Governance of Cyberinfrastructure

Motivation, Concepts, Approach, Call to Arms

Munindar P. Singh

singh@ncsu.edu

Department of Computer Science
North Carolina State University
http://www.csc.ncsu.edu/faculty/mpsingh/

How has e-Science (or IT) evolved?

- **Applications:** Control of computations hidden in code; integration a nightmare
- **Workflows:** Control abstracted out; integration still difficult
- **Standards-driven orchestration:** Integration improved; limited support for autonomy
- **Messaging:** Integration simplified by MoM and transformations; limited support for autonomy
- **Choreography:** Model conversations over messages; limited support for autonomy
- **Governance:** Manage resources via interactions among autonomous parties

What problem does governance solve?

- **Challenge:** How can we manage and share resources in virtual organizations (VOs)? effectively?
- **Idea:** Why not use policies: tried and true?
- **Problem:** Won't applying policies on VOs require specialized
  - Vocabulary and conceptual model?
  - Architecture for monitoring, compliance checking, and enactment?
  - Operational semantics?
  - Design patterns?

Why is governance important?

- Optimal use of resources to serve scientists
  - Share resources in a controlled manner
  - Configure and reconfigure
  - Enable unanticipated uses for resources
  - Administer respecting human organizational needs
- *Comment:* No fundamental difference between virtual and other organizations
- Currently, humans apply subtle considerations manually: make them explicit; “create” new CI assets
- Community-based CIs require governance

Why is governance difficult?

Herding cats

- *Autonomy:* Members behave independently, constrained only by their agreements
- *Heterogeneity:* Members are independently constructed, constrained only by interface descriptions
- *Membership dynamism:* VO configuration changes at runtime
- *Structural dynamism:* Members exhibit complex, evolving relationships

What principles apply here?

- *Separation of concerns*
  - Implementation from concepts
  - Governance from process; process from data
  - “Ilities” from correctness
- *Reification*: create CI assets
  - What was hidden (in code) becomes explicit
  - Metadata is separate from resources
  - All key aspects are inspectable

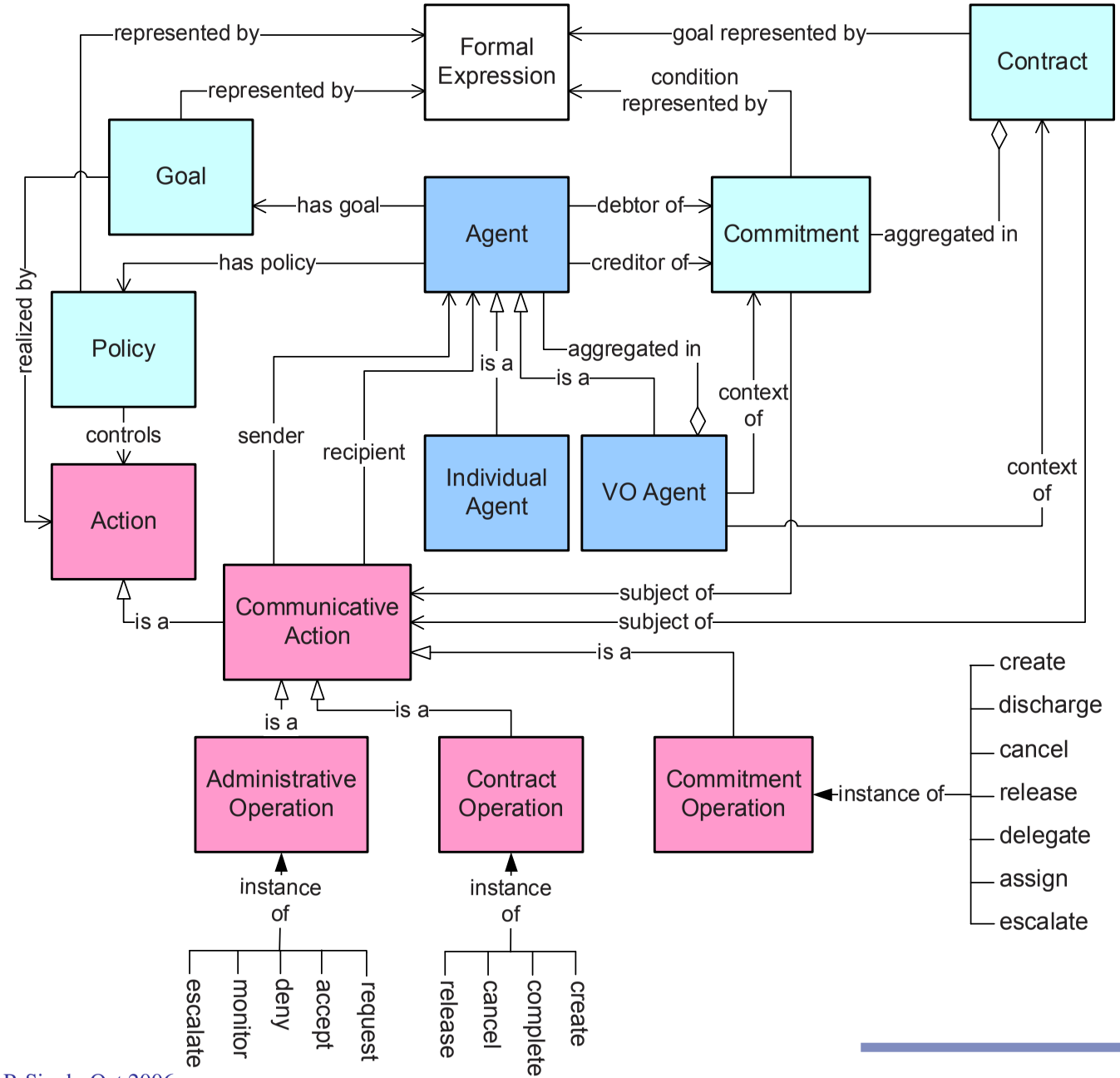
How are these put into practice?

