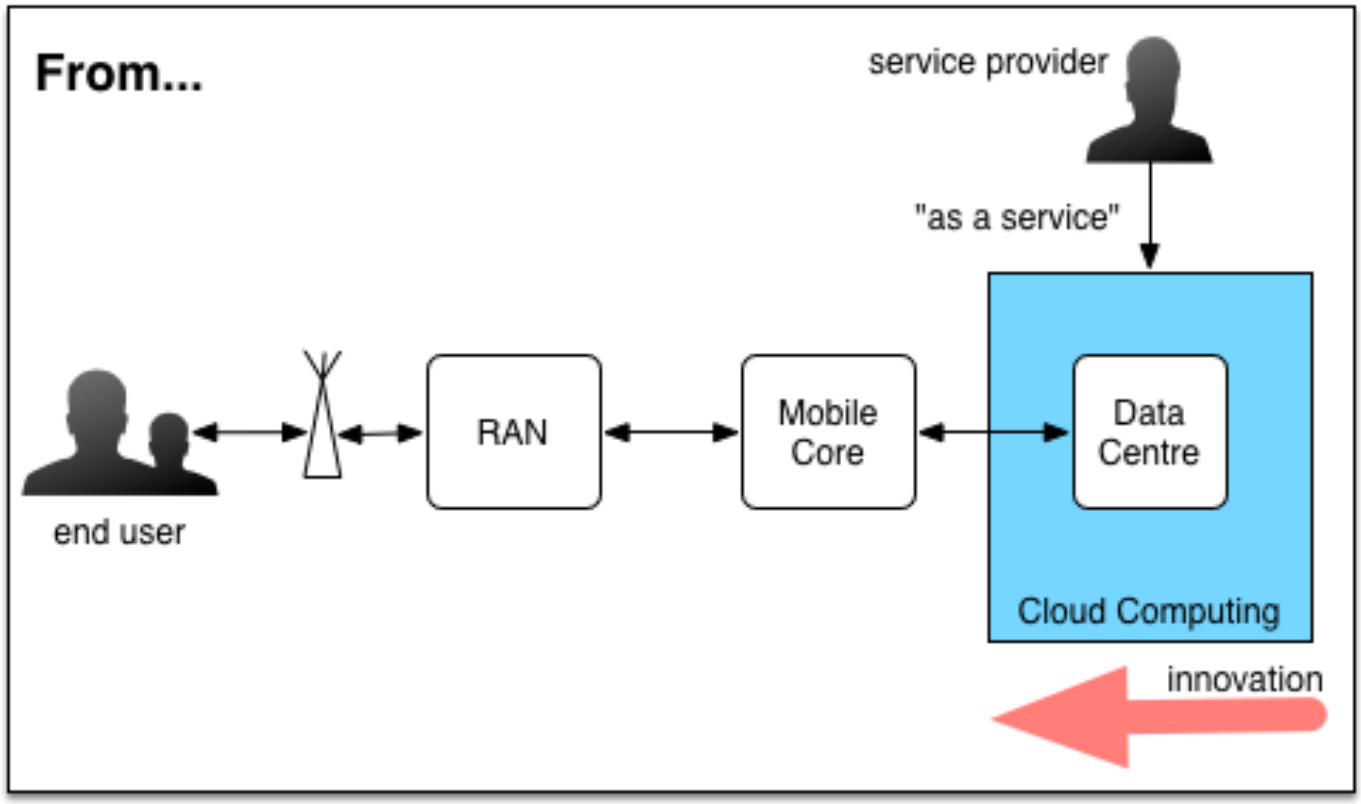
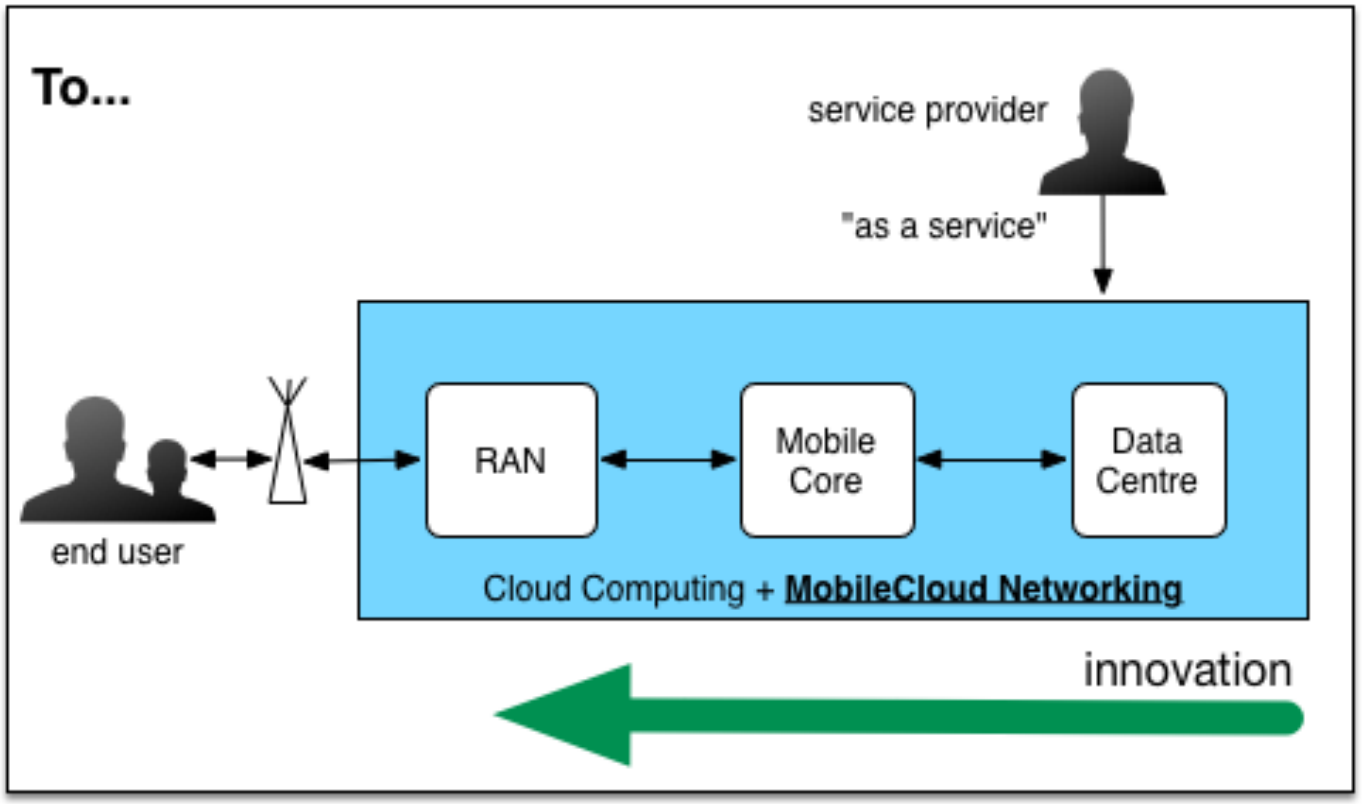
MCN: Beyond NFV

