Invited Presentations

Keynote

Mechanized Policy, Fact or Fancy?

Joe Pato, Principal Scientist, Trusted E-Services Lab – HP Labs, Cambridge MA, USA

Policy and the IETF – Theory and Practice

John Strassner, Past chairman IETF Policy Working Group, Cisco, USA

Provisioning Your Future through Policy-based Management

Rick Roeling, Openview PolicyXpert Architect, Hewlett-Packard Company, USA

Trust Management and Security Policy

Matt Blaze, AT&T Research Laboratories, USA

On the negotiation of Access Control Policies

Virgil Gligor, University of Maryland, USA

Policy in the Standards Bodies - Theory & Practice

John Strassner
Cisco Fellow

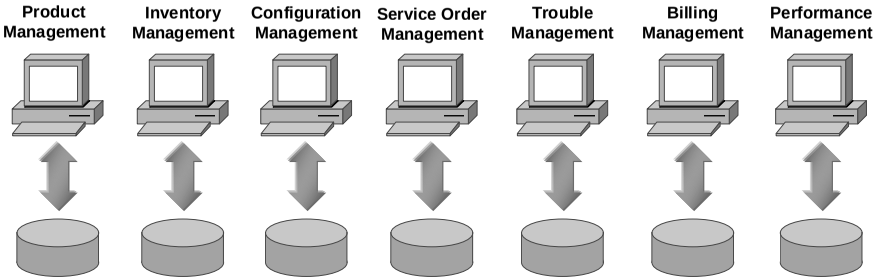
Agenda

- Introduction and Background
- Overview of the PCIM
- Overview of the QPIM
- Roles of the IETF and the DMTF

Policy Networking Scenario

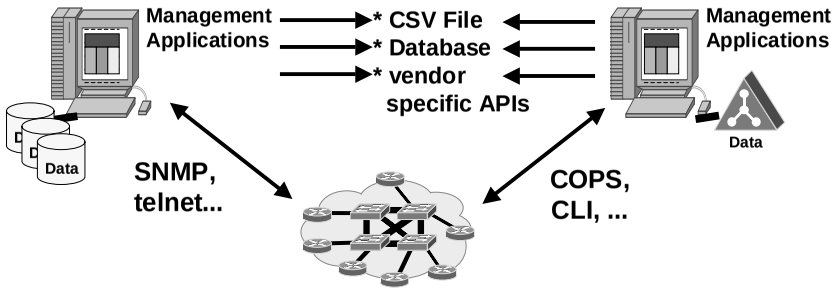
- Network Configuration/Management
  - » Financial Institution must cope with different operational scenarios
    - – normal operation, high-volume, and emergency meltdown policies
    - – configure the network, not just individual isolated device interfaces
    - – map business rules and procedures to applications that use the network

Shortcomings of Today's Network Operations



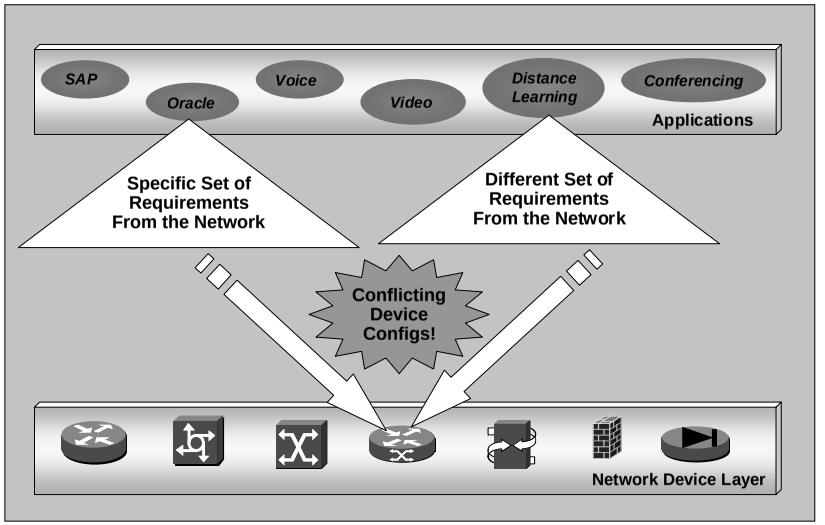
- Architectural issues
  - » Data redundancy
  - » Synchronization problems
  - » Application authorization issues
  - » Vendor and Application "lock in"
- Integration issues
  - » Isolated Data Silos
  - » Administrative nightmare
  - » Integration/customization nightmare
  - » Transition from legacy systems to a more flexible new operational architecture

Integration and Data Exchange Issues in the Past



- Vendor promise: single database…
- Integration: hasn't happened!
- Missing standards:
  - Application integration
  - Data exchange

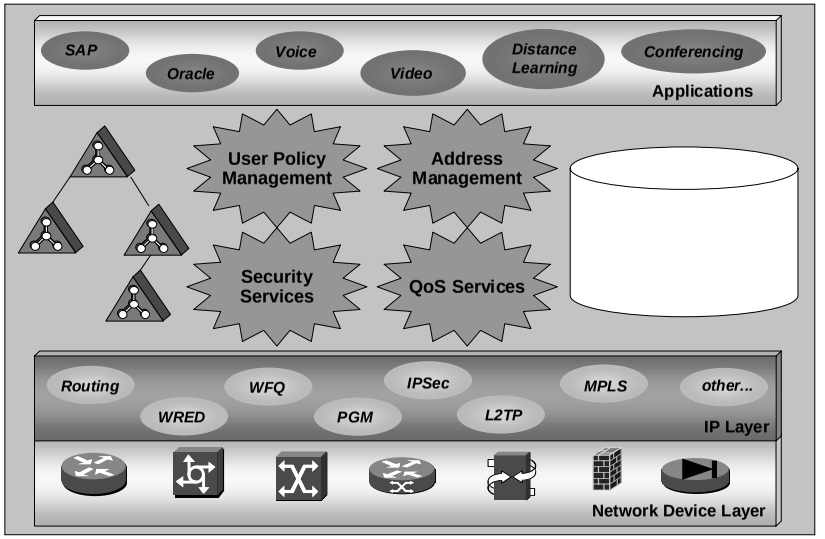
Need for DEN and Policy



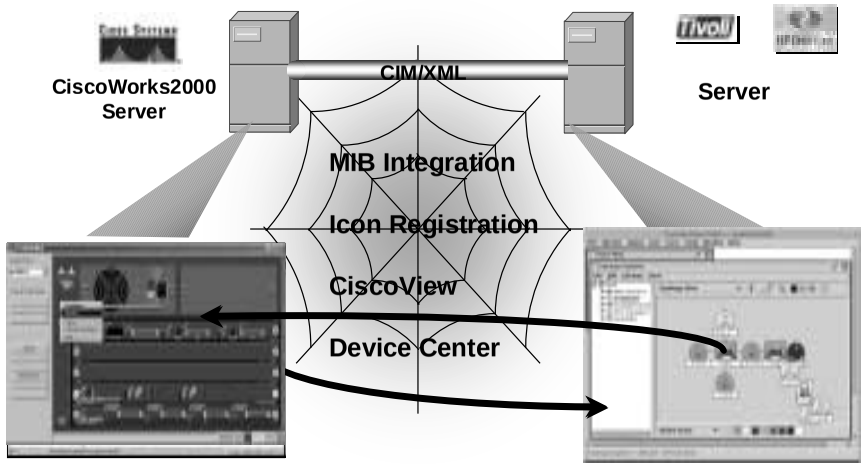
What is the Solution?

