



# European Middleware Initiative (EMI)

Alberto Di Meglio (CERN)  
Project Director

# What is EMI?

The European Middleware Initiative (EMI) project represents a close collaboration of the three major middleware providers - ARC, gLite and UNICORE, together with other software providers - to establish a sustainable model to support, harmonise and evolve the grid middleware for deployment in EGI and other distributed e-Infrastructures



# FP7 Program

## FP7 Capacities Work Programme 2010: Infrastructures Call **FP7-INFRASTRUCTURES-2010-2**

### Sub-topic: 1.2.1.3 – Middleware and repositories

Develop middleware that strengthens European presence by consolidating or even going beyond existing DCIs (e.g. exploiting emerging developments like virtualisation), while improving their stability, reliability, usability, functionality, interoperability, security, management, monitoring and accounting, measurable quality of service, and energy efficiency

Starting date: May 1<sup>st</sup> (?)

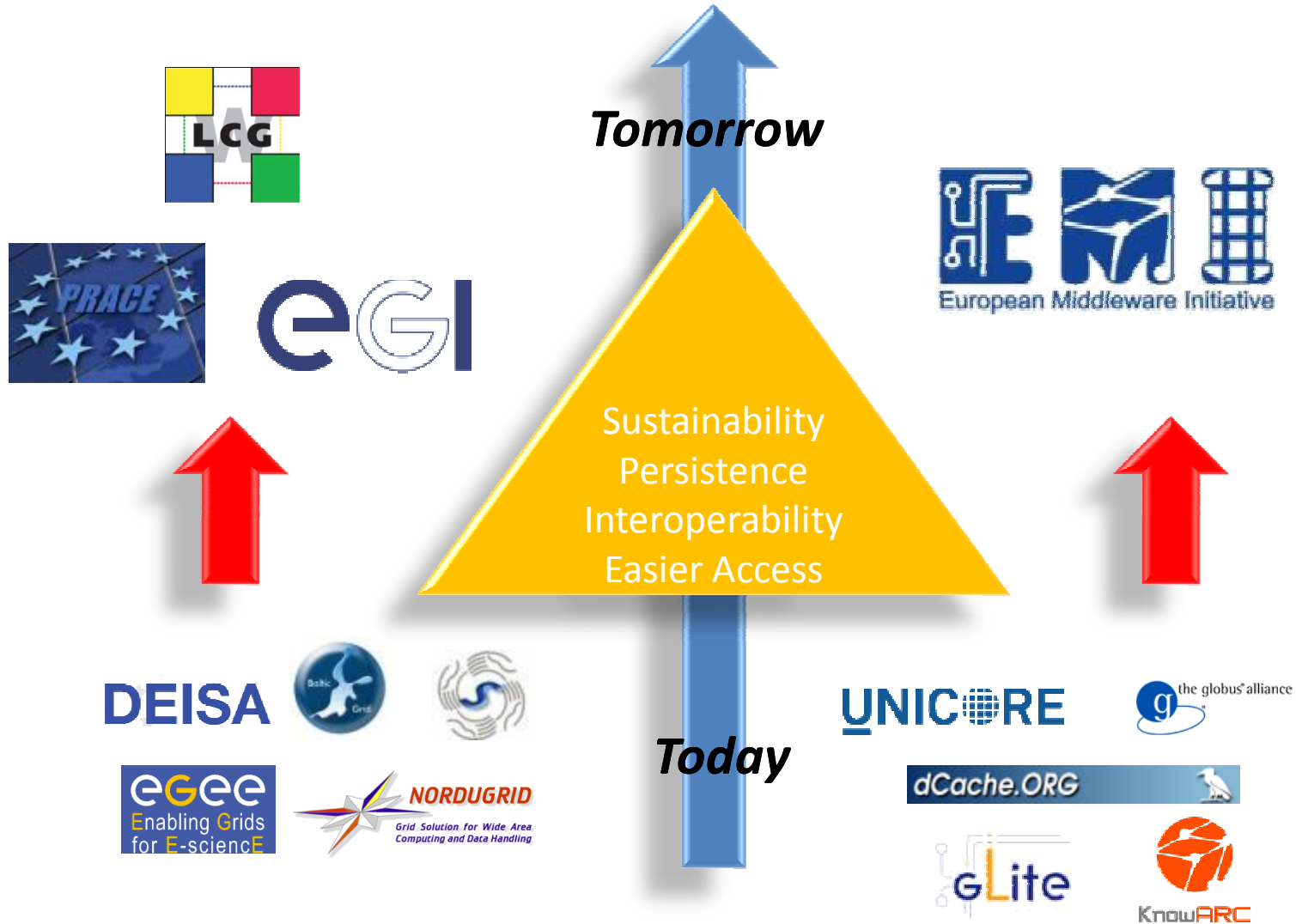
Duration: 3 years

Total budget: 13M EUR (EC) + 13M EUR (partners)

Effort: 65 FTEs



# A European Vision



# Objectives

## Consolidate

Consolidate the existing middleware distribution simplifying services and components to make them more sustainable (use of off-the-shelf and commercial components whenever possible)

