DyMRA: Dynamic Market Deployment for Decentralized Resource Allocation

Daniel Lázaro, Xavier Vilajosana and Joan Manuel Marquès

Universitat Oberta de Catalunya

Introduction

- Virtual communities:
  - Formed by users with common goals that need to collaborate to achieve their objectives.
  - Managed as Virtual Organizations that need computational resources.
- These resources can come from one of the following sources:
  - a) resources phisically belong to the VO.
  - b) members may contribute resources to the community
  - c) users may join resources in a cooperative way that results in benefit of all the participants

Introduction

- While a VO may lack resources, other computers may have surplus resources.

  →***inter-VO resource allocation***

- Markets
  - allocate resources efficiently.
  - promote incentives to resource owners to provide or trade their resources.
  - a centralized approach.

Introduction

- We developed DyMRA,
  - a decentralized resource allocation system based on markets that allows inter-VO resource allocation.
  - specially designed for dynamic and peer-to-peer environments
    - dynamically reallocate resources and services that manage the overall system.
  - many local ad hoc markets are created at will and run as services within the VO.

Introduction

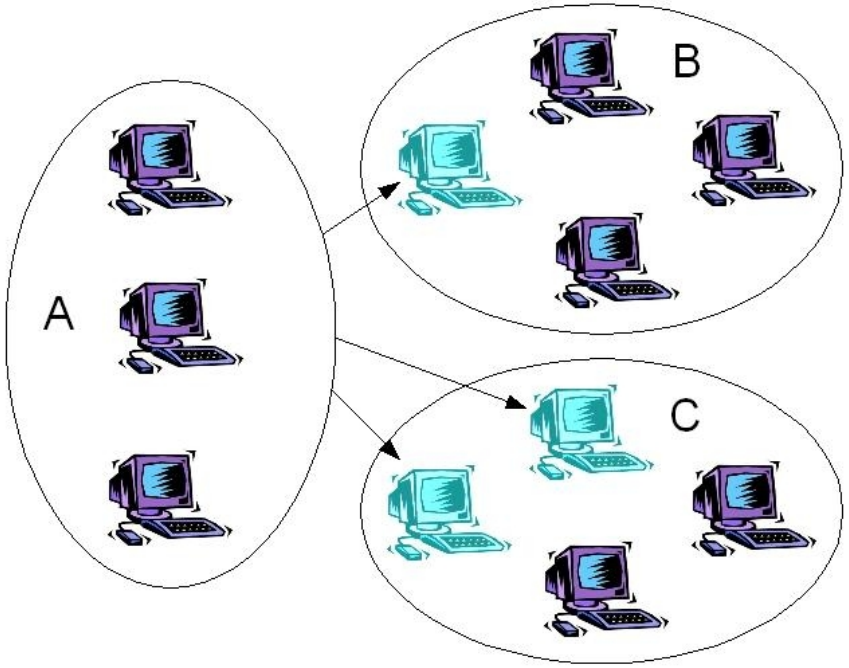
- DyMRA is built on top of LaCOLLA:
  - a peer-to-peer middleware that
    - allows a group of users scattered across the Internet to share resources in a cooperative manner
    - allows the deployment of stateless services using the resources provided by the members of the VO.
- DyMRA components are deployed as services in LaCOLLA middleware.

Scenario

- Three virtual communities:
  - A: online gaming community; members don’t usually contribute resources, but pay a fee to access the services.
  - B: scientific community; members contribute resources.
  - C: photo sharing community; members usually contribute resources.

Scenario

At a time, the VO A may require more resources than available to match the required quality of service.



VOs B and C, which have resources contributed by their users, may have a surplus.

B and C can sell access to their resources to A, obtaining benefits for their members (real money or something else).

