



the globus alliance  
www.globus.org

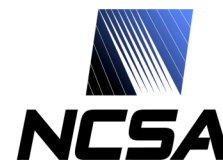
# The Globus Authorization Processing Framework

## New Challenges for Access Control Workshop

April 27, 2005, Ottawa, Canada

Frank Siebenlist (Argonne National Laboratory), Takuya Mori (NEC),  
Rachana Ananthakrishnan (ANL), Liang Fang (Indiana Uni.),  
Tim Freeman (UofChicago), Kate Keahey (ANL), Sam Meder (ANL),  
Olle Mulmo (KTH), Thomas Sandholm (KTH)

[franks@mcs.anl.gov](mailto:franks@mcs.anl.gov) - <http://www.globus.org/>





# Outline

- The Globus Toolkit (GT)
- (Grid) Use Cases
  - ◆ Virtual Orgs (VOs), multiple admin realms, delegation
- Policy, Policy, Policy....
  - ◆ Attributes
    - Shibboleth, SAML, X509-ACs, VOMS, etc.
  - ◆ Authorization
    - Call-out, SAML Authz, XACML, PC, PERMIS, AAA-tk, Delegation...
- Authorization Processing Framework
  - ◆ Attribute collection, generic PDP-abstraction, Master-PDP, Delegation/Rights-Admin
- Big Picture & Futures
  - ◆ Proto-type=>real-thing, XACML-3, job/agreement-language integration



# Globus Toolkit (GT-4.0)

- WS, WS-I & WSRF compliant toolkit
- WSS, WS-I, X509/(GGF-)SAML Identity/Attribute Certificates, X509 Proxy-Certificate, XACML, PERMIS, VOMS compliant toolkit
  - ◆ Message Level Security & TLS support
- Different platform support
  - ◆ Java, C/C++, Python, .Net/C#
- (Security-)Integrated with higher-level Svcs
  - ◆ GridFTP, GRAM, MDS, MyProxy, PURSE, OGSA-DAI...
- Many, many parties involved
  - ◆ Customer-requirements driven
  - ◆ ... with commercial “versions”...
- Open Source
  - ◆ Apache-style license



