Unicore Security
and its Way to Interoperability

Daniel Mallmann – Research Centre Juelich

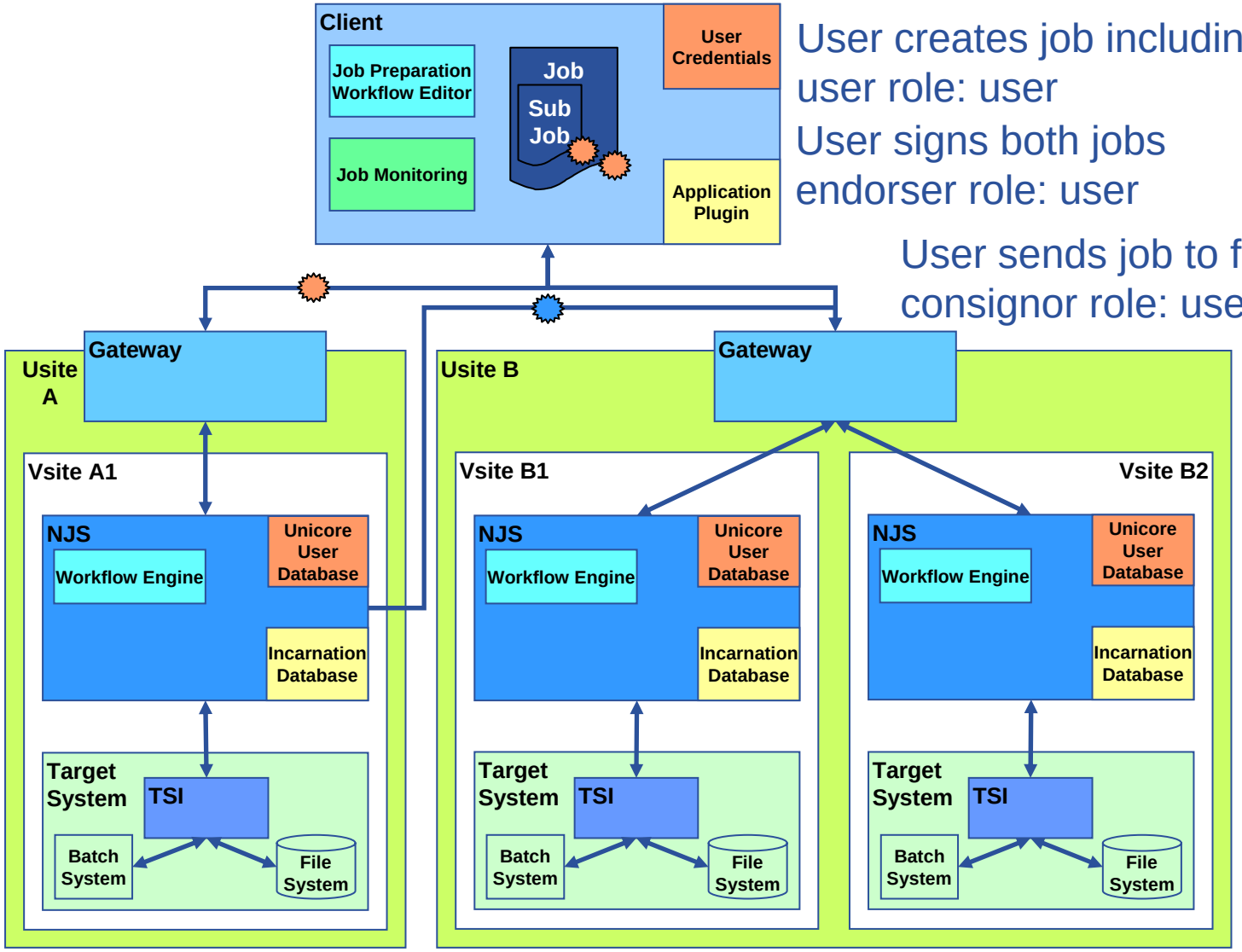
MWSG Meeting, CERN 14-15 November 2006

www.eu-egee.org

Information Society and Media

Unicore Architecture and Roles

Enabling Grids for E-sciencE



User creates job including subjob
user role: user
User signs both jobs
endorser role: user

