Project no. IST-033576

XtreemOS

Integrated Project
BUILDING AND PROMOTING A LINUX-BASED OPERATING SYSTEM TO SUPPORT VIRTUAL ORGANIZATIONS FOR NEXT GENERATION GRIDS

Training report and plan
D5.2.3

Due date of report: May 31st, 2009
Actual submission date: March 22th, 2010 (re-submission)

Start date of project: June 1st 2006

Type: Deliverable
WP number: 5.2

Name of responsible person: Michael SCHÖTTNER
Editor & editor's address:
Institution & address: Heinrich-Heine University Duesseldorf
Universitaetsstr. 1
40225 Duesseldorf, Germany

Version 1.8 / Last edited by Michael Schöttner / Date: March 22th, 2010

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Keywords: Training.

Revision history:

Version	Date	Authors	Institution	Sections Affected / Comments
1.0	8/05/2009	Michael Schöttner	UDUS	First draft
1.1	12/05/2009	Sandrine L'Hermitte	INRIA	Extensions
1.2	12/05/2009	Michael Schöttner	UDUS	Polishing
1.3	13/05/2009	Sandrine L'Hermitte	INRIA	Some more extensions
1.4	12/06/2009	Michael Schöttner	UDUS	Integration of reviewer comments
1.5	26/08/2009	Michael Schöttner	UDUS	Integration of EC comments
1.6	26/08/2009	Sandrine L'Hermitte	INRIA	Extensions + polishing
1.7	27/08/2009	Michael Schöttner	UDUS	Final polishing
1.8	22/03/2010	Michael Schöttner	UDUS	Added training material

Reviewers
Alvaro Arenas (STFC) and Gregor Pipan (XLAB)

Tasks related to this deliverable

Task No.	Task description	Partners involved
T5.2.3	Training engineers and users	UDUS*, INRIA, STFC, CNR, BSC, ULM, VUA, XLAB, ZIB
T5.2.4	XtreemOS summer school	STFC*, INRIA, UDUS, CNR, BSC, ULM, VUA, XLAB, ZIB
T5.2.5	XtreemOS day for key players	XLAB*, INRIA, STFC, CNR, BSC, ULM, VUA, XLAB, ZIB, UDUS

* task leader

Table of Contents

Executive Summary

This deliverable provides a report on training activities of the third year of the project and is also a planning document for future activities during the last year of the project. This document is based on Annex 1 and the two previous training deliverables provided in D5.2.1 and D5.1.2 (updated annually).

Internal training activities, e.g. how to install and configure XreemOS were organized co-located with general technical meetings. External training activities include an XtreemOS contribution to the CoreGRID summer school, a security-related tutorial that has been given at the ICS 2009 conference, and a tutorial presented during a INRIA/EDF/CEA joint summer school. Finally, the XtreemOS summer school (September 2009) preparation has been started.

During the last year of the project, training activities will have to extend external training activities and reach external audiences with different profiles (users & developers) with training on the whole XtreemOS software stack. The first audience that will be targeted includes all related European ICT-/Grid-projects, the Linux and the Grid communities. One of the main external training events will be the XtreemOS summer school in September 2009.

Finally, key players from industry and research need to be “trained”, too. The courses planned for this target audience shall give an overview and a clear insight into the commercially interesting know-how and benefits of XtreemOS (scheduled for spring 2010).

1. Introduction

The main goal of the WP5.2 in XtreemOS is the implementation of training activities for internal and external purposes.

Internal training targets primarily the project partners who require additional expertise on particular topics in order to ensure an effective and efficient design, implementation and integration of the different parts of the system.

In contrast, external training targets parties, which are not members of the consortium, but are interested in the topics related to the project. External training includes educational activities (e.g., contribution in summer schools and training programs and organization of the XtreemOS first summer school) for potential users and developers. Finally, decisions makers need to be "trained" too, in order to learn about potentially interesting commercial benefits of the XtreemOS technology.

2. Training report

2.1 Internal training activities

During the last general technical meeting in Amsterdam (March 23-27, 2009), several tutorials were given by XtreemOS members to other XtreemOS colleagues: theses internal trainings are essential to ensure a common understanding of the XtreemOS mechanisms and practises.

The first release of the system highlighted a need for a tutorial on how to install XtreemOS (installation and configuration procedures were difficult for inexperienced users and developers), on how to manage credentials in XtreemOS and on how to better understand the security API. Members from INRIA, XLAB and STFC organized these tutorials that were plenary sessions (targeting all project members).

Other tutorials were given on specific topics and focused specific WPs and/or developers.

Yvon Jégou (INRIA) gave a specific tutorial on the XtreemOS permanent testbed.

Nicolas Vigier and Antoine Giniès (EDGE) organized another tutorial about packaging for all the developers of the project. This tutorial was used for internal training.

It is also worth mentioning that EDGE had also previously visited different partners, e.g. CNR and STFC, training them on automated branching, testing and packaging scripts in the new SVN tree (December 2008).

2.2 External training activities

CoreGRID summer school
XtreemOS was presented at the CoreGRID summer school (7.7–11.7.2008) at Technische Universitaet Dortmund, Germany (webpage: http://www.it.irf.uni-dortmund.de/IT/CoreGRID/index.php). XtreemOS had a 45min. time slot and around 40 people attended the talk of Joerg Domaschka (ULM) entitled “Reliability and Availability in the XtreemOS Project”. The slides of the presentation are available on the CoreGRID summer school webpage: http://www.it.irf.uni-dortmund.de/IT/CoreGRID/talks/Domaschka.pdf and attached in Appendix A.

Gridforum Netherlands
Christine Morin and Sylvain Jeuland (INRIA) gave a masterclass (including demo) at Vrije Universiteit Amsterdam in the framework of Gridforum Netherlands on October 23, 2008 (http://isoc.nl/activ/2008-ChristineMorin.htm). Around 55 persons (students, PhD, business etc.) attended this talk. The slides of the presentation by Christine Morin are available online: http://isoc.nl/activ/2008-ChristineMorin.pdf

ENVOL school
Furthermore, Christine Morin (INRIA) gave two talks on Kerrighed technology, which is leveraged in XtreemOS, during the school on dEveloppemeNt et la ValOrisation des Logiciels en environnement de recherche (ENVOL) organized by CNRS/PLUME, Annecy, France, October 2008. The audience was composed of software development managers and developers from research labs aiming at developing and promoting software developed in laboratories (namely in open source). Christine Morin took the opportunity to include some features of XtreemOS in the presentation (kDFS for instance).

Kiberpipa event

On April 14th 2008 XLAB made an XtreemOS training session in the Kiberpipa slot. Kiberpipa ('Cyberpipe') is a hacker space in Ljubljana, Slovenia, established in 2001 as a part of the K6/4 Institute. The hacker space operates as a cultural centre, computer laboratory and Internet café (with free wireless access). Kiberpipa engages primarily in open source programming and the recycling of computer devices. It organises workshops, lectures, and entertainment and information events. Marjan Šterk and Matej Artač presented the XtreemOS project and its goals, and demonstrated its usage from the perspective of a VO administrator, a resource administrator, and a VO user. The audience consisted of Linux users, who showed interest in the project's results. The video of the presentation (in Slovenian) is available on Kiperpipa's website (http://video.kiberpipa.org/media/POT_XtreemOS/play.html).

Tutorial at ICS'09 conference

Beyond the above-mentioned talks a tutorial proposal titled "Security and VO Management in Grids" was submitted at various conferences (SC'08, IPDPS 2008) and accepted at ICS'09 (23rd International Conference on Supercomputing) taking place in New York, USA, June 2009.

This tutorial provided an overview of security and Virtual Organization management in established and new Grid systems. We surveyed the security and Virtual Organization management features provided by some major Grid middleware packages, and introduced the comparable functionality in XtreemOS, a Grid-based operating system. The training material was prepared by Corina Stratan (VUA), Alvaro Arenas (STFC), Christine Morin (INRIA), Haiyan Yu (ICT) and Yvon Jégou (INRIA).

The content and schedule of this tutorial is as follows:

Grid security and VO Management: concepts and issues - 1 hour

- Concepts of user identity/authentication, authorization & access control to resources
- Challenges to Grid security
- Single-Sign On and Federation
- VO concepts and models

Security and VO management in the state-of-the-art Grid systems - 1 hour

- Globus – authentication & Single-sign on, authorization, delegation, Community Authorization Service, plugins for VOMS
- gLite - authentication, authorization, delegation, VOMS
- UNICORE - clients and authentication Gateway
- VOMS Attribute Authority for UNICORE using SAML
- Security and VO management in XtreemOS

XtreemOS: a Grid-based Operating System – 1 hour

- XtreemOS objectives
- XtreemOS Foundation layer
  (credential storage via Key Retention Service, mention use of PAM)
- UID/GID mapping from VO attributes
- XtreemOS Grid layer - Services and Applications
- Show a job submission workflow invoking XtreemOS services
- XtreemOS advantages
- XtreemOS roadmap for interoperability

This tutorial has been widely promoted and a dedicated webpage has been created on the project website: http://www.xtreemos.eu/xtreemos-events.

The slides are attached in Appendix B.

Other events
Sylvain Jeuland (INRIA) visited the Laboratoire Bordelais de Recherche en Informatique (LABRI) and gave a detailed presentation on XtreemOS and presented a demonstration of XtreemOS (Bordeaux, France, October 2008).

Furthermore, Christine Morin presented a tutorial about XtreemOS at the INRIA/EDF/CEA joint summer school, near Paris, June, 2009. The slides are attached in Appendix C.

2.3 Preparation of the XtreemOS summer school

The first XtreemOS summer school will take place in Oxford, UK in September 7-11, 2009. This is the major event for the third period of the project in terms of training. The goal of this summer school is to attract potential developers and users for the XtreemOS system. As a matter of fact summer schools target PhD students and master students who are good testers and could potentially become future developers and users of the technology.

The preparation of the XtreemOS summer school (planned for M40) has been initiated by Alvaro Arenas (STFC): room rental, contact of potential invited speakers, draft programme and topics to be addressed, grants etc.

The organising committee is composed of:
- Alvaro Arenas, STFC Rutherford Appleton Laboratory (Organisation chair)
- Christine Morin, INRIA Rennes – Bretagne Atlantique (Scientific coordination)
- Sandrine L'Hermitte, INRIA Rennes – Bretagne Atlantique (Administrative support)
- Benjamin Aziz, STFC Rutherford Appleton Laboratory (Local organisation)
- Ian Johnson, STFC Rutherford Appleton Laboratory (Local organisation)

Promotional actions also started and the call for participation was widely disseminated (See D5.1.8).

A webpage was created on the project website: http://www.xtreemos.eu/xtreemos-events/xtreemos-summer-school-2009 where registration has been opened. A dedicated flyer was also created is also downloadable from this webpage (http://www.xtreemos.eu/xtreemos-events/xtreemos-summer-school-2009/xtreemos-summer-school-2009-2/2009-06-23.8405986064/download).

3. Training plan

All training activities will be coordinated with communication (WP5.1) and liaison activities (WP5.3).

3.1 Training researchers and XtreemOS users (T5.2.3)

The existing training material will be refined and extended depending on the XtreemOS members' needs. We will continue to organize internal training sessions whenever necessary.

During the last year of the project, T5.2.3 will have to extend to external training activities. The first audience that will be targeted includes all related European ICT-/Grid-projects, the Linux and the Grid communities in general. One of the main external training events will be the XtreemOS summer school, see Section 3.2.

Beyond training engineers and researchers this task will also provide training courses for external and internal users after the second software release. The different requirements of users will be addressed by specific training material. User training is planned to be co-located with training engineers, e.g. ½ day at the beginning of a training workshop and tutorials will also be submitted at international and national events (Supercomputing, ISC, …).

An initial course portfolio on various aspects of XtreemOS will be developed from tutorial submissions to conferences (e.g. ISC'09) and contributions to other EU project summer schools.

The consortium will organize a ½ day XtreemOS summit at EuroPar09 in August 2009, in Delft, The Netherlands. This summit will include talks from the consortium and demonstration sessions and discussions and it will aim at attracting people from the external world (raising interest, showing the appeal of the XtreemOS technology, discussing about XtreemOS-related topics etc.).

The programme and schedule of this XtreemOS summit is the following:

Time	Subject	Speaker	Type
14:30-15:00	Main objectives of XtreemOS	**Thilo Kielmann** Vrije universiteit Amsterdam, the Netherlands	Talk
15:00-15:30	Security model	**Alvaro Arenas** Science and Technology Facilities Council, the United Kingdom	Talk
15:30-16:00	Resource matching	**Guillaume Pierre** Vrije Universiteit Amsterdam, the Netherlands	Talk
16:00-16:30	Coffee break		
16:30-17:00	Parallel IO and replication in XtreemFS	**Björn Kolbeck** Zuse Institute Berlin, Germany	Talk
17:00-17:45	Applications and demonstrations	**Peter Linnell** INRIA Rennes, France	Demonstration
17:45-end	Open Q&A session	All	Discussion

3.2 XtreemOS summer school (T5.2.4)

The first XtreemOS summer school will take place in Oxford, UK in September 7-11, 2009.

The aims of the XtreemOS Summer School are:

- To introduce participants to emergent computing paradigms such as Grid computing and Cloud computing
- To provide lectures and practical courses on novel techniques to achieve scalability, highly availability and security in distributed systems
- To present Grid applications in the domains of E-science and business.
- To provide a forum for participants to discuss your research work and share experience with experience researchers.

The topics covered in the Summer School include:

- Introduction to Grids, Clouds, SOA and network-centric operating systems
- Grid programming interfaces
- VO Management and security
- Distributed data management
- Application execution management
- Scalability
- Grid Checkpointing

Both internal and external speakers will make lectures. The following external speakers have been invited:

- Paolo Costa, Microsoft Research Cambridge, UK
- Kate Keahey, Argonne National Laboratory, USA
- Cedric Le Goater, IBM, France
- Kathrin Peter, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Germany
- David Wallom, Oxford e-Research Centre, UK

Some XtreemOS members will also actively take part in the lectures:

- Christine Morin, INRIA Rennes-Bretagne Atlantique, France
- Tony Cortes, Barcelona Supercomputing Center, Spain
- Thilo Kielmann, Vrije Universiteit Amsterdam, The Netherlands
- Massimo Coppola, ISTI-CNR, Italy
- Bjorn Kolbeck, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Germany
- John Mehnert-Spahn, Heinrich-Heine Universitaet Duesseldorf, Germany
- Alvaro Arenas, STFC Rutherford Appleton Laboratory, UK

A preliminary program is available, as shown hereafter:

XtreemOS Summer School 2009 Wadham College, Oxford University, Oxford, UK					
Time	**Monday** **September 7**	**Tuesday** **September 8**	**Wednesday** **September 9**	**Thursday** **September 10**	**Friday** **September 11**
09:00 – 10:30	**Arrival of Participants**	VO Management and Security Alvaro Arenas, STFC	Data Management in D-Grid Community Projects Kathrin Peter, ZIB	Invited Speaker Paolo Costa, MSR Cambridge	Grid Checkpointing John Mehnert-Spahn, U. Dusseldorf
			XtreemFS – A Distributed and Replicated File System Bjorn Kolbeck, ZIB	Highly Scalable Services Massimo Coppola, CNR	Distributed State Game Management Michael Sonnenfroh, U. Dusseldorf
10:30 – 11:00		**Coffee Break**			
11:00 – 12:30		Grid Programming Interface – SAGA Thilo Kielmann, VUA	Practical on XtreemFS Bjorn Kolbeck, ZIB	Virtual Nodes Jorg Domaschka, ULM	Invited speaker Cedric Le Goater, IBM
				Practical on Highly Scalable Services Massimo Coppola, CNR	
13:00 – 14:00	**Lunch Break**				
14:00 – 15:30	Invited speaker Kate Keahey, Argonne Lab	Practical on Grid Programming Interface Thilo Kielmann, VUA	Application Execution Management Toni Cortes, BSC	Doctoral symposium	**Departure of Participants**
15:30 - 16:00	**Coffee Break**				
16:00 – 17:30	Introduction to Grids, SOA, and network-centric OS Christine Morin, INRIA	Invited speaker David Wallom, Oxford e-Research Centre	Practical on Application Execution Management Toni Cortes, BSC	Continuation Doctoral symposium	
19:00 -		Welcome reception, including poster session and project demos		XSS Dinner	

3.3 XtreemOS day for key players (T5.2.5, leader: XLAB)

The governing board decided to move this event to spring 2010. The major goal of this workshop is to raise awareness of XtreemOS to key players in the fields of grid systems, Linux, and operating systems and to demonstrate the value of our "products" (XtreemOS system and stand-alone components). The majority of participants of this workshop will be invited-only. It is planned to co-locate this XtreemOS major event with another major ICT fair/event and to invite decision makers. We also aim to produce a small book at the end of this workshop.

The workshop material content will partially emerge from the XtreemOS exploitation plans (WP5.1) and demo scenarios (WP4.4).

4. Conclusion

During the past year our external training activities have mostly targeted academic people (summer school participants, master class…) who can be seen as potential developers and users of the XtreemOS systems. The main training activity of the third year was the preparation of the first XtreemOS summer school.

During the last year of the project, we will focus on business key players and try to attract users and developers for the XtreemOS system and software components. The major events of the last year will be the XtreemOS summer school and the XtreemOS day for key players. XtreemOS members will in parallel continue to submit tutorials to summer schools and scientific conferences.

Appendix

A. XtreemOS talk at Coregrid summer school
The summer school was held in Dortmund, Germany, July, 2009.

ulm university universität ulm

Availability and Reliability in the XtreemOS Project

Jörg Domaschka | joerg.domaschka@uni-ulm.de | 11 July 2008

Institute of Distributed Systems | Ulm University

Christian Spann, Franz J. Hauck
Aspectix Research Team
Institute of Distributed Systems
Ulm University
Germany

Jeff Napper, Guillaume Pierre,
Maarten van Steen
Computer Systems Department
Vrije Universiteit Amsterdam
The Netherlands

Rüdiger Kapitza
Informatik 4
University Erlangen
Germany

Michal Szymaniak
Google Research

USA

Hans P. Reiser
LASIGE
University of Lisbon
Portugal

The XtreemOS Project

Overview:

- Eurpean project: 17 partners
- Investigate grid support in operating systems
- Personal computers, clusters, mobile devices

The XtreemOS Project

Overview:

- Eurpean project: 17 partners
- Investigate grid support in operating systems
- Personal computers, clusters, mobile devices

Targeting large peer-to-peer grids

- Off-the-shelf computers
- Connected via the Internet
- No central infrastructure, fully decentralised
- Churn, unreliable nodes

The XtreemOS Project

Overview:

- Eurpean project: 17 partners
- Investigate grid support in operating systems
- Personal computers, clusters, mobile devices

Targeting large peer-to-peer grids

- Off-the-shelf computers
- Connected via the Internet
- No central infrastructure, fully decentralised
- Churn, unreliable nodes

Unreliable environment

Need for reliable services
(e.g. security, monitoring, ...)

The XtreemOS Project

Overview:

- Eurpean project: 17 partners
- Investigate grid support in operating systems
- Personal computers, clusters, mobile devices

Targeting large peer-to-peer grids

- Off-the-shelf computers
- Connected via the Internet
- No central infrastructure, fully decentralised
- Churn, unreliable nodes

Unreliable environment $\Rightarrow$ **?** $\Leftarrow$ Need for reliable services (e.g. security, monitoring, ...)

Questions to answer

- How can reliability be achieved?

Reliability

Snapshots:

- Save state of application from time to time
- In case of failures: load snapshot

Reliability

Snapshots:

- Save state of application from time to time
- In case of failures: load snapshot

But:

- Application may be composed of 100s (1000s) of processes
  - Snapshotting requires coordination, communication

  $\Rightarrow$ Nothing is for free

Reliability

Snapshots:

- Save state of application from time to time
- In case of failures: load snapshot

But:

- Application may be composed of 100s (1000s) of processes
  - Snapshotting requires coordination, communication

  $\Rightarrow$ Nothing is for free
- User may experience downtime
  - Bad for login or security services

  $\Rightarrow$ Reliability $\neq$ Availability

Reliability

Snapshots:

- Save state of application from time to time
- In case of failures: load snapshot

But:

- Application may be composed of 100s (1000s) of processes
  - Snapshotting requires coordination, communication

  $\Rightarrow$ Nothing is for free
- User may experience downtime
  - Bad for login or security services

  $\Rightarrow$ Reliability $\neq$ Availability
- What entity monitors the application?
  - Has to be reliable and available
  - Has to be distributed

  $\Rightarrow$ Self-containment

Questions to answer

- How can reliability be increased?

Questions to answer

- How can reliability be increased?
- How can availability be increased?
- Is there a self-contained solution?

Availability

Replication:

- Availability by redundancy
- Provide identical entities at multiple sites
- Contains snapshots as special case
- Consistency protocol ensures reliability

Outline

What to replicate?

- Data
- Database
- Computing task
- Object
- Service

What to replicate?

- Data
- Database
- Computing task
- Object
- Service

$\Rightarrow$ Sophisticated algorithms for all fields ...

What to replicate?

- Data
- Database
- Computing task
- Object
- Service

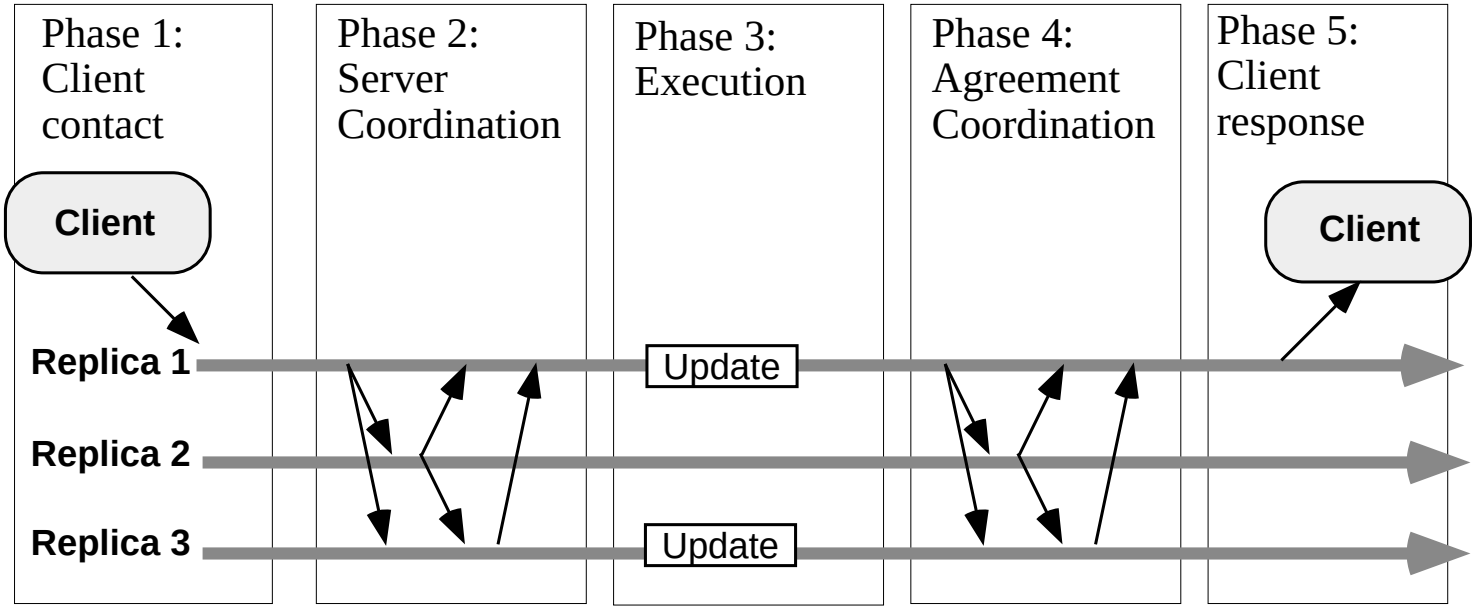
$\Rightarrow$ Sophisticated algorithms for all fields ...
$\Rightarrow$ ... and a general model

General Replication Model

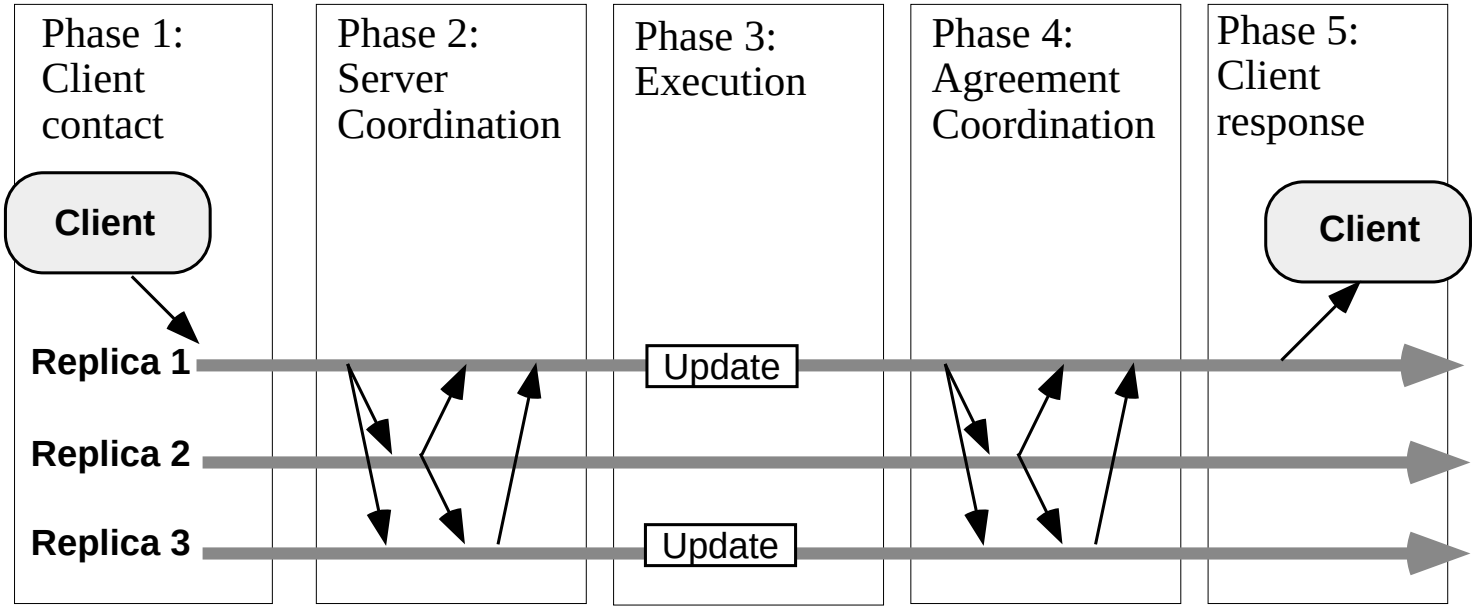
5 Phases

1. Request: client submits operation
2. Server coordination: synchronize the execution
   (e.g., message ordering)
3. Execution: operation is executed
   (by one or more replicas)
4. Agreement coordination: result of the operation
   (e.g., guarantee atomicity)
5. Response: send outcome back to client

General Replication Model



General Replication Model



Replication protocol:

- Decides on the use of a phase
- Different approaches per phase
- Different demands to the code

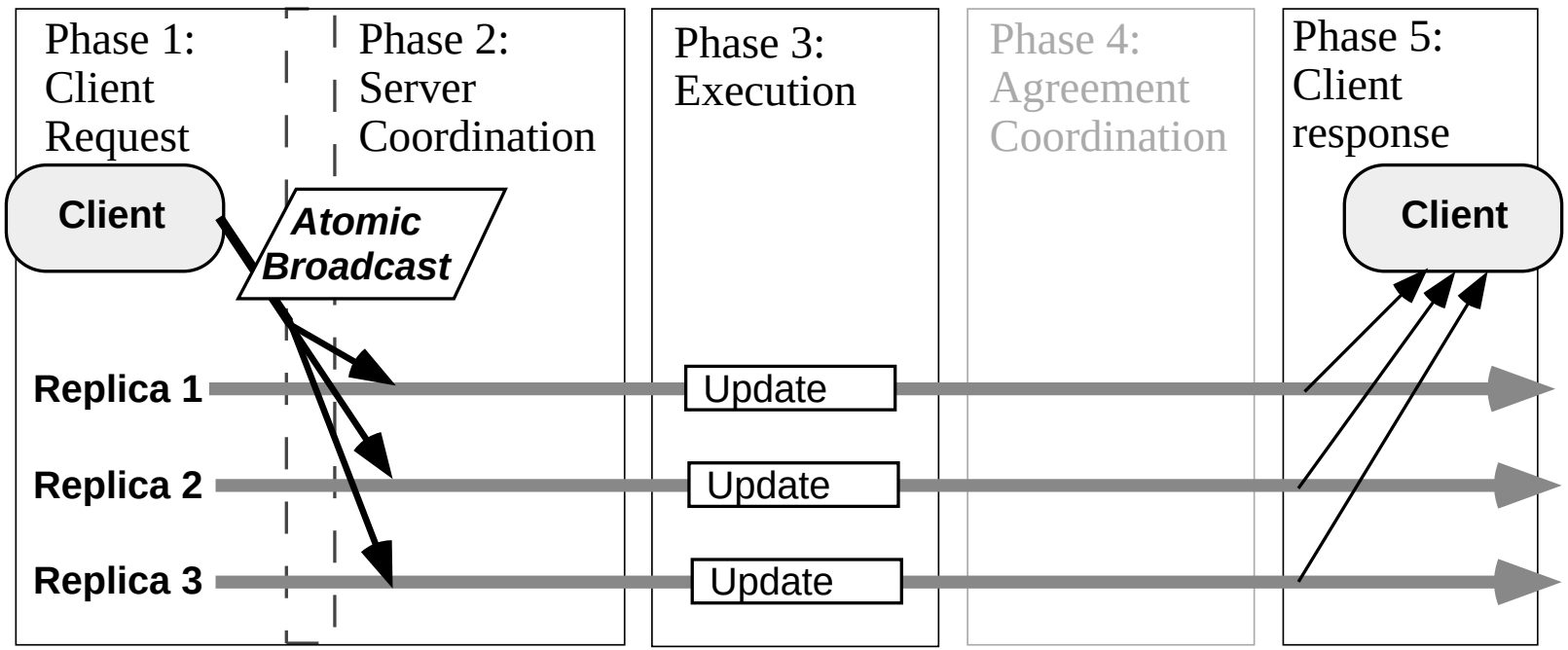
Classification

Active replication:

- State-machine replication
- Decentralised approach
- Request processed by all replicas
- Simple due to symmetry
- Quick reaction to failures
- Demanding with respect to determinism
  - Message ordering
  - Execution order

Classification

Active replication:



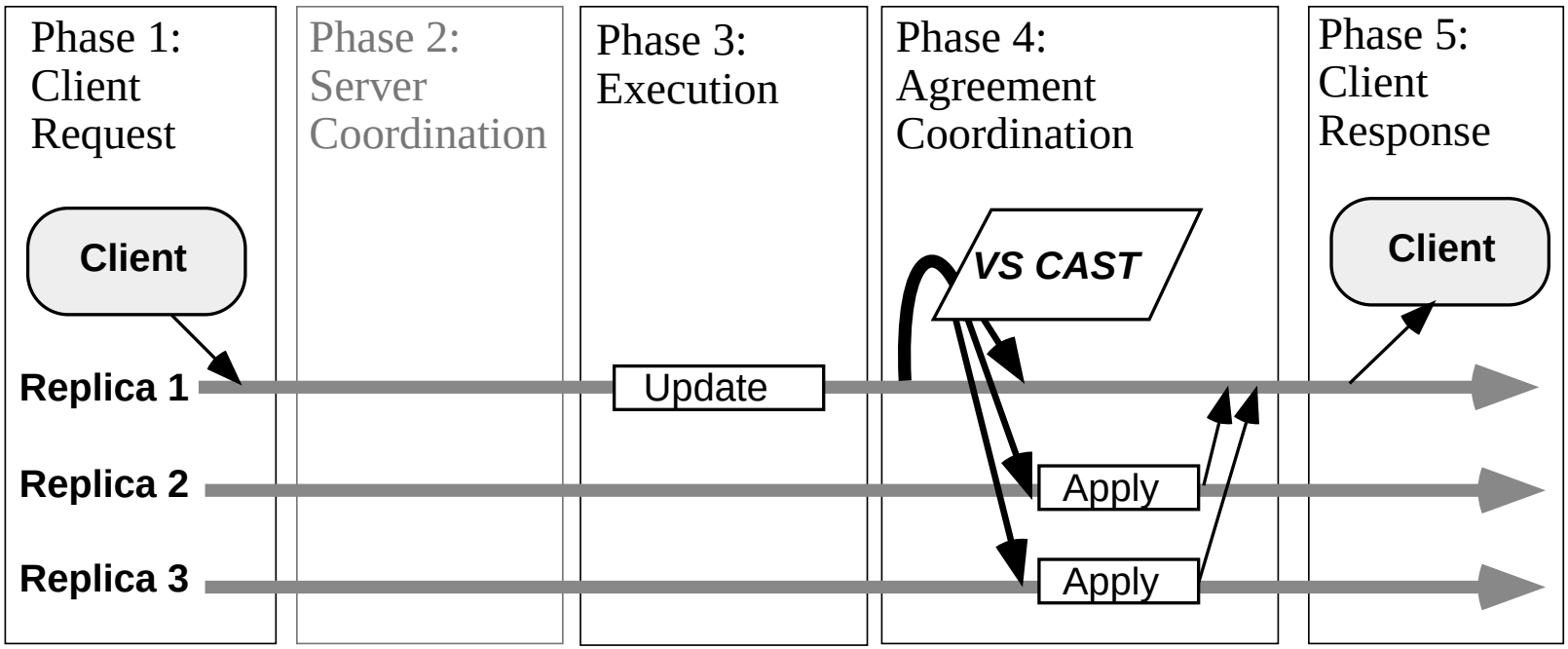
Classification

Passive replication:

- Primary backup replication
- Centralised approach
- Request processed by a single replica (primary)
- New state/state changes transferred to backups
- Failure of primary requires re-election
- Can handle nondeterminism (sometimes)

Classification

Passive replication:



Outline

Environment

What to replicate?

