GENI Architecture: Transition

Facility Architecture Working Group

July 18, 2007

Outline

- Requirements
  - Top Level
  - Derived
- Facility Design
  - Hardware Configuration
  - Software Configuration
- Minimal Core
- Construction Plan

Top-Level Requirements

1. Generality
   A. Minimal Constraints
      - allow new packet formats, new functionality, new paradigms,…
      - allow freedom to experiment across the range of architectural issues (e.g., security, management,...)
   B. Representative Technology
      - include a diverse and representative collection of networking technologies, since any future Internet must work across all of them, and the challenges/opportunities they bring
2. Slicability
      - support many experiments in parallel
      - isolate experiments from each other, yet allow experiments to compose their experiments to build more complex systems

Top-Level Req (cont)

3. Fidelity
   A. Device Level
      - expose the right level of abstraction, giving the experimenter the freedom to reinvent above that level, while not forcing him or her to start from scratch (i.e., reinvent everything below that level)
      - these abstractions must faithfully emulate their real world equivalent (e.g., expose queues, not mask failures)
   B. Network Level
      - arrange the nodes into a representative topology and/or distribute the nodes across a physical space in a realistic way
      - scale to a representative size
      - expose the right network-wide abstractions (e.g., circuits, lightpaths)
   C. GENI-Wide
      - end-to-end topology and relative performance
      - economic factors (e.g., relative costs, peering)

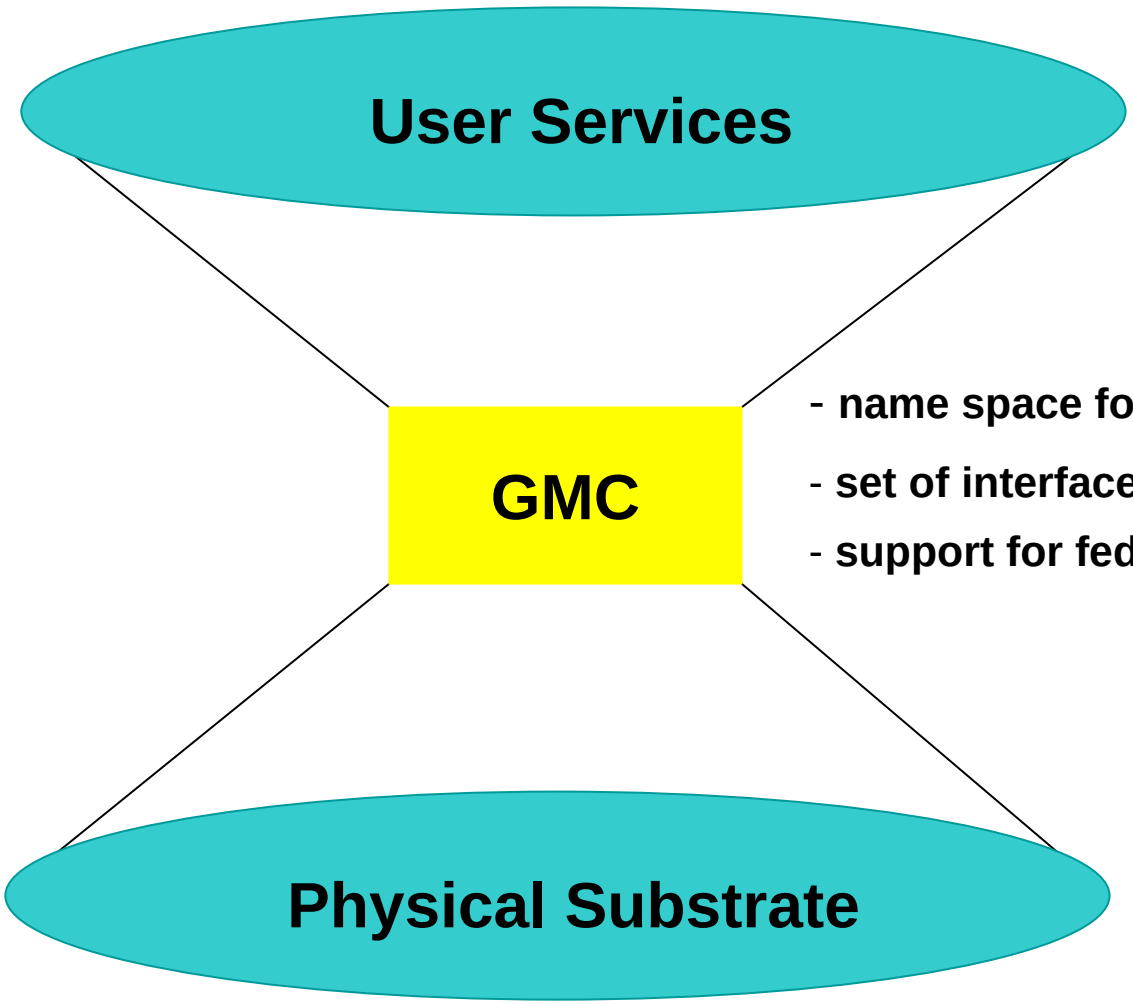
Top-Level Req (cont)

4. Real Users
   - allow real users to access real content using real applications
5. Experiment Support
   A. Ease-of-Use
      - provide tools and services that make the barrier-to-entry for using GENI as low as possible (e.g., a single PI and one grad student ought to be able to use GENI)
   B. Observability
      - make it possible to observe and measure relevant activity

Top-Level Req (cont)

6. Sustainability
   A. Extensible
      - accommodate network technologies contributed by various partners
      - accommodate new technologies that are likely to emerge in next several years
      - support technology roll-over without disruption
   B. Operational Costs
      - the community should be able to continue to use and support the facility long after construction is complete

Facility Architecture



- name space for users, slices, & components
- set of interfaces (“plug in” new components)
- support for federation (“plug in” new partners)

Derived Requirements

1. Generality
   - A. Minimal Constraints →
     - i. All devices are programmable with open interfaces
     - ii. All devices expose multiple levels of abstraction, with the lowest level coming as close to the raw hardware as possible:
       - a. sockets
       - b. virtual links
         - point-to-point
         - multipoint
       - c. virtual radio
       - d. virtual wire
         - framed TDM electrical
         - unframed TDM electrical
         - unframed raw wavelength

Derived Req (cont)

1. Generality
   - B. Representative Technology →
     - i. Architected to accommodate a wide range of technologies
     - ii. Initially selected technologies:
       - a. national fiber backbone
       - b. connected edge clusters
       - c. mobile client devices & sensors
       - d. 802.11 access network
       - e. WiMAX access network
       - f. cognitive radio access net

Derived Req (cont)

2. Slicability ⟶
   i. Virtualize resources
   ii. Partition resources (time & space)
   iii. Isolate and contain slices
   iv. Support slice composition
   v. Minimum capacity
      a. 1000-10000 total projects
      b. 100-1000 active experiments
      c. 10-100 continuously running services
      d. 1-10 high-performance/low-jitter nets

Derived Req (cont)

3. Fidelity
   - A. Node Level ⟶ i. For each selected technology (1.B.ii)
     - a. support interfaces as appropriate (1.A.ii)
     - b. sliver behavior (e.g., jitter)
     - c. node capacity
     - d. transmission capacity
   - B. Network Level ⟶ i. For each selected technology (1.B.ii)
     - a. points-of-presence
     - b. distribution/topology
   - C. GENI-Wide ⟶ i. Rich topology
     - ii. Federation of autonomous domains

Derived Req (cont)