- *Recombination:* Encourage novel combinations
  - Formal representations that support software engineering abstractions: refinement and aggregation
  - Tools for customized repository access
  - Power tools to support unambiguous composition for enhancing repositories
- *Collective intelligence:* Capture social knowledge about above
  - Correctness of particular compositions
  - Contexts where compositions work or fail

What unique features are needed?

- *Intelligent* and *intelligible*
- *User* perspective that complements data, application, system perspectives
  - *Contractual basis:* Key relationships are reflected in contracts
  - *Management of context:* A VO recursively provides the context for interactions and policies of its members
  - *Policy:* An implementation-independent model and operational semantics
  - *Protocol orientation:* How agents apply policies to enter into, monitor, and enact contracts

What are its main ingredients?



How do contracts and VOs relate?

They are duals

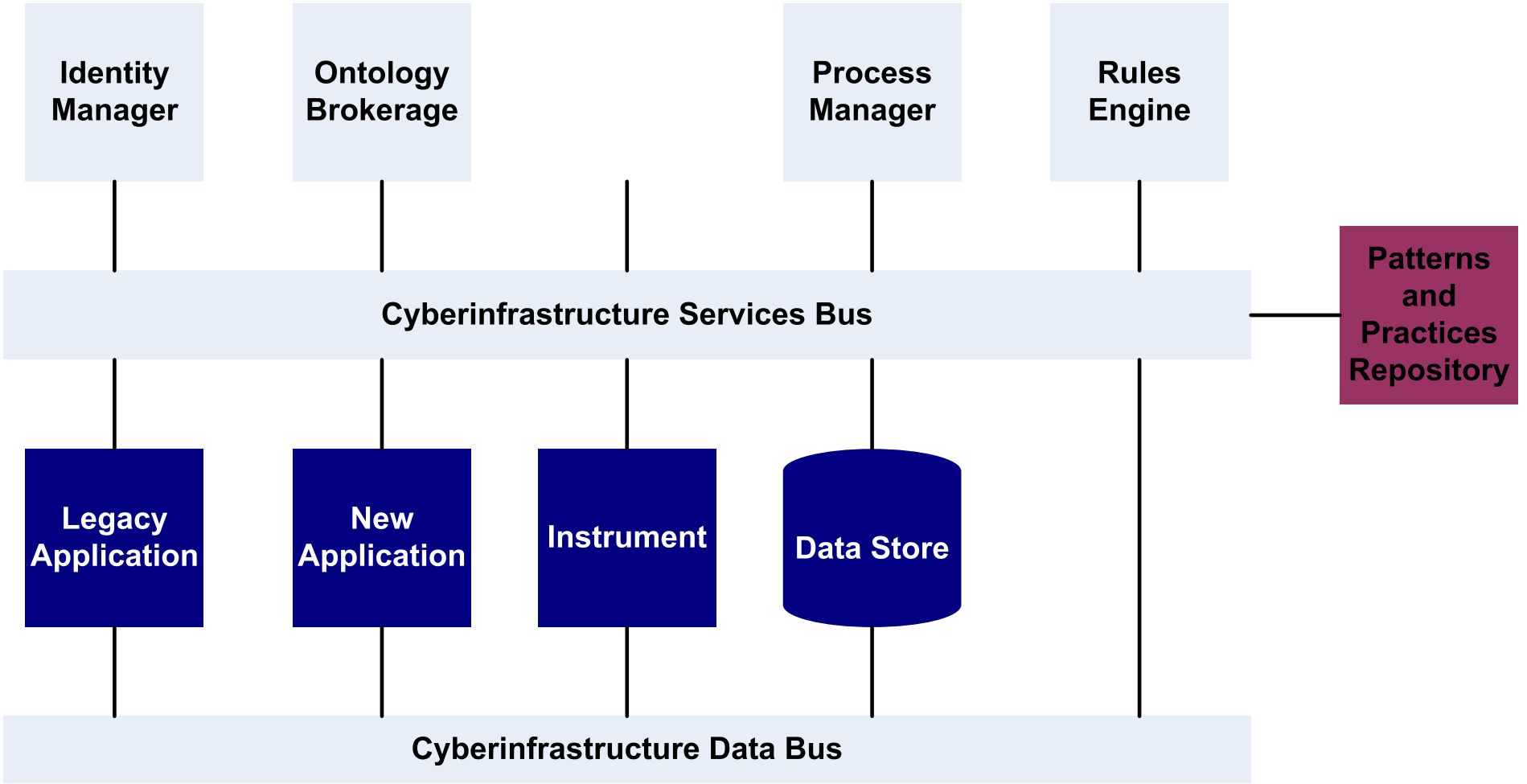
- *Contract:* static entity capturing relationships among two or more agents, reflected in a VO
  - A contract arises within a VO where the contracting agents are peers
  - The enclosing VO would have been created by a prior contract
- *VO:* dynamic (evolving) entity: hosts commitments, contracts, authorities
  - Created through a contract
  - Provides a basis for creating, manipulating, and enacting contracts