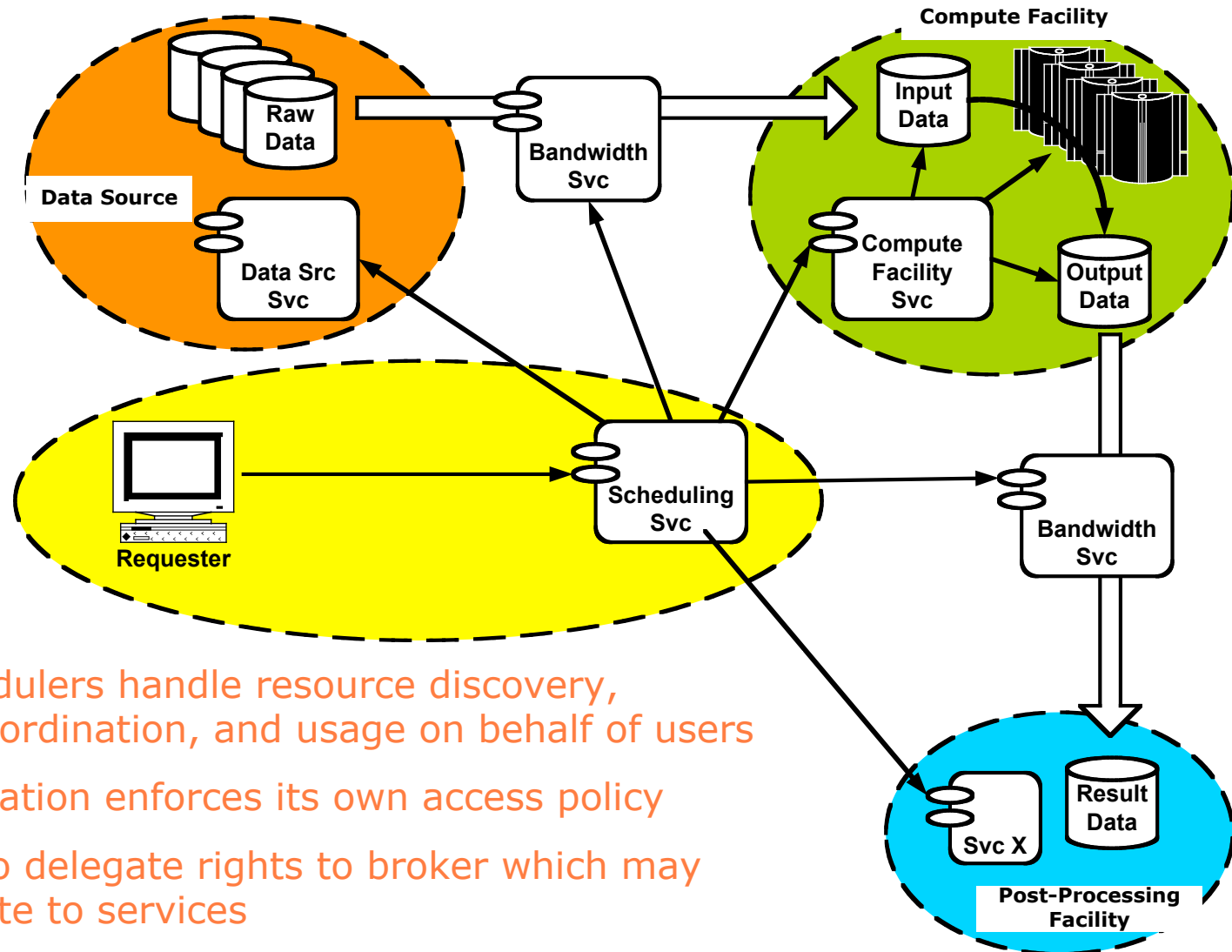
the globus alliance

www.globus.org

## Leverage (Open Source) Security Service Implementations

- **OpenSSL**
  - ◆ “native” Proxy Certificate support coming...  
(thanks to OpenSSL hacker Richard Levitte and KTH!)
- **Internet2’s OpenSAML**
  - ◆ Part of GT - used by CAS/GridShib/AuthzCallout/...
- **Internet2’s Shibboleth**
  - ◆ NSF funded GridShib project to “Grid-enable” Shibboleth
- **Sun’s open source XACML effort**
  - ◆ Integrate sophisticated policy decision engine in the GT
- **Futures: Permis, Handle System, XKMS, XrML, ...**

# Security of Grid Brokering Services



- Brokers/Schedulers handle resource discovery, reservation, coordination, and usage on behalf of users
- Each Organization enforces its own access policy
- User needs to delegate rights to broker which may need to delegate to services
- QoS/QoP Negotiation and multi-level delegation



the globus alliance

www.globus.org

# Security Services Objectives

- It's all about "Policy"
  - ◆ (Virtual) Organization's Security Policy
  - ◆ Security Services facilitate the enforcement
- Security Policy to facilitate "Business Objectives"
  - ◆ Related to higher level "agreement"
- Security Policy often delicate balance
  - ◆ More security ⇔ Higher costs
  - ◆ Less security ⇔ Higher exposure to loss
  - ◆ Risk versus Rewards
  - ◆ Legislation sometimes mandates minimum security



# Agreement ⇔ VO Security Policy

## (Business) Agreement

Price  
Cost  
Obligations  
QoS  
T&Cs  
.....  
Security  
.....

