OpenStack on the Edge
BoF Session - Boston 2017

Adrien Lebre
Fog Computing / Edge Computing/ Massively Distributed Clouds Working Group

https://etherpad.openstack.org/p/BOS-Fog-Edge-MassivelyDistributed-BoF
(or just go on google and looks for ‘’Fog Edge OpenStack’’)

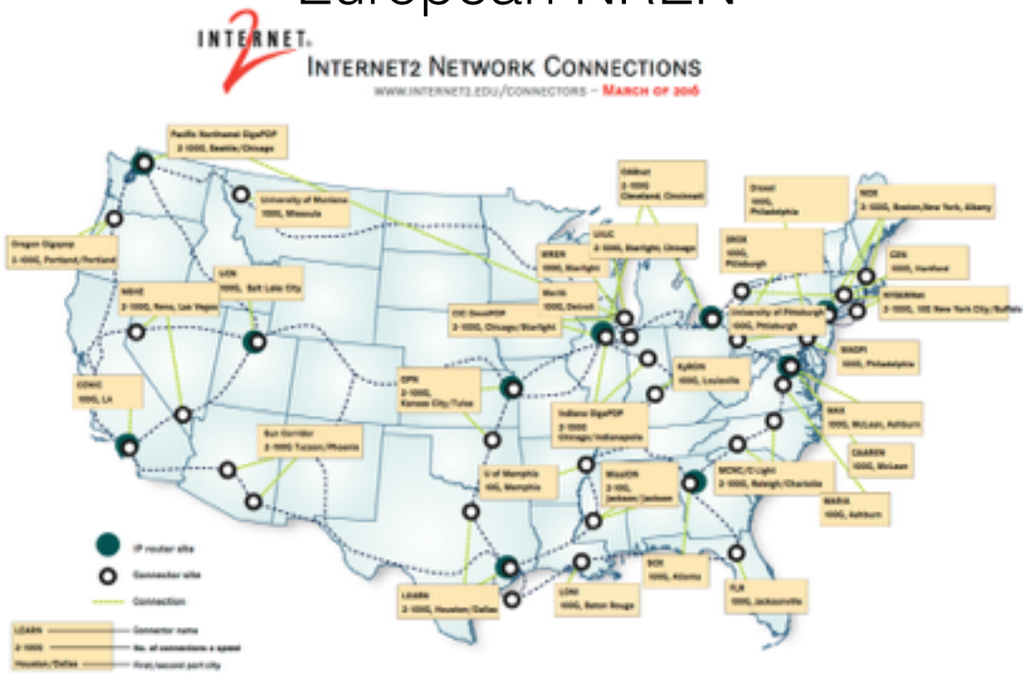
Fog/Edge/MDC Infrastructures

- Leverage network backbones

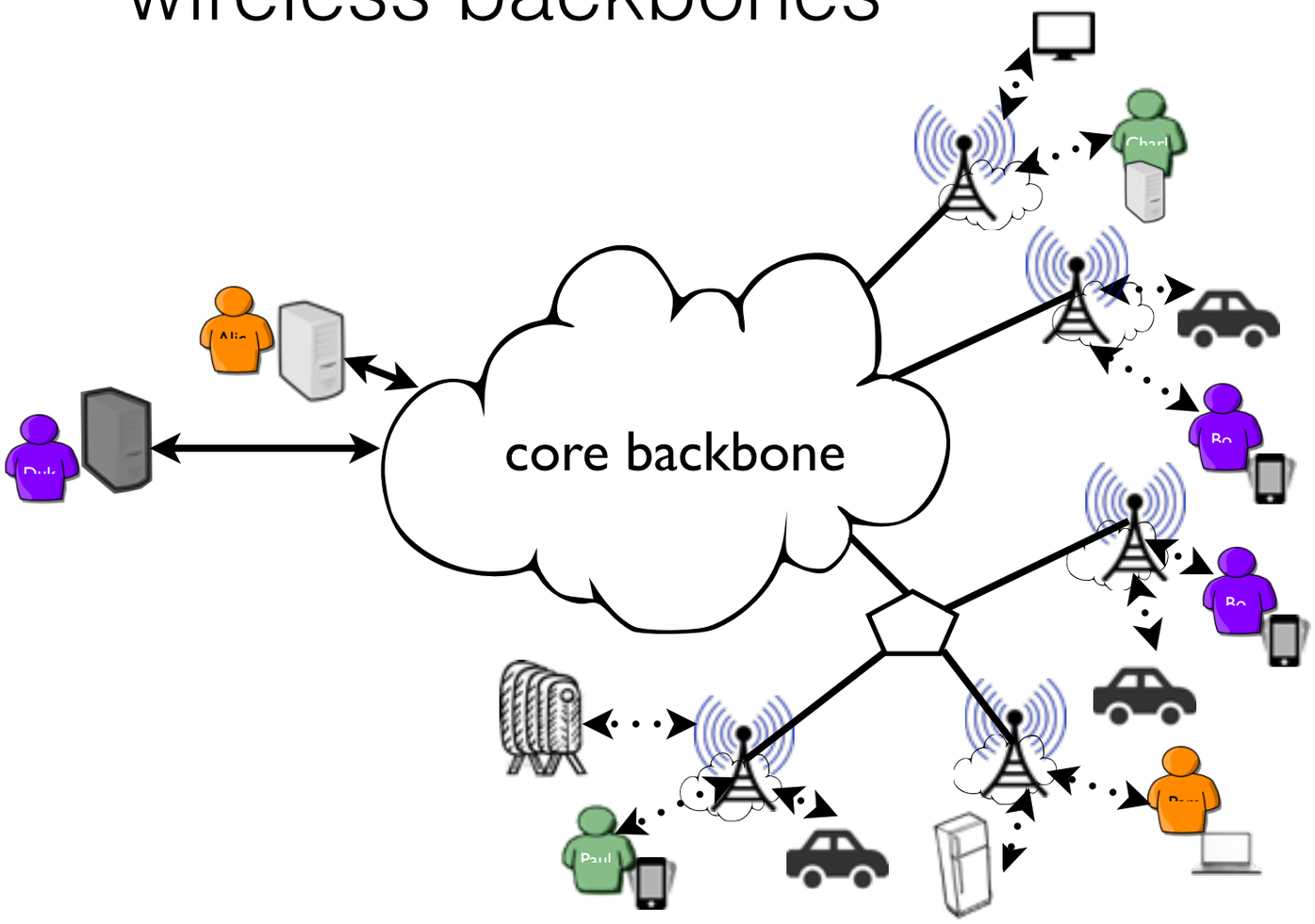
  Extend any point of presence of network backbones (aka PoP) with servers (from network hubs up to major DSLAMs that are operated by telecom companies, network institutions…).

