



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 D5.2.4

Due date of report: September 30st, 2010
Actual submission date: October 15th, 2010

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 1.7 / Last edited by Michael Schöttner / Date: October 14, 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	02/09/2010	Michael Schöttner	UDUS	Outline
1.1	15/09/2010	Michael Schöttner	UDUS	First draft
1.2	30/09/2010	Michael Schöttner	UDUS	Added executive summary, introduction, and conclusions
1.3	12/10/2010	Massimo Coppola	CNR	Added key player event
1.4	13/10/2010	Michael Schöttner	UDUS	Include numbers for all events, overview table, polishing
1.5	13/10/2010	Sandrine L'Hermitte	INRIA	Extensions + statistics on summer school participants
1.6	13/10/2010	Michael Schöttner	UDUS	Polishing
1.7	14/10/2010	Michael Schöttner	UDUS	Final polishing

Reviewers

Franz Hauck (ULM) and Guillaume Pierre (VUA)

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

1. Introduction	5
2. Training engineers and users	6
2.1 Internal training activities.....	6
2.2 External training activities	6
4. XtreamOS summer schools	9
4.1 First Summer School (September 2009)	9
4.2 Second Summer School (July 2010)	11
5. XtreamOS day for key players.....	14
5.1 Industrial workshop in Italy (June 2010)	14
5.2 Industrial workshop in Slovenia 2010.....	15
6. Conclusion.....	16
Appendix	17
A.1 Tutorial: “Security and VO Management in Grids” at ISC09	
A.2 Tutorial: “Easing Application Execution in Grids with XtreamOS Operating System” at OGF28	
A.3 Tutorial: “Grid and Cloud Computing with XtreamOS” at Eurosys 2010	
A.4 2 nd XtreamOS Summit – XtreamOS challenge part	

Executive Summary

This deliverable provides the final report on training activities during the fourth year of the project (June 2009 – September 2010). This document is based on the actions planned in Annex 1 and completes the three previous training reports provided in D5.2.1, D5.2.2, and D5.2.3.

During the last year of the project no internal training activities were necessary and all the efforts were spent on training for external users. Several tutorials were presented at important scientific conferences (e.g. EuroSys 2010) and at OGF meetings (e.g. OGF28). Both summer schools, which are major training events directly linked to the project research topics, were very successful and attracted many active external participants. We have also organized two XtremOS summits (i.e. one-day workshop) co-located with the yearly EuroPar conference. These summits were privileged occasions to meet people interested people in the XtremOS technology.

Finally, one event targeting key industrial players was organized in Pisa and another one is planned in Slovenia this fall.

Overall all training activities have attracted 168 external people and resulted in valuable feedbacks on the developed software.

1. Introduction

The main goal of WP5.2 in XtreamOS was the implementation of training activities for internal and external people. 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. As expected at this stage of the project, there was no need for internal training during the last year of the project.

In contrast, external training targets parties, which are not members of the consortium, but are interested in the topics related to the project. During the last year many external training activities were organized by partners ranging from tutorials at scientific conferences to the XtreamOS summits and summer schools.

Finally, decisions makers were targeted by a special industrial workshop organized in Pisa. Another event for key players is planned in Slovenia shortly after the contractual end of the project.

2. Training engineers and users

2.1 Internal training activities

No internal training sessions were performed as at this stage of the project, there were no more requests from XtremOS members for internal trainings.

2.2 External training activities

Tutorial: “Security and VO Management in Grids” (June 2009)

Yvon Jegou (INRIA), Christine Morin (INRIA), Haiyan Yu (ICT), Corina Stratan (VUA) gave a half-day tutorial “Security and VO Management in Grids” at the International Conference on Supercomputing (ICS’09), New York, USA, on June 12, 2009. See also conference website: <http://pcsostres.ac.upc.edu/ics-conference/archive/ics09/workshops.html#5>



This tutorial provided an overview of security and Virtual Organization management in established and new Grid systems. The security and Virtual Organization management features provided by some major Grid middleware packages were surveyed, and the comparable functionality in XtremOS, a Grid-based operating system was introduced. Concepts in Grid security were introduced, including their respective challenges and protection mechanisms. Globus, gLite and UNICORE middleware packages have been described, showing the services they provide, their VO management functions, and security abilities. The tutorial then explored the features of the XtremOS Grid operating system, demonstrating the advantages of close integration between Grid functionality and operating system facilities. The slides of this tutorial can be found in Appendix A.1.

The target audience were grid users, grid developers and grid administrators. Overall there have been 19 participants: 4 from the consortium and 15 registered external participants.

First XtremOS Summit (August 2009)

The 1st XtremOS Summit was a one-day workshop co-located with the EuroPar 2009 conference in Delft, The Netherlands, on August 25, 2009. See also our website:

<http://www.xtreemos.eu/project/xtreemos-events/xtreemos-summit-at-euro-par-2009>.



The XtremOS summit included talks about XtremOS services and components. The summit started with a talk about the main objectives of XtremOS (Thilo Kielmannm, VUA), followed by talks about the security model (Alvaro Arenas, STFC), the resource matching approach (Guillaume Pierre, VUA), and parallel IO and replication in XtremFS (Bjoern Kolbeck, ZIB). Afterwards selected demonstrations showed the benefits of using XtremOS technology (Peter Linnell, INRIA). Finally, the summit concluded with an open discussion slot. The summit was moderated by Michael Schöttner (UDUS).

The summit was targeting grid users, grid application developers, grid system and middleware designers. Overall there have been 24 participants: 6 from the consortium and 18 external participants.

Tutorial: “Easing Application Execution in Grids with XtreamOS Operating System” (March 2010)

Christine Morin (INRIA), Yvon Jégou (INRIA), Toni Cortes (BSC), Michael Schöttner (UDUS) and Thilo Kielmann (VUA) gave a half-day tutorial “Easing Application Execution in Grids with XtreamOS Operating System” at OGF28 meeting at Ludwig-Maximilians-Universität München, Munich, Germany, on March 15, 2010, see also http://www.ggf.org/gf/event_schedule/index.php?id=1948.



This tutorial presented XtreamOS Grid system from the user point of view. Its goal was to show why XtreamOS is a system of choice for Grid users who want to easily execute their applications while taking advantage of advanced features for job management. We have showed what can be done with XtreamOS with regards to application execution, how it can be done and how the available features (AEM, resource discovery, and reliable job execution, support for interactive applications, SAGA API) are implemented. Demonstrations have been presented throughout the tutorial. The slides can be found in Appendix A.2.

The target audience was also grid users, grid developers and grid administrators. Overall there have been 10 participants: 5 from the consortium and 5 external participants.

Tutorial: “Grid and Cloud Computing with XtreamOS” (April 2010)

Corina Stratan (VUA), Massimo Coppola (CNR) and Guillaume Pierre (VUA) gave a half-day tutorial “Grid and Cloud Computing with XtreamOS” at the European Conference on Computer Systems (EuroSys), Paris, France, on April 13, 2010, see also: <http://eurosys2010.sigops-france.fr/tutorials.html>.



The goal of this tutorial was to teach the audience about XtreamOS giving users the illusion of working with a traditional computer while removing the burden of complex resource management issues of typical distributed environments. Furthermore, how existing POSIX and Grid applications are transparently handled, automatically exploiting container and virtualization support when needed.

After introducing the general concepts of the domain, the tutorial presented the XtreamOS functionalities (VOs, job submission and management, XtreamFS and XtreamOS administration) and demonstrated how they can be used. The slides are listed in Appendix A.3.

This tutorial targeted users, application developers, system and middleware designers of cloud and grid systems. Overall there have been 23 participants: 3 from the consortium and 20 external participants.

Tutorial: “Distributed Computing with XtreamOS” (July 2010)

Given by Corina Stratan (VUA) at the GridInitiative summer school, Bucharest, Romania, July 23, 2010. The summer school took place at the Politehnica University of Bucharest. This was a 1.5h presentation, pretty much based on the EuroSys 2010 tutorial, but this time also including a small live demo on the XtreamOS development testbed.

This tutorial targeted users, application developers, system and middleware designers of cloud and grid systems. Overall there have been 31 participants: 1 from the consortium and 30 external participants.

Second XtreamOS Summit (August 2010)

The 2nd XtreamOS Summit was a one-day workshop co-located with EuroPar conference in Ischia, Italy, on August 30, 2010. Beside a tutorial, live demos were presented and the best competitors of the XtreamOS challenge presented the results of their experimentations on XtreamOS, see also [http://www.xtreemos.eu/project/xtreemos-events/XtreamOS summit 2010](http://www.xtreemos.eu/project/xtreemos-events/XtreamOS%20summit%202010)

The objectives of this summit were to teach participants the usage of main XtreamOS services (Virtual Organization management & grid security mechanisms, application execution management, XtreamFS - distributed data storage etc.) and to present XtreamOS from the user's point of view including demonstrations of main XtreamOS functionalities.



One of the highlights were the presentation of the best XtreamOS computing challenge competitors, see also <http://www.xtreemos.eu/project/xtreemos-events/xtreemos-challenge/>. The slides of these presentations can be found in Appendix A.4.

The tutorial targeted users, application developers, system and middleware designers of cloud and grid systems. Overall there have been 25 participants: 5 from the consortium and 20 external participants.

4. XtremOS summer schools

4.1 First Summer School (September 2009)

The 1st XtremOS Summer School was held at Wadham College in Oxford, UK, on September 7-11, 2009. The objectives were 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, and to provide a forum for participants to discuss your research work and share experience with experience researchers. Detailed information can be found here: <http://www.xtreemos.eu/project/xtreemos-events/xtreemos-summer-school-2009>.



An overview of the program and structure of the summer school is shown in Figure 1. There were three categories of talks: invited lectures, presentations from consortium members, and talks from school participants in the doctoral symposium. These talks were complemented by various practical sessions.

XtremOS 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
10:30 – 11:00		Coffee Break			
11:00 – 12:30		Grid Programming Interface – SAGA Thilo Kielmann, VUA	Practical on XtremFS Bjorn Kolbeck, ZIB	Highly Scalable Services Massimo Coppola, CNR	Distributed State Game Management Michael Sonnenfroh, U. Dusseldorf
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	

Figure 1, First XtremOS summer school program

All invited talks are listed below:

- “*The Hitchhiker’s Guide to the Data Centers Galaxy*”,
Paolo Costa, Microsoft Research Cambridge, UK
- “*Cloud Computing with Nimbus*”,
Kate Keahey, Argonne National Laboratory, USA

- “*The UK National Grid Service and University of Oxford Campus Grid, Production E-Infrastructures Supporting Interdisciplinary Research*”,
David Wallom, Oxford e-Research Centre, UK
- “*Distributed Checkpoint / Restart for the HPC*”,
Daniel Lezcano, IBM, France

Thursday afternoon was dedicated to discuss research topics with PhD and MSc students attending the school. A call for participation was launched beforehand and students were strongly encouraged to present their research work.

- “*Managing security of grid architecture with a grid security operation center*”,
Syed Raheel Hassan
- “*A software distribution network in a virtualized infrastructure*”,
Jakob Blomer
- “*Beernet: a relaxed-ring approach for P2P networks with transactional replicated DHT*”,
Boriss Mejias
- “*Dynamic process re-organisation for a grid-wide DSM data sharing service*”,
Jesus de Oliveira
- “*Fine-grained parallelism with SVP*”,
Raphael Poss
- “*Towards an auxiliary agent architecture for cloud computing*”,
Xiaobin Xiao

All the slides of the summer school are available here: <http://www.xtreemos.eu/project/xtreemos-events/xtreemos-summer-school-2009/slides>. Furthermore videos from school can be found here: <http://www.xtreemos.eu/project/xtreemos-events/xtreemos-summer-school-2009/videos>

Overall there have been 29 participants: 9 from the consortium and 20 external participants (6 of them were awarded a grant). Figure 2 shows the countries where the external participants came from and Figure 3 their position.

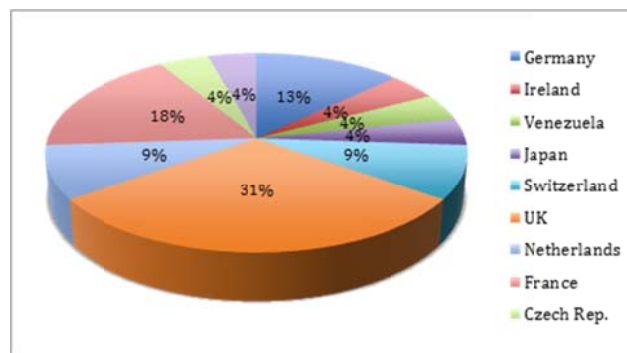


Figure 2, home countries of the external participants

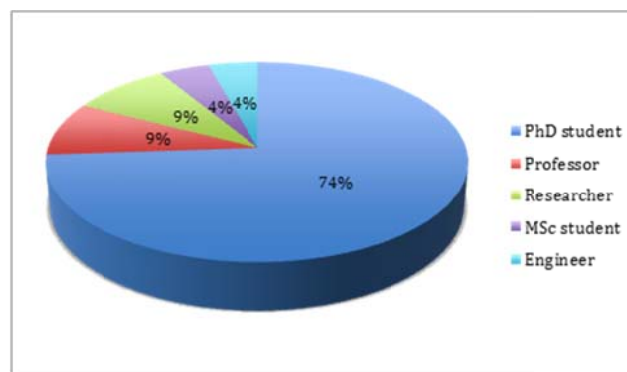


Figure 3, positions of the external participants

4.2 Second Summer School (July 2010)

The 2nd XtreamOS Summer School was held at Reisenburg Castle, Science Center of Ulm University, on July 5-9, 2010. The objectives were 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, and to provide a forum for participants to discuss your research work and share experience with experience researchers. Detailed information can be found here: <http://www.xtreemos.eu/project/xtreemos-events/summer-school-2010>



An overview of the program and structure of the summer school is shown in Figure 4. As the first summer school was very successful the format was kept. There were again three categories of talks: invited lectures, presentations from consortium members, and talks from external people in the doctoral symposium. And these talks were complemented by various practical sessions.

The invited Lectures were:

- *“Building Clouds with OpenNebula: A Grid Computing Perspective”*,
Ruben Montero, Complutense University of Madrid, Spain
- *“DGSI: Federation of Distributed Compute Infrastructures”*,
Bernhard Schott, Platform Computing, Frankfurt, Germany

Thursday afternoon was dedicated to discuss research topics with PhD and MSc students attending the school. A call for participation was launched beforehand and students were strongly encouraged to present their research work.

- *“Storage Deduplication in Cloud Computing”*,
João Paulo, University of Minho, Portugal
- *“Twitter workload for NoSQL databases”*,
Francisco Cruz, University of Minho, Portugal
- *“Broker Overlay for Decentralized Grid Management”*,
Abdulrahman Azab, University of Stavanger, Norway
- *“Grid Workflow Job Execution Service ‘Pilot’*,
Lev Shamardin, Moscow State University, Russia
- *“Dynamic Counter-Based Broadcast in MANETs”*,
Sarah Omar al-Humoud, University of Glasgow, UK

The demo sessions included following demonstrations:

- Core demo: VO administration and job submission and resource submission
- Mobile device Grid Player
- Mobile device 3G connection sharing
- Wissenheim virtual presence game
- XtreamFS demos: first steps, replication with failover - disconnect cable and video still plays, read-only replication
- Other applications: COMP Superscalar - protein analysis, SpecWeb2005 - web server benchmarking

All the slides of the summer school are available here: <http://www.xtreemos.eu/project/xtreemos-events/summer-school-2010/summer-school-2010-program>.

-Draft programme -

Time	Monday July 5	Tuesday July 6	Wednesday July 7	Thursday July 8	Friday July 9
09:00-10:30	Arrival of participants	VO Mgmt & security (A. Arenas)	Scalaris: Pub/Sub system (T. Schuett)	Virtual Nodes (J.Domaschka)	Grid Checkpointing (J. Mehnert-S.)
10:30-11:00		XtreemFS File System (J. Stender)	Application Execution Mgmt (R. Nou)	Byzantine Fault Tolerance (C. Spann)	Object Sharing Service (M. Schoettner)
11:00-12:30		Practical session XtreemFS (J.Stender)	Practical session AEM (R.Nou/J.Giralt)	Practical session vnodes (J.Domaschka / S.Kächele/ C.Spann)	Invited talk: Bernhard Schott (DGSi coordinator, Platform Computing)
12:30-14:00	Lunch break				
14:00-14:30	Registration	Cluster flavour/ Kerrighed (J. Parpaillon)	Tuto:how to port an application to XOS? (M. Sterk)	XtreemOS applications	Departure of participants (optional: guided tour in Ulm)
14:30-14:45	Welcome				
15:00-16:30	Opening talk (C. Morin)	Testbed / deployment (Y. Jégou)	Grid Application Programming (T. Kielmann)	Coffee break	
16:30-17:00	Invited talk: Ruben S. Montero (OpenNebula, UCM)	Doctoral symposium (1)	Practical session SAGA (T.Kielmann)	Doctoral symposium (2)	
17:00-17:45					
		XOS technical demo (M. Sterk)			
19:00-20:00	Welcome reception (incl. Poster/demo)	Dinner	Dinner	School dinner	
from 20:00		FIFA world cup semi-final 1	FIFA world cup semi-final 2		

Figure 4, Second XtreemOS summer school program

Overall there have been 35 participants: 21 from the consortium and 14 external participants (2 of them were awarded a grant). Figure 5 shows the countries where the external participants came from.

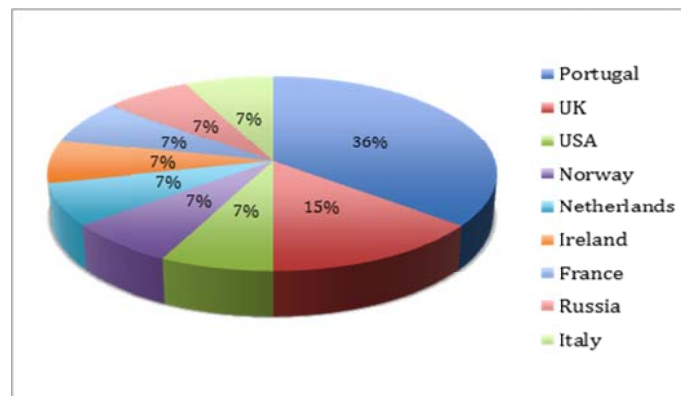


Figure 5, home countries of the external participants

Furthermore, we took the opportunity of this second summer school to shoot a promotional video for the XtreemOS system. The idea was to make a friendly video in which XtreemOS key members explained the benefits of using our Grid OS.

The video is now online both on our project website and on YouTube:

http://www.youtube.com/watch?v=tgfKWe_ruUY

A technical report completes these video tutorials: Yvon Jégou, “Installing XtremOS on a Virtual Machine”, XtremOS technical report #5, October 2010. Available for download here:

<http://www.xtreemos.eu/project/publications/VMInstall.pdf>

Almost all the participants in the 2010 XtremOS summer school were PhD students.



5. XtreamOS day for key players

5.1 Industrial workshop in Italy (June 2010)

On June 16th 2010, CNR organized in Pisa, the following full-day workshop “X-HPC: Experimenting with XtreamOS to provide HPC services”.



The objectives of this workshop were to strengthen the liaison with industrial partners of the XtreamOS operating system. To advertise among potential industrial adopters the project results, as well as the already existing industrial involvement in the project, to increase awareness of academic and research institutions of both XtreamOS and its positioning with respect to the EU foreseen roadmap toward Future Internet and Virtualized Platforms for Cloud computing, and to foster the involvement of the IT centre of the University with the XtreamOS experimentation.

This international event mainly targeted key players, mainly from Italy but also from abroad, both from industries and academia. It was organized in collaboration with the newly instituted IT centre of the University of Pisa (<http://www.itc.unipi.it>, formed in December 2009, with official inauguration end of October 2010), leveraging the IT centre and XtreamOS partners into a synergy that could disseminate the results of the XtreamOS Consortium.

XtreamOS partners as well as several industrial groups leveraging Hi-tech IT and HPC were contacted. In spite of the effort to set up the workshop in a suitable period, some of the contacted partners could not attend due to other international meetings.

More information about the workshop can be found here http://hpc.isti.cnr.it/?page_id=62.

The workshop was also publicly advertised via the ERCIM News bulletin N.81, April 2010, on page 65: <http://ercim-news.ercim.eu/images/stories/EN81/EN81-web.pdf>.

Technical Program

- Opening and welcome,
Prof. P.Ferragina (IT center Director)
- “*XtreamOS an Open-Source Distributed Operating System approaches Cloud Computing*”,
Dr. Christine Morin (INRIA)
- “*Cross roads: industry, research and teaching at the IT Center*”,
Dr. Antonio Cisternino (University of Pisa)
- “*Cloud Computing in the Strategies for the Future Internet*”,
Dr. Domenico Laforenza (CNR-IIT, director)
- “*The XtreamOS VO-oriented Security Model*”,
slides by Alvaro Arenas (STFC), presented by Massimo Coppola (CNR-ISTI)
- “*Kerrighed - XtreamOS Cluster Flavour*”,
Jean Parpaillon (KER)
- “*Octopus, Scheduling Virtual Machines*”,
Davide Morelli, Marco Mura (IT centre, University of Pisa)
- “*Parallel programming issues, achievements and trends in HPC and adaptive computing*”,
Prof. Marco Danelutto (University of Pisa)
- “*Innovation with Acer Servers*”,
Sean Stacey (ACER/Gateway)
- “*HPC data intensive applications: case studies from the CASPUR end-user community*”,
Dr. Nico Sanna (CASPUR)

- *“High Performance Computing with NVIDIA Tesla GPUs”*,
Dr. Timothy Lanfear (NVIDIA)
- *“General Purpose Programming on GPUS”*,
Cristian Dittamo (University of Pisa)

Attending Representatives

From the XtreamOS consortium:

- Christine Morin (INRIA), Jean Parpaillon (Kerlabs), Massimo Coppola, Susanna Martinelli, Emanuele Carlini, Nicola Tonello (CNR-ISTI), Domenico Laforenza (CNR-ISTI and CNR-IIT), Fabio Martinelli, Paolo Mori (CNR-IIT)

From the IT center

- Prof. Antonio Cisternino, Maurizio Davini, Prof. Paolo Ferragina (director), Davide Morelli, Marco Mura (IT Center and University of Pisa)

From other institutions

- Sean Stacey, Edy Schwartz (Acer-Gateway HPC division)
- Timothy Lanfear (NVIDIA)
- Nico Sanna (CASPUR - Italian Inter-University Consortium for the Application of Super-Computing for Universities and Research, <http://www.caspur.it/en/>)
- Prof. Enzo Barone (Scuola Normale Superiore di Pisa, <http://www.sns.it/en/>)
- Domenico Dato (Tiscali SpA, R&D Manager)
- Dr. Tommaso Cucinotta (SSSUP, Scuola Superiore di Studi Universitari S. Anna, Pisa, <http://www.sssup.it>)
- Prof. M. Danelutto, Cristian Dittamo, Giacomo Righetti (Dept. C.S., University of Pisa)
- Sebnem Erturk (Dept. Physics, University of Pisa)

Other guest attendees were from the Computer Science, Physics and Engineering Depts. of the University of Pisa. Overall there have been 35 participants: 9 from the consortium and 26 external participants (from 7 institutions / groups).

5.2 Industrial workshop in Slovenia 2010

XLAB has applied to host a local key players event co-located with the EuroCloud event organized by the Slovene Chamber of Commerce and scheduled for October 2010. The event will be attended by the decision-makers of many of the largest Slovene IT and telecommunication companies. We plan to promote XtreamOS to cloud computing vendors and are currently waiting for the confirmation from the organizing committee of EuroCloud.

6. Conclusion

During the last year all training activities targeted external people who are potential developers and users of the XtreamOS systems. These training activities include tutorials at important scientific conferences and at OGF meetings. The events that attracted the highest number of external participants were our two summer schools (Oxford, UK, September 2009 and Ulm, Germany, July 2010) and the 2nd XtreamOS summit co-located with EuroPar 2010 (Ischia, Italy, August 2010). Furthermore one event for decision makers was organized in Pisa, Italy, June 2010 showing commercial benefits of the XtreamOS technology.

Several external feedbacks and contributions came from participants in these training events, especially from summer school participants.

Overall all training activities during the last reporting period targeted external people. Below is a summary of participants per event:

Event	Overall number of participants	Number of participants from the consortium	Number of external participants
Tutorial: "Security and VO Management in Grids" (June 2009)	19	4	15
Tutorial: "Easing Application Execution in Grids with XtreamOS Operating System" (March 2010)	10	5	5
Tutorial: "Grid and Cloud Computing with XtreamOS" (April 2010)	23	3	20
Tutorial: "Distributed Computing with XtreamOS" (July 2010)	31	1	30
1st XtreamOS Summit (August 2009)	24	6	18
2nd XtreamOS Summit (August 2010)	25	5	20
1st Summer School (September 2009)	29	9	20
2nd Summer School (June 2010)	35	21	14
Industrial workshop in Italy 2010	35	9	26

Table 1, Number of participants per event

All together 168 external people have attended XtreamOS training activities. Compared with the last years of the project this is a success although these numbers could have been even better. One of the lessons learned is that it would have been easier to attract more external people by providing earlier an open testbed for public access (with ready-to-use XtreamOS configuration). Thus people could more easily test the software without spending much time on installation and configuration.

Appendix

A.1 Tutorial: “Security and VO Management in Grids” at ISC09

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*





▪ **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



- **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



- **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**



- **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**



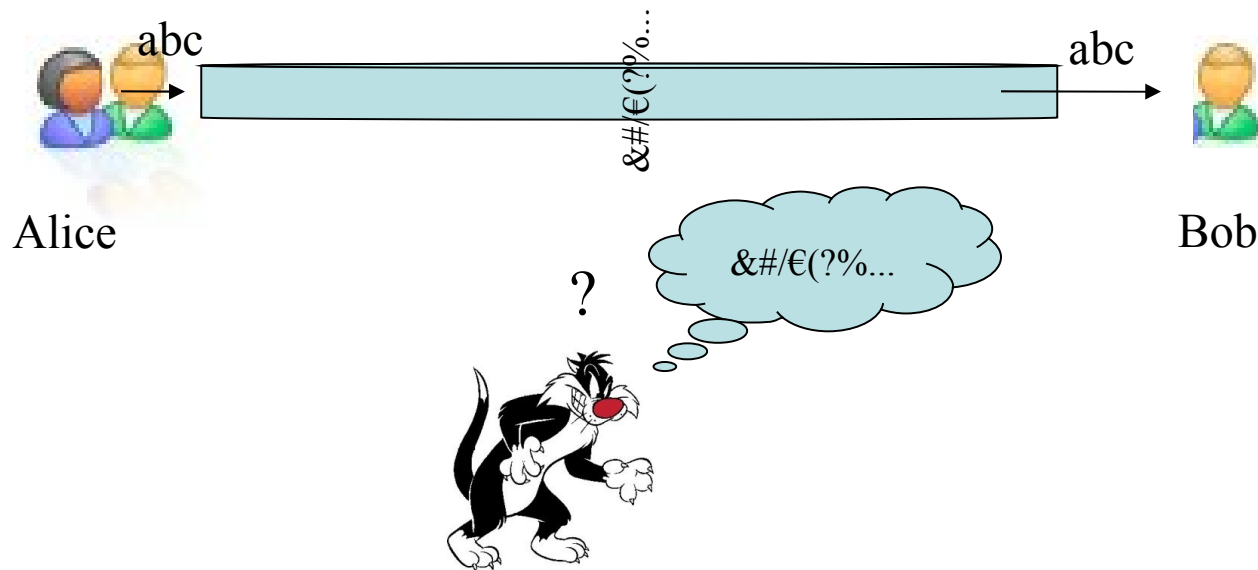
Basic Security Concepts

- **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-repudiation:** 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)



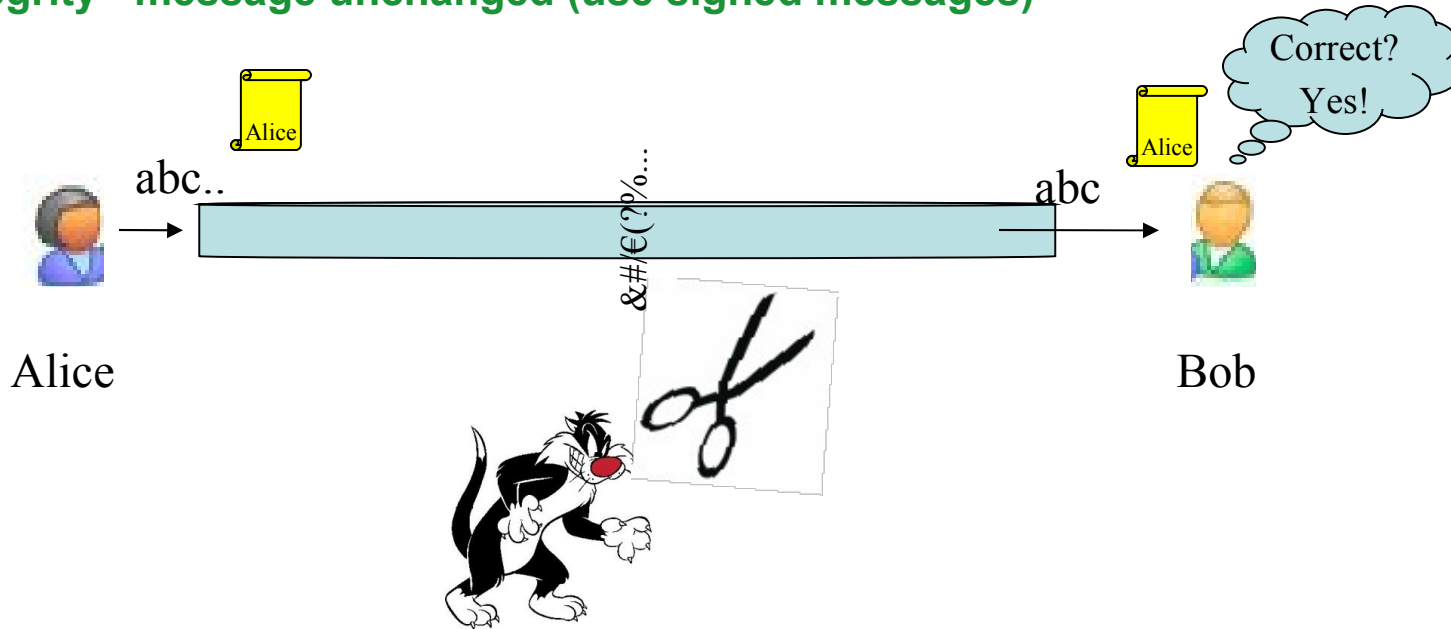
Confidentiality (privacy) - A secure conversation should be private. In other words, only the sender and the receiver should be able to understand the conversation. If someone eavesdrops on the communication, the eavesdropper should be unable to make any sense out of it.

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



Security fundamentals

Integrity - message unchanged (use signed messages)

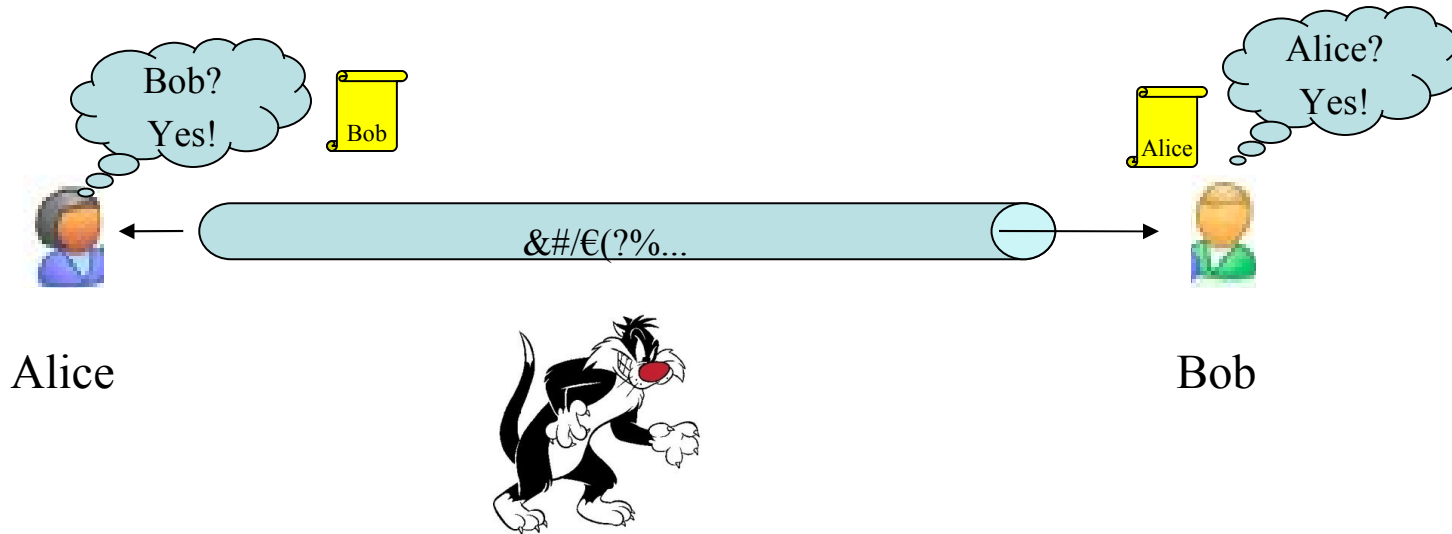


Integrity - A secure communication should ensure the integrity of the transmitted message. This means that the receiving end must be able to know for sure that the message he is receiving is exactly the one that the transmitting end sent him. Take into account that a malicious user could intercept a 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)

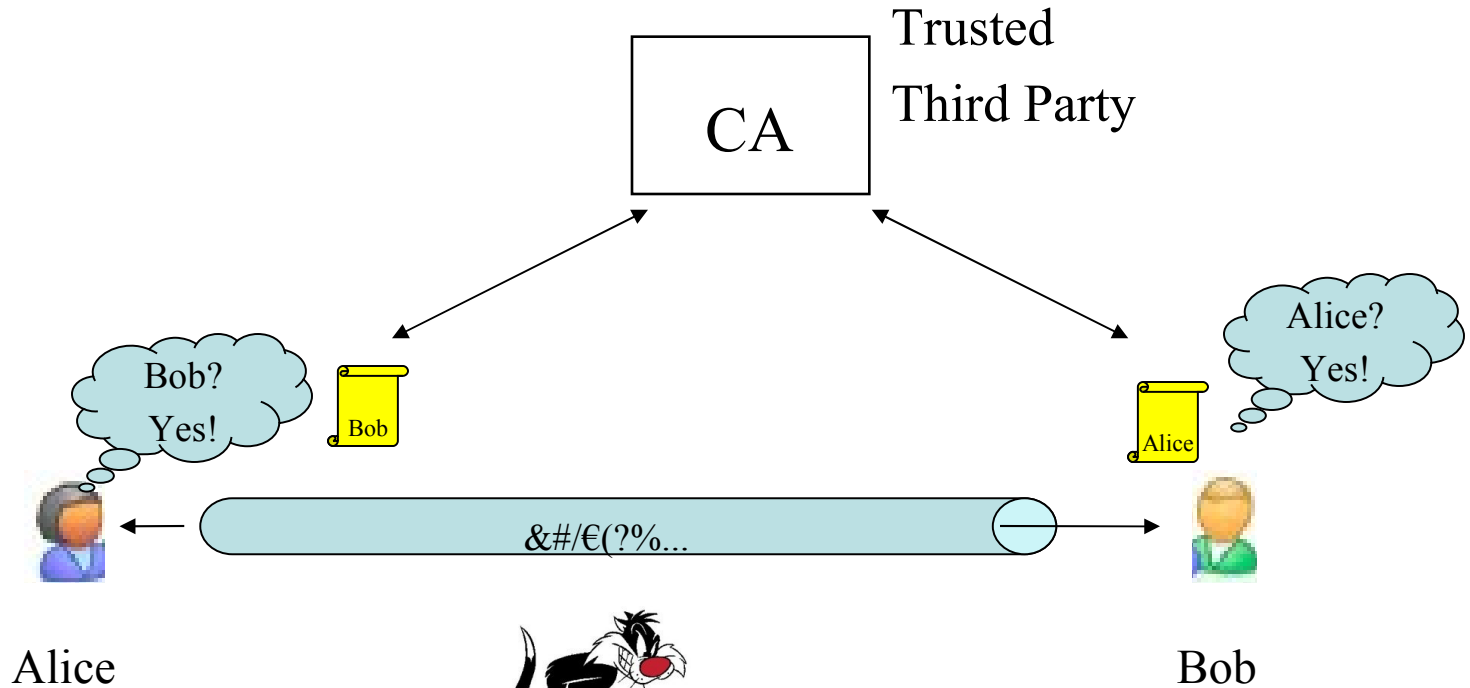


AuthN - A secure communication should ensure that the parties involved in the communication are who they claim to be. In other words, we should be protected from malicious users who try to impersonate one of the parties in the secure conversation.



CA - Certification Authority

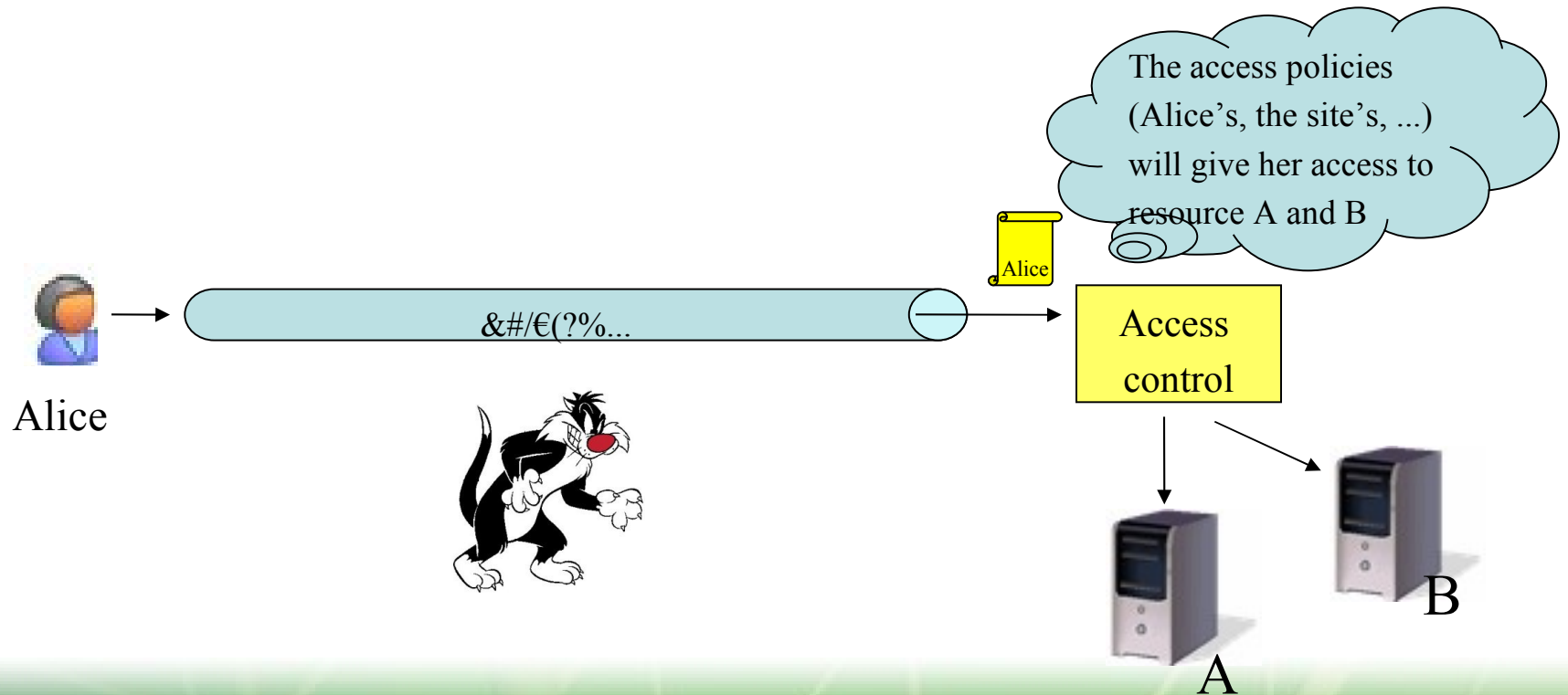
- The role of the CA is manage the certificate life cycle: create, store, renew, revoke





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-mortem 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

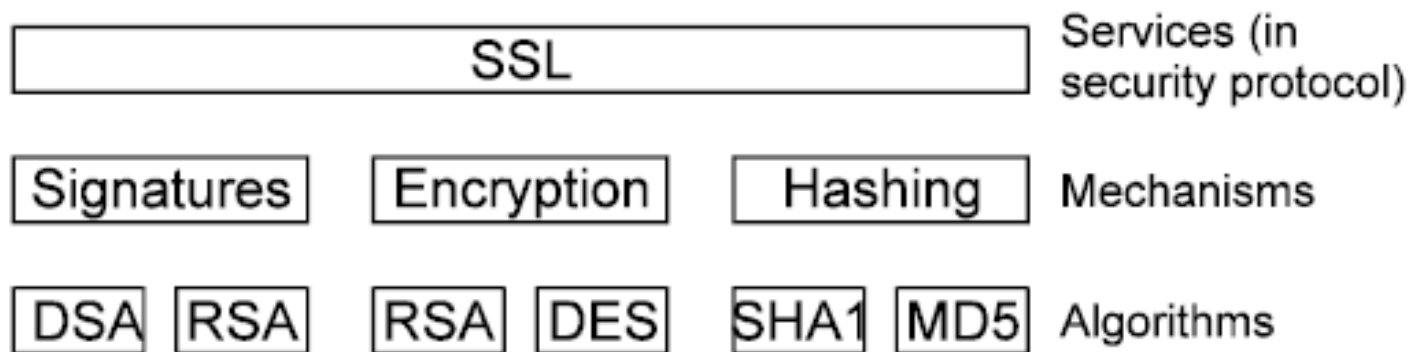
- **Three basic building blocks are used:**
 - **Encryption** is used to provide confidentiality, can also provide authentication and integrity protection
 - **Digital signatures** are used to provide authentication, integrity protection, and non-repudiation
 - **Checksums/hash algorithms** are used 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



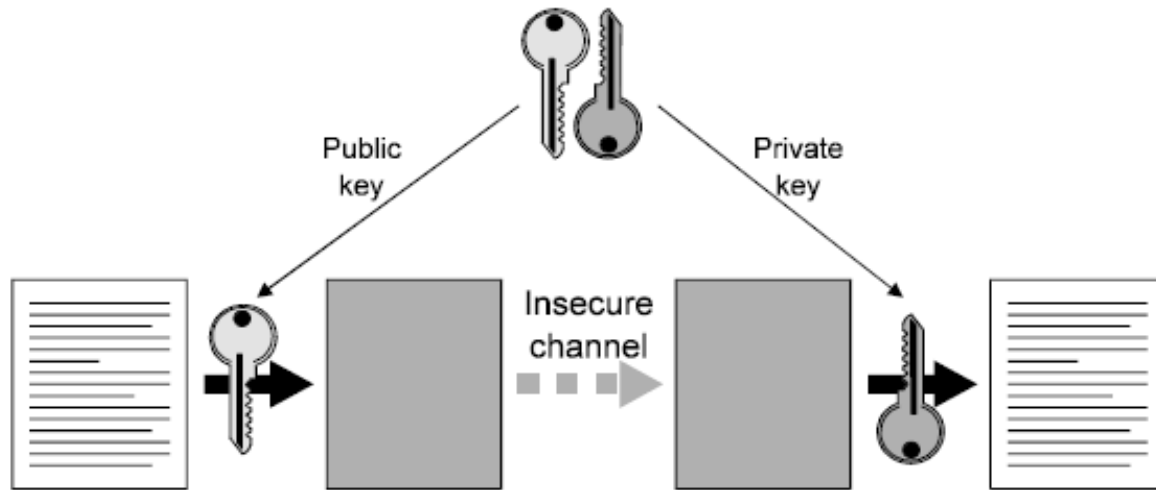
Trust through Cryptography

- **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 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



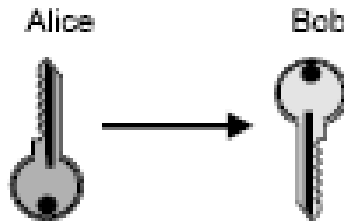
Key Management Problems

- **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

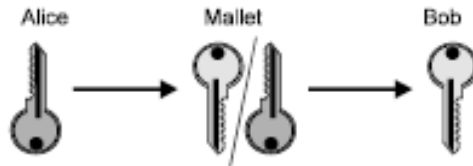


Key Distribution Problems

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



- Mallet intercepts the key and substitutes his own key

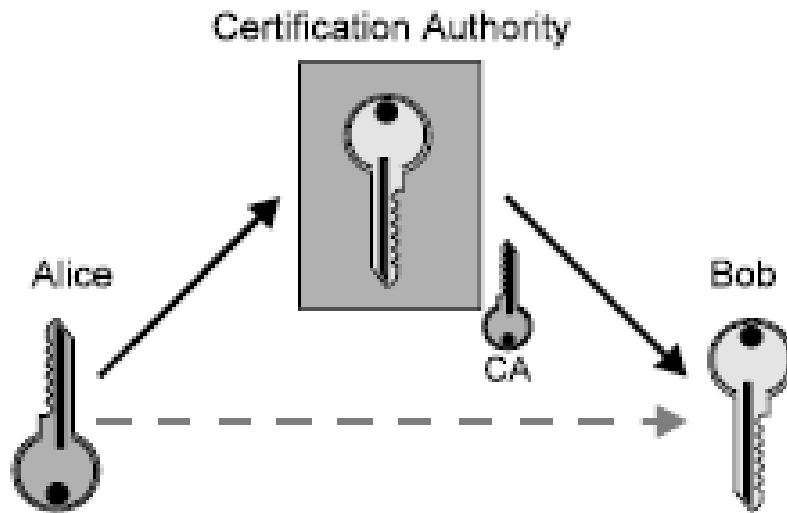


- Mallet can decrypt all traffic and generate fake signed message



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)



"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"



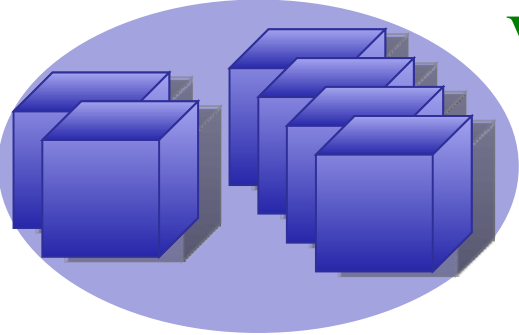
Virtual Organization (VO)

- 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

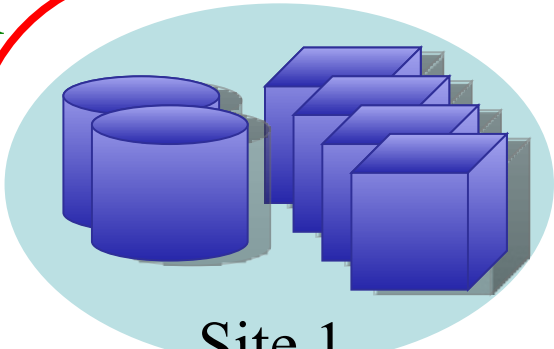


Virtual Organizations

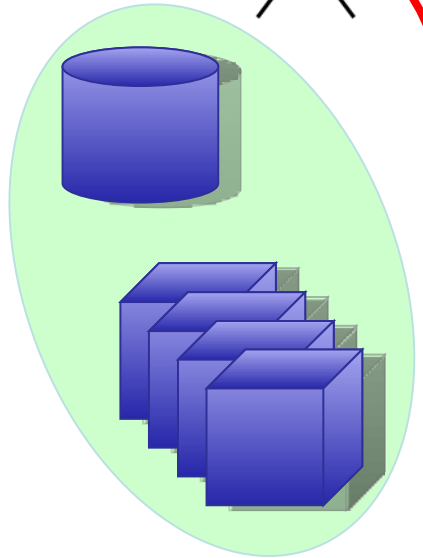
VO A



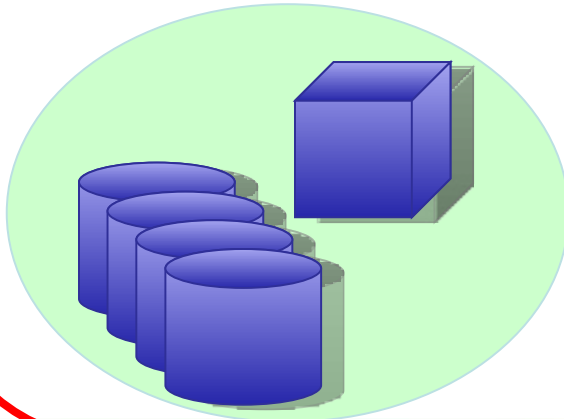
Organization 3



Site 1



Organization 2

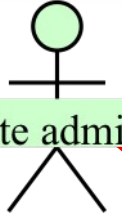


Site 2



Organization 1

Site admin



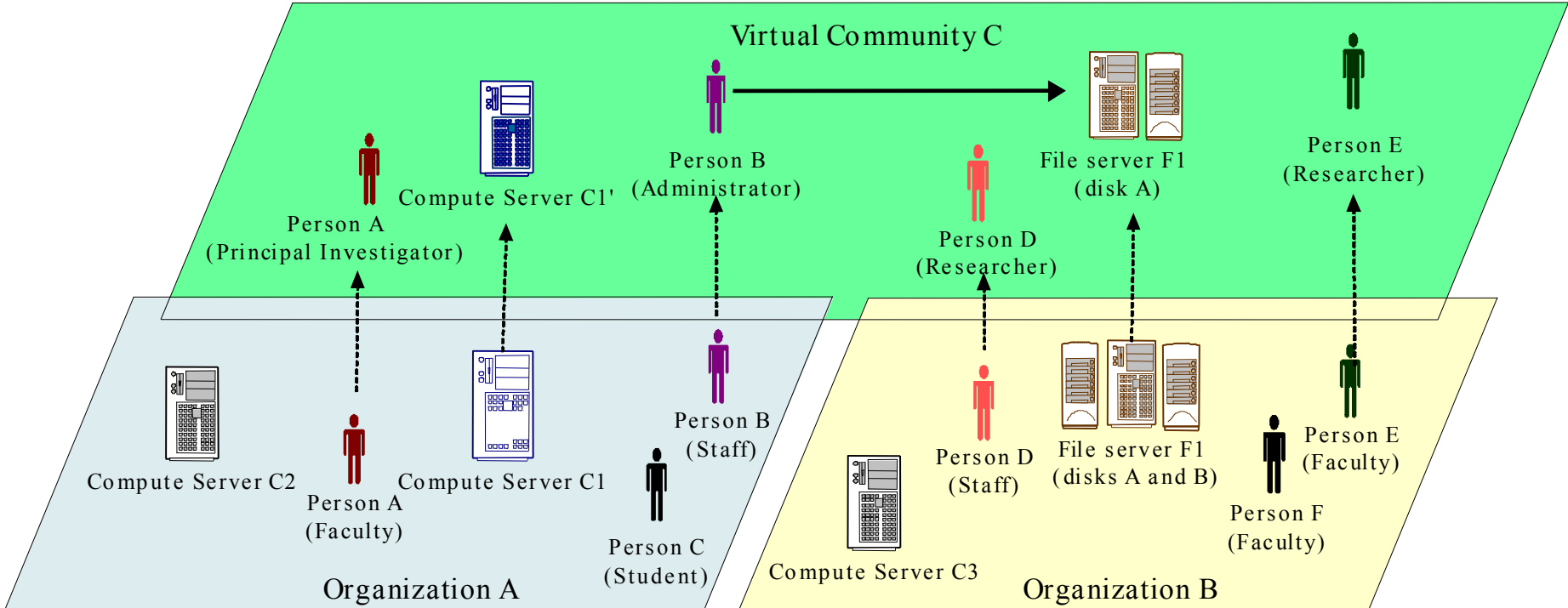
VO admin

VO B



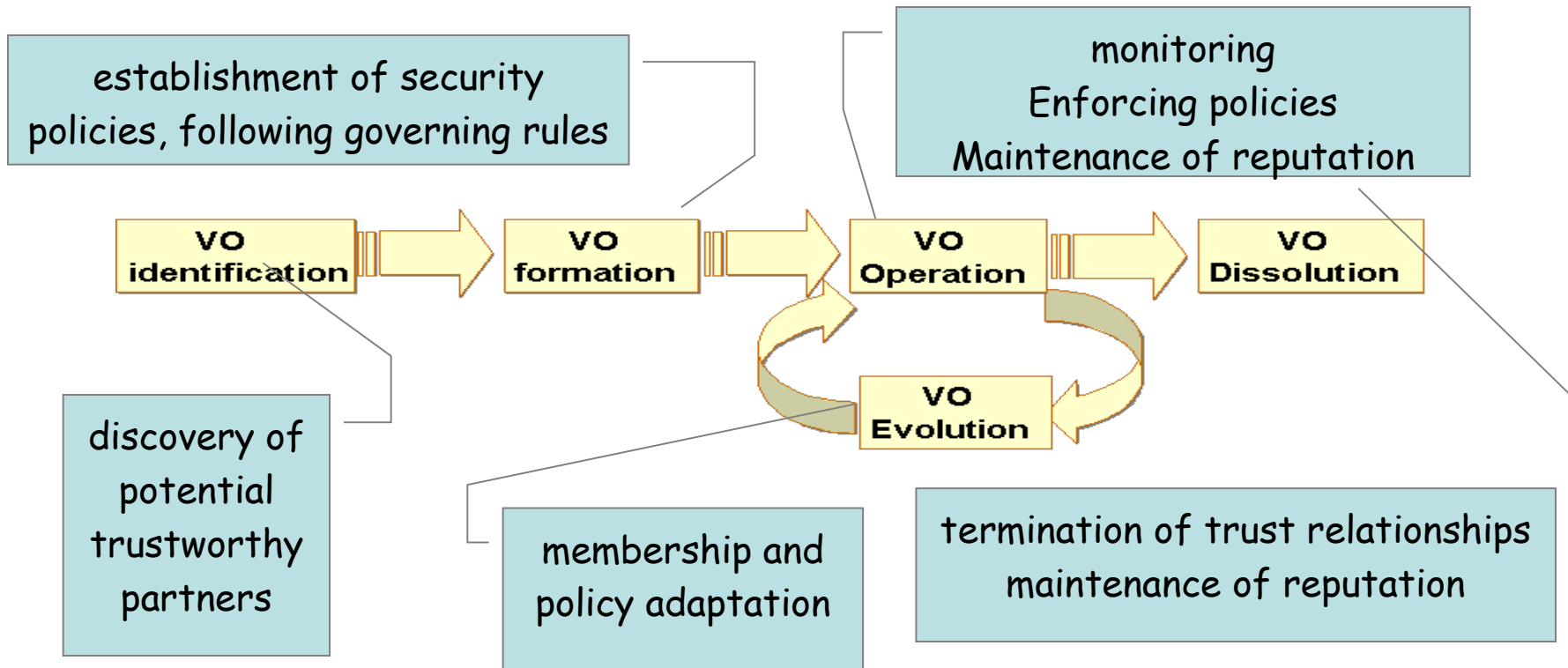


Virtual vs. Organic structure





VO Lifecycle



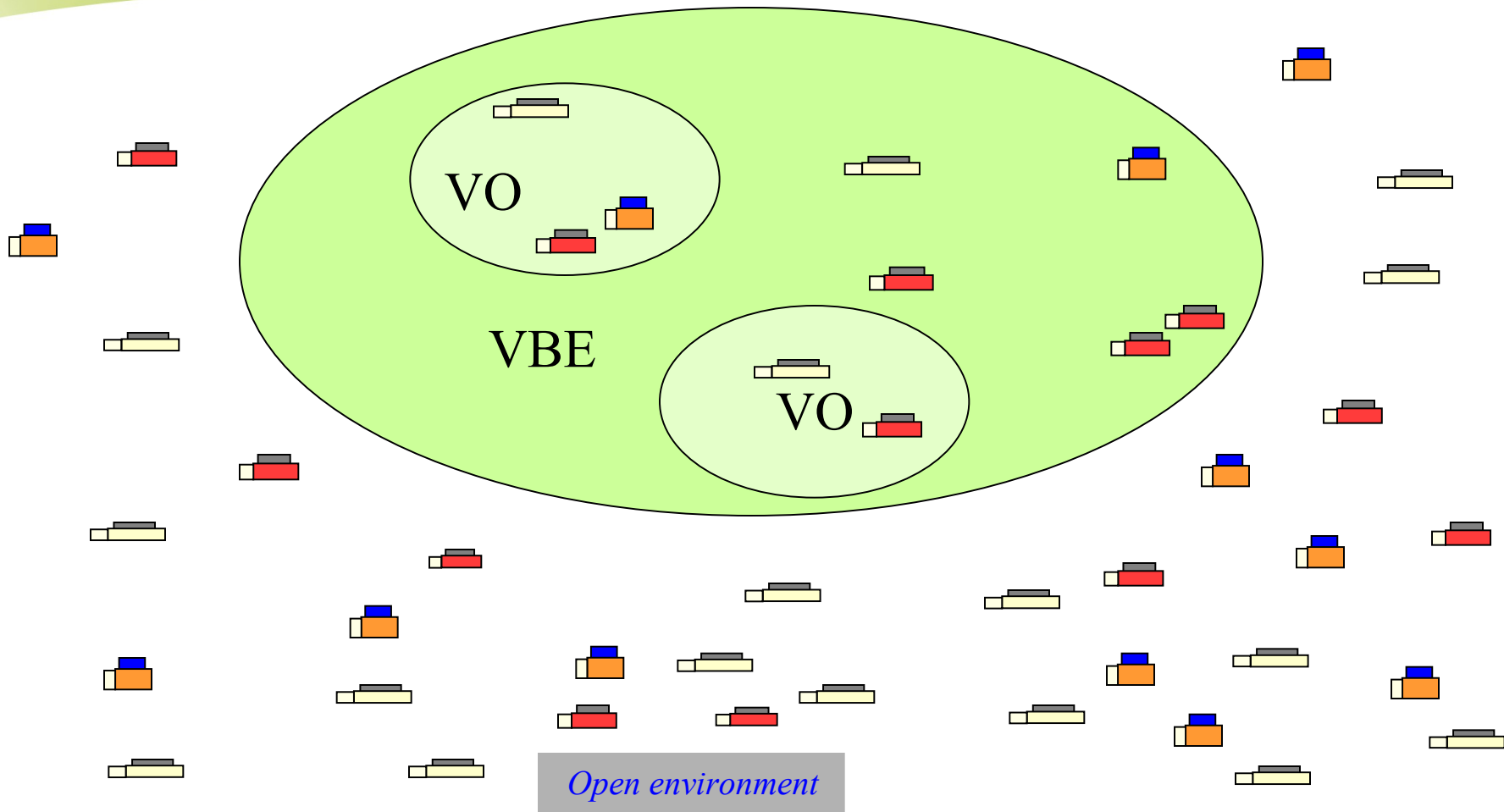


Virtual Breeding Environment

- **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





- **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*





- **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!**



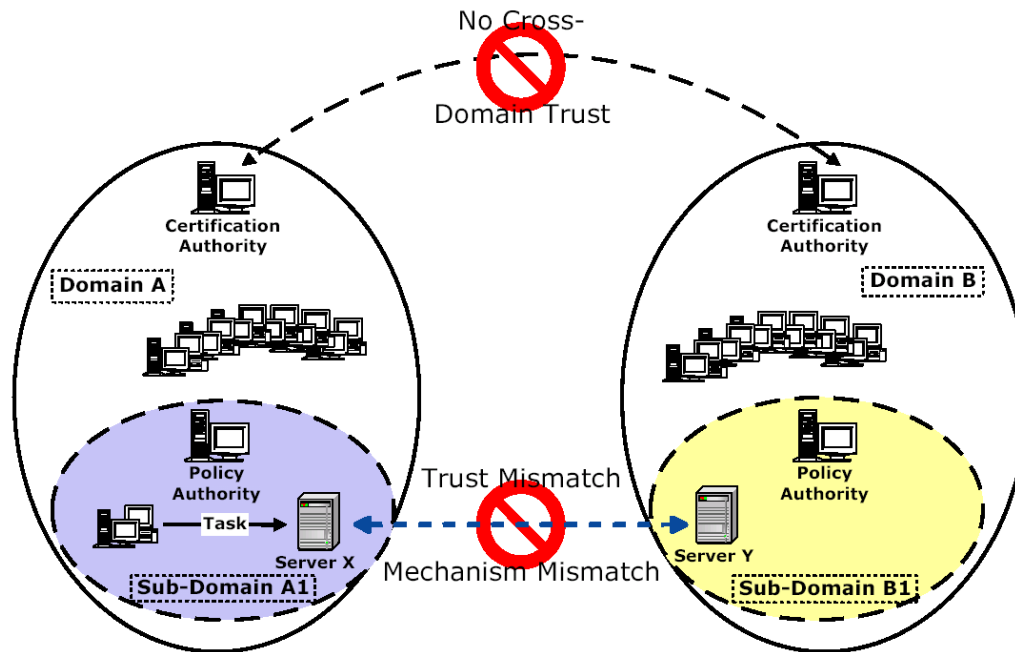
GSI – Grid Security Infrastructure

- **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**

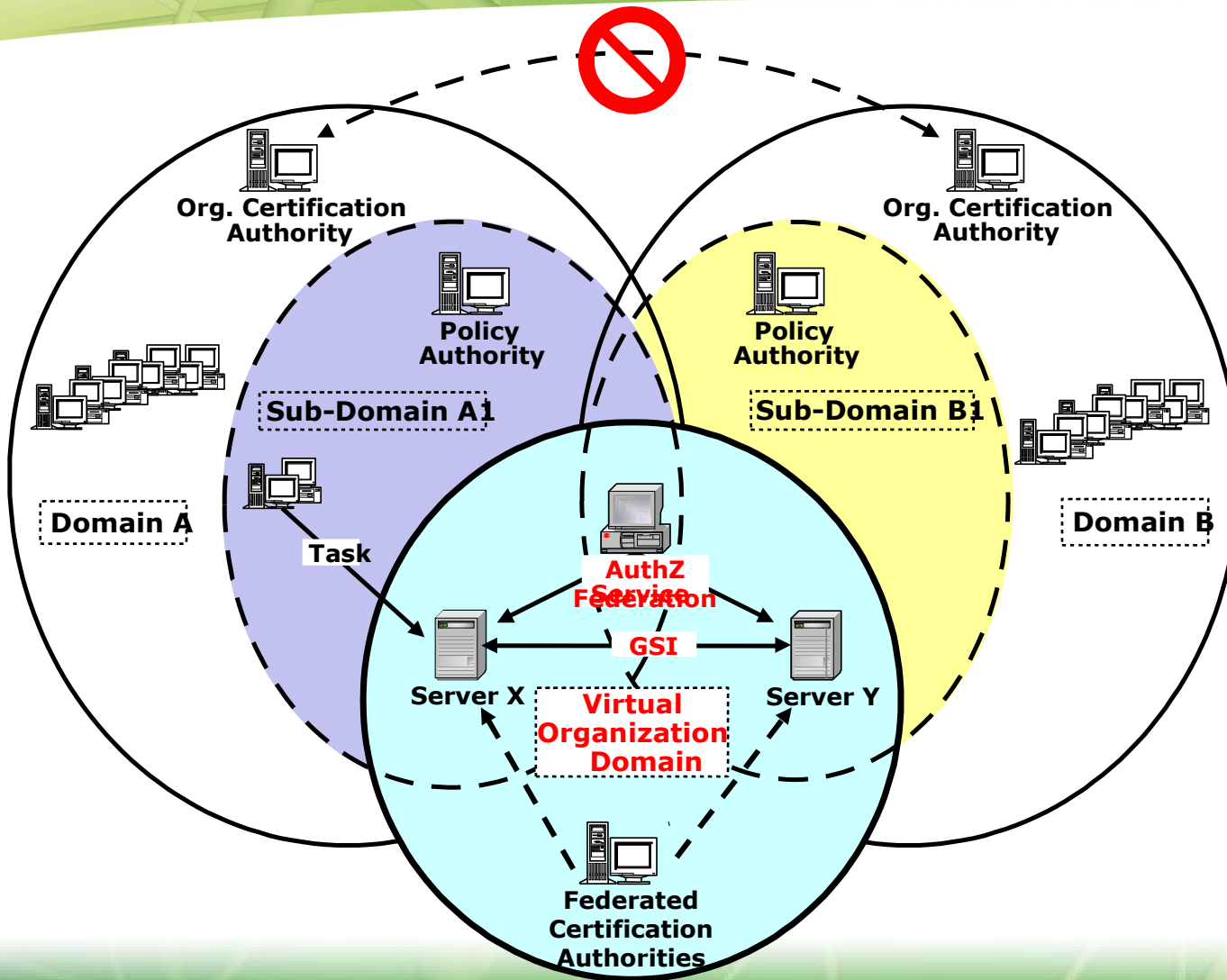
- Organizations trust each other to identify their own users

