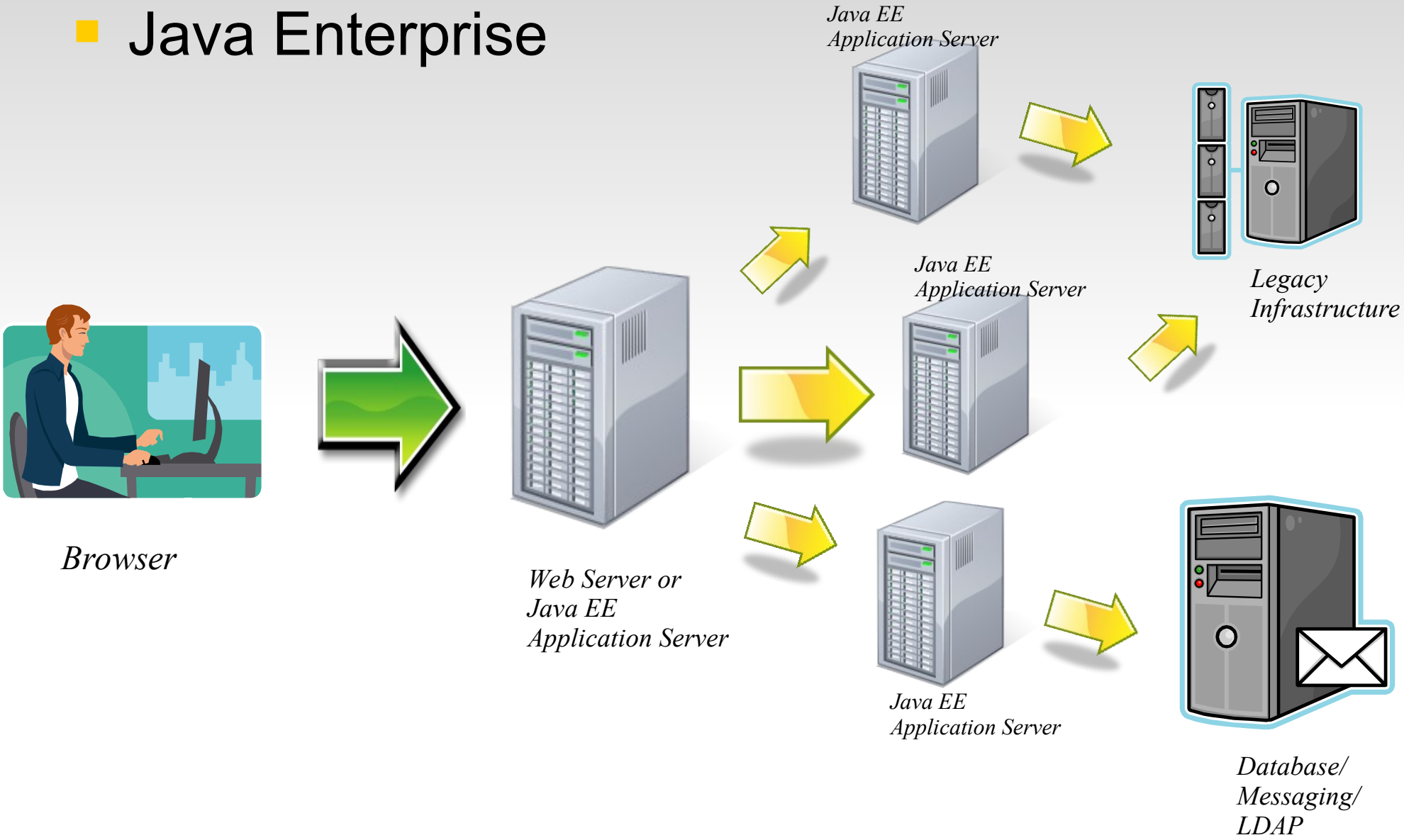
