

GENI: Global Environment for Network Innovations

Architecture

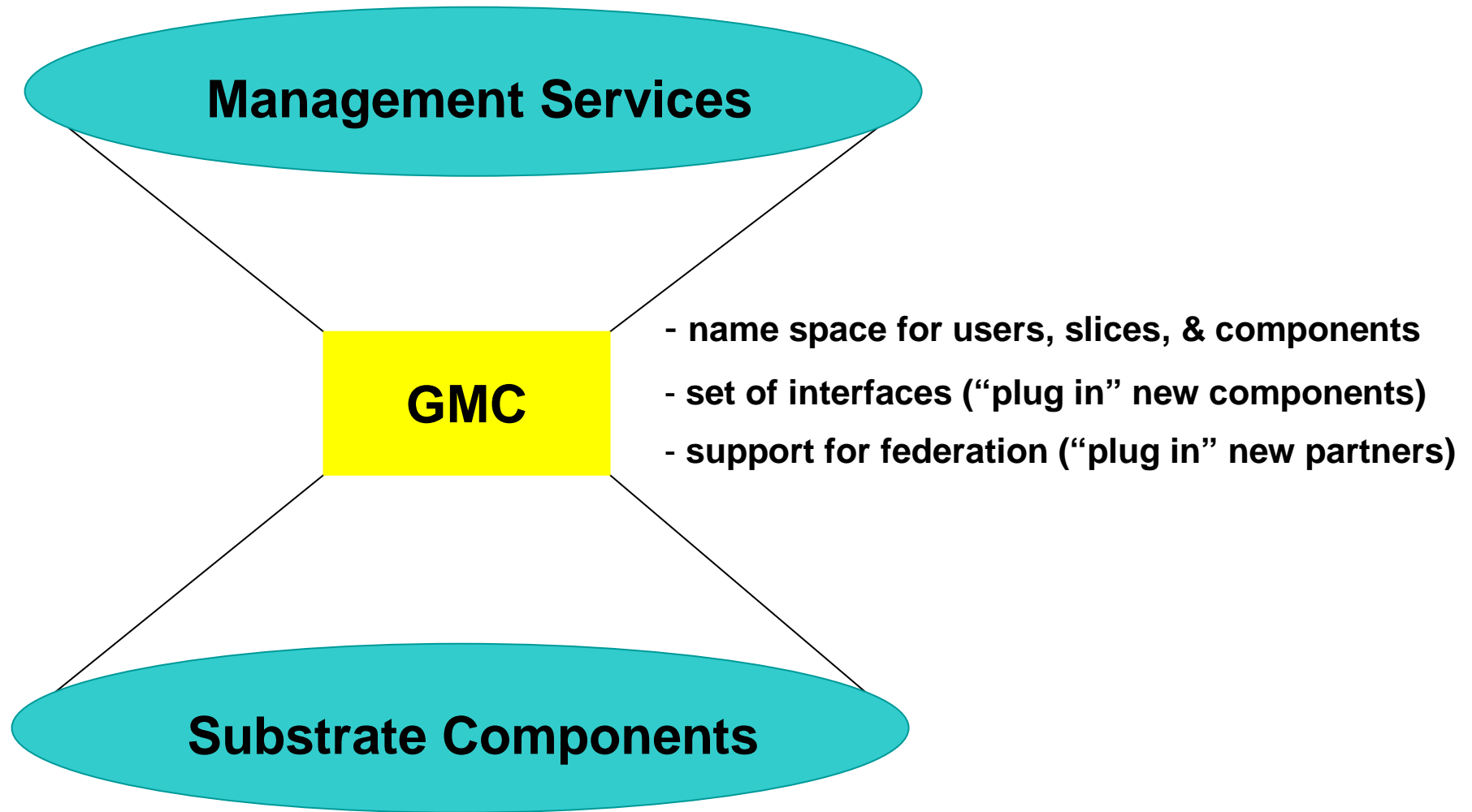
Larry Peterson

November 13, 2006

Outline

- Abstractions
- Examples
- Working Group Responsibilities

Overview



Abstractions

- Components
 - multiplexed among a set of slivers
 - specify with an **rspec**
 - resources / privileged operations / instrumentation sensors
- Slices
 - name them (approved to use GENI)
 - embed them (instantiate a set of slivers)
 - configure them (run an experiment)
- Aggregates
 - researcher portal
 - coordinate resource allocation
 - manage set of components