- **Web of trust**

- Users trust each other to do identify others



Certification Authorities (CAs) in Grid





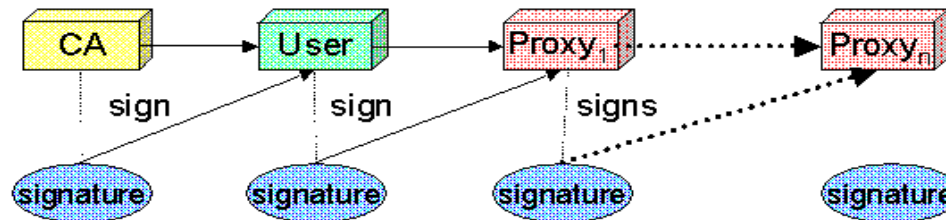
Single sign-on and delegation

- **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



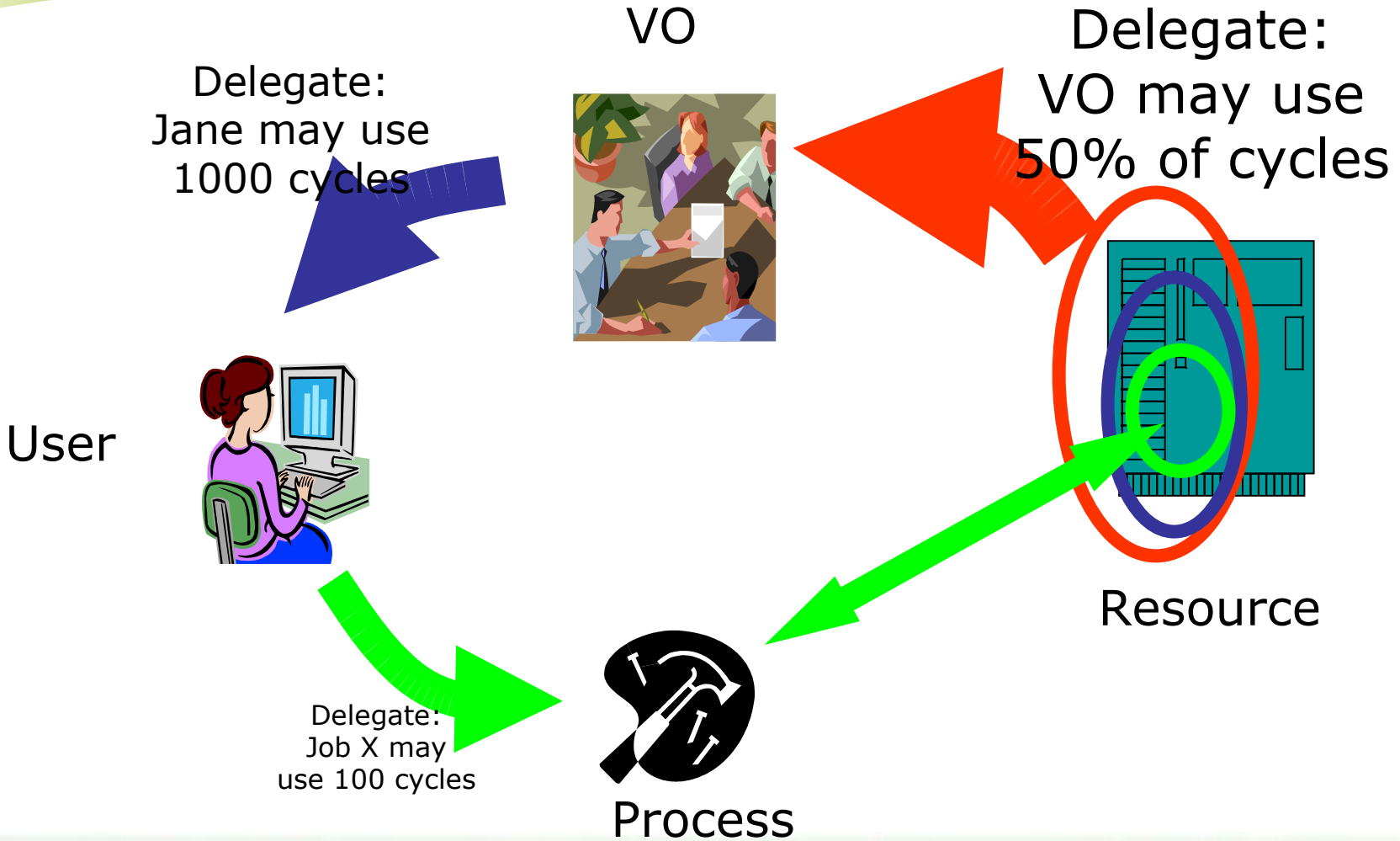
Delegation and limited proxy

- **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)



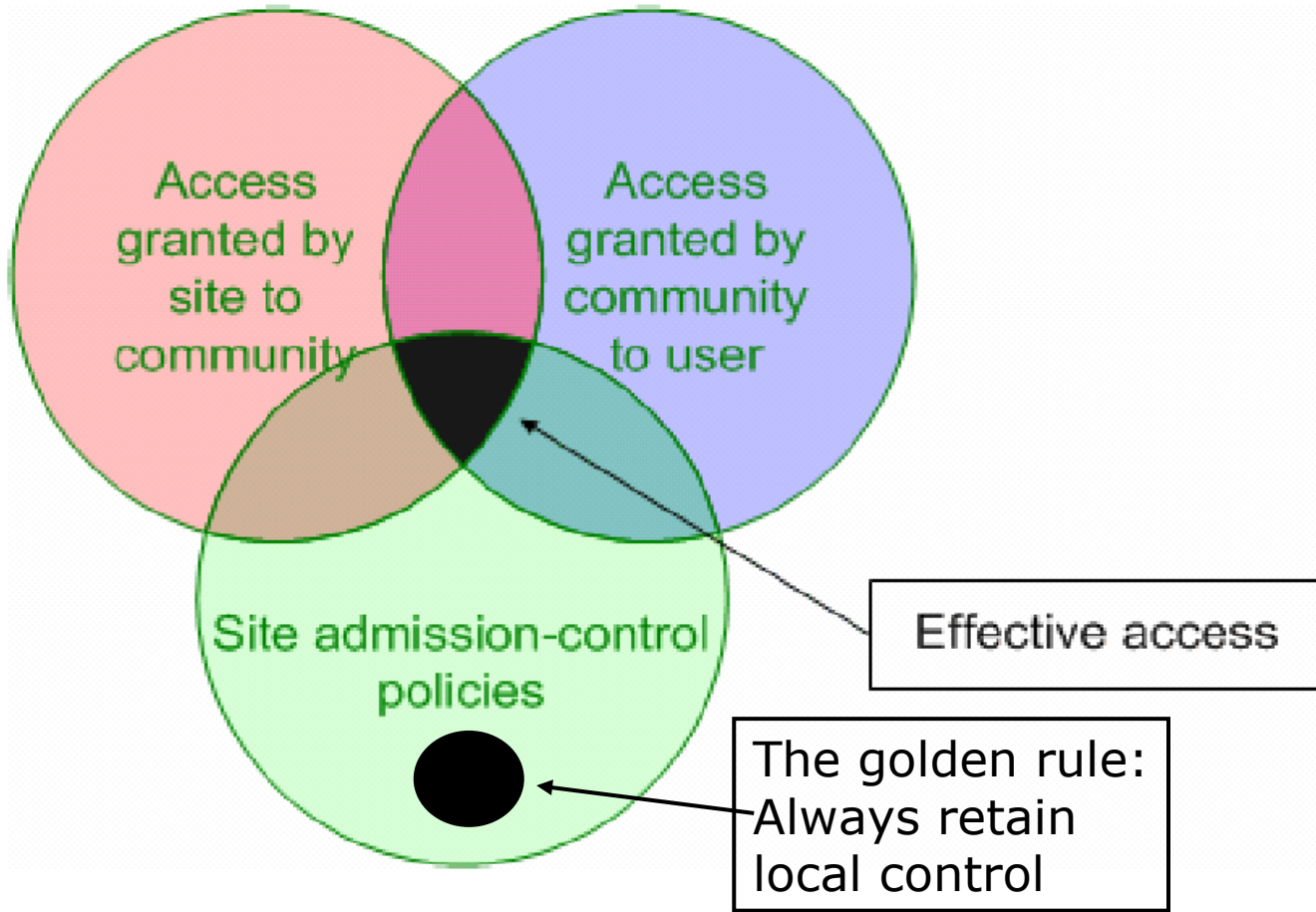


Authorization





Policies for accessing resources





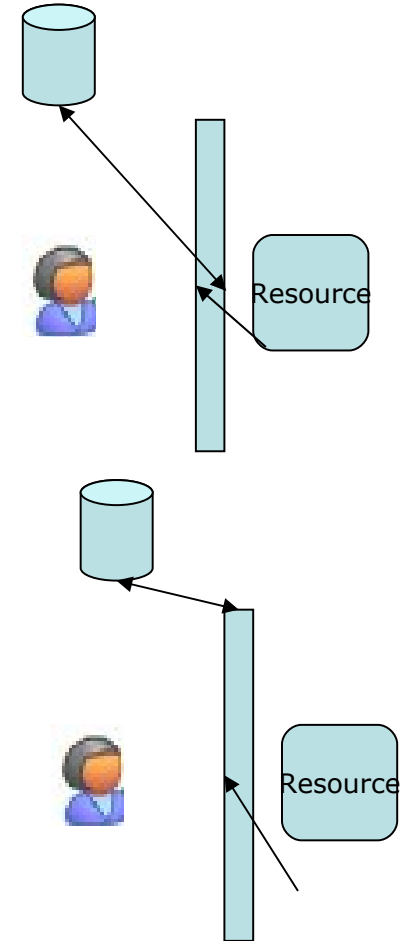
Authorization to a resource - alternatives

■ 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

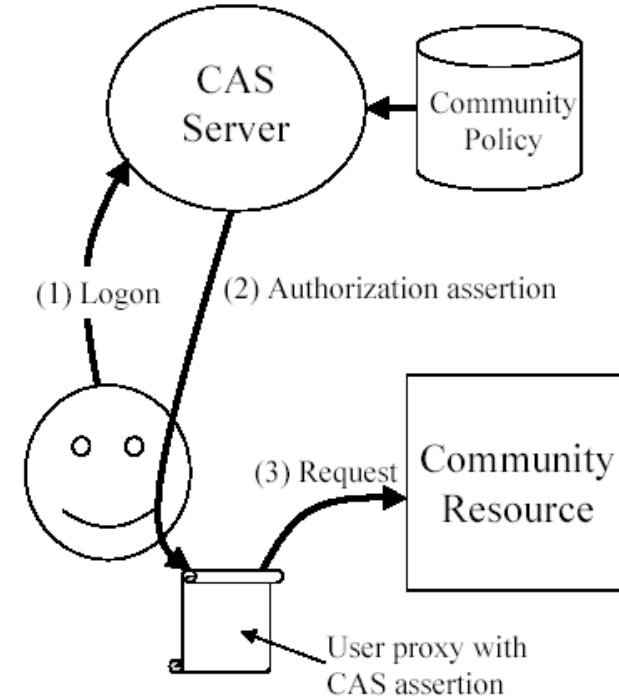
- Go to the access control point and ask for access (just showing who you are, showing your ID, nothing about what you're authorized to do).
- The access controller uses your ID to pull the access policies from a database.
- Depending on the access policies, you're authorized to run your program on the 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**

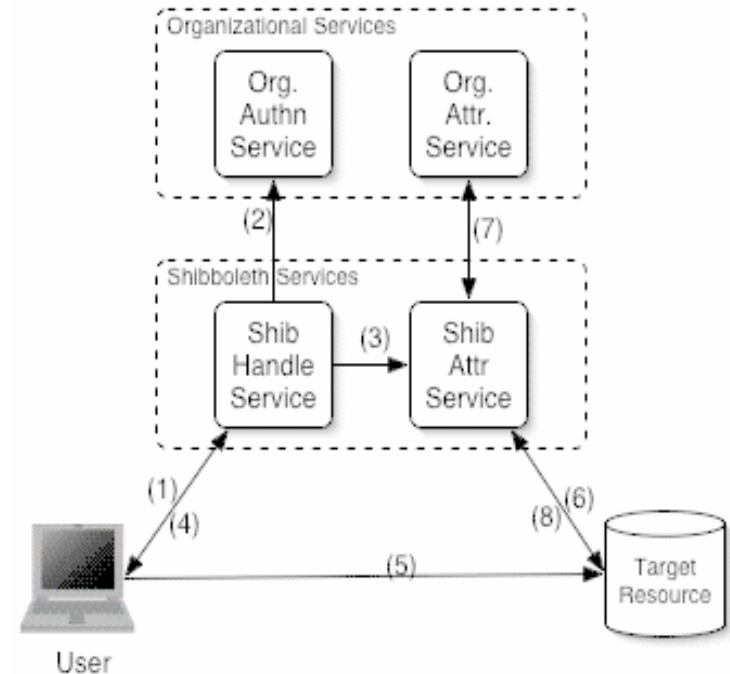




- **VOMS = Virtual Organization Membership Service**
 - Developed by the EU DataGrid and DataTag projects
- **Provides a way to delegate the authorization of users to VO managers:**
 - The user credentials are associated with a set of membership information (VO name, group, roles, generic attributes)
 - 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**



- **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



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



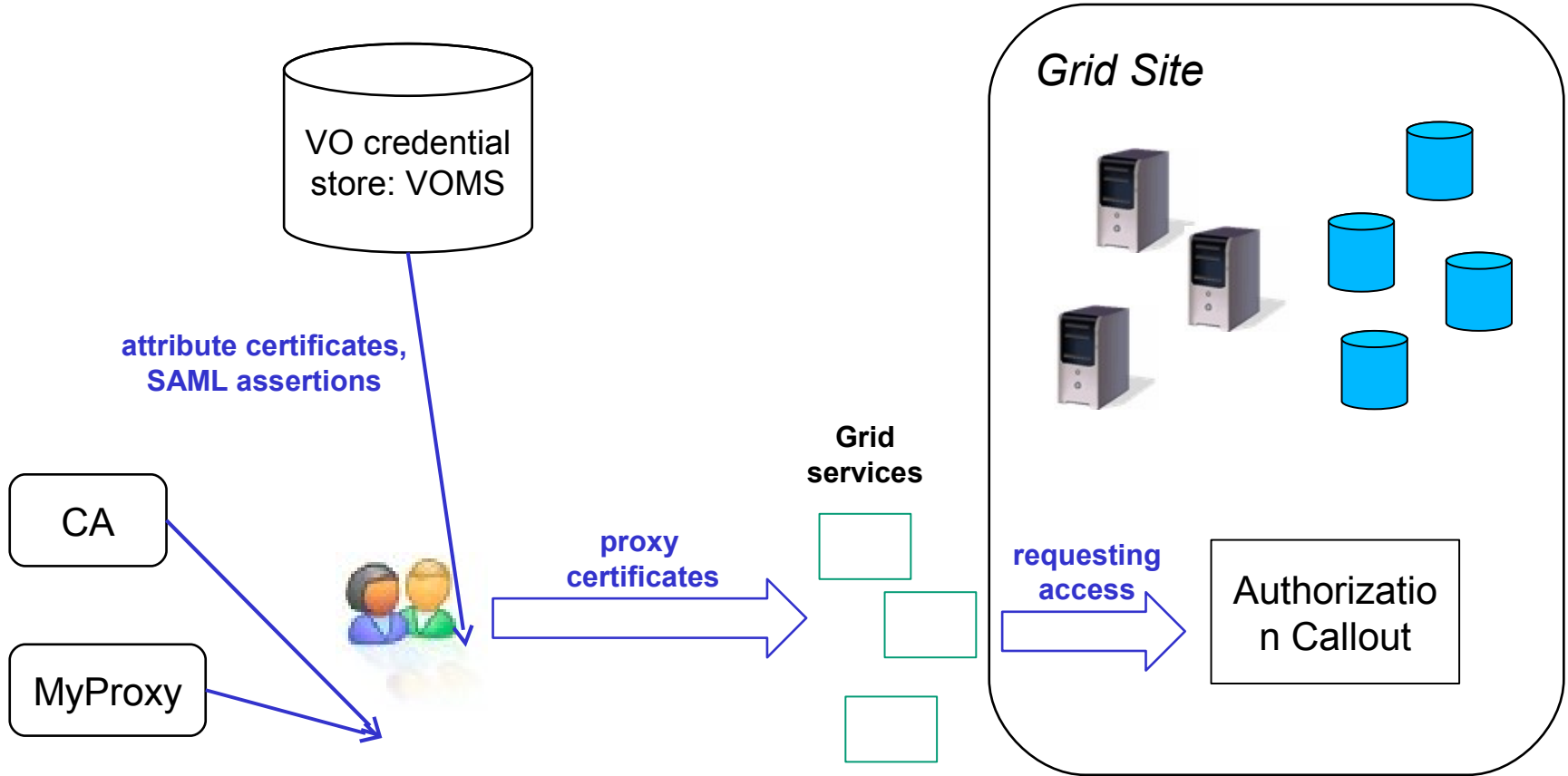
- **Implements the Grid Security Infrastructure (GSI)**
- **X.509 proxy certificates**
 - Enable single sign-on
 - The users can dynamically assign rights to services
- **MyProxy – storing and retrieving GSI credentials**
 - “convert” from username/passphrase to a GSI certificates
 - Renewing credentials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



- **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)

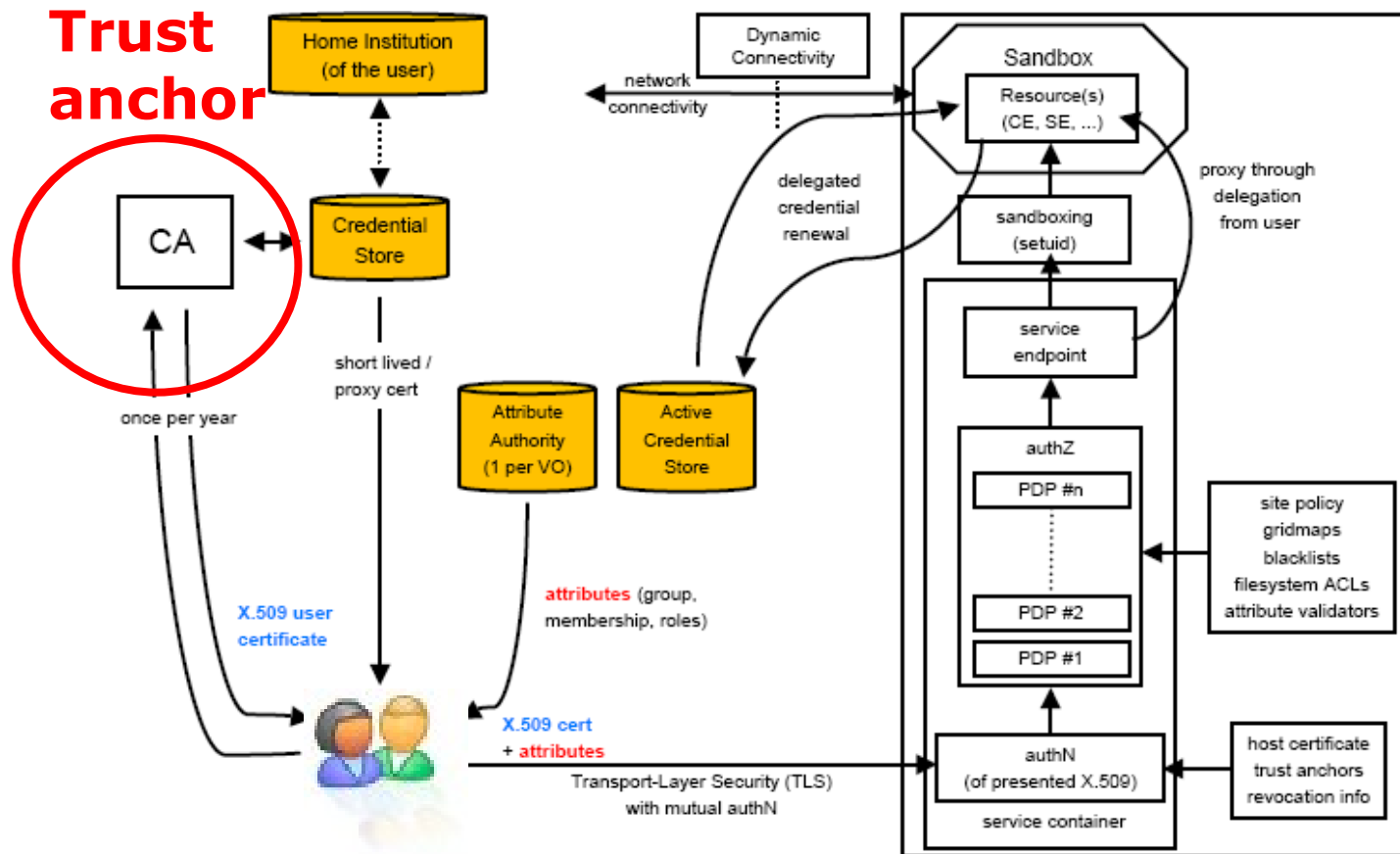


Security in gLite - Overview

- **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



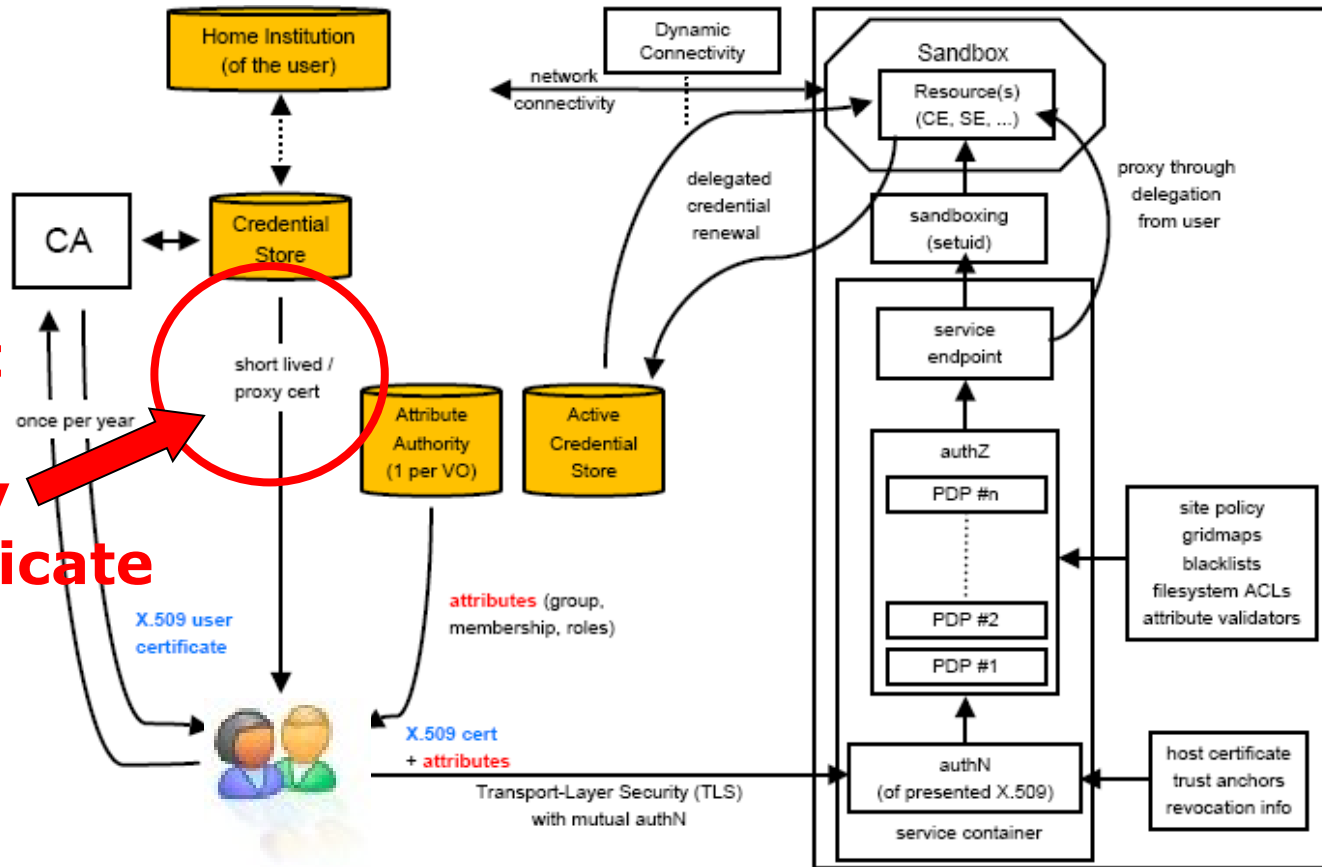
gLite - Security flow (1)





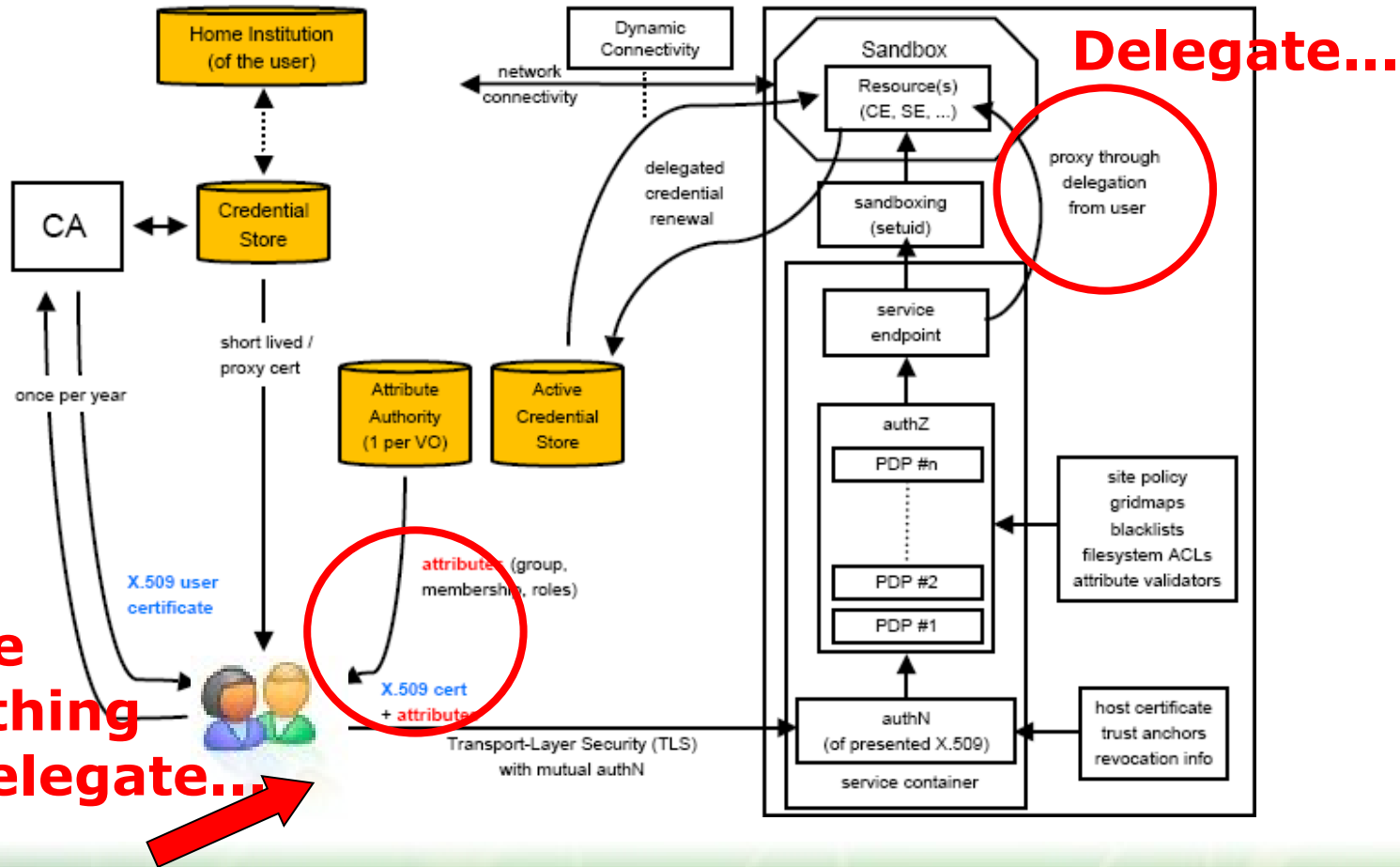
gLite - Security flow (2)

Short lived proxy certificate



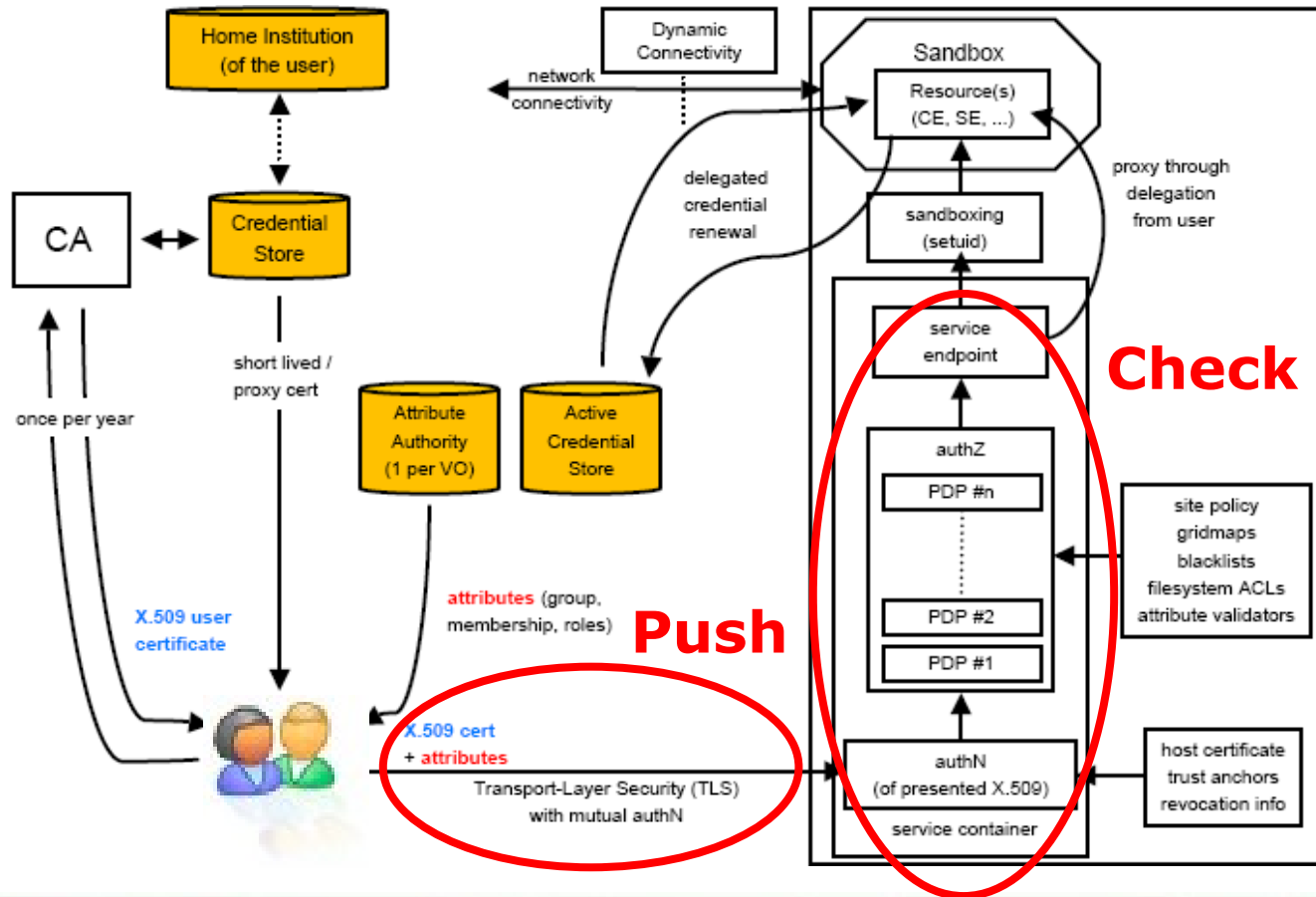


gLite - Security flow (3)



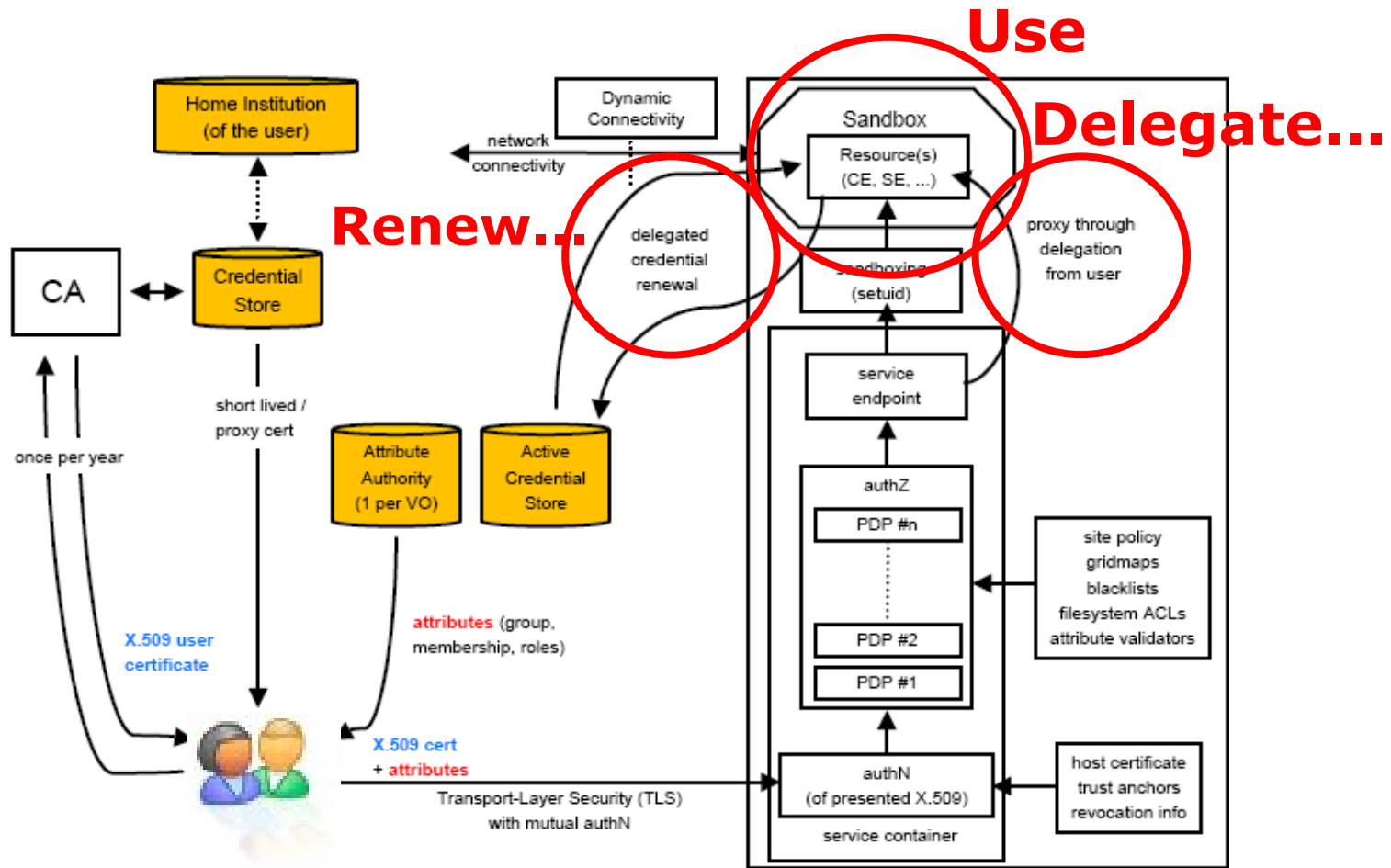


gLite - Security flow (4)





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

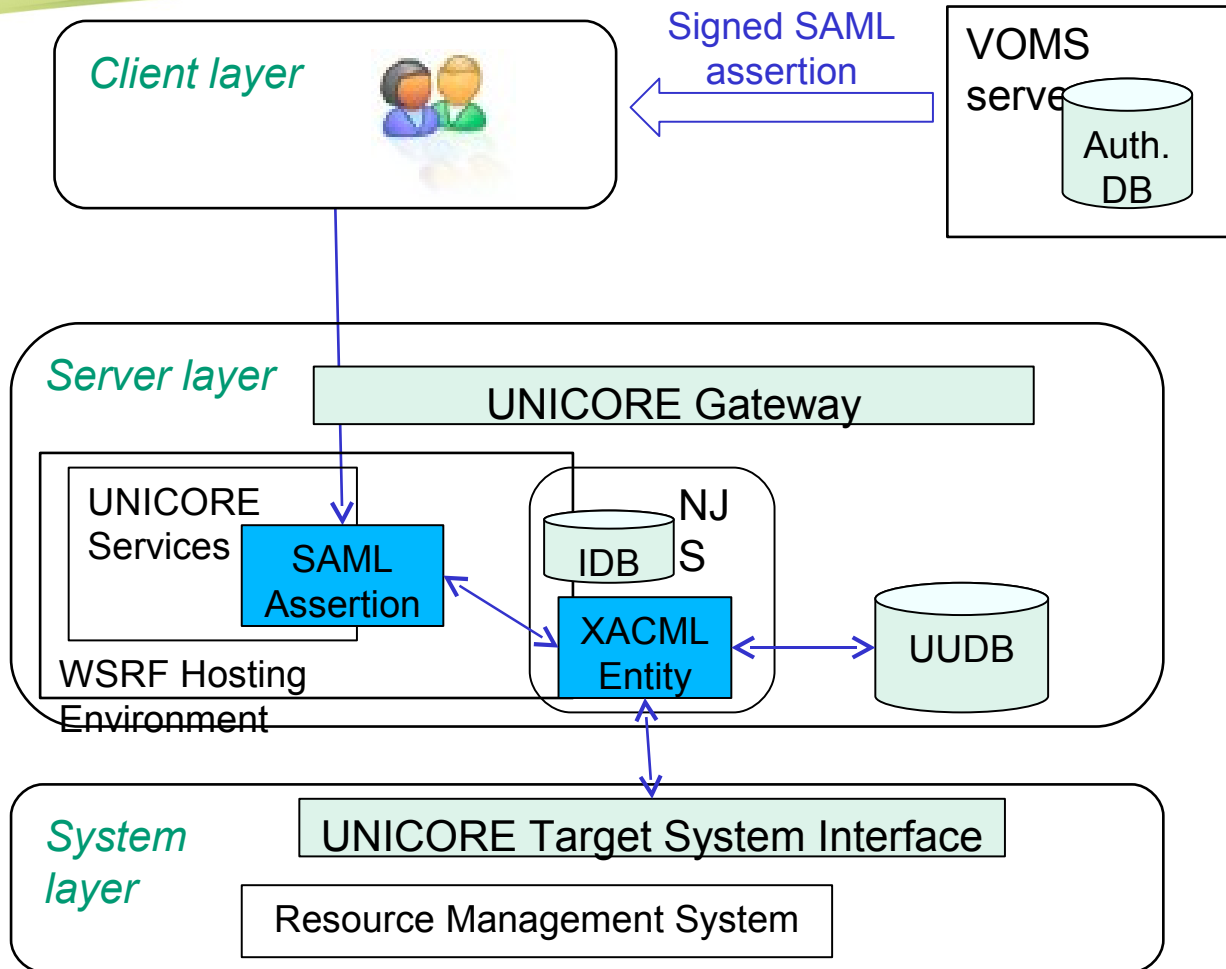
- **Mutual authentication (between Gateway/NJS and User) using X509 Certificates**
- **No proxy certificates, no generalized delegation**
- **Authorization:**
 - Performed by NJS (and thus moved away from the target system)
 - Using UADB (Unicore User Database)
 - More recent extensions to support both 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**



- **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)





- **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/put>
- 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*





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



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)



- **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 negotiation
 - 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**



- **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

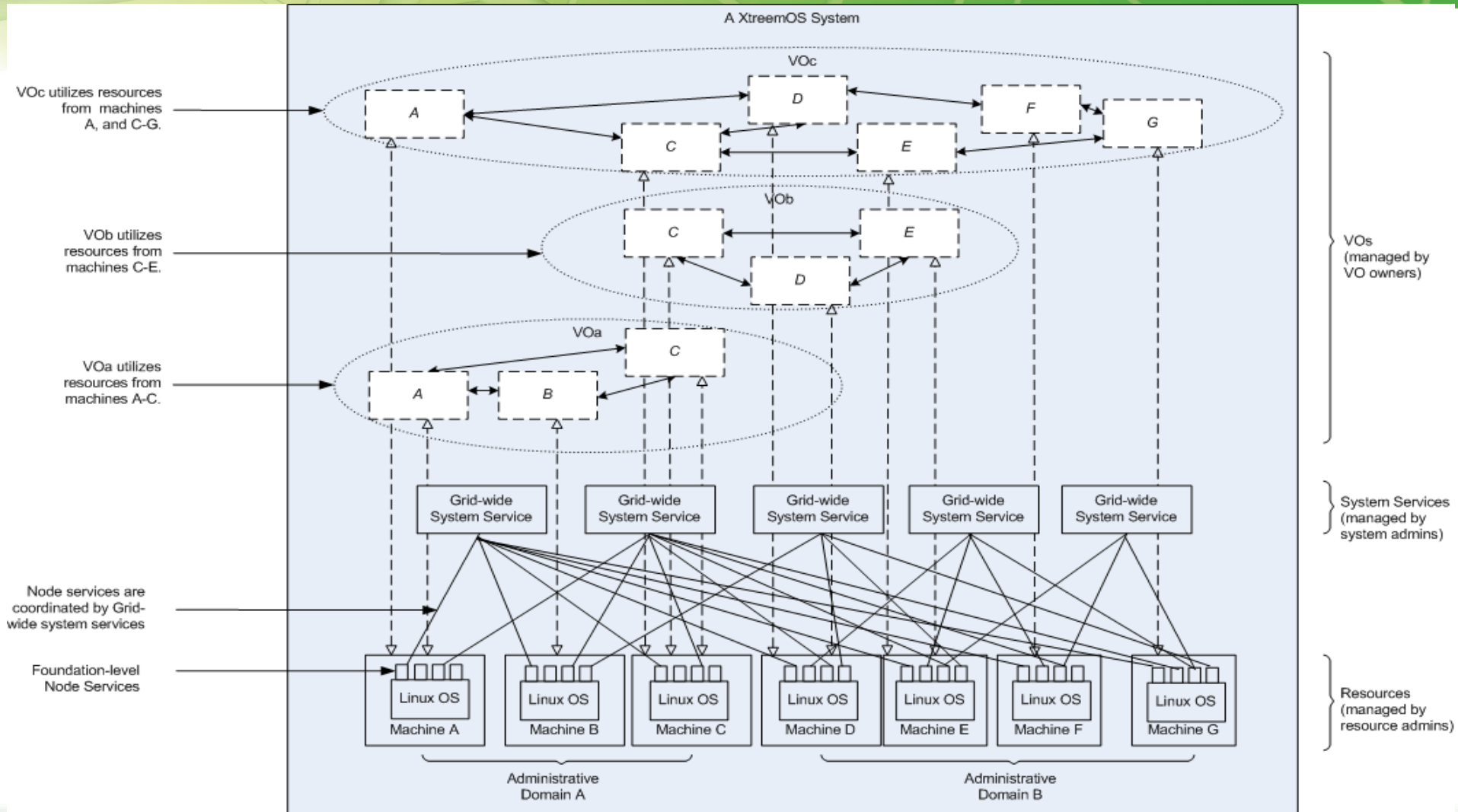


Virtual Organisation Administration

- **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 System

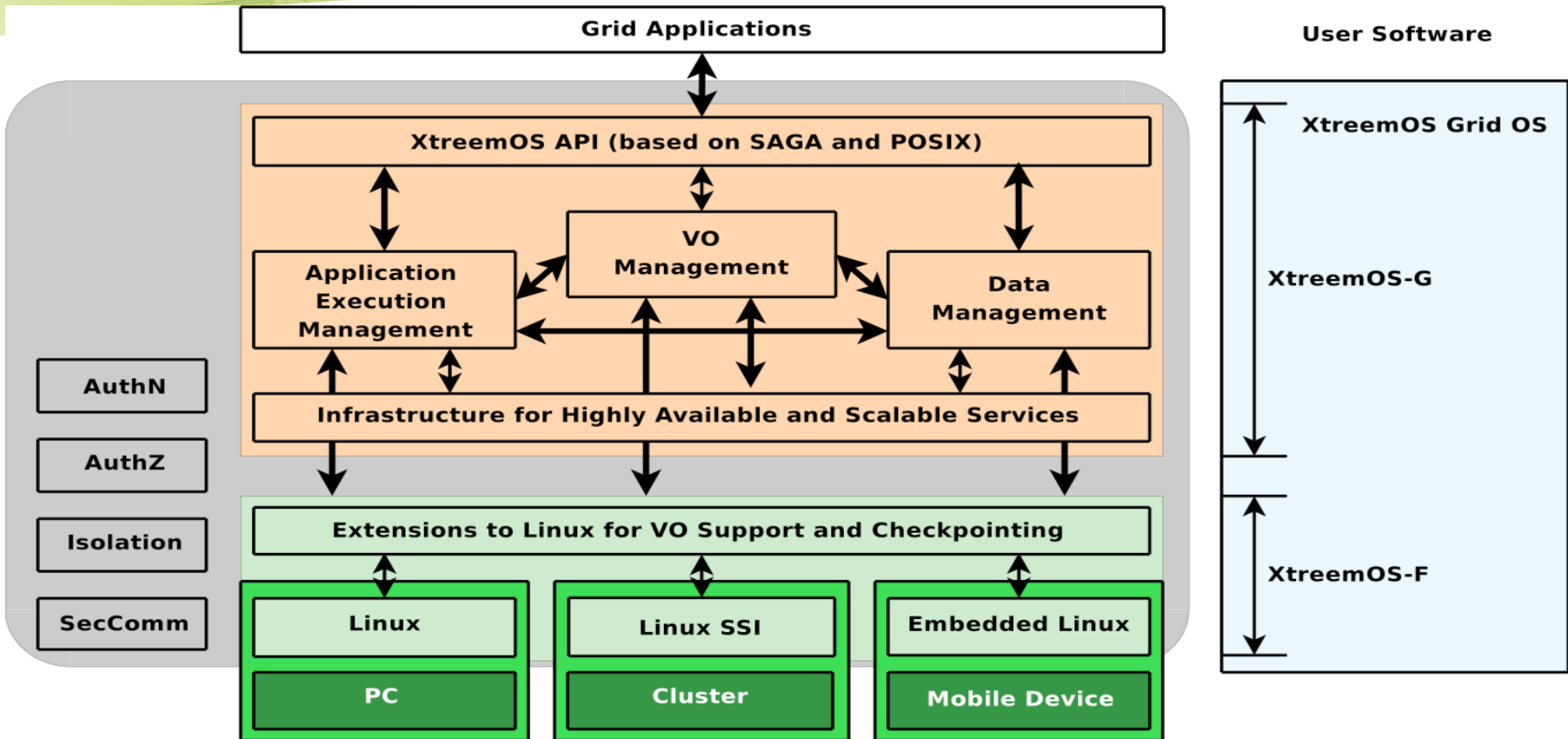




- **Distributed file system**
 - Spanning the grid
 - Replication
 - Striping
- **Access control based on Grid attributes**
- **Each XtreemOS users has one home volume in XtreemFS**



XtreemOS Architecture





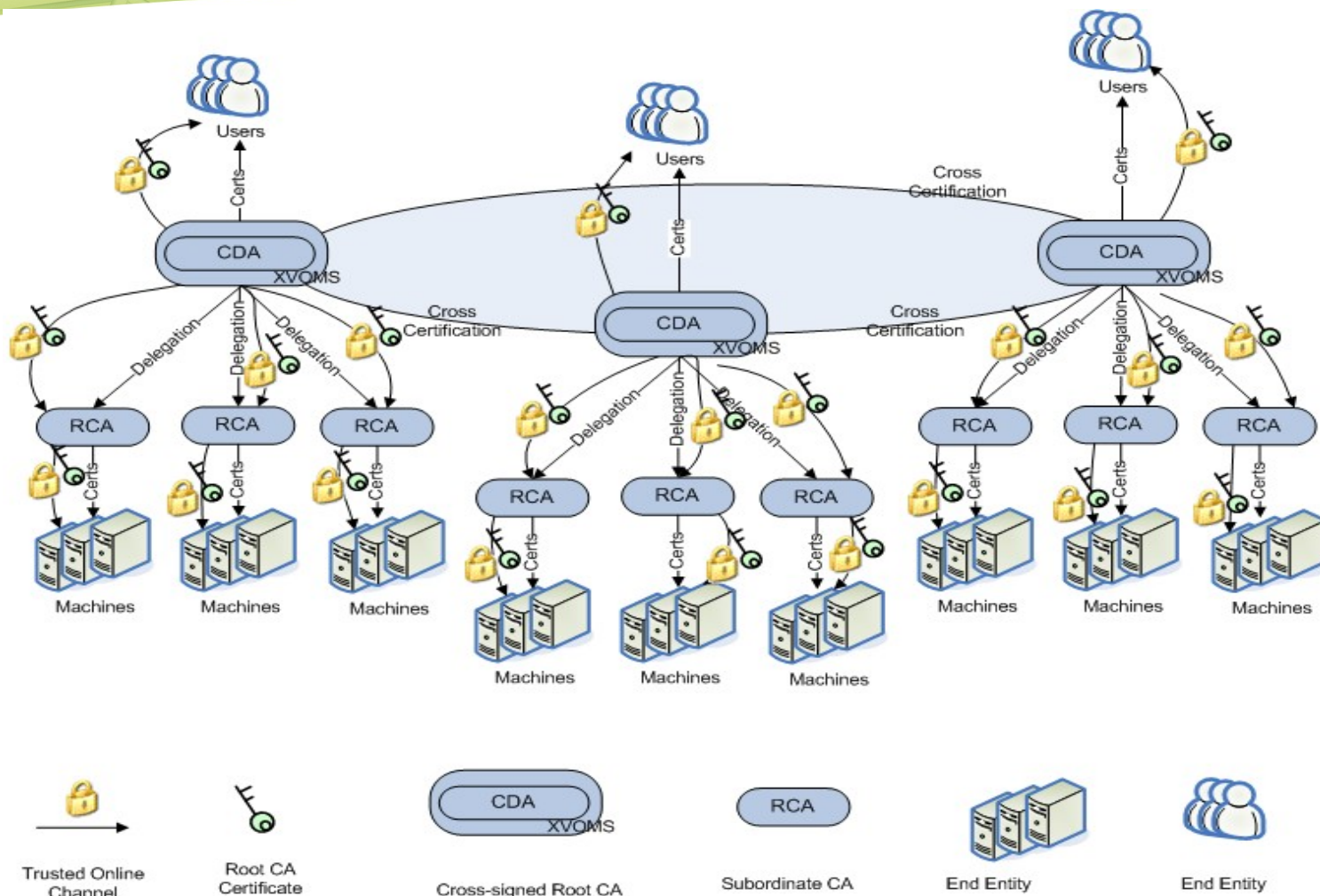
Security Model in XtreemOS



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



Trust Model





- **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)



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



Foundations



- **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



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



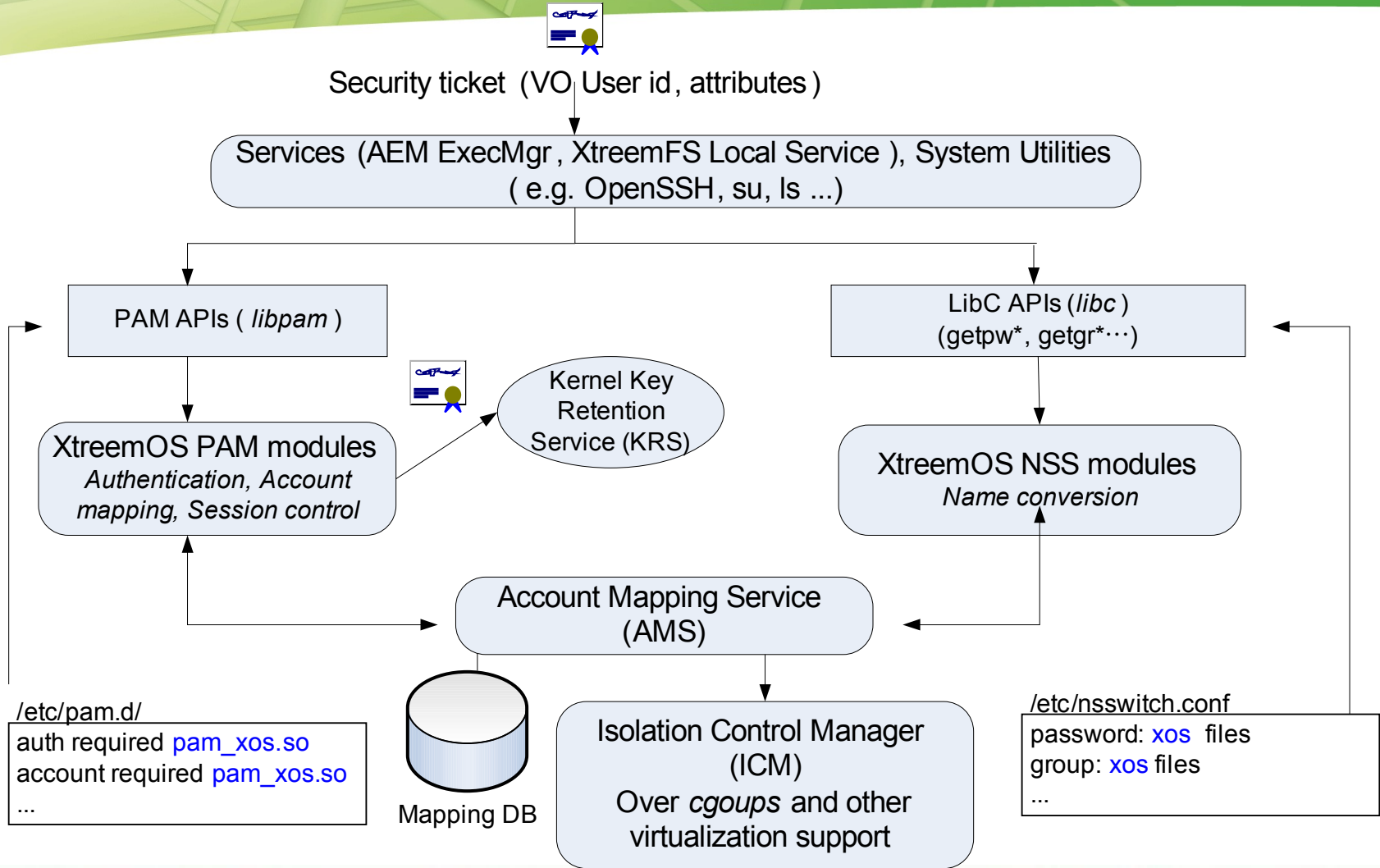
- **Job context created**
 - When a user session is opened on some resource
 - Can be
 - Simple Unix account
 - Control groups
 - limit/protect resource usage
 - Accounting, billing
 - Namespaces (PID, user, net, ...)
 - Restrict visibility from job context
 - Net namespaces restrict access to Internet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



- **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

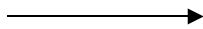


- **XVOMS**
 - User and RCA registration
 - VO lifecycle management
 - Creation/dissolution



Web frontend, VO Creation

VOLife
Frontend



The screenshot displays the XtreemOS web interface. At the top, the header reads "Virtual Organizations in Action" and "XtreemOS Enabling Linux for the Grid" with the Tux penguin logo. Below the header is a navigation bar with tabs: "Home", "Manage Users", "Manage My VOs", and "Manage My Resources". A welcome message on the right says "Welcome to VoLifeCycle , admin [logout]".

