Distributed Data Access Control Mechanisms and the SRM

Peter Kunszt

Manager Swiss Grid Initiative
Swiss National Supercomputing Centre CSCS
GGF Grid Data Management Working Group Co-Chair

HPDC Workshop on Management of Rights in Production Grids
Paris, 19 June, 2006

Building the Science Database of the Sloan Digital Sky Survey,
Johns Hopkins University Baltimore

EU Grid Projects, leading data management middleware development
CERN, Geneva

Grid Storage Management Working Group Co-Chair
Global Grid Forum

Manager Swiss Grid Initiative,
Swiss National Supercomputing Centre
CSCS, Manno

Outline

Introduction: Distributing Data Security Aspects

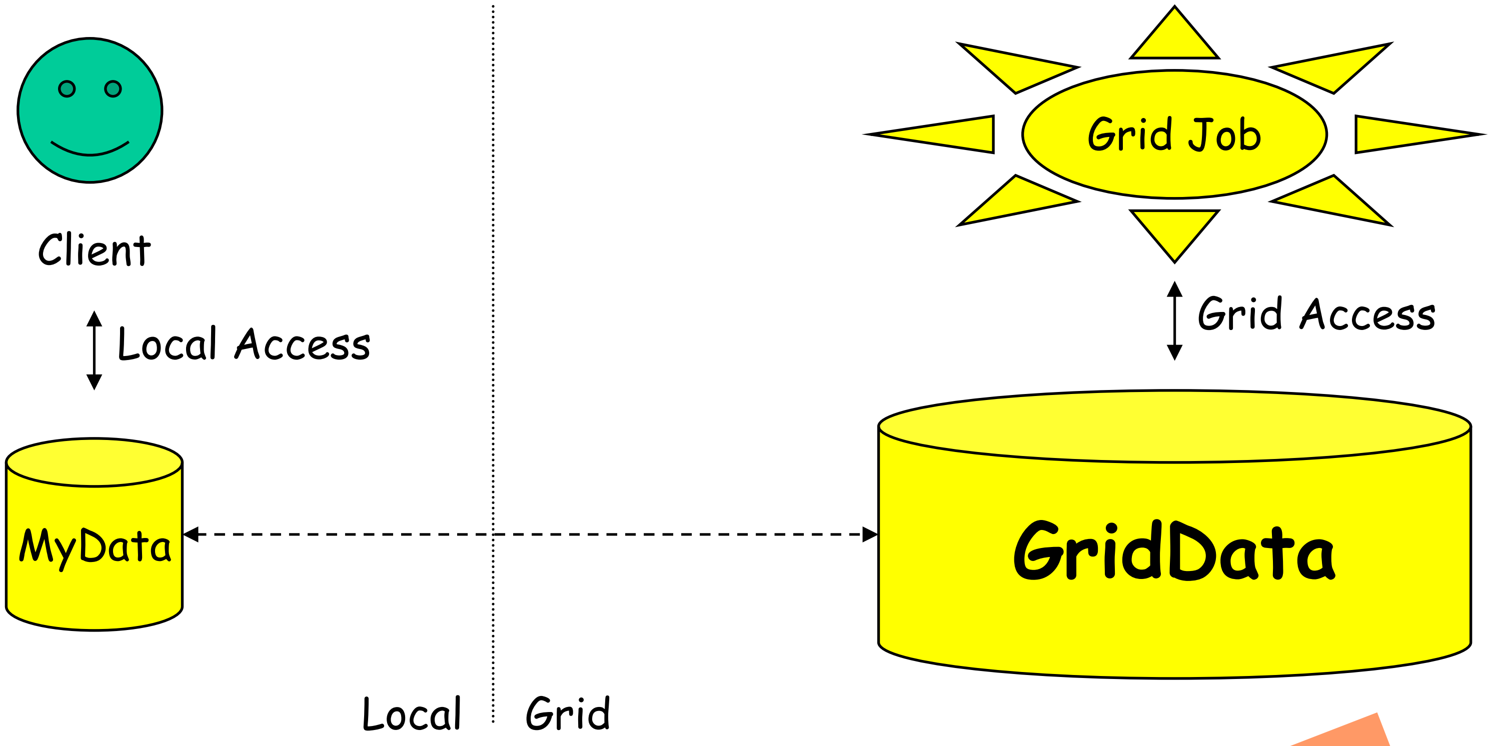
- Ubiquitous Access to Data(?)
- Semantic models
- Trust relationships

