Experimenter Workflow and Services WG

Global Environment for Networked Innovation

GENI Engineering Conference (GEC)

www.geni.net

Clearing house for all GENI news and documents

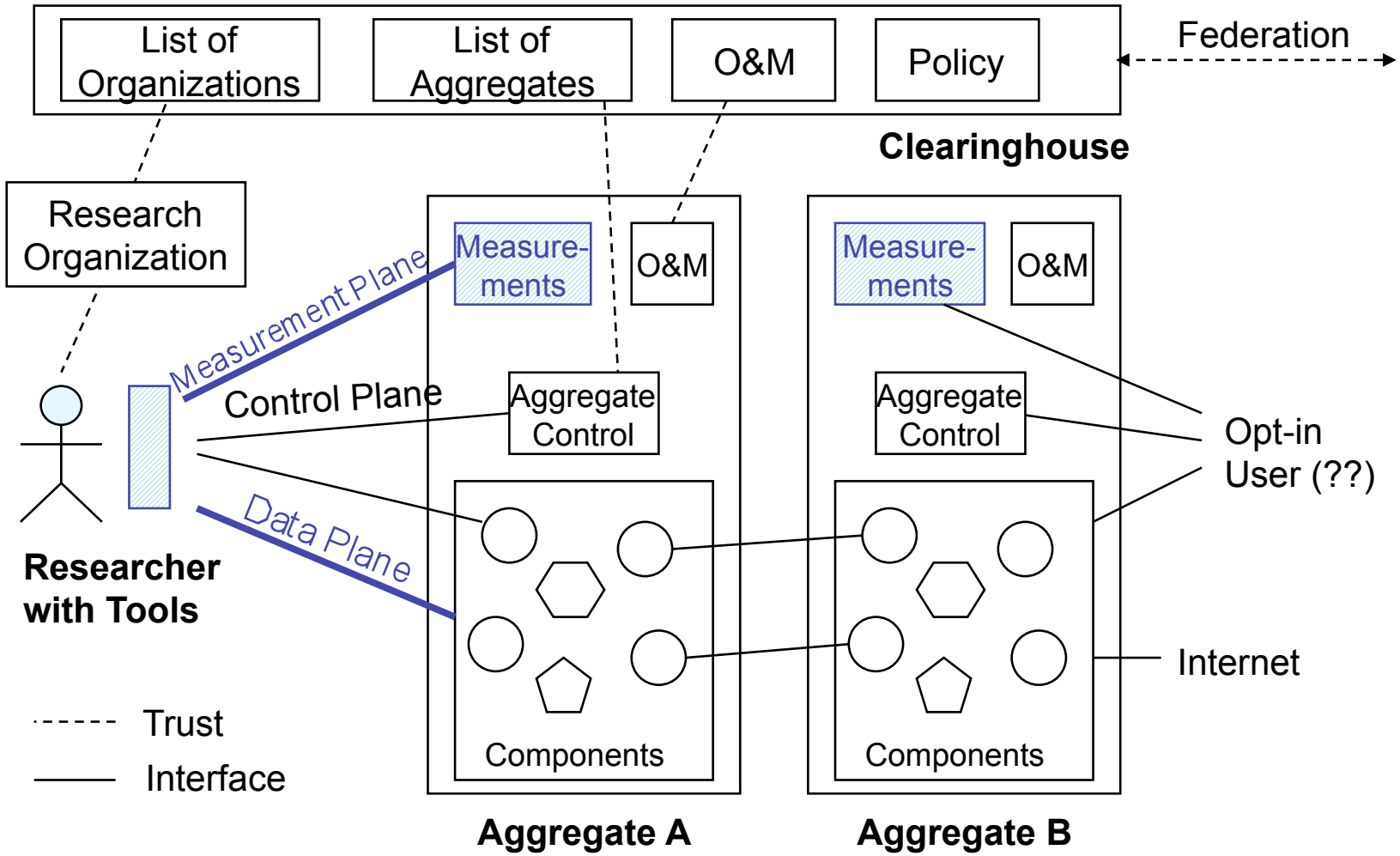
Workflow & Services WG

- Chair: Jay Lepreau, Jeff Chase
- Scope:
  - *What do experimenter-users need from GENI? Consider planning, scheduling, running, debugging, analyzing experiments; long running experiments & how they grow; archiving data.*
- Activities:
  - Develop sample workflows, use cases
  - Define experimenter-support services
  - Develop measurement strategies