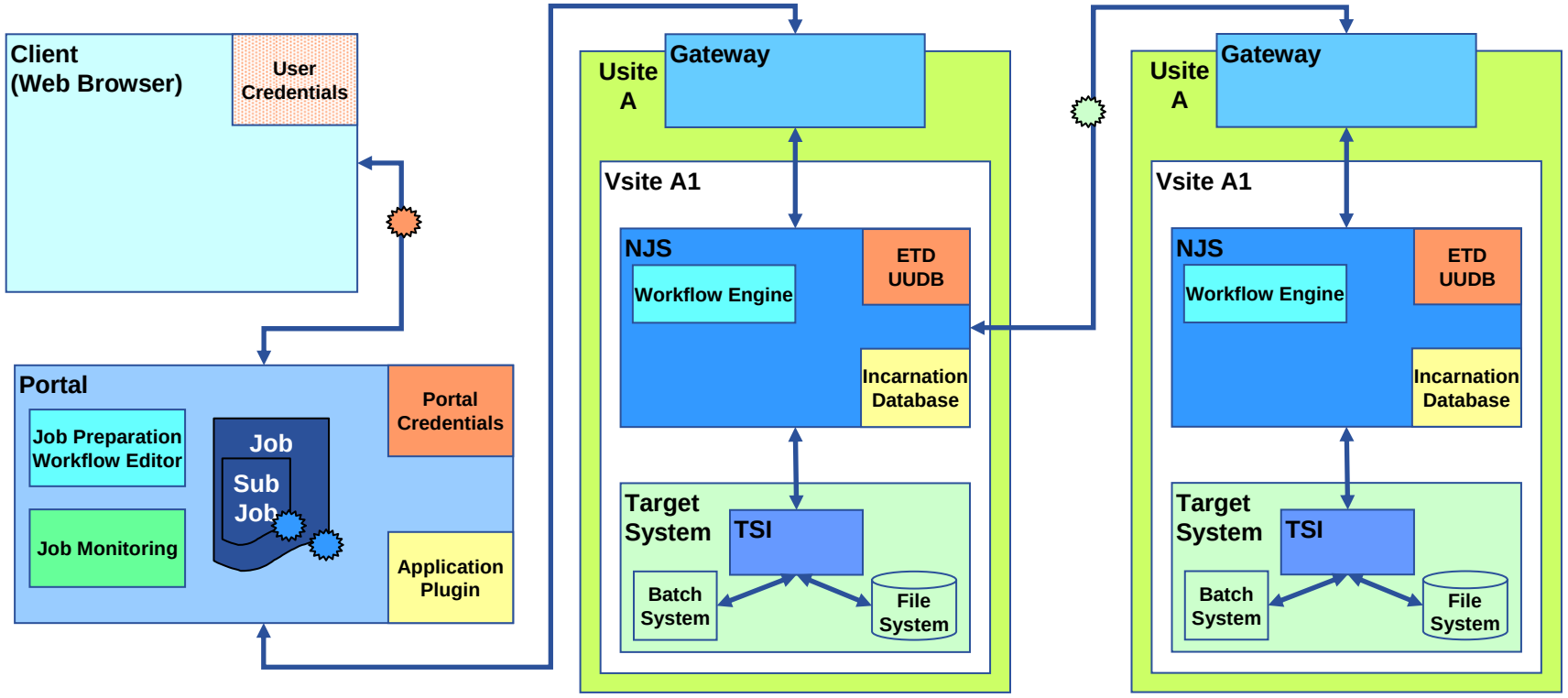
User sends job to first NJS
consignor role: user

NJS unpacks job and sends subjob to second NJS
user role:user
endorser role: user
consignor role: NJS

