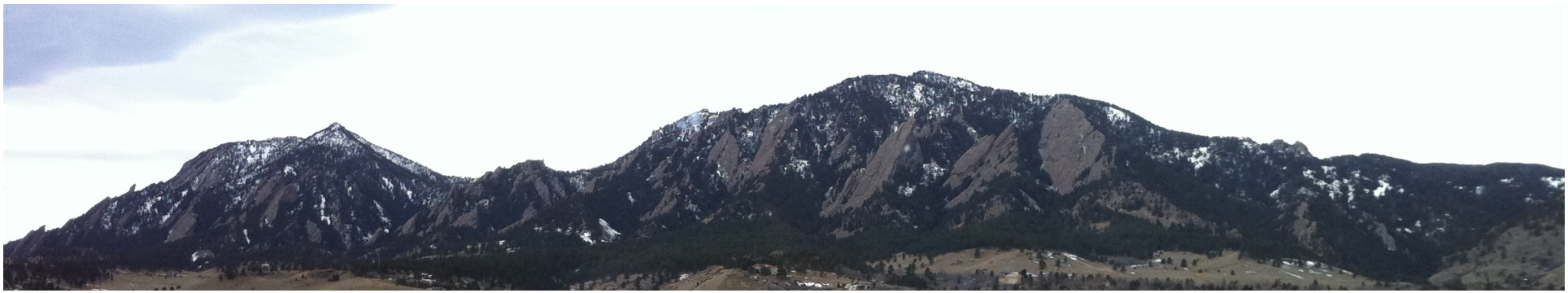
Duke Systems

Foundations of a Future "Inter-Cloud" Architecture

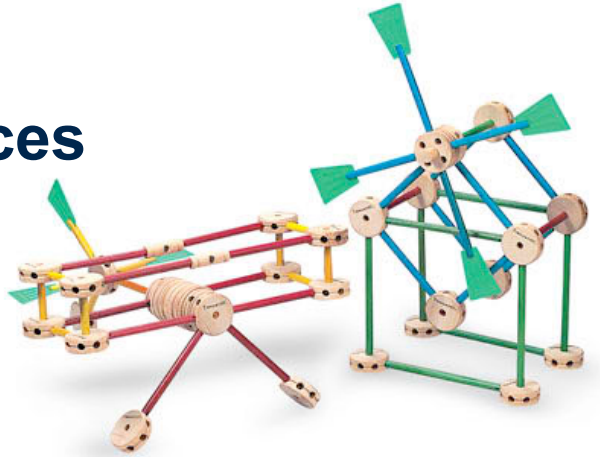
Jeff Chase

Duke University / RENCI



Some challenges

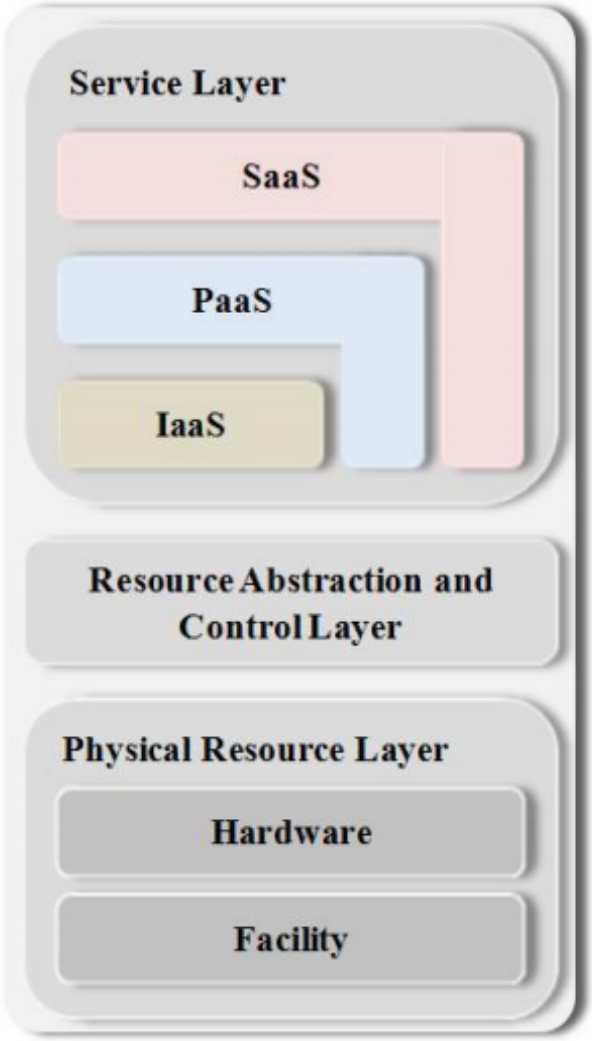
- Cloud infrastructure resources
  - diversity and representation
- Programmability and isolation
  - working at the bottom of the stack
  - configurations and compatibility
- Architecture: elements and combinations
  - assembling virtual infrastructure
  - orchestrating multi-domain services
- Trust



NIST Cloud Computing Reference Architecture

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 500-292



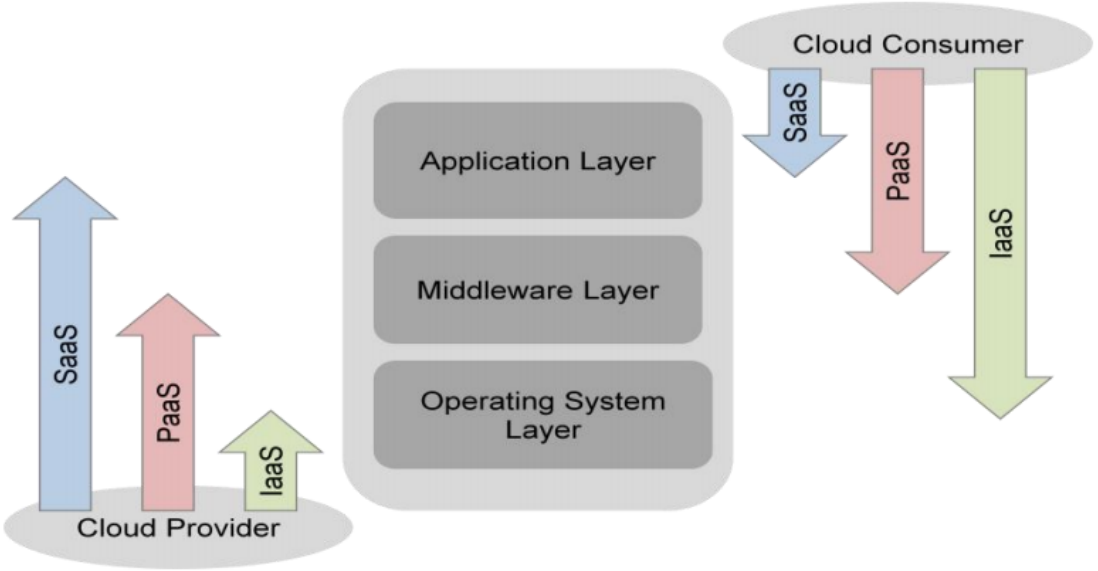


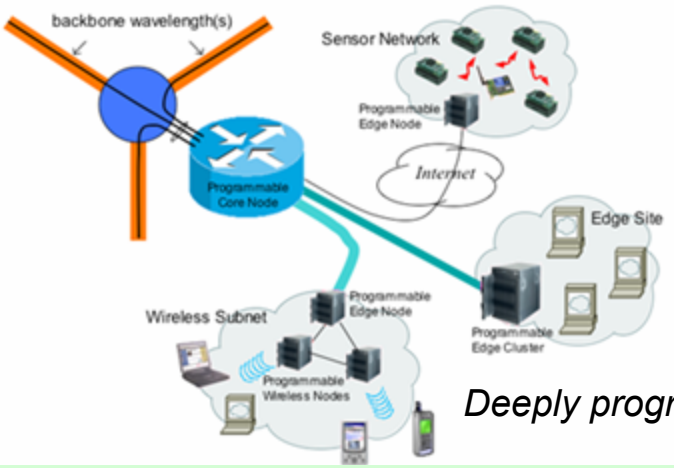
Figure 8: Scope of Controls between Provider and Consumer

Infrastructure as a Service (IaaS)

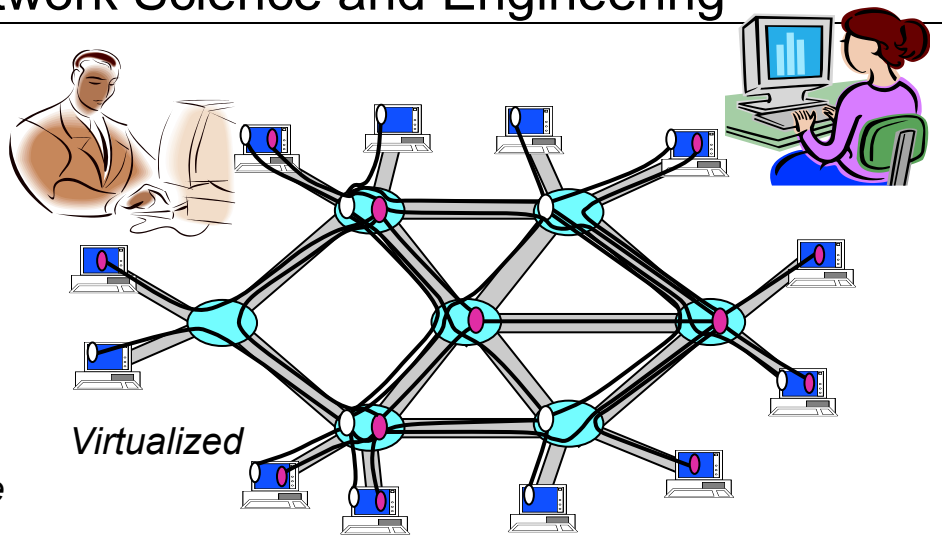
"Consumers of IaaS have access to virtual computers, network-accessible storage, network infrastructure components, and other **fundamental computing resources**…and are billed according to the amount or duration of the resources consumed."

The GENI Vision

A suite of infrastructure for long-running,
realistic experiments in Network Science and Engineering



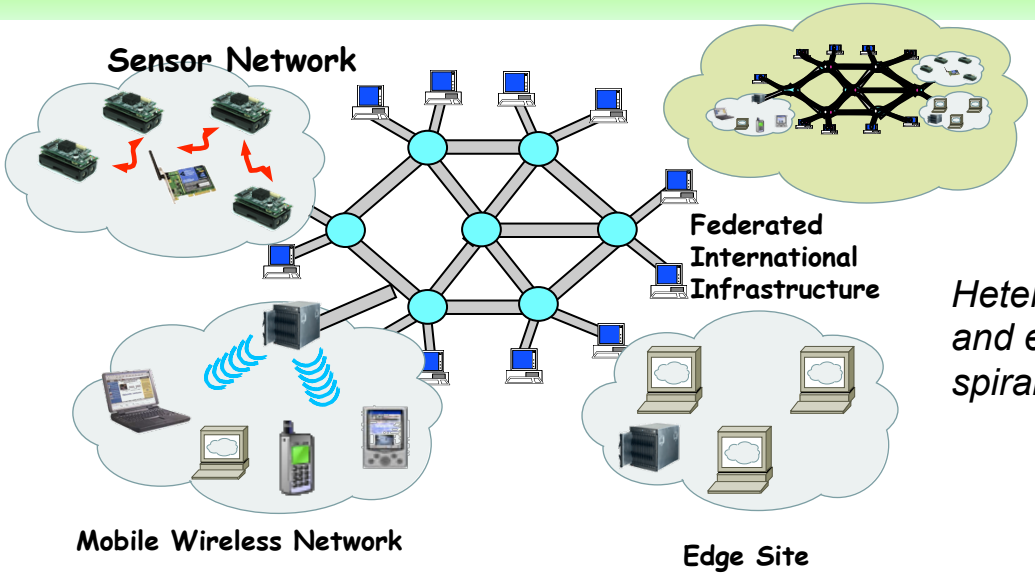
Deeply programmable



Virtualized

Federated substrate with end-to-end virtualized "slices"

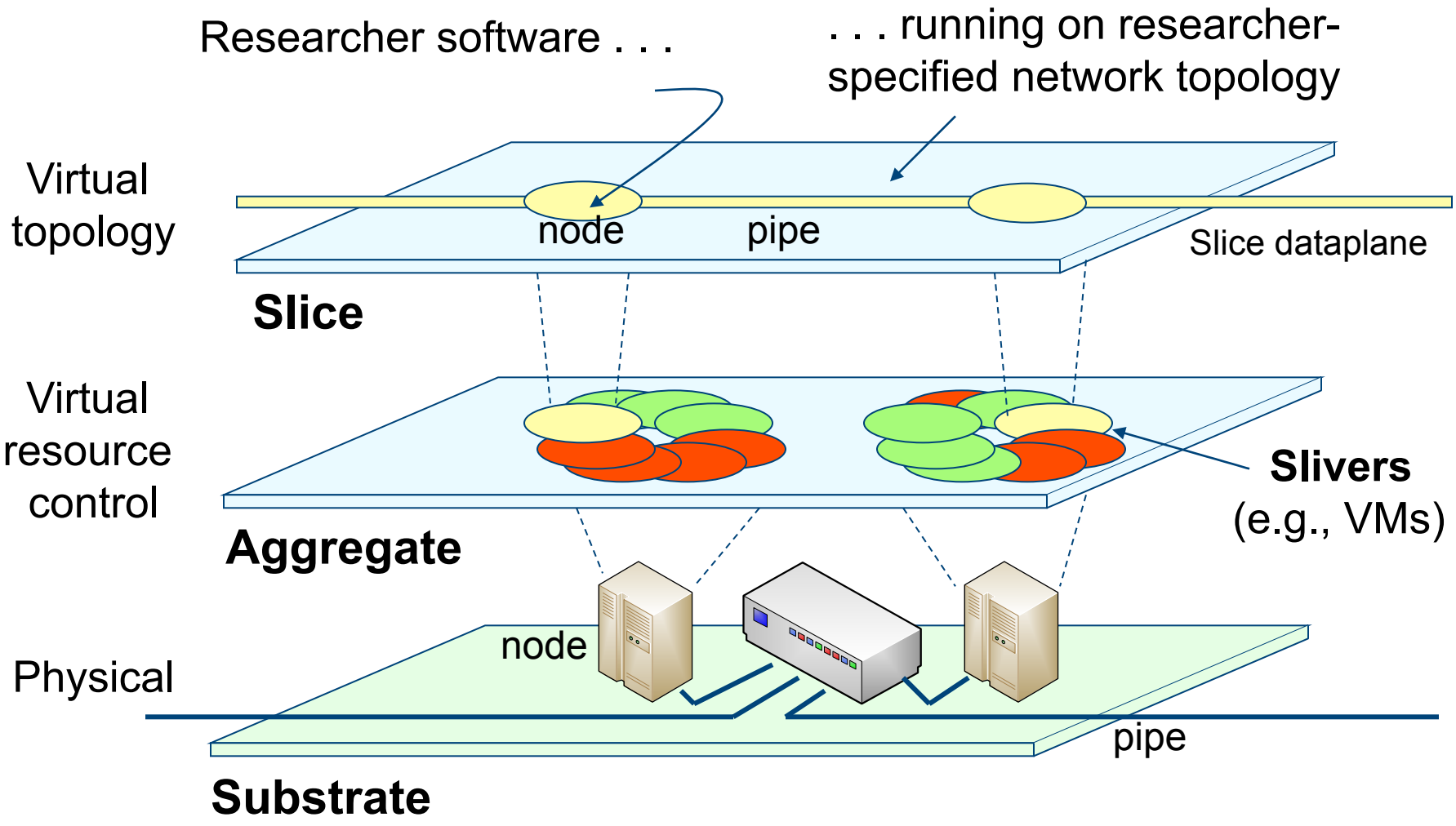
"aggregates"