## Static Initial VO Security Policy

trust anchors  
(initial) members  
(initial) resources  
(initial) roles

Access rules  
Privacy rules

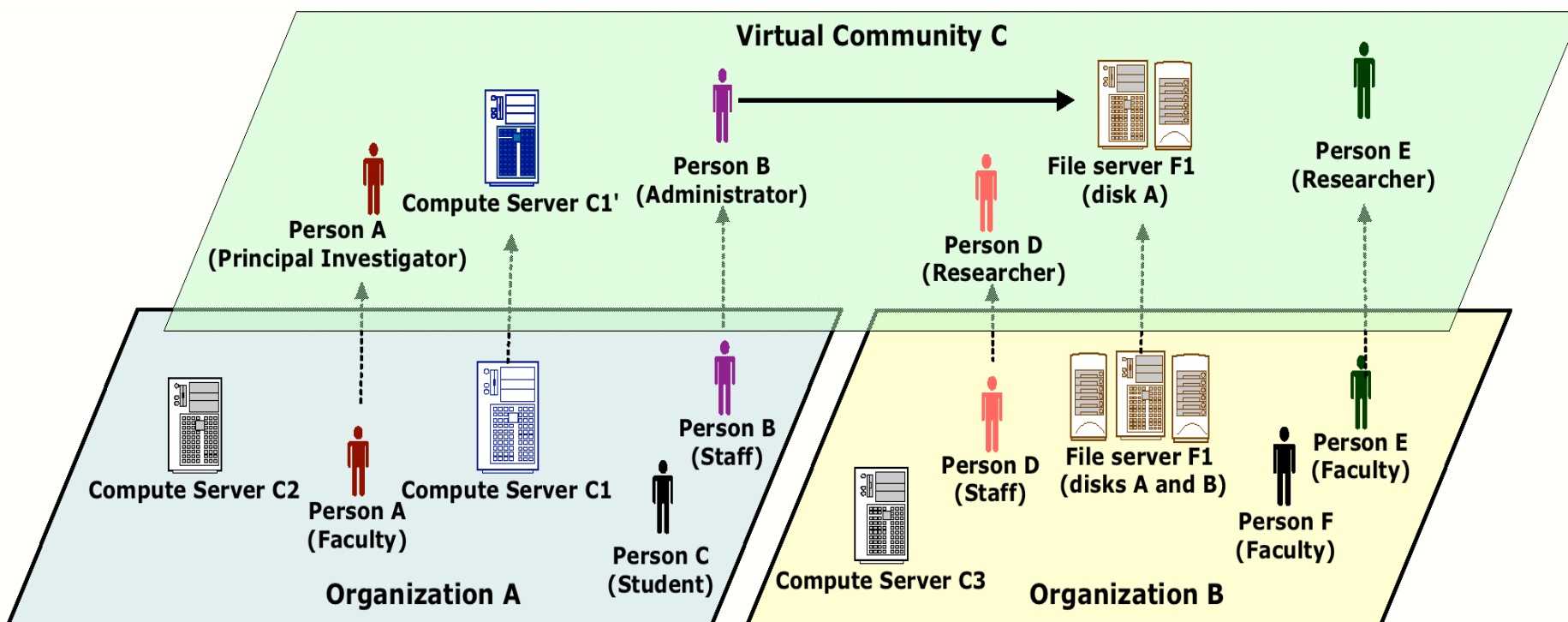
## Dynamic VO Security Policy

members  
resources  
roles

Attribute mgmt  
Authz mgmt



# Virtual Organization Concept



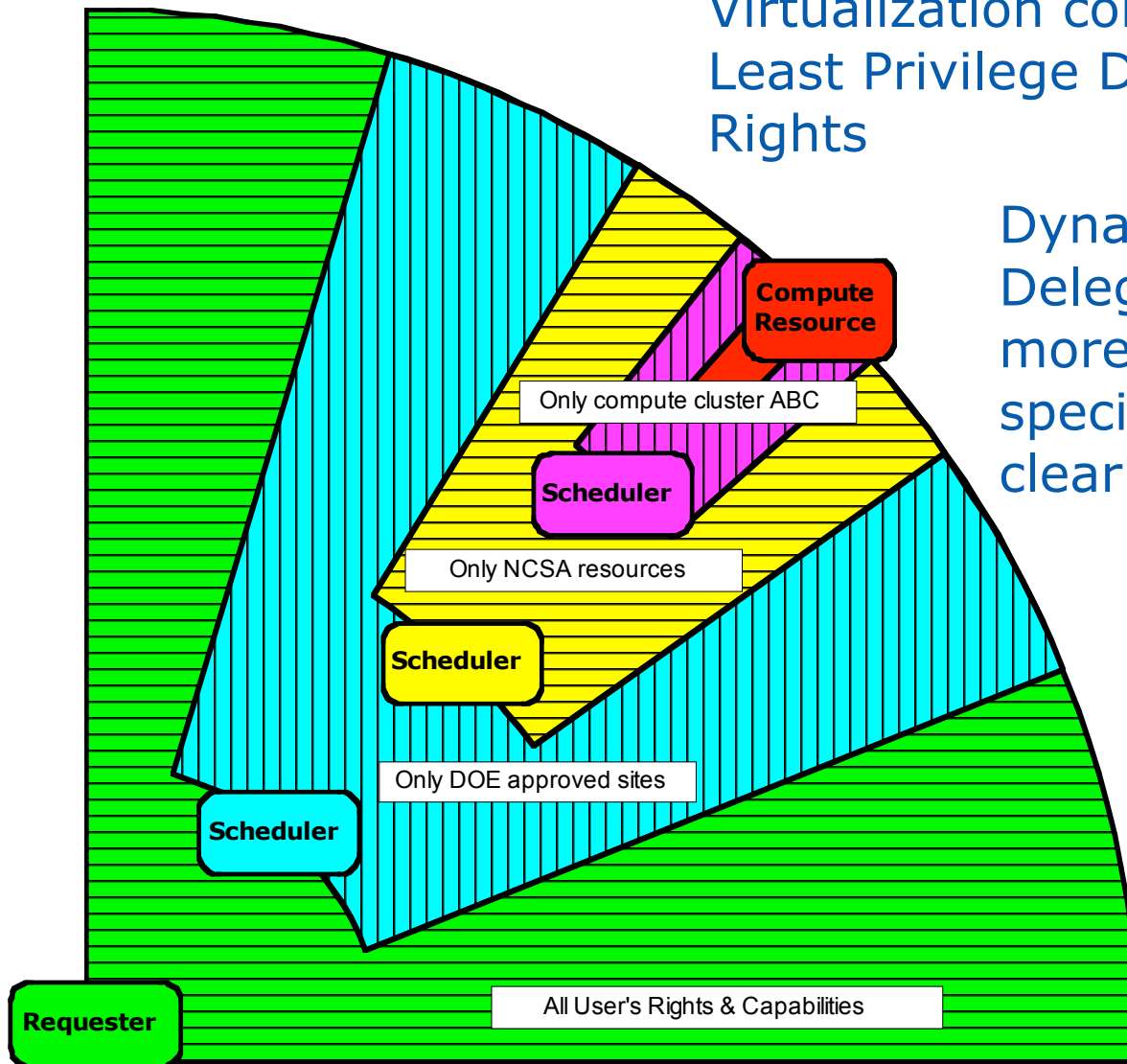


# Propagation of Requester's Rights through Job Scheduling and Submission Process

Virtualization complicates Least Privilege Delegation of Rights

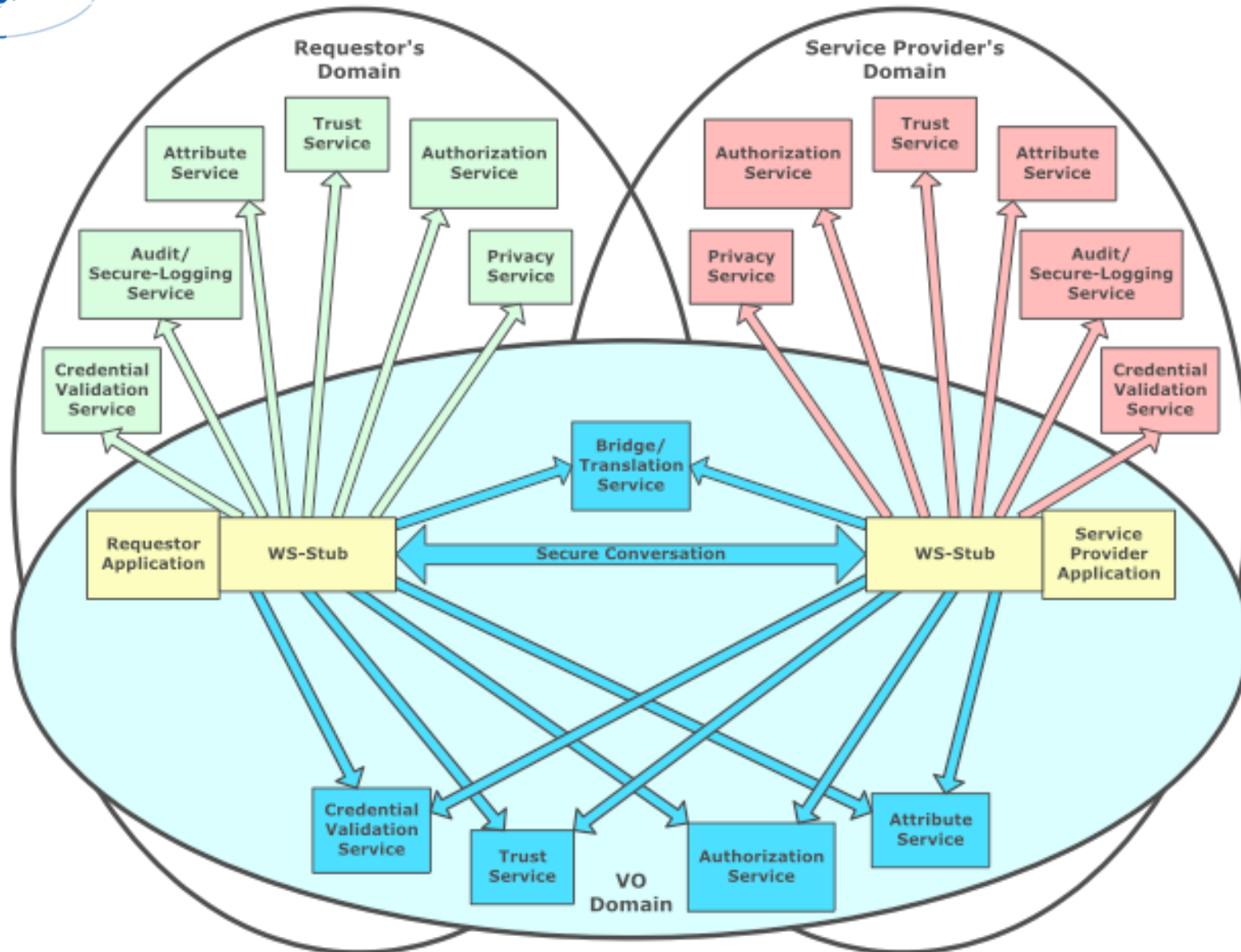
Dynamically limit the Delegated Rights more as Job specifics become clear

Trust parties downstream to limit rights for you... or let them come back with job specifics such that you can limit them





# Security Services with VO





## GT's Attribute Assertion Support

- VOMS/Permis/X509/Shibboleth/SAML identity/attribute assertions
- Assertions can be pushed by client, pulled from a service, or are made locally available

**GT-runtime has to mix and match all Attribute information a consistent manner, and present it to the subsequent Authz stage...**



## GT's GGF's Authorization Call-Out Support

- GGF's OGSA-Authz WG:  
"Use of SAML for OGSA Authorization"
  - ◆ Authorization service specification
  - ◆ Extends SAML spec for use in WS-Grid
  - ◆ Recently standardized by GGF
- Conformant call-out integrated in GT
  - ◆ Transparently called through configuration
- Permis interoperability
  - ◆ Ready for GT4!
- Futures...
  - ◆ SAML2.0 compliance ... XACML2.0-SAML2.0 profile



the globus alliance

[www.globus.org](http://www.globus.org)

# GT-XACML Integration

- eXtensible Access Control Markup Language (XACML)
  - ◆ OASIS standard
  - ◆ Open source implementations
- XACML: sophisticated policy language
- Globus Toolkit will ship with XACML runtime