Requirements

- Interoperability
- Group self-sufficiency
- Decentralization and self-organization
- Individual autonomy
- Market availability
- Location transparency

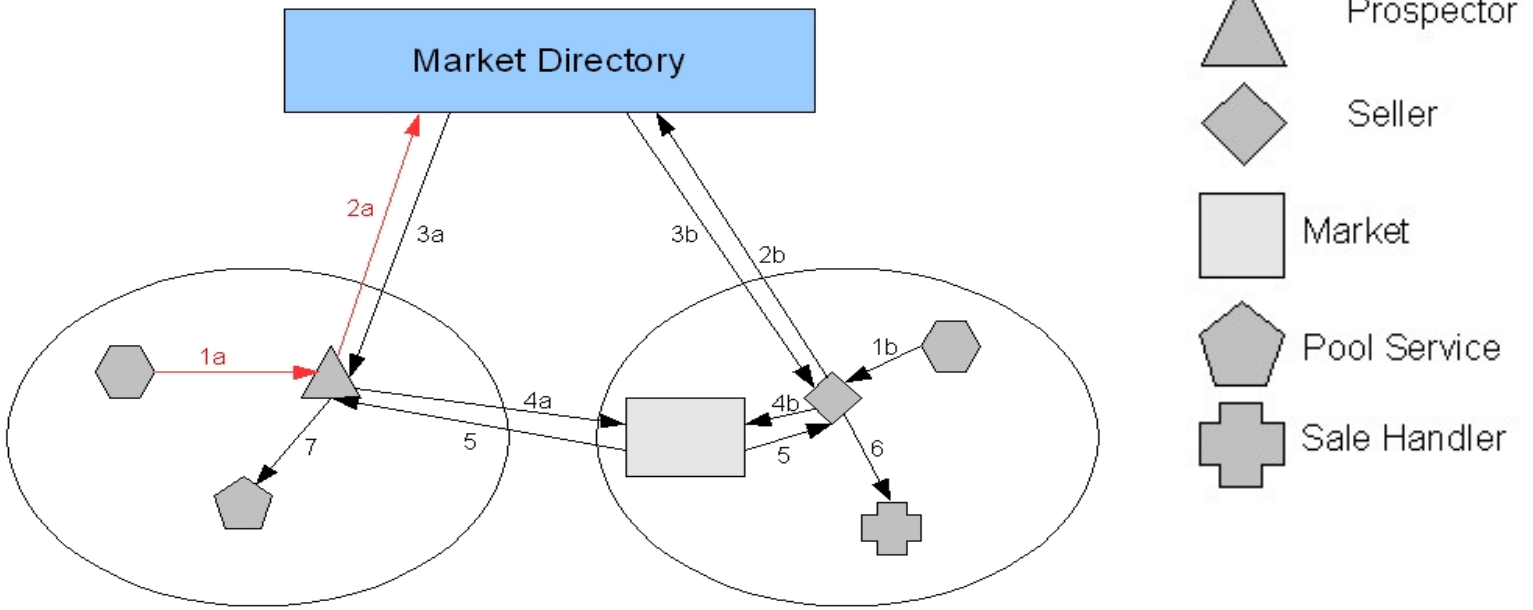
Architecture

- *Prospector*: finds suitable markets to obtain the desired resources.
- *Seller*: offers the aggregated surplus of resources of the VO in a suitable market.
- *Pool service*: controls access of the VO members to external resources.
- *Sale Handler*: controls external access to the resources of a VO.
- *Accounting service*: monitors the resources available in a VO, and decides when to start a trading process.
- *Market*: it mediates the trading of resources between VOs.
- *Market Directory*: Contains an index of existing markets and their locations.

Architecture

- All components except the Market Directory (MD) are deployed as services inside a VO using LaCOLLA.
- The MD is not part of a VO, but an external service which is known and can be accessed by all groups.
  - Possible implementations:
    - Centralized index.
    - DHT distributed among the VOs.