4. Real Users ⟶
   i. Wide coverage
   ii. Opt-in mechanisms
      i. Generic proxies
      ii. GENI-aware applications
   iii. Legacy Internet connectivity
      a. number of edge connection points
      b. number of peers
      c. capacity per peer

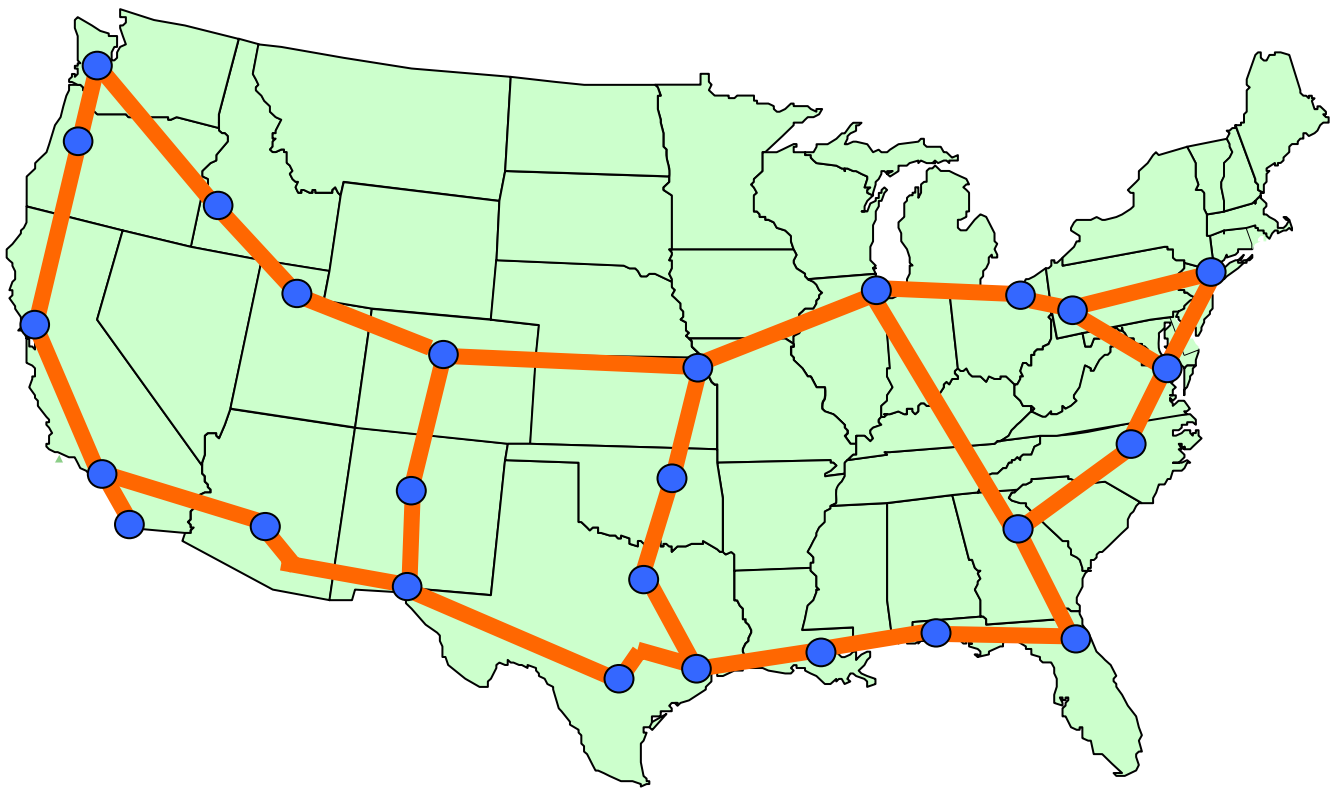
Derived Req (cont)

5. Experiment Support
   - A. Ease-of-Use ⟶
     - i. Rich set of user services
       - a. slice embedding
       - b. experiment management
       - c. legacy Internet
       - d. building block services
   - B. Observability ⟶
     - i. For each selected technology (1.B.ii)
       - a. instrumentation sensors
     - ii. Data collection/archiving/analysis tools

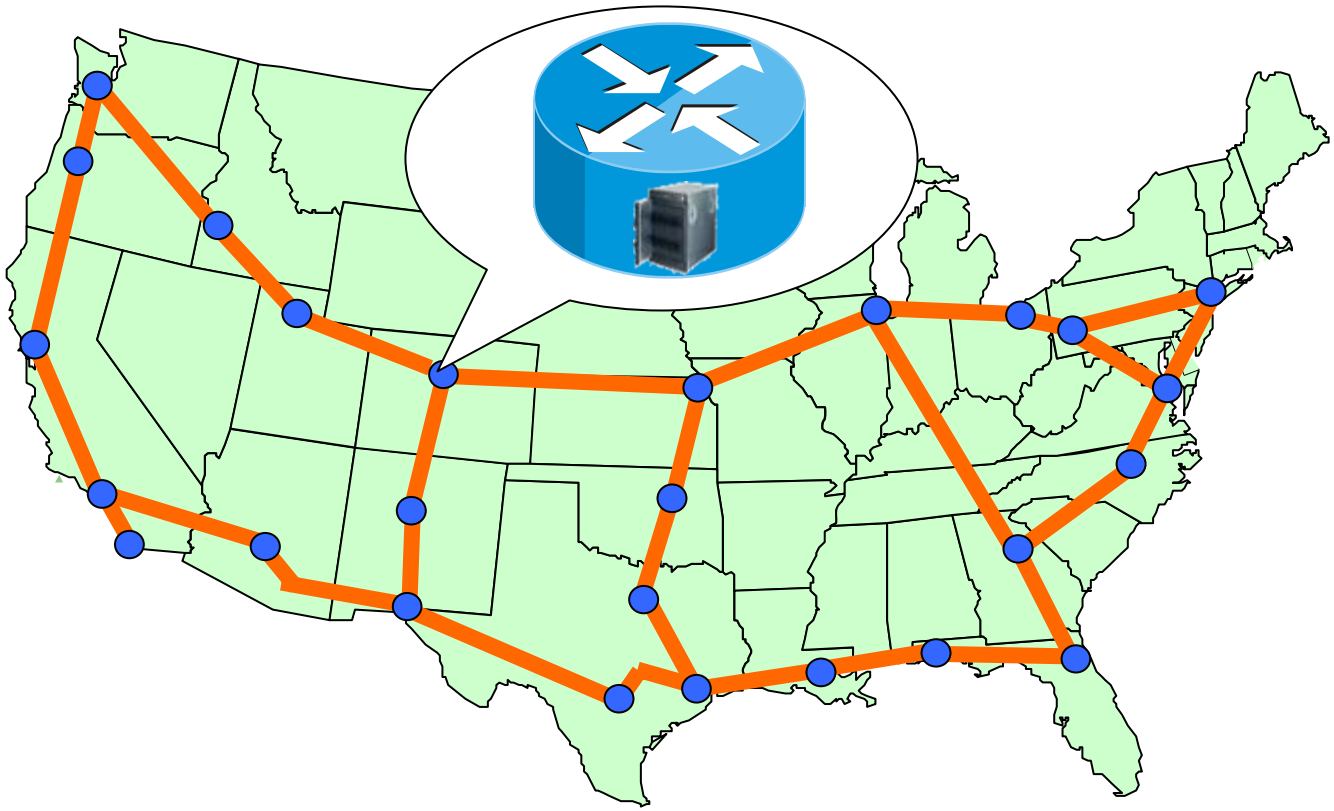
Derived Req (cont)