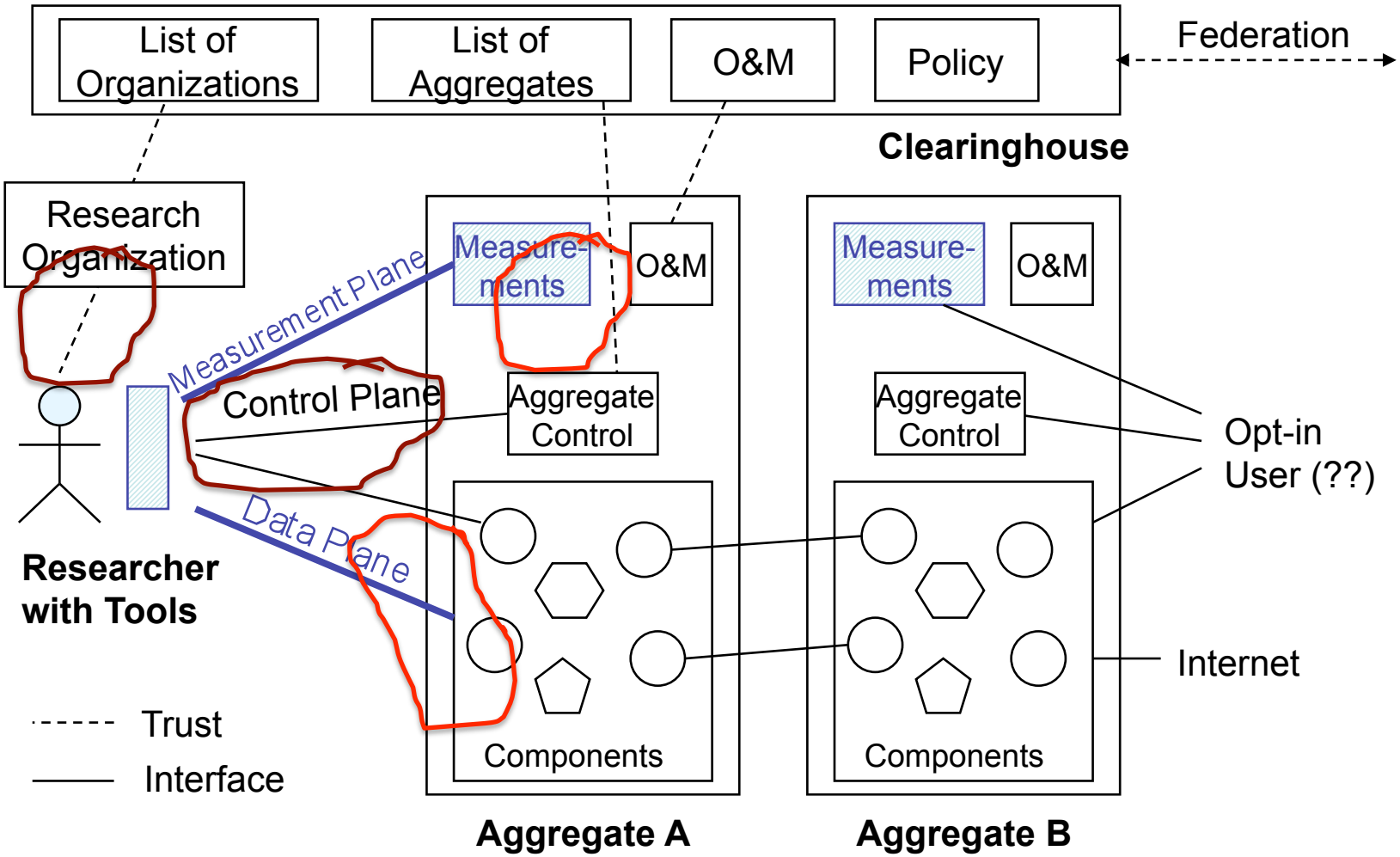
Agenda/Goals

- Explore interactions with other WGs
  - Requirements/risk planning
- Propose new subgroup structure
- Perspectives on “experimenter workflow”
  - Jeannie Albrecht (Williams)
  - Rob Ricci (Utah)

Experimenter Workflow & Services WG



Experimenter Workflow & Services WG



The Researcher-Facing WG

- Control-wg does rspecs
  - Tools that match and query rspecs are here.
- Substrate collects measures
  - Enabling, gathering, processing, querying, archiving are here in services-wg.
- Control-wg does slice (sliver?) control.
  - Services-wg does planning, scheduling, running…
  - *Where is policy?*
- Substrate provides storage
  - Many ways experimenters will use it…

Focus questions

- Specify/design the “core services”:
  - Important enough and hard enough to argue about
  - Must be part of facilities planning
  - Directly motivated by usage scenarios
  - Deliver maximum bang for ease-of-use
  - User-centric, network-centric

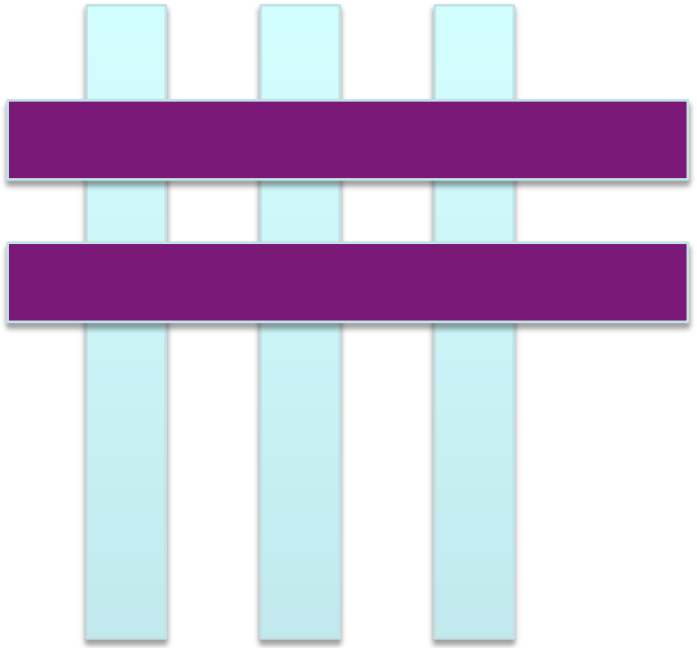
- Enable flowering of extensions/plugins
  - Find/integrate technology pieces of value
- What requirements do these services place on other WGs?

Subgroups (Old)

- Security Architecture
- Edge cluster definition
- Storage
- Resource Allocation
- Experiment Support
- Operations Support
- Communications Substrate
- Legacy Systems

Proposed Refactoring

- Refactor along broad functional groupings.
- Cut across “traditional” research areas.
  - wireless/mobile, optical, storage, security, etc.
- Proposed subgroups
  - Slices
  - Information plane
  - Experiment Building Blocks
  - FDR: Faults/Diagnosis/Repair
    - Fault injection
    - Diagnosis/Debugging
    - Repair/Recovery/Restart



Subgroup: Slices

- Programmatic control/orchestration for GENI experiments
- Specifying, discovering, allocating, and configuring end-to-end resources for a slice
- Slice visibility, containment, and reach
- Incremental allocate/release of resources for adaptive slices, e.g., long-running

Slices: Questions

- What “helper” tools/interfaces and what do they require from the control plane?
- Will GENI enable research on new management services and control plane?
  - If software is the “secret weapon”, what parts of the platform are programmable/replaceable?
- Co-allocation/scheduling of an end-to-end slice?
  - What does “predictable and repeatable” mean?
  - What assurances are components permitted to offer?
- What level of control/stability do we assume over the substrate?