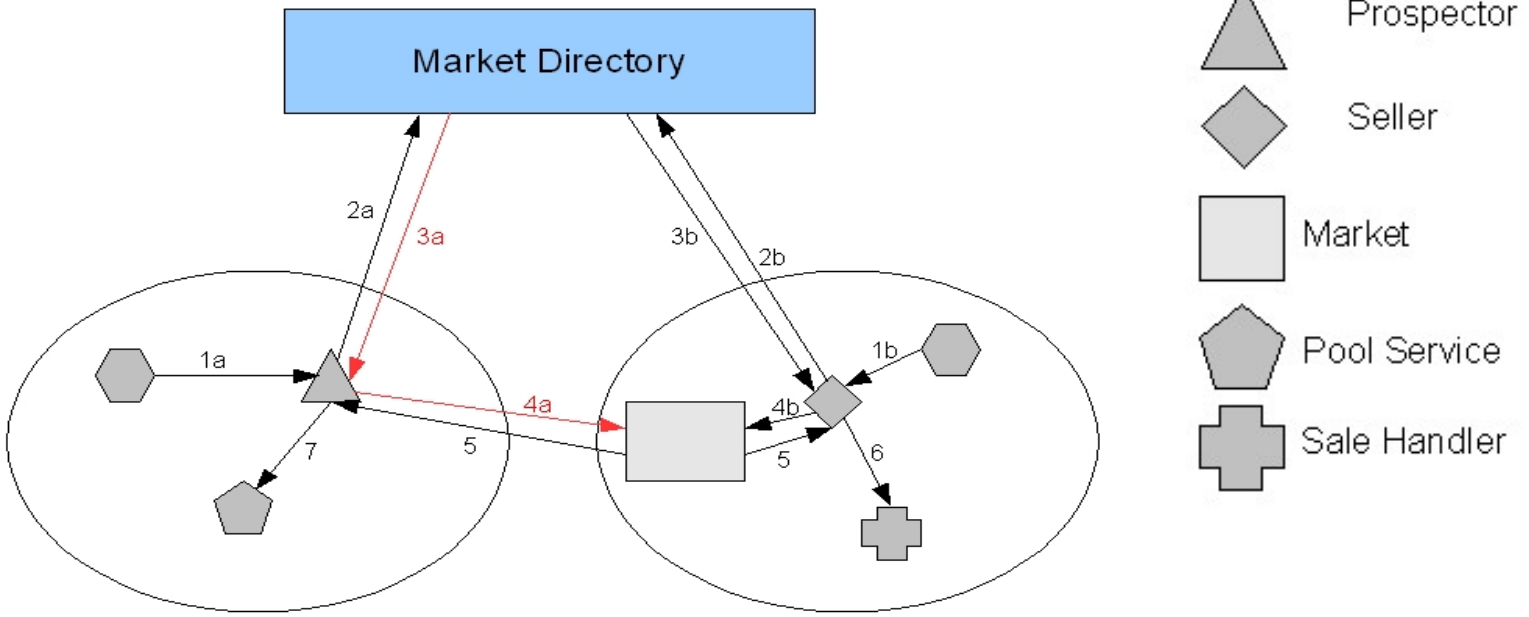
Trading process
Looking for a market



1a. The Accounting service detects that the resources are below a certain threshold and contacts the Prospector to acquire such resources.

2a. The Prospector looks for a suitable market in the Market Directory.