- An *information model* that defines management abstractions of
  - » profiles and policies that can be applied to
    - – devices, protocols, and services
- This provides
  - » a unified model for integrating users, applications, and networking services
  - » an extensible *service-oriented framework*

Intelligent Network Services



CIM Data Exchange



Agenda

- Introduction and Background
- Overview of the PCIM
- Overview of the QPIM
- Roles of the IETF and the DMTF

Purpose of the PCIM

- Provide a set of classes and relationships that provide an extensible means for defining policy control of managed objects
  - » Represents the structure, not the contents, of a policy
  - » Content provided by subclassing classes to derive technology- and vendor-specific conditions, actions, and other elements

PCIM Overview (1)

- Policy-based management assumes that the network is modeled as a state machine
  - » Classes and relationships are used to model:
    - – the state of an entity
    - – the actions that either maintain an entity's state or move the entity to a new state
    - – settings to be applied to an entity to maintain or change its state

PCIM Overview (2)

- Thus, policy is applied using a set of rules
  - » Each rule has a set of conditions that specify when the policy should be applied
    - – Conditions can be specified in CNF or DNF
  - » Each rule has a set of actions that are executed if the conditions are TRUE
    - – Execution order can be specified
  - » Rules may be prioritized and grouped together to model an administrative hierarchy

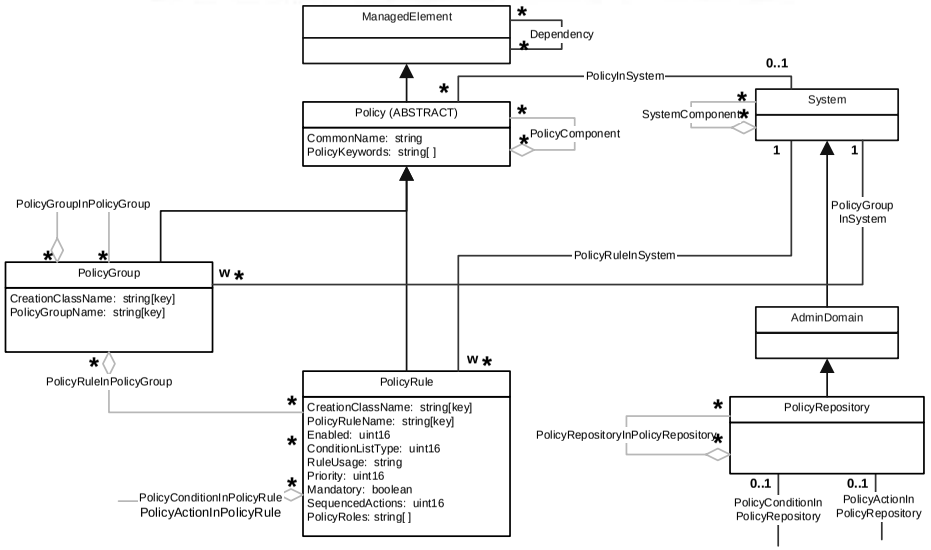
Reusable Components

- Policy components can be specific to a rule or reusable among many rules
  - » Rule-specific information is attached to the rule itself
  - » Reusable information is stored in a container that is referenced by the rule
- The only difference between a reusable and a rule-specific component is in the intent of the administrator
  - » *No difference in functionality*

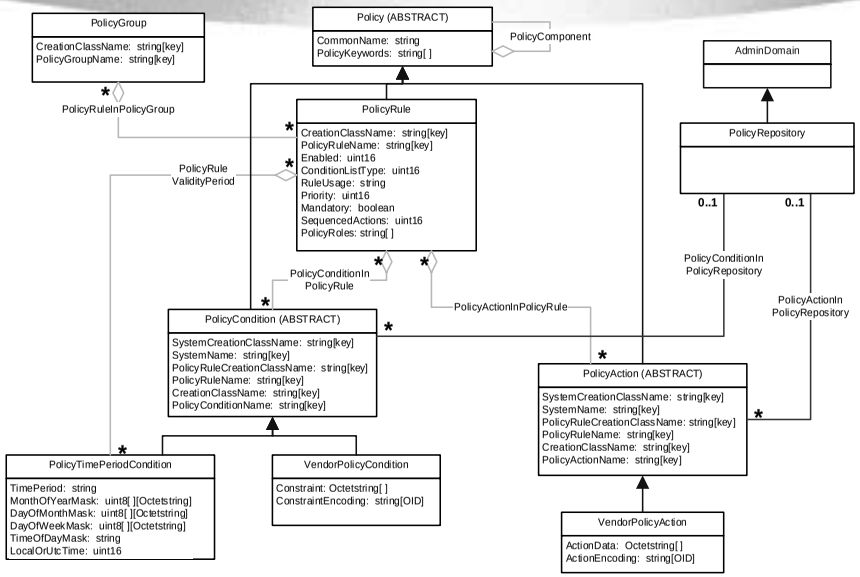
Reusable Components (2)

- PCIM defines a policy repository to store reusable information. This causes some subtle differences, including:
  - » access control can be specified for rule-specific conditions and actions, but not for reusable ones
  - » referential integrity should be enforced for rule-specific elements; harder to due in the reusable case
  - » mapping to a data model is more difficult

