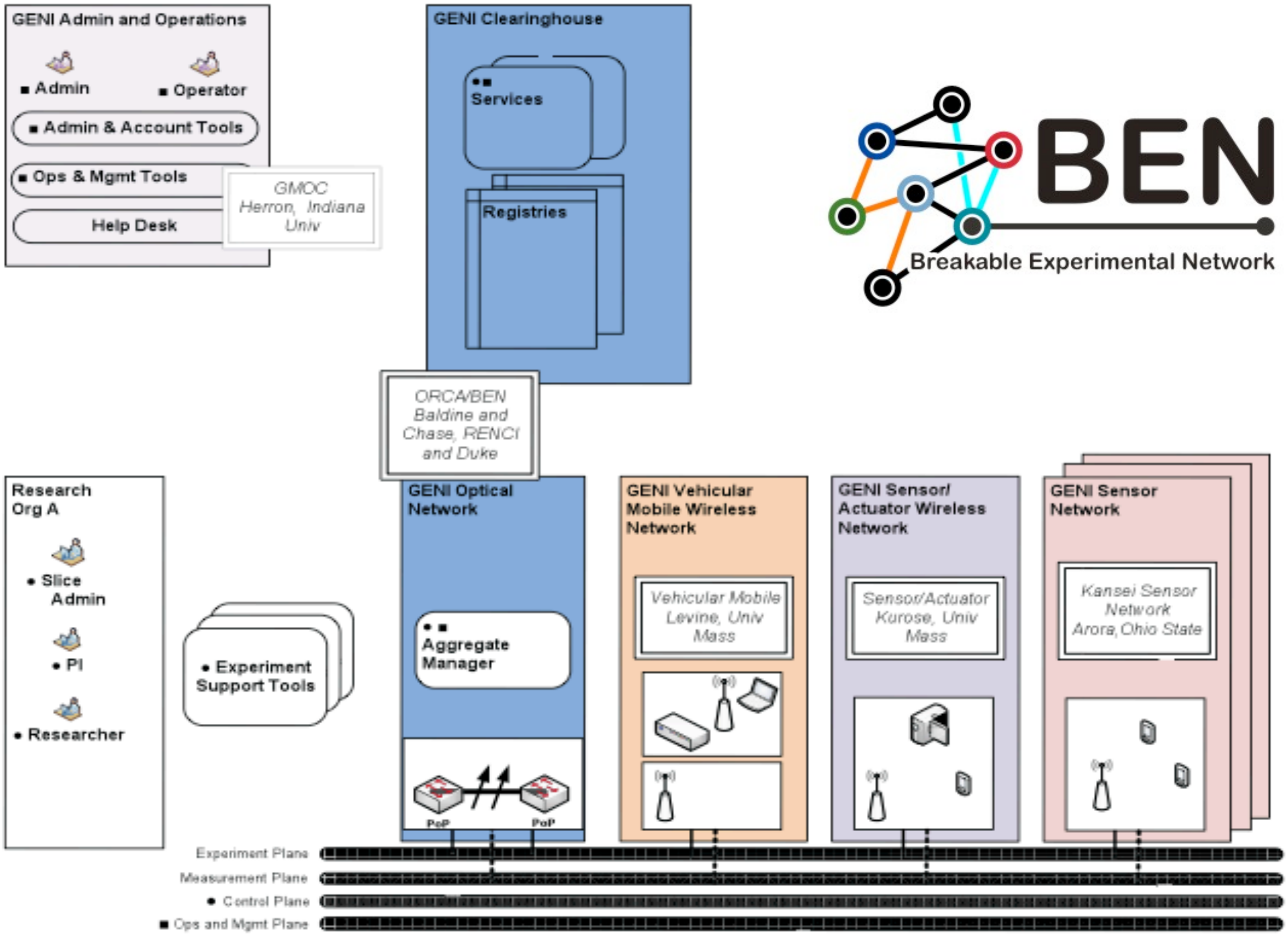
Open Resource Control Architecture (Orca/Shirako) Control Framework

Jeff Chase

FIRE-GENI workshop 12/8/08


GENI Admin and Operations
■ Admin
■ Operator
■ Admin & Account Tools
■ Ops & Mgmt Tools
Help Desk
GMOC Herron, Indiana Univ
GENI Clearinghouse
●■ Services
Registries
ORCA/BEN Baldine and Chase, RENCI and Duke
Research Org A
● Slice Admin
● PI
● Researcher
● Experiment Support Tools
GENI Optical Network
●■ Aggregate Manager
PoP
PoP
GENI Vehicular Mobile Wireless Network
Vehicular Mobile Levine, Univ Mass
GENI Sensor/ Actuator Wireless Network
Sensor/Actuator Kurose, Univ Mass
GENI Sensor Network
Kansei Sensor Network Arora, Ohio State
Experiment Plane
Measurement Plane
● Control Plane
■ Ops and Mgmt Plane

