AEROSPACE
Assuring Space Mission Success

Cloud Service Level Agreements and Cloud Federation

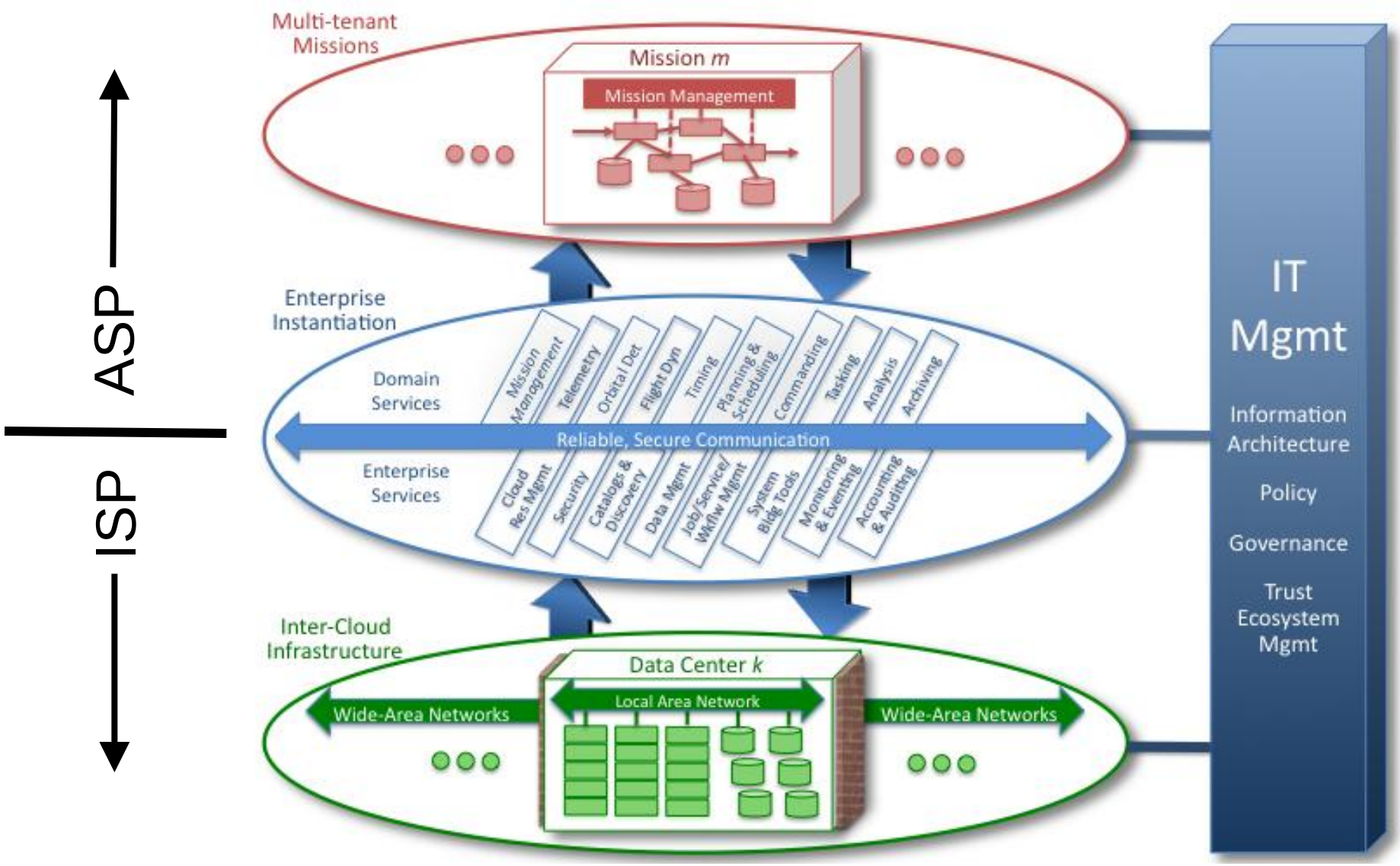
Dr. Craig A. Lee, Senior Scientist, lee@aero.org
The Aerospace Corporation

GSAW, February 26, 2014

Introduction

- NIST Definition of Cloud Computing
  - *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*
- Potential Benefits
  - *Improved mission-effectiveness*
    - Improved reliability using on-demand resources to recover after failure
    - Surge capacity provided using rapid, on-demand elasticity
    - Improved access to data sets, services, and other resources
  - *Improved cost-effectiveness*
    - Economies of scale achieved through consolidation
    - Generic hosting environment provided for many missions
    - Reduced power, space, cooling, physical infrastructure requirements

A Cloud-Based Reference Model



Issues and Observations

- How can we manage sets of applications that will have differing resource and performance requirements?
  - *How to ensure that mission goals are met (e.g., throughput), while ensuring that aggregate cloud requirements are met (e.g., overall server utilization)?*
  - *Individual and aggregate requirements could be competing and conflicting*
  - *Service Level Agreements (SLAs) are commonly considered to address the issue of individual requirements*
- How to Manage Security and Governance across Organizations
  - *Distributed Infrastructures and Organizations*
  - *Federated Authentication & Authorization Are Critical*
  - *Virtual Organizations (VOs) have been used to address this issue*