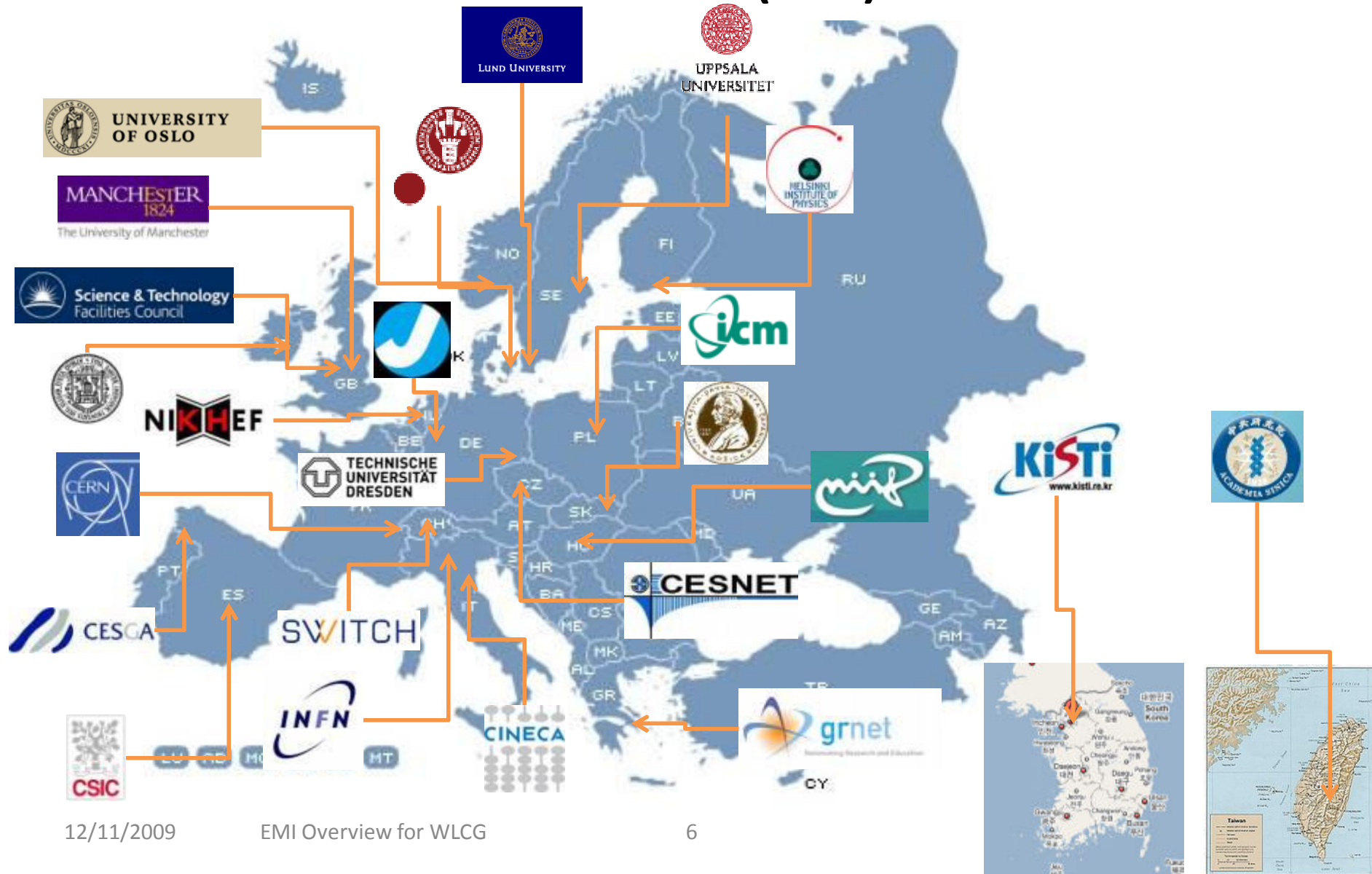
## Evolve

Evolve the middleware services/functionality following the requirement of infrastructure and communities, mainly focus on operational and interoperability aspects