- Extend to the edge by including wireless backbones

European NREN



USA NREN



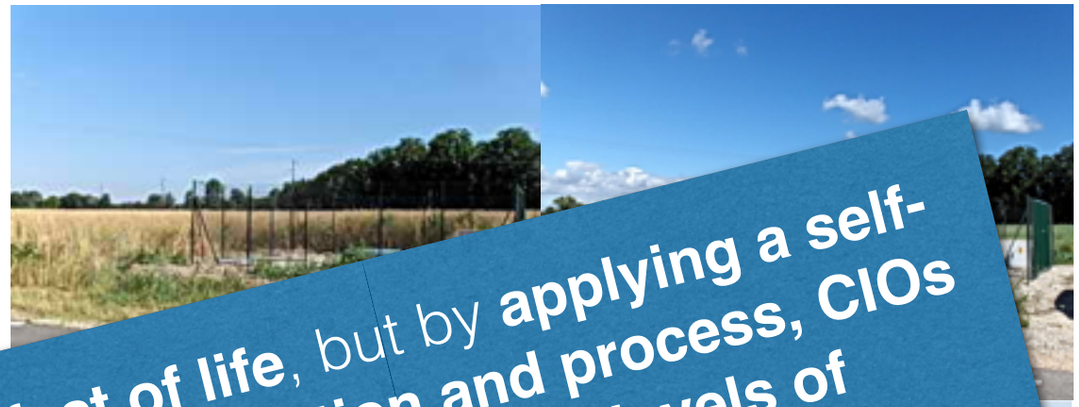
Micro/Nano DCs

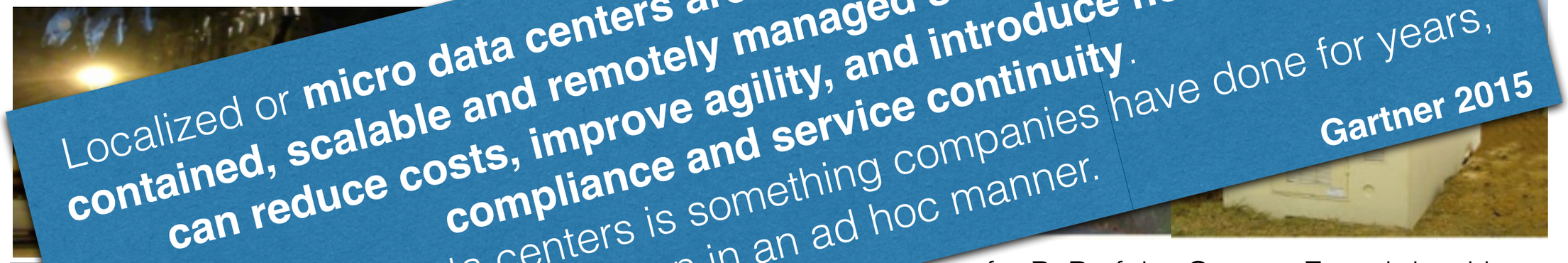
Sagrada Familia microDC
(Barcelona, Spain)

Deployment of a PoP of the Orange French backbone

MDC Industry - Brazil

Micro/Nano DCs





Localized or **micro data centers are a fact of life**, but by **applying a self-contained, scalable and remotely managed solution and process, CIOs can reduce costs, improve agility, and introduce new levels of compliance and service continuity**.

Creating micro data centers is something companies have done for years, but often in an ad hoc manner.

Gartner 2015

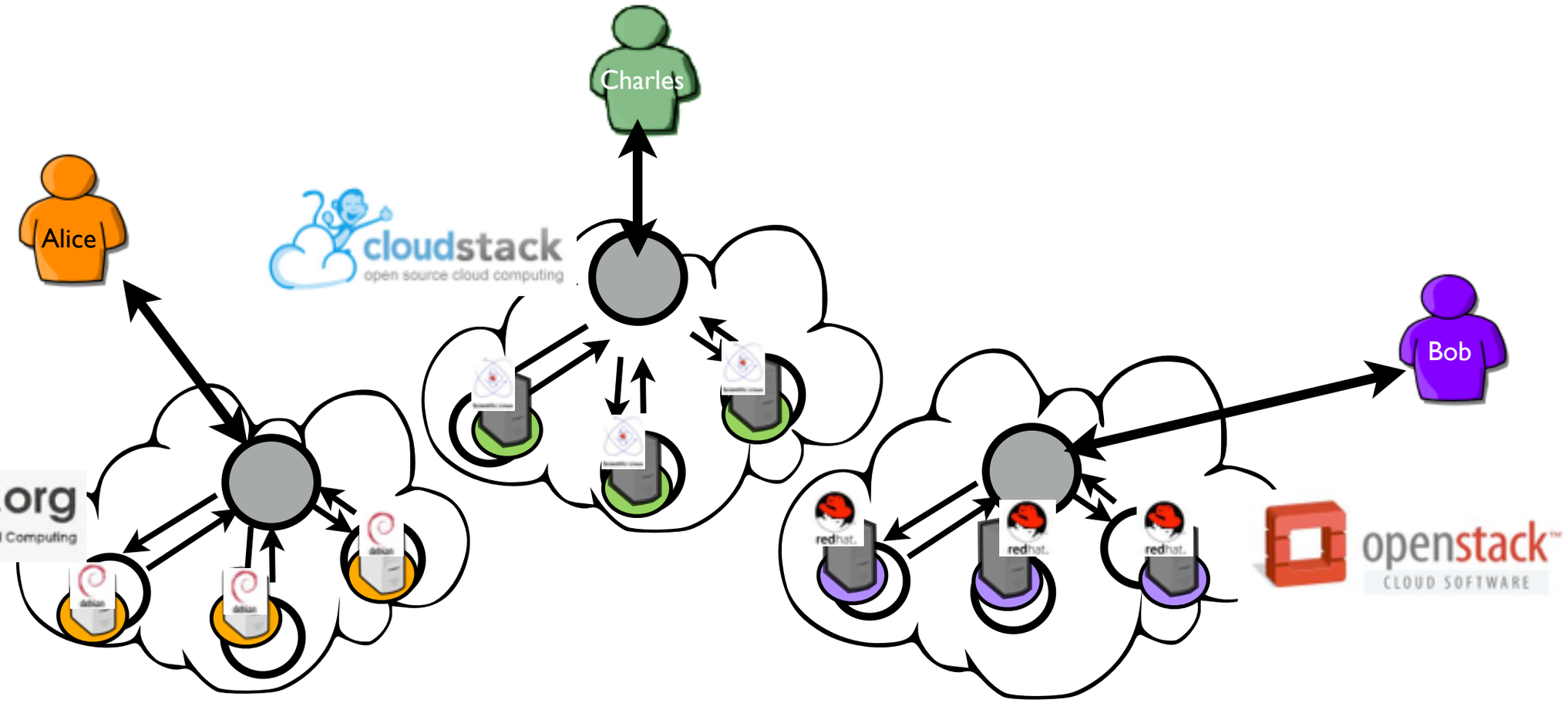
ployment of a PoP of the Orange French backbone

Sagrada Familia microDC
(Barcelona, Spain)

MDC Industry - Brazil

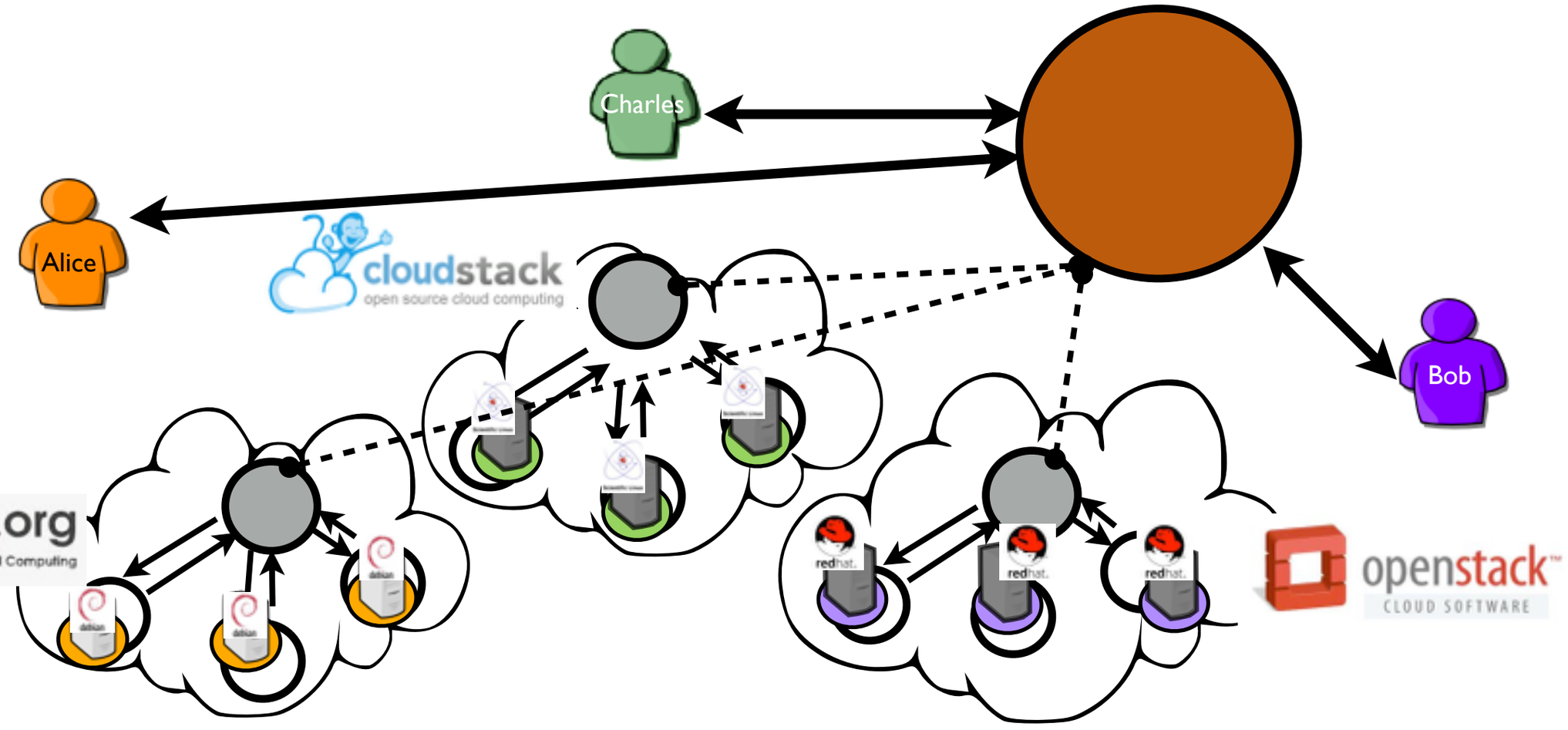
What's about Brokering Approaches?