On the left side, there is a vertical menu with the following items: "Create a VO", "Join a VO", "My Pending Requests", "Get an XOS-Cert", "Generate new keypair", "About me", "Change Password", and "Logout".

The main content area shows the "Create a VO" form. It includes a text input field for "VO Name:", an "Options:" section with a checkbox for "Automatic approving of requests(disabled)", and a "VO Description:" section with a rich text editor toolbar (containing bold, italic, underline, font size, text color, background color, bulleted list, numbered list, and link icons) and a large text area. At the bottom of the form are "Create" and "Cancel" buttons.



■ **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

Virtual Organizations in Action

Select VO
and
send
joining
requests

Home Manage Users Manage My VOs Manage My Resources Welcome to V

Create a VO
Join a VO
My Pending Requests
Get an XOS-Cert
Generate new keypair
About me
Change Password
Logout

Join a VO

Search: JoinVO LeaveVO Refresh

<input type="checkbox"/>	GVID	VO Name	VO Owner	Is Member	Description
<input type="checkbox"/>	2fd9bc8f-a8a4-4195-85d0-272d1f63f093	testvo	admin	false	
<input type="checkbox"/>	4ecc77d7-c153-4a57-8430-b06df3825aa2	testvi	admin	false	
<input type="checkbox"/>	9d2dbf39-a754-4cc8-9b00-c6c83f218bd3	testes	admin	false	
<input type="checkbox"/>	7206ce2-4d38-4432-9100-1aa0a5ec8152	ette	admin	false	
<input type="checkbox"/>	f39c6568-35c1-4f50-b7b8-d8c785dba11a	test11	admin	false	
<input type="checkbox"/>	1047e048-3739-45b6-ba04-d729832e539d	test1	admin	false	
<input type="checkbox"/>	94c0658a-4d15-4f15-b9aa-9340813253ce	asdf	admin	false	
<input type="checkbox"/>	9d705a80-6fcf-4a9c-a666-af51673e9f5b	11	admin	false	
<input type="checkbox"/>	276683d2-ed17-40d6-8f19-d52d1aa969b1	ppp	admin	false	
<input type="checkbox"/>	036bdc25-d01d-46b4-a56a-99a2aededfa0	xc	admin	false	
<input type="checkbox"/>	bac5795-823c-43b3-890b-3a556fef9290	test	admin	true	



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

Virtual Organizations in Action

Home Manage Users **Manage My VOs** Manage My Resources

My Owned VOs
Approve Requests
Manage Groups/roles
Manage Policies

Managing groups/roles

- ppp
- test
- test11
- testvo
- test1

Id	Name	Realname	Affiliation	Email
----	------	----------	-------------	-------

Context menu for 'test11':

- AddGroup
- AddRole
- AddUser
- Refresh



■ **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

Virtual Organizations in Action

Home Manage Users Manage My VOs Manage My Resources

- Create a VO
- Join a VO
- My Pending Requests
- Get an XOS-Cert →
- Generate new keypair
- About me
- Change Password
- Logout

Get an XOS-Cert

Choose your joined VO:

VO Name:

Specify Cert generating parameters:

Passphrase:

Retype-Pass:

Valid days:

Submit



Manage VO Resources

Manage resources
in a VO

Virtual Organizations in Action

Home Manage Users Manage My VOs **Manage My Resources**

- Register a RCA
- Add a Resources**
- Approve Resources
- Get Machine Certificates

Managing RCA Resources

Search: AddResource DelResource Refresh

<input type="checkbox"/>	Id	Name	RCA	VOs	Desc
--------------------------	----	------	-----	-----	------

Search: AddToVO Refresh

<input type="checkbox"/>	Id	Name	Is Member	Owner	Desc
<input type="checkbox"/>	1	testvo	false	admin	
<input type="checkbox"/>	2	testvi	false	admin	
<input type="checkbox"/>	3	testes	false	admin	
<input type="checkbox"/>	4	ette	false	admin	
<input type="checkbox"/>	5	test11	false	admin	
<input type="checkbox"/>	6	test1	false	admin	



■ **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



■ 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 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



- **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**



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



- **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 ?



<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



- **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

A.2 Tutorial: “Easing Application Execution in Grids with XtremOS Operating System” at OGF28

XtreemOS



*Enabling Linux
for the Grid*

OGF 28 - Tutorial

**Easing Application Execution in Grids
with XtreemOS Operating System**

**Christine Morin, INRIA Rennes-Bretagne Atlantique
XtreemOS scientific coordinator**

March 15, 2010

XtreemOS IP project

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



Information Society
Technologies





- **Toni Cortes, BSC, Spain**
- **Yvon Jégou, INRIA, France**
- **Thilo Kielmann, VUA, The Netherlands**
- **Christine Morin, INRIA, France**
- **Michael Schöttner, University of Düsseldorf, Germany**



- **Overview of XtreemOS**
- **Application Execution Management (AEM)**
 - What XtreemOS-AEM offers that other systems do not
 - How are the main tasks done (user view)
 - AEM internals
- **Support for interactive applications**
- **Reliable job execution**
 - XtreemOS-GCP checkpointing service for distributed applications
 - Integration of different checkpointer packages
 - Generic callbacks for application-level optimizations
 - Channel checkpointing with heterogenous checkpointer packages
- **Examples of job execution**
 - MPI applications
 - Workflow
 - Executing SAGA applications on XtreemOS



- **XtreemOS: a distributed operating system for Grids**

- VO natively supported

- **Distributed operating system**

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**



■ Transparency

- Bring the Grid to standard users
 - Ease of use, management & programming
 - Provide Posix/Unix interface
 - Based on Linux operating system
 - Efficient, reliable and secure application execution
 - Legacy applications
 - Grid applications (SAGA)



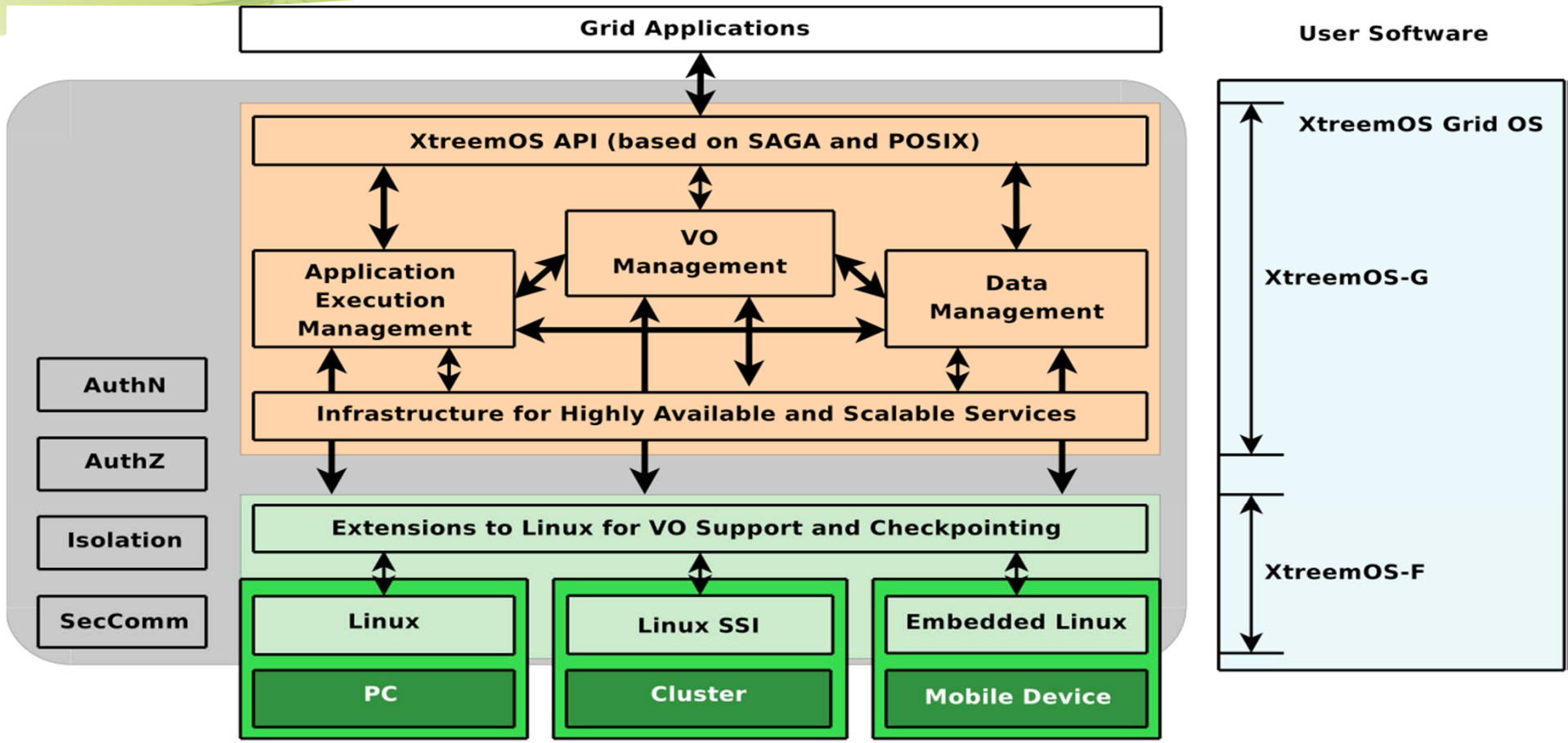
■ Scalability

- Scale with the number of entities and adapt to evolving system composition
 - Target large scale highly dynamic grids spanning multiple administrative domains
 - Large number of heterogeneous resources with high churn
 - Large number of users & applications
 - Dependable system
 - Distribution, replication, migration of XtreemOS services
 - Overlay as underlying communication system
 - Security services





XtreemOS Architecture





- **Improving usability**
 - **Local resource administrator:** autonomous management of local resources
 - **VO administrator:** flexibility management of credential and VO policies
 - **End user:** login as a Grid user into a VO; the Grid should be as much as possible invisible
 - Single sign on (SSO)
 - Posix interface as far as possible
- **Secure and reliable application execution**
 - Fine-grained control of resource usage

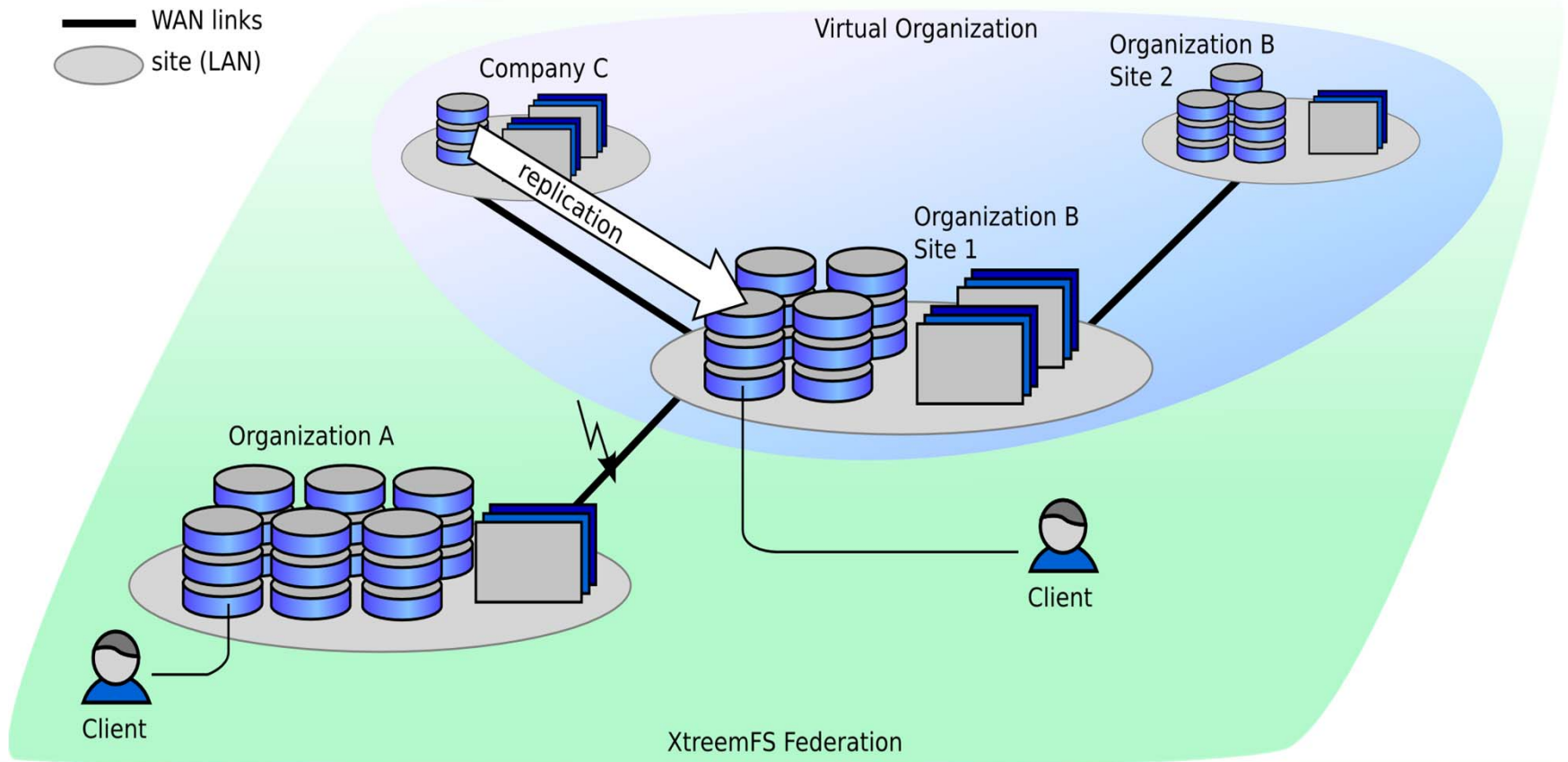


- **Scalable VO management**
 - Independent user and resource management
 - On-the fly mapping of Grid users onto local accounts
 - Interoperability with VO management frameworks and security models
 - Customizable isolation, access control and auditing



XtreemFS: A Grid File System

Federating storage in different administrative domains





- **Transparent location-independent access to data**
 - Data storage in different administrative domains
 - Grid users from multiple VO
- **Provide users a global view of their files in a Grid**
 - Posix compatible file system (API, behaviour)
 - Each XtreemOS user has a home volume in XtreemFS
- **Consistent data sharing**
 - Client-side caching & cache consistency
- **File striping & replication**
 - Autonomous data management with self-organized replication and distribution
- **Advanced metadata management**
 - Replication, partitioning
- **Access control based on Grid attributes**
 - VO member credentials



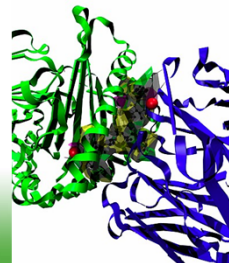
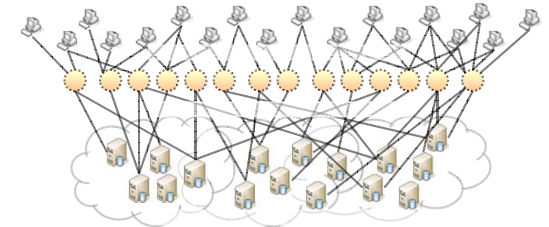
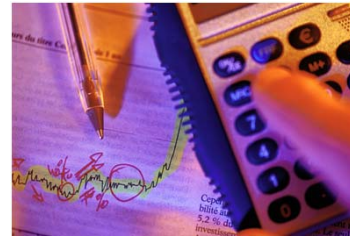
Application Spectrum

- **Wide range of applications...**
 - Grid aware distributed applications
 - Grid unaware (legacy) applications executed in a Grid

- **... in different domains**

- E-business
 - Services...
- Scientific applications

... XtreemOS is an OS!

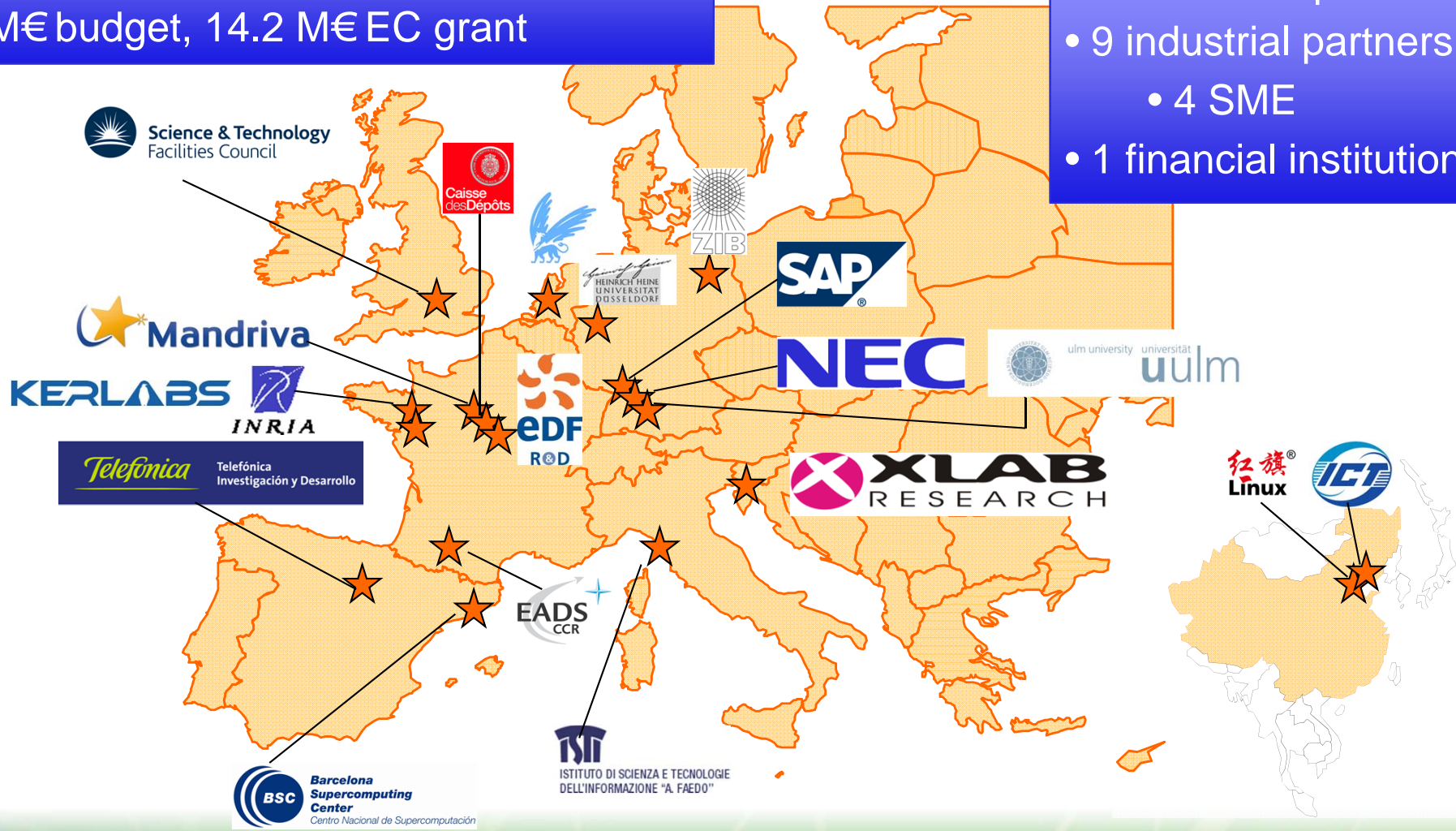




XtreamOS European Project

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

- 9 academic partners
- 9 industrial partners
 - 4 SME
- 1 financial institution



- What AEM in XtreemOS offers that other systems do not
- How can I do with XtreemOS
- AEM internals



Grid awareness

- **Users may be unaware of Grid issues**
 - Grid used like any interactive system
 - If you know Linux you know Grid
 - Application can be interactive
 - “Grid parameters” used
 - Default ones (system, vendor, ...)
 - Learned ones
- **Grid-aware users may use all potential**
 - Define “Grid parameters”
- **Current systems** are only for Grid-aware users
- **Current systems** only allow batch jobs



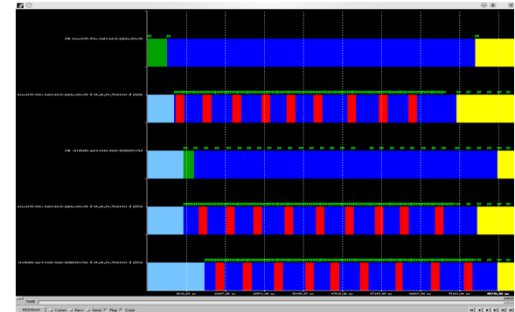
- **XtreemOS tries to reuse Unix/Linux concepts**
 - Not invent new ones
- **Parent hierarchy**
 - XtreemOS implements parent hierarchy
 - Including `jobWait` (mimicking process wait)
- **Processes are to jobs as threads to processes**
- **Job control is managed via signals to jobs**
 - Including new Grid signals
- **Current systems reinvent new mechanisms**
 - Make users life more complex

- **Jobs may not need to run in exclusive access**
 - Not all jobs require exclusive access
 - Especially interactive ones
- **XtremOS allows the user to decide whether**
 - To use exclusive node
 - To use shared nodes
 - Nodes will run more than one job at a time
- **Current systems do not allow the user to decide the environment**

- **XtremOS allows parallel applications**
 - Several resources allocated to the same application
 - Resources can be coordinated if needed
 - All managed via reservations
 - May be implicit if the user does not care about them
- **One reservation may be used by several jobs**
 - Simplify the work of workflow managers
- **Current systems, at least not all of them, offer reservations**

- **Extensible job monitoring**
 - The system monitors its own events
 - Any component can add information
 - Including the application itself
 - The user can decide what is monitored and what is not

- **Monitoring is done at thread level**



- **Current systems** have very limited monitoring
- **Current systems** only monitor at job level

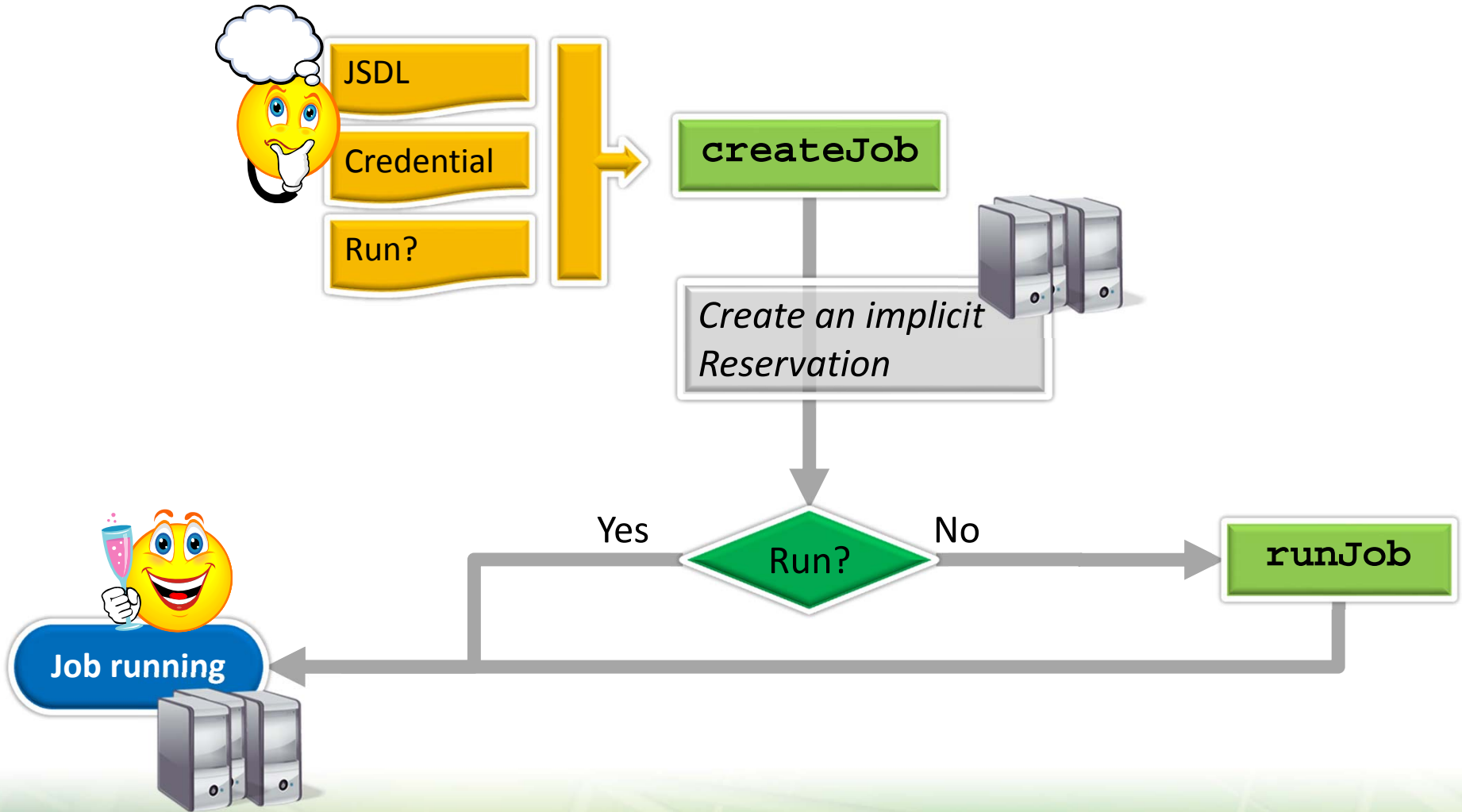
- **XtremOS allows users to define dependencies among jobs**
 - Dependency trees are tagged
 - User can have one for each need (workflow, monitoring, ...)
- **The meaning of dependencies is user-decided**
- **Implemented examples**
 - Monitor a dependency tree
 - Kill a dependency tree
 - ...
- **Current systems, at most, have predefined ones**

- **XtreemOS is aware that jobs use files**
 - When selecting the resources, file location will be taken into account
 - Nodes close to the files will be requested
 - The user needs to specify the files used
 - If cannot find resources close to files
 - Replicas will be requested to XtreamFS
- **Current systems are not file closeness aware**

- What AEM in XtreemOS offers that other systems do not
- How can I do with XtreemOS
- AEM internals



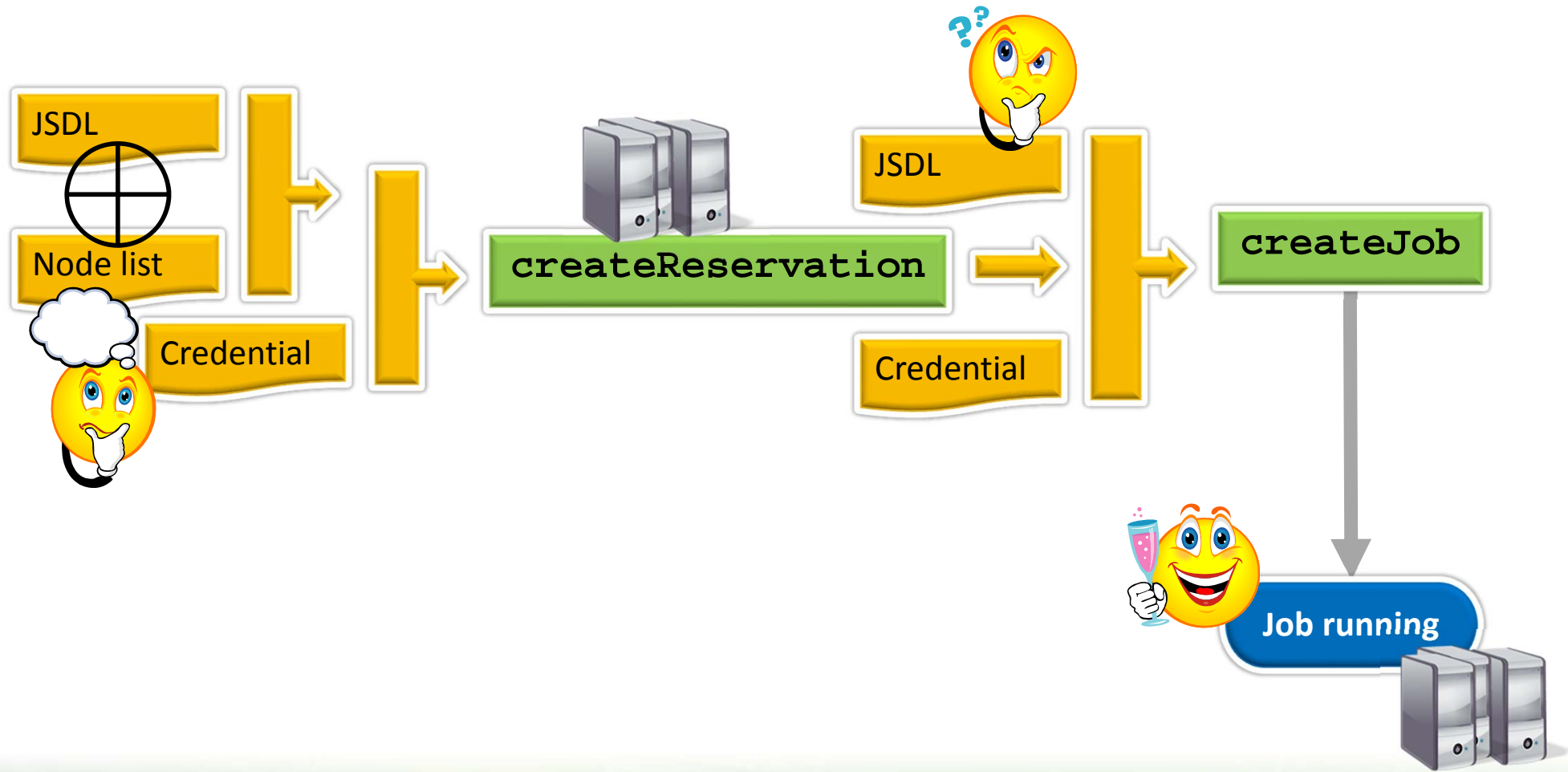
Executing a job



- **Ways to execute a job from the shell**
 - Option 1
 - `$ executable.jsdl [params] -in f -out ff`
 - If this file is empty, the system will fill it
 - This is the most Unix-like version
 - Credential will be taken automatically
 - Parameters and redirections can also be inside JSDL
 - Option 2
 - `$ xsub -f executable.jsdl`
 - Option 3
 - `$ xsub.sh executable [params] -in f -out ff`

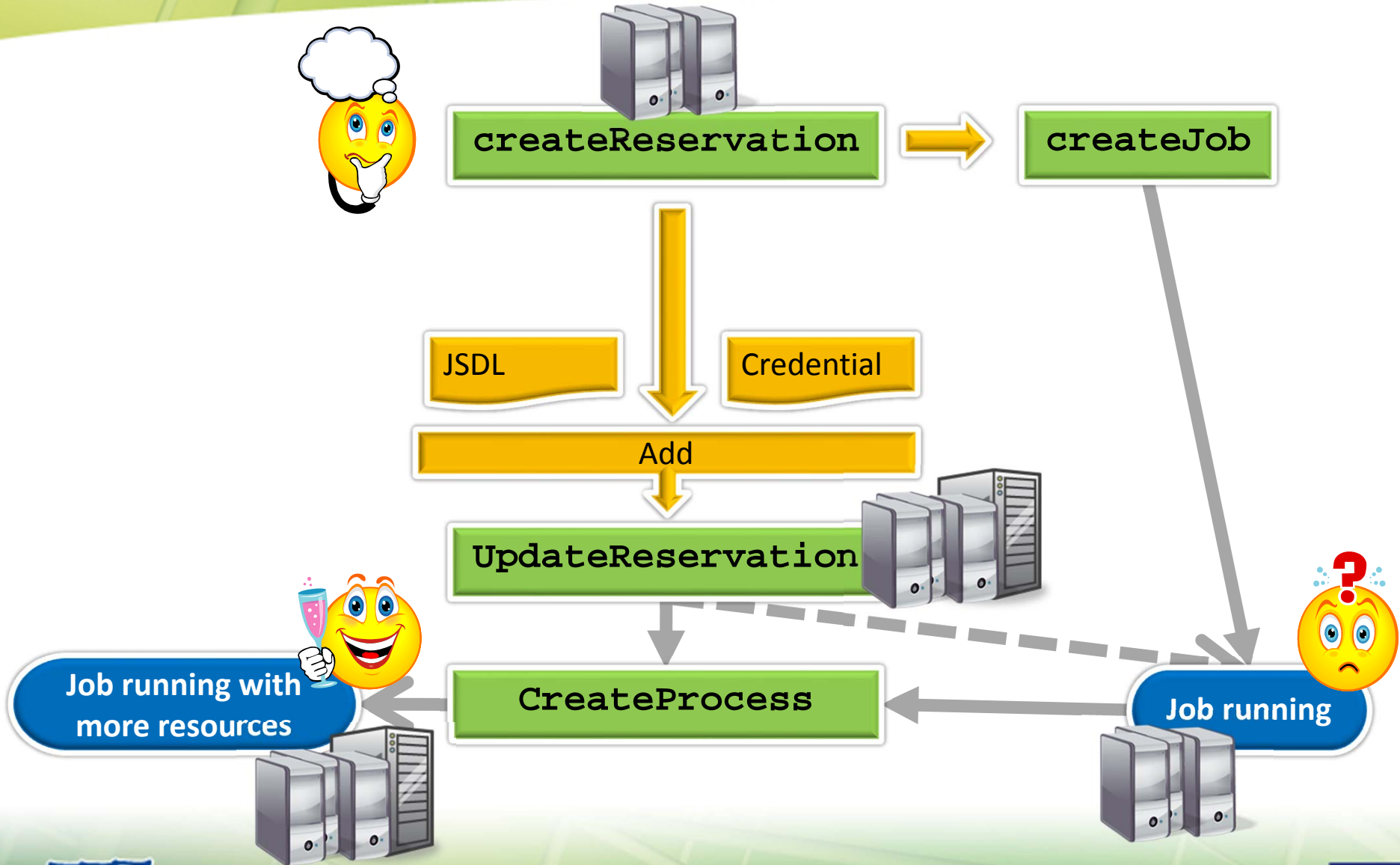


Explicit reservations



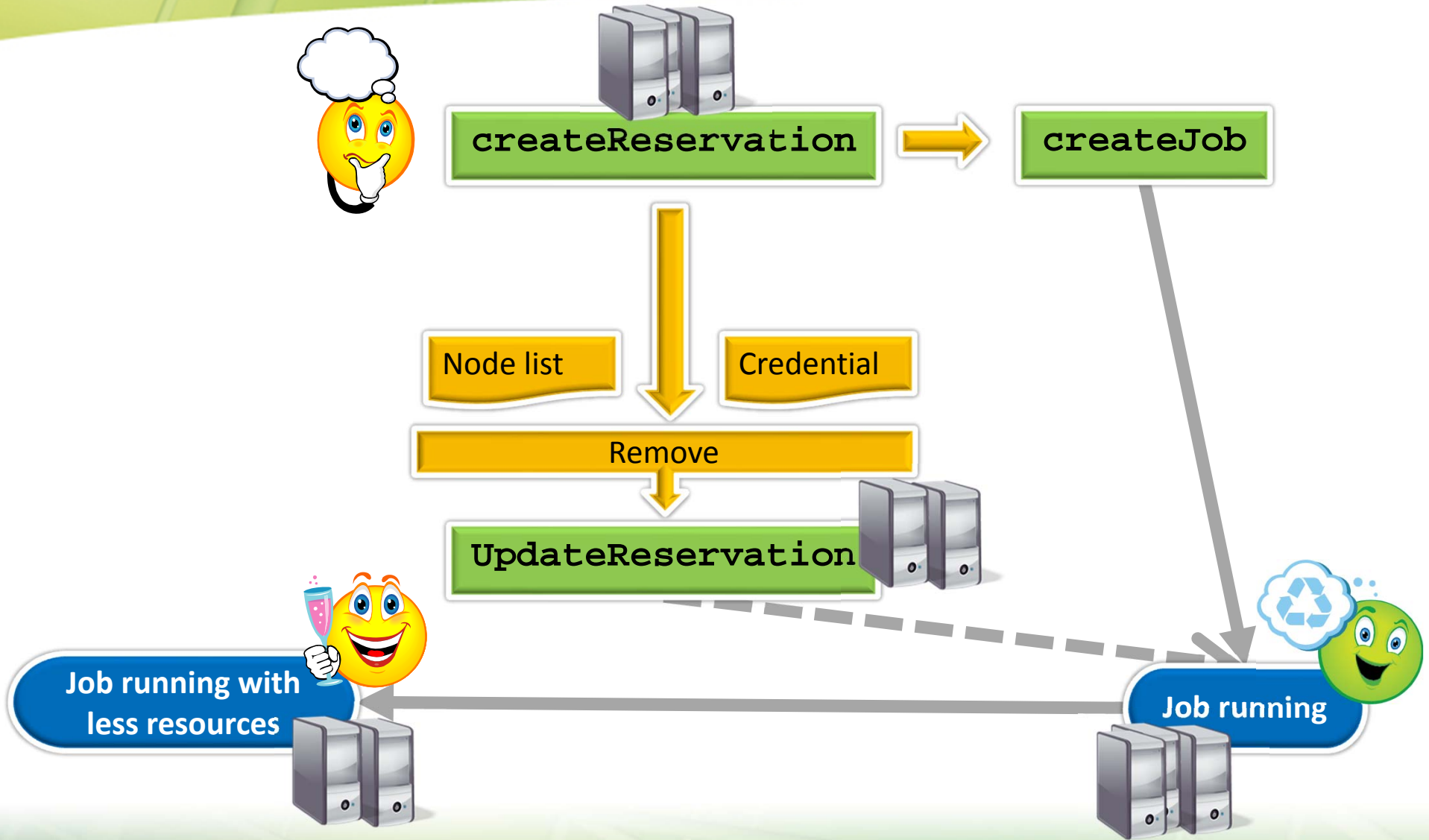


Dynamic reservations





Dynamic reservations

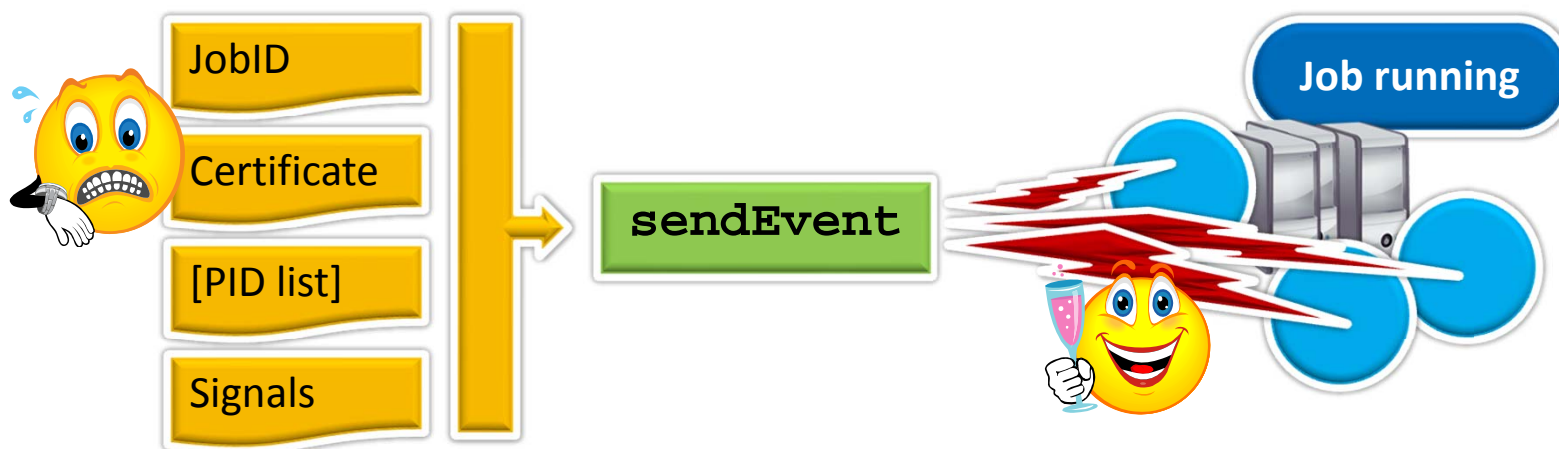


Multi-job reservation

- **A job can only use 1 reservation**
- **A reservation may hold many jobs**
 - Easy to implement workflow tools
 - Easy to implement applications with several jobs
 - User/programmer responsibility to coordinate them



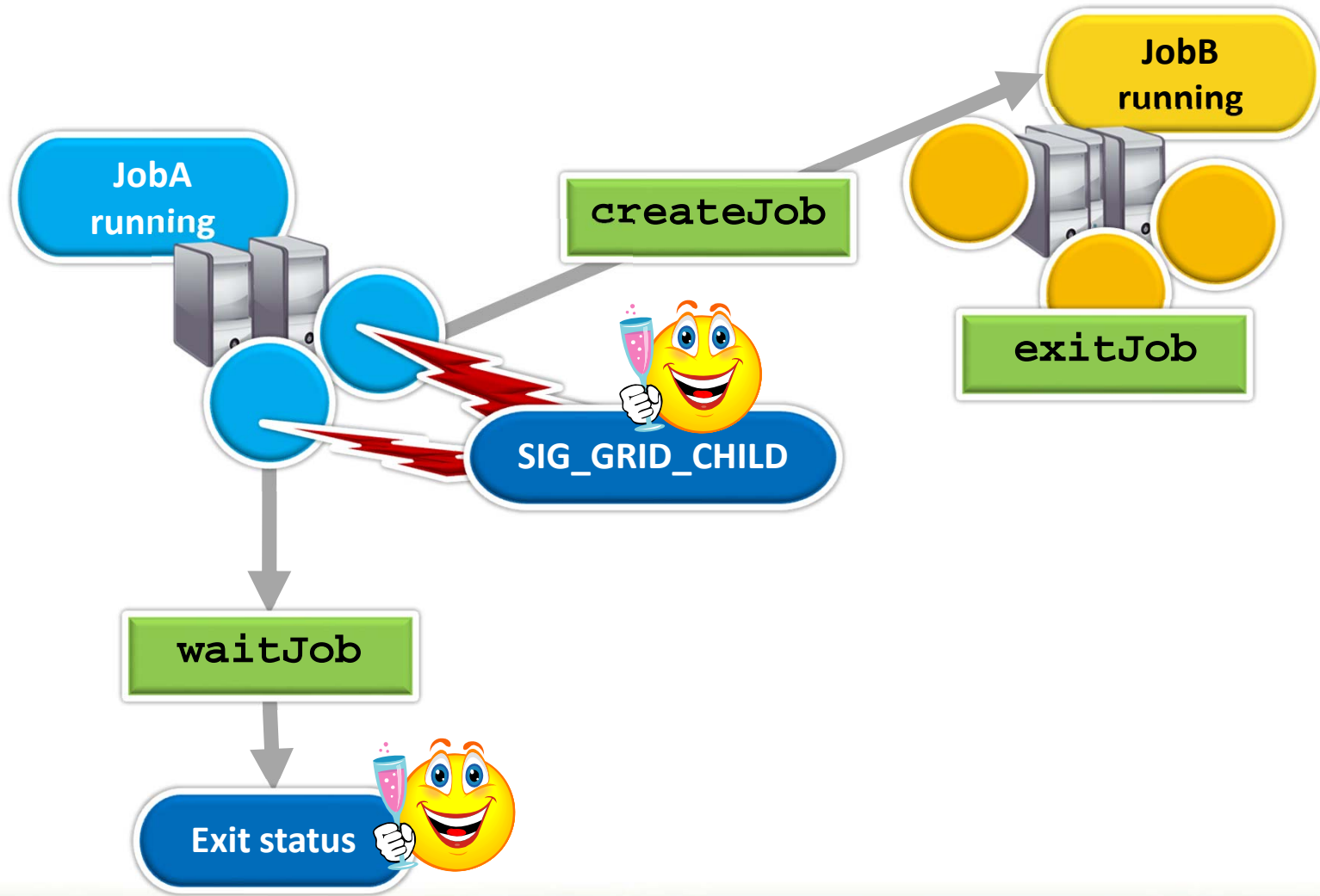
Signals to jobs



- **“Duplicated” call**
 - `jobControl (jobId, ctrOp, userCtx)`
 - Mapped to a signal event
- **Current control operations**
 - STOP → `SIG_STOP`
 - CONTINUE → `SIG_CONT`
 - KILL → `SIG_KILL` or `SIG_TERM`
 - Or any Linux process control event
- **Sending signals Linux-like**
 - Kill signal `jobID`

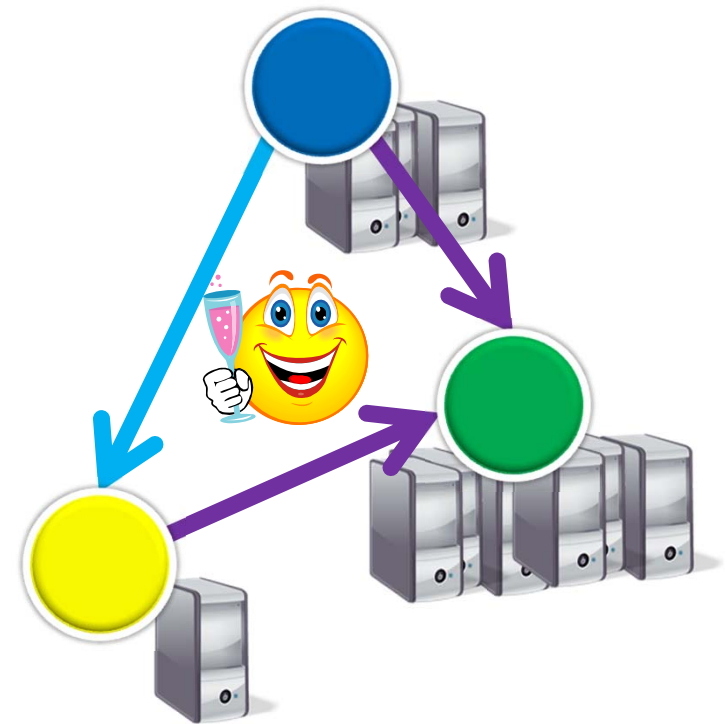
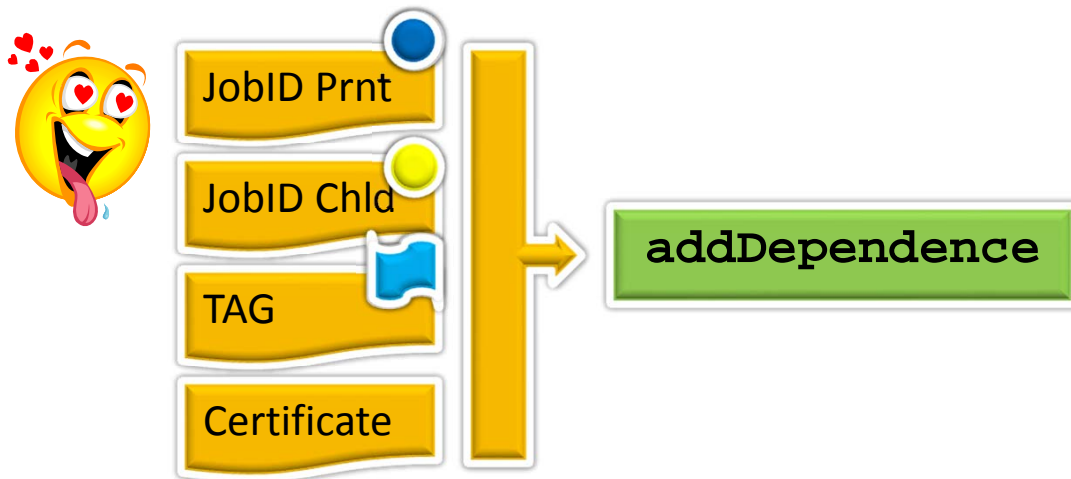


Special Grid signals



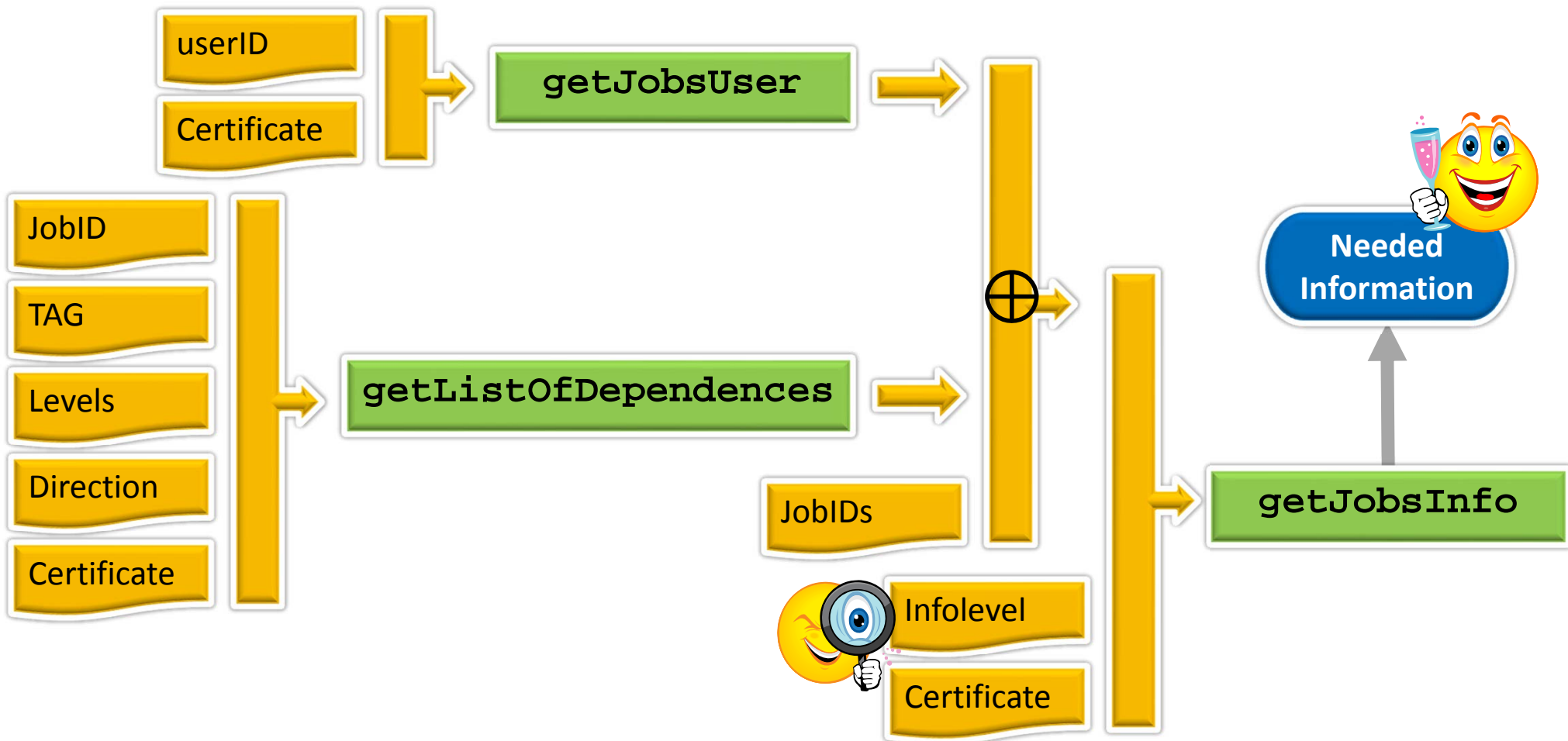


Dependences





Getting job information





Job info. Linux-like

- **Job information will appear on /proc**



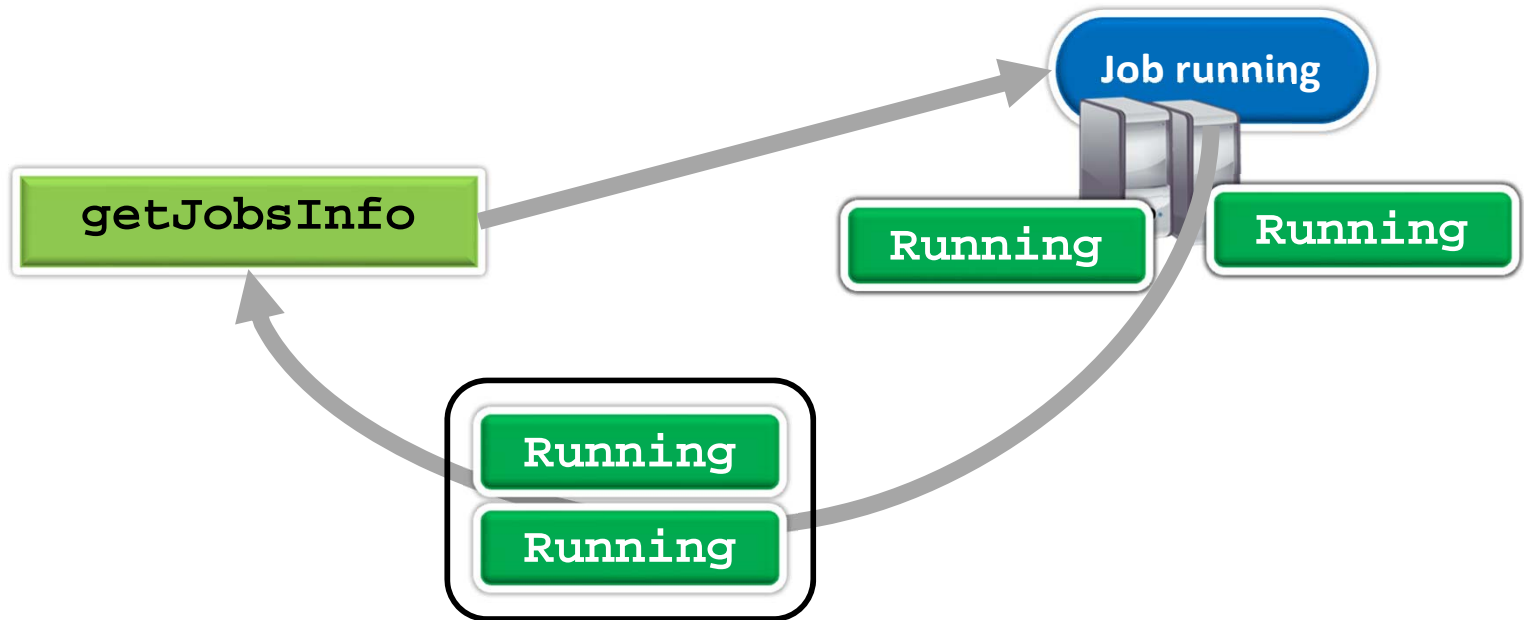
- Another way to get information instead of using special calls