Names

- Signed statements of identity
 - signed by a hierarchy of *naming authorities*
 - implemented as X.509 certificates
- Includes
 - object type (e.g., component)
 - unique identifier (UUID)
 - address for object manager (URI)
 - human-readable string
 - e.g., top-level_authority.sub_authority.sub_authority.name
- Examples
 - slice: geni.us.princeton.codeen
 - component: geni.us.backbone.nyc

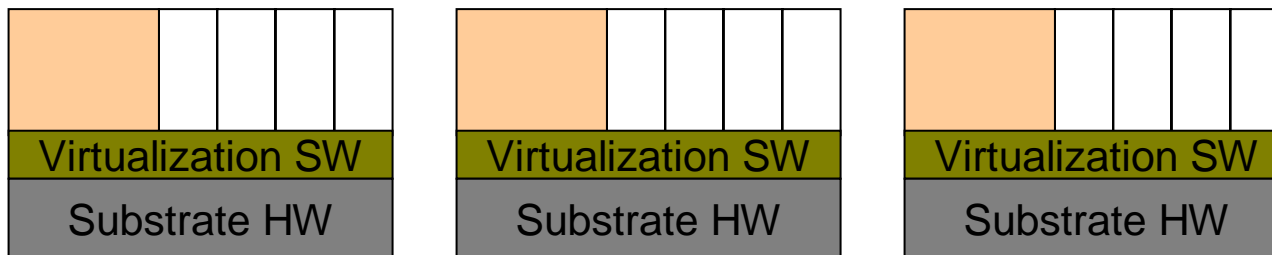
Substrate Hardware

Substrate HW

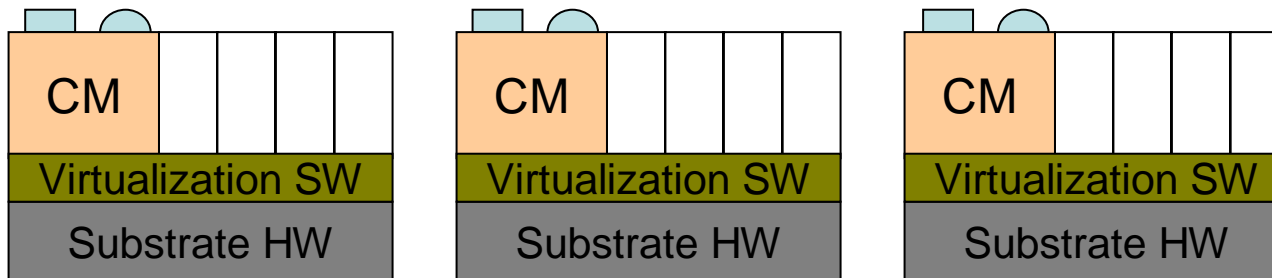
Substrate HW

Substrate HW

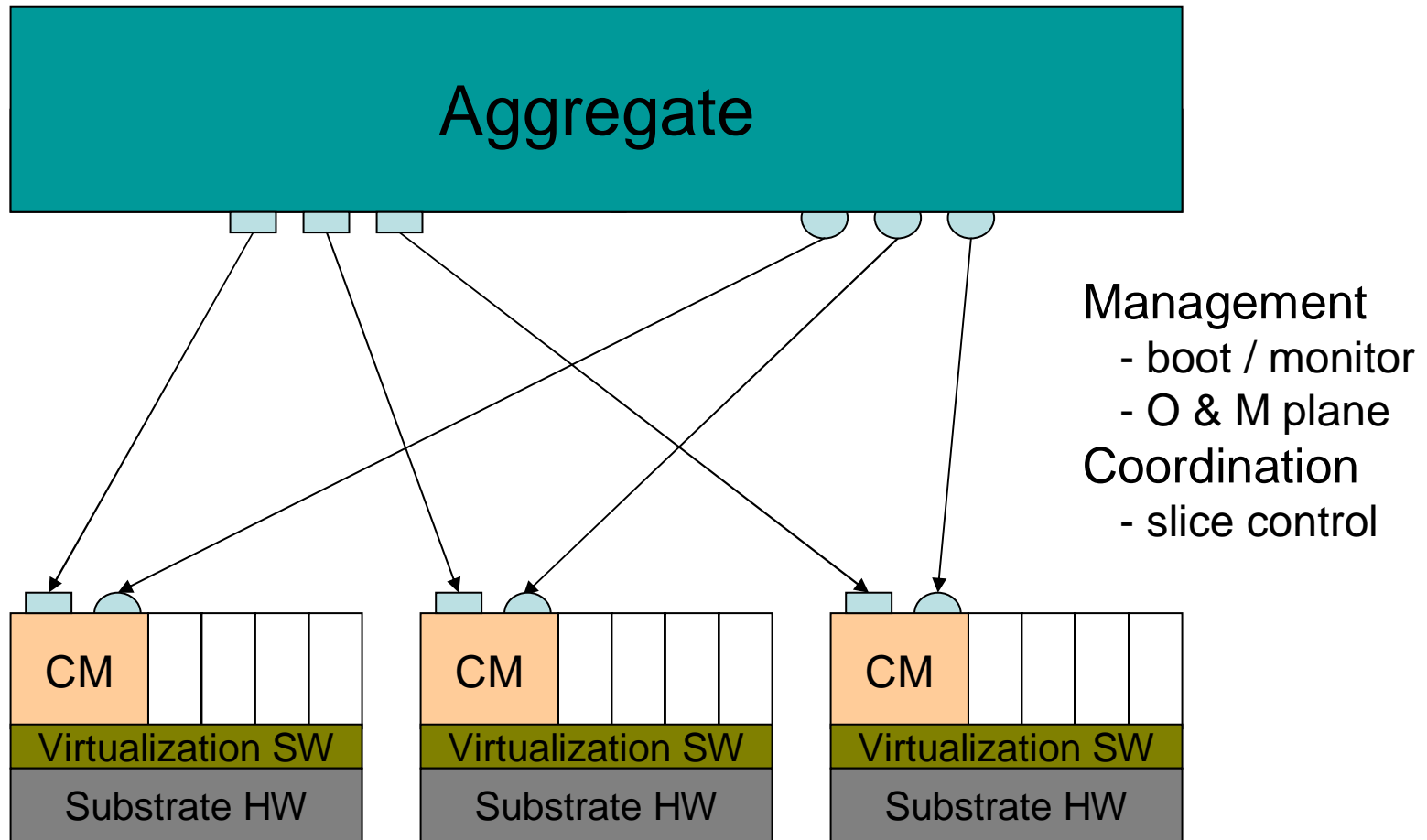
Virtualization Software



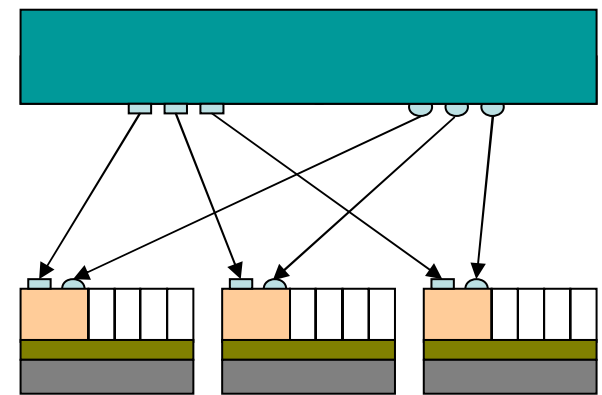
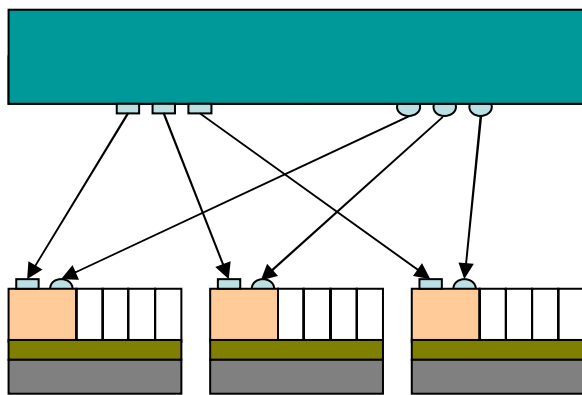
Components