- Sporadic (hybrid computing/cloud bursting) almost ready for production
- While standards are coming (OCCI…), current brokers are rather limited to simple usages and not advanced administration operations



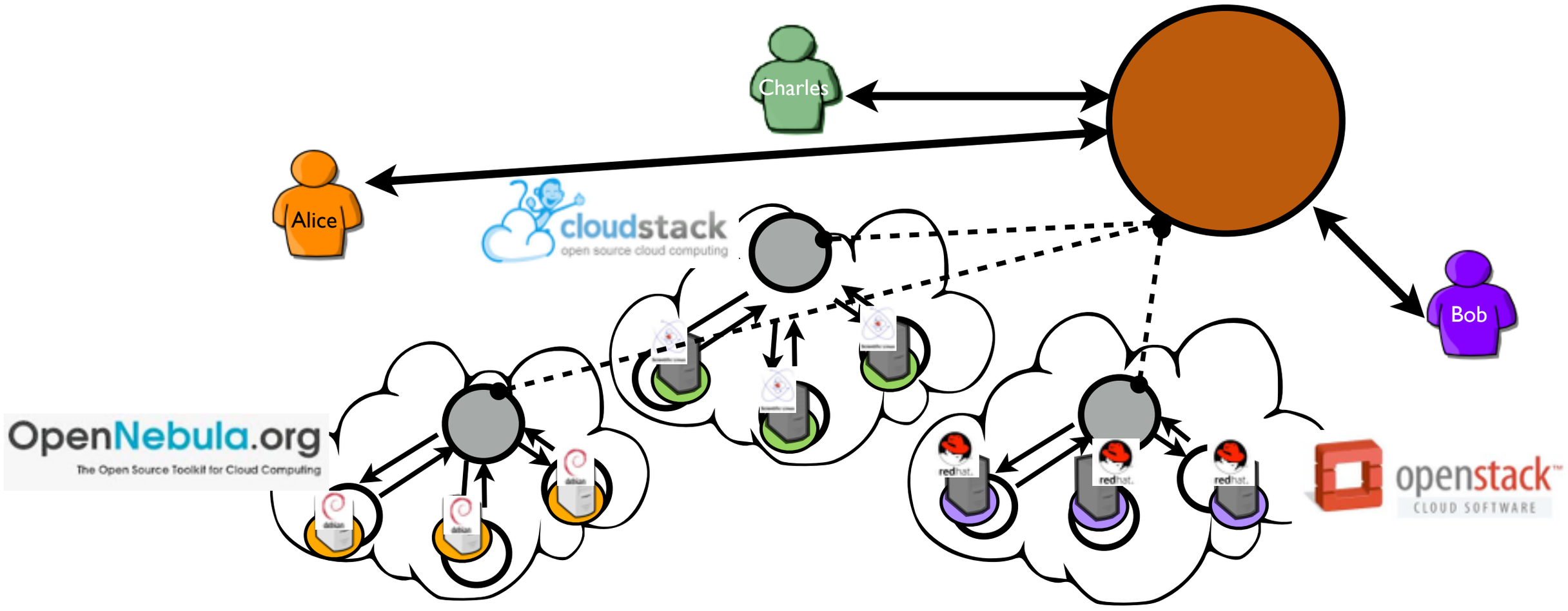
What's about Brokering Approaches?

- Sporadic (hybrid computing/cloud bursting) almost ready for production
- While standards are coming (OCCI…), current brokers are rather limited to simple usages and not advanced administration operations



What's about Brokering Approaches?

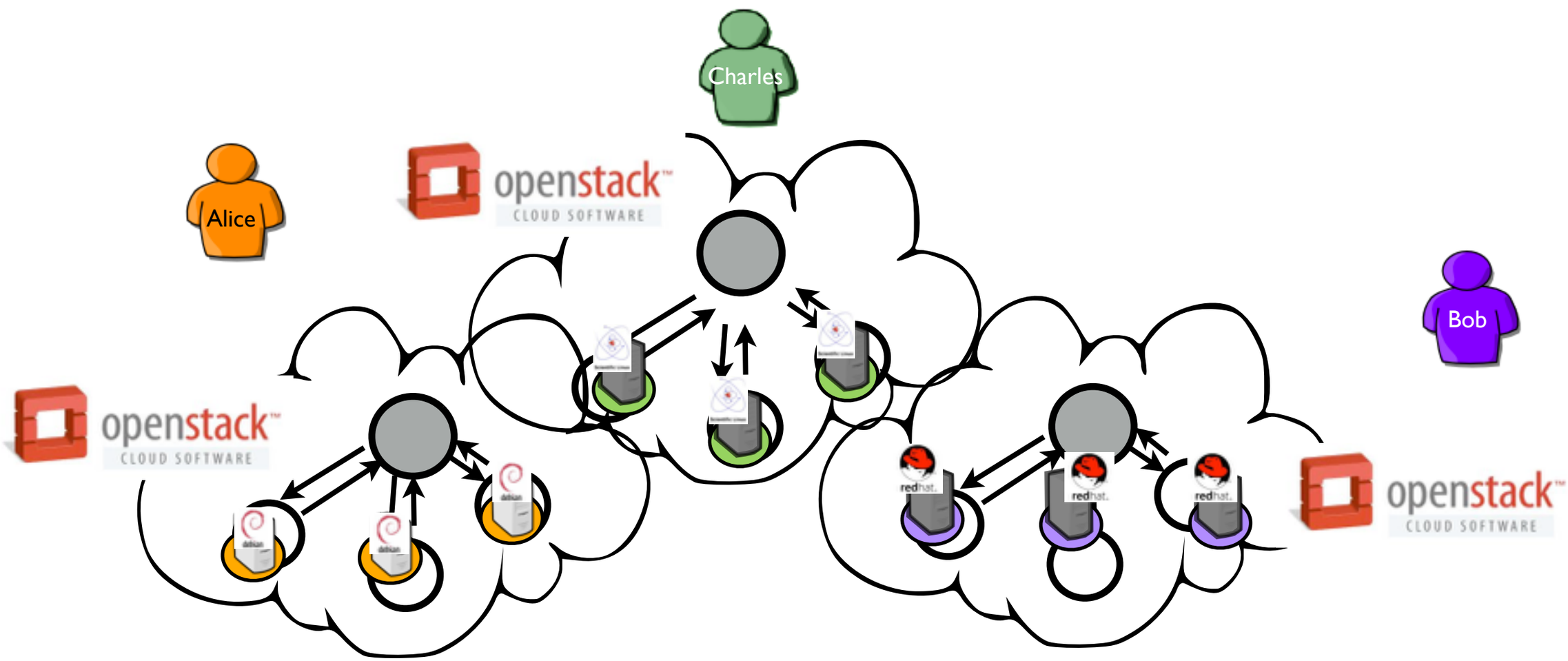
- Sporadic (hybrid computing/cloud bursting) almost ready for production
- While standards are coming (OCCI…), current brokers are rather limited to simple usages and not advanced administration operations



Advanced brokers must reimplement standard IaaS mechanisms while facing the API limitation