Andy Edmonds, Thomas Michael Bohnert, Giovanni Toffetti



- On-demand and self-service
- Elastic
- Multi-tenant
- Pay-as-you-go

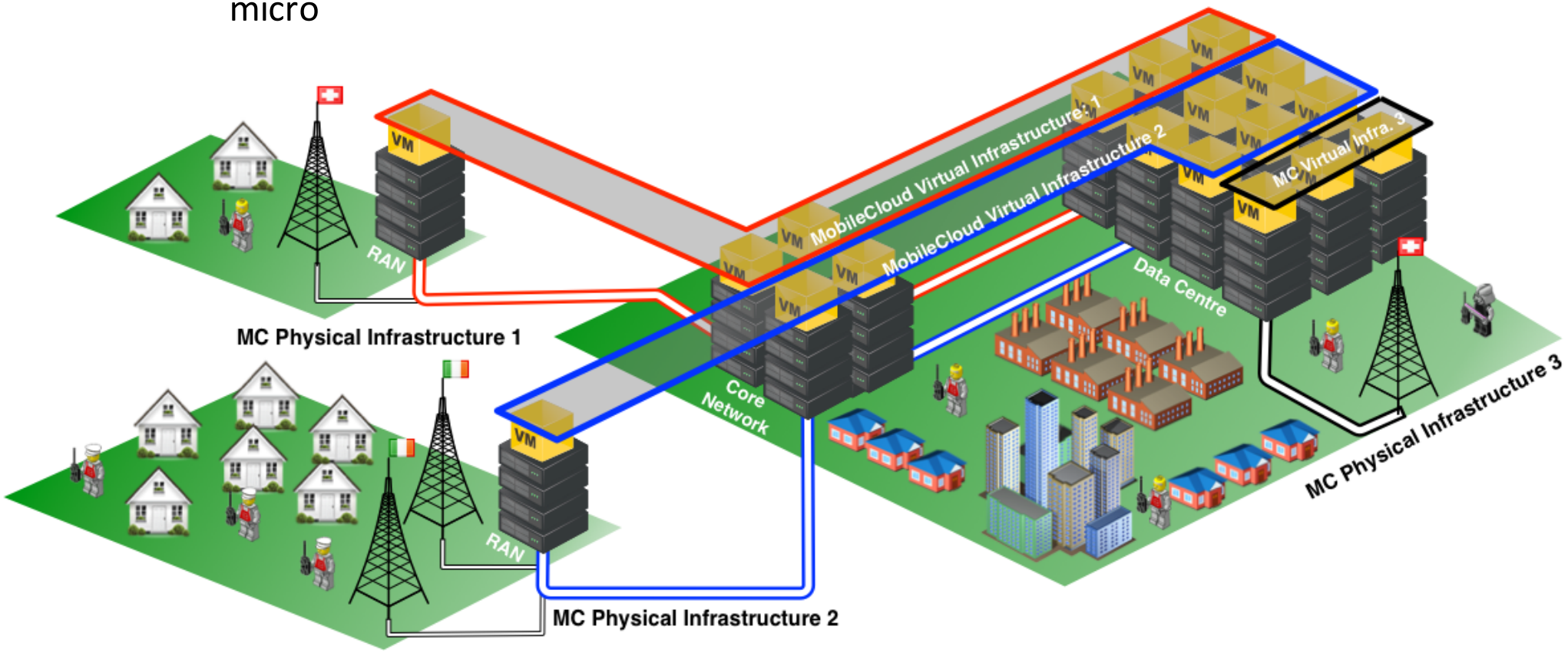


- On-demand and self-service
- Elastic
- Multi-tenant
- Pay-as-you-go

MCN
Mobile Cloud Networking

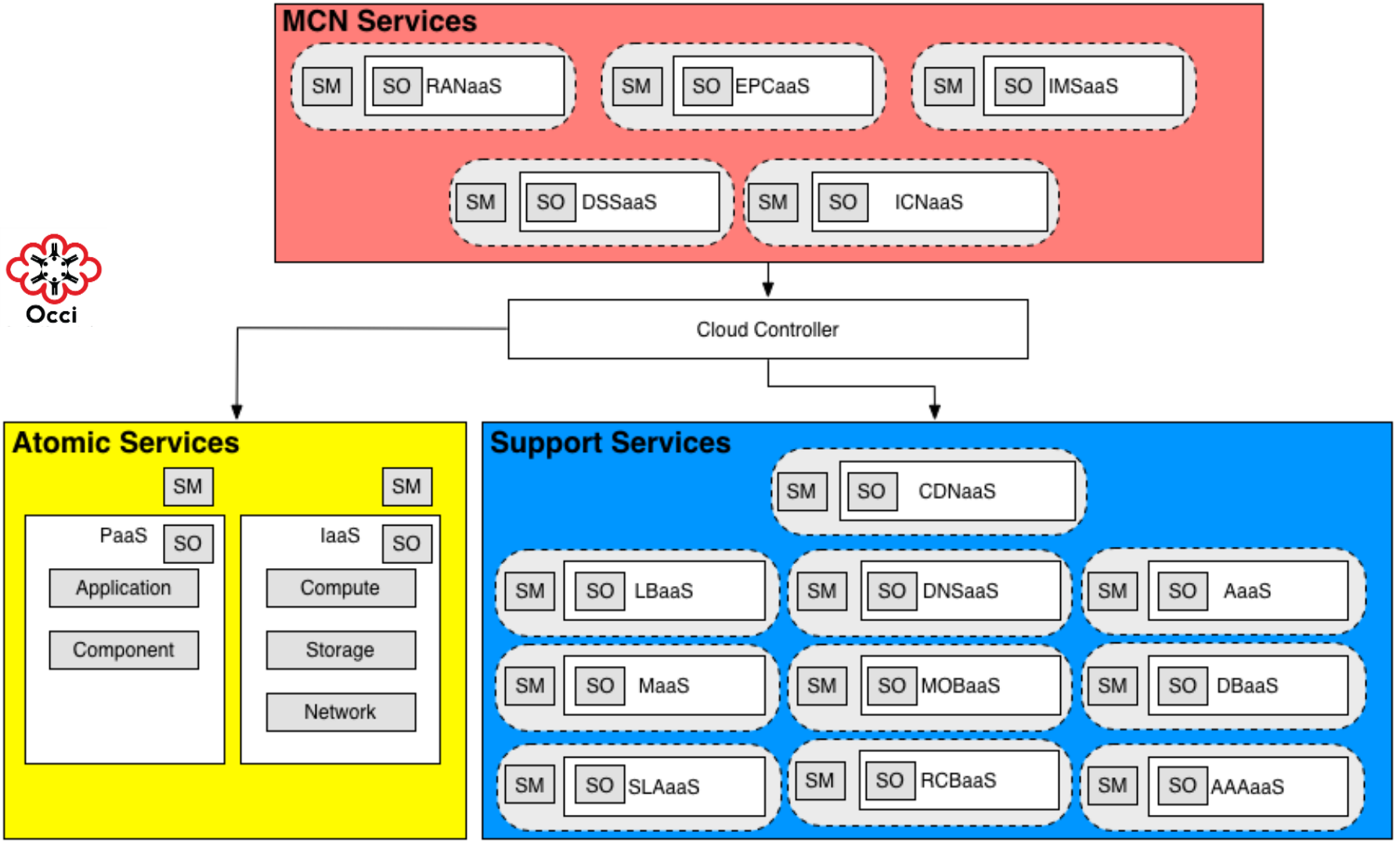
