can address Fog/Edge Computing use-cases **https://wiki.openstack.org/wiki/Fog_Edge_Massively_Distributed_Clouds**

http://beyondtheclouds.github.io

Thanks to all Discovery colleagues and OpenStack Folks


Domestic
Enterprise
Hvbrid
Cloud
Cloud
Inter Micro DCs latency
[50ms-100ms]
Micro/Nano DC
Intra DC latency
< 10ms
Edge
Edge
Extreme
Wired
Wireless link



Industrial Internet / Internet of Skills