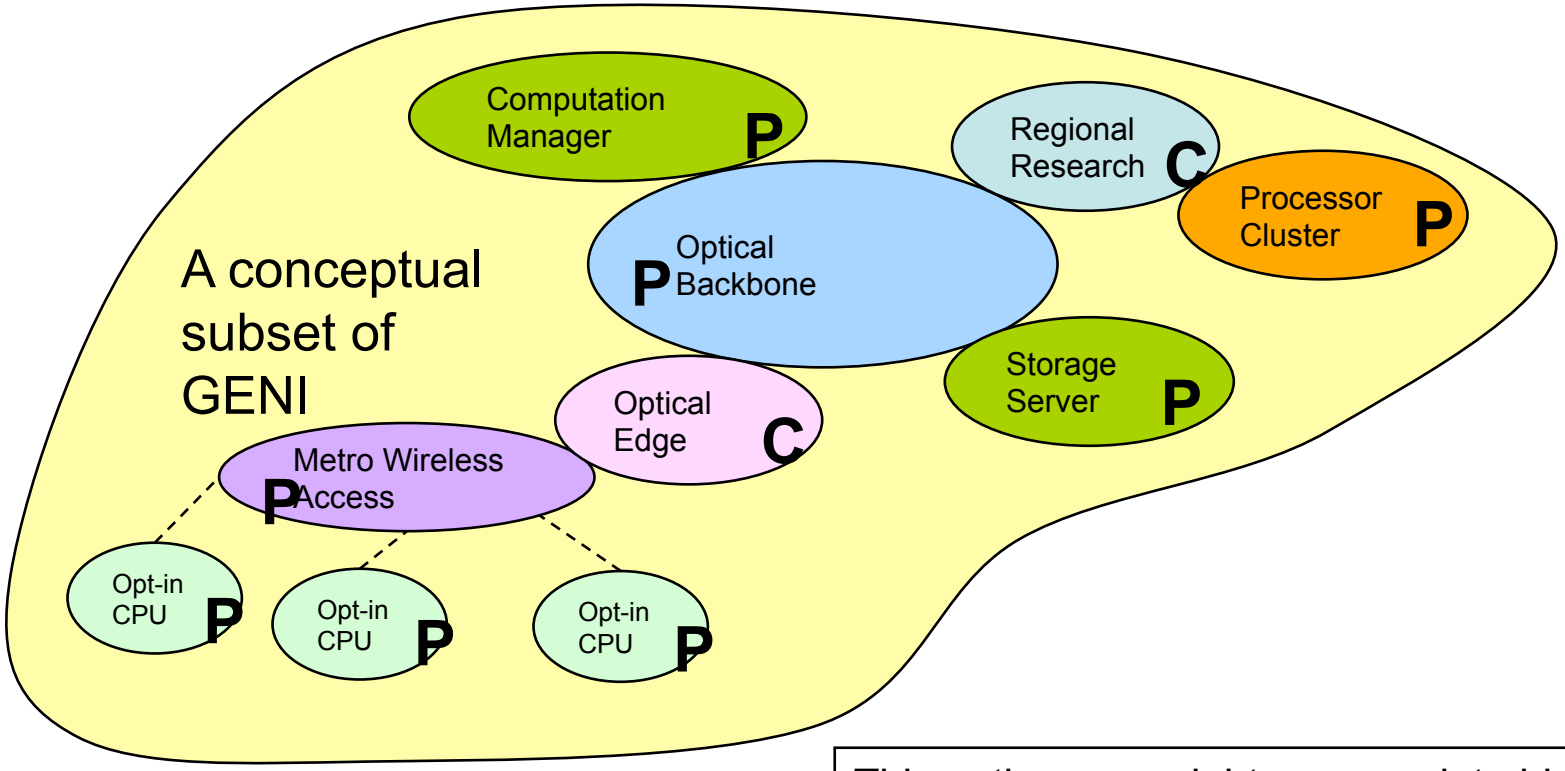
Subgroup: Information Plane

- What can an experimenter (or a facility monitor) know about a slice?
  - Open exploration vs. privacy
- What facilities are offered to collect, process, and archive data from an experiment?
- Measurement plane
  - Monitoring and instrumentation of slices
  - Querying/filtering/routing (pub/sub)
- Time and space
  - Geographic localization, topology
  - Events and allocations

Subgroup: Building Blocks

- Artifact repository
  - Software packages, preconfigured virtual appliances
  - Workloads/traffic, faultloads, standard data sets, etc.
- How do researchers combine these artifacts into a complete configuration?
  - Validating/certifying packages and configurations
  - Packages and slice resources must match
  - Assemble tool chains to process the data
- Provision the slice for the software running in it.
- Select the software for the slice components.

