Development Roadmap of the EMI middleware

Balázs Kónya, Lund University, Sweden
Project Technical Director

and L. Field, J. White, P. Fuhrmann, M. Cecchi, J.K. Nilsen

CHEP 2012, 21-25 May 2012, New York
Roadmap: a high level view

• EMI project in a nutshell
• Year 2 development through showcases
• Year 3 development in the pipeline

for details see the numerous EMI posters and presentations

24/5/2012  www.eu-emi.eu
• European FP7 project
• 24 partners with CERN as the coordinator
• 3 years with April 2013 end
• 1115 person months
• 12 M Euro EU funding

EMI brings together the major EU middleware developers
3 years of harmony

EMI common products

Standards, New technologies, Users and Infrastructure Requirements

Repositories

Implemented Agreements

Before EMI

3 years

After EMI

Implemented Agreements

EMI common products

Repositories

24/5/2012 www.eu-emi.eu
Main goals

- Quality Improvements
- Maintenance Support Release
- Software Eng.
- Innovation and Development
- Dissemination Training
- Exploitation Sustainability

Web site, communication channels
DCI and other collaborations
ScienceSoft
Commercial collaborations

EMI 1 Kebnekaise Legacy pre-EMI MW
EMI 2 Matterhorn

24/5/2012 www.eu-emi.eu
support period: 12+6+6
One year maintenance & support

119 component updates during 14 EMI-1 update cycles

1218 bugs fixed

493 incidents closed

Number of EMI Releases

Bugs closed

Incidents closed

24/5/2012

www.eu-emi.eu
Year of the Development

**EMI 2 Matterhorn**
- New functionality
- New products
- Agreements and standardization

- **29** Product Teams
  - **57** products
- **101** planned development tasks
  - **75** completed
  - **19** prototypes
  - **7** postponed

**EMI 1 Kebnekaise**
- Open Source process adoption
- Distribution consolidation

---

[24/5/2012]

[www.eu-emi.eu]
EMI 2: congratulation!

EMI 2 (Matterhorn) has been released

21/05/12 12:47
The EMI project is pleased to announce the availability of the EMI 2 (Matterhorn) release.

http://www.eu-emi.eu/emi-2-matterhorn

This release provides many new features and enhancements for many middleware components from ARC, dCache, gLite, and UNICORE, including important new services not available in EMI 1.

The new functionality includes a new common Service Registry Service (EMIR), the Worker Nodes on Demand Service (WNoDeS), Nagios probes for most services, better integration of the Argus authorization framework with many services, the initial implementation of the common EMI Execution Service interface and more.

All services are available for Scientific Linux 5 64 bit and Scientific Linux 6 64 bit. A subset of services is also available for Debian 6 64bit and more will be released in the near future.

http://emisoft.web.cern.ch/emisoft
Major new features and services

• Implementation of a common EMI interface for job management (EMI-ES)
• Extended back-end support for all EMI CEs to Torque, SGE, LSF, SLURM
• All the EMI services publish GLUE2 conformant information
• A common service registry for all EMI services (EMIR)
• Comprehensive set of Nagios probes for every EMI service
• A prototype of the common EMI security library (CANL)
• ARGUS as the official EMI authorization solution via integration with all the SEs and Ces
• ...
• ...

Technical Agreements and Standardization

• EMI Delegation agreement
• Storage Accounting record (StAR) and Compute Accounting record (CAR)
• A common EMI SAML profile
• ...
• ...

Year of the Development

24/5/2012

www.eu-emi.eu
Show-case 1: Data Industry Standards

- Industry standard protocols for accessing SEs and the catalog
  - DPM and dCache ready for NFS4.1
  - HTTPS offered by DPM, StoRM and dCache
  - WebDAV support in DPM and dCache
  - WebDAV support being developed in FTS3 and LFC
- Vital part of the greater vision for EMI Data
Show-case 2: EMI service Registry

**EMIR**: from implementation through performance testing to deployment planning

- One central registry where all services can be discovered
- *No registry supported by all three middleware providers prior to EMI*
- **Unified service discovery**
- Quorum based, replicated Global Service Registry DB ensures HA
- Service providers register to and push information to Domain Service Registries
- Deployment hierarchy of DSRs in investigation
- All services to publish information in EMIR
Show-case 3: EMI Execution Service