Discussion of Possible Security Models

EGEE and SRM models

Ubiquitous Distributed Data Security

Use Case : Ubiquitous Data Access



Same Semantics Expected

Or Not??

Data Access Semantics

Local: Security fully controlled by the Owner

- Setting and Getting of Permissions and ACLs
- Changes have Instantaneous effect

Distributed: Different Security Models

- **Single Master, Read-Only Copies**
  - Changes only possible in one place, effect on copies not instantaneous
- **Multi Master**
  - Changes possible everywhere, complex synchronization
  - Race Conditions
  - Resolution of Conflicts may involve human decisions
- **No Synchronization** (Peer-to-Peer)
- **Any combination of the three above**
  - Hierarchical models
  - Caches

Trust Relationships

Local : Client and Resource interact directly

- Client trusting the local Storage to enforce the access model
- Client trusting the local Resource Owners not to abuse data
- Resource Owners trusting Clients not to put ‘bad’ data on resource

Grid : VO trust layer between Client and Resource

- VO trusting Resource to enforce access
- VO trusting Resource Owners not to abuse data
- Resource trusting VO not to place ‘bad’ data into resource
- Client trusting VO to maintain the trust relationships properly on its behalf

Distributed Data Security Models

Policy Decision Point PDP

- Decisions about Clients being able to access Data

Policy Enforcement Point PEP

- Enforcing the PDP decision, strong trust relationship between PEP and PDP
- Enforcement can only be done by the Recource Owners

Models different depending on the placement of PDP and PEP in the Grid Layered Architecture

- 5 Models possible

Policy Enforcement and Decision Points

			PDP	PDP	Service Owner:**VO** Application Layer
	PDP	PDP PEP		PEP	Service Owner:**Site** Middleware Layer
PDP PEP	PEP		PEP		Service Owner:**Site** Resource Layer
Model 1	Model 2	Model 3	Model 4	Model 5	