Experiment A's Components



P – GENI resources programmed and configured

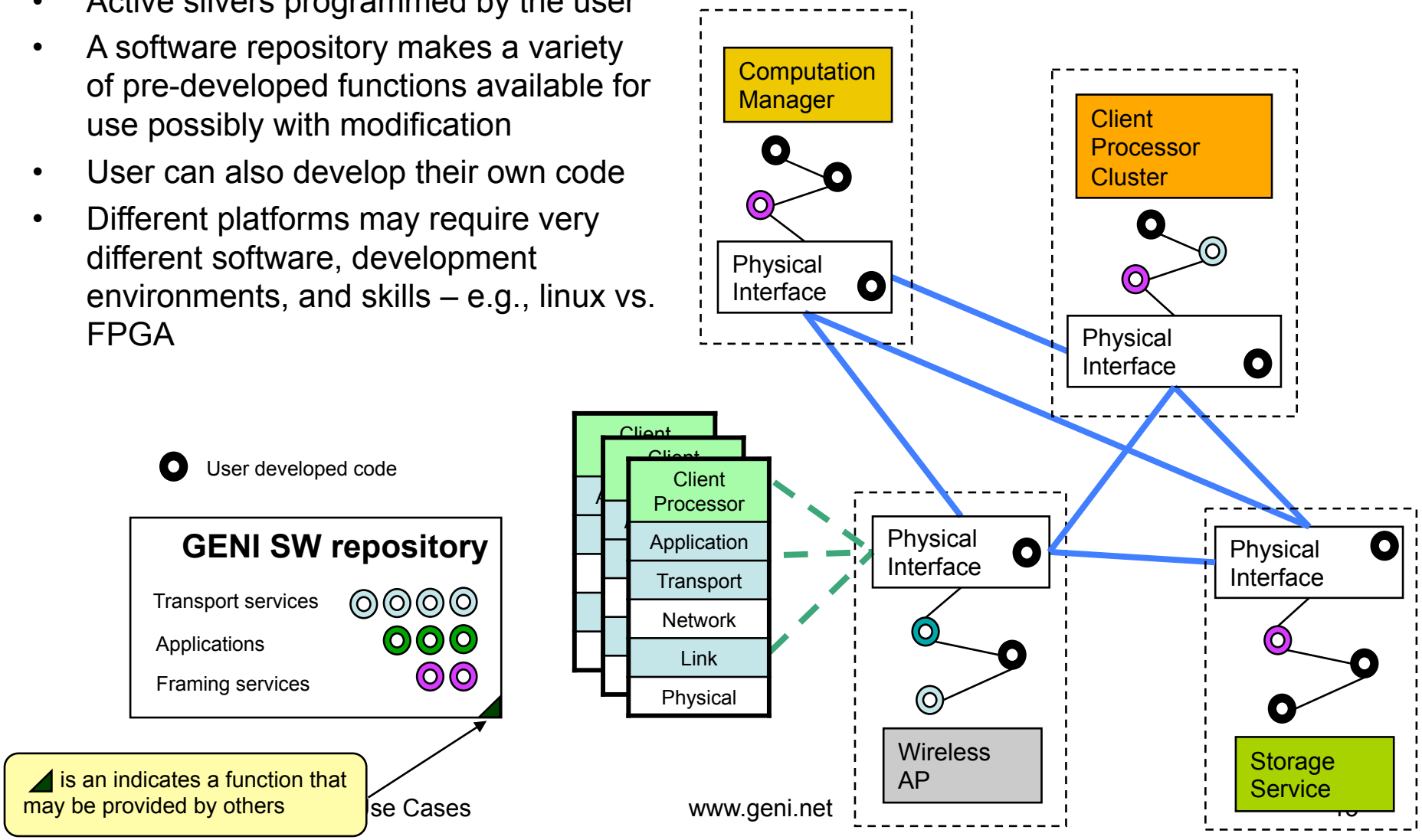
C – GENI resources configured only

Things the user might program into his slice:

- Application functions in computation manager, processor cluster, storage array, opt-in CPUs
- Transport functions in computation manager, storage server, and opt-in CPUs
- Link functions in wireless AP and opt-in CPUs
- Physical functions in optical backbone

Software Repository

- Active slivers programmed by the user
- A software repository makes a variety of pre-developed functions available for use possibly with modification
- User can also develop their own code
- Different platforms may require very different software, development environments, and skills – e.g., linux vs. FPGA



Building Blocks: Questions

- Programmable substrate is key part of GENI value proposition
- Few experiments will program any substrate component from scratch
- What packages/artifacts to draw “off the shelf”?
- How to select them and link them together?

Subgroup: FDR

- Sliver control interfaces for fault injection?
  - How to package and apply faultloads?
- Event log processing to detect and localize faults
  - Invariant checking and anomaly detection
  - Capture, monitor, audit traffic across containment boundaries (e.g., GGW, PlanetFlow)
  - Fault reporting
- Adaptation/repair
  - Fault narratives/signatures
- Suspend snapshot, restart, rewind, etc.

Edge Components

- Range of virtualization technologies, balancing:
  - Overhead
  - Isolation/assurance
  - Customizability/programmability
- Storage is still hard.
  - Roots, sharing, persistence, slivering