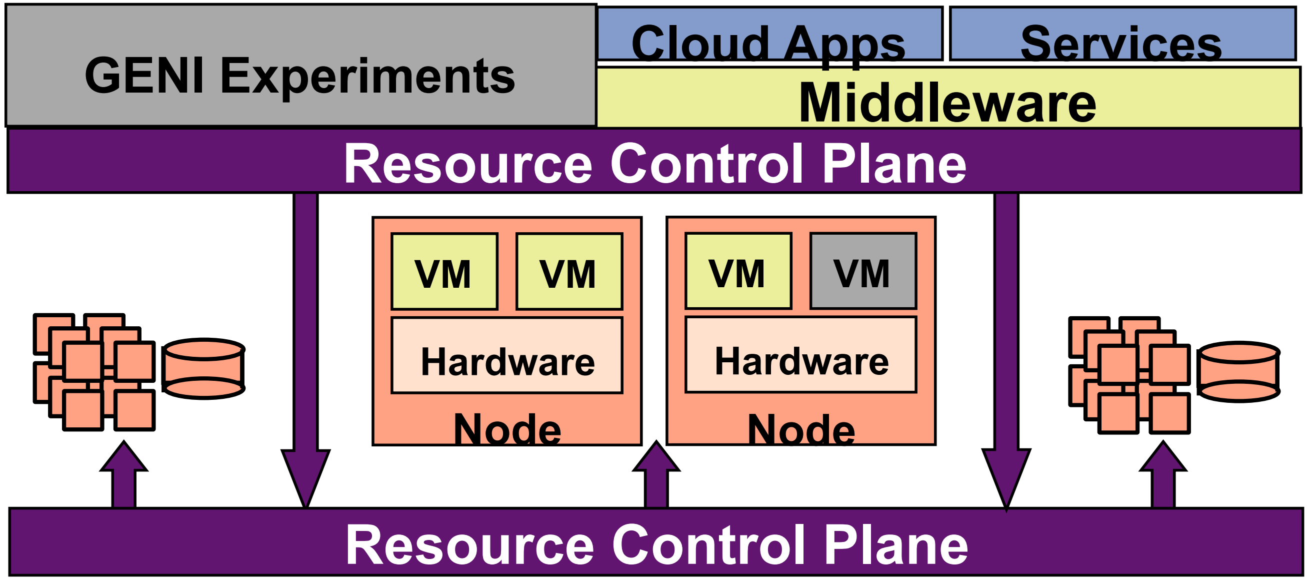
BEN
Breakable Experimental Network

Figure 5-4. Cluster D utilizing ORCA control framework

Cluster-D Substrate

- "Cloud computing" ensembles at the edge
  - Cluster-on-Demand (COD)
  - Eucalyptus (UCSB)
  - NCSU VCL
- Cross-layer network slivering
  - BEN: nested allocation at optical transport and below
  - Configurable IP overlays (e.g., VRF)
- Sensor fields/testbeds
  - Vise (U. Mass Amherst)
  - Kansei (Ohio State)
- Mobility and intermittent connectivity
  - DieselNet (U. Mass Amherst)

Open Resource Control



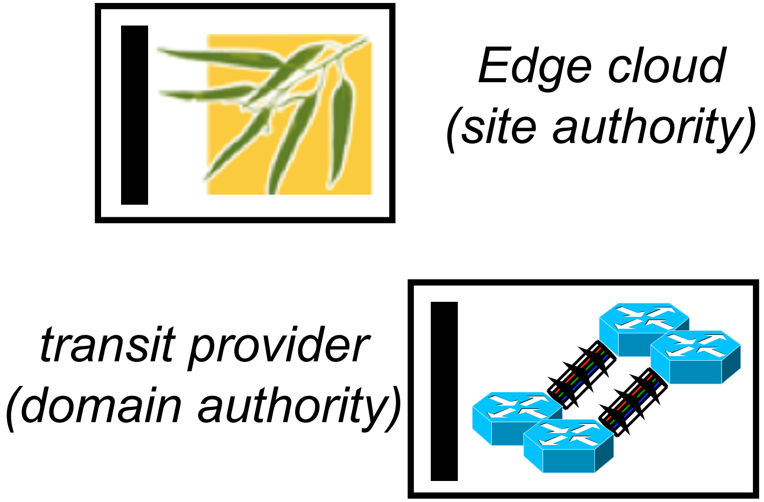
- Contract model for resource peering/sharing/management
- Automated lease-based allocation and assignment
- Flexible resource/contract representations
- Aggregation and accountable delegation
- *http://www.cs.duke.edu/nicl/*

Orca Control Framework

- Open
  - Protocol-centric: servers (*actors*) representing providers (*hosts*) and consumers (*guests*) negotiate lease contracts.
  - Open to innovation: plugin structure for contract exchange, representation, allocation policy, and configuration.
- Resource
  - Allocate/sliver diverse substrate elements
  - “Virtual network resources”
- Control
  - Resource contracts confer specific rights and assurance of isolation; holder controls/programs leased resources.
- Architecture
  - Actor roles, protocols, and interfaces
  - Java-based toolkit implementation: Shirako leasing core

Control Framework: Actors

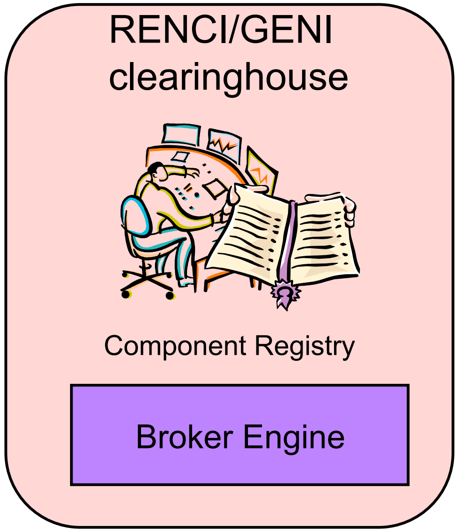
- Principle #1: actors represent primary stakeholders.
  - Provider (host): aggregate manager for an ensemble
  - Consumer (guest): controller acting for researcher
  - Broker represents loose federation of hosts and guests



Edge cloud (site authority)

transit provider (domain authority)

Provider aggregates

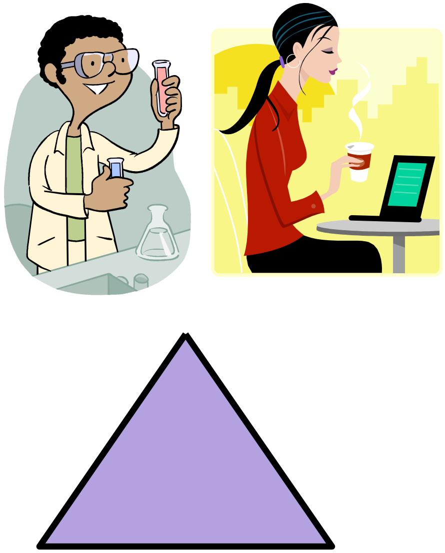


RENCI/GENI clearinghouse

Component Registry

Broker Engine

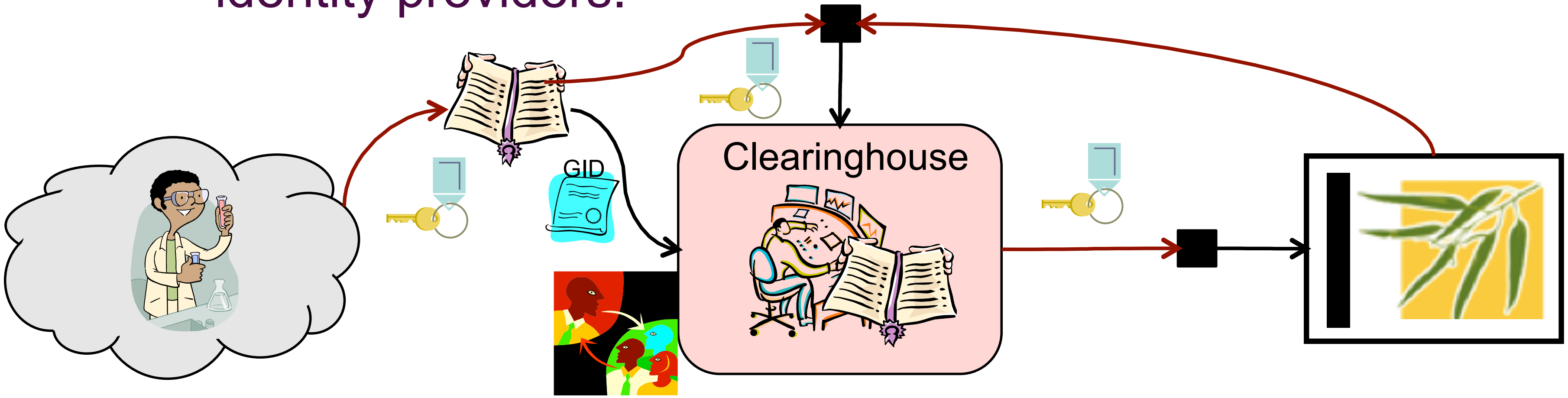
Facility clearinghouse



Guest slice controller

Identity and Trust

- Other entities (e.g., institutions) endorse identities and their security attributes to establish trust chains.
  - No direct role in the control plane protocols, although they may wield powers of the entities they endorse.
- Actor operators register public keys of partners.
  - User and aggregate authority (MA) pick clearinghouse.
  - Clearinghouse registers qualified aggregates and user identity providers.