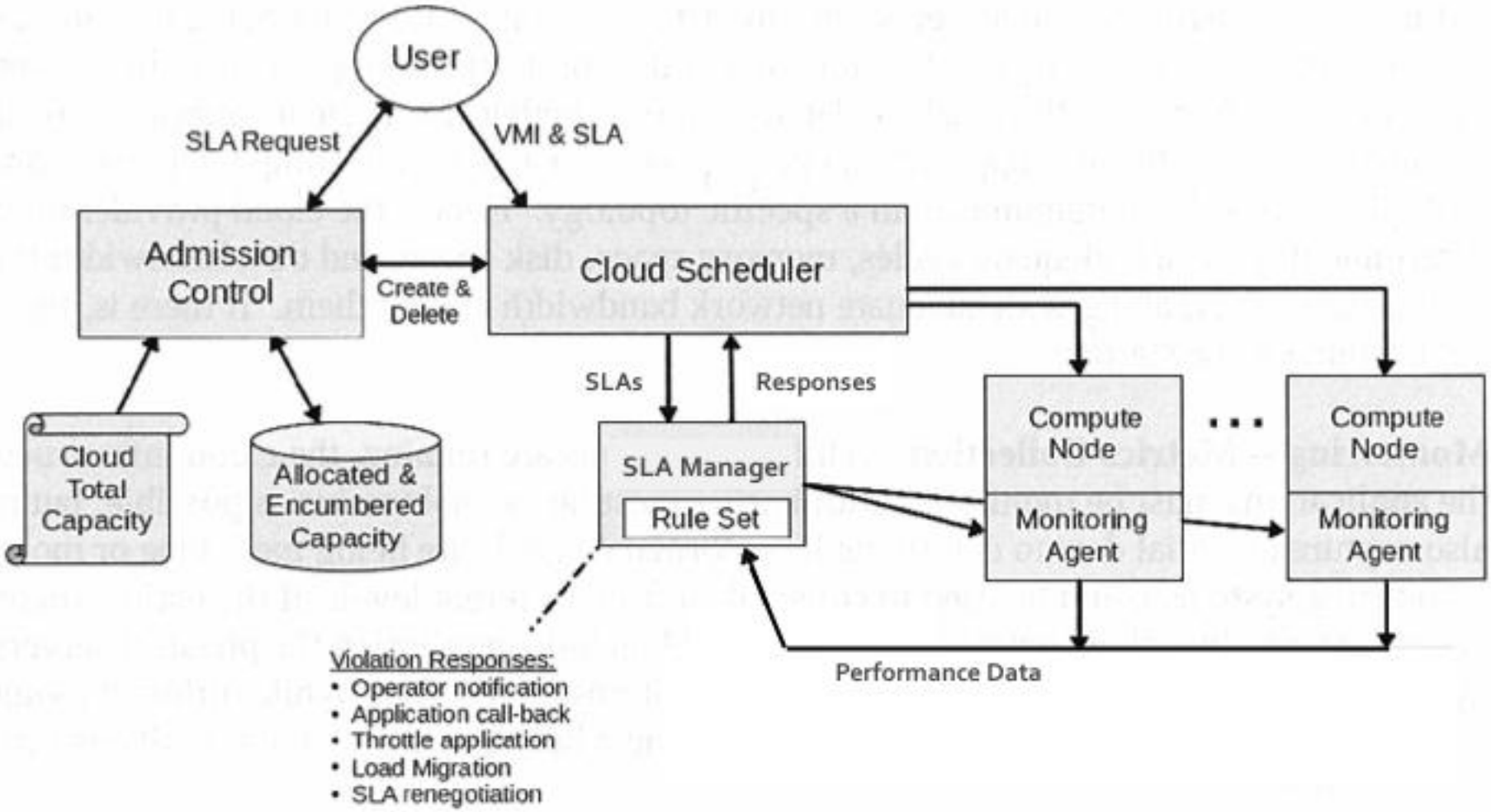
More About SLAs

- Some applications will be performance-critical or performance-sensitive
  - *"Best effort" cloud resources may not suffice to meet app requirements*
- Some applications will have dynamic requirements
  - *Some apps will have varying demands – surge*
  - *Some apps may surge at unpredictable times*
- Previously addressed by over-provisioning with dedicated hardware
  - *Dedicated system was sized for the worst-case, rather than the average case*
  - *Drove entire system cost*
- This is antithetical to cloud computing
  - *Multi-tenant environment where utilization and costs can be better managed*
- ***Hence, the goal is to provide the user with a reasonable expectation that performance requirements will be met, through mechanisms that are reasonable for the provider to implement and support for multiple apps***
- Dynamic, machine-enforceable SLAs
  - *These are not simply "contractual" SLAs*
  - *These are services that a provider may provide and a user may use*

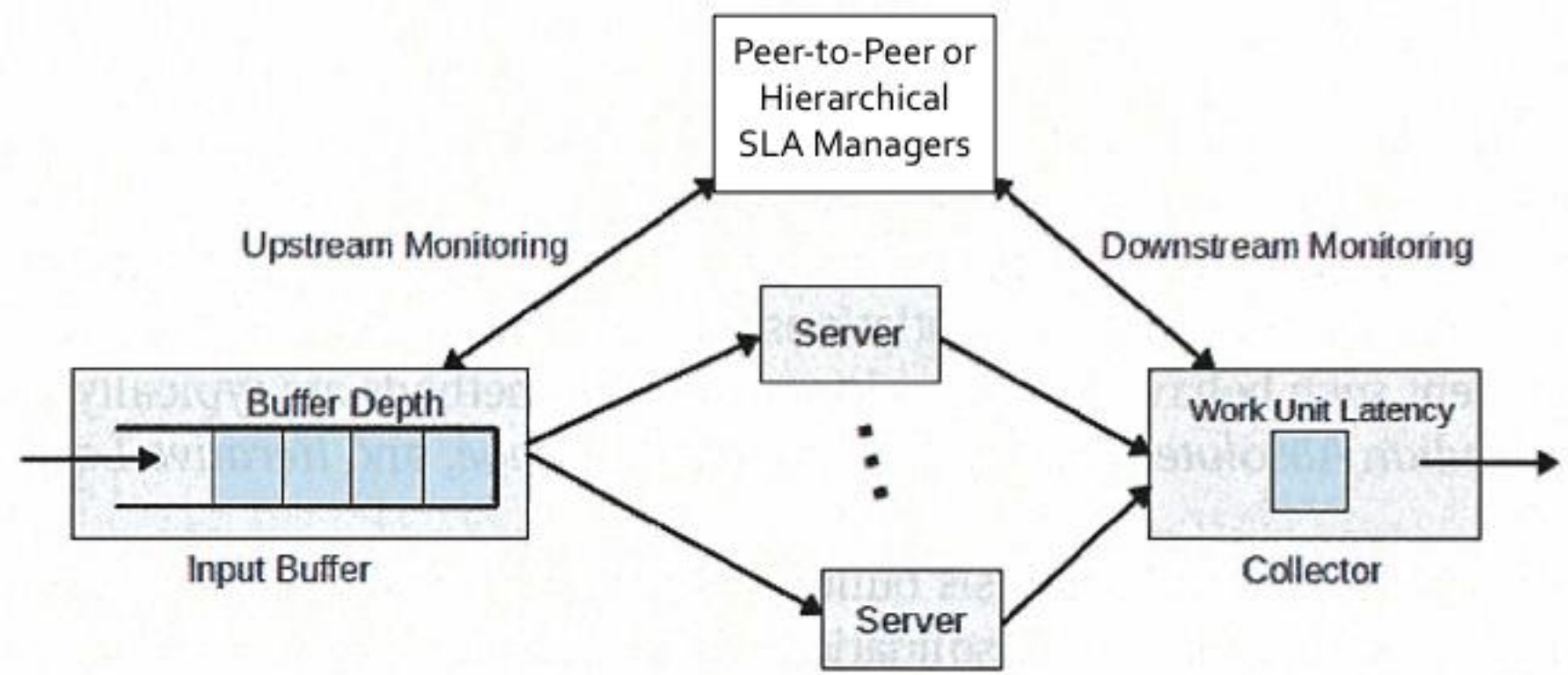
Basic SLA Functions
– *an Autonomic Control Cycle*

- Admission Control
  - *Mapping of app-level requirements to infrastructure-level metrics*
  - *WS-Agreement and WS-Agreement Negotiation*
  - *Term language needed*
- Monitoring - Metrics Collection
  - *Where: host OS/hypervisor, guest OS, application-level*
  - *When: upstream vs. downstream*
- SLA Evaluation
  - *Hysteresis*
  - *Statistical methods, e.g., Median Absolute Deviation, Interquartile Range, Iterative Local Regression*
- SLA Enforcement -- Violation Response
  - *Throttling*
  - *Load migration – both process and VM*
  - *On-demand resources*
  - *SLA re-negotiation*