Model 1: Site Security Only

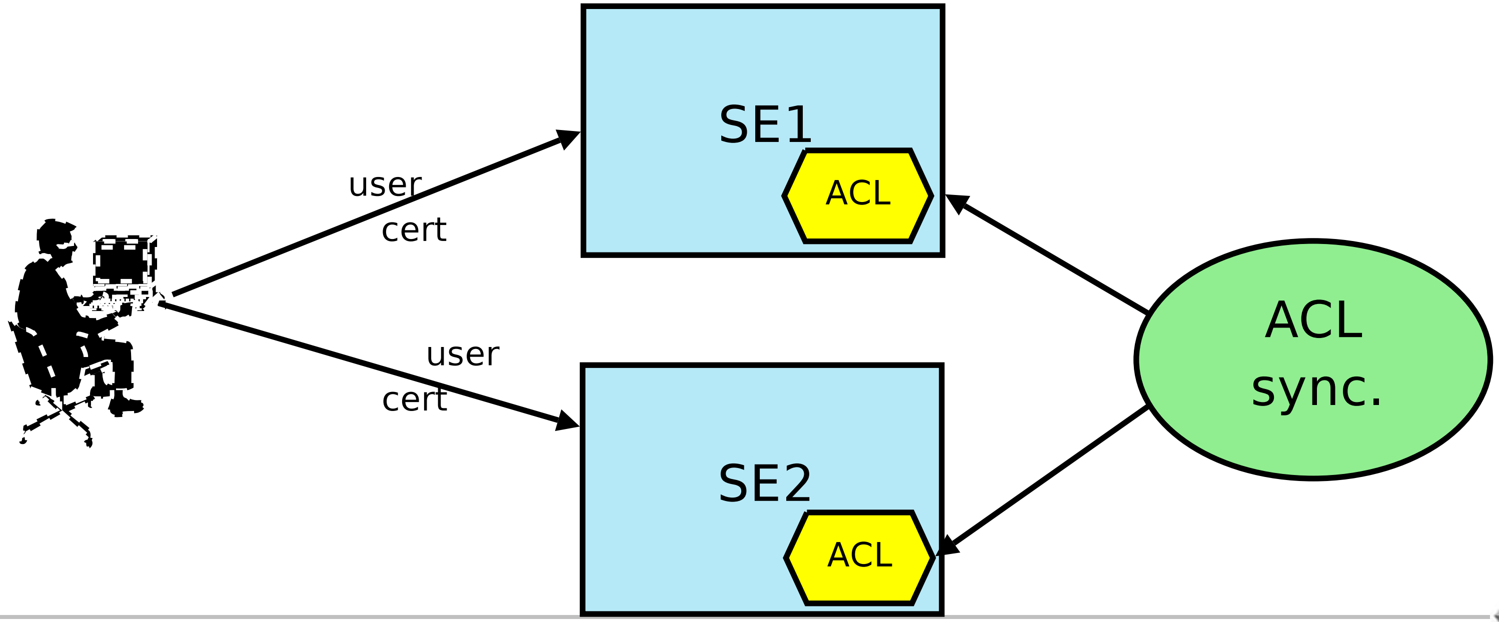
Both PDP and PEP are in the Resource Layer

- Integrated Storage Resource model
- Full control of the Storage of the Site/Resource Owner
- Mapping necessary between
  - Grid Users and Site Users
  - Grid Data names and Site Data names (logical vs. physical names)
  - Local Namespace is relevant as it holds access semantics
- Issues with distributed access
  - Peer to peer maps well onto this model
  - For single master and multi master, synchronization between local instances is needed
  - Job scheduling needs to take individual access capabilities into account in addition of the data being present or not

Model 1: Site Security Only

Implementation possibility:

- Client is accessing each storage individually, directly
- Client has proper credentials for **each** local storage element
- For non P2P models, synchronization is necessary
- Possible standardization problems as individual storages have potentially different ACL interpretations



Model 2: Middleware PDP, Site PEP

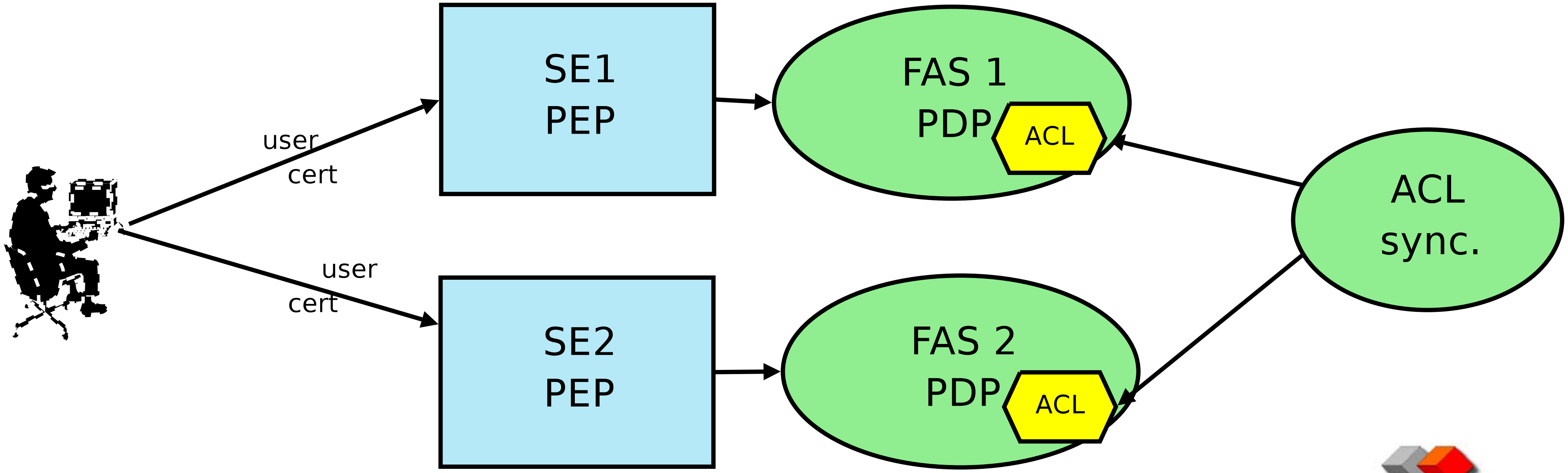
PDP is on site but in the Grid Layer. PEP in the Resource Layer