Control Framework Principles

(2) Actors enter into contracts to lease resources.

- Specific assurances for performance and isolation
- Reservations, best effort, etc.
- Specific time intervals
- Flexible representation
- Accountability

(3) Contract model supports delegation (subcontracting).

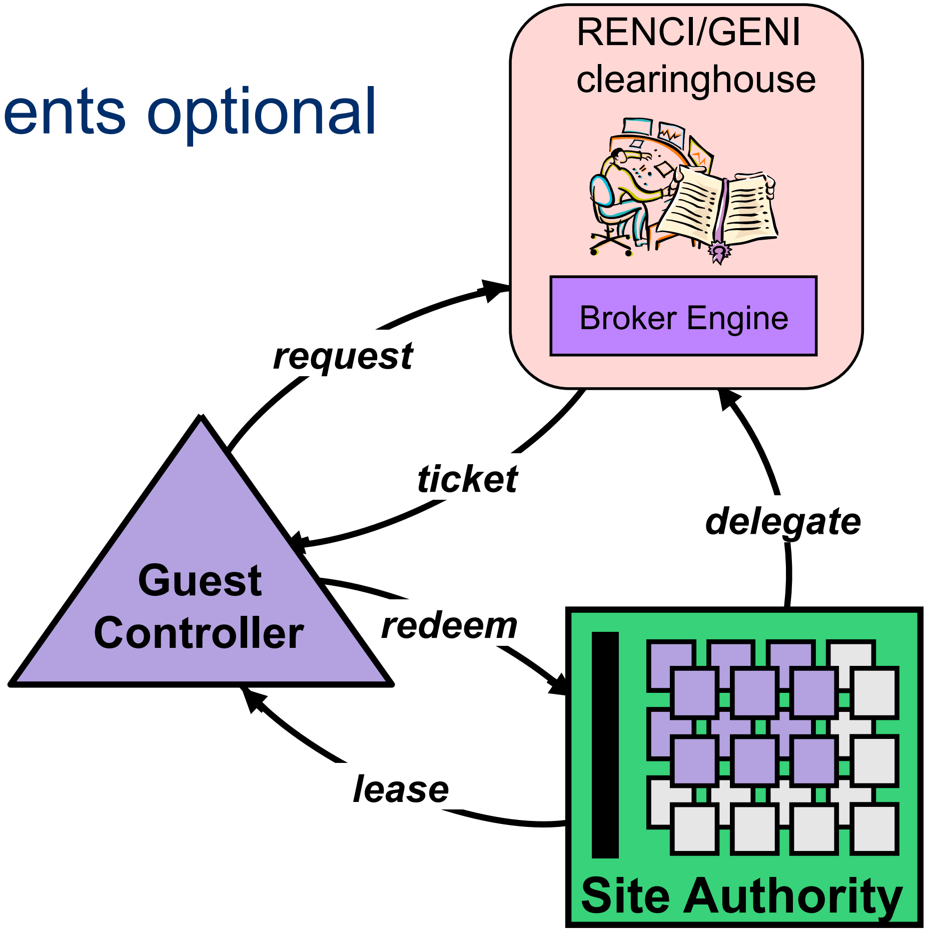
- Providers delegate allocation to clearinghouse
- Ticket broker service with “ticket splitting”

(4) Use aggregation.

- AM is a “wrapper” around existing resource manager technologies for substrate components.

Ticket Broker Service

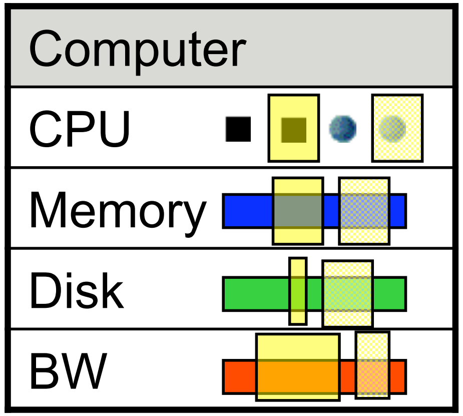
- Orca clearinghouse implements optional GENI *ticket broker service*.
- AMs delegate to broker.
- Broker issues resource tickets to user slices.
- Match request properties.
  - Coordinated allocation
  - Calendar scheduling by user identity or group
- Foundation for discovery
  - resources, paths, topology

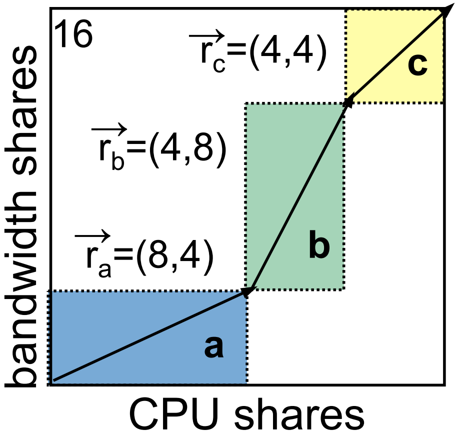


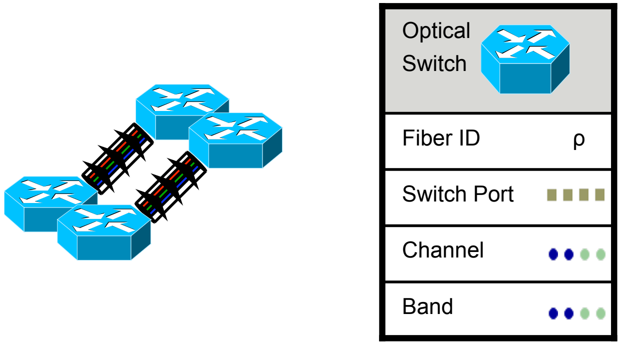
SHARP [SOSP 2003]
w/ Vahdat, Schwab

Issue: Ticket Splitting

- Clearinghouse/broker maintains a registry of components, attributes, and relationships.
- Some aggregates are pools of interchangeable instances of a given type.
  - E.g., edge cluster cells, storage
- Broker plugins process representations and issue tickets for subsets that match guest requests.

