



European Commission



# VO management in Grid4All

Nikos Parlavantzas (INRIA) , **Vladimir Vlassov (KTH)**

XtremOS – Grid4All meeting  
Paris, July 2008



Information Society

# Grid4All

# Scenarios



■ A school teacher plans an education project involving resources from other schools

□ How can he set up necessary IT organisation?

■ A domestic user wants to convert a set of movies to some format in a reasonable amount of time

□ Where can he get sufficient amount of resources?

# Critical Aspects

- Non-professional users
- Social behaviours with churn and high volatility
- Need for incentives to make the system spread and thrive
- Need for support for different styles of collaboration

# Our approach

- Self-configuring overlay infrastructure
- Aggressive support for autonomic management at different levels
  - Self-managing services & applications
- Flexible access to shared mutable data in volatile environments
- Job execution in volatile environments
- Market-based allocation of resources across VOs

# VOs in Grid4All (1/2)

- VO is a (highly) dynamic, virtualized collection of users that pool their resources into a single virtual administrative domain for some common purpose
- A VO might provide (possible member expectations)
  - Shared workspaces
  - Collaborative applications & social networks
  - Storage
  - Resources for execution (software, CPU cycles)

# VOs in Grid4All (2/2)

- Resources are donated (currently altruistically) to the VO by its members and/or allocated at resources marketplaces
  - Currently, we do not set obligations to members
    - Not guarantees on availability of resources
  - Currently, no resource sharing policies
  - Lease agreements with external providers
- Inter-VO resource markets
  - Resource auctions operating across VOs
  - VO may lease external resources at marketplaces
  - VO may offer its resources at marketplaces

# Requirements to VO management

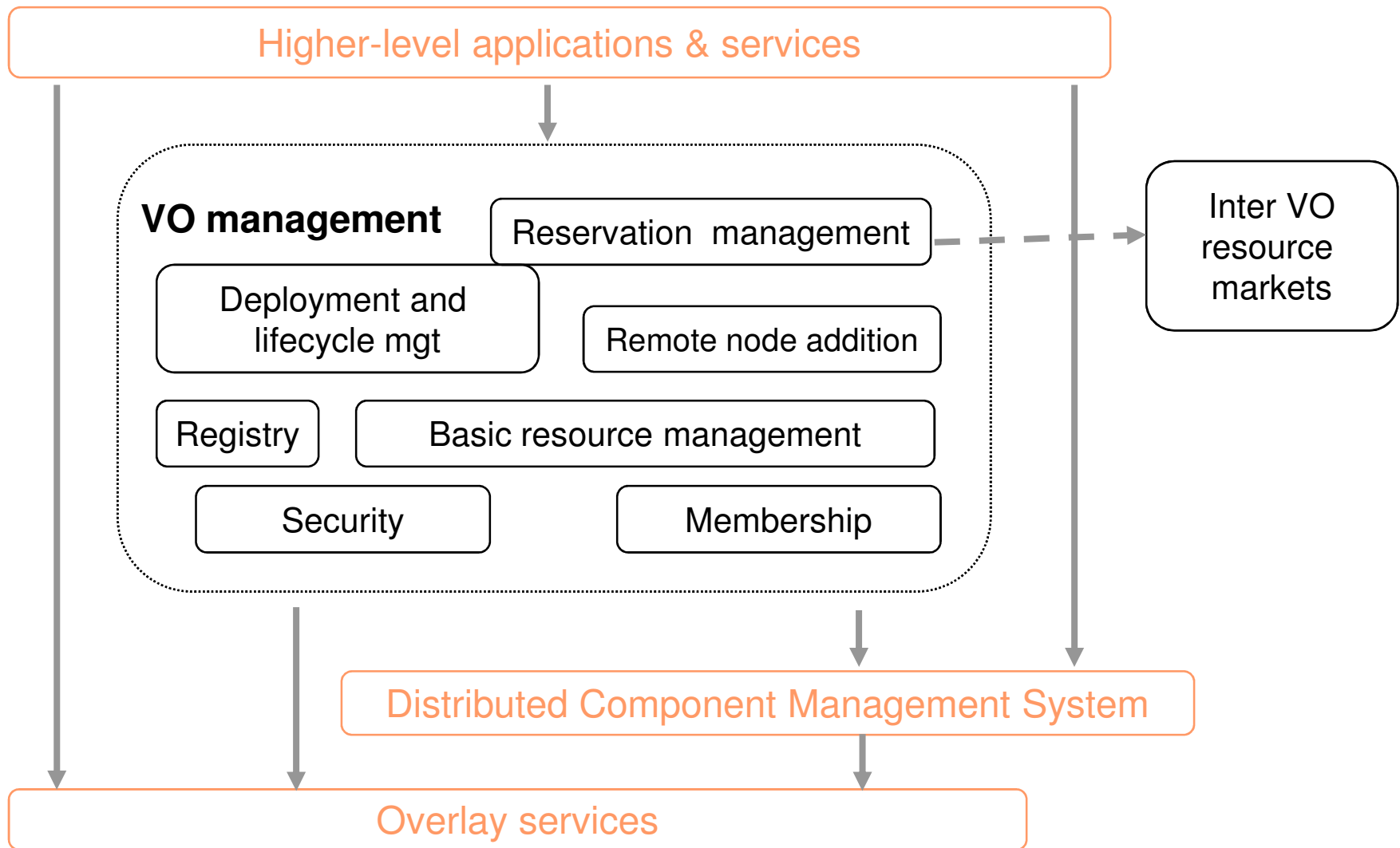
- Functional:

- Membership and security;
- Resource provision and management;
- Deployment and application management

- Non-functional:

- Ease of use and administration
  - Autonomous management wherever possible
- Availability, scalability and decentralisation

# Architecture



# Scenario (1/2)

- Interactions between membership (VOMS), resource management (RM), deployment (D), registry (R), and security (S)
  1. A user invokes VOMS to log in to the VO
  2. The user invokes RM to contribute some computational and storage capacity to the VO
  3. The user invokes R to ask for a given service; service is not available.

# Scenario (2/2)

4. The user invokes D to deploy an application that implements the given service.
5. D uses RM to discover and allocate appropriate resources for hosting the application components. RM verifies that the user has the right to allocate these resources based on S.
6. D instantiates the application components on the allocated resources, binds them, and activates the application. D also registers the given service using R.
7. The user invokes R to obtain the service and then invokes it.

# Implementation (1/2)

- Set of components integrated into the Grid4All container
  - customised Jade system
  
- Grid4All container
  - Based on Java implementation of the Fractal component model
  - Has architecture described in Fractal ADL
  - Hosts:
    - VO management
    - DCMS system
    - Niche overlay services
    - Higher-level services and applications built or wrapped as components

# Implementation (2/2)

- VO physically corresponds to a set of Grid4All containers connected through Niche-based overlay network
- Two container configurations: *JadeBoot* and *JadeNode*
  - *JadeBoot* bootstraps the system and includes any “centralised” VO functionality (e.g., LDAP server)