- Storage Resource delegates decision on access to the middleware
- On-site middleware so resource owners are still in control
  - Well suited for sites running local storage with limited semantics
- Mappings: same as M1
  - User, filename mappings still needed
  - ACLs/Security metadata stored now in the Grid Layer
- Issues with distributed access
  - Peer to peer maps well onto this model too
  - For single master and multi master, synchronization between local instances is still needed
  - Job scheduling needs to take individual access capabilities into account in addition of the data being present or not.

Model 2: Middleware PDP, Site PEP

Implementation possibility:

- A middleware: File Authorization Service FAS needed as PDP
- Client is accessing each storage individually, directly
- Client has proper credentials for **each** local storage element
- For non P2P models, synchronization is still necessary
- Lesser standardization problems as M1: middleware abstraction



Model 3: Middleware Control

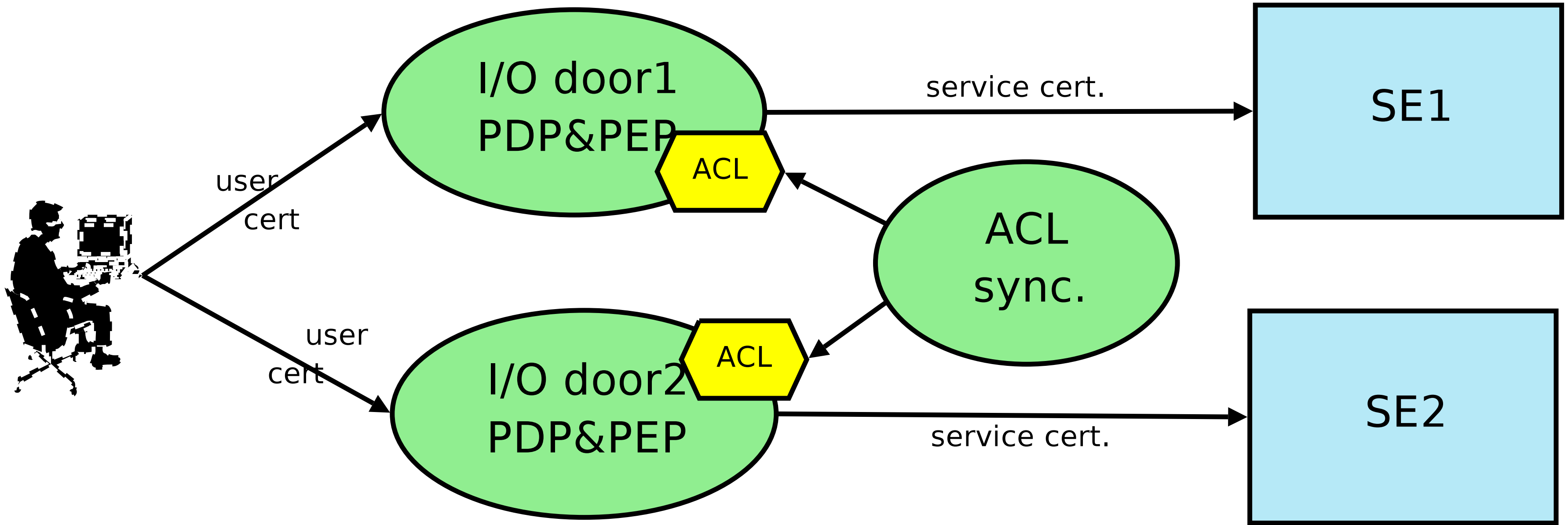
Middleware Layer controls both PEP and PDP

