The Security model of the ALICE next generation Grid framework

Miguel Martinez Pedreira[1], Costin Grigoras[2], *Volodymyr Yurchenko*[3], Maksim Melnik Storetvedt[4]

[1]Johann-Wolfgang-Goethe Univ. (DE), [2]CERN, [3]National Academy of Sciences of Ukraine (UA), [4]Western Norway University of Applied Sciences (NO)

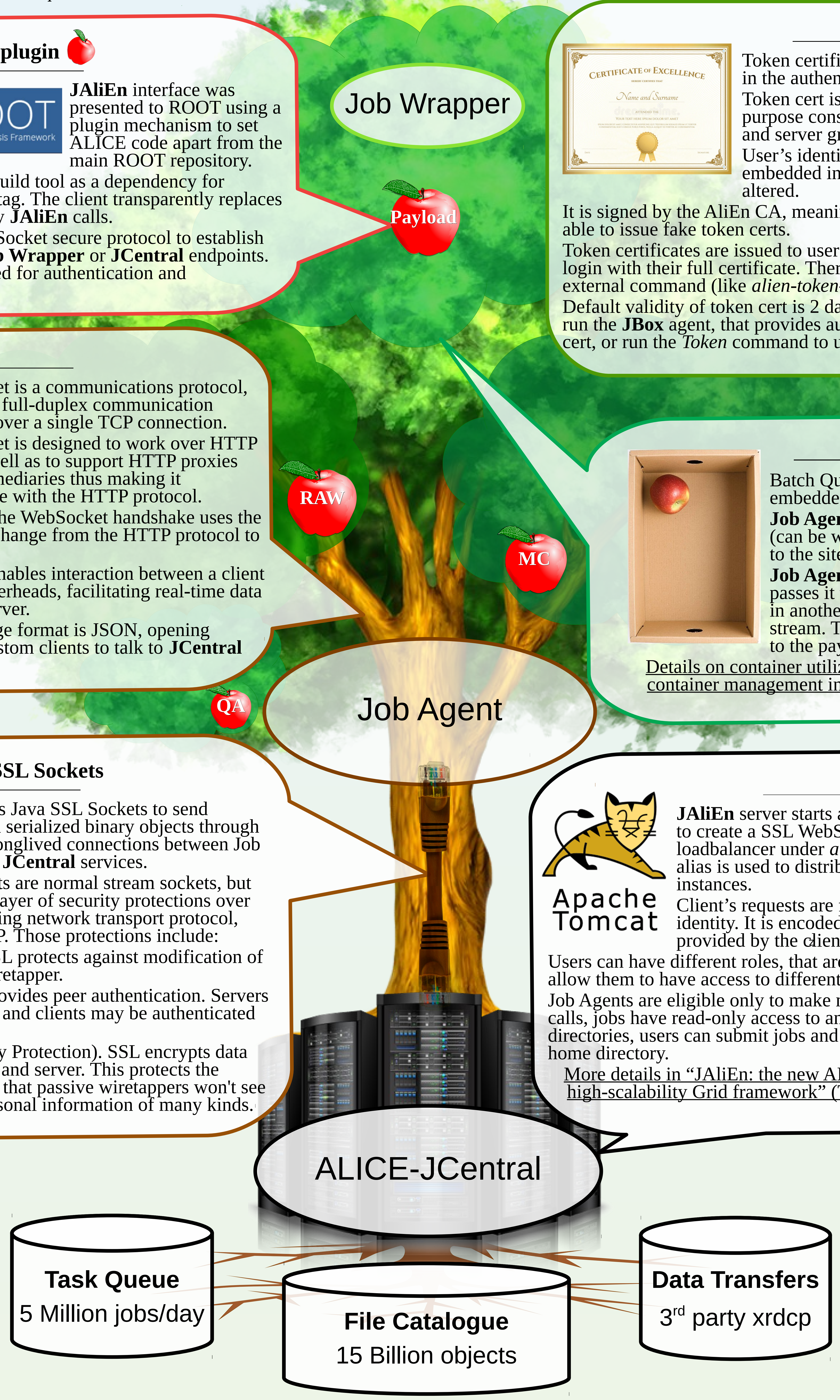
Abstract

JAliEn (Java-AliEn) is ALICE's next generation Grid framework which will be used for the top-level distributed computing resources management during the LHC Run3 and onward. While preserving an interface familiar to the ALICE users, its performance and scalability are an order of magnitude better than the currently used system.