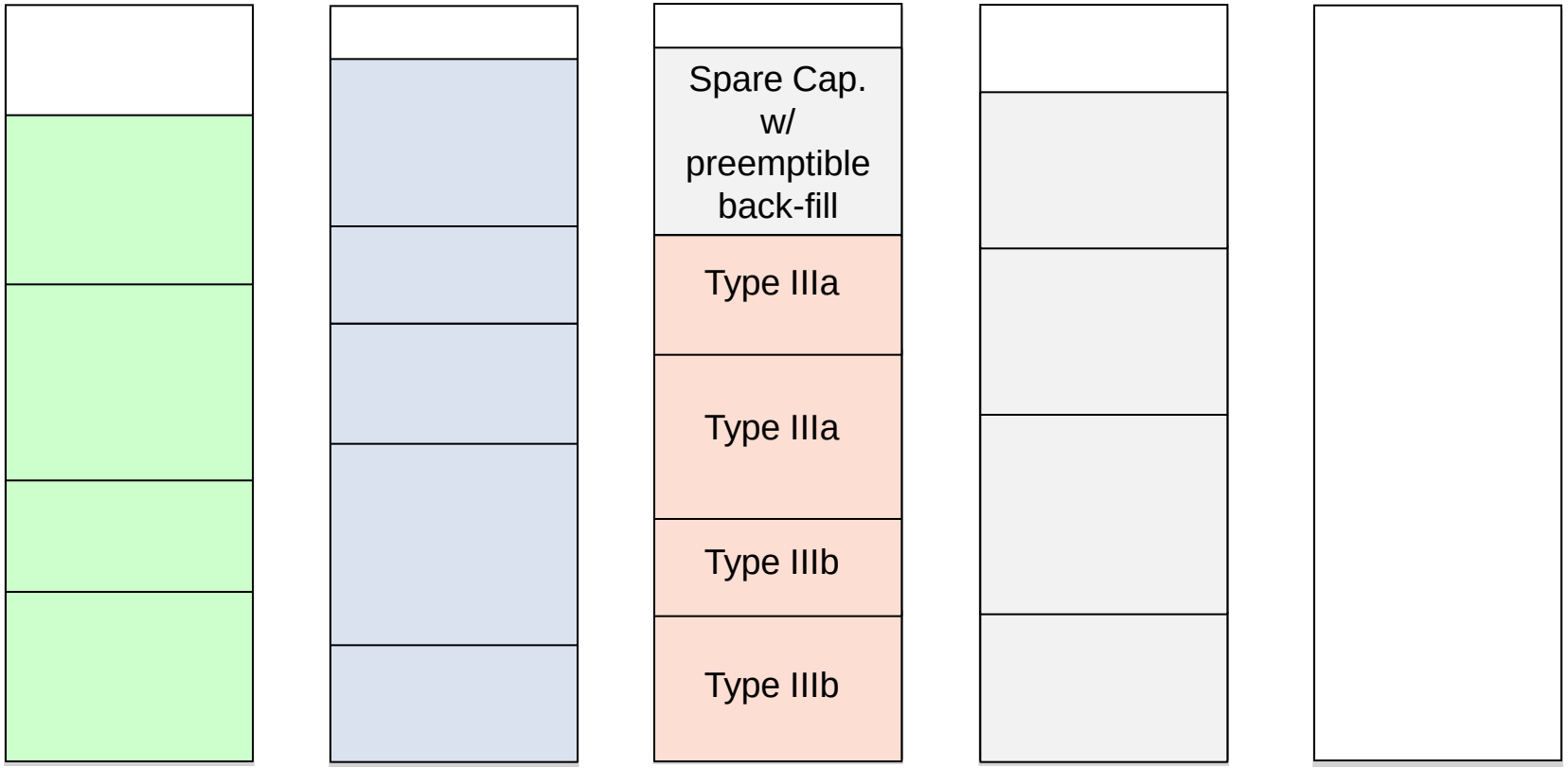
A General SLA Architecture



Distributed Monitoring of Different Mission Components



Server Load Types



Type I
Best Effort

Type II
Strict Throttling

Type III
Live Migration w/ back-fill

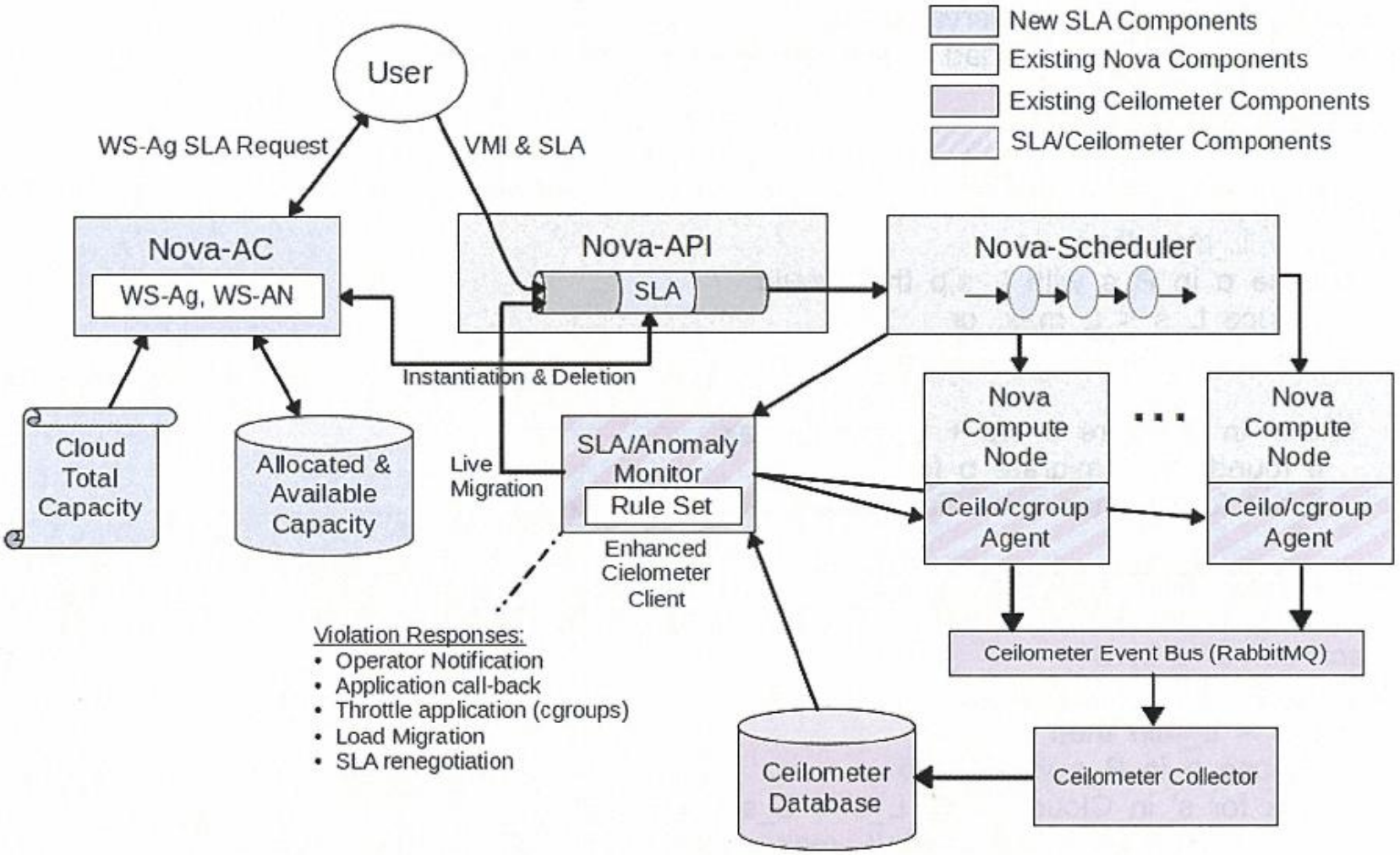
Type IV
Preemptible

Type V
Idle

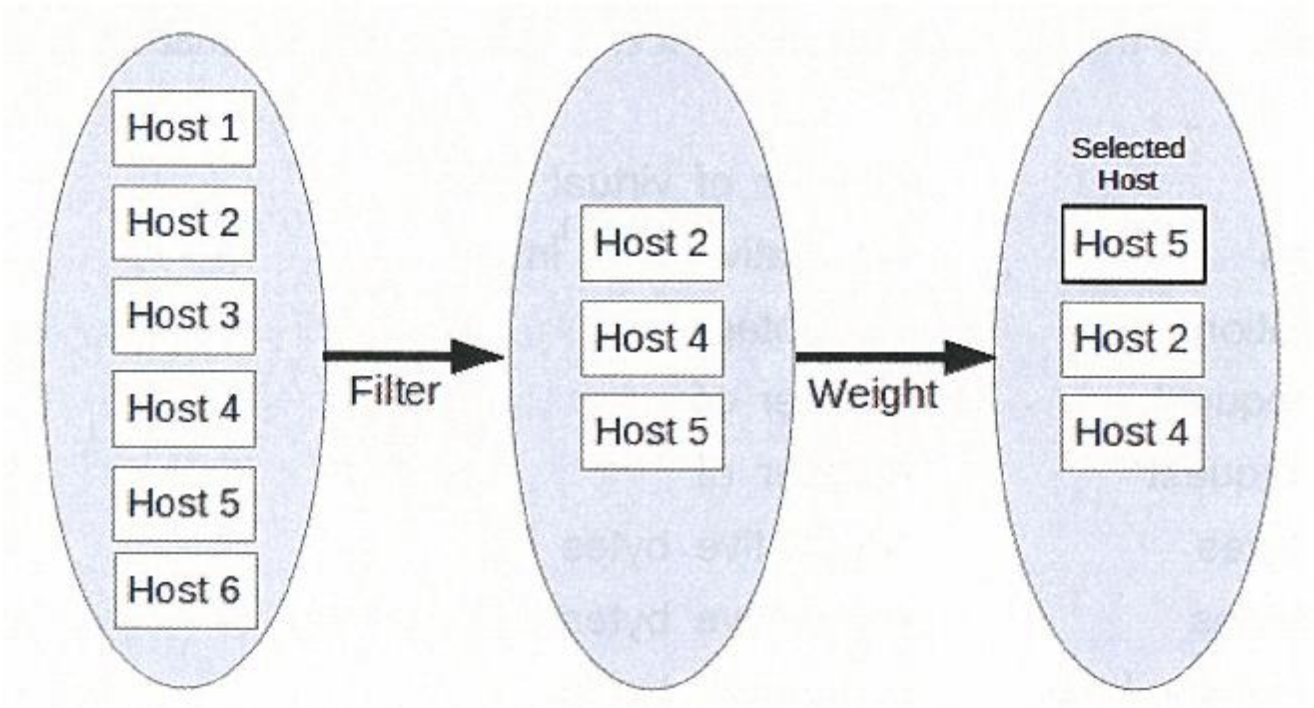
Migration Overhead Metrics & Policies

- Overhead Metrics
  - *Memory footprint*
  - *File descriptors-- network/disk connections*
  - *Accurate migration overhead model needed*
- Possible Policies: what to move where
  - *First Fit Decreasing*
  - *Fastest Migration Time (least time needed)*
  - *Maximum Load Reduction*
  - *Load Reduction to Just Below Maximum*
  - *Highest Correlation with Causing Excessive Load*
  - *Application Value (priority)*
  - *Application Availability*