micro

macro



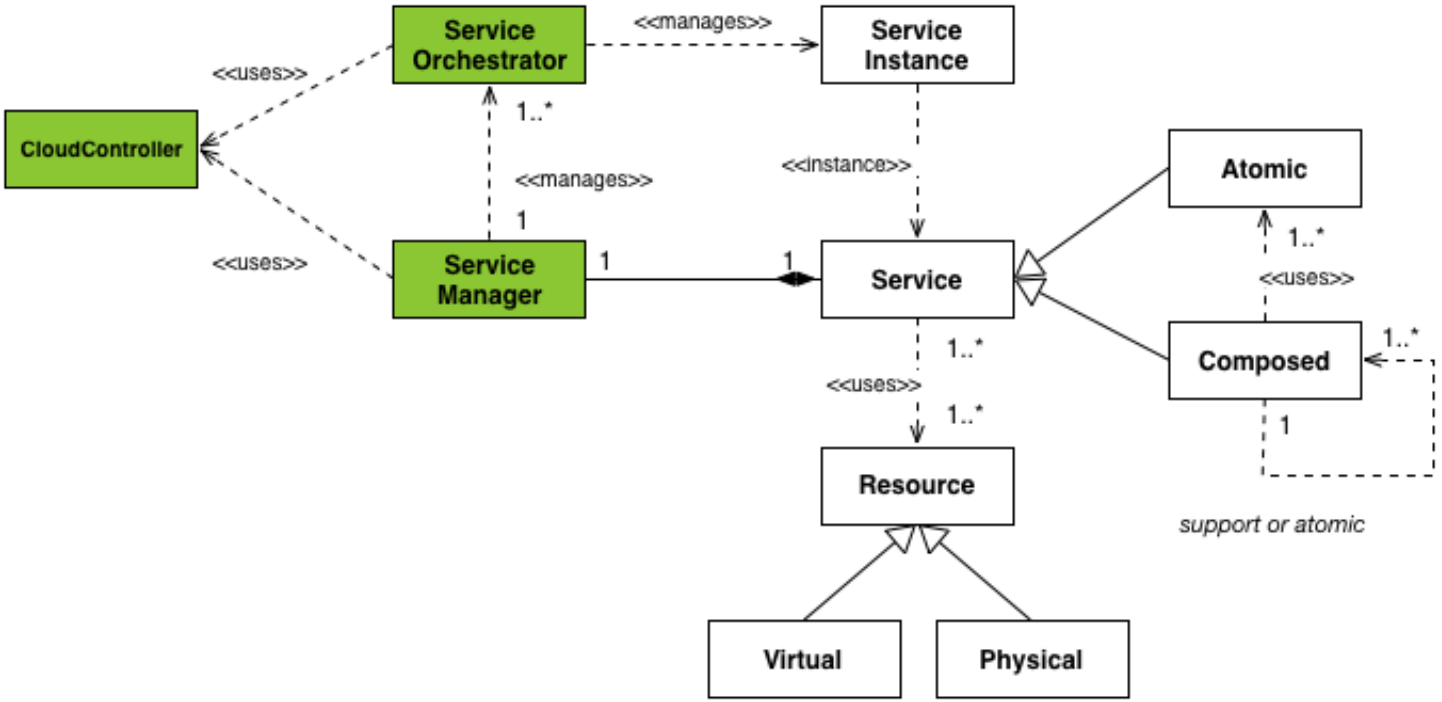
ICC LAB

MCN
Mobile Cloud Networking



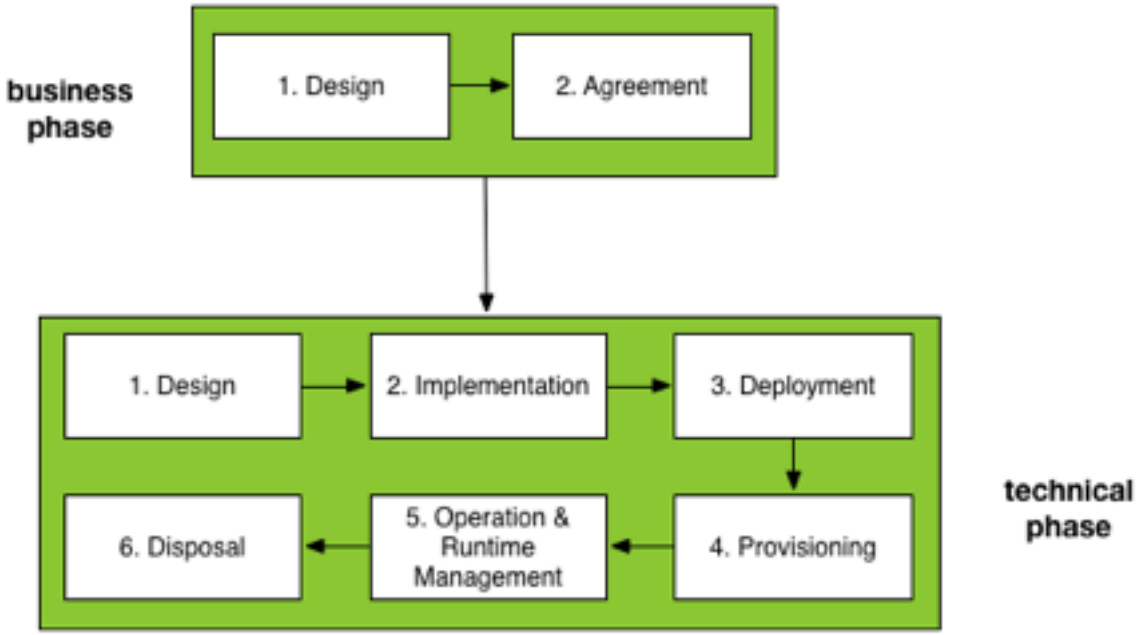
ICC LAB

key entities



- **Service Manager**: receives requests for new tenant service instances
- **Cloud Controller**: manages the lifecycle of a tenant service instance