6. Sustainability
   - A. Extensible ⟶
     - i. Same as 1.B.i (representative technology)
     - ii. Support federated ownership
   - B. Operational Costs ⟶
     - i. Renewal costs contribute to 1.B.ii.
     - ii. Adopt sustainable software engineering practices
     - iii. Architect for security
     - iv. Develop monitoring & diagnosis tools

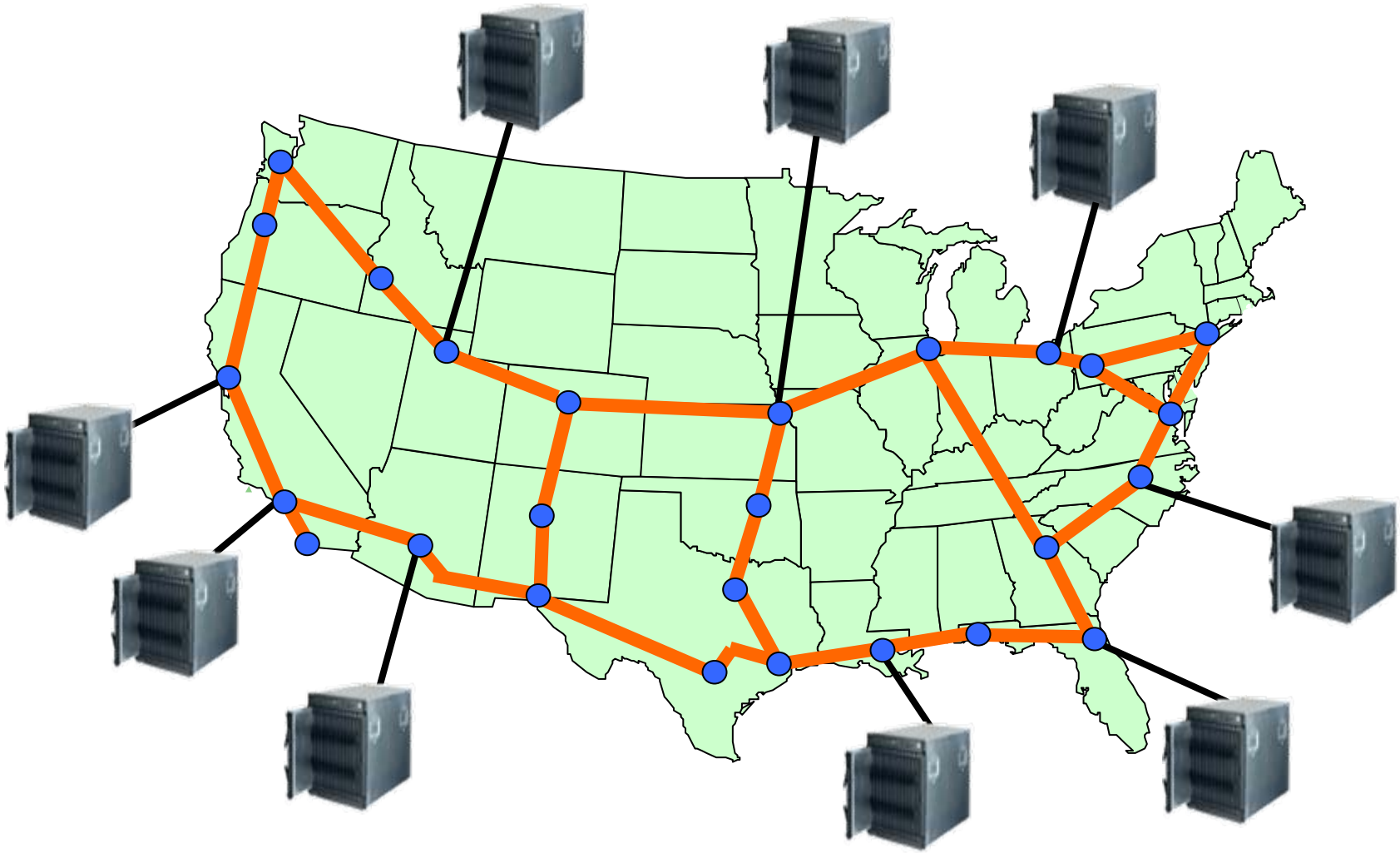
National Fiber Facility



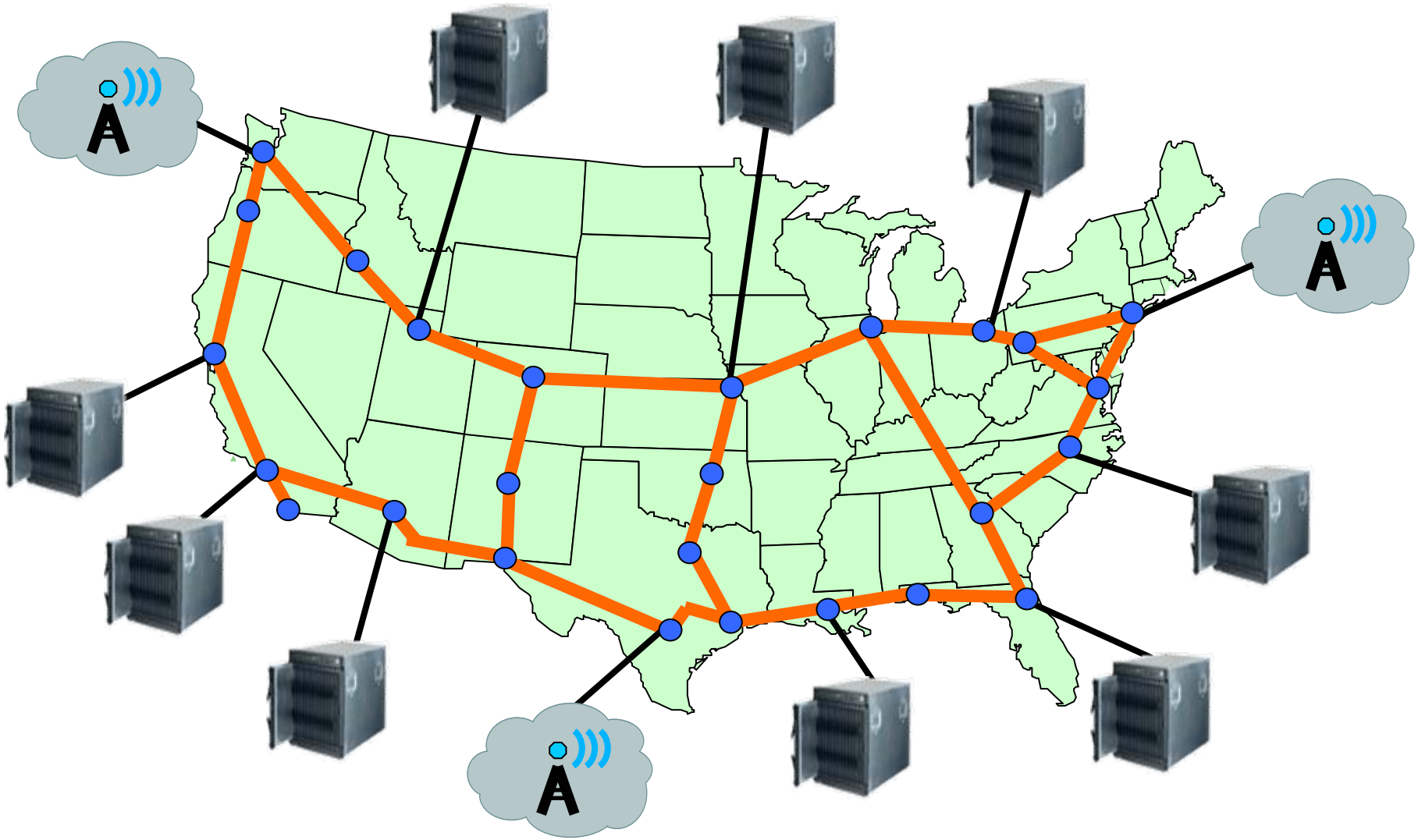
+ Programmable Routers



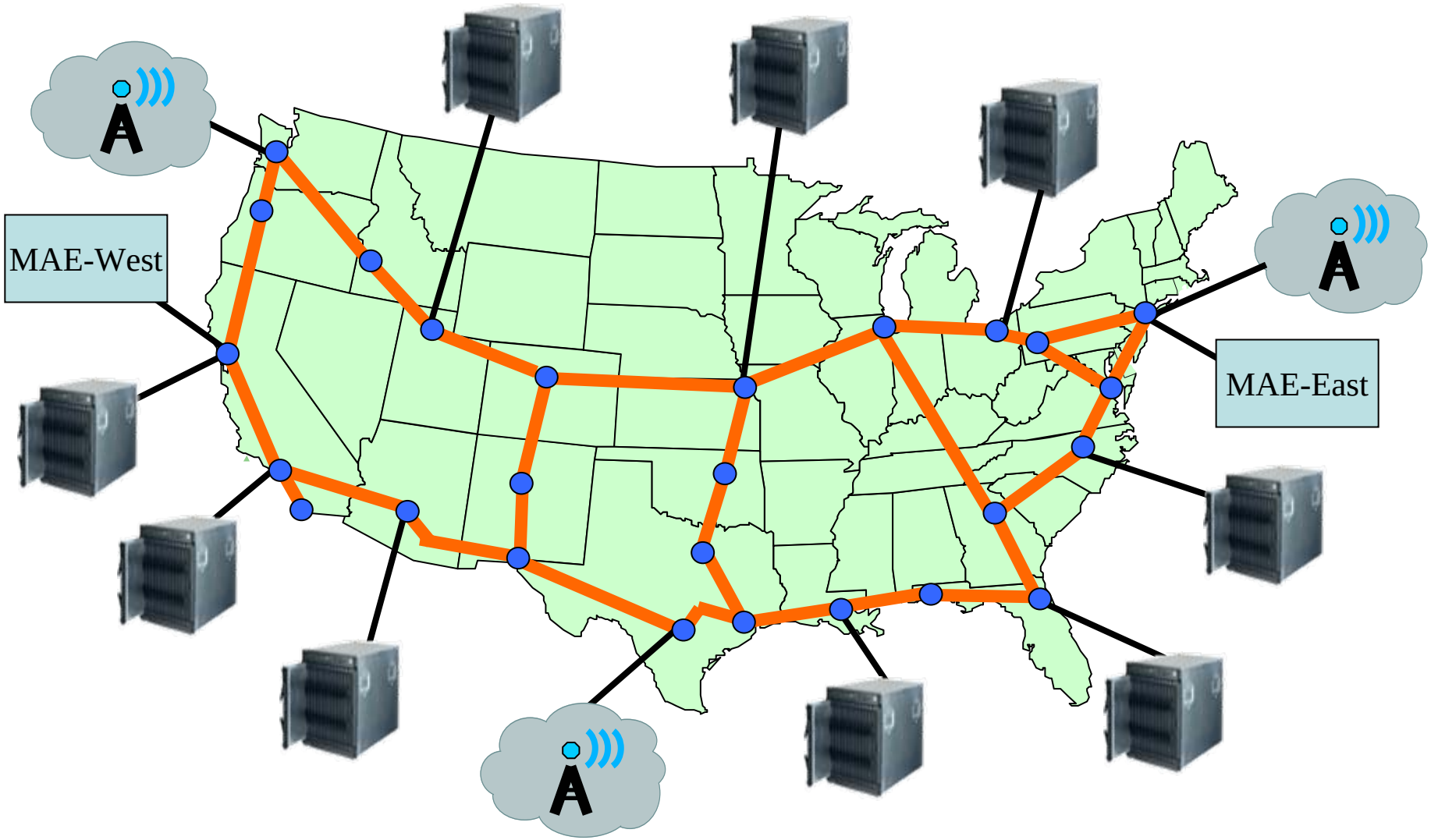
+ Clusters at Edge Sites

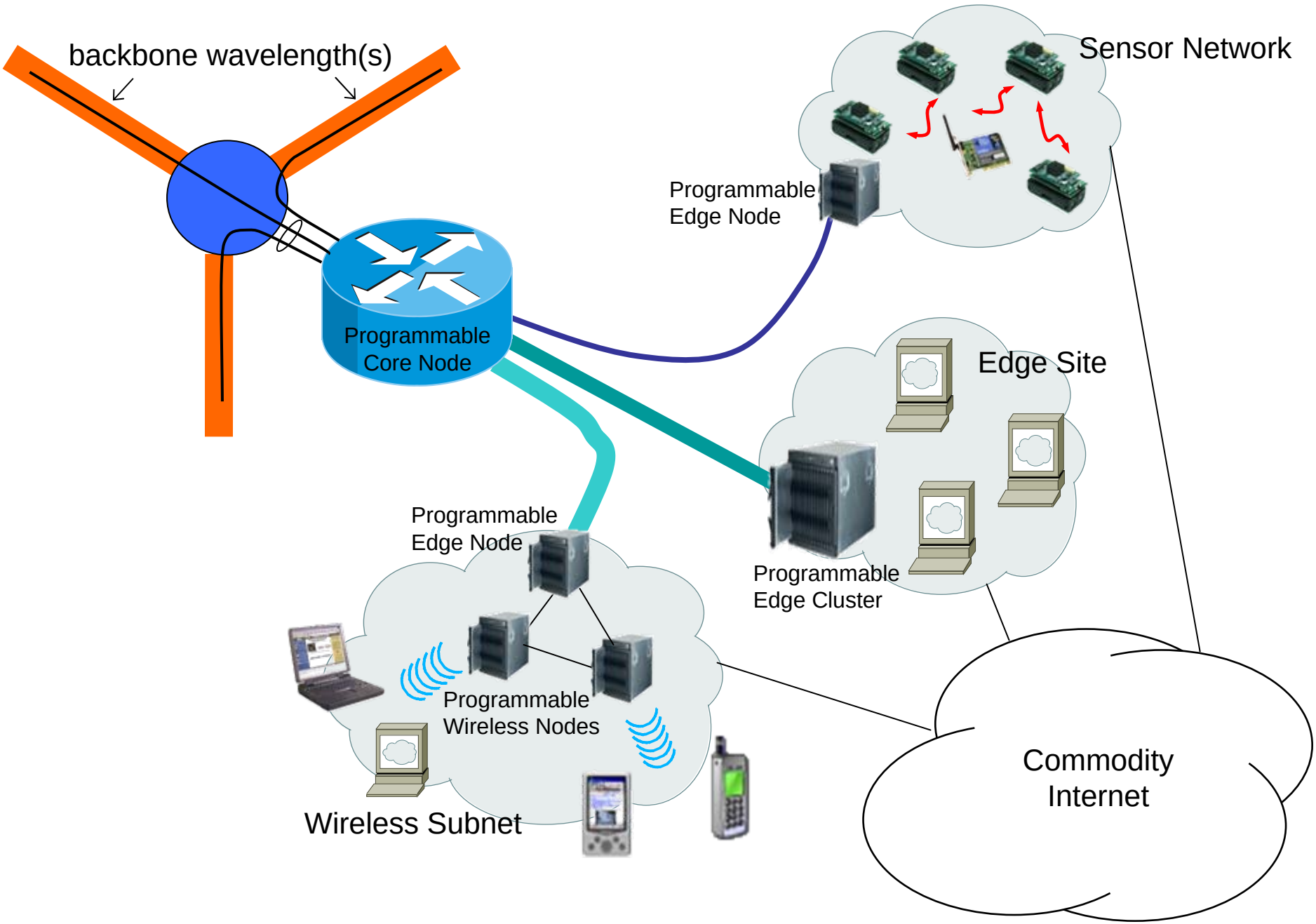


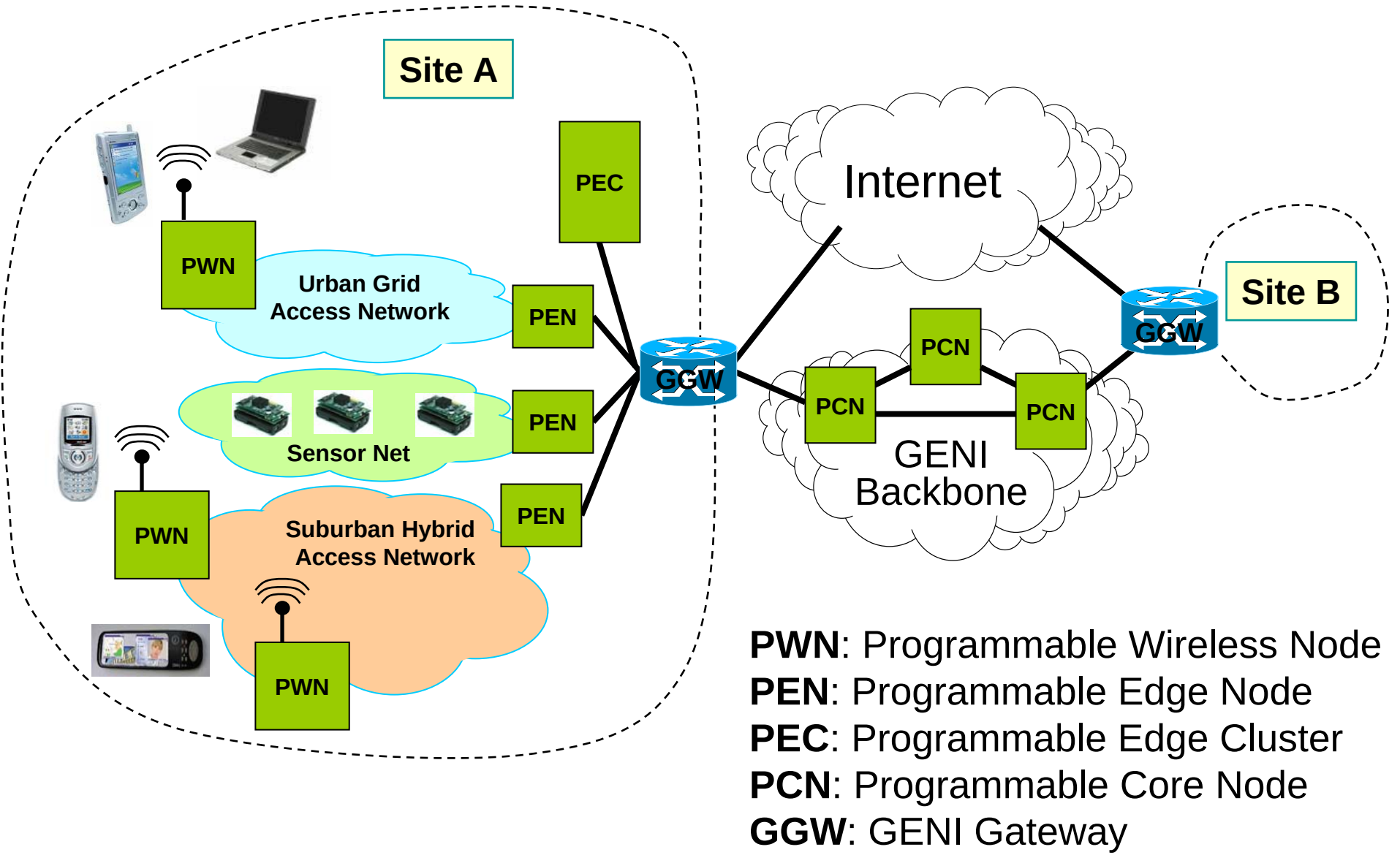
+ Wireless Subnets



+ ISP Peers

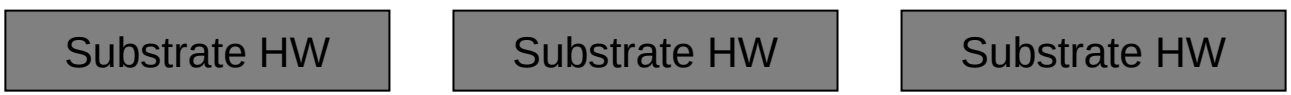



backbone wavelength(s)