- Data
- Database
- Computing task
- **Object/Service**

Environment

What to replicate?

- Data
- Database
- Computing task
- **Object/Service**

Why Objects and Services?

"Can't you just use databases?"

- Many applications do not need stable storage
- Uniform programming model
- Support for legacy applications

Virtual Nodes: XtreemOS Approach to Reliability

Replication Framework

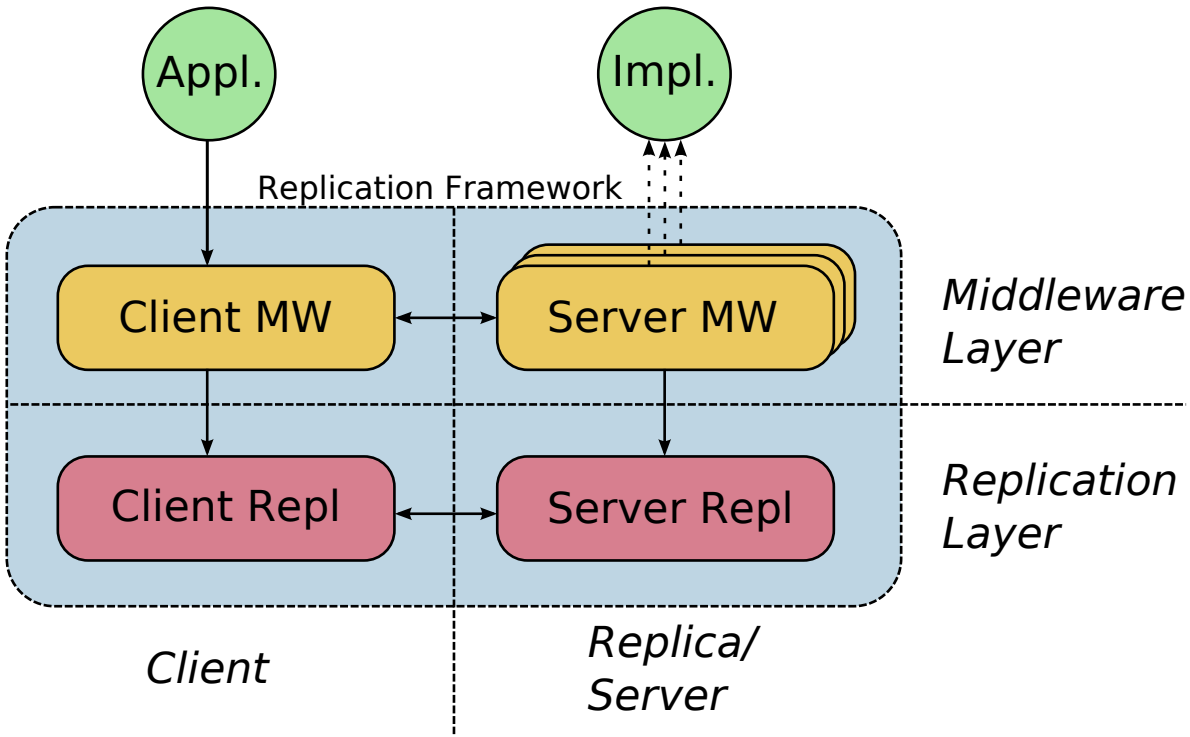
- Java-based
- Support for changing replica groups
- Multiple replication protocols
- Multiple middleware interfaces (CORBA, J-RMI, SOAP, ...)
- Support for nested invocations (SOA)
- Optimization for *read-only* invocations
- Support for deterministic multithreading
- Self-contained: independent of other nodes and services
- Service implementation orthogonal to replication

Virtual Nodes: XtreemOS Approach to Reliability

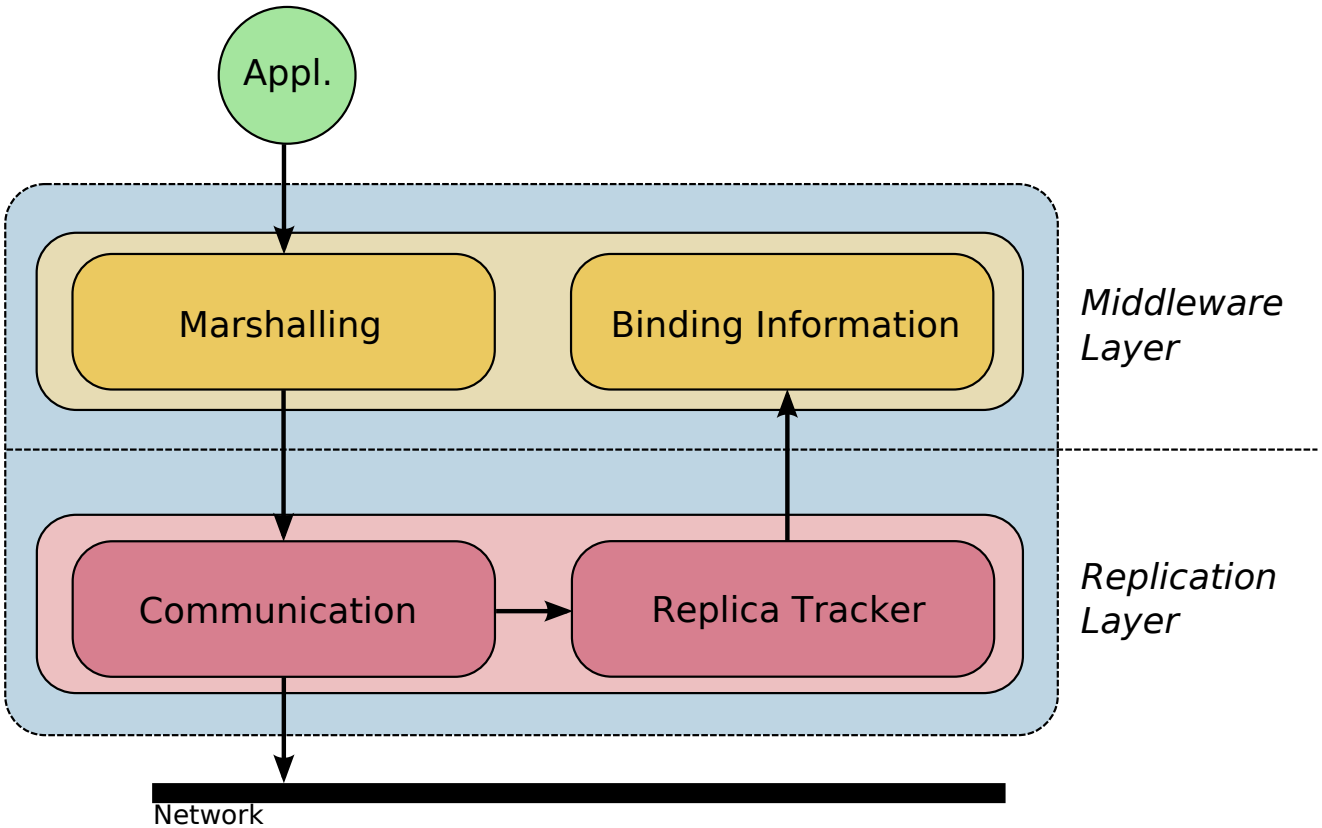
Replication Framework

- Java-based
- Support for changing replica groups
- Multiple replication protocols
- Multiple middleware interfaces (CORBA, J-RMI, SOAP, ...)
- Support for nested invocations (SOA)
- Optimization for *read-only* invocations
- Support for deterministic multithreading
- Self-contained: independent of other nodes and services
- Service implementation orthogonal to replication
  - Except for non-deterministic methods
  - Except for state transfer
  - ...

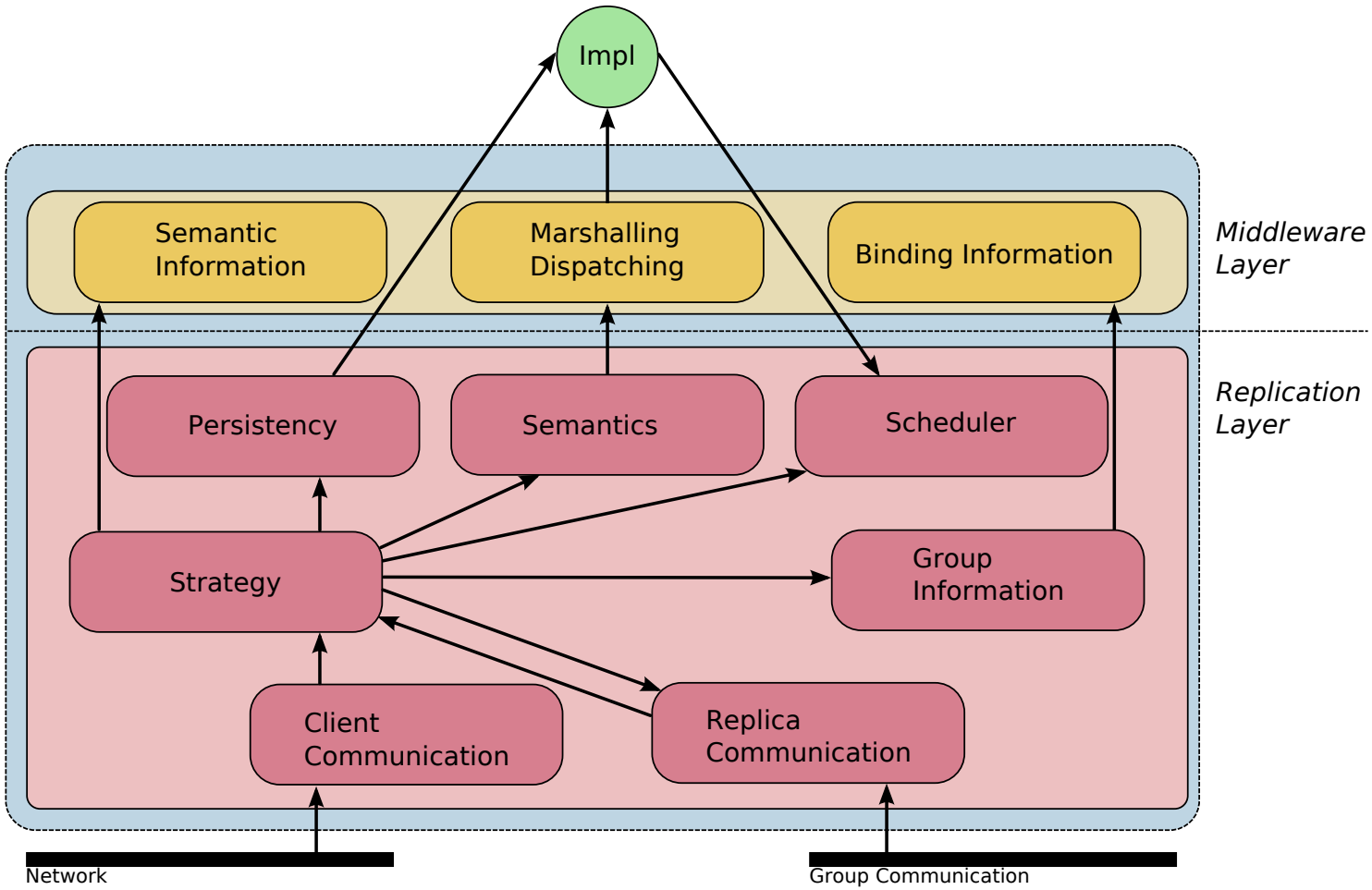
Architecture: Overview



Architecture: Client-side



Architecture: Server-side



Deterministic Scheduling

Active Replication Requires Determinism

- Multithreading is non-deterministic
- Single-threaded execution
  - Slow and dead-lock prone
  - Denies use of condition variables (`wait`, `notify`)
  - Does not make use of multi-core architectures

Deterministic Scheduling

Active Replication Requires Determinism

- Multithreading is non-deterministic
- Single-threaded execution
  - Slow and dead-lock prone
  - Denies use of condition variables (`wait`, `notify`)
  - Does not make use of multi-core architectures

Deterministic Multithreading

- Deterministic thread switching: limited concurrency
- Four algorithms with different properties
  - Single active thread (SAT, Reiser et al.)
  - Multiple active threads (MAT, Reiser et al.)
  - Lose synchronization algorithm (LSA, Basile et al.)
  - Preemptive deterministic scheduling (PDS, Basile et al.)

Deterministic Scheduling

Active Replication Requires Determinism

- Multithreading is non-deterministic
- Single-threaded execution
  - Slow and dead-lock prone
  - Denies use of condition variables (`wait`, `notify`)
  - Does not make use of multi-core architectures

Deterministic Multithreading

- Deterministic thread switching: limited concurrency
- Four algorithms with different properties
  - Single active thread (SAT, Reiser et al.)
  - Multiple active threads (MAT, Reiser et al.)
  - Lose synchronization algorithm (LSA, Basile et al.)
  - Preemptive deterministic scheduling (PDS, Basile et al.)
- No one-size-fits-all solution

Scheduler Integration

Intercept Java Synchronisation Statements:

- `synchronized` methods and blocks
  - `synchronized` instance methods
  - `synchronized` static methods
  - `synchronized` blocks
- `wait()`, `notify()`, and `notifyAll()` calls

Scheduler Integration

Intercept Java Synchronisation Statements:

- `synchronized` methods and blocks
  - `synchronized` instance methods
  - `synchronized` static methods
  - `synchronized` blocks
- `wait()`, `notify()`, and `notifyAll()` calls

Interception: Replace Statements by Calls to Scheduler

- `synchronized`: pair of lock/unlock invocations
- All other: simple replacement
- On source code or byte code level
- Transparent to service developer
- Appropriate also for legacy applications

Interception by Code Transformation

```
public class Queue extends ... {
    public synchronized
            String remove()
    {
        while(data.size()==0)
            wait();
        return data.remove(0);
    }


    public synchronized
        void append(String x)
    {
        data.add(x);
        notify();
    }


}
```

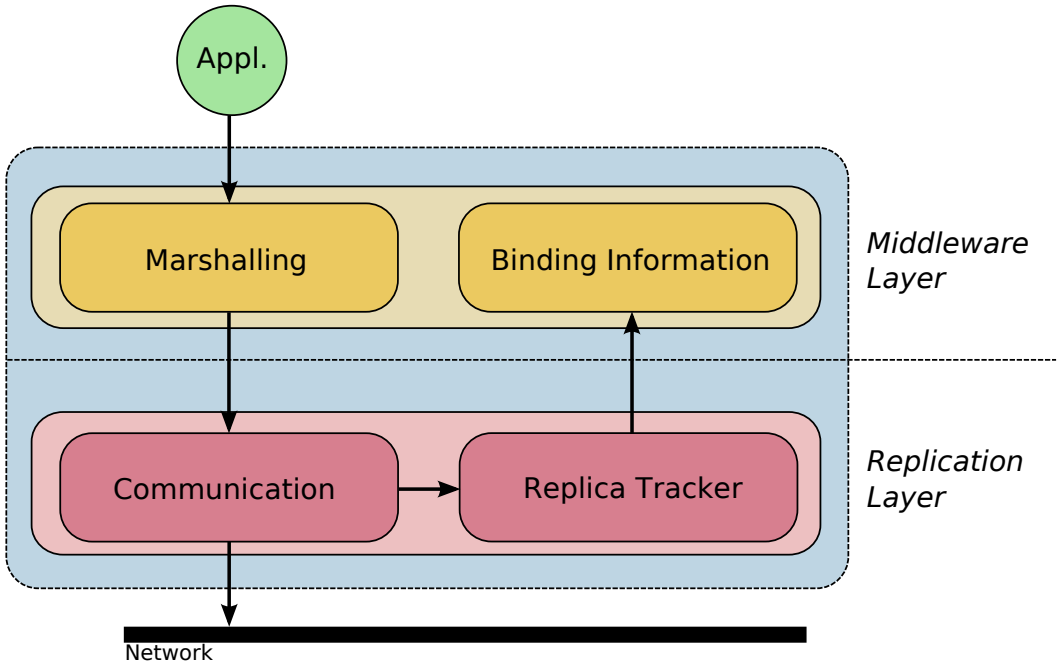
Interception by Code Transformation

```
public class Queue extends ... {
    public synchronized
            String remove()
    {
        while(data.size()==0)
            wait();
        return data.remove(0);
    }

    public synchronized
        void append(String x)
    {
        data.add(x);
        notify();
    }

}
```

⇒

```
public class Queue extends ... {
    public String remove() {
        _scheduler().lock(this);
        try {
            while(data.size()==0)
                _scheduler()._wait(this);
            return data.remove(0);
        } finally {
            _scheduler().unlock(this);
        }
    }
    public void append(String x) {
        _scheduler().lock(this);
        try {
            data.add(x);
            _scheduler()._notify(this);
        } finally {
            _scheduler().unlock(this);
        }
    }
}
```

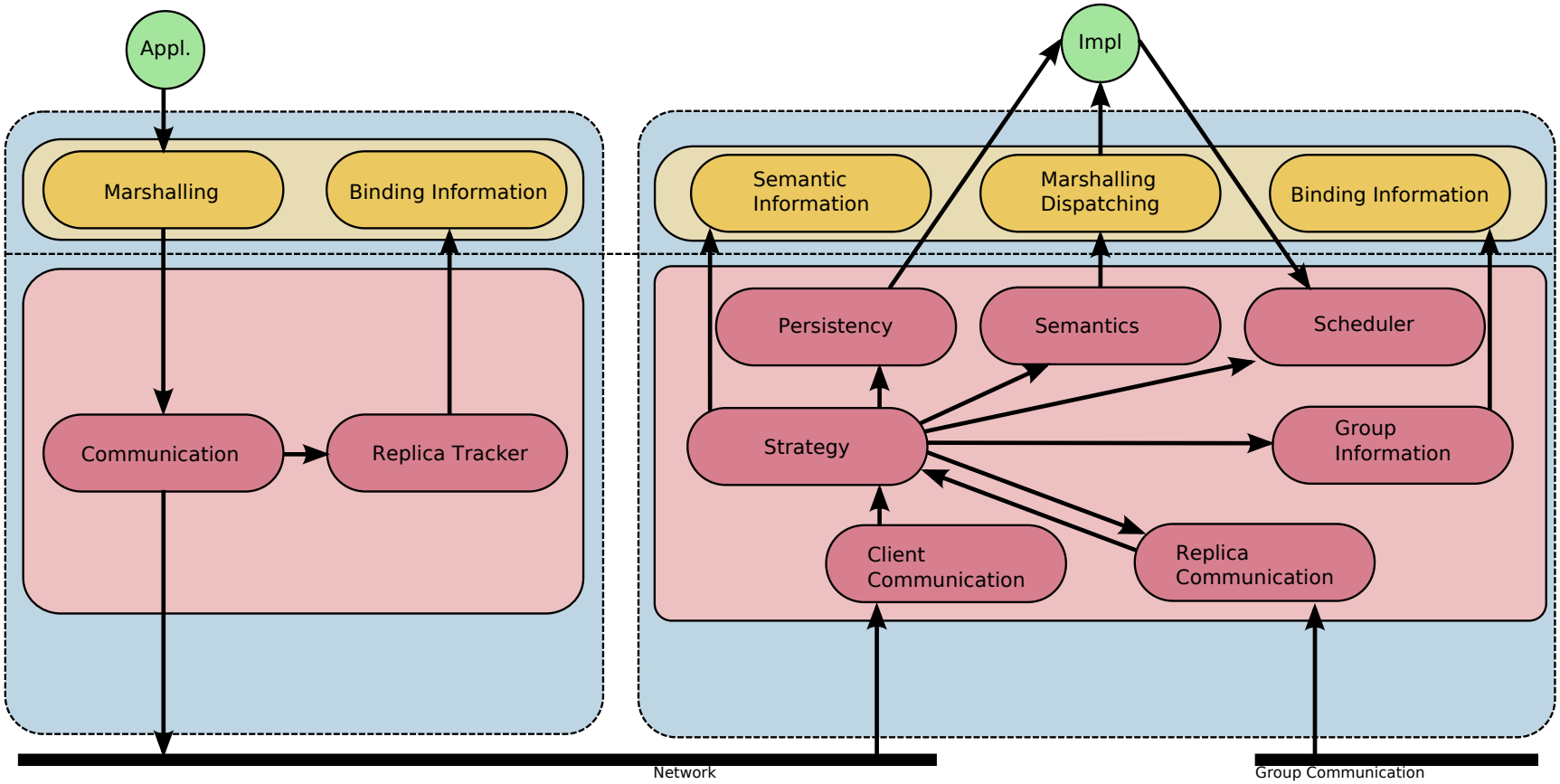
Client Transparency



- Client has to install additional software
- Application developer has to be aware of replication
- Violates the goal of transparency

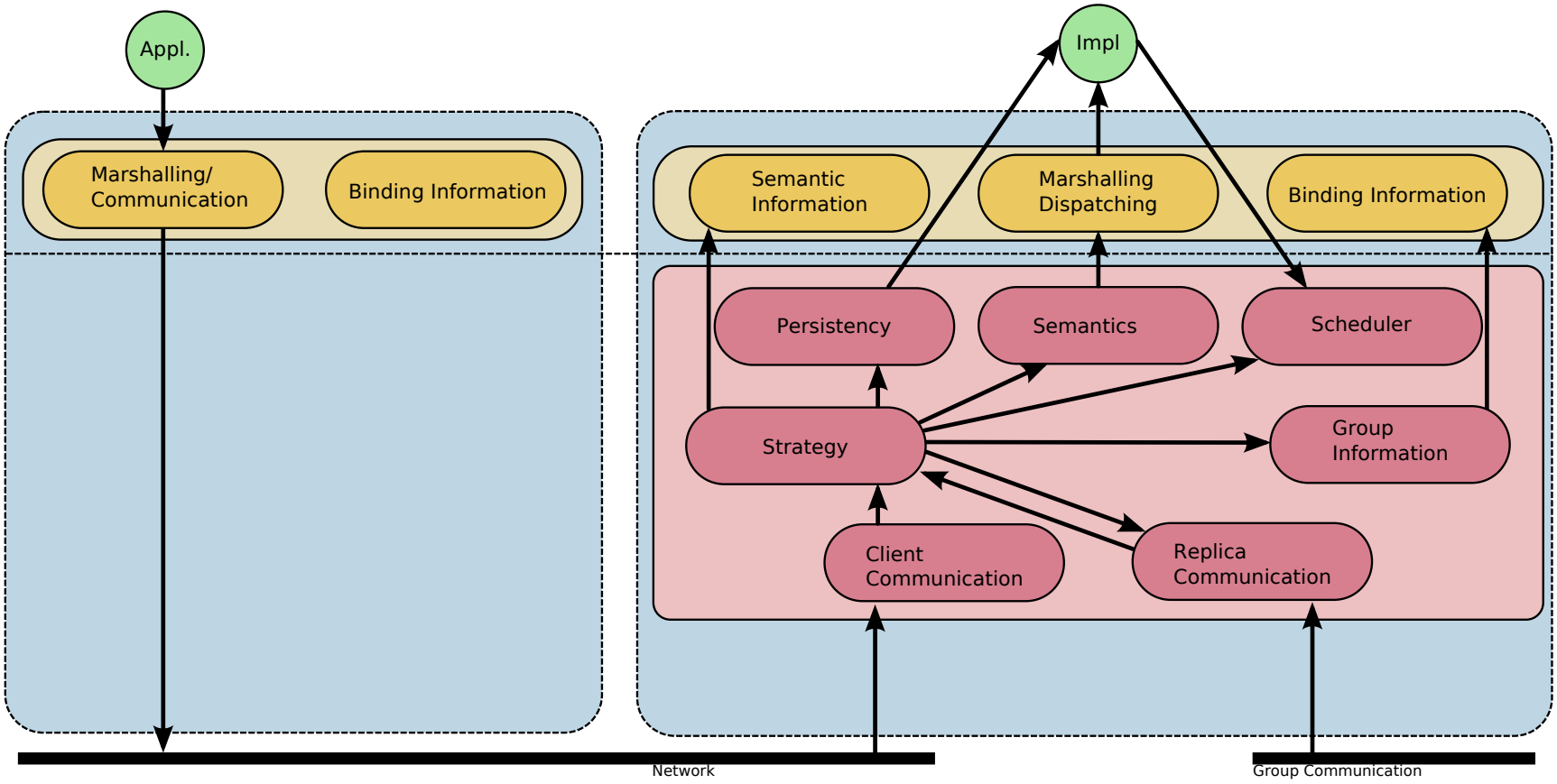
Increase Client Transparency

Remove Replica Layer



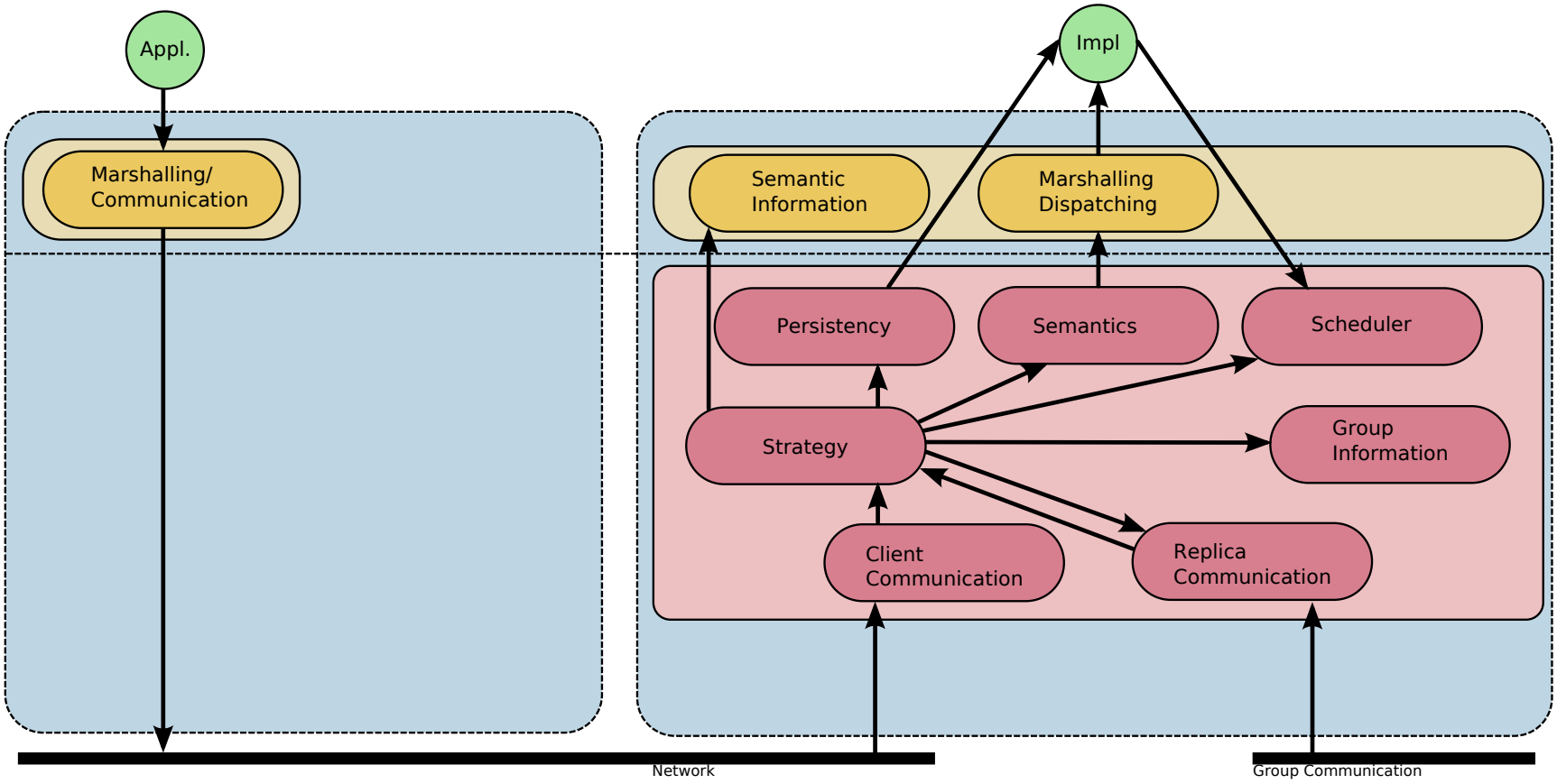
Increase Client Transparency

Remove Binding Information

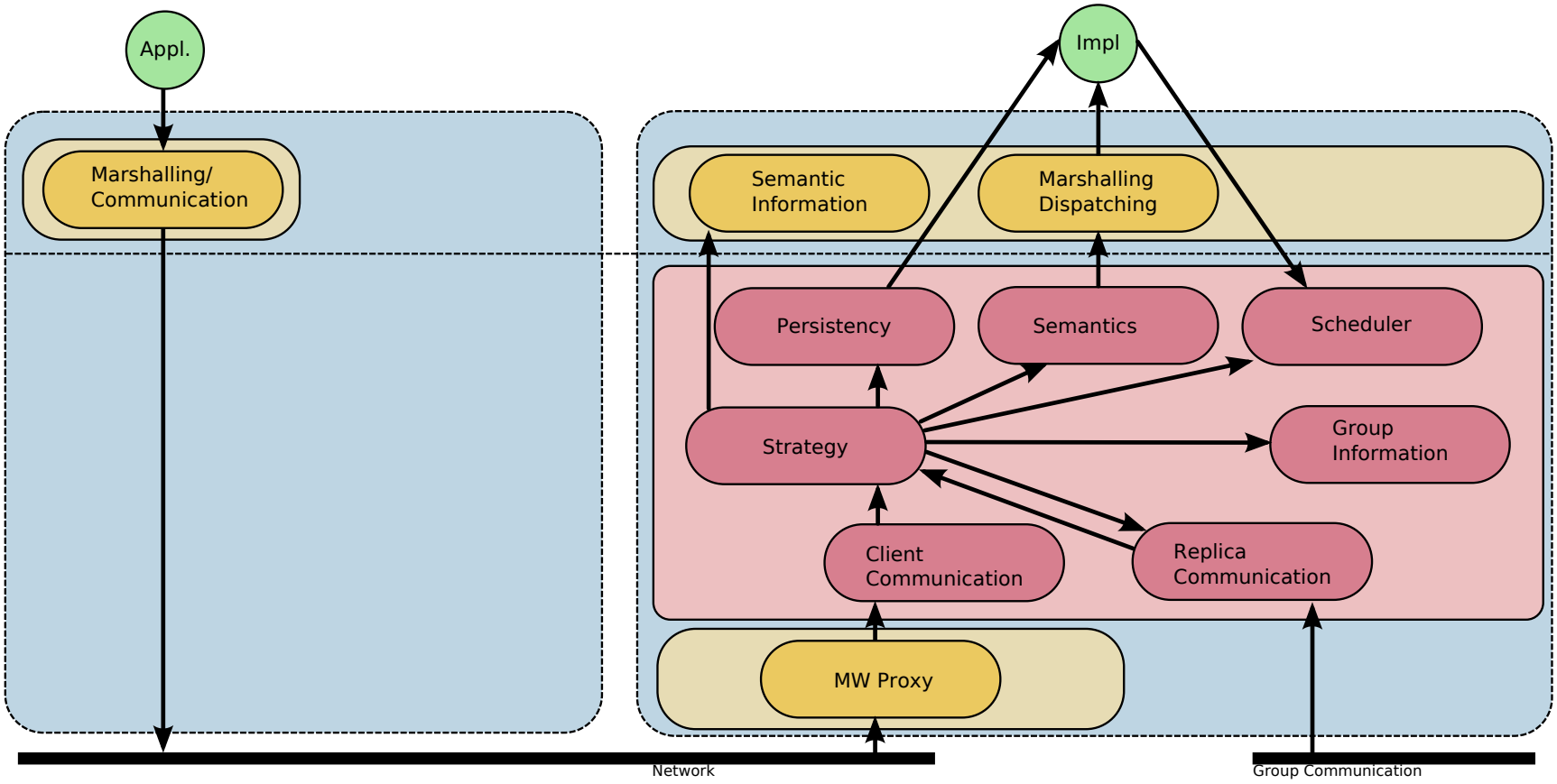


Increase Client Transparency

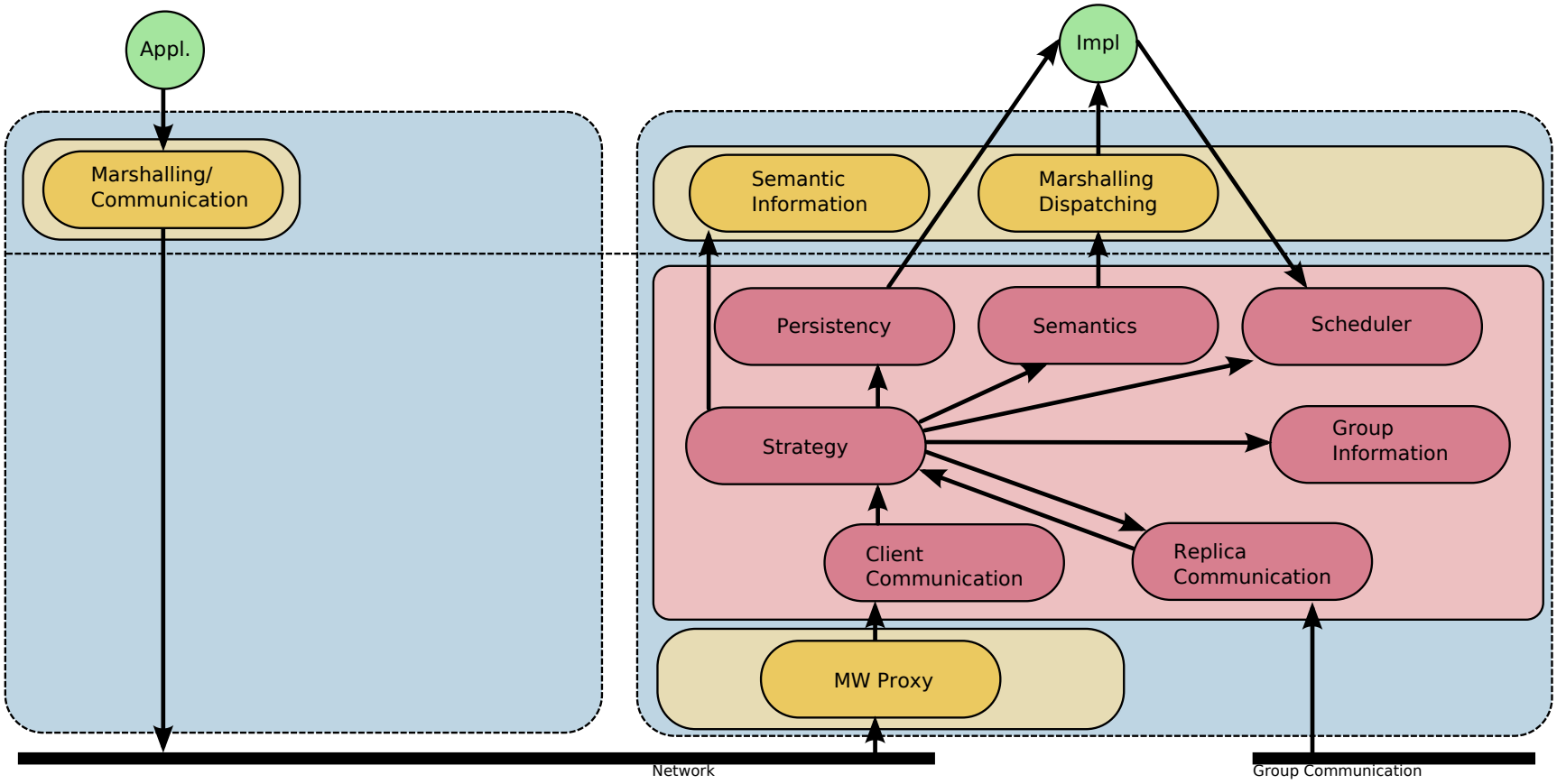
Add Middleware Proxy



Increase Client Transparency



Increase Client Transparency



"How do clients keep track of service location?"

