



EMI Registry

JRA1, Infrastructure

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

Overview

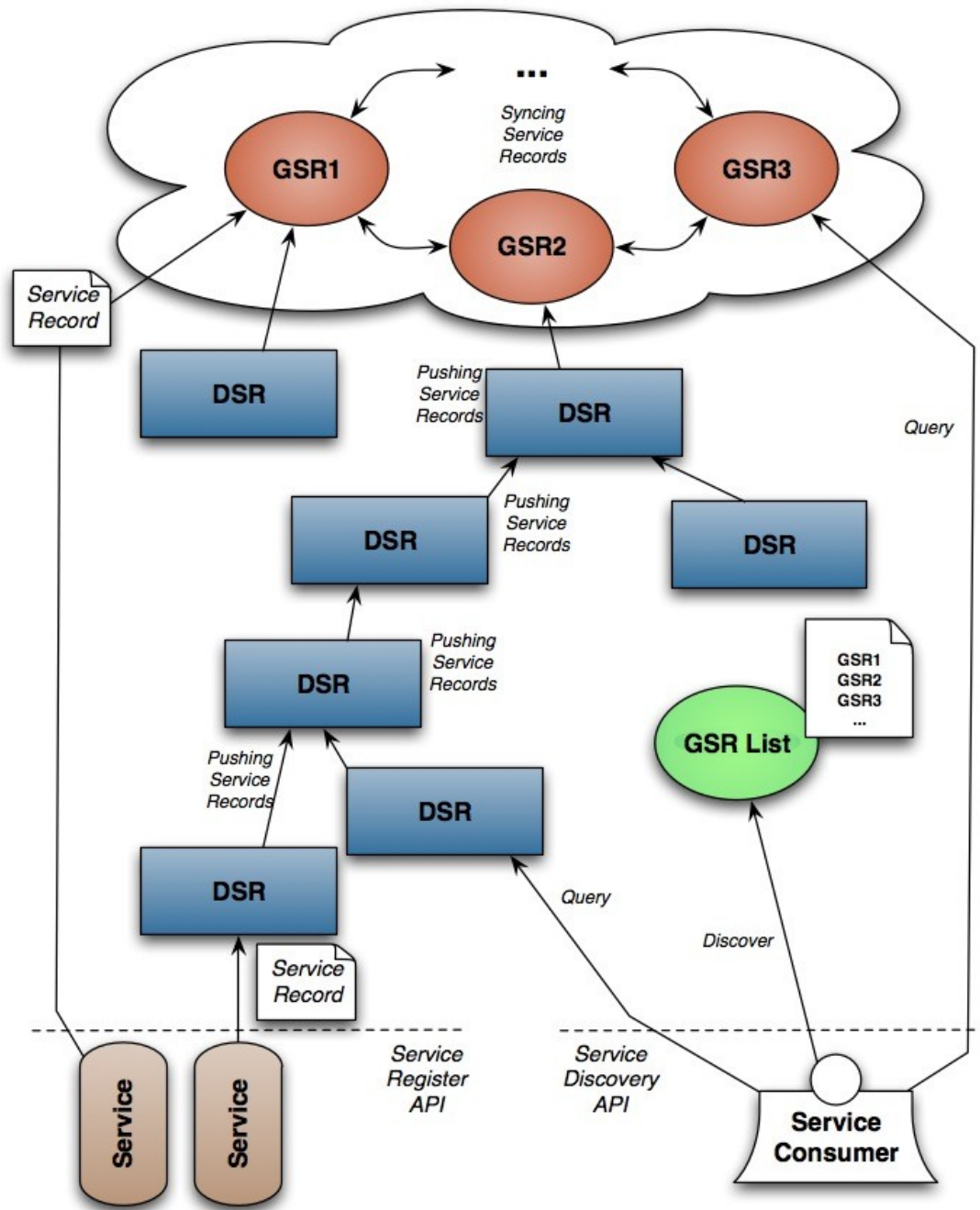
- Service discovery
- Requirements
- EMI Registry design plan
- The answers
- Status report

Service discovery

- Available solutions
 - GOCDB, ARC LDAP Infosys, Unicore registry, (ARC ISIS – Information System Indexing Service)
- Used by
 - clients and (other) services as well
- Global Scope
- Various type of information
- Different protocols, security, etc.

Requirements

- Discover the existence of services
- Scalable
- Robust
- Federated (!)
- Other requirements may appears:
 - Secure
 - Up-to-date
 - Reliable
 - ...