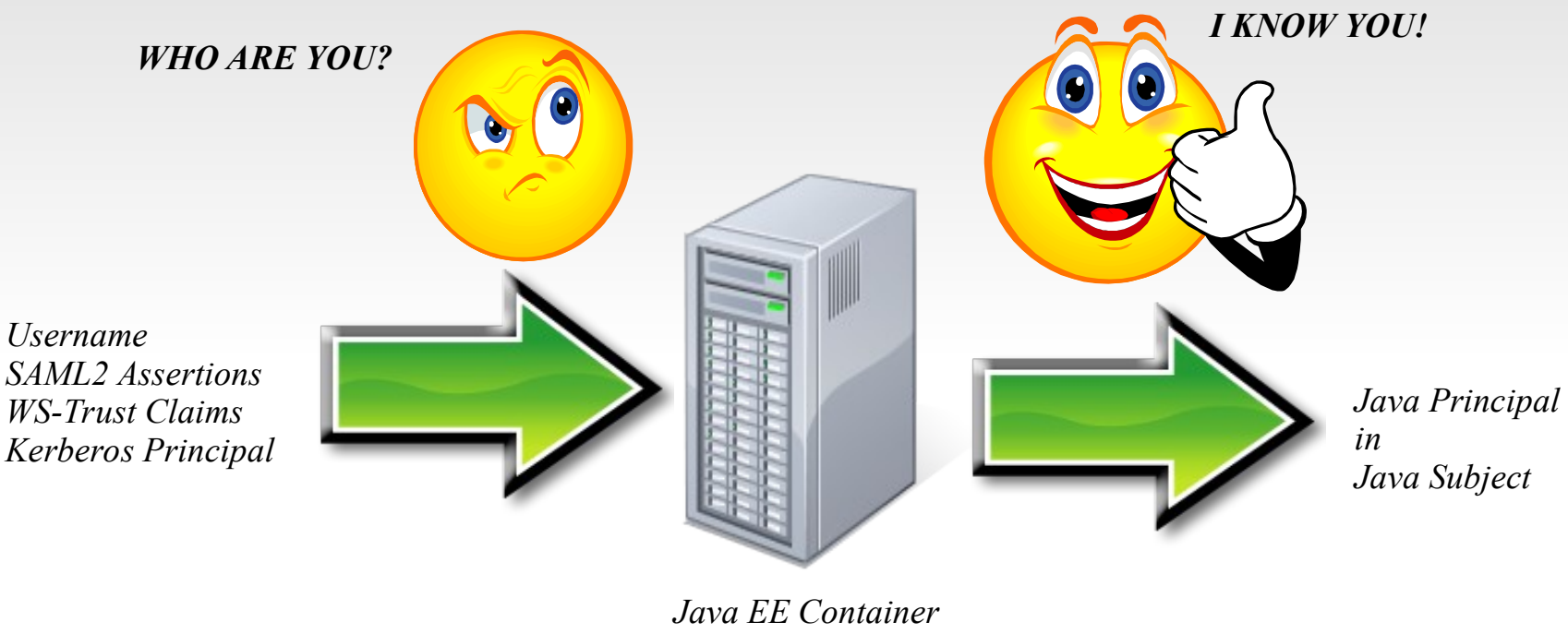
Java EE Security