- **Service Orchestrator**: manages and abstracts underlying resources and SOs

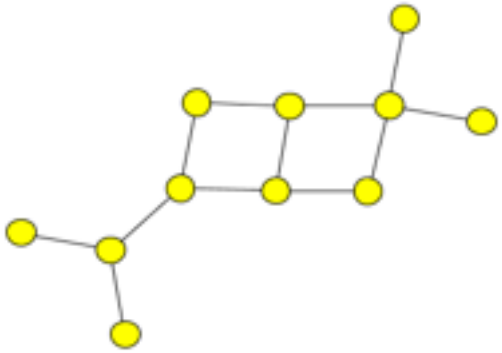
lifecycle



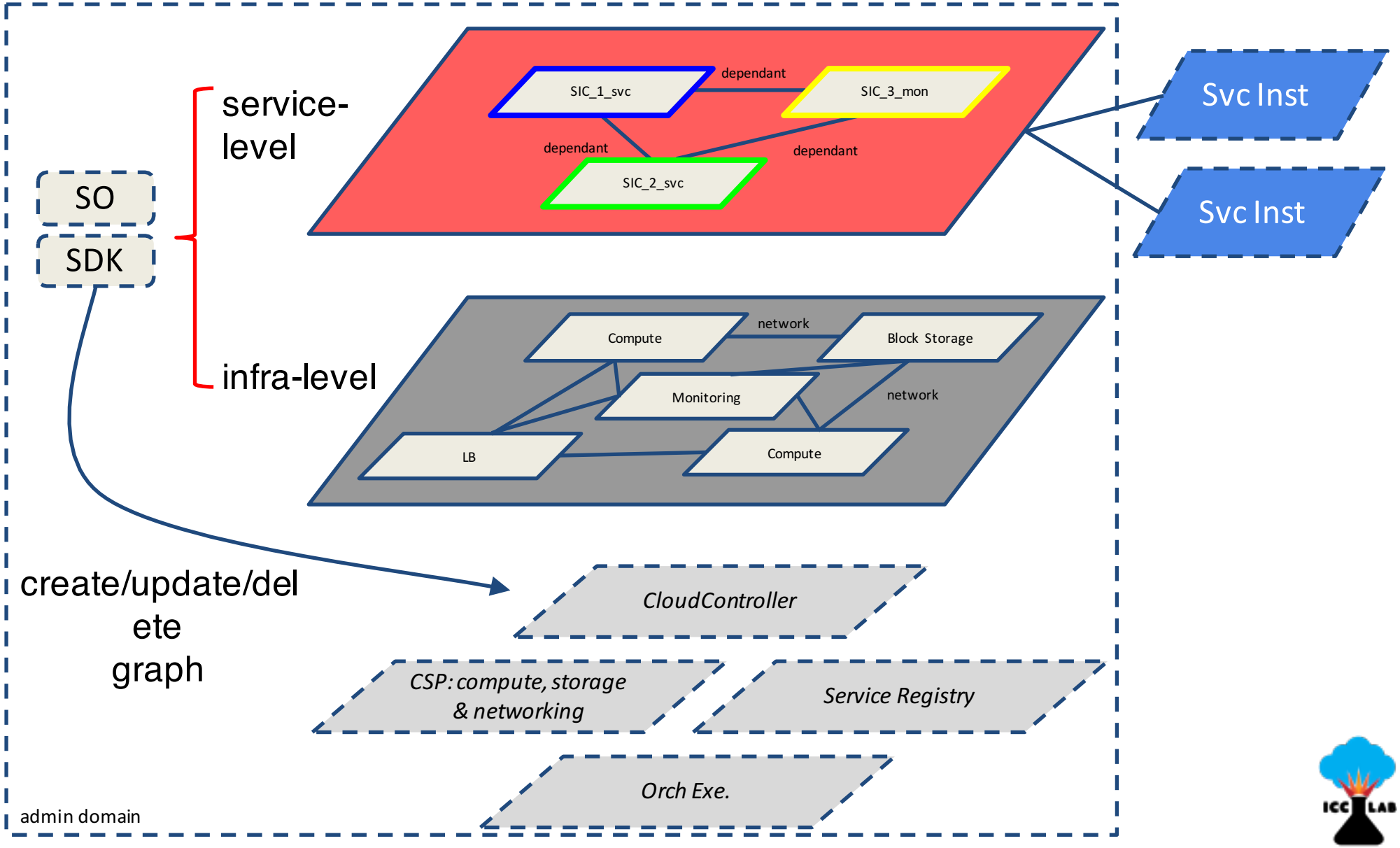
- applies to all entities
- service/resource entities also and their **graphs** too