  - ◆ Integrated in every client and server build on GT
  - ◆ Turned-on through configuration
- ...and we're using the XACML-"model" for our Authz Processing Framework...
- ...can be called transparently from runtime and/or explicitly from application...



## GT's Assertion Processing "Problem"

- VOMS/Permis/X509/Shibboleth/SAML/Kerberos identity/attribute assertions
- XACML/SAML/CAS/XCAP/Permis/ProxyCert authorization assertions
- Assertions can be pushed by client, pulled from service, or locally available
- Policy decision engines can be local and/or remote
- Delegation of Rights is required "feature" implemented through many different means

**GT-runtime has to mix and match all policy information and decisions in a consistent manner...**



# Basic Access Control Policy

Bob's policy:

Alice is my friend and I'll share my lemonade with her  
Mallory is not my friend and he can go #\$\$%^&



Alice



Ivan

Can I have glass of lemonade?

Sure, here is a glass

Can I have glass of lemonade?

No way, I don't like you



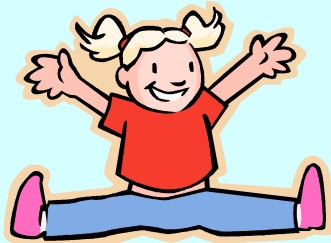
Mallory



# Basic Access Control Policy (2)

Bob's policy:

Alice is my friend and I'll share my lemonade with her  
Mallory is not my friend and he can go #\$\$%^&

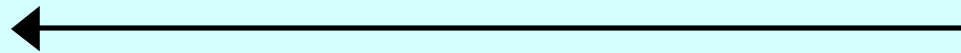


Alice

Can I have glass of lemonade?



Sure, here is a glass



Ivan

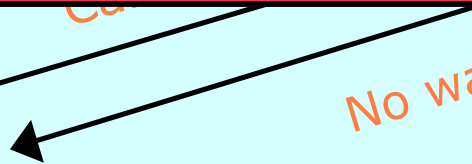
**Resource Owner decides!**

(ultimate source of authority for access)



Mallory

No way, I don't





# Delegation of Rights (1)

Ivan's policy:  
Carol is my friend and I'll share my lemonade with her  
I'll share my lemonade with any friend of Carol  
I don't know any Bob...(?)

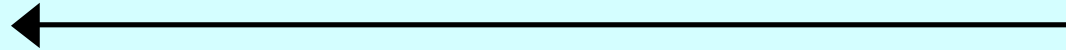


**Bob**

Can I have glass of lemonade?