How about security and integrity?

Authentication, credentials, authorization, accounting, audit

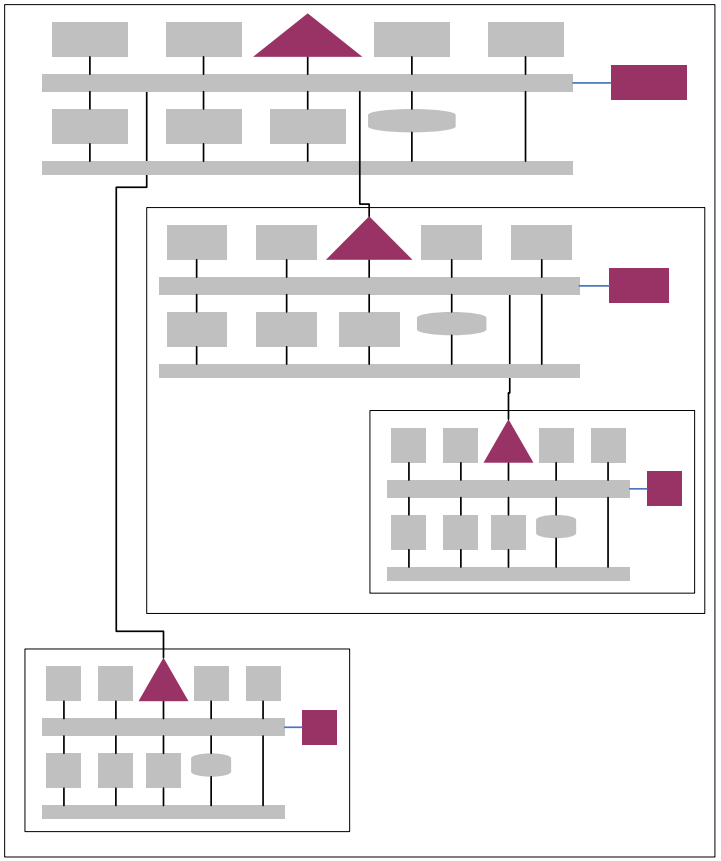
- VOs define namespaces and roles
- Each VO specifies its policies for AAA
  - Authentication could be centralized, e.g., on a “root VO”
  - Peer VOs’ roles may be recognized, but suitably limited
  - Resources can be committed to satisfy upper-level commitments

What is our system architecture?

