EMI Service Registry (EMIR)

JRA1, Infrastructure

Shiraz Memon (FZJ), Ivan Marton (NIIF), Gabor Szigeti (NIIF)

Outline

- Motivation
- Goals
- EMIR Highlights
- Architecture
- Information Model
- REST API
- Security
- EMI Integration Scenario
- Performance Analysis
- Implementation Remarks
- Status report

Motivation

- No common Index Service
  - Site/Top level BDII
  - ARC ISIS
  - UNICORE Registry
- No common “service” to store the service information
  - Various Service record formats
  - Service publish and discovery interfaces

Goals

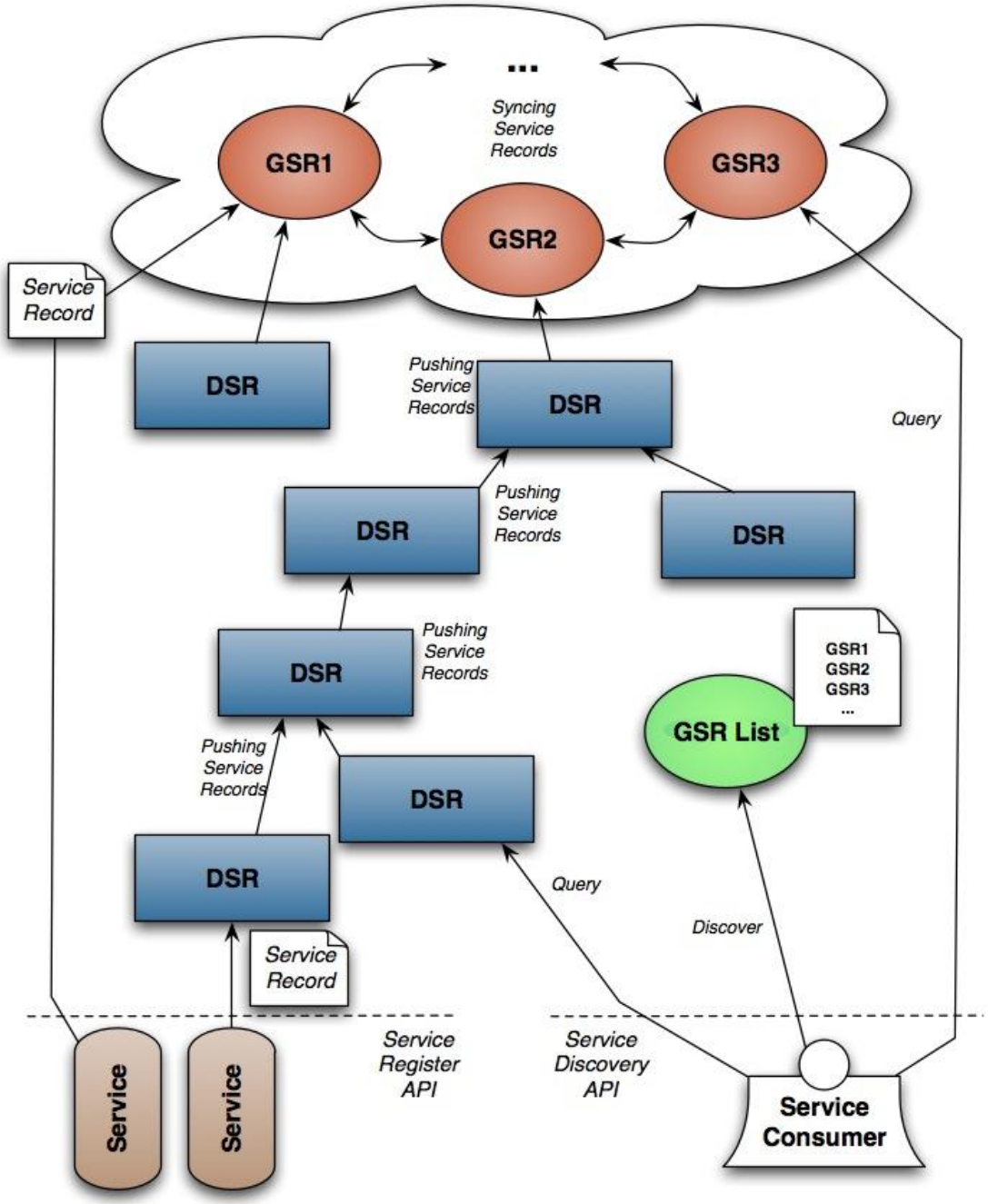
- Common Index Service to Publish and Discover the Services
- Unified Information Model
- Support for federations (Natural to existing Infrastructures)
  - Inter-Federation Lookups
  - Multi-Federation service discovery
- Common Security Model
- Scalable and Performance