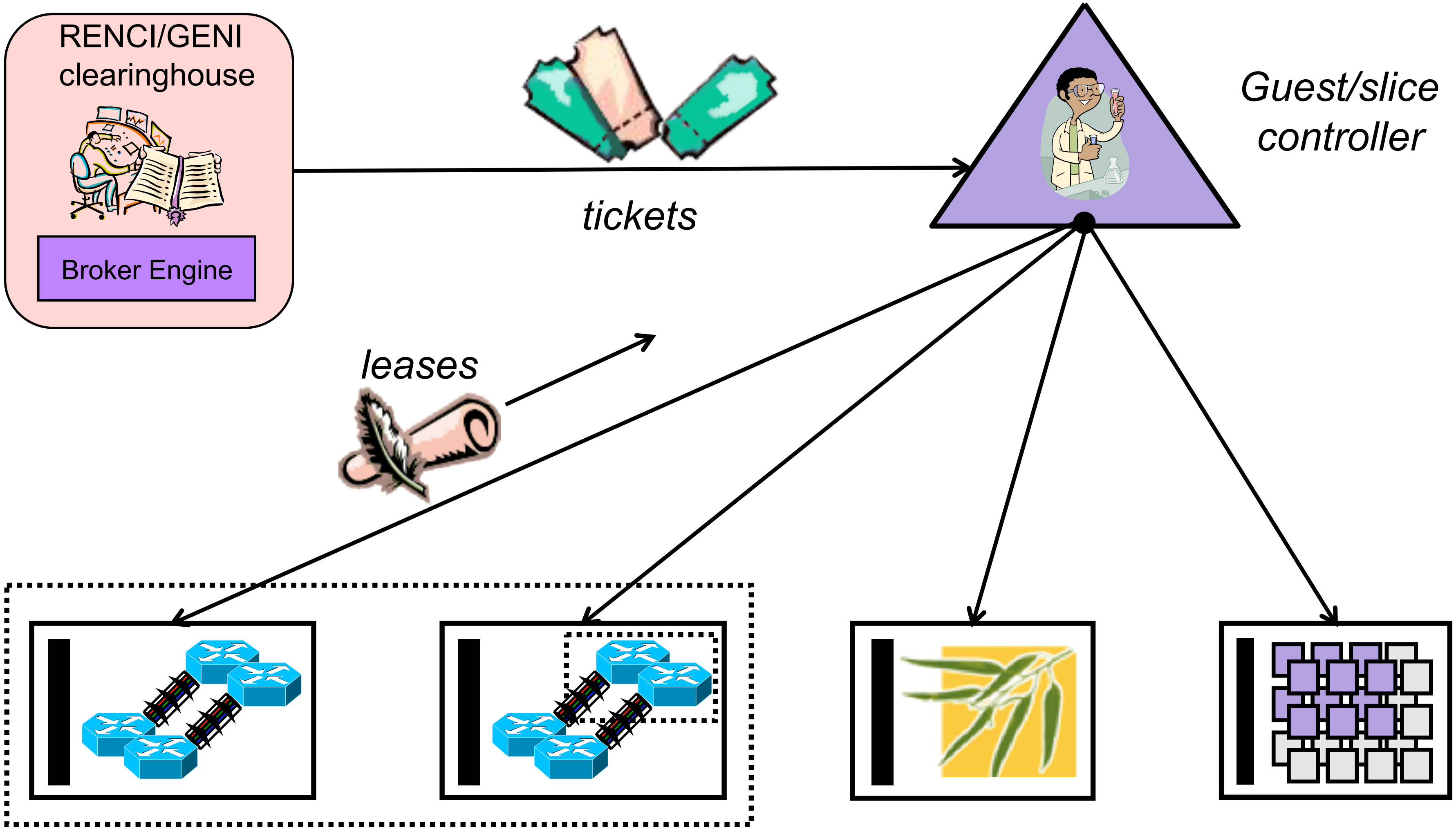






NDL + GMPLS?

