Security in EMI

John White
Helsinki Institute of Physics
(for the gLite and EMI Security teams)

www.eu-egee.org

e-infrastructure

Information Society and media

CAPACITIES

Overview

- **EMI Background.**
- **Progress to present.**
  - **gLite++**
  - **EMI editorial team process.**
  - **Technical breakout process.**
- **Documents.**
  - **Summary**
- **Discussion.**

Some definitions

- EMI - European Middleware Initiative.
  - Main software provider to EGI.
- EGI - The European Grid Initiative.
  - Main customer of EMI.
- UMD - Unified Middleware Distribution
  - `http://knowledge.eu-egi.eu/knowledge/index.php/UMD`
- DCI - Distributed Computing Infrastructure.

EMI Project

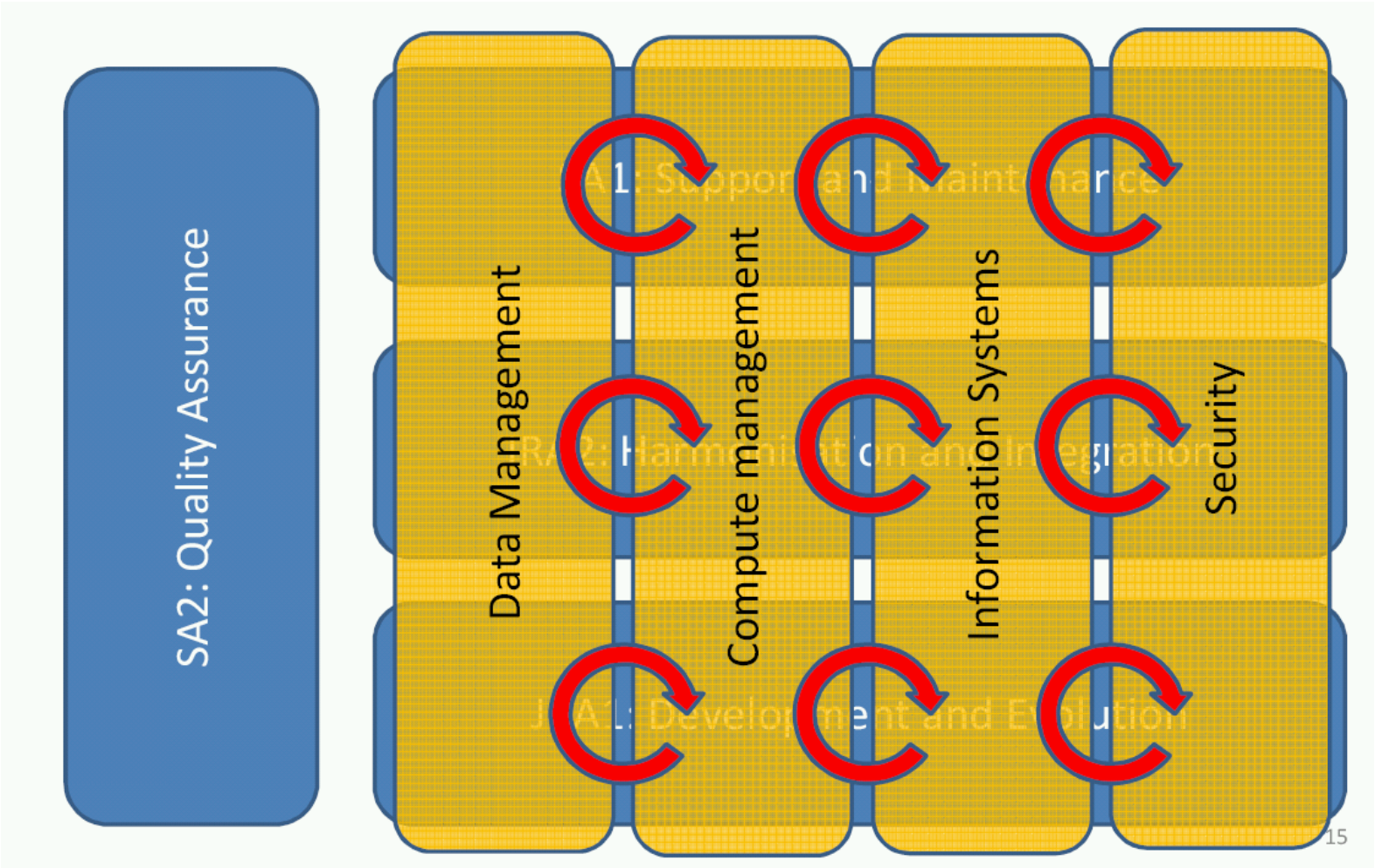
- **The European Middleware Initiative.**
- Tentative list of partners:
  - **ARC:**
  - KU, NIIF, LU, UiO, UPJS, UU.
  - **gLite.**
  - CERN, CESNET, CSIC/CESGA, INFN, NIKHEF, SFTC-RAL, SWITCH, UH.HIP, UoM.
  - **UNICORE.**
  - CINECA, FZJ, ICM, TUD.
  - Up to two more partners for "non-development" tasks.
- EMI will provide a coherent Middleware stack primarily to EGI.