User credentials NJS server credentials

Unicore Explicit Trust Delegation

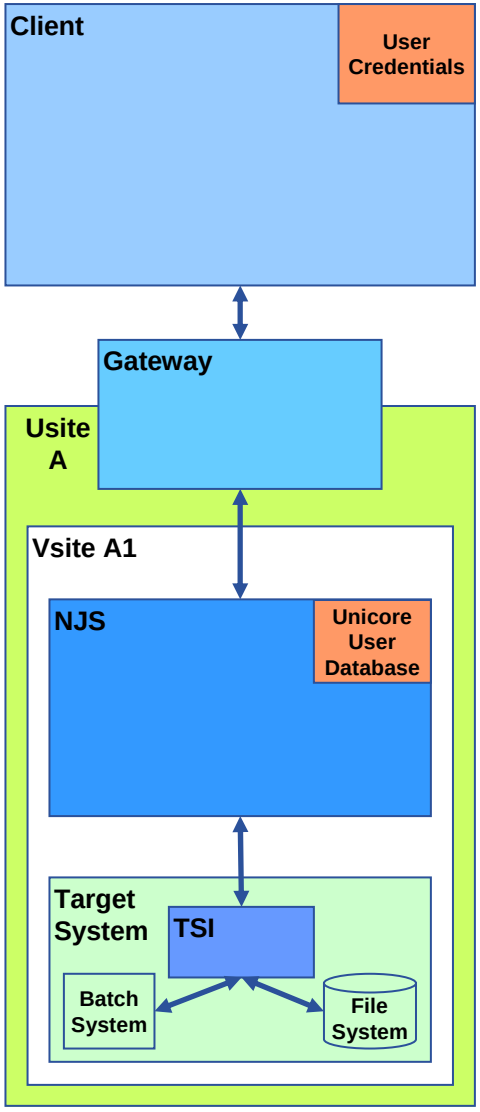
User authenticates at portal (not necessarily using credentials)
User creates job in portal - user role: user
Portal signs job - user role: user - endorser role: portal
Portal sends job to NJS - user role: user - endorser role: portal - consignor role: portal
NJS unpacks job and sends subjob to second NJS
user role: user - endorser role: portal - consignor role: NJS



User credentials portal credentials NJS credentials

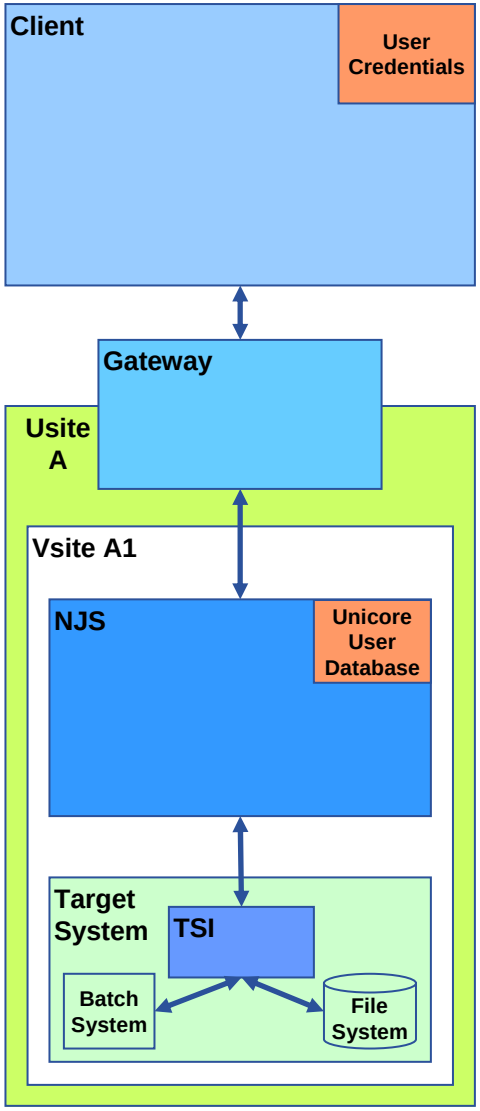
Unicore Security Components

- **Transport Level**
  - Client-Gateway and Gateway-NJS connections are mutually authenticated client-server SSL (consignor key and Gateway/NJS key)
- **Message Level**
  - All Messages are signed with the endorser key
    - Still looking for a high-performance signing mechanism for the Unicore 6 Web services implementation
- **NJS and Gateway Credentials**
  - X509 certificates
  - PKCS12 format
  - Password usually in configuration file