An SLA Architecture for OpenStack



Nova’s Filter and Weight Design



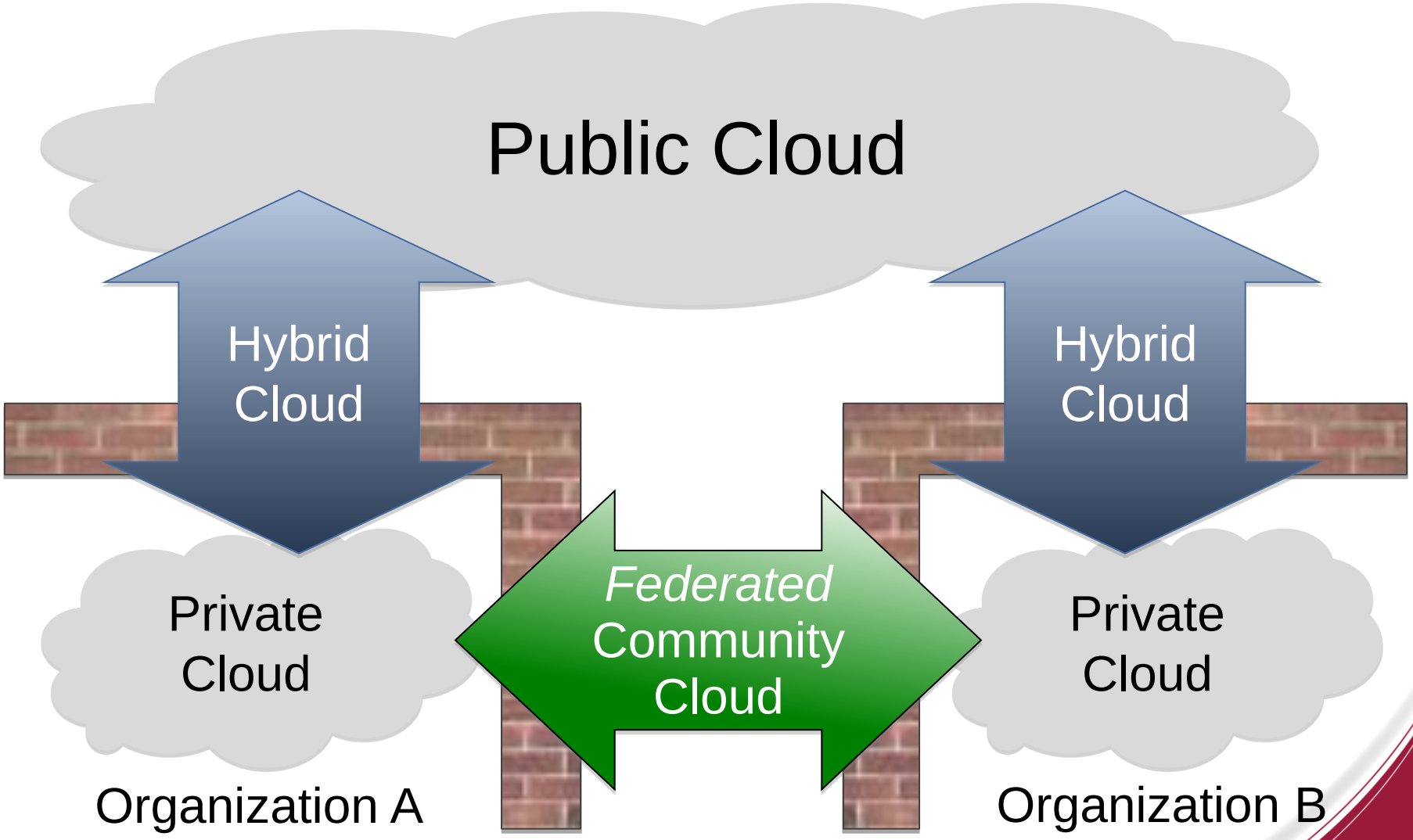
Comments About Cloud SLAs

- Development & Test plan needed
  - *What are the simplest SLA mechanisms that "scratch the itch" for the most users?*
- Capacity Planning & Management
  - *How to estimate query requirements, load demand, time-to-completion*
  - *How to support reasonable load requirements to produce reasonable times-to-completion*
  - *How to manage sets of users such that no one user is disruptive*
  - *How to on-board requirements from other organizations*
- Cyber-security Implications
  - *As clouds become larger and more widely used, there will be more automated tools, i.e., autonomic behaviors*
  - *Autonomic agents become a threat surface -- compromising an agent that controls system behavior would have broad impact*
- Track and exploit on-going SLA work in the marketplace
  - *OGF WS-Agreement, WS-Agreement Negotiation*
  - *TeleManagement Forum (TMF), IEEE, EU, ...*
  - *NSF Center for Cloud and Autonomic Computing*

Changing Gears: Why Federate Clouds?

- Fundamental Cloud Technology:
  - *On-Demand Provisioning of Resources, e.g., servers, storage, communication, platforms, services*
- The Leap: *a Global, Inter-Cloud*:
  - *Everything is available anywhere, anytime, securely, transparently, without having to worry about infrastructure*
- Fundamental cloud technology says *nothing* about:
  - *Distributed Data and Workflow Management*
  - *Wide-Area Network Management*
  - *Federated Identity Management*
  - *Single Sign-on*
  - *Delegation of Trust*
  - *Virtual Organizations*
  - *Managing the Trust Ecosystem*
- This requires *Cloud Federation*

Cloud Deployment Modes



Use Case: Disaster Response Using Federated Clouds

NGA/NCOIC GeoInt Community Cloud Prototype (GCC) demonstrated the integration of cloud resources and geospatial information tools in response to a Haiti-like earthquake event. Participants: Aerospace, NJVC, Boeing, Raytheon, Telos, Winthrop Mgmt, OGC.

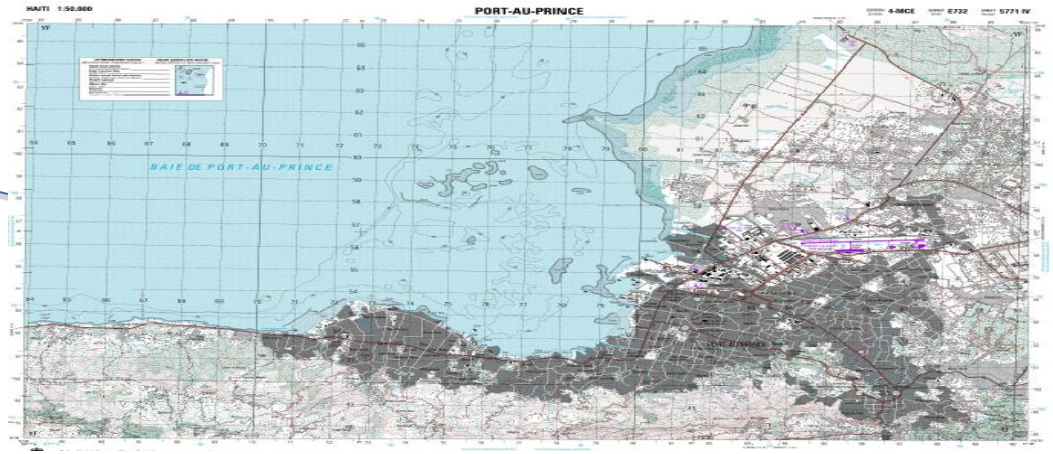
Stakeholder Cloud #1

Stakeholder Cloud #2

Stakeholder Cloud #3

First Responders

On-Demand Federated Cloud



Port-au-Prince, Haiti

Use Case: GEOSS

- Ten year plan to build out architecture with data management tools for full and open exchange of data, metadata and data products, recognizing national and international policies and legislation.
- http://www.earthobservations.org