General principle statements

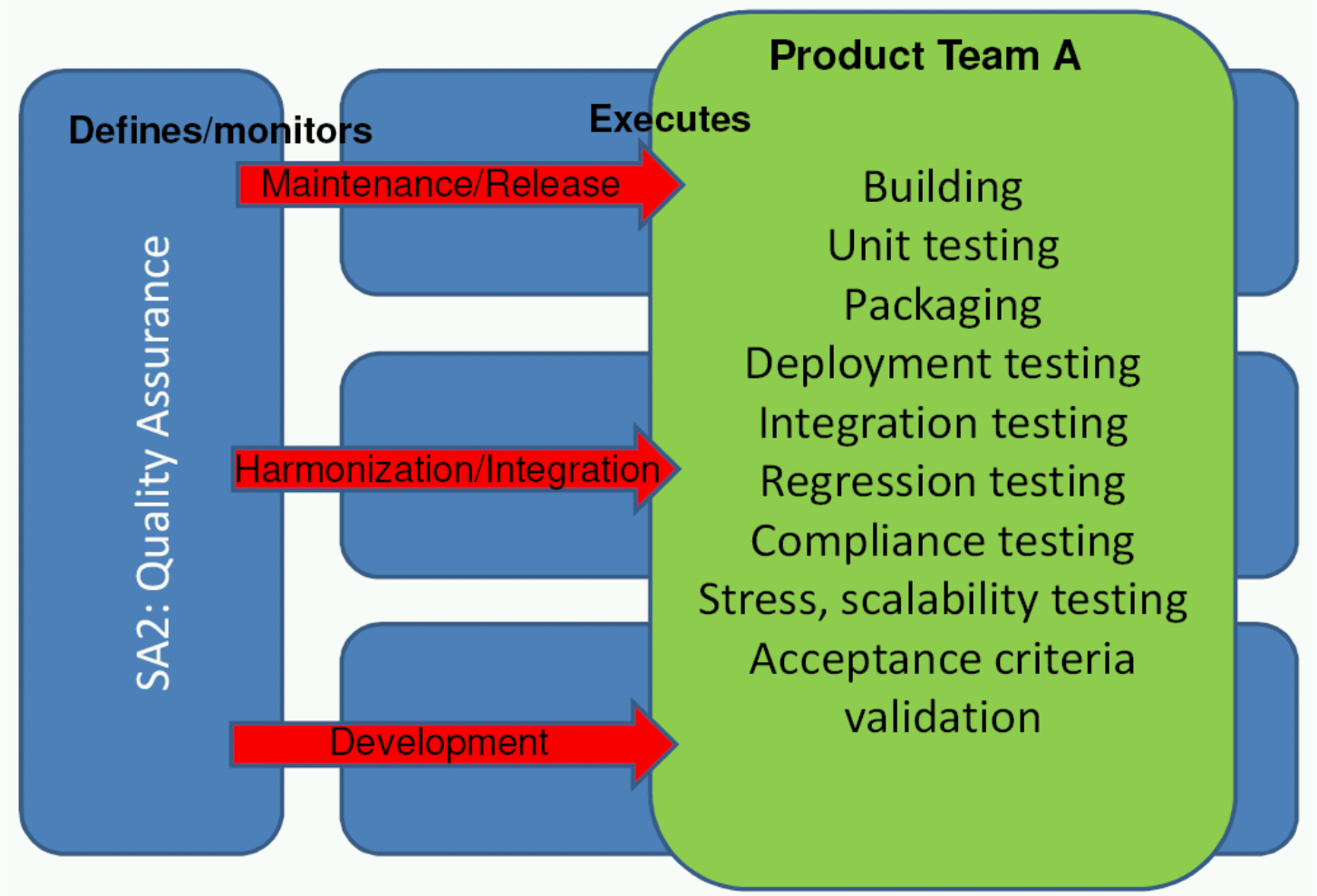
- EMI must maintain and evolve the middleware services for which it is responsible so that they meet or exceed the needs of EGI and other distributed infrastructures (DCI).
- The middleware services provided by EMI and needed by EGI and other DCIs have to be maintained as necessary. Maintenance includes bug fixing and all changes necessary to improve usability, reliability, stability and scalability of the services according to EGI and other DCIs requirements.
- Services must be evolved, harmonized, standardized, replaced, merged or discontinued as needed following a clear work plan. Not all services may need evolution or harmonization, but this doesn't exclude them from EMI if they are used by EGI and other DCIs.