Replica Tracking

Use a location service?

- How is the location service being tracked?

Replica Tracking

Use a location service?

- How is the location service being tracked?

Use client-side daemon?

- Will work most of the time
- Still no guarantee
- Additional traffic due to pulling
- No fix address: initial contact difficult

Replica Tracking

Use a location service?

- How is the location service being tracked?

Use client-side daemon?

- Will work most of the time
- Still no guarantee
- Additional traffic due to pulling
- No fix address: initial contact difficult

Our Approach: Exploit Mobile IPv6

- Uses standardized techniques
- Does not require any modifications at client side

Outline

Mobile IPv6 in a Nutshell

Mobile nodes reachable while away from home networks

- Correspondent node (CN): any node talking to mobile node

Mobile IPv6 in a Nutshell

Mobile nodes reachable while away from home networks

- Correspondent node (CN): any node talking to mobile node

Mobile Node: Two Addresses

- Home address (HoA): identifies mobile node, never changes
- Careof address (CoA): represents mobile node's current location

Mobile IPv6 in a Nutshell

Mobile nodes reachable while away from home networks

- Correspondent node (CN): any node talking to mobile node

Mobile Node: Two Addresses

- Home address (HoA): identifies mobile node, never changes
- Careof address (CoA): represents mobile node's current location

Transparency for High-level Protocols:

- Mobile nodes addressed by HoA
- IP-level translates HoA to CoA
- Location changes are annouced by the mobile node

Mobile IPv6 in a Nutshell

Mobile nodes reachable while away from home networks

- Correspondent node (CN): any node talking to mobile node

Mobile Node: Two Addresses

- Home address (HoA): identifies mobile node, never changes
- Careof address (CoA): represents mobile node's current location

Transparency for High-level Protocols:

- Mobile nodes addressed by HoA
- IP-level translates HoA to CoA
- Location changes are annouced by the mobile node

"Sounds nice, but how does the IP-level know?"

Mobile IPv6 in a Nutshell (II)

Home Agent (HA)

- Router in home network
- Mobile node informs HA about CoA
- Knows mapping from HoA to CoA

Mobile IPv6 in a Nutshell (II)

Home Agent (HA)

- Router in home network
- Mobile node informs HA about CoA
- Knows mapping from HoA to CoA

"Hey, wait a second! You do use a central entity! Isn't this cheating!?!"

Mobile IPv6 in a Nutshell (II)

Home Agent (HA)

- Router in home network
- Mobile node informs HA about CoA
- Knows mapping from HoA to CoA

"Hey, wait a second! You do use a central entity! Isn't this cheating!?!"

Yes, but ...

- Routers are not switched off spontaniously
- Routers run a small software system and tend to be less buggy
- No network depends on a single router

Distributed Servers

Distributed Server

- Group of nodes pretending to be a mobile node
- Identified by the home address
- Node addresses represent careof addresses

Distributed Servers

Distributed Server

- Group of nodes pretending to be a mobile node
- Identified by the home address
- Node addresses represent careof addresses

Features

- One node registers at home agent (contact node)
- Nodes can hand back and forth single connections (cooperatively)
- Contact node can change (cooperatively)

Connection Handoff

IP layer: Change address mapping at client

- Part of mIPv6 protocol
- Involves client, donor, home agent, and receiver
- Requires kernel patch at server machines

Connection Handoff

IP layer: Change address mapping at client

- Part of mIPv6 protocol
- Involves client, donor, home agent, and receiver
- Requires kernel patch at server machines

Transport Layer

- Connectionless protocols (UDP): —
- Connection-based protocols (TCP): copy socket state
- Requires kernel patch at server machines

Connection Handoff

IP layer: Change address mapping at client

- Part of mIPv6 protocol
- Involves client, donor, home agent, and receiver
- Requires kernel patch at server machines

Transport Layer

- Connectionless protocols (UDP): —
- Connection-based protocols (TCP): copy socket state
- Requires kernel patch at server machines

Application Layer

- Stateless applications: —
- Stateful applications: copy application state
- Requires cooperation of application

Outline

Integrated Approach

Benefit:

- Virtual Nodes: fault-tolerance for Distributed Servers
- Distributed Servers: anycast mechanism for Virtual Nodes

Integrated Approach

Benefit:

- Virtual Nodes: fault-tolerance for Distributed Servers
- Distributed Servers: anycast mechanism for Virtual Nodes

Facts:

- Handover requires an old socket state
  $\Rightarrow$ Replication of state
- Only reasonable with active replication

Integrated Approach

Benefit:

- Virtual Nodes: fault-tolerance for Distributed Servers
- Distributed Servers: anycast mechanism for Virtual Nodes

Facts:

- Handover requires an old socket state
  $\Rightarrow$ Replication of state
- Only reasonable with active replication

Failure Detection:

- Minimize experienced downtime: change contact node quickly
- Minimize false positives: exclude group members slowly

Invocation

1. Client sends request to contact node
2. Contact node copies socket state
3. Contact node broadcasts request and socket
4. All nodes process request
5. Contact node sends reply to client
6. Contact node broadcasts new socket state

Discussion

Fault-tolerance:

- No fault-tolerance during steps 1 and 2
- Steps 3 – 5: Handover reveals #bytes sent and received
  - Allows to send remaining bytes of reply

Discussion

Fault-tolerance:

- No fault-tolerance during steps 1 and 2
- Steps 3 – 5: Handover reveals #bytes sent and received
  - Allows to send remaining bytes of reply

Minimal overhead (copying socket)

- Step 6 purely for garbage collection
- Piggyback on other requests

Discussion

Fault-tolerance:

- No fault-tolerance during steps 1 and 2
- Steps 3 – 5: Handover reveals #bytes sent and received
  - Allows to send remaining bytes of reply

Minimal overhead (copying socket)

- Step 6 purely for garbage collection
- Piggyback on other requests

Changing contact node

- No effect on client
- Other replicas need to know
- Causes an additional group message

Conclusion

XtreemOS:

- Challenge for reliability and availability
- Replication can solve both issues

Conclusion

XtreemOS:

- Challenge for reliability and availability
- Replication can solve both issues

XtreemOS Virtual Nodes:

- Configurable replication framework for fault-tolerance
- Support for multiple middleware systems at client-side
- Deterministic multithreading

Conclusion

XtreemOS:

- Challenge for reliability and availability
- Replication can solve both issues

XtreemOS Virtual Nodes:

- Configurable replication framework for fault-tolerance
- Support for multiple middleware systems at client-side
- Deterministic multithreading

XtreemOS Distributed Servers:

- Anycast due to mobile IPv6
- Group of nodes pretends to be a mobile node
- Handing over of connections

Conclusion

XtreemOS:

- Challenge for reliability and availability
- Replication can solve both issues

XtreemOS Virtual Nodes:

- Configurable replication framework for fault-tolerance
- Support for multiple middleware systems at client-side
- Deterministic multithreading

XtreemOS Distributed Servers:

- Anycast due to mobile IPv6
- Group of nodes pretends to be a mobile node
- Handing over of connections

Integration:

- Both systems are orthogonal
- Increases client-side transparency

Papers

- Matthias Wiesmann et al: *Understanding Replication in Databases and Distributed Systems*. ICDCS '00
- Hans P. Reiser et al: *Consistent Replication of Multithreaded Distributed Objects*. SRDS'06
- Hans P. Reiser et al: *Deterministic Multithreading for Replicated CORBA Objects*. PDCS'06
- Claudio Basile et al: *Active Replication of Multithreaded Applications*. Transactions on Parallel and Distributed Systems, May 2006
- Michal Szymaniak et al: *Enabling Service Adaptability with Versatile Anycast*. Concurrency and Computation: Practice and Experience, September 2007.

B. XtreemOS tutorial at ICS09

The tutorial was given at the INRIA/EDF/CEAR joint summer school, near Paris, June, 2009.

XtreemOS

Enabling Linux for the Grid

ICS'09

Tutorial on Security and Virtual Organization Management in Grids

Part 1 – Fundamentals in Security and VO

New York, June 12, 2009

Information Society Technologies

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Contributors

- **Presenters**
  - Yvon Jégou, INRIA Rennes, France
  - Christine Morin, INRIA Rennes, France
  - Corina Stratan, Vrije University Amsterdam, The Netherlands
- **Acknowledgements**
  - Alvaro Arenas, STFC
  - Haiyan Yu, ICT/CAS, China

Credits

- **Some slides are based on presentations given by:**
  - Alvaro Arenas' Grid security tutorial at CoreGRID Summer School 2008
  - Matej Artac's presentation on XtreemOS VOPS
  - Ake Edlund's security course at ISSGC'07
  - Peter Gutmann's tutorial on Security
  - Syed Naqvi's Grid security tutorial at CGW 2006
  - Philippe Massonet, CETIC, presentation on Grid security requirements, OGF 25, March 2009

General Outline

- Fundamentals in Security & VO
- State of art on security & VO management in Grid systems
- VO management in XtreemOS Grid OS and security architecture

Fundamentals in Security and VO Management

- Basics on security
- Virtual organization concept

Security

- **What is computer security?**
  - Computer security deals with the **prevention** and **detection** of **unauthorised actions** by user of a computer system

- **Why is security important in Grids?**
  - Grids are open distributed systems
  - Opening our systems to others implies **security risks**

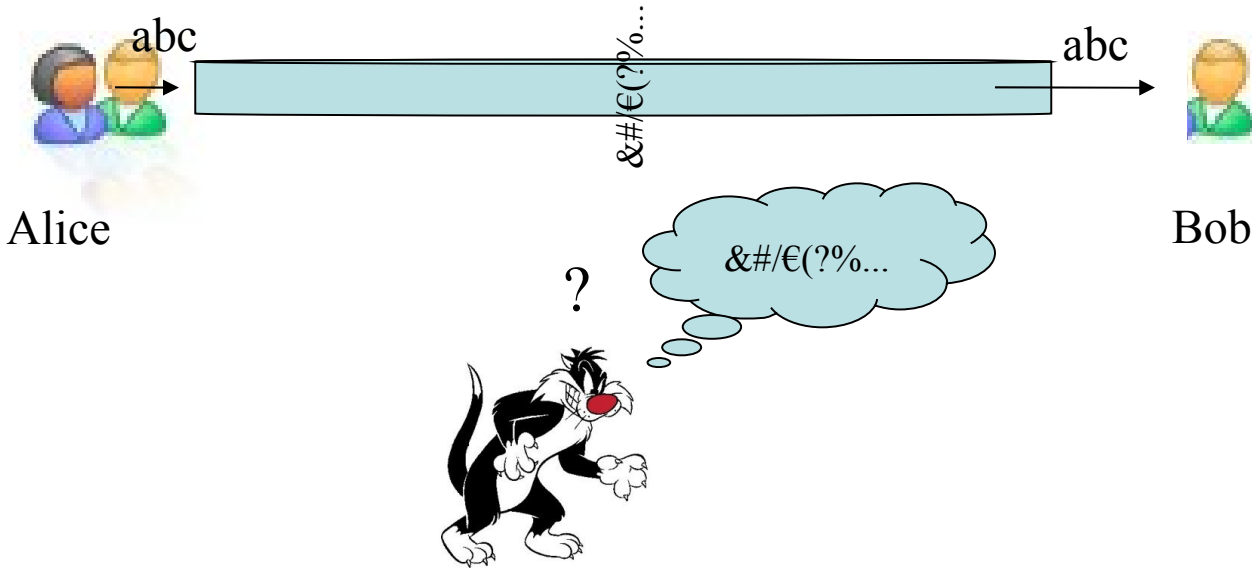
Basic Security Concepts

- **Authentication.** Assurance of identity of person or originator of data
- **Authorisation.** Being allowed to perform a particular action
- **Integrity.** Preventing tampering of data

- **Availability**: Legitimate users have access when they need it
- **Non-repudation**: Originator of communications can't deny it later
- **Confidentiality:** Protection from disclosure to unauthorised persons
- **Auditing**: Provide information for post-mortem analysis of security related events

Security fundamentals

Confidentiality - only invited to understand conversation (use encryption)

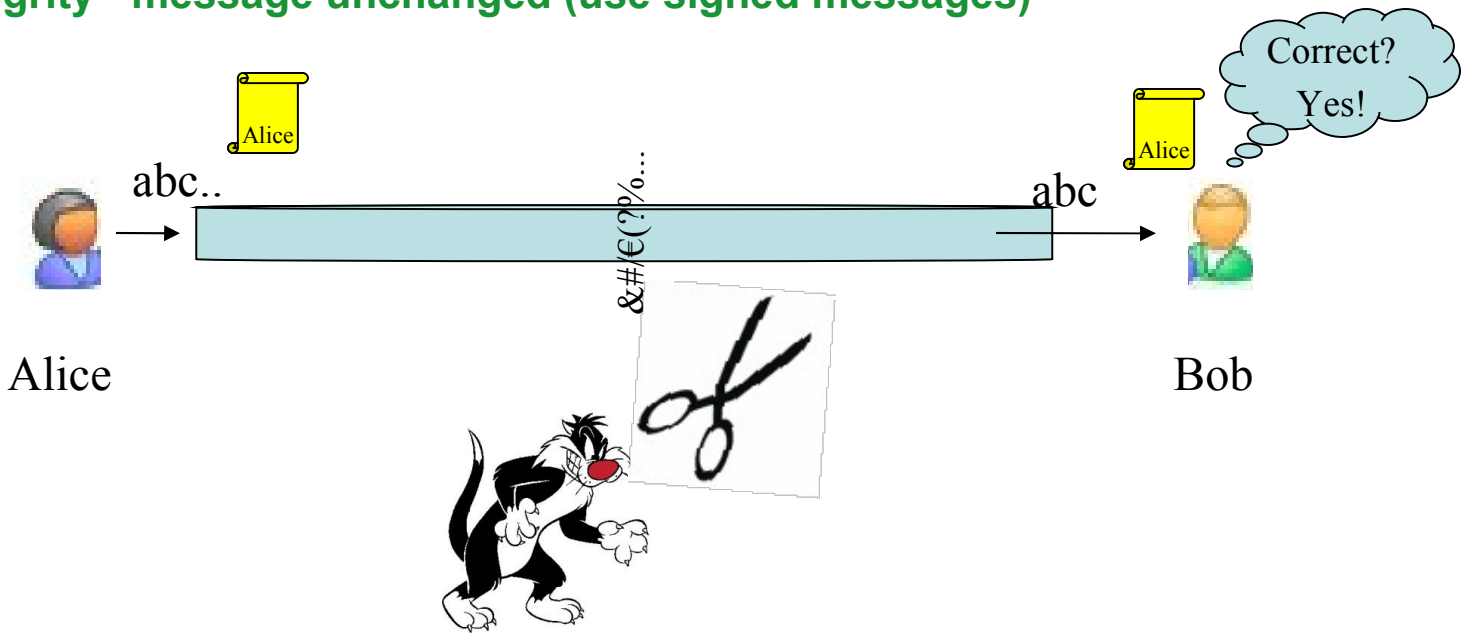


Confidentiality (privacy) - A secure conversation should be private. In other words, only the sender and the receiver should be able to understand the conversation. If someone eavesdrops on the communication, the eavesdropper should be unable to make any sense out of it.

(This is generally achieved by encryption/decryption algorithms.)

Security fundamentals

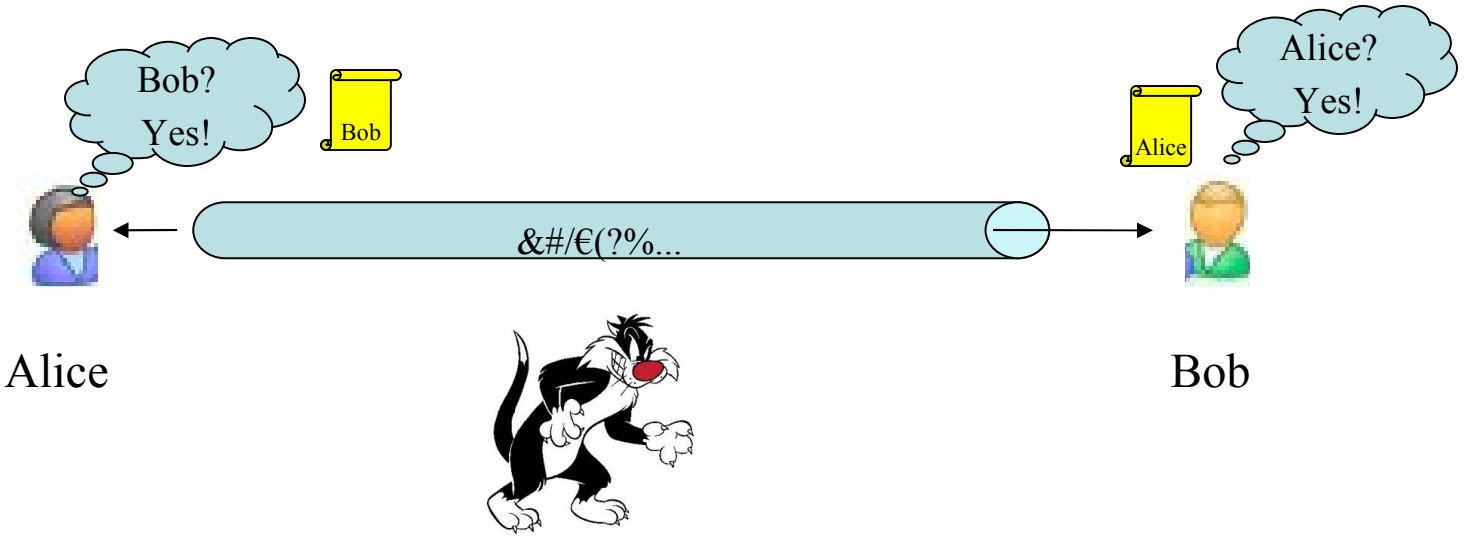
Integrity - message unchanged (use signed messages)



Integrity - A secure communication should ensure the integrity of the transmitted message. This means that the receiving end must be able to know for sure that the message he is receiving is exactly the one that the transmitting end sent him. Take into account that a malicious user could intercept a communication with the intent of modifying its contents, not with the intent of eavesdropping.

Security fundamentals

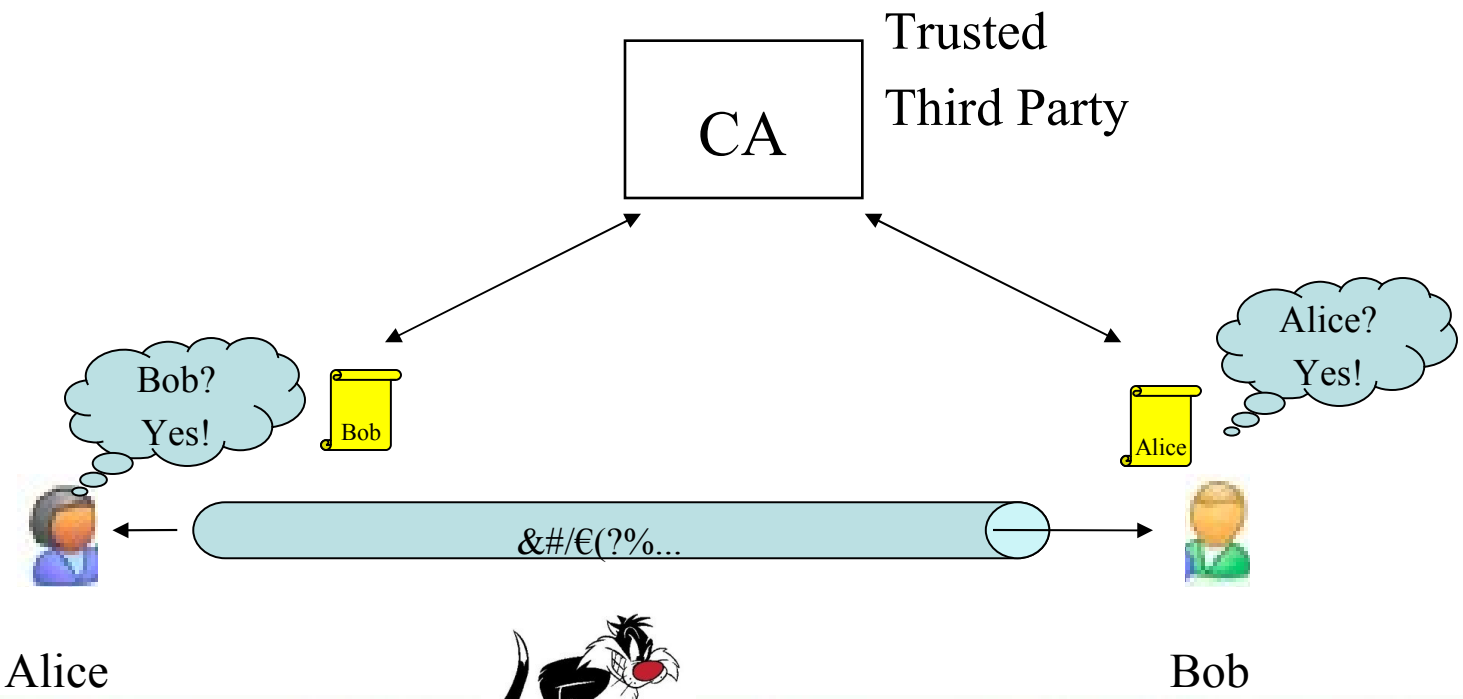
Authentication - invited are who they claim to be (use certificates and CAs)



AuthN - A secure communication should ensure that the parties involved in the communication are who they claim to be. In other words, we should be protected from malicious users who try to impersonate one of the parties in the secure conversation.

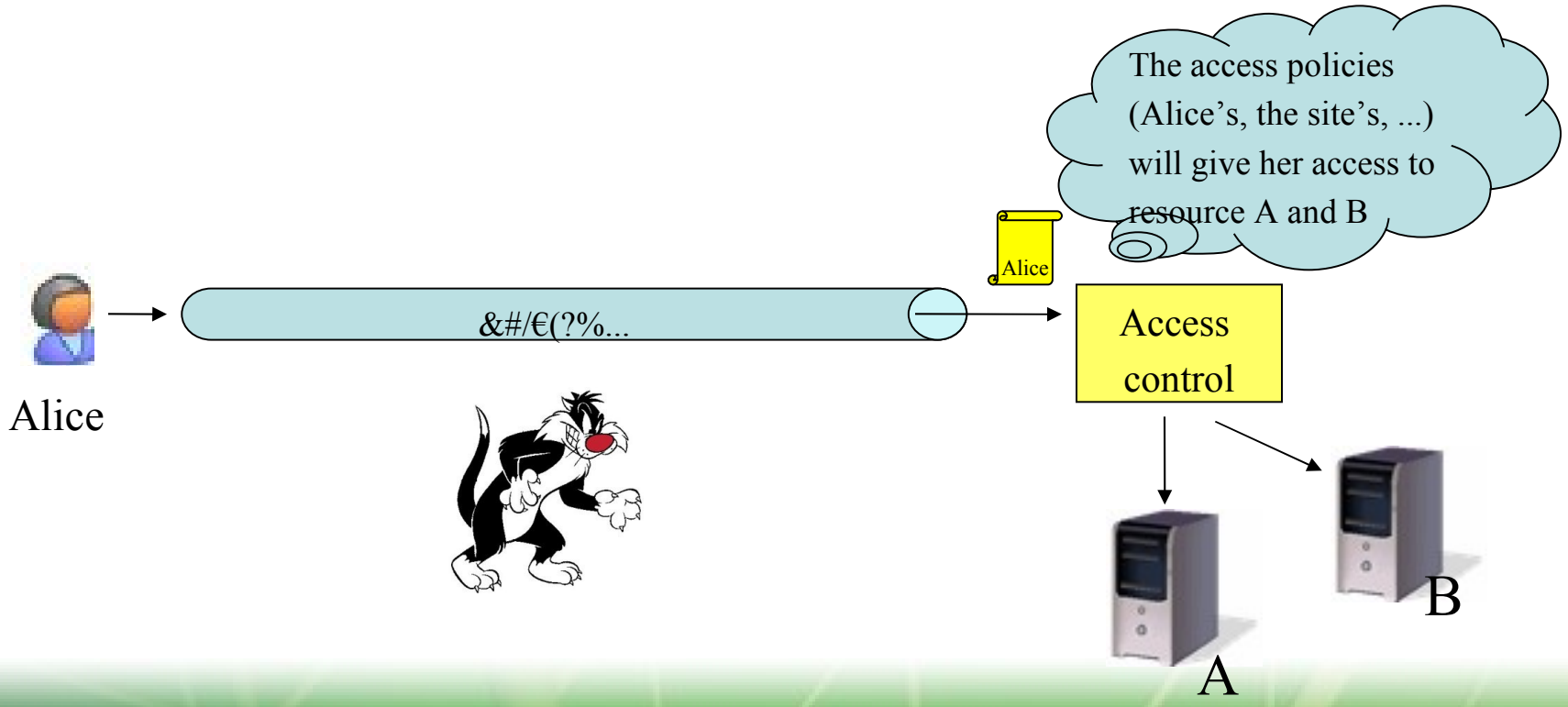
CA - Certification Authority

- The role of the CA is manage the certificate life cycle: create, store, renew, revoke



Security fundamentals

Authorization - allowing or denying access to services based on policies



Security fundamentals

To be able to analyse the communication we also need auditing providing information for post-morten analysis of security related events...

A common way to organize these concepts is ’AAA’ - Authentication, Authorization and Auditing.

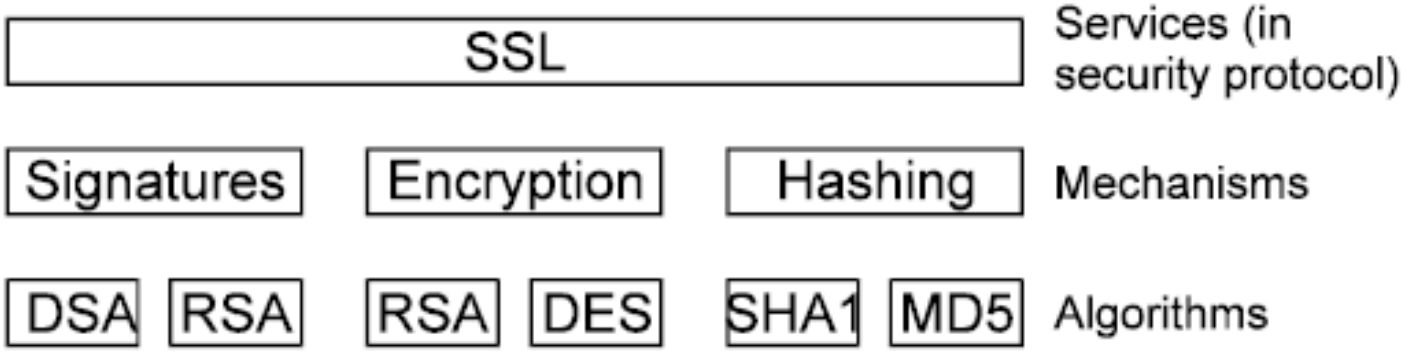
- enable the identification (Authentication) of entities (users, systems, and services),
- allow or deny access to services and resources (Authorization),
- and provide information for post-mortem analysis of security related events (Auditing).

Security Mechanisms

- **Three basic building blocks are used:**
  - **Encryption** is used to provide confidentiality, can also provide authentication and integrity protection
  - **Digital signatures** are used to provide authentication, integrity protection, and non-repudiation
  - **Checksums/hash algorithms** are used to provide integrity protection, can provide authentication
- **One or more security mechanisms are combined to provide a security service**

Security Services and Mechanisms

- A typical security protocol provides one or more services



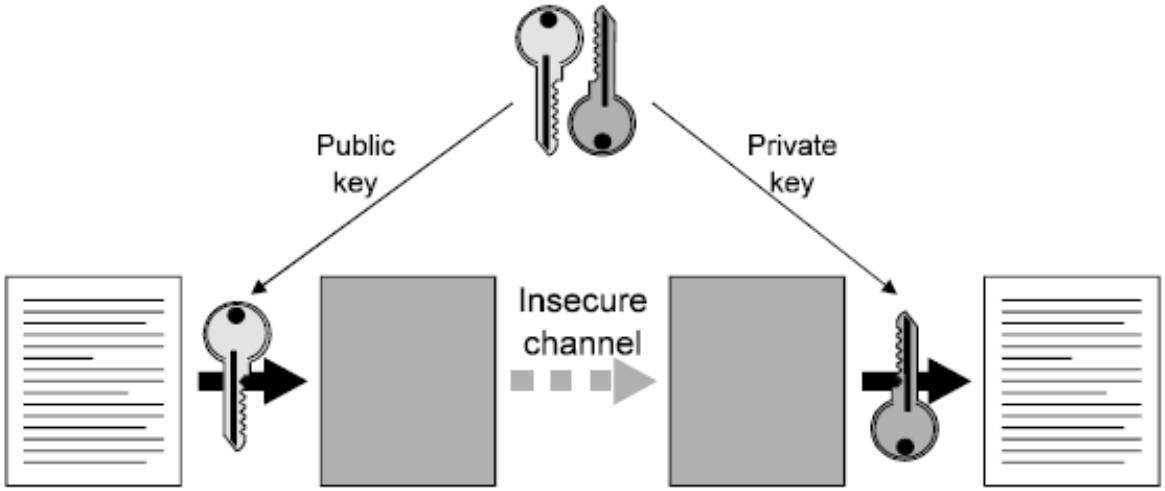
- Services are built from mechanisms
- Mechanisms are implemented using algorithms

Trust through Cryptography

- **An entity uses computer programs to cryptographically verify the information given**
  - If everything is ok, then trust of the information is established
  - Otherwise, there is not trust

Public-Key Encryption

- Users possess public/private key pairs



- Anyone can encrypt with the public key, only one person can decrypt with the private key

Key Management

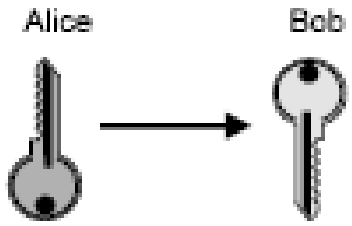
- **Key management is the hardest part of cryptography**
- **Two classes of keys**
  - Short-term session keys
    - Generated automatically and invisibly
    - Used for one message or session and discarded
  - Long-term keys
    - Generated explicitly by the user
- **Long-term keys are used for two purposes**
  - Authentication (including access control, integrity, and non-repudiation)
  - Confidentiality (encryption)
    - Establish session keys
    - Protect stored data

Key Management Problems

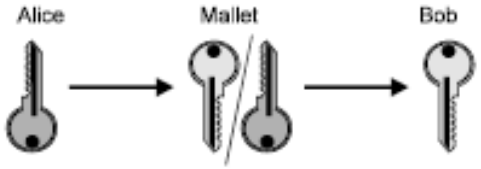
- **Key certification**
- **Distributing keys**
  - Obtaining someone else's public key
  - Distributing your own public key
- **Establishing a shared key with another party**
  - Confidentiality: Is it really known only to the other party?
  - Authentication: Is it really shared with the intended party?
- **Key storage**
  - Secure storage of keys
- **Revocation**
  - Revoking published keys
  - Determining whether a published key is still valid

Key Distribution Problems

- Alice retains the private key and sends the public key to Bob



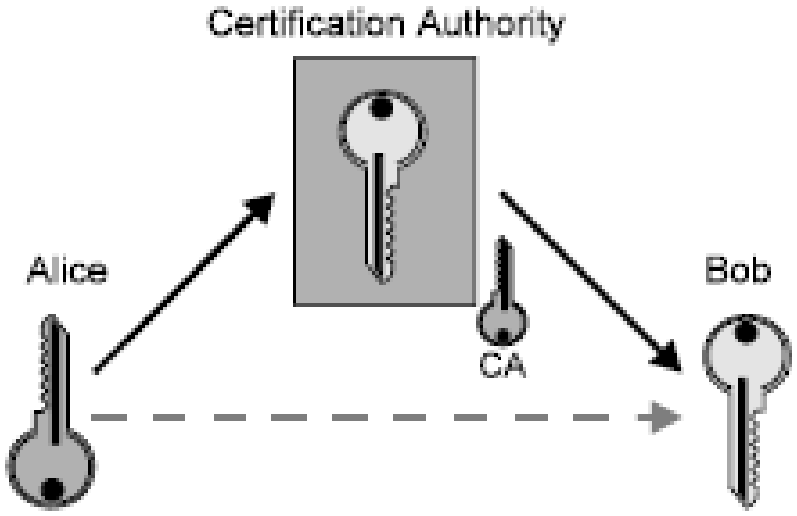
- Mallet intercepts the key and substitutes his own key



