Site Access Control Arch (DJRA3.2)

David Groep

NIKHEF

www.eu-egee.org

Information Society

Outline

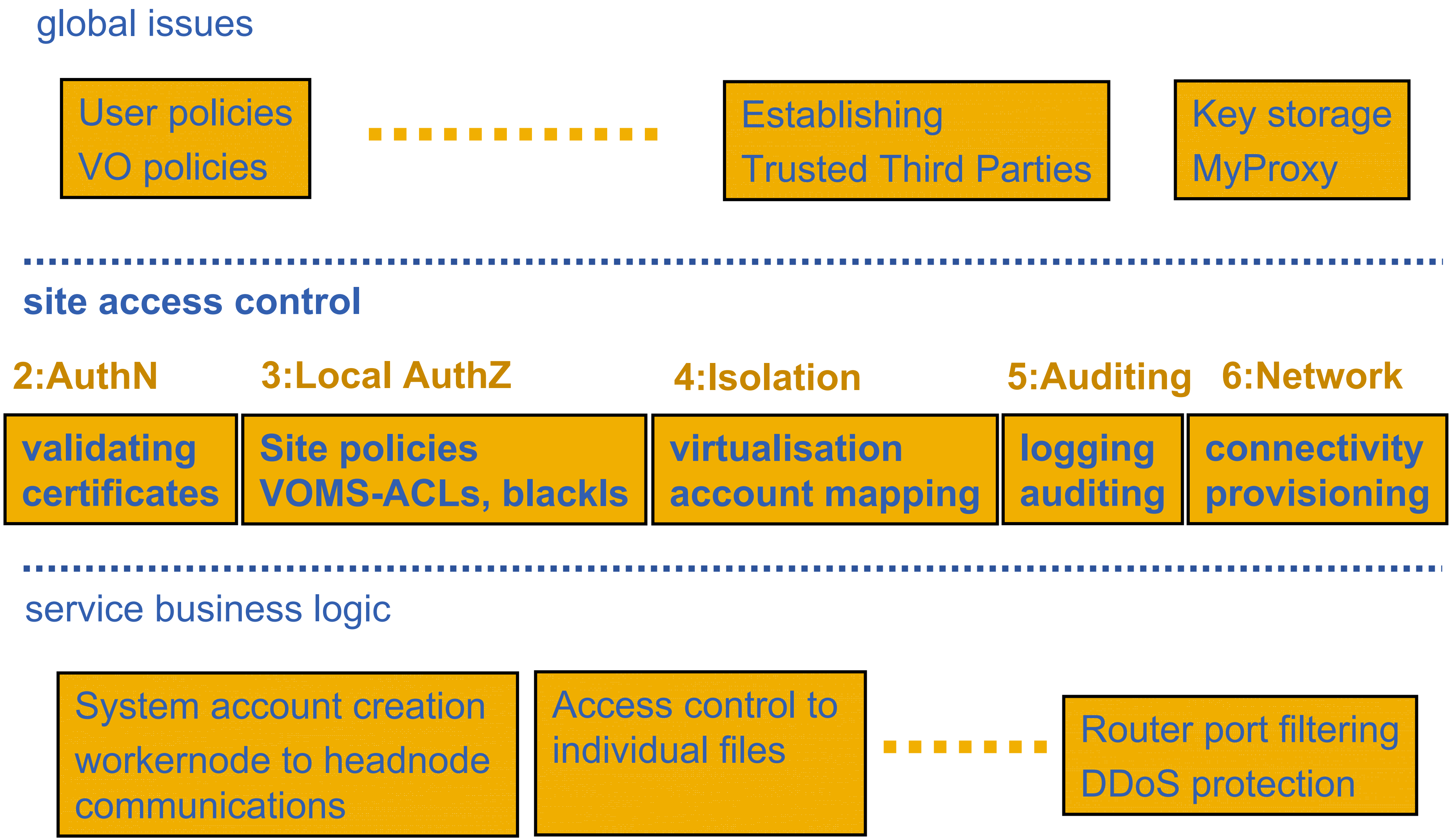
- Timeline
- Positioning and scope
- Document structure
- Overview of Site Access Control Mechanisms