Top-Level Policy Classes



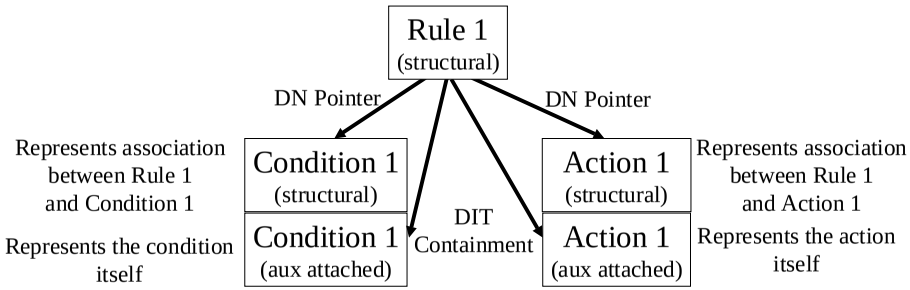
Policy Conditions and Actions



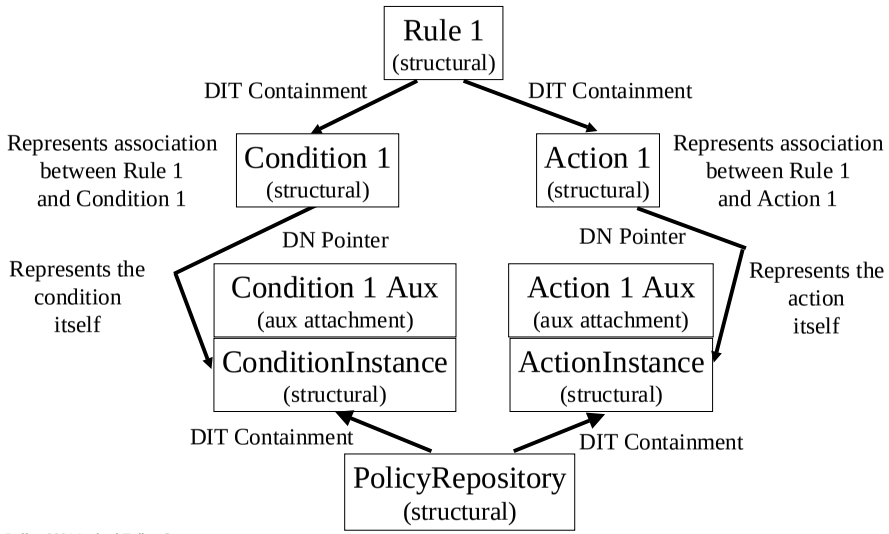
Structure of a Rule-Specific Policy

- **PolicyRule is a container that holds PolicyConditions and PolicyActions**
  - **» QPIM extends this so that a condition is treated as a container**
- **To do this attachment**
  - **» PolicyRule is a structural class**
  - **» PolicyCondition and PolicyAction are both auxiliary classes**

Rule-Specific Example



Reusable Rule Example



Agenda

- Introduction and Background
- Overview of the PCIM
- Overview of the QPIM
- Roles of the IETF and the DMTF

Purpose of the QPIM

- QPIM provides a set of classes and relationships that manage and control device-specific IntServ and DiffServ QoS mechanisms using policy
  - » Extends PCIM
  - » Info model provides interoperability between policy servers, policy management applications, and devices

Problems Solved

- QPIM solves two different problems
  - » Different behaviors are abstracted
    - – QPIM defines device-independent abstractions that build QoS policies that manage these behaviors
  - » Prevents the same policy rule from provisioning different devices differently
    - – QPIM defines common semantics and definitions to build, interpret and enforce high-level policy rules

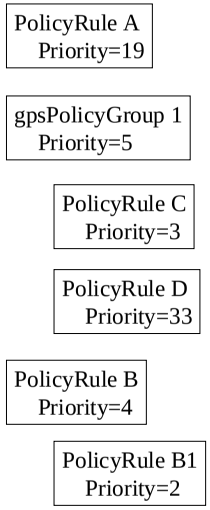
Extensions to PCIM

- Four main extensions
  - » Data organization
  - » Hierarchical policy repositories
  - » Extensions to reusable objects
  - » Extensions to the structure of a PolicyRule

Data Organization

- Establish notion of a PolicyDomain
- Define two additional grouping constructs
  - » qosPolicyDomain
    - – Defines root of admin domain, contains policy rules and other data of a domain
    - – Defines decision-match strategy for evaluating policy rules
  - » gpsPolicyGroup
    - – Specialized container having its own decision-match strategy, priority, and role-combinations

Using the gpsPolicyGroup Class



Decision-Match Strategy

PCIM
D, then A, then B, then C, then B1
no concept of evaluating a policy container

QPIM, match-first
A, then D, then C, then B1, then B

QPIM, match-all
same as above, except set aside all matching rules for further processing

QPIM, domain is match-all, container is match-first
A, then either D or C, whichever matches first, then B1, then B

Hierarchical Policy Repositories

- Objects in different parts of the data tree are independent of each other
  - » Policy repositories usd to share data
  - » Hierarchical policy repositories used for hierarchical administration and scoping
  - » Extend to storing reusable information, not just conditions and actions

Reusable Object Extensions

- Reusable objects extended from PCIM
  - » Condition defined as an ordered triplet {variable, operator, value}
  - » Variables and values can be reused
  - » Values can be constrained
  - » Binding between variables and values is defined in QPIM

PolicyRule Structural Extensions

- Reuse PolicyRule for interoperability
- Add new subclasses of PolicyCondition and PolicyAction to handle additional semantics

Additional Concepts Not In PCIM

- Five new additions
  - » Rule nesting
  - » Rule decision strategies
  - » Compound conditions
  - » Pre-defined variables and constants
  - » Per-Hop Behavior definitions

Rule Nesting

- PCIM defines the PolicyGroup class for grouping together PolicyRules
  - » Treats a PolicyRule as an atomic object
- QPIM defines the ability to nest a PolicyRule within another PolicyRule
  - » Example
    - – High-level rule for logon
    - – Lower-level, nested rules for doing each piece of the logon

Rule Decision Strategies

- Two different strategies defined, match-first and match-all
  - » Match-first exits as soon as a condition is satisfied
  - » Match-all logs all of the matching conditions and then revisits them in priority order to execute their actions

Decision Strategy

- **defined per** QoSPolicyDomain **and can be overridden per** qosNamedPolicyContainer**.**
- **The** order **of decision making of nested rules is defined by the combination of their** internal priority**, the** priority **of the** policy rule **containing the nested rule and the** priority of their containers**.**

Compound Conditions

- **PCIM defines a generic condition as a single term**
  - » **Multiple conditions are implemented as multiple terms requiring multiple objects**
- **QPIM defines a compound condition that can aggregate a set of simpler conditions**
  - » **One object now contains multiple condition terms, providing better performance**

Pre-Defined Variables and Constants

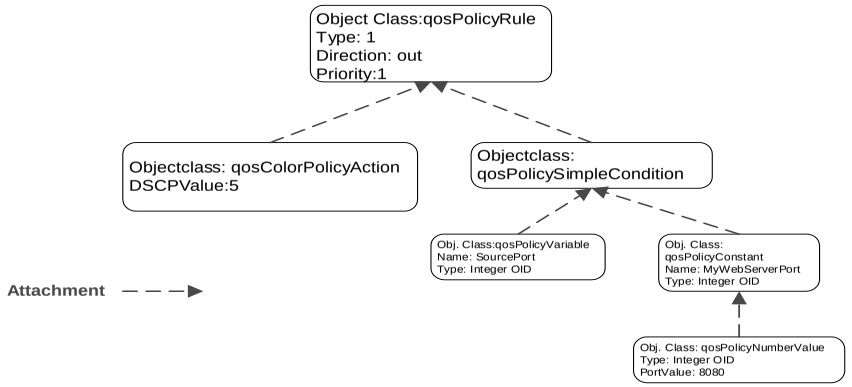
- **Not present in PCIM**
- **QPIM defines variables and constants as classes**
  - » **needed to represent fine-level details of a condition as an ordered triplet**
  - » **manages the binding between variables and values**
  - » **can optionally constrain the values that a variable can take**

Per-Hop Behaviors

- **Not present in PCIM**
- **QPIM defines the ability to define PHBs and apply them to network devices using policies**

Simple QoS Policy Rule (Attachment)