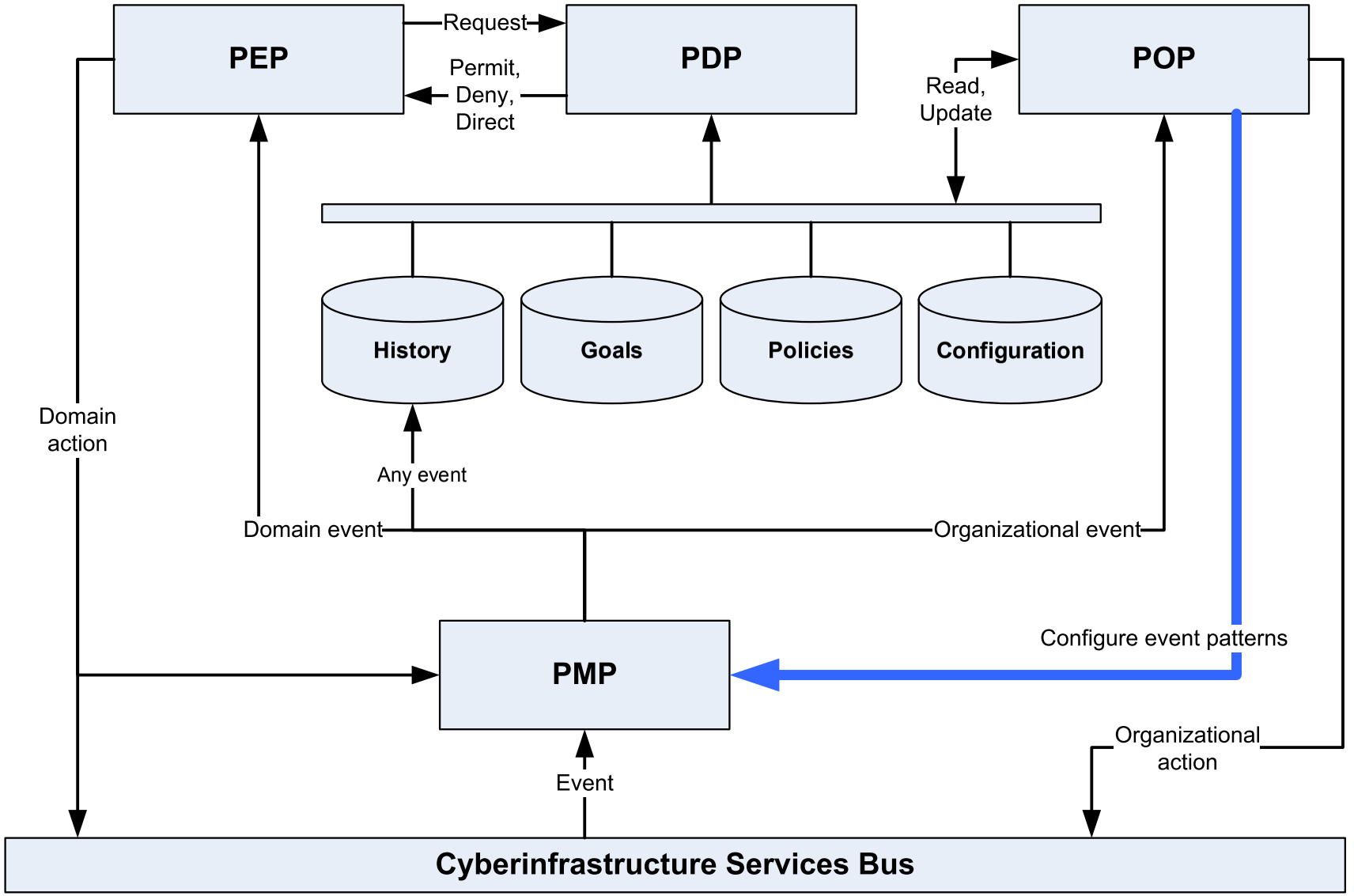


How is this a fractal structure?

VOs nested in VOs: it's turtles all the way!



What is a point of governance?