Slice Setup



RENCI/GENI clearinghouse

Broker Engine

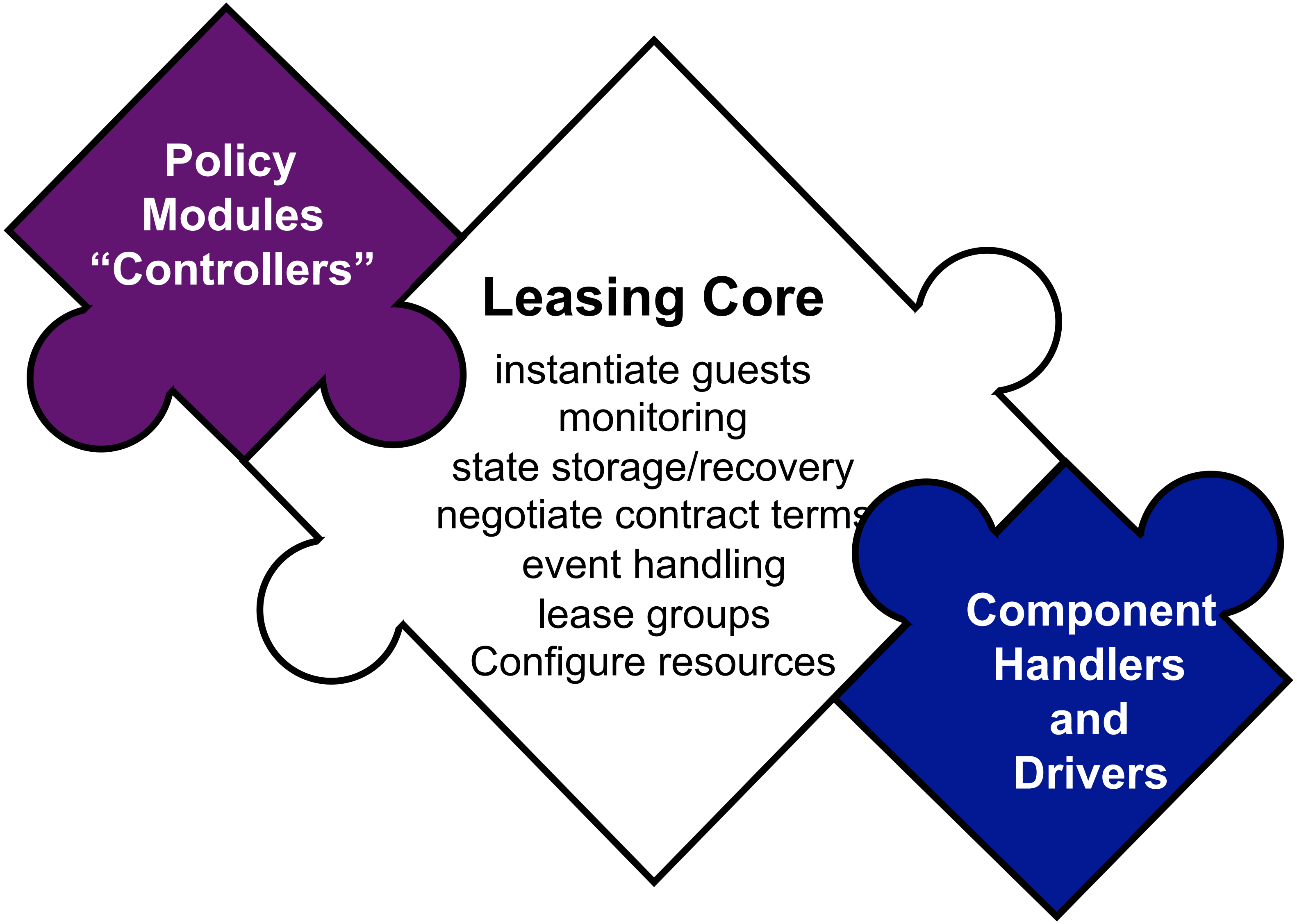
tickets

Guest/slice controller

leases

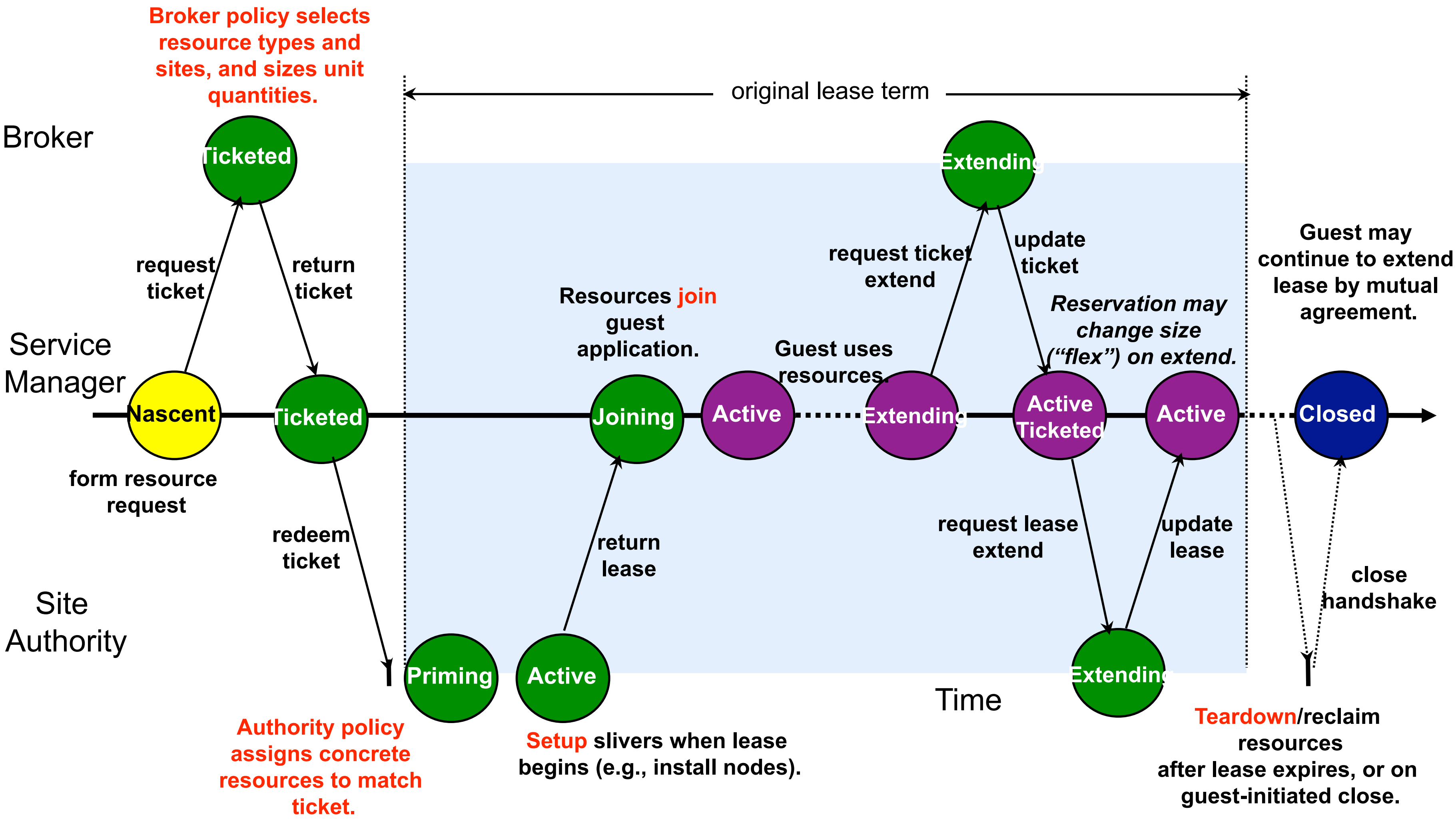
Exchange of labels, tokens, configuration attributes, etc.

Pluggable Resources and Policies



Leasing Core

instantiate guests
monitoring
state storage/recovery
negotiate contract terms
event handling
lease groups
Configure resources



Broker policy selects resource types and sites, and sizes unit quantities.

original lease term

Broker

Ticketed

request ticket

return ticket

Service Manager

Nascent

form resource request

Ticketed

redeem ticket

Site Authority

Authority policy assigns concrete resources to match ticket.

Priming

Active

Setup slivers when lease begins (e.g., install nodes).

return lease

Resources join guest application.

Joining

Active

Guest uses resources.

Extending

request ticket extend

Extending

update ticket

Reservation may change size (“flex”) on extend.