Unified Index Service: EMI Service Registry (EMIR)

- Discover all the EMI Services (ARC, gLite, UNICORE, dCache)
- Hierarchies and Federations
- Common Access Point to Service Discovery (even at each hierarchical level)
- REST-ful Interface to Publish and Discover the Services
- Service Information: GLUE 2.0 Based Service Entity

System Architecture

EMI INFSO-RI-261611


...
Syncing
Service
Records
GSR1
GSR2
GSR3
Service
Record
DSR
Pushing
Service
Records
DSR
Query
DSR
Pushing
Service
Records
DSR
DSR
Pushing
Service
Records
GSR1
GSR2
GSR3
...
GSR List
Pushing
Service
Records
DSR
DSR
Query
Discover
Service
Record
Service
Register
API
Service
Discovery
API
Service
Service
Service
Consumer

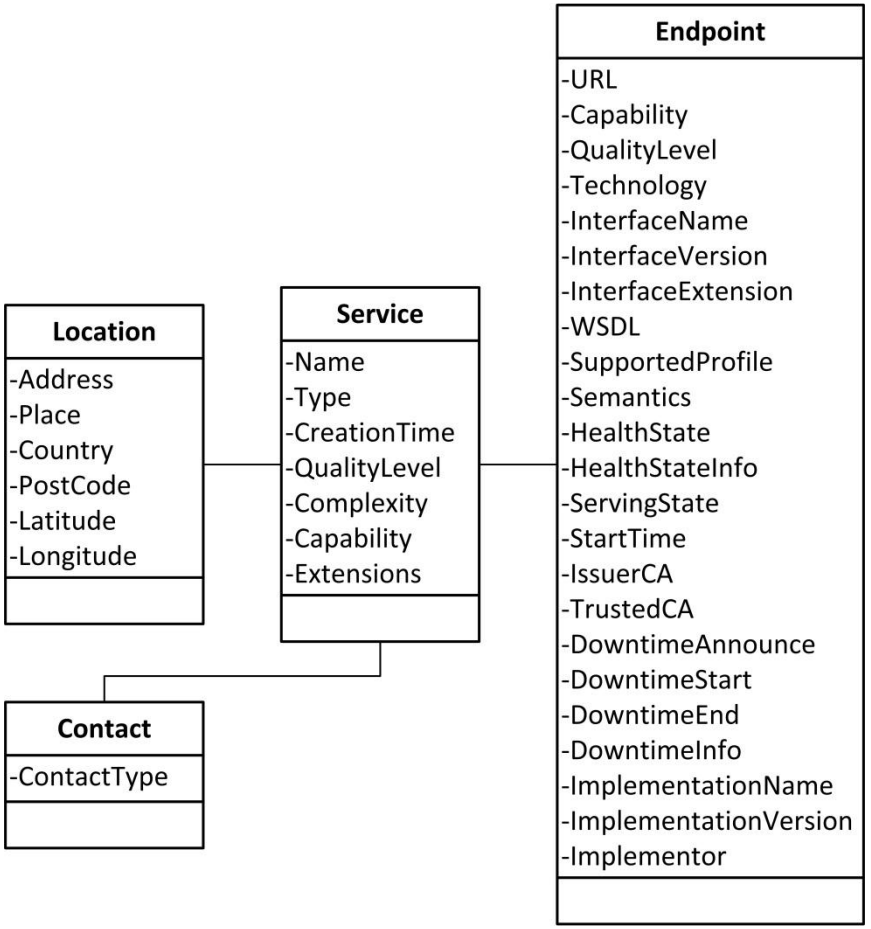
EUROPEAN MIDDLEWARE INITIATIVE

Service Information Model

EMI INFSO-RI-261611

GLUE 2.0: Information Model

- Service
  - General Information
  - Meta-data (Last-update, Publisher/Owner, Lifetime)
- Endpoint
  - Access Information
- Location
  - Physical Location
  - Geo-Coordinates
- Contact
- Additional Attributes (can be added on the fly)
- TTL for every Service Entry