Necessary Security Capabilities for Federation

- Federated Identity Management (Authentication – AuthN)
  - *"Are you who you say you are?"*
  - *"Where are you from? Who is your Identity Provider?"*
  - *"Can I trust your Identity Provider?"*
  - *"How do I interact with your Identity Provider to verify your identity?"*
- Federated Access Control (Authorization - AuthZ)
  - *In a distributed env, how are user identities mapped to authorization privileges?*
  - *Use a trusted Attribute Server to manage dynamic sets of authorization attributes*
    - Provides a "security context" to manage sets of authorization attributes that are not tied to any particular home institution
    - Authorization attributes must be well-known, or negotiated and agreed upon
  - *This is the Virtual Organization (VO) concept*
    - Could also be called *workspaces, enclaves, …*
- Trust Federations
  - *Establishes trust among potential federation participants*
  - *Example: International Grid Trust Federation, www.igtf.net*
    - Provides "Good Housekeeping Seal of Approval" to CA operators
    - Certificates from approved CAs are trusted among IGTF participants

Federated Authorization Management with Virtual Organizations (VOs)

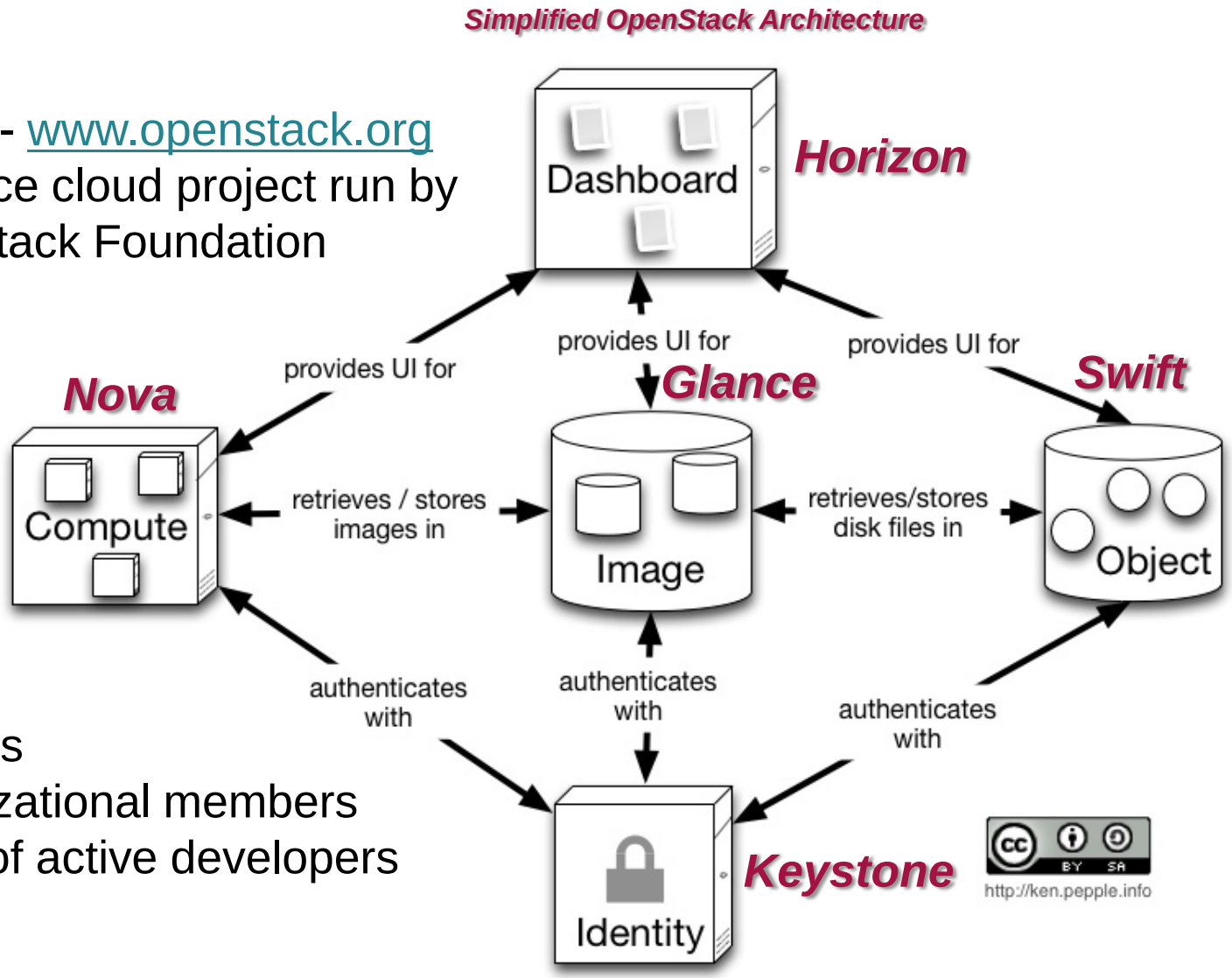
- A VO is a security and collaboration context not exclusively associated with any one physical organization or site
  - *Participating partners agree upon structure, rules and processes*
  - *A VO partner can be a single person, a group or an entire organization*
- A VO has members that are assigned roles and/or attributes
  - *Membership roles or attributes grant specific capabilities within a given VO as determined by each resource/service provider*
- Partners participating in a VO contribute resources, i.e., data and services
  - *They retain complete control over their own resources!*
  - *Access by VO members can be modified or revoked at any time by both the VO administrator and the resource administrator*
- VOs enable federated, community clouds by being the Trusted Third Parties who assert user identity attributes, and who may authenticate users as well