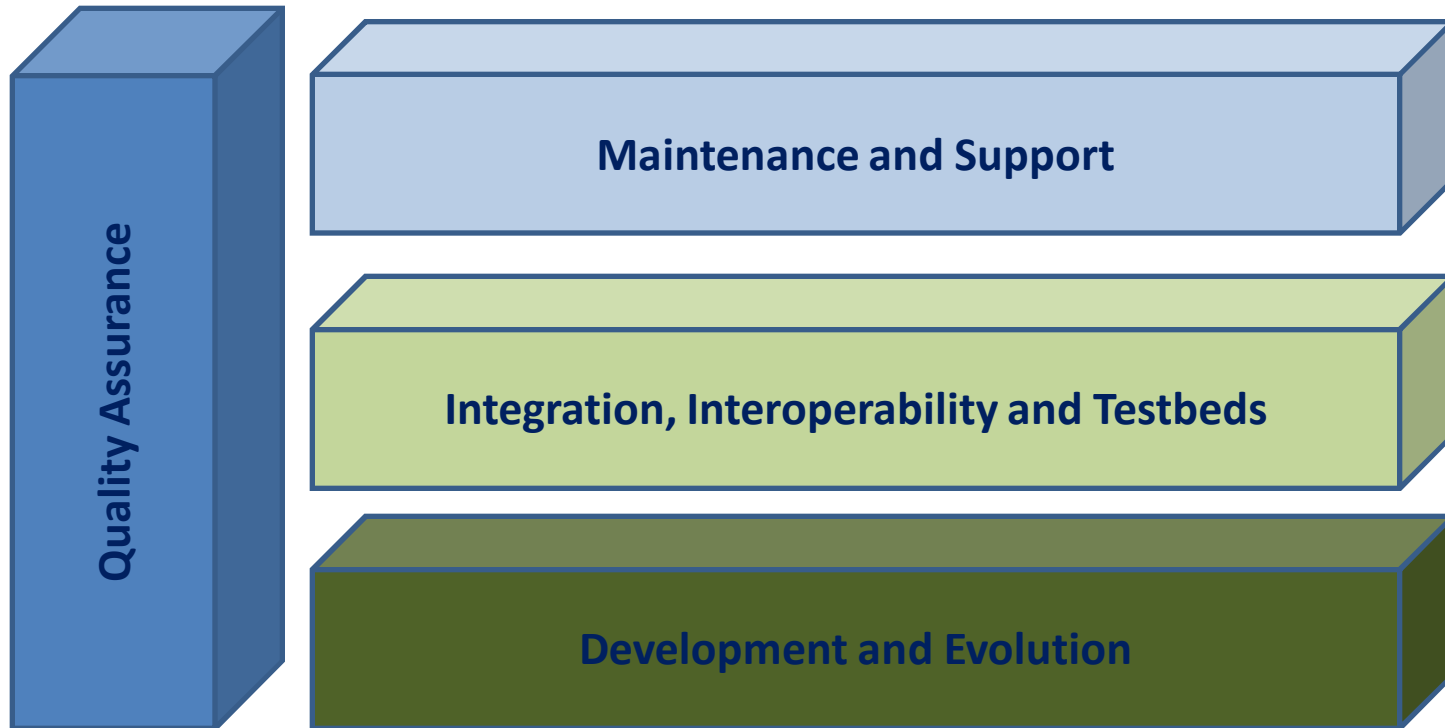
## Support

Reactively and proactively maintain the middleware distribution to keep it in line with the growing infrastructure usage

# Partners (24)



# Project Structure



# Technical Areas

## Job Management Services

A-REX, UAS-Compute, WMS, LB, CREAM, MPI support

## Data Management Services

Classic SE, dCache, StoRM, UAS-Data, DPM, LFC, FTS, Hydra, AMGA