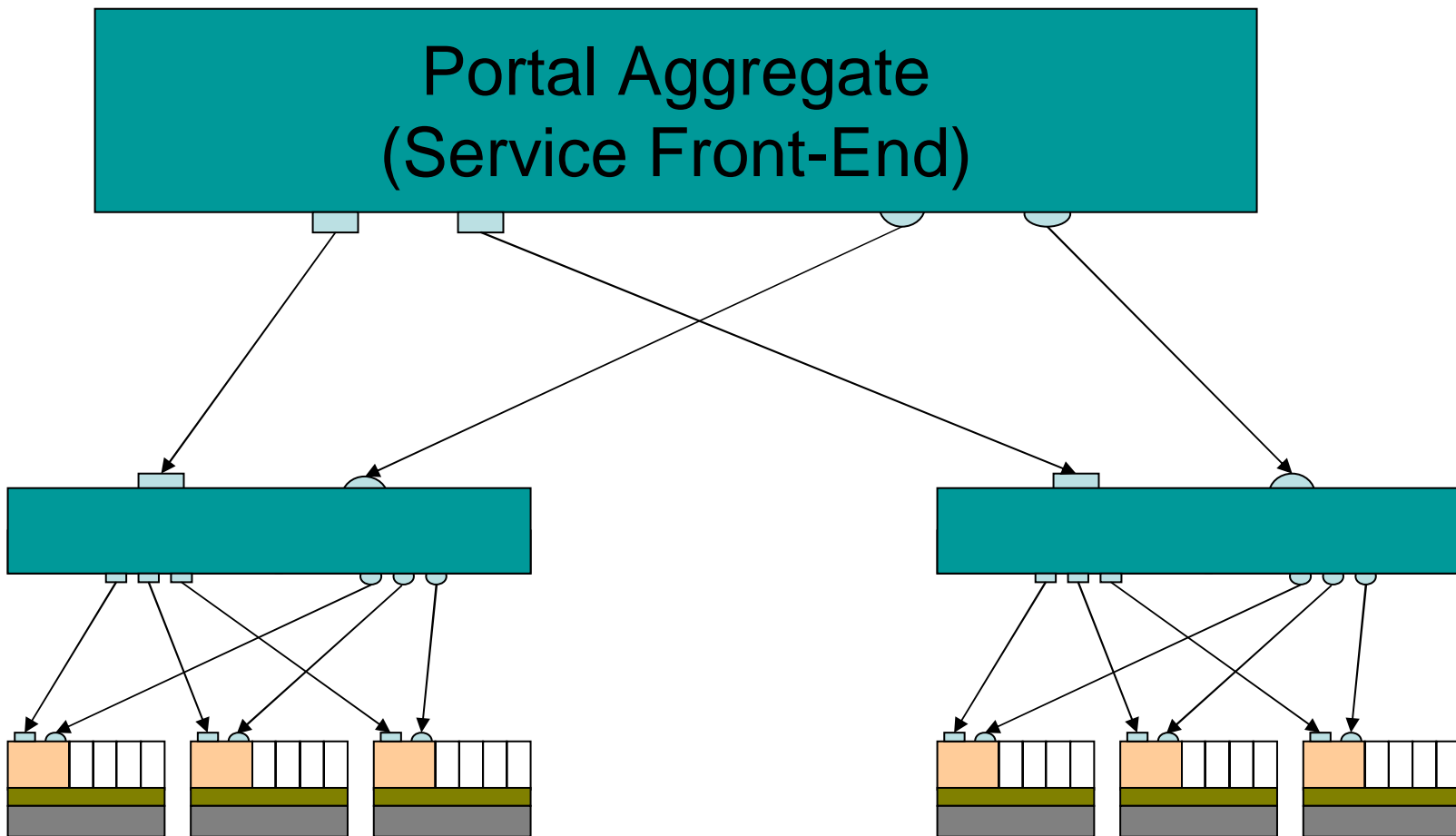
Aggregates



Federation



User Portal



Virtualization

- Multiple levels possible
 - different level required by different experiments
 - different level depending on the technology
 - in the limit, we'll need to slice “physical component array”
- Example “base cases”
 - virtual server (socket interface / overlay tunnels)
 - virtual router (virtual line card / static circuits)
 - virtual switch (virtual control interface / dynamic circuits)
 - virtual AP (virtual MAC / fixed spectrum allocation)
- Specialization
 - the ability to install software in your own virtual-*