- Storage can only be accessed through the Middleware Layer
- On-site middleware so resource owners are still in control
- Mappings are managed by the Middleware
  - Abstraction of local storage semantics and namespace
  - ACLs/Security metadata stored and enforced in the Grid Layer
- Issues with distributed access
  - Peer to peer is still a good model, each site now has uniform semantics
  - For single master and multi master, synchronization between local instances is still needed
  - Job scheduling needs to take individual access capabilities still into account in addition of the data being present or not.

Model 3: Middleware Control

Implementation possibility:

- Middleware service acting as Door to the storage, keeping ACLs locally
- Client cannot access the Storage Element directly
- Client needs credentials **only** for the middleware
- Middleware **owns** the data on the SE and accesses it using a service cert



Model 4: VO PDP, Site PEP

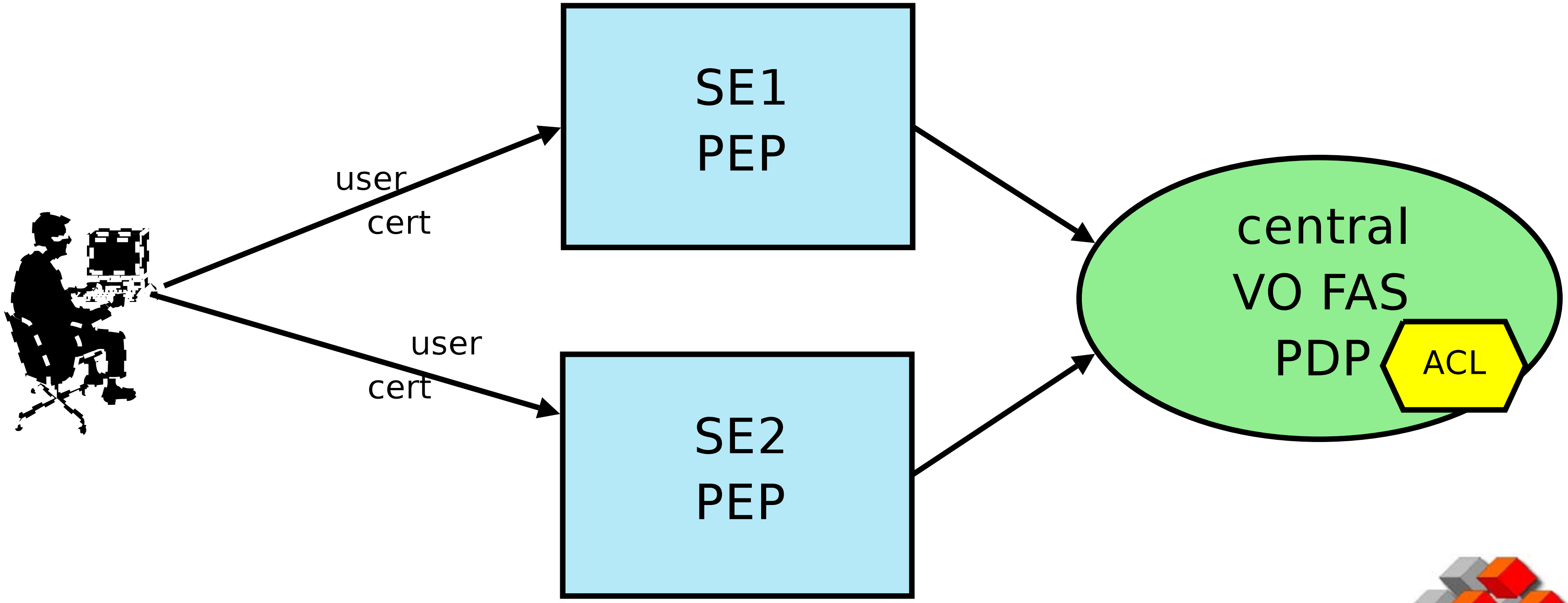
VO Layer controls the Decision making