Heterogeneous,
and evolving over time via
spiral development

2007

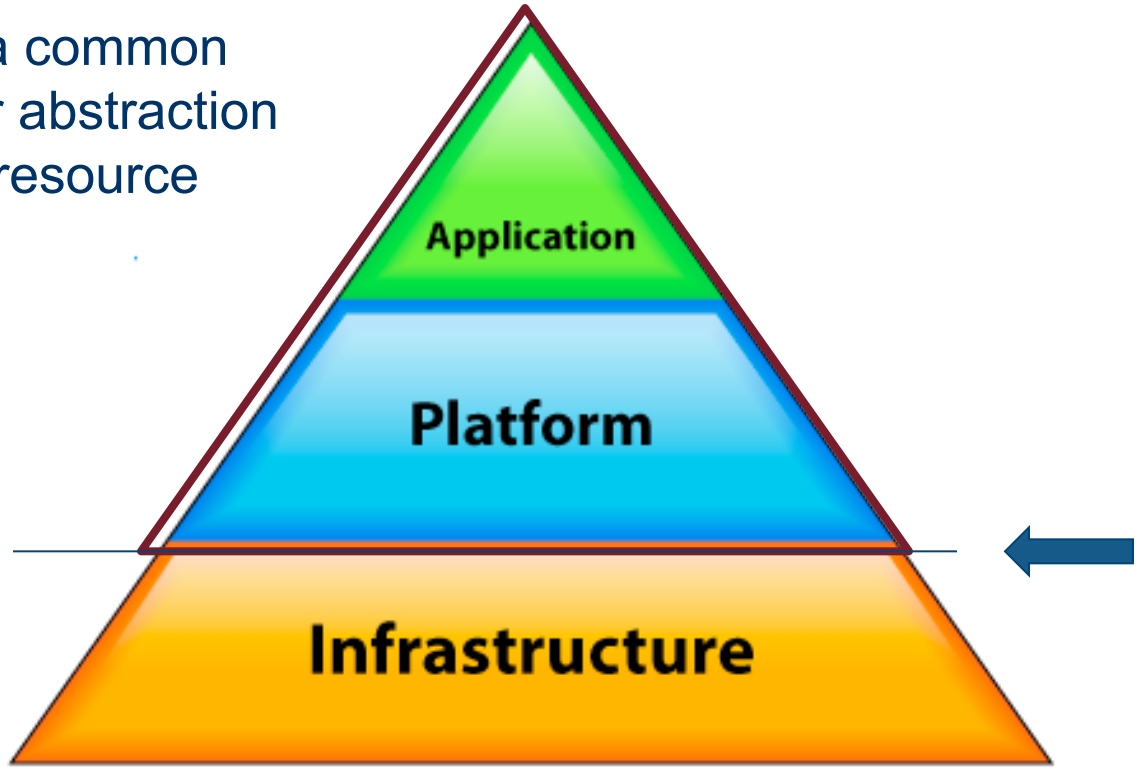
GENI resource model



GENI is IaaS

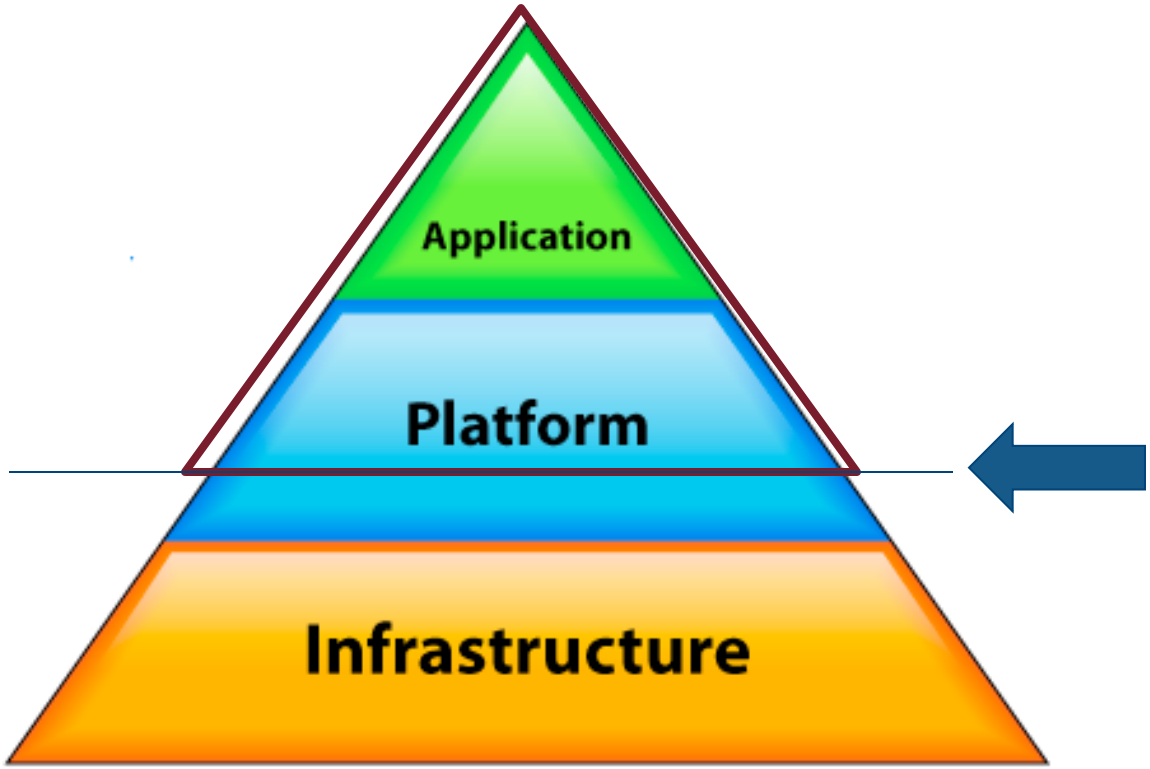
- **GENI is diverse Infrastructure-as-a-Service**
  - **Every IaaS; all connected.**

Can we build a common resource/sliver abstraction spanning “all” resource types?



GENI is IaaS

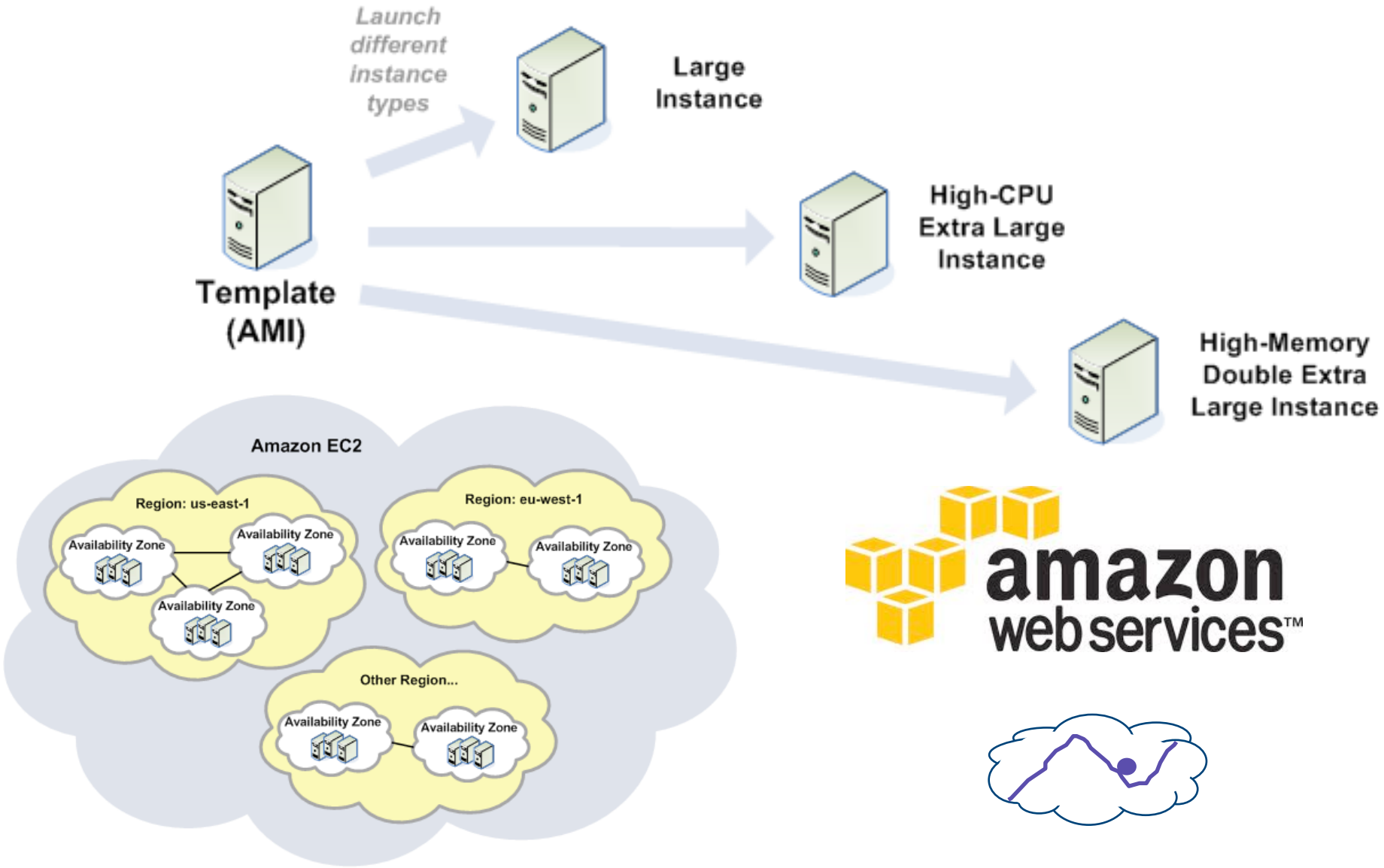
- GENI is diverse Infrastructure-as-a-Service
  - Every IaaS; all connected.
- How much platform?



Infrastructure

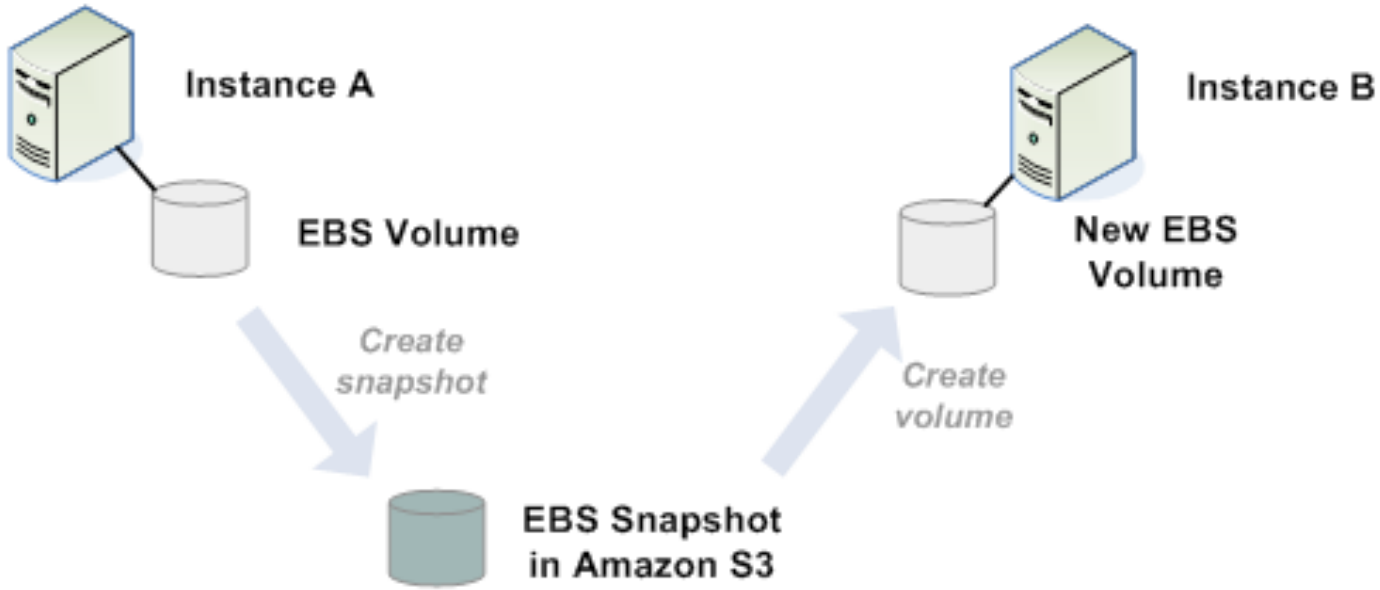
Resources and “Slivers”

EC2: The Canonical IaaS Cloud

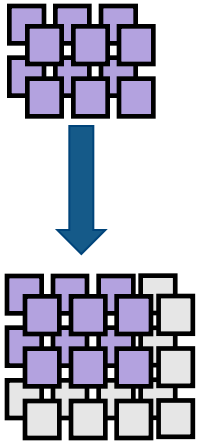


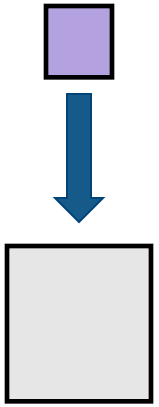
Adding storage

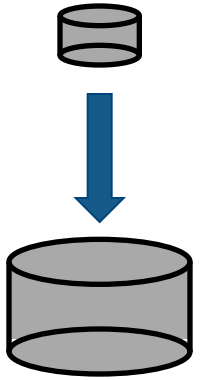
amazon web services™

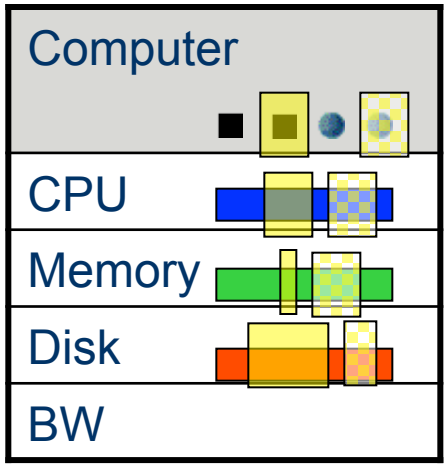


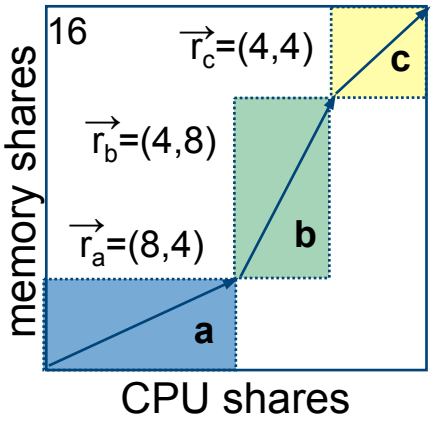
Adaptations: Describing IaaS Services







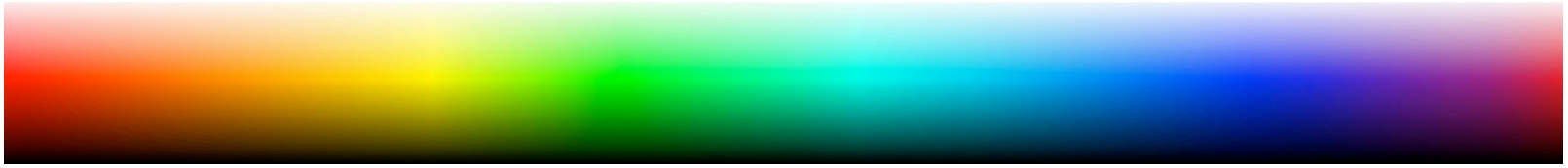




Adaptations: service classes

- **Must adaptations promise performance isolation?**
- **There is a wide range of possible service classes…to the extent that we can reason about them.**

Continuum of service classes



Available surplus | Weak effort | Best effort | Proportional share | Elastic reservation | Hard reservation

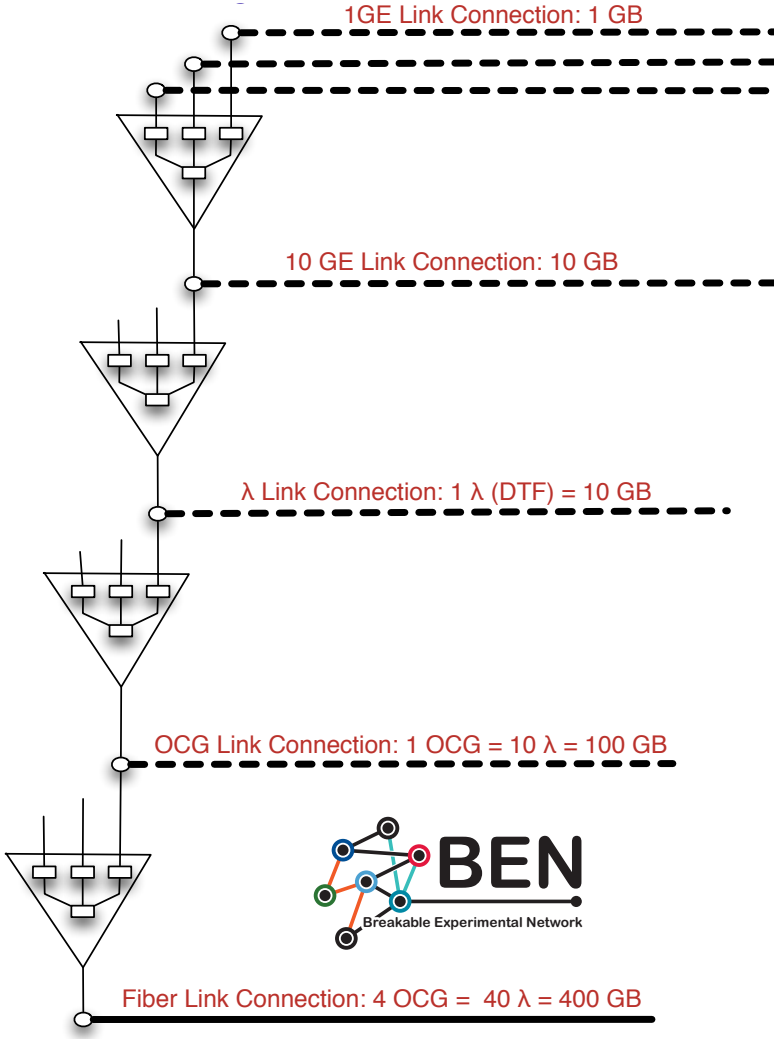
Reflects load factor or overbooking degree

