Spitfire use cases within the EU DataGrid

Leanne Guy

WP2 — CERN

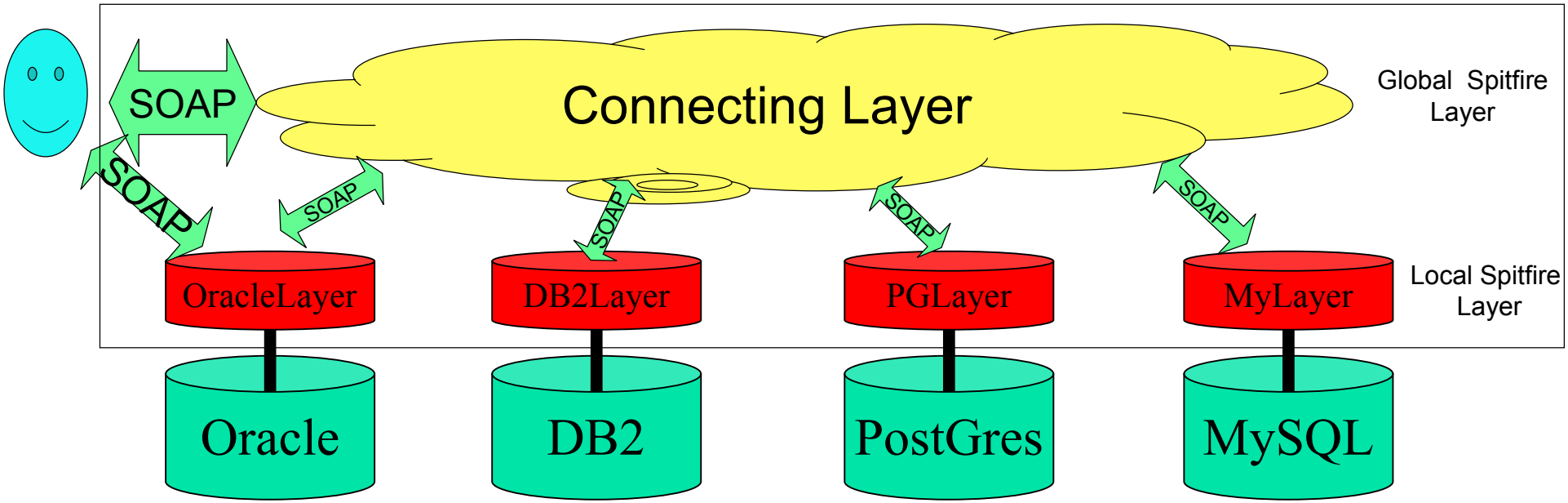
Leanne.Guy@cern.ch http://cern.ch/leanne

Spitfire: *A Relational DB Service for the Grid*

- Spitfire is the MetaData service provided by WP2 in the context of the EU-DataGrid Project
  - complex, distributed, database independent catalogue
- The client is decoupled from the backend RDBMS by a mediator
  - enables ease of use
  - interoperaility, performance, plugability
- providing access to many different database implementations through generalised commands issued by a remote client.
- The SQLDatabaseService is a grid enabled front end for reading and writing from/to a relational database.
- Communication over HTTP(S)

Spitfire architecture

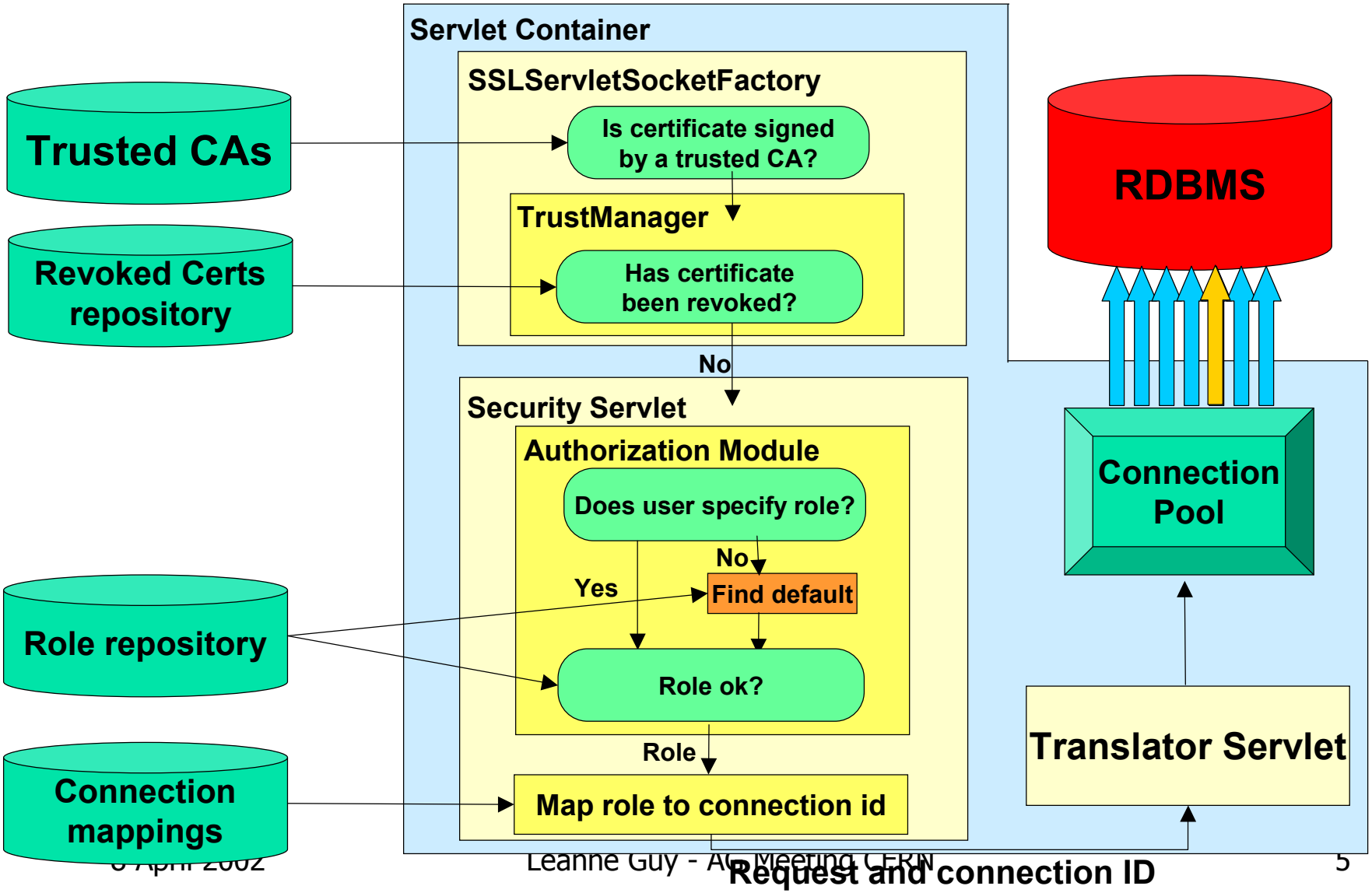
- XSQL Servlet as one access mode for ‘simple’ web access
- Web/Grid Services Paradigm
  - SOAP & WSDL interfaces
  - JDBC interface to RDBMS
- Pluggability and extensibility
- Atomic RDBMS is always consistent
- No local replication of data
- Role-based authorization