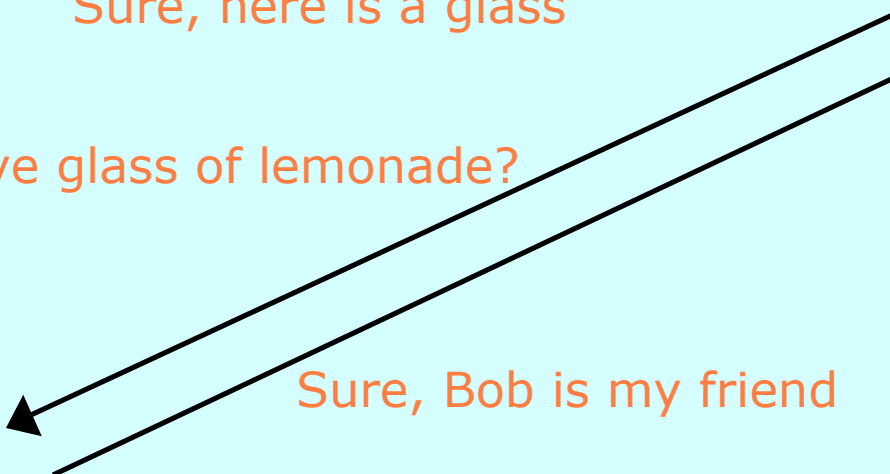


Sure, here is a glass



**Ivan**

Can Bob have glass of lemonade?



Sure, Bob is my friend



**Carol**

Carol's policy:  
Bob is my friend and I'll share my lemonade with him

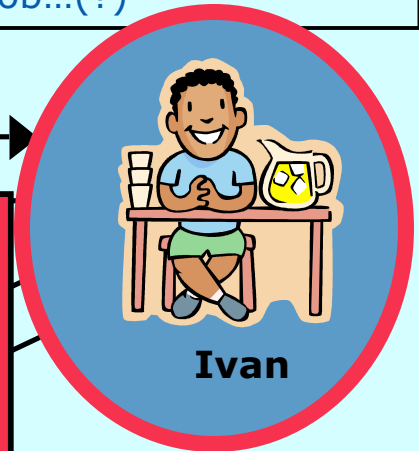


# Delegation of Rights (2)

Ivan's policy:  
Carol is my friend and I'll share my lemonade with her  
I'll share my lemonade with any friend of Carol  
I don't know any Bob...(?)



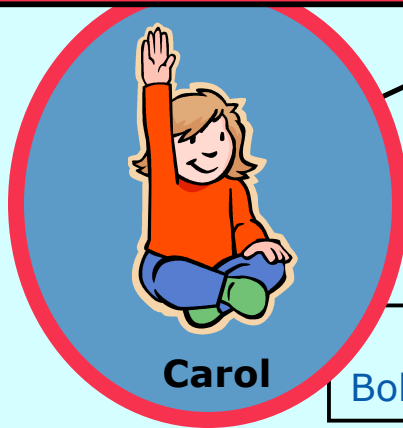
Can I have glass of lemonade?



Ivan

**Ivan likes Carol + Carol likes Bob  
=> Ivan likes Bob**

(non-normative delegation logic ;-)