Reflects priority

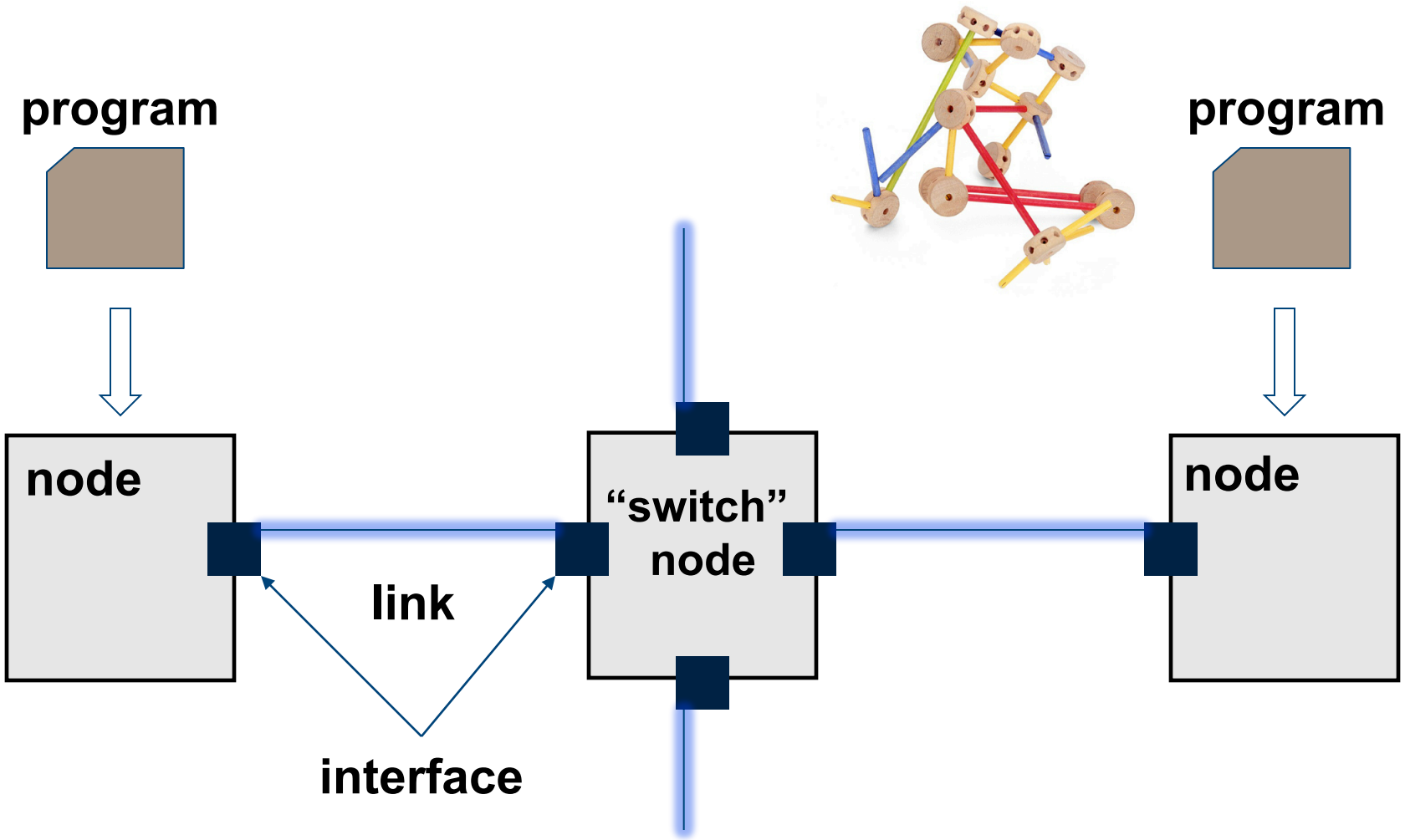
Adaptations: multi-layer networks

NDL semantic description: a snippet

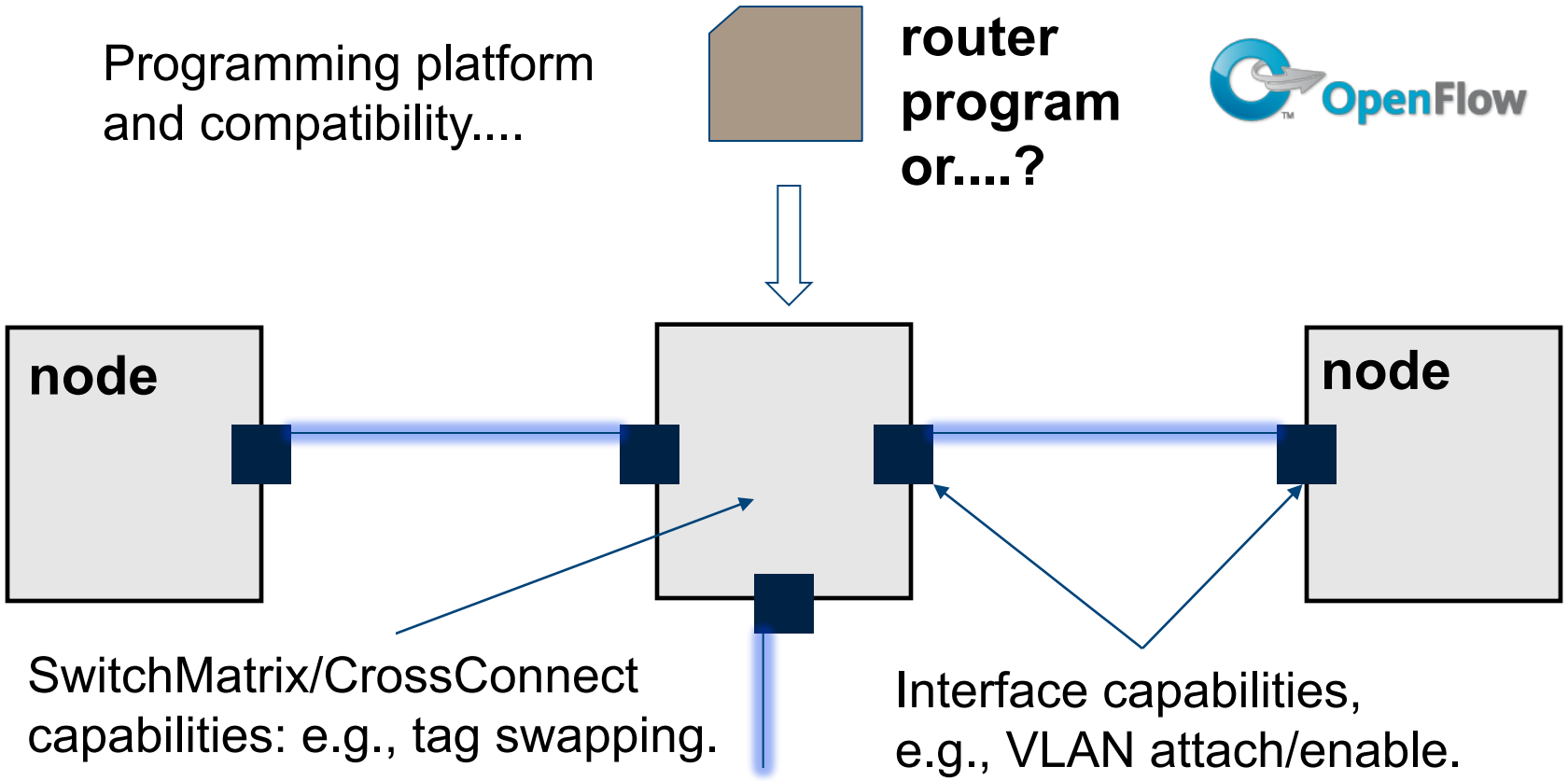
```
<!--Polatis-Renci:f1-->
<ndl:Interface rdf:about="#Polatis-Renci:f1'
<rdf:type rdf:resource="http://…/ndl/wdm#F
  <rdfs:label>Polatis-Renci:f1</rdfs:label>
  <ndl:connectedTo rdf:resource="#Polatis-
</ndl:Interface>
```



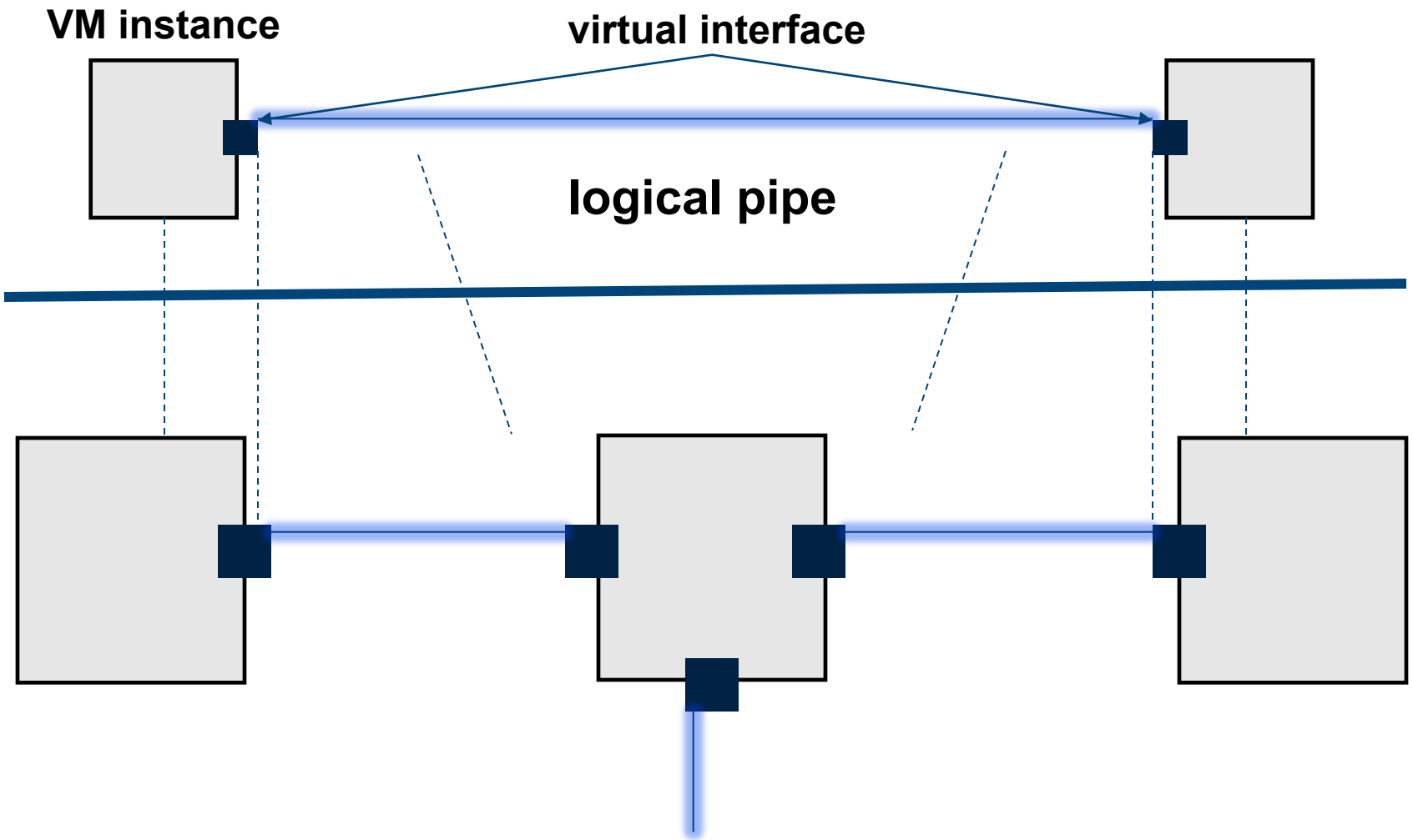
Making connections



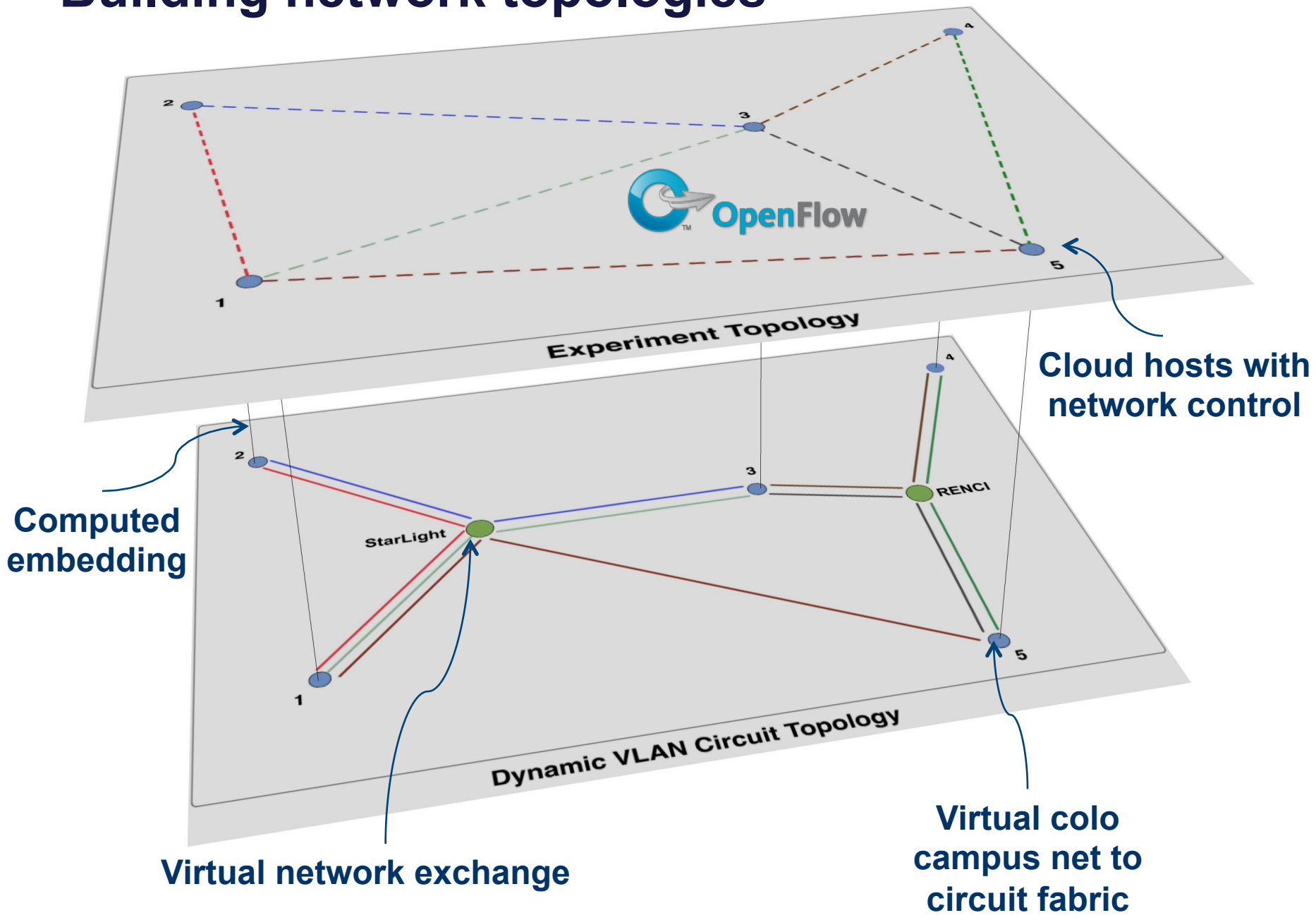
Resource functions and attributes



An adaptation

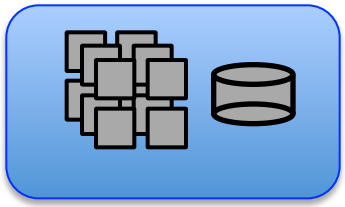


Building network topologies

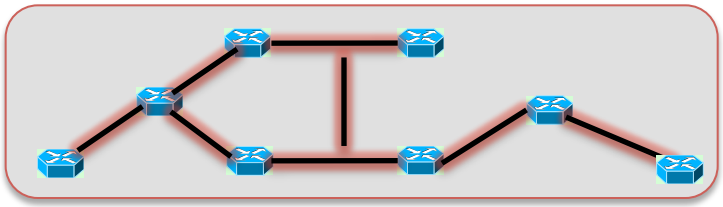


IaaS: clouds and network virtualization

Virtual Compute and Storage Infrastructure



Virtual Network Infrastructure



Cloud APIs (Amazon EC2 ..)