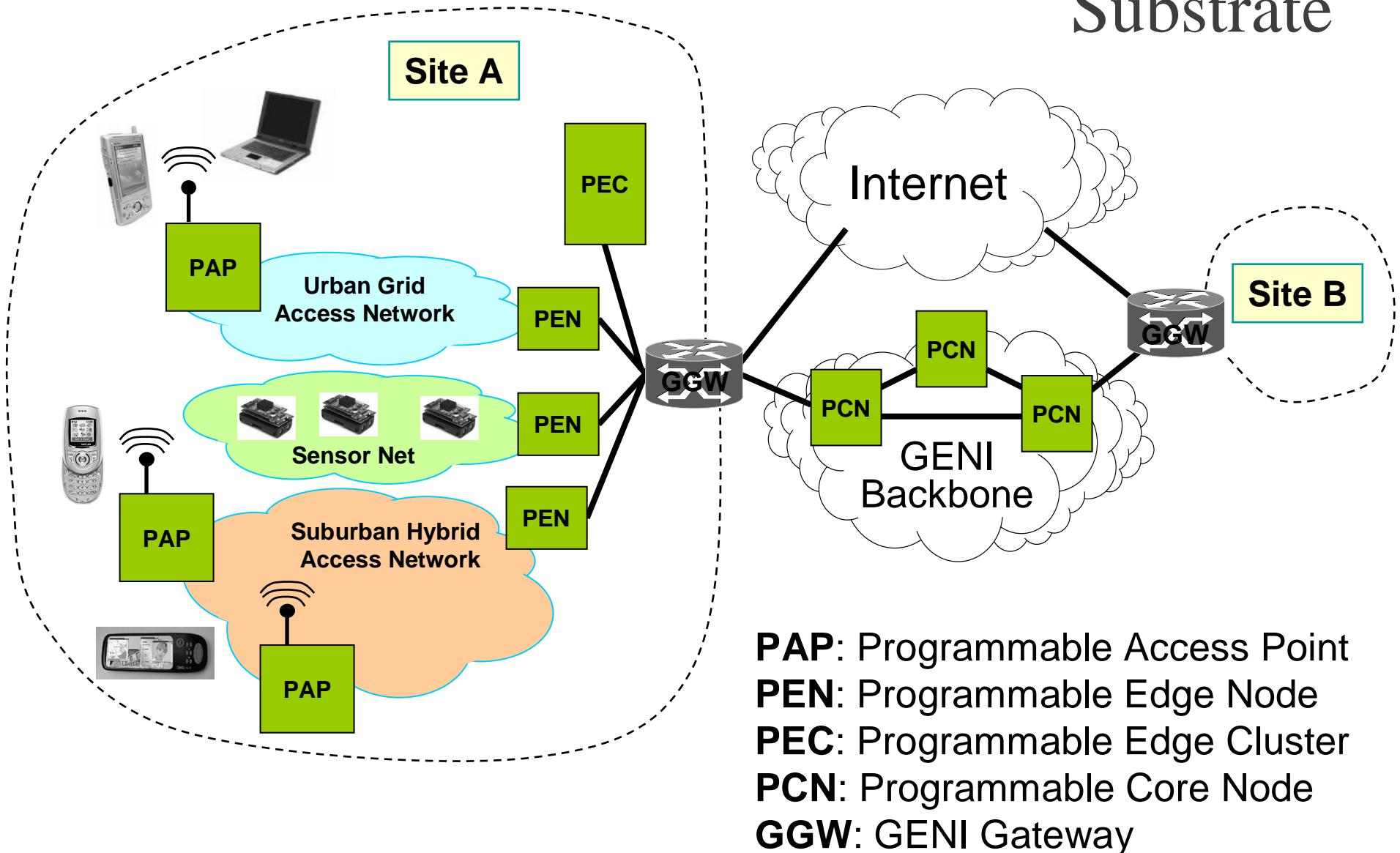
Canonical Component

- 1) create and destroy slivers, bind a set of resources to a sliver
- 2) isolate slivers from each other, such that
 - a) the resources consumed by one sliver do not unduly affect the performance of another,
 - b) one sliver cannot, without permission, eavesdrop on network traffic to or from another,
 - c) one sliver cannot access objects (e.g., files, ports, processes) belonging to another, and
 - d) users are allowed to install software packages in their sliver without consideration for the packages installed in other slivers running on the same component;
- 3) allow users to securely log into a sliver that has been created on their behalf;
- 4) deliver signals to slivers, including a “reboot” signal that is delivered whenever the sliver starts up;
- 5) grant privileged operations to select slivers, including the ability of one sliver to access private state associated with another sliver (thereby supporting sliver interposition);

Compliant Site (GGW)

- 6) disconnect the component(s) from the network and bring it into a safe state;
- 7) rate-limit the network traffic generated by the component(s);
- 8) for site that supports connections to the existing Internet, limit (filter) how the component(s) interacts with (exchanges packets with) the Internet;
- 9) for a site that supports connections to the existing Internet, support a mechanism to audit all packet flows transmitted by the component(s) to the Internet, and determines what sliver (slice) is responsible for a given packet;