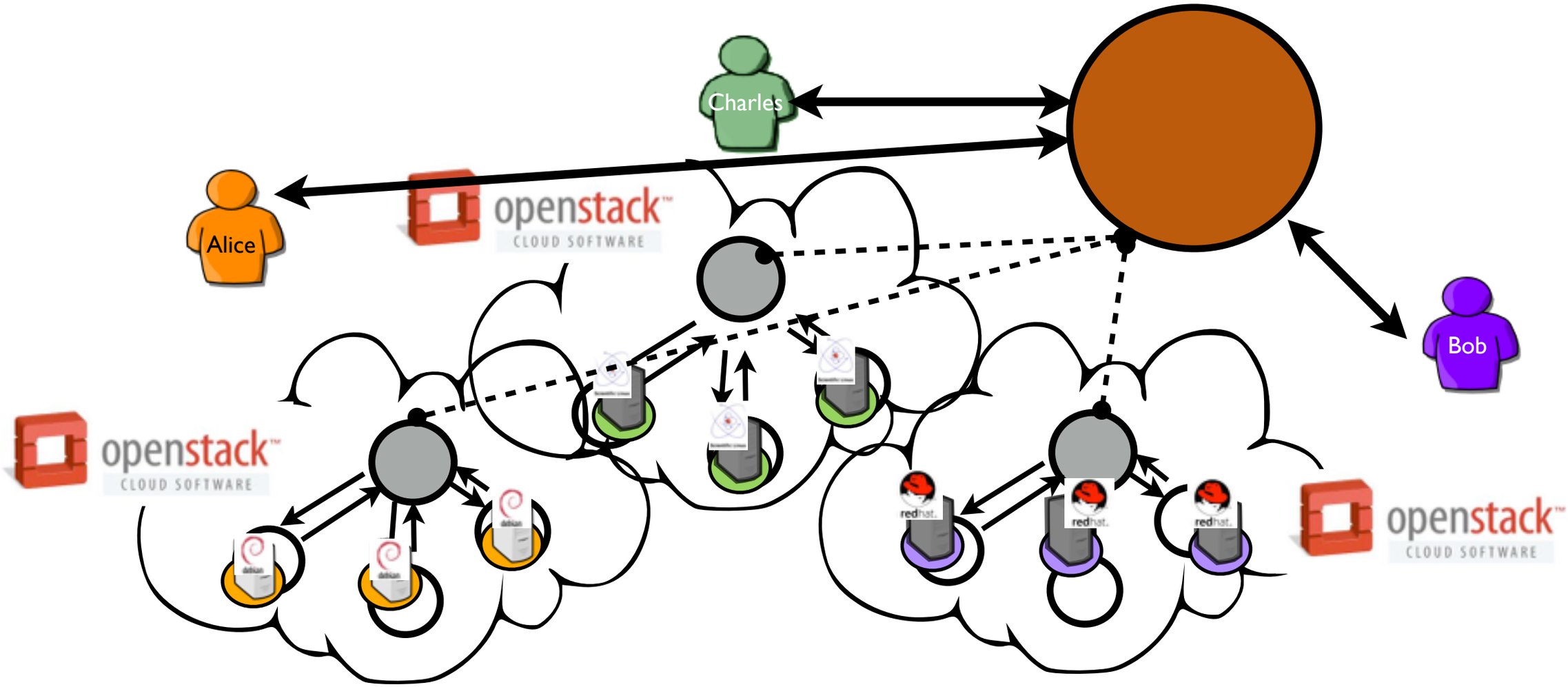
Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late



Would OpenStack be the solution?

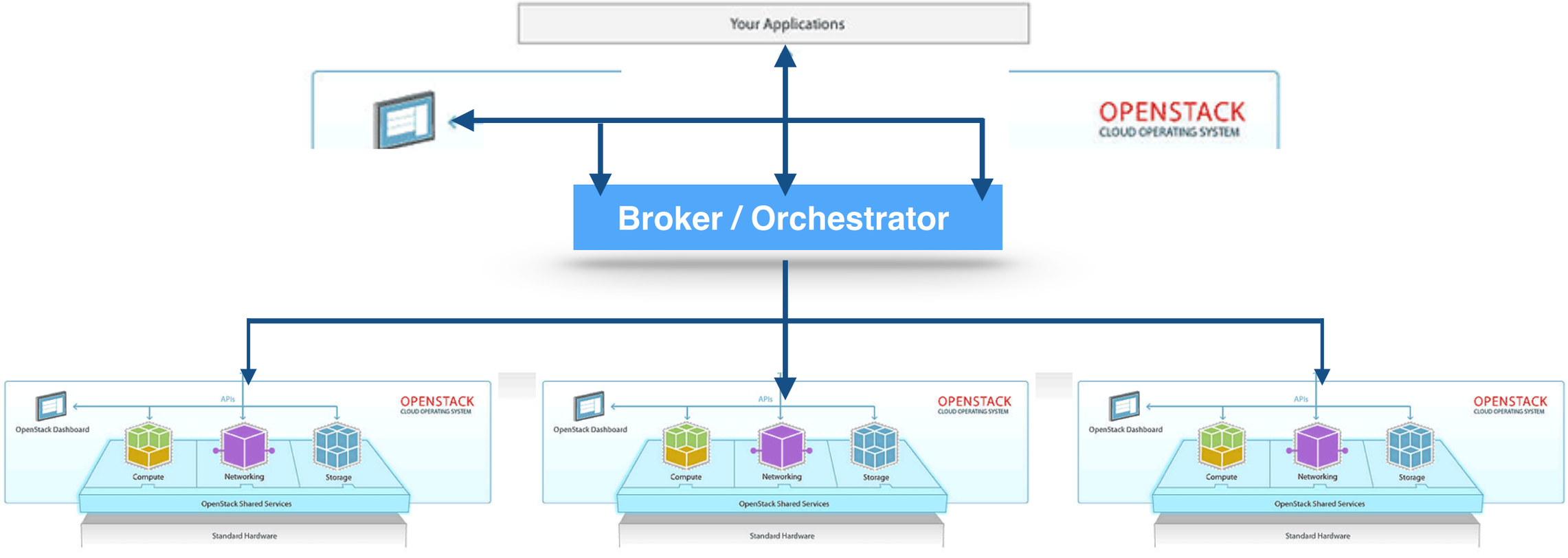
- Do not reinvent the wheel… it is too late



Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

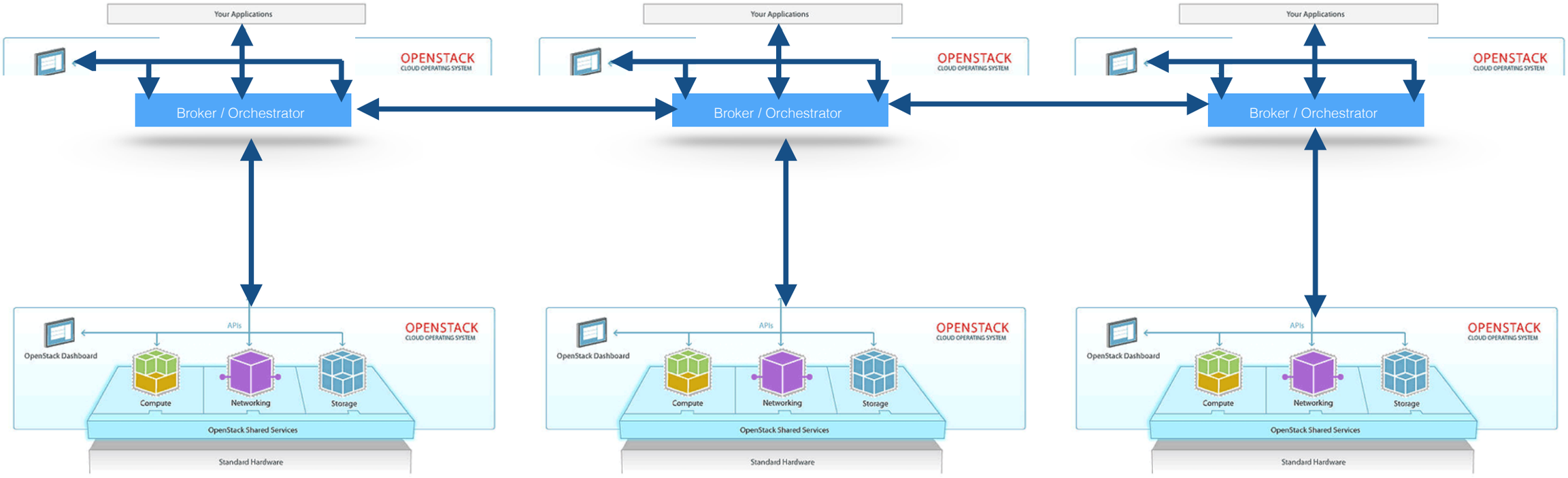
Top/Down: add a substrate to pilot independent OpenStack instances.



Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

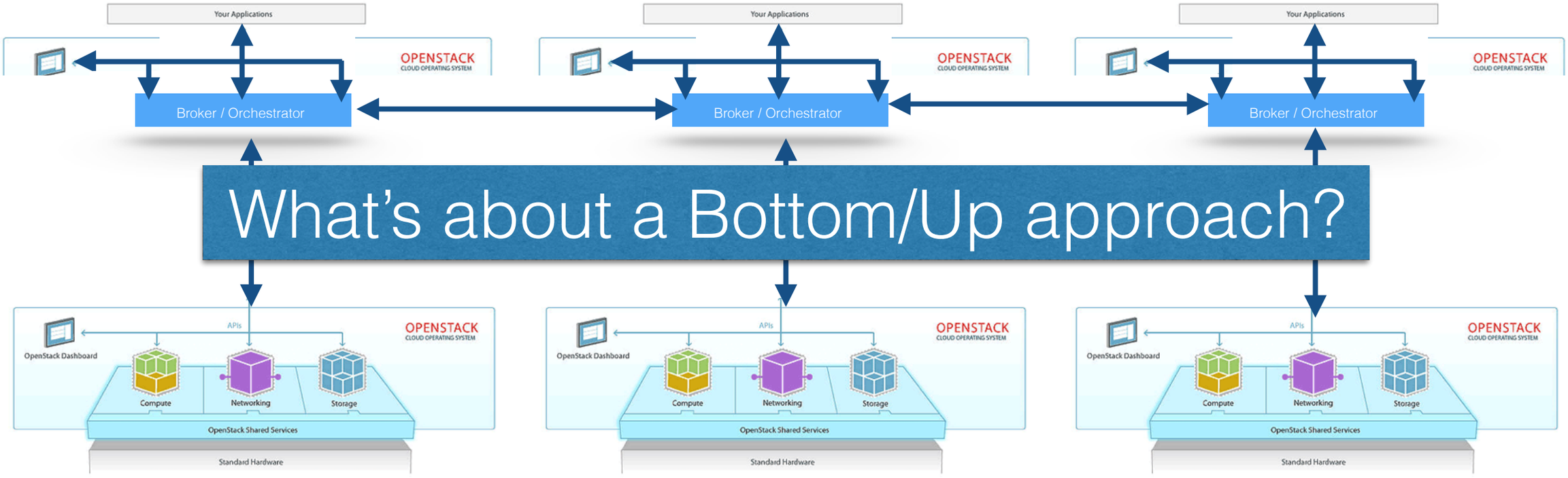
Top/Down: add a substrate to pilot independent OpenStack instances.



Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

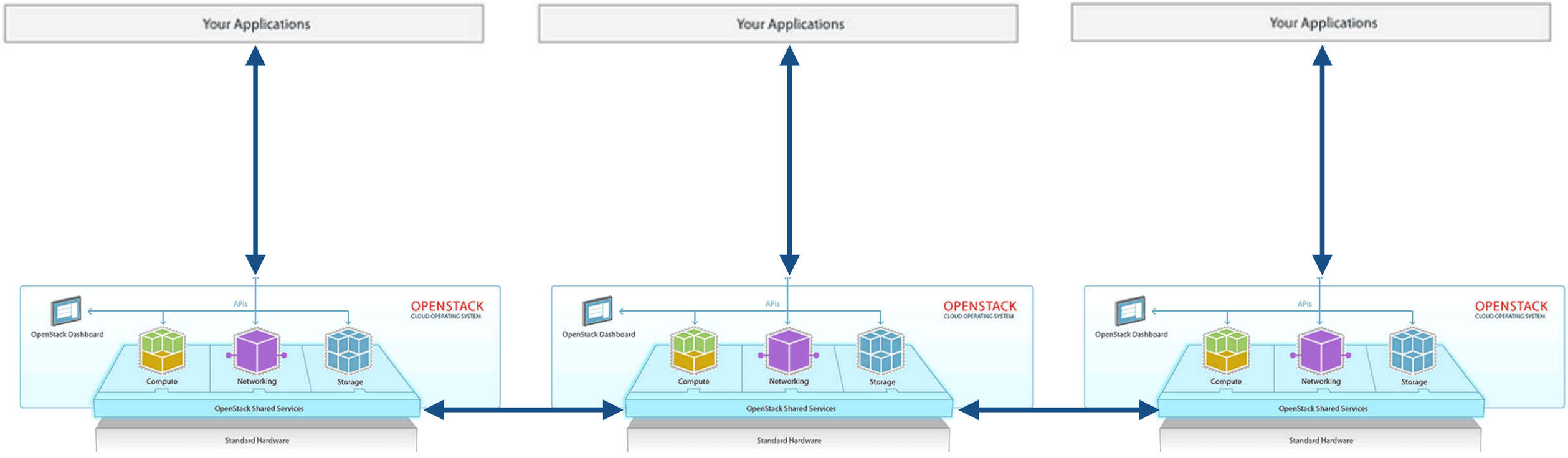
Top/Down: add a substrate to pilot independent OpenStack instances.



Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

Bottom/Up - investigate whether/how OpenStack core services can be cooperative by default using Self* and P2P mechanisms

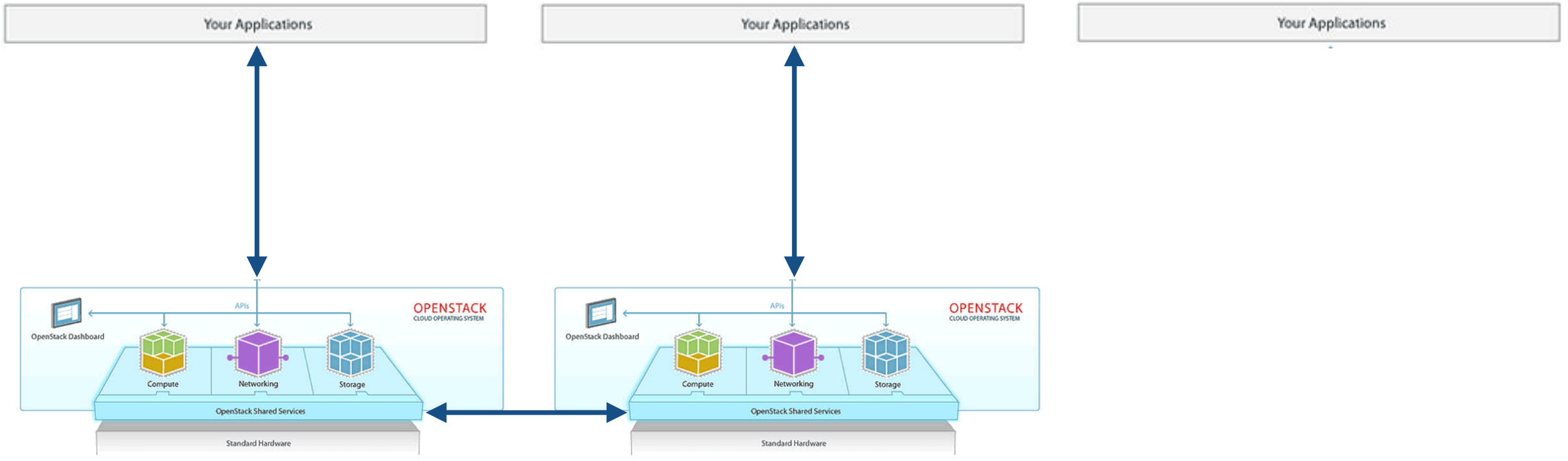


Natively distributed/cooperative

Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

Bottom/Up - investigate whether/how OpenStack core services can be cooperative by default using Self* and P2P mechanisms