- Mallet can decrypt all traffic and generate fake signed message

Certification Authority

- **A Certification Authority (CA) solves this problem**



- **CA signs Alice's key to guarantee its authenticity to Bob**
  - Mallet can't substitute his key since the CA won't sign it

Certification Authorities (CAs)

- **CAs are entities that are trusted by different systems**
- **The CAs are responsible for certifying the public keys of different users who subscribe to the CA**
  - Guarantee the connection between a key and an end entity
- **An end entity is**
  - Person, role (“Director of marketing”), organisation, pseudonym, a piece of hardware or software, an account (bank or credit card)
- **CA manages key lifecycle: creation, store, delete, renew**

Obtaining a Certificate (1)

1. **Alice generates a key pair and signs the public key and identification information with the private key**
   - Proves that Alice holds the private key corresponding to the public key
   - Protects the public key and ID information while in transit to the CA
2. **CA verifies Alice's signature on the key and ID information**
3. **Optional: CA verifies Alice's ID through out-of-band means**
   - email/phone callback
   - Business/credit bureau records, in-house records

Obtaining a Certificate (2)

4. **CA signs the public key and ID with the CA key, creating a certificate**
   - CA has certified the binding between the key and ID

5. **Alice verifies the key, ID, and CA's signature**
   - Ensures the CA didn't alter the key or ID
   - Protects the certificate in transit

6. **Alice and/or the CA publish the certificate**

Public Key Infrastructure (PKI)

- **PKI allows one to know that a given key belongs to a given user**
  - Based on asymmetric encryption
- **The public key is given to the world encapsulated in a X.509 certificate**
- **Certificates: Similar to passport or driver license**
  - Identity signed by a trusted party (a CA)

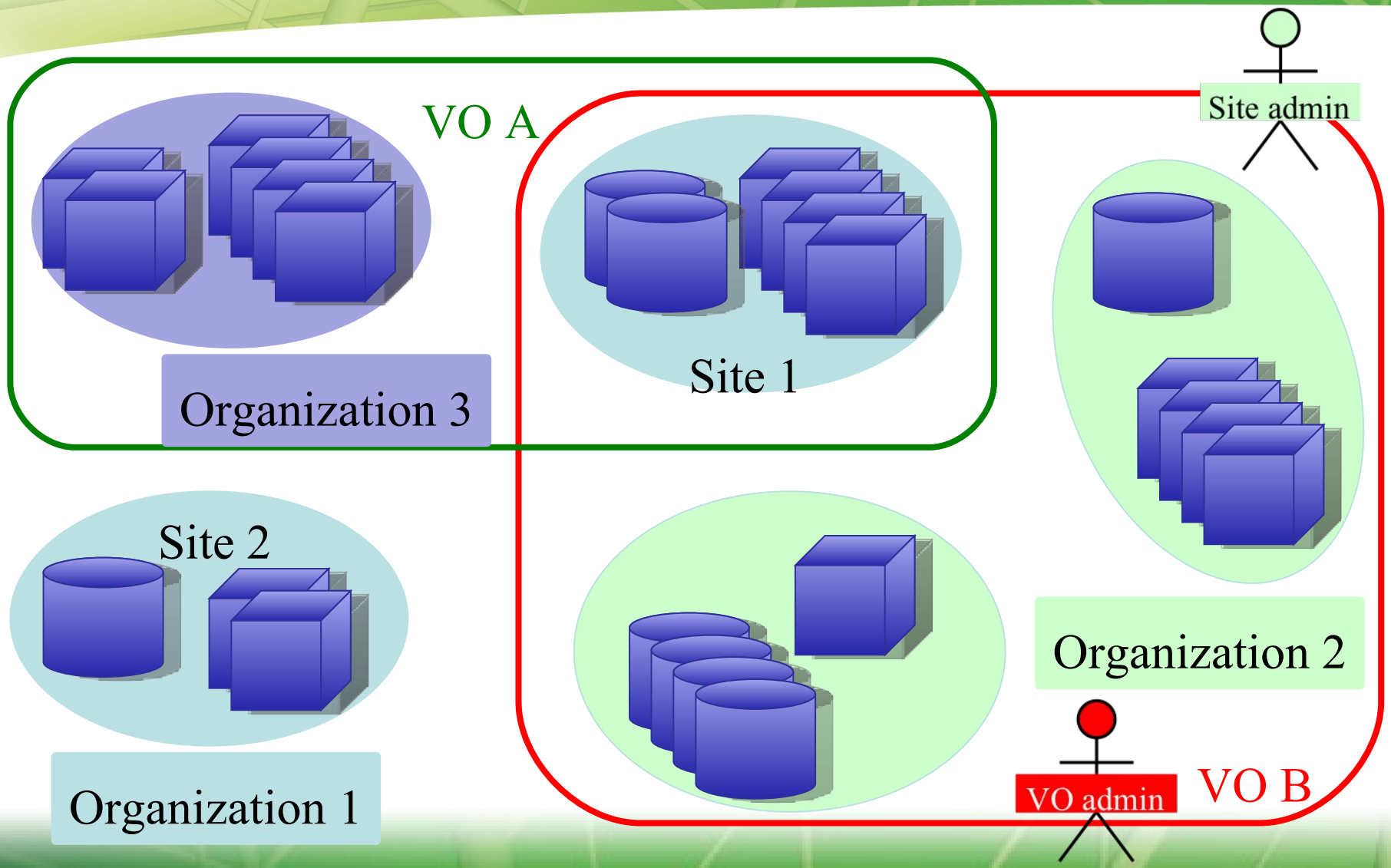
Grid

"A fully distributed, dynamically reconfigurable, scalable and autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources (computing power, storage, instruments, data, etc.) in order to generate knowledge"

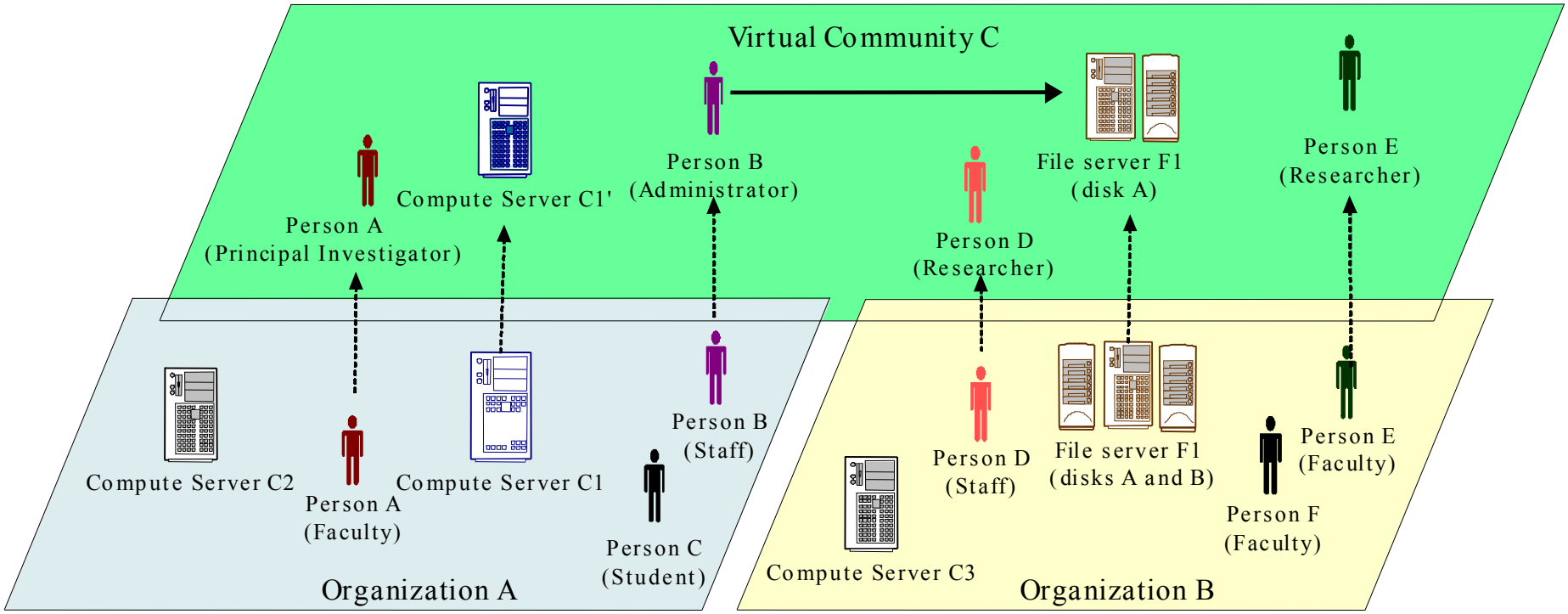
Virtual Organization (VO)

- VO = set of users that pool resources in order to achieve common goals - Rules governing the sharing of the resources
- A VO can be seen as a distributed organization which has the task of managing access to resources that are accessed through computer network and located in different domains

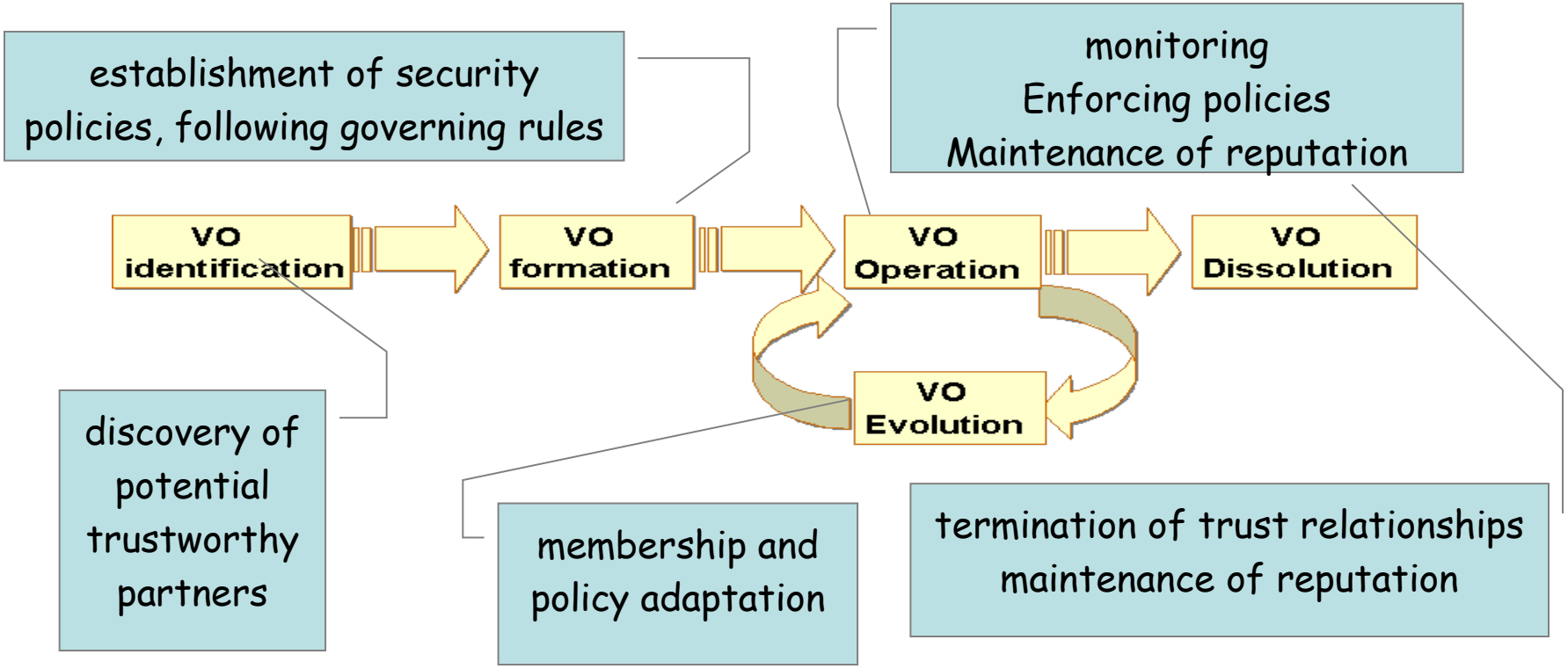
Virtual Organizations



Virtual vs. Organic structure



VO Lifecycle

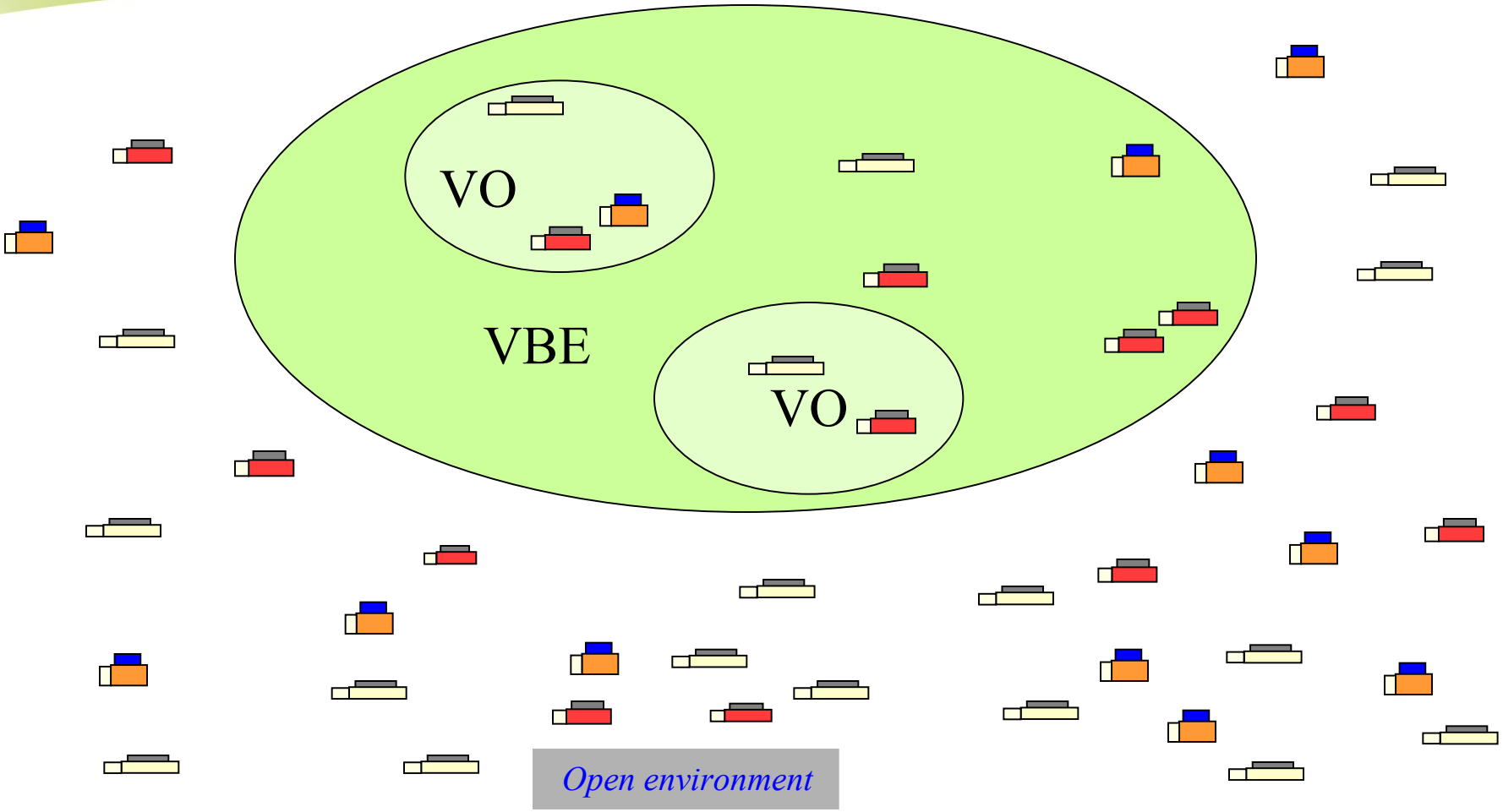


Virtual Breeding Environment

- VO are created in the context of a Virtual Breeding Environment (VBE)

- A Virtual Breeding Environment is composed of users and service providers. It provides user and service provider registration, certificate management, and VO lifecycle management.

VBE & VO



Actors

- VBE administrator
- VO administrator
- Domain/site administrators
- End-users – VO members

XtreemOS

Enabling Linux for the Grid

ICS'09

Tutorial on Security and Virtual Organization Management in Grids

PART 2 - Security and VO Management in Grids

Information Society Technologies

XtreemOS IP project is funded by the European Commission under contract IST-FP6-033576

Outline

- **Grid security & VO management overview**
  - Grid security essentials
  - Establishing trust, policies
  - Single sign on and delegation
  - Authorization
  - Monitoring - logging, auditing and accounting
- **Real-life examples**
  - Globus Toolkit
  - EGEE/gLite
  - Unicore

Grid security & VO management overview

- Grid security essentials
- Establishing trust, policies
- Single sign on and delegation
- Authorization
- Monitoring - logging, auditing and accounting

Requirements for Grid Security

- **Access to shared services**
  - cross-domain authentication, authorization, accounting, billing
- **Support multi-user collaboration**
  - organized in one or more ‘Virtual Organisations’
  - may contain individuals acting alone – their home organization administration need not necessarily know about all activities
- **Leave resource owner always in control**

Issues in making Grid security work

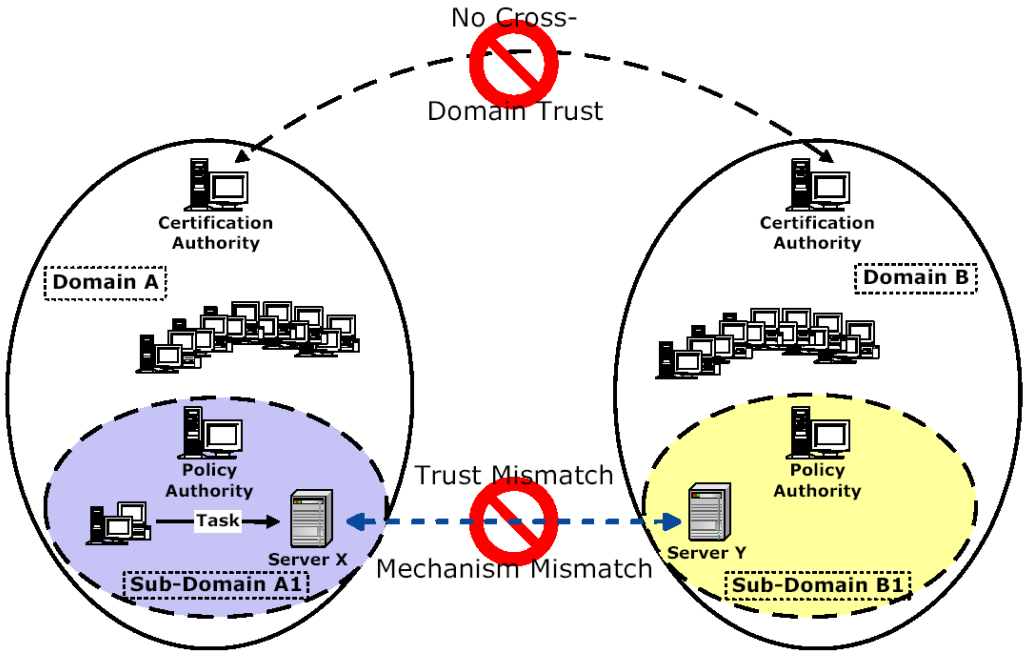
- **Resources may be valuable & the problems being solved sensitive**
  - Both users and resources need to be careful
- **Resources & users often located in distinct administrative domains**
  - Can't assume cross-organizational trust agreements
  - Different mechanisms & credentials
- **Dynamic formation and management of communities (VOs)**
  - Large, dynamic, unpredictable, self-managed …
- **Interactions are not just client-server, but service-to-service on behalf of the user**
  - Requires delegation of rights by user to service
- **Policy from sites, VO, users need to be combined**
  - Varying formats
- **Want to hide as much as possible from applications!**

GSI – Grid Security Infrastructure

- A reference specification for Grid security architectures
- Protocols and APIs to address Grid security needs
- Based on public-key encryption technology
  - SSL protocol for authentication, message protection
  - X.509 certificates
- Each user as a Grid id, a private key, and a certificate signed by a CA
- First implementation – in the Globus Toolkit

Establishing trust

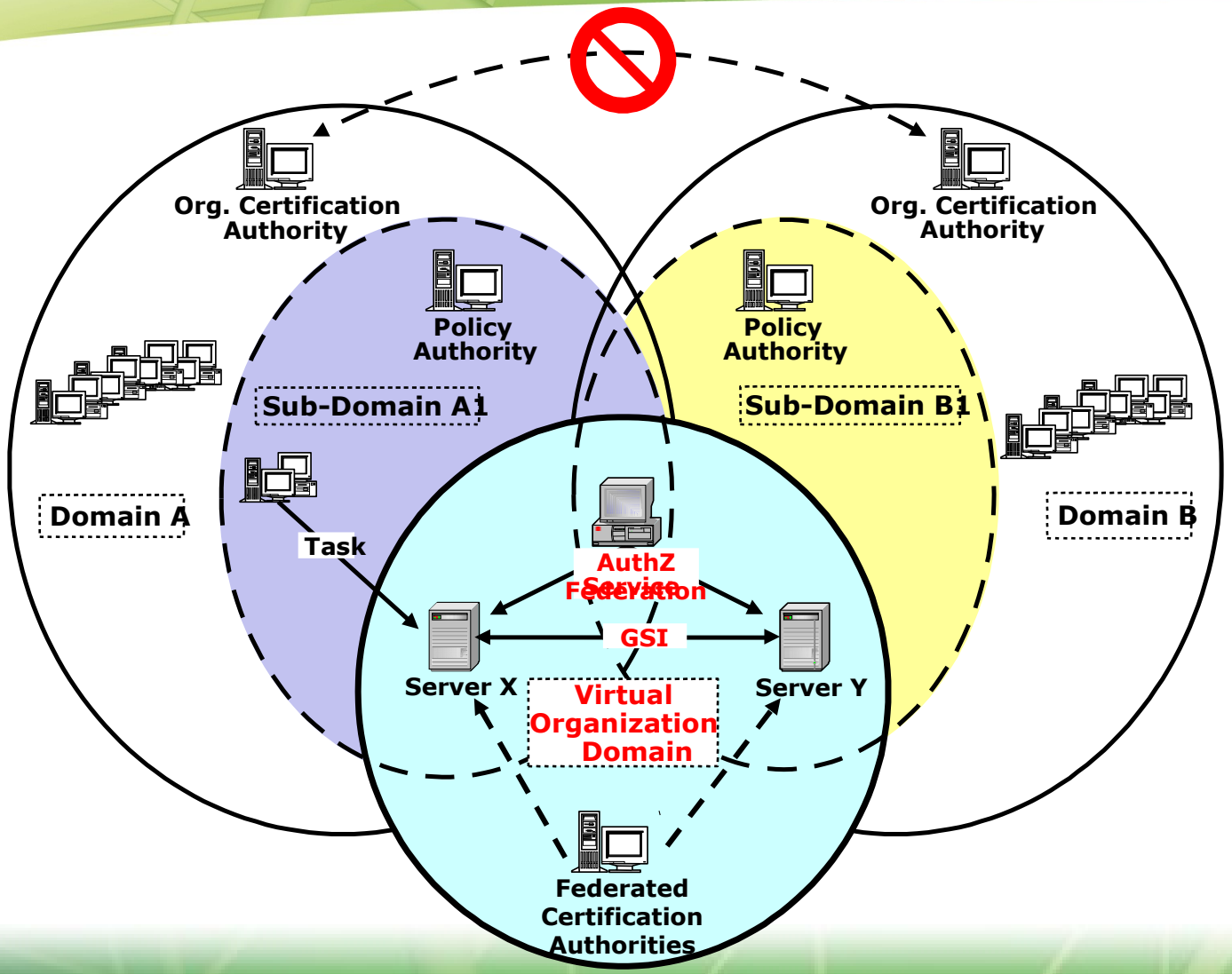
- It is the dynamic cross-organizational resource sharing that gives us a problem
- VOs are user-to-user, not organization-to-organization
- No trust, different policies, different mechanisms



Solving the trust problem

- **Trusted Third Parties**
  - Independent identity assessment providers
  - The most commonly used today – e.g., Certificate Authorities
  - Example: www.gridpma.org
- **Federations**
  - Organizations trust each other to identify their own users
- **Web of trust**
  - Users trust each other to do identify others

Certification Authorities (CAs) in Grid

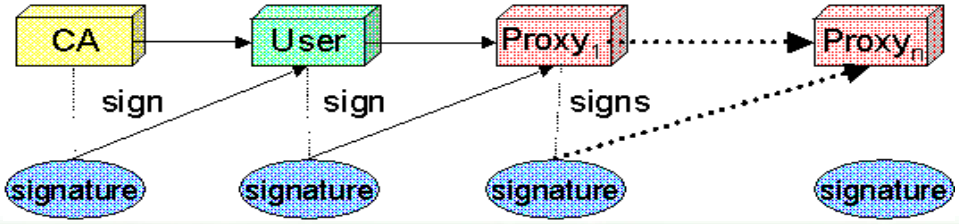


Single sign-on and delegation

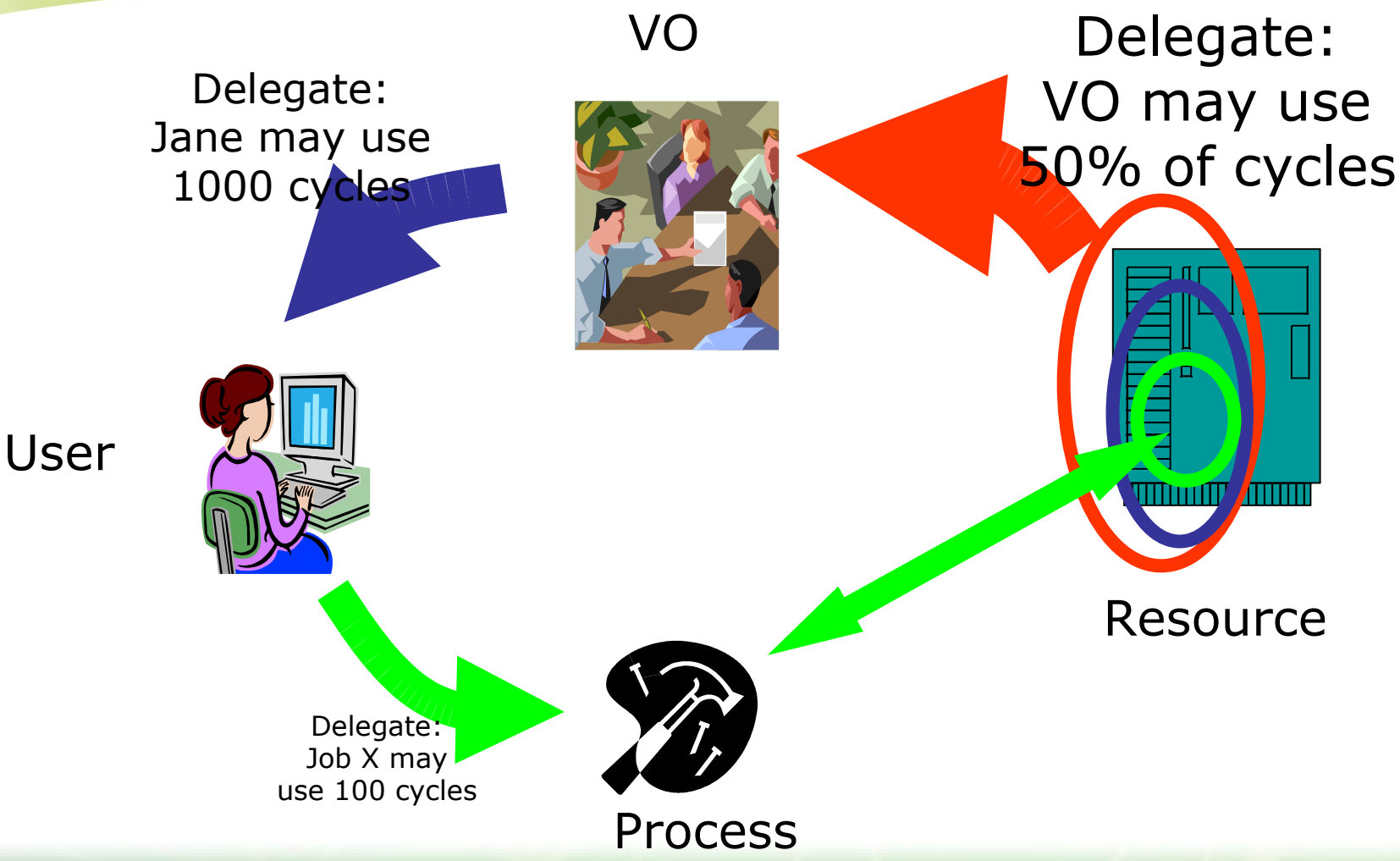
- **Jobs require access to multiple resources**
  - To authenticate with your certificate directly you would have to type a passphrase every time
- **Need to automate access to other resources: Authenticate Once**
  - Important for complex applications that need to use Grid resources
  - Allows remote processes and resources to act on user's behalf - also known as **delegation**
  - Also you need a way to send you VO details (Groups membership, roles and capabilities) across
- **Solution adopted in the Grid Security Infrastructure: *proxy certificates***
  - A temporary key pair
  - in a temporary certificate signed by your 'long term' private key
  - valid for a limited time (default: 12 hours), but can be renewed

Delegation and limited proxy

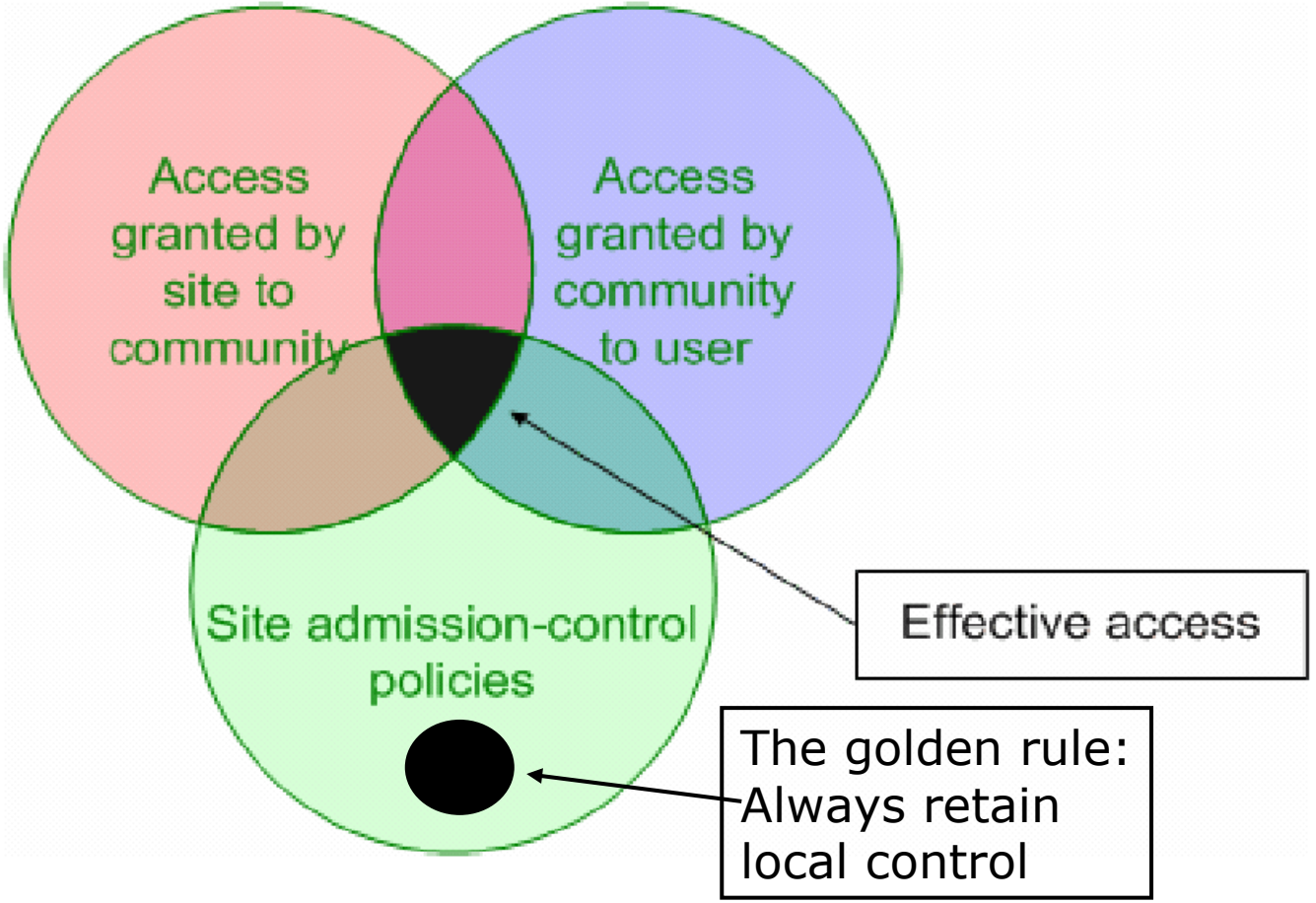
- **Delegation = remote creation of a (second level) proxy credential**
  - Agents and brokers act on behalf of users, with (a subset of) their rights
  - you don't know beforehand where your task will end up
  - definition of attribute release policies to these 'unknown' entities is virtually impossible
  - need to support restricted delegation
- **Allows remote process to authenticate on behalf of the user**
- **The client can elect to delegate a "limited proxy"**
  - Each service decides whether it will allow authentication with a limited proxy
  - The proxy can also be used as a container for other elements (e.g. extensions that contain user credentials)



Authorization

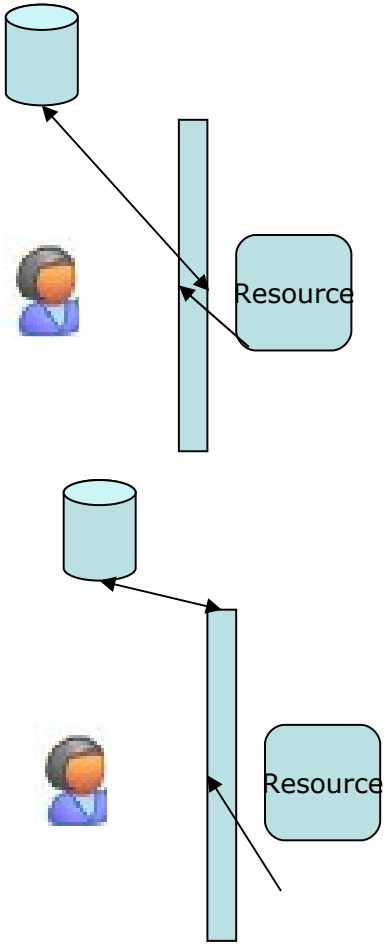


Policies for accessing resources



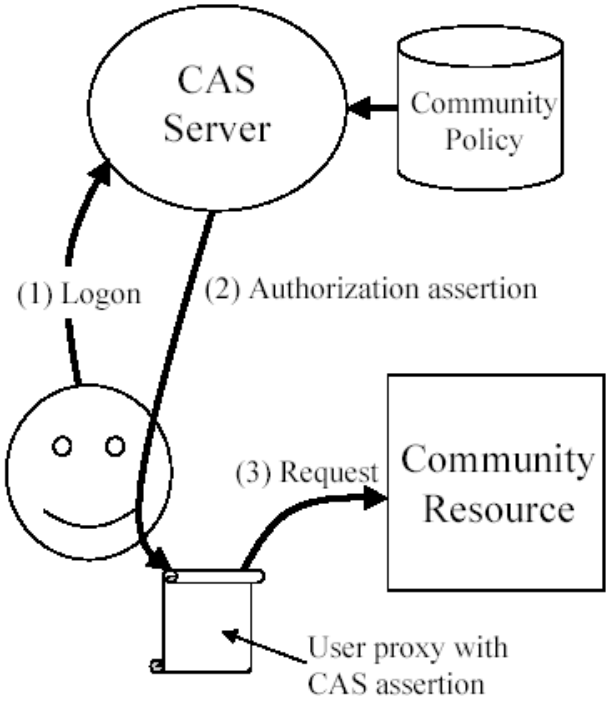
Authorization to a resource - alternatives

- **Push Authorization**
  - Produce a proof (proxy certificate) that you are authorized to use the requested resource
  - Bring (push) this proof to an access control point, who will make sure the proof is valid
- **Pull Authorization**
  - Go to the access control point and ask for access (just showing who you are, showing your ID, nothing about what you're authorized to do).
  - The access controller uses your ID to pull the access policies from a database.
  - Depending on the access policies, you're authorized to run your program on the resources, or parts of the resources, or not at all.