Active Ticketed

request lease extend

Extending

update lease

Active

Time

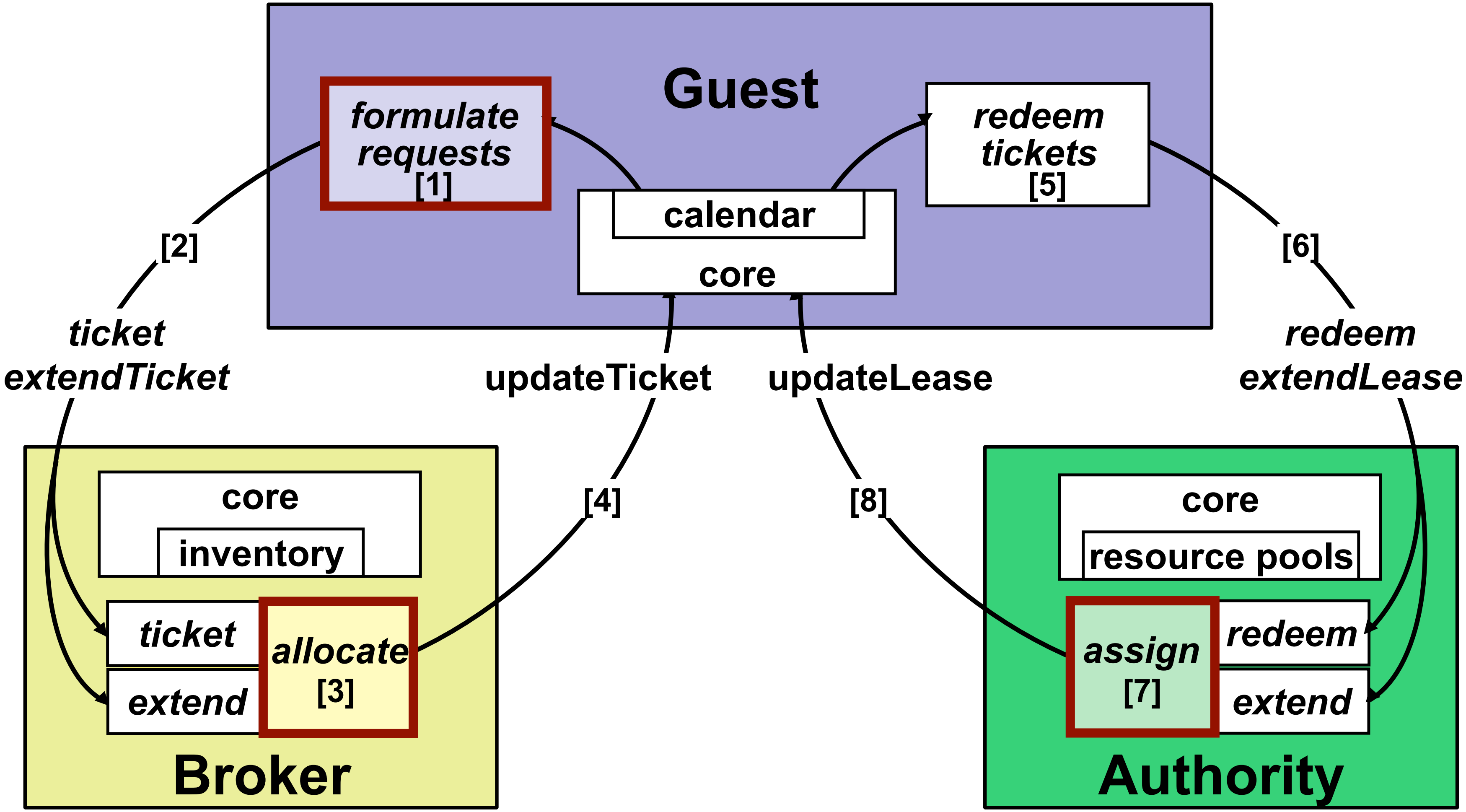
Guest may continue to extend lease by mutual agreement.

Closed

close handshake

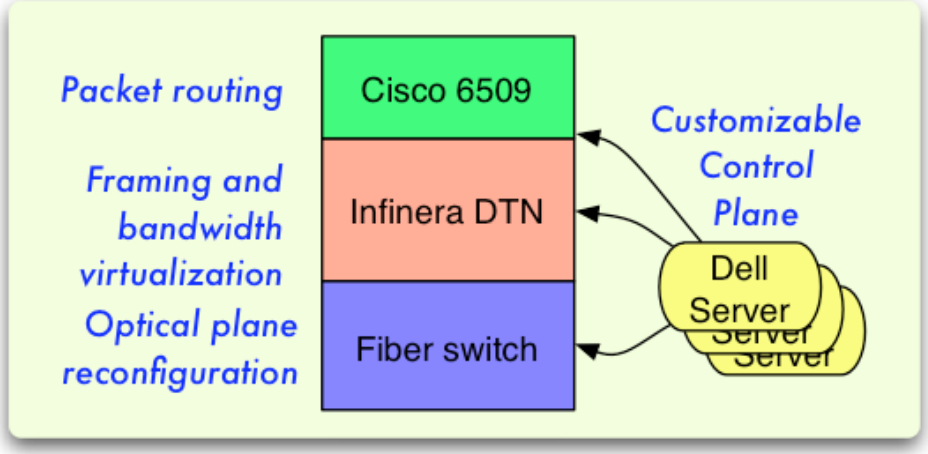
Teardown/reclaim resources after lease expires, or on guest-initiated close.

Orca: Actors and Protocols



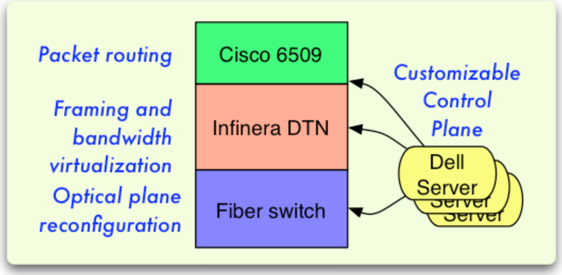
Breakable Experimental Network (BEN)

- Research/Triangle optical network
- RENCI PoPs, e.g., campus RECs
  - Duke, NCSU, UNC, MCNC
  - Cisco/MPLS bridge to campus/NLR
- Sliverable at multiple levels
  - From the fiber up (BYOT)
  - Lambda WDM + TDM
  - Infinera Bandwidth Virtualization
  - VLANs and flows


Packet routing
Framing and bandwidth virtualization
Optical plane reconfiguration
Cisco 6509
Infinera DTN
Fiber switch
Customizable Control Plane
Dell Server
Server
Server

Experimentation w/ Orca/BEN

- Extend Orca to enable slivering of
  - Network elements:
    - Fiber switches
    - DWDM equipment


Packet routing
Framing and bandwidth virtualization
Optical plane reconfiguration
Cisco 6509
Infinera DTN
Fiber switch
Customizable Control Plane
Dell Server
Server
Server