- Storage is accessed directly by Clients as in Model 2
- The Decision is not an on-site service so the resource owners have to delegate the decision making to the VO, only enforcing it locally
- Mappings are managed by the VO
  - Abstraction of local storage semantics and namespace now up to the VO layer
  - ACLs/Security metadata managed by the applications but is a single point of failure
- Issues with distributed access
  - Peer to peer is not necessarily a good model as a potentially central VO service would need to be contacted for every operation
  - The VO PDP needs to decide whether to enforce single master or multi master, but can do so relatively easily
  - Job scheduling does not need to take individual site access into account.

Model 4: VO PDP, Site PEP

Implementation possibility:

- Storage needs a callout to the (central) VO PDP service
- Client has proper credentials for **each** local storage element
- No standardization needed



Model 5: VO PDP, Middleware PEP

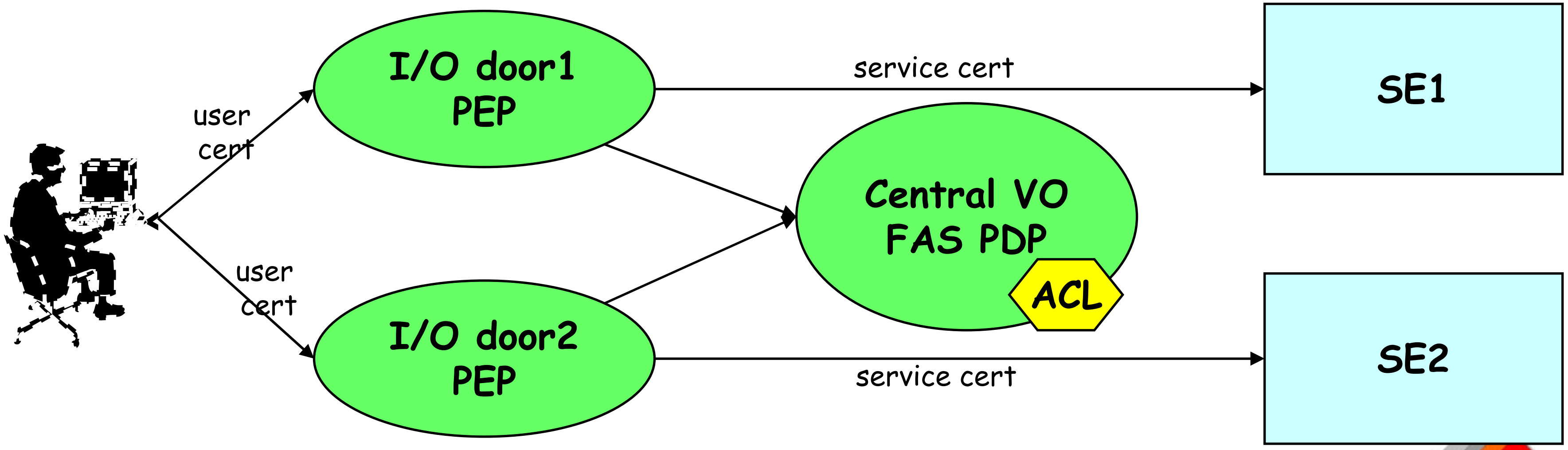
Middleware Layer controls PEP while VO maintains PDP