Unicore Security Components

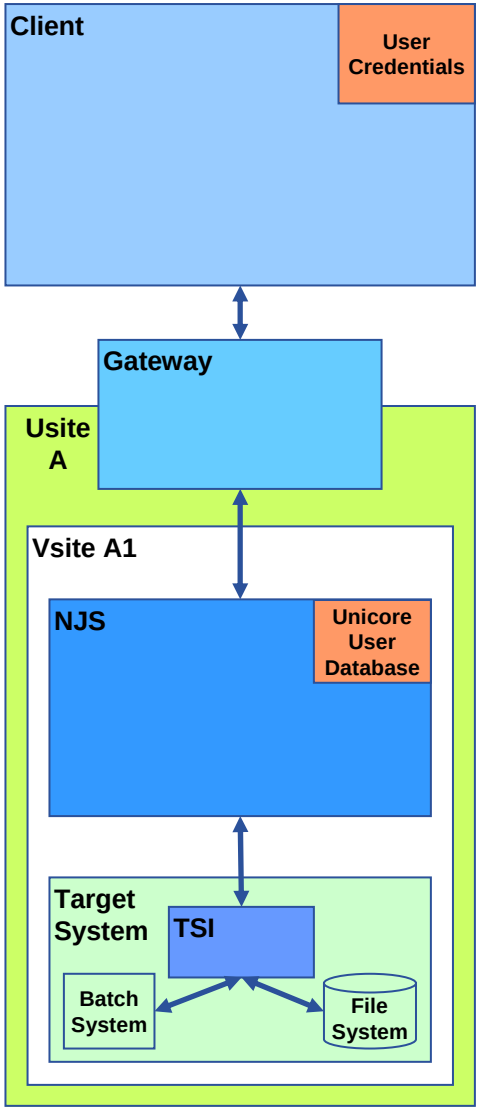
- **User Credentails: Unicore Keystore**
  - File in configuration directory of the Unicore client
  - X509 certificate
  - Private key PKCS12 format
  - List of trusted CAs
  - List of trusted developer certificates for application plugins
- **User Authentication: Unicore Gateway**
  - List of trusted CAs
  - List of URLs of the certificate revocation lists (CRLs)



Unicore Security Components

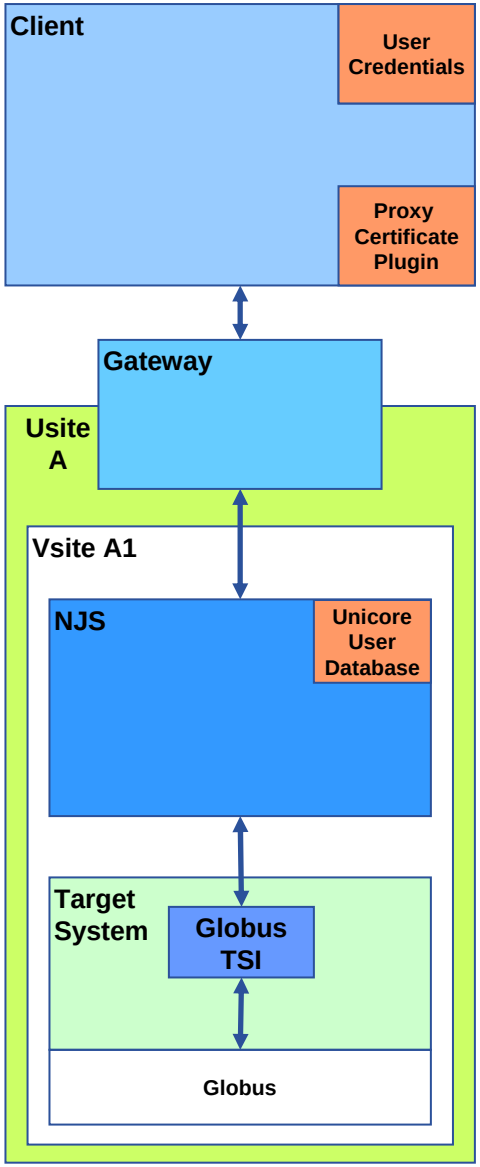
- **User Authorization:**
  Unicore User DataBase
  - Mapping of
    user certificates to Xlogin on target system
  - Different implementations
    - Java class with plain file
    - Web service with xml file
    - DEISA evaluates only Distinguished Name of certificate

