jAliEn security model

Nikola Hardi
nhardi@cern.ch

14/05/2019 - ALICE T1/T2 Workshop, Bucharest, Romania

Security on the Grid

- **Two kinds of grid services**
  1. File storage
  2. Grid jobs

- **Three entities**
  1. Central services
  2. User machines
  3. Grid sites

Objectives	On the grid
authentication	obtaining identity
authorization	determining permissions
integrity	keeping data unmodified
confidentiality	encrypting secret information
resilience	Isolating workspaces
non-repudiation	protecting ownership and origin

Security characteristics of the Grid

1. **Distributed** services, communicating over **public and insecure network**
2. Computation on **shared resources**
3. **Uncontrolled client environment**, users' personal devices
4. The framework connects users and computing sites as **legal entities**
5. **Multiple origins** of data and code

Security requirements for the Grid

1. **Authentication**: mutually authenticated communication parties
2. **Integrity**: assured integrity control
3. **Confidentiality**: possible confidential transmission of credentials
4. **Verifiable Grid files**: certified identity and originator of a Grid file entry
5. **Verifiable origin of Grid jobs**: certified originator
6. **Verifiable processing of Grid jobs**: certified processing, modif. and delegation

Presentation summary

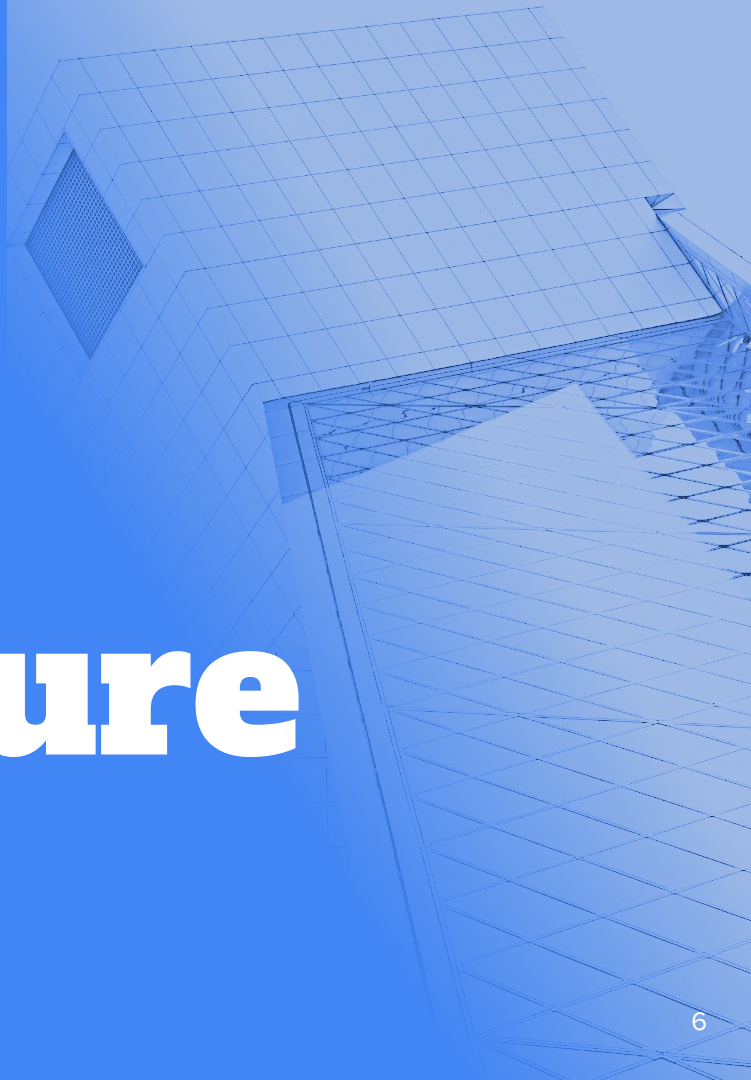
JAlien infrastructure overview

1. JAlien components
2. Communication channels
3. Component identities

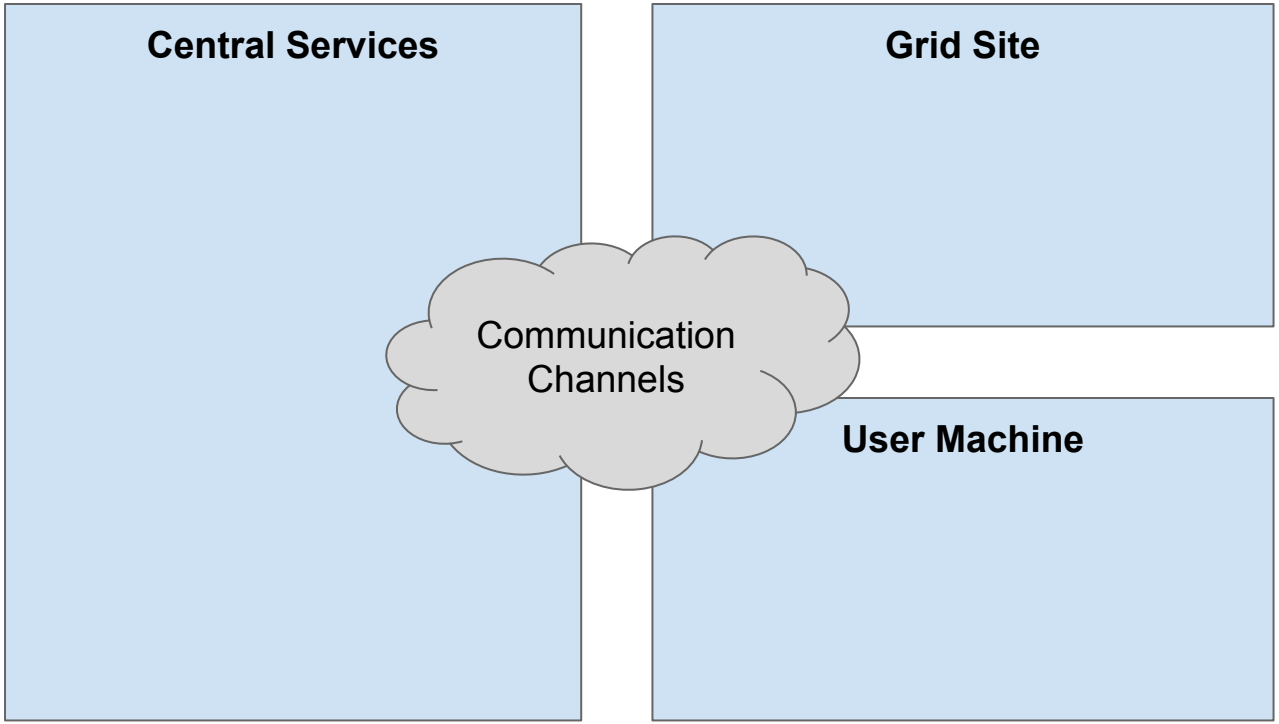
JAlien commands examples

1. Listing files
2. Transferring files
3. Running grid jobs

Infrastructure overview

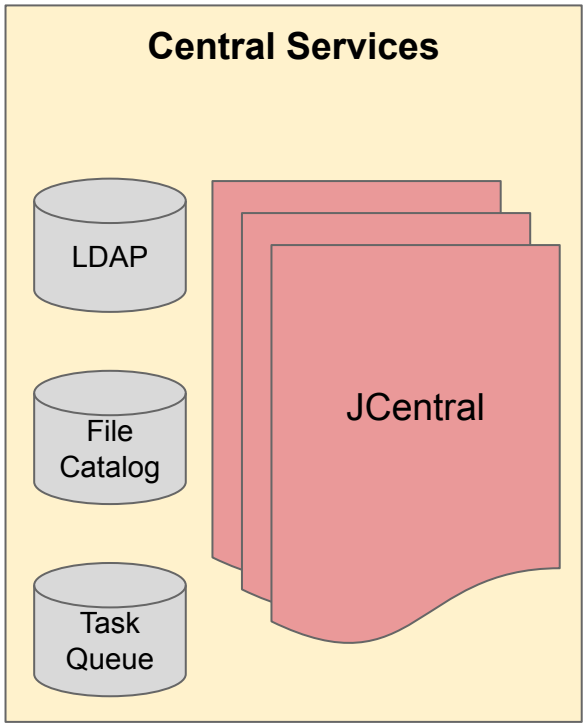


Entities on the Grid



JAlien components:

1. Central Services



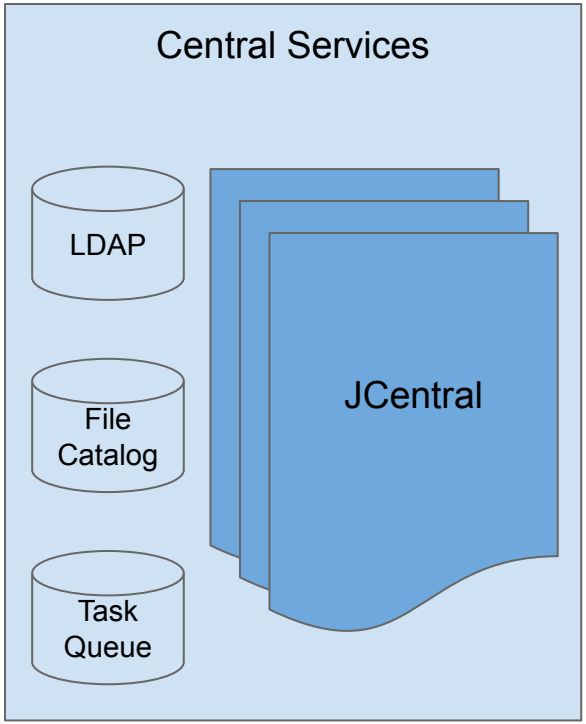
Grid Site

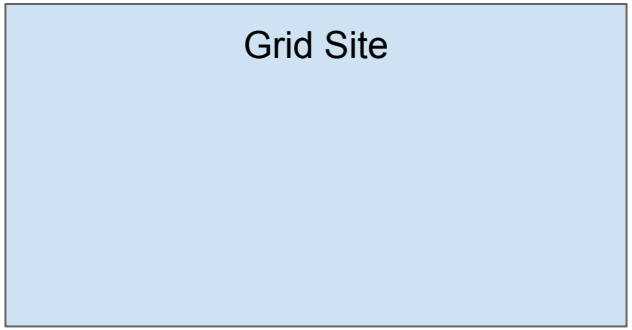
User Machine

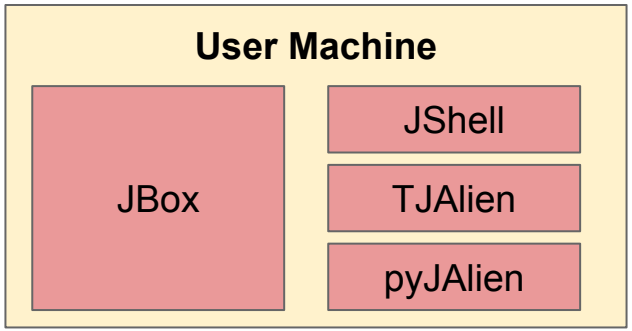
JAlien components:

- JCentral

2. User Machines



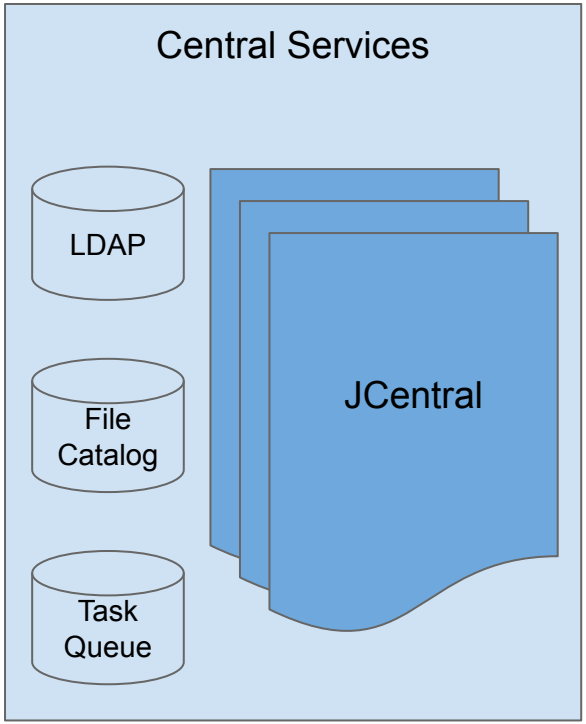


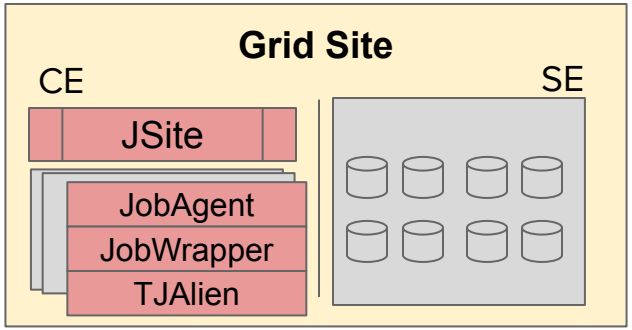


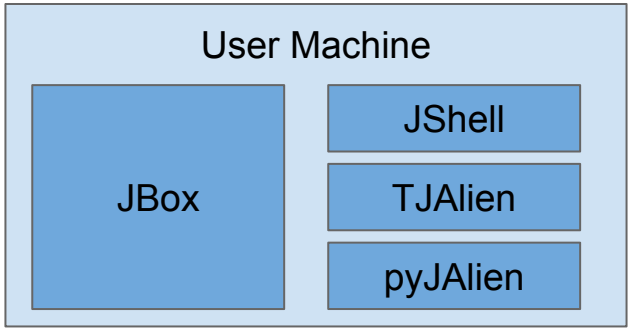
JAlien components:

- JCentral
- JBox
- JShell
- TJAlien
- pyJAlien

3. Grid Sites



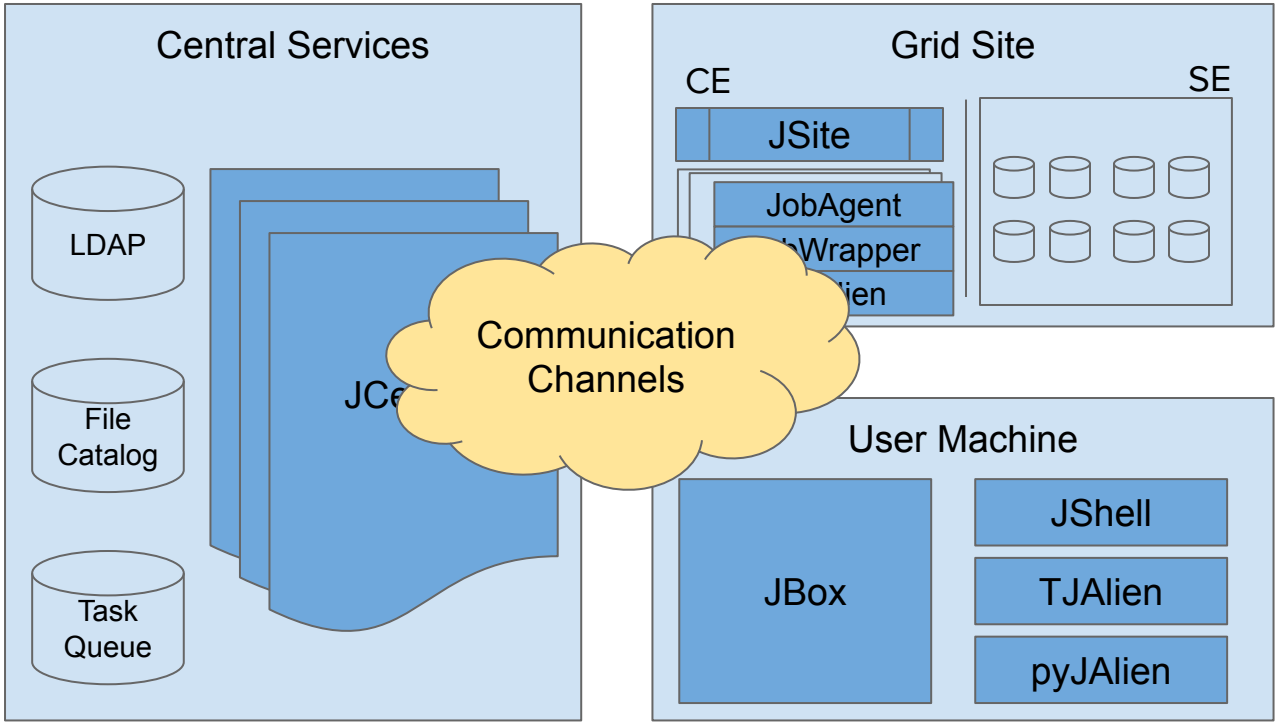




JAlien components:

- JCentral
- JBox
- JShell
- TJAlien
- pyJAlien
- JSite
- JobAgent
- JobWrapper

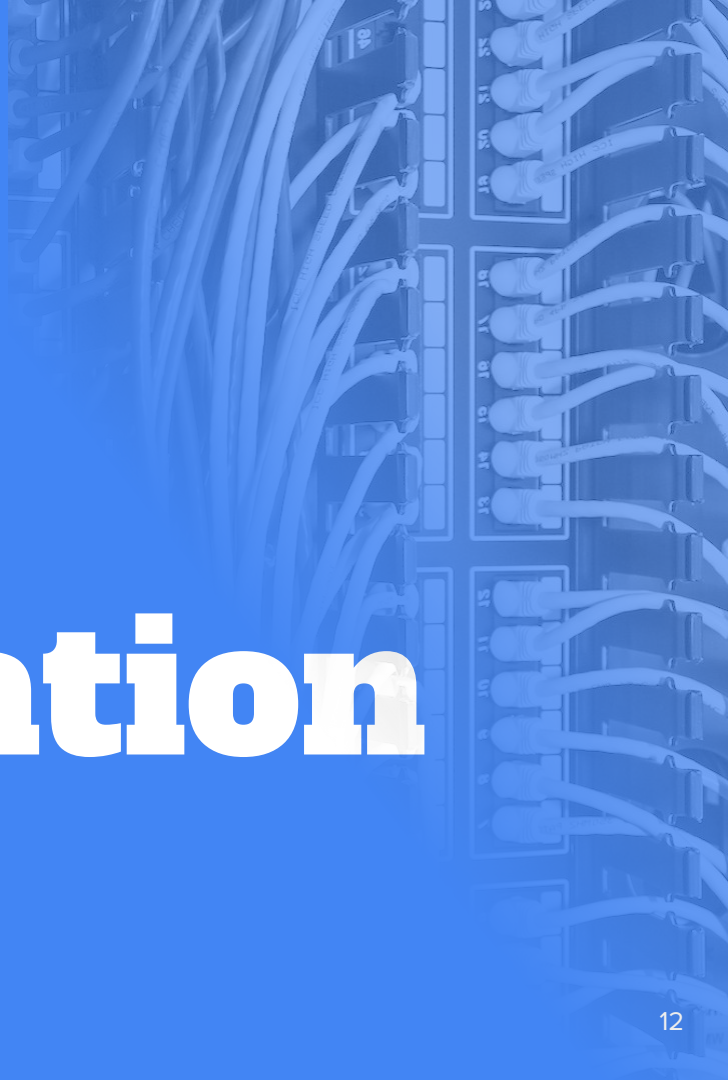
3. Grid Sites



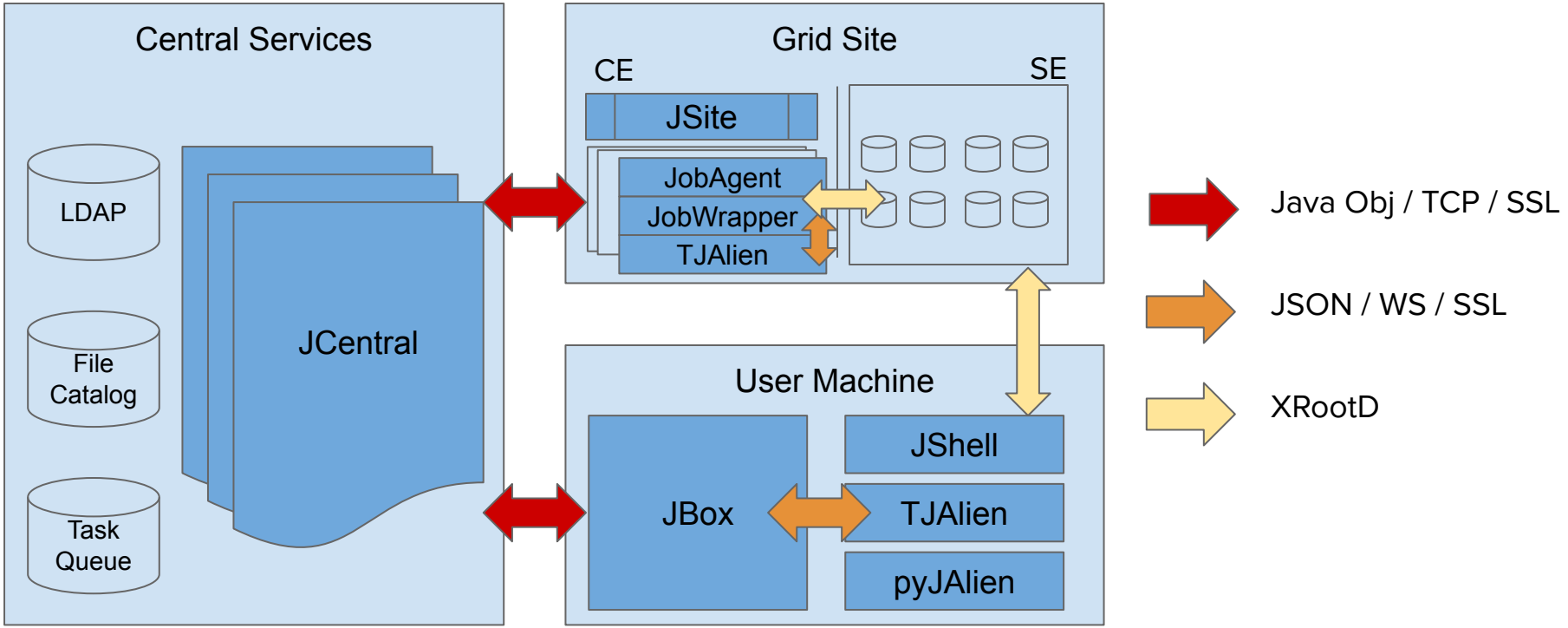
JAlien components:

- JCentral
- JBox
- JShell
- TJAlien
- pyJAlien
- JSite
- JobAgent
- JobWrapper

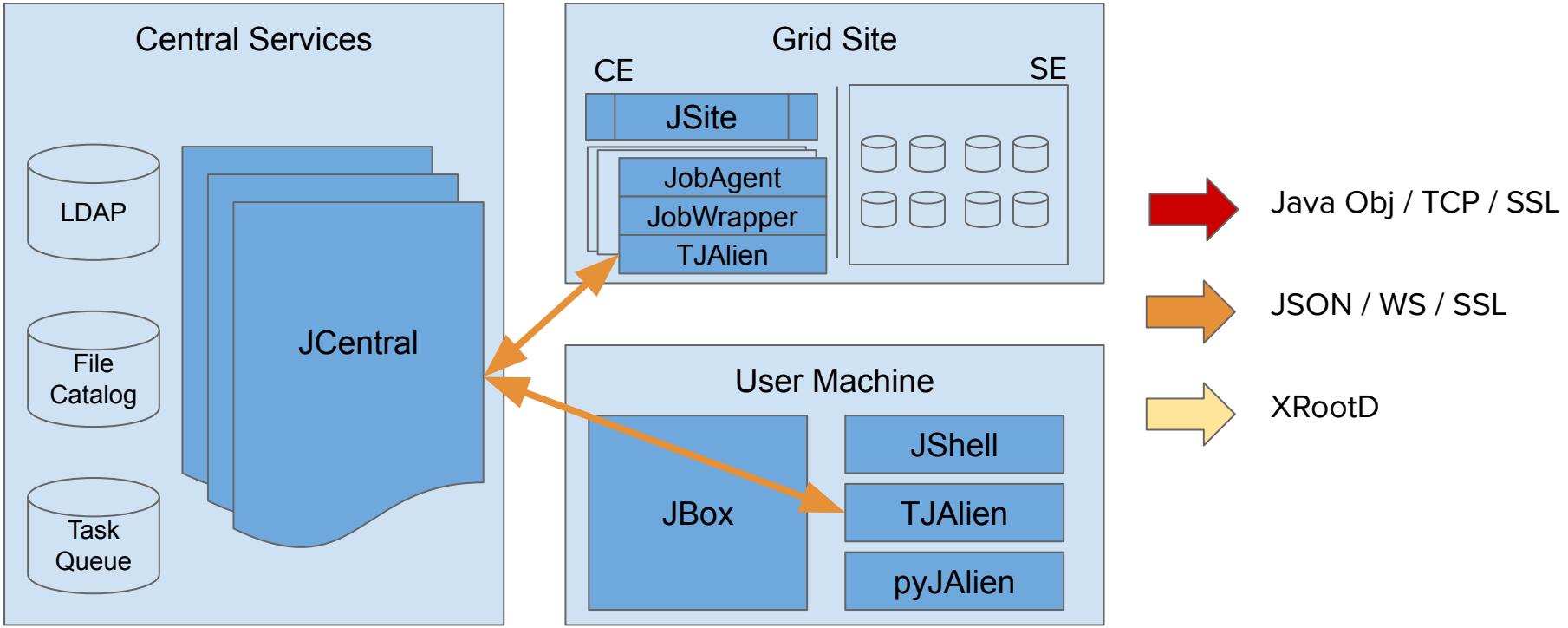
Communication channels



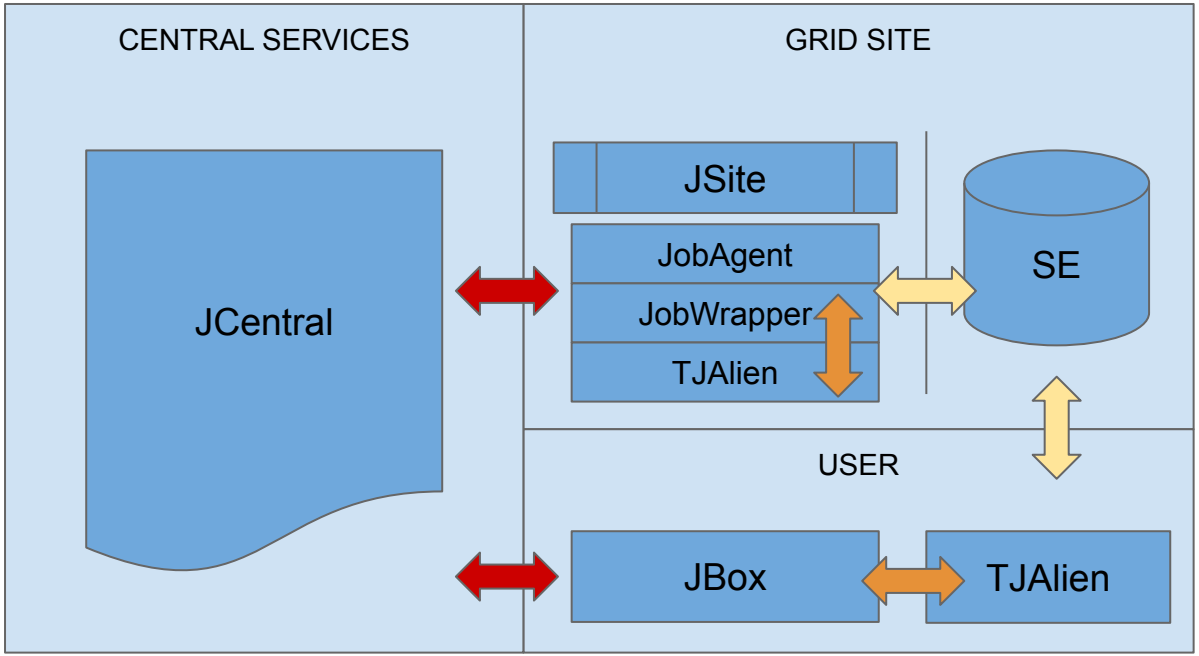
Communication channels - optimal conf



Communication channels - failback



Communication channels - simplified



Java Obj / TCP / SSL

JSON / WS / SSL

XRootD

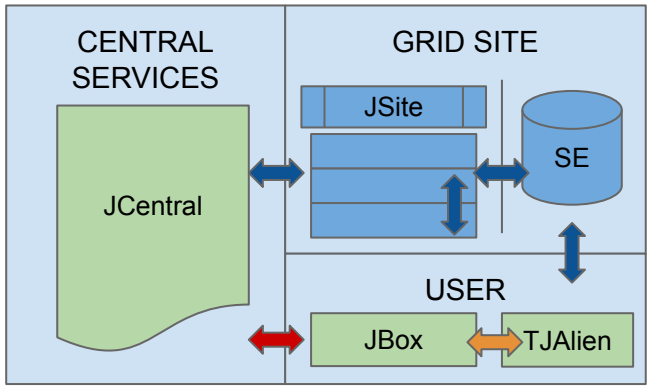
Security Credentials

Connecting to the Grid

Crypto-material
- Alien CA
- JCentral host certificate
- Grid certificate with key pair
- Token certificate signed by Alien CA

Connection procedure
1. JCentral is running with a host certificate
2. JBox starts
   - Loads grid certificate and decrypts priv. Key
   - Connects to JCentral (Java / SSL / TCP)
   - Requests a token from JCentral
   - Stores token certificate locally
3. An end-client starts (JShell / TJAlien)
   - TJAlien loads token
   - TJAlien connects to JBox (JSON / S WebSocket)

Tokens are signed by Alien CA, and have limited capabilities



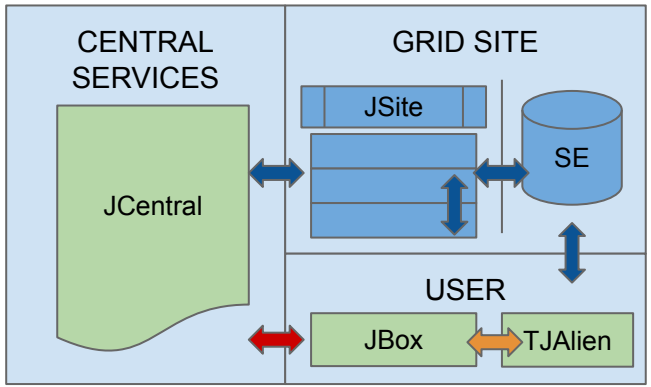
Security properties of JAlien tokens

- Signed by Alien CA, in contrast to the X.509 proxy certificates signed by users
- Limited capabilities
  - Full grid certificate is required for requesting another token
- Multiple kinds of token
  - For end-user clients