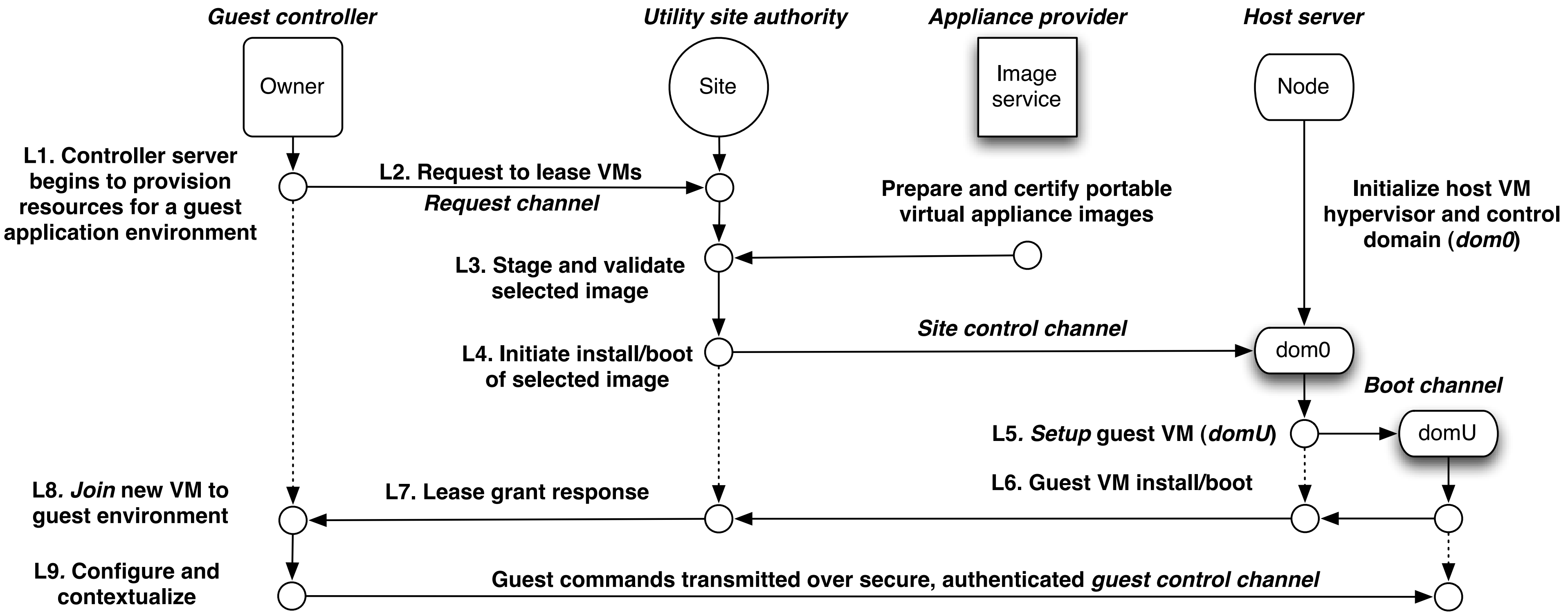
- Adapt mechanisms to enable flexible description of network slices
  - NDL
- Demonstrate end-to-end slicing on BEN
  - Create realistic but “prefab” slices containing compute, storage and network resources
  - Launch sample applications that expose topology properties
- Link to self-contained edge resources, e.g., over NLR

Thanks!

Some Observations

- The Classic Internet is “just an overlay”.
  - GENI is underlay architecture (“underware”)…an exokernel for the Internet.
- Incorporate edge resources: GENI is “cloud computing” + sliverable network
- Multiple domains (MAD): not a “Grid”, but something like dynamic peering contracts
  - Decouple services from substrate; manage the substrate; let the services manage themselves.
- Requires predictable (or at least “discoverable”) allocations for reproducibility
  - QoS at the bottom or not at all?

Example: Guest VM Setup



Guest controller

Utility site authority

Appliance provider

Host server

L1. Controller server begins to provision resources for a guest application environment

L2. Request to lease VMs

Request channel

Prepare and certify portable virtual appliance images

Initialize host VM hypervisor and control domain (*dom0*)

L3. Stage and validate selected image

Site control channel

L4. Initiate install/boot of selected image

Boot channel

L5. *Setup* guest VM (*domU*)

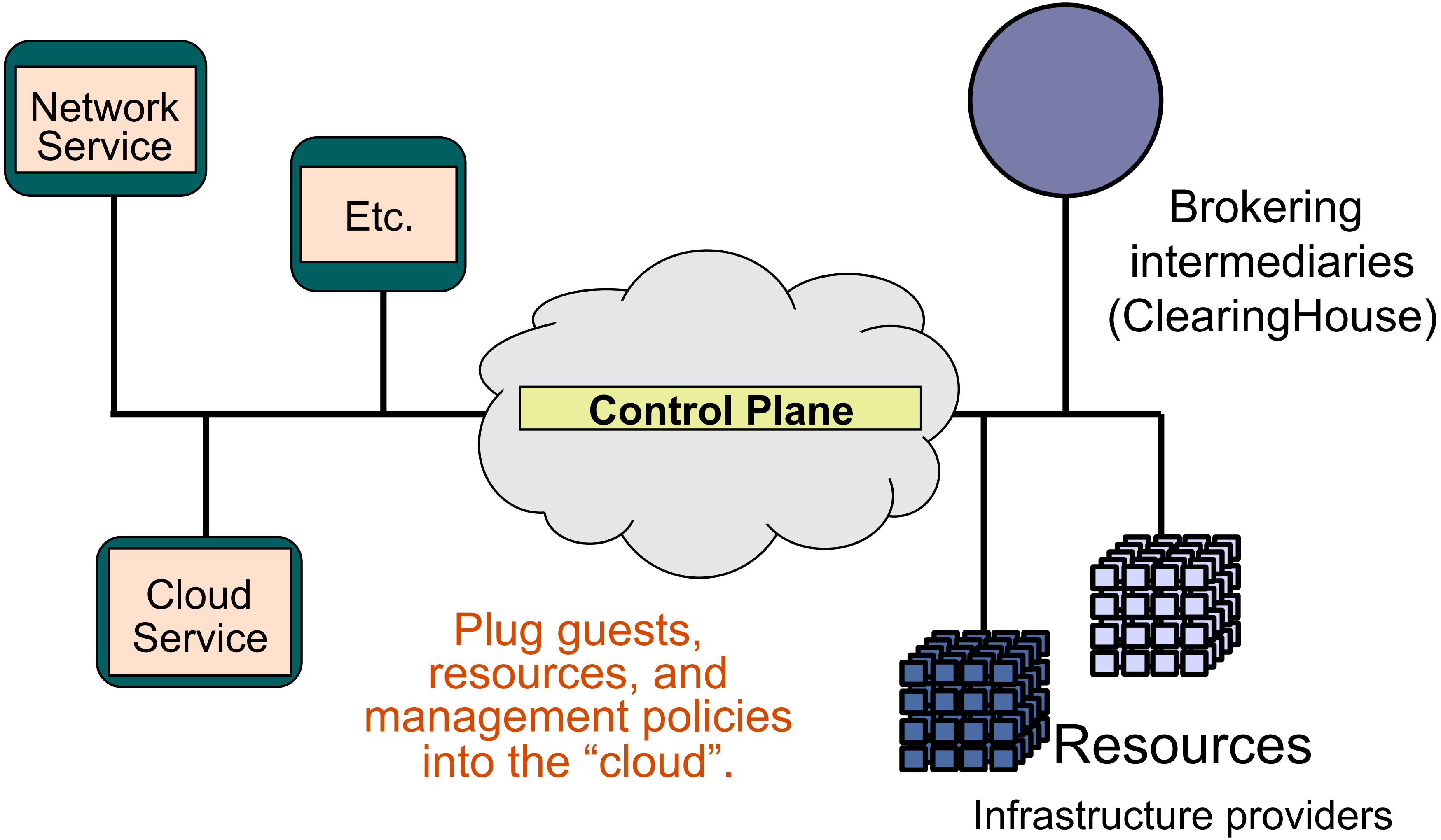
L8. *Join* new VM to guest environment

L7. Lease grant response

L6. Guest VM install/boot

L9. Configure and contextualize

Guest commands transmitted over secure, authenticated *guest control channel*



Plug guests, resources, and management policies into the “cloud”.