Sensor Network
Programmable
Edge Node
Programmable
Core Node
Edge Site
Programmable
Edge Node
Programmable
Edge Cluster
Programmable
Wireless Nodes
Wireless Subnet
Commodity
Internet

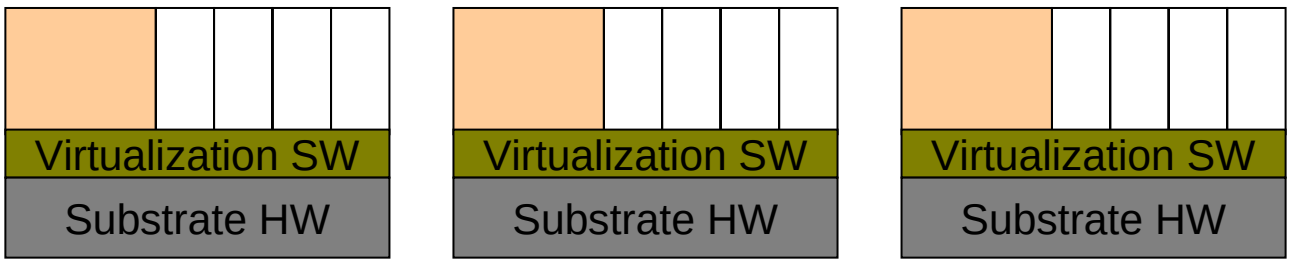


PWN: Programmable Wireless Node
PEN: Programmable Edge Node
PEC: Programmable Edge Cluster
PCN: Programmable Core Node
GGW: GENI Gateway