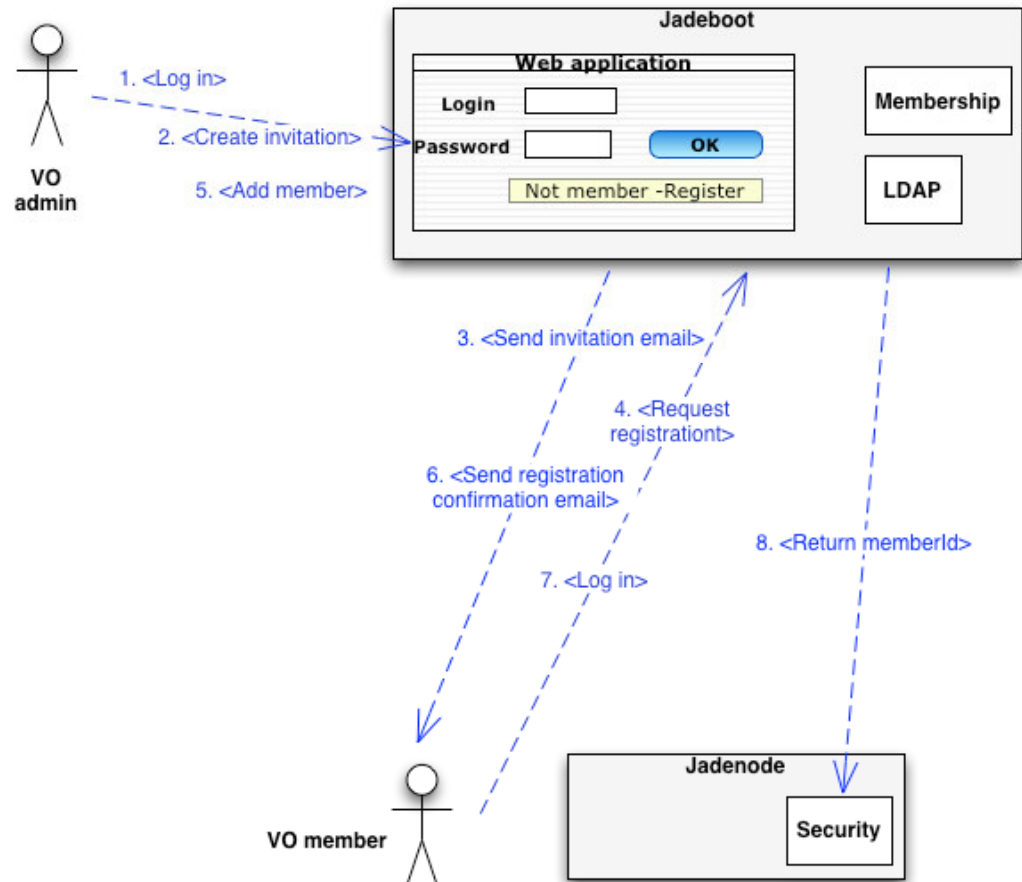
# **VO management services**

# Membership (1/2)

- Handles authentication of VO members and keeps associated information (roles, on-line status, policies, etc)
- VO creator
- VO administrator
  - dissolves the VO; adds and removes members
  - creates roles and associated VO-level policies, assigns roles to members
  - creates invitations to members
- Member (as a user or/and a resource provider/owner)
  - registers with a VO; logs in/logs out
  - retrieves information on members
  - defines resource-owner policies

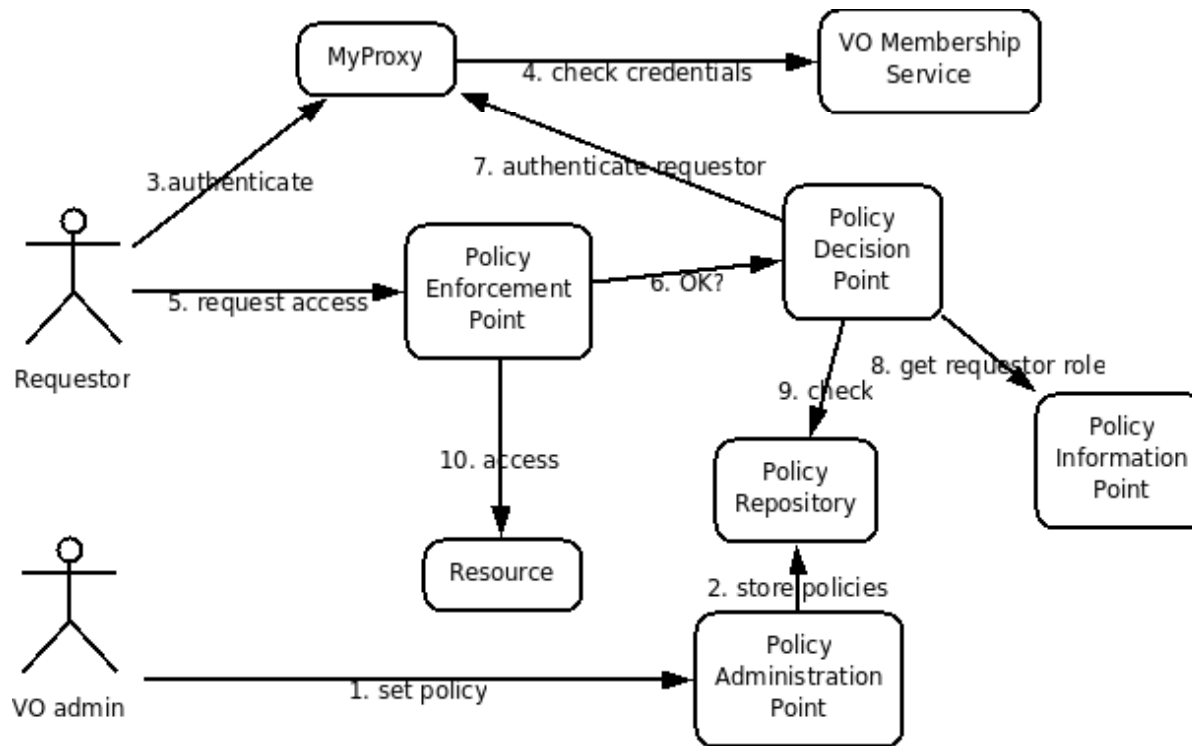
# Membership (2/2)

- Currently centralised architecture that relies on LDAP server for storing information
- Decentralised implementation using the DHT is possible
- Web-based interface for administrators and members



# Security infrastructure

- Framework to support policy-based authorization
- Prototype using Sun's XACML implementation
- VO policies, resource owner policies



# Basic resource management (1/3)

- Deals with management of resources of a single VO
- Provides simple, lightweight mechanisms for resource sharing:
  - Discovering, allocating, and monitoring VO resources
  - Contributing resources to the VO
- Supported operations subjected to VO authorisation policies by exploiting the security infrastructure
- Intended as basis for more sophisticated Grid resource management services

# Basic resource management (2/3)

## ■ Conceptual model

- A *node* is the unit of resource contribution by a VO member.
- A *resource* is a share of the processing, storage, communication capacity of a node. It is used to deploy components
- Discovery and allocation are based on resource properties
  - e.g., “((storage>=500000000) and (networkSpeed=medium))”