Delegation

- Principle #3: Contracts enable delegation of powers.
  - Delegation is voluntary and provisional.
- It is a building block for creating useful concentrations of power.
  - Creates a potential for governance
  - Calendar scheduling, reservation
  - Double-edged sword?
    - Facility can Just Say No

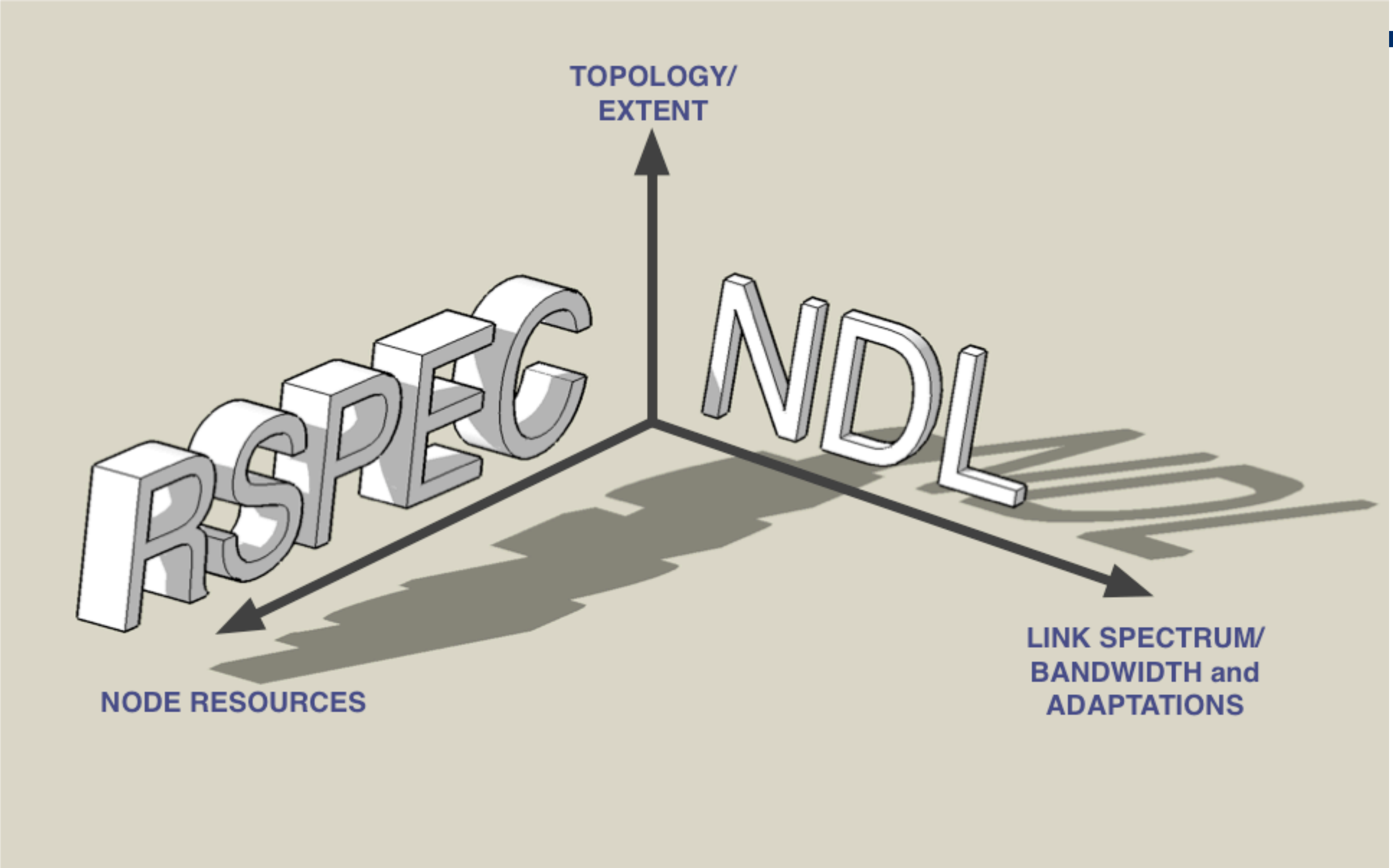
Aggregation

- Principle #4: aggregate the resources for a site or domain.
  - Primary interface is domain/site authority
- Abstraction/innovation boundary
  - Keep components simple
  - Placement/configuration flexibility for owner
  - Mask unscheduled outages by substitution
  - Leverage investment in technologies for site/domain management

Network Description Language

```
<ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:501/3">
  <ndl:name>tdm3.amsterdam1.netherlight.net:501/3</ndl:name>
  <ndl:connectedTo
    rdf:resource="http://networks.internet2.edu/manlan/manlan.rdf#manlan:if1"/>
  <ndl:capacity
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.244E+9</ndl:capacity>
</ndl:Interface>
<ndl:Interface rdf:about="http://networks.internet2.edu/manlan/manlan.rdf#manlan:if1">
    <rdfs:seeAlso rdf:resource="http://networks.internet2.edu/manlan/manlan.rdf"/>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#”
   xmlns:ndl="http://www.science.uva.nl/research/sne/ndl#”
   xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
<!-- Description of Netherlight -->
<ndl:Location rdf:about="#Amsterdam1.netherlight.net">
   <ndl:name>Netherlight Optical Exchange</ndl:name>
   <geo:lat>52.3561</geo:lat>
   <geo:long>4.9527</geo:long>
</ndl:Location>
<!-- TDM3.amsterdam1.netherlight.net -->
<ndl:Device rdf:about="#tdm3.amsterdam1.netherlight.net">
      <ndl:name>tdm3.amsterdam1.netherlight.net</ndl:name>
      <ndl:locatedAt rdf:resource="#Amsterdam1.netherlight.net"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/1"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/2"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/3"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/4"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:502/1"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:502/2"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:502/3"/>
>
```


TOPOLOGY/
EXTENT
RSPEC
NDL
NODE RESOURCES
LINK SPECTRUM/
BANDWIDTH and
ADAPTATIONS

Elements of Orca Research Agenda

- Automate management inside the cloud.
  - Programmable guest setup and provisioning
- Architect a guest-neutral platform.
  - Plug-in new guests through protocols; don't hard-wire them into the platform.
- Design flexible security into an open control plane.
- Enforce fair and efficient sharing for elastic guests.
- Incorporate diverse networked resources and virtual networks.
- Mine instrumentation data to pinpoint problems and select repair actions.
- Economic models and sustainability.