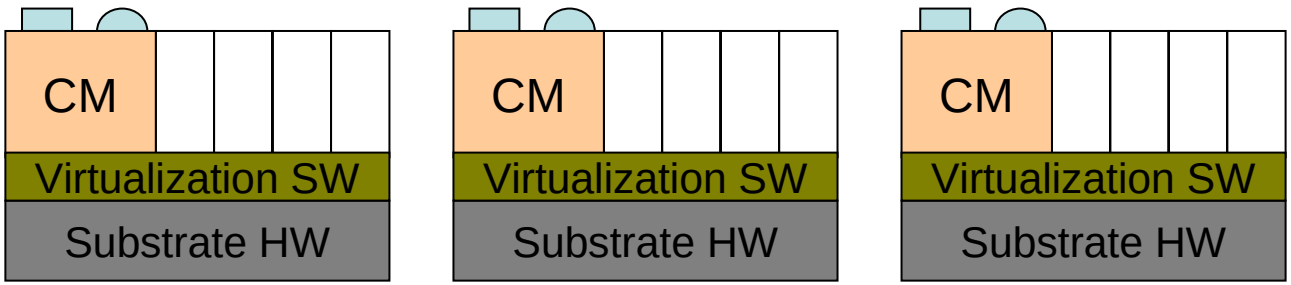
Substrate Hardware



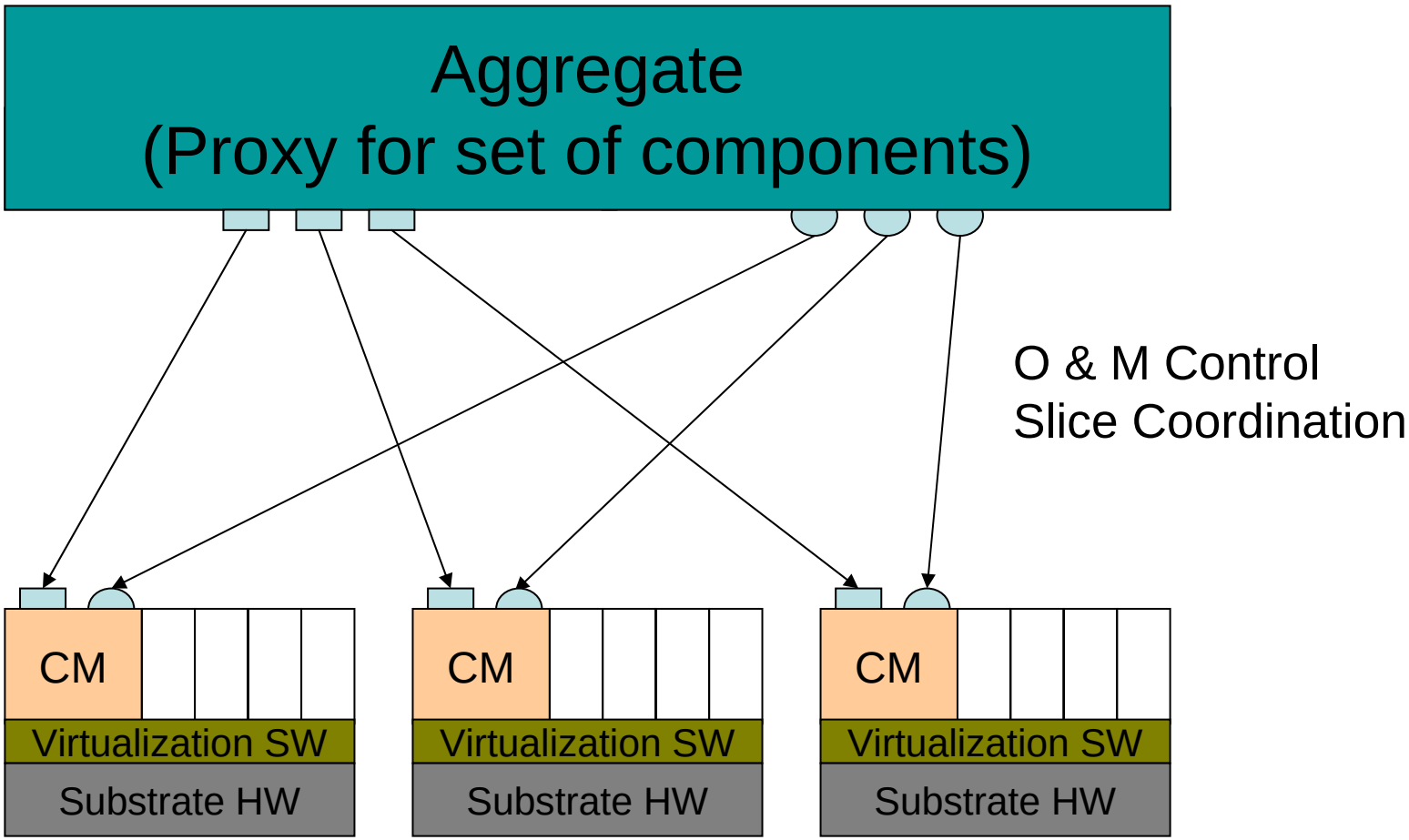
Virtualization Software



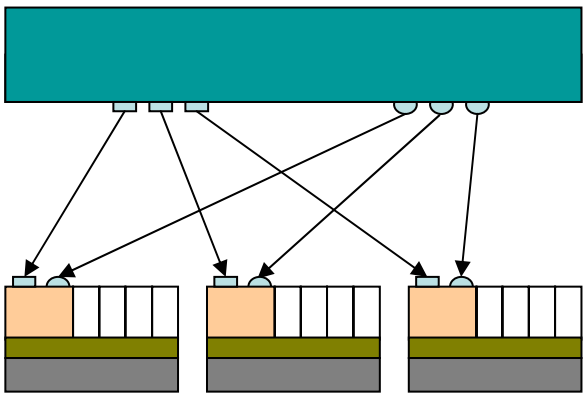
Components

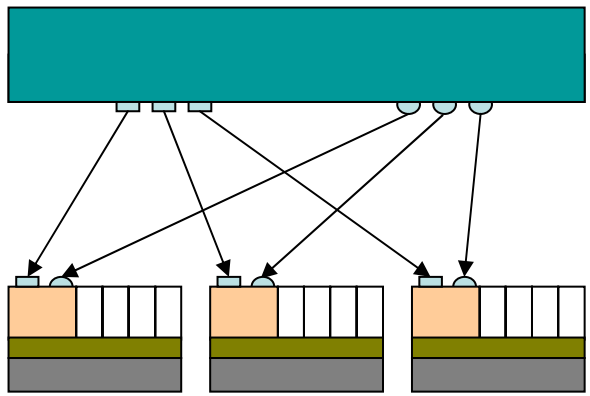

CM
Virtualization SW
Substrate HW
CM
Virtualization SW
Substrate HW
CM
Virtualization SW
Substrate HW