Network Provisioning APIs (NLR Sherpa, DOE OSCARS, I2 ION, OGF NSI …)

Cloud Providers

NLR

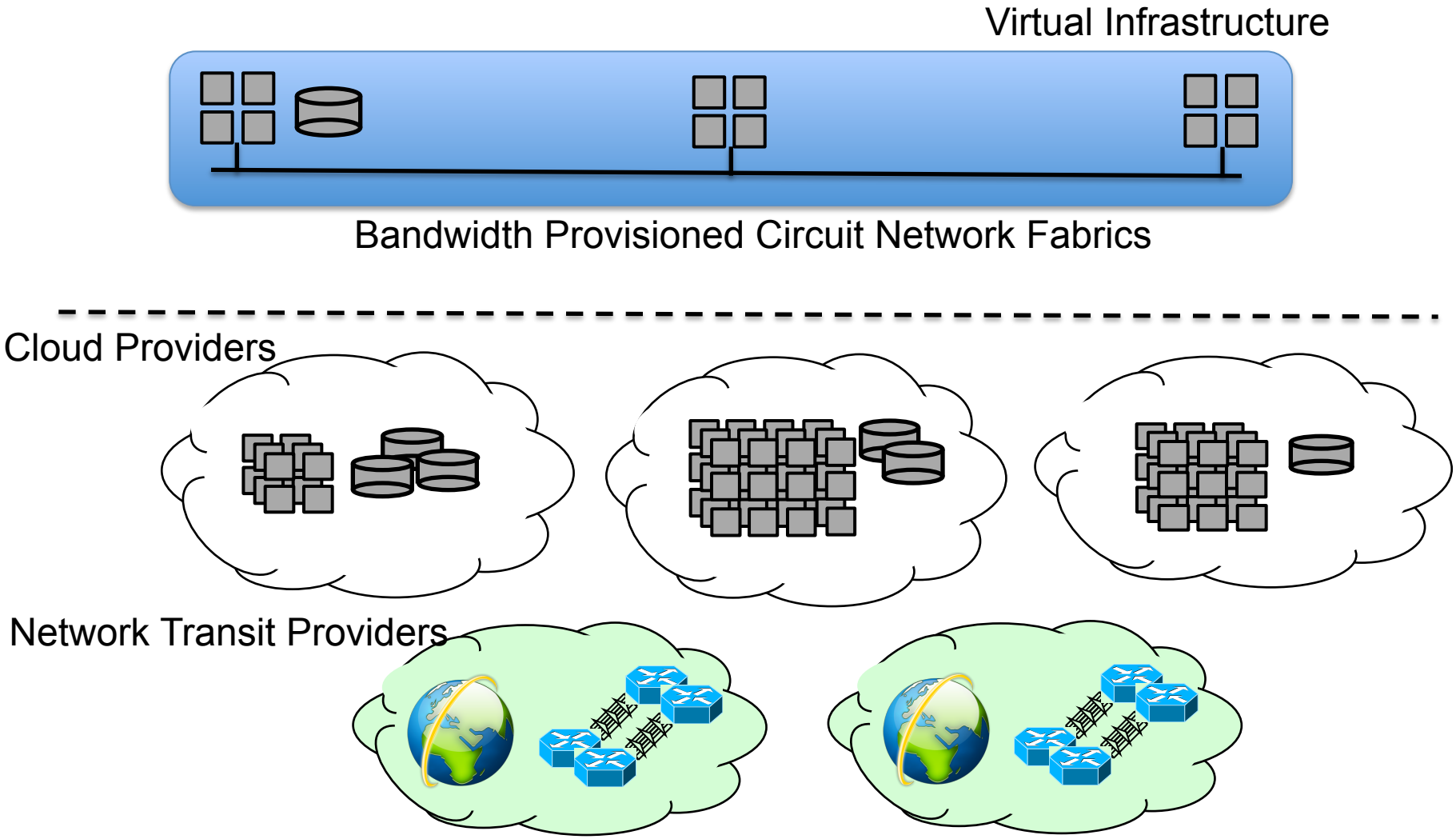
ESnet
Energy Sciences Network

INTERNET2

BEN
Breakable Experimental Network

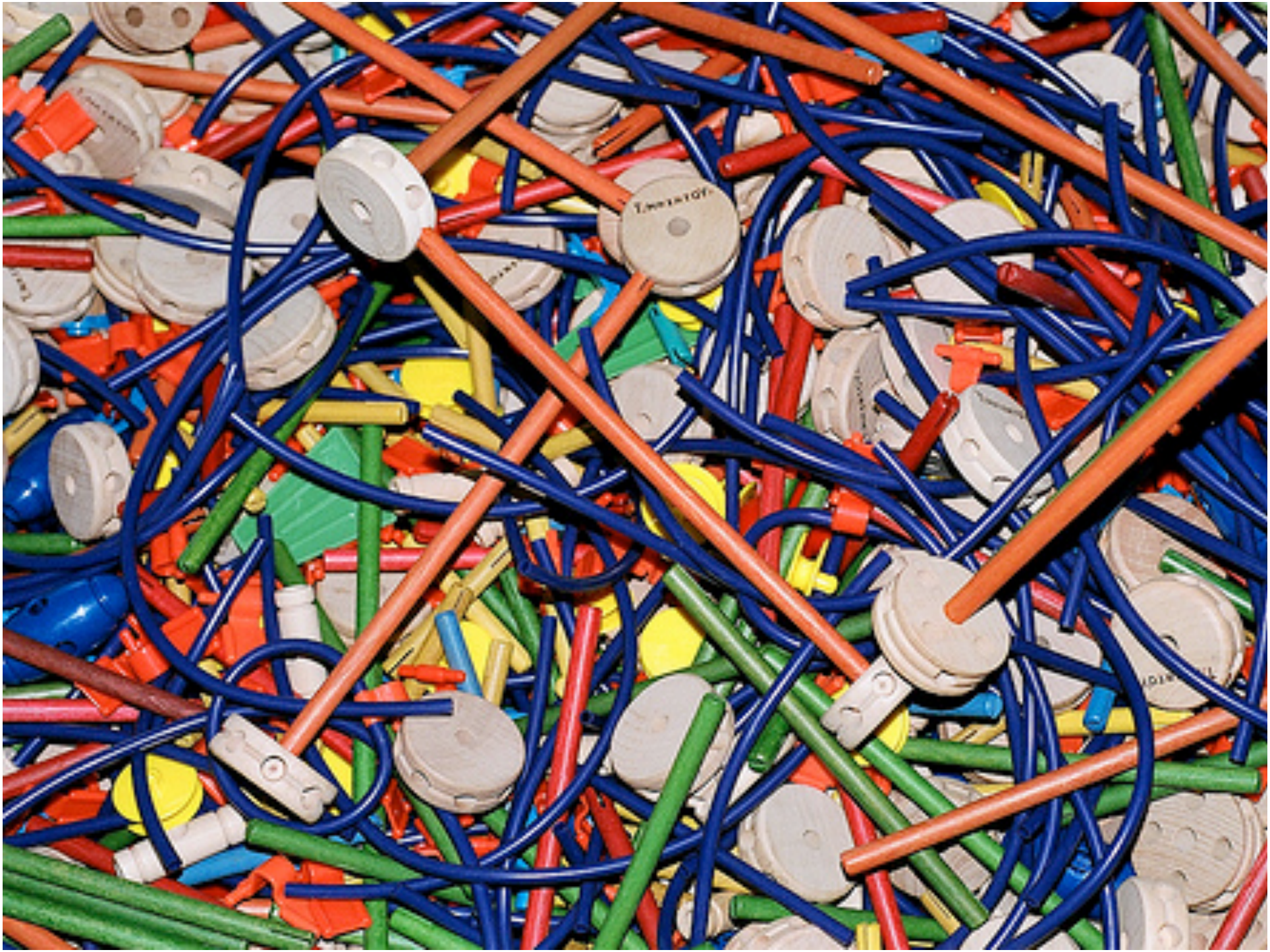
Transport Network Providers

“The missing link”

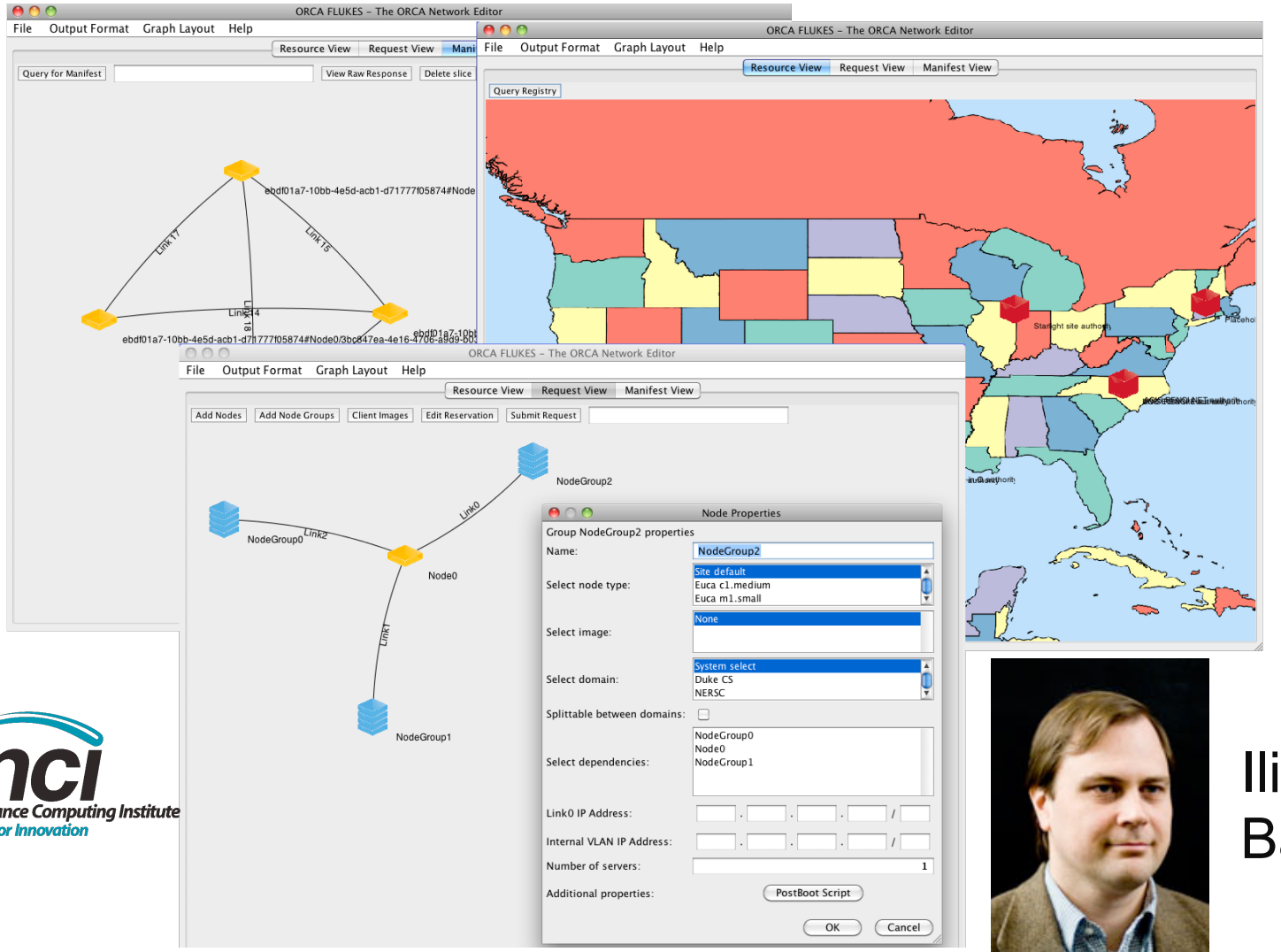


Orchestration

Keeping it together



Flukes Semantic GUI



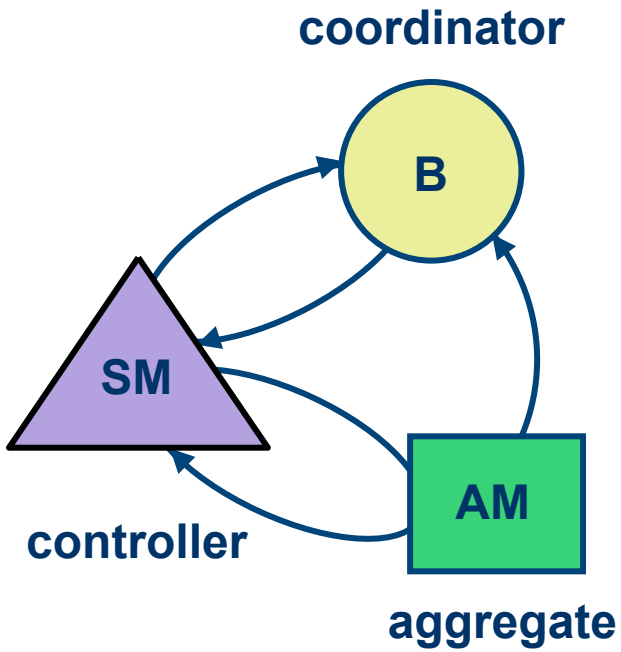
Renci
Renaissance Computing Institute
Catalyst for Innovation

Ilia Baldine

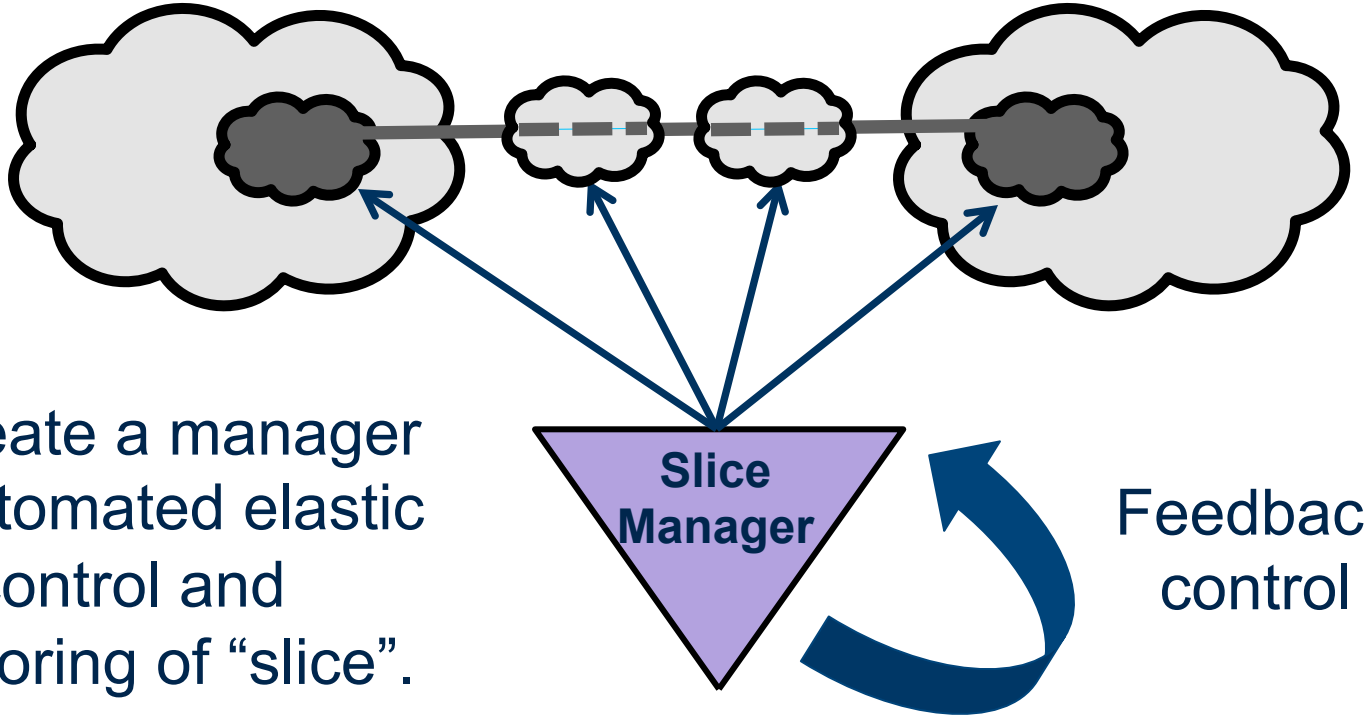
ORCA

Open Resource Control Architecture

- **ORCA is a "wrapper" for off-the-shelf cloud and circuit nets etc., enabling federated orchestration:**
  - + **Resource brokering**
  - + **VM image distribution**
  - + **Topology embedding**
  - + **Stitching**
  - + **Authorization**
- **GENI, DOE, NSF SDCI+TC**
- **http://networkedclouds.org**
- **http://geni-orca.renci.org**



Overview (1)