  - For the job agent
  - For the grid job
- Valid for limited period
  - On a user machine the token is transparently refreshed by JBox in presence of the full grid cert
- No more alien-token-init!
- Unified authentication mechanism for all JAlien components

jAliEn commands

1. Listing files

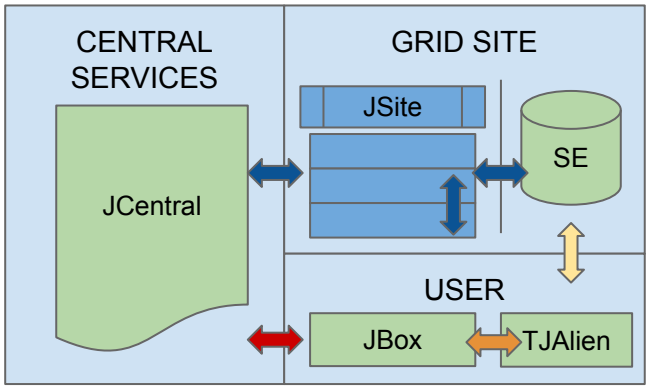
- The connection from client to central services is up (TJAlien / JBox / JCentral)
- TJAlien sends an “ls” command (JSON/WS)
- JBox receives it, sends a request to JCentral (Java/TCP)
- JCentral
  - Authorizes the request
  - Accesses the File Catalogue database (MySQL)
  - Replies with compressed serialized Java collection
- JBox forms the final reply (JSON)
- The client uses the result of “ls” command



2. File transfers to/from the Grid

- The connection is up
- The client issues a “cp” command
  - `cp file://hello.txt  alien://world.txt`
- JBox sends requests to JCentral
- JCentral
  - Records the request in the booking table
  - Replies with a signed envelope
- TJAlien receives the envelope
- TJAlien transfers the file to a SE
- SE replies with an auth envelope
- TJAlien confirms the transfer to JCentral
- JCentral clears the booking table

Grid file attributes
GUID: random identifier
LFN: symbolic name (path)
PFN: a physical location (replica)