JSON Rendering

- Developed within EMI
- Used for Publishing and Discovering the Services


```
[
 {
  "Service_Name": "EMI Execution Service",
  "Service_CreationTime": {"$date": "2011-10-16T11:47:24Z"},
  "Service_Type": "eu.emi.es",
  "Service_Capability": ["activity submission", "activity creation"],
  "Service_QualityLevel": "production",
  "Service_Endpoint_URL": "http://eu-emi.eu/emi-es",
  "Service_Endpoint_Technology": "SOAP over HTTP",
  "Service_Endpoint_InterfaceName": "ActivityManagement",
  "Service_Endpoint_InterfaceVersion": ["1.0" ],
  "Service_Endpoint_WSDL": "http://eu-emi.eu/emi-es ?wsdl",
  "Service_Endpoint_HealthState": "ok",
  "Service_Endpoint_HealthStateInfo": "it is OK, I can see it",
  "Service_Endpoint_ServingState": "production",
  "Service_Endpoint_StartTime": {"$date": "2011-07-21T11:47:24Z"},
  "Service_Endpoint_DowntimeAnnounce": {"$date": "2011-07-21T11:47:24Z"},
  "Service_Endpoint_DowntimeStart": {"$date": "2011-07-21T11:47:24Z"},
  "Service_Endpoint_DowntimeEnd": {"$date": "2011-07-21T11:47:24Z"},
  "Service_Endpoint_QualityLevel": "production",
  "Service_ExpireOn": {"$date": "2011-07-21T11:47:24Z"},
  "Service_UpdateOn": {"$date": "2011-07-21T11:47:24Z"},
  "Service_OwnerDN": "CN=Shiraz Memon,O=FZJ,OU=JSC"
 }
]
```

Querying the XML documents (1/2)

```
< !--   Document  Start   -- >

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:QueryResult count="1" xmlns:ns2="http://www.eu-emi.eu/emiregistry" xmlns="http://schemas.ogf.org/glue/2009/03/spec_2.0_r1">
  <Service BaseType="Entity" CreationTime="2011-07-21T13:47:24.111+02:00" Validity="12313">
    <Name>ComputingService</Name>
    <Extensions/>
    <Type>job-management</Type>
    <QualityLevel>production</QualityLevel>
    <Location>
      <Address>A Street 1</Address>
      <Country>Germany</Country>
      <PostCode>53119</PostCode>
      <Latitude>53.3</Latitude>
      <Longitude>4.0</Longitude>
    </Location>
    <Contact>
      <Detail>http://contactlink</Detail>
      <Type>sysadmin</Type>
    </Contact>
    <Contact>
      <Detail>http://contactlink</Detail>
      <Type>developer</Type>
    </Contact>
```

Querying the XML documents (2/2)

```
      <Endpoint BaseType="Entity" CreationTime="2011-07-21T13:47:24.111+02:00">
        <Name>ComputingService</Name>
        <URL>http://eu-emi.eu/emies</URL>
        <Capability>Activity Management</Capability>
        <Capability>Activity Submission</Capability>
        <Technology>technology</Technology>
        <InterfaceName></InterfaceName>
        <InterfaceVersion>1.0</InterfaceVersion>
        <HealthState>ok</HealthState>
        <HealthStateInfo>Its OK</HealthStateInfo>
        <ServingState>production</ServingState>
        <StartTime>2011-07-21T13:47:24.111+02:00</StartTime>
        <DowntimeAnnounce>2011-07-21T13:47:24.111+02:00</DowntimeAnnounce>
        <DowntimeStart>2011-07-21T13:47:24.111+02:00</DowntimeStart>
        <DowntimeEnd>2011-07-21T13:47:24.111+02:00</DowntimeEnd>
      </Endpoint>
    </Service>
</ns2:QueryResult>

< !--    Document End     -- >
```