- `/proc/XtreemOS/jobID/...`

- **Will be integrated in the `ps`**



Monitoring buffers



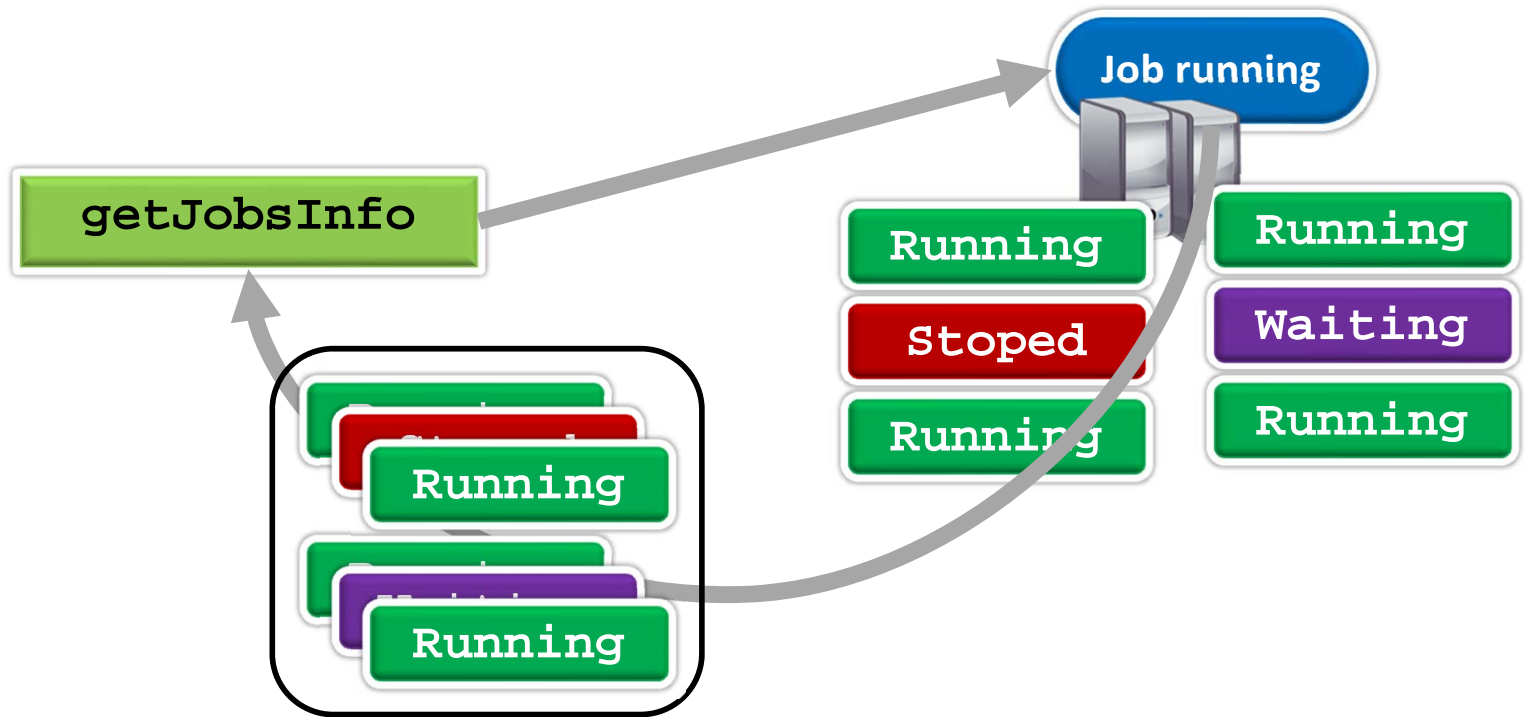


Monitoring buffers





Monitoring buffers





Monitoring user metrics



`addJobMetric`

Job running



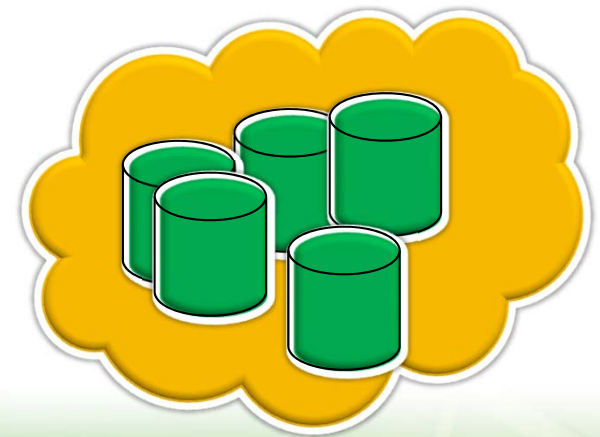
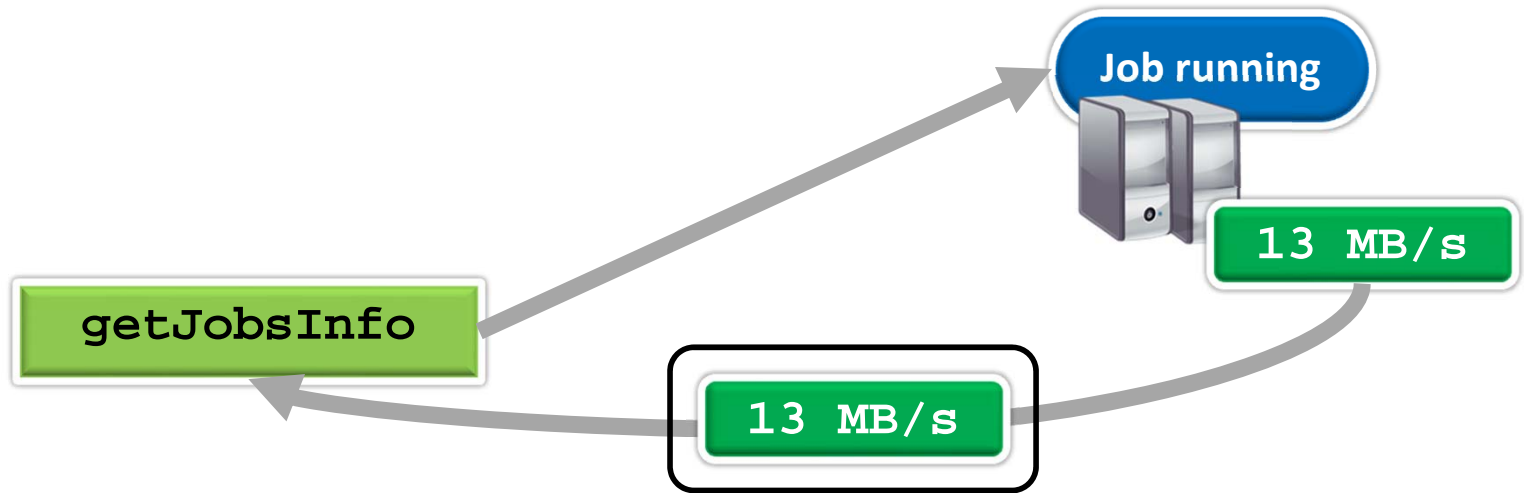
13 MB/s

`setMetricValue`



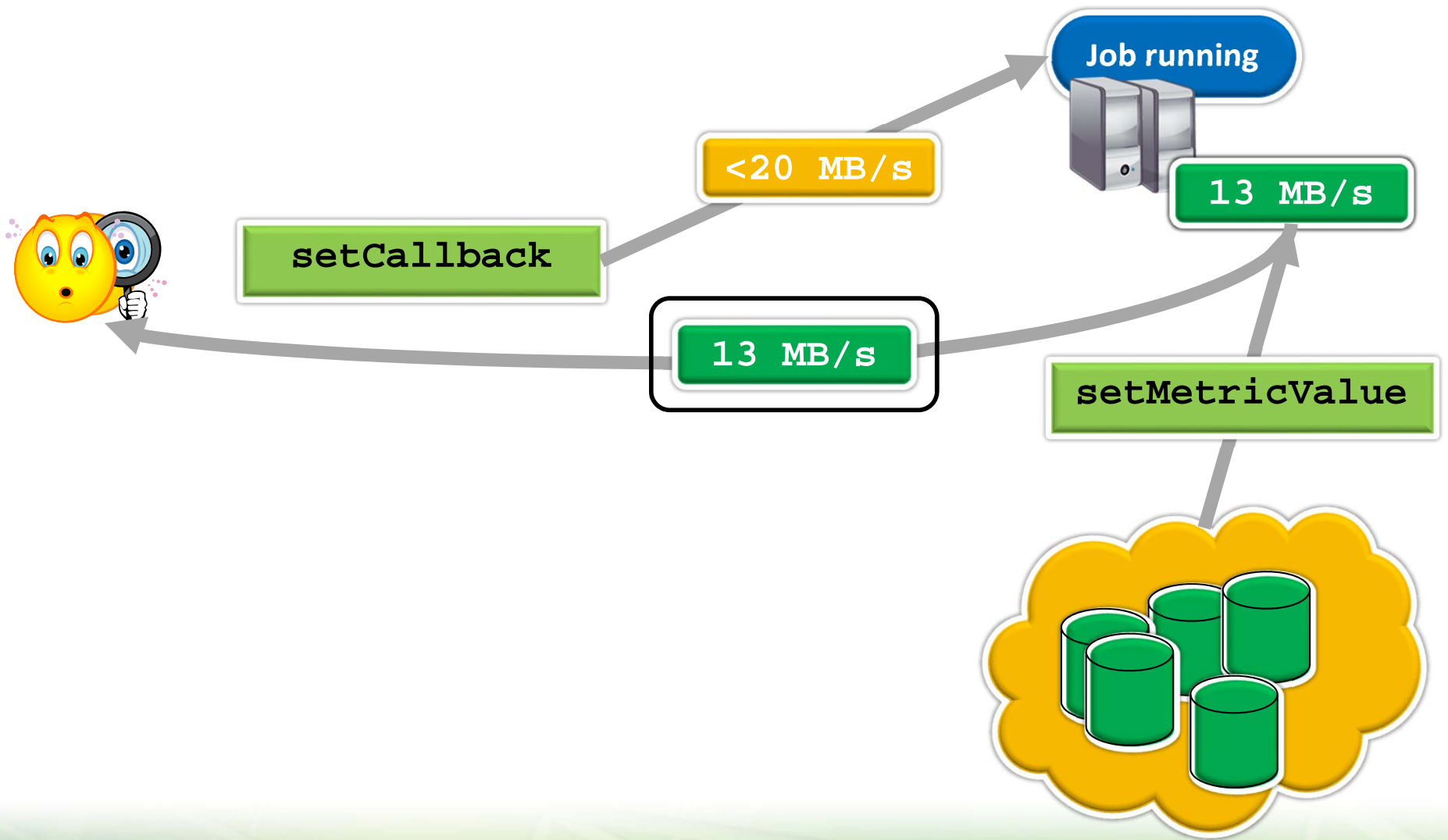


Monitoring user metrics



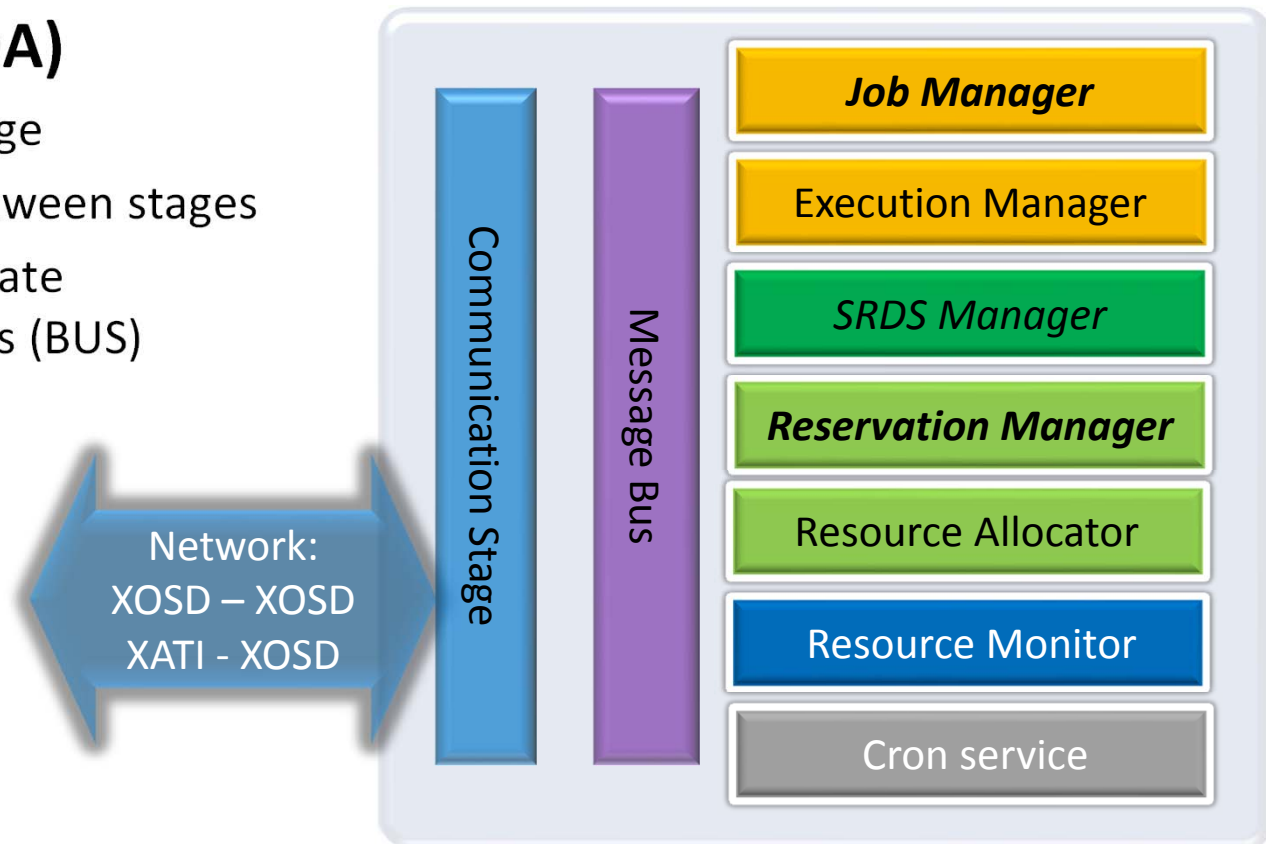


Monitoring callback



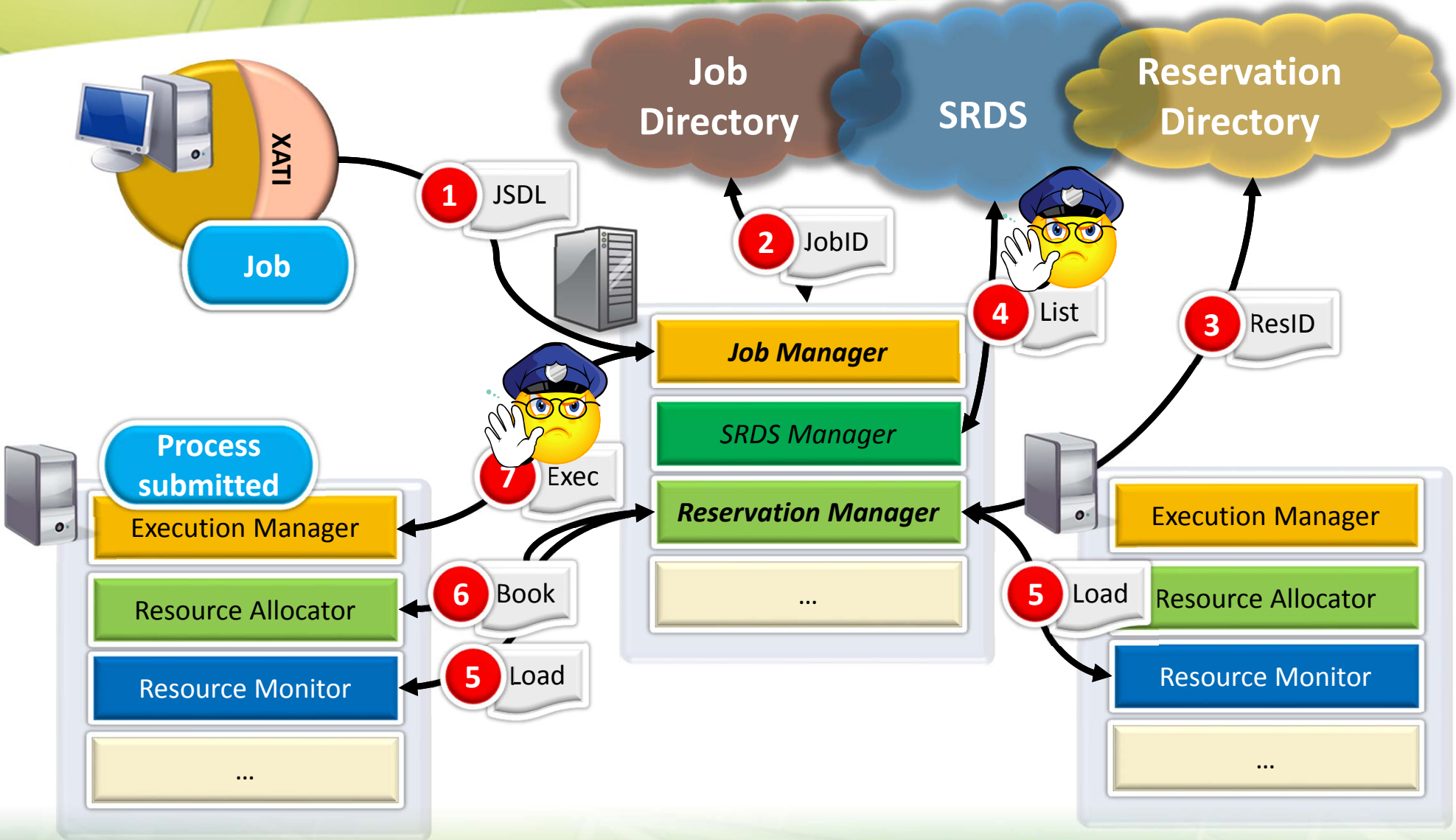
- What AEM in XtreamOS offers that other systems do not
- How can I do with XtreamOS
- **AEM internals**

- **Staged event driven architecture (SEDA)**
 - One thread per stage
 - No shared data between stages
 - Threads communicate via message queues (BUS)





Example: job submission



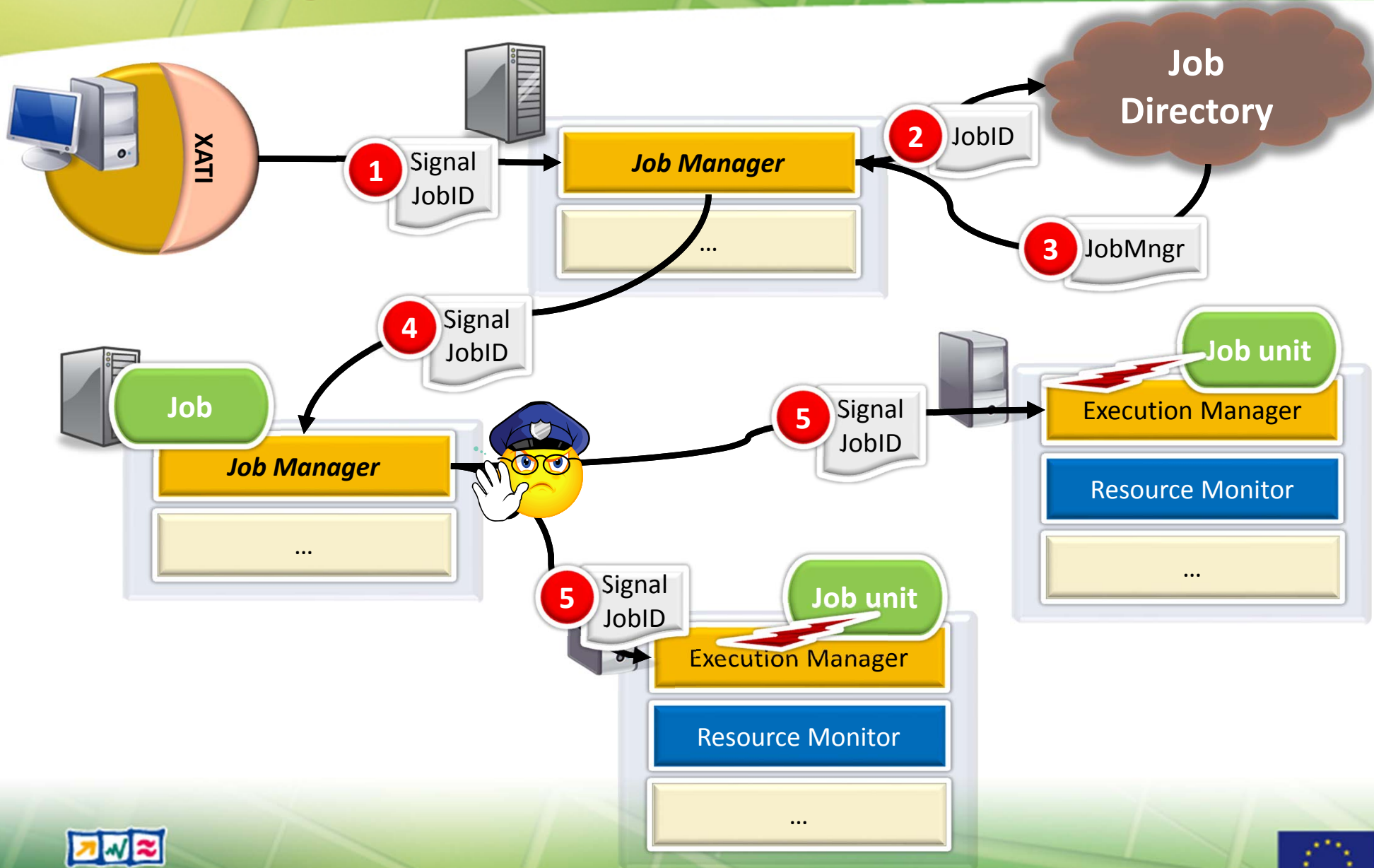
- **Nodes may have different times**
 - Different time zones
 - Skews when setting the time
- **Solutions**
 - Times are converted to GMT+0
 - Ntpd required to synchronize times
 - Some skew will always exist
 - Threshold to queue sent jobs/procs
 - If a job/proc arrives a bit early, it is queued and started later
 - If a job/proc arrives too early an error is returned

- **Algorithms (system configuration file)**
 - Random
 - Round robin
 - global on a per reservation basis
 - Several jobs may share a reservation
 - Load obtained during the negotiation phase
 - File closeness
- **Scheduling hints (user defined)**
 - Shared/exclusive
 - 1 process per node
 - Do not repeat node till necessary

- **Cooperate to reduce**
 - File transfer and remote access
- **Background**
 - File system maps nodes in a 2D space (Vivaldi)
 - Exports the coordinates to SRDS
- **Two step cooperation**
 - Scheduler will request nodes “close to files” to SRDS
 - X,Y coordinates and a radius
 - Scheduler will inform of files to be used
 - File system will try to create replicas in advance (if possible)

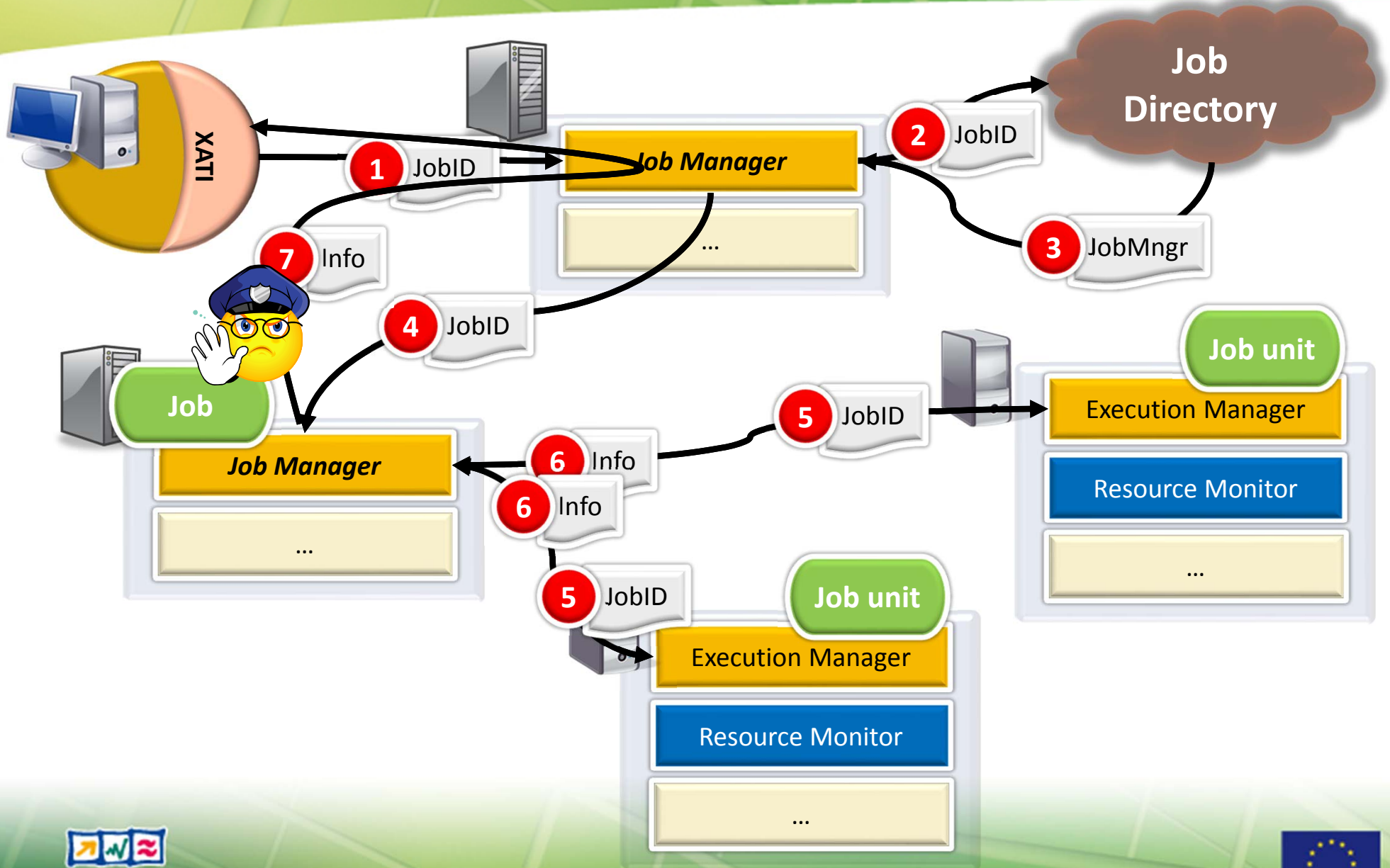


Example: Job signal





Example: job information





Buffering

- **All monitoring events can be buffered**

- Reduces monitoring traffic and overhead
- Buffers have a finite size
 - Configurable per event: small, medium, large
- If too many events, old events are lost
- When events are read, the buffer is emptied

- **To reduce overwriting unread information**

- Call back when the buffer is half full
 - Will be available as soon as call backs are available



- **LTTng**
 - Linux Trace Toolkit new generation
 - Monitors kernel events
 - Also implements buffering
- **Best option for a detailed kernel information**
 - No kernel modifications needed in XtreemOS packages
- **Example**
 - Monitor thread/process status changes
 - Without LTTng it means kernel changes

Control of new processes

- **Control non XtreemOS events**
 - Forks done by a process do not go through XtreemOS
 - But... have to be known
- **Linux informs of these events via connectors**
 - Execution manager learns about them
 - Execution manager informs job manager
 - If necessary

- **Services have a job/resource view**
 - Exceptions: Job Directory
 - Implemented using DHT → scalable
 - Some times a few hops are needed
 - The performance price is reasonable
- **No global scheduler**
 - Schedule a job in a “good enough” way
 - Not make the best potential system schedule
 - It would be impossible → do not try



- **Fault tolerance**

- Services keep no vital state
- Exceptions:
 - Job and reservation Managers
 - Job Directory

- **Job and reservation managers**

- Built on top of virtual nodes
- Master/slave replication

- **Job directory**

- Uses DHT replication mechanisms



- **Application execution management ...**
 - Learns from Linux, not invent new things
 - Offers dynamic reservations & resource management
 - Allows complete and fine-grained monitoring
 - Is aware of data
- **Application execution management is ...**
 - Transparent
 - Scalable
 - Fault tolerant



Interactive Jobs in XtreemOS

Yvon Jégou

INRIA – Rennes, France

- Usual Grid middleware
 - Prepare job data
 - Prepare job JSDL
 - Submit the job
 - Wait for job termination
 - Exploit results
- No support for interaction between users and jobs
 - Jobs are run in batch mode



Why Support Interactive Jobs?

- Many computing environments are interactive
 - MATLAB, Scilab
 - Meshing tools
 - Visualisation
- Full integration of Grid to user desktop environment
 - `job1 < infile | job2 > outfile`

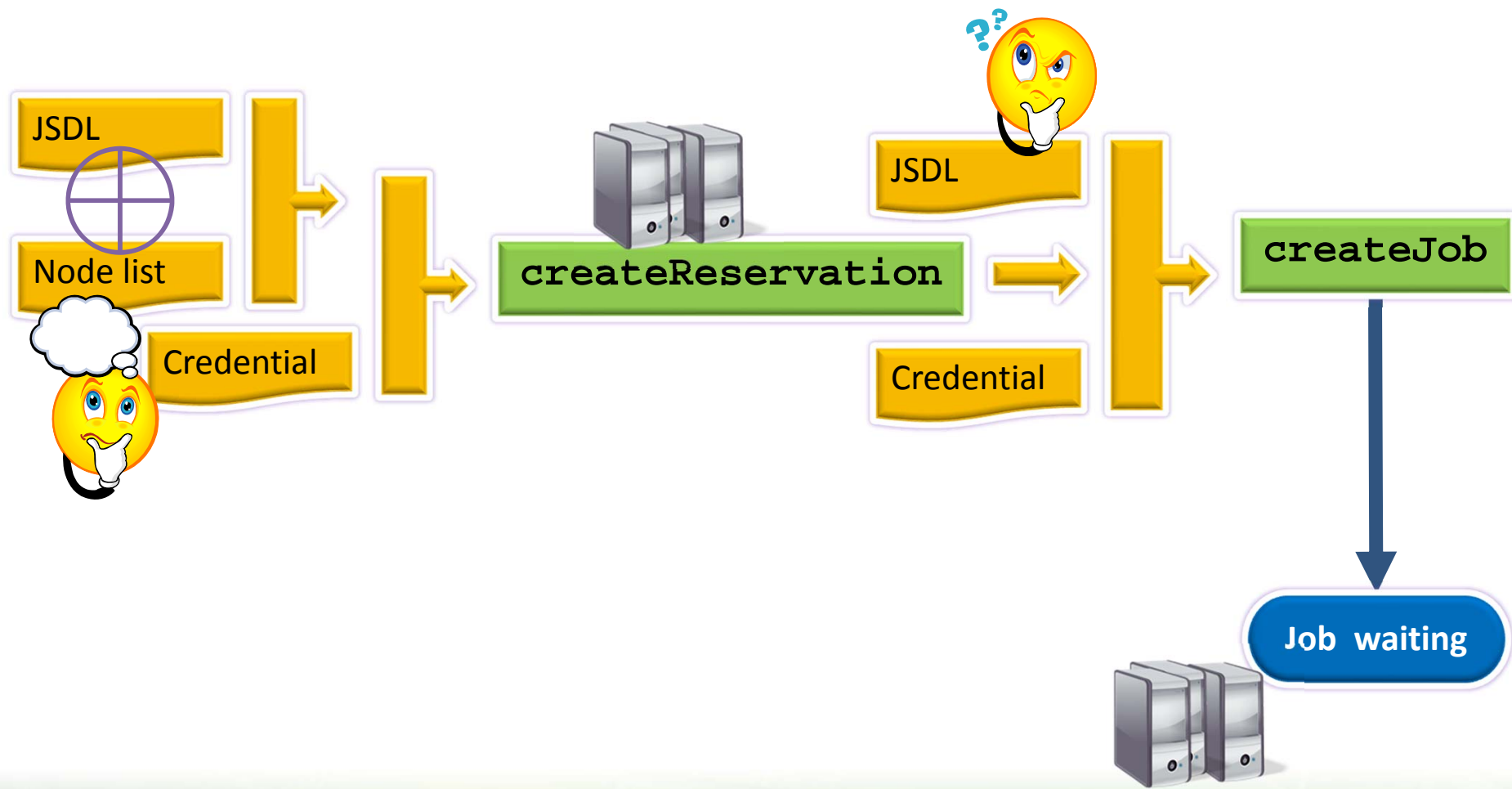


Job Submission in XtreemOS

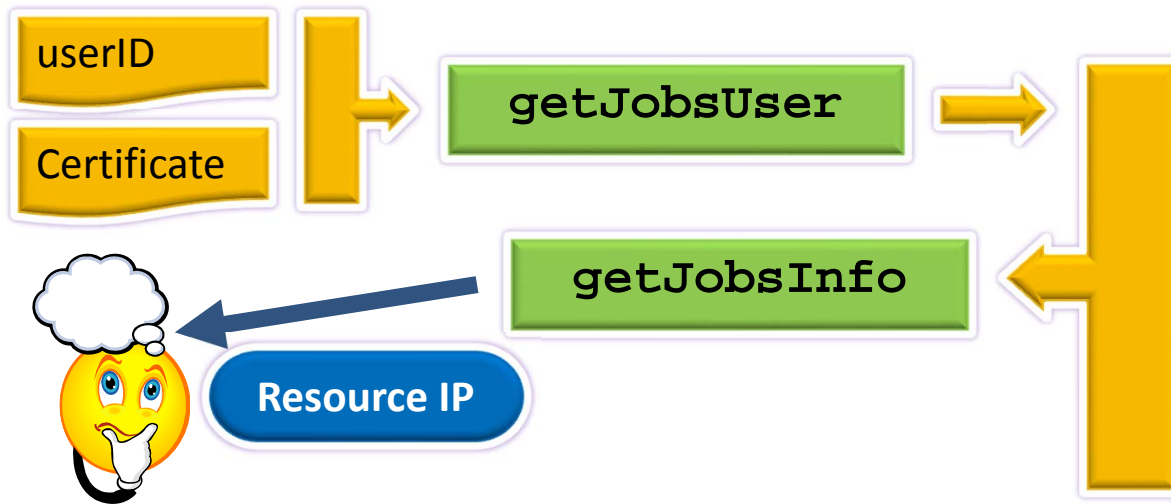
- Job submission
 - Asynchronous
- Distributed XtreemOS Infrastructure
 - Even driven
- SSH-XOS
 - Extension of SSH
 - User connected in their Grid environment
 - User authenticated by their XtreemOS certificates



Interactive Job Submission



Interactive Job Submission





Interactive Job Submission



Resource IP

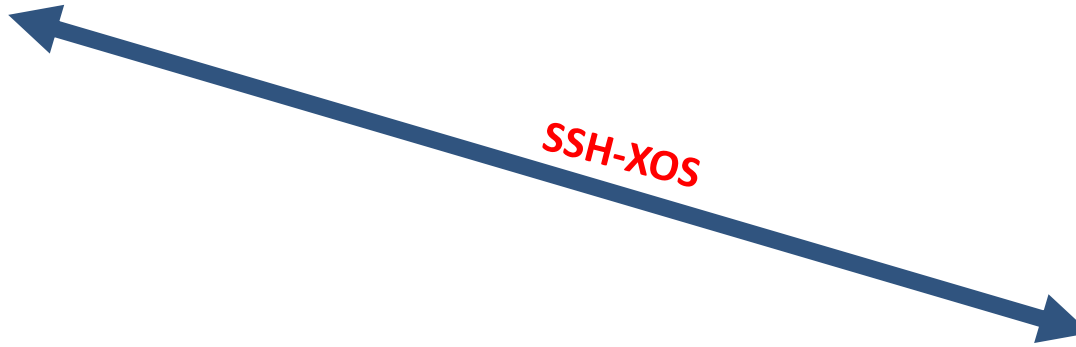
SSH-XOS

Job waiting

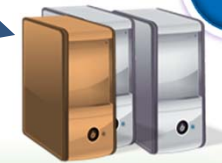




Interactive Job Submission



SSH-XOS



Job running

- XOSJobDaemon
 - User process (service) started for the job on the resource nodes
 - In charge of starting the user application
 - User can communicate with his daemons using ssh-xos (subsystem)

Interactive application

- **xjob -T bash.jsdl**
- **XJOBDaemon** started in the waiting state
- **xjob** connects to the resource using **ssh-xos** using **XOSExec** subsystem
- **XOSExec** connects to **XJOBDaemon**
 - Transfers input/output/error fds and **tty**
 - Starts the application (**/bin/bash**)

- Client side:
 - Need to know if **in/out/err/X11** must be forwarded
- Server (resource) side
 - Need to know if **XOSJobDaemon** must be started
 - When to start the application
 - Immediately: batch job
 - After receiving **in/out/err/X11**: interactive job
 - Immediately+allow connection: batch job + debug

XtreemOS



*Enabling Linux
for the Grid*

Reliable Job Execution

Michael Schöttner

Heinrich-Heine University Duesseldorf, Germany

OGF28, Munich, Germany, 2010



Information Society
Technologies

XtreemOS IP project

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





- **Many checkpointers exist but ...**
- **XtreemOS-GCP checkpointing service**
- **Integration of different checkpointer packages**
- **Communication channel checkpointing with heterogeneous checkpointers**



Grid Jobs


Paris



 Job unit A1

London



 Job unit A2


Duesseldorf



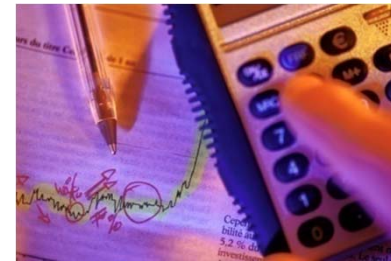
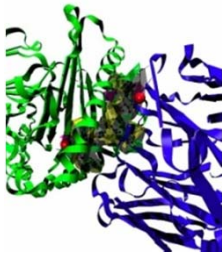
 Job unit A3

Barcelona



 Job unit A4

Job A running in a VO






Paris



 Job unit A1

London



 Job unit A2


Duesseldorf



 Job unit A3

Barcelona



 Job unit A4

Job A running in a VO

solution → backward error recovery



Checkpointing & Restart

- **Checkpointing: Saving periodically the state of the application in stable storage**
- **Restart: In case of a fault we can restart from a checkpoint and do not fall back to the initial state**





Checkpointing & Restart

- **Checkpointing: Saving periodically the state of the application in stable storage**

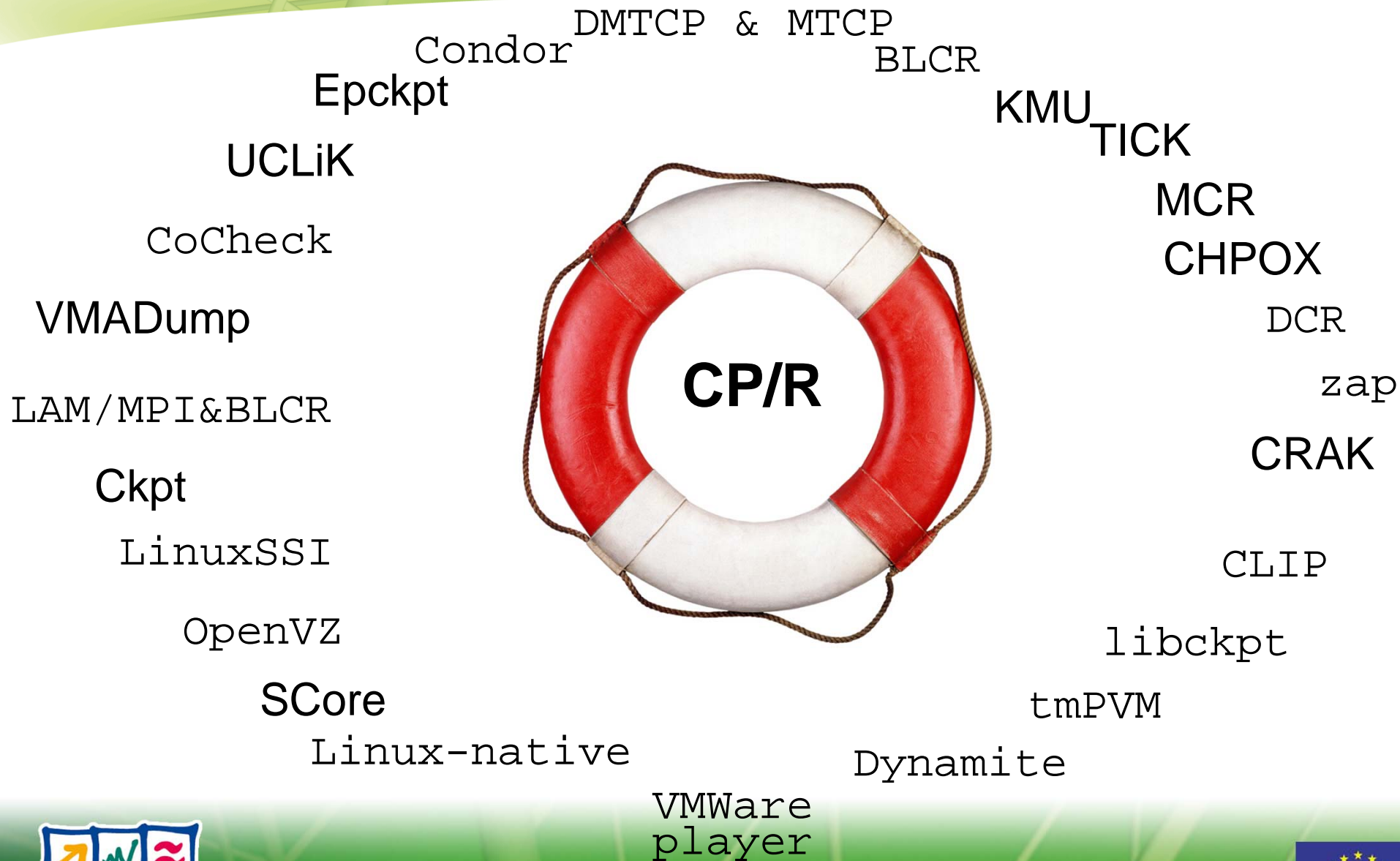
- **Restart: In case of a fault we can restart from a checkpoint and do not fall back to the initial state**

- **Challenges:**
 - Trade-off between costs during fault-free execution and costs at recovery
 - Size of the distributed state may be very large
 - Checkpointing images must be replicated
 - Heterogeneity of checkpointer packages





Many Checkpointers exist



The background of the slide is a green-to-yellow gradient with a perspective grid of white lines that recede towards the top right.

XtreemOS-GCP checkpointing service



- **A grid service integrated within AEM implementing job migration and job fault tolerance for grid jobs**
- **Aims at integrating existing checkpointer packages**
- **Supports transparent and application-level checkpointing**





- **User/application commands**

```
$xjobcheckpoint JobID
```

```
$xjobrestart JobID CPversion
```

- **JSDL file extensions**

- Extended by checkpointing tags
- Checkpointer requirements
- Protocols and parameters
- ...





JSDL File Sample: Header

```
<?xml version="1.0" encoding="UTF-8"?>  
  <JobDefinition xmlns="http://schemas.ggf.org/jsdl/2005/10/jsdl">  
    <JobDescription>  
      <JobIdentification>  
        <Description>Your application</Description>  
        <JobProject>XtreemOS-GCP Test</JobProject>  
      </JobIdentification>  
  
      <Application>  
      ....
```





<Application>

```
<POSIXApplication xmlns:ns1="http://schemas.ggf.org/jsdl/2005/06/jsdlposix">
```

```
  <Executable>/home/schoettner/XtreemGCPtest</Executable>
```

```
  <Argument>192.168.2.1</Argument>
```

```
  <Output>/tmp/out_cl-cf.txt</Output>
```

```
  <Error>/tmp/err_cl-cf.txt</Error>
```

```
</POSIXApplication>
```

<Resources>

```
  <TotalResourceCount>
```

```
    <exact>3</exact>
```

```
  </TotalResourceCount>
```

</Resources>

</Application>





<JobCheckpointing>

<Initiator>System</Initiator>

<ProtocolManagement>

<Name>CoordinatedCheckpointing</Name>

<Parameter>1hour</Parameter>

</ProtocolManagement>

<FileManagement>

<ReplicationLevel>5</ReplicationLevel>

</FileManagement>

<JobCheckpointingMatching>

<MultiThread>Yes</MultiThread>

<Sockets>Yes</Sockets>

</JobCheckpointingMatching>

</JobCheckpointing>





Grid level

Job Checkpointer
(Job Manager extension)

Node Level

Job-unit Checkpointer
(Execution Manager extension)

Job-unit Checkpointer
(Execution Manager extension)

Common Checkpointer API

SSI-Translib

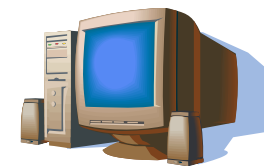
BLCR-Translib

LinuxSSI Kernel Checkp.

BLCR Checkpointer



XtreemOS-SSI cluster



XtreemOS PC



Common Checkpointer API

- **Uniform access to different checkpointer packages**
→ implemented by a translib (shared library)
- Translate function signatures
- Translate job-to-Linux process group
- Translate user ids: grid user id -> local userID
- Translate/provide callbacks for applications





Common Checkpointer API

- **To which extent must existing checkpointers be adapted to support various checkpointing protocols?**

- **We need the following sequences**

- **Stop**
 - **Checkpoint**
 - **resume_cp**
- } **Checkpoint**
-
- **Rebuild**
 - **resume_rst**
- } **Restart**





Callback Management

- **Implemented in generic part of translib**
- **Called before and after a checkpoint and after restart**
- **Common API for application callback registration**
- **Useful for:**
 - **Application optimizations**
 - **Complement checkpointer incapacibilities**
 - **Checkpointing communication channels**





- **Currently, supported checkpointer packages**
- **BLCR**
- **OpenVZ**
- **MTCP**
- **LinuxSSI**
- **(Linux native)**





Checkpoint files

- **Must be replicated**
- **And accessible from each grid node**
- **Stored in XtreemFS, providing:**
 - **Stripping**
 - **Automatic replication**
 - **Location-transparent access**
 - **Access control via XtreemOS user accounts**





▪ **Coordinated Checkpointing**

- All involved nodes are synchronized during cp-time
- Last set of checkpoints are always consistent
- Allows fast recovery
- Used for migration

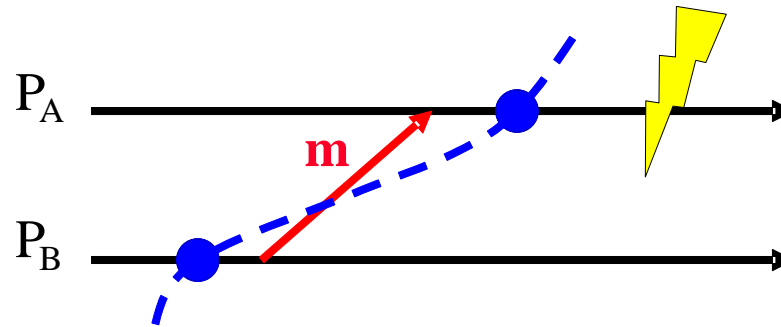
▪ **Independent Checkpointing**

- No coordination overhead during checkpointing
- But recovery requires calculation of consistent set of cps
- System needs to store older checkpoints too

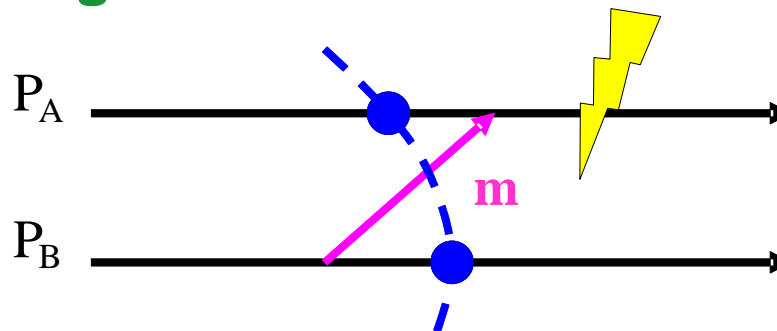


Consistent Checkpoints

- Must avoid orphan messages



- And lost messages:





Channel checkpointing with heterogeneous checkpointers

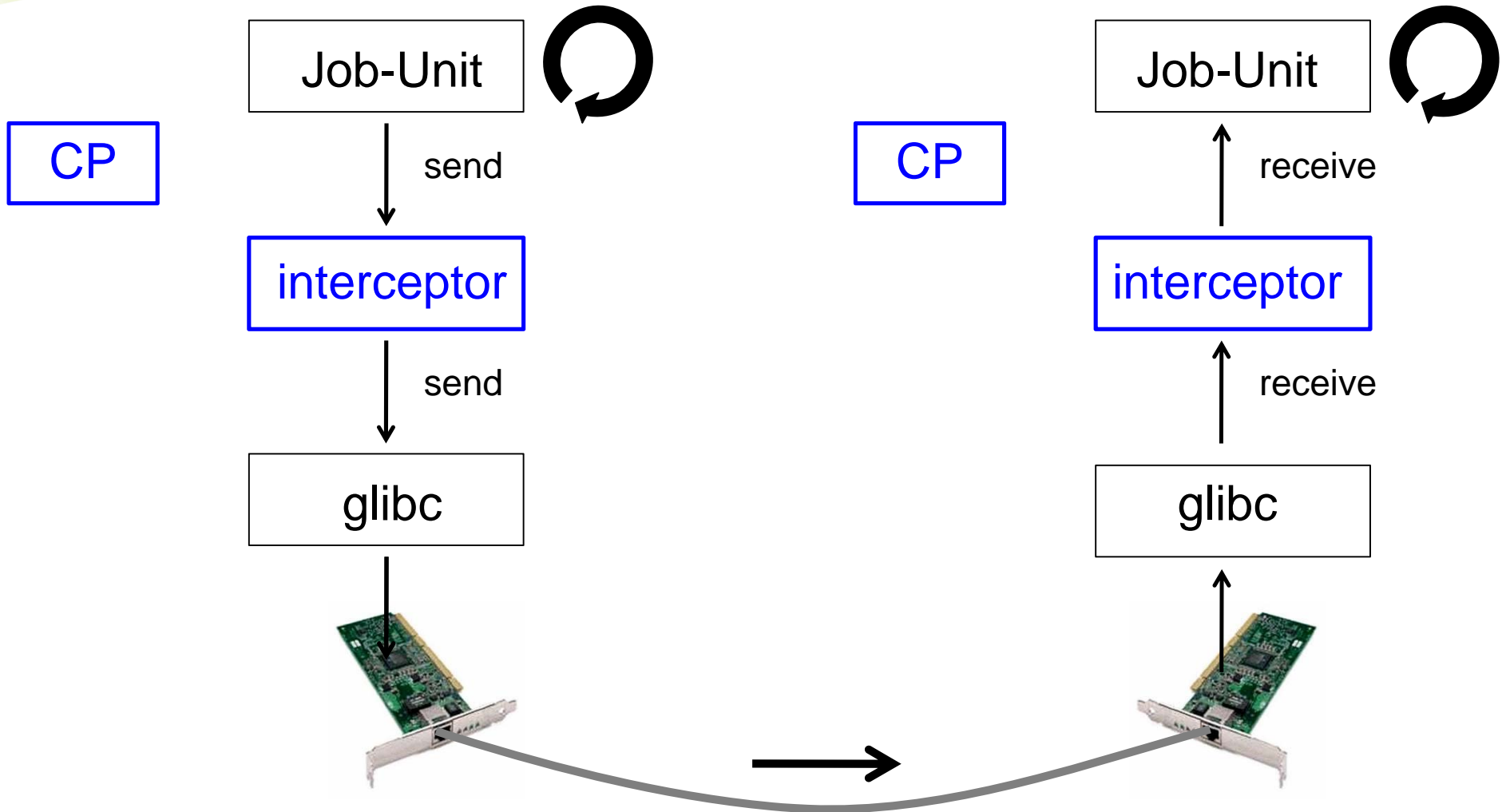


- **Distributed CP must handle in-transit messages**
- **Single node CP do not care about messages**
- **Distributed CPs cannot operate in a heterogeneous setup**

- **Aims of XtreemOS-GCP:**
 - **no checkpointer modifications**
 - **application transparency**
 - **migration support**