To enhance the **JAliEn** security, we have developed the so-called Token Certificates – short lived X.509 certificates, generated by central services automatically or on client's request. The new system provides fine-grained control over user/client authorization, e.g. filtering out unauthorized requests based on the client's type: generic user, job agent, job payload. These and other parameters (like job ID) are embedded in the token's DN by the issuing service and cannot be altered.

Client-side security implementation is also described in the aspect of interaction between user jobs and job agents. User jobs will use **JAliEn** tokens for authentication and authorization to the central **JAliEn** services. These tokens are passed to payload from the job agent through a pipe stream, thus are not stored on disk or in environment visible to anyone except the job process. Furthermore, we foresee improvement at the level of isolation of users' payloads by running them in containers.

While **JAliEn** doesn't rely on X.509 proxies, the backward compatibility is kept to assure interoperability with site services, which require these.



ROOT6 with JAliEn plugin

JAliEn interface was presented to ROOT using a plugin mechanism to set ALICE code apart from the main ROOT repository.

It can be built using AliBuild tool as a dependency for AliRoot with *user-root6* tag. The client transparently replaces legacy AliEn API calls by **JAliEn** calls.

JAliEn plugin uses WebSocket secure protocol to establish connections to **JBox**, **Job Wrapper** or **JCentral** endpoints. Token certificates are used for authentication and authorization.

Token Certificates

Token certificates replace X.509 proxies in the authentication scheme for **JAliEn**.

Token cert is a full certificate, but without purpose constraints, that do exist for user and server grid certificates.

User's identity and permissions are embedded in token cert and cannot be altered.

It is signed by the AliEn CA, meaning third-party entities are not able to issue fake token certs.

Token certificates are issued to users automatically after users login with their full certificate. Therefore there is no need to run external command (like *alien-token-init* or *grid-proxy-init* today).

Default validity of token cert is 2 days (configurable). Users can run the **JBox** agent, that provides automatic renewal of token cert, or run the *Token* command to update it manually.

Secure WebSockets

WebSocket is a communications protocol, providing full-duplex communication channels over a single TCP connection.

WebSocket is designed to work over HTTP ports as well as to support HTTP proxies and intermediaries thus making it compatible with the HTTP protocol.

To achieve compatibility, the WebSocket handshake uses the HTTP Upgrade header to change from the HTTP protocol to the WebSocket protocol.

The WebSocket protocol enables interaction between a client and a server with lower overheads, facilitating real-time data transfer from and to the server.

JAliEn WebSocket message format is JSON, opening approaches for creating custom clients to talk to **JCentral** services.

Isolated environment

Batch Queue starts a **Job Agent** with an embedded Job Agent token.

Job Agent instance is started in a slot (can be wrapped singularity / docker, up to the site to configure this).

Job Agent requests a Job token and passes it to the **Job Wrapper**, that runs in another isolated env, through a pipe stream. Thus the Job token is visible only to the payload and is not stored on disk.

Details on container utilization: "Grid services in a box - container management in ALICE" (Track 7, Thursday at 15:00)

Java Object Stream SSL Sockets

JAliEn uses Java SSL Sockets to send compressed serialized binary objects through persistent longlived connections between Job Agents and **JCentral** services.

Such sockets are normal stream sockets, but they add a layer of security protections over the underlying network transport protocol, such as TCP. Those protections include:

- Integrity Protection. SSL protects against modification of messages by an active wiretapper.
- Authentication. SSL provides peer authentication. Servers are usually authenticated, and clients may be authenticated as requested by servers.
- Confidentiality (Privacy Protection). SSL encrypts data being sent between client and server. This protects the confidentiality of data, so that passive wiretappers won't see sensitive data such as personal information of many kinds.

Server Security

JAliEn server starts an embedded Apache Tomcat to create a SSL WebSocket endpoint. A loadbalancer under *alice-jcentral.cern.ch* DNS alias is used to distribute connections to **JAliEn** instances.

Client's requests are filtered based on the requester identity. It is encoded in a peer token certificate provided by the client during handshake.

Users can have different roles, that are defined in LDAP. Roles allow them to have access to different resources in the Catalogue.

Job Agents are eligible only to make matching and job tracing calls, jobs have read-only access to anything except their output directories, users can submit jobs and have access only to their grid home directory.

More details in "JAliEn: the new ALICE high-performance and high-scalability Grid framework" (Track 3, Thursday at 14:45)

Poster #355

Computing in High Energy Physics conference, 9-13 July 2018, Sofia, Bulgaria