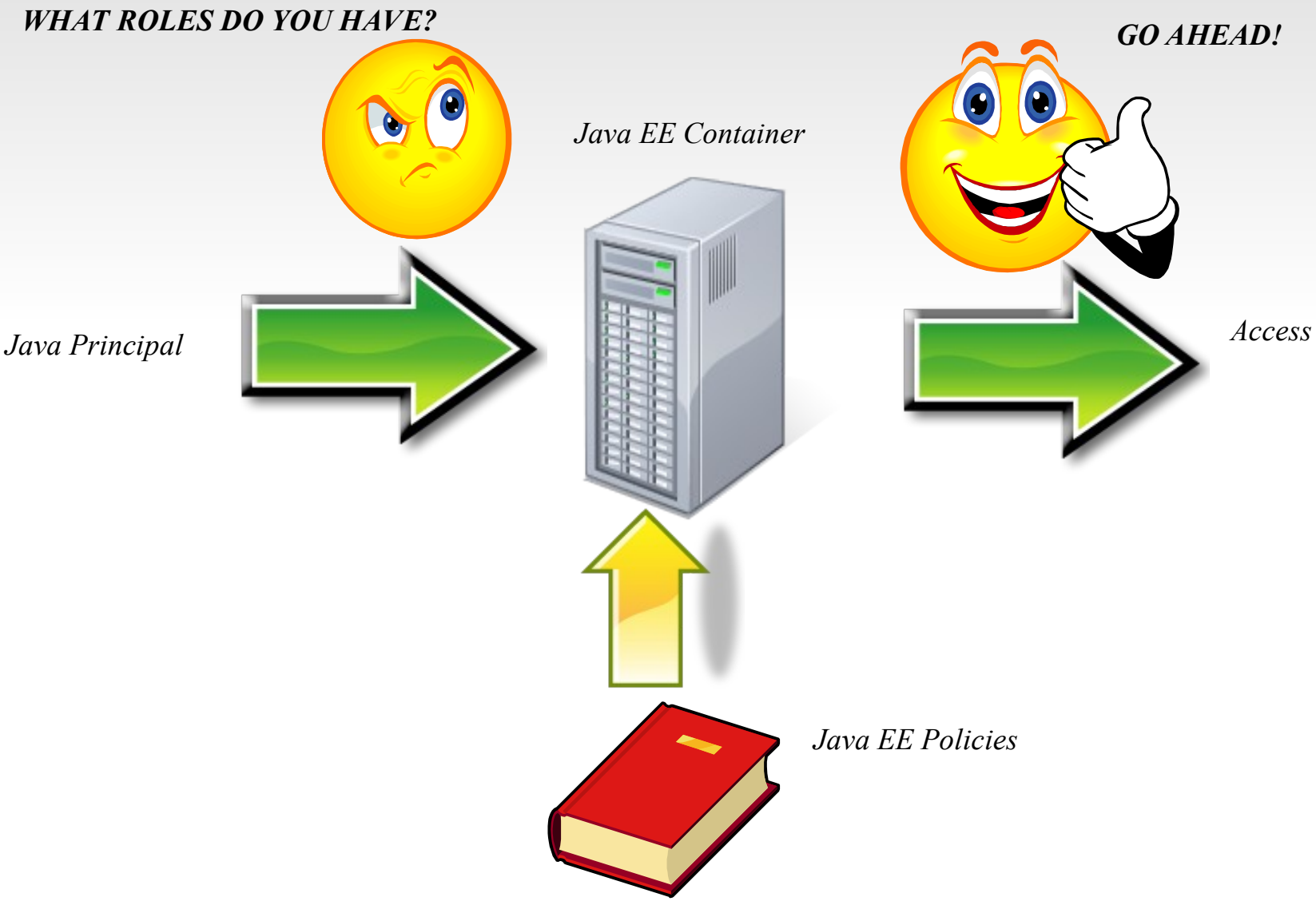
Java Enterprise



Java EE Containers Authentication



Java EE Containers Authorization



AUTHENTICATION TYPES

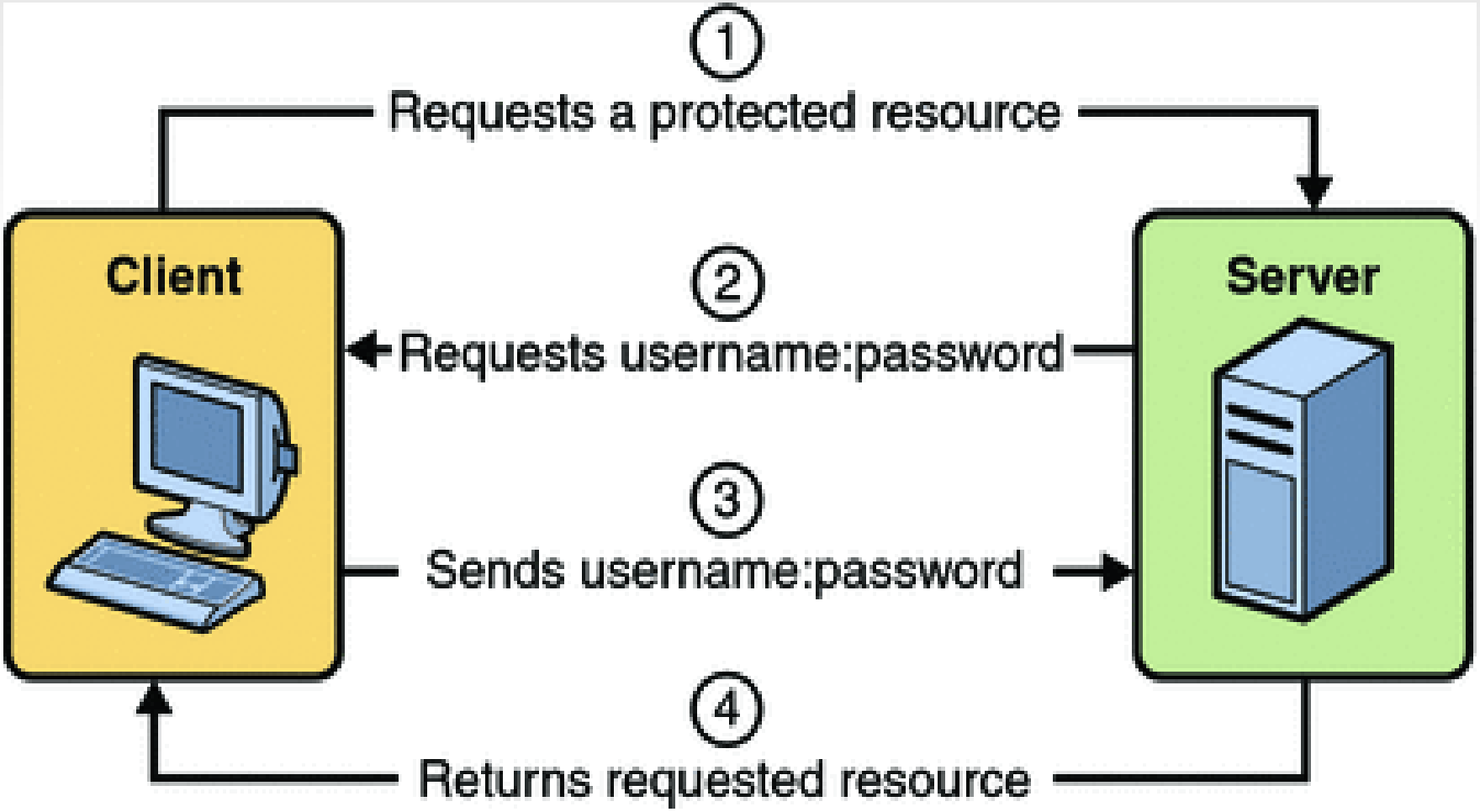
- BASIC AUTHENTICATION - security credential are required to authenticate
- FORM BASED AUTHENTICATION – A html form providing security credential
- SSL AUTHENTICATION authentication by certificates

- Important Terms in Java EE Security
- Realm
- Groups
- Users
- Principal
- Role
- The application Server will provide the facility to create realm, users and groups