graphs

- There are two graphs
  - 1 for the SO's Services - STG
  - 1 for the infrastructure - ITG
    - that is: **services enabling the SO's service**
- Both are inter-related
  - “horizontal” & “vertical”



graphs


service-level
infra-level
SO
SDK
SIC_1_svc
dependant
SIC_3_mon
dependant
dependant
SIC_2_svc
Svc Inst
Svc Inst
Compute
network
Block Storage
Monitoring
network
LB
Compute
create/update/delete graph
CloudController
CSP: compute, storage & networking
Service Registry
Orch Exe.
admin domain

orchestration

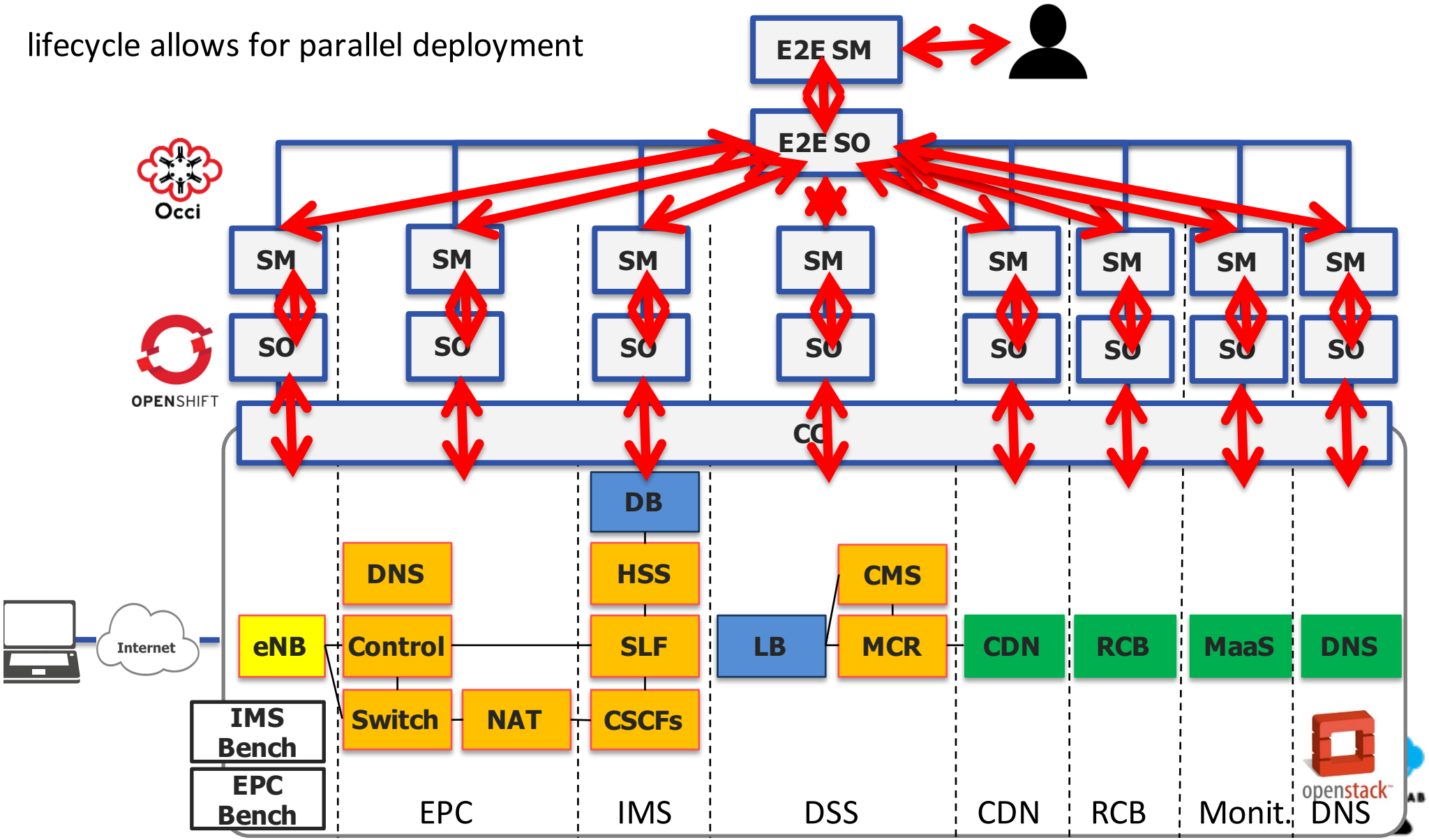
providing a service instance to tenant
service deployment unit: service bundle

- **Service Orchestrator**: Your service's logic
- **Service Manifest**: Your service dependencies
- **Resource Manifest**: The resources your service needs