Timeline

- November 30th:
  RFC from the Joint Security Policy Group
- December 13th:
  RFC from ROC managers & MWSG
  Preview version to moderator & reviewers
- December 31st:
  Official 1st version, sent to reviewers Jan 3rd

- January 10th:
  Approved by the moderator & reviewers
  Version 1.0 (this version) released

Position of Site Access Control



global issues

User policies
VO policies

Establishing
Trusted Third Parties

Key storage
MyProxy

site access control

2:AuthN

validating certificates

3:Local AuthZ

Site policies
VOMS-ACLs, blackls

4:Isolation

virtualisation
account mapping

5:Auditing

logging
auditing

6:Network

connectivity
provisioning

service business logic

System account creation
workernode to headnode communications

Access control to individual files

Router port filtering
DDoS protection

Goals

- **Generic access control to services**
  - Authentication
  - Authorization
  - for legacy applications & file access, networks, …
- **Sites are always in control of their resources**
- **Flexibility, scalability**
- **Allow for central control in a site**
- **Converge to a single policy format**
- **Standardization of configuration**
- **Address requirements from NA4, SAAA-RG, and others (incorporated in MJRA3.1 “user requirements”)**

Problem approach

Each chapter deals with a particular issue at three levels:

- **Roadmap**
  direction of the proposed solution regarding interoperability and sustainability of the solution
- **EGEE architecture**
  which part of the roadmap correspond to current requirements, and what part is achievable within the timeframe of the project.
- **EGEE-1 release implementation**
  if the implementation differs from the proposed architecture, this describes what is actually there

Where relevant, deployment considerations are given

Structure: high-level

Five main areas:

- Ch. 2: Authentication validation
  - *certificates, trusted CAs,*
  - *interoperation with other AAIs & the e-IRG Roadmap,*
  - *dynamic federations,*
  - *use of MyProxy to bridge mechanisms*
- Ch. 3: Local Authorization
  - *site policy decision points (PDPs)*
  - *allow a VO and its groups, ban specific users, constrain user proxies,*
  - *interoperate with local access to mass stores*

Structure: local fabric

- **Ch. 4: Isolation**
  - *resource virtualization,*
  - *account mapping & management of pool accounts*
- **Ch. 5: Auditing**
  - *what information is logged, deployment issues in logging*
  - *keeping account history*
- **Ch. 6: Network issues**
  - *matching the site requirement ‘no network access from worker nodes’*
  - *with the user requirement for ubiquitous access from worker nodes*
  - *Dynamic Connectivity*

Authentication validation

- **Currently a common third-party trust federation: IGF**

- **Roadmap**
  - allow interoperability with non-certificate-based systems (Shibboleth, EduRoam, RADIUS, GN2)
  - scale better to large groups of identical users (students) and allow for easier attribute release policies (privacy preservation)
- **EGEE Architecture**
  - certificates, but use MyProxy if needed to bridge federations, e.g. using A-Select
  - make certificate validation mode direct (OCSP) and simpler (RFC3820 proxies, standard OpenSSL/javax.security)
  - allow checking mode extensions like policy OIDs
- **EGEE-1:** only move to standard TLS + RFC3820

Local Authorization

- **Roadmap**
  - all assertions carried as SAML statements
  - all local (and global) policies expressed in XACML
  - separate authorization service using standard protocols
  - site policy, *AND*-ed with user and VO policy, evaluated together
  - policy evaluation never requires special local privs (`root')
- **EGEE Architecture**
  - Authorization Framework (Java) and LCAS (C/C++ world)
  - both provide set of PDPs
    (but slightly more PDPs will be there for the AuthZ FW)
  - Authorization Service via OGSA-AuthZ-WG spec
  - PDPs:
    user white/blacklist, VOMS-ACL, Proxy-lifetime, OID checks, peer-system name validation, central CRL checking