CAS – Community Authorization Service

- **CAS manages a data base of VO policies**
  - What each grid user can do as VO member
- **A Grid user contacts CAS**
  - Proxy cert. is exploited for authentication on CAS
  - CAS returns a signed policy assertion for the user
- **Grid user creates a new proxy that embeds the CAS assertion**
- **Exploits this proxy certificate to access services**

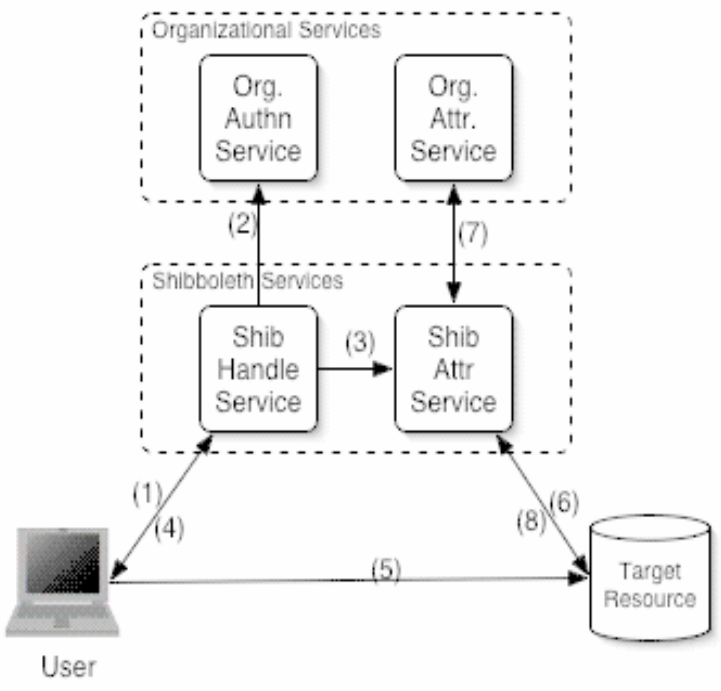


VOMS

- **VOMS = Virtual Organization Membership Service**
  - Developed by the EU DataGrid and DataTag projects
- **Provides a way to delegate the authorization of users to VO managers:**
  - The user credentials are associated with a set of membership information (VO name, group, roles, generic attributes)
  - The information is stored in an account database
  - The VOMS service can provide signed assertions containing these attributes
- **VOMS allows for dynamic & fine-grained access control on Grid resources**

Shibboleth

- **Attribute Authority Service for distributed cross domain environments**
  - User authentication is done on a local Shibboleth server that returns a handle to the user
  - Users use the handle to access remote services
  - Remote services use the user handle to retrieve user's attributes from a Shibboleth Attribute Server
  - Remote Service determines user access rights exploiting his attributes



Monitoring – logging, auditing and accounting

- **Important for security handling (and not only)**
- **Auditing**
  - uses information recorded (logged) about system activity for the purposes of accountability and security assurance
- **Logging**
  - a common infrastructure for the recording of system events for tracking, accountability and auditing purposes
- **Accounting**
  - All relevant system interactions can be traced back to a user

Outline

Real-life examples

- Globus Toolkit
- EGEE/gLite
- Unicore

Example #1: Globus Toolkit (GT)

- **Open source middleware for computing grids**
- **Has evolved to an implementation based on web services**
  - implements the Open Grid Services Architecture (OGSA) and the Web Services Resource Framework (WSRF)
  - includes components that provide resource management, data management, security, information infrastructure, communication, fault detection etc.
- **Probably the most widely used Grid middleware**
- **Included in other Grid software stacks**
  - OSG
  - LCG

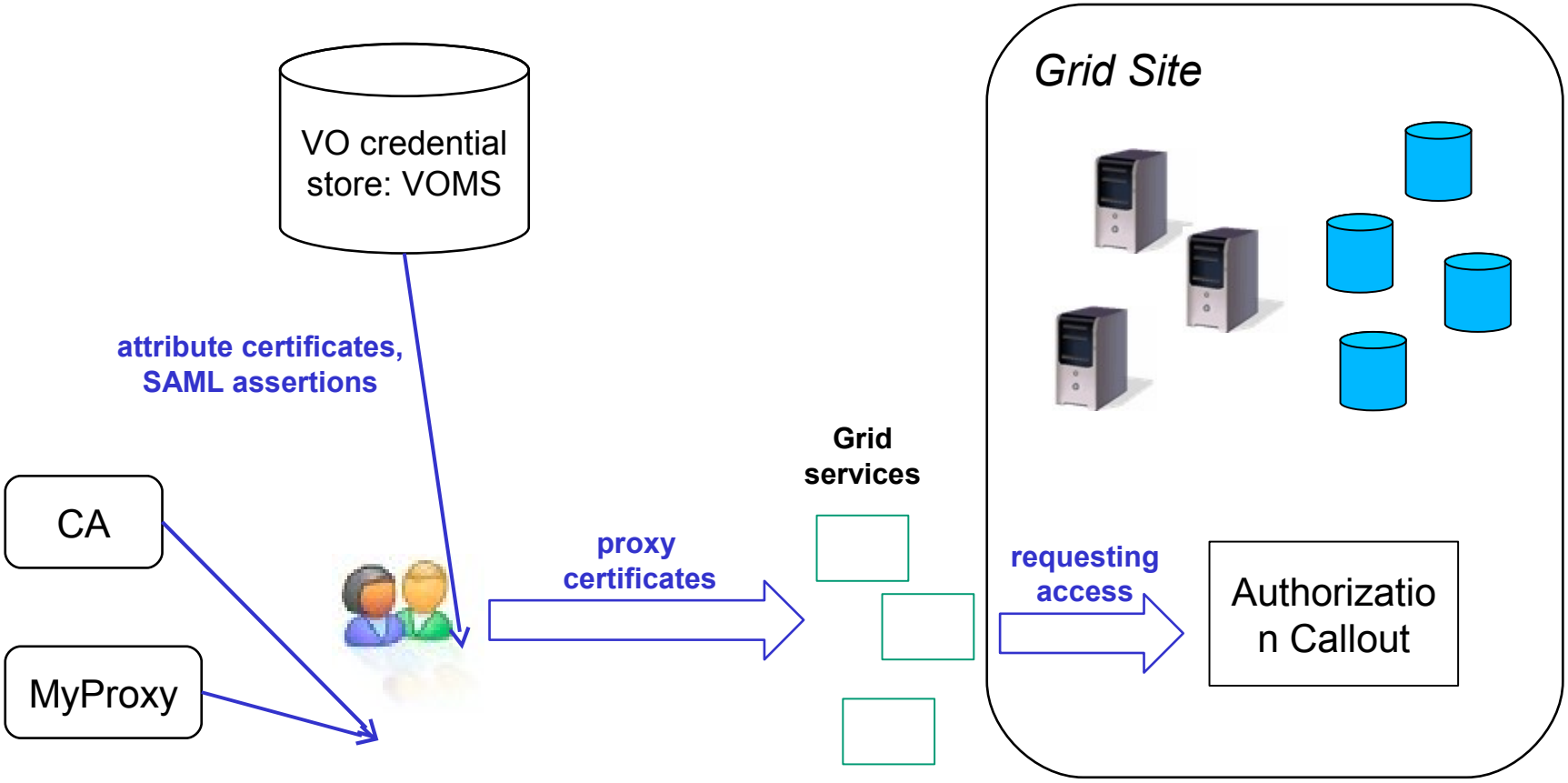
Globus Toolkit – Security Overview (1)

- **Implements the Grid Security Infrastructure (GSI)**
- **X.509 proxy certificates**
  - Enable single sign-on
  - The users can dynamically assign rights to services
- **MyProxy – storing and retrieving GSI credentials**
  - “convert” from username/passphrase to a GSI certificates
  - Renewing credentials for long-running tasks
  - Support for One Time Password

Globus Toolkit – Security Overview (2)

- **GridShib – GT integration with Shibboleth**
  - Policy controlled attribute service
  - Interactions through WS protocols
- **Authorization – many types of policy information:**
  - Attribute assertions: VOMS, X509, Permis, Shibboleth, SAML, Kerberos, …
  - Authorization assertions: XACML, SAML, CAS, XCAP, Permis, …
- **Authorization processing**
  - Policy Decision Point (PDP) abstraction
  - after validation, all attribute assertions are mapped to XACML Request Context Attribute format
  - mechanism-specific PDP instances are created for each authorization assertion and call-out service

Globus Toolkit – Security flow



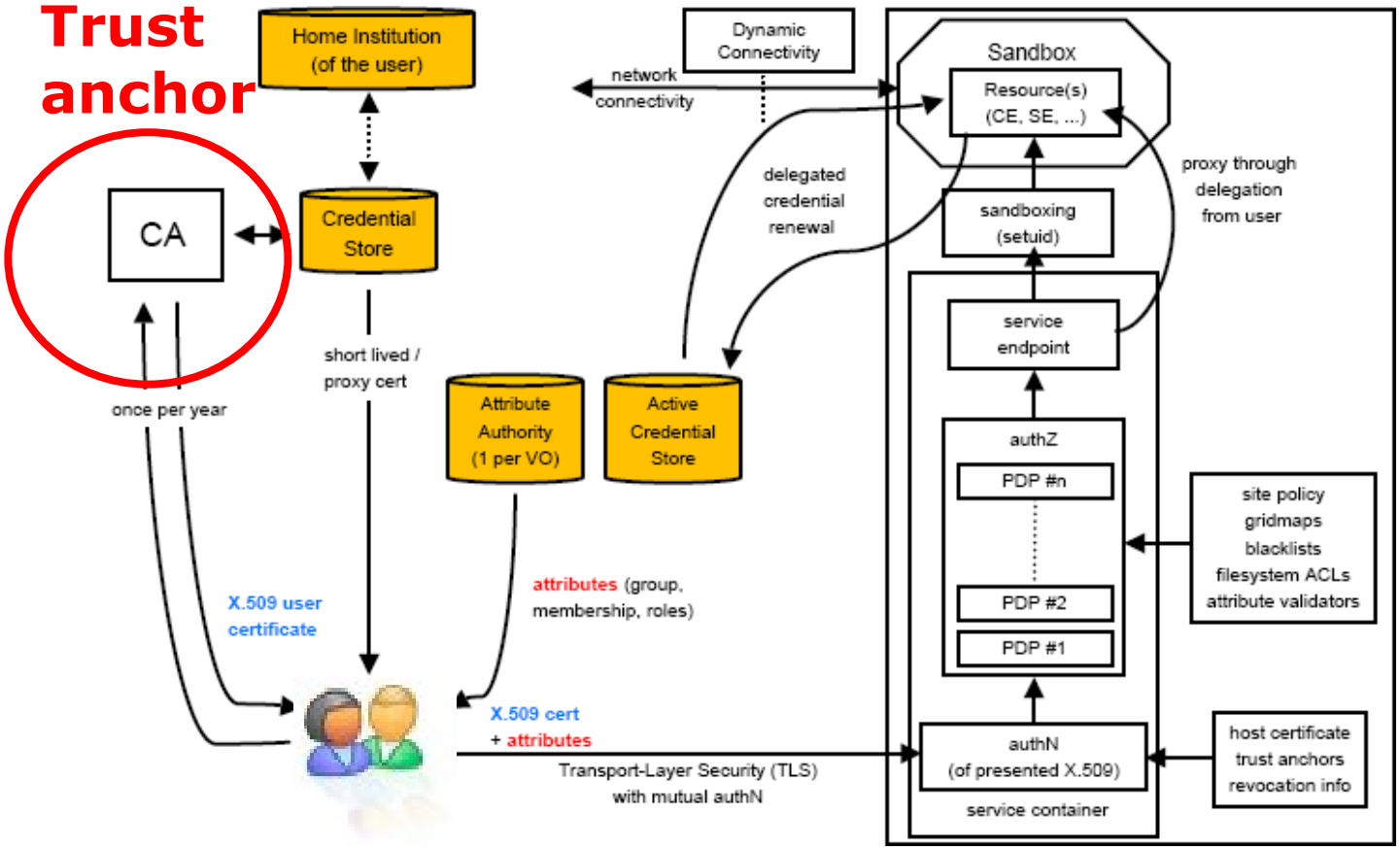
Example #2: gLite

- gLite: Grid middleware developed at CERN, in the context of the LHC experiments
- Used by more than 15000 researchers around the world
- gLite components:
  - User Interface (UI)
  - Computing Element (CE)
  - Storage Element (SE)
  - Resource Broker (RB)
  - Information Service (IS)

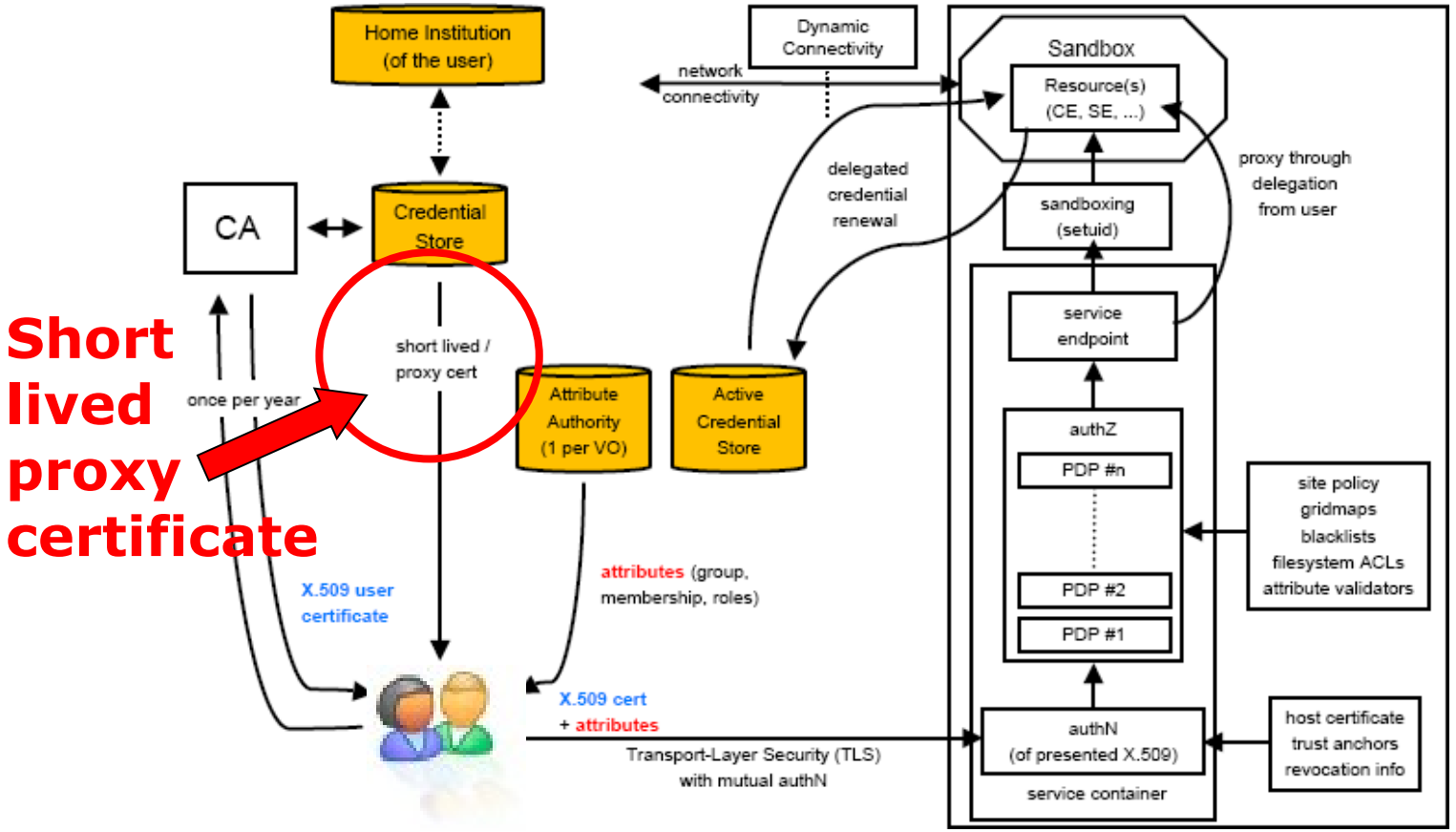
Security in gLite - Overview

- Security system based on X.509 certificates
- Single sign-on enabled by proxy certificates
- VOMS service used to stored information about groups, roles and capabilities for the users
- Local Centre Authorization Service (LCAS)
  - Checks if the user is authorized or banned at the site
  - And if the site can currently accept jobs
- Local Credential Mapping Service (LCMAPS)
  - Maps the Grid credentials (including groups, roles etc.) to local credentials

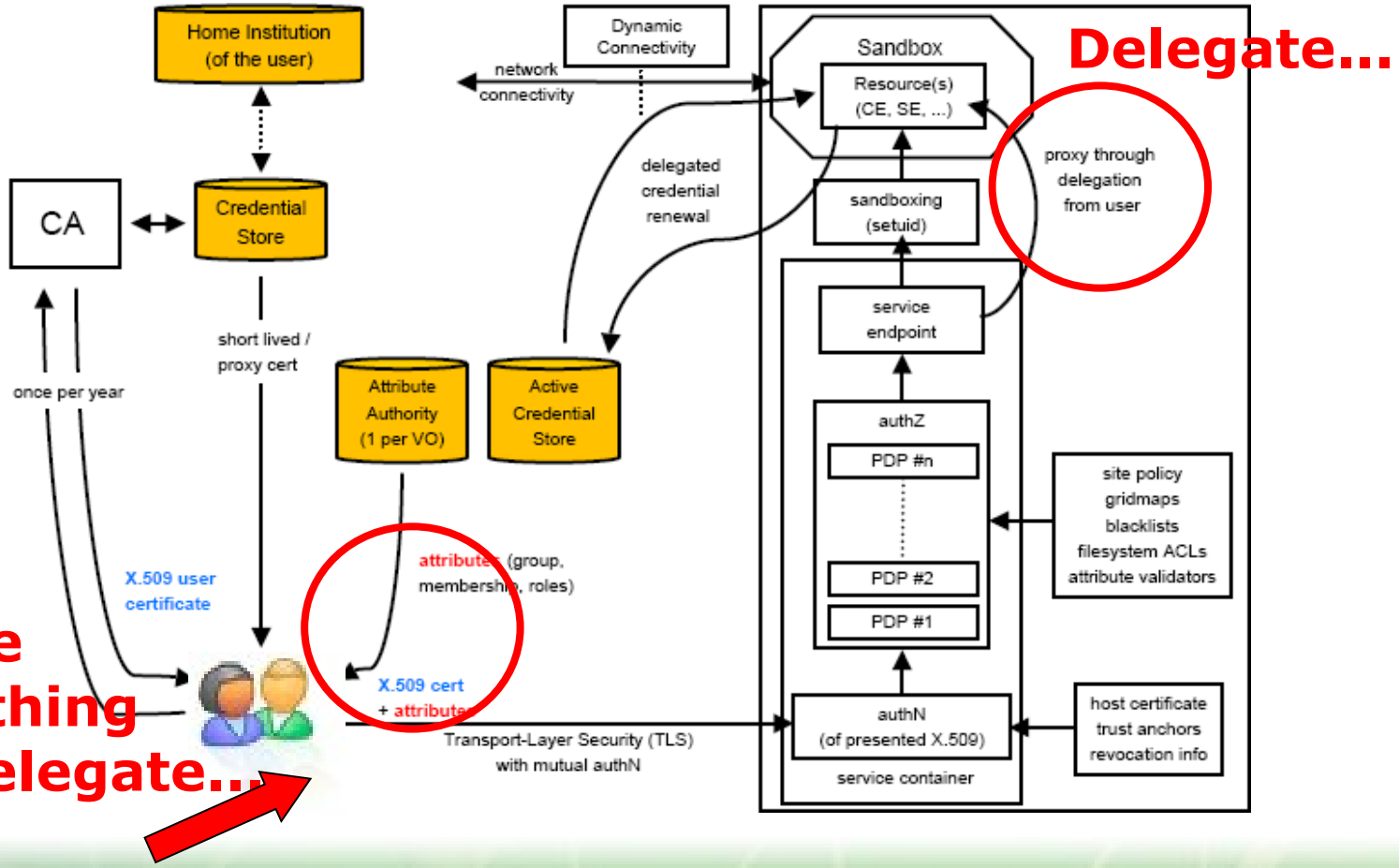
gLite – Security flow (1)



gLite – Security flow (2)

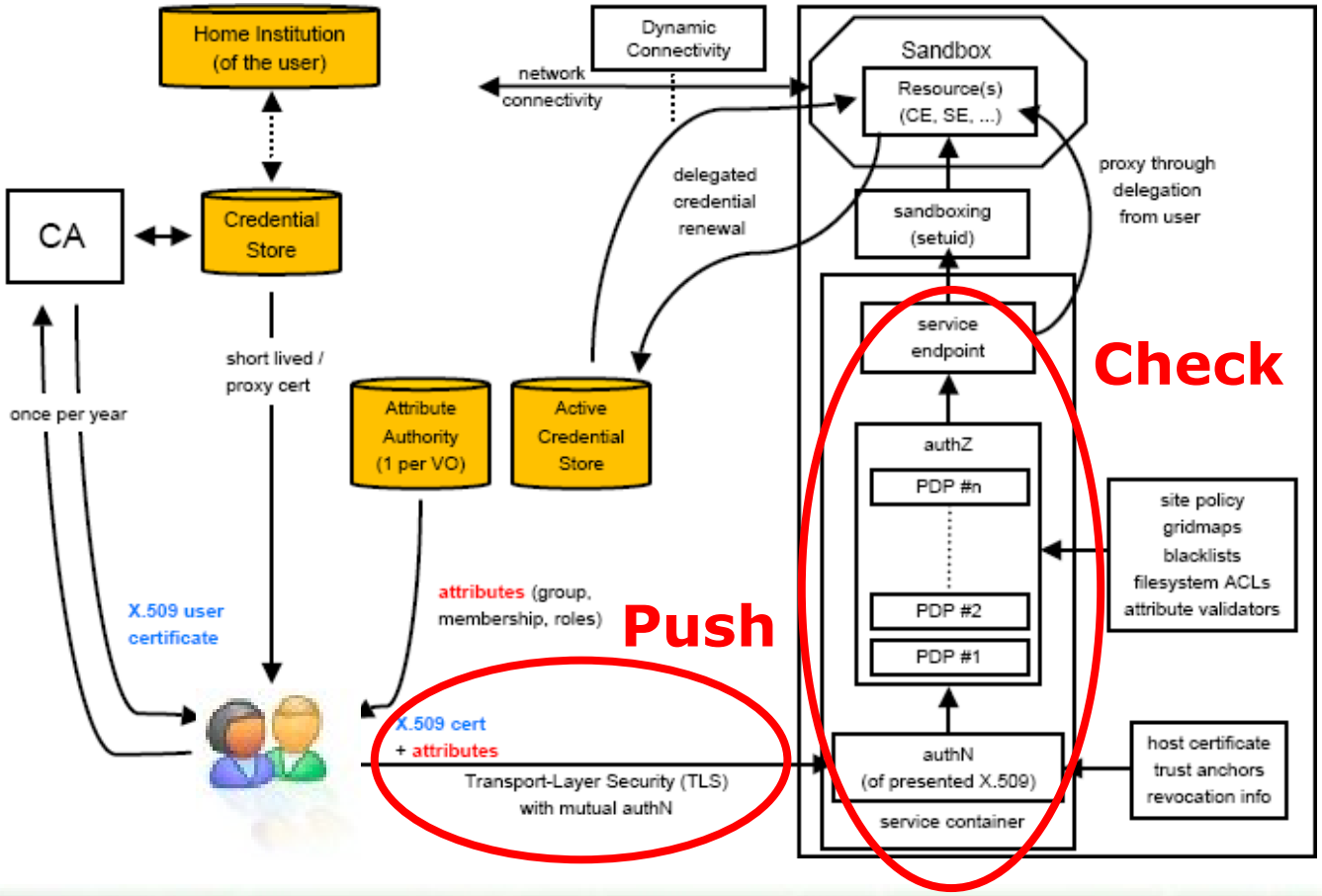


gLite – Security flow (3)

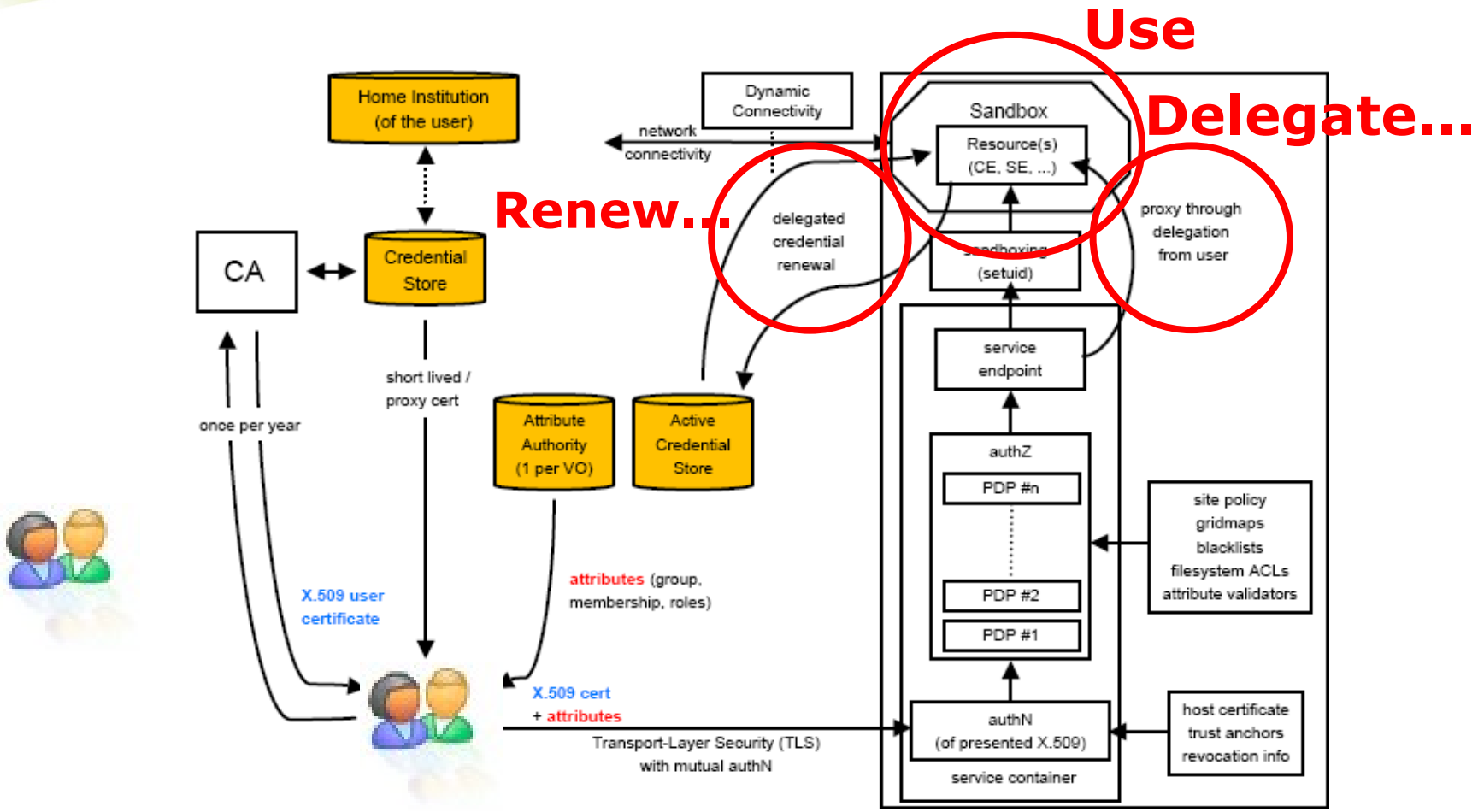


Bundle everything and delegate...

gLite – Security flow (4)



gLite – Security flow (5)



Example #3: UNICORE

- **Grid middleware used by many European research projects**
  - DEISA (Distributed European Infrastructure for Scientific Applications) uses the UNICORE technology
- **UNICORE layers:**
  - Client: graphical interfaces, command line, APIs
    - *The UNICORE services can also be accessed through portals (e.g. GridSphere)*
  - Service: components of the Unicore Service Oriented Architecture
    - *Gateway – entry point to a Unicore site*
    - *NJS – job management & execution engine*
    - *Global service registry*
    - *…*
  - System: interface between Unicore and the local resource management systems / operating systems

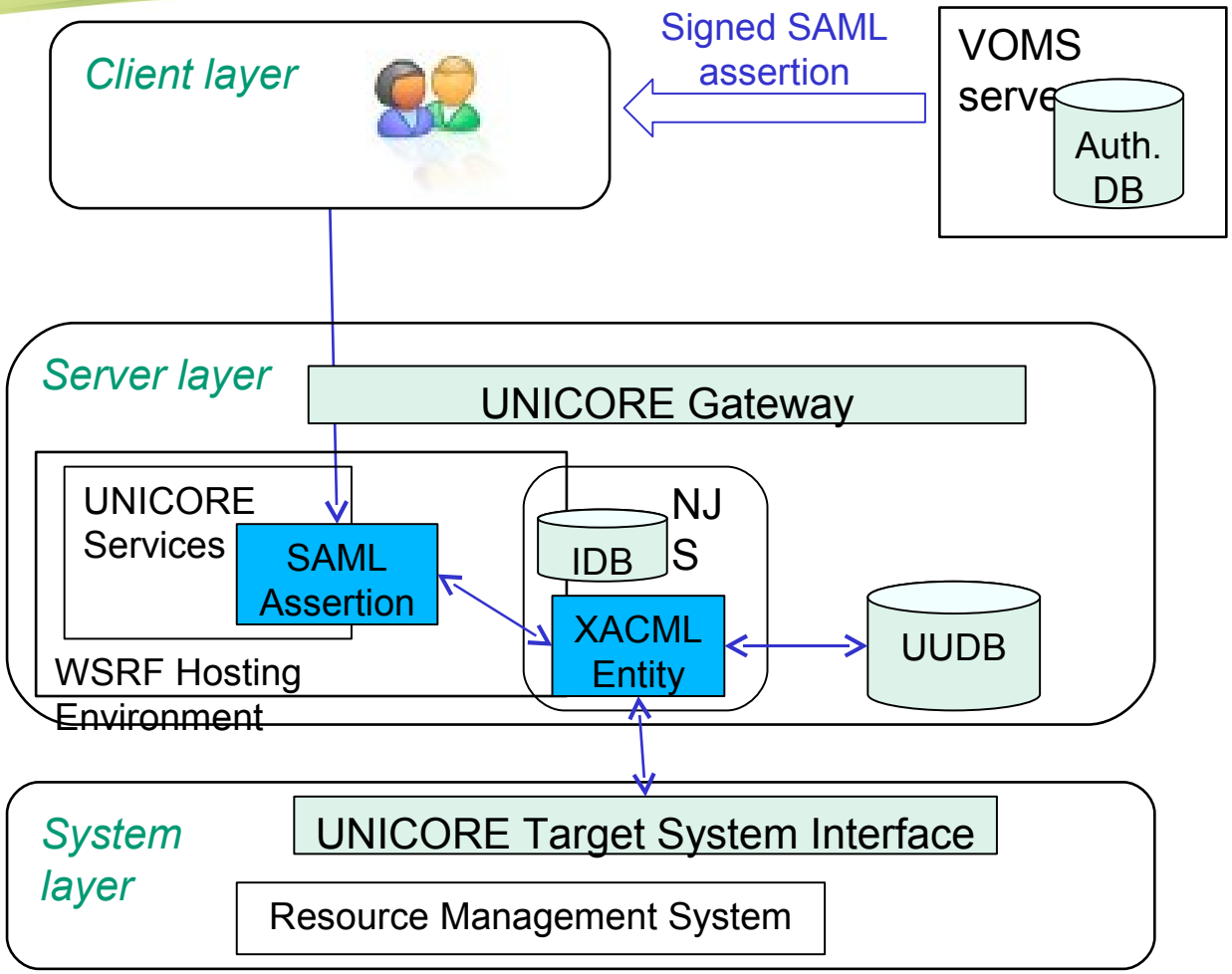
UNICORE security – overview

- **Mutual authentication (between Gateway/NJS and User) using X509 Certificates**
- **No proxy certificates, no generalized delegation**
- **Authorization:**
  - Performed by NJS (and thus moved away from the target system)
  - Using UUDB (Unicore User Database)
  - More recent extensions to support both role and attribute based authorization (VOMS, Shibboleth)
- **Separation of consigner and endorser: only a user can *endorse* a job; an NJS or a user can *consign* a job**

UNICORE security – VOMS

- **VOMS releases SAML assertions containing user attributes**
  - The assertions are included in the SOAP headers,
  - and signed with the VOMS server's certificate
- **The authorization decisions are taken in the service tier**
  - PDP – Policy Decision Point
  - uses XACML policies,
  - and information obtained from the UNICORE User Database

UNICORE – Security flow (with SAML based VOMS)



Interoperability

- **Can services that are hosted in different environments (with different security mechanism) interoperate?**
- **This is a difficult problem**
  - We cannot expect all the organizations to adopt a single security technology
  - Or to share their user registries with other organizations
- **Ongoing work in many of the current Grid projects**
  - standardized protocol to communicate authorization assertions across OSG, EGEE, Globus and Condor
  - XtreemOS: interoperability solution based on SAGA (Simple API for Grid Applications)

References

- M. Coppola, Y. Jegou, B. Matthews, C. Morin, L.P. Prieto, O.D. Sanchez, E.Y. Yang, H. Yu. *Virtual Organizations Support within a Grid-Wide Operating System*. IEEE Internet Computing, 12(2):20-28, March/April 2008. Available at http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=4463381
- M. Adamski, A. Arenas, A. Bilas, P. Fragopoulou, V. Georgiev, A. Hevia, G. Jankowski, B. Matthews, N. Meyer, J. Platte, and M. Wilson. *Trust and Security in Grids: A State of the Art*. CoreGRID White Paper WHP-0001, May 2008. Available at http://www.coregrid.net/mambo/images/stories/WhitePapers/whp-0001.pdf
- E.Y. Yang (editor). *D3.5.11 - 3rd Specification and Design of XtreemOS Security and VO Services*. XtreemOS Project Deliverable, 2008. Available at http://www.xtreemos.eu/publications/plonearticlemultipage.2008-06-26.0232965573/pub
- V. Venturi, M. Riedel, A.S. Memon, M.S. Memon, F. Stagni, B. Schuller, D. Mallmann, B. Tweddell, A. Gianoli, V. Ciaschini, S. van de Berghe, D. Snelling, and A. Streit. Using SAML-based VOMS authorization within Web Services-based UNICORE Grids. Proceedings of 3rd UNICORE Summit 2007 in conjunction with EuroPar 2007, Rennes, France, LNCS 4854. Available at http://www.unicore.eu/documentation/documents.php

Thank you!