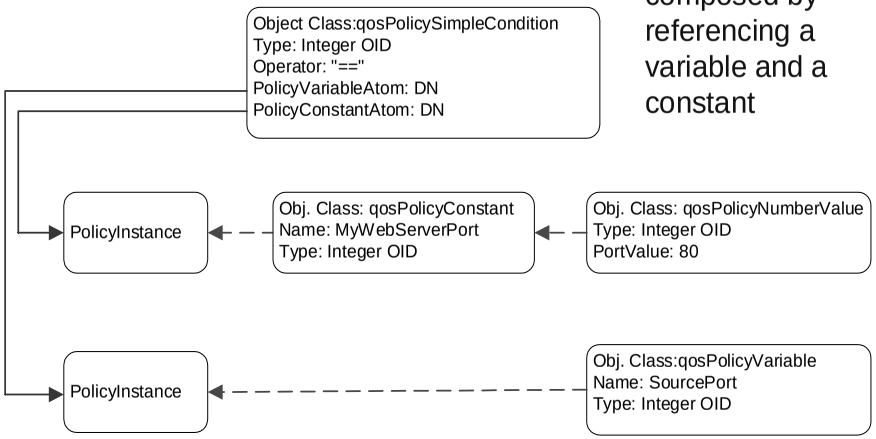
if (SourcePort == MyWEBServerPort) then Color DSCP=5



Simple Condition Example (Reference)

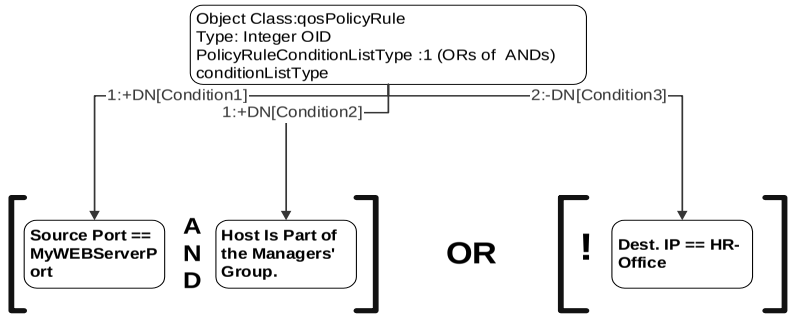
SourcePort == MyWEBServerPort

Simple condition composed by referencing a variable and a constant



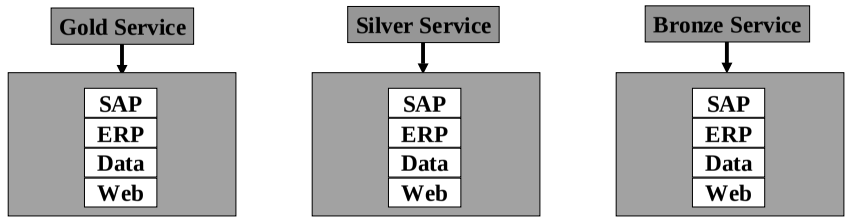
Complex Condition Example

[SourcePort == MyWEBServerPort & Host Is Manager] **OR** [Destination IP != HR-Office]

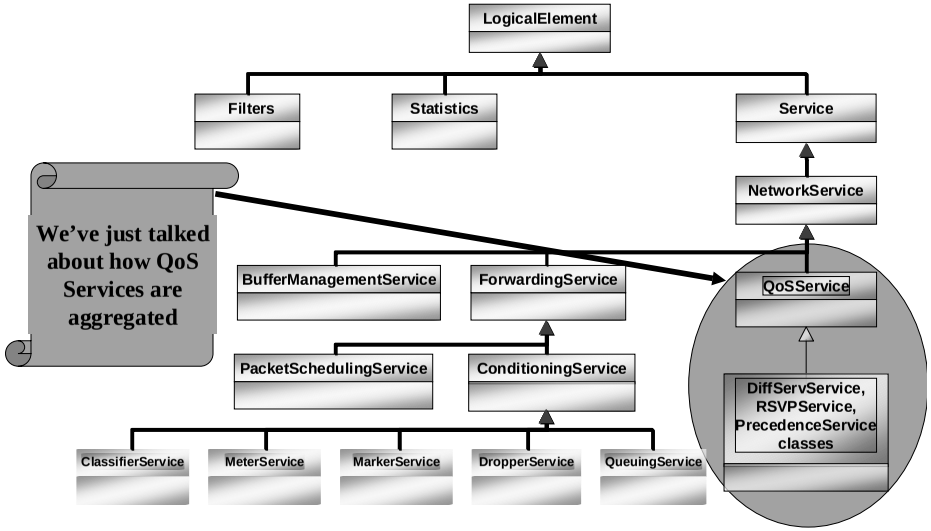


Modeling Services That Need QoS

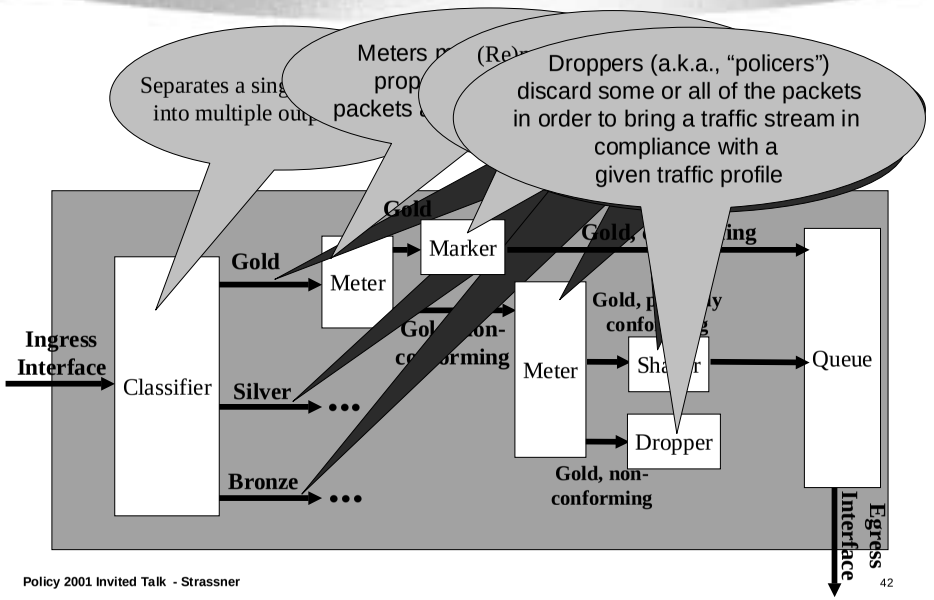
- **The difference between Gold, Silver and Bronze is:**
  - **FUNCTIONALITY (Silver isn't allowed to send SAP, and all Bronze can send is Data and Web)**
  - **QUALITY (Data is a common service, but the drop probability is highest for Bronze and lowest for Gold)**
  - **Services associated with each Olympic service are treated COLLECTIVELY better**