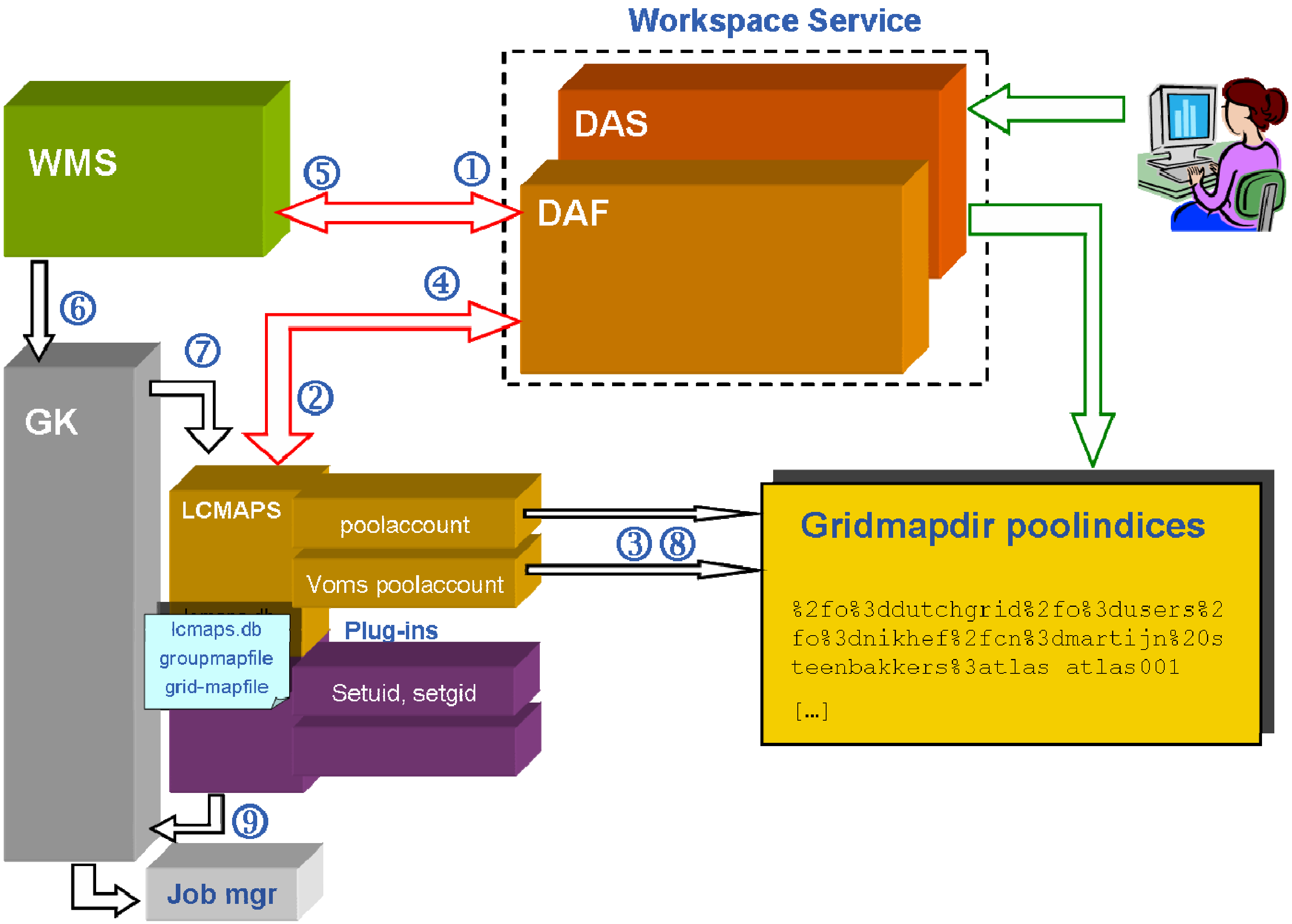
Local Authorization in EGEE-1

- **EGEE-1 Implementation**
  - Only a limited set of PDPs:
    - ban/allow and VOMS-ACL
  - Authorization interface is non-standard (C/C++)
  - All evaluation is in-line:
    - source modifications needed to old services (GT gatekeeper, GridFTP server)
  - No separate authorization service (no site-central checking)
  - Policy format is not XACML everywhere (but GACL)

Isolation

- **Roadmap**
  - virtualization of resources (VM) *or* assigning of local credentials
  - should be indistinguishable from ‘outside’
- **EGEE Architecture**
  - only based on credential mapping
  - do as little as possible with ‘root’ privileges: su-exec
  - minimizing local management: poolaccounts & poolgroups
  - credential mapping and manipulation: LCMAPS
  - management capabilities on these accounts: WSS
- **EGEE-1 implementation**
  - LCMAPS and WSS available
  - limited access control capabilities to the WSS (mapfile only)
  - lightweight su-exec implemented by heavy-weight Gatekeeper

LCMAPS and the WSS



```
%2fo%3ddutchgrid%2fo%3dusers%2
fo%3dnikhef%2fcn%3dmartijn%20s
teenbakkers%3atlas atlas001
[…]
```

System Integration components

Additional options for system integration

- **NSS-grid**
  - make regular commands (ls, top) show grid DNs
  - linked to credential mapping and auditing system
- **grid-PAM**
  - retrofit existing services with grid security
  - gsi-ssh, gsi-cvs, …

- **No explicit requirements within the project**
- **highly popular outside, with smaller installations**
- **SAC architecture should allow for these options**
- **no effort assigned until real requirement is there**

Auditing

- **Common logging (format) & reporting is a prerequisite *but not yet defined for the middleware suite as a whole***

- **But a minimum must and will be provided**
  - events are traceable through the system
  - storing audit trails left to conventional means (*syslog*)
  - deployment suggestion provided: secure *syslogs*

- **Credential mapping repository: “JR”**
  - linked to credential system LCMAPS and the JobManager (JR)
  - a version will be available