Database Technology

- MongoDB: NOSQL database
- Supports Native JSON format
- Dynamic Indexing
- Replication
- Horizontal scalability with sharding (Similar to Google's BigTable)
- Available for most of the major OSs

- REST-ful API: Publishing & Querying

EMI INFSO-RI-261611

Service Registration

- /serviceadmin
  - Register

  -> HTTP POST - /serviceadmin

  - Body: JSON

  <- OK
  - Read

  -> HTTP GET – /serviceadmin?Service_Endpoint_URL=service_url

  <- JSON document
  - Update (full replace)

  -> HTTP PUT - /serviceadmin

  - Body: JSON

  <- OK
  - Delete

  -> HTTP DELETE - /serviceadmin?Service_Endpoint_URL=service_url

  <- OK

Service Querying

Showing all the services

- /services
  - /query

  -> HTTP GET

  <- OK, JSON

Returns the 100 services, skipping the first 20

- /query?limit=100&skip=20

  -> HTTP GET

  <- OK, JSON

Returns 100 services skipping the first 20 records with specific name

- /query?limit=100&skip=20&ServiceName=ComputingService

  -> HTTP GET

  <- OK, JSON

Returns 100 services skipping the first 20 records, filtered by “name”, as GLUE 2.0 XML document

- /query.xml?limit=100&skip=20&ServiceName=ComputingService

  -> HTTP GET

  <- OK, XML

Interacting with EMIR

- Register/Update

Header Method Data

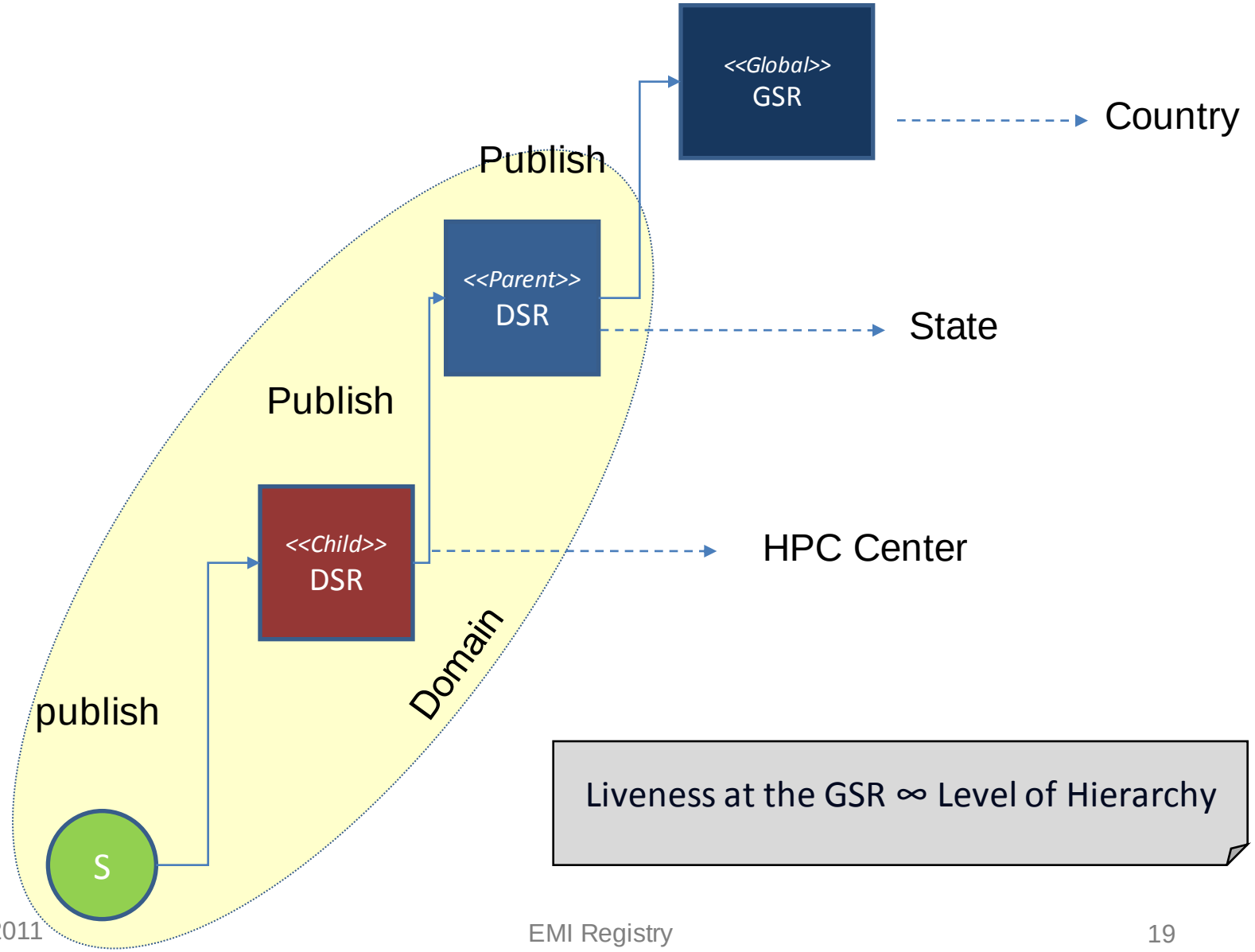
curl -v -H "Content-Type: application/json" -X POST (PUT) -d @info.json /
http://emiregistry1.grid.niif.hu:54321/serviceadmin