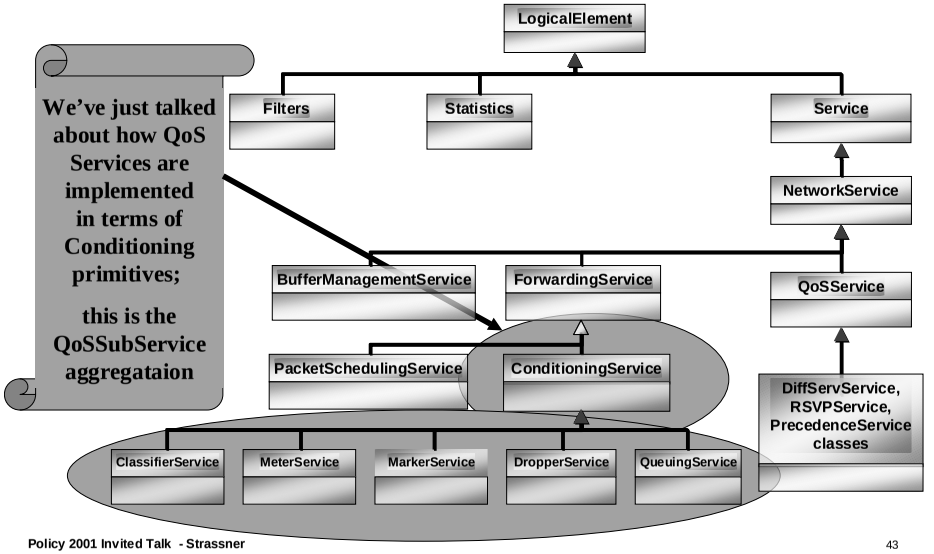
Overview of the QoS Model



Modeling the Forwarding Path



Overview of the QoS Model



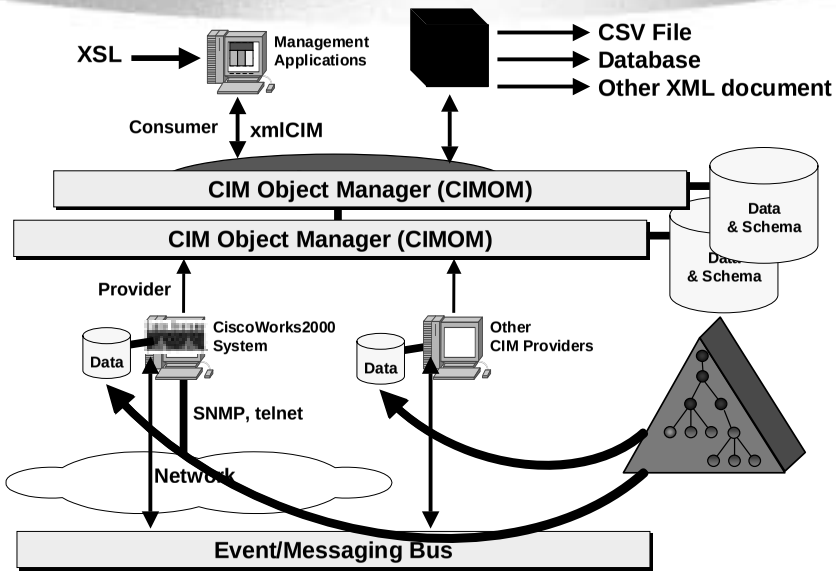
Agenda

- Introduction and Background
- Overview of the PCIM
- Overview of the QPIM
- Roles of the IETF and the DMTF

Standards Efforts

- IETF is focusing on
  - » use of info and data models to support technology-specific mechanisms in protocols
  - » configuration/management use of a protocol and info/data models
- DMTF is focusing on
  - » information models, and mapping to LDAP
  - » xmlCIM as a way to build applications that share and reuse data

How Does This Work?



CISCO SYSTEMS

EMPOWERING THE INTERNET GENERATION℠

Provisioning Your Future Through Policy-Based Management

Rick Roeling

Architect
OpenView Provisioning

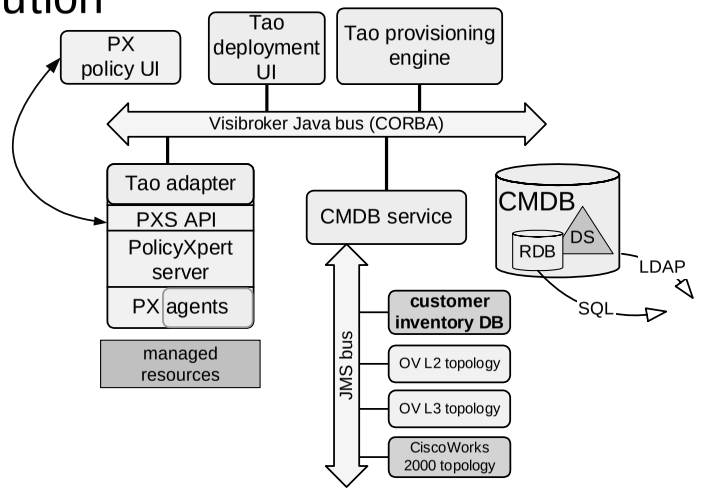
HP OpenView
Works | Right | Now

Contents

Evolution of Policy-based Management

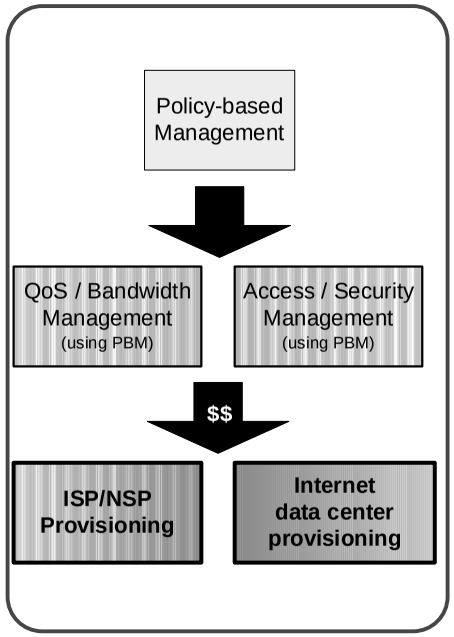
- **Architectural**
  - Canonical IETF policy management system has become a robust distributed management system
- **Commercial**
  - A wave of companies developed policy-based QoS or Security Managers -- Cisco, HP, IP Highway, Nortel, Novell, Orchestream, SolSoft, etc.
- **Behavioral**
  - Policy-based configuration and control is maturing into automated service provisioning, driven by the needs of service provide

Architectural Evolution



Behavorial Evolution

1. Policy-based management of QoS or Access Control
2. QoS and/or Security Management; using a PBM
3. Automated Provisioning, with policy and configuration management features