Aggregates

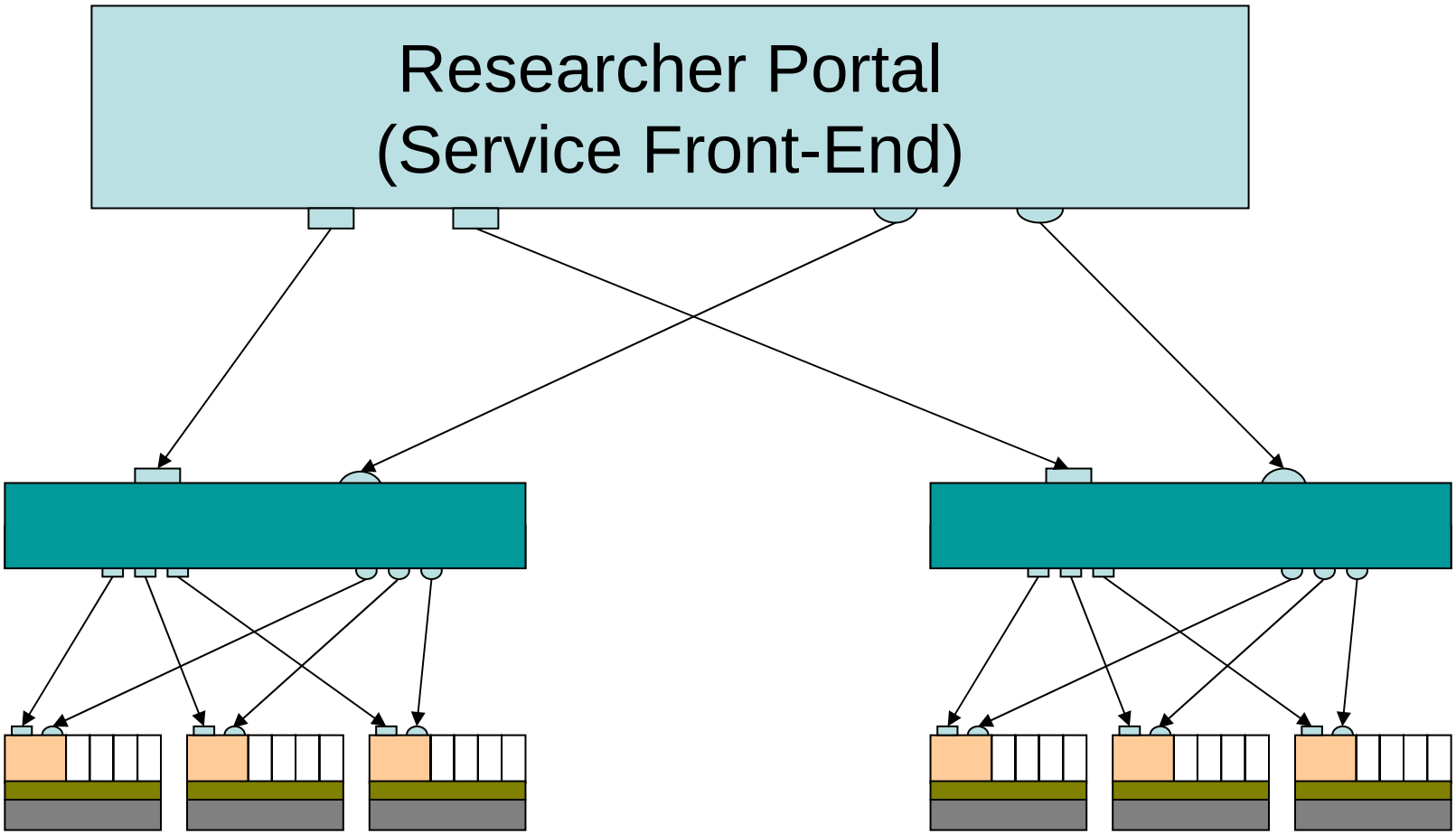

Aggregate
(Proxy for set of components)
O & M Control
Slice Coordination
CM
Virtualization SW
Substrate HW
CM
Virtualization SW
Substrate HW
CM
Virtualization SW
Substrate HW