Intellectual Heritage:
VOs Developed for Global Grids

OpenStack: an Open Source Cloud Project

OpenStack -- www.openstack.org

- Open source cloud project run by the OpenStack Foundation

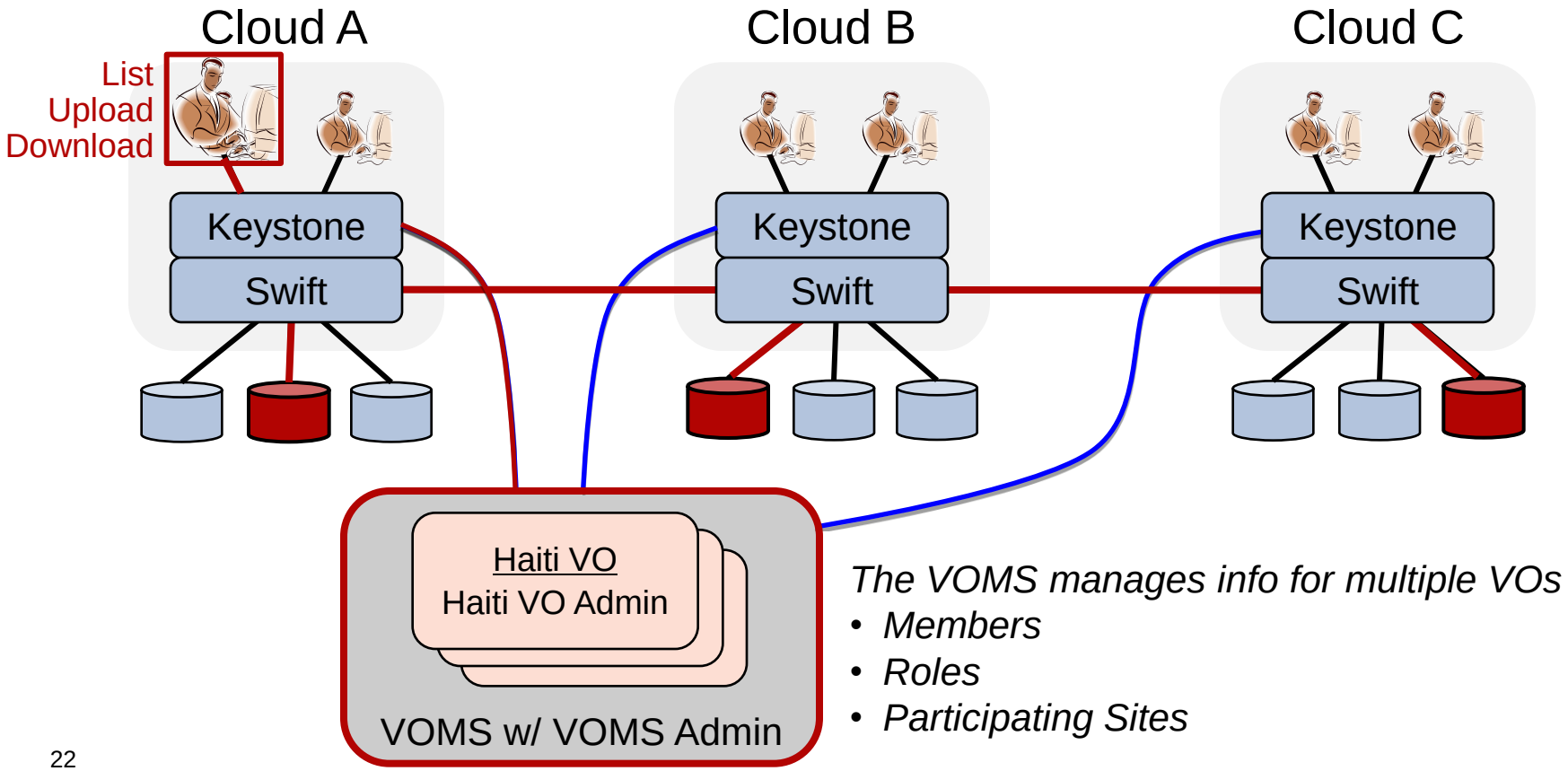


- 87 countries
- 140 organizational members
- Hundreds of active developers

Adding an External VO Mgmt System to OpenStack

NGA/NCOIC GeoInt Community Cloud (GCC) Prototype

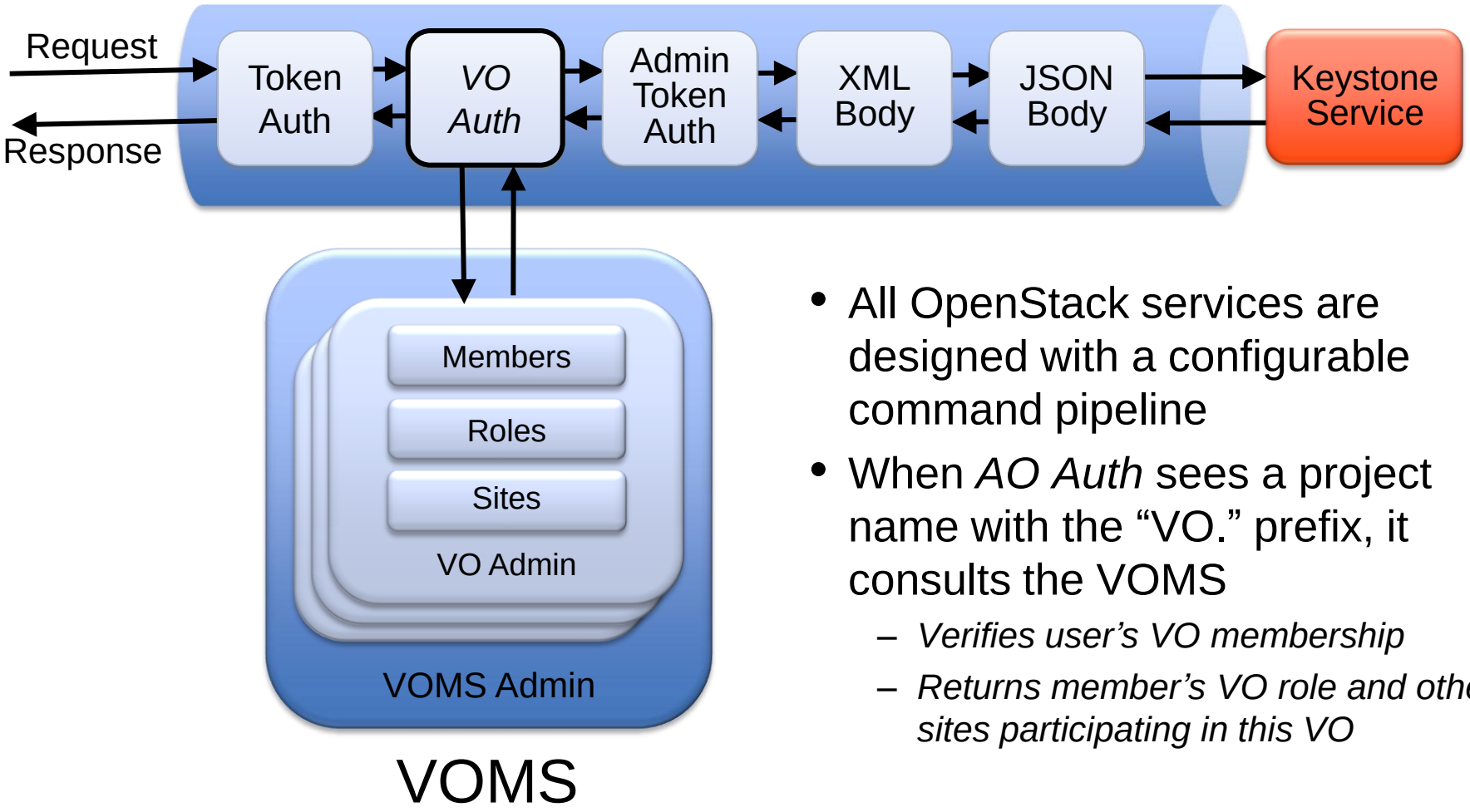
- The Keystone project concept extended to include *VO projects*
  - *Project names starting with "VO." are VO projects*
  - *VO projects could be denoted by internal Keystone attribute*
- Keystone and Swift clients and servers modified to demonstrate the management of container access using VO roles



The VOMS manages info for multiple VOs

- *Members*
- *Roles*
- *Participating Sites*

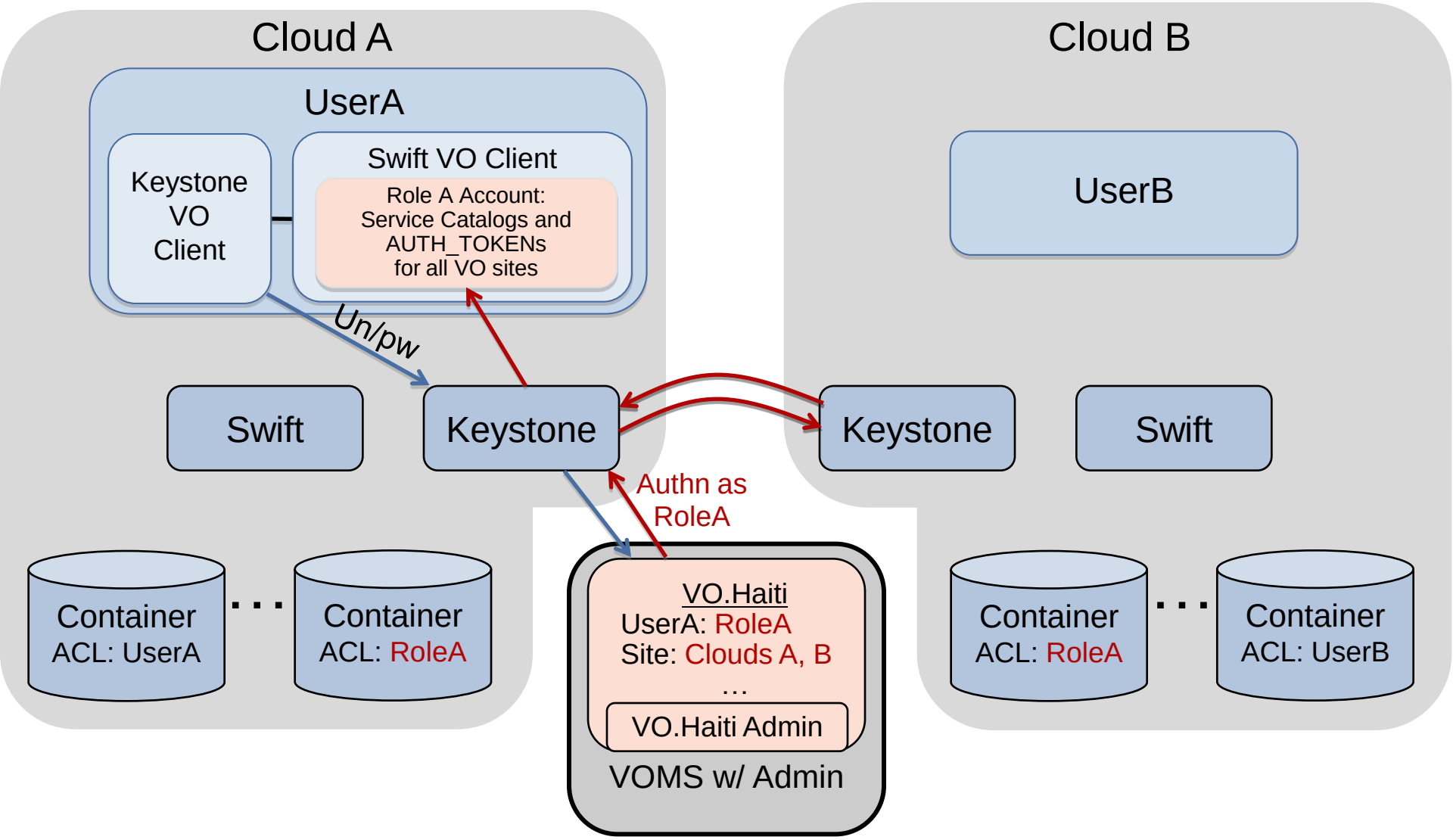
Adding a *VO Auth* Stage to the Keystone Command Pipeline