Interfaces

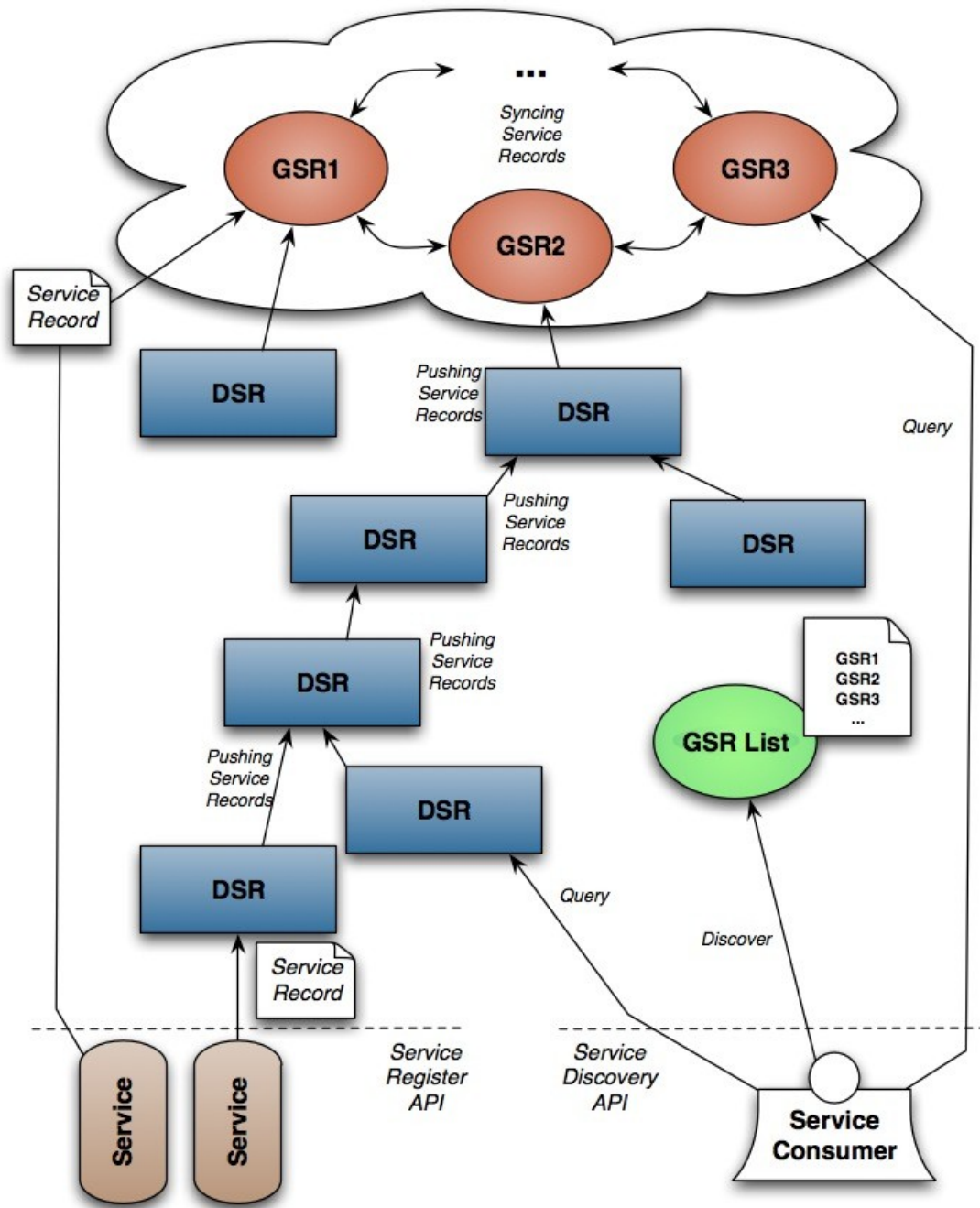
- 3 different interfaces
 - Service register interface
 - Service discovery interface
 - Intra-registry (synchronisation) interface
- Need to be...
 - Standardized
 - Lightweight
 - Easy-to-implement and adapt

Service examples

- A-REX
- WMS
- CREAM
- UAS
- OSGA-BES
- SRM
- EMI ES
- ARGUS
- StoRM SE
- VOMS
- SLCS
- STS
- Hydra
- SCAS
- gLExecStoRM SE
- ARC gridftp-
job-plugin interface
- ARC gridftp server
- FTS
- DPM
- LFC
- dCache server

Service Consumer examples

- CLI's
 - provided by the partners: ARC, dCache, gLite, Unicore has also different set of command line tools
 - graphical user interfaces
 - portals
 - thick clients
 - etc.
 - libraries (like the arclib)
 - services(!)



Internal issues

- Information model
 - Highly based on GLUE2
 - At the beginning as small amount of data as possible (Service type, URL, and some more)
 - can be extended later
 - Static information (depends on the point of view) – during the lifetime of a service
- Security
 - Rely on ARGUS Service
 - Using XACML profiles

Federated model

- The DSRs (Domain Service Registries) can be organised into a tree structure following the borders of the federation.
- The registrations (belong to the federation) will be propagated above
- Consumers can prefer “local” results
- Policy decisions could be also propagated signed by a trustworthy federation member.

Global Service Registry

- The top level of the hierarchy
- Fully replicated databases
- Also implements the Query interface
- Robust solution for storing and querying information
- Receives reliable registration - just from the DSRs.
- Synchronize using peer-to-peer solution between each other
- Addresses can be known from GSR list

The answers

- Discover the existence of services –
Providing interface for query service database
- Scalable – Dynamically extendable structure
- Robust – Replicated database
- Federated (!) – supported by design
- Other requirements may appears:
 - Secure – ARGUS, XACML
 - Up-to-date – Aliveness checking
 - Reliable – Controlled information from controlled sources

Status report

- Has a detailed design plan with some open questions, that are also recently discussed.
- Managed to decide some technical issues:
 - Which programming language (Java), database backend (MongoDB), kind of interface (RESTful), or type of message (JSON) will be used.
- Implementation has been started

Status report

- A kind of “Agile” development is in progress
- Short prints are defined
 - 1st: DSR functionalities, working interfaces, integrated security
 - 2nd: Building of internal hierarchical structure
 - 3rd: Implementing peer-to-peer structure, and additional features like aliveness checking, etc.



Thank you!

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