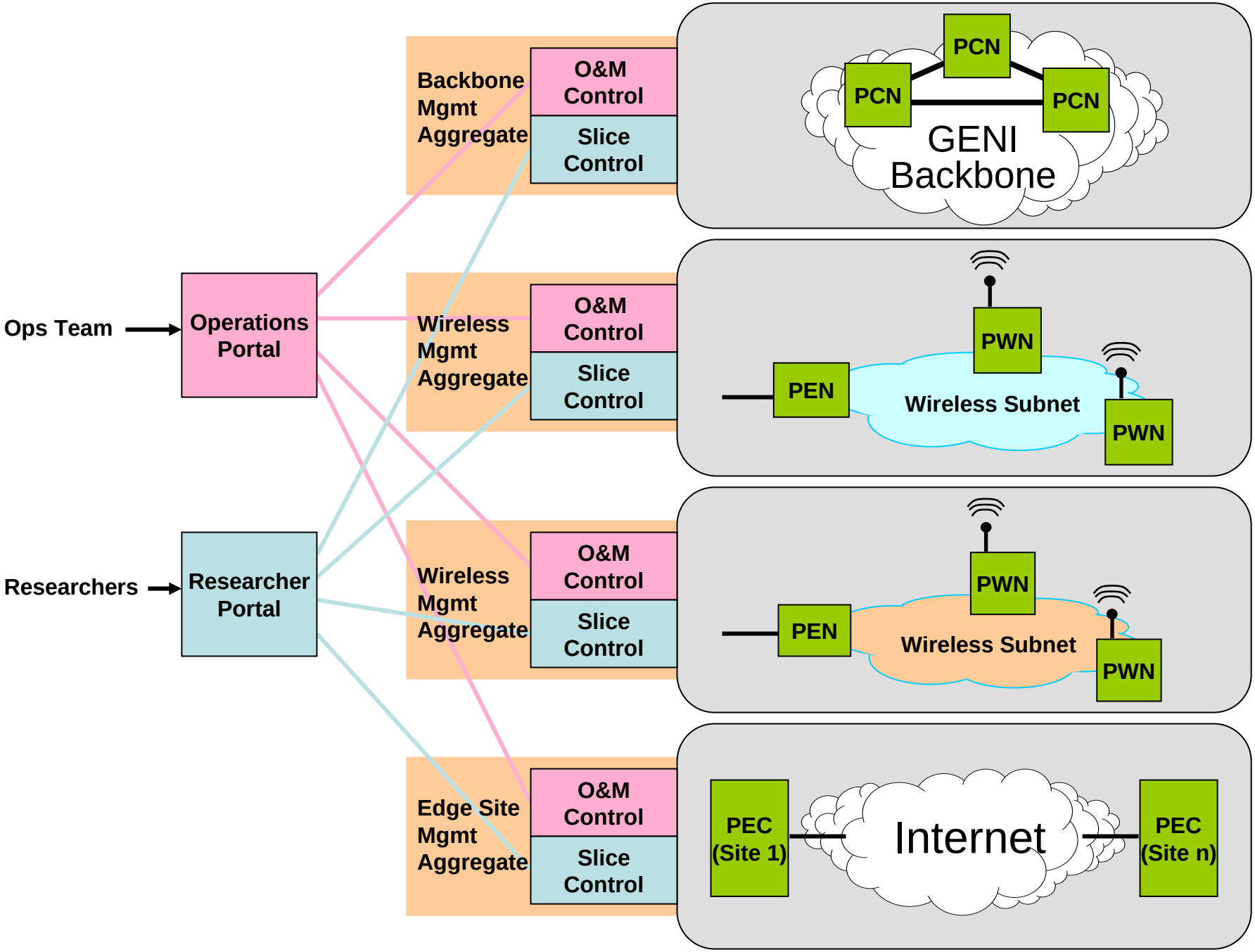
Federation



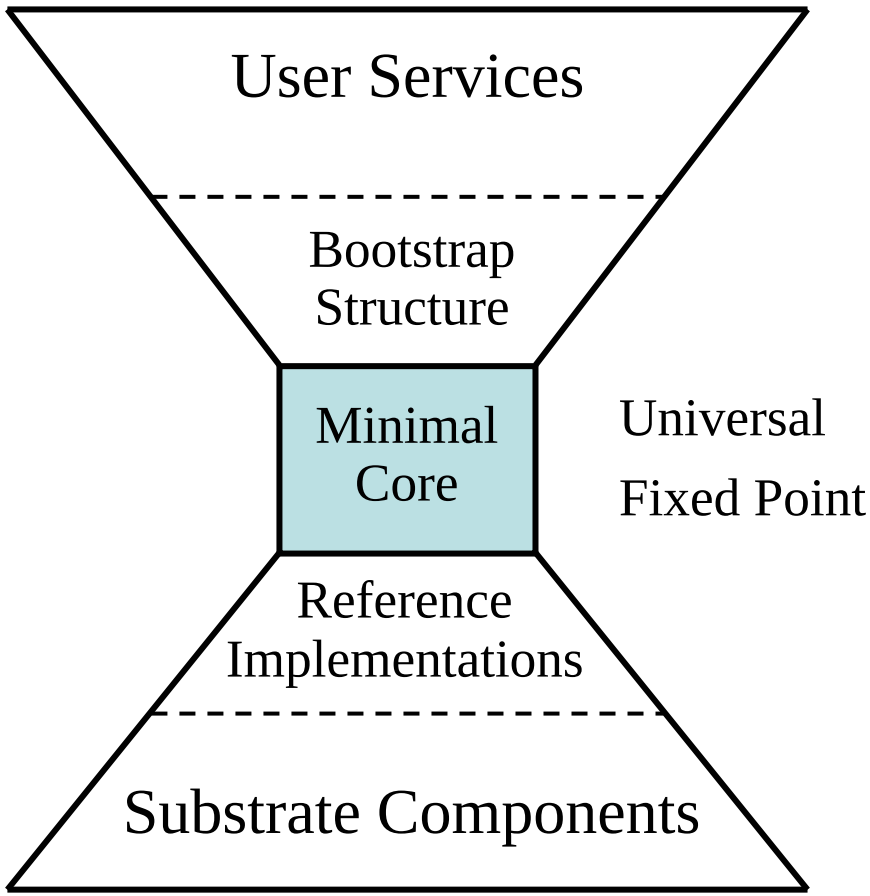


User Portals




Ops Team
Researchers
Operations Portal
Researcher Portal
Backbone Mgmt Aggregate
Wireless Mgmt Aggregate
Wireless Mgmt Aggregate
Edge Site Mgmt Aggregate
O&M Control
Slice Control
O&M Control
Slice Control
O&M Control
Slice Control
O&M Control
Slice Control
PCN
PCN
PCN
GENI Backbone
PWN
PEN
Wireless Subnet
PWN
PWN
PEN
Wireless Subnet
PWN
PEC (Site 1)
Internet
PEC (Site n)

Hour-Glass Revisited



Minimal Core

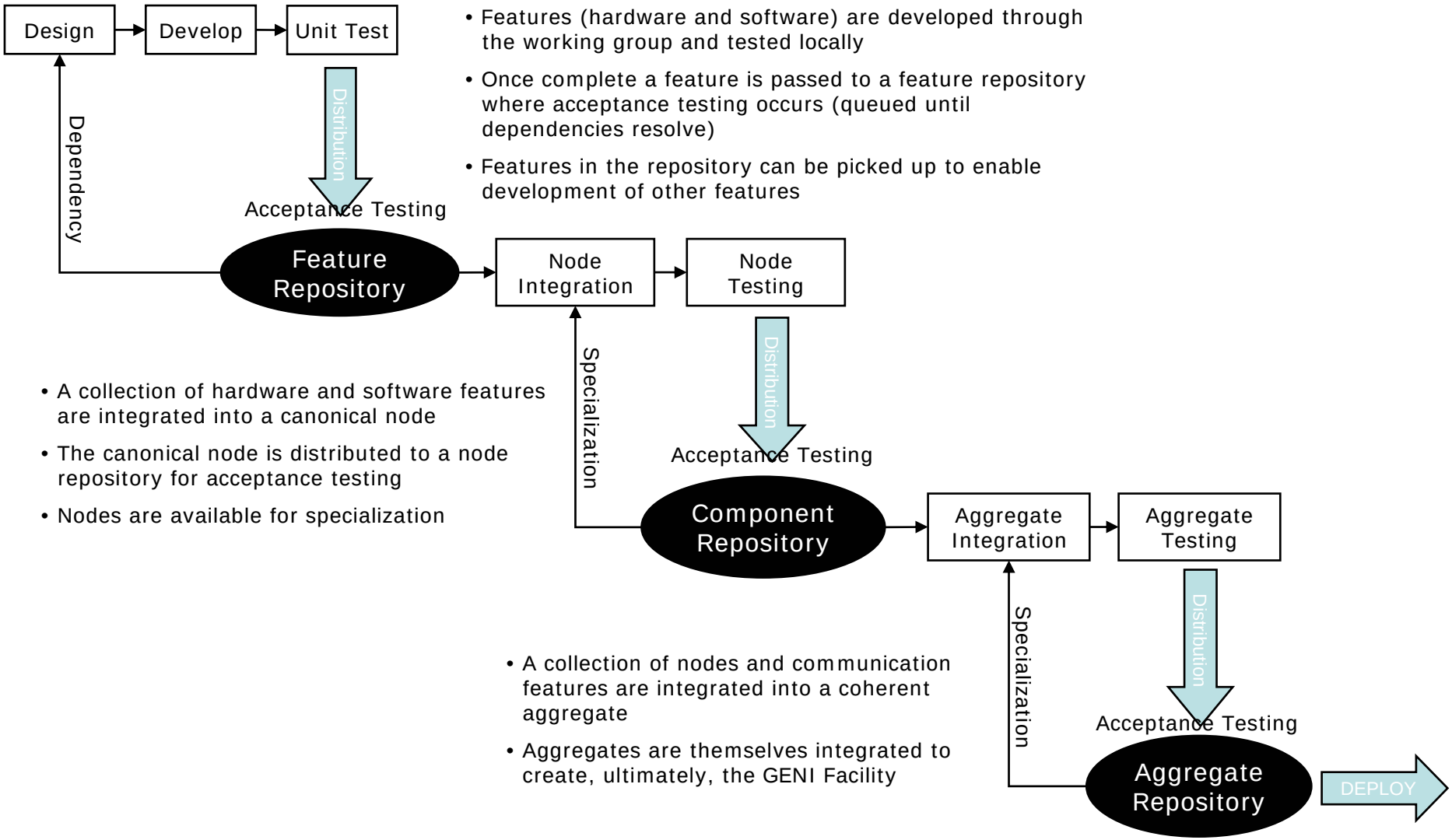
- Principals
  - Slice Authorities (SA)
  - Management Authorities (MA)
  - User (experimenter, not “end user”)
- Objects
  - Slices
    - Registered, Embedded, Accessed
  - Components
- Data Types
  - GENI Global Identifiers (GGID)
  - Tickets (credentials issued by component MA)
    - rspec: resource specification
  - Slice Credentials (express live-ness, issued by SA)

Core (cont)

- Default Name Registries
  - Slice Registry (e.g., geni.us.princeton.codeen)
  - Component Registry (e.g., geni.us.backbone.nyc)
- Component Interface
  - Get/Split/Redeem Tickets
  - Control Slices
  - Query Status

Construction Plan

- Objectives
  - Allow broader community to contribute (not just subs)
  - Scale (federate) the integration effort
- Strategy
  - Feature Development
    - Roughly equivalent to open source development process
  - Component/Aggregate Integration
    - Roughly equivalent to preparing a Linux distribution



- Features (hardware and software) are developed through the working group and tested locally
- Once complete a feature is passed to a feature repository where acceptance testing occurs (queued until dependencies resolve)
- Features in the repository can be picked up to enable development of other features

- A collection of hardware and software features are integrated into a canonical node
- The canonical node is distributed to a node repository for acceptance testing
- Nodes are available for specialization

- A collection of nodes and communication features are integrated into a coherent aggregate
- Aggregates are themselves integrated to create, ultimately, the GENI Facility