Questions?

XtreemOS

Enabling Linux for the Grid

ICS'09

Tutorial on Security and Virtual Organization Management in Grids

PART 3 - Security and VO Management in XtreemOS

Information Society Technologies

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Outline

- Introduction to XtreemOS
- Administration of Grid Systems
- Security Model on XtreemOS
- Foundations for Security Enforcement
- XtreemOS Security Infrastructure
- On-going Work

XtreemOS

Introduction to XtreemOS

- **XtreemOS is a Grid Operating System**
- **Targets**
  - Large number of users
  - Large number of resources
  - High dynamicity
- **XtreemOS**
  - POSIX/UNIX interface for developers
  - POSIX/UNIX interface for users
  - Supports legacy applications
  - Supports Grid standards (ex: SAGA)

Global Operating System Services

- **Distributed services**
- **Scalability**
  - Provided through replication
- **Dependability**
  - Replication
  - Migration
- **Virtual Nodes**
  - Framework for scalable and resilient services
- **Service Discovery**

Resource Allocation for Applications

- **No global Scheduler**
  - Job manager service created for each job
- **Resource Discovery on peer-to-peer Overlay**
  - Structured overlay for faster access to requested resources
  - Resource negociation
  - VO policies checked during discovery

Administration of XtreemOS Grids

Domain Administrators

- Domain administrators delegate user administration to Virtual Breeding Environments (VBE)
  - SLA
  - PKI infrastructure
- Users create VOs
- Domain administrators provide resources to VOs
- Resource owners always in control

VBE / VO

- **Virtual Breeding Environment – VBE**
  - Infrastructure for hosting Virtual Organisations (VO)
  - User registration
  - VO lifecycle
  - Implements core services
- **Virtual Organisations**
  - Manage VO models (groups, roles, capabilities)
  - Manage user credentials (attributes)
- **VO administration**
  - Geographically distributed
  - Autonomous, independent from administration domains

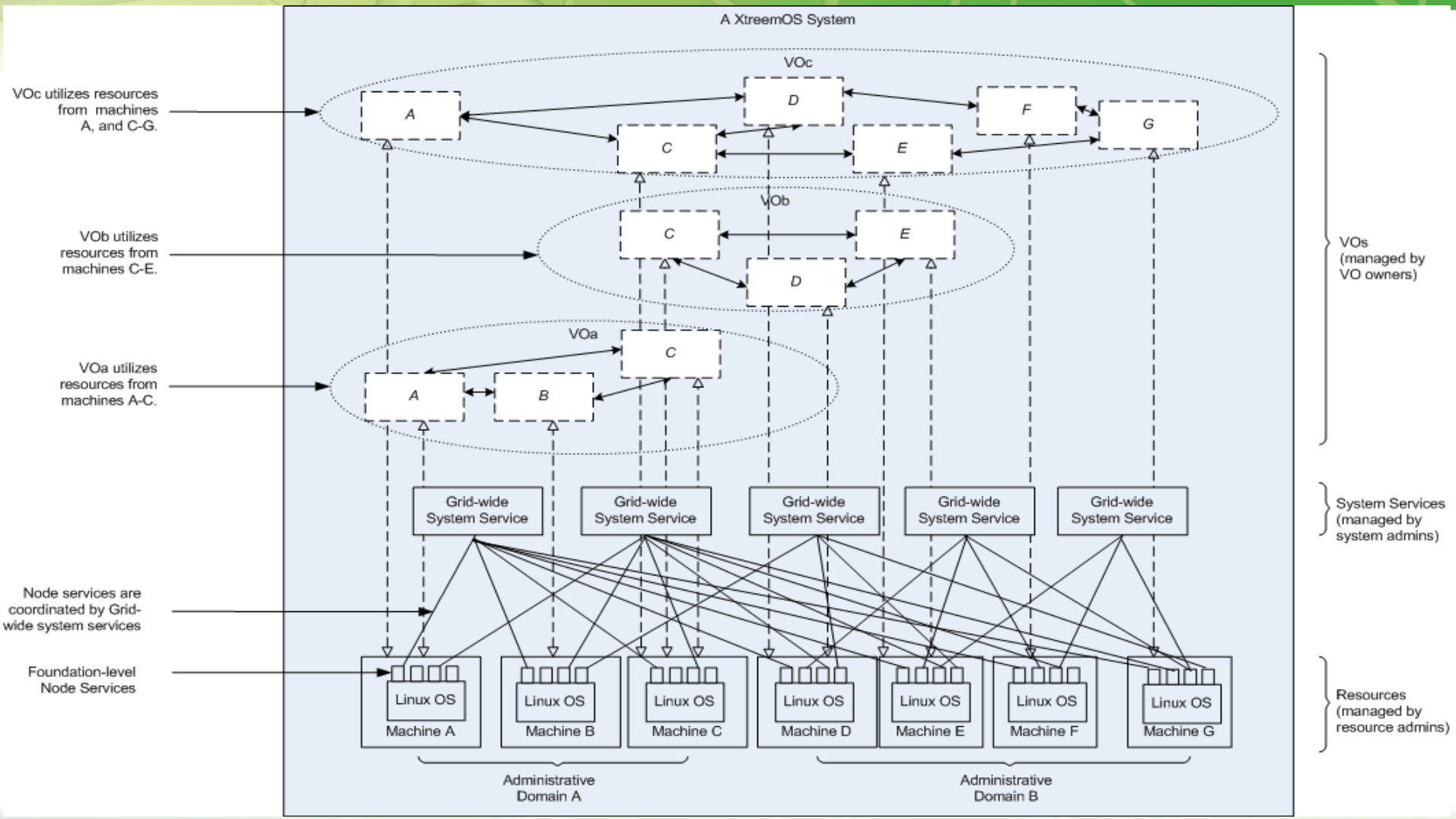
Virtual Organisation Administration

- **A VO can be seen as a distributed organisation which has the task of managing access to resources that are accessed through computer network and located in different domains**
- **Administration through the distribution of**
  - Identity certificates (X.509)
  - Attribute certificates
    - Bind credentials to identities

  to users and resources

XtreemOS
Enabling Linux for the Grid

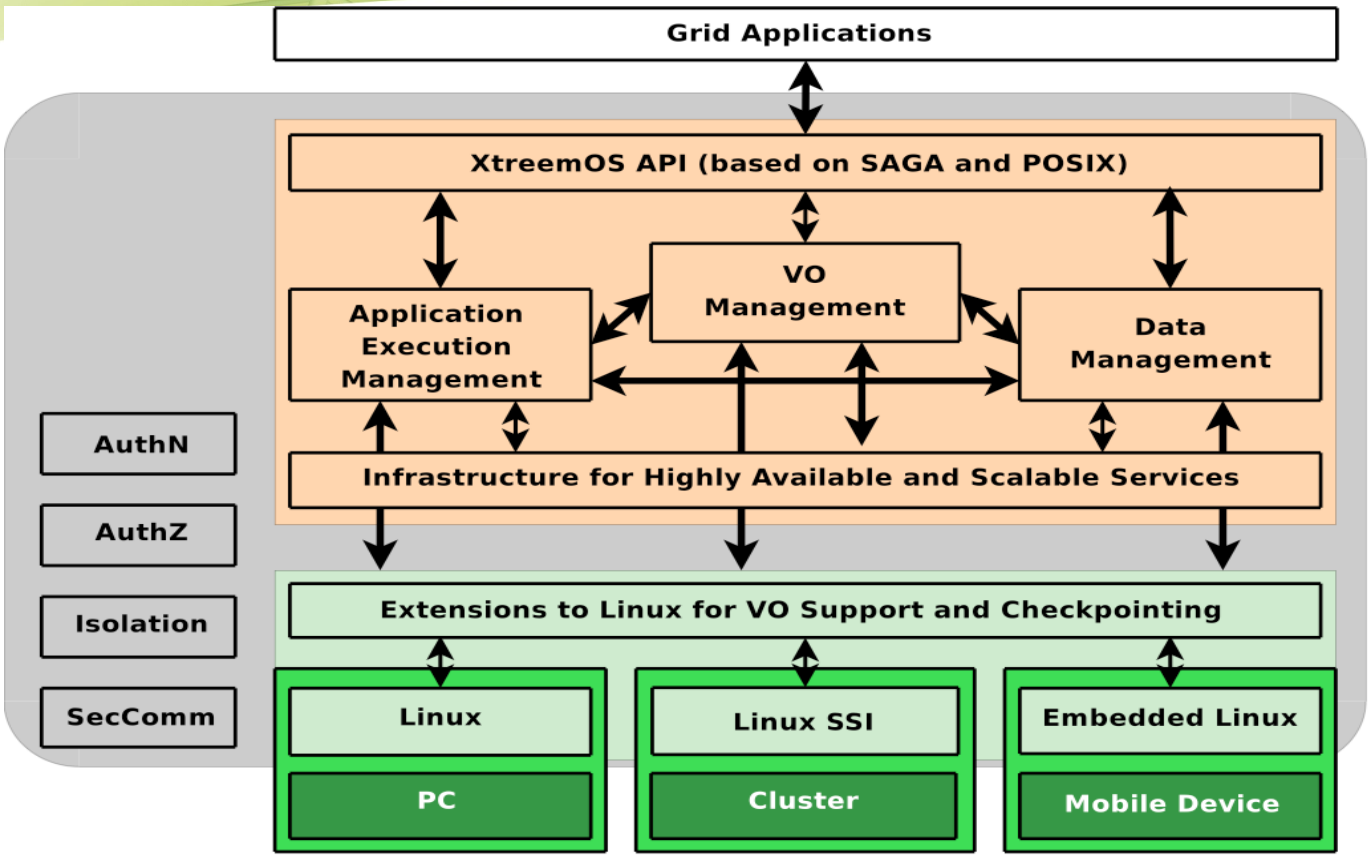
XtreemOS System

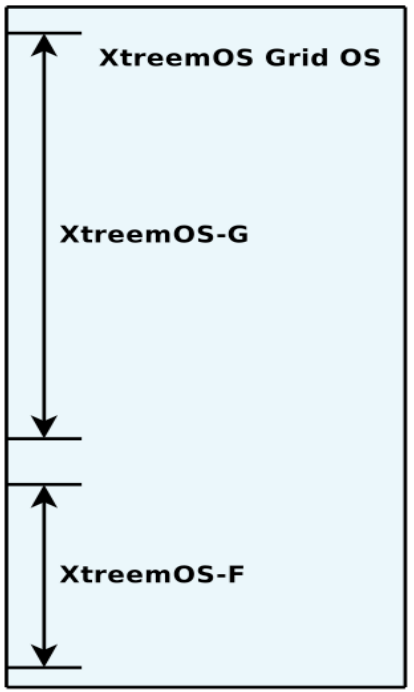


XtreemFS

- **Distributed file system**
  - Spanning the grid
  - Replication
  - Striping
- **Access control based on Grid attributes**
- **Each XtreemOs users has one home volume in XtreemFS**

XtreemOS Architecture



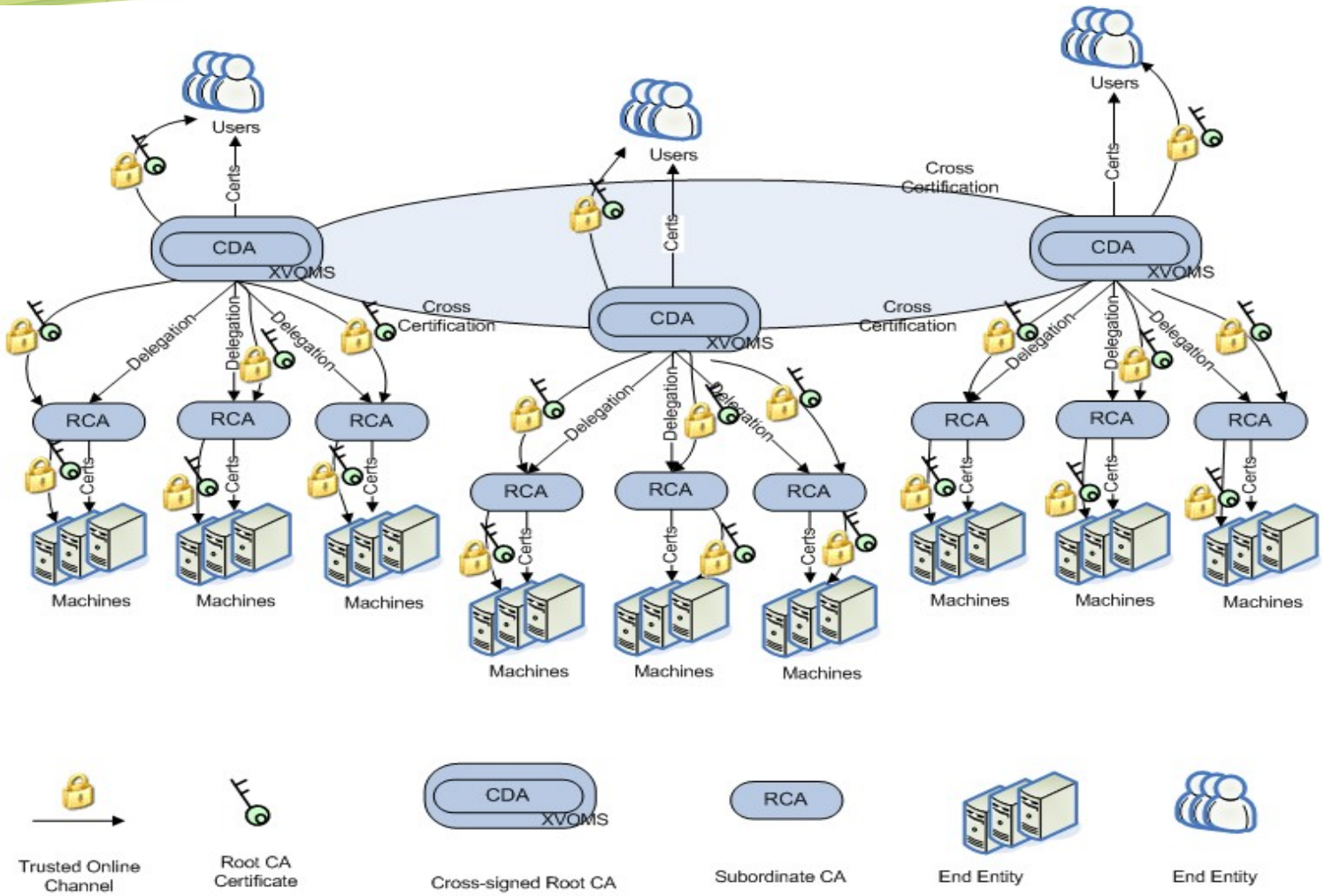


Security Model in XtreemOS

Security Model

- **PKI-based trust model**
  - Top: a set of cross-certified root CAs
  - Underneath: subordinate CAs (RCAs)
  - Identifiers and attributes

Trust Model


Users
Certs
CDA
XVOMS
Cross Certification
Delegation
RCA
Machines
Trusted Online Channel
Root CA Certificate
Cross-signed Root CA
Subordinate CA
End Entity
End Entity

VBE and VO

- **Virtual Breeding Environments – VBE**
  - Provide security based on trust
    - Services running on behalf of a VBE trust each other
    - Trust established through cryptography
    - Secure communications
  - Provide means to manage VOs in a scalable way
- **Authorization based on node-level and VO-level policies**

Single-Sign-On and Delegation

- **Single-Sign-On**
  - User session management services trusted by XtreemOS services
  - In charge of validating user credentials and user requests
  - Provides the interface between the user space and the operating system space
- **Delegation**
  - User session management services can be replicated on resource nodes
  - User can run Grid requests from resource nodes (same capabilities as from their access node)

Isolation

- **Protection**
  - Security
  - Performance, quality of services
  - Resource usage

Foundations

Namespaces

- **Global namespaces**
  - GUID, GVID, GGID, GNID
    - Identifiers
    - Global IDs are unique
  - Users and nodes have X.509 certificates
    - Identity stored in the distinguished name (DN)
- **Node-level (local to resources) namespaces**
  - OS users (UID/GID)
  - Files (inodes)
  - Processes (PIDs)
- **VO namespaces**
  - Groups, role, capability

Namespaces, cont.

- **Mapping between different namespaces managed by local service `xos-amsd`**
  - GUID ↔ UID
  - GGID ↔ GID
- **With the support of `nsswitch`**
  - `ls -l` shows the GUID of the file owner

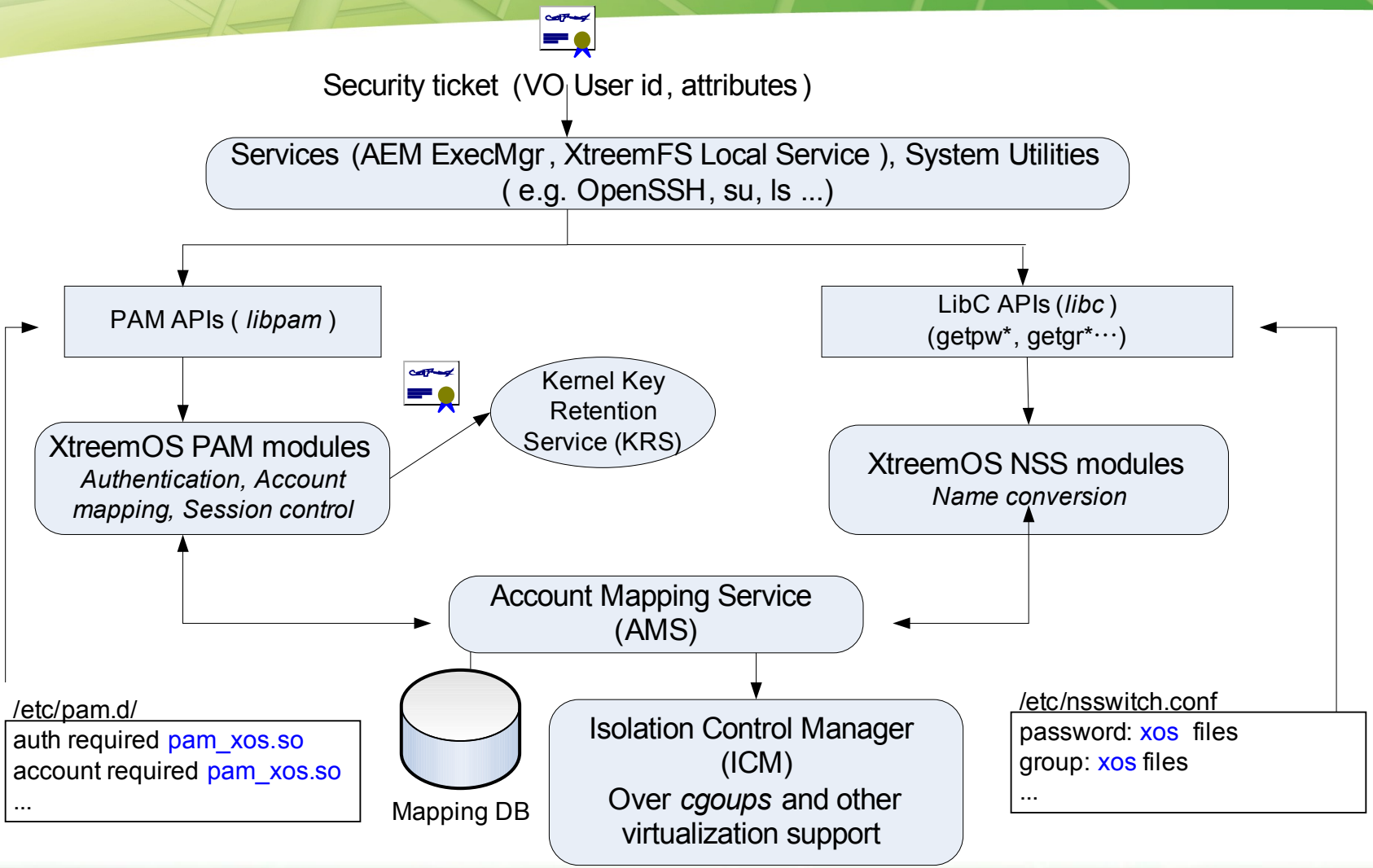
Isolation, Job context

- **Job context created**
  - When a user session is opened on some resource
  - Can be
    - Simple Unix account
    - Control groups
      - limit/protect resource usage
      - Accounting, billing
    - Namespaces (PID, user, net, ...)
      - Restrict visibility from job context
      - Net namespaces restrict access to Internet
    - Containers (~ cgroups + namespaces)
    - Virtual machines

Resource-level XtreemOS components

- `xos-amsd`: management of global to local entity mapping
- `pam-xos`: modules in charge of authentication, autorisation and session management
- `nsswitch`: POSIX namespace management
- `ssh-xos`: extends ssh authentication with XOS certificates
  - Provides same account mappings as for jobs

Internal Components of Node-level VO Support



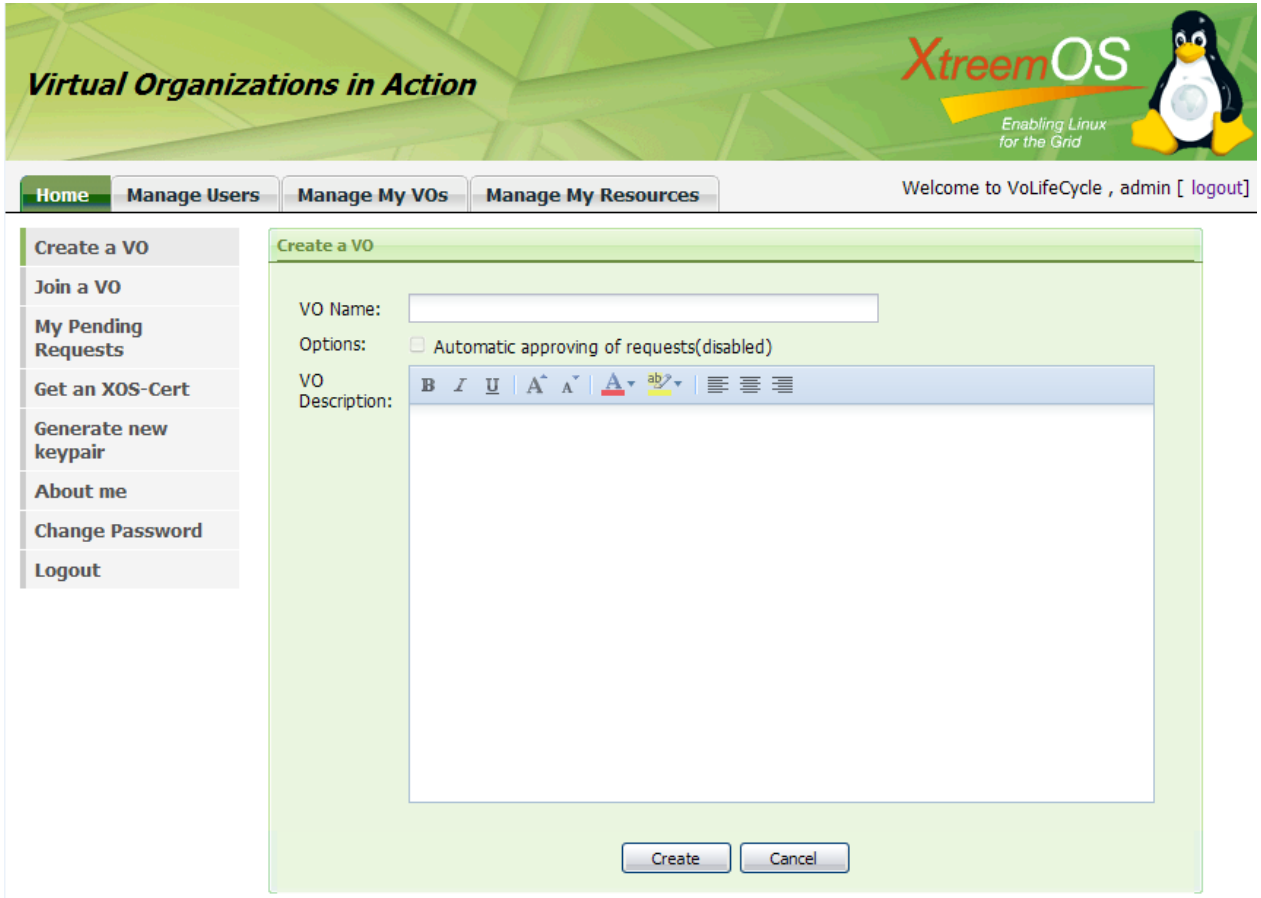
XtreemOS Security Architecture Components

XVOMS

- **XVOMS**
  - User and RCA registration
  - VO lifecycle management
    - Creation/dissolution

Web frontend, VO Creation

VOLife Frontend →

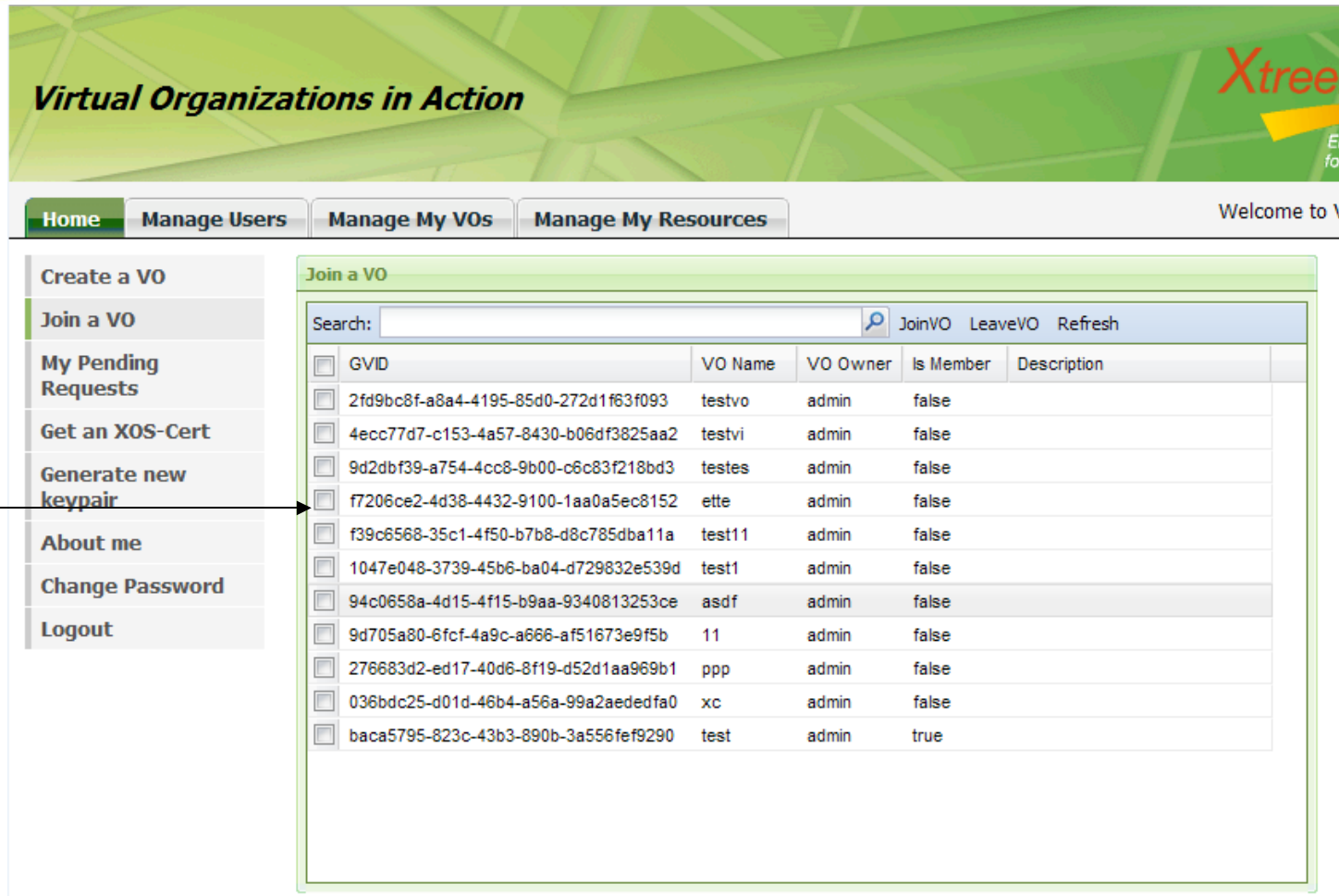


VO Lifecycle (2)

- **XVOMS**
  - User and RCA registration
  - VO lifecycle management
    - Creation/dissolution
    - User and node registration
    - Define and manage attributes (ex: roles and groups)
      - Associate attributes to users