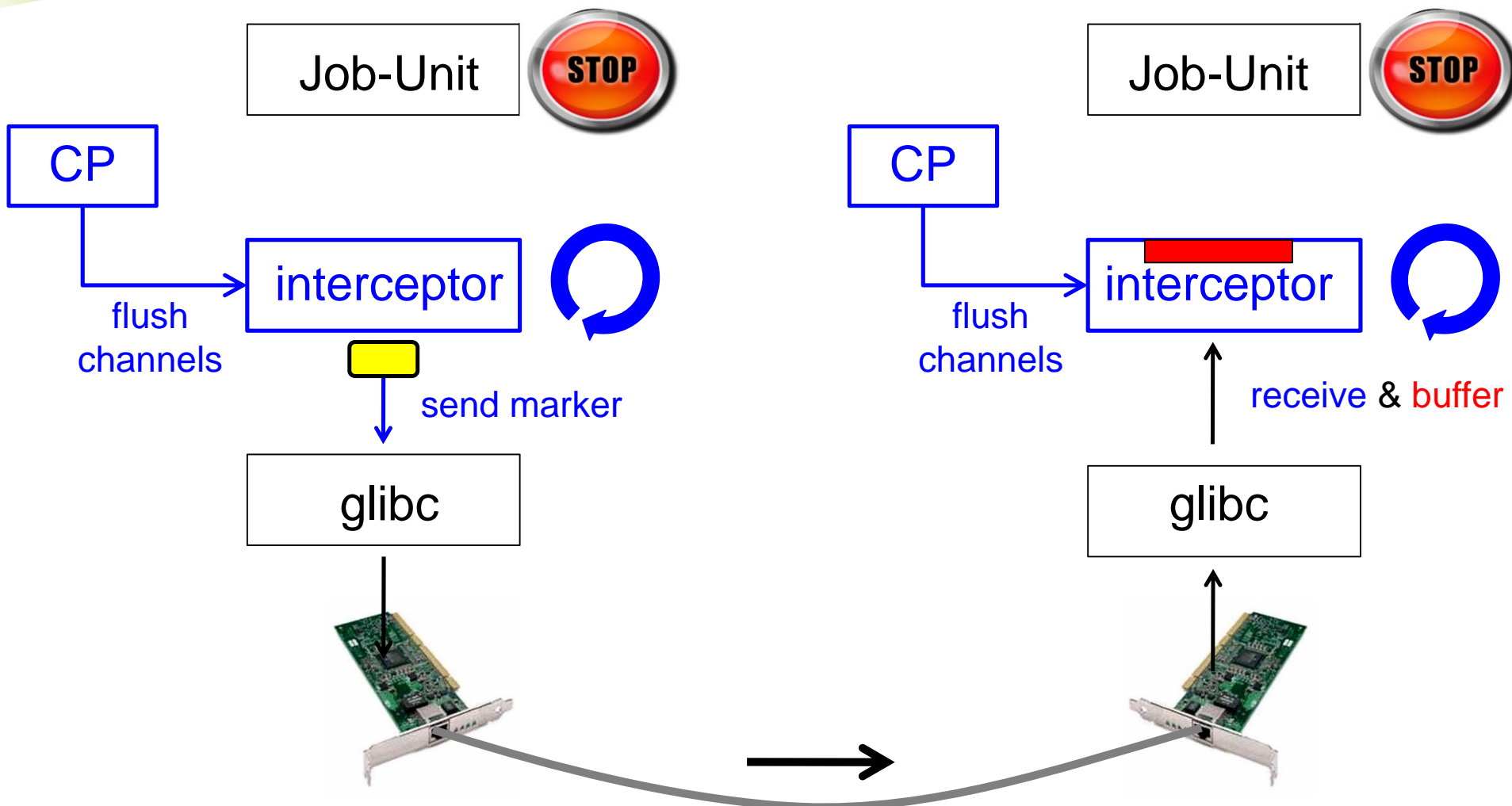


Channel CP: normal operation



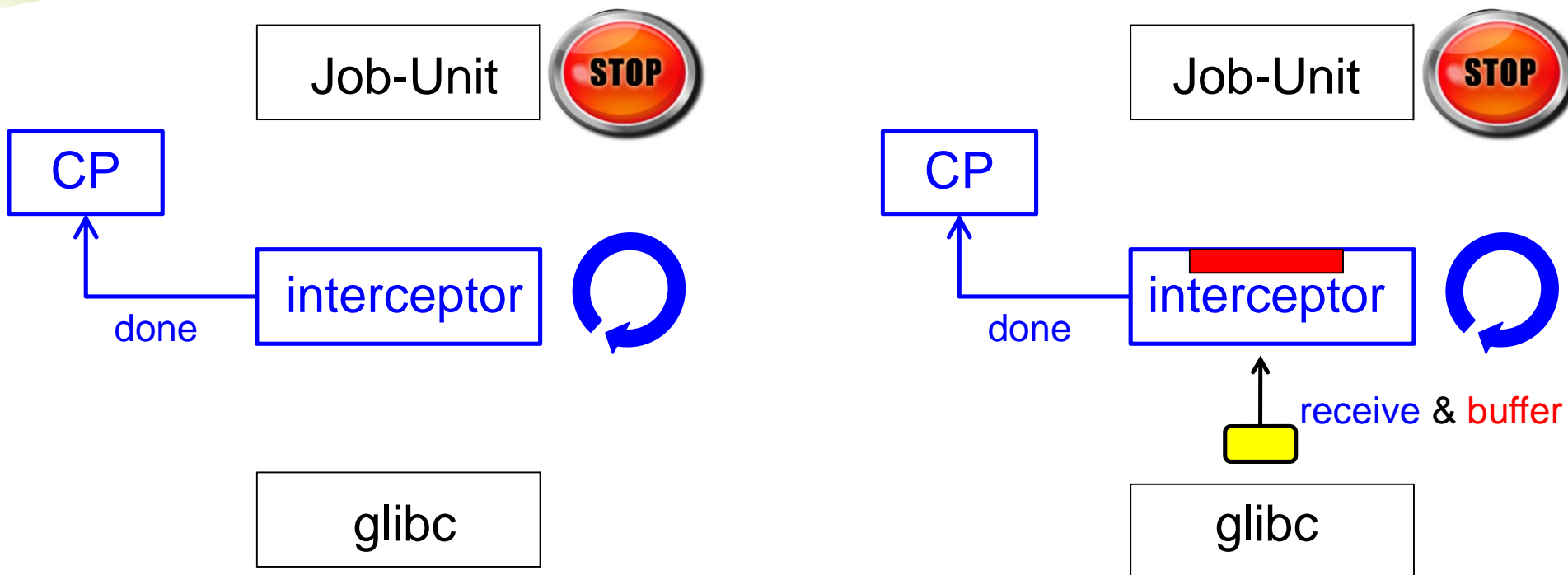


Channel CP: flush operation



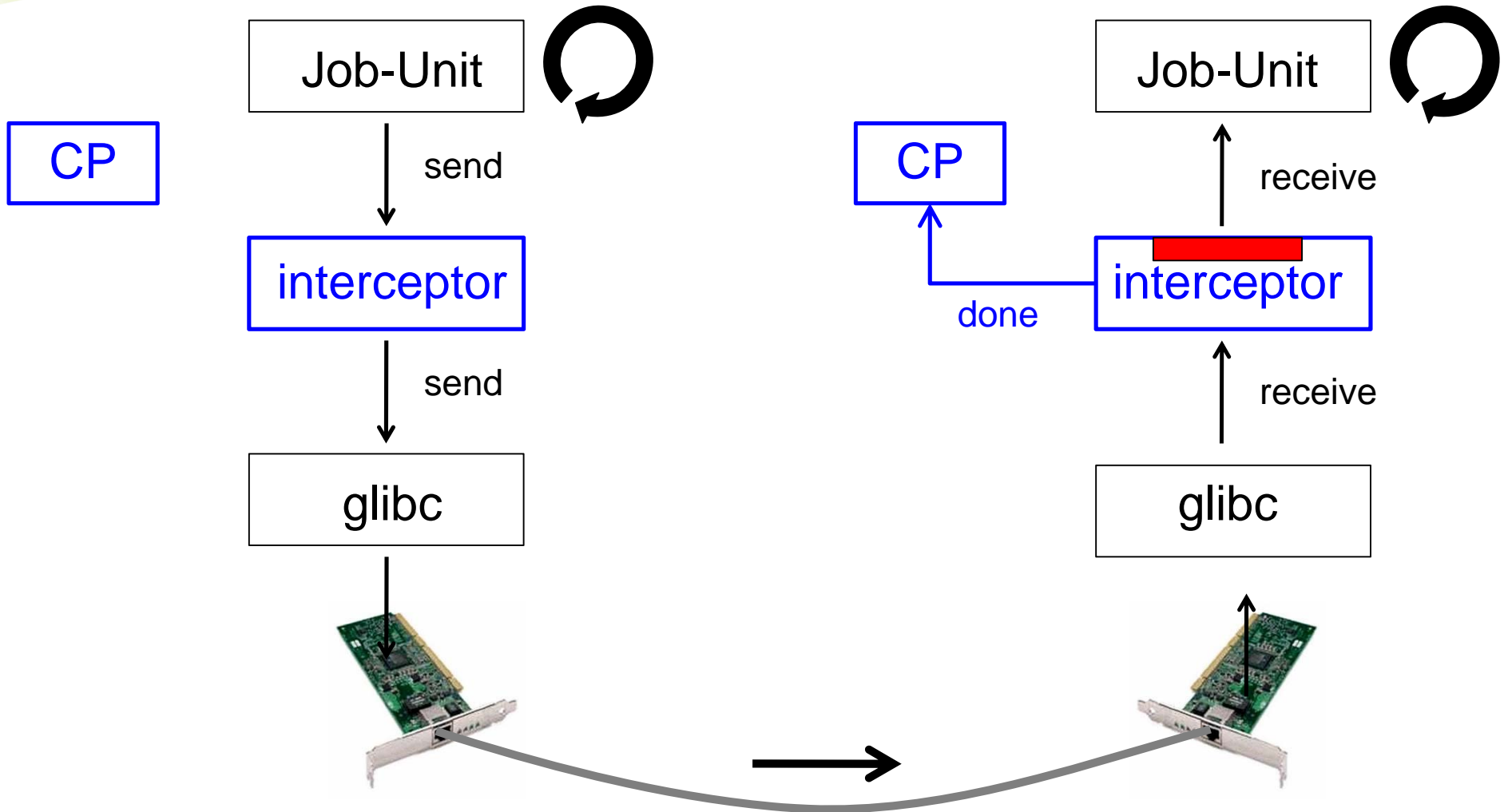


Channel CP: flush operation





Channel CP: resume operation





- **XtreemOS-GCP offers migration and fault tolerance in grids by providing checkpointing and restart**
- **It is designed for heterogeneous setups integrating existing checkpointing packages**
- **Future work:**
 - virtual machine support & adaptive checkpointing**



Acknowledgment

- **EC for funding XtreamOS**
- **XtreamOS- GCP contributors:**
 - **Heinrich-Heine Universität Düsseldorf
John Mehnert-Spahn**
 - **INRIA, Rennes, France
Christine Morin, Thomas Ropars, Surbi Chitre**





Backup



Common Checkpointer API

- **xos_prepare_environment**
 - Identify relevant process and process group reference type

- **xos_stop_job_unit**
 - Execute potential callbacks
 - Stop all processes of a job unit (pre-checkpoint)

- **xos_checkpoint_job_unit**
 - Checkpoint job unit by calling underlying checkpointer

- **xos_resume_job_unit_cp**
 - Resume after checkpointer operation
 - Execute potential callbacks (post-checkpoint)





- **xos_rebuild_job_unit**
 - Rebuild job-unit from checkpoint image

- **xos_resume_job_unit_rst**
 - Resume processes
 - Execution of potential callbacks (post-restart)

Notes



Checkpointing Differences

- **Some do not handle all resources**
- **Most of them focus on single nodes**
- **Distributed CPs implement mainly coordinated protocols**
- **Some are embedded in a framework, e.g. MPI**
- **Some offer transparent checkpointing others not**





- **Application**
 - Application implements checkpointing itself

- **Library**
 - Application linked against checkpointer library
 - Transparent checkpointing

- **Kernel**
 - OS supports checkpointing/restart
 - Transparent checkpointing

- **Virtual machines**
 - Suspend & resume functionality

- **MPI side (mpich-1.2.7-p1) :**
 - Ability to change “ssh” or “rsh” for our process.
 - `./configure -rsh=/usr/bin/xsubMPI -prefix=/usr/bin`
 - `P4_RSHCOMMAND=/usr/bin/xsubMPI`
- **MPIrun cannot be used, the list of nodes**
 - Needs IP and port (not just IP) and reservation ID
- **XOS_mpirun**
 - 99% samecode as MPIrun
 - `EXTERNALRES="xreservation -f resfil.jsdl -n <numProcs> -t <duration> -m"`
 - Already available in MPIRun

How it works?

- **EXTERNALRES** command called inside mpirun
 - Gets the list of resources from an external application.
- **A reservation is created with the num of nodes**
 - According to jsdl restrictions
 - The nodes are saved to a file, MyResource
- **MyResource is readed and issued to a node**
 - This node will be the master.
- **Slaves are creates in the other nodes**
 - On behalf of the user

Some Considerations

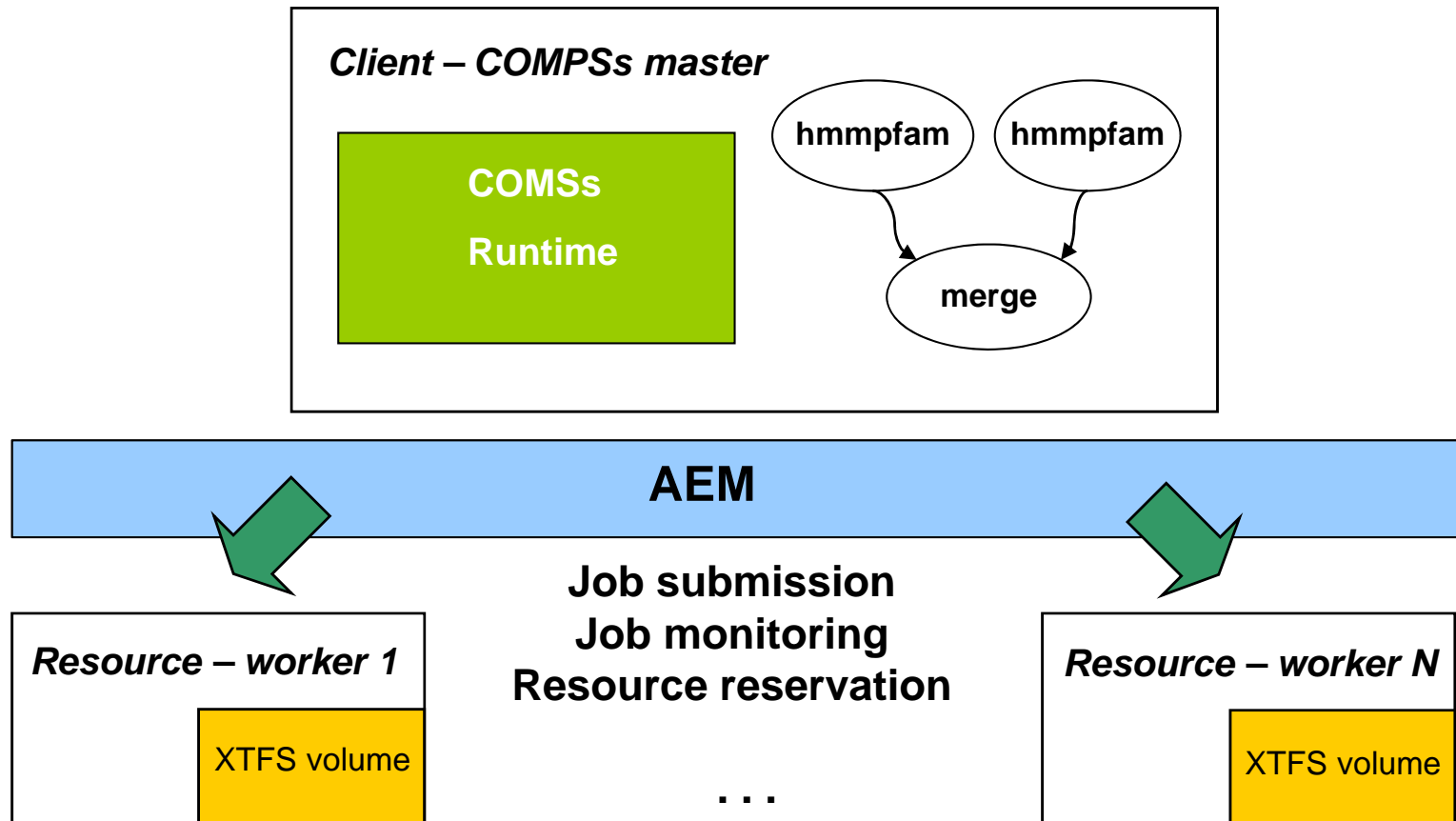
- **XtremFS has to be used**
 - Executable
 - User certificate
 - AEM client configuration file
 - No node IP (as we do not know which node will use it)

- **Framework**
 - Ease development & execution of Grid-unaware apps.
- **Programming model**
 - User selects the methods to be run on the Grid
- **Runtime**
 - Calls to selected methods become remote tasks
 - A task dependency graph (workflow) is created

- **Hmmpfam application**
 - Bioinformatics field
 - Compares aminoacid sequences with protein families
 - Embarrassingly parallel
- **COMPSs-hmmpfam on top of XtreemOS**
 - Tasks generated by COMPSs are submitted to the Grid using the XtreemOS AEM component
 - Currently using the internal interface
 - SAGA interface is work in progress
 - COMPSs exploits parallelism as much as possible



COMPSSs and AEM



- **Resource reservation**

- COMPSs requests to AEM a number of resources
 - Before starting the application
 - Will be used as workers by COMPSs
- Once the application ends, the reservation is released

- **Job submission**

- Each COMPSs task is mapped to a XtreemOS job
 - Submitted to the above mentioned reservation

AEM functionality used

- **Resource management**
 - AEM is in charge of scheduling
 - This is done by COMPSs in other environments
- **Job monitoring**
 - COMPSs uses AEM to check the state of submitted jobs
 - Each time a job ends COMPSs is informed and
 - The task dependency graph is updated
 - Newly dependency-free tasks are submitted for execution

- **Dependencies**
 - Currently not being used, but it is work in progress
- **XtreemFS**
 - Executables and data are in XtreemFS
 - Currently replications is managed by COMPSs
 - Stage in/out
 - Working on relying on XtreemFS to avoid explicit stage in/out

XtreemOS

Enabling Linux
for the Grid



XOSAGA: the XtreemOS API

Thilo Kielmann

VU University, Amsterdam

kielmann@cs.vu.nl

XtreemOS IP project

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



Information Society
Technologies



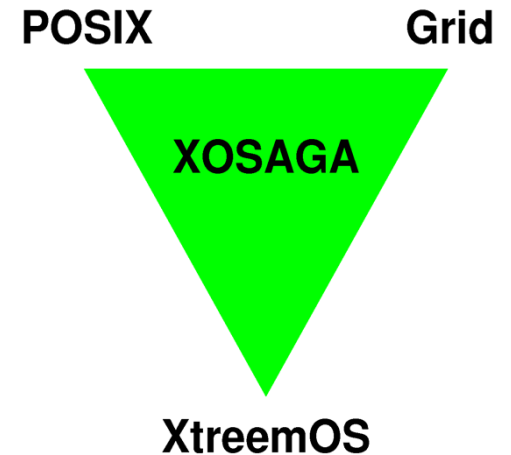


Design Challenges:

- POSIX look-and-feel (for Linux apps)
- grid look-and-feel (for grid apps)
- provide XtreemOS functionality

Approach:

- Start from OGF standardized SAGA API
 - POSIX look-and-feel
 - accepted grid API
- build XtreemOS-specific extensions (XOSAGA)





Grid Programming Nightmare: Copy a File with Globus GASS

```

int copy_file (char const* source,   char const* target)
{
    globus_url_t                source_url;
    globus_io_handle_t          dest_io_handle;
    globus_ftp_client_operationattr_t source_ftp_attr;
    globus_result_t              result;
    globus_gass_transfer_requestattr_t source_gass_attr;
    globus_gass_copy_attr_t      source_gass_copy_attr;
    globus_gass_copy_handle_t     gass_copy_handle;
    globus_gass_copy_handleattr_t gass_copy_handleattr;
    globus_ftp_client_handleattr_t ftp_handleattr;
    globus_io_attr_t              io_attr;
    int                            output_file = -1;

    if ( globus_url_parse (source_URL, &source_url) != GLOBUS_SUCCESS )
    {
        printf ("can not parse source_URL \"%s\"\n", source_URL);
        return (-1);
    }

    if ( source_url.scheme_type != GLOBUS_URL_SCHEME_GSIFTP &&
        source_url.scheme_type != GLOBUS_URL_SCHEME_FTP   &&
        source_url.scheme_type != GLOBUS_URL_SCHEME_HTTP   &&
        source_url.scheme_type != GLOBUS_URL_SCHEME_HTTPS  ) {
        printf ("can not copy from %s - wrong prot\n", source_URL);
        return (-1);
    }

    globus_gass_copy_handleattr_init (&gass_copy_handleattr);
    globus_gass_copy_attr_init       (&source_gass_copy_attr);

    globus_ftp_client_handleattr_init (&ftp_handleattr);
    globus_io_fileattr_init           (&io_attr);

    globus_gass_copy_attr_set_io      (&source_gass_copy_attr,
                                       &io_attr);

    globus_gass_copy_handleattr_set_ftp_attr
                                       (&gass_copy_handleattr,
                                       &ftp_handleattr);

    globus_gass_copy_handle_init      (&gass_copy_handle,
                                       &gass_copy_handleattr);

```

```

    if (source_url.scheme_type == GLOBUS_URL_SCHEME_GSIFTP ||
        source_url.scheme_type == GLOBUS_URL_SCHEME_FTP   ) {
        globus_ftp_client_operationattr_init (&source_ftp_attr);
        globus_gass_copy_attr_set_ftp (&source_gass_copy_attr,
                                       &source_ftp_attr);
    }
    else {
        globus_gass_transfer_requestattr_init (&source_gass_attr,
                                               source_url.scheme);
        globus_gass_copy_attr_set_gass(&source_gass_copy_attr,
                                       &source_gass_attr);
    }

    output_file = globus_libc_open ((char*) target,
                                   O_WRONLY | O_TRUNC | O_CREAT,
                                   S_IRUSR | S_IWUSR | S_IRGRP |
                                   S_IWGRP);

    if ( output_file == -1 ) {
        printf ("could not open the file \"%s\"\n", target);
        return (-1);
    }

    /* convert stdout to be a globus_io_handle */
    if ( globus_io_file_posix_convert (output_file, 0,
                                       &dest_io_handle)
         != GLOBUS_SUCCESS) {
        printf ("Error converting the file handle\n");
        return (-1);
    }

    result = globus_gass_copy_register_url_to_handle (
        &gass_copy_handle, (char*)source_URL,
        &source_gass_copy_attr, &dest_io_handle,
        my_callback, NULL);

    if ( result != GLOBUS_SUCCESS ) {
        printf ("error: %s\n", globus_object_printable_to_string
              (globus_error_get (result)));
        return (-1);
    }

    globus_url_destroy (&source_url);
    return (0);
}

```



Relief: Copy a File with SAGA

```
import org.ogf.saga.error.SagaException;
import org.ogf.saga.file.File;
import org.ogf.saga.file.FileFactory;
import org.ogf.saga.url.URL;

public class CopyFile {

    void copyFile(URL sourceUrl, URL targetUrl) {
        try {
            File f = FileFactory.createFile(sourceUrl);
            f.copy(targetUrl);
        } catch (SagaException e) {
            System.err.println(e);
        }
    }
}
```

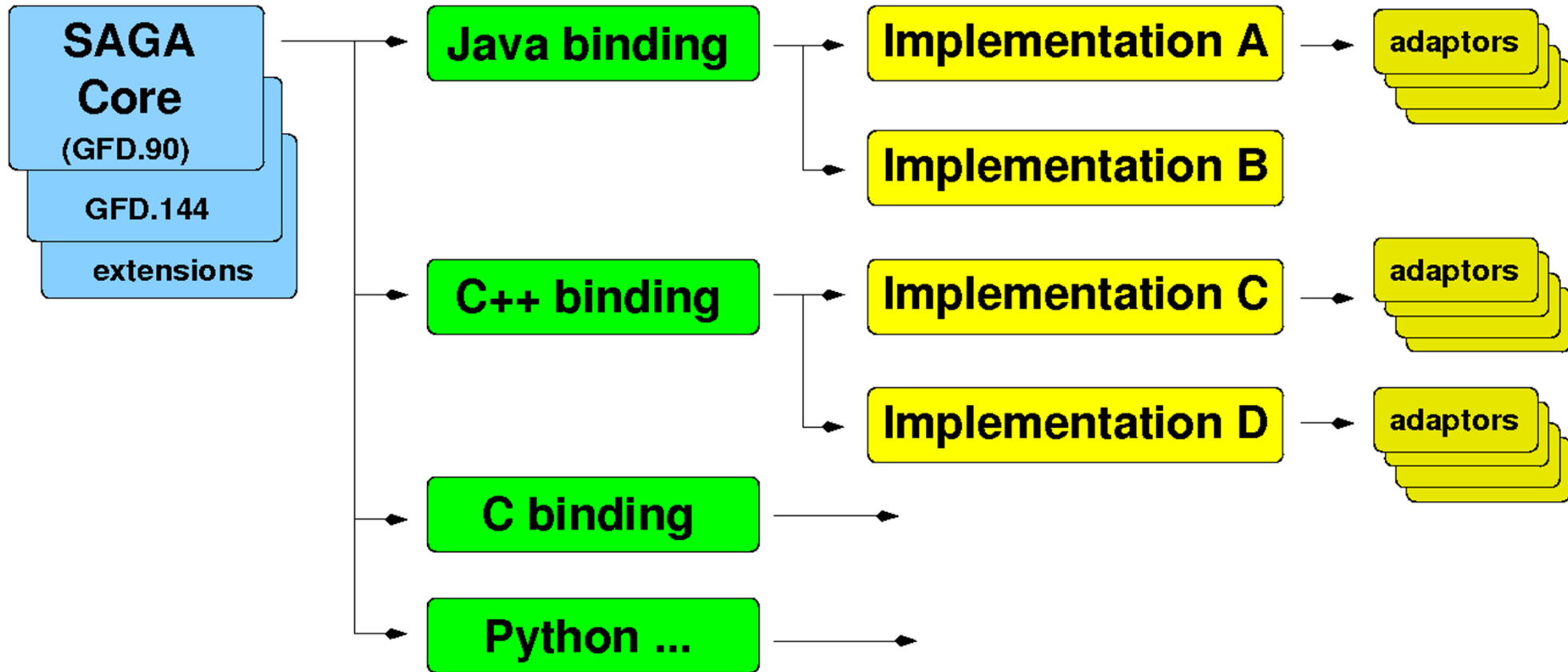
- Provides the high level abstraction that application programmers need; will work across different systems
- Shields the gory details of lower-level middleware systems



- **A programming interface for grid applications**
 - provides common grid functionality
 - deals with (remote) resources explicitly
 - e.g., files, jobs
 - simple (80/20 rule, limited in scope)
 - integrated (“consistent”)
 - stable: does not change (incompatibly)
 - uniform, across middleware platforms
 - high level, what applications need



The SAGA Landscape

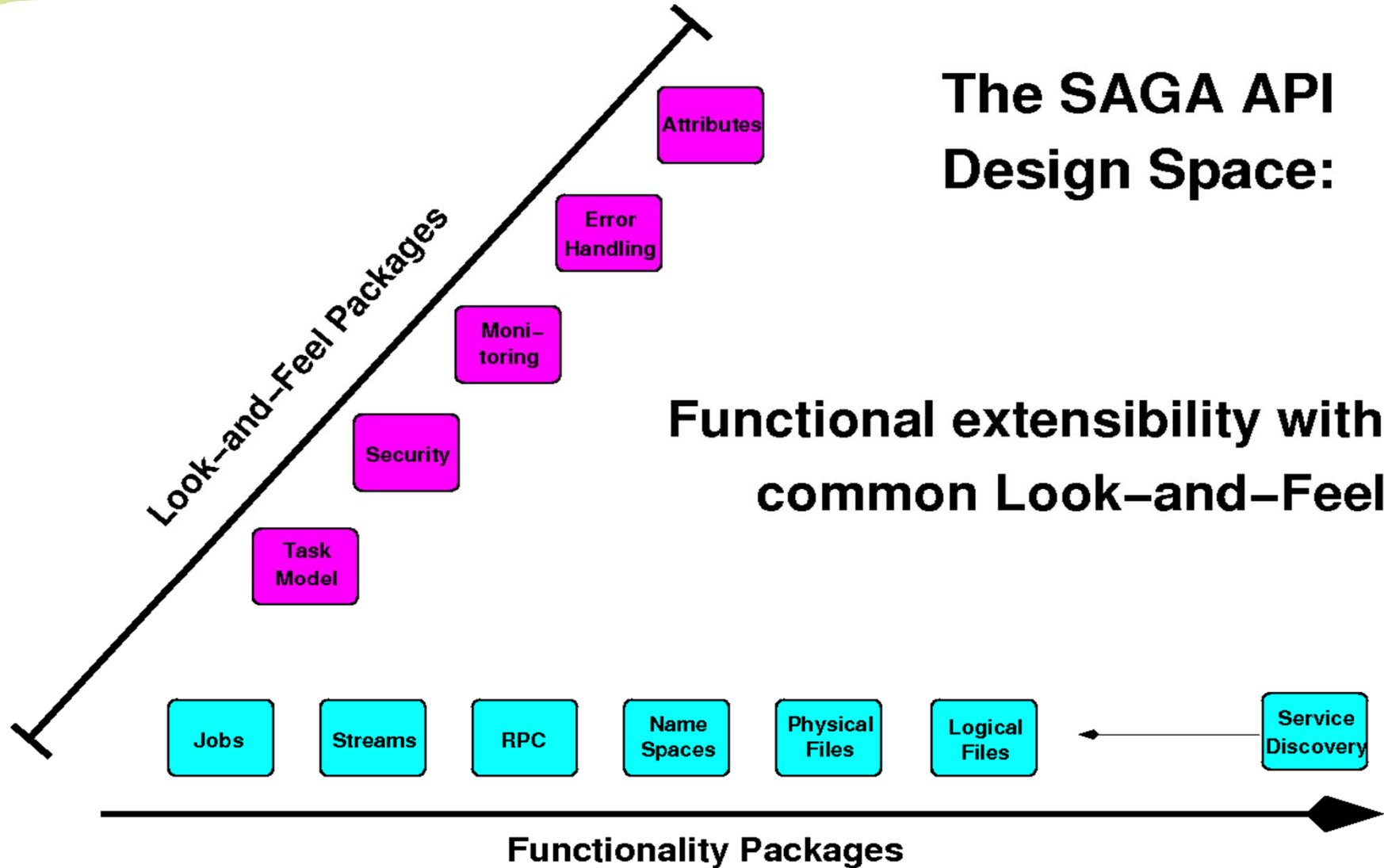




SAGA API Design Overview

The SAGA API Design Space:

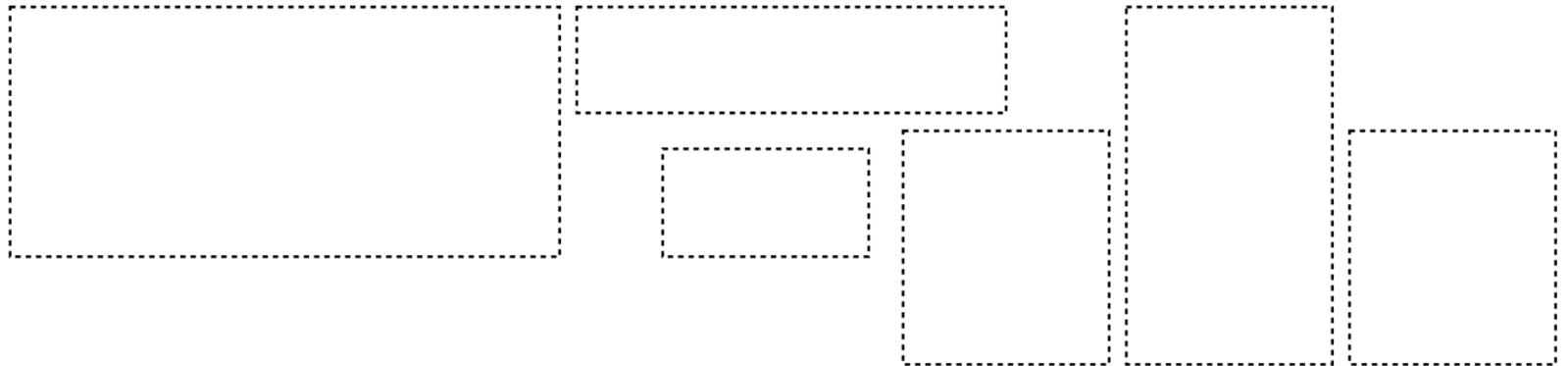
Functional extensibility with common Look-and-Feel





SAGA Interface Hierarchy

Look & Feel



Look and feel: Top level Interfaces; Core SAGA objects needed by other API packages that provide specific functionality -- capability providing packages e.g., jobs, files, streams, namespaces etc.



SAGA Interface Tour

Look & Feel

Base Object

object

● → inherits
● → implements

interface

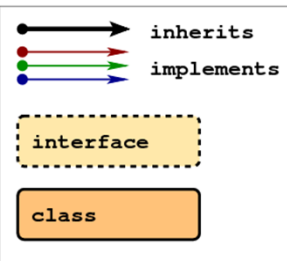
class

The common root for all SAGA classes.

Provides unique ID to maintain a list of SAGA objects. Provides methods (`get_id()`) essential for all SAGA objects



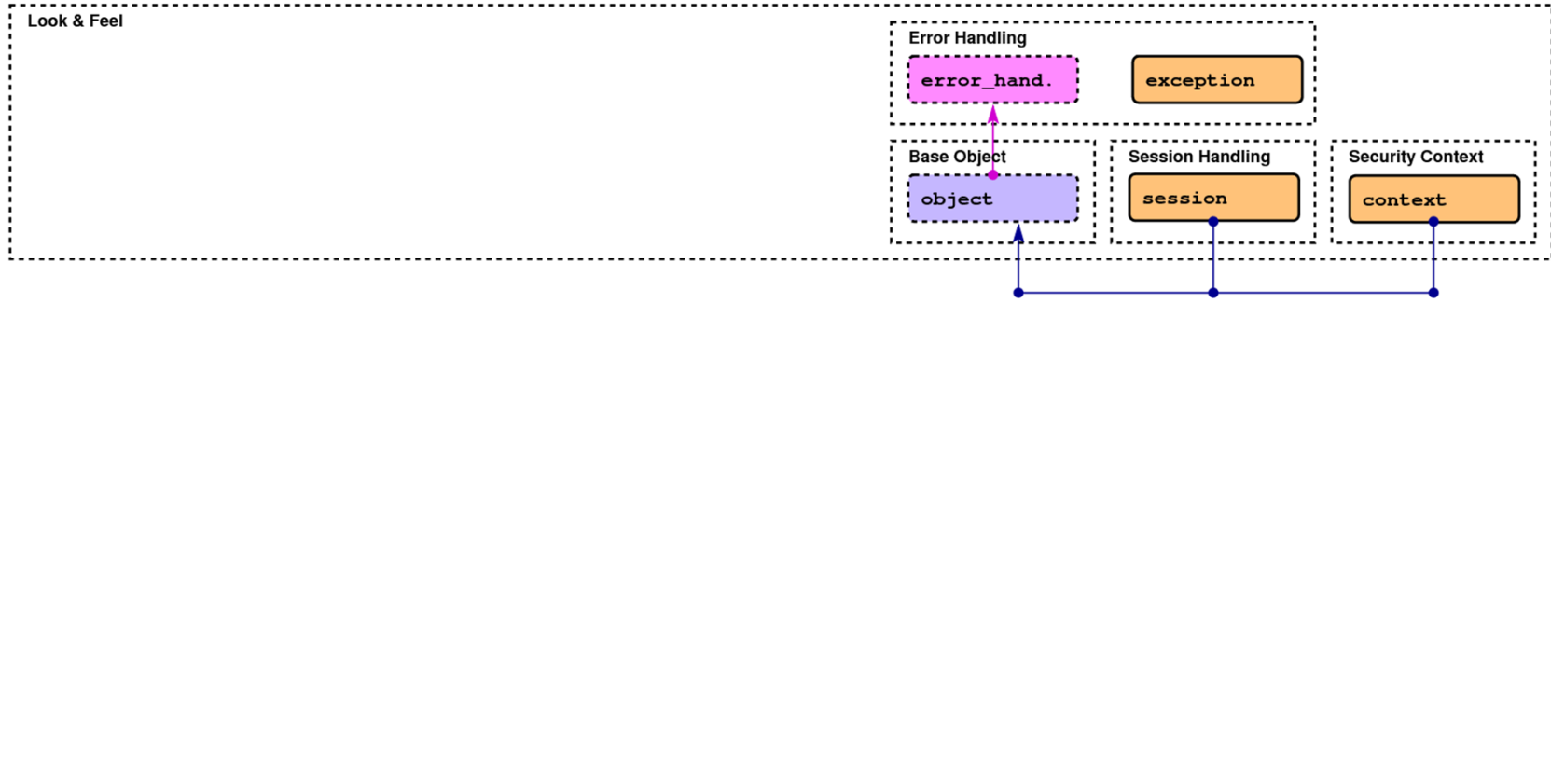
Errors and Exceptions



SAGA defines a hierarchy of exceptions
(and allows implementations to fill in specific details)



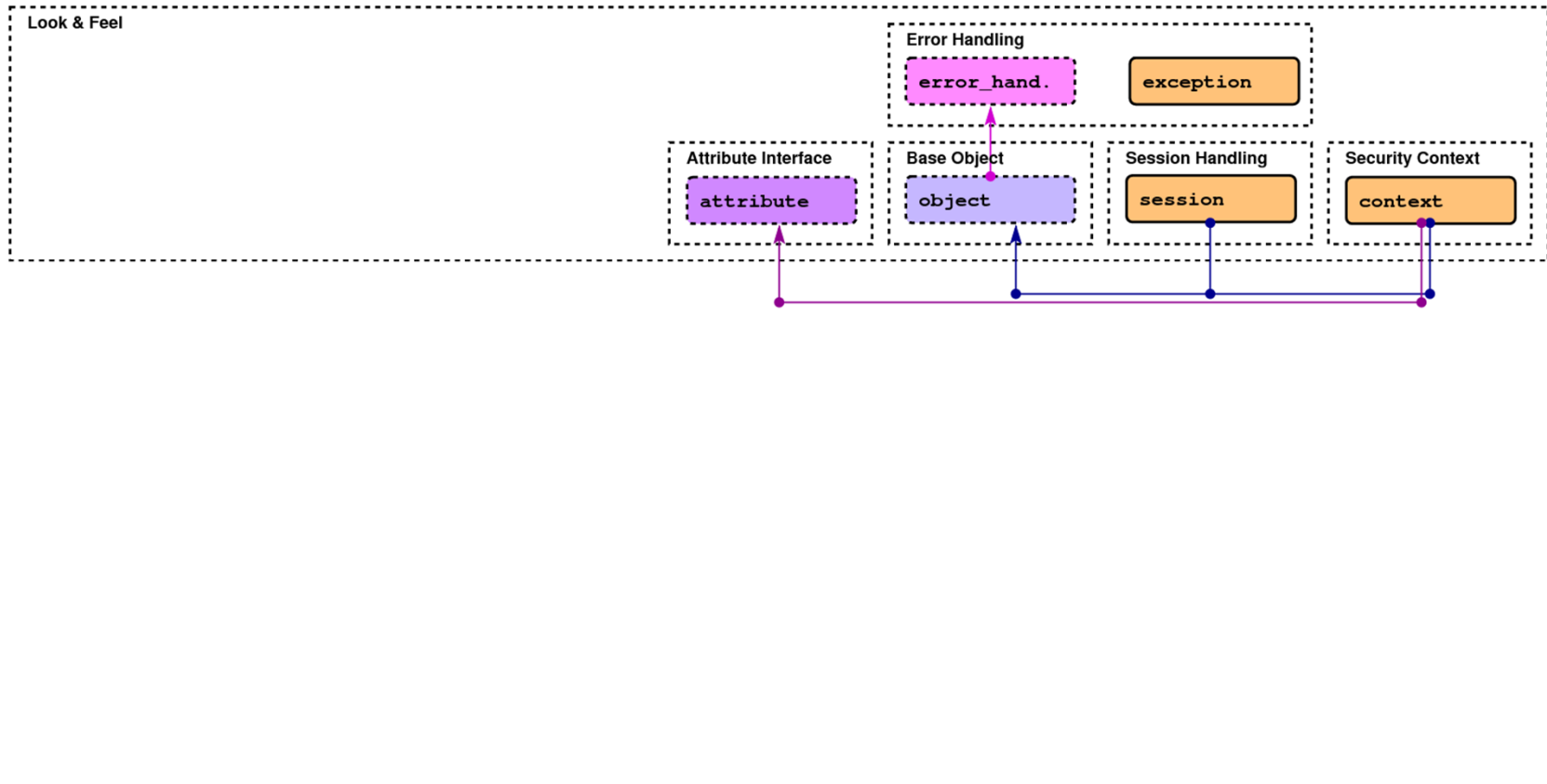
Session, Context, Permissions



Context provides functionality of a session handle and isolates independent sets of SAGA objects. Only needed if you wish to handle multiple credentials. Otherwise *default context* is used.



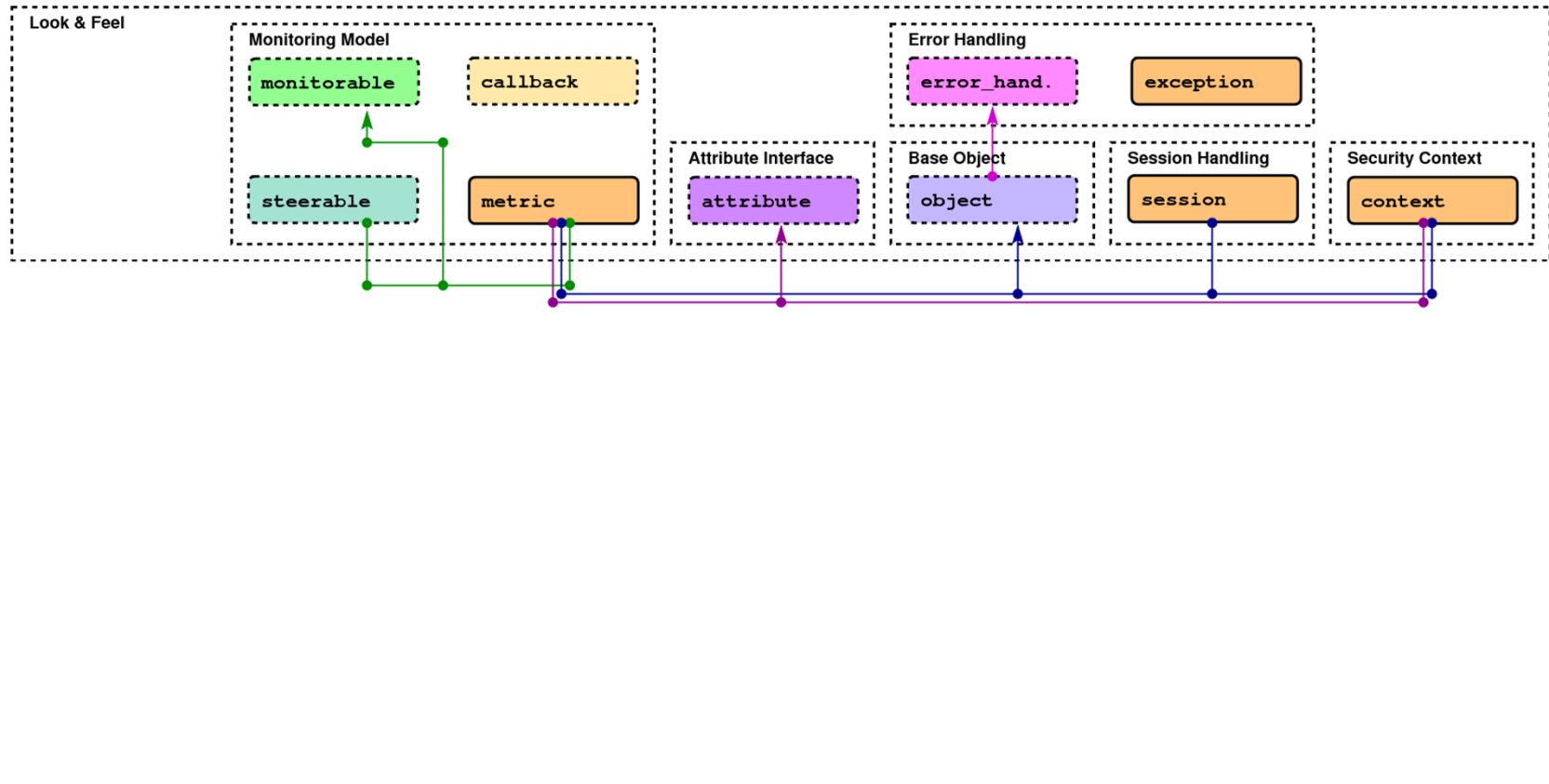
Attributes



Where attributes need to be associated with objects, e.g. Job-submission. Key-value pairs, e.g. for resource descriptions attached to the object.



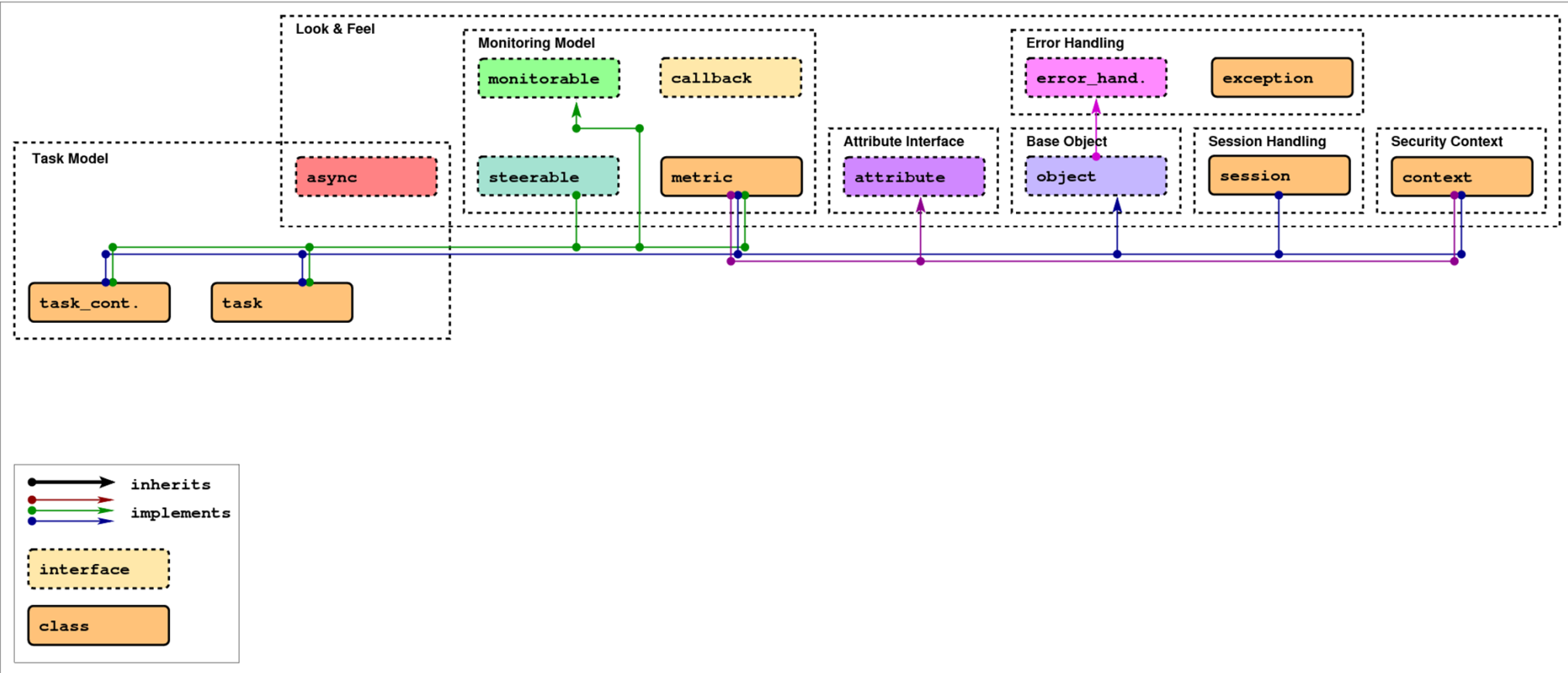
Application Monitoring



Metric defines application-level data structure(s) that can be monitored and modified (steered). Also, task model requires state monitoring.



Asynchronous Operations, Tasks



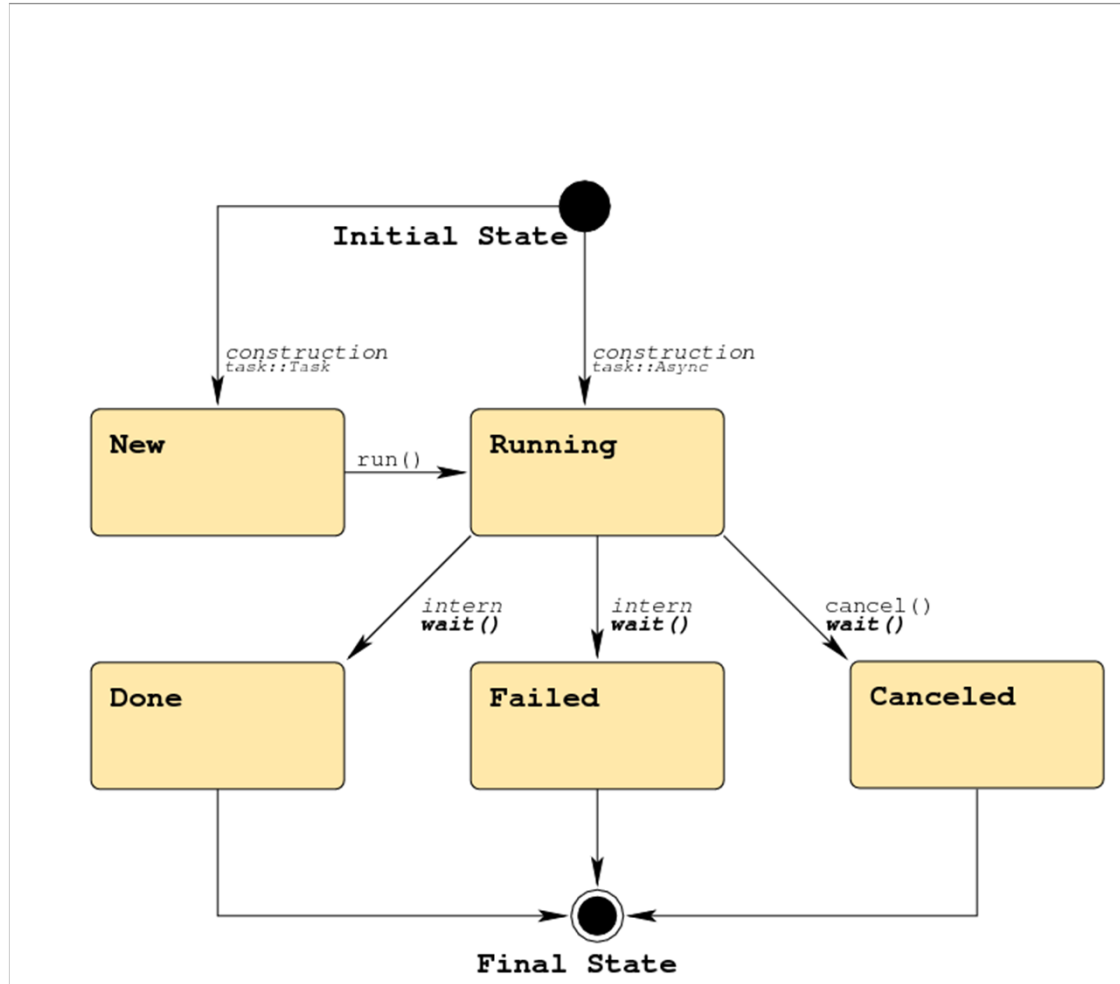
Most calls can be synchronous, asynchronous, or tasks (need explicit start.)



- **All SAGA objects implement the task model**
- **Every method has three “flavours”**
 - synchronous version - the implementation
 - asynchronous version - synchronous version wrapped in a task (thread) and started
 - task version - synchronous version wrapped in a task but not started (task handle returned)



SAGA Task Model





SAGA Task Model

```
import org.ogf.saga.error.SagaException;
import org.ogf.saga.file.File;
import org.ogf.saga.file.FileFactory;
import org.ogf.saga.task.Task;
import org.ogf.saga.task.TaskMode;
import org.ogf.saga.url.URL;
import org.ogf.saga.url.URLFactory;

public class TaskModelExample {

    void foo() throws SagaException {

        URL src = URLFactory.createURL("any://host.net/data/src.dat");
        URL dst = URLFactory.createURL("any://host.net/data/dest1.dat");

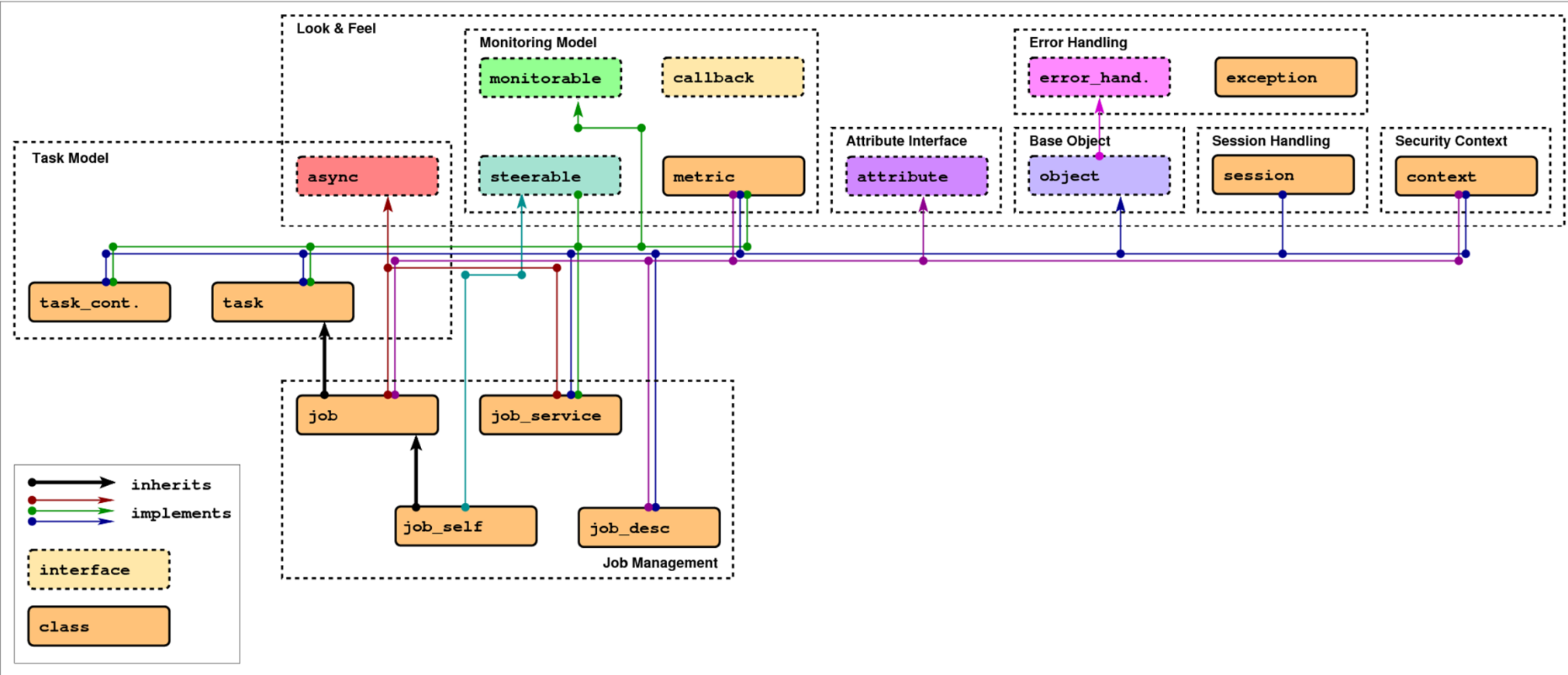
        File f = FileFactory.createFile(src);

        // normal sync version of the copy method
        f.copy(dst);

        // the three task versions of the same method
        Task t1 = f.copy(TaskMode.SYNC, dst); // in 'Done' or 'Failed' state
        Task t2 = f.copy(TaskMode.ASYNC, dst); // in 'Running' state
        Task t3 = f.copy(TaskMode.TASK, dst); // in 'New' state

        t3.run();

        t2.waitFor();
        t3.waitFor();
    }
}
```

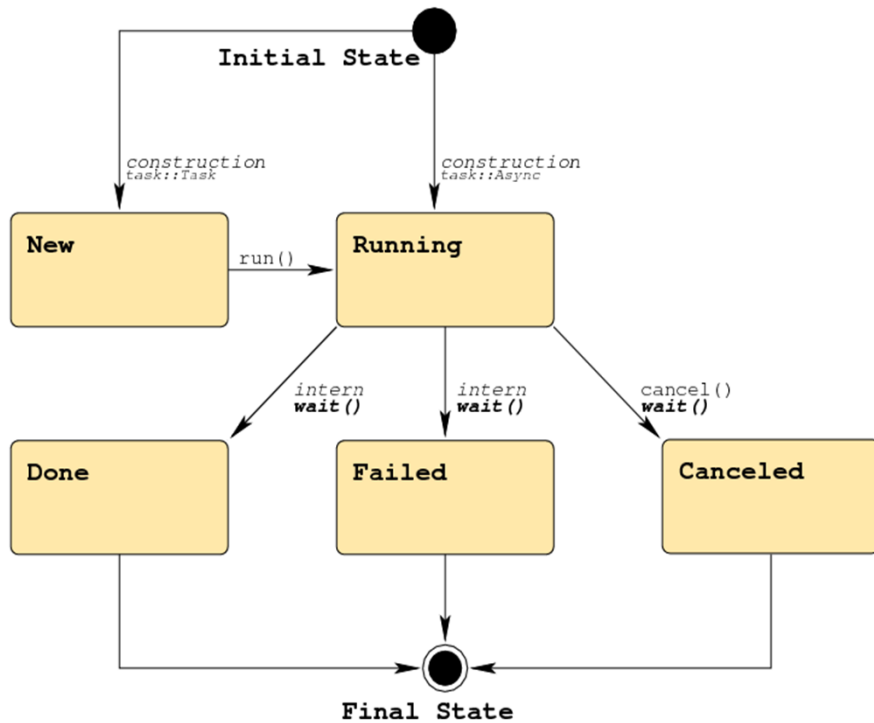


Jobs are submitted to run somewhere in the grid.