## Information Services

Service Registry, BDII, Service Discovery, info providers

## Security Services

HED security, Gateway, UVOS, VOMS/VOMS-Admin, ARGUS, SLCS, glExec, Gridsite, Proxyrenewal

## Infrastructure Services

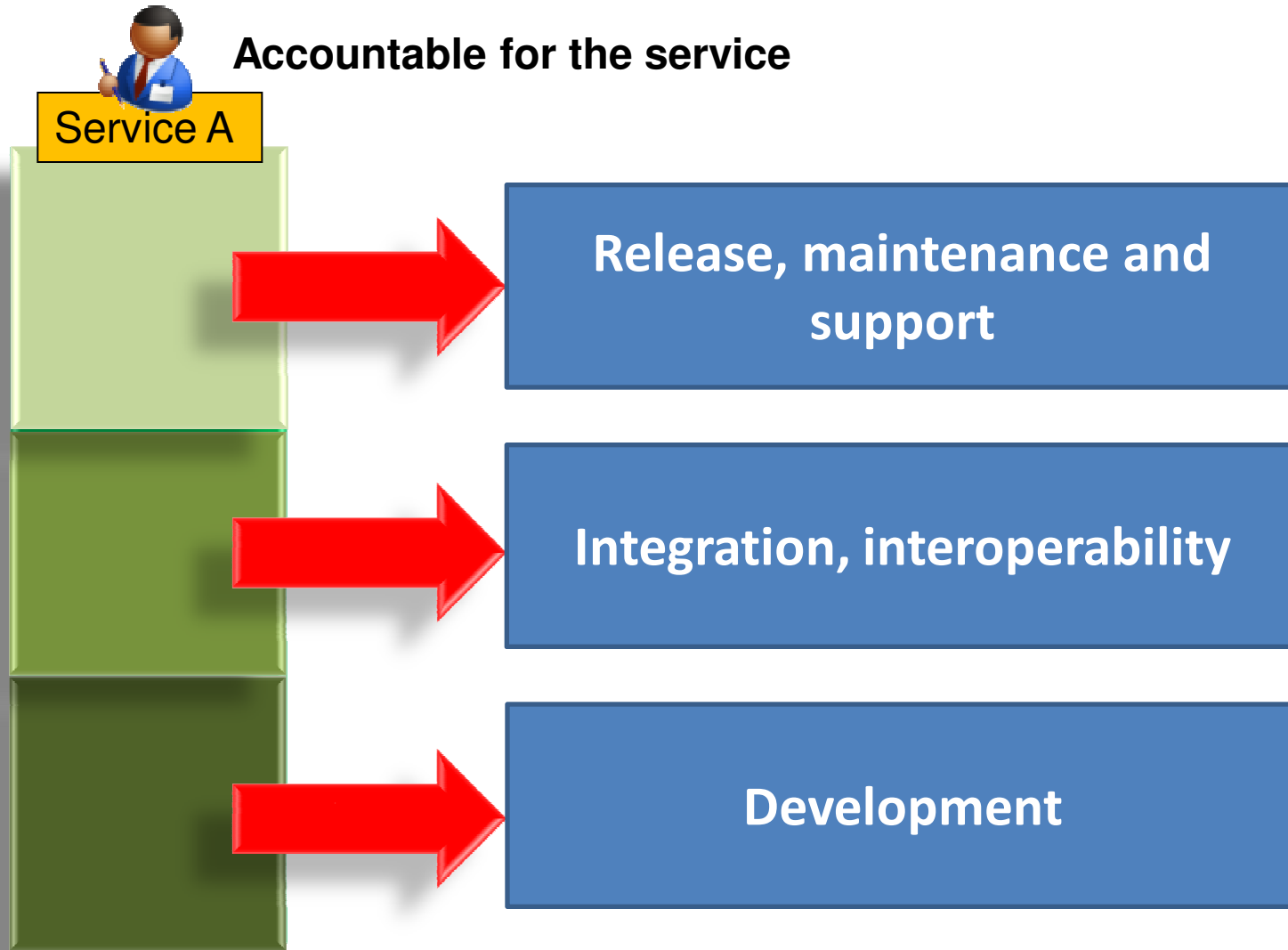
Messaging Service, APEL and accounting probes, monitoring interfaces, virtualization/clouds support