What this means

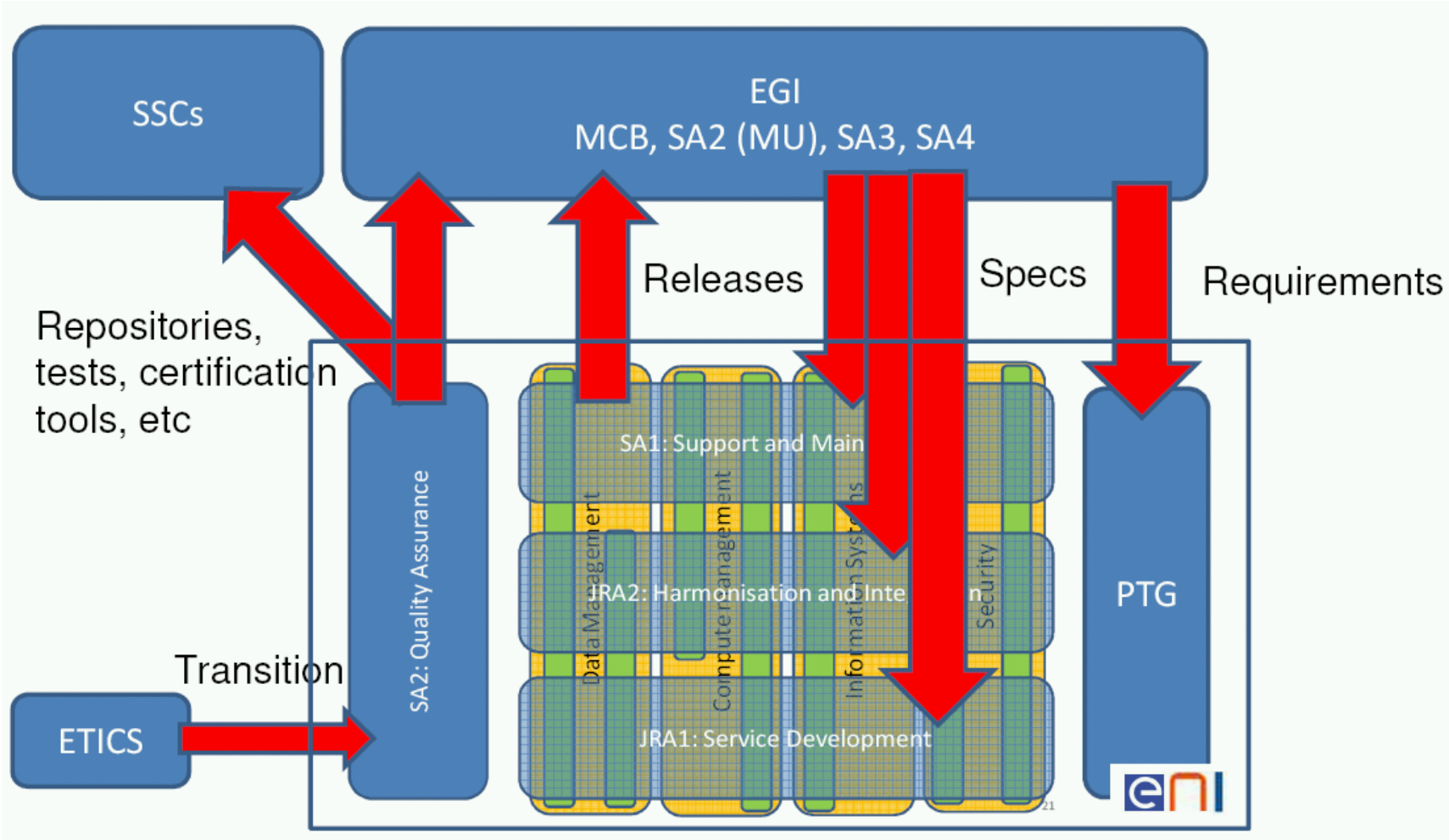
- EGI is our main customer. They should provide a list of components and their requirements to EMI. The responsibility for software may differ for other customers.
- At the start there has to be a list of EMI components that are vital to keep the "infrastructure" running. No gaps. Some maintenance is foreseen to be "proactive" ie. development required for scalability.
- In order to converge the three MW stacks and, over time, reduce the maintenance effort we are told there must be a reduction in the number of components and adoption of standards.