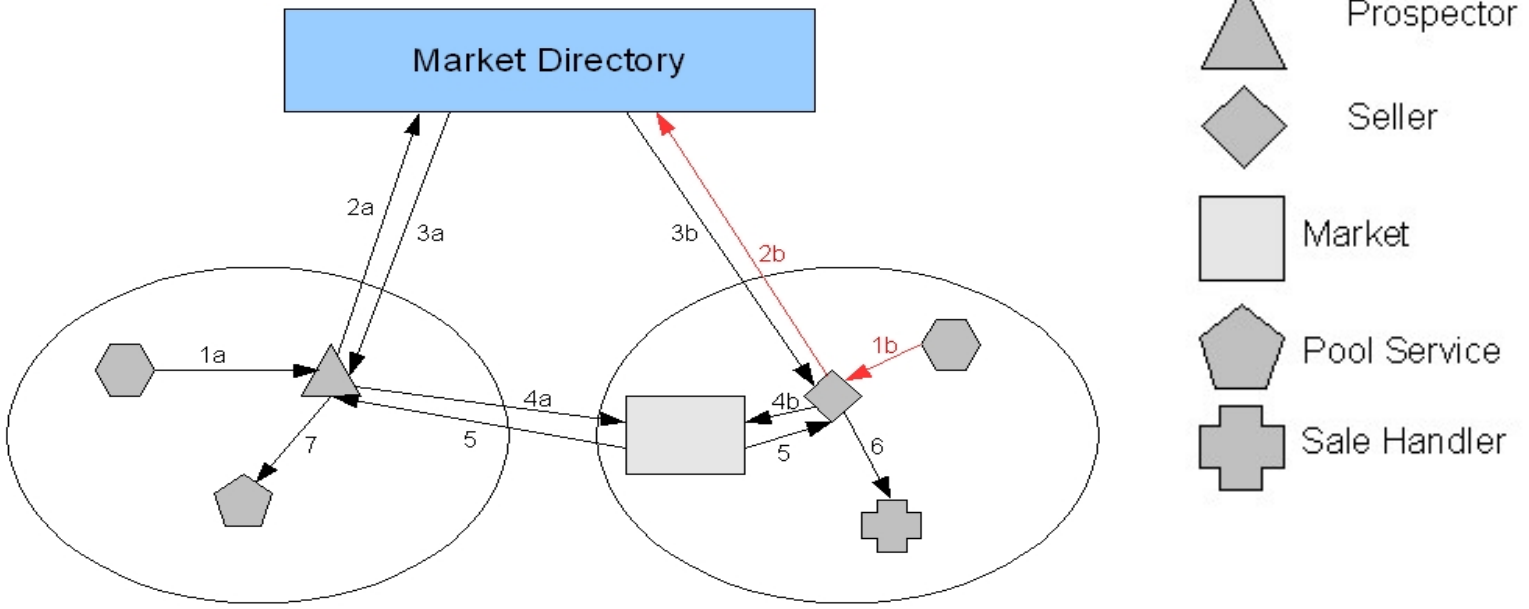
Trading process
Accessing the market



3a. The Market Directory sends the Prospector a list of markets which suit the specified needs.

4a. The Prospector chooses one of the markets of the list. If there is no suitable market, it creates a new one. The Prospector sends its bid.

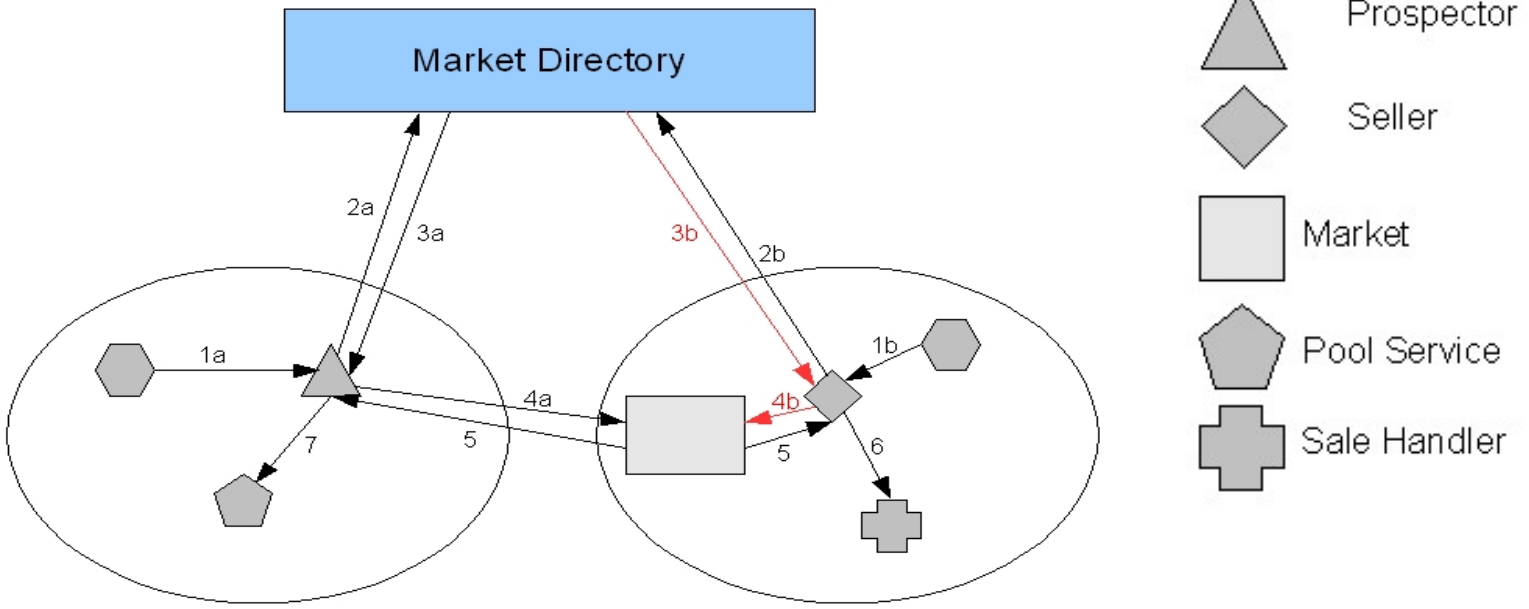
Trading process
Looking for a market



1b. The Accounting service detects that the resources are above a certain threshold and contacts the Seller to sell the surplus resources.