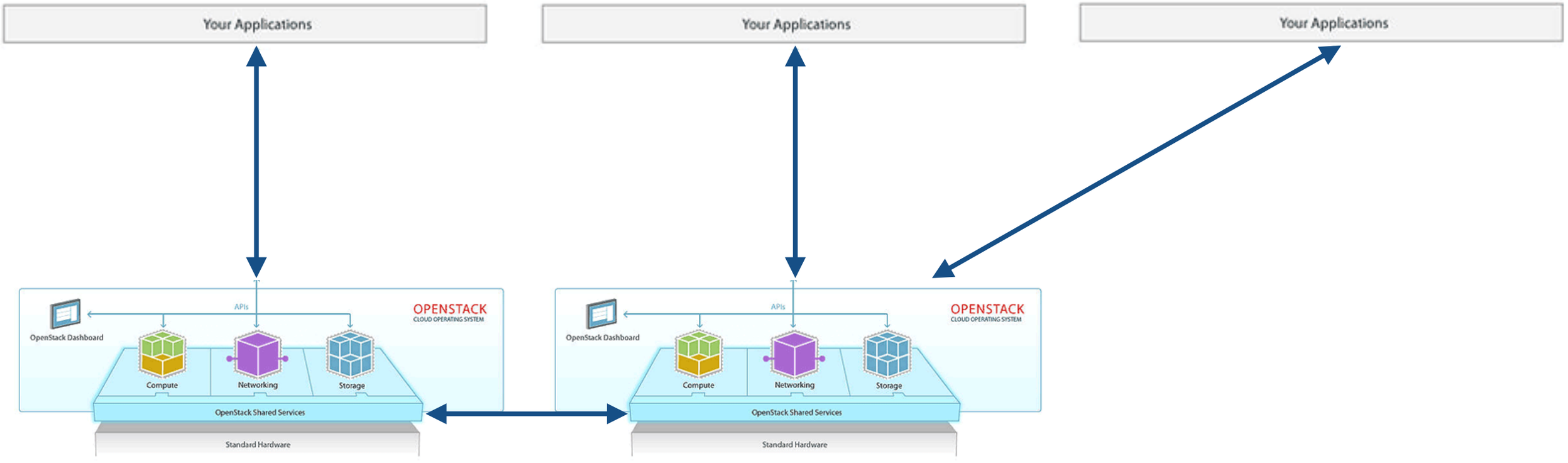


Natively distributed/cooperative

Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

Bottom/Up - investigate whether/how OpenStack core services can be cooperative by default using Self* and P2P mechanisms

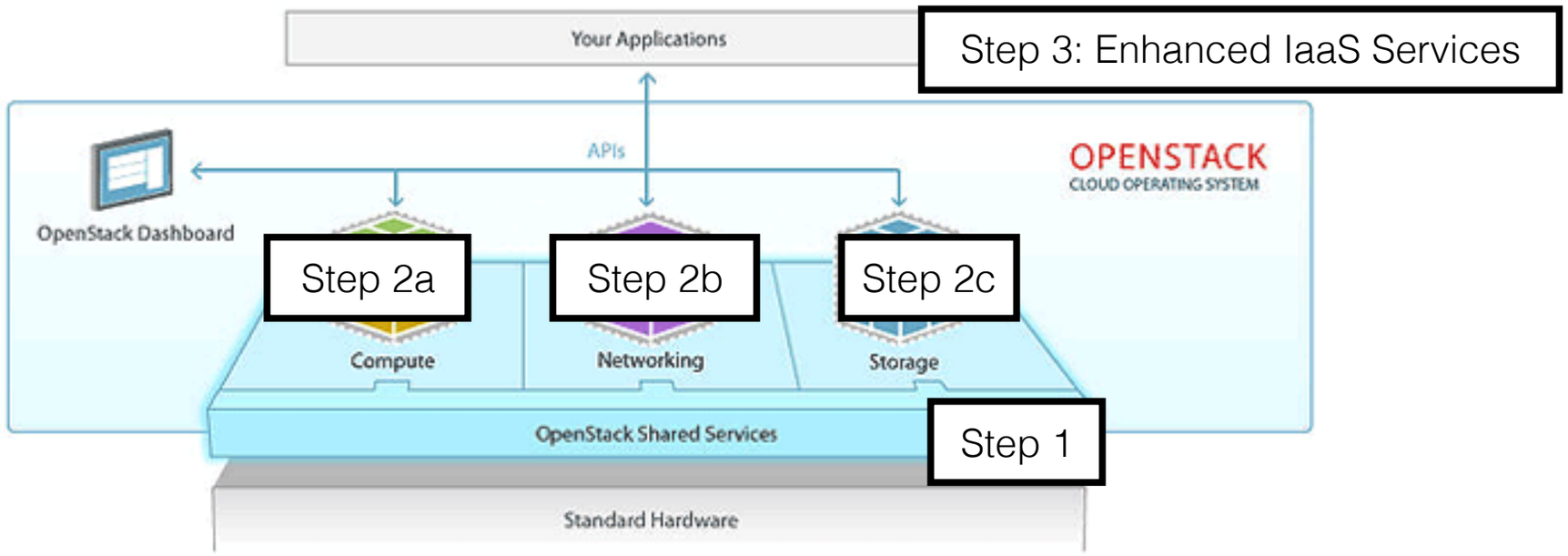


Natively distributed/cooperative

Would OpenStack be the solution?

- Do not reinvent the wheel… it is too late
- Few proposals to federate/operate distinct OpenStack DCs

Bottom/Up - investigate whether/how OpenStack core services can be cooperative by default using Self* and P2P mechanisms



Looking back to the Future

- Austin Summit - May 2016 - Nova PoC

openstack CLOUD SOFTWARE

Technical considerations

SIDEBAR ◂ PREV | UP | NEXT ▸

CONTENTS SEARCH

OPENSTACK MANUALS › OPENSTACK ARCHITECTURE DESIGN GUIDE - CURRENT

Technical considerations

Repurposing an existing OpenStack environment to be massively scalable is a formidable task. When building a massively scalable environment from the ground up, ensure you build the initial deployment with the same principles and choices that apply as the environment grows. For example, a good approach is to deploy the first site as a multi-site environment. This enables you to use the same deployment and segregation methods as the environment grows to separate locations across dedicated links or wide area networks. In a hyperscale cloud, scale trumps redundancy. Modify applications with this in mind, relying on the scale and homogeneity of the environment to provide reliability rather than redundant infrastructure provided by non-commodity hardware solutions.

Infrastructure segregation

OpenStack services support massive horizontal scale. Be aware that this is not the case for the entire supporting infrastructure. This is particularly a problem for the database management systems and message queues that OpenStack services use for data storage and remote procedure call communications.

Traditional clustering techniques typically provide high availability and some additional scale for these environments. In the quest for massive scale, however, you must take additional steps to relieve the performance pressure on these components in order to prevent them from negatively impacting the overall performance of the environment. Ensure that all the components are in balance so that if the massively scalable environment fails, all the components are near maximum capacity and a single component is not causing the failure.

Looking back to the Future

Technical considerations

SIDEBAR ◂ PREV | UP | NEXT ▸

OPENSTACK MANUALS ▸ OPENSTACK ARCHITECTURE DESIGN GUIDE - CURRENT

Technical considerations

Repurposing an existing OpenStack environment to be massively scalable is a formidable task. When building a massively scalable environment from the ground up, ensure you build the initial deployment with the same principles and choices that apply as the environment grows. For example, a good approach is to deploy the first site as a multi-site environment. This enables you to use the same deployment and segregation methods as the environment grows to separate locations across dedicated links or wide area networks. In a hyperscale cloud, scale trumps redundancy. Modify applications with this in mind, relying on the scale and homogeneity of the environment to provide reliability rather than redundant infrastructure provided by non-commodity hardware solutions.