SO executes in a container
SO manages according to lifecycle

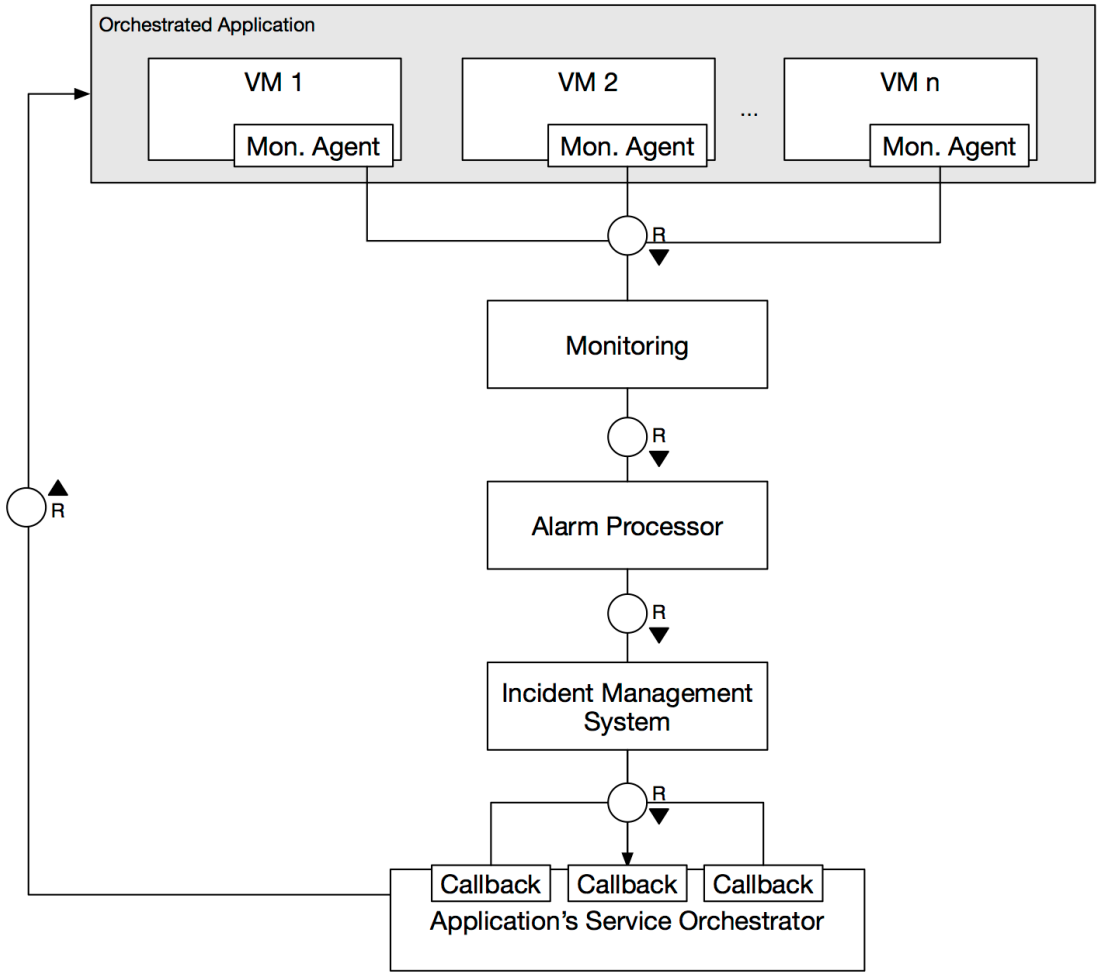
composition

lifecycle allows for parallel deployment



reliability

- SMs, SO, CC are stateless processes, backed by keyval stores
- Monitoring and scaling of resources is provided by the CC



In progress

reliability

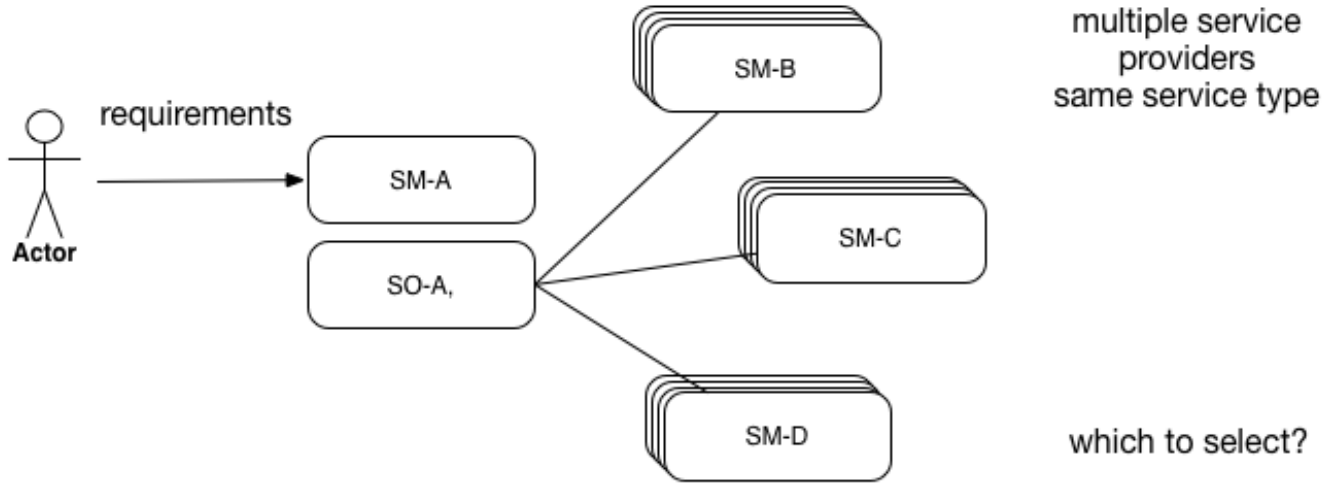
Problem: Upgrade a running distributed application without stopping it?

Updating the graph requires:

- safe service routing
- possibly state transfer
- correct replacement service

In progress

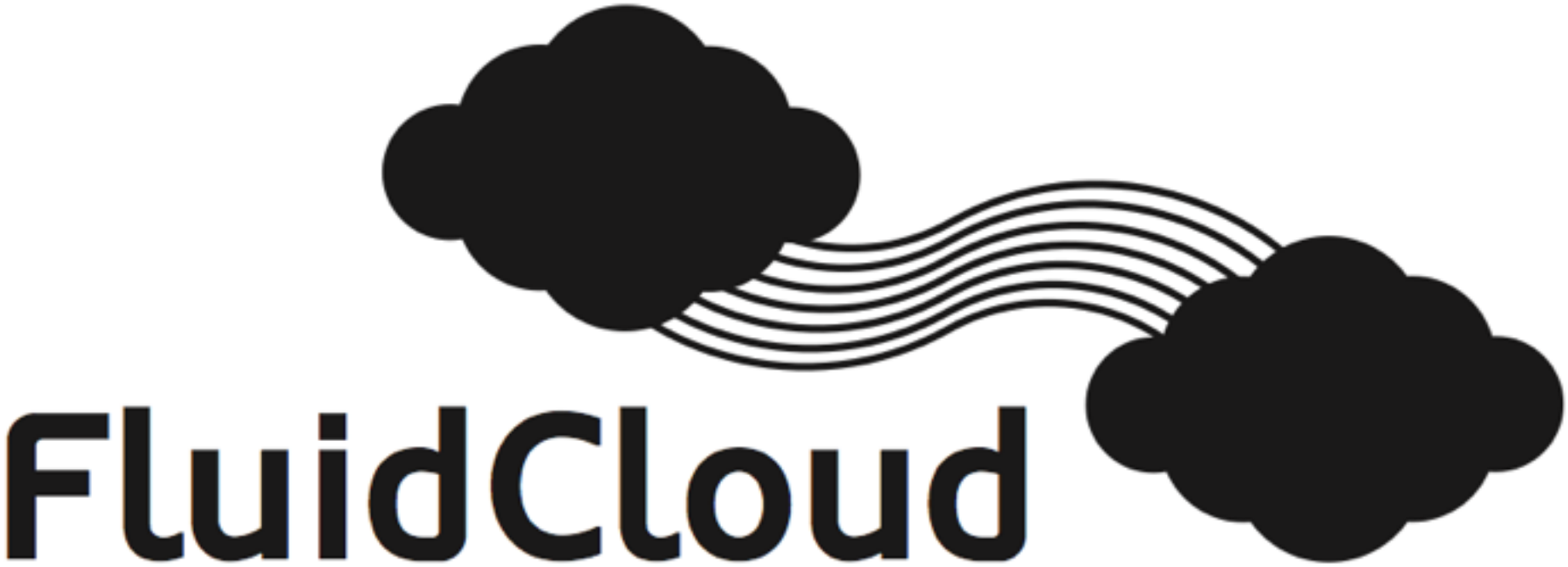
(re)placement



Problem: right service…

- Multi-parameter selection…

very much the placement needed in FluidCloud...

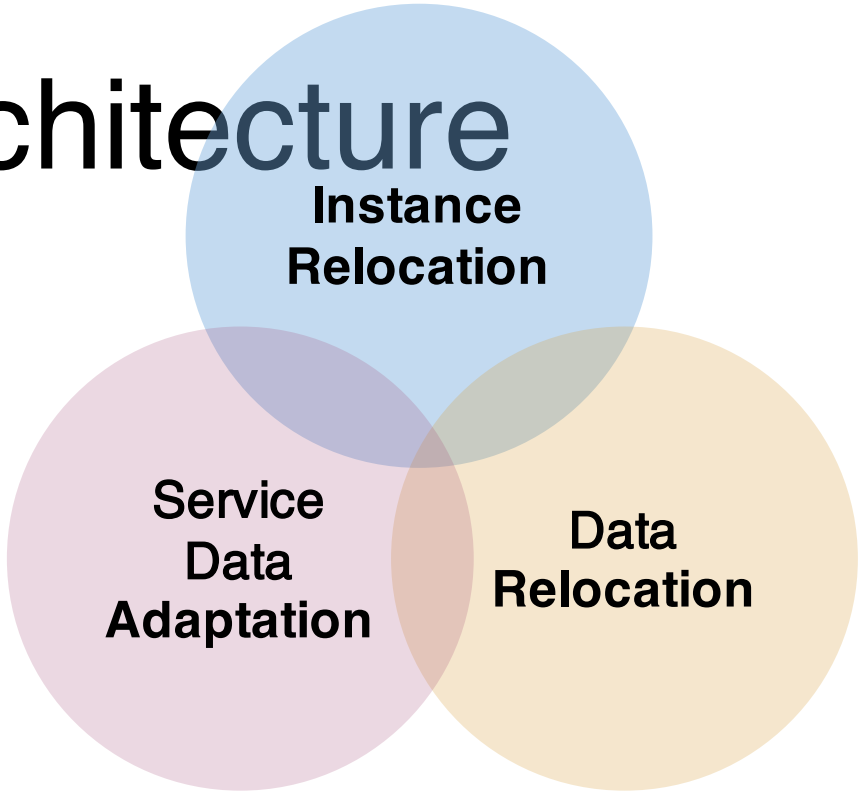


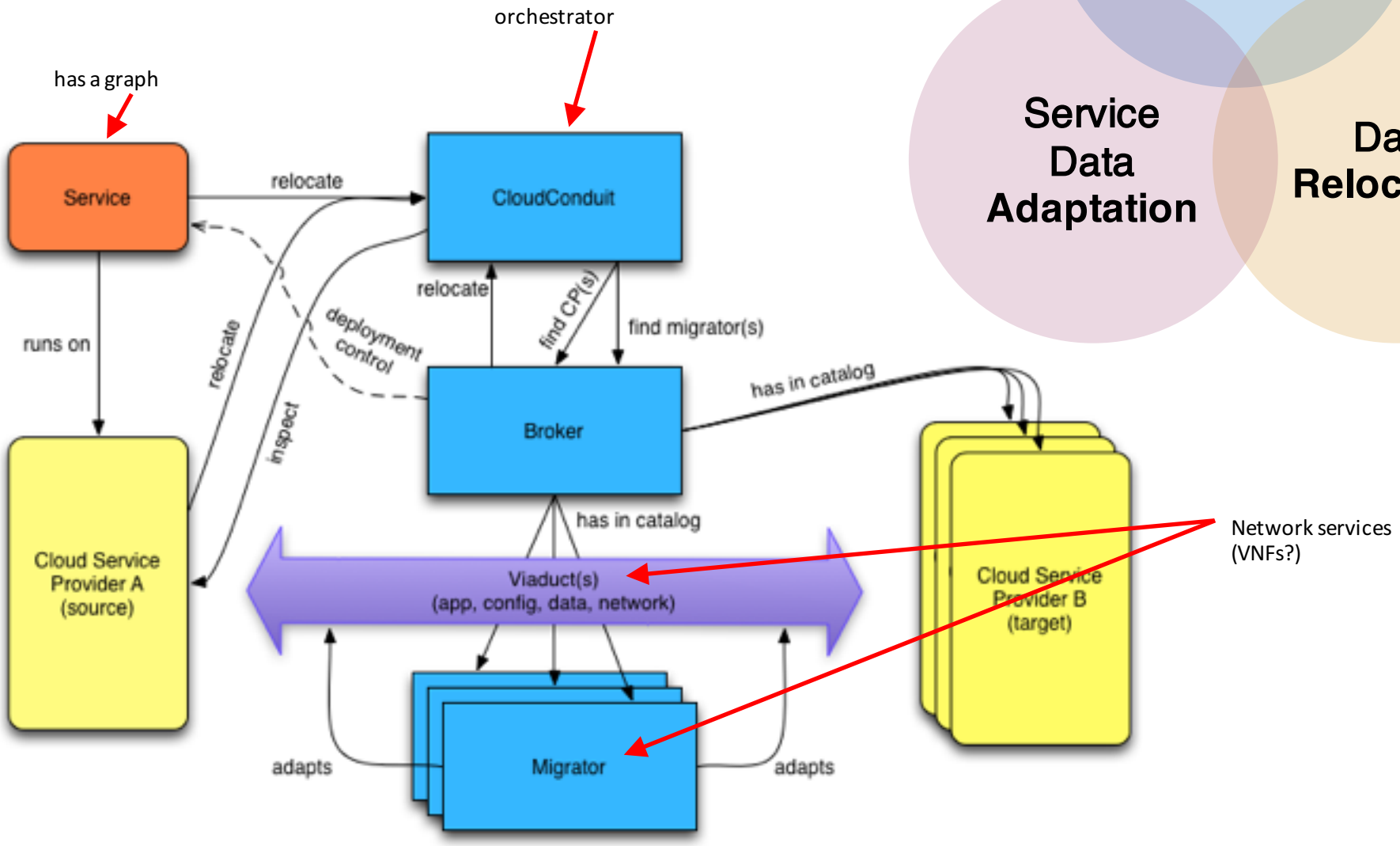
How to intrinsically enable and fully **automate relocation** of service instances between clouds?