Registry URL

- Query
  - curl -v -X GET http://emiregistry1.grid.niif.hu:54321/services/query
  - curl -v -X GET http://emiregistry1.grid.niif.hu:54321/services/query.xml
- Delete
  - curl -v -X DELETE http://emiregistry1.grid.niif.hu:54321/serviceadmin?Service_Endpoint_URL=1

Federations

Hierarchical Structure

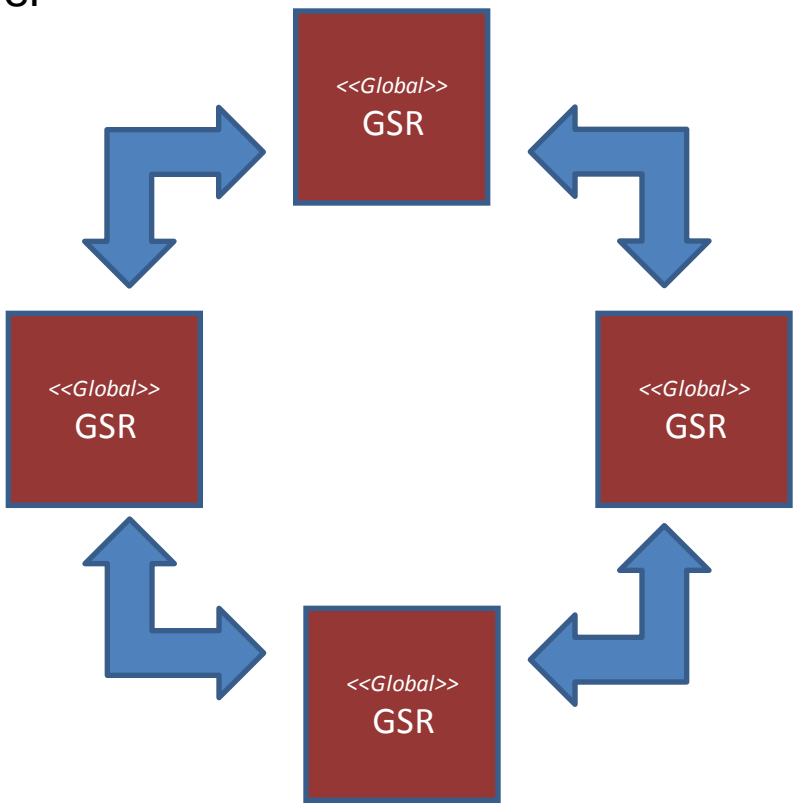


EMIR Hierarchy

- Static Construction
- Events (changes in the registry) are sent directly to the Parent as they occur
- Failure 1: Parent Goes Offline
  - Caching the occurred events in between
  - Send as soon as the Parent comes online
- Failure 2: Child found a Parent (got newly added)
  - Send whole local information to the parent in Chunks