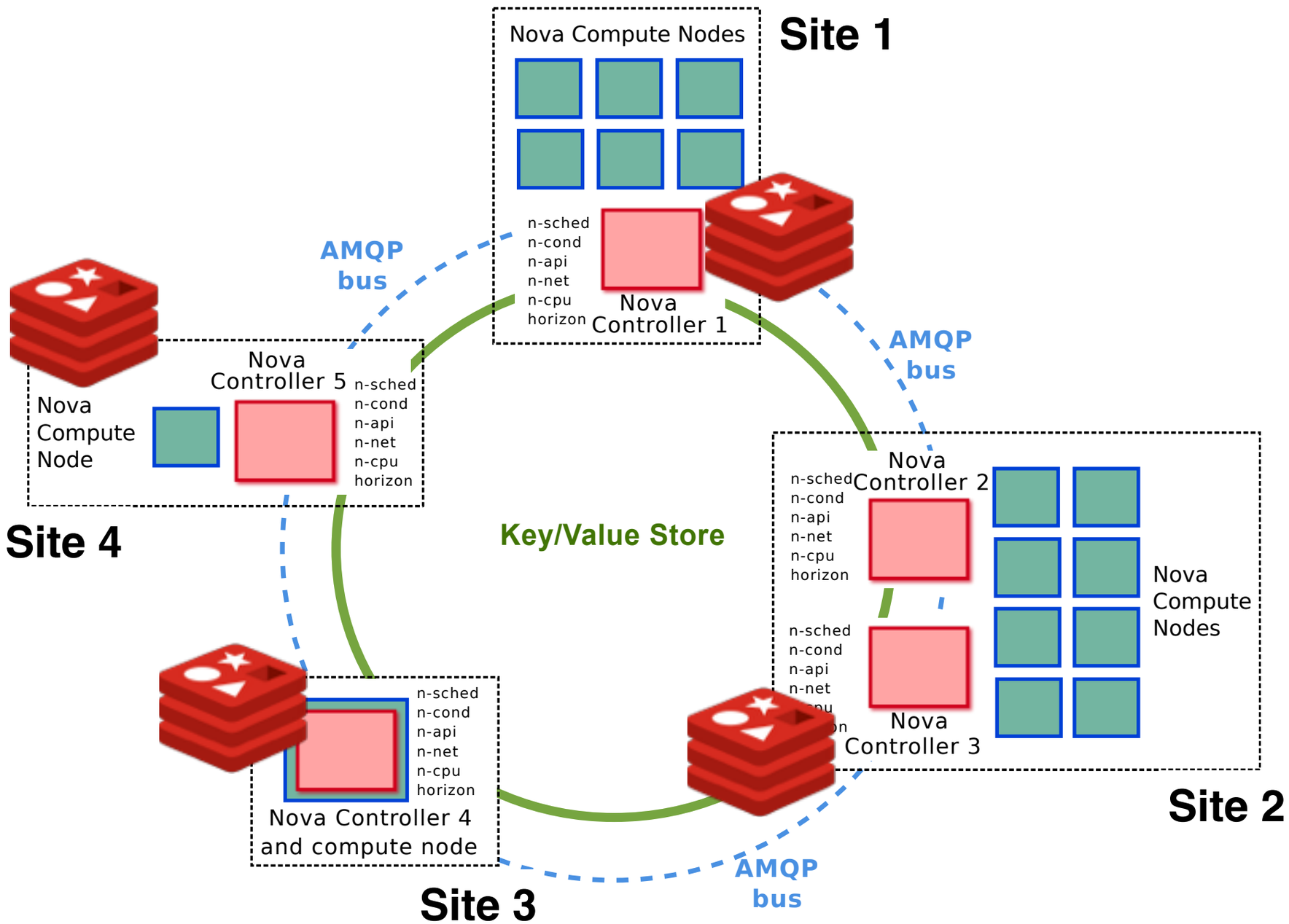
Infrastructure segregation

OpenStack services support massive horizontal scale. Be aware that this is not the case for the entire supporting infrastructure. This is particularly a problem for the database management systems and message queues that OpenStack services use for data storage and remote procedure call communications.

Traditional clustering techniques typically provide high availability and some additional scale for these environments. In the quest for massive scale, however, you must take additional steps to relieve the performance pressure on these components in order to prevent them from negatively impacting the overall performance of the environment. Ensure that all the components are in balance so that if the massively scalable environment fails, all the components are near maximum capacity and a single component is not causing the failure.

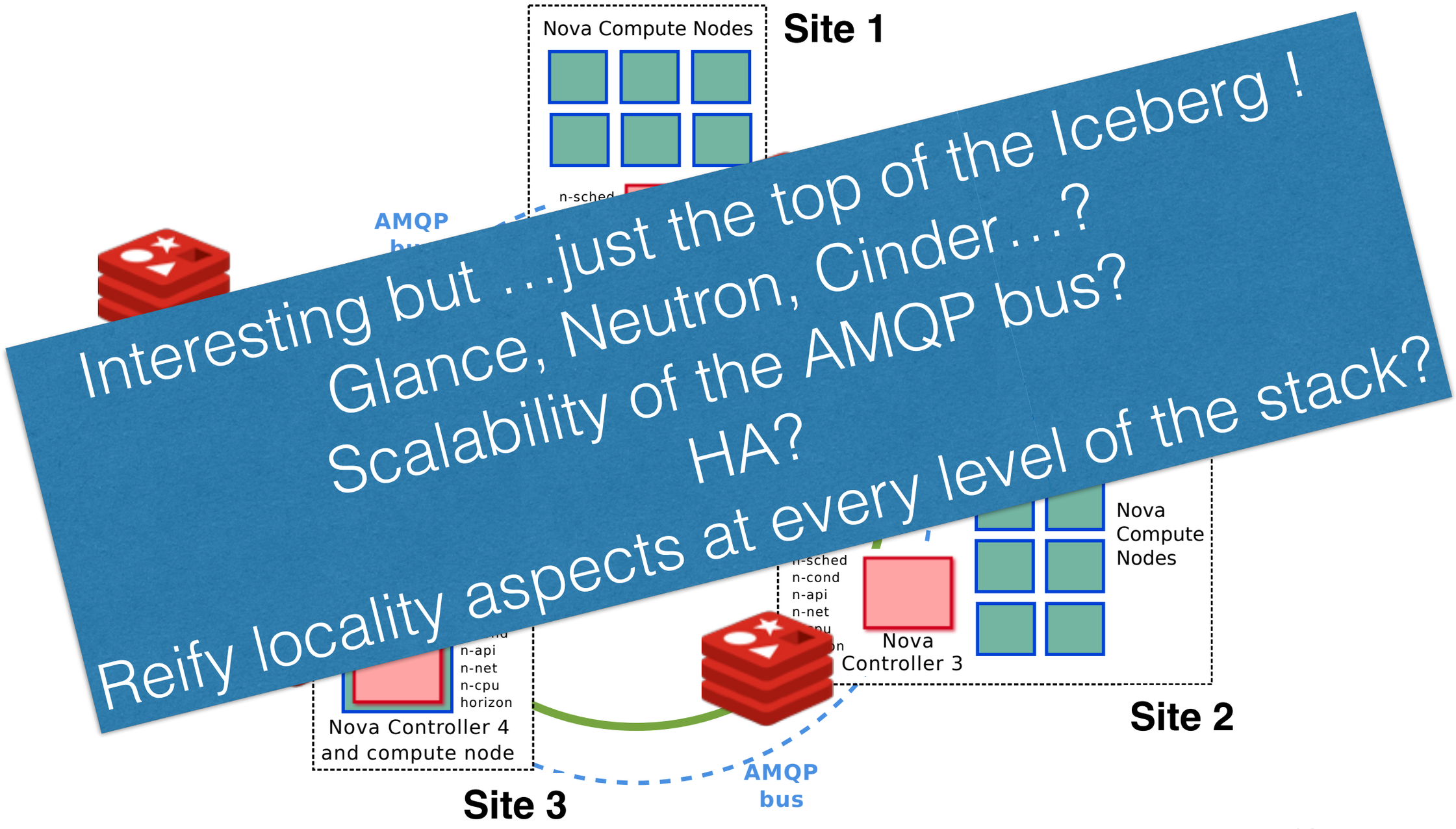
Looking back to the Future

- Austin Summit - May 2016 - Nova PoC (based on Juno)
  Replaced MySQL DB by Reddis (NoSQL backend)



Looking back to the Future

- Austin Summit - May 2016 - Nova PoC (based on Juno)
  Replaced MySQL DB by Reddis (NoSQL backend)



Takeaway message

- Goal of the WG: do not reinvent the wheel (upstream first).

  Study to what extent current mechanisms can handle Fog/Edge infrastructures

  Propose revisions/extensions of internal mechanisms when appropriate.