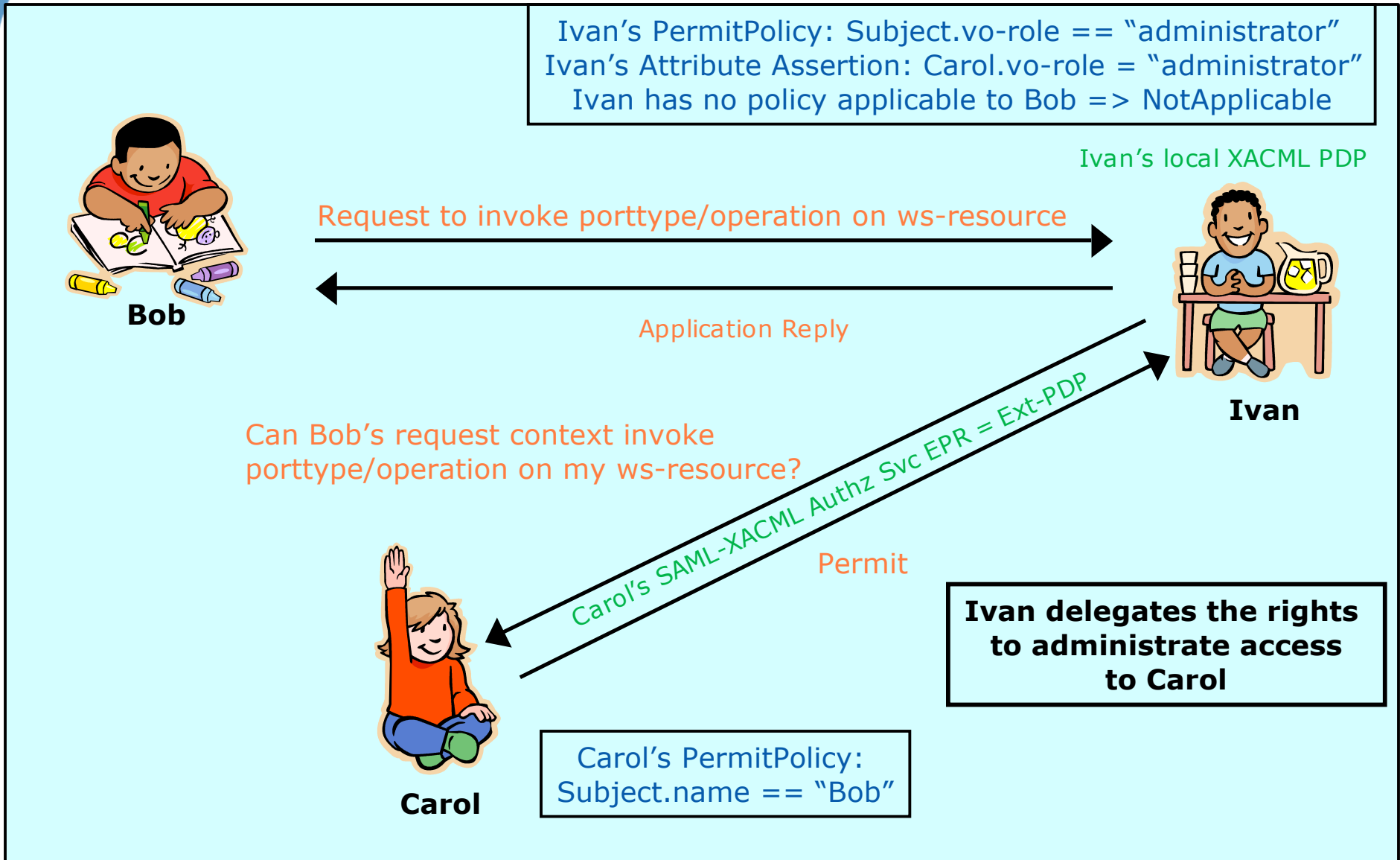
Carol

Sure, Bob is my friend

Carol's policy:  
Bob is my friend and I'll share my lemonade with him



# Delegation of Rights (3)





# Authz Processing Assumptions (1)

- All Policy Statements, PDPs and Authz-Decisions have Issuer associated with them
  - ◆ “someone” has to take responsibility for statements and associated decisions
- Resource Owner is the Ultimate Authority
  - ◆ Any statement/decision that can not be directly traced back to the owner is NotApplicable
    - “traced back”: delegation chain that starts with owner
- Two different Policy Statements and Queries
  - ◆ Admin Policy Statements
    - Issuer states that certain admin-subject are allowed to administer the rights of certain access-subjects to invoke certain operations on certain resources.
  - ◆ Access Policy Statements
    - Issuer states that certain access-subject are allowed to invoke certain operations on certain resources.