- **Delegation:**
  NJS – Explicit Trust Delegation
  - Each trusted agent has to be added to the UUDB
    - Xlogin prefix = agent-



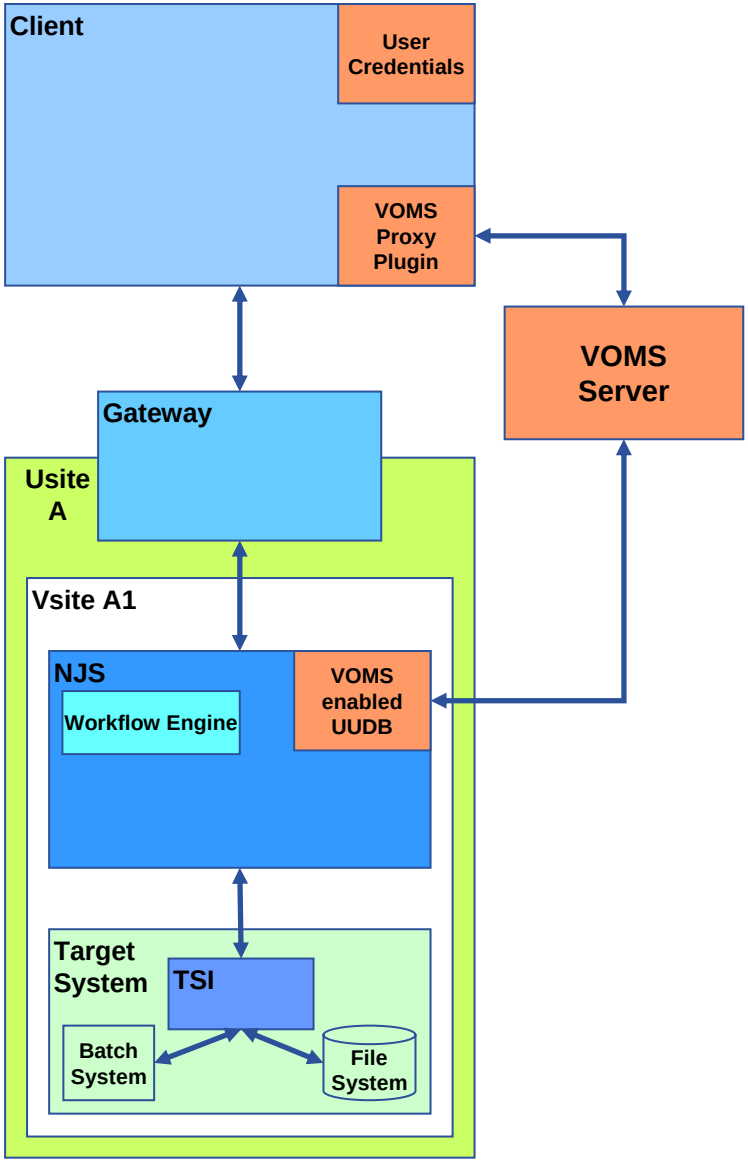
Unicore Security Components

- **Unicore – Globus Interoperability: Globus Proxy Certificates**
  - Generated by Proxy Certificate Plugin
  - Extracted from Unicore job at NJS
  - Send to the Globus TSI