ICC LAB

FluidCloud Architecture

read more:
http://blog.zhaw.ch/icclab/fluidcloud-presented-at-usenix/





Orchestration software now released

www.hurtle.it

Thanks!

Questions?

Backup

The network differs from the computing environment in 2 key factors...

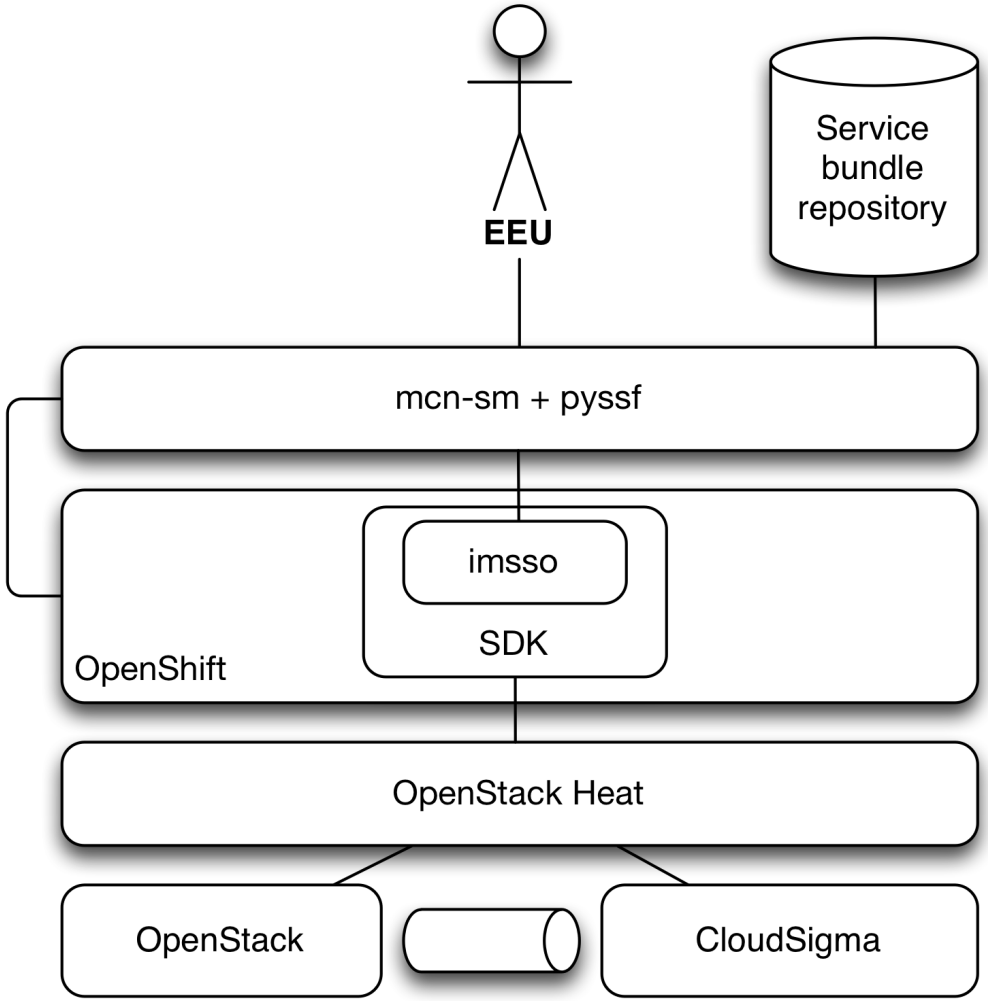
1. **Data plane workloads** (which are huge!) → HIGH PRESSURE ON **PERFORMANCE**
2. **Network requires shape** (+ E2E interconnection) → GLOBAL NETWORK VIEW IS REQUIRED FOR **MANAGEMENT**

...which are big challenges for vanilla cloud computing.

AN ADAPTED VIRTUALISATION ENVIRONMENT IS NEEDED
TO OBTAIN **CARRIER-CLASS** BEHAVIOUR

CLOUD COMPUTING	NFV
1. PERFORMANCE BOUND TO CPU	1. PERFORMANCE BOUND TO I/O & MEMORY ACCESS
2. AGGREGATED VIEW OF RESOURCES (CPU, memory, etc.)	2. NUMA VIEW Internal architecture is relevant for guests
3. ENDPOINTS Applications need the OS	3. MIDDLEPOINTS Data-plane network functions bypass the OS
4. NODE-CENTRIC Shapeless interconnection	4. NETWORK-CENTRIC The network has a shape
5. MANY AND SMALL VMs	5. FEW AND LARGE VMs

http://www.fi-


EEU
Service bundle repository
mcn-sm + pyssf
imsso
SDK
OpenShift
OpenStack Heat
OpenStack
CloudSigma