2b. The Seller looks for a suitable market in the Market Directory.

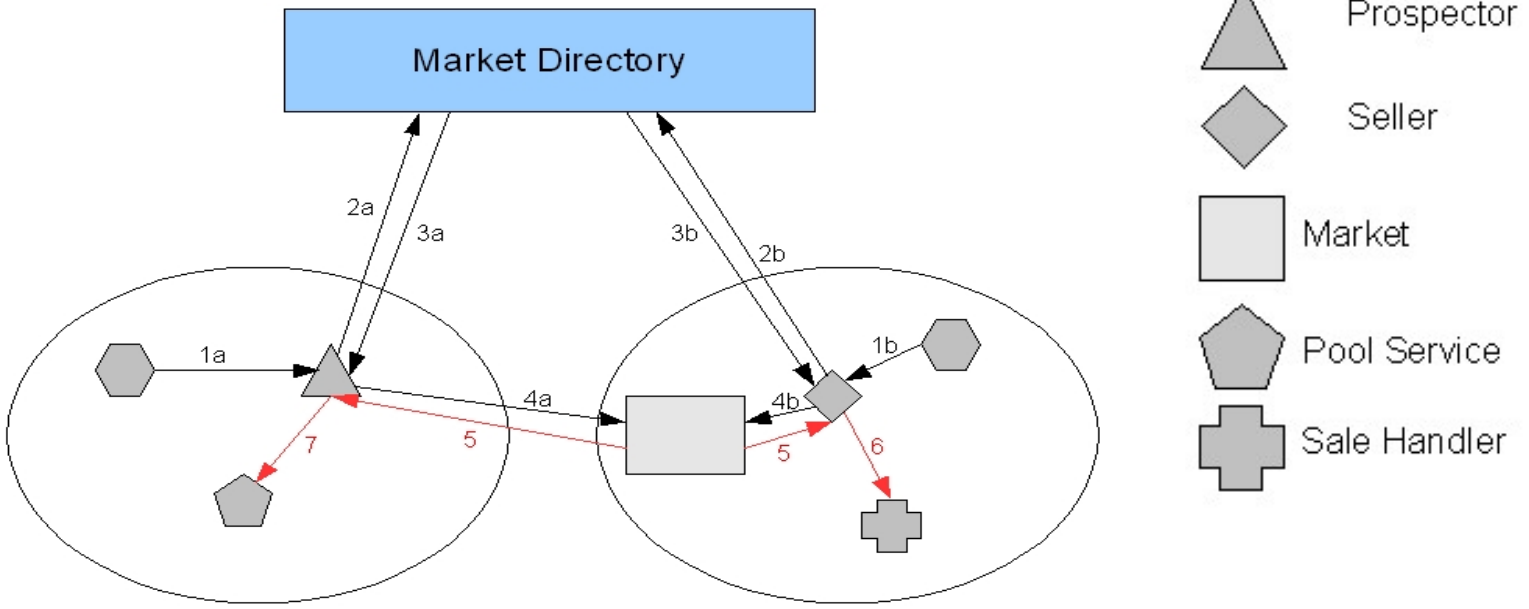
Trading process
Accessing the market



3b. The Market Directory sends the Seller a list of markets which suit the specified needs.

4b. The Seller chooses one of the markets of the list. If there is no suitable market, it creates a new one. The Prospector sends its bid.