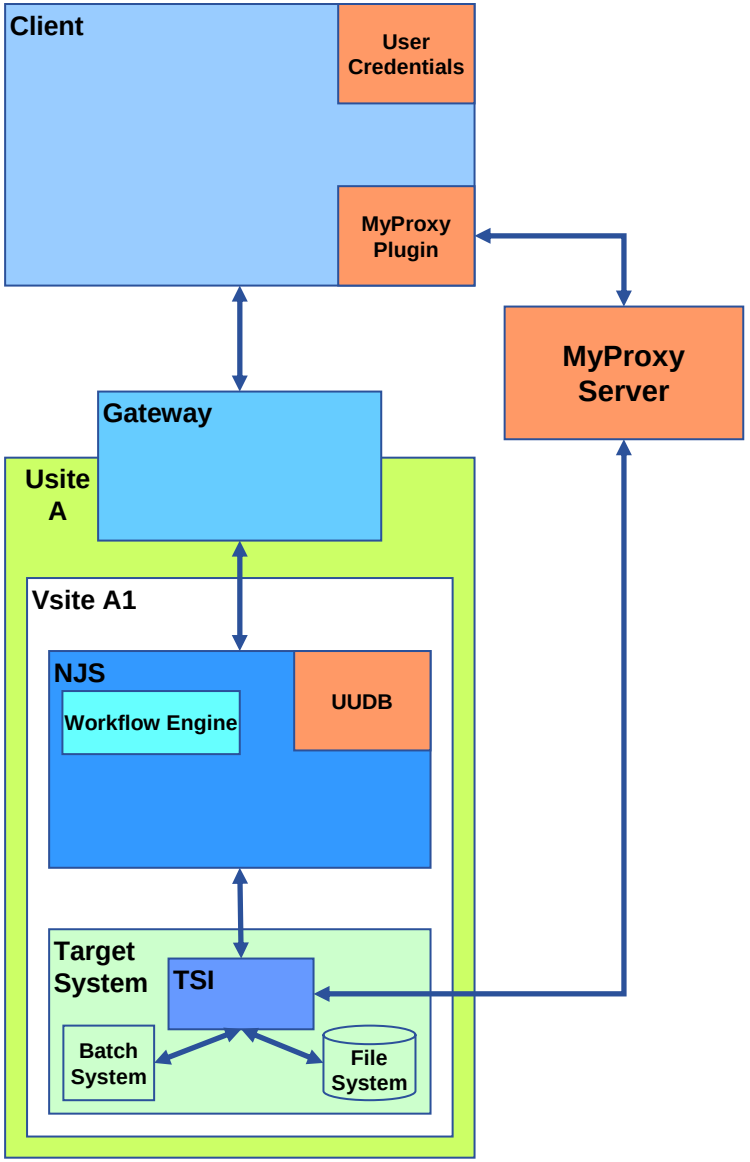
Missing Components in Unicore

- **VO Management**
  - HPC background:
    access granted to single users
  - Possible integration scenario:
    - VOMS proxy plugin generates VOMS certificate (voms-proxy-init)
    - NJS uses VOMS enabled UUDB for user authorization