Network Considerations

Conflicting requirements

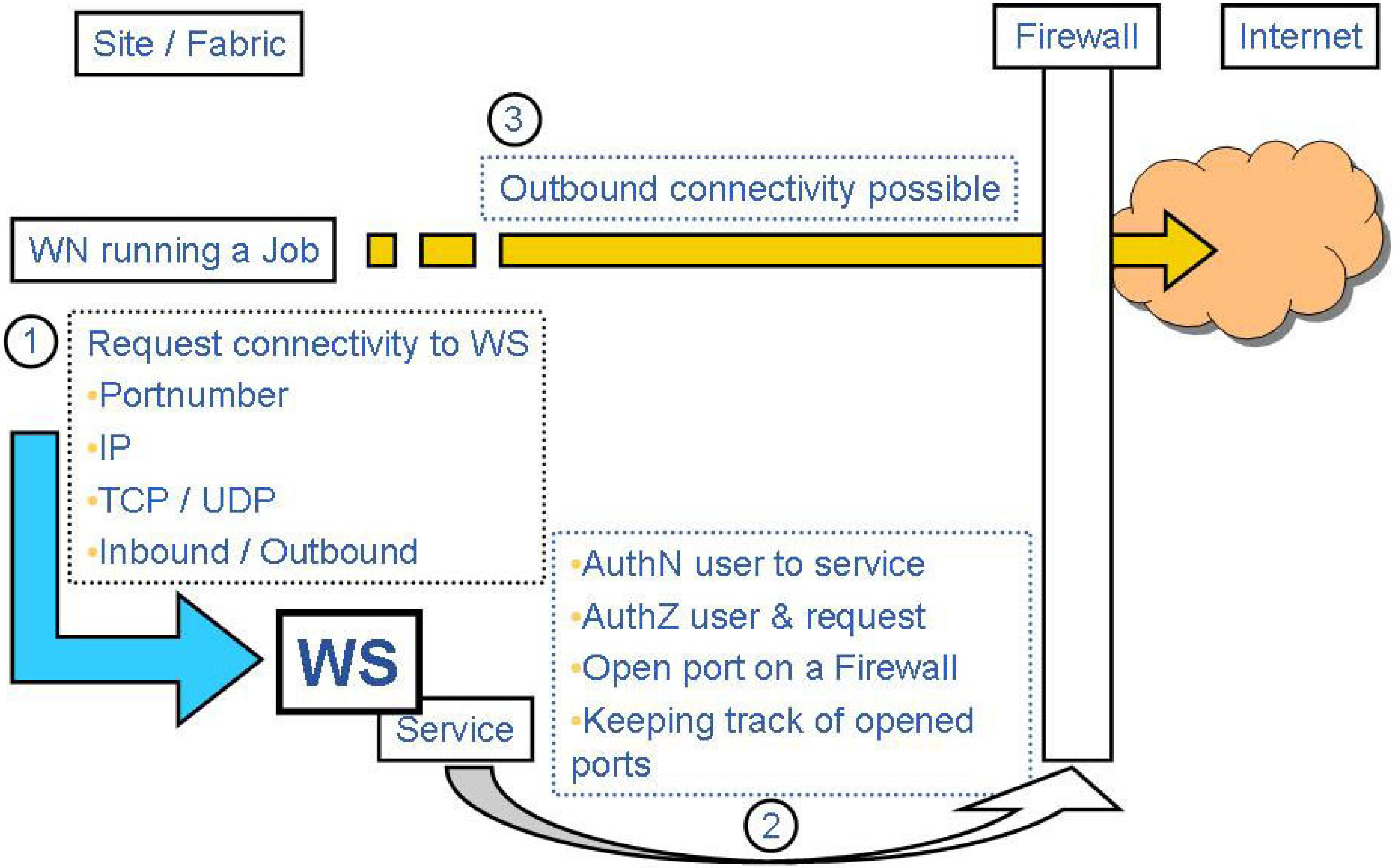
- Sites: worker nodes shall have no global connectivity
- Apps: worker nodes must have full connectivity

Proposed solution (JRA3 part)

- Dynamic Connectivity Service (DCS) “Site Proxy”
- policy-controlled connections to the outside world
- grid service interface,
  common (interface & development) with JRA4
- deployment scenario:
  dedicated boxes, flexible packet routing

Not in EGEE-1

DCS: logical workflow



Site / Fabric

Firewall

Internet

3 Outbound connectivity possible

WN running a Job

1 Request connectivity to WS
- Portnumber
- IP
- TCP / UDP
- Inbound / Outbound

WS

Service

- AuthN user to service
- AuthZ user & request
- Open port on a Firewall
- Keeping track of opened ports

2

Summary

DJRA3.2 Site Access Control architecture

- **Roadmap**
  - sites must be in control and always secured
  - users will not know they have single sign-on,
    but will complain if they don’t have it
  - large number of efforts world-wide to address this:
    AuthN and AuthZ are extremely active fields
  - Roadmap takes these developments into account
    e-IRG Roadmap, GN2, Globus, initiatives in academia
- **EGEE architecture**
  - aim for better mechanisms, but with consistency in mechanisms
- **EGEE-1**
  - deployment of proven technology
    (but which had not been used before in LCG2 yet)

References

DJRA3.2: Site Access Control Architecture

https://edms.cern.ch/document/523948/

JRA3 Team (NIKHEF, UvA, KTH/PDC, UH/HIP, UiB)
with help from JRA1 DM cluster (security) (CERN)