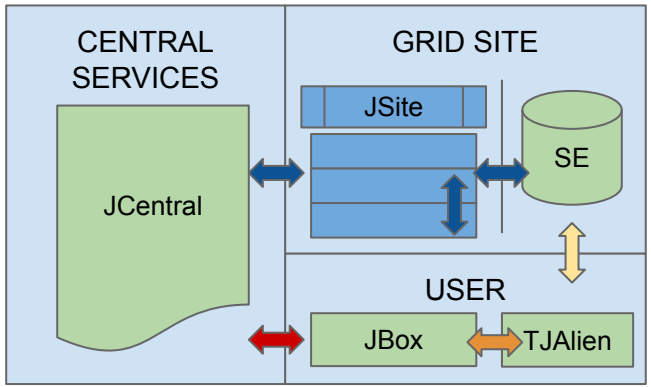


2. File transfers - comments

- The command and follow-up requests are communicated over secure channel
- JCentral reserves LFN using the booking table to prevent data race
- File transfer is done using high performance xroot protocol directly to an SE
- The clients must present authentic envelope for file transfers to SE
- On successful transfer, SE responds with another envelope
  - Containing the hash sum and size of actually stored file
- JCentral requires the second envelope to clear the booking table with OK
- If the transfer failed, JCentral deletes the entry from booking table
- The envelopes cannot be forged and file integrity is guaranteed

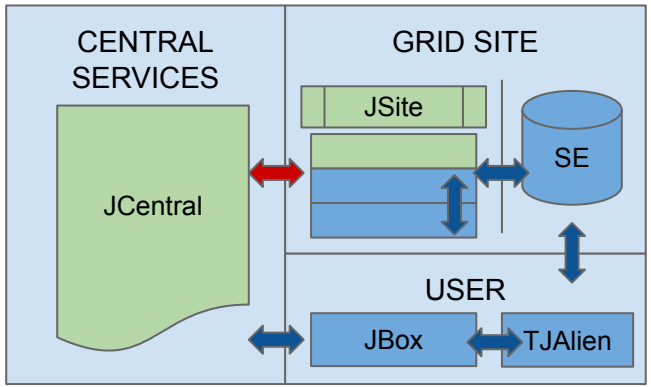
3. Running the Grid jobs - submit

- Connection from user client to JCentral is established
- The job description and input files are transferred to the Grid
- The user issues a “submit” command
- JCentral
  - Authorizes the request
  - Assigns a username to this request
  - Puts the request into the task queue
  - Waits for a JobAgent (pilot job) to call in



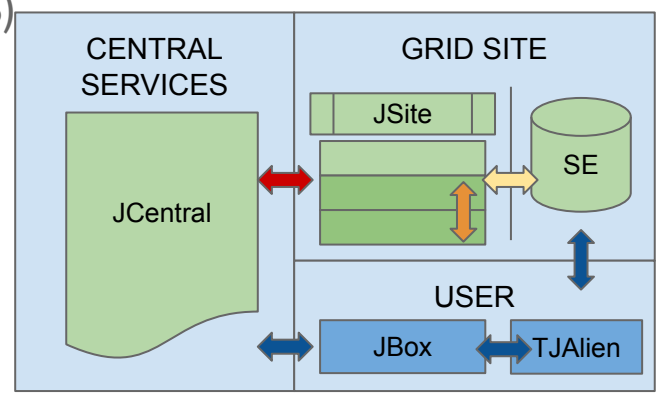
3. Running the Grid jobs - WN setup

- JSite runs with a host certificate on the VO box
- JSite requests a new set of JobAgent token from JCentral
- JSite inserts a JobAgent into batch queue
- A worker node becomes available
- JobAgent starts with its own credentials and connects to JCentral
- JobAgent requests a job
- JCentral replies with a job and job token



3. Running the Grid jobs - starting job

- JobAgent starts a container (isolated sandbox)
- JobWrapper starts inside of the container
- JobWrapper loads the grid job token
- JobWrapper prepares the workspace for running the payload
- The actual payload starts (aliroot / O2)
- TJAlien can connect to JobWrapper (JSON/WS)



Conclusion

JAlien token certificates

- The most interesting and important new feature of JAlien security model
- Tokens unify authentication methods for all components
- Credentials are signed by the centralised CA (Alien CA)
- Multiple kinds of tokens
  - User token
  - JobAgent token
  - Grid job token
- User tokens automatically renewed by a background agent (JBox)

Thank you!
Questions?

jAliEn security model

Nikola Hardi
nhardi@cern.ch

May 2019 - ALICE T1/T2 Workshop, Bucharest, Romania

Backup slides

List of publications about jAliEn security

- **Authorization of data access in distributed storage systems**, D. Fichtiger, 2005
- **A mediated definite delegation model**, S. Schreiner et al, 2011
- **Securing the AliEn file catalogue [...]**, S. Schreiner, 2011
- **A security architecture for the ALICE Grid Services**, S. Schreiner et al, 2012
- **Security architecture for e-Science Grid computing**, S. Schreiner, 2015
- **Integration of XRootD into the cloud infra for ALICE data analysis**, M. Kompaniets et al, 2015
- **The security model of the ALICE next generation Grid framework**, M. Pedreira et al, (unpublished)