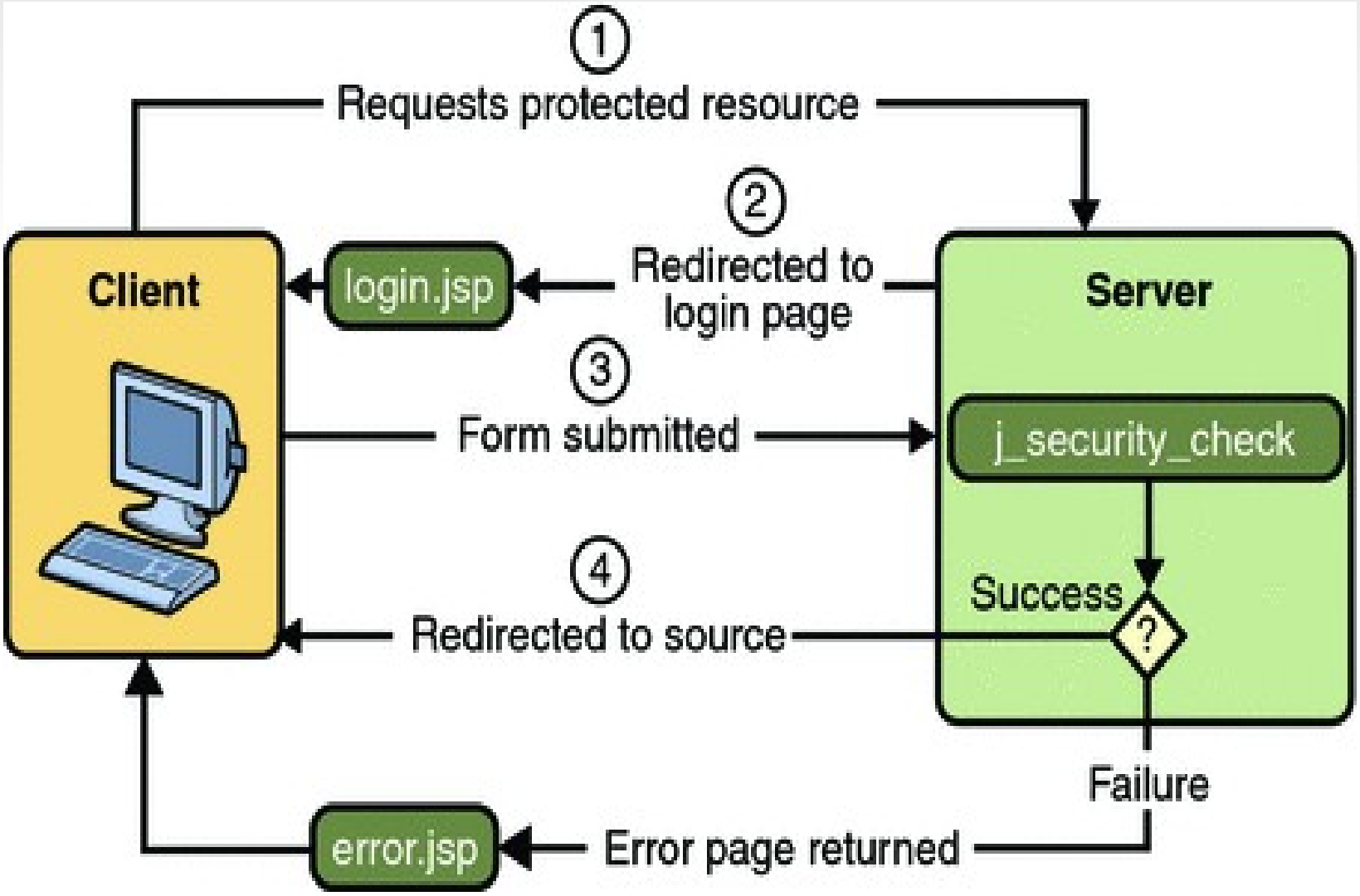
- In BASIC AUTHENTICATION and FORM-BASED AUTHENTICATION

- Roles are created in the application context like sun-web.xml with predefined tags
- The resources required to be protected are listed in web.xml with role and group