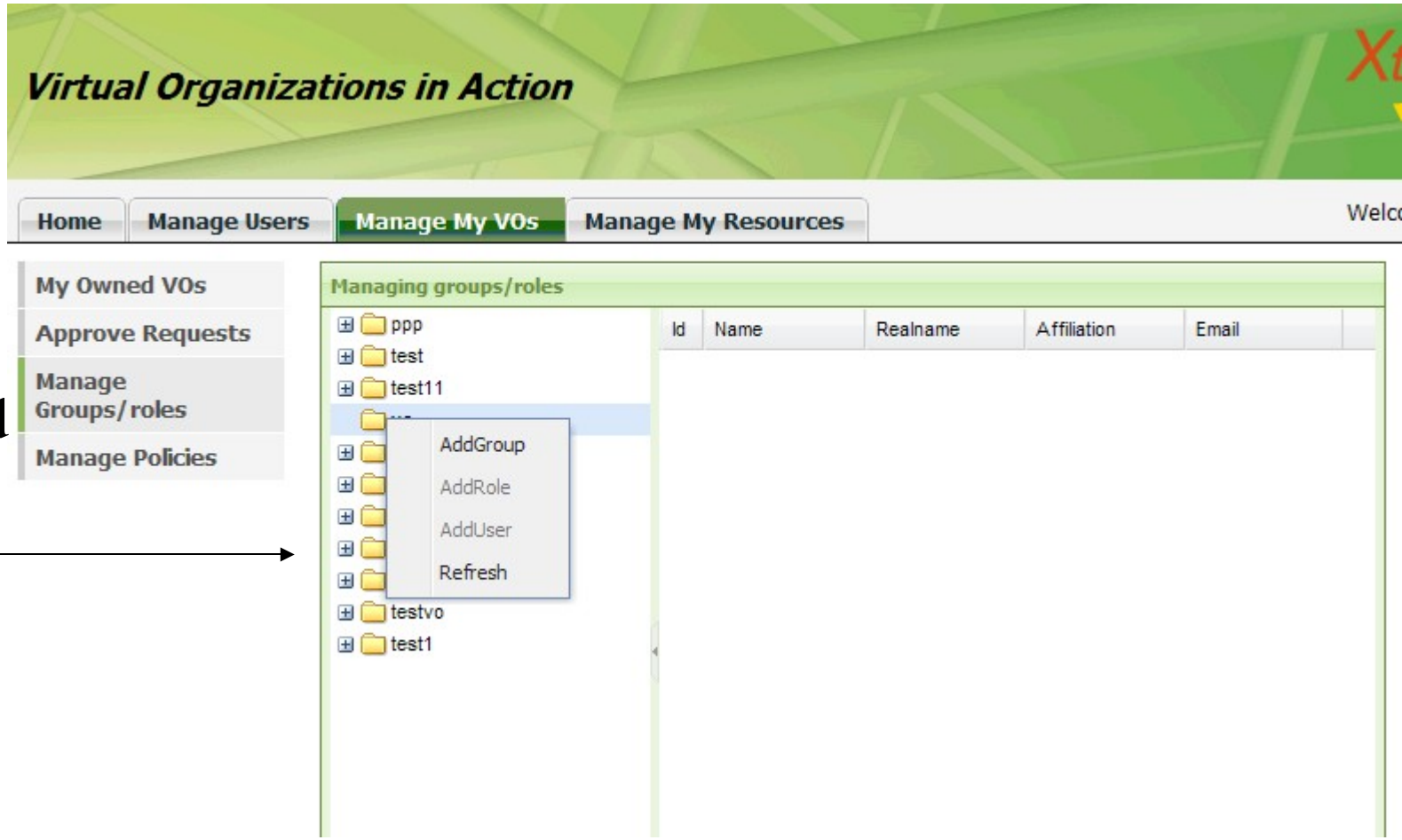
Joining a VO

Select VO and send joining requests



Manage VOs

Manage your own VOs, e.g. adding groups and roles, or policies

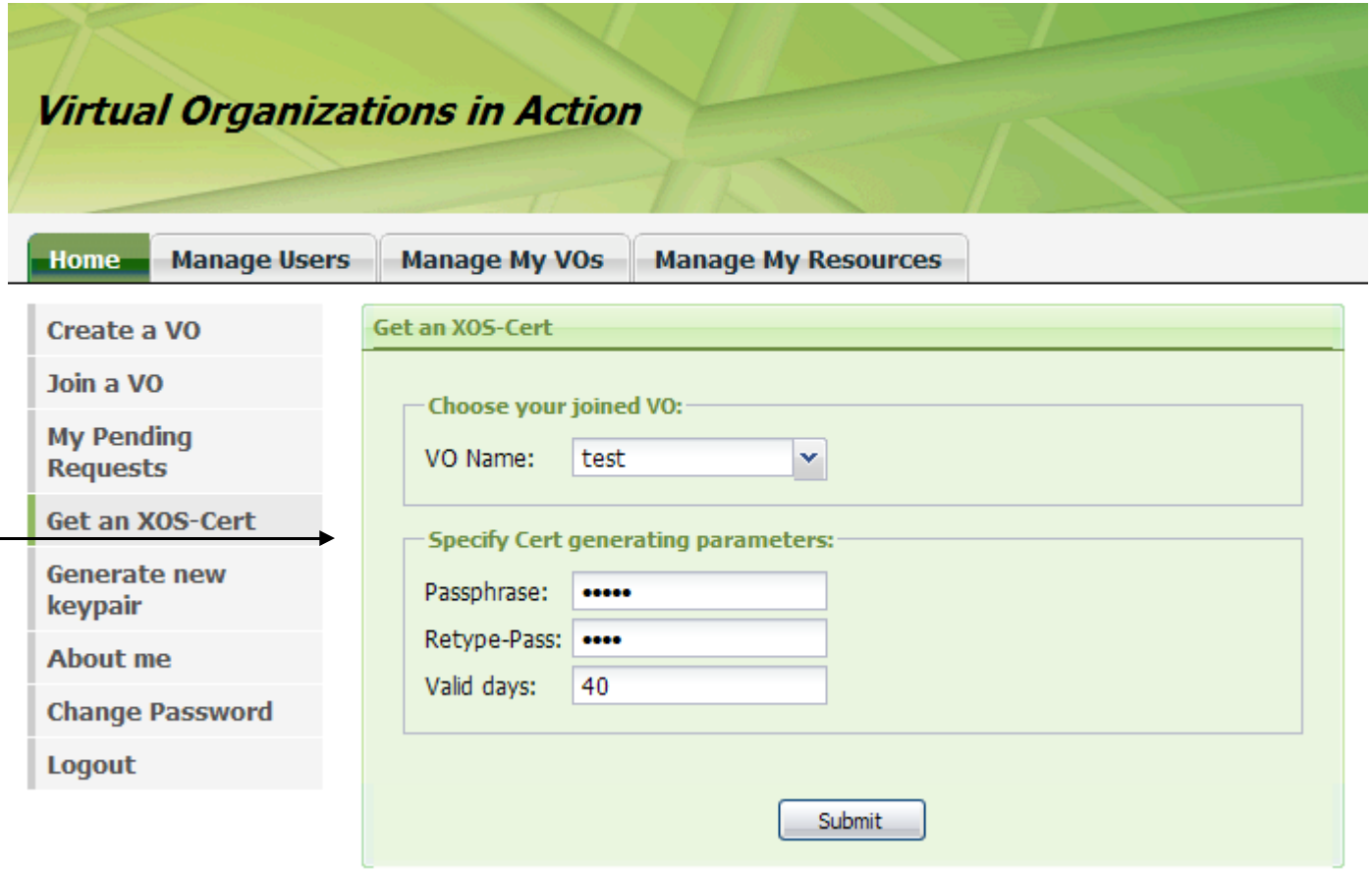


VO Lifecycle (3)

- **XVOMS**
  - User and RCA registration
  - VO lifecycle management
    - Creation/dissolution
    - User and node registration
    - Define and manage attributes (ex: roles and groups)
      - Associate attributes to users
  - User credential distribution
    - Attribute certificates

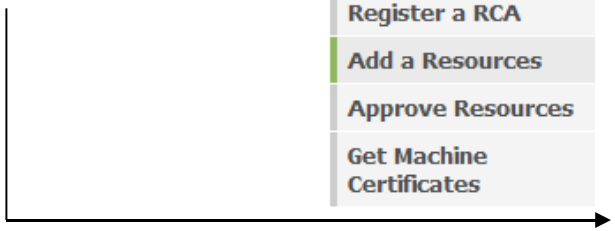
Get an XOS certificate

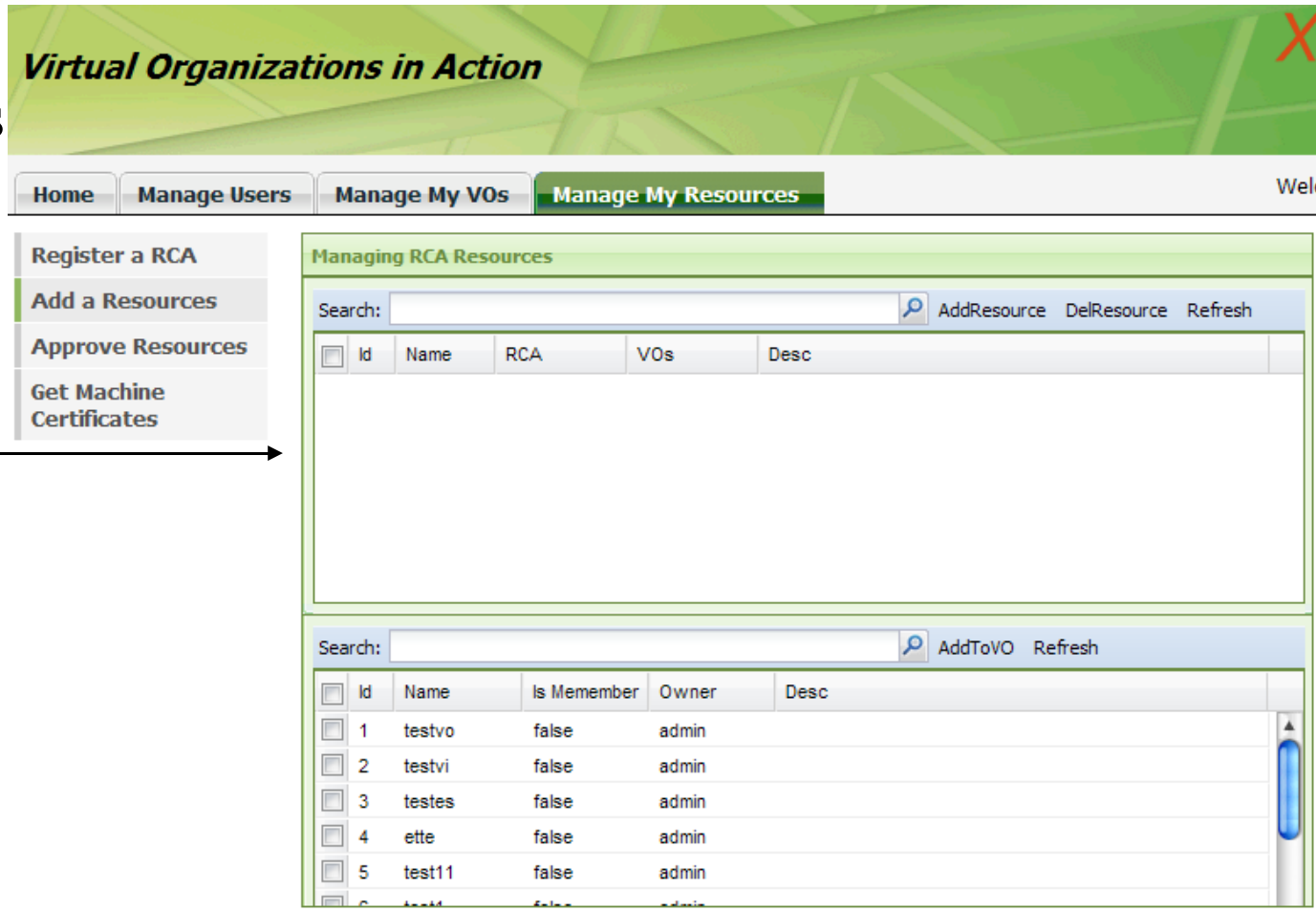
After the request is approved, getting an XOS-cert online



Manage VO Resources

Manage resources in a VO





VO Lifecycle (4)

- **XVOMS**
  - User and RCA registration
  - VO lifecycle management
    - Creation/dissolution
    - User and node registration
    - Define and manage attributes (ex: roles and groups)
      - Associate attributes to users
  - User credential distribution
    - Attribute certificates
  - RCA: resource credential management

XtreemOS security components

- **VOPS**
  - Policy management point
  - Policy decision point
  - Filters to distribute policy decisions in a scalable way
- **RCA**
  - Resource registration
  - Distributes certificates to resources
  - Attributes define resource capabilities for resource discovery (#cpus, memory, ...)

User Session Service

- **User session services**
  - Started when the user logs in
  - In charge of validating user credentials
  - Trusted by XtreemOS operating system services
  - Bridging the user space with the operating system space
  - All grid requests go through the user session service
  - Support untrusted client nodes
- **Provide Single-Sign-On**
- **Provide Delegation**
  - Can be replicated on resource nodes

XtreemOS Security Components

- **Node-level security services**
  - Secure communication (certificate+SSL)
  - Policy for account mapping and credential management
  - Node-level and VO-level policies
  - Isolation
    - Visibility / protection
    - performance

Conclusion

What we want to achieve ?

- **Local resource administrator**
  - Autonomous management of local resources
- **VO administrator**
  - Ease of management
  - Flexibility in VO policies

What we want to achieve ?

- **Users, service administrators**
  - Ease of use
    - **Simple login as a Grid user in a VO**
    - **The Grid should be as much as possible invisible**
    - **Posix interface as far as possible**
  - Secure and reliable application execution
    - **Fine-grained control of resource access**
    - **Accurate monitoring of application execution**
  - High performance
  - Ubiquitous access to services, applications & data from mobile devices

What do we want to achieve?

- **Application, service programmers**
  - **Linux applications should run with little (no) modifications**
  - **Grid applications should run with little (no) modifications**
  - **XtreemOS functionality must be provided to applications**

XtreemOS
Enabling Linux for the Grid

What could not be done before?

- Linux distribution including Grid support
  - Transparent remote application execution
  - Integration of Grid level authentication with system level authentication
  - Ease of management and use

- Three flavours of XtreemOS in contrast to most Grid middleware targeting machines exploited with a batch system
  - PC, clusters, mobile devices
  - Single system image clusters
    - Kerrighed Linux based SSI

What could not be done before?

- **Scalable VO management**
  - Independent user and resource management
  - Interoperability with VO management frameworks and security models
  - Customizable isolation, access control and auditing
- **Distributed application management**
  - No global job scheduler
  - Resource discovery based on an overlay network
- **Grid file system federating storage in different administrative domains**
  - Transparent access to data

On-going work

- **Very Dynamic VOs**
  - Created automatically for the duration of an application/workflow
    - Multi-users
  - Lightweight configuration of resources
  - Predefined policies (VO-based)
- **Interoperability**
  - GridShib (Shibboleth)

Thank you !

Questions ?

XtreemOS
Enabling Linux for the Grid

More Information on XtreemOS

http://www.xtreemos.eu

To contact us: contact@xtreemos.eu

Second open source XtreemOS release planned in Summer 2009

Public Deliverables related to Security and VO Management in XtreemOS

- **All deliverables in http://www.xtreemos.org/publications/plonearticlemultipage.2008-06-26.0232965573/public-deliverables**

- **Security services**
  - D3.5.11 - 3rd specification and design of security & VO services
  - D3.5.5 - First prototype of implementation of security services
  - D3.5.4 - Second draft specification of XtreemOS security services
  - D3.5.3 - First draft specification of XtreemOS security services
  - D3.5.2 - Security requirements for a Grid-based OS download
  - D3.5.1 - State of the art in the security for OS and Grids

Public Deliverables

- **Node level VO support mechanisms**
  - D2.1.6 - Evaluation of Linux native isolation mechanisms for XtreemOS flavours
  - D2.1.5 - Design and Implementation of Advanced Node-level VO Support Mechanisms
  - D2.1.4 - Prototype of the basic version of Linux-XOS
  - D2.1.2 - Design and implementation of basic version of node-level VO support mechanisms
  - D2.1.1 - Linux XOS specification

- **Other deliverables related to security in XtreemOS**
  - D3.5.10 - 1st report on modelling, evaluation and testing for XtreemOS Security Assurance
  - D3.5.8 - Specification of application firewall
  - D3.5.7 - Security for the XtreemFS File System
  - D3.5.6 - Report on formal analysis of security properties

C. XtreemOS tutorial at INRIA/EDF/CEA joint summer school
The summer school was organized near Paris in June, 2009.

XtreemOS

Enabling Linux for the Grid

Computing School 2009 CEA-EDF-INRIA

XtreemOS

Christine Morin, INRIA Rennes-Bretagne Atlantique

XtreemOS scientific coordinator

June 16, 2009

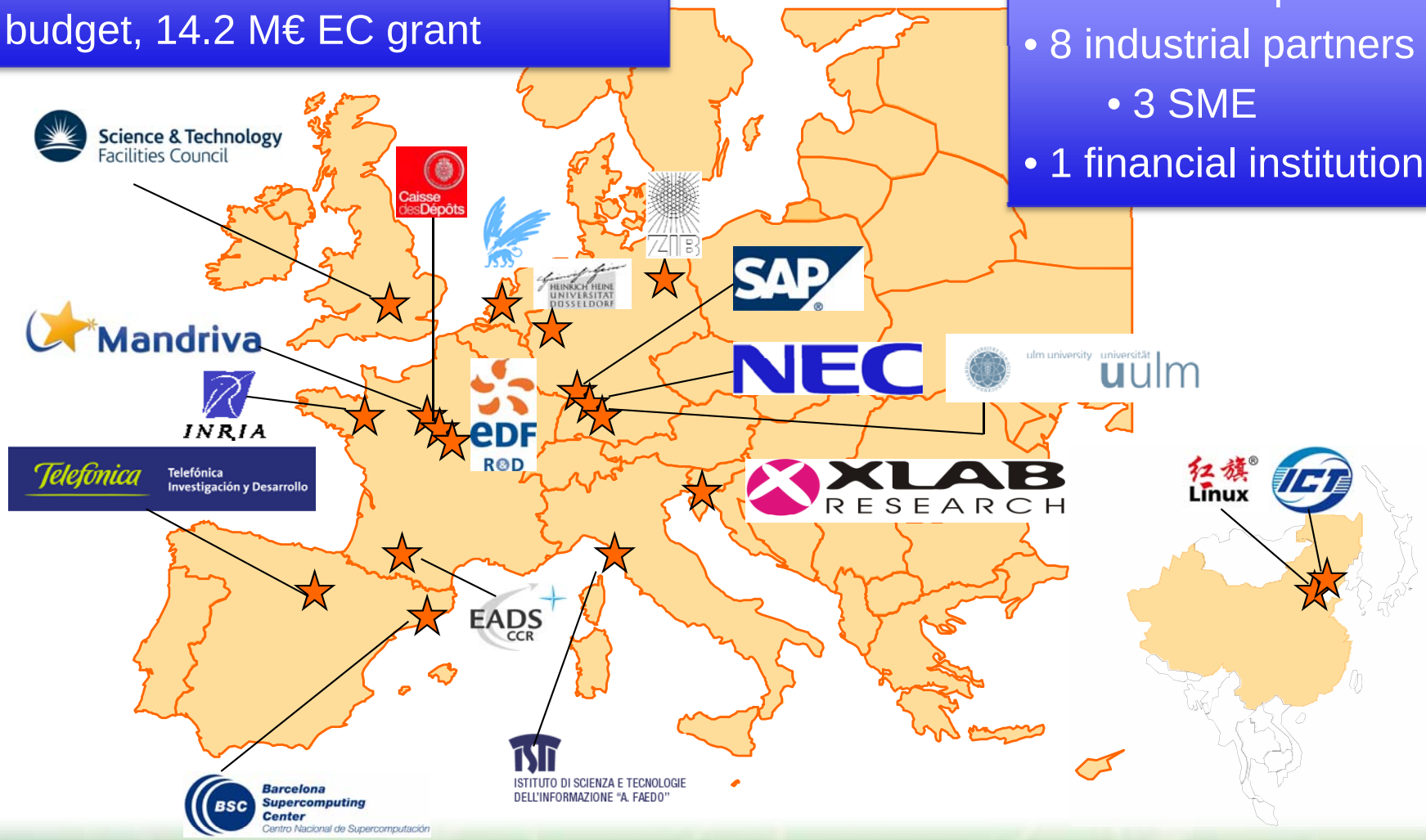
XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Information Society Technologies

XtreemOS European Project

- 4-year IP project started in June 2006 in the FP6 framework
- 30 M€ budget, 14.2 M€ EC grant

- 9 academic partners
- 8 industrial partners
  - 3 SME
- 1 financial institution



- **XtreemOS is a distributed operating system for Grids**
- **Targets**
  - Large-scale highly dynamic grids spanning multiple administrative domains
    - Large number of heterogeneous resources
    - Large number of users
  - Ease of use, management and programming
    - Posix/Unix interface for users & programmers
  - Efficient, reliable and secure application execution
    - Legacy applications
    - Grid applications (SAGA)

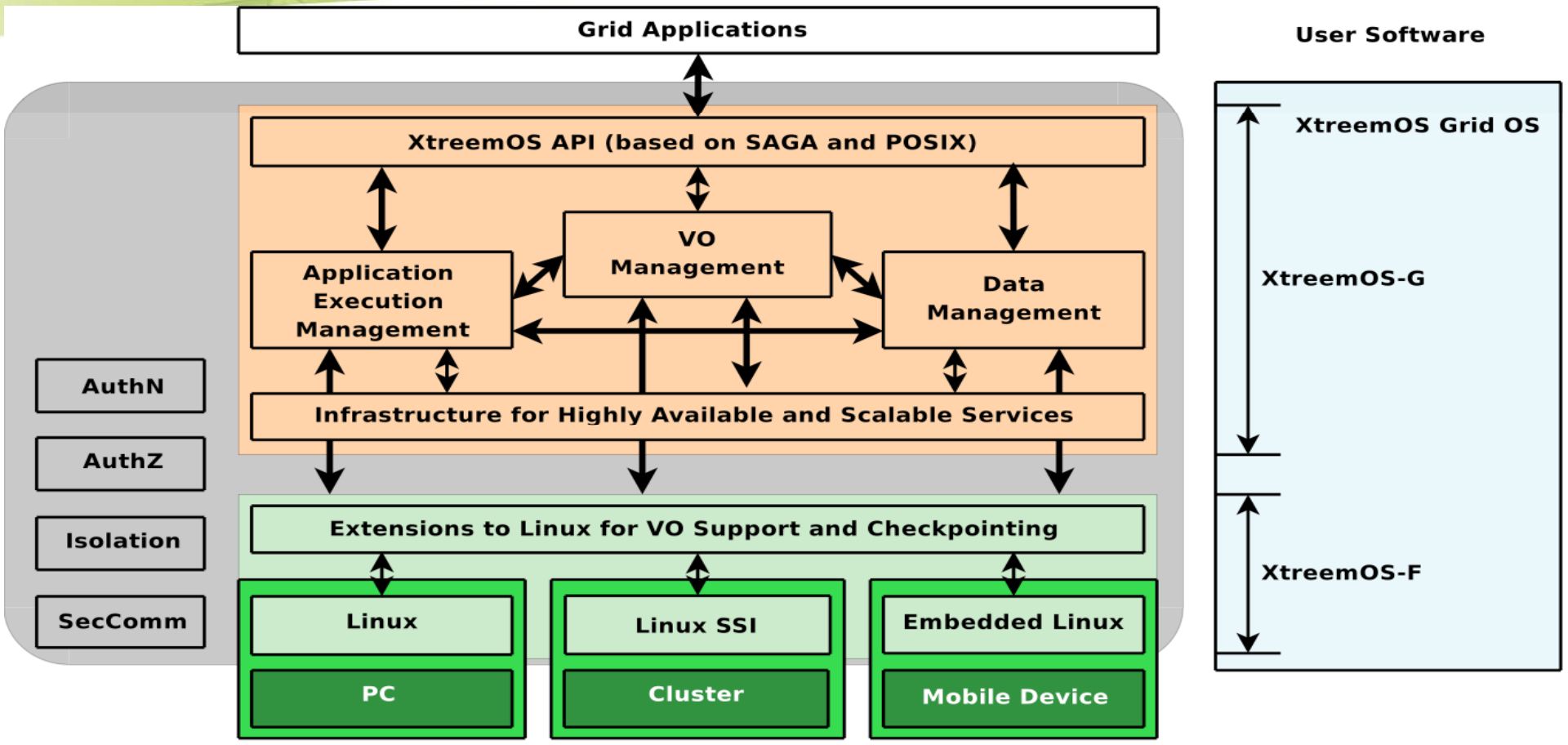
A **comprehensive** set of **cooperating** system services providing a **stable interface**

for a **wide-area** dynamic distributed infrastructure composed of **heterogeneous resources** and spanning **multiple administrative domains**

XtreemOS Fundamental Properties

- **Two fundamental properties: transparency & scalability**
  - Bring the Grid to standard users
  - Scale with the number of entities and adapt to evolving system composition
- **Scalability & dependability of XtreemOS system**
  - Distribution, replication, migration of XtreemOS services
  - Overlay as underlying communication system

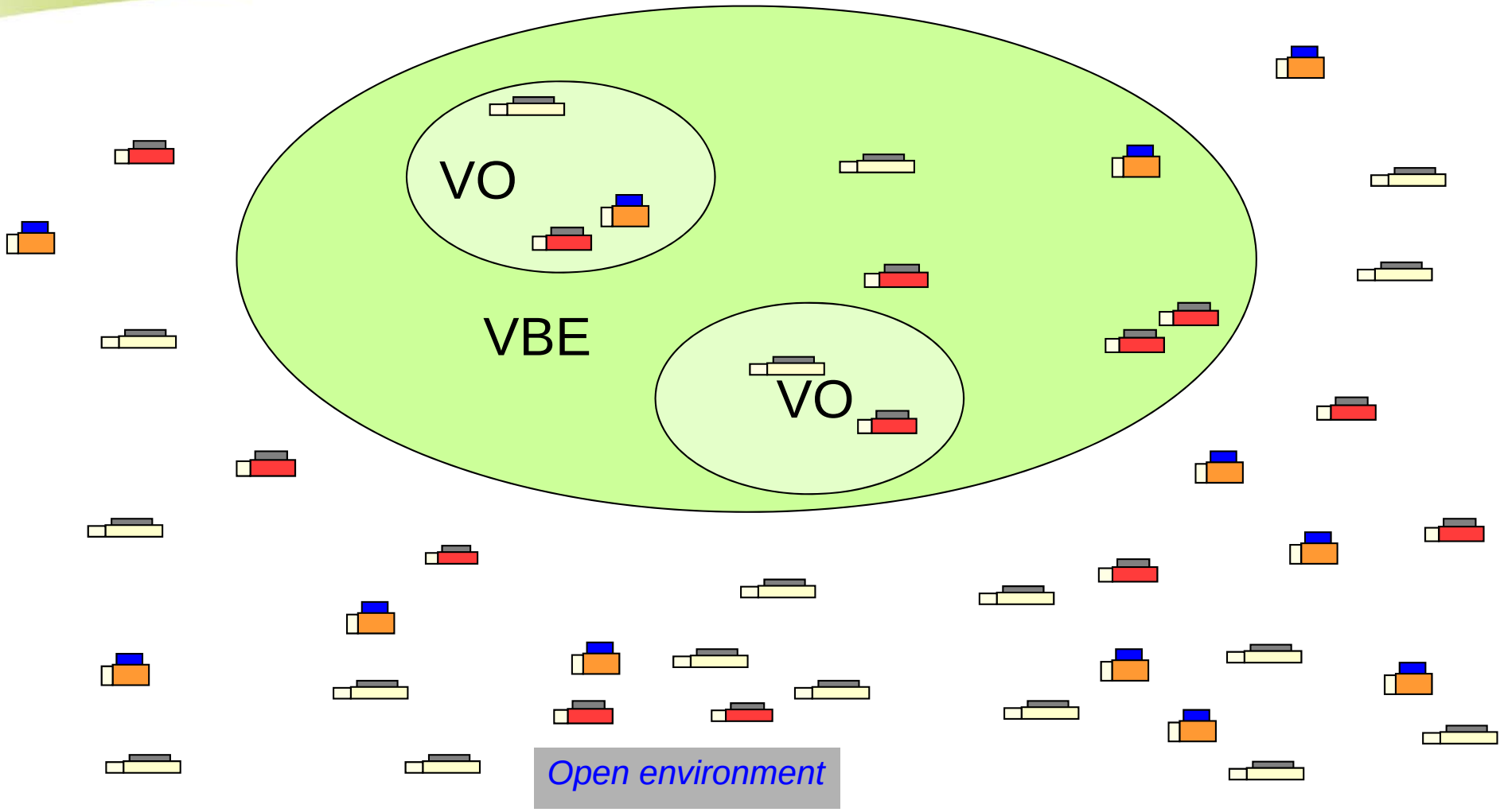
XtreemOS Architecture



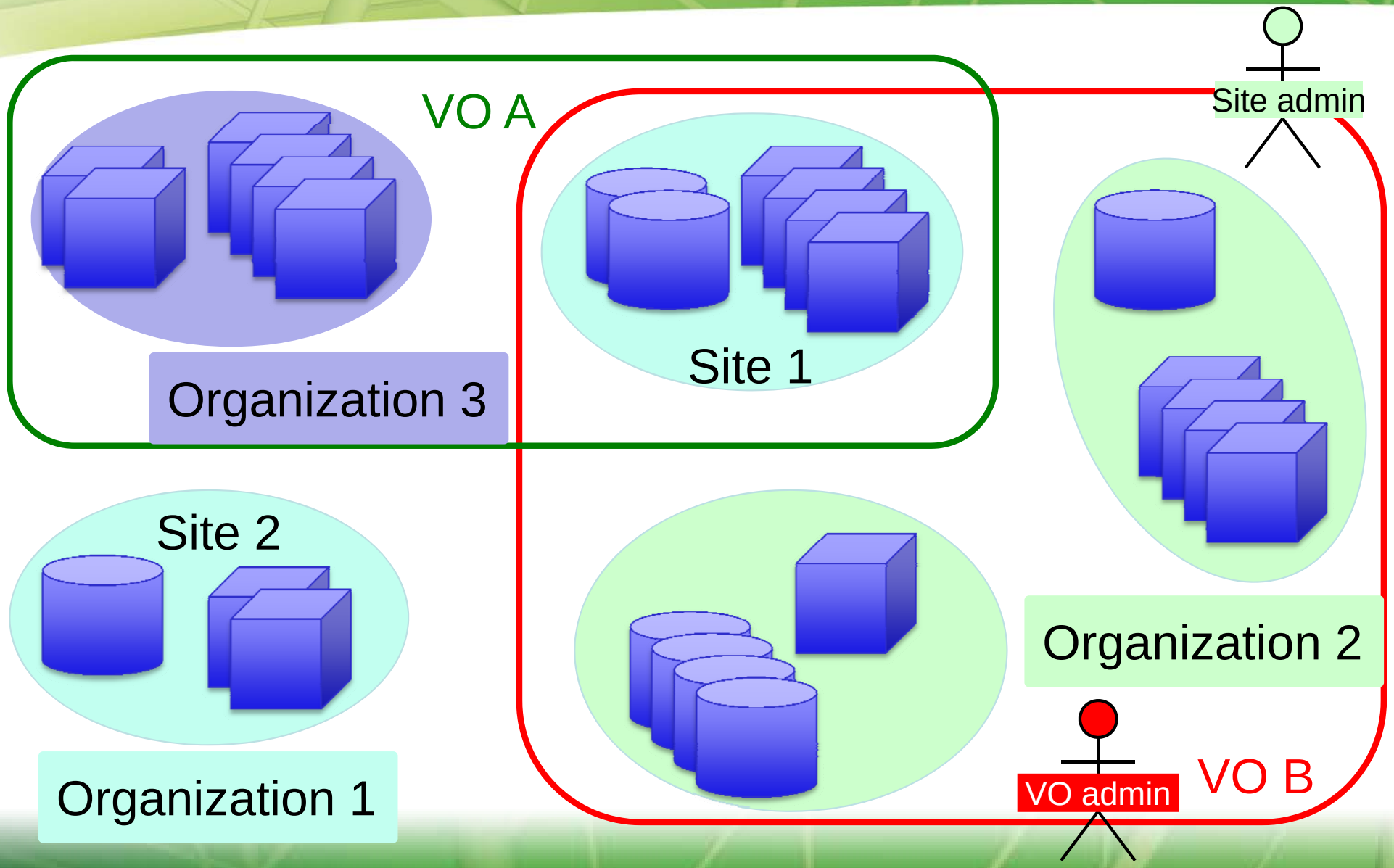
XtreemOS Grids

- **Virtual Organization (VO)**
  - VO = set of users that pool resources in order to achieve common goals - Rules governing the sharing of the resources
  - A VO can be seen as a distributed organization which has the task of managing access to resources that are accessed through computer network and located in different domains
- **Virtual Breeding Environment (VBE)**
  - VO are created in the context of a Virtual Breeding Environment (VBE)
  - A Virtual Breeding Environment is composed of users and service providers. It provides user and service provider registration, certificate management, and VO lifecycle management.