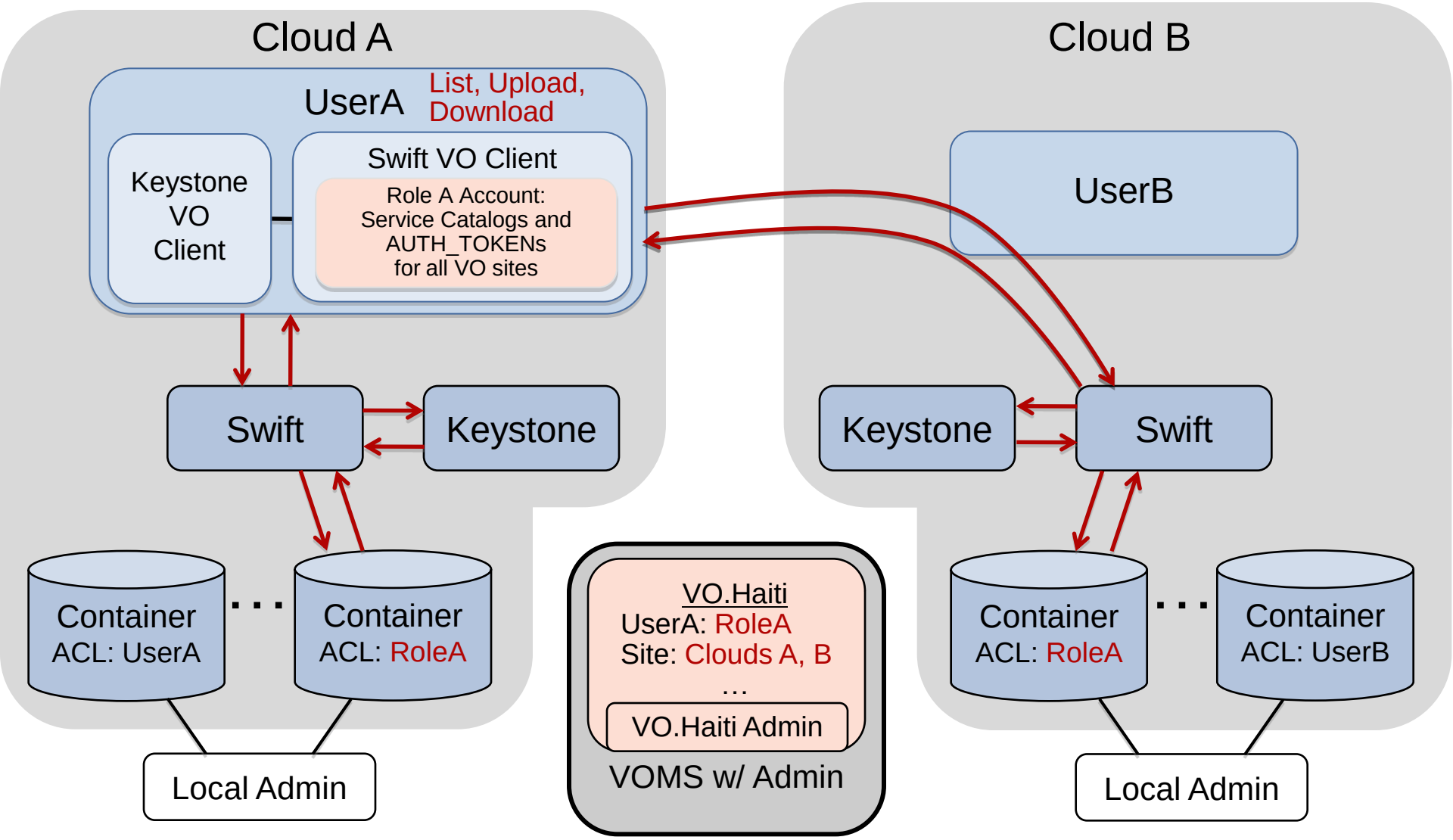
- All OpenStack services are designed with a configurable command pipeline
- When *AO Auth* sees a project name with the “VO.” prefix, it consults the VOMS
  - *Verifies user’s VO membership*
  - *Returns member’s VO role and other sites participating in this VO*

GCC Example:

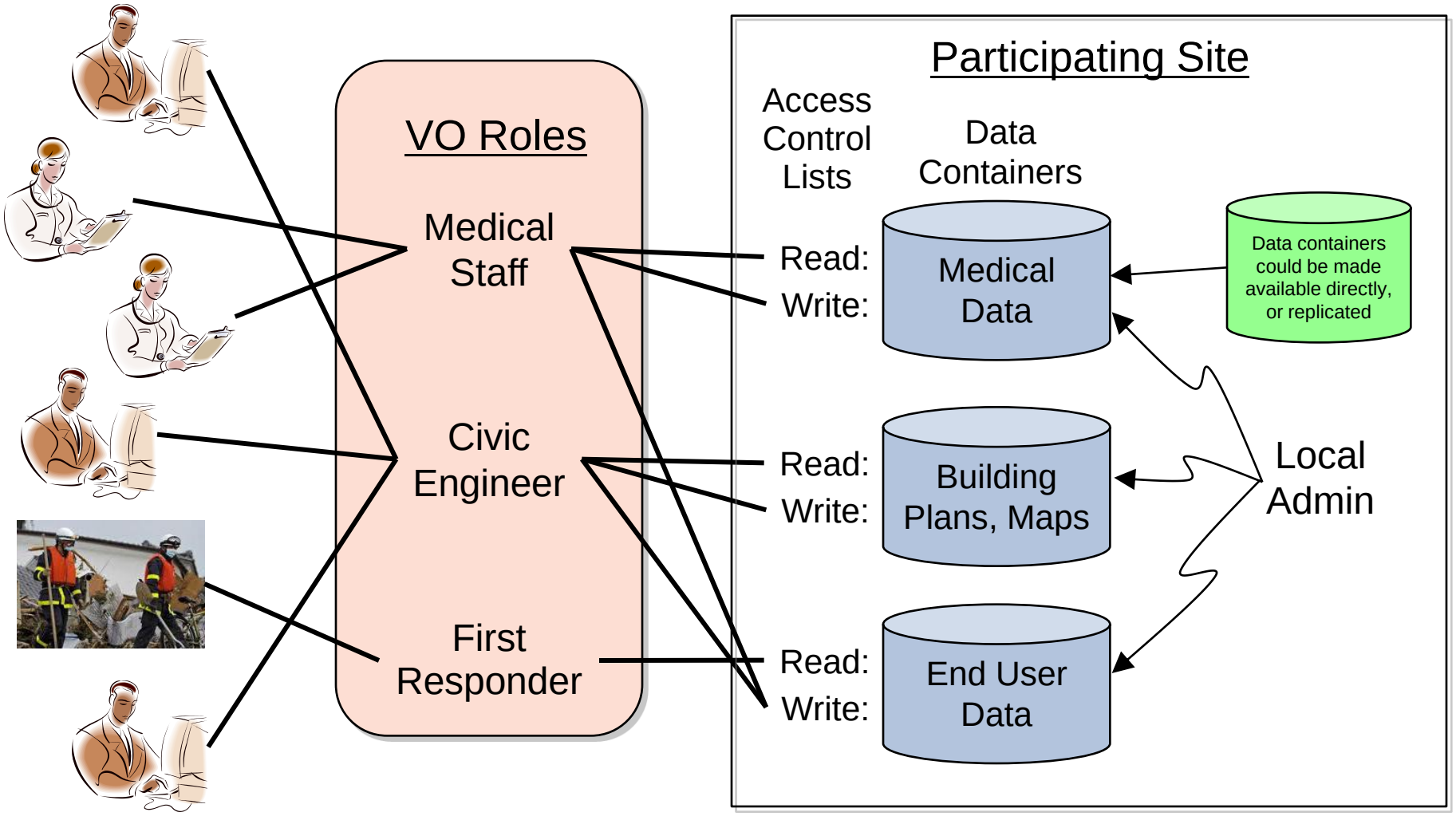
User Authentication using Keystone with external VOMS