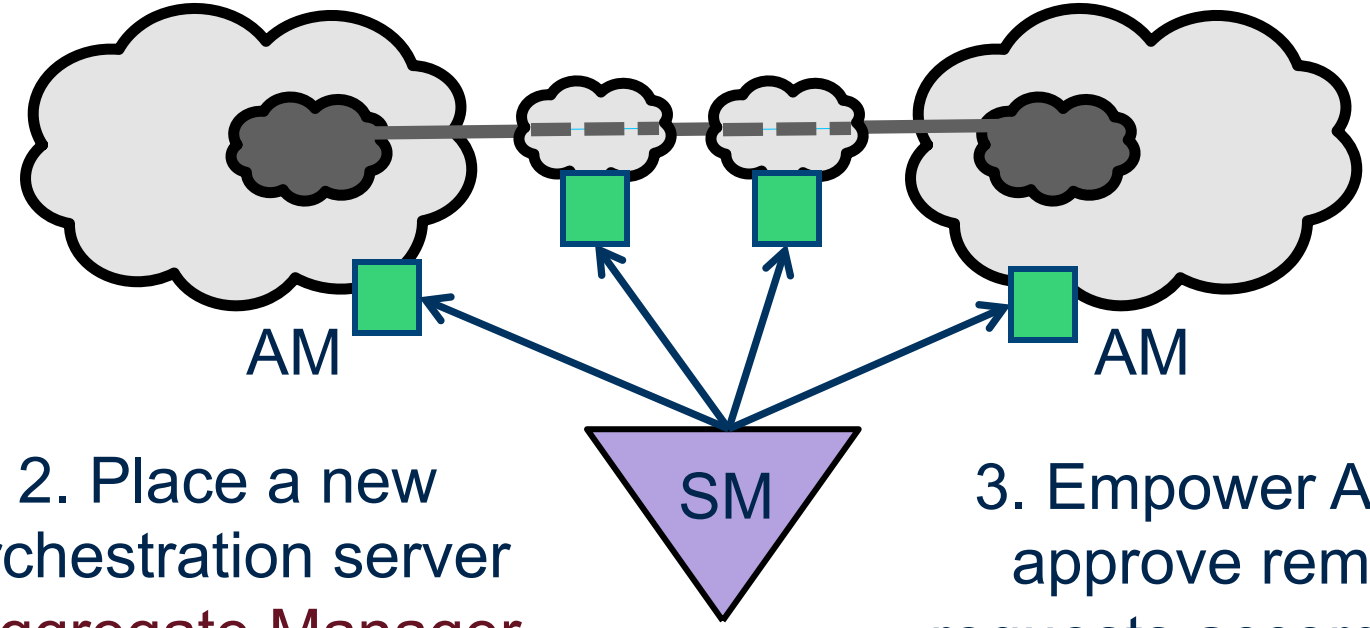


1. Create a manager for automated elastic control and monitoring of “slice”.

Feedback control

Guest-programmable
Pluggable controller policies
Deployment and monitoring

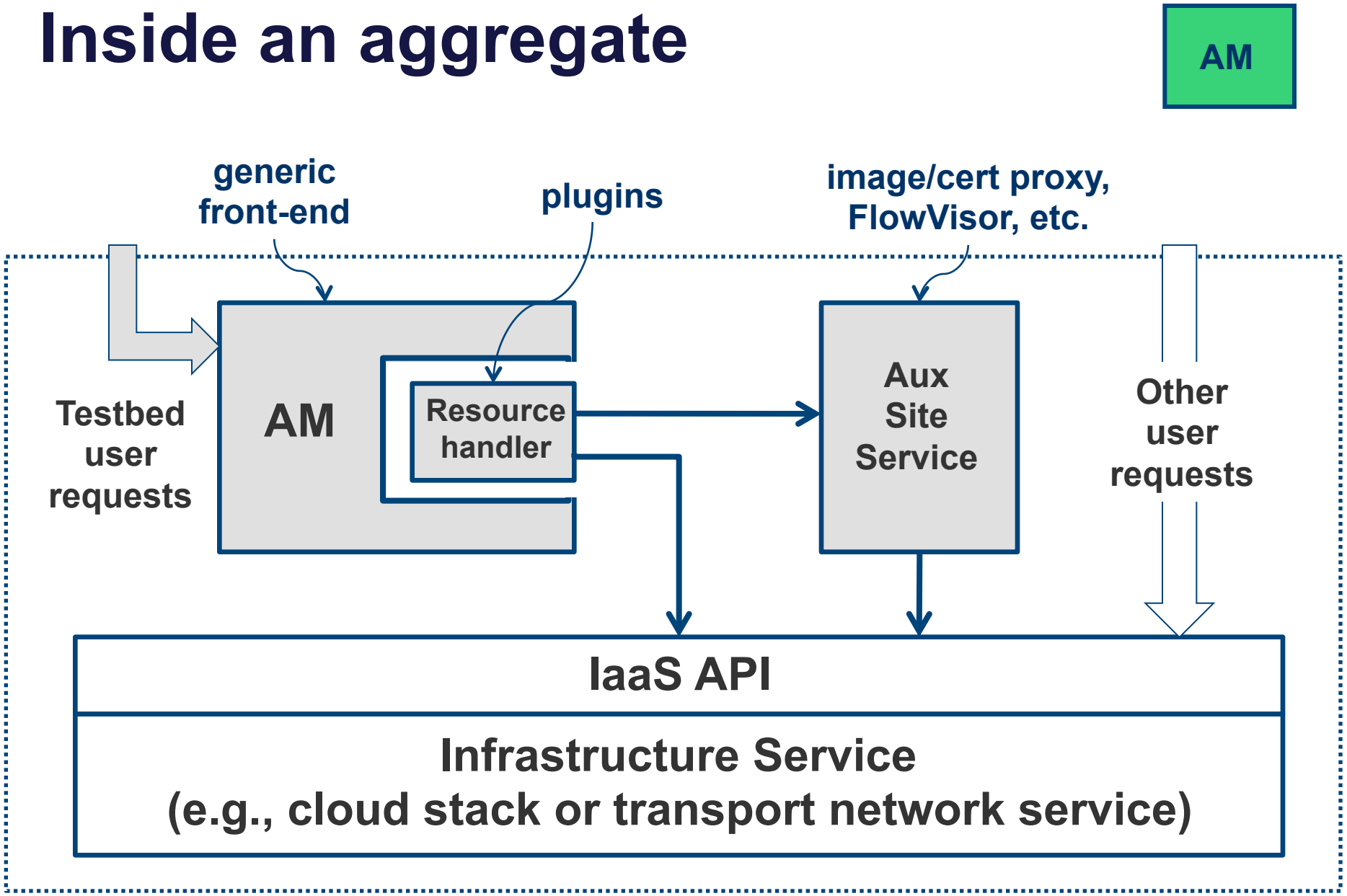
Overview (2)



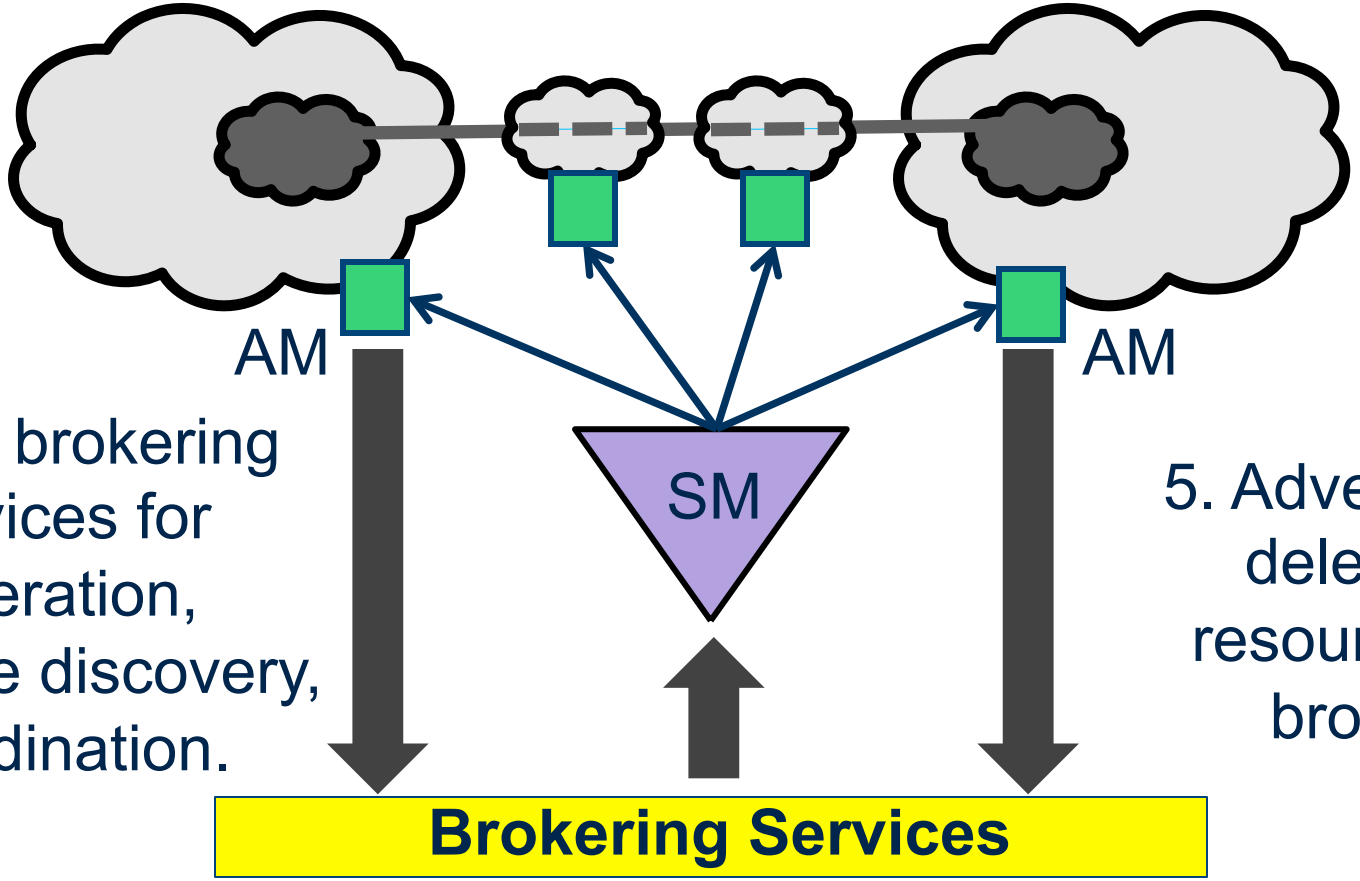
2. Place a new orchestration server (Aggregate Manager or authority) at each provider.

3. Empower AM to approve remote requests according to local policies, and invoke local control interfaces as needed.

Inside an aggregate



Overview (3)

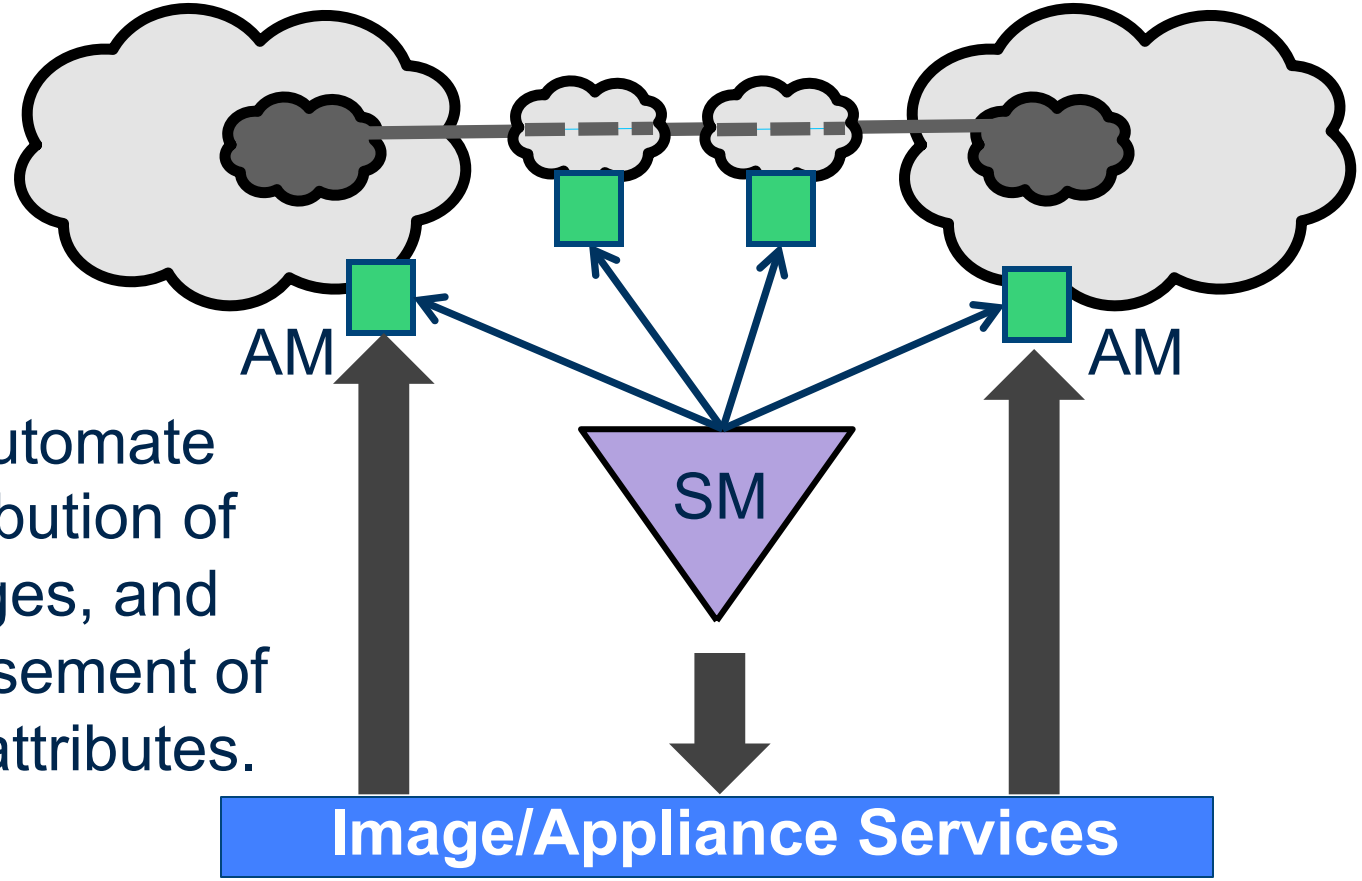


4. Add brokering services for federation, resource discovery, coordination.

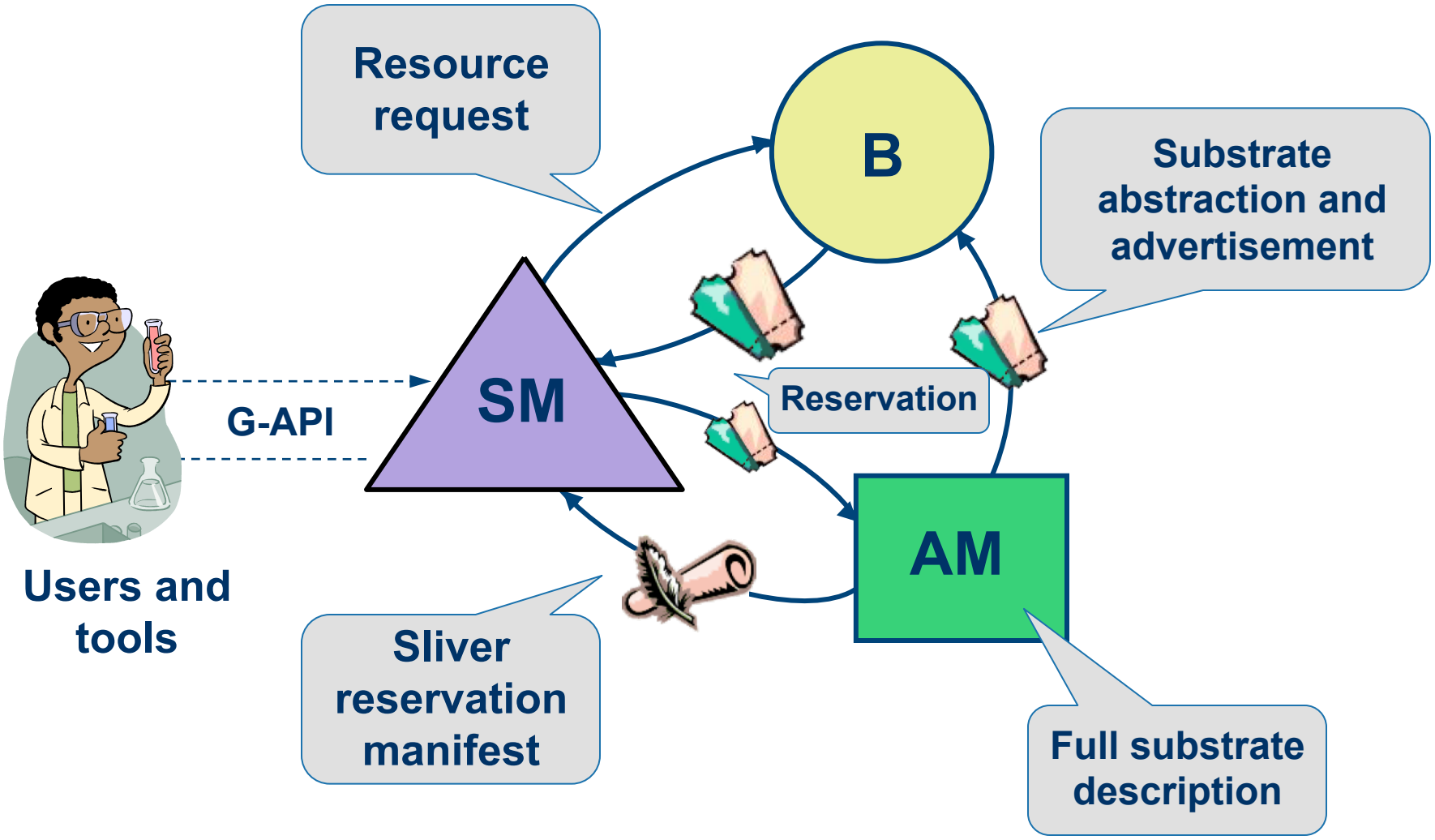
5. Advertise or delegate resources to broker.