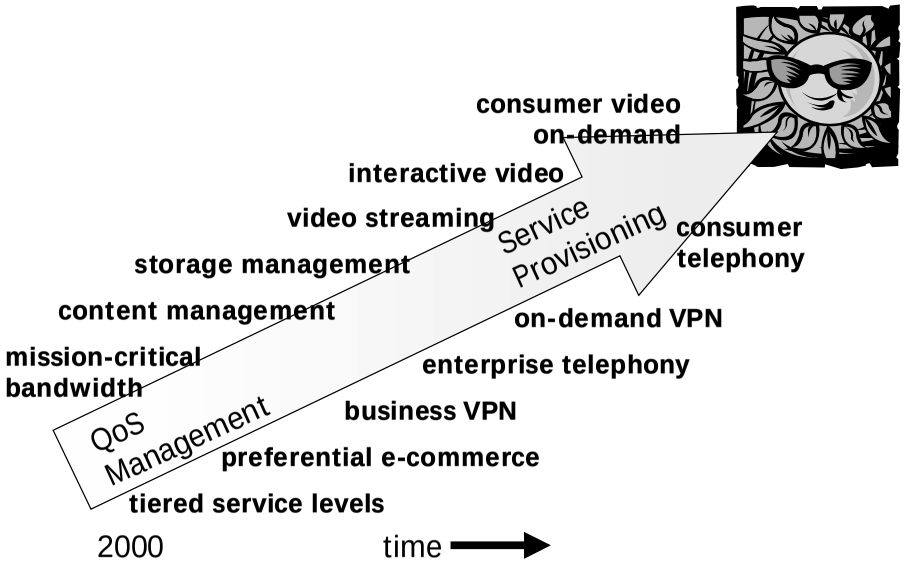
Commercial Evolution

- PolicyXpert 1.0 November 1999
  - Enterprise QoS manager controlling WAN bandwidth congestion
- PolicyXpert 2.0 November 2000
  - ISP/NSP QoS manager focused on DiffServ QoS management
- PolicyXpert 2.1 Q1 2001
  - PX 2.0 running on Solaris and HP-UX
- **Tao Q4 2001 and Q2 2002**
  - Central to HP OpenView provisioning solution
  - IDC provisioning (VLANs, firewalls/access, and QoS)
  - IDC provisioning (server ignition, SANs)
  - ISP/NSP provisioning (MPLS paths, VPN assignments, DiffServ QoS)
  - Configuration management database (CMDB)
  - OV topology and customer inventory data integration

The Provisioning Future

- Providers are everywhere
  - ASP, ISP, MSP, NSP, CSP, HSP, FSP, WISP, ...
- Expertise is scarce
  - IT drain compounded by waves of new technology
- E-business, outsourcing, and broadband accelerate demands
- **Configuring network, system, and application services in a large, multi-vendor environment requires automation and abstraction**

The Provisioning Future



Technologies

"The best way to predict the future is to invent it."
-- Alan Kay

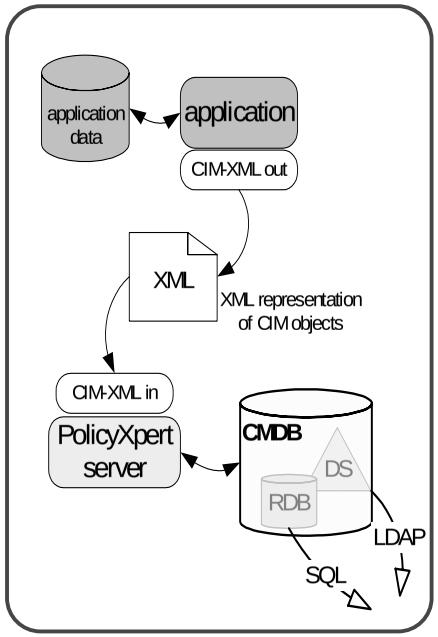
- Technical enablers for policy-based service provisioning are finally ready
- QoS, Traffic Engineering, and Security standards
- Data modeling, data exchange, and data access technologies and standards

Relevant Standards and Technologies

- **DiffServ** - Scalable QoS architecture being adopted by ISPs
- **MPLS** - Traffic-engineering and VPN path creation standard gaining rapid support
- **IPSec** - Authenticated and encrypted tunneling standards, plus key exchange standard, finally taking hold
- **COPS** - Policy enforcement protocol with provisioning enhancements that is simpler than other configuration methods
- **LDAP** - Directory access protocol that will be pervasively used in Internet services and management by end of 2001
- **XML** - Data exchange/integration language in pervasive use across all computing industries by the end of 2001
- **CIM** - INSM/PBM information model now mature enough to build systems upon, **DEN** enhancements nearly compete

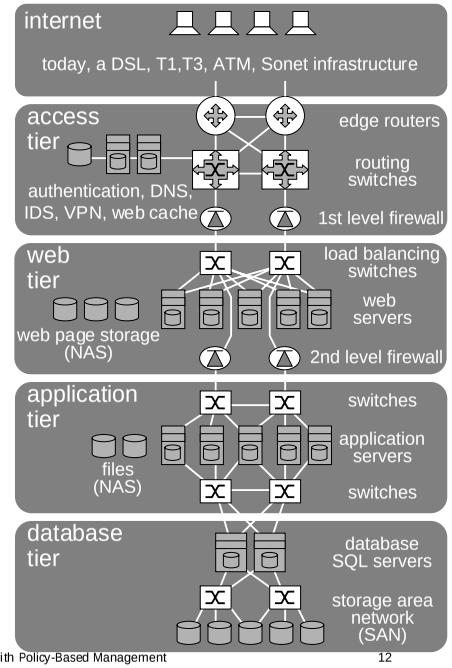
Amazing attraction of XML

- Unquestionably the next, big technology
  - Simple, extensible, flexible
- PolicyXpert experience
  - Research schema ➔ automated test suite ➔ import/export tool ➔ **...**
- Data integration is XML's real power
  - CIM plus XML will allow a PBM to unite data sources into a configuration management database (CMDB)
  - E.g. Novell's DirXML



The IDC

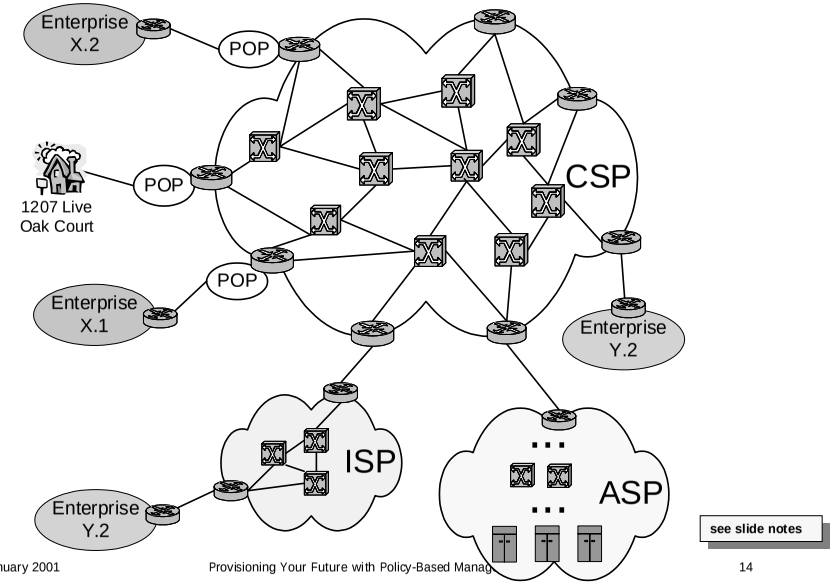
- a four tier layered architecture
- a service is provisioned by configuration tasks within the tiers
- a provider implements the tiers many times; one for each customer or application
- the provider wants rapid re-configuration of resources as customers and application demands change



Internet data center Perspectives on Provisioning

- Network, servers, and storage
- VLAN configuration
- VPN gateway configuration
- Unix and Windows server ignition
- SAN management
- L2 QoS, load balancing, NAT, etc
- **Secure**

The ISP/NSP (a.k.a. CSP)