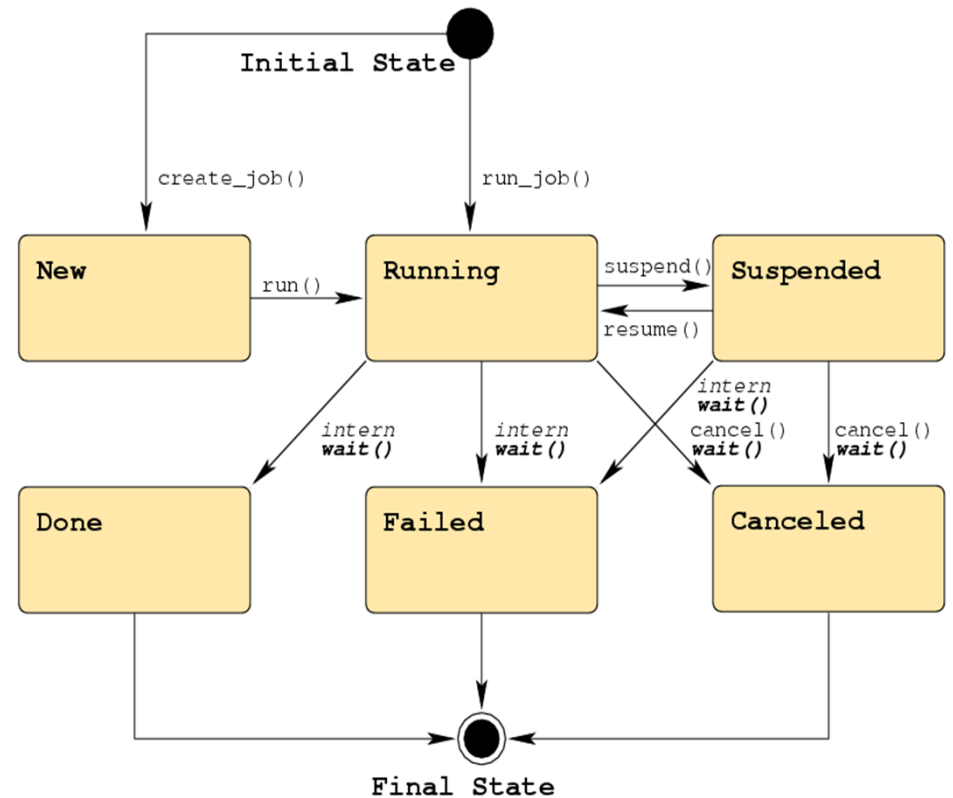


SAGA Task and Job States

Tasks:



Jobs:





Job Submission API

```
import org.ogf.saga.job.Job;
import org.ogf.saga.job.JobDescription;
import org.ogf.saga.job.JobFactory;
import org.ogf.saga.job.JobService;
import org.ogf.saga.task.State;
import org.ogf.saga.url.URL;
import org.ogf.saga.url.URLFactory;
```

```
public class JobSubmissionExample {
```

```
    void foo() throws SagaException { // submit a simple job and wait for
completion
```

```
        JobDescription d = JobFactory.createJobDescription();
        d.setAttribute(JobDescription.EXECUTABLE, "job.sh");
```

```
        URL u = URLFactory.createURL("any://remote.host.net");
        JobService js = JobFactory.createJobService(u);
```

```
        Job job = js.createJob(d);
        job.run();
```

```
        while(job.getState().equals(State.RUNNING)) {
            String id = job.getAttribute(Job.JOBID);
            System.out.println("Job running with ID: " + id);
            Thread.sleep(1000);
```

```
        }
```

```
    }
```

```
}
```



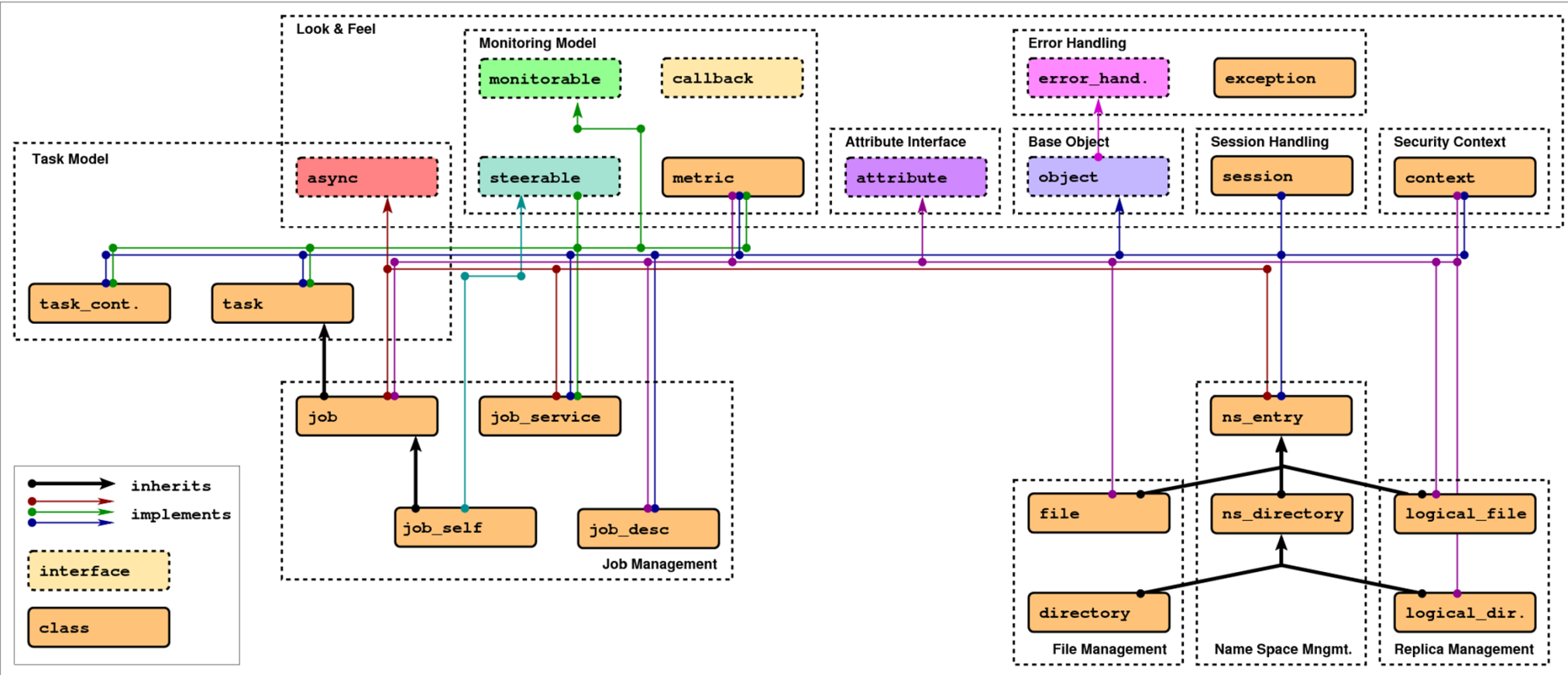
Job Submission API

`job_service` uses `job_description` to create a `job`

- `job_description` attributes are based on JSDL [OGF, GFD.56]
 - JSDL files can be imported/exported separately
- State model is based on OGSA BES [OGF, GFD.108]
- `job_self` represents the SAGA application



Files, Directories, Name Spaces



Both for physical and replicated (“*logical*”) files



File API Example

```
import org.ogf.saga.buffer.Buffer;
import org.ogf.saga.error.SagaException;
import org.ogf.saga.file.FileFactory;
import org.ogf.saga.job.JobDescription;
import org.ogf.saga.job.JobService;
import org.ogf.saga.url.URL;

import org.ogf.saga.buffer.BufferFactory;
import org.ogf.saga.file.File;
import org.ogf.saga.job.Job;
import org.ogf.saga.job.JobFactory;
import org.ogf.saga.task.State;
import org.ogf.saga.url.URLFactory;

public class FileAPIExample {
    void foo() throws SagaException {
        // read the first 10 bytes of a file if file size > 10 bytes

        URL u = URLFactory.createURL("file://localhost/etc/passwd");
        File f = FileFactory.createFile(u);
        long size = f.getSize();

        if (size > 10) {
            Buffer buf = BufferFactory.createBuffer(10);
            int readBytes = 0;
            while (readBytes < 10) {
                readBytes += f.read(buf, readBytes, 10 - readBytes);
            }

            String s = new String(buf.getData());
            System.out.println(s);
        }
    }
}
```



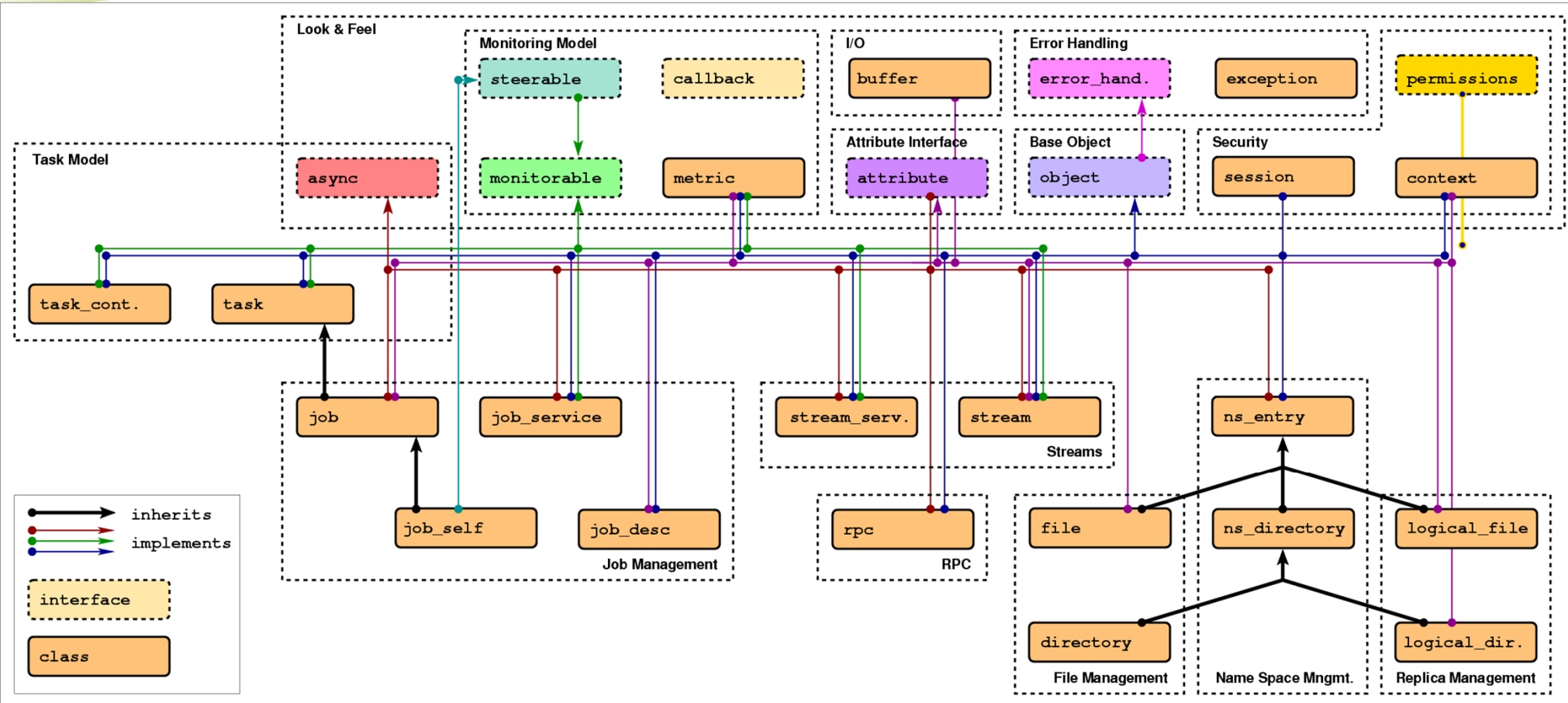
FileReadExample.java

```
import org.ogf.saga.buffer.Buffer;           import org.ogf.saga.buffer.BufferFactory;
import org.ogf.saga.error.SagaException;     import org.ogf.saga.file.File;
import org.ogf.saga.file.FileFactory;       import org.ogf.saga.url.URL;
import org.ogf.saga.url.URLFactory;

public class FileReadExample {
    public static void main(String[] argv) {
        if (argv.length < 1) {
            System.out.println("usage: java FileRead <URL>");
        } else {
            try {
                Buffer buf = BufferFactory.createBuffer(64);
                URL u = URLFactory.createURL(argv[0]);
                File f = FileFactory.createFile(u);
                int readBytes = 0;
                do {
                    readBytes = f.read(buf);
                    String s = new String(buf.getData(), 0, readBytes);
                    System.out.print(s);
                } while (readBytes > 0);
            } catch (SagaException e) {
                System.err.println(e);
            }
        }
    }
}
```



Streams, RPC, Permissions, I/O Buffers



Data streaming endpoints

Permissions for access rights

GridRPC

Buffers for I/O operations



- **XtreemOS comes with 3 SAGA implementations:**

- C++, collaboration with SAGA team @ LSU
- Java
- Python

- **Supported platforms:**

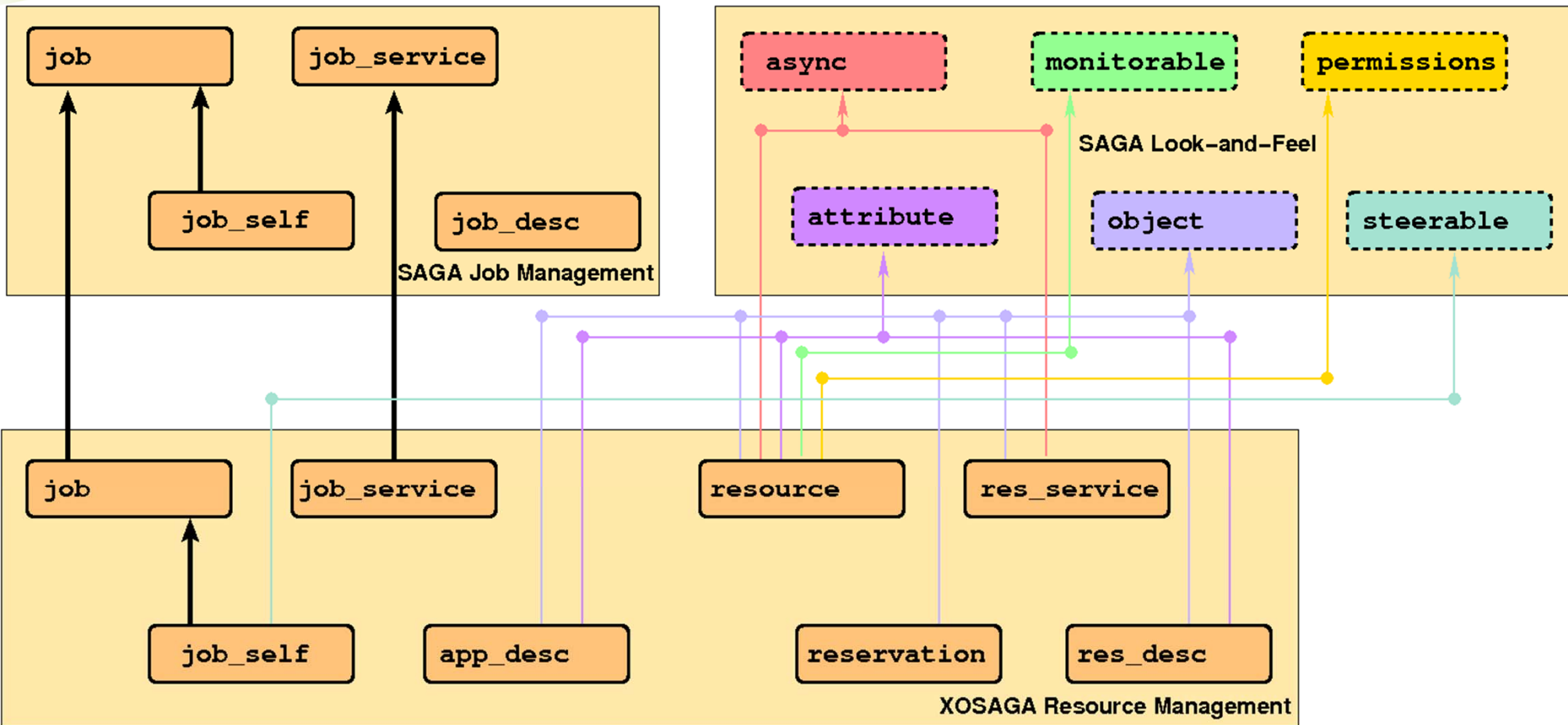
- XtreemOS, XtreemFS
- Globus (2.0 to 4.2), GRAM and GridFTP
- GridSAM (OMII-UK), gLite (not C++)
- localhost, ssh, XMLRPC



- **Provide support for resource reservations**
 - Split resource acquisition from application execution
- **Add Reservation class**
 - Submit jobs to active reservation
- **Split SAGA's job description (back) to application and resource descriptions**
- **Implemented as a separate XOSAGA package**



XOSAGA Resource Management Extension Package





Side Track: XOSAGA for IaaS (EC2)

- The XOSAGA package for resource reservations is *very* close to an IaaS interface
 - starting a VM means creating a resource and implicitly reserving it
- Ongoing work: extend XOSAGA to cover EC2 as compute resources
 - add `image`'s and `image_description`'s
 - add attributes to `resource_description`'s
 - image id, machine type, zone, ...
 - the call to `reserve` creates a VM, behind the scenes



- **XtreemOS comes with its XOSAGA API**
 - Inspired by POSIX
 - Inspired by SAGA
 - Providing XtreemOS-specific packages
- **Implementations in**
 - C++
 - Java
 - Python
- **Support for**
 - XtreemOS
 - Globus, gLite, GridSAM
 - localhost, ssh



Acknowledgements

- **The SAGA Team, at and with OGF:**
 - Andre Merzky, Shantenu Jha, Pascal Kleijer, Malcolm Illingworth, Hartmut Kaiser, Ole Weidner, Stephan Hirmer, Cerial Jacobs, Kees Verstoep
- **The European Commission via grants to**
 - The **CoreGRID** network of excellence
 - The **XtreemOS** project
 - Mathijs den Burger, Ana Oprescu, Emilian Miron, Manuel Franceschini, Tudor Zaharia, Pravin Shinde, Paul van Zoolingen
- **The Dutch VL-e project, OMII-UK, CCT LSU**



XtreamOS: a distributed operating system for large-scale dynamic Grid infrastructures

Key Aspects

- Easy of use & management
- Posix like & SAGA interface
- Advanced features for job execution & control
 - Reservation, co-allocation, signals, monitoring...
- Reliable application execution based on checkpointing
- Support for interactive jobs, workflow, MPI Grid applications



- **Download XtreemOS open source software (GPL/BSD)**
 - XtreemOS 2.1 version coming very soon
 - <http://www.xtreemos.eu/software>
 - RPM packages for Mandriva Linux
 - Ready to use Virtual Machine images for VirtualBox and KVM
- **On-line demonstrations**
 - <http://www.xtreemos.eu/demonstrations>
- **XtreemOS public deliverables and papers**
 - <http://www.xtreemos.eu>

A.3 Tutorial: “Grid and Cloud Computing with XtremOS” at Eurosys 2010

XtreemOS

*Enabling Linux
for the Grid*



Grid and Cloud computing With XtreemOS

Guillaume Pierre, VU University Amsterdam

**with contributions by Christine Morin, Thilo
Kielmann and other XtreemOS folks**

XtreemOS IP project

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



Information Society
Technologies





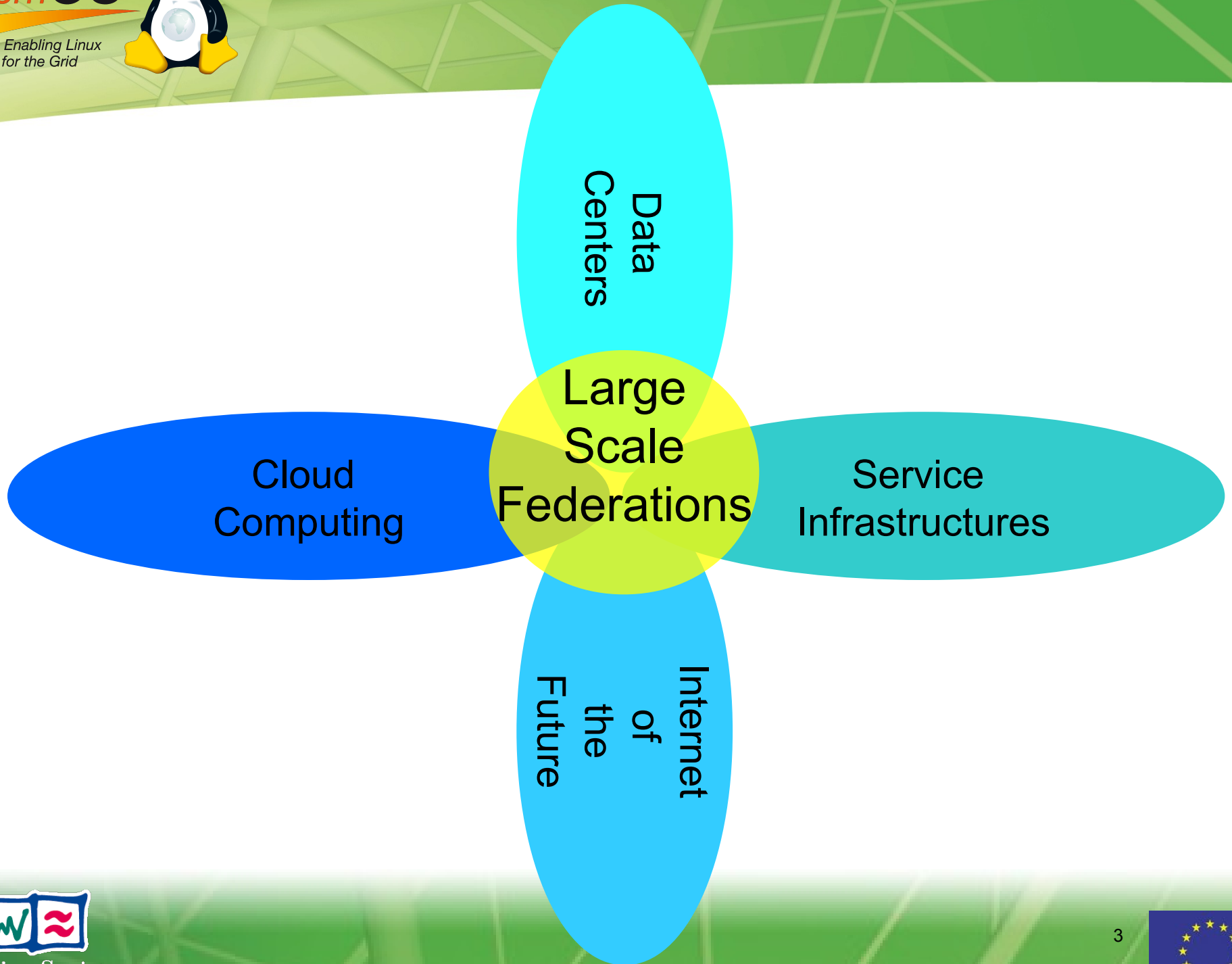
What is XtreemOS?

A Linux-based **Operating System**

with native **Virtual Organization** support

for **Large-scale Federations (like Grids or Clouds)**

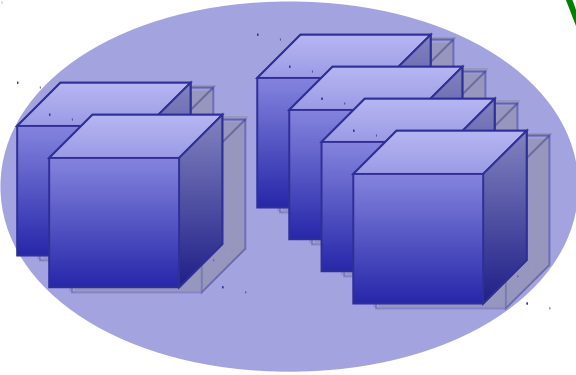




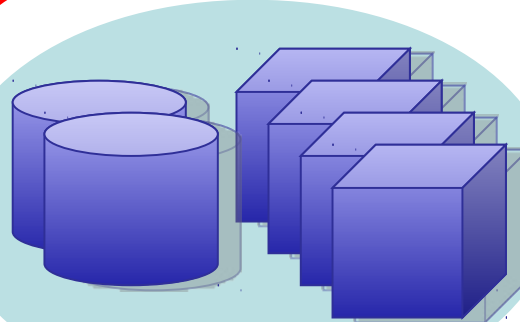


Virtual Organizations

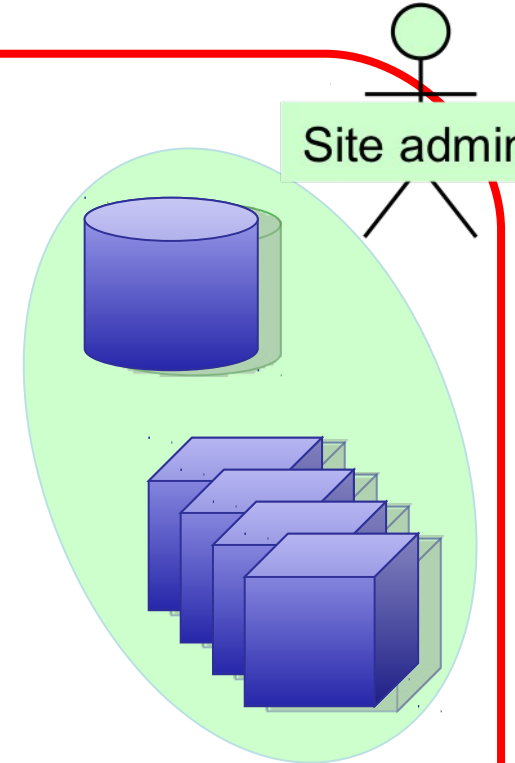
VO A



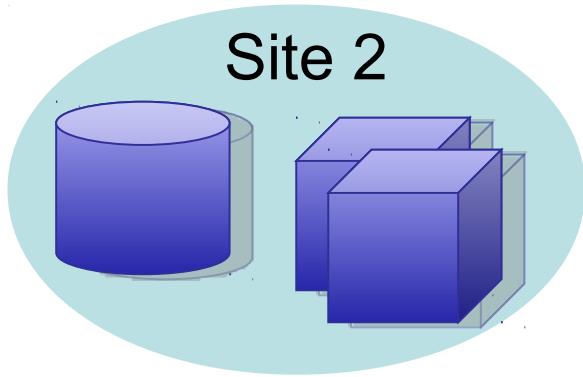
Organization 3



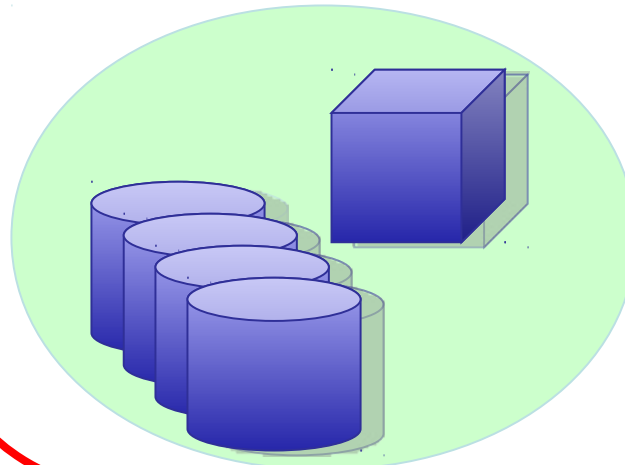
Site 1



Organization 2

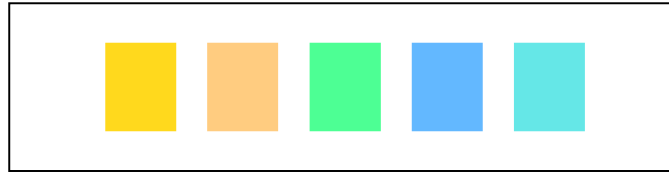
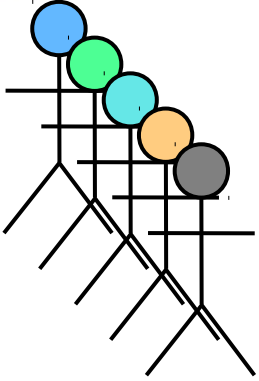


Organization 1



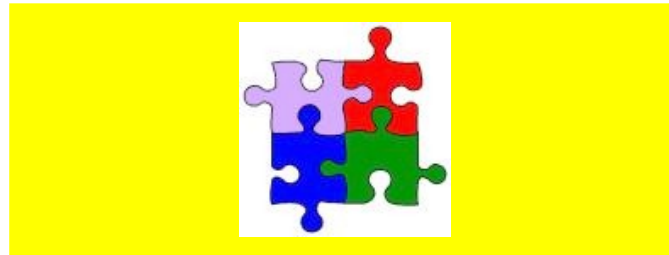


Traditional Operating System



Application

Set of integrated services
(process, file, memory
segment, sockets, user
account, access rights)



Operating System



Single computer

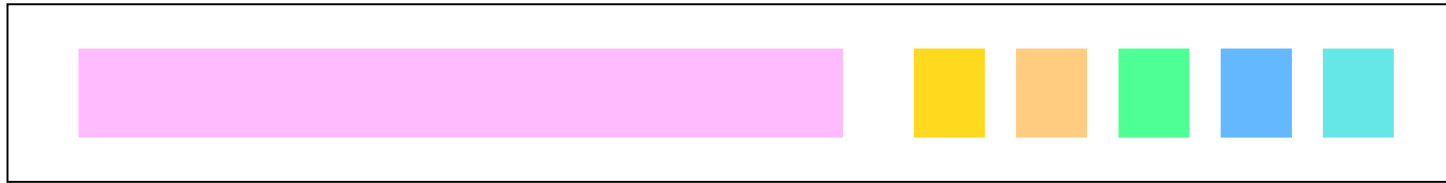


Hardware

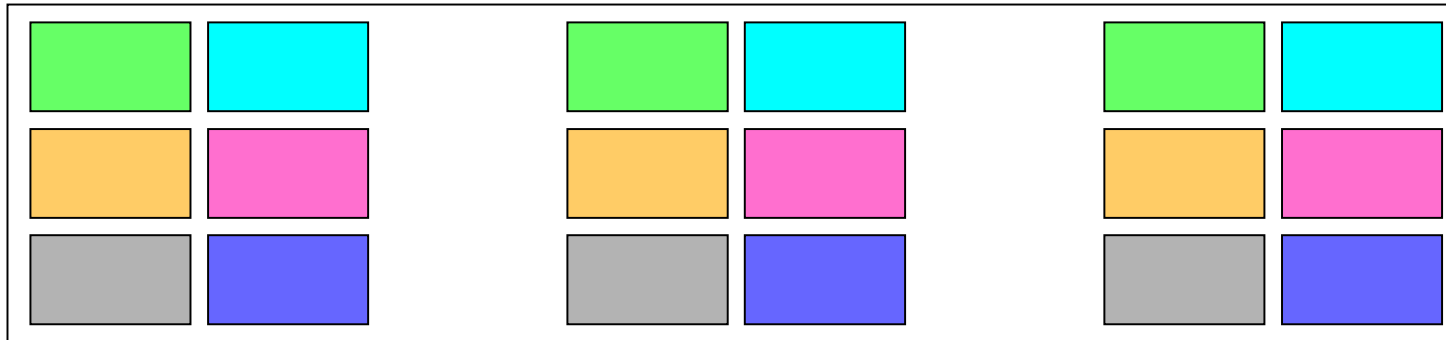




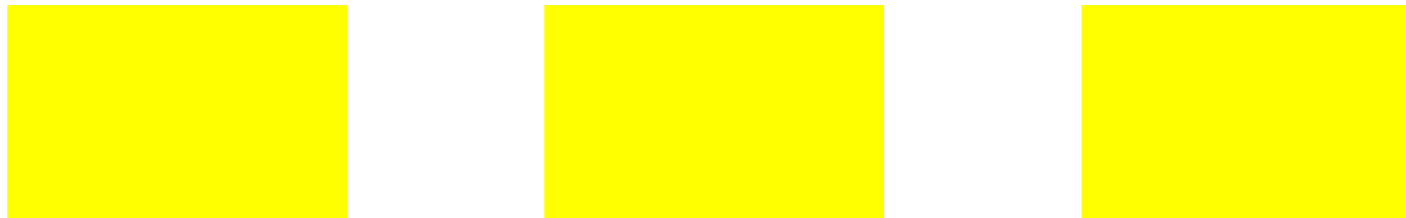
Middleware Approach



Grid
Middleware



OS

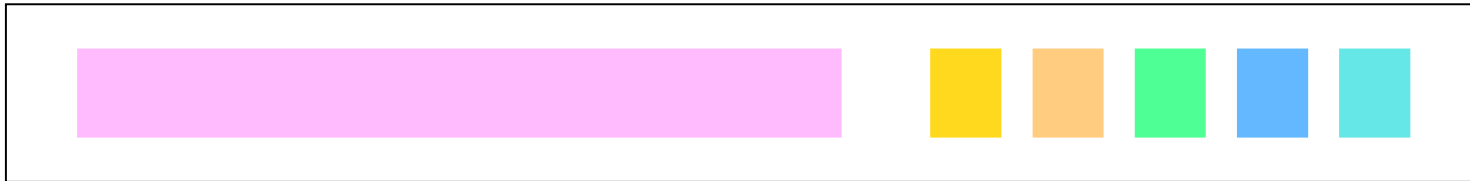


Hardware

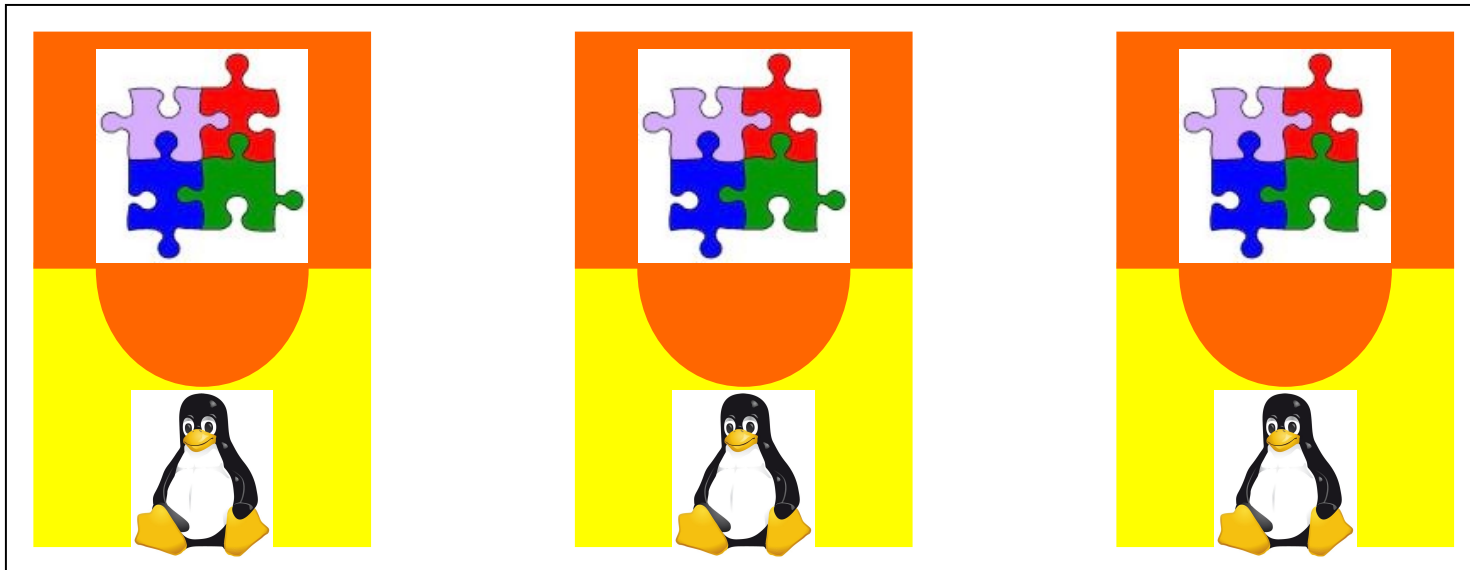




XtreemOS A Grid Operating System



XtreemOS





A **comprehensive** set of **cooperating**
system services
for a
wide-area dynamic distributed
infrastructure





- **Two fundamental properties: transparency & scalability**
 - Bring the grid to “standard” users
 - Scale with the number of entities and adapt to evolving system composition





- **Scale**
 - Thousands of nodes in thousands sites in a wide area infrastructure
 - Thousands of users

- **Consequences of scale**
 - Heterogeneity
 - Node hardware & software configuration
 - Network performance
 - Multiple administrative domains
 - High churn of nodes





- **Scalability with the number of entities & their geographical distribution**
 - Avoid contention points & save network bandwidth (performance)
 - Run over multiple administrative domains (security)

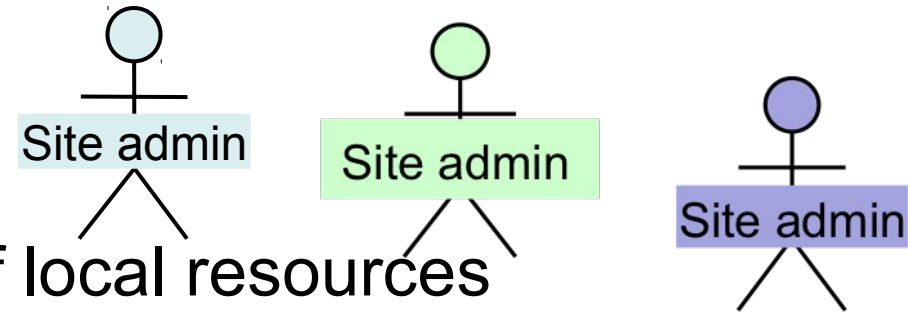
- **Adaptation to evolving system composition (dynamicity)**
 - Run with partial vision of the system
 - Self-managed services
 - Transparent service migration
 - Critical services highly available
 - No single point of failure





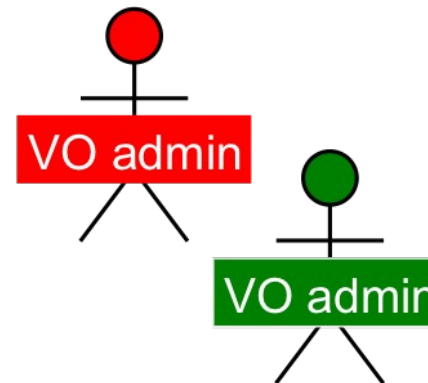
Site administrators

- Ease of management
- Autonomous management of local resources
- Should not be impacted by every single change in a VO



VO administrators

- Ease of management
- Flexibility in VO policies
- Accounting





- **Bring the Grid to standard Linux users**
 - **Feeling to work with a Linux machine (familiar interfaces)**
 - Standard way of launching applications
 - `ps` command to check status of own jobs
 - **No limit on the kind of applications supported**
 - Non Grid aware legacy applications
 - Interactive applications
 - **Grid-aware user sessions**
 - **Grid-aware shell** taking care of Grid related issues





- **VO can be built to isolate or share resources**
 - Parameter defined by VO administrator
- **Security without too much burden**
 - Single-Sign-On
 - Simple login as a Grid user in a VO





- **Conformance to standard API**

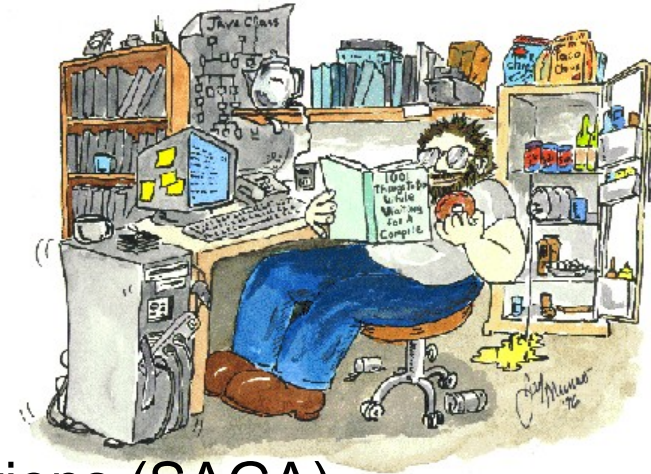
- Familiar Posix interface
- Grid application standards
- XOSAGA: The Simple API for Grid Applications (SAGA) with XtremOS extensions

- **Make Grid executions transparent**

- Hierarchy of jobs in the same way as Unix process hierarchy
- Same system calls: wait for a job, send signals to a job
- Processes in a job treated as threads in a Unix process

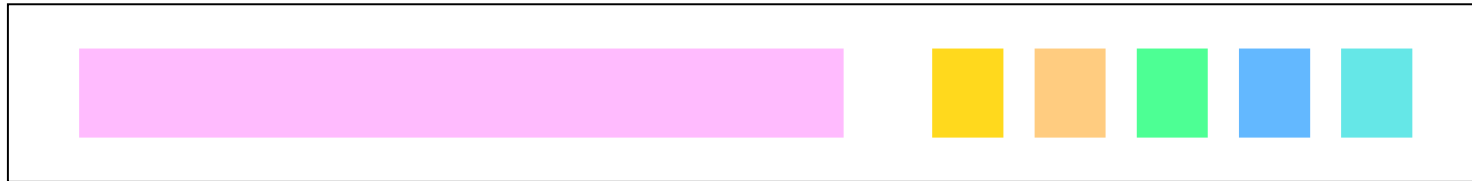
- **Files stored in XtremFS Grid file system**

- Posix interface and semantics to access files regardless of their location

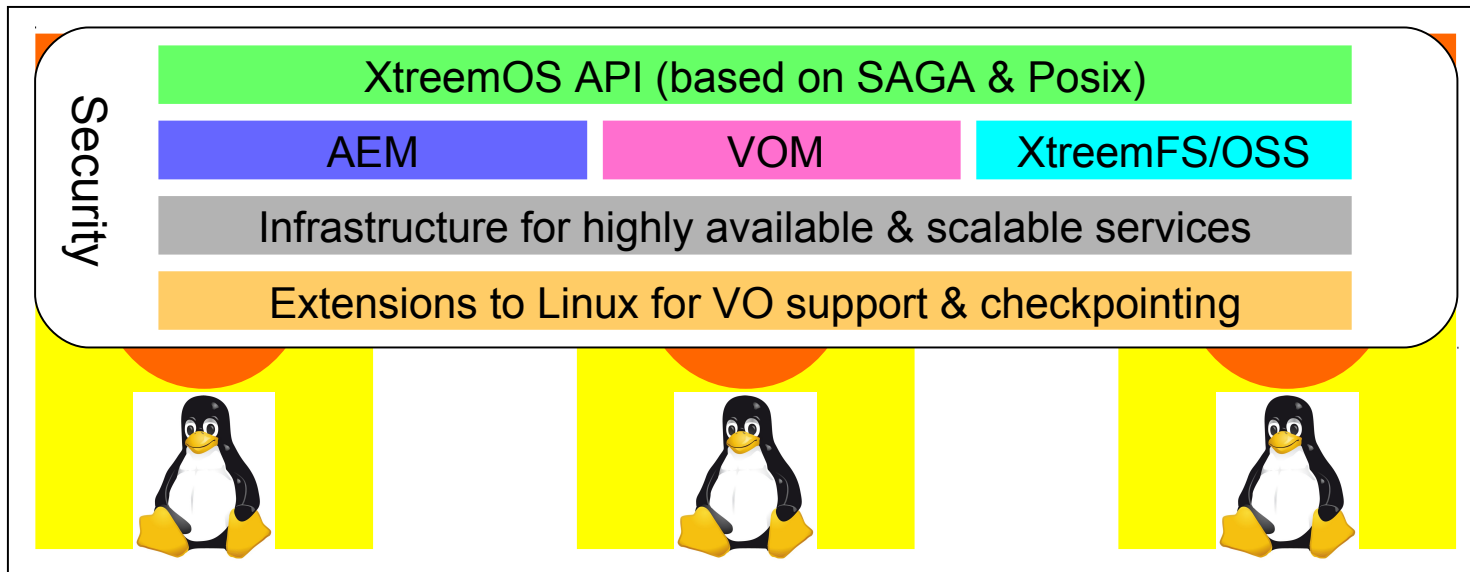




XtreemOS Services



XtreemOS





XtremOS Flavours



Stand-alone PC

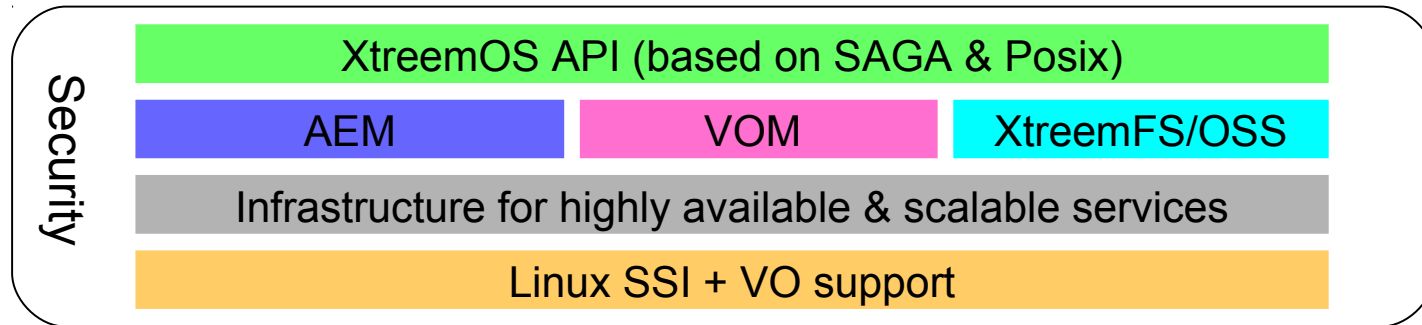


Cluster



Mobile device





■ Based on LinuxSSI foundation layer

- Linux based Single System Image cluster OS
 - Illusion of a powerful SMP machine running Linux
- Leverage Kerrighed full SSI
 - Posix compliant interface validated by successfully running the standard Linux Test Suite





■ Objectives

- Integration of XtreemOS services in mobile Linux OS enabling grid operation efficiently and transparently

■ Targets

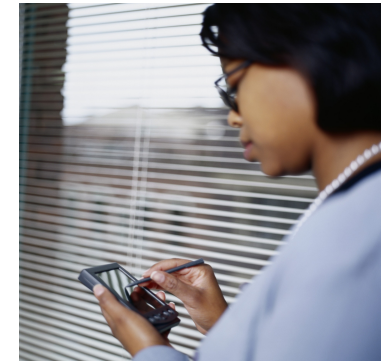
- Grid aware use cases

- Grid users on the move

- Grid-transparent use cases

- Services provided by a Grid infrastructure without the end users knowing it (Mobile Linux integrators)

■ Portability





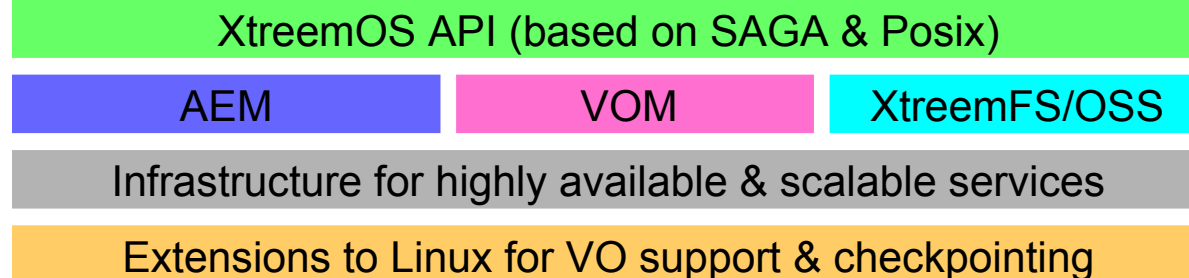
Objectives

- To allow secure interaction between users and resources
 - Authentication, authorization, accounting

Challenges

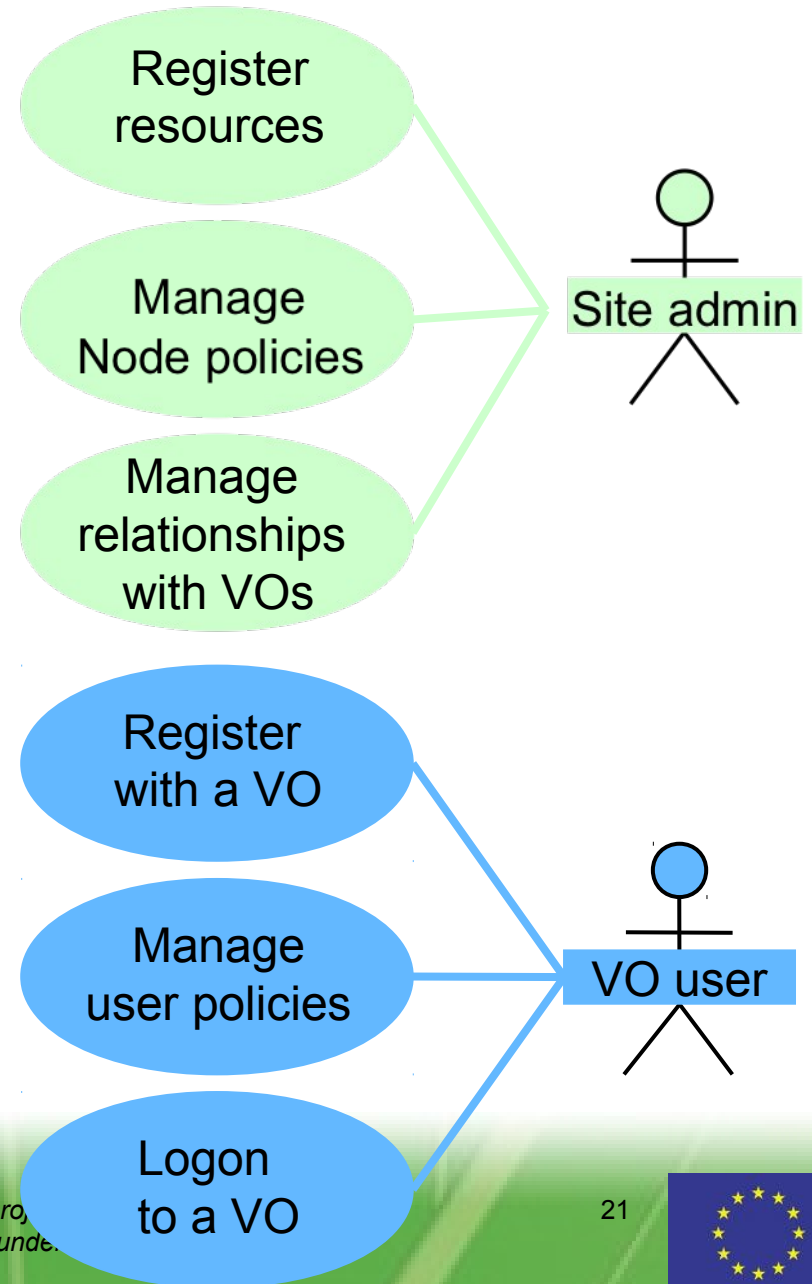
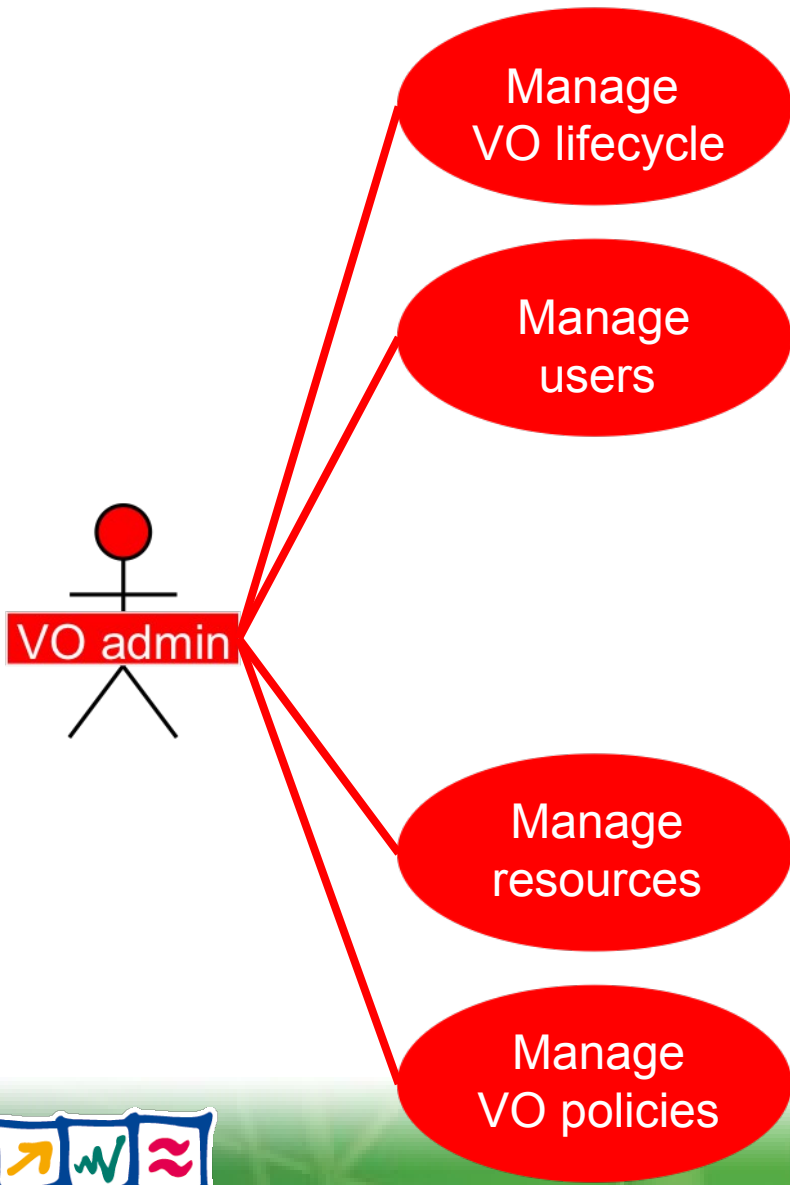
- Scalability of management of dynamic VOs
- Interoperability with diverse VO frameworks and security models
- Flexible administration of VOs
 - Flexibility of policy languages
 - Customizable isolation, access control and auditing
- Embedded support for VOs in the OS
- No compromise on efficiency, backward compatibility

Security





VO-related Interactions





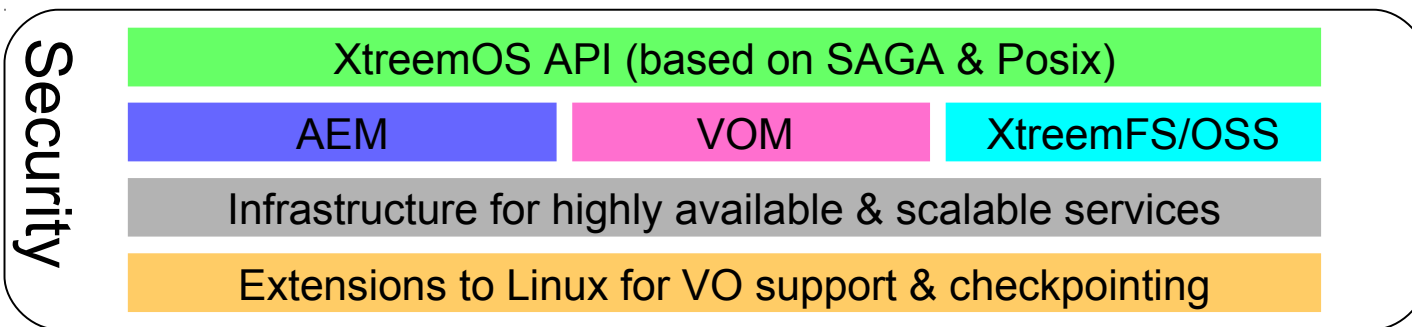
- **Maximum transparency**
 - Grid unaware applications & tools can be used without being modified or recompiled
- **Integration of Grid level authentication with node level authentication**
 - Creation of dynamic on-the-fly mappings for Grid users in a clean & scalable way
 - No centralized Grid wide data base
- **Grid user mappings invisible to local users**
- **VO's are easy to setup and manage**
 - No grid map file needed
 - Independent user and resource management
 - User management does not necessitate any resource reconfiguration





▪ Objectives

- Start, monitor, control applications
- Discover, select, allocate resources to applications





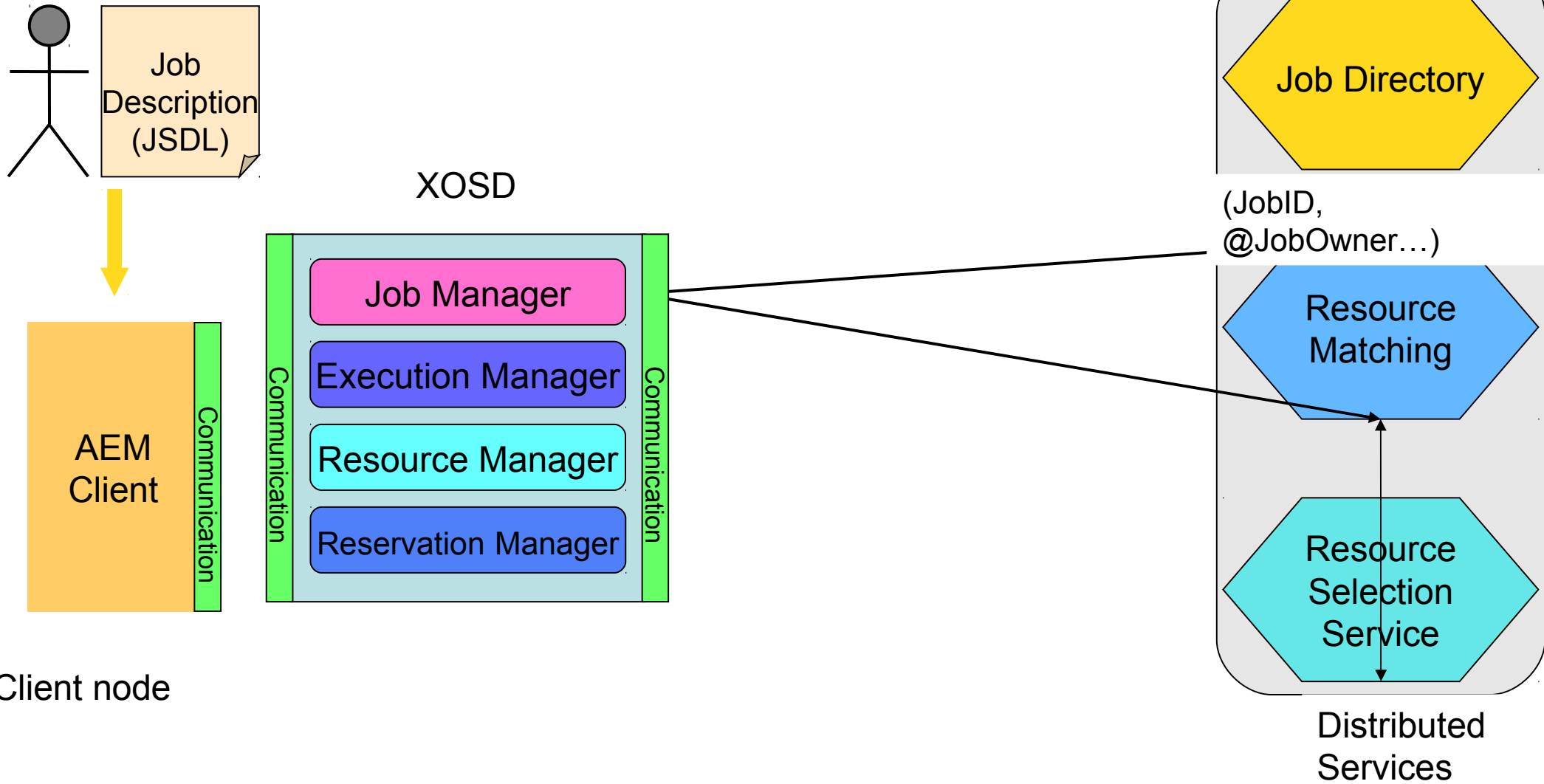
■ Features

- “Self-scheduling” jobs
 - No global job scheduler
- Resource discovery based on overlay networks
- Unix-like job control
- Monitoring & accounting
 - Accurate and flexible monitoring of job execution
- Resource reservation & co-allocation
- Interface for workflow engine
- Checkpointing service for grid jobs