Overview (4)

6. Automate distribution of images, and endorsement of their attributes.



NDL-OWL: Making the most of semantic resource descriptions



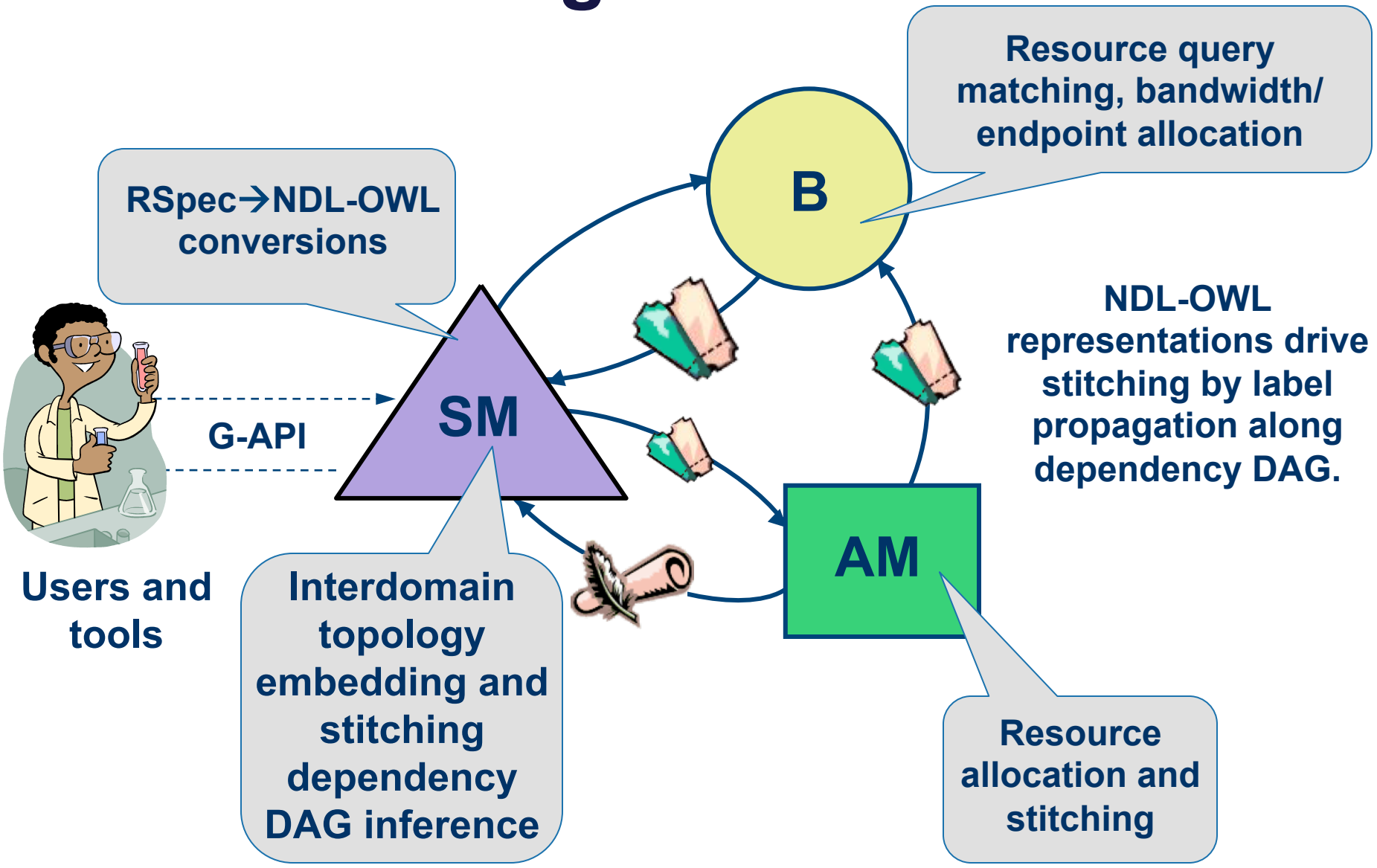
NDL-OWL

- NDL-OWL ontology (Semantic Web) extends Network Description Language (NDL), based on G.805 model.
  - uva.nl SNE: http://www.science.uva.nl/research/sne/ndl
  - NDL represents substrate: topology, connectivity, layers, adaptations (e.g., circuits, virtualization)
  - NDL enables path computation by SPARQL queries
- NDL-OWL adds abstracted views of resources
  - Abstracted view of substrate (resource advertisements)
  - Resource requests, embeddings, and allocation
  - Edge resources and their configuration
  - Label sets (e.g., VLAN tags), label produce/consume tags

NDL-OWL in ORCA

- Represent resource semantics for IaaS systems declaratively.
  - Not “baked in” to control framework.
  - Drive control framework actions by queries on models.
- NDL-OWL adds semantically rich labels for dynamic provisioning.
  - Capacity constraints, QoS, usage tracking
- NDL-OWL modules in AMs drive templated configuration commands automatically.
  - setup/teardown handler calls from Aggregate Manager
- NDL-OWL modules in SM generate dependency graph for sequenced stitching.
  - Declarative stitching framework: propagate typed labels from producers to consumers along dependency DAG

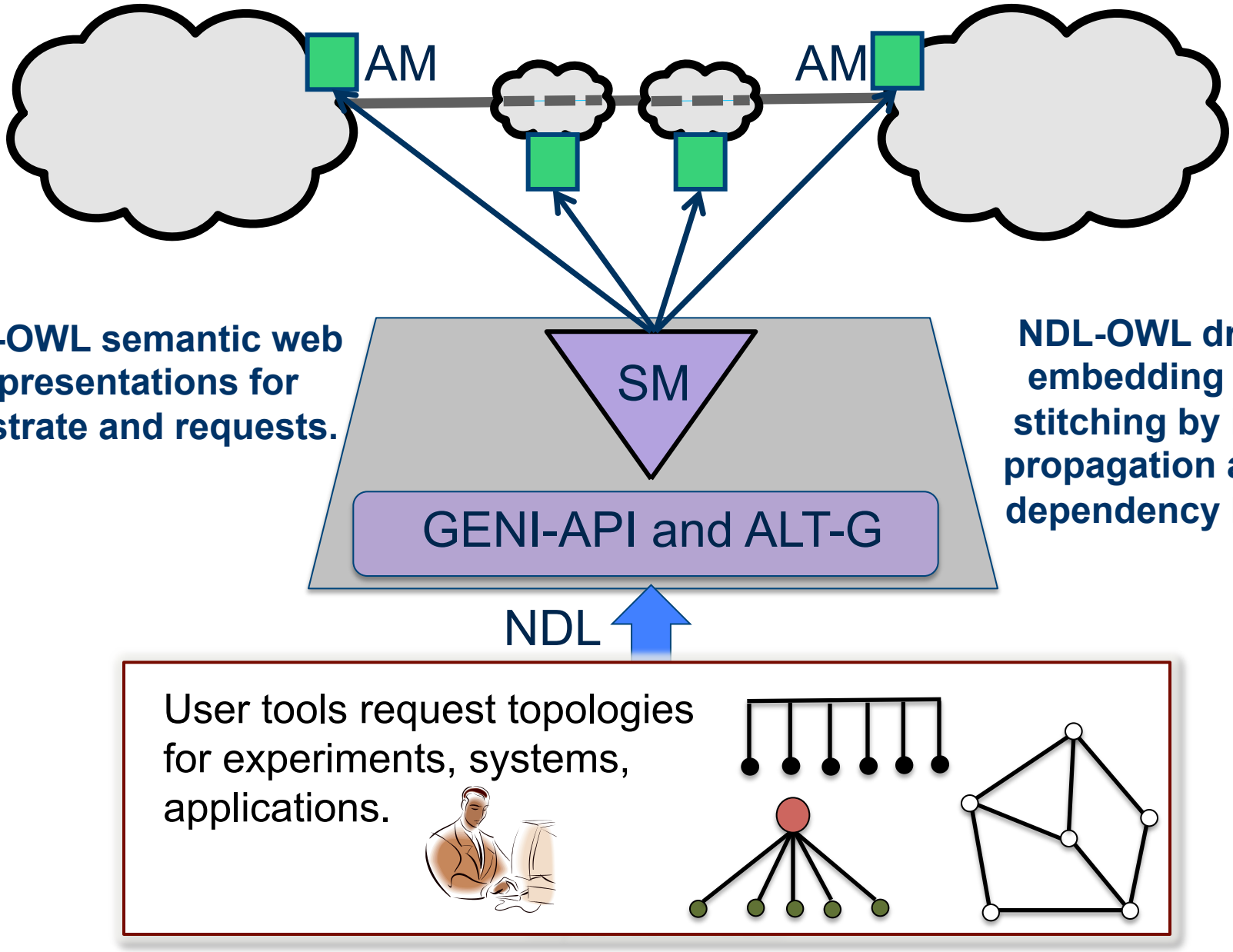
NDL-OWL Logic



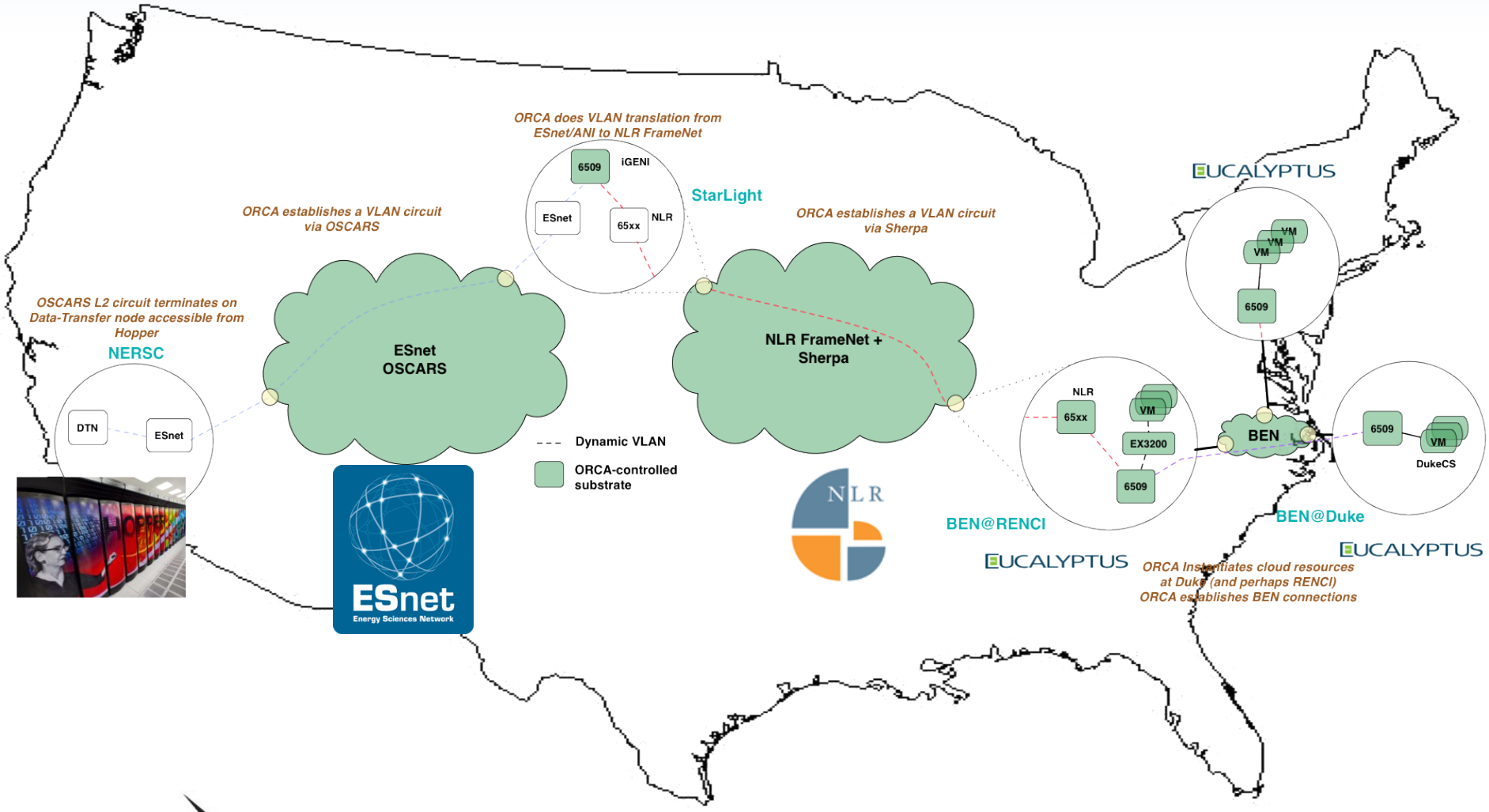
ExoGENI

- **Every Infrastructure as a Service, All Connected.**
  - Substrate may be volunteered or rented.
  - E.g., public or private clouds and transit providers
- **ExoGENI Principles:**
  - Open substrate
  - Off-the-shelf back-ends
  - Provider autonomy
  - Federated coordination
  - Dynamic contracts
  - Resource visibility

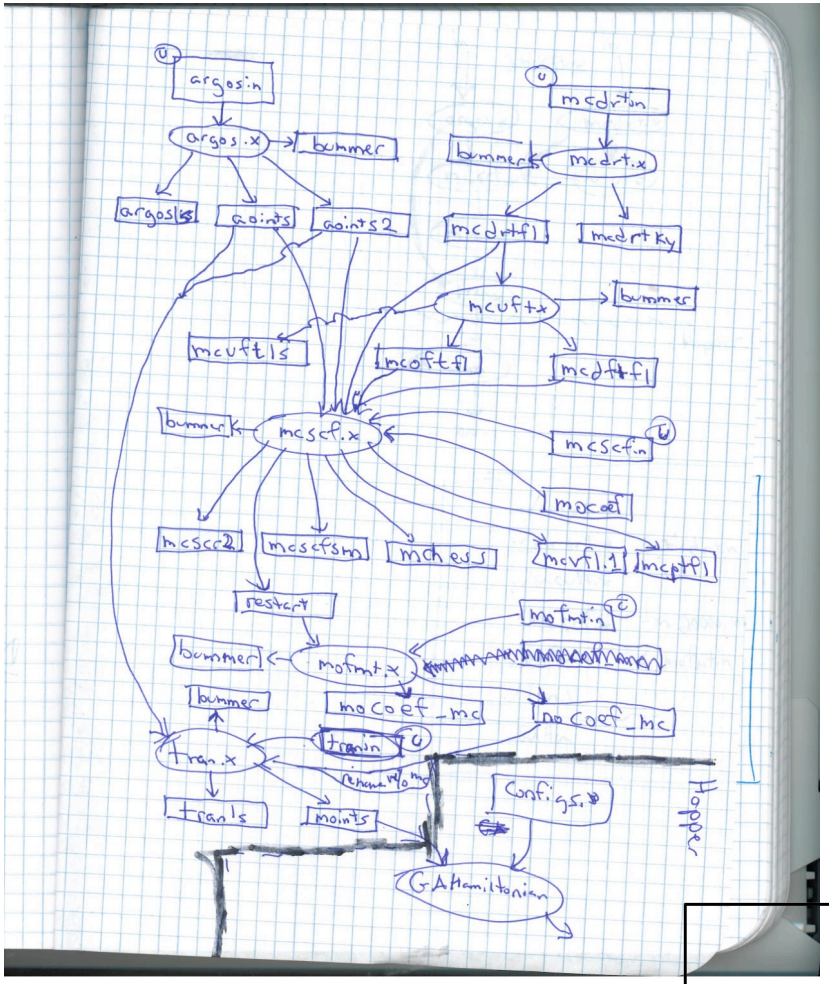
Applications and Platforms

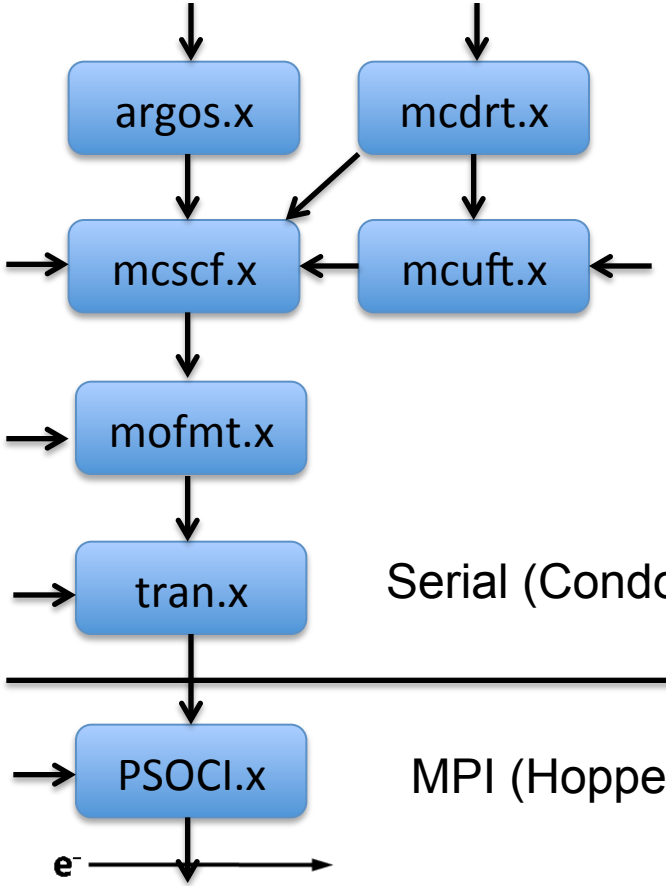

AM
AM
NDL-OWL semantic web representations for substrate and requests.
SM
GENI-API and ALT-G
NDL-OWL drives embedding and stitching by label propagation along dependency DAG.
NDL
User tools request topologies for experiments, systems, applications.