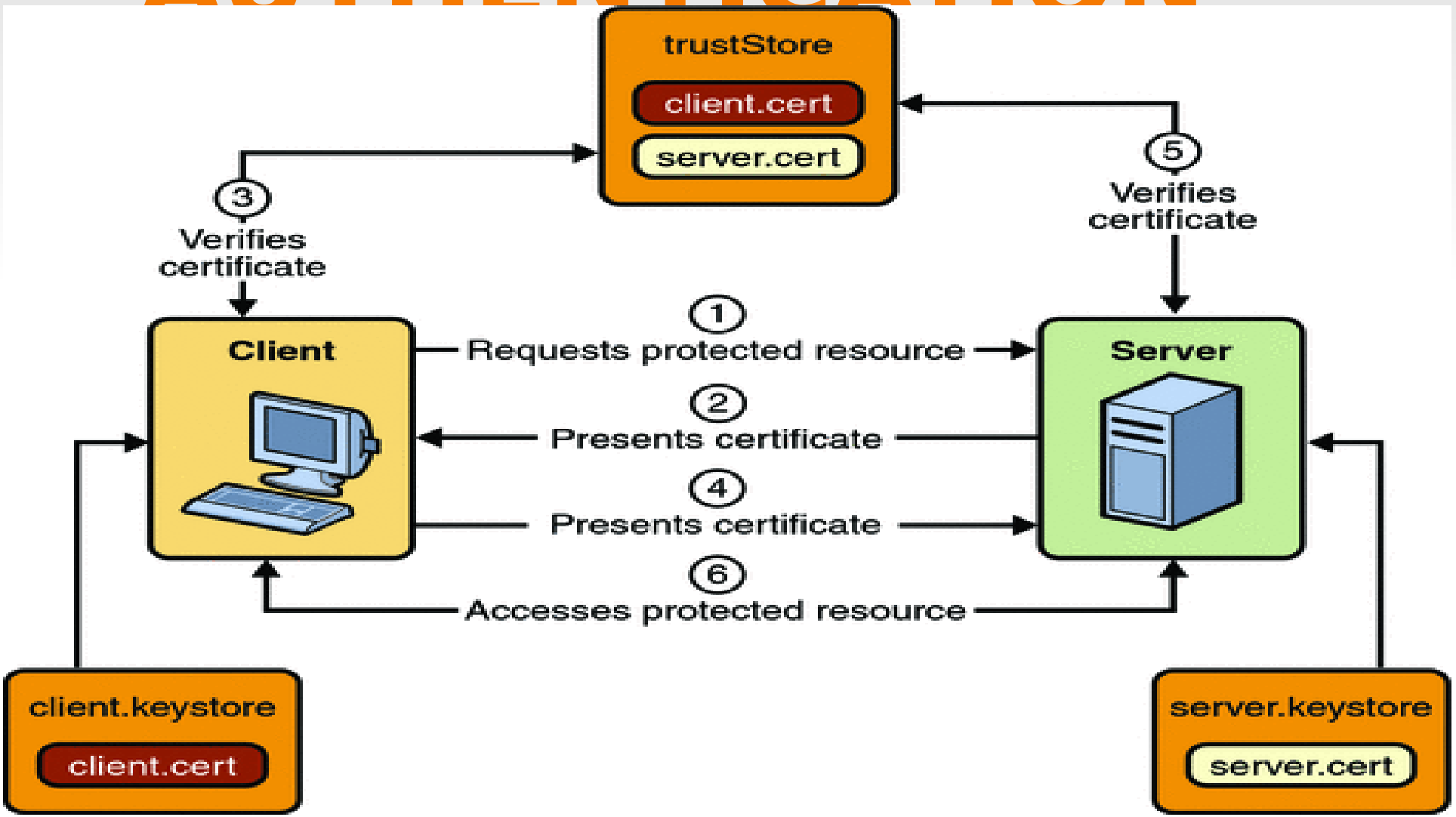
BASIC AUTHENTICATION



FORM BASED AUTHENTICATION



SSL AUTHENTICATION



Why Java EE 8 security

Java EE 8 includes a Security API specification that defines portable, plug-in interfaces for authentication and identity stores, and a new injectable-type SecurityContext interface that provides an access point for programmatic security. You can use the built-in implementations of these APIs, or define custom implementations.

Components of Java EE 8 security

- *Credentials*
- *Identity Stores*
- *Identity Store Handlers*
- *Authentication Mechanism*
- *AuthorizationMechanism*
- *SecurityContext Interface*

Credentials

Credentials are Objects which encapsulate all the information of User Principal

- User Name
- Password
- Roles/Groups
- Tokens

Supported Credentials

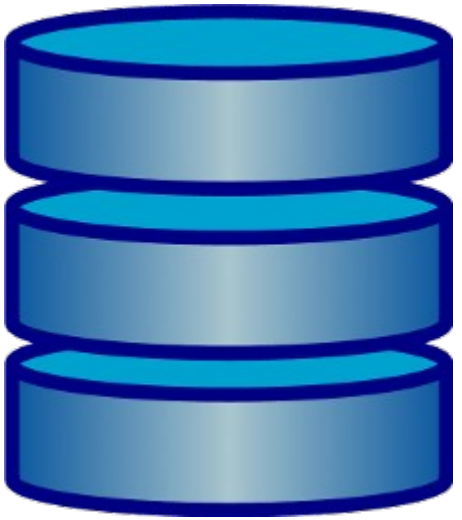
Following Credentials are Supported in Java EE 8

- UserName Password
- JSON Web Tokens (JWT)
- SSL Certificate CN Based
- OAuth Tokens
- *Any Custom Credential*