Work areas cross activity boundaries.



Product teams are responsible for all aspects.



Client-driven work processes?

- Meeting on June 11th 2009.
  - Discussion of the possible gLite contributions to UMD.
- August 26th 2009.
  - Weeks of telephone conferences...
  - Produced a document based on provided template.
  - "Recommendations for gLite Security Components in UMD Middleware".
  - V. Ciaschini, C. LaJoie, C. Witzig, J. White.
  - `https://edms.cern.ch/document/1016213/1.`
  - **Some items still under discussion.**
- **Not all gLite security group members in accord.**
- **But parts of this document forms the basis for later work.**

General overview of results...

- Reduce the number of maintained security services in the future.
- LCAS/LCMAPS and SCAS would be replaced by the ARGUS Authorization system.
- Merge VASH into VOMS.
- Minimal option: use of "Exclusive Short-lived Certificates"
  - Rationalize and simplify the current authentication model.
  - Removal of support for long-lived (more than a day) proxy certificates.
  - Issue only short-lived (24 hours or less) proxy certificates Bootstrapped by a users home organization or other means.

General overview of results...

- Unified init driver - capable of obtaining credentials from pre-existing authentication systems
  eg: Kerberos tokens; username/password pairs; Shibboleth and others.
- X.509-based Authentication Library.
  - Create a single library with APIs, in Java and C.
  - Geared towards authentication and user/service information retrieval.
- Adoption the "Unicore Security Model"
- The "Security Token Service" model would require a significantly higher amount of effort.
  - Would produce a large disruptive change to the existing infrastructure.
  - Would need strong client interest in moving in either of these directions.

EMI Editorial Process

- Initially formed a four member editorial team.
  - ARC (1), UNICORE* (1), gLite (2).
  - First tasks to produce a project structure.
  - Presented to the core group August 4th.
  - Re-design started almost immediately!
- Lead to instructions (Aug 18th) for "Technical Team Breakout" sessions.
  - **Compute.**
  - **Data.**
  - **Information.**
  - **Security.**
- Experts recruited per area and consortium.
- Results due (and delivered) by August 28th.