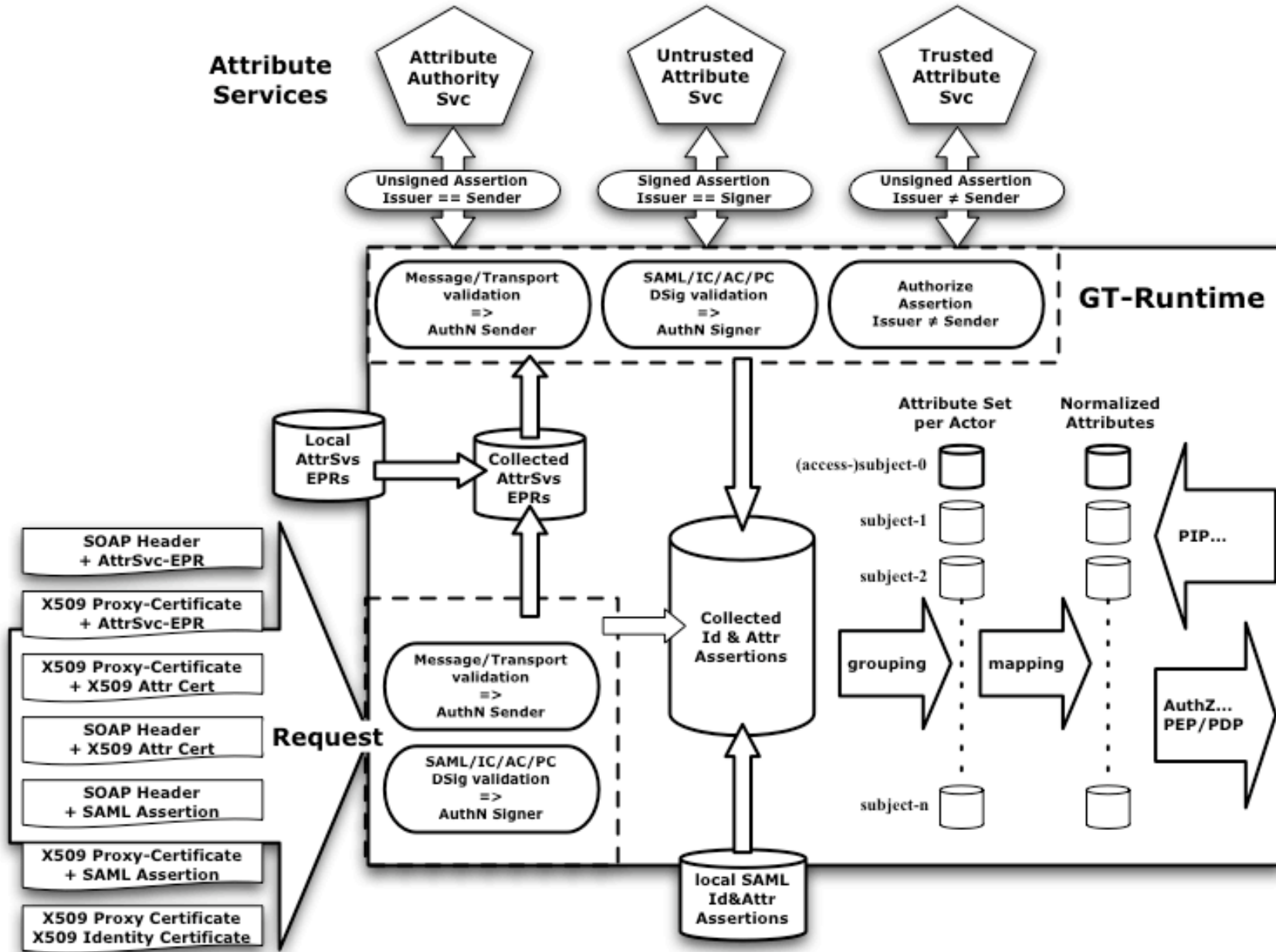


## Authz Processing Assumptions (2)

- “Push-Pull” Equivalence
  - ◆ Pushing authz-assertion and evaluating it locally renders same decision as evaluating the same policy statements remotely behind an external PDP
- Authz-Decisions are Policy Statements
  - ◆ Folded over the request context
  - ◆ Could optimize by only considering the attributes used to render a decision...
  - ◆ If attributes don’t specify a “invocation context”, then only the invoker’s identity would suffice...
  - ◆ Conservative: mandate that all request context’s attributes values are equal to the ones that rendered the decision.



# Attribute Collection Framework





## GT's Authorization Processing Model (1)

- Use of a Policy Decision Point (PDP) abstraction that conceptually resembles the one defined for XACML.
  - ◆ Normalized request context and decision format
  - ◆ Modeled PDP as black box authorization decision oracle
- After validation, map all attribute assertions to XACML Request Context Attribute format
- Create mechanism-specific PDP instances for each authorization assertion and call-out service
- The end result is a set of PDP instances where the different mechanisms are abstracted behind the common PDP interface.

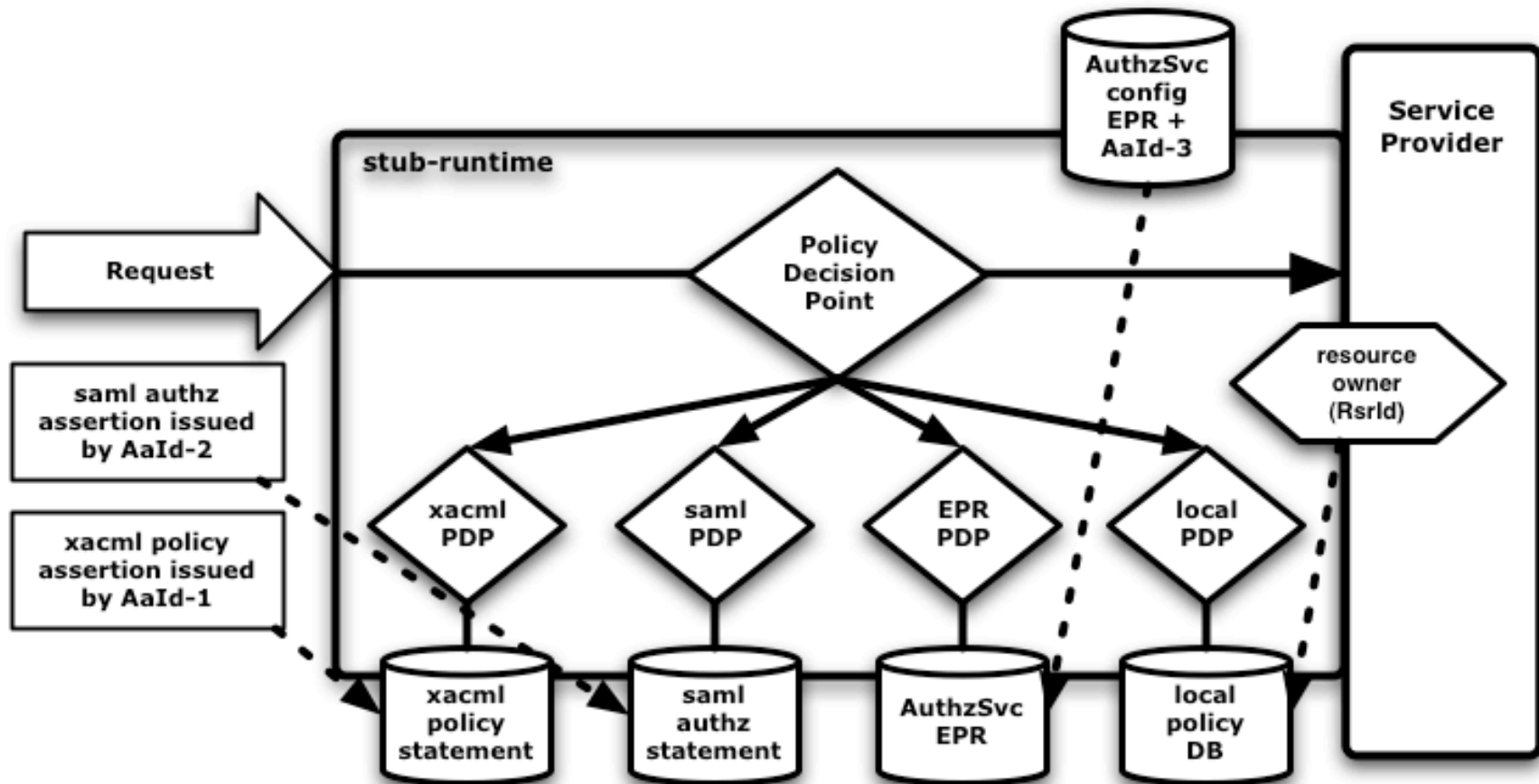


## GT's Authorization Processing Model (2)

- The Master-PDP orchestrates the querying of each applicable PDP instance for authorization decisions.
- Pre-defined combination rules determine how the different results from the PDP instances are to be combined to yield a single decision.
- The Master-PDP is to find delegation decision chains by asking the individual PDP instances whether the issuer has delegated administrative rights to other subjects.
- the Master-PDP can determine authorization decisions based on delegated rights without explicit support from the native policy language evaluators.