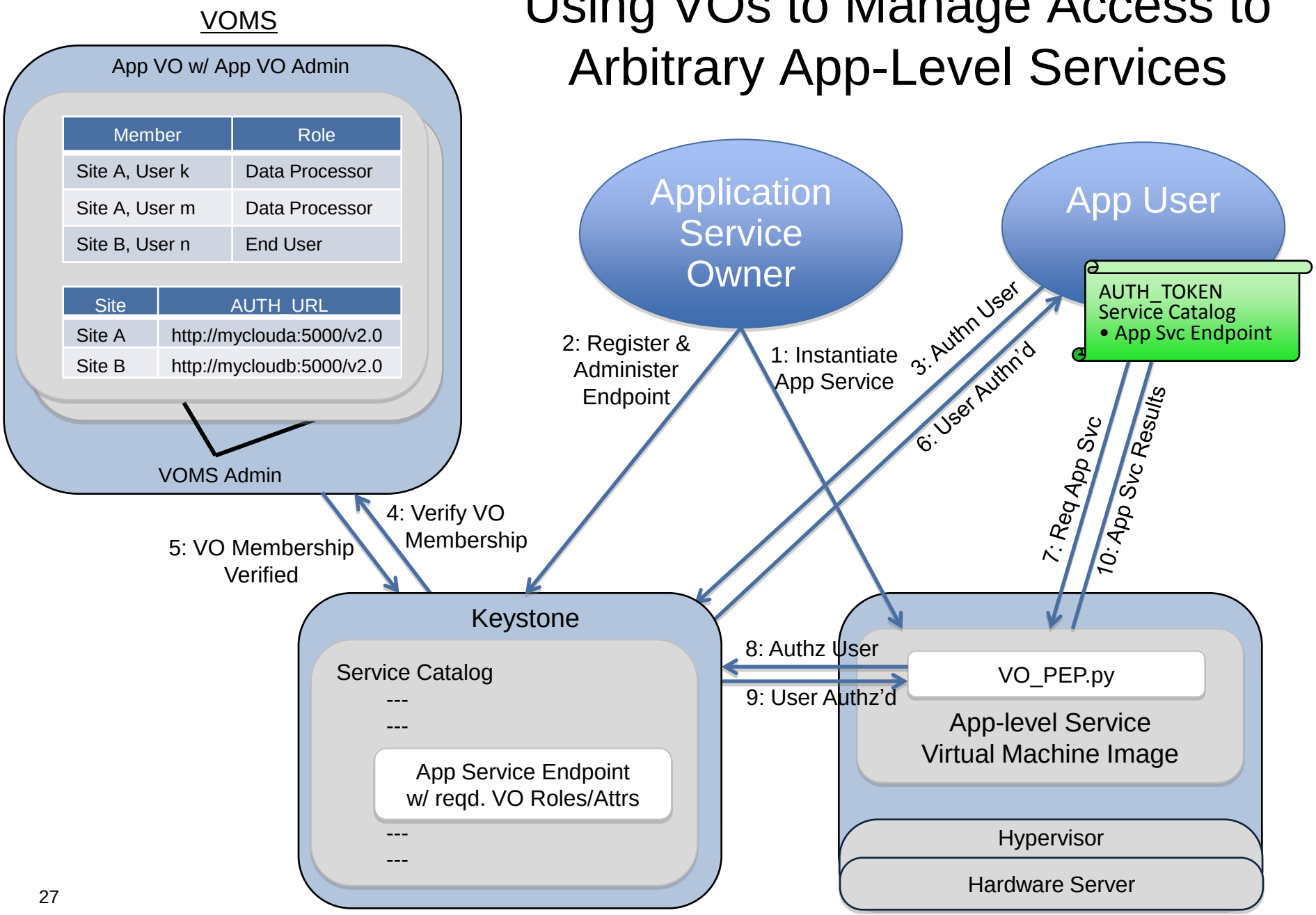
GCC Example: VO Container Access



GCC Example:
Using VO Roles to Manage Container Access



Using VOs to Manage Access to Arbitrary App-Level Services



Key Design Issues for VOs in OpenStack

- How to store the VO attributes
  - *As separate user entries with VO attributes (as per VOMS)*
  - *As a set of general mapping rules (as per attribute mappings)*
- How to Integrate VOs with multiple clouds
  - *Authenticating identity credentials from different Identity Providers*
  - *Aggregating IdP attributes with VO attributes*
- External VOMS DB vs. Peer-to-Peer Keystone VO DBs
  - *External VOMS easier to implement, quicker authz revocation process, but single point of failure*
  - *P2P doesn't rely on third party VOMS, not a single point of failure, but introduces consistency issue, and authz revocation takes longer*
- Whether to check that user is a VO member or not
  - *Carries an overhead (esp. if external VOMS DB) so when to avoid it?*
- Ensure that VOs can be used by arbitrary application-level services
  - *Database access, RSS feeds, any kind of standard services, e.g., geospatial tools: Web Map Service, Web Feature Service, Web Coverage Service, …*
- What should be at app-level vs. infra-level, i.e., what should be in Keystone
- Who manages the VO
  - *VO admin (distributed) or VOMS admin (centralized)*
- Infrastructure-level federation vs. application-level federation
- Scalability

Additional Important Issues

- Many VO management and administrative issues must be addressed
  - *Creating and terminating a VO*
  - *Joining and leaving a VO*
  - *Who is the VO Admin, and the VOMS Admin*
- Common understanding of VOs
  - *VO resources, roles, attributes and trusted VOMS*
  - *Data publishing and discovery*
  - *Semantic interoperability*
- Granularity of data/resource protection
  - *Resource protection carries an overhead*
  - *There will be a granularity at which resource protection overhead becomes excessive and intolerable*
- *Trust Federations*
  - *Human organizations where common agreements are made in advance of need*
  - *Existing Example: International Grid Trust Federation, www.igtf.net*
  - *Possible Analogy: International Disaster Response Trust Federation*

Summary and Discussion

- Two major topics explored:
  - *Cloud SLAs*
  - *Cloud federation – Virtual Organizations*
- Capacity and performance management will be central to how cloud-based ground systems are deployed and managed
  - *Autonomic control will be crucial since it will simply not be possible -- nor desirable – to have humans in the loop*
- Cloud federation will be central to how ground systems are operated across a distributed infrastructure of inter-clouds
  - *Identity attributes and resource attributes ("data tagging") will be crucial to enforcing security and governance policies*
- *Lots of work to be done*

AEROSPACE
Assuring Space Mission Success

Thank you!

Any Questions?

All trademarks, service marks, and trade names are the property of their respective owners.