Identity Stores

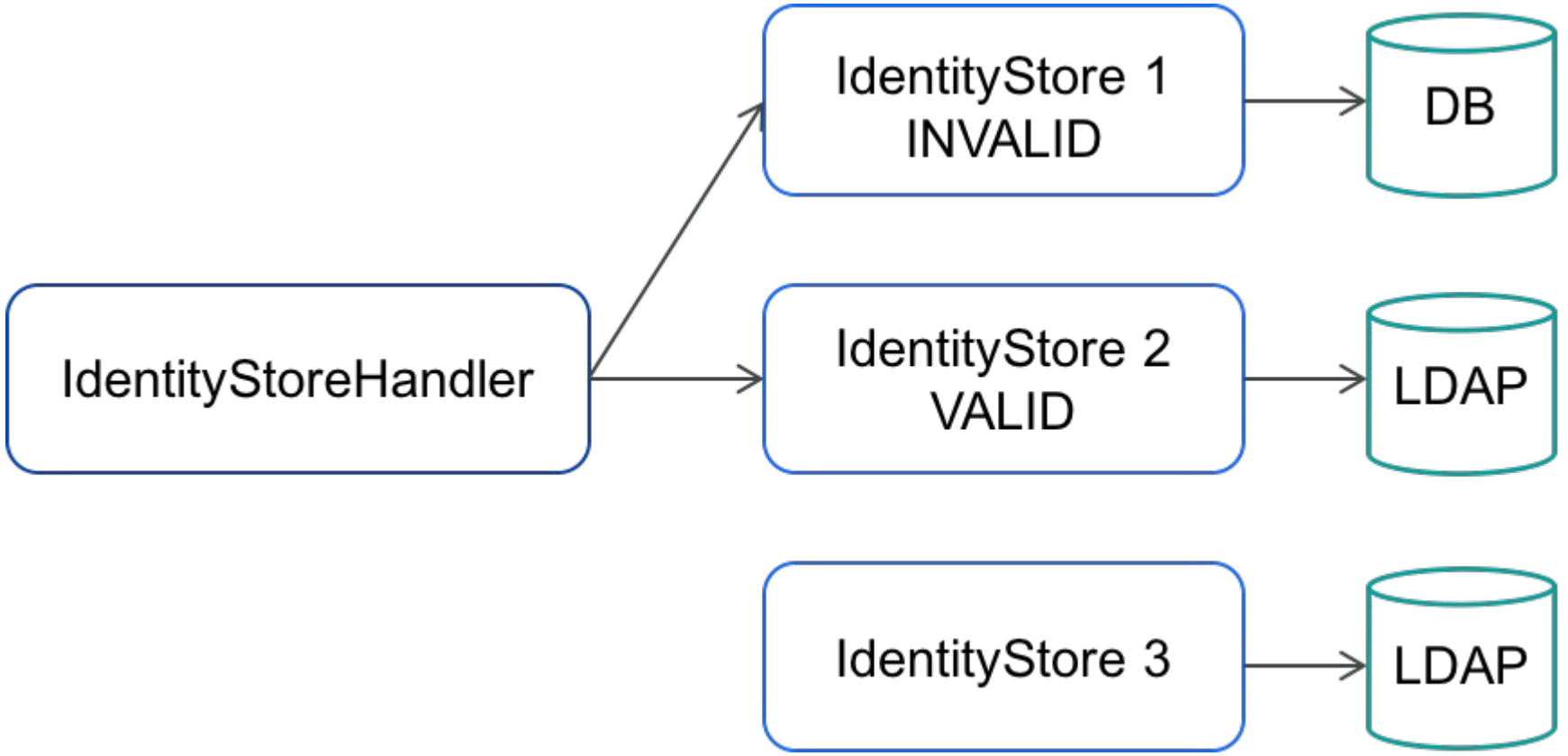
Identity Stores are the Objects encapsulating the repository of registered user credentials. The Identity Stores can represent data in