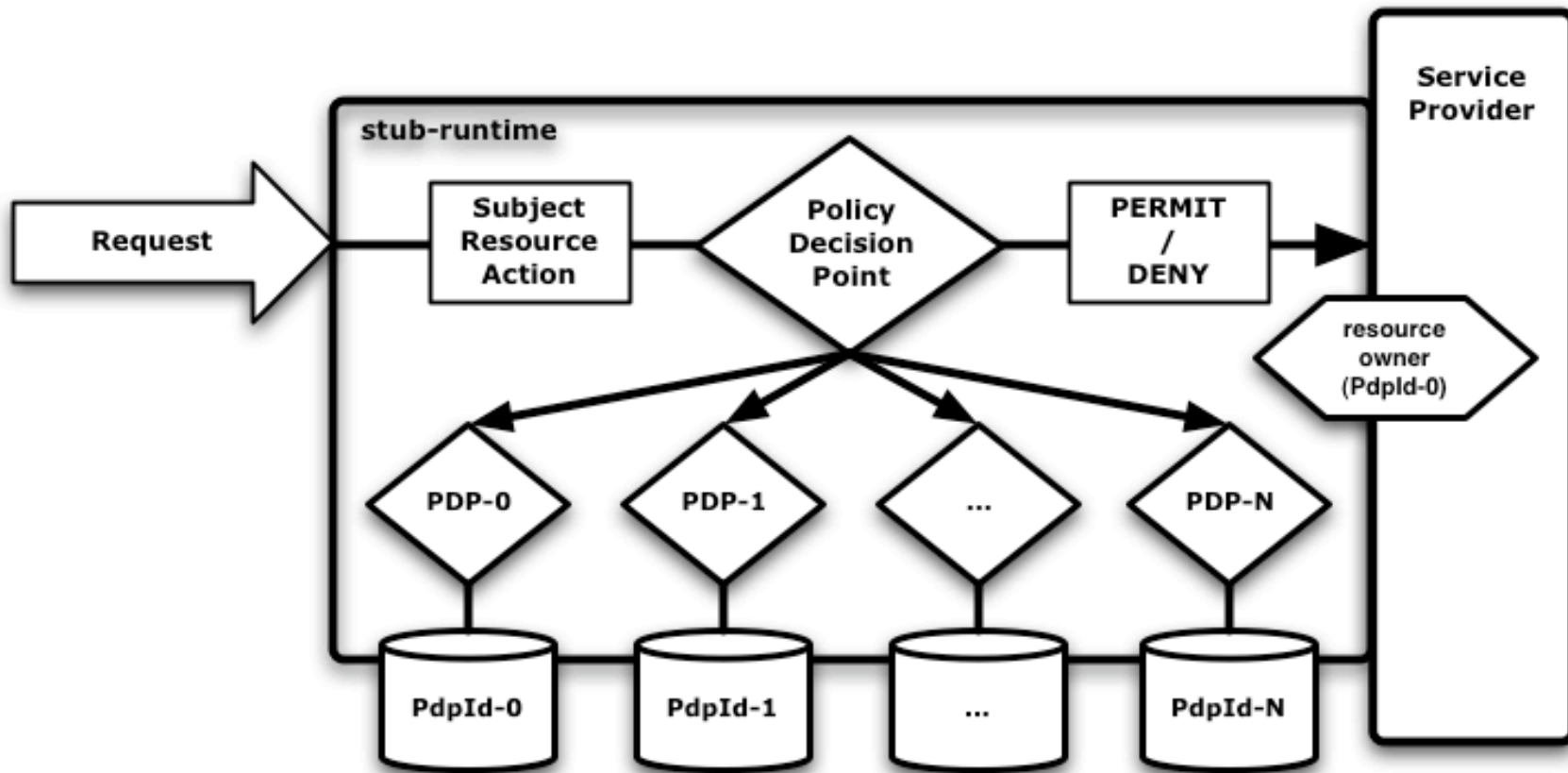


# GT Authorization Framework (1)





# GT Authorization Framework (2)





## GT Authorization Framework (3)

- Master-PDP accessed all mechanism-specific PDPs through same Authz Query Interface
  - ◆ SAML-XACML-2 profile
- Master PDP acts like XACML “Combinator”
  - ◆ “Permit-Overrides” rules
    - Negative permissions are evil...
- Delegation-chains found through exhaustive search
  - ◆ ...with optimization to evaluate cheap decisions first...
- “Blacklist-PDPs” are consulted separately
  - ◆ Statically configured, call-out only PDPs
  - ◆ Deny-Overrides only for the blacklist-PDPs...
    - Pragmatic compromise to keep admin simple



# GT-Authz Summary & Futures

- Generic Authz Processing Framework
  - ◆ Mix, match and combine different authz mechanism
  - ◆ Supports delegation as “side-effect”
- Proto-type => GT-4.2 integration
  - ◆ Both Attribute Collection & Authz Processing
  - ◆ Java, Python, C/C++ (,.Net) ... WS & GridFTP & httpd
- XACML-3 (?)
  - ◆ May be able to incorporate “all” our processing requirements
- Focus on higher-level Policy Integration
  - ◆ (Security) Policy Negotiation/Publishing/Discovery
  - ◆ Job Execution & Agreement Language Integration (?Semantic Web?)
  - ◆ Infrastructure Svc Integration to enable the “5-min VO”
  - ◆ ... stay requirement driven - listen to our “customers” ...