  Investigate how should current cloud APIs be extended to take the advantage of the geo-distribution (latency-aware applications…)

- Ongoing action: Analyze OpenStack Performance under the Fog/Edge perspective (scalability, traffic characterisation…) using EnOS (a dedicated framework for conducting performance evaluations of OpenStack)

  Toward Fog, Edge, and NFV Deployments:
  Evaluating OpenStack WANwide
  Wednesday 2:40pm-3:20pm, Hynes CC, Level 3, Ballroom A

- Face-to-face meeting (for current and new WG members)
  Wednesday 5:20pm - 6:00pm, Hynes CC, Level 2, MR 201

Fog Computing / Edge Computing Massively Distributed Clouds Working Group

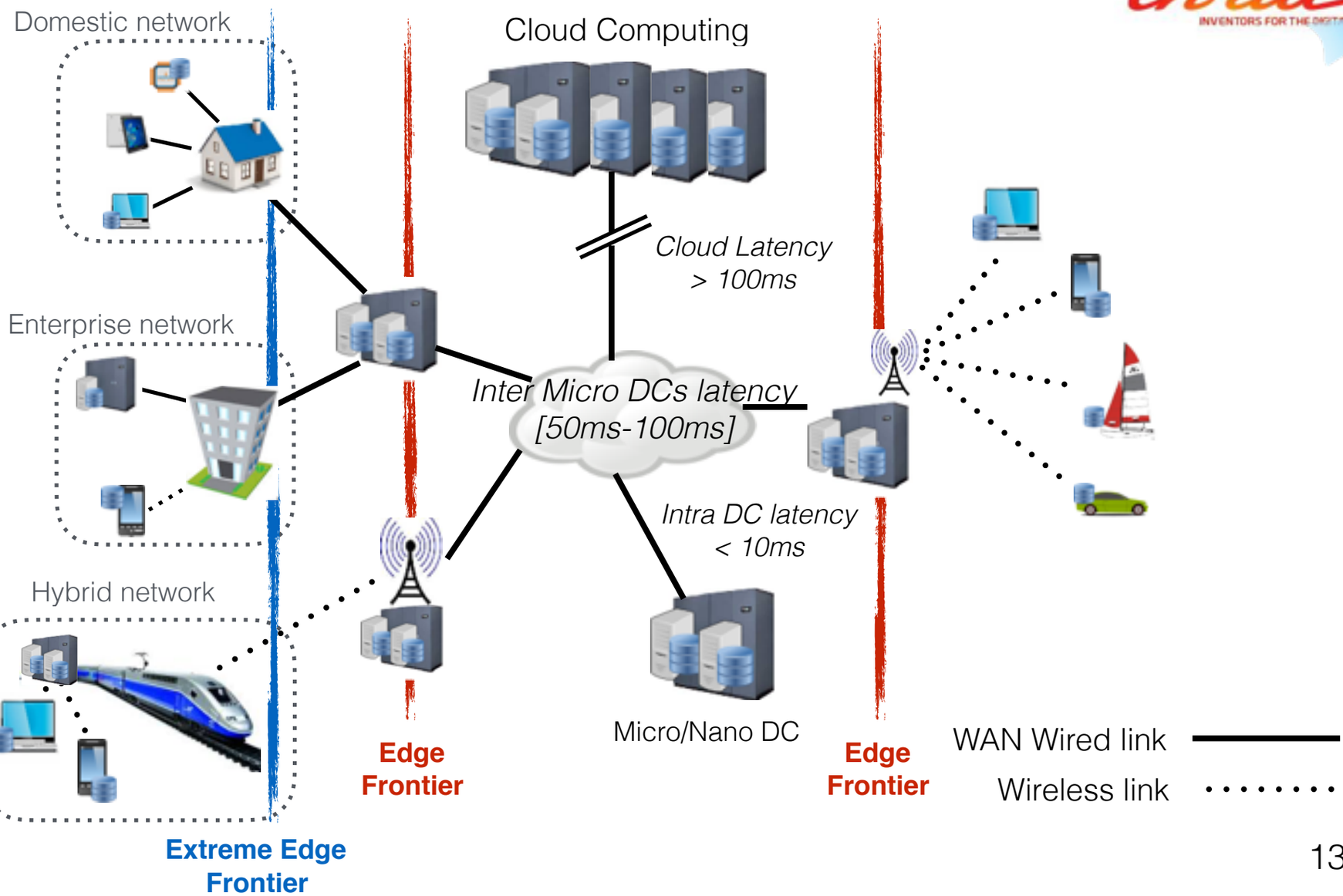
Adrien Lebre

BoF Session - OpenStack Summit - Bos

https://etherpad.openstack.org/p/BOS-Fog-Edge-MassivelyDistributed-BoF
(or just go on google and looks for ‘’Fog Edge OpenStack’’)

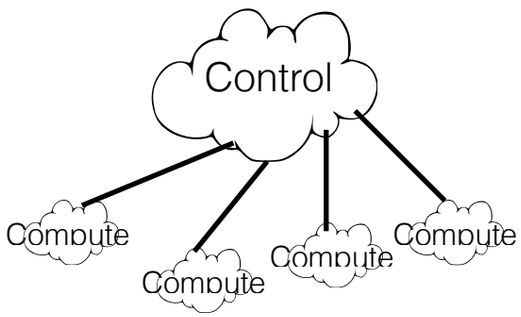
Ongoing Action

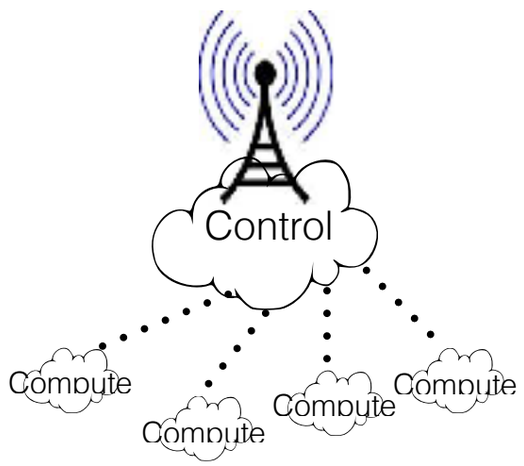
- Collaboration with the Performance Team to understand OpenStack Performance (scalability, traffic characterisation…)
- EnOS: Experimental Environment for Conducting performance evaluations of OpenStack

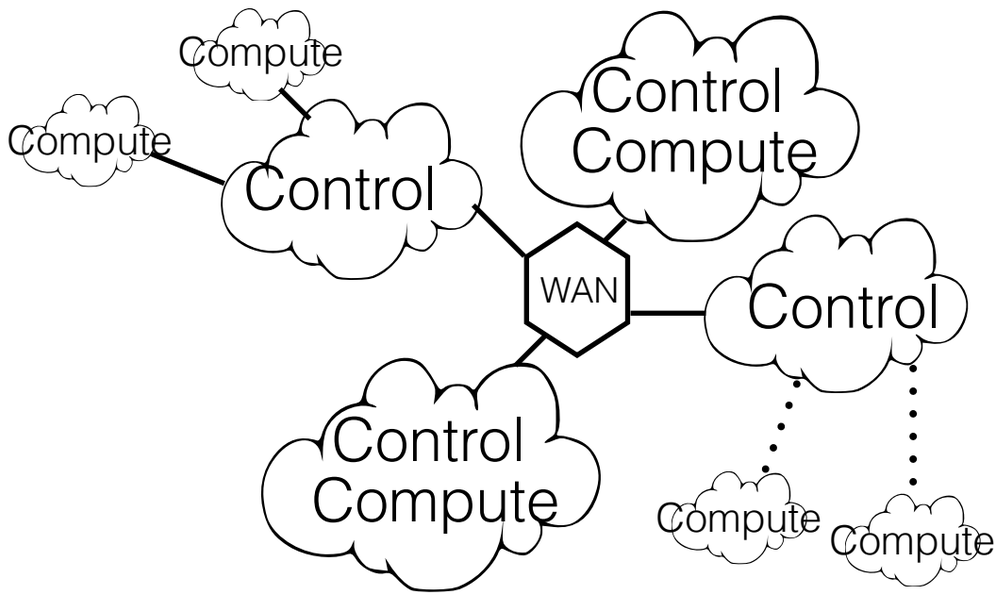


Ongoing Action

- Ongoing collaboration with the Performance Team to understand OpenStack Performance (scalability, traffic characterisation…)
- EnOS: Experimental Environment for Conducting performance evaluations of OpenStack
- Current focus: placement constraints/opportunities
  how many instances of each service? one global bus? one central Glance? severals? Where should we locate them?…







WAN Wired link ——————

Wireless link ·····