SC11 Demo: Solar Fuels Workflow



renci





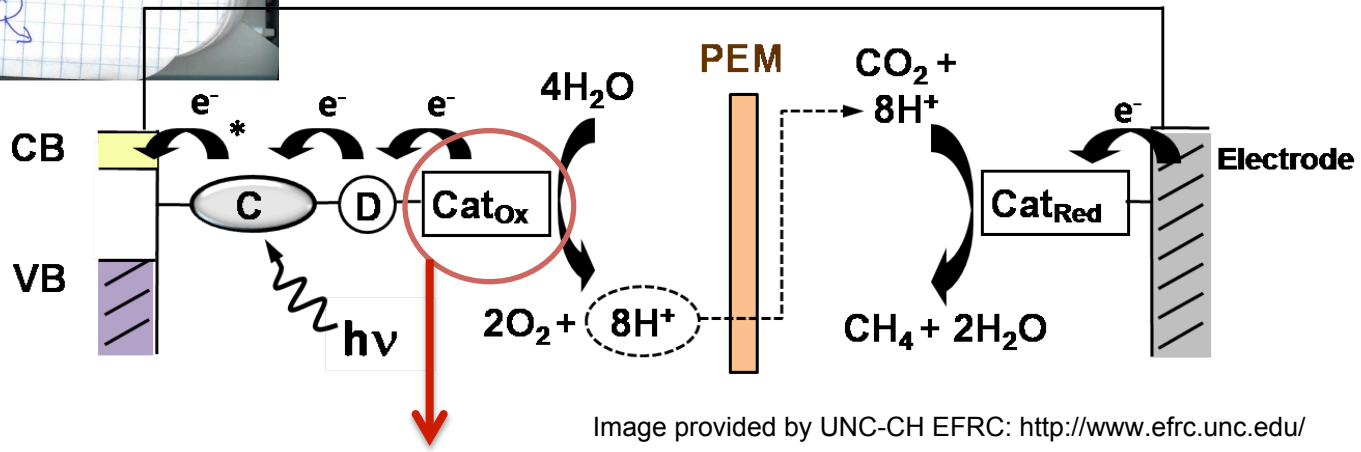


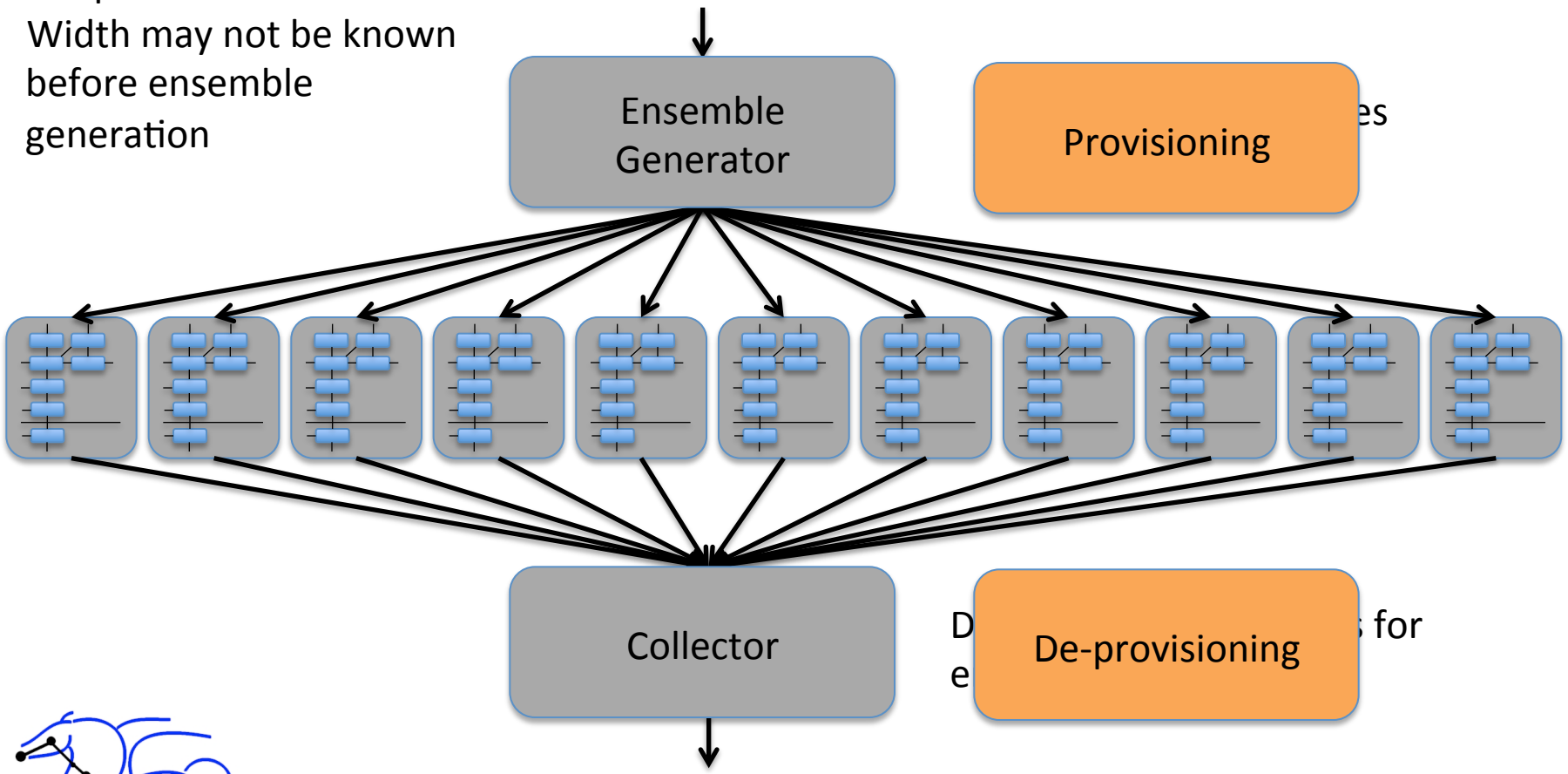
Image provided by UNC-CH EFRC: http://www.efrc.unc.edu/

Oxidation catalysts

Renci
Renaissance Computing Institute
Catalyst for Innovation

Example Dynamic Workflow: Ensemble

- Wide step of Ensemble is temporal
- Width may not be known before ensemble generation



pegasus

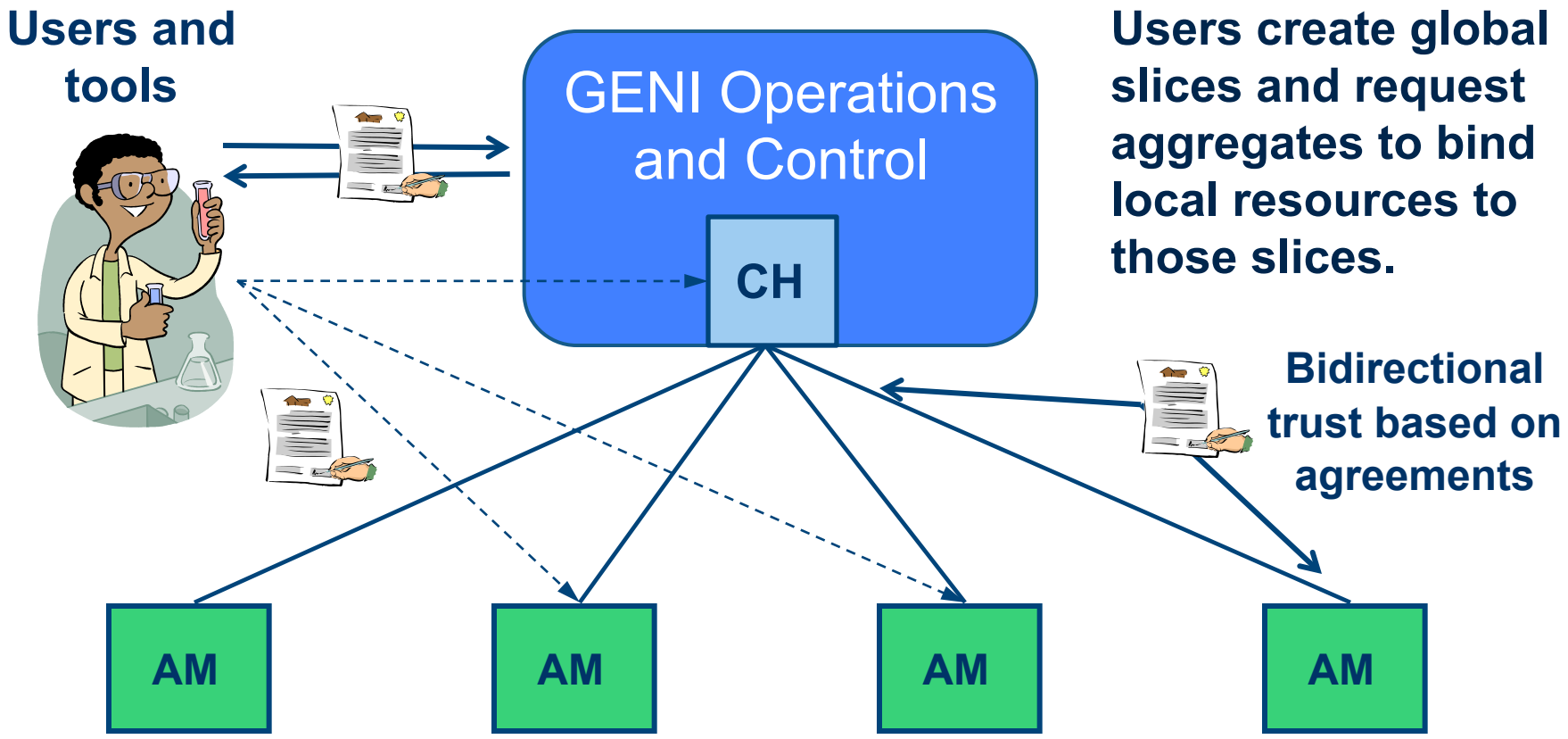
Condor
High Throughput Computing

Trust

Federated trust challenges

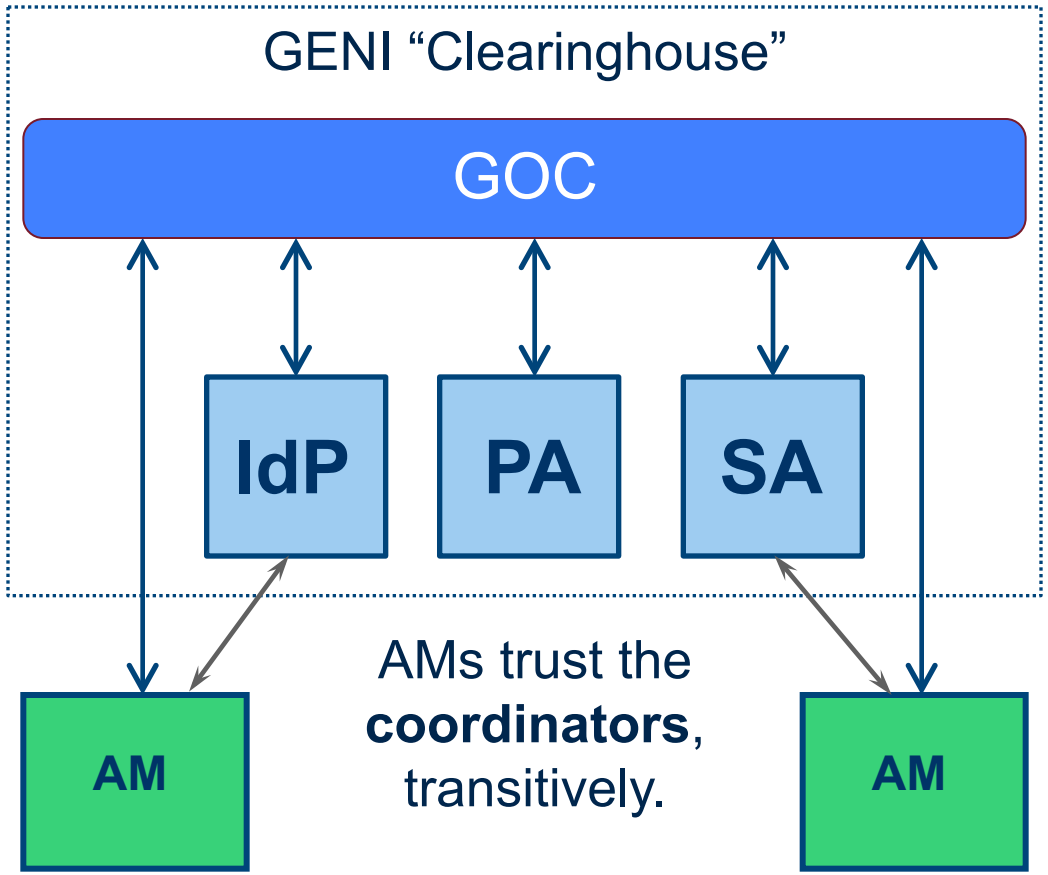
- External identity providers (SAML SSO)
  - Assertion of identity attributes
- Trust structures for federated coordination
  - Multiple overlapping federations/policies
- Declarative trust structure and policy
  - Delegation, inference, audits, accountability
  - Global slices with capability-based protection
  - Software-defined networking services and rights
  - Image certifications/endorsements
  - stitching

GENI trust structure: overview (v1.0)



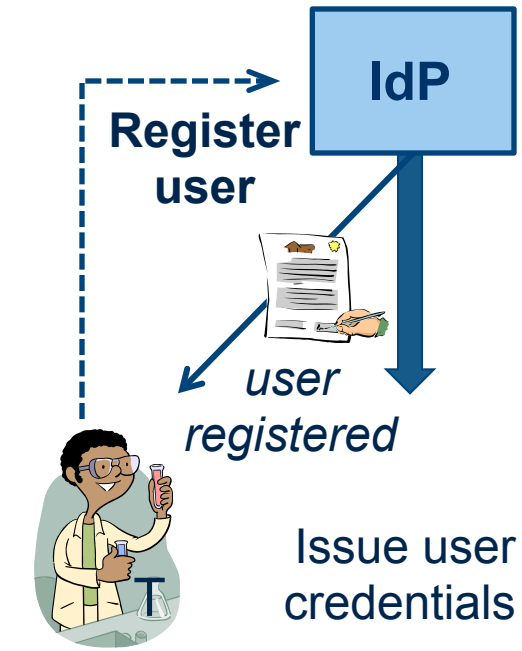
Principals are users and organizations, and tools or servers acting on their behalf.

GENI trust structure (v2.0)



NSF GENI Federation provides identity and authorization services (coordinators) for GENI aggregates.

Example: Identity Provider (IdP)



Issue user credentials

InCommon

- **An IdP asserts facts about users.**
- User attributes may include inCommon attributes harvested through indirect delegation to Shibboleth IdPs.
- These attributes may have parameters with simple values (strings or numbers).

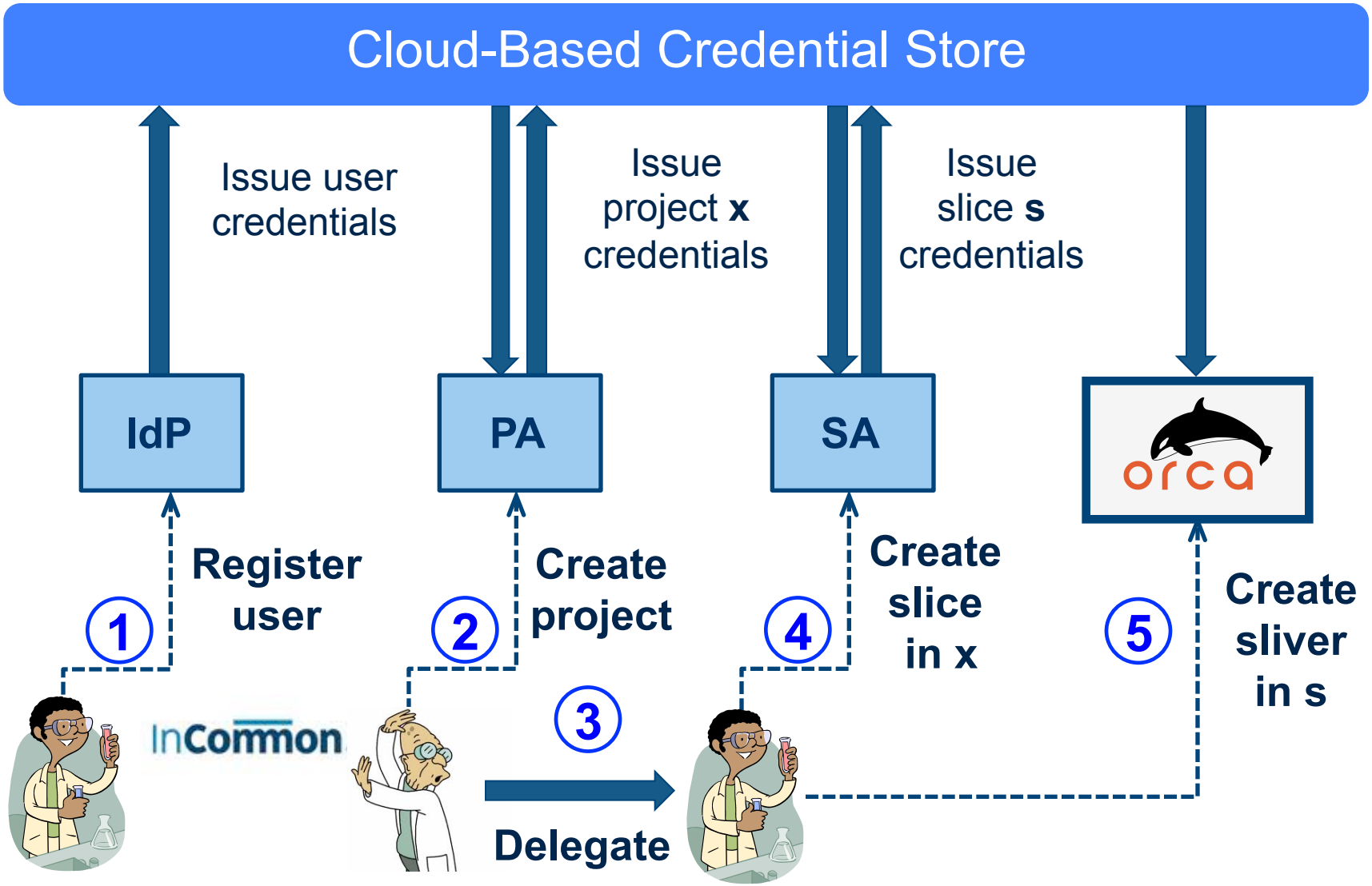
IdP.**geniUser**←T
IdP.**student**←T
IdP.**enrolled**(CS-114)←T

IdP.**geniUser**←D
IdP.**faculty**←D

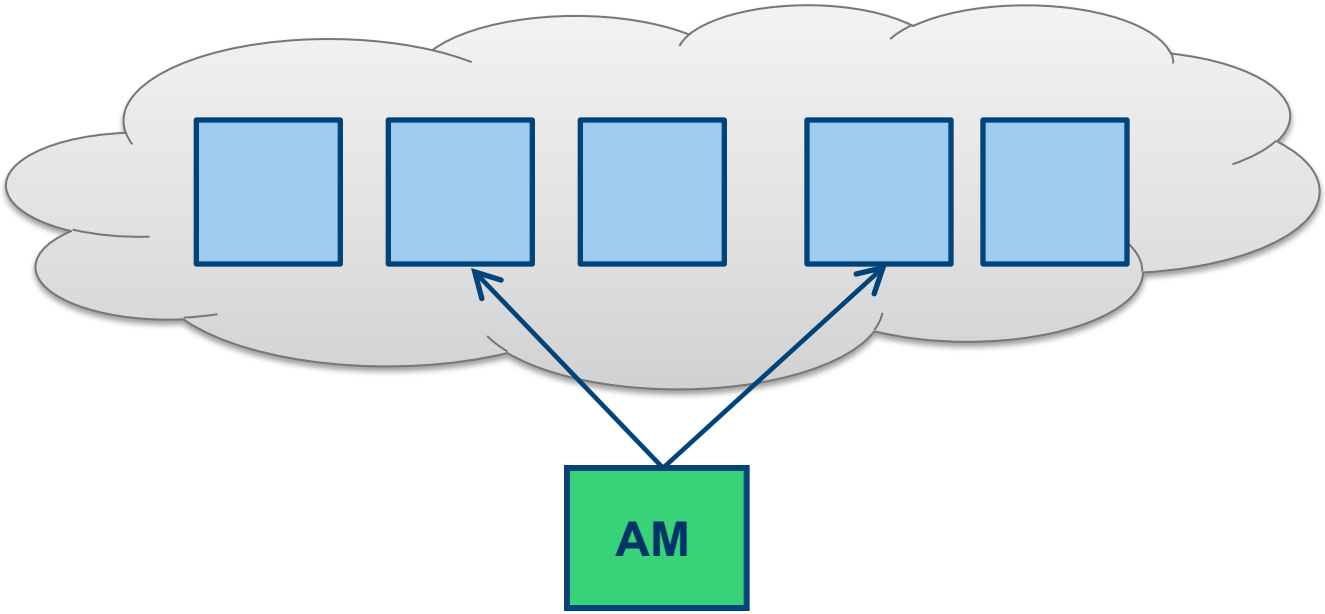
Users have roles e.g., student, faculty.