# Basic resource management (3/3)

- Decentralised implementation based on overlay services and DCMS
  - Discovery currently uses broad(multi)cast
  - Deployment, given a Fractal ADL description
  
- Overlay Services
  - Replicated DHTs, pub/sub event dissemination, name-based communication
  - Based on the DKS structured overlay
  
- Distributed Component Management System (DCMS)
  - Framework for building self-managing services and applications
  - Based on Fractal component model

# Reservation management

- Supports advance reservation of resources available within the VO or obtained on demand from resource markets.
  - If allocation is not possible using currently available resources, it initiates negotiation at the resource market and leases additional resources
- Example of a higher-level service that builds on basic resource management as well as the Grid4All resource brokering facilities

# Deployment (1/3)

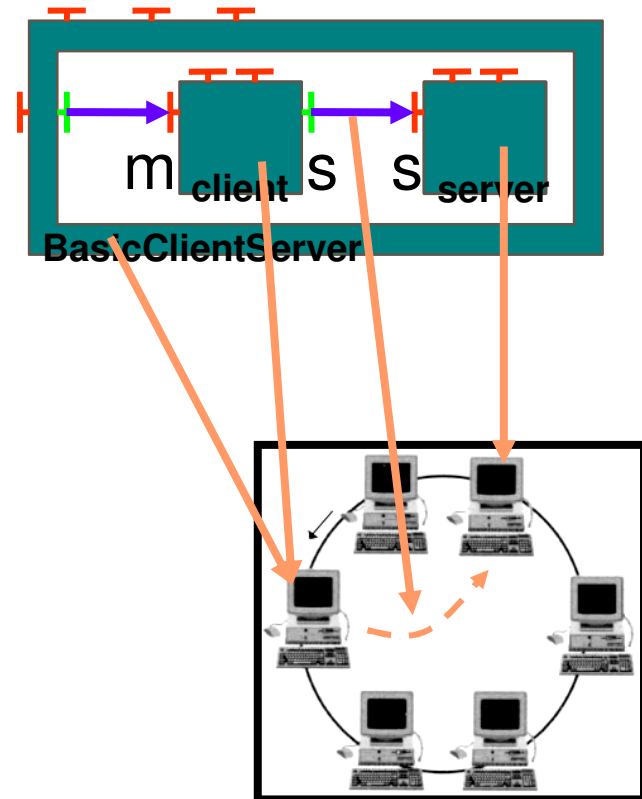
- Facilitates the installation, configuration, and activation of distributed, component-based applications within the VO
  - applications include functional and self-management part
- Takes as input an *application description* that contains
  - architecture (components, containment, bindings)
  - component attributes
  - packaging and placement information
  - resource requirements

# Deployment (2/3)

## Example

```
<definition name="BasicClientServer">
  <interface name="service" role="server"/>

  <component name="client">      ....
    <virtual-node name="node1"
      resourceReqs="MemorySize>1GB"/>
  </component>
  <component name="server"
    ...
    <virtual-node name="node2"
      resourceReqs="MemorySize>4GB"/>
  </component>
  <binding client="this.service" server="start.service" />
  <binding client="start.rsrc_apache" server="apache.resource"
    bindingType="oneway" />
  <virtual-node name="node1"/>
</definition>
```



# Deployment (3/3)

- Implementation builds on
  - extensible Fractal ADL toolset to add abstractions for bindings, resource requirements, and management elements
  - DCMS to create/bind/activate functional components and management elements
  - basic resource management for discovering/allocating appropriate resources
  - OSGI for packaging and installing components
  
- Work in progress
  - Adding support for complex, parameterised deployment workflows through Oz-based software framework

# Collaborative & federative services

- VO-aware File System (VOFS)
  - A mountable VOFS: a web of files
  - Allows aggregating files and storage; disconnected operation; policy-based access control
  - Supports shared workspaces
- Semantic Store
  - *Telex* middleware takes application semantics into account and leverages VOFS for persistency.
  - Demonstrator: shared calendar application
- Execution service
  - Scheduling and managing to completion work units
  - Arbitration of resources among several applications
  - Demonstrator: a movie format convertor

# Inter-VO Services

- Information services
  - Semantic-based resource/service discovery
- Auction-based markets for resources
  - Operate across VOs
  - Finding and negotiating external resources traded at marketplaces
  - Trading resources
  - Framework for developing new market mechanisms
  - Includes market information service and currency management system

# Summary

- VO management in Grid4All provides basic support for membership, security, resource management, and deployment
  
- Main features
  - Deployment of complex applications through high-level descriptions and automatic resource selection
  - Decentralised, overlay-based implementation that enables reliable operation in dynamic environments
  - Fully componentised structure that promotes adaptability and extensibility
  - Basis for sophisticated resource management services and policy-driven, VO-wide management processes