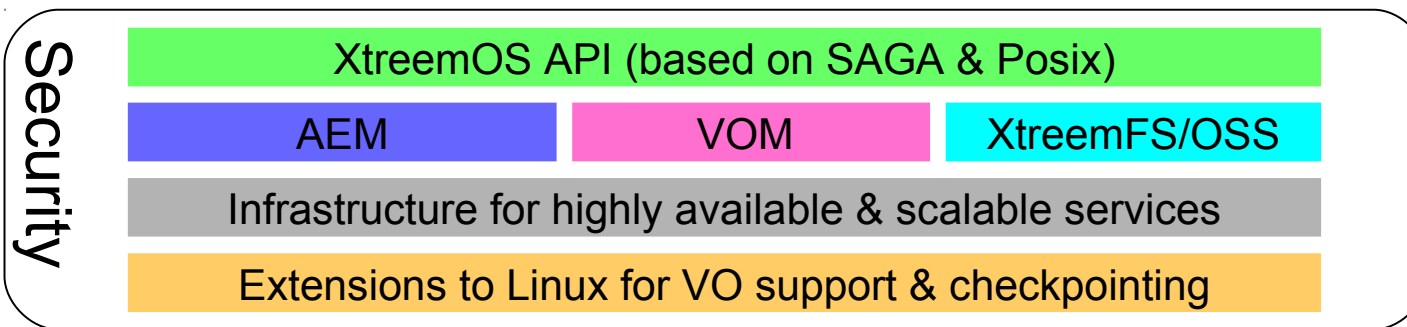


AEM Architecture





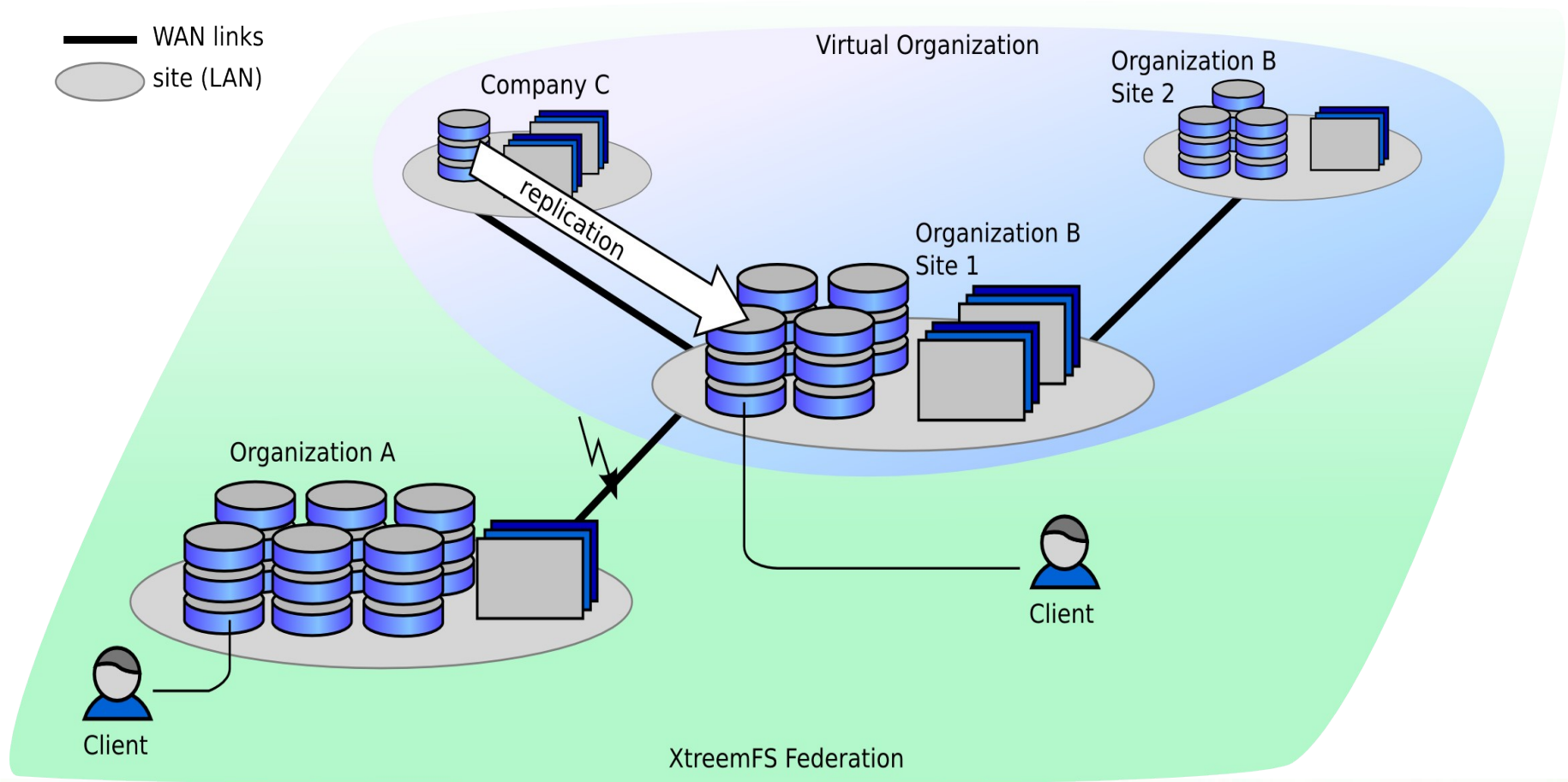
- **XtreemFS Grid file system**
 - Persistent data
- **Object Sharing System (OSS)**
 - Shared objects in memory





XtreemFS: A Grid File System

Federating storage in different administrative domains





▪ Objectives

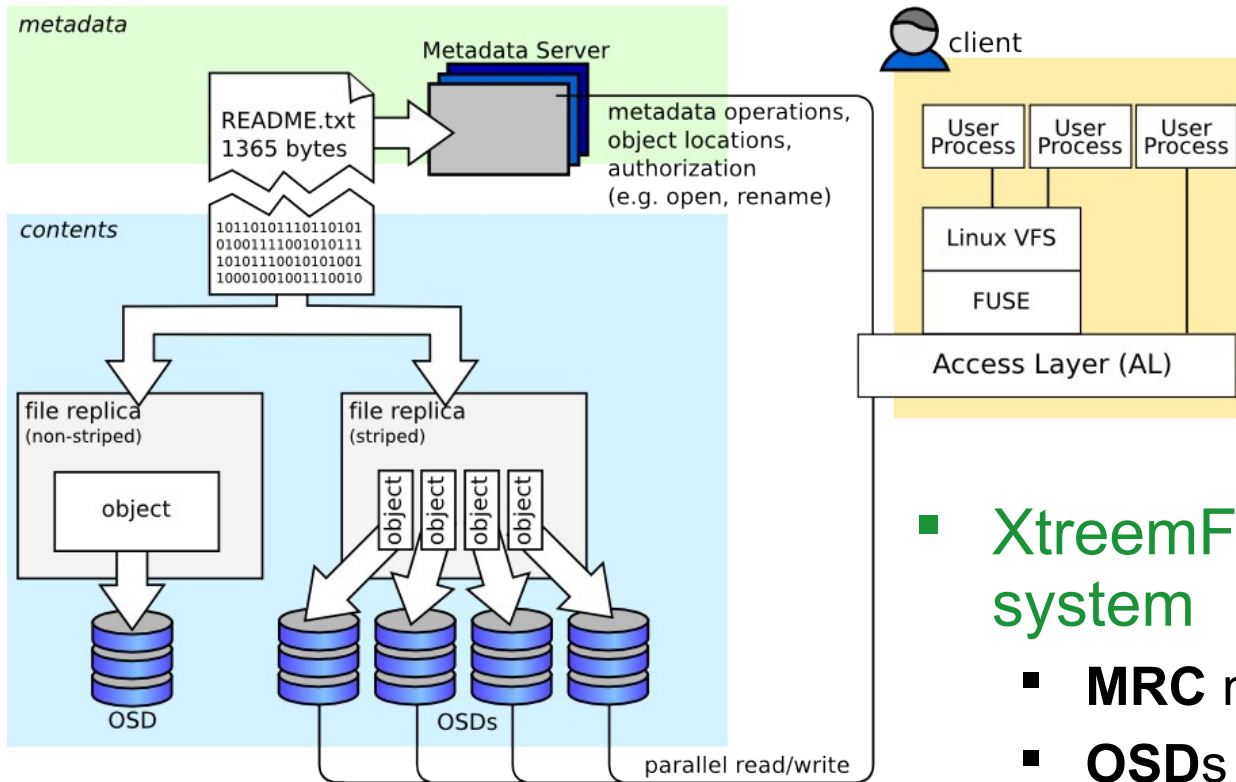
- Transparent access to data
- Providing to users a global view of their files through a Grid file system

▪ Challenges

- Efficient location-independent access to data through standard Posix interface in a Grid environment
 - Data storage in different administrative domains
 - Grid users from multiple VO's
- Autonomous data management with self-organized replication and distribution
- Consistent data sharing



XtreemFS: Architecture



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





- **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



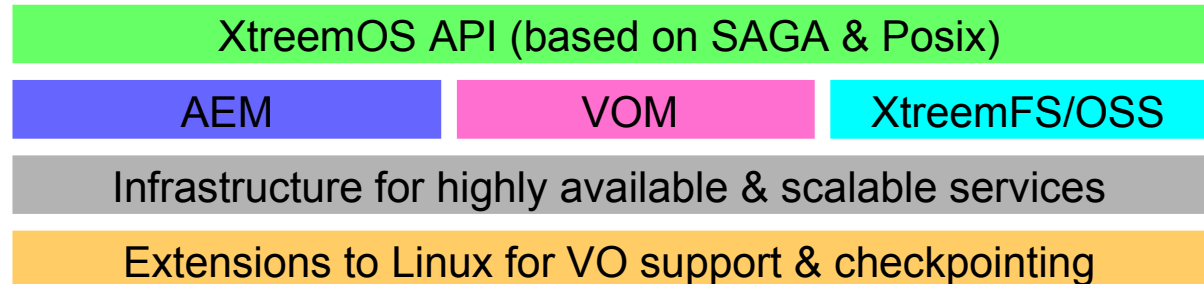
- **Replication of files**
 - primary/secondary with automatic failover
 - fully synchronous to lazy data replication
 - POSIX compatible by default
- **Striping (parallel read and write)**
- **RAID and end-to-end checksums**
- **Client-side caching and cache consistency**
- **Access pattern-based replica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





- **A Linux-based Grid Operating System**
 - flavours for PC's, clusters, and mobile devices
 - VO management integrated without kernel changes or central administration
- **XOSAGA and POSIX API's serve both Grid and Linux applications**
- **AEM, VOM, and XtreemFS provide global services**
- **Infrastructure for highly available services**

Security





- **Information**

- www.xtreemos.eu

- **XtreemOS 2.1**

- *ready for download*

- **Open source software repository**

- <http://gforge.inria.fr/projects/xtreemos/>

- **Contact**

- info@xtreemos.eu

- gpierre@cs.vu.nl



XtreemOS

*Enabling Linux
for the Grid*



Grid and Cloud computing with XtreemOS Part 2: Using XtreemOS

Corina Stratan, VU University Amsterdam
with contributions by Toni Cortes, Thilo Kielmann and other XtreemOS folks



Information Society
Technologies

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





- **Setting up a user account in XtreemOS**
- **Storing your data with XtreemFS**
- **Running applications with AEM**





- **To use XtreemOS you need to:**
 - get a user account on an XtreemOS machine
 - join a Virtual Organization (VO)
- **Joining a Virtual Organization - steps:**
 - send a request to the VO administrator
 - get a private key and a certificate
 - copy the key & certificate on the XtreemOs machine
- **This can be done with:**
 - XtreemOS commands, or
 - the **VoLifeCycle** web application (easier)



Account setup - step 1: create a VoLifeCycle account

default - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://node004:8080/volifecycle/view/uni_view.jsp?module=default&js=user_signup

Most Visited Getting Started Latest Headlines

Virtual Organizations in Action

Download the [Root certificate](#) (right-click and select 'Save as...')

XtreamOS
Enabling Linux
for the Grid

Home Manage Users Manage My VOs Manage My Resources

Welcome to VoLifeCycle

Login

Create an account

User Signup

Account Information

Login id:

Password:

Re-type Password:

Contact Information

First Name:

Last Name:

Organization:

Email:

Submit

Done

After this step:

- The VoLifeCycle admin will approve the new user account





Account setup - step 2: joining a VO

File Edit View History Bookmarks Tools Help

http://node004:8080/volifecycle/view/uni_view.jsp?module=user&js=user_join_vo

Most Visited Getting Started Latest Headlines

Virtual Organizations in Action

Download the [Root certificate](#) (right-click and select 'Save as...')

Home Manage My VOs Manage My Resources Welcome to VoLifeCycle , eurosys01 [logout]

Create a VO

Join a VO

My Pending Requests

Get an XOS-Cert

Generate new keypair

About me

Change Password

Logout

Join a VO

Search: [input] JoinVO LeaveVO Refresh

GVID	VO Name	VO Owner	Is Member	Description
<input type="checkbox"/> 2c0e8cb2-4453-461e-85b7-74874e76e7c2	exampleVO	xtreemos-vc	false	An example VO containing just t
<input type="checkbox"/> d81617b1-af89-4d9e-90fb-b9e65912cc71	VU_XOS20_admin	admin	false	
<input checked="" type="checkbox"/> 27c11e8-922a-4f9d-b301-579617a25a3c	EuroSys10	admin	false	VO for the EuroSys 2010 tutorial

Each VO has an unique ID (GVID)

Help Hints

Choose one VO from the table, then click JoinVO or LeaveVO button

Or you could [create your own VO](#) !

Done

After this step:

- The VoLifeCycle admin will approve the new user account



Account setup - step 3: getting a private key and a certificate

After this step:

- The user will download the key and cert, and copy them on an XtreemOS machine





Account setup - step 4: setting up the key & certificate

```
[euros01@node007 ~]$ ls -l .xos/truststore/
total 8
drwxrwxr-x 2 euros01 euros01 4096 2010-03-26 17:09 certs/
drwxrwxr-x 2 euros01 euros01 4096 2010-03-26 17:10 private/

[euros01@node007 ~]$ ls -l .xos/truststore/certs/
total 4
-rw-r--r-- 1 euros01 euros01 1303 2010-03-26 16:47 user.crt

[euros01@node007 ~]$ ls -l .xos/truststore/private/
total 4
-r----- 1 euros01 euros01 964 2010-03-26 16:47 user.key
[euros01@node007 ~]$
```

- The user key and certificate must be placed in the user's home account on an XtreemOS machine,
- in the `.xos/truststore` directory

Ready to use XtreemOS!





- A **user** can have the following identifiers:
 - **GUID: user ID**
 - **GVID: VO ID**
 - **GGID: group ID**
- ... and the following VO attributes:
 - **Group**
 - **Role**
 - **Capabilities**
- A **node** can have the following identifiers/attributes:
 - **GNID**
 - **GVID**
 - **Service**

These are global
(Grid-wide) identifiers





- **Single-Sign-On**
 - User session management services trusted by XtreemOS services
 - In charge of validating user credentials and user requests
- **Delegation**
 - User can run Grid requests from resource nodes (same capabilities as from their access node)
- **Mapping between different namespaces managed by local service xos-amsd**
 - GUID \square (Linux) UID
 - GGID \square (Linux) GID





```
[eurosos01@node007 ~]$ view-xos-cert
.xos/truststore/certs/user.crt
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      01:27:9a:d5:66:c3
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=VU, OU=cda, CN=node004das2.cs.vu.nl/cda
    Validity
      Not Before: Mar 26 14:12:09 2010 GMT
      Not After : Jan 20 14:22:09 2011 GMT
    Subject: CN=68ed46e2-df45-4bb7-9bb9-686b1b9b640e
    ...

XtreemOS VO Attributes:
  GlobalPrimaryVOName:
    27cf1fe8-922a-4f9d-b301-579617a25a3c
  GlobalSecondaryVONames:
    null
  GlobalUserID:
    68ed46e2-df45-4bb7-9bb9-686b1b9b640e
  GlobalPrimaryGroupName:
    null
  GlobalSecondaryGroupNames:
    null
```

- The **view-xos-cert** command can be used to display a human-readable format for any XtreemOS certificate





Looking at a machine certificate

```
[root@node007 certs]# view-xos-cert
.xos/truststore/certs/resource.crt
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      01:27:96:27:42:35
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=VU, OU=rca, CN=node004das2.cs.vu.nl/rca
    Validity
      Not Before: Mar 25 16:23:27 2010 GMT
      Not After : Jan 19 16:33:27 2011 GMT
    Subject: C=NL, L=Amsterdam, OU="XtreemOS Project
VU Certification Authority", O=VU, CN=Address =
[://13037.199.135:60000(130.37.199.135)]
    X509v3 extensions:
      ...

      DirName:/CN=VU CA/O=XtreemOS Project/OU=XtreemOS VU
Root Certification Authority
      serial:02

    X509v3 Subject Key Identifier:

7C:34:40:BE:10:4A:9F:41:B4:31:05:D4:C5:7C:06:8A:DD:80:96:
8A
```

**A machine has two types
of certificates:**

- Resource certificate
- Attribute certificate



- **Setting up a user account in XtreemOS**
- **Storing your data with XtreemFS**
- **Running applications with AEM**





- **The XtreemFS services:**
 - **DIR** – Directory Service
 - **MRC** – Metadata and Replica Catalog
 - **OSD** – Object Storage Device
- **The XtreemFS client**
 - Used to access an XtreemFS installation
- **XtreemFS volumes**
 - Containers for files and directories
 - Each volume has its own policy settings





- **Creating a volume: `mkfs.xtreemfs`**
 - Access policy: “authorize all”, POSIX, Volume
 - Striping policy: RAID0
- **Deleting a volume: `rmfs.xtreemfs`**
- **Listing all volumes: `lsfs.xtreemfs`**
- **Mounting a volume: `mount.xtreemfs`**
 - A volume must be mounted to a local directory before it can be used
- **Unmounting a volume: `umount.xtreemfs`**



Managing XtreemFS volumes: example

```
[euros01@node007 ~]$ mkfs.xtreemfs -a POSIX -p RAID0 -s 256 -w 1
node004.das2.cs.vu.nl:32636/euros01Vol
```

```
[euros01@node007 ~]$ lsfs.xtreemfs node004.das2.cs.vu.nl:32636
vol02 -> b93b75e8-0c49-4749-8e93-5d5632e9dc81
vol01 -> 9da8dabf-64ab-4f06-a2be-7f5ff1b038eb
vol-18f71ca3-fda7-497d-9a97-e14dbad5ad90 -> 952abbed-faed-4f64-8914-b1220e021932
euros01Vol -> 5170b340-dab4-4c3d-8d90-d0f076fa8bb5
vol-beda3fe0-44c2-4487-840d-4314314776b9 -> bf72e00b-53e5-4696-a882-b018d0bdd848
vol-c9c24d9b-b933-41ed-abdc-73a3262b1ad5 -> 8a8674f4-3a74-4c37-9eb7-69a4c797a55b
user-77530d3d-cfe2-4e1f-8ab6-f570afec9462 -> 030243a8-2fb4-4847-9336-6b6bef0c1af5
user-18f71ca3-fda7-497d-9a97-e14dbad5ad90 -> 8922f98c-f54a-444b-8d7d-e1a57ee7e7a8
```

```
[euros01@node007 ~]$ mkdir mount_point
[euros01@node007 ~]$ mount.xtreemfs node004.das2.cs.vu.nl/euros01Vol mount_point/
[euros01@node007 ~]$ cd mount_point/
[euros01@node007 mount_point]$ echo "hello world" > hello.txt
[euros01@node007 mount_point]$ ls
hello.txt
```

```
[euros01@node007 mount_point]$ cd ..
[euros01@node007 ~]$ umount.xtreemfs mount_point/
```





XtreemFS MRC @ default-MRC - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://core.xtreemos.cs.vu.nl:30636/

Most Visited Getting Started Latest Headlines

XTREEMFS **MRC default-MRC**

Version

XtreemFS	MRC 1.2.0 (Luscious Lebkuchen)
RPC Interface	2009121110
Database	0.3.0

Configuration

TCP & UDP port	32636
Directory Service	oncrpc://localhost:32638
Debug Level	6

Load

# clien connections	4
# pending client requests	0
Processing Stage queue length	

Requests

'open'	40
'readdir'	14
'xtreemfs_update_file_size'	2
'getattr'	283
'getxattr'	70

Volumes

selectable OSDs	default-OSD
striping policy	STRIPING_POLICY_RAID0, 128, 1
access policy	2
osd policy	1000,3000

Done

- All XtreemOS services can be monitored through a web interface
- The DIR service also has an e-mail notification system





- **Currently XtreemFS supports read-only replication**
- **To replicate a file:**
 - Set it to be read-only: `xtfs_repl --set_readonly /xtreemfs/file.txt`
 - Add new replicas: `xtfs_repl --add_auto --full --strategy random /xtreemfs/file.txt`
 - List all the replicas: `xtfs_repl -l /xtreemfs/file.txt`
- **Automatic on-close replication**
 - automatic creation of new replicas when files are closed after having been initially written



- **Filtering policies**
 - Default filter
 - FQDN-based filter
- **Grouping policies**
 - Data center map-based grouping
 - FQDN-based grouping
- **Sorting policies**
 - Shuffling
 - Data center map-based sorting
 - Vivaldi coordinates-based sorting
 - DNS-based sorting





- **Each Grid user has his/her own XtreemFS home volume**
 - The home volume can be created automatically
 - ... and can be accessed from any XtreemOS node
- **The home volume is mounted locally in a directory named as follows: /home / <GUID>**
- **A convenient place to store the input and output files of Grid jobs**





- **Command for logging into an XtreemOS node**
 - **Similar with the “regular” ssh, but can be used to log into any VO node**
 - **mounts automatically the user's home volume**

```
[eurosyst01@node007 ~]$ ssh-xos node007
Enter passphrase for key '/home/eurosyst01/.xos/truststore/private/user.key':
Last login: Tue Mar 30 10:32:42 2010 from node007.das2.cs.vu.nl
-bash-3.2$ pwd
/home/68ed46e2-df45-4bb7-9bb9-686b1b9b640e
-bash-3.2$
```





- **Setting up a user account in XtreemOS**
- **Storing your data with XtreemFS**
- **Running applications with AEM**





- **AEM: Application Execution Management**
- **Utilities for running jobs:**
 - Submitting jobs: `xsub`, `xsub.sh`
 - Advance reservations: `xreservation`
 - Job monitoring: `xps`
 - Checkpointing: `BLCR`, `LinuxSSI`
 - JobMA: graphical interface for mobile devices





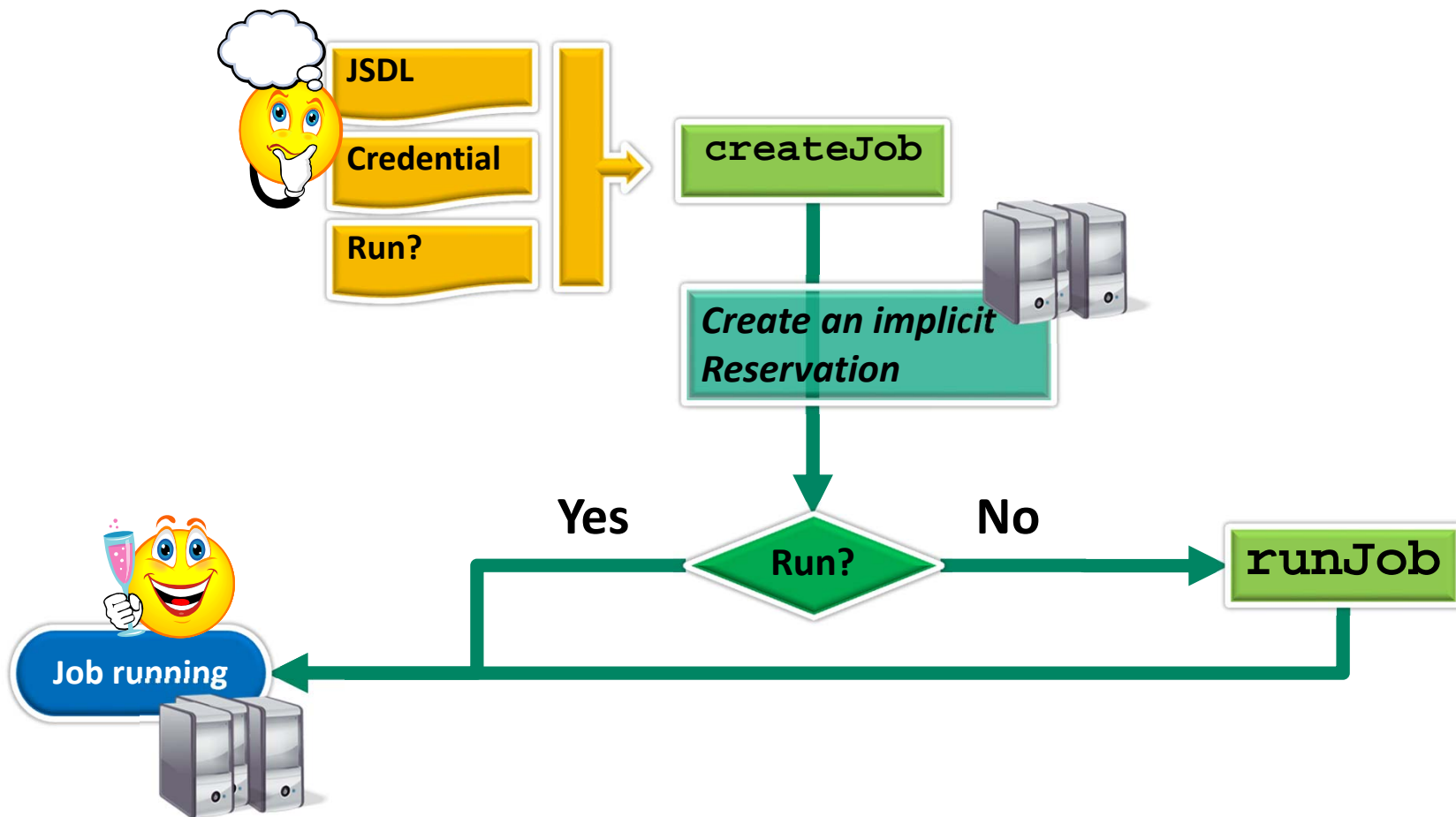
- **JSDL: standard language used in grids**
- **Purpose: to describe the job requirements in terms of resources**
- **... and the job specification (executable file, parameters etc.)**

```
<JobDefinition xmlns="http://schemas.ggf.org/jsdl/2005/11/jsdl">
  <JobDescription>
    <JobIdentification>
      <Description>Execution of cal</Description>
      <JobProject>XtreamOS_Test</JobProject>
    </JobIdentification>
    <Application>
      <POSIXApplication xmlns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix">
        <Executable>/usr/bin/cal</Executable>
        <Output>/home/68ed46e2-df45-4bb7-9bb9-686b1b9b640e/out_cal.txt</Output>
        <Error>/home/68ed46e2-df45-4bb7-9bb9-686b1b9b640e/err_cal.txt</Error>
      </POSIXApplication>
    </Application>
    <Resources>
      <TotalResourceCount>
        <Exact>1</Exact>
      </TotalResourceCount>
    </Resources>
  </JobDescription>
</JobDefinition>
```





Executing a job





- `xsub`: for submitting and executing a job described by a JSDL file

```
[eurosyst01@node007 ~]$ xreservation -qf  
Address = [://130.37.199.135:60000]: * : *  
  
[eurosyst01@node007 ~]$ xsub -f cal.jsdl  
Job submitted successfully: 2e9af6d5-6c24-45b4-  
85d9-91d3323c0696
```

... but it can be easier:

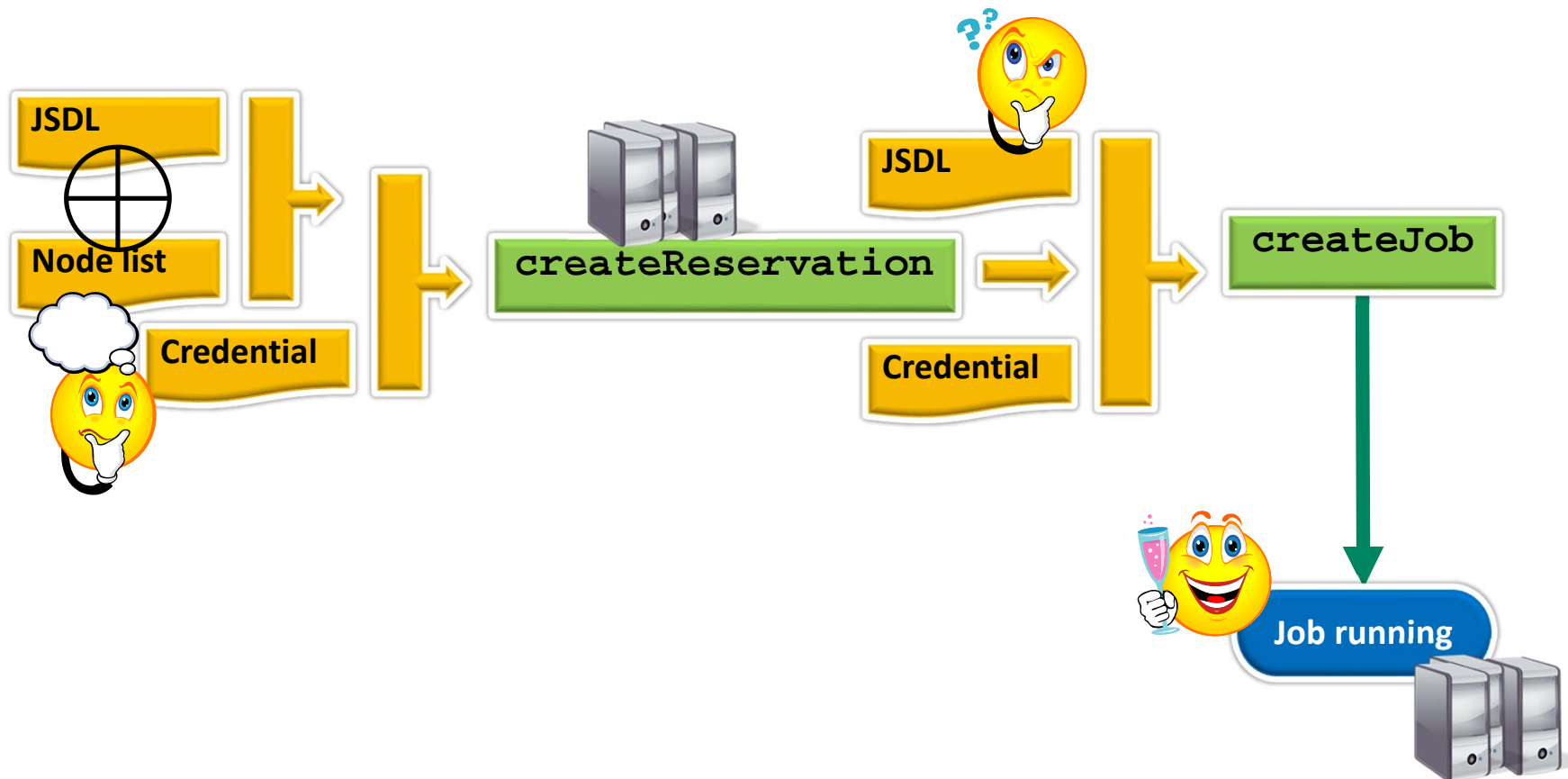
- `xsub.sh`: a script that automatically generates a JSDL file and submits it

```
xsub.sh executable [ parameters][-in input ][-out output ][-err error ]
```





Explicit reservations





How to make explicit reservations

- Use the `xreservation` command
- You can specify the number of needed resources, the starting time, the duration etc.
- A JSDL file can be given as input to provide criteria for selecting resources
- `xreservation -qf` shows the resources currently available for executing jobs
 - A VO member can only execute jobs on resources that belong to the VO

```
[euros01@node007 ~]$ xreservation -qf  
Address = [://130.37.199.135:60000]: * : *
```

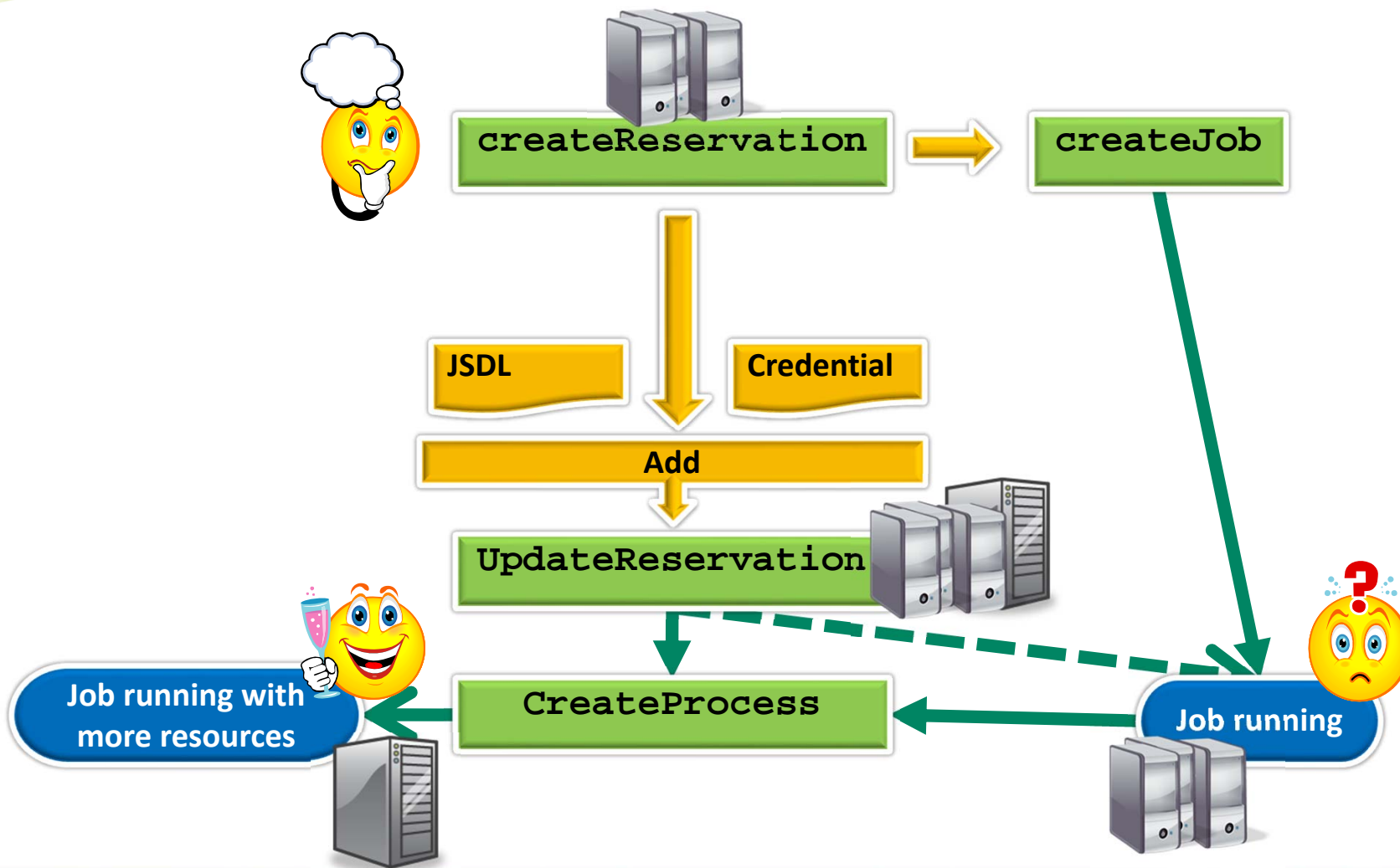
```
[xosuser@node007 ~]$ xreservation -qf  
Address = [://130.37.199.134:60000]: * : *  
Address = [://130.37.199.135:60000]: * : *
```

This node belongs to 2 VOs



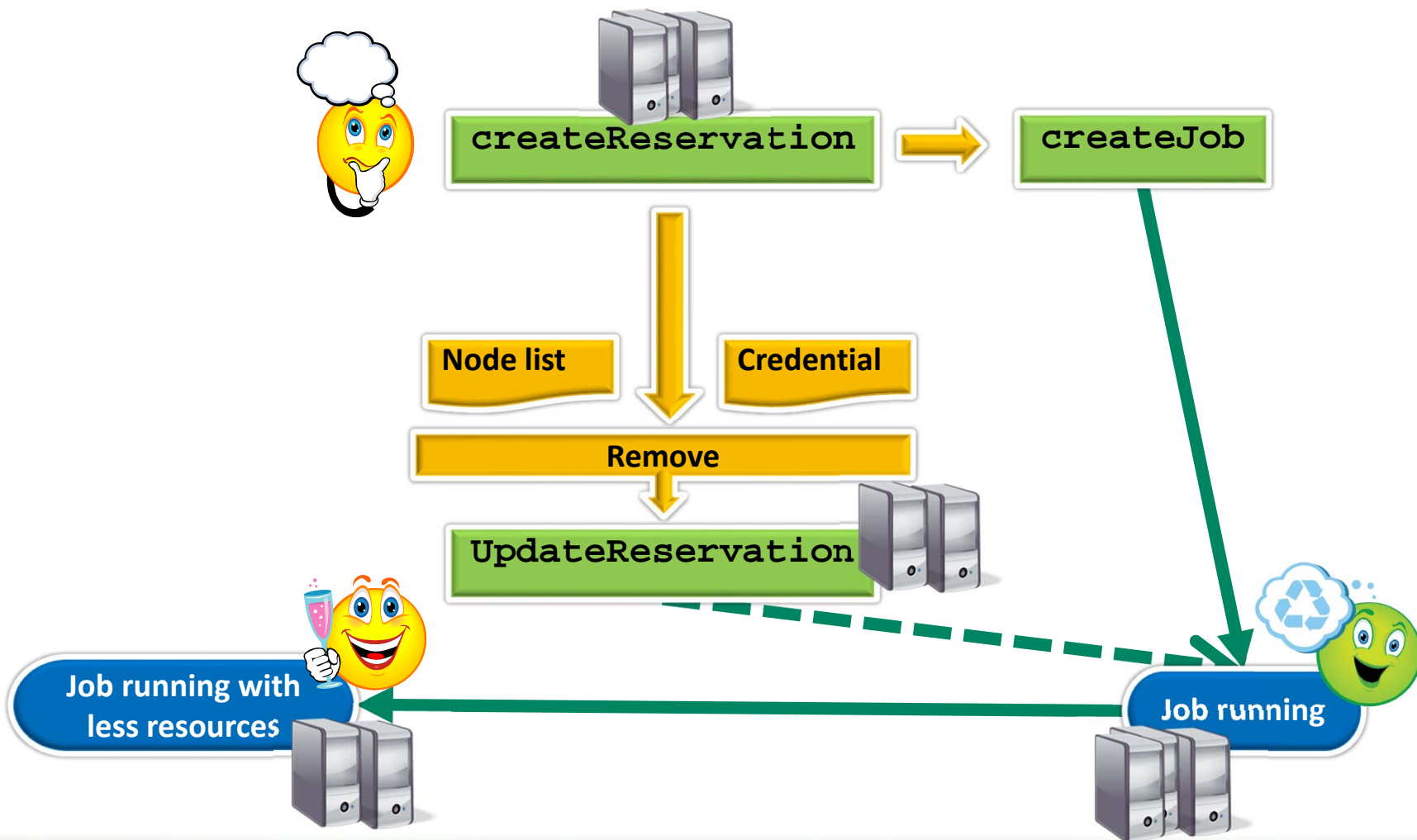


Dynamic reservations (1)





Dynamic reservations (2)





- **xps: interface for the monitoring infrastructure of XtreemOS**
- **Displays information about:**
 - all user's jobs
 - a job with a given ID
 - a group of jobs with a tagged dependence

```
[eurosyst01@node007 ~]$ xps -a
2e9af6d5-6c24-45b4-85d9-91d3323c0696 @ 1269938007356 :
    jobID = 2e9af6d5-6c24-45b4-85d9-91d3323c0696
    userDN = 68ed46e2-df45-4bb7-9bb9-686b1b9b640e
    VO = 27cf1fe8-922a-4f9d-b301-579617a25a3c
    jobStatus = Done
    submitTime = Tue Mar 30 10:33:20 CEST 2010
```





- **XATI (XtreemOS Application Toolkit Interface)**
 - Support for running multi-process jobs across multiple nodes
- **Support for MPI jobs: beta status**
- **Running interactive jobs**
 - Exporting the user's desktop environment
 - Opening interactive sessions inside a job context





- **SAGA: Simple API for Grid Applications**
 - Provides common grid functionality
 - Uniform across middleware platforms
 - Community effort (OGF, Berkeley, VU, LSU, NEC)
- **XOSAGA: SAGA + XtreemOS extensions:**
 - VO management (XtreemOS credentials)
 - Application Execution Management (reservations)
 - XtreemFS (URL schema)
 - Distributed Servers (hand-over sockets)
 - Data sharing (OSS, Scalaris publish/subscribe)
- **Languages: C++, Java, Python**

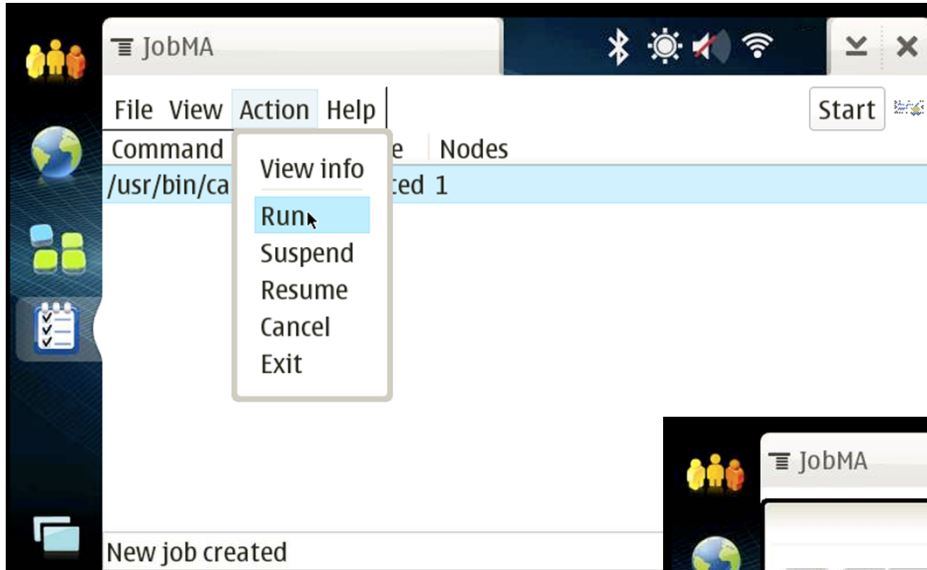




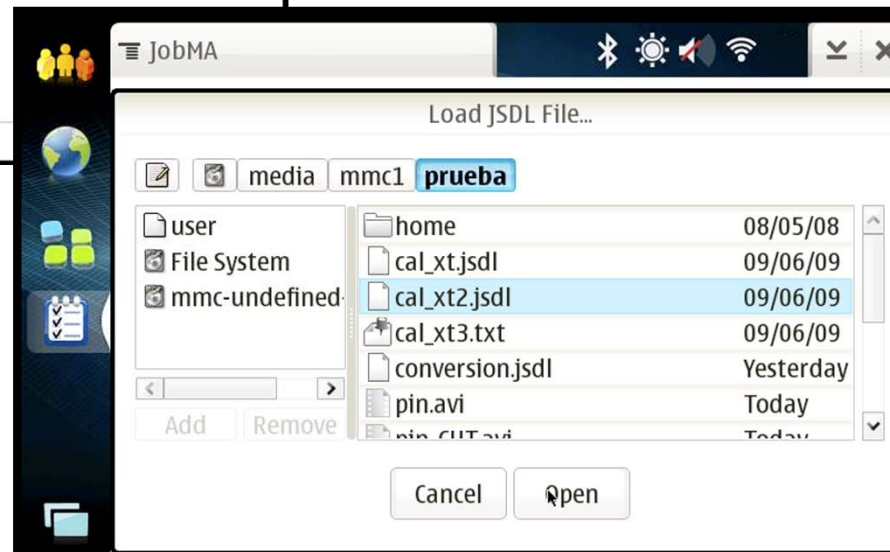
XOSAGA example: copying a file in XtreemFS

```
#include <saga.hpp>
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    if (argc != 3) {
        cout << "usage: " << argv[0] << " srcFile dstFile" << endl;
        return 1;
    }
    saga::url srcUrl(argv[1]);
    saga::url dstUrl(argv[2]);
    try {
        saga::session s;
        saga::filesystem::file f(s, srcUrl, saga::filesystem::Read);
        cout << "Copying " << srcUrl << " to " << dstUrl << endl;
        f.copy(dstUrl);
    } catch (saga::exception const & e) {
        cout << "caught SAGA exception: " << e.get_message() << endl;
    }
}
```





- **Graphical job manager for mobile devices**
- **Provides an interface to the main AEM features**
- **Based on GTK**
- **Jobs can be defined interactively or by loading a JSDL file**





- *Users mailing list:*
 - *<http://lists.gforge.inria.fr/cgi-bin/mailman/listinfo/xtreemos-users>*
- *Users wiki:*
 - *<http://xtreemos-user.wiki.irisa.fr>*
- *IRC:*
 - *[#xtreemos-dev](irc://freenode.net)*



XtreemOS

Enabling Linux
for the Grid



DEMO SESSION



Information Society
Technologies

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



XtreemOS

*Enabling Linux
for the Grid*



Grid and Cloud computing with XtreemOS Part 3 - Basic of System Administration

Massimo Coppola ISTI-CNR, Italy
with contributions by Christine Morin and countless
collaborators within XtreemOS
Eurosys 2010, Paris



Information Society
Technologies

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





- **VO management lifecycle**
- **Security background : Public Key Infrastructures**
- **Scalable Virtual Organizations in XtreemOS**
- **XtreemOS VO creation and management GUI**
- **Monitoring resources**





- **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**





What are the administrator's tasks?

- **Basic set-up of virtual organizations consists in**
 - Establishing trust among resources and users
 - Providing the resources
 - Administrating the resources via policies
- **We already saw that**
 - Users / resources have global ids
 - There's no need to set up any id mapping
 - This is done by XtreemOS via LINUX functionalities
- **So, how is this done?**
- **What's left to the administrator?**





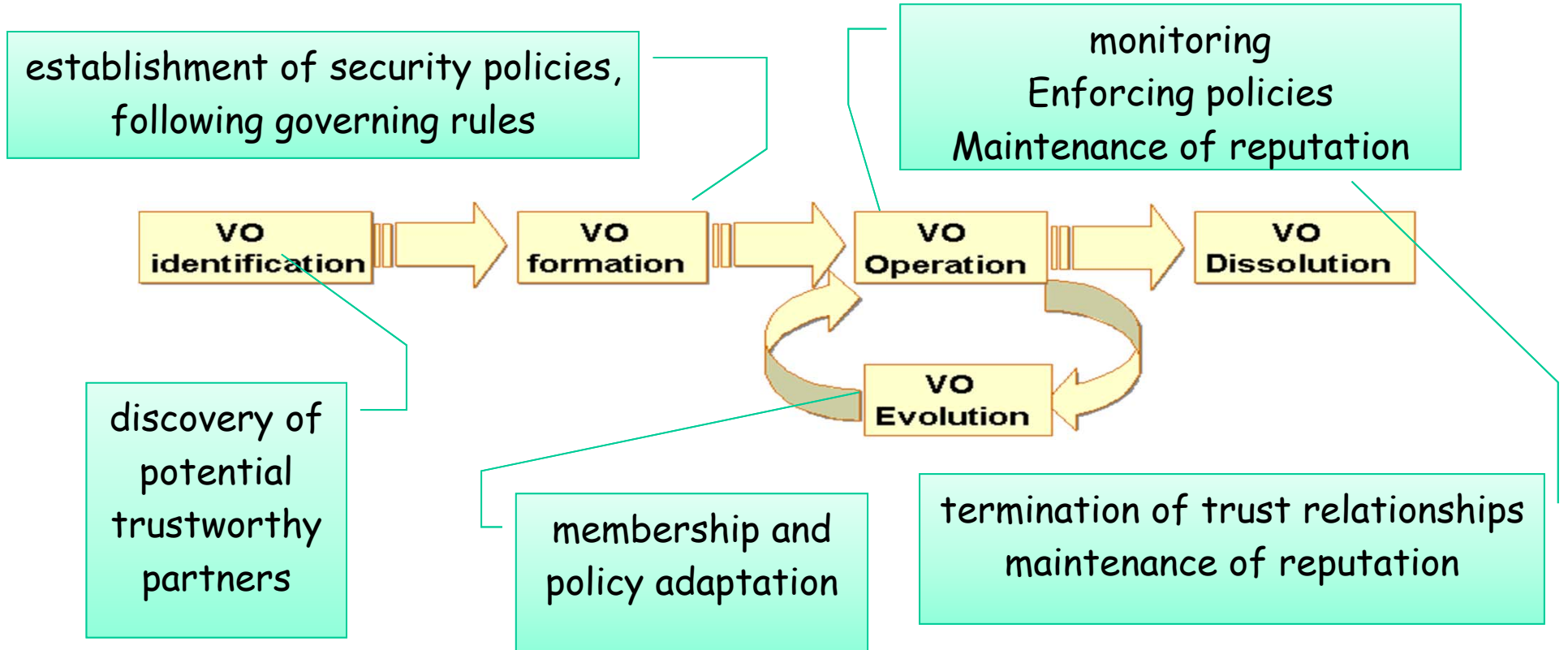
Basic Security Concerns over Grids and Clouds

- **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!