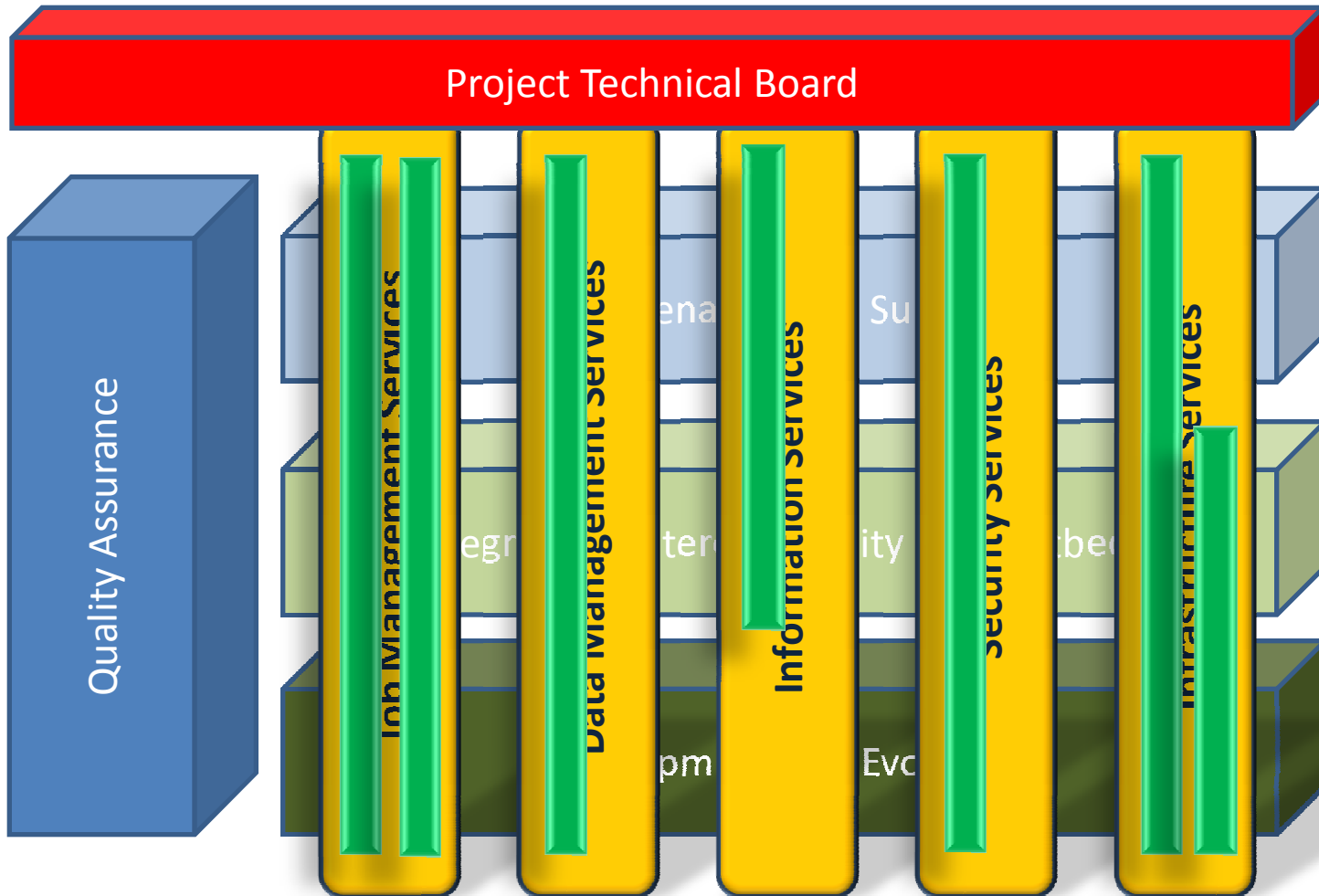
# Product Teams

- Product teams are the services implementation teams within each area responsible to deliver software releases and all associated material.
- They perform the required technical tasks from design to release through implementation, testing, certification and 3<sup>rd</sup>-level support as part of one or more Work Package and according to what specified by the Project Technical Group and the QA policies.
- Product Teams are flexible, in the sense that they can be formed or closed as the corresponding products are introduced or obsoleted in EMI and allow adding or removing services as needed even from external contributors
- They provide a transparent and direct method to assign responsibility for a service