- Discovery/allocation/policy services in scope for services WG
  - IP address/port allocation/binding

How much storage for 28E?

1. Local filesystem interface (1E)
2. SQL database (1E)
3. Services for creating new storage services and intercepting storage system calls (1E)
4. Raw disk interface (3E)
5. Block-based storage interface (3E)
6. A wide-area filesystem for administration and experiment management (4E)
7. A high-performance cluster filesystem (4E)
8. Fast write-once/read only storage services. (3E)
9. A reduced complexity storage interface for constrained nodes. (3E)
10. Maintenance and bug fixing throughout life cycle. (5E)

Storage: New (?) Directions

- Decouple services from infrastructure
  - Common “raw” sliverable storage infrastructure?
  - “Let a thousand flowers bloom.”
- Consider separate services separately
- Focus on key storage services for workflow
  - Repositories: Image/appliance, snapshots, source (?)
  - Operational: auditing, instrumentation (write-once)
  - On-demand storage for experiment use
    - Node sliver instantiation (roots)
    - S3

GIMS: New (?) Directions

- One view: filtering and routing are replaceable systems that run within slices.
  - Acquire pipes to the data and slivers close to it
  - Installable filters
  - Routing by underlay or overlay
- Function-ship queries

Path forward

- Bash/populate subgroups
- Participate in joint design groups with other WGs
  - E.g., substrate-specific use cases
  - Boundary tussle with control-wg
  - GIMS
- Continuing focus on requirements planning
- Design: quick and simplistic and open
- Leverage other open technologies wherever possible