Shibboleth

GENI Authorization: Workflow

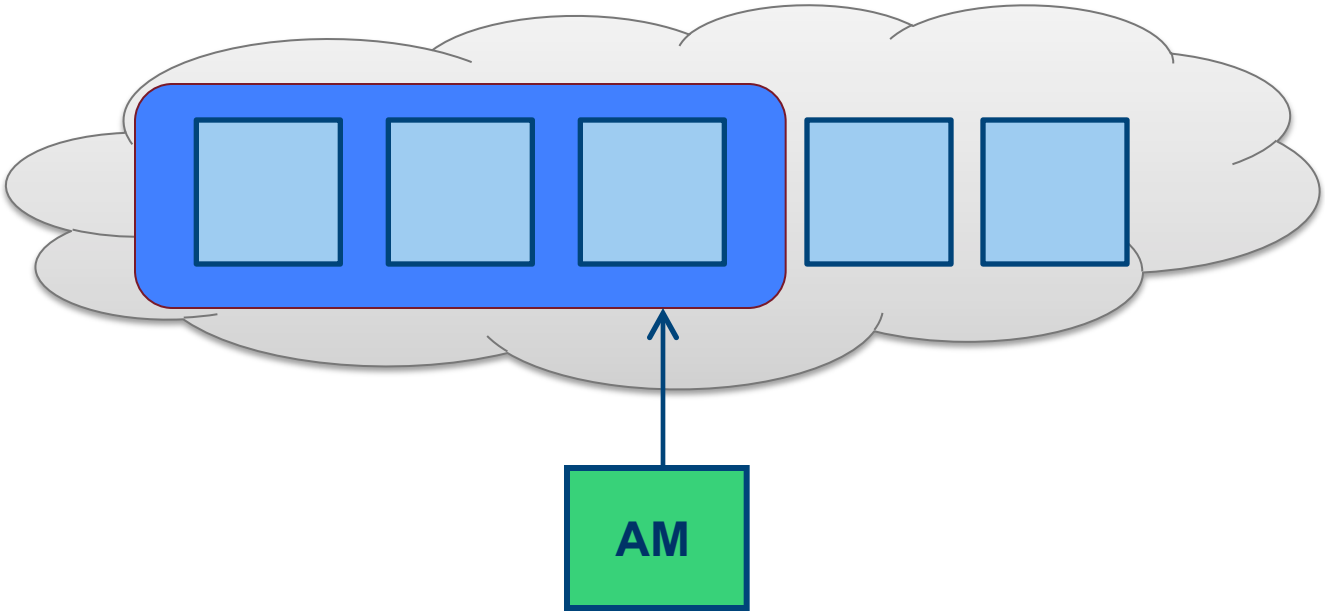


The view from an AM



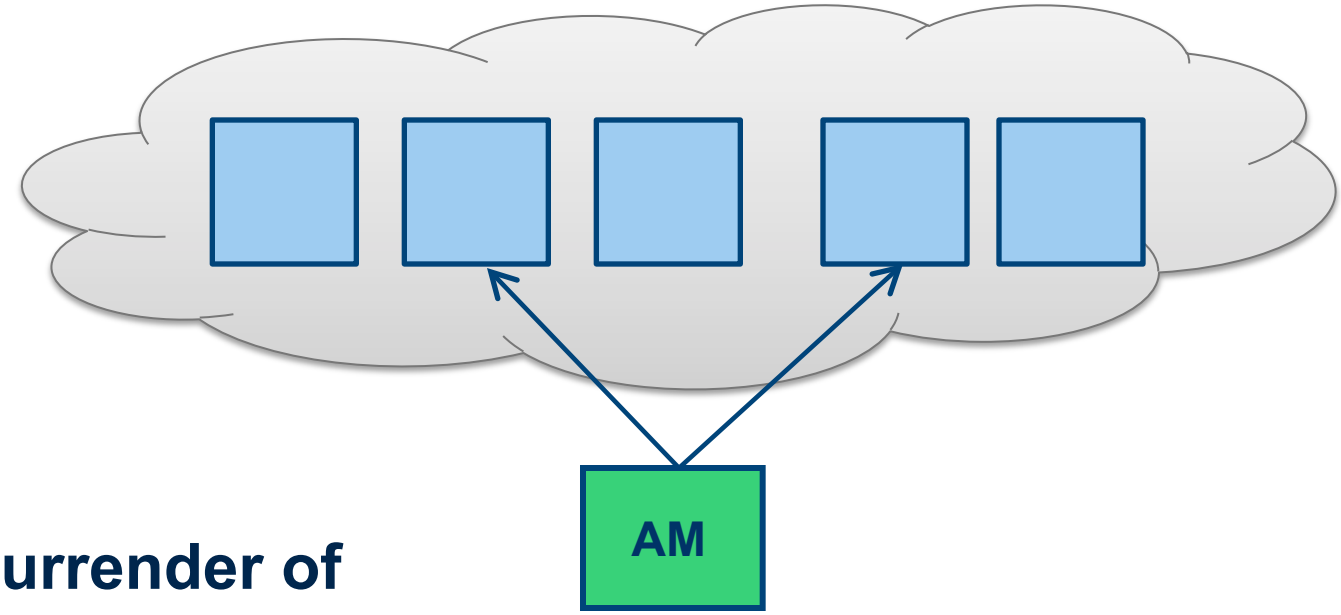
- To an aggregate, the world is (has) a cloud of coordinators that the aggregate may accept, or not.
- Think about EC2: it only cares about VISA and MasterCard.

The view from an AM



- A federation is just a convenient grouping of coordinators for an AM to accept as a bundle.
- An AM may even accept them without “joining” the Federation, i.e., agreeing to conform to its dictates.

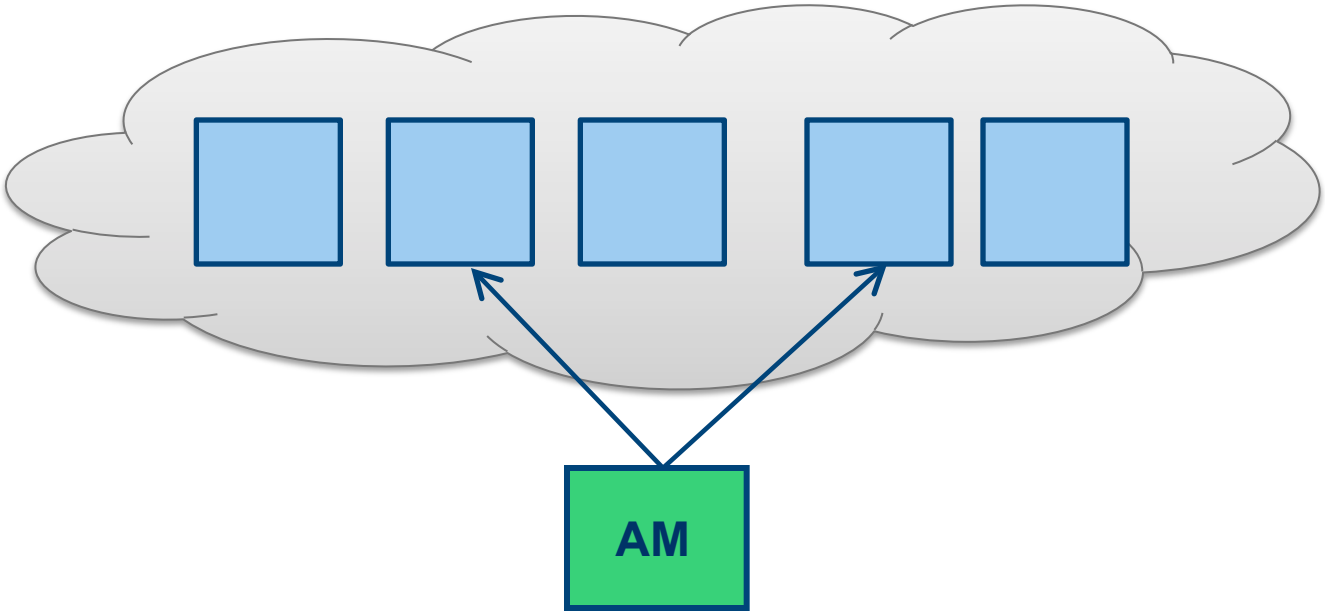
AM-centric view of coordination



Any surrender of sovereignty to a Federation is voluntary, or induced by socio-political factors outside the trust structure.

The NSF GENI AMs choose to surrender.

AM-centric view of coordination



Therefore, we should design coordinators that leave AMs free to choose from the menu of coordinators available.

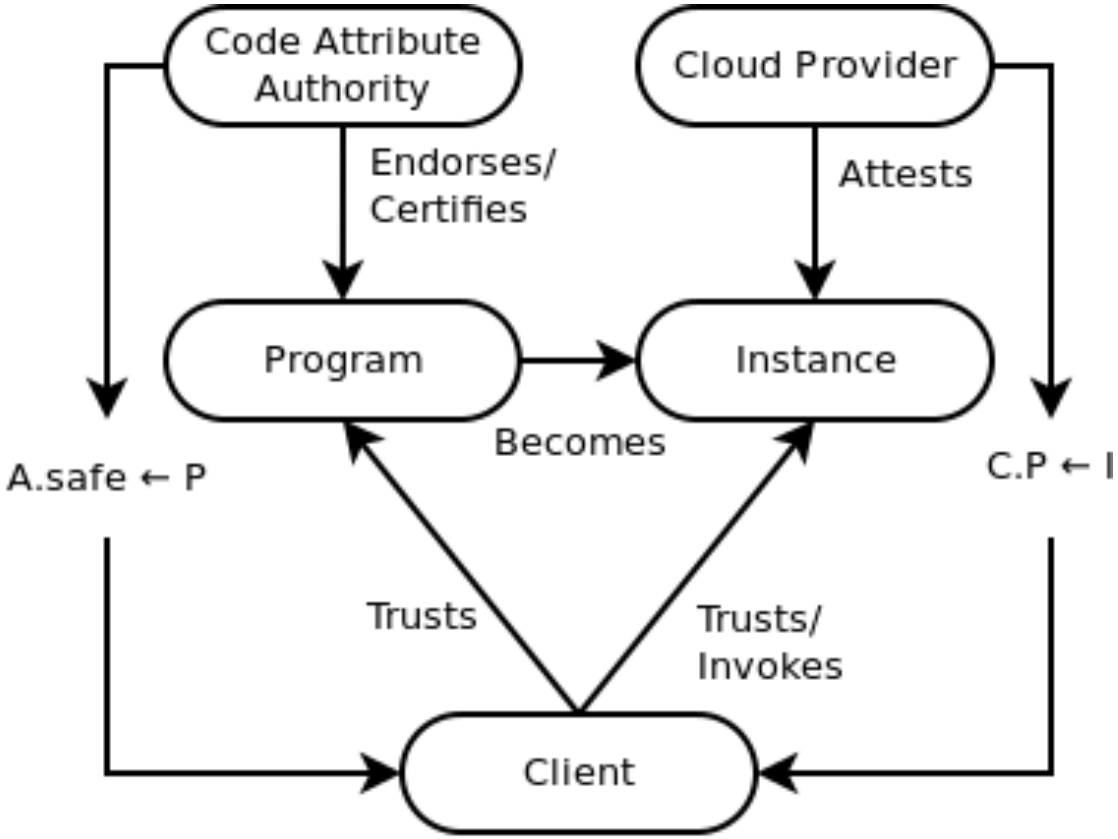
To the extent that AMs choose the same coordinators, they should work.

Software Service as a Subject

- Trust is “typically” based on a human or organizational identity.
- Can we trust a service or program instance independent of our trust in its owner (SP)?
- Can a slice or cloud-hosted service be “its own” subject?
- Add new building blocks:
  - Assert/certify program attributes
  - Remote attestation

“Trusted Platform Cloud”

- **Trusted entity certifies program properties.**
- **AM attests to loaded program or image**
- **Client infers instance attributes from program properties**
- **Sealed instances**

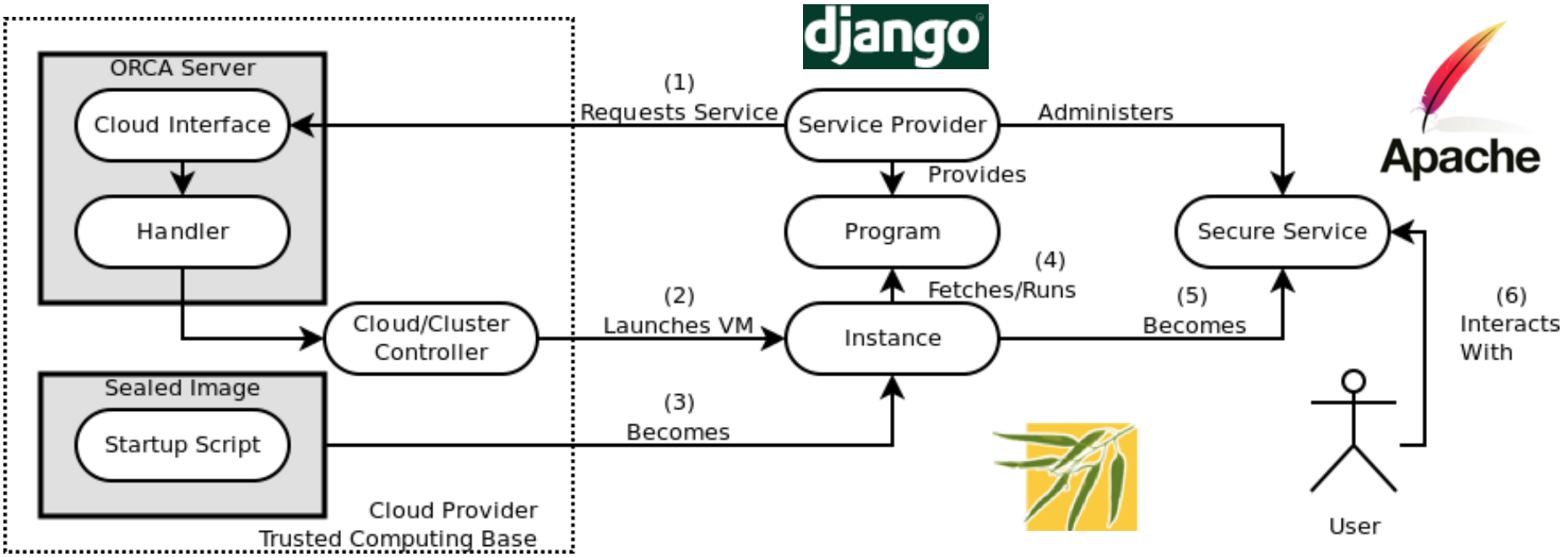


Service Instance Identity (SID): autonomous, independent of owner.

Signed, Sealed, Delivered

- Program (or image) must be signed and endorsed by entity trusted by clients.
- Program is delivered to the cloud sites where it runs, under control of SP (instance owner).
- Cloud provider (AM) seals the instance to prevent post-launch tampering by SP.
  - Administer through well-defined interfaces only.
  - No “log in as root”
  - Cloud provider issues fresh key and attestation.

Example: Trusted Platform as a Service



A Service Provider requests a new instance to run a Python/Django program. ORCA launches a trusted image designated for running secured code. The image has a script that fetches the program, launches it, and seals itself.

Andrew Brown et. JC, **Trusted Platform-as-a-Service: A Foundation for Trustworthy Cloud-Hosted Applications**, CCSW 2011

Toward an "InterCloud" and FIA

- Networked clouds present a range of challenges
  - Identity, trust, governance, policy
  - Resource representation and resource control
- An "InterCloud" must provide some some coordination services trusted by aggregates.
- The trust graphs must match the inter-cloud governance/ agreement structure, which may be complex and fluid.
  - Specify them declaratively with a trust delegation logic.
  - Evolve them according to events at the socio-political layer.
- Attestation of hosted services enables a trustworthy ecosystem of cloud application.
- Next: pricing, economics, and adaptation