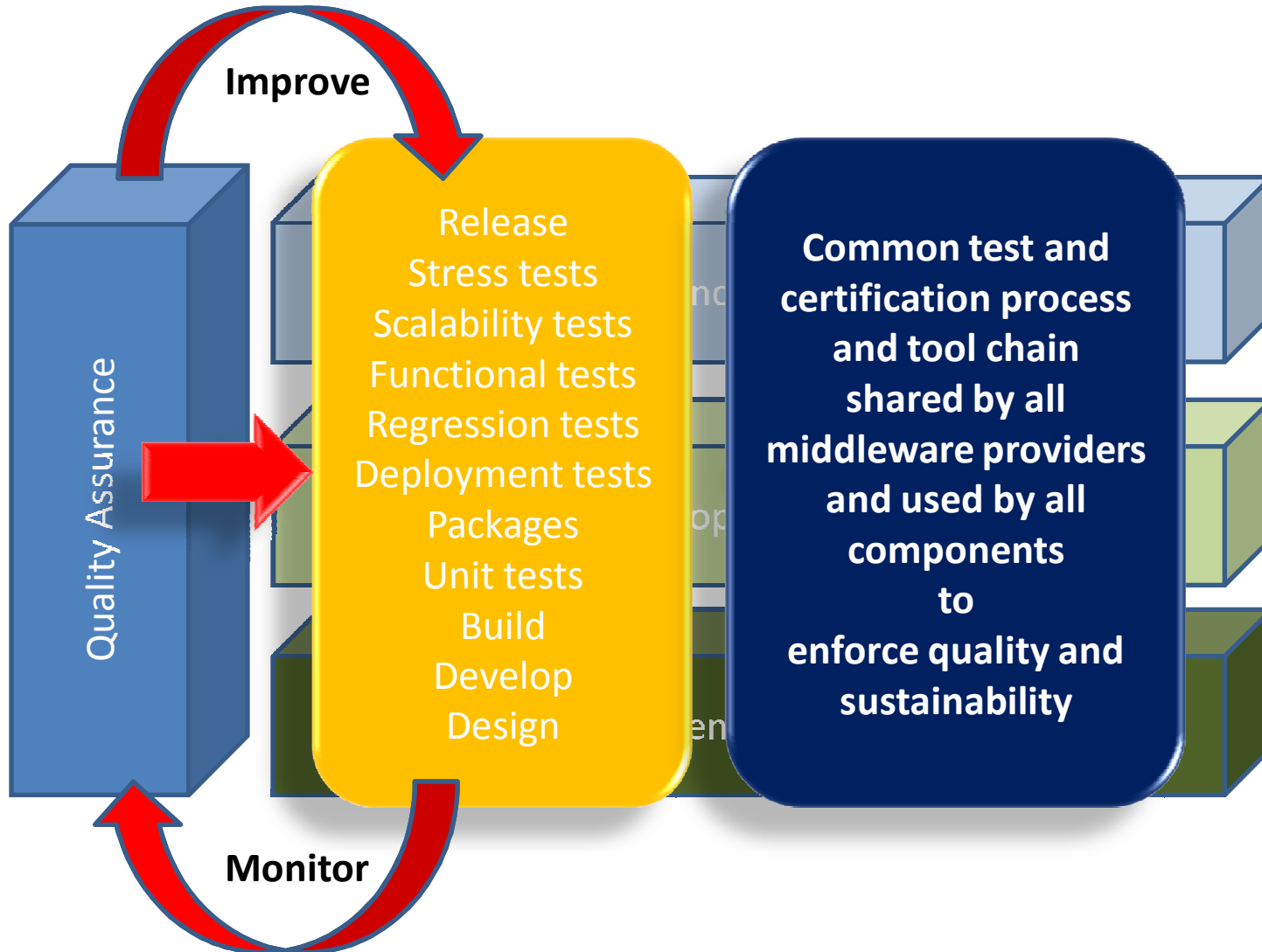
# Product Teams



# Full Matrix Structure



# Quality Assurance





# Releases and Release Policies

**This applies to individual components**

- **Major releases:** once or twice per year, may contain non-backward compatible changes
- **Minor releases:** a few times per year, fully backward compatible, may contain new functionality
- **Revisions:** every week or two weeks, only bug fixes
- **Emergency:** as needed, only specific bug fixes, use emergency release procedures

# Selection Criteria

EMI includes:

- Middleware services and components from the participating providers
- All services that are or will be in production by next year
  - EGI, PRACE operations must be fully supported
- Services not currently in production, but already known to replace existing services
  - Ex: ARGUS to replace LCAS/LCMAPS
- New services satisfying identified new user needs
  - EMI requirements are user-driven, satisfy the needs of the user communities filtered through the coordination of major initiatives (EGI, PRACE, WLCG, SSCc projects)