- Storage can only be accessed through the Middleware Layer as in Model 3
- Everything else as in Model 4

Model 5: VO PDP, Middleware PEP

Implementation possibility:

- Middleware service acting as Door to the storage but access control info is in the central PDP
- Client cannot access the Storage Element directly
- Client needs credentials **only** for the middleware
- Middleware **owns** the data on the SE and accesses it using a service cert

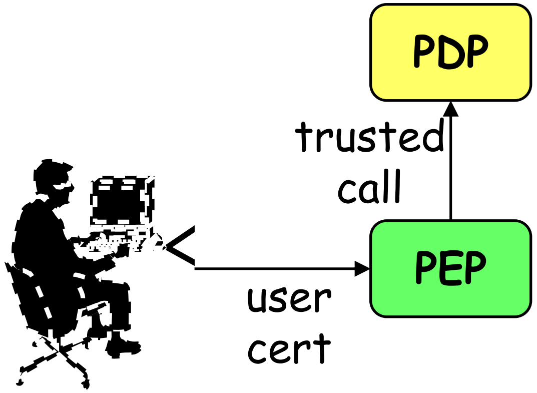


Alternative Implementations

Possible depending on how the communication between PDP and PEP is being done:

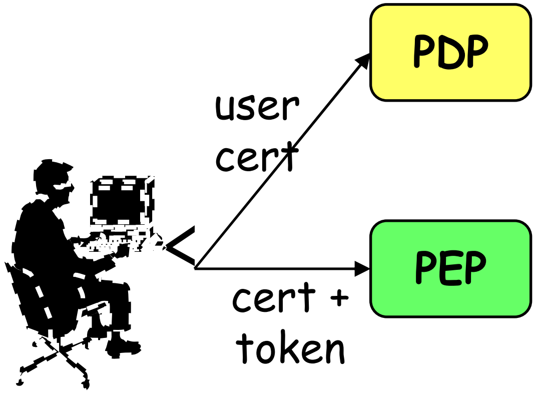
Pull

- Call-out from the PEP to the PDP
- Also called 'late' authorization as the decision is made very late in the process



Push

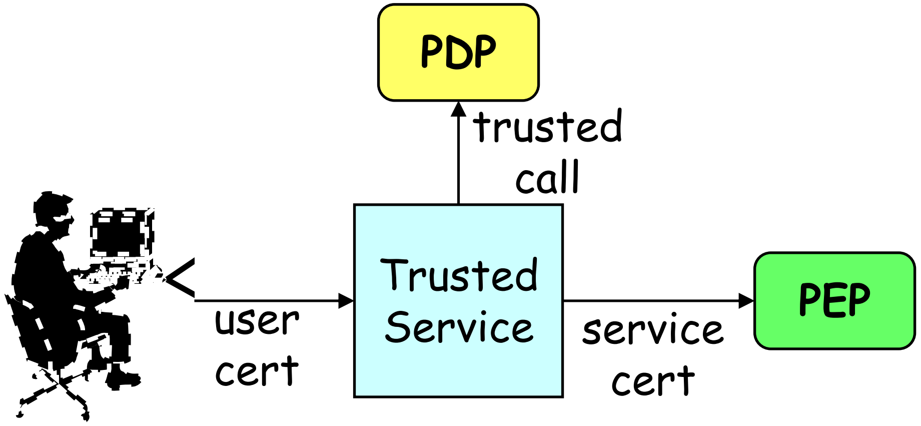
- User gets a token signed by the PDP
- User claims to have the rights on the data directly based on the secure token
- Also called 'external' authorization