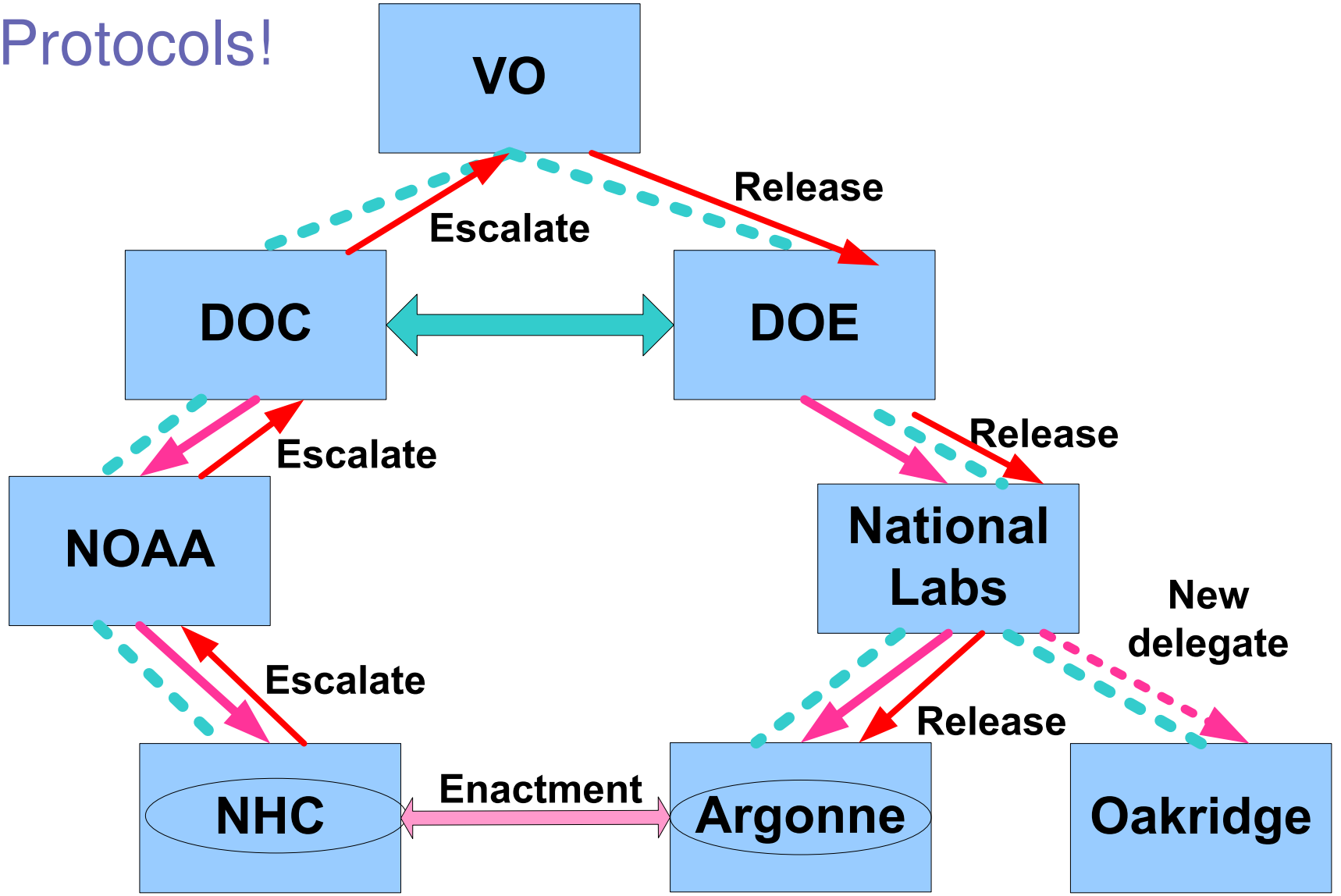
What are protocols?

Semantic choreographies: the interactive, reusable portions of processes

- Describe interactions as classes describe objects
- Specify messages and how they affect interaction state
- Specify well-defined roles
  - Capturing obligations on an endpoint
  - Setting local policies while complying with a protocol
- Stored in a repository, i.e., a CI asset
- Refined and composed for implementation

How do we compute preemption?

Protocols!



What is our current agenda?

Bridging the gap between current architectures (e.g., ESB or enterprise service bus) and user needs

- Capture and generalize scenarios known to be of user interest
- Develop a repository of validated protocols
- Extend and incorporate current tools: OWL-P (protocols) and MAVOS (multiagent virtual organization system)

Have we heard of these?

- Policies for VOs (Foster; Feeney; . . .)
- Policy languages (Ponder; Datalog; Rei; . . .)
- Organization theory
- IETF Policy Framework
- XACML
- DMTF's Common Information Model
- WS-Agreement
- WS-Policy