Local layer functionality

- DB Administration
  - Create Database
  - Delete Database
  - Create Table
  - Drop Table
- Role Administration
  - Create Role
  - Delete Role
  - Update Role
- DB Information
  - Quotas
  - Memory, Disk space
  - User Info
  - Schemata
- Core Functionality
  - Insert
  - Update
  - Delete
  - Select
- Timestamps
  - Set Table Timestamp
  - Set Row Timestamp
- Connections
  - Open Dedicated Connection
  - Close Connection

Spitfire security mechanism



RM metadata catalogue

Replica Catalogue

- stores PFN <-> LFN mappings
- metadata for files (PFNs) on the local storage element
  - file size
  - file status : durable, volatile, permanent

Metadata Catalogue: RepMeC

- instance of SpitFire
- stores attributes of LFNs independent of any associated PFNs
  - location of master and secondary copies for an LFN
  - VO specific data, e.g replication policy for LFNs
  - high level access control
- RepMec API
  - provides methods to define, store and extract metadata on replicas

All access to RepMeC is via the Replica Manager to ensure consistency

Query optimisation service

Optor: WP2 Query Optimisation service

- Goal: Maximise the throughput for a set of job
  - locate the “best replicas”, “best CE”
- requires a file prefetch catalogue
  - store PFNs of prefetched files for a given job
  - used by getBestFile() to determine if the best PFN for a job is
  - removal oftemporary files after a job finishes

Prefetch catalogue wil be implemented using SpitFire

- lists of prefetched PFNs
- plan for first implement by the end of May

http://ppewww.ph.gla.ac.uk/ScotGRID/code_doc/queryopt/simulation/Optor/index.html

Service index

Problem:

- Want to maintain and query dynamic and timely information about Grid services
- How should a database node maintain information populated from a large variety of unreliable, frequently changing, autonomous and heterogeneous remote data sources - without sacrificing reliability, predictability and simplicity?

Use a Service Index:

- A service description is a set of interfaces with associated operations. that define how it is bound to one or more network protocols and endpoints (see OGSA).
- Search for descriptions of resource brokers, computing elements, storage elements, etc.

There are no plans for any release prior to or for tesbed2

Service index (2)

Service Index uses Spitfire as it's storage solution to fetch and store sets of service descriptions

Link	Context	Type	Timestamps	Metadata	Content
http://repcat.cern.ch/getServiceDescription	Parent	Service	10,15,20,30	<owner name = "http://cms.org'/>	<service> … </service>
http://sched.infn.it:8080/getServiceDescription	Parent	Service	50,60,70,80	<owner name = "http://cms.org'/>	<service> … </service>

Wolfgang Hoschek, A Unified Peer-to-Peer Database Framework for XQueries over Dynamic Distributed Content and its Application for Scalable Service Discovery, PhD Thesis, Technical University of Vienna (submitted), 2002

Medical metadata : WP10

- Plan to use SF to store metadata for medical images
  - patient specific & image specific data
  - database will contain LFNs of collections of images
  - separation of confidential and non confidential data
- One SF metadata database instance per hospital
  - queries on numerous distributed databases

Examples:

- patient will query the SF database for all her/his medical images
- physician will query SF databse for all records on one patient
- researcher queries SF database for all ultrsound images

See also presentation of Johan Montagnat

- Have a prototype with local SF installation & insecure connections

Current release information

Spitfire 1.2 : released first week April:

- Integrated XSQL-Spitfire
- Installation decoupled from Tomcat & MySQL
- Java API
- SOAP interfaces designed and we have a basic understanding of the technology. Not currently provided

Future releases

Spitfire 1.3: end May 2002

- Alpha release of SOAP interface
  - Schema and User Management
  - Connection Management (PoolMan)
- Test-suite
  - designed with Junit based tests implemented
- Test-driven by Biology Applications, maybe by HEP-Atlas

Spitfire 1.4: end September (testbed 2.0)

  - Version 1.0: Full integration of all components
  - More running applications