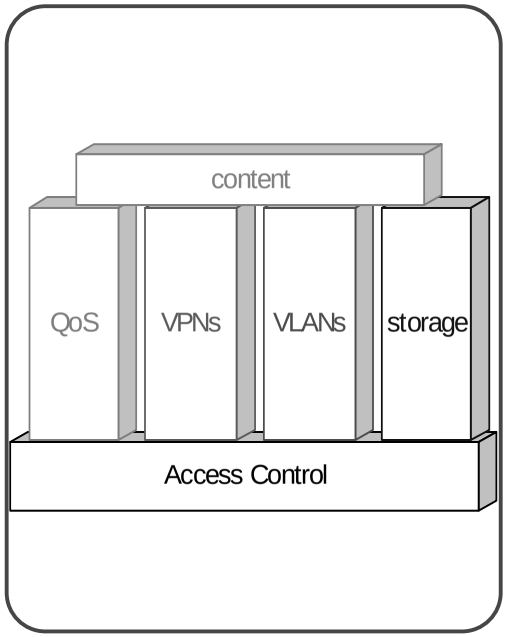


ISP/NSP Perspectives on Provisioning

- Aggregated Traffic
- Differentiated Classes of Service, e.g. VoIP
- MPLS path provisioning for traffic engineering and VPNs
- IPSec Secure tunnels
- Topology-aware, directory-enabled PBM
- Cisco++ (Cisco plus one or two select vendors)
- **Scalable**

Domains of Provisioning

- **Access Control**: allow or deny system, device, and network access
- **QoS**: aggregated CoS
- **VPNs**: configure MPLS paths for VPN or TE uses; and/or configure IPSec tunnels
- **VLANs**: configure VLAN assignments for partitioning and/or traffic shaping uses
- **Storage**: configure SAN and NAS
- **Content**: web infrastructure (caching, redirection, load-balancing)



Keys to a Compelling, Policy-based, Provisioning Solution

- CMDB
- Can integrate into the provider's "Big P" provisioning solution
  - APIs, CIM-XML exchange
- Integrates with assurance, and with usage
  - E.g. HP OpenView
- Two-phase commit, with rollback
- Scalable and secure
- Extensible
  - Conditional and unconditional policy rules

Summary

- Policy-based management is evolving to be a core component of service provisioning
  - Configuration management
- OV policy-driven provisioning solutions will integrate via XML, and a CMDB
  - Multi-vendor
  - Scalable
  - Extensible
  - OpenView integrated
  - "Big P integrate-able"

www.openview.hp.com/policyxpert

Negotiation of Access Control Policies

Virgil D. Gligor (*)
Electrical and Computer Engineering Department
University of Maryland
College Park, Maryland 20742
gligor@eng.umd.edu

January 29, 2001

(*) Work performed with **H. Khurana** and **R. Koleva** as part of a project on **Integrated Security Services for Dynamic Coalition Management** funded by DARPA under contract F30602-00-2-0510 monitored by AFRL, Rome Research Site, NY.

The views and conclusions presented herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or AFRL.

Negotiation in Secure Communication Protocols

- Notion of negotiation is *common* here

  Examples: ISAKMP, Oakley, SSL 3.0 protocols

  – what is negotiated ?
    - *(common) security associations*
      – common keying state; e.g., DH group parameters, shared key, identities, nonces
      – encryption modes, hashing, authentication algorithms
    - *common protocol modes*
      – aggressive (few messages with complex semantics)
      – conservative (many messages with simple semantics)
    - *common security services/features*
      – identity privacy (yes/no)
      – perfect forward secrecy for keys, identities

Policies, Policy Models, and Model Interpretations

Negotiation => a "common policy" representation

Example of Policy Specification

Attribute (AT) ***properties*** ∧

Access Management (AM) ***properties*** ∧

Access Authorization (AA) ***properties***

. . . . a conjunction of ***specified properties***

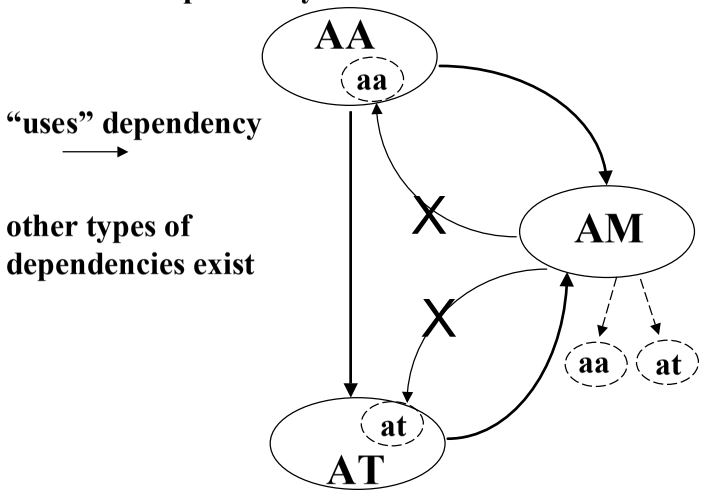
Example of of Properties We Specify / Negotiate

- Attributes and attribute (**AT**) properties
  – Security (integrity) levels, partial order, lattice property
  – Roles, permission-, membership-inheritance (DAG) properties

- Access management (**AM**) properties
  – Distribution, review, revocation of permissions
    - Selectivity, transitivity, independence ...
  – Object / subject, domain creation and destruction properties
  – Object encapsulation, domain protection properties

- Access authorization (**AA**) properties
  – Required subject and object attributes for access
    - viz., access rules of RBAC, UNIX, Win2K, etc.

Do we specify/ negotiate *Access Control* properties independently of each other ?



. . . no, *not all* properties can be specified and negotiated independently

Negotiation of Access Control Policies

What does this mean ?

Negotiation of *Common State* (coalitions, ad-hoc networks)
- assumption: same *interpretation of a common policy model*
- common state: state obtained by assignment of privileges for coalition objects/services to coalition users of foreign domains
- constraints on common state
  none, global constraints, local-domain constraints

Negotiation of *Policy Model* (interpretation)
- different AA, AT, AM property types;
- establishment of common properties => negotiation

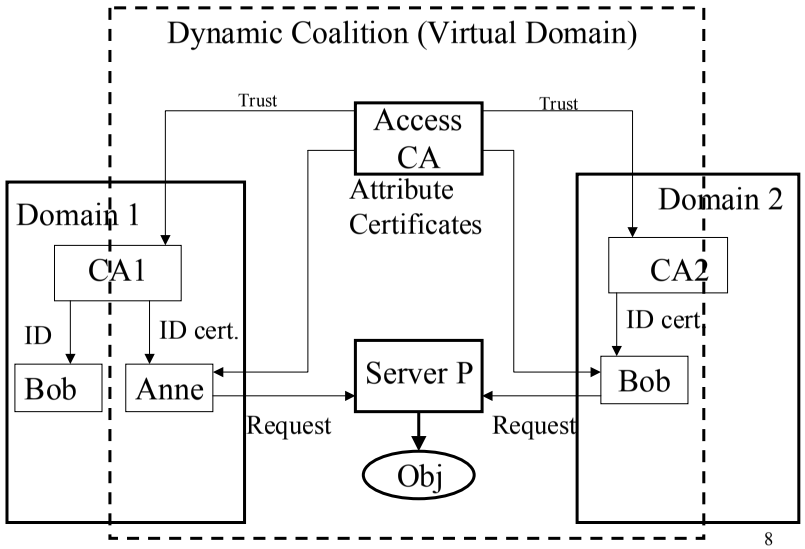
Negotiation of both Policy Model (interpretation) and Common System State