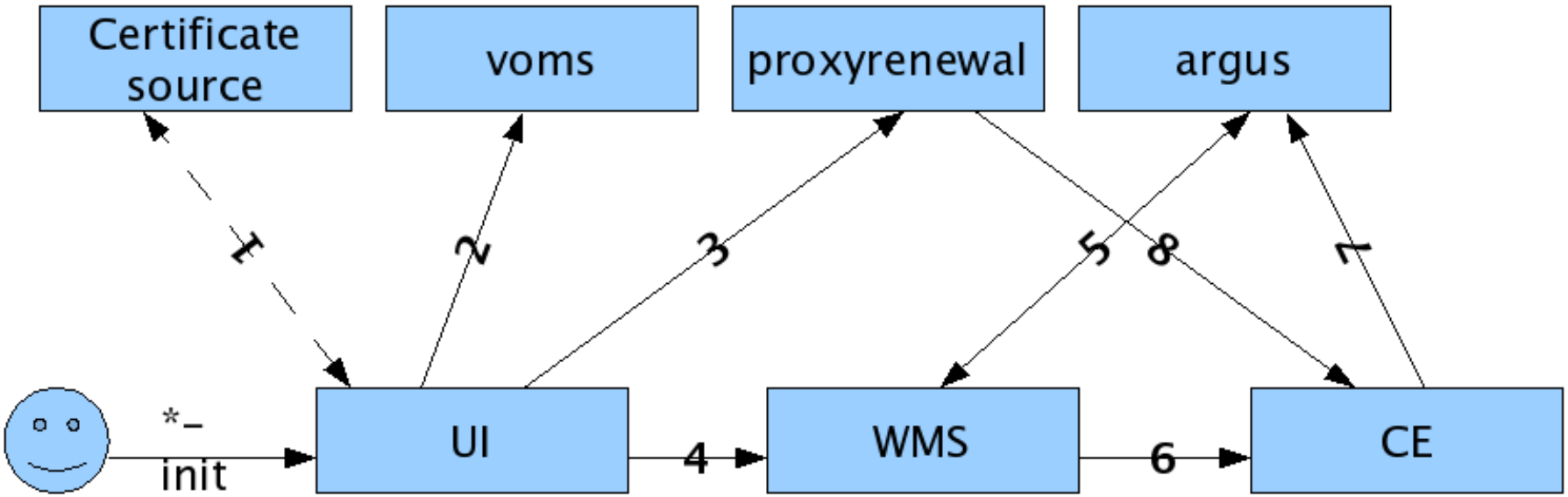
Security Technical Breakout

- Krzysztof Benedyczak
- Vincenzo Ciaschini
- Aleksandr Konstantinov
- Ales Krenek
- Morris Riedel
- Bernd Schuller
- Oxana Smirnova
- Weizhong Qiang
- Christoph Witzig
- John White

- Initial architecture, not a fully precise design.
  - A level of confidence that harmonization will succeed.
- Not a focus on individual harmonization opportunities.
- Should cover the prioritized harmonization opportunities.
  - State clearly why additional opportunities are needed.
  - Input: Initial list of components in context.
  - Input: Priorization PPT page with identified opportunities.
- Output:
  - 1 page design architecture diagram.
  - 1-3 pages of text with refinements from annex text.
  - 1-3 lines for the roles of the components in context.
  - Expected effort in FTE.
  - Harmonization type: eg.
    - Unified Interfaces, e.g. three different CEs with the same interface.
    - Convergence, e.g. one CE merged from all three.
    - Obsoletion, e.g. one particular CE is not needed anymore.
- **Deadlines for teams end of business 27.8.2009**

- After discussions, smaller team proposed:
  - Vincenzo Ciaschini
  - Daniel Kouril
  - Ales Krenek
  - John White
- Quicker route to a architecture proposal, approved by overall group.

Security Architecture Results



- New AuthZ system, removal of LCAS/LCMAPS.
- Variable “certificate source”.

"Harmonization" opportunities:

- Communication among services: SSL/TLS + X.509 proxies, but compatibility with GSI.
  - Delegation requires GSI. SSL/TLS+X.509 proxies not compatible with GSI, can be made so with minimal effort.
- Delegation and proxy storage collapsed into one service.
- Policy services: each MW stack implements own.
  - The policies inside the PDP are (or can be) XACML.
  - Communication between PEP and PDP in the SAML/XACML protocol.
- Harmonization achieved by:
  - PDP must implement the full XACML spec.
  - SAML/XACML protocol implementations are compatible.
  - Common set of attributes to be used in policies.
- Implementations of services relying on SLCS-type CAs desirable.
  - Create a common framework into which plugins for specific implementations to be inserted

Security Architecture Results

- No initial diversions from wLCG (OSG and NAREGI).
  - Evolutionary changes only.
- Authentication: include short or long-lived certificate CAs.
  - IGTF-defined profiles for SLCS , MICS
  - eg. Shibboleth, Kerberos, MyProxy in CA mode.
- VOMS attributes from eg. Shibboleth (VASH).
- Generic client to generate X.509 credential (proxy).
  - Short-lived RFC 3820-compliant, GT2 proxies for backwards compatibility.
- Authorization: a global, distributed authz system (Argus?).
  - Able to handle complicated policies.
  - Able to associate attributes also to services.
- Proxy delegation: restriction necessary.
- Common authN Library across all services.
- Anonymization: For Biomed. See RFC 5636.
- Session tokens: Needed for Data Management.

- Effort in FTE...
  - Not provided by this group.
  - Under discussion between EMI member consortia.
  - Currently shifting.
- This document used as input to EMI project.

p.s. There's a LaTeX template available:

https://edms.cern.ch/document/964536/1