VBE & VO



Virtual Organizations



VBE and VO

- **Virtual Breeding Environments – VBE**
  - Provide security based on trust
    - Services running on behalf of a VBE trust each other
    - Trust established through cryptography
    - Secure communications
  - Provide means to manage VOs in a scalable way
- **Authorization based on node-level and VO-level policies**

Actors in XtreemOS Grids

- **VBE administrator**
  - VO life cycle
  - User registration
  - PKI infrastructure
- **VO administrator**
  - Manage VO models (groups, roles, capabilities)
  - Manage user credentials (attributes)
  - Manage VO membership
  - Define VO policies

Actors in XtreemOS Grids

- **Site/domain administrator**
  - Resource administrator (eg. computing & storage resources)
  - Provide resources to VOs
  - Local policies for resource access & usage
  - Resource owners always in control
  - Autonomous resource management
- **End Users**
  - First need to register to a VBE
  - Create VO and/or register to VOs (in the scope of their VBE)

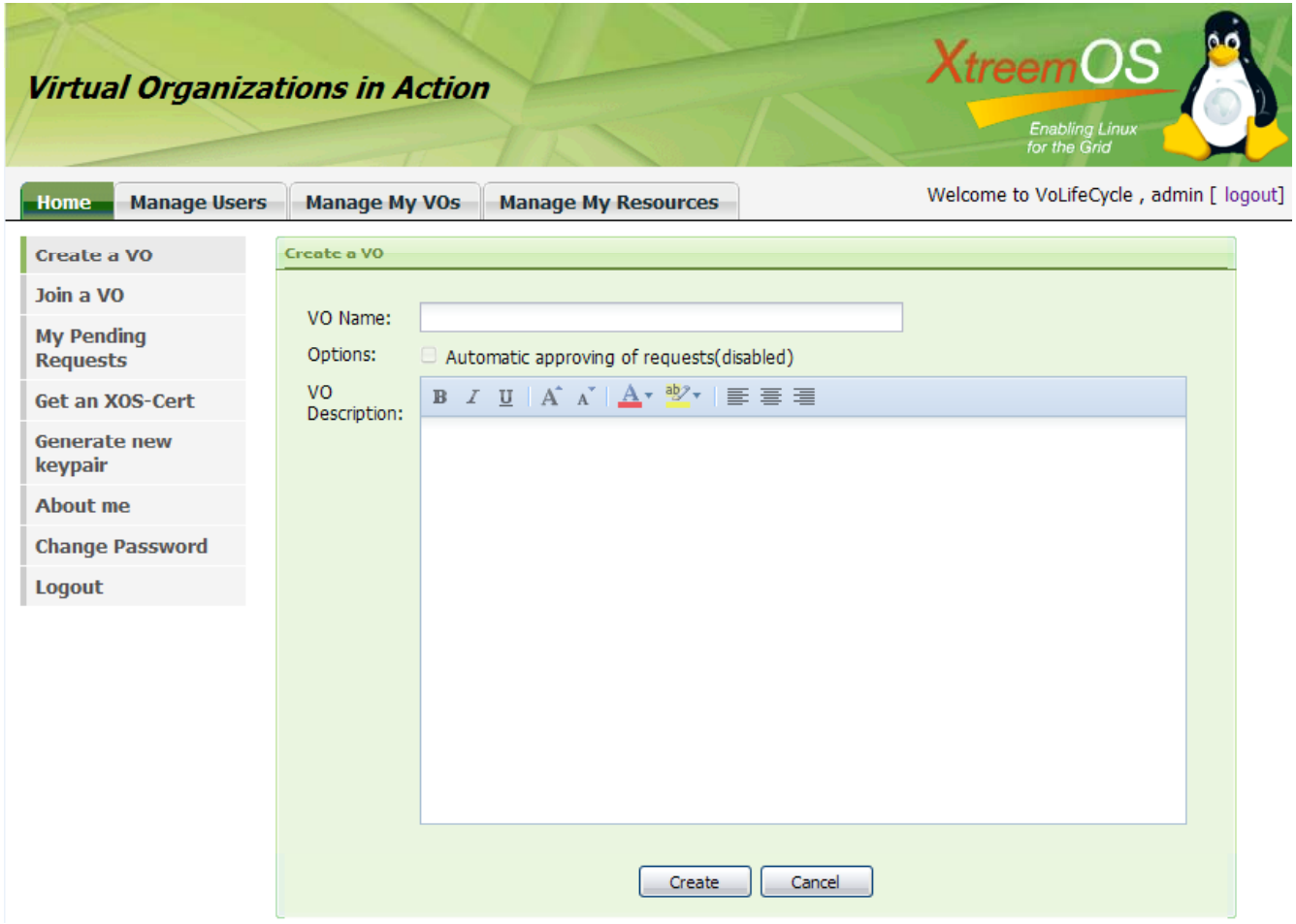
Virtual Organisation Administration

- **Administration through the distribution of**
  - Identity certificates (X.509)
  - Attribute certificates
    - Bind credentials to identities

  to users and resources

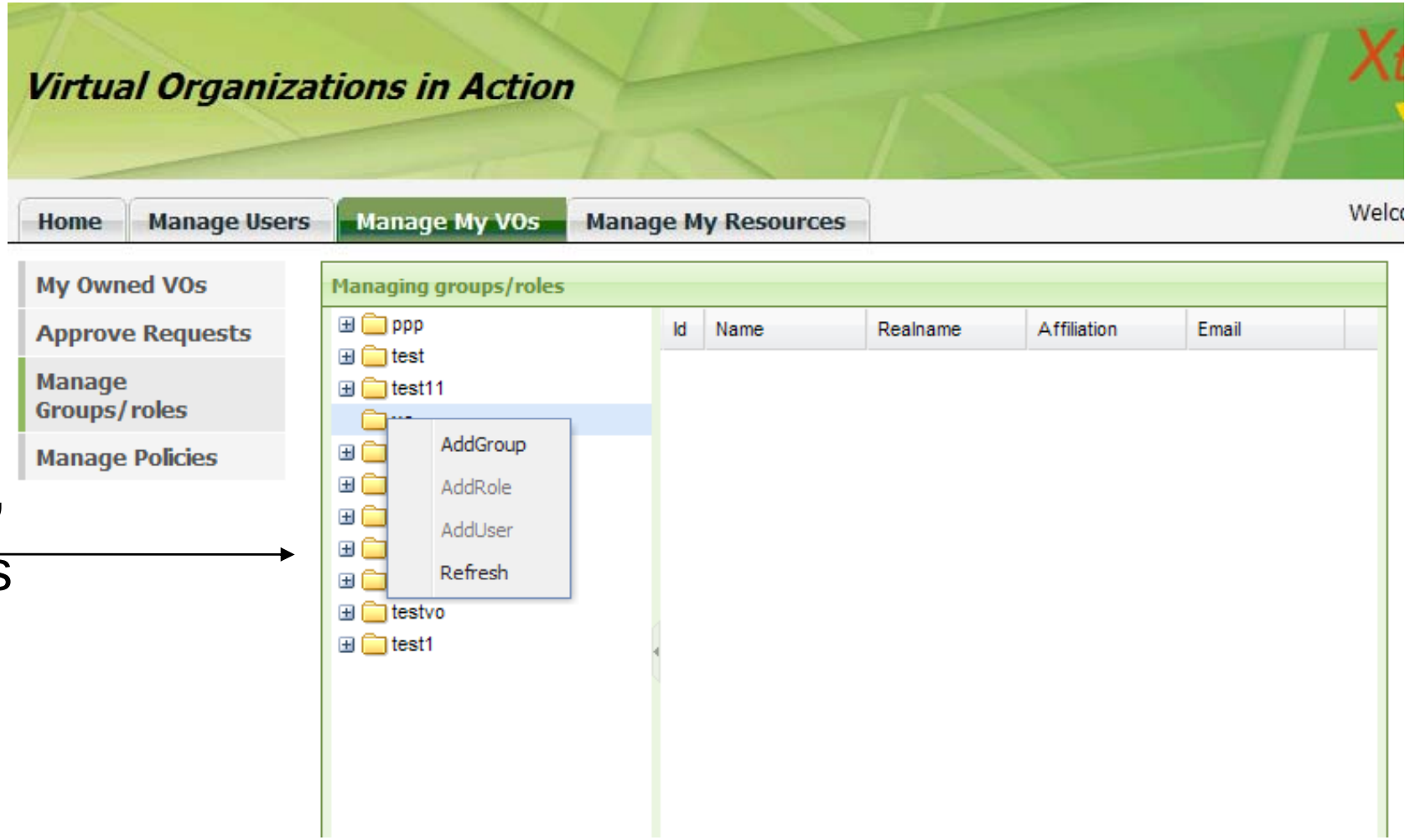
Web frontend, VO Creation (end-user)

VOLife Frontend →



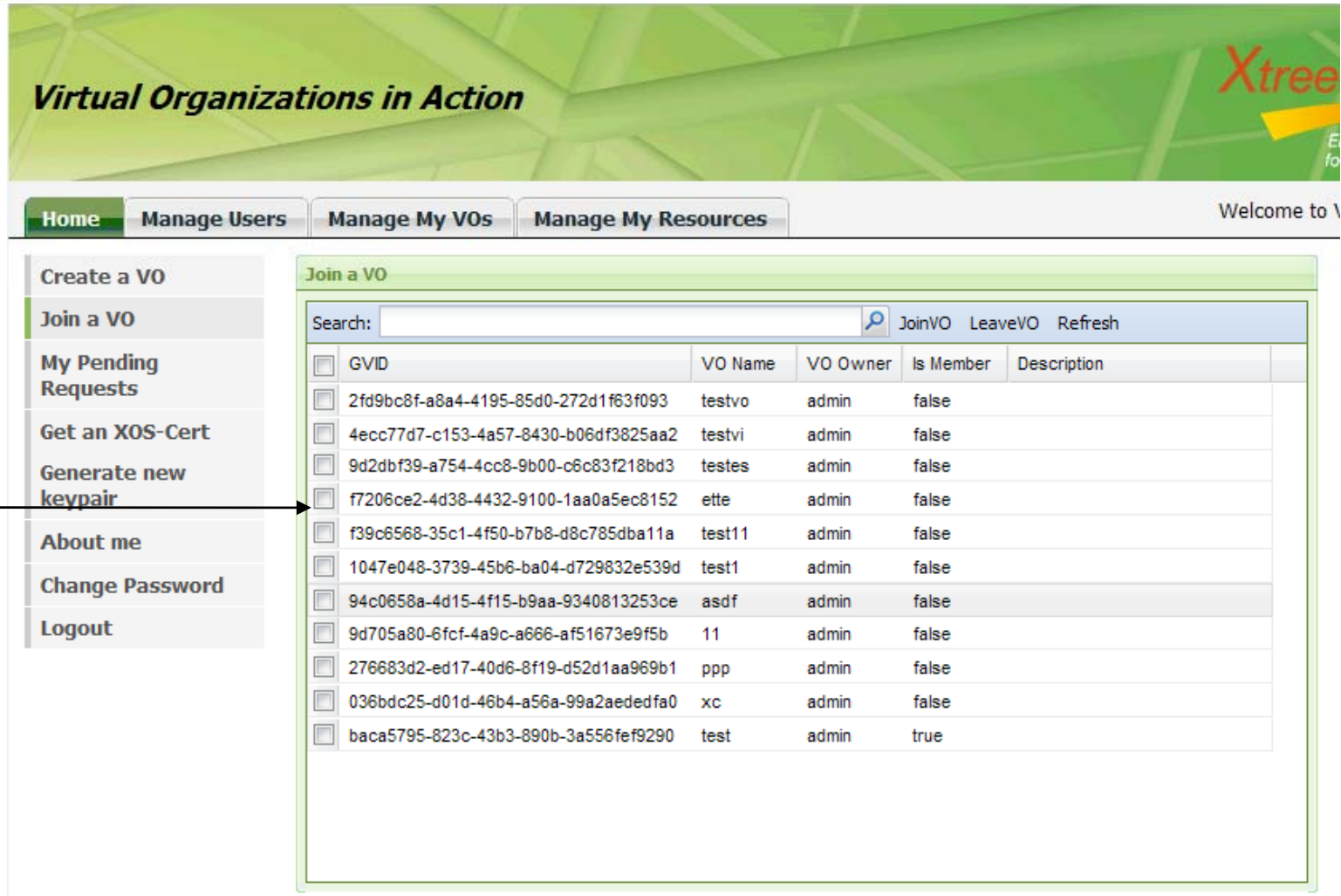
Manage VOs (VO admin)

Manage your own VOs, e.g. adding groups and roles, or policies



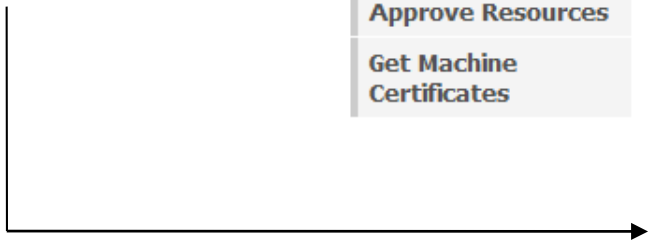
Joining a VO (end user)

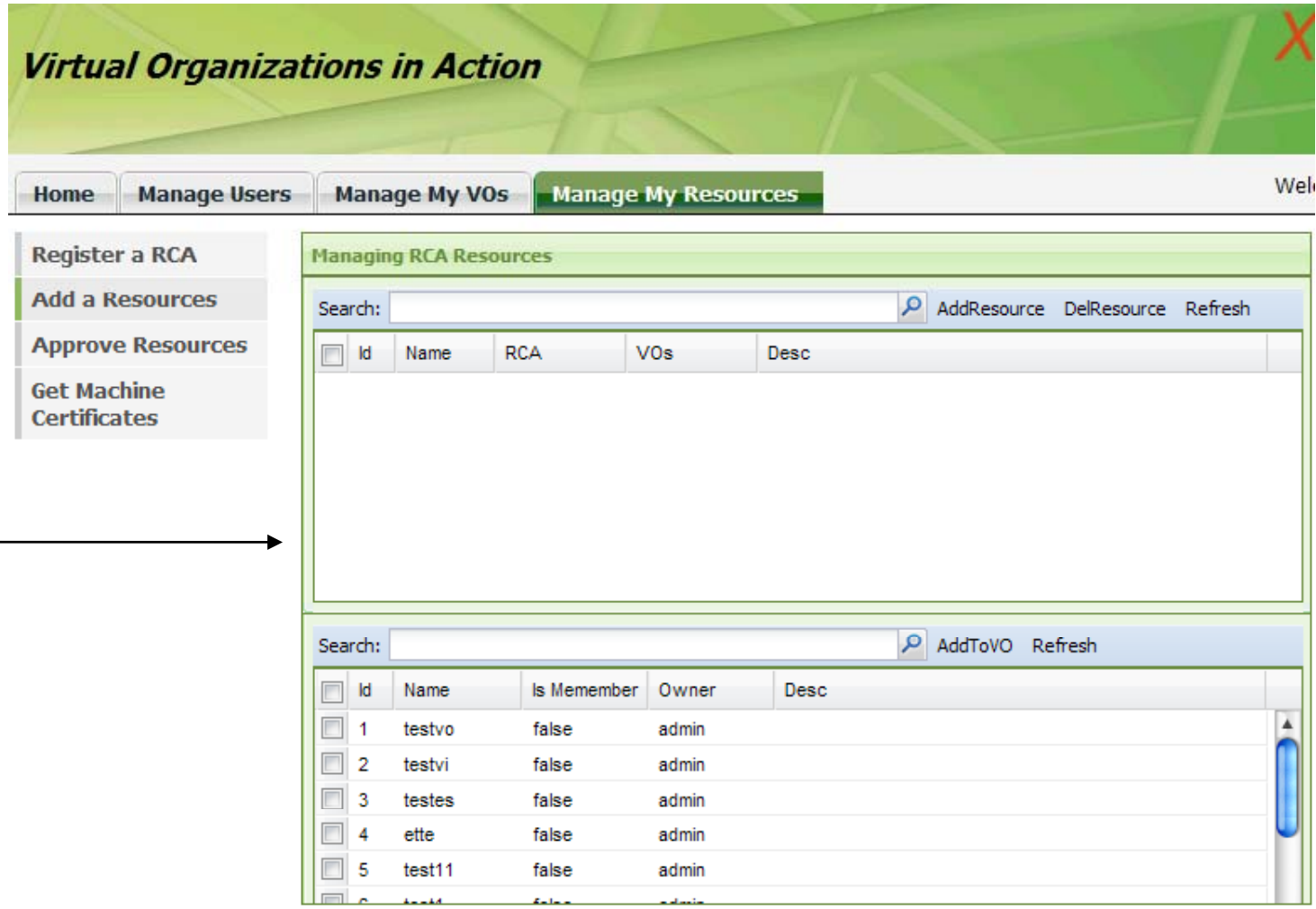
Select VO and send joining requests



Manage VO Resources (site admin)

Manage resources in a VO

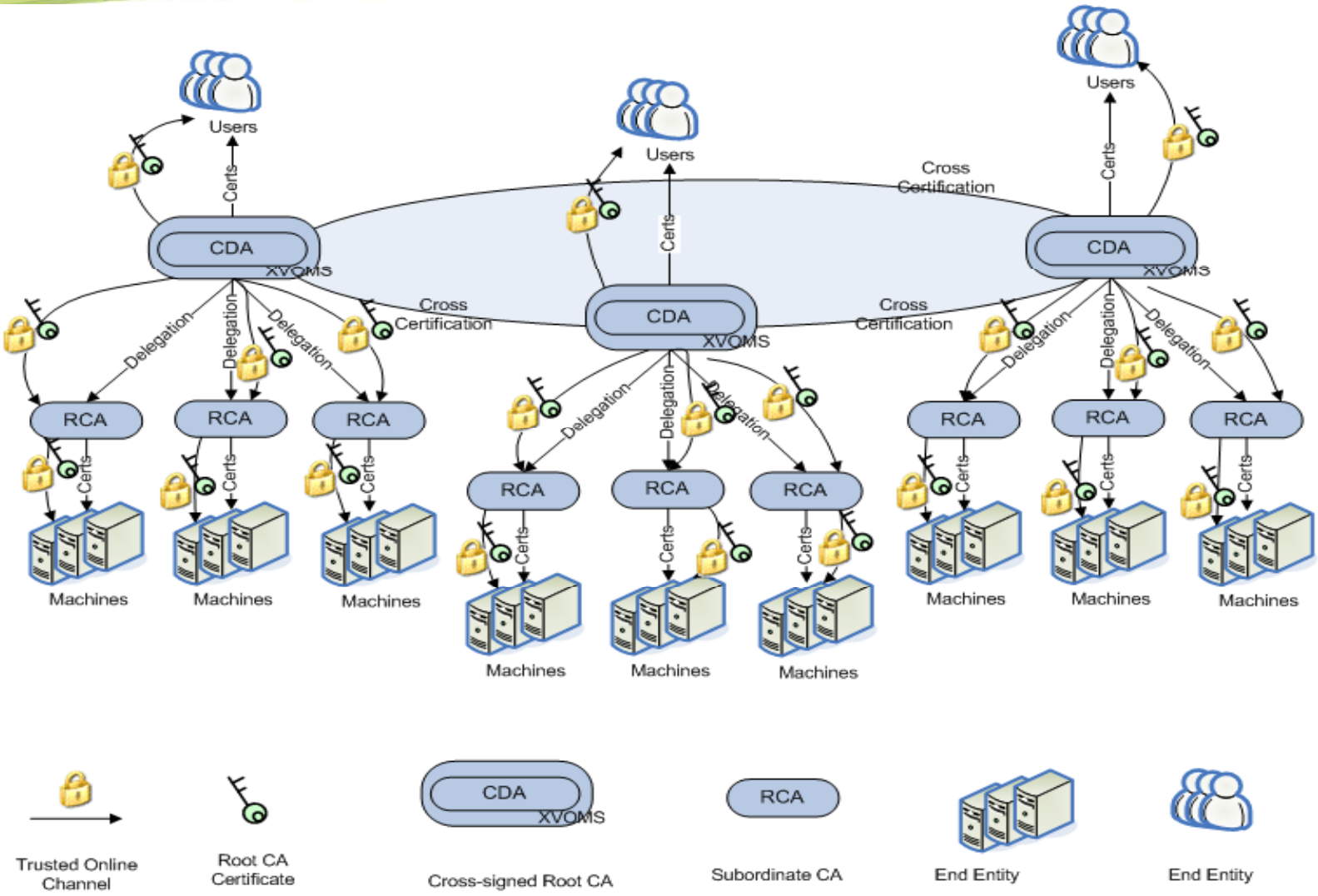




PKI Trust Model in XtreemOS

- CDA: cross-certified root CAs for VO users & RCA certification
- RCA: subordinate CAs for resource certification
- Identifiers & attributes

Trust Model


Users
Certs
CDA
XVOMS
Cross Certification
Delegation
RCA
Machines
Trusted Online Channel
Root CA Certificate
Cross-signed Root CA
Subordinate CA
End Entity
End Entity

Namespaces

- **Global namespaces**
  - GUID, GVID, GGID, GNID
    - Identifiers
    - Global IDs are unique
  - Users and nodes have X.509 certificates
    - Identity stored in the distinguished name (DN)
- **Node-level (local to resources) namespaces**
  - OS users (UID/GID)
  - Files (inodes)
  - Processes (PIDs)
- **VO namespaces**
  - Groups, role, capability

Node-level Mapping between Namespaces

- **On-the-fly mapping between namespaces**
- **Mapping between different namespaces managed by local service `xos-amsd`**
  - GUID ★ UID
  - GGID ★ GID
- **With the support of `nsswitch`**
  - `ls -l` shows the GUID of the file owner

User Session Service

- **User session services**
  - Started when the user logs in
  - In charge of validating user credentials
  - Trusted by XtreemOS operating system services
  - Bridging the user space with the operating system space
  - All grid requests go through the user session service
  - Support untrusted client nodes
- **Provide Single-Sign-On**
- **Provide Delegation**
  - Can be replicated on resource nodes

Isolation, Job context

- **Job context created**
  - When a user session is opened on some resource
  - Can be
    - Simple Unix account
    - Control groups
      - limit/protect resource usage
      - Accounting, billing
    - Namespaces (PID, user, net, ...)
      - Restrict visibility from job context
      - Net namespaces restrict access to Internet
    - Containers (~ cgroups + namespaces)
    - Virtual machines

Node-level Session Management

- **`pam-xos`: modules in charge of authentication, authorisation and session management**
- **`ssh-xos`: extends ssh authentication with XOS certificates**
  - Provides same account mappings as for jobs

Executing Applications in XtreemOS

- **JSDL file to describe the application & its resource requirements**
  - Will be made transparent in future versions of XtreemOS
- **No global scheduler**
  - Job manager service created for each job
- **Resource discovery on peer-to-peer overlay**
  - Structured overlay for faster access to requested resources
  - Resource negotiation
  - VO policies checked during discovery

Application Execution Management

- **Features**
  - No assumption on local node RMS
    - AEM can be used without any batch system
  - Job “self-scheduling”
    - Best effort resource allocation strategy
  - Resource reservation & co-allocation
  - Unix-like job control
  - Monitoring & accounting
    - Accurate and flexible monitoring of job execution
  - Support for interactive applications
  - Interface for workflow engine
  - Checkpointing service for grid jobs

AEM Architecture

AEM Client

Communication

XOSD

Communication

Job Manager

Execution Manager

Resource Manager

Reservation Manager

Communication

Job Directory

(JobID,
@JobOwner…)

Resource Matching

Resource Selection Service

Distributed Services

Resource Description (GLUE)

Communication

Job Manager

Execution Manager

Resource Manager

Reservation Manager

Communication

Communication

Job Manager

Execution Manager

Resource Manager

Reservation Manager

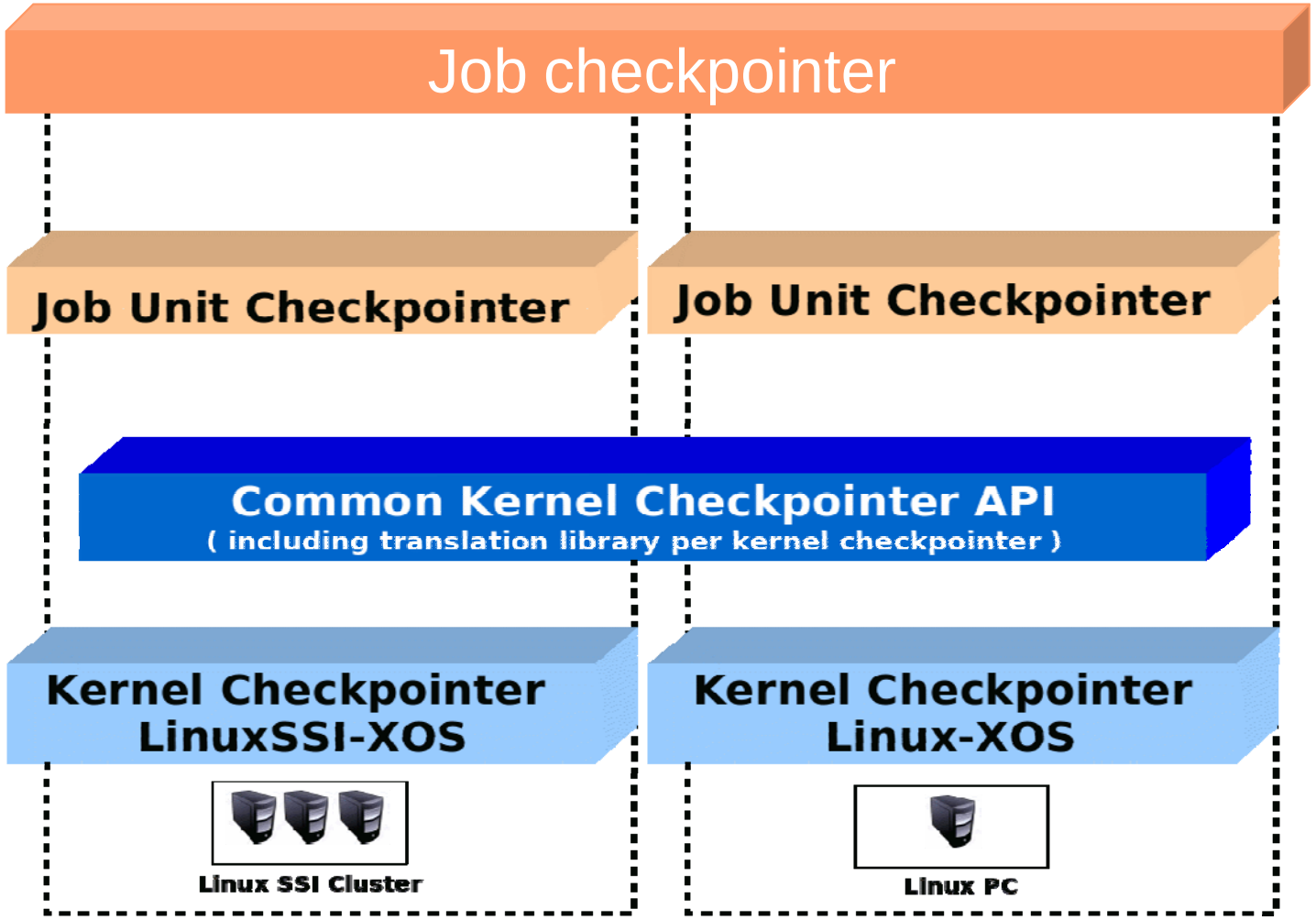
Communication

Resource nodes

Checkpointing Service

- **Goal: checkpointing and restart for grid jobs**
  - Fault tolerance
  - Migration (scheduling / load balancing)

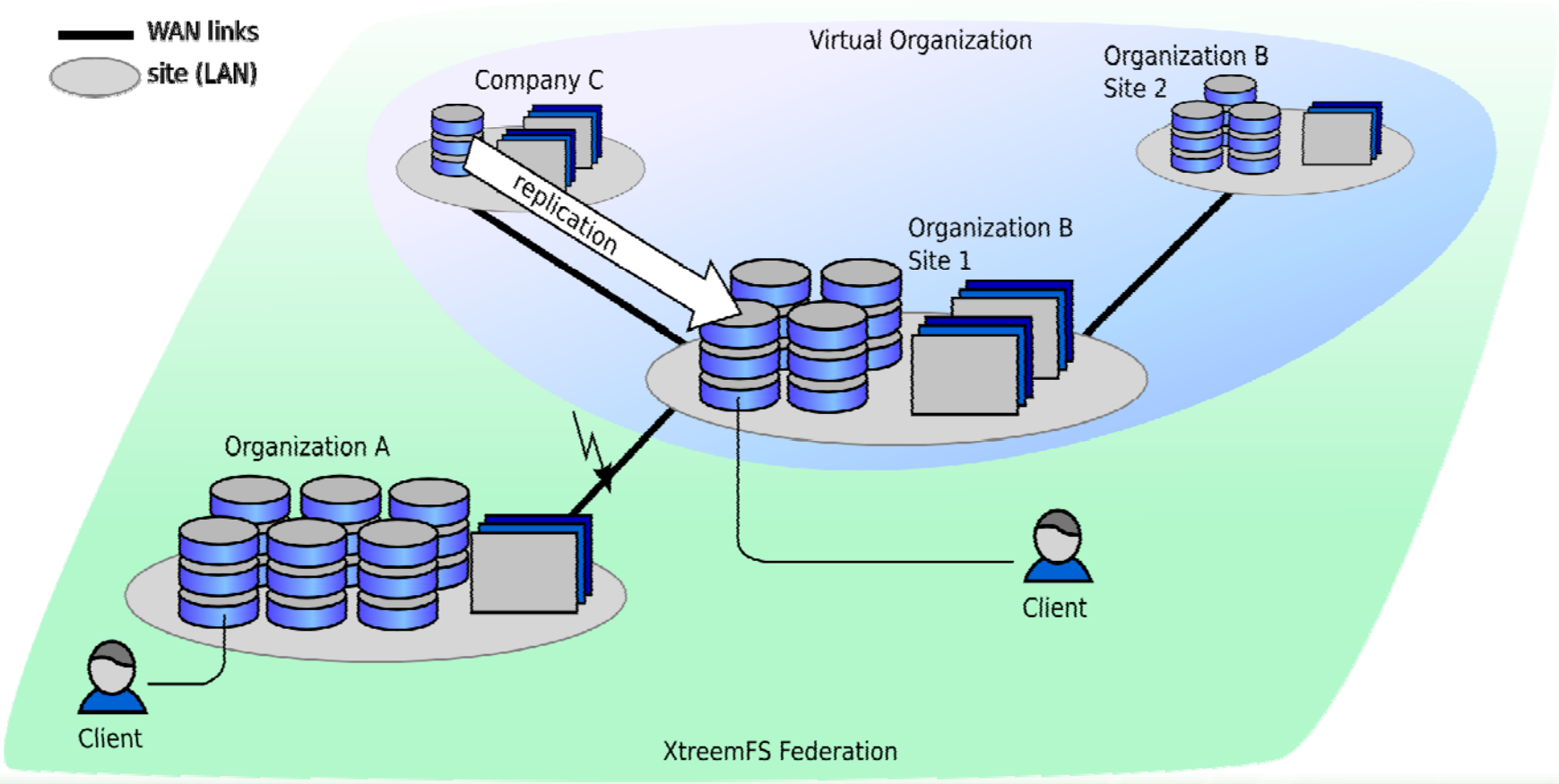
Checkpointing Service



Checkpointing Service

- **Checkpointing protocols for Grid applications**
  - Coordinated checkpointing
  - O2P protocol (optimistic message logging protocol)
- **Kernel checkpointers**
  - BLCR & Kerrighed checkpointer
    - Adapted for Grid usage (callbacks)
    - Steps for applications running on several Grid nodes
  - Checkpointers based on Linux kernel virtualization mechanisms
    - containers, name spaces…
- **Security**
- **Checkpoint storage**
  - XtremFS
  - Garbage collection

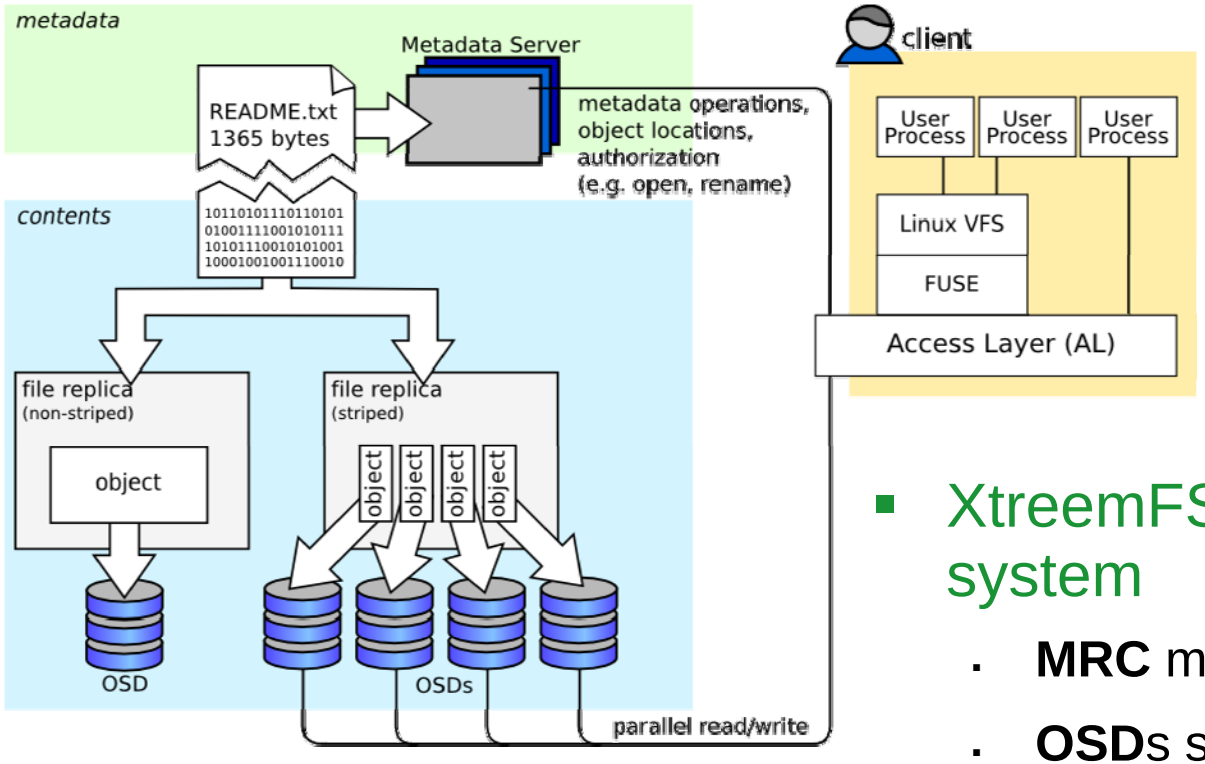
Federating storage in different administrative domains



XtreemFS Features

- **Provide users a global view of their files in a Grid**
- **Transparent location-independent access to data**
  - Data storage in different administrative domains
  - Grid users from multiple VO
- **Consistent data sharing**
- **Access control based on Grid attributes**
  - VO member credentials
- **Each XtreemOS user has a home volume in XtreemFS**

XtreemFS: Architecture



- XtreemFS: an object-based file system
  - **MRC** maintains metadata
  - **OSD**s store file content
  - **Client** (Access Layer) provides client access

XtreemFS Features

- **POSIX compatible file system**
  - File system API
  - Behaviour as defined by POSIX or local file system
- **Advanced metadata management**
  - Replication
  - Partitioning
  - Extended attributes and queries

XtreemFS Features

- **Replication of files**
  - primary/secondary with automatic failover
  - fully synchronous to lazy data replication
  - POSIX compatible by default
- **Striping (parallel read and write)**
- **RAID and end-to-end checksums**
- **Client-side caching and cache consistency**
- **Autonomous data management with self-organized replication and distribution**
  - Access pattern-based replica management (RMS service)

Conclusion

- **Validation with a set of reference applications in progress**
  - Aladdin/G5K large-scale platform
  - Permanent test bed
- **On-going research activities**
  - **Advanced features**
    - Accounting, isolation
    - Very dynamic VO
    - High availability of XtreemOS critical services
  - **XtreemOS & cloud computing**
    - XtreemOS as a software infrastructure for cloud federations
    - Exploitation of virtual machines dynamically provisioned from clouds in an XtreemOS Grid

More about XtreemOS

- **Open software Development**
  - http://gforge.inria.fr/projects/xtreemos/
  - Second major release under preparation (GPL/BSD)
    - Packaged for Mandriva & RedFlag Linux distibutions
- **Information**
  - http://www.xtreemos.eu
  - Public deliverables and technical reports
- **Contact**
  - info@xtreemos.eu
  - Christine.Morin@inria.fr

Meet XtreemOS Team

- **XtreemOS Summer School, Oxford, UK**
  - September 7-11, 2009
  - Registration open, see http://www.xtreemos.eu/xtreemos-events/xtreemos-summer-school-2009

- **Demonstrations on XtreemOS booth**
  - June 23-26, 2009**: ISC '09**, Hamburg, Germany
  - November 14-20, 2009: **SC '09**, Portland, Oregon, USA

- http://www.xtreemos.eu
- http://gforge.inria.fr/projects/xtreemos/

Thank you for your attention

Questions?