EMI-ES: from specification to implementations

- EMI-ES interface specification agreed as the common job management interface
- Web-service interface with
  - Integrated support for data staging
  - Delegation capability
  - Re-engineered state model
  - Revised job description
  - Glue2-based service and activity description
  - Clearly defined Port Types
- All CEs implemented EMI-ES job management specification at 80-90% completion level
- Iterative schema definition to protect users from child deceases
  - EMI-ES v1.2 revision
  - Client side development ongoing
  - Central piece in the EMI harmonisation efforts
Show-case 4: Common Authentication Library

CANL: from implementation to adoption

- Defined and documented API for common security library
- Main features
  - Credentials handling
  - Trust store handling
  - Name constraints checking
  - CRL
  - Proxy: verification, generation, proxy CSRs, utilities
  - Partially unified error codes and messages
  - OCSP support (on-line revocation)
  - PKCS 11 (support for smart cards & soft tokens)
- Implementation available in C, C++ and Java
- EMI products migrating to CANL
  - VOMS, ARC HED, Trustmanager, L&B, UNICORE, dCache, CREAM, Argus, Pseudonymity, Hydra, STS
Show-case 5: Car & StAR

- EMI agreement on accounting records
  - Compute accounting CAR
  - Storage accounting StAR
- To be used “everywhere”
  - accounting sensors
  - APEL
- Feeding all this back to OGF
  - UR 2.0 (?)
Show-case 5: Car & StAR

EMI Compute Accounting Record (CAR)

• Mainly a profile of the OGF UR1.0

• Some differences:
  – Added support for groups/VOs
    • Using Group+GroupAttribute like in StAR
  – Definition encompasses both usage records and aggregated records
  – Aggregated schema follows APEL SSM guidelines
StAR:

- Definition of usage record for storage
- Inspired by OGF UR1.0
- Summarises used space
- XML-based schema
- Non-overlapping/non-contiguous records means no space used
  - ValidDuration configured by sys-admin
- Supports handling of groups (aka VOs)
- Scope is limited to consumption of storage space
The last peak (Y3 development plans)

General strategy:

- Complete product developments:
  - FTS3, GFAL2
  - STS
  - EMI Datalib
- Product hardening, focus on usability
- Integration and adoption of common EMI solutions (EMIR, CANL)
- Migration plans, compatibility
In the pipeline: FTS3 (GFAL2)

• See talk later Today
In the pipeline: STS

• Security Token Service
  – For simplified credential management
  – STS transforms an existing security token into another security token
  – SOAP-based Web Service

forge.switch.ch/redmine/projects/sts/wiki
In the pipeline: emi_datalib

ARC and GLITE JOIN FORCES TO WORK ON THE UNIFIED EMI DATA LIBRARY

WE DELIVER. TOGETHER.
• file-based data-moving library – libarcdata2
• pluggable architecture
• libarcdata2 is used by the arc command tools, the ARC CE and external clients to upload and download files
• written in C++

• posix-like data access
• pluggable architecture
• several external clients (e.g. experiments software) use GFAL and lcg_util libraries
• written in C

24/5/2012

www.eu-emi.eu
- posix-like interface from GFAL2
- higher-level file-based interface from libarcdata2
- adds transfer interface to GFAL2 for initiation and monitoring of 3rd-party transfers
- libarcdata2 will use GFAL2 through plugin
- external clients requiring posix-based byte-wise data access can use GFAL2 directly
- file-based data moving clients (e.g. lcg_util CLI, ARC CLI and ARC CE) can use libarcdata2
- FTS3 and parts of the lcg_utils and ARC CLIs will use the 3rd-party transfer interface of GFAL2
- python library will be created to the needed functionality from the lcg_util python API
- all plugins are moved under GFAL2 though some plugins are file-based and not posix – allows posix access without needing to know physical file locations
Summary

• Three-year-long EMI project has given an excellent one-time opportunity to work together
• Most development intensive phase of EMI is just behind us
• With Matterhorn EMI delivered
  – Common libraries (CANL)
  – Common service (EMIR)
  – Common implemented interfaces (EMI-ES over CEs)
  – Agreements and adoption plans (CAR-STAR, ...)
• The third last year still to bring some interesting stuff but the focus is now on product hardening
  – Planned for EMI 3: STS, FTS3, GFAL, emi_datalib
EMI is partially funded by the European Commission under Grant Agreement INFSO-RI-261611