Global Registries

- Replicated Registries
- Data Synchronization at Global level
- Why?
  - Global Service Discovery
  - Failover
  - Robustness
  - Scalability
  - Load Balancing
- Offers query interface
- Where to find GSRs?
  - Publicly known list



EUROPEAN MIDDLEWARE INITIATIVE

Security

EMI INFSO-RI-261611

Goals

- ARGUS based access control
- Attribute Source: VOMS-SAML
- EMI Common Authentication Library (CAL)

Currently Supported

- SSL/TLS based authentication – requires valid X.509 credentials
- Attribute Source: User Map File
  - Defines "Users" with associated "roles" (Service Provider, Service Consumer/User, Administrator)
- Access Control
  - Access Control Policies
    - e.g. allows a user with "serviceowner" role can register and manage/control his registrations
    - defined in a XML file containing raw XACML policies

Performance Analysis

EUROPEAN MIDDLEWARE INITIATIVE

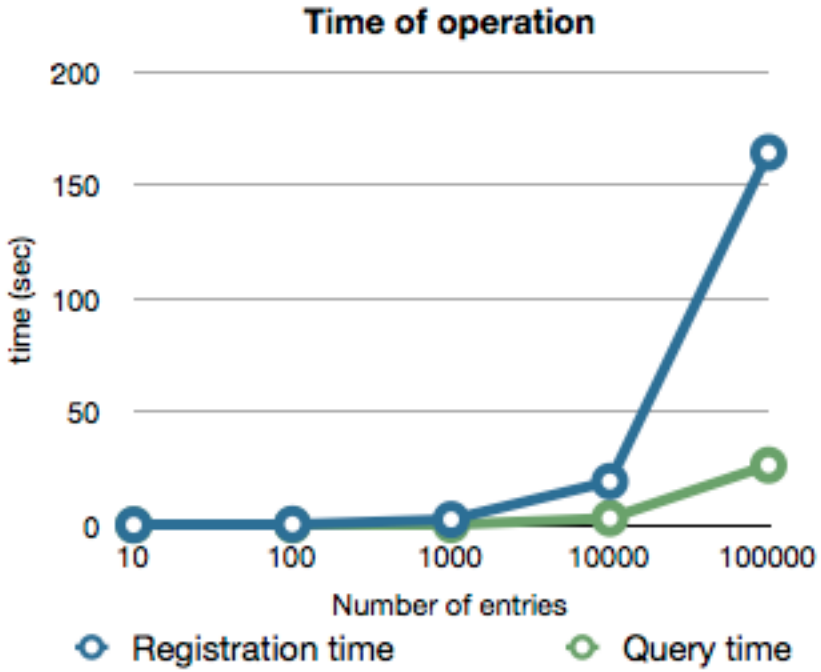
System Environment

- Single Registry Instance without Security
- 1 GB JVM Memory
- 4 Cores
- ~3 GB for MongoDB