# New Development

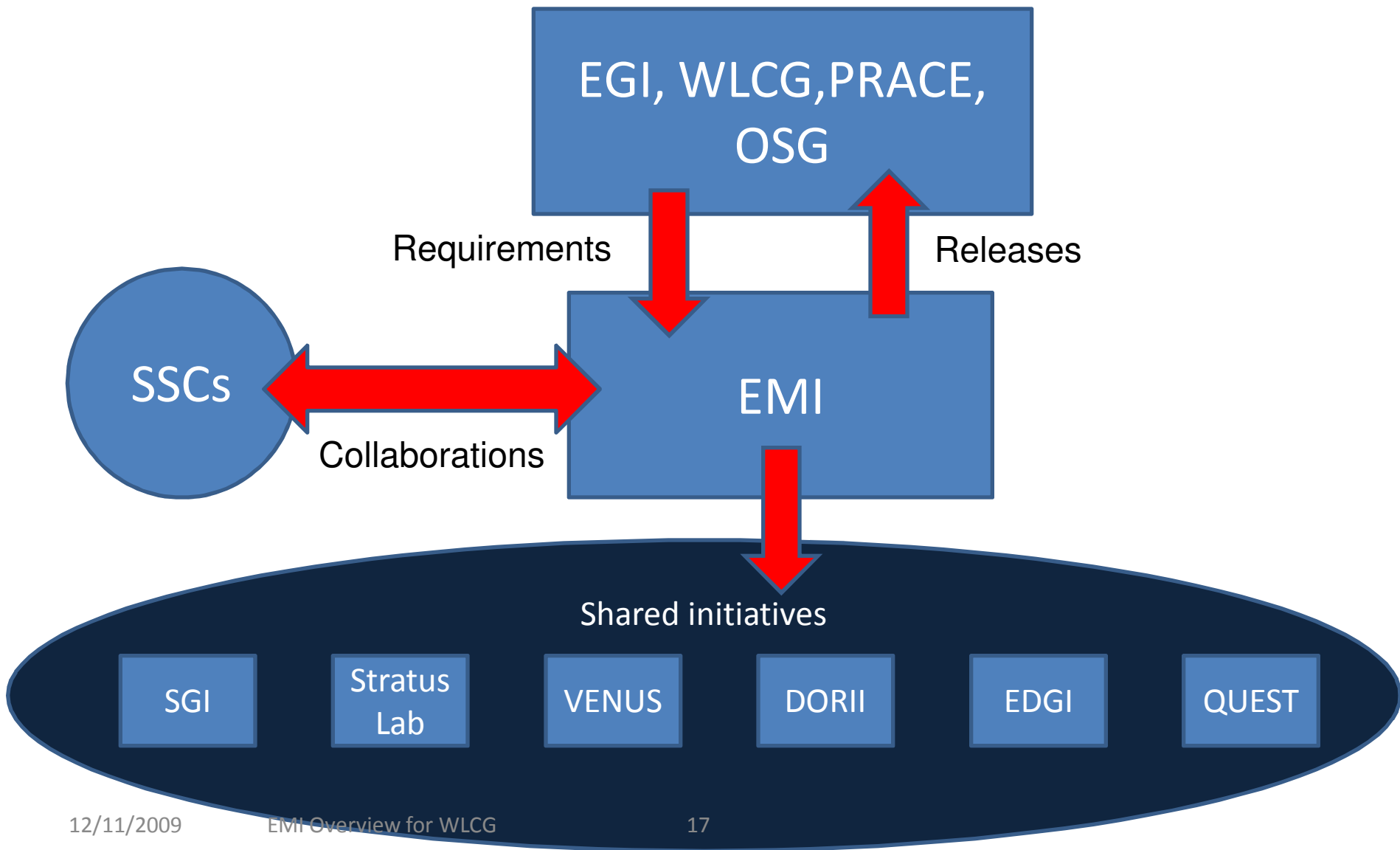
- New services or functionality are introduced as needed based on:
  - New user requirements
  - Need to evolve existing services or replace older technologies to support the growing infrastructures
  - Special focus on integration of virtualization, monitoring interfaces, accounting, support for gateways and portals
- Trade between new research and stability of the infrastructure operations
  - Large-scale realistic testing (in collaboration with the infrastructure providers), phase-out/migration plans, support policies

# Middleware Evolution Targets

- **Messaging:** messaging service part of the middleware architecture, used for various information tasks (monitoring, accounting, policies, etc). Based on activeMQ or MRG
- **Accounting:** minimum necessary modifications to the most used sensors and collectors to use messaging and a standard data format. Accounting tools are not in scope
- **Monitoring:** monitoring and instrumentation interfaces following an agreed standard (QMF?) to be used by higher level monitoring apps
- **SRM and security** 'harmonization' and **common storage accounting** across SE services, **POSIX support** extension for data access services
- **Consolidation of security models** across the three middleware stacks
- **Consolidation of clients/APIs** (keep backward compatibility, but simplify)
- **MPI:** support added as necessary, convergence HTC/HPC?
- **Virtualization and clouds access:** support for existing or new systems (depends on what systems will be most used)
- **Interoperability** between HTC and HPC
- **Portlets specs implemented**, but no portals



# Collaborations





**Thank you**