Negotiation of Common State:

Example 1 - No Constraints on Common State

- **Coalition Goal**: Need based (unconstrained) assignment of privileges for coalition objects to users of foreign domains
  - assigning membership of local roles to foreign users as required
  - examples: Shands [NDSS2000] and Herzberg et al. [IEEE Sec. & Priv 2000]
- **Negotiation:** who gets access to what objects/services to reach a common coalition state
- **Constrains** on negotiated state: *none*

Access Control Server of a Dynamic Coalition



Negotiation of Common State:
Example 2 - Global Constraints

- **Coalition *Goal***: Negotiate 6 supply routes between 3 domains, where each domain controls a subset of the 6 routes.

  Route *sharing:* an user (administrator) of a foreign domain may *execute route applications* (e.g., communication system).

- **Negotiation**: assignment of coalition-object privileges to users as required by: application execution (compatibility); administrability; etc.

- **Before Negotiation**:

  Domain **D1** controls routes {1, 4, 5 and 6},
  Domain **D2** controls routes {1, 2, 3, and 4}, and
  Domain **D3** controls routes {1, 2, 4, and 5}
  Some routes may be shared already; e.g., routes 1, 2, 4, 5

Negotiation of Common State:
Example 2 (ctnd.) - Global Constraints

- **Global constraints**
  - A domain must *share* any unique route
  - For *common routes* choose the one that follows *least privilege principle*; e.g., access to smallest number of application objects; weakest privileges):
    - Domain **D1**: *rt. 1* -> 5 objects, *rt. 4* -> 8 objects, *rt. 5* -> 4 objects
    - Domain **D2**: *rt. 1* -> 5 objects, *rt. 2* -> 6 objects, *rt. 4* -> 9 objects
    - Domain **D3**: *rt. 1* -> 7 objects, *rt. 2* -> 4 objects, *rt. 4* -> 6 objects
      rt. 5 -> 3 objects

- **After Negotiation**:

  Domain **D1** shares routes {1, 6},
  Domain **D2** shares routes {3},
  Domain **D3** shares routes {2, 4, and 5}.

- **Other global constraints**: *Separation of Duty* [IEEE S&P 1998]

Negotiation of Common State:
Example 3 - Local Constraints

- **Coalition Goal**: Negotiate sharing of *private routes* among domains
- **Before Negotiation**:

  Domain **D1** controls routes {1, 2, and 3},
  Domain **D2** controls routes {4, 5, 6, 7 and 8}, and
  Domain **D3** controls routes {9, 10, 11, and 12}

- **Local constraints:** some local constraints *cannot be (apriori) revealed*
  - Domain **D1**: willing to share half as many routes as other domains because of some (e.g., geographic) advantage (can reveal this)
  - Domain **D2**: willing to share twice as many routes as **D1;** unwilling to share route 6 (cannot reveal this constraint)
  - Domain **D3**: willing to share twice as many routes as **D1**; needs one of the routes {2, 5 or 6} and is unwilling to share route 11 (cannot reveal this)

Negotiation of Common State:
Example 3 (ctnd.) - Local Constraints

- **Negotiation:**
  - Domains construct *preference lists* (under local constraints), propose corresponding *common state* and *wait for responses*
  - Agreement on *common state* => state is *committed*; else lower-preference states are proposed, and negotiation continues;
  - Application of game theory
- **Examples (negotiations that terminate):**
  - **D1** proposes route-sharing state { **D1**: 1, **D2**: (4, 5), **D3**: (9, 10)}
    - all domains agree and common state is committed
  - **D3** proposes rout-sharing state: { **D1**: 2, **D2**: (5, 6), **D3**: (9, 10) }
    - **D2** rejects it (unwilling to share 6), and instead
    - **D2** proposes: { **D1**: 2, **D2**: (4, 5), **D3**: (10, 11) }
    - **D3** rejects it (unwilling to share 11) and instead
    - **D3** proposes: { **D1**: 2, **D2**: (4, 5), **D3**: (9, 10) }
    - all domains agree and common state is committed

Negotiation of Policy Models

- **Example 1**: ***No negotiation***
  Same **AA**, **AT** and **AM** properties - both domains use RBAC as in Shands [NDSS 2000]
  - must negotiate membership to roles with/without constraints

- **Example 2**: ***Negotiation of different AM properties***
  Domains have *different* "rules" for assigning membership to roles in an RBAC system as in Herzberg [IEEE Sec. & Priv. 2000]

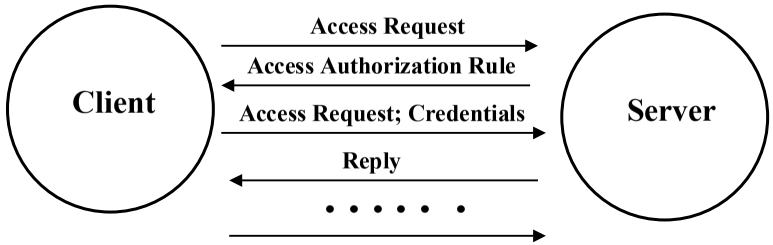
- **Negotiation**: common role membership "rules" with/without constraints

Negotiation of Policy Models

- **Example 3**: ***Negotiation of different AA properties***
  Domains discover the **AA** "rules" of foreign domains (as in "trust negotiations" of Seamons, Winslett and Yu [NDSS 2001])

- **Negotiation**: server domains reveal **AA** "rules" and client domains reveal *credentials* (with/without) constraints

- **Local constraints:** some local constraints *cannot be (apriori) revealed*
  Server: unwilling to disclose all **AA** rules (e.g., to keep privileged association private)
  Client: unwilling to disclose all *credentials* (e.g., to keep privileged association private; to protect sensitive private information such as bank account, social-security number)



Local Constraints: partial disclosure; gradual disclosure of both "rules" and credentials

Negotiation of Policy Models

- **Example 4**: ***Negotiation of different AM properties***
  A domain supports *selective revocation* of attribute certificates (rev. of identity certificate => rev. of dependent attribute certificate) while the another does not

- **Negotiation**: Three options and their dependencies; each option may cause a change in **AA** properties (authentication checks)
  - (i) both domains support *selective revocation* - need not verify identity certificate on access request
  - (ii) both do *not* support *selective revocation* - need to verify identity certificate on access request
  - (iii) both retain their individual **AM** properties - servers may need to distinguish requests from the domains and use corresponding authorization policies

- **Further Negotiation**: assignment of privileges to users with/without constraints

Progress

Selected an Access Control Policy, language and server:

- **Role Based Access Control (RBAC) Policy**
- **Policy Language is implemented via GUI**
- **Policy Server enforces RBAC within a Domain**

Will add negotiation components to Policy Language and GUI

Will demonstrate policy negotiation and enforcement