- Database
- LDAP
- Files
- Remember Me Cookies
- Any Custome Storage



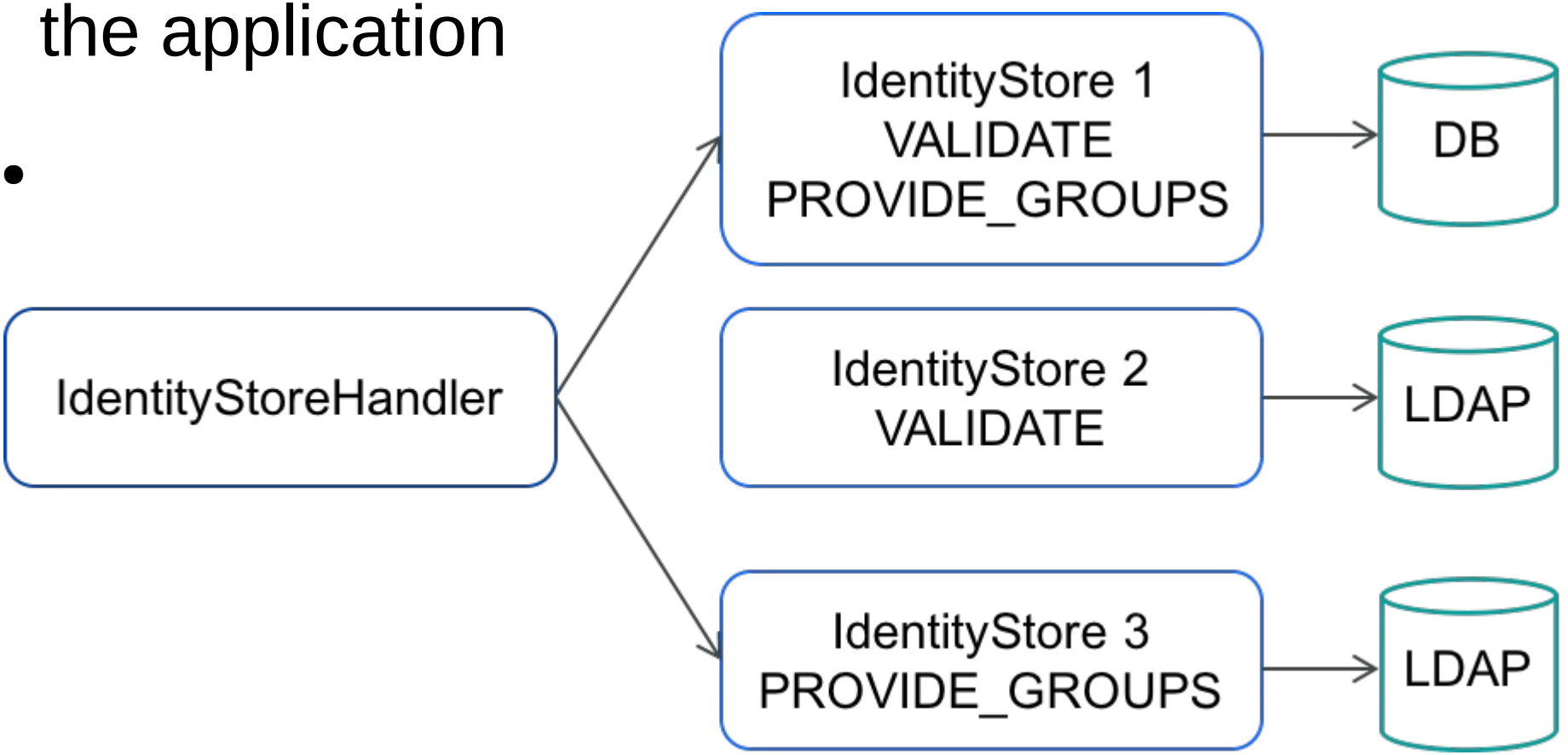
Identity Store Handler

Identity Handler continuously scans all the Identity Stores mentioned in the application.

- 

Identity Store Handler

Identity Handler continuously scans all the Identity Stores for GROUPS/ROLES associated with the credentials mentioned in the application



Authentication Mechanism

It is the mode in which application will like to authenticate. Some inbuilt Authentication Mechanisms are

- BasicAuthenticationMechanism
- FormBasedAuthenticationMechanism
- CustomFormBasedAuthenticationMechanism
- OAuthAuthenticationMechanism
- *Http Authentication Mechanism*

Authorization Mechanism

It is the mode in which application will like to authenticate. Some inbuilt Authorization Mechanisms are

- DB Based Authorization Mechanism
- Basic Authorization Mechanism
- Cookie Based Authorization Mechanism
- JWT/ Auth Based Authorization Mechanism
- *Any Custom Authorization Mechanism*

Security Context

It is an Injectable Object used to call authenticate method and check the logged in Principal and its Role using methods like

- Authenticate (request, response, Credential)
- CallerPrincipal() for name of user
- isCallerInRole(<role name>) returns boolean
- And other useful metadata regarding user and role

Other Features

It is the mode in which application will like to authenticate. Some inbuilt Authentication Mechanisms are

- It is compulsory to use SSL in all the calls
- Supports all RoleBased Annotations in EJB and Rest Objects
- Completely Stateless
- More Customization by Developer
- *CDIs must be used as far as possible instead of EJBs*