Trading process
Making a deal

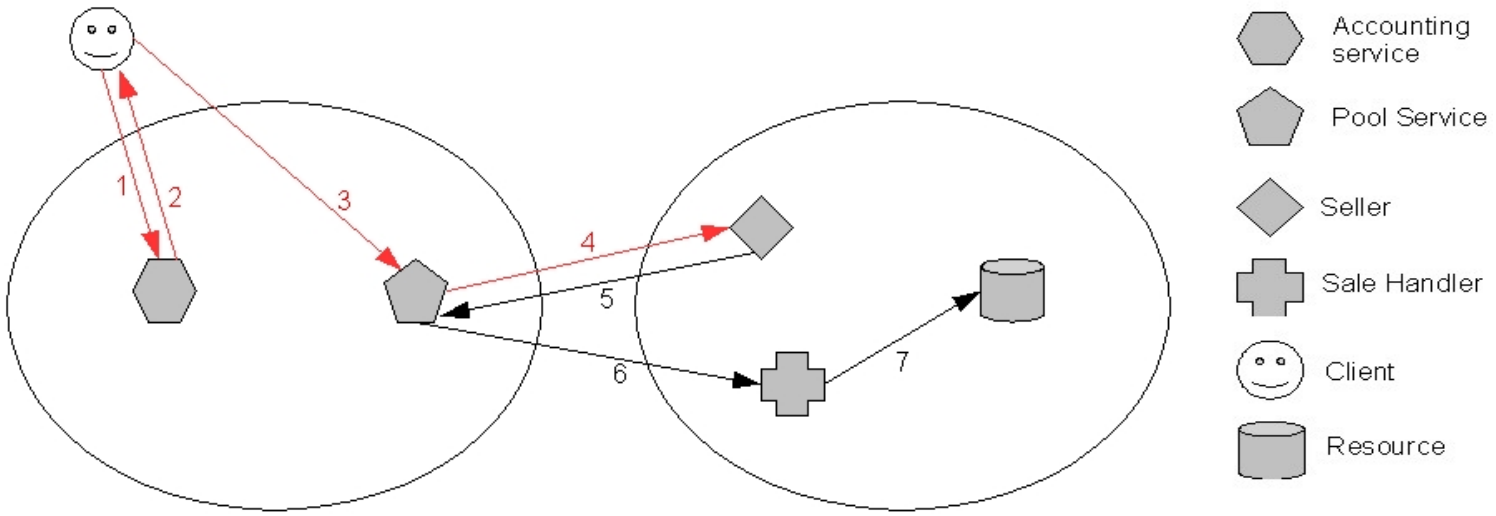


5. The market makes an agreement and notifies the sale to both the Prospector and the Seller.

6. The Seller starts a Sale Handler, which is deployed in its VO and mediates the use of the resources.

7. The Prospector informs the Pool about the resources bought.

Access process



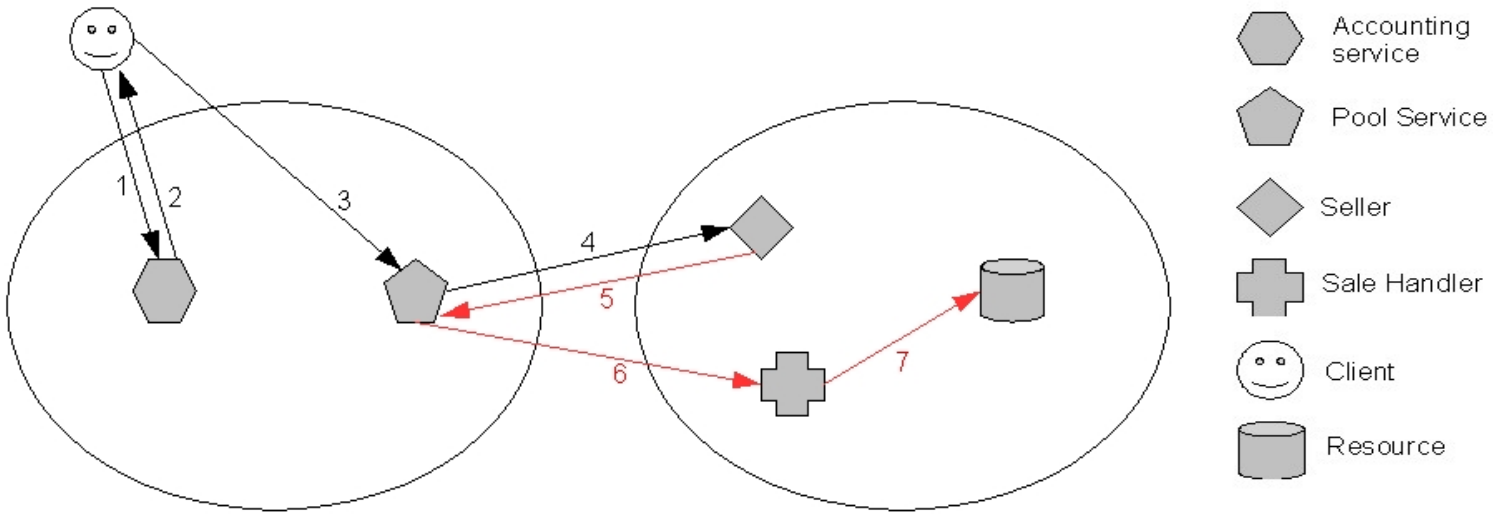
1. When a client needs resources, it contacts the Accounting service.