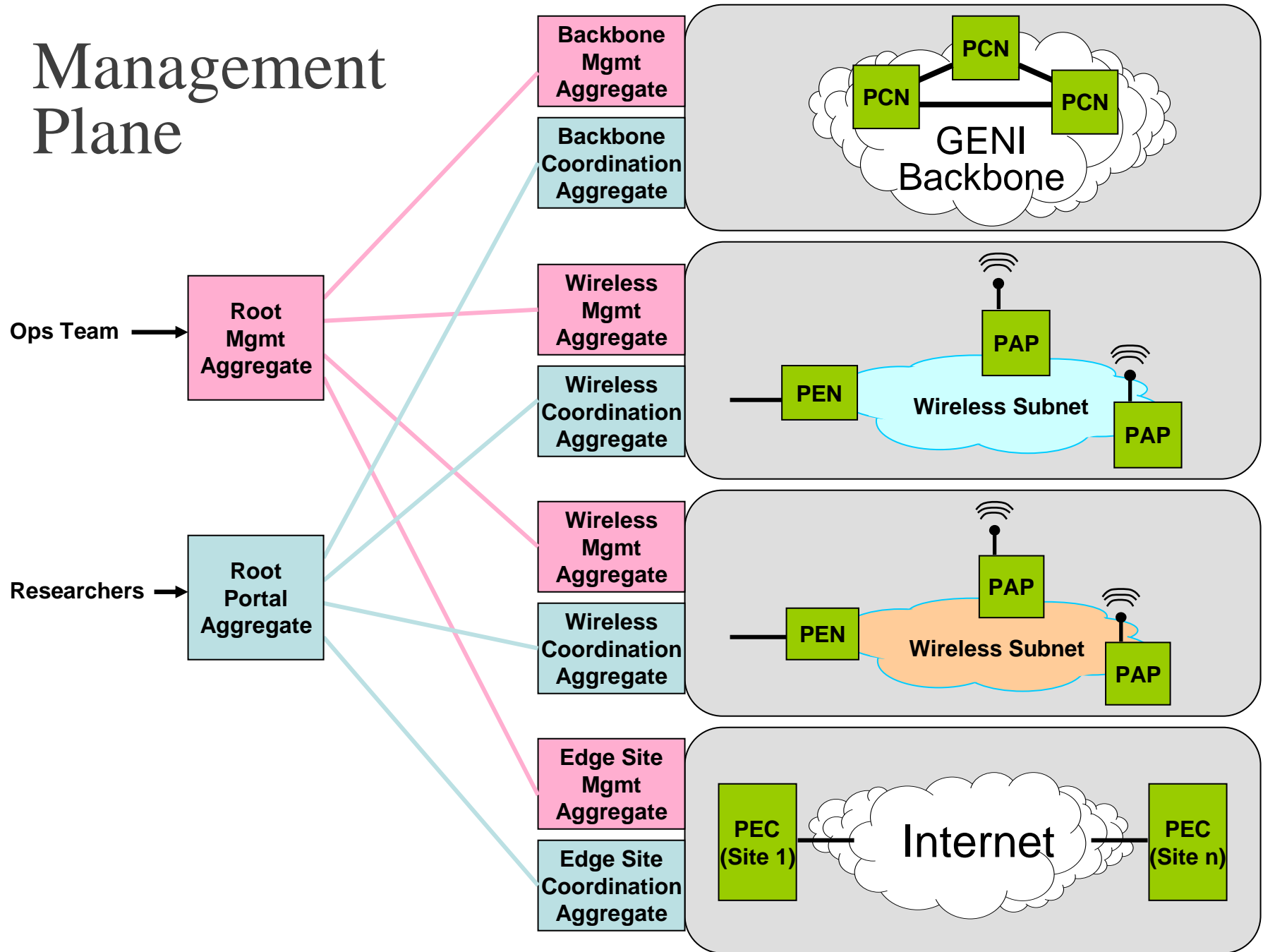
Example Substrate



Reference Design Responsibilities

- Architecture WG
 - GMC / Naming Service
 - Canonical Component
 - corresponds to Programmable Edge Cluster (PEC)
 - Site Configuration
 - including tail circuits
- Backbone WG
 - Programmable Core Node (PCN)
 - Backbone Network
- Wireless WG (x subnet types)
 - Programmable Edge Nodes (PEN)
 - Programmable Access Points (PAP)
 - Access Network
- Services WG
 - Resource Allocation
 - Slice Embedding
 - ...

Management Plane



Aggregates, Services, & Planes

- Management Aggregate (Backbone/Wireless WG)
 - Operations & Maintenance Control Plane
 - securely boot and update
 - diagnose & debug failures
- Coordination Aggregate (Backbone/Wireless WG)
 - Slice Control Plane
 - coordinate slice embedding across a subnet
- Portal Aggregate (Services WG)
 - slice embedding service
 - resource discovery
 - resource allocation (implement GSC policy)
 - end-to-end topology “stitching”
 - experiment management service
 - configuration management
 - development tools
 - diagnostics & monitoring
 - data logging

Web Services

- Components register their availability
- Root portal aggregate is a web services registry
- Sub-services build on this and register their availability
 - etc., recursively
- High-level (user-friendly) services export web-based user interfaces