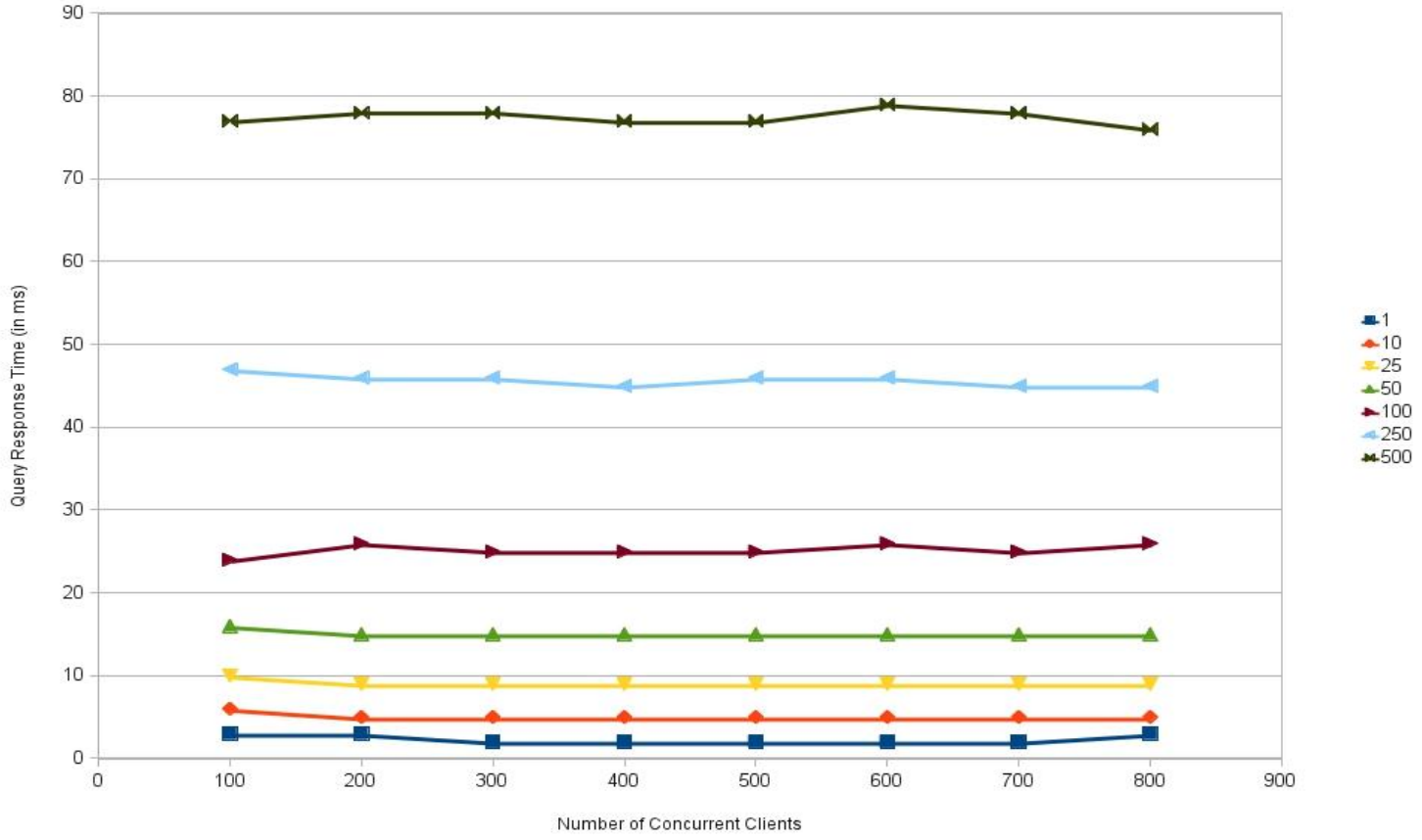
EMI INFSO-RI-261611

Performance Analysis:

Single Threaded Response Time (Without Security)



Performance Analysis: Multithreaded Response Time



Implementation Remarks

- JAVA 6
- Maven 2
- JAX-RS (Java Standard for REST-ful WS)
- Jetty 7 Web Server
- MongoDB

Status Report

- Implementation
  - DSR:
    - Provided: Most of the basic features already implemented
    - Missing: Argus and/or VOMS-SAML support
  - GSR: P2P synchronization/replication will be available by EMI 2
  - Clients: ARC, UNICORE (under development)
- Documentation
  - Technical Document has been revised to reflect the implementation
  - Comparative Survey Document is progressing
  - Development Plan on EMI TWiki
- Packaging
  - EPEL distribution, will be available by EMI 2
  - Source and Binary Tarball
  - ETICS Integration

Further Extensions

- Service Registrants from Service providers
- SAGA-SD API
- GUI to administer and browse registry(s)
- Triggers, etc…

Links

- Twiki: https://twiki.cern.ch/twiki/bin/view/EMI/EMIRegistry
- Source Code on GitHub (public repo.): https://github.com/eu-emi/emiregistry

Questions ???

EMI INFSO-RI-261611

Thanks!

EMI is partially funded by the European Commission under Grant Agreement RI-261611