2. The Accounting service checks the resources currently available to the VO and tells the client which ones to use.

3. If the client must use external resources, it contacts the Pool.

4. The Pool service chooses which of the external resources should be used, and contacts its corresponding Seller.

Access process



5. The Seller tells the Pool service the location of the Sale Handler that manages the specific agreement.

6. The Pool service contacts the Sale Handler, according to the conditions of the agreement.

7. The Sale Handler checks that the request of the Pool service does not violate the conditions of the agreement. After this, it uses the resources of the VO to fulfill the request of the Pool service.

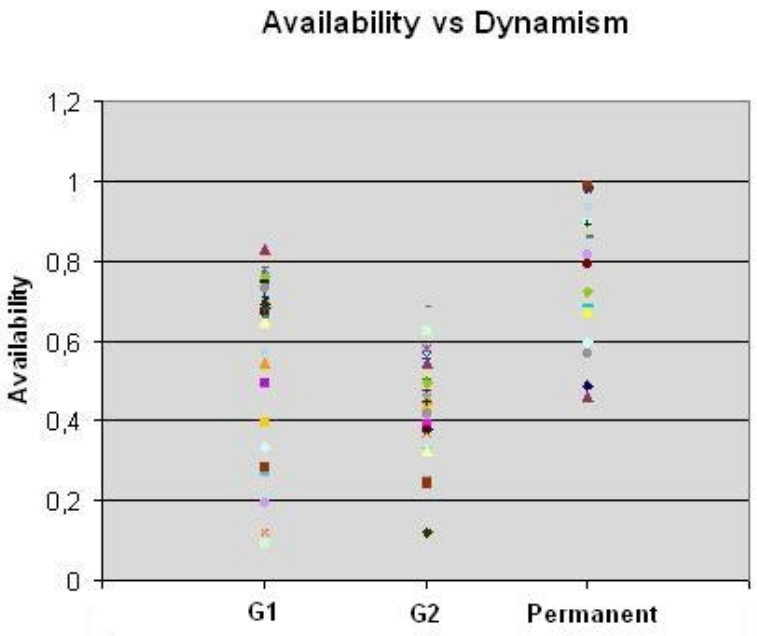
Validation

- We implemented a prototype of the proposed architecture to test its usefulness.
  - The Prospector, Seller, Pool, SaleHandler and the Market are deployable services over the LaCOLLA middleware.
  - The Market was implemented as a double auction protocol that enables buyers and sellers to submit bids for multiple units of a single resource.
- The MarketDirectory has been implemented as a centralized index. It stores pairs of < key, value > where the key identifies the type of traded resource and the value refers to the location of the market where it is traded in.

Validation

- The objective of our test is to validate the trading process, focusing on availability.
- We executed the following components and processes:
  - one process which periodically tried to buy resources.
  - one process which periodically tried to sell resources.
  - Prospector, Pool and Seller services were active inside the VO.
  - the Market Directory was available in a static location.
  - markets were deployed as services inside the VO, and can be:
    - created on demand. This reduces the perceived availability, as when a market needs to be created it is counted as a failed attempt.
    - permantently active. This gives a better availability, but at the cost of spending resources for these markets even when they’re not being used.

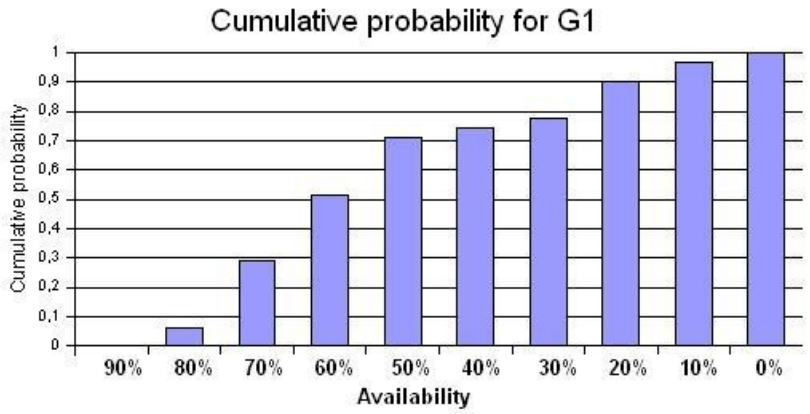
Validation

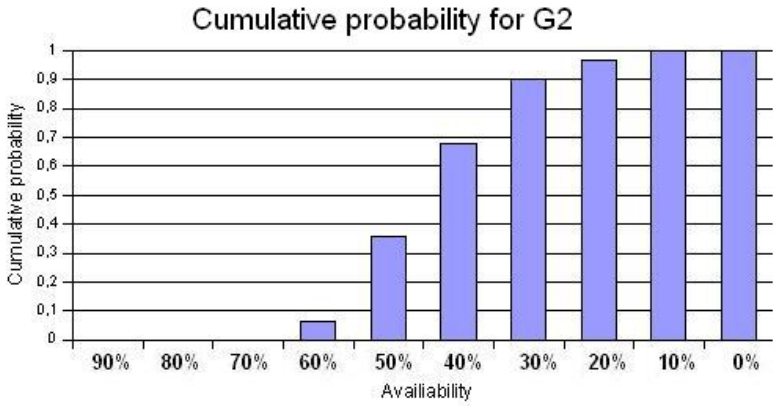


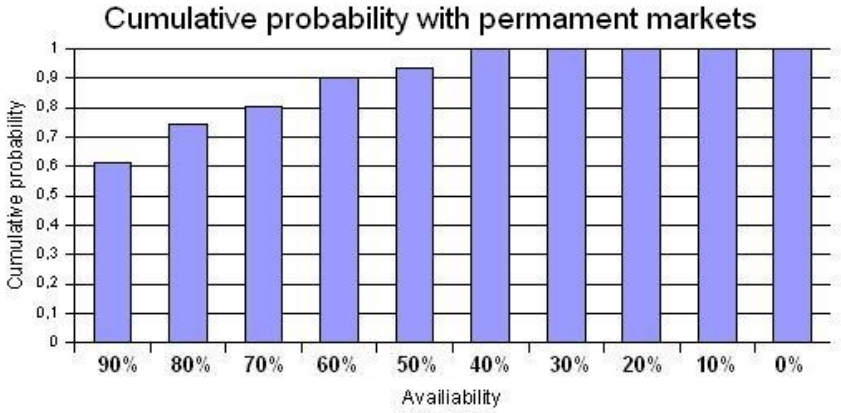
Availability vs. level of dynamism

- We tested two configurations with different levels of dynamism (G1 a lower level of dynamism than G2) with markets created on demand, and configuration G1 with permanent markets.
- As expected, G1 has a better availability than G2.
- Permanent markets substantially increase availability.

Validation


Cumulative probability for G1
Cumulative probability
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Availability


Cumulative probability for G2
Cumulative probability
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Availiability


Cumulative probability with permament markets
Cumulative probability
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Availiability

Conclusions

- We’ve presented DyMRA, a framework for inter-VO resource allocation that uses centralized markets in a decentralized environment without introducing bottlenecks or single points of failure.
- We’ve presented the preliminary results of evaluating our proposed architecture.
- Our future work includes:
  - the complete development of the DyMRA components, such as a decentralized Market Directory
  - further defining the set of mechanisms to control the access to external allocated resources.
  - considering the duration of the allocations of resources (lease times), to allow the application of our framework in a real environment.