Alternative Implementations

Trusted

- User entrusts a service to act on its behalf
- Service retrieves PDP info on behalf of user
- Service has ‘admin’ capabilities on PEP



SRM security

The current SRM security runs with **Model 1**

SRM v1 and v2

- Model 1 not by design but by default. For Model 1, the Pull implementation is trivial as the PDP and PEP are the same
- Discussions did not even start!
- If SrmCopy is not used, any model can be implemented on top of the SRM.
- SrmCopy in v1 and v2 only allows Model 1 by default, using the Peer to Peer security option (no synchronization on update).

SRM v3

- Foresees proper handling of ACLs with the necessary methods
- May be extended to include synchronization between sites also for SrmCopy
- Detailed discussions also to be had. SRM v3 is simply flexible enough to allow for any security model.

EGEE security model: gLite 1.5

The EGEE gLite versions up to v1.5 were designed to run with **model 3 or 5.**

Model 5

- gLite I/O: middleware service acting as door ('trusted' implementation as the actual enforcement is done by the SE)
- gLite Fireman Catalog: central VO-owned service

Model 3

- gLite I/O as door (trusted impl)
- gLite Fireman deployed locally at each site
- Synchronization between Fireman catalogs through a messaging service

EGEE security model: LCG and gLite3

The EGEE versions up to LCG 2.7 and starting gLite3 run with **model 1.**

Model 1

- Just using the SRM
- Alternatively, directly talking to the storage at each site over the native interface (dcap, rfio) or over GFAL

Not quite Model 2

- LCG File Catalog LFC acting as namespace service only, ACLs but not enforced
- Push implementation using VOMS groups but these are not signed by the LFC
- No synchronization

AliEn

The LHC Alice experiment’s own Grid Middleware called AliEn that largely influenced the EGEE design is working with **model 4**

- ‘Push’ implementation with a token given to the service by the Alien System
- ‘Central’ Alien File Catalog – distributed instance
- Direct access to file through xrootd protocol with the service token

SRB

The Storage Resource Broker from SDSC seems to be using **Model 3**

- Central namespace service
- Every access goes through the SRB layer and is tightly controlled by it

Summary

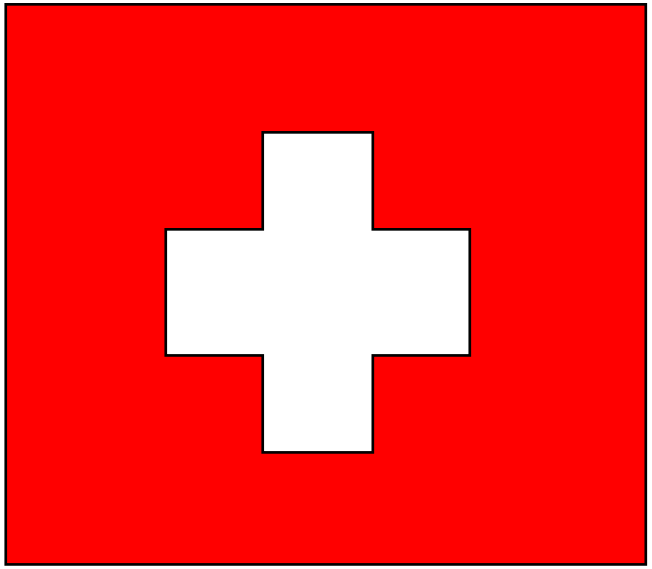
The users of a distributed Grid infrastructure need to be aware of the security semantics of their distributed data access model

- Not an easy decision which model to use
- Every model has advantages and disadvantages
- Possibly no one size fits all solution
- SRM v3 might accommodate any solution on top

- Still a lot of thinking/deciding to be done!

- .. And I haven’t even touched on the subject of co-scheduling…

..And now for the Really Important Stuff



HOPP SCHWIIZ