VO Lifecycle





- **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-repudiation:** 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

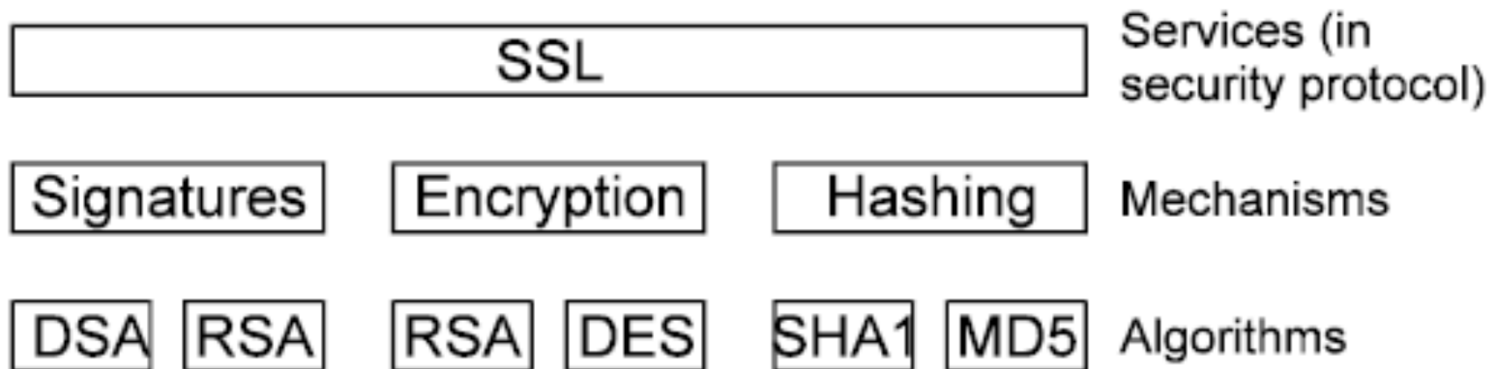


- **Authentication Authorization Auditing**
- **Three basic building blocks are used:**
 - **Encryption** is used to provide confidentiality, can also provide authentication and integrity protection
 - **Digital signatures** are used to provide authentication, integrity protection, and non-repudiation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





- A typical security protocol provides one or more services

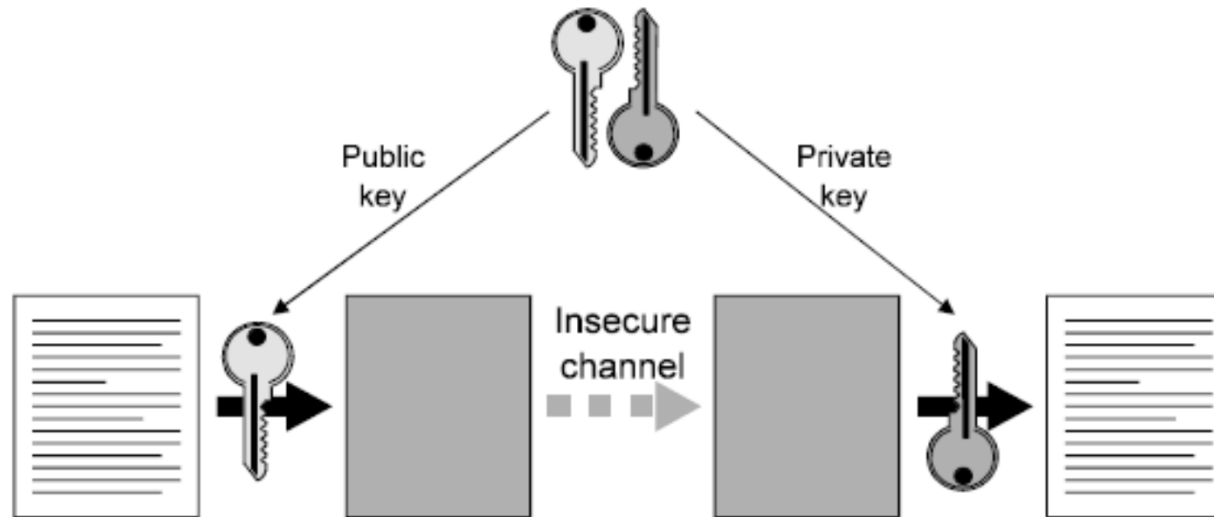


- Services are built from mechanisms
- Mechanisms are implemented using algorithms
- Algorithms and mechanism are carefully developed
 - Huge amount of work in verification and debugging





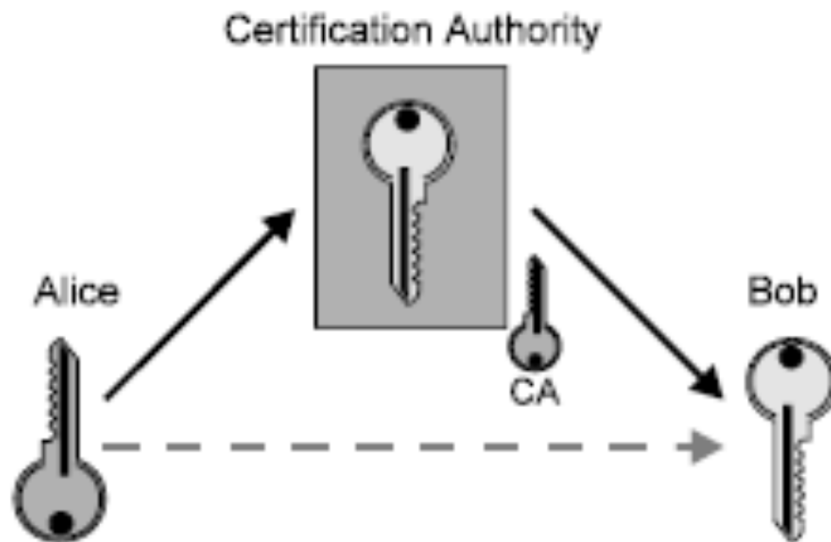
- Users possess **public/private key pairs**



- **Anyone can encrypt with the public key, only one person can decrypt with the private key**
 - Communication can be made secure
 - The problem is how to authenticate the keys



- **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



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)





- **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.**





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





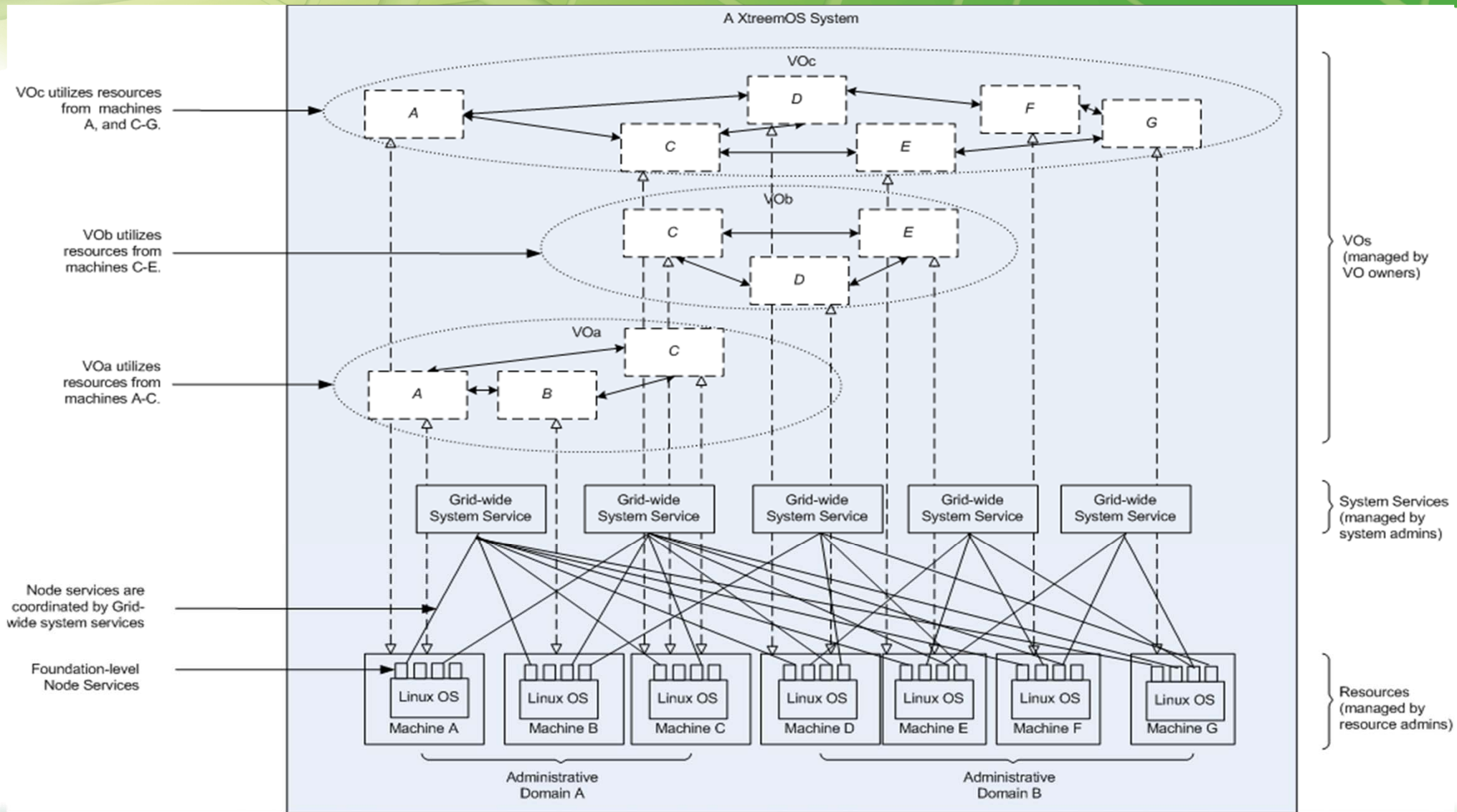
- **Domain administrators delegate user administration to Virtual Breeding Environments (VBE)**
 - PKI infrastructure
- **Users create VOs**
- **Domain administrators provide resources to VOs**
- **Resource owners always in control**
 - On site policies local to each machine





- **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





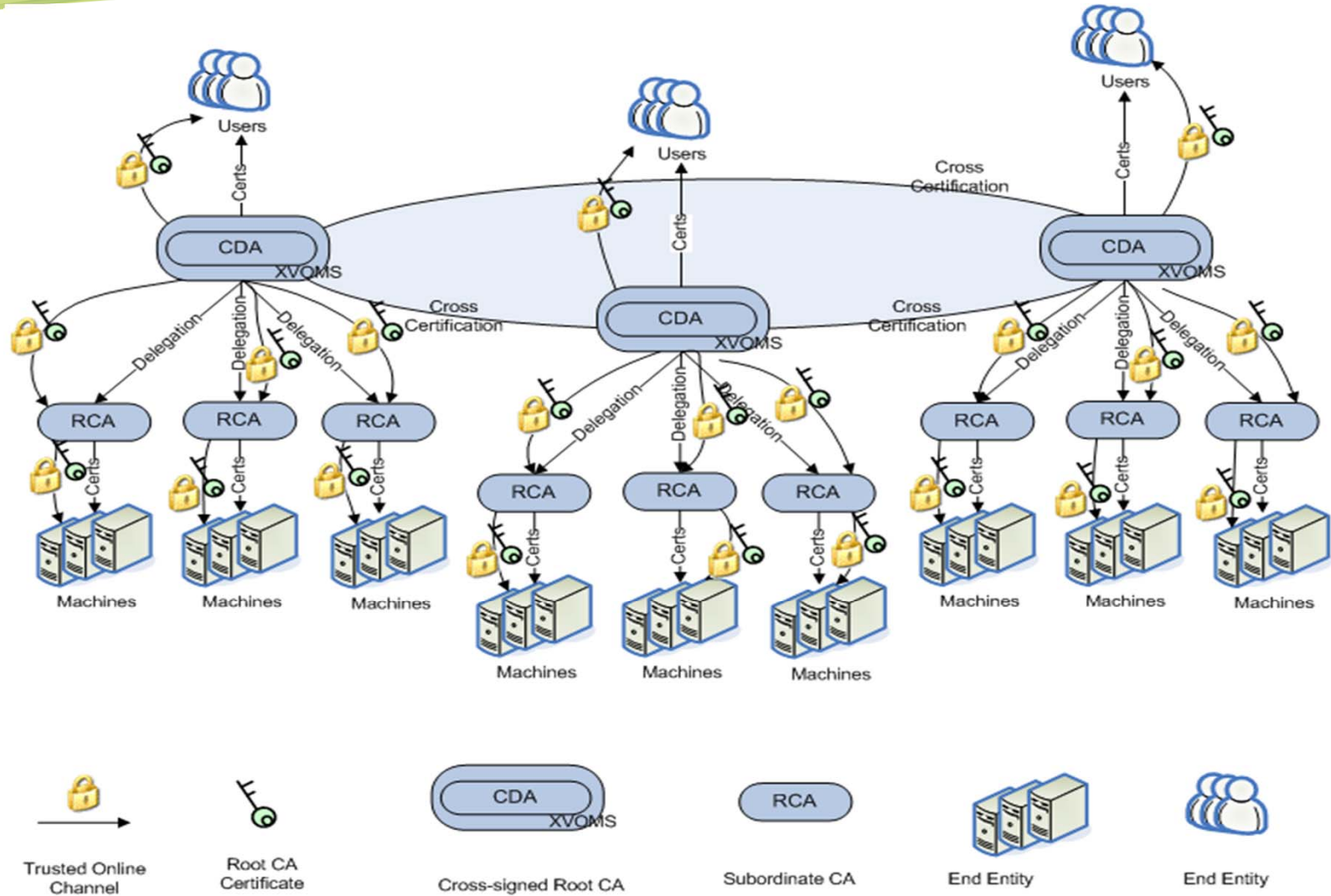


- **Distributed file system**
 - Spanning the grid
 - Replication
 - Striping
- **We saw in previous part how to set up an XFS volume**
 - Access control based on Grid attributes
 - Each XtreemOS users has a home volume in XtreemFS





Trust Model





- **XtreemOS Security Architecture Components and VO management**
- **At least one node (a core node) will host a CDA**
- **XVOMS: the database holding all information about active VOs within an XtreemOS platform**
 - Controls the other key services providing security and platform management
 - We will see the web GUI
 - Same functionalities available via shell commands, thus scriptable





- **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 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



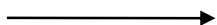


- **XVOMS**
 - **User and RCA registration**
 - **VO lifecycle management**
 - **Creation/dissolution**





VOLife Frontend



The screenshot displays the 'Virtual Organizations in Action' web interface. At the top, it features the XtreemOS logo and the slogan 'Enabling Linux for the Grid'. Below the header is a navigation menu with tabs for 'Home', 'Manage Users', 'Manage My VOs', and 'Manage My Resources'. A welcome message reads 'Welcome to VoLifeCycle , admin [logout]'. On the left, a sidebar menu lists various actions: 'Create a VO', 'Join a VO', 'My Pending Requests', 'Get an XOS-Cert', 'Generate new keypair', 'About me', 'Change Password', and 'Logout'. The main content area is titled 'Create a VO' and contains a form with the following fields: 'VO Name' (text input), 'Options' (checkbox for 'Automatic approving of requests(disabled)'), and 'VO Description' (rich text editor with a toolbar). At the bottom of the form are 'Create' and 'Cancel' buttons.



- **Create a VO**
 - Create an account, and use it to log in
 - VOLife show a link to obtain a copy of the Root certificate for your grid
 - Copy the root certificate in
`/etc/xos/truststore/certs/xtreemos.crt`
- **Home → Create a VO**
- **Home → My pending Requests**
 - Typical administrator tasks
- **Home → Join a VO, Home → Get an XOS Cert**





■ 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**





Select VO
and
send
joining
requests

Virtual Organizations in Action

Home Manage Users Manage My VOs Manage My Resources

Welcome to V

Create a VO

Join a VO

My Pending Requests

Get an XOS-Cert

Generate new keypair

About me

Change Password

Logout

Join a VO

Search:



JoinVO LeaveVO Refresh

<input type="checkbox"/>	GVID	VO Name	VO Owner	Is Member	Description
<input type="checkbox"/>	2fd9bc8f-a8a4-4195-85d0-272d1f63f093	testvo	admin	false	
<input type="checkbox"/>	4ecc77d7-c153-4a57-8430-b06df3825aa2	testvi	admin	false	
<input type="checkbox"/>	9d2dbf39-a754-4cc8-9b00-c6c83f218bd3	testes	admin	false	
<input type="checkbox"/>	f7206ce2-4d38-4432-9100-1aa0a5ec8152	ette	admin	false	
<input type="checkbox"/>	f39c6568-35c1-4f50-b7b8-d8c785dba11a	test11	admin	false	
<input type="checkbox"/>	1047e048-3739-45b6-ba04-d729832e539d	test1	admin	false	
<input type="checkbox"/>	94c0658a-4d15-4f15-b9aa-9340813253ce	asdf	admin	false	
<input type="checkbox"/>	9d705a80-6fcf-4a9c-a666-af51673e9f5b	11	admin	false	
<input type="checkbox"/>	276683d2-ed17-40d6-8f19-d52d1aa969b1	ppp	admin	false	
<input type="checkbox"/>	036bdc25-d01d-46b4-a56a-99a2aededfa0	xc	admin	false	
<input type="checkbox"/>	baca5795-823c-43b3-890b-3a556fef9290	test	admin	true	





- **Manage My VO → My Owned VOs**
 - Lists currently owned Vos
- **Manage My VO → Approve Requests**
 - Grant membership to other user/resources
- **Manage My VO → Manage Group Roles**
 - Each group or role is a set of users
 - Multiple groups in a VO, multiple roles in a group
 - Managed by clicking in the cells of the users table





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

Virtual Organizations in Action

Home Manage Users **Manage My VOs** Manage My Resources

My Owned VOs
Approve Requests
Manage Groups/roles
Manage Policies

Managing groups/roles

	Id	Name	Realname	Affiliation	Email
+		ppp			
+		test			
+		test11			
+		testvo			
+		test1			

Context menu for selected group:

- AddGroup
- AddRole
- AddUser
- Refresh





- **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**





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

Virtual Organizations in Action

Home Manage Users Manage My VOs Manage My Resources

- Create a VO
- Join a VO
- My Pending Requests
- Get an XOS-Cert
- Generate new keypair
- About me
- Change Password
- Logout

Get an XOS-Cert

Choose your joined VO:

VO Name: test

Specify Cert generating parameters:

Passphrase:

Retype-Pass:

Valid days: 40

Submit



Manage resources in a VO

Virtual Organizations in Action

Home Manage Users Manage My VOs **Manage My Resources**

- Register a RCA
- Add a Resources**
- Approve Resources
- Get Machine Certificates

Managing RCA Resources

Search: AddResource DelResource Refresh

<input type="checkbox"/>	Id	Name	RCA	VOs	Desc
--------------------------	----	------	-----	-----	------

Managing VOs

Search: AddToVO Refresh

<input type="checkbox"/>	Id	Name	Is Member	Owner	Desc
<input type="checkbox"/>	1	testvo	false	admin	
<input type="checkbox"/>	2	testvi	false	admin	
<input type="checkbox"/>	3	testes	false	admin	
<input type="checkbox"/>	4	ette	false	admin	
<input type="checkbox"/>	5	test11	false	admin	
<input type="checkbox"/>	6	test1	false	admin	





- **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**





- **Manage My Resources → Register an RCA**
 - Register an RCA in the database
- **Manage My Resources → Add a resource**
 - Add a resource to the RCA
- **Manage My Resources → Approve Resources**
 - Approve resource joining requests





- **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





Resource Monitoring

- **XtreemOS is a distributed platform**
 - Heavily relies on P2P mechanism to monitor resources
 - Fault-tolerant: resources can join and leave
- **SRDS – Service/Resource Directory Service**
 - Several P2P networks connect XOS resources
 - Many P2P daemons on each resource node
 - HTTP interfaces are provided to monitor the platform and the P2P network status





SRDS control web-interface

Menu

- [Overlay Weaver monitor](#)
- [Scalaris monitor](#)
- [Xosd Logger](#)

Filter search: more filters separated by spa

```
INFO: send(not resolved)/146.48.83.198:3997, TERMINATE_INVOKE)
Aug 7, 2009 4:00:51 PM ow.messaging.Message encode
INFO: tag:TERMINATE_INVOKE # of contents:6
Aug 7, 2009 4:00:51 PM ow.messaging.timeoutcalc.RTTBasedTimeoutCalculator updateRTT
INFO: To brunello.isti.cnr.it/146.48.83.198:3997: RTT: 46, ave. RTT: 36, ndev: 2, mdev_max: 500, rttvar: 500, timeout: 2036
Aug 7, 2009 4:00:51 PM ow.routing.impl.IterativeRoutingDriver$Querier call
INFO: Callback result: {(RODABXiyAC1ld554d1JL2W1vcySh2IHJc2VyaWVsaXpLZC5TZXJpYWhpenVKSzGFzaAAAMACSH7oE
AgABTAAYYXR0cnliLdXRlc3QAFUxqYXZlL3V0bWw5SGZaHRhYx10SHw3IAE2phdEueXRpOCS1
YXNodGFibGUtUw81UFRkaAAWAKYACmxyYmR5M053J1JAA10aHJlc2hvOGRIcD9AAAGAAIdwGk
AAALAAABnQADU3h0wRBonFpoGFibGV0AAAYkJE2MjLDAAZVclR0bWV0AAQ0NzQwdaAANRGlza0F2
50F...
EF2YwFsYwJsZXQACj1xNdc800MNd0BA5J2x61
nxldAAKHjK2MTG3HDg0Chg=,10770,0x6f775f736563726574}]
l.AbstractRoutingDriver invokeCallbacks
ll, onRootNode: false, on paraxos1.irisa.fr/131.254.201.16:3997
s1cCHTirpLS2 process
03133312e3235342e3230312e3136
cp.ConnectionPool get
e:
cp.TCPMessageSender send
201.17:3997, ROUTE_NONE)
message encode
TimeoutCalc.RTTBasedTimeoutCalculator updateRTT
1.17:3997: RTT: 3, ave. RTT: 2, ndev: 1, mdev_max: 500, rttvar: 500, timeout: 2002
cp.ConnectionPool get
e:
cp.TCPMessageSender send
201.17:3997, ADJUST_LAST_HOP_REO)
message encode
ntents:1
Aug 7, 2009 4:00:51 PM ow.messaging.timeoutcalc.RTTBasedTimeoutCalculator updateRTT
INFO: To paraxos2.irisa.fr/131.254.201.17:3997: RTT: 2, ave. RTT: 2, ndev: 0, mdev_max: 500, rttvar: 500, timeout: 2002
Aug 7, 2009 4:00:51 PM ow.messaging.tcp.ConnectionPool get
INFO: A Socket found in the hash table:
Aug 7, 2009 4:00:51 PM ow.messaging.tcp.TCPMessageSender send
INFO: send(not resolved)/146.48.83.198:3997, TERMINATE_NONE)
Aug 7, 2009 4:00:51 PM ow.messaging.Message encode
INFO: tag:TERMINATE_NONE # of contents:2
Aug 7, 2009 4:00:51 PM ow.messaging.timeoutcalc.RTTBasedTimeoutCalculator updateRTT
INFO: To brunello.isti.cnr.it/146.48.83.198:3997: RTT: 34, ave. RTT: 35, ndev: 2, mdev_max: 500, rttvar: 500, timeout: 2035
Aug 7, 2009 4:00:51 PM ow.routing.impl.IterativeRoutingDriver$Querier call
INFO: Callback result: null
Aug 7, 2009 4:00:51 PM ow.messaging.tcp.ConnectionPool get
INFO: A socket found in the hash table:
```

xuser@brunello.isti.cnr.it: /root/shots

[root@brunello ~]# curl 'SY-2a7G_RuFu_logger.png' -e -u SF

XChat: Dialog with bokim @ FreeNode

Screen Captur... X mrxvt XChat: Dialog... ..SRDS webint... Downloads xuser@brun... 10:01

HTTP localhost Port 9000
Main entry point to HTTP
monitoring interface
Offers error logging for the
brave





The screenshot shows a Mozilla Firefox browser window displaying the SRDS control web-interface. The address bar shows the URL `http://paraxos1.irisa.fr:9000/`. The page title is "SRDS control web-interface" and the main heading is "Overlay Weaver Node Status".

Menu

- [Overlay Weaver monitor](#)
- [Scalaris monitor](#)
- [Xosd Logger](#)

Node Information

URL: <http://paraxos1.irisa.fr:3998/>
Node ID: e9457528c174e5cae9c81f10be0ddf408ca0f372
Lookup algorithm: Chord
Lookup style: Iterative
of stored keys: 1

Routing Table

Predecessor

<http://brunello.isti.cnr.it:3998/> a2d13312afb9fa9d76f2c1d7bba219055d50fab5

Successor List

- <http://paraxos2.irisa.fr:3998/> f2cc2d0df0b7866d8520ae69d5bbb82c26feeed
- <http://brunello.isti.cnr.it:3998/> a2d13312afb9fa9d76f2c1d7bba219055d50fab5
- <http://paraxos1.irisa.fr:3998/> e9457528c174e5cae9c81f10be0ddf408ca0f372

Finger Table

- 1 <http://paraxos2.irisa.fr:3998/> f2cc2d0df0b7866d8520ae69d
- 157 <http://brunello.isti.cnr.it:3998/> a2d13312afb9fa9d76f2c1d7b

Put, Get and Remove Operations

operation	key	value	TTL (sec)	secret
get	<input type="text"/>			
put	<input type="text"/>	<input type="text"/>	600	<input type="checkbox"/> (option)

Buttons: submit, submit

A terminal window is visible in the foreground showing a shell prompt: `xuser@brunello.isti.cnr.it: /root/shots`. The terminal command is `scrot 'ZY-Zn-Xd_Bux$H3.png' -e 'mv $f "/shots/'`.

OW HTTP monitoring interface





..SRDS webinterface.. - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://paraxos1.irisa.fr:9000/
Most Visited Mandriva Mandriva Store Mandriva Expert Community Mandriva Wiki Jamendo Strumenti per le lin...
..SRDS webinterface.. Risultato della ricerca immagi...
Menu
[Overlay Weaver monitor](#)
[Scalaris monitor](#)
[Xosd Logger](#)

SRDS control web-interface

Number of nodes: 3

Total Load	Average Load	Load (std. deviation)	Rea
0	0.0	0.0	{{[131,254,201,16],14195,<0.99.0>}, {[131,254,201,17],14196,<6920.98.0>}}

Pred	Node
[893265177134352942183 2548441797358615757522747]	5726382988105907927169767348548375306 130823093654722548441 5726382988105907927169
07927169767348548375306]	8932651771343529421838435283374258190 [130823093654722548441 5726382988105907927169
29421838435283374258190]	130823093654722548441797358615757522747 [572638298810590792716 130823093654722548441

xuser@brunello.isti.cnr.it: /root/shots
[root@brunello shots]# scrot 'ZY-Zn-Zd_\$wz\$hd.png' -e 'mv \$f "/shots/'

1 2 3 4 Screen Captur... Xmrvt XChat: Dialog... ..SRDS webint... Downloads xuser@brun... 09:58

Scalaris HTTP monitoring interface





- **Scalable VO management**
 - Independent user and resource management
 - Interoperability with VO management frameworks and security models
 - Customizable isolation, access control and auditing
 - Scalable Hierarchical and P2P management of resources
- **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





www.xtreemos.eu

- **From there:**

- Official WWW <http://www.xtreemos.eu>
- XtreamOS Blog <https://www.xtreemos.org/blog>
- IRC channel for user support
[irc.freenode.net/channel #xtreemos](irc://irc.freenode.net/channel/#xtreemos)
- Mirrors for ISO Downloads and Package Updates

<http://www.xtreemos.eu/software/mirror-websites>
<http://www.xtreemos.eu/software/experimenting-xtreemos-on-virtual-machines>



A.4 2nd XtreamOS Summit – XtreamOS challenge part

XtreemOS



*Enabling Linux
for the Grid*

XtreemOS Summit

Christine Morin, INRIA Rennes-Bretagne Atlantique

XtreemOS scientific coordinator

August 30, 2010

XtreemOS IP project

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



Information Society
Technologies





- **9:35-13:00 – Tutorial**
 - Easing Application Execution in Grids with XtreemOS operating system
 - 11:00-11:30 - *Coffee break*
- **13:00-14:30 – *Lunch***
- **14:30-15:40 – XtreemOS demonstrations**
- **15:40-16:00 – XtreemOS testbed presentation**
- **16:00 – 16:30 – *Coffee break***
- **16:30-17:15 – Results of the XtreemOS Challenge**
 - Presentation of experimental results
 - Awards
- **17:15-17:45 - Open discussions & closing remarks**

XtreemOS



*Enabling Linux
for the Grid*

XtreemOS Challenge

August 30, 2010



Information Society
Technologies

XtreemOS IP project

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





- **First computing challenge to demonstrate an application running on XtreemOS**
- **Opportunity to test applications on a Linux-based Grid operating system and perform large-scale experiments on the Grid**



- **Enlarge XtreemOS user community**
- **Increase the pool of demonstrable use-cases, user's experiences and success stories**
- **Get feed-back from external users**



- **Application (real/complex):** how it takes advantage of the XtreemOS system
- **Scalable system:** size of the deployment
- **Reproducibility of experience**
- **Other selection criteria:**
 - Complexity of the application
 - Appropriateness for XtreemOS Grid system
 - Potential impact, work done to port the application to XtreemOS
 - Tools developed to deploy the demonstration
 - Presentation of the demonstration in a paper and video.



- Bernd Scheuermann (SAP - Germany)
- Yvon Jégou (INRIA - France)
- Marjan Šterk (XLAB - Slovenia)
- Nicolas Vigier (EDGE-IT/Mandriva - France)
- Massimo Coppola (Consiglio Nazionale delle Ricerche - Italy)
- Alvaro Arenas (Science and Technology Facilities Council - UK)
- Santiago Prieto (Telefónica I+D - Spain)
- Björn Kolbeck (Zuse Institute Berlin - Germany)
- Thilo Kielmann (Vrije Universiteit Amsterdam - Netherlands)
- Louis Rilling (Kerlabs - France)



- **Parallel Kriging**
 - Alvaro Parra, Exequiel Sepulveda & Felipe Lema from ALGES lab at Universidad de Chile
 - <http://www.youtube.com/watch?v=o2183WfVBuk>

- **Grid Security Operation Center**
 - Syed Raheel Hasan, Laboratoire Informatique, Université de Franche-Comté, France
 - <http://www.youtube.com/watch?v=VUUaRpzusdo>

- **Monte Carlo Simulation for Single-Photon Emission Computed Tomography with XtreemOS**
 - Emanuele Carlini, Sebnem Erturk & Giacomo Righetti from University of Pisa, Italy
 - http://hpc.isti.cnr.it/xos/XOS_Challenge_Video.qt



And the Winner is ...

Monte Carlo Simulation for Single-Photon Emission Computed Tomography with XtreemOS

Emanuele Carlini, Sebnem Erturk & Giacomo Righetti from
University of Pisa, Italy

XtreemOS



*Enabling Linux
for the Grid*

Closing Remarks

August 30, 2010



Information Society
Technologies

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





- **Release 2.1.1 available for PC, cluster, mobile device**
- **Packaged in Mandriva & Asianux Linux distribution**
- **VM images available for KVM and Virtual Box**
- **Open permanent testbed**



- **Web site: <http://www.xtreemos.eu>**
- **Software: <http://gforge.inria.fr/projects/xtreemos/>**
 - GPL/BSD licence
- **Email: contact@xtreemos.eu**

XtreemOS Challenge 2010

PhD Student: Syed Raheel Hasan
(raheel.hasan@univ-fcomte.fr)

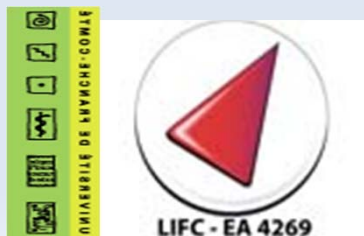
Masters Students:

Jasmina (jasmina.pazardzievska@yahoo.com)

Maxime Syrame (msyrame@gmail.com)

Supervisor: Prof. Julien Bourgeois
(julien.bourgeois@univ-fcomte.fr)

Computer Science Laboratory (LIFC),
University of Franche-Comte (UFC),
1 Cours Leprince-Ringuet, 25201,
Montbéliard. France.



Agenda

- Need For developing Grid Security Operation Center (GSOC)
- Introduction of GSOC
 - Cbox, Abox, Dbox and SvoBox
 - GSOC General Diagram
 - Security Alert Composition in GSOC
 - GSOC Reports Specific for XtremOS.
 - Basic and Advance Correlation of Security Alerts in GSOC
- Security Flaws in XtremOS
- Scalability of GSOC in multi grid and cloud computing networks
- Conclusion
- References

Motivation for Developing GSOC

At present grid networks are,

- working within very trusted sites,
- they share limited number of resources,
- but the trend of interconnecting multiple grids is evolving,
- therefore with the passage of time and especially by the innovation of desktop grids the risk is very high that grid computing networks will also be attacked like brute force attacks, denial of service attacks(DoS) or distributed denial of service attacks(DDoS) etc...
- No solution is present to date to counter these attacks in grid computing networks.

Contd.....

- The main difficulty in grid computing networks is to deal with the specificities of grid infrastructure, that are:
 - multi-sites networks, multi-administrative domains,
 - dynamic collaboration between nodes and sites,
 - high number of nodes to manage,
 - no clear view of the external networks
 - and exchange of security information among different administrative domains [1].

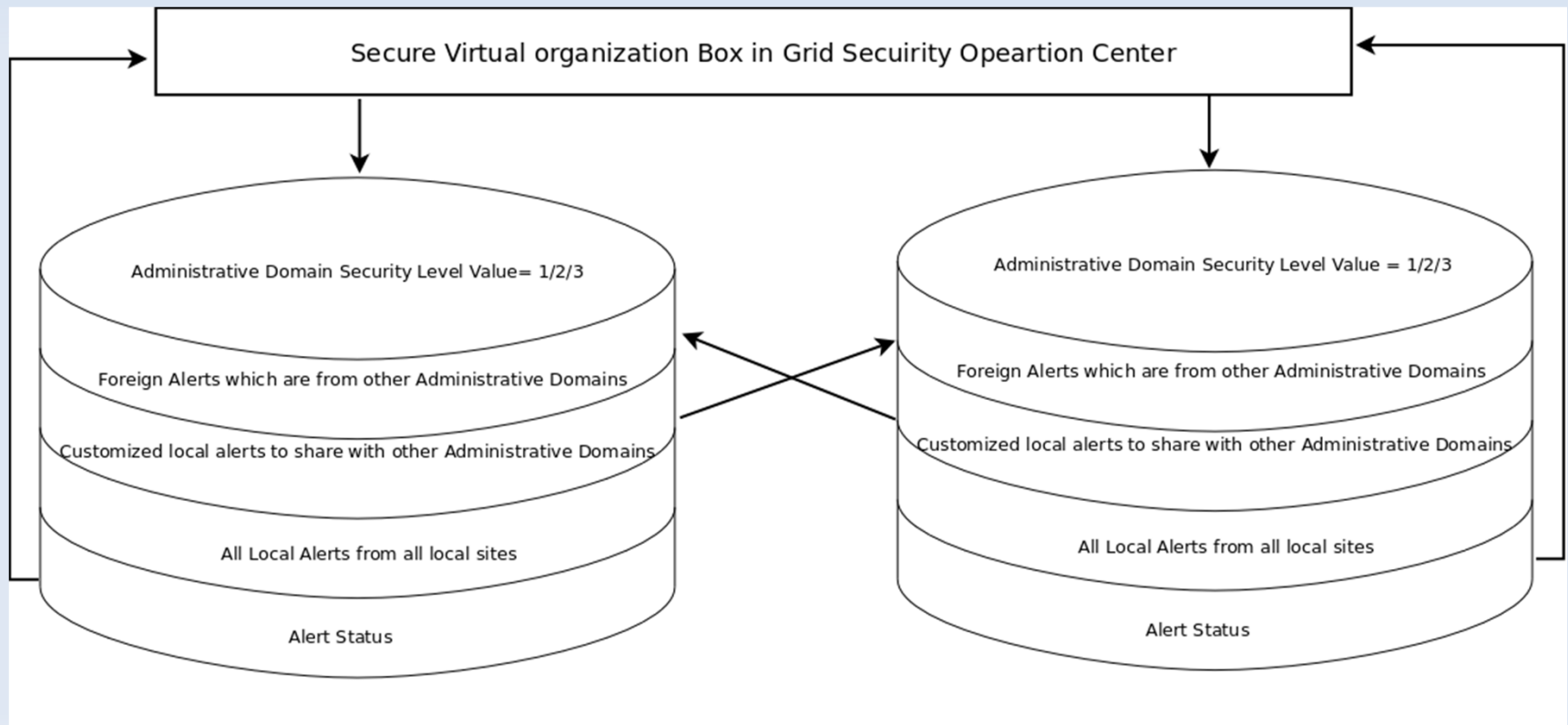
Grid Security Operation Center (GSOC)

GSOC is an Intrusion Detection System for grid computing networks . It has four main componets which work together in distributed locations within a grid network.

- CBOX: is a log collecting modules which collects logs from any computing nodes like computers, switches, routers, IDS/IPS, anti-virus servers, firewalls etc.
- ABOX: contains security rules created by an administrator.
- DBOX: is a local intrusion data base which is based on Common Vulnerability Exposures(CVE) [5]
- SVOBox: to provide a mechanism to share customized security alerts within multiple administrative domains in a grid network. This means that an administrator has a choice that he can share any specific security alerts at a specific time period [1].

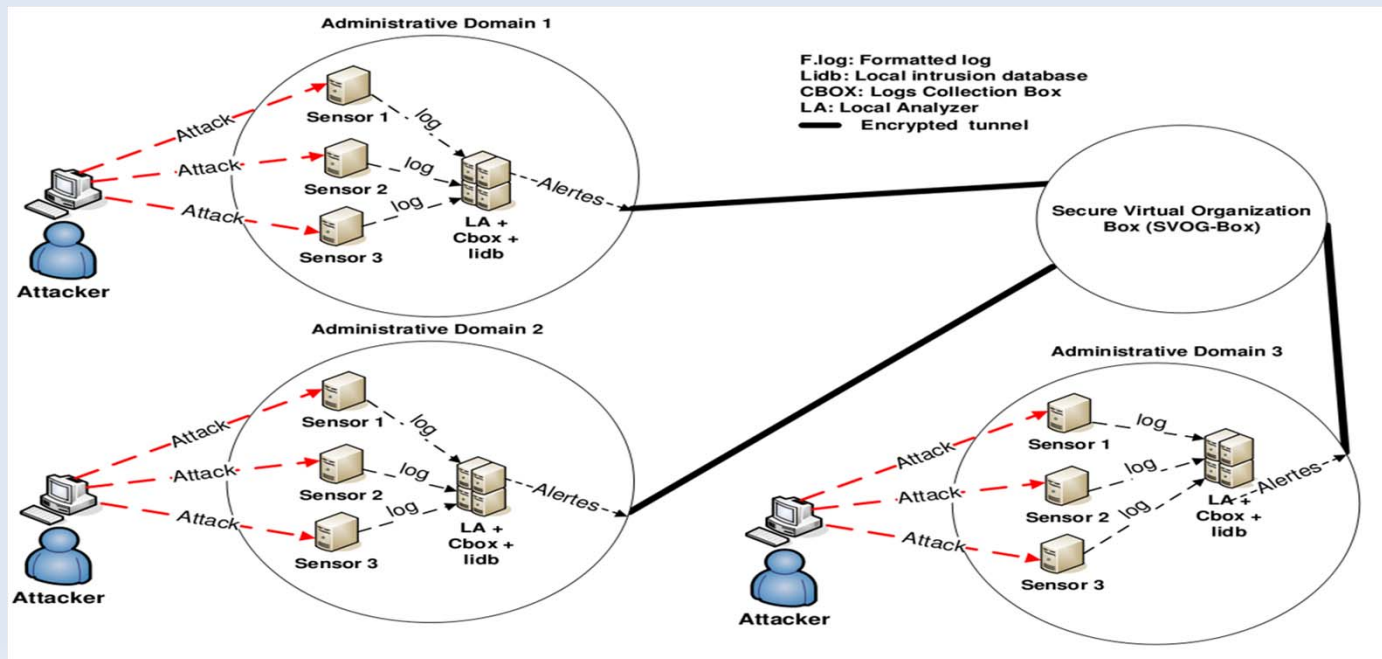
Secure Virtual Organization Box (SVOBox)

- The job is SVOBox is to share the security alerts with other grid networks at real time.



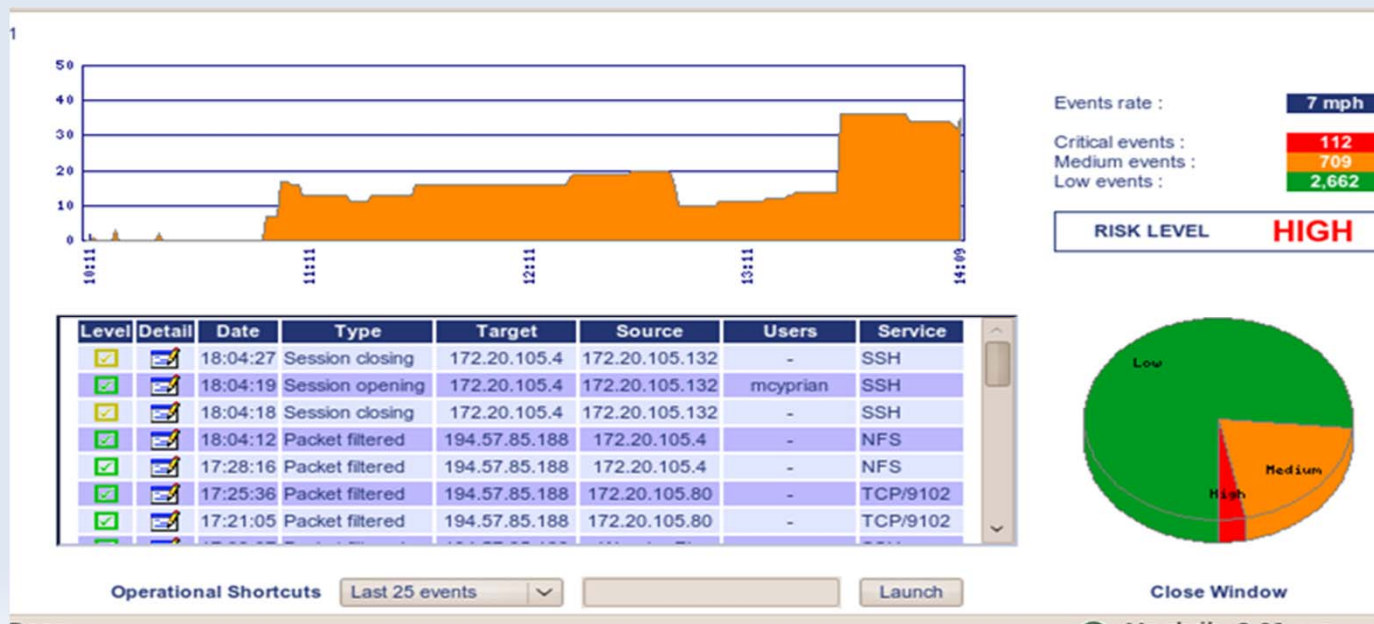
GSOC General Diagram

By using Grid Security Operation Center (GSOC) an administrator of a grid network can see what is happening at administrative domain 1, 2 or 3. Similarly the administrators of AD1, AD2 and AD3 can also see the security status of other external domains. This will help the grid network administrators to isolate the administrative domains if it will be under attack.



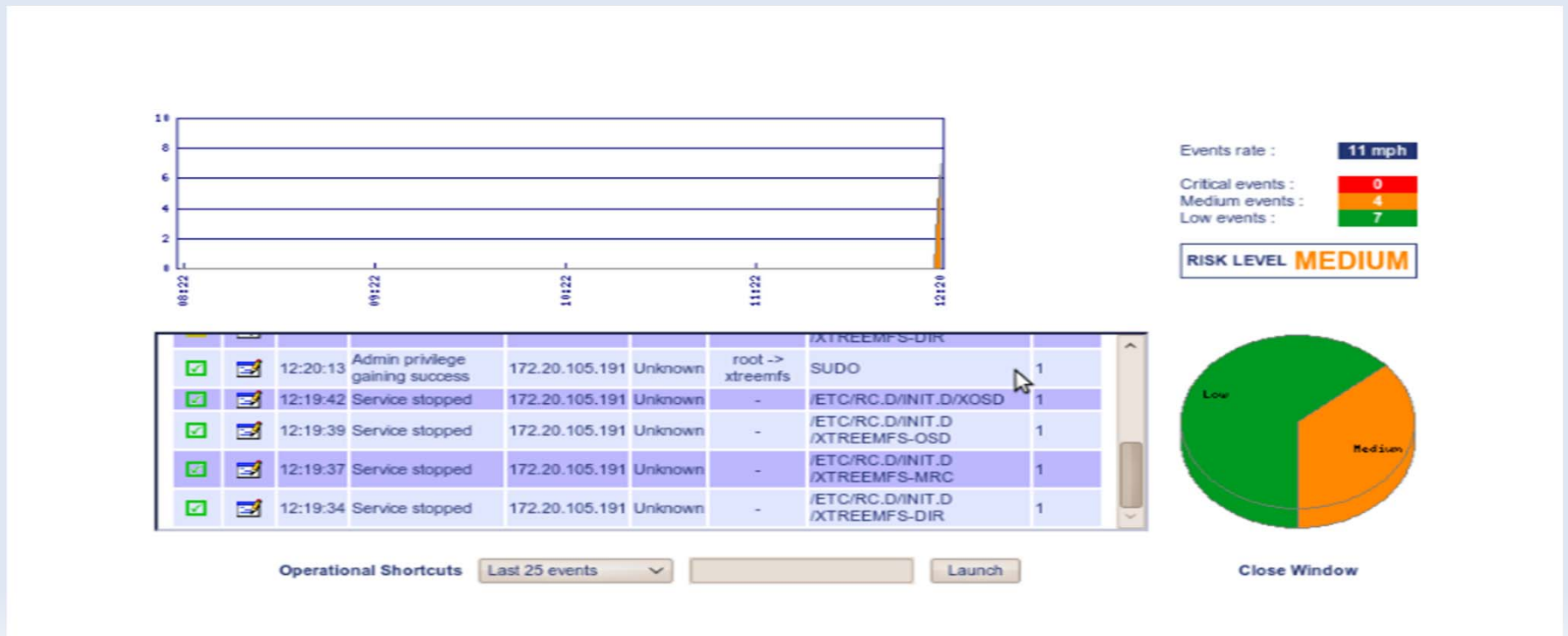
Security Alert (SA) Composition

SA categorizes the user events in three ways. The minimum starts with green color which are the lowest and indicates minor incidents, orange color which are the medium level incidents and then the red color which are the most high level incidents.



GSOC Reports Specific for XtremOS

Generating alerts if the XtremOS Directory Service (DIR), MetaData and Replica Catalog Service (MRC) and Object Storage Device Service (OSD) has been stopped/started or restated. Similarly other critical services and operations can be reported after configuring in GSOC.



Contd.....

Using GSOC an administrator can detect ping of death, DoS/DDoS, brute force attacks, user privileges gaining attempt , user privilege gaining success , user group modification , user account activation , session opening/closing on any computer, multiple authentication failures, service restarted/stopped/started, password change attempt, password changed, login refused, packet filtering etc...

For minimum deployment of GSOC in one administrative domain only two machines are required; one for collecting logs (CBOX) and another one which holds the intrusions database called DBox, Apache Web Server and a local analyzer called LA. There is no need to install any software on any node.

Basic and Advance Correlation in GSOC

- Cbox is responsible for collecting logs for one local site therefore basic correlation at each Cbox reduces the size of logs to be saved in local intrusion database. Then these saved logs identifies the type of attack which is then further forwarded to LA for advance correlation.
- Local Analyzer is responsible for collecting correlated messages from multiple Cboxes. All the correlated security alerts are then finally reported at Dashboard and these alerts will then be shared with multiple grid networks.

Security Flaws in XtreamOS

- In the technical documentation of XtreamOS, it has been mentioned that in XtreamOS the grid middleware has been merged with the kernel [3].
- Due to this merger, now XtreamOS is more vulnerable as compare to other grid systems which are using operating system and grid middlewares separately.
- Now XtreamOS could be compromised due to system critical errors that could be due to normal networks attacks like brute force, ping of death and DDoS. If not compromised then the performance of the XtreamOS could be degraded once the kernel security patches will not be applied regularly.

Scalability of GSOC in multi grid and cloud computing networks

- The architecture of GSOC has been designed to be further extend-able in multi grids and cloud computing networks.
- For one local site minimum one cbox required. But the number of cboxes can be increased depending on the size if local site.
- One LA is required for one administrative domains which controls all the cboxes.
- One SvoBox is required for interconnecting multiple grid networks and cloud computing networks.

Conclusion

- Intrusion detection systems for distributed networks like SNORT [6], BRO[7], Tripwire[8], DSOC[2] etc... does not allow grid network administrators to share their security alerts with other grid networks. Whereas GSOC provides this feature which helps to counter,
- Local attacks within locally administered network
- External attacks which are occurring within multiple grids
- Basic and advance correlation reduces the size of logs while saving in database which saves disk space and minimizes consuming network bandwidth while correlating for more sophisticated attacks.
- GSOC when deployed in XtremOS can alert the administrator about the attacks even the kernel or security patches are not

References

- [1] Julien Bourgeois and Raheel Hassan. Managing Security of Grid Architecture with a Grid Security Operation Center. Vol. ISBN:978-989-674-005-4:403-408, SECRYPT International Conference on Security and Cryptography, Milan, Italy, INSTICC Press, 7-10 July 2009
- [2] Abdoul Karim Ganame, Julien Bourgeois, Renaud Bidou and François Spies. A Global Security Architecture for Intrusion Detection on Computer Networks. In Computers & Security, Elsevier Publisher, Vol. 27(1-2):30-47, 2008
- [3] State of the art in the security for OS and Grids (D3.5.1) - December 2006(http://xtreemos.org/science-and-research/plonearticlemultipage.2007-05-03.6637185684/copy_of_virtual-organisations-and-security-management)
- [4] Northcutt, S. and Novak, J. (2002). Network Intrusion Detection. ISBN: 0-73571-265-4. New Riders, third edition edition. September.
- [5] Common Vulnerabilities and Exposures (CVE). <http://cve.mitre.org/>
- [6] Open source network intrusion prevention and detection system (IDS/IPS). www.snort.org/
- [7] Bro is an open-source, Unix-based Network Intrusion Detection System (NIDS). www.bro-ids.org/
- [8] Tripwire . www.tripwire.com/

XtreemOS



*Enabling Linux
for the Grid*

Monte Carlo Simulation for Single Photon Emission Computed Tomography with XtreemOS

XtreemOS Challenge

E.Carlini

S.Erturk

G.Righetti



Information Society
Technologies

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





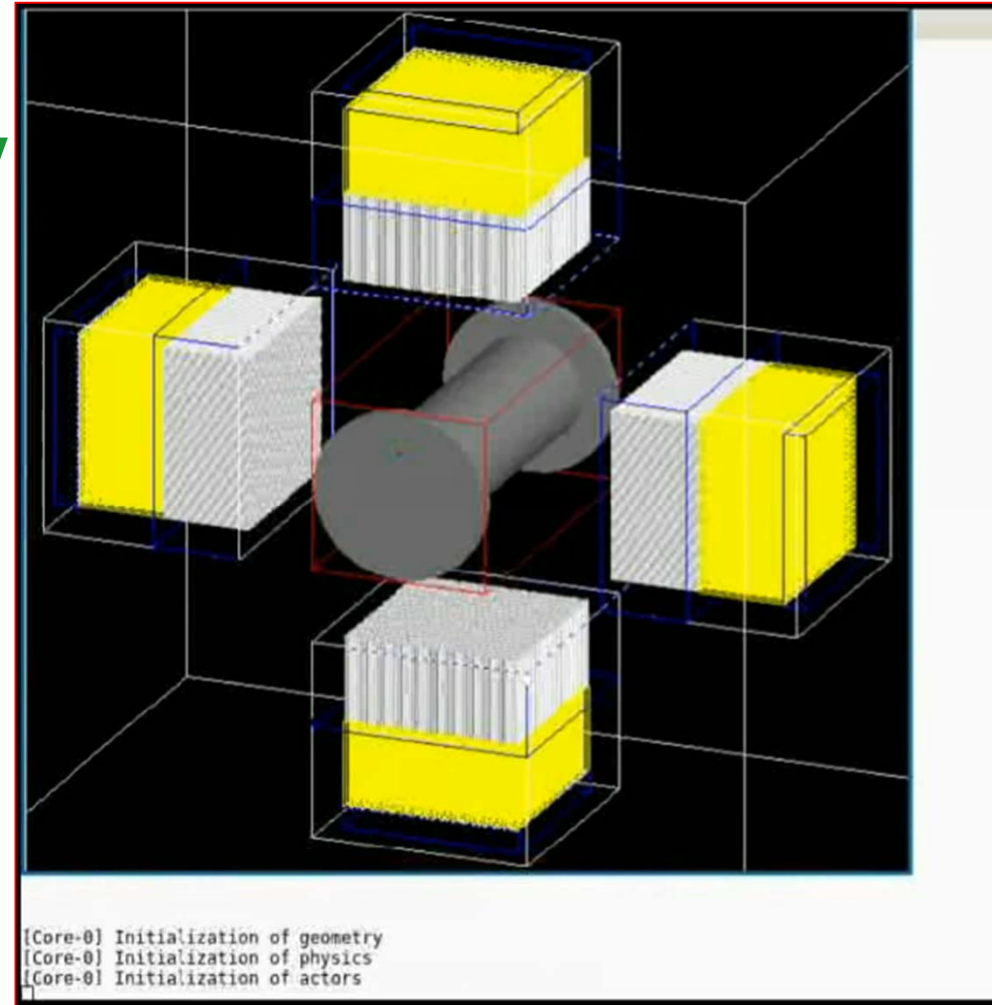
- **The SPECT case study**
- **Parallelization**
- **XtreemOS**
 - Motivations
 - Integration
- **Conclusions and Future Works**





Single-Photon Emission Computed Tomography

- Imaging technique
- In-vivo volumetric analysis of the distribution of the radiopharmaceuticals
- Result: 3D image of the probed body

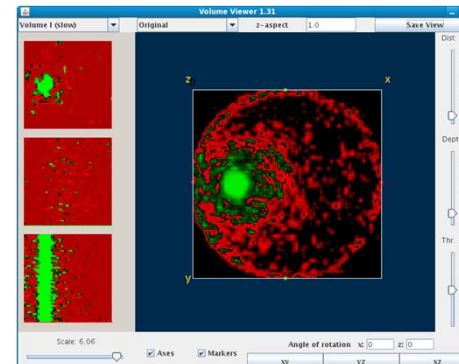




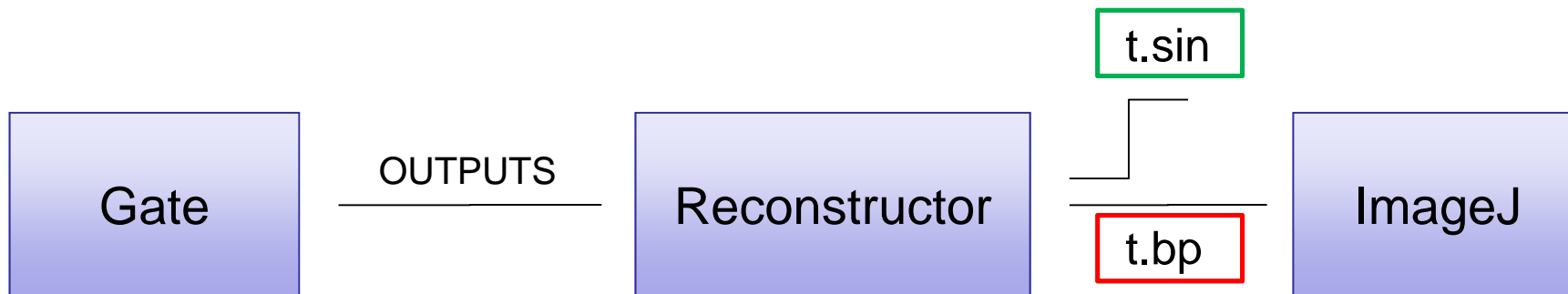
What is Image Reconstruction?

views
=
projections

projections together
=
image reconstruction

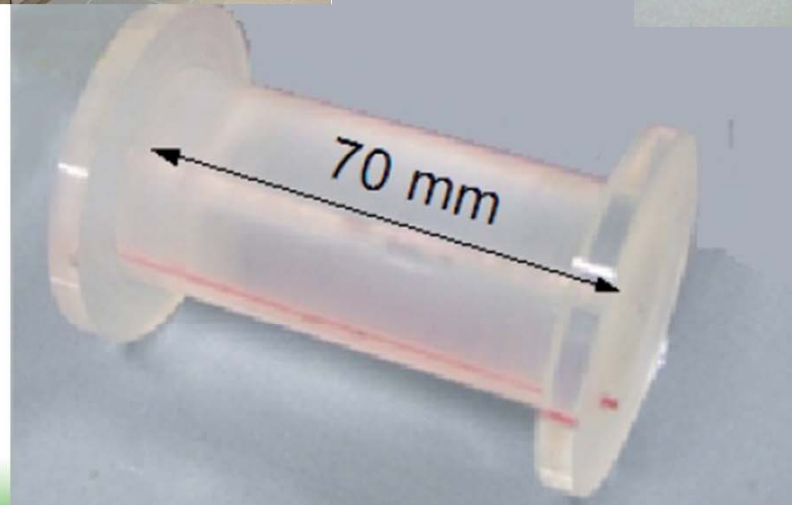
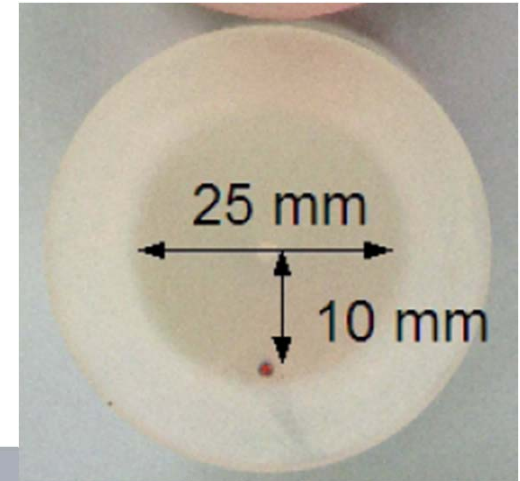


System Response Matrix : transforms projections into a 3D image





YAP-(S)PET Small Animal Scanner





Why the computational simulation?

- **Optimization purposes**
 - **Collimator** design is a lengthy process
 - High-precision metallic honeycomb-like structures put in front of particle detectors
 - Material and design affect detection **sensitivity**, **directionality**, image **resolution**
 - **Reconstruction** algorithms
 - Collimator design affects **algorithm optimizations** and the **System Response Matrix**
- **Lower the costs of evaluating different designs**
 - Shorter time-to-market, cheaper and better device





Monte Carlo simulation

deterministic



Monte Carlo
probabilistic
approach

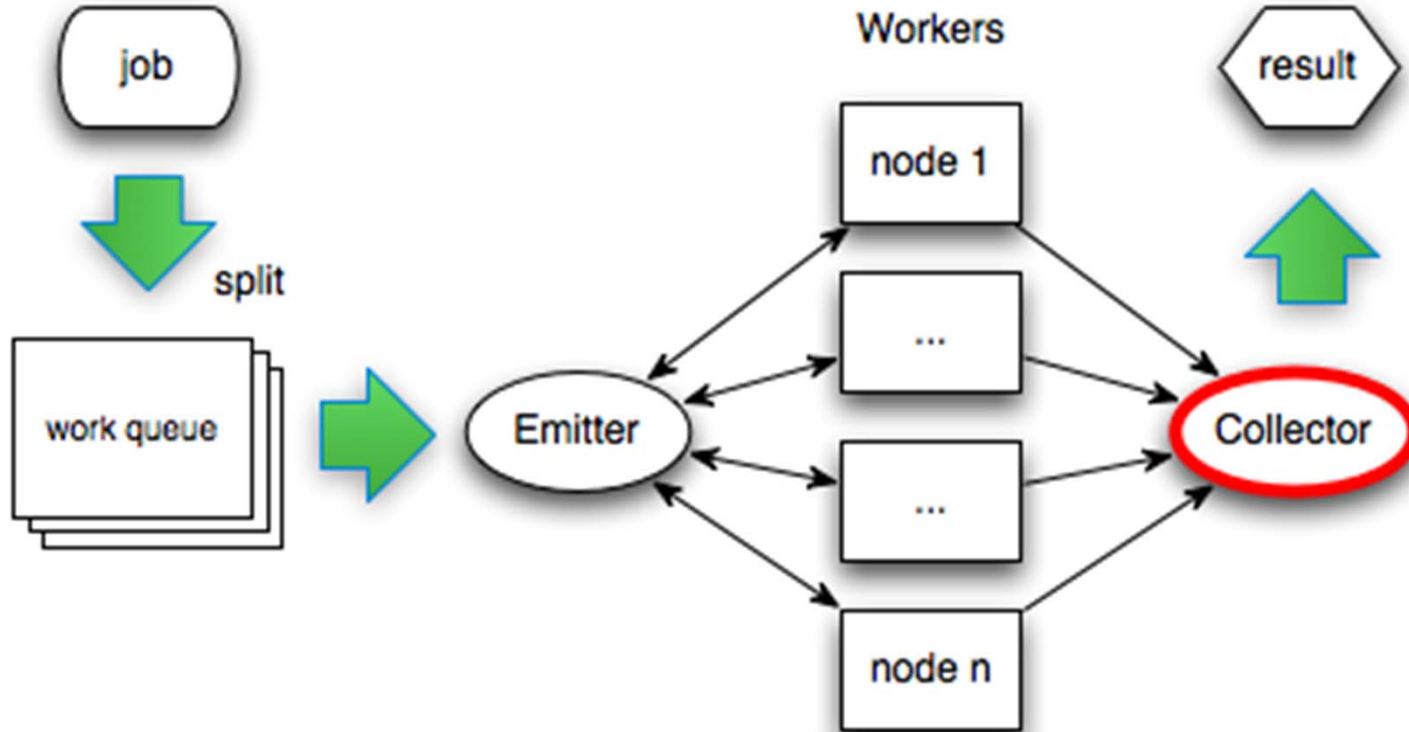
High
computing
power

High degree
of accuracy

Independent
tasks



How we parallelized





- <http://www.opengatecollaboration.org/>
- **Numerical simulations in medical imaging**
- **Developed by OpenGATE collaboration**
- **Macro-based scripting language**





- **Click to edit Master text styles**

Second level

Third level

Fourth level

Fifth level





- **Tweaking the XtreemOS distribution**
 - Add libraries used by GATE simulation/reconstruction
 - Recompile all GATE tools (v6 patch 01)
 - Functional tests on the experimental & open testbeds
- **Enabling distributed, heterogeneous execution**
 - Custom parallelism control on top of XtreemOS
 - Introduce Fine parallelism grain, Fault Tolerance, Load Balancing, I/O distribution
 - Implementation is flexible, back-compatible with GATE for clusters





- **Making ready-to-run XtreemOS images**
 - Disk-image size issues with XtreemOS v2.1+GATE
 - Large images can timeout at deploy time
 - Developed a workaround
 - Deployed XtreemOS platforms up to 100 nodes
 - G5k clusters in Rennes and Grenoble
 - Up to two cluster at a time for a simulation
- **G5K was a precious resource**
 - to develop a large-scale XtreemOS application
 - deploying a large platform each time before use is still a complex task





- **Ongoing work: gather and exploit results**
 - Analyze different collimators design for the YAP-(S)PETII scanner
 - Computing SRM for different prototypes
- **Future Work**
 - Research heuristic methods to simplify SRM calculations
 - Big Challenge in Applied Physics!
We need a lot of simulation data :
huge amount of radioactivity = very heavy simulations





- **Improve integration with XtreemOS AEM & monitoring services**
 - Fully exploit dynamic resource allocation in AEM
 - Can job support be used for application sub-jobs?
 - Further tests with XtreemFS features
 - Rewrite bash scripts with other languages
 - XOSSAGA, XtreemOS API
- **Experiment other XtreemOS installations**
 - Open testbed, hybrid XtreemOS platforms, examples:
 - G5K + open testbed
 - G5K + owned cluster using XtreemOS virtual machines





Thanks for your attention!

Any questions?