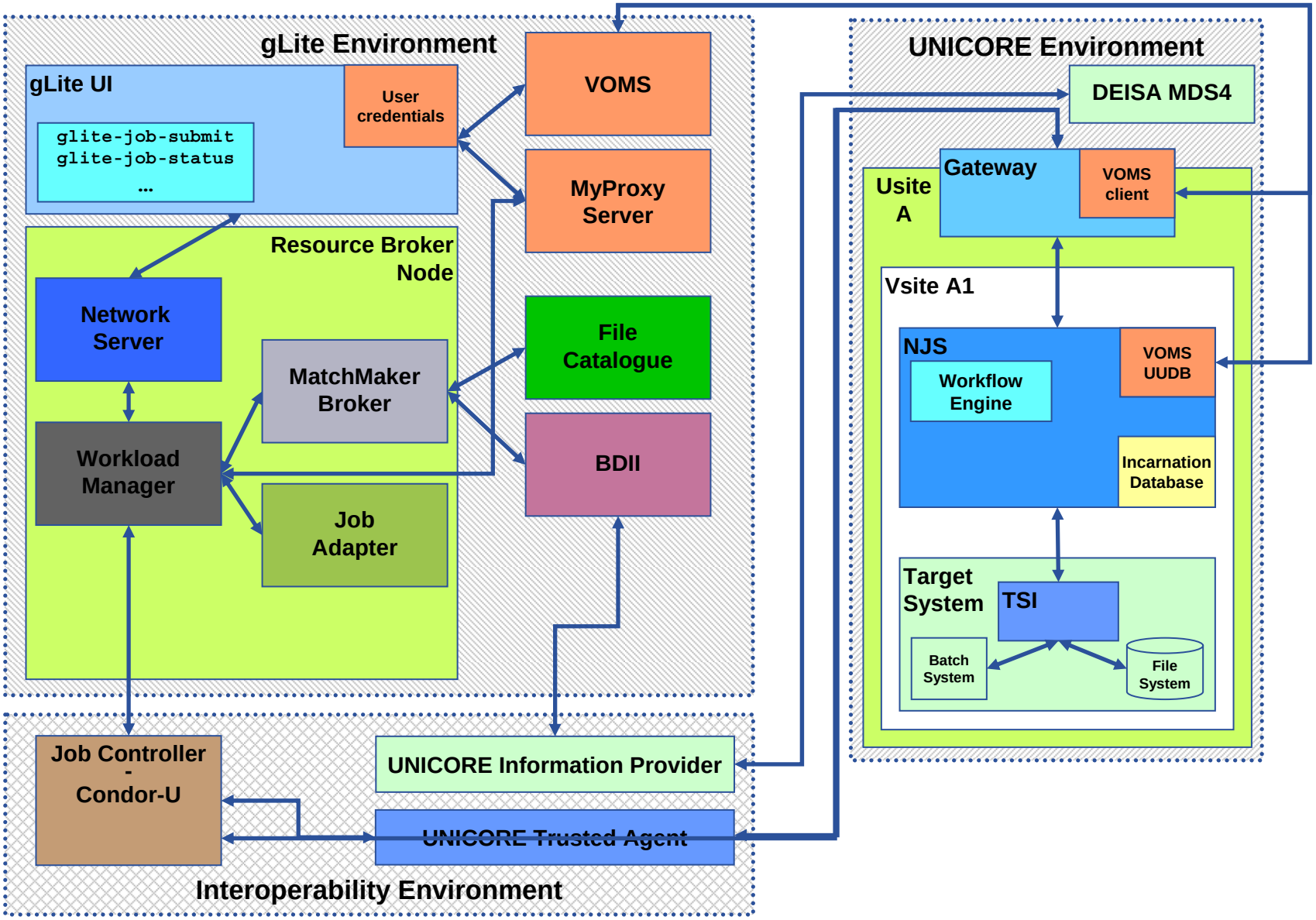


Missing Components in Unicore

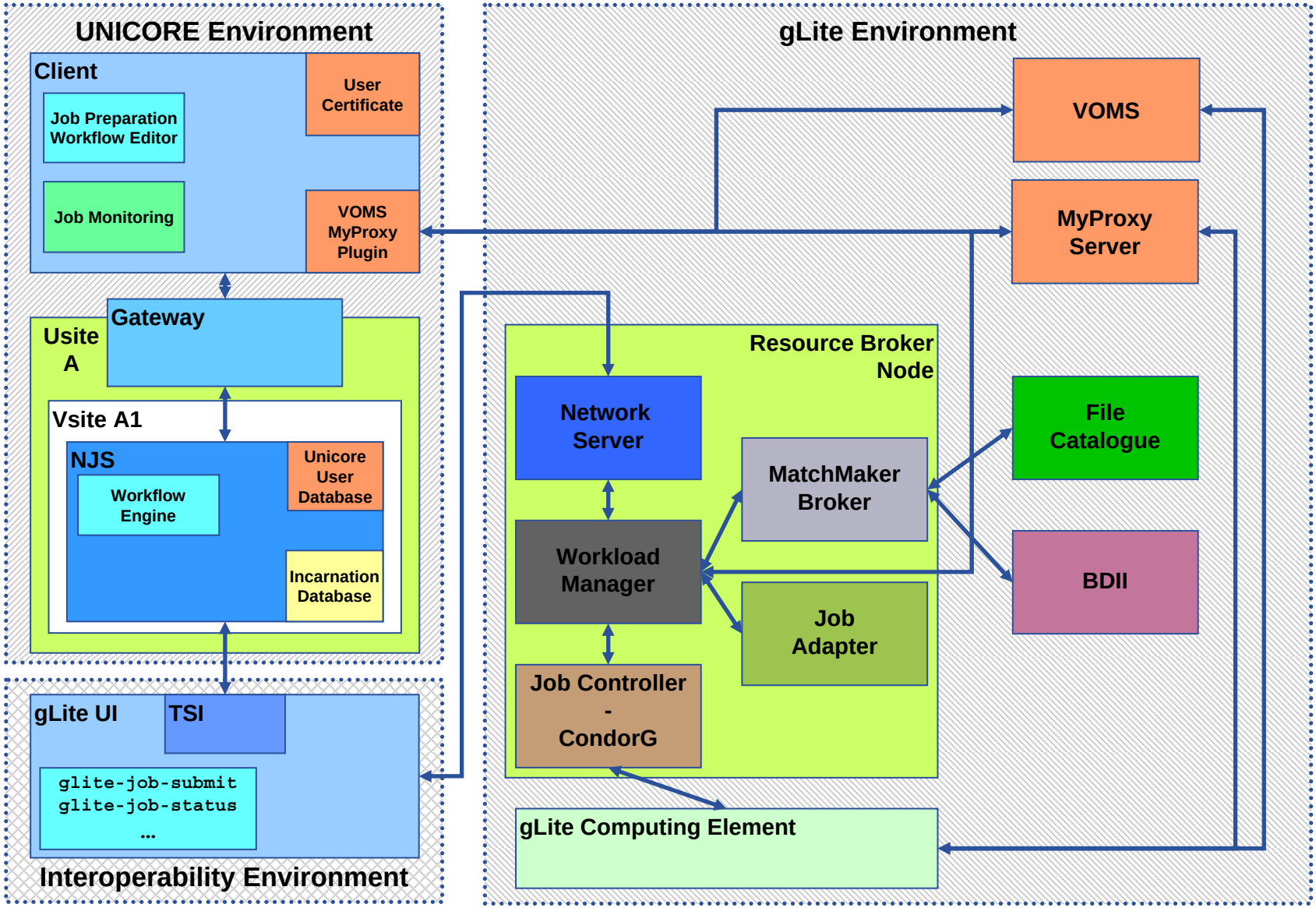
- **Proxy Service**
  - Job send to batch system
  - Access only to local file systems (GPFS, NFS, …)
  - No additional “Grid authorization” necessary (and possible)
  - Possible integration scenario:
    - MyProxy plugin generates and stores proxy certificate in MyProxy Server
    - TSI accesses MyProxy server to obtain user credentials



JobSubmit gLite to Unicore



Unicore to gLite



Next Steps

- **VOMS Integration**
  - Addressed in OMII-Europe JRA1
    - Focus on Unicore 6
    - EGEE-II needs solution for Unicore 5
- **MyProxy Integration**
  - Has to be addressed in OMII-Europe JRA3
  - Offers access to
    - “Grid storage”
    - OGSA-DAI (?)
    - Applications using remote services
  - Strong reservations within Unicore community
- **Fine grained Authorization**
  - Application level
  - Methods on properties

Some Questions

- **VOMS-Proxy-Init**
  - Java version available?
- **VOMS Client (similar to component running on CE)**
  - Java version available?
- **MyProxy Client**
  - Java version available?
- **WMS**
  - Does it access VOMS server?
- **Server Credentials**
  - How are they stored?
- **Integration of OGSA-BES Interface into ICE (Interface to CREAM Environment)**
  - Access to Unicore, gLite, Globus
  - How is authentication and authorization handled?

Future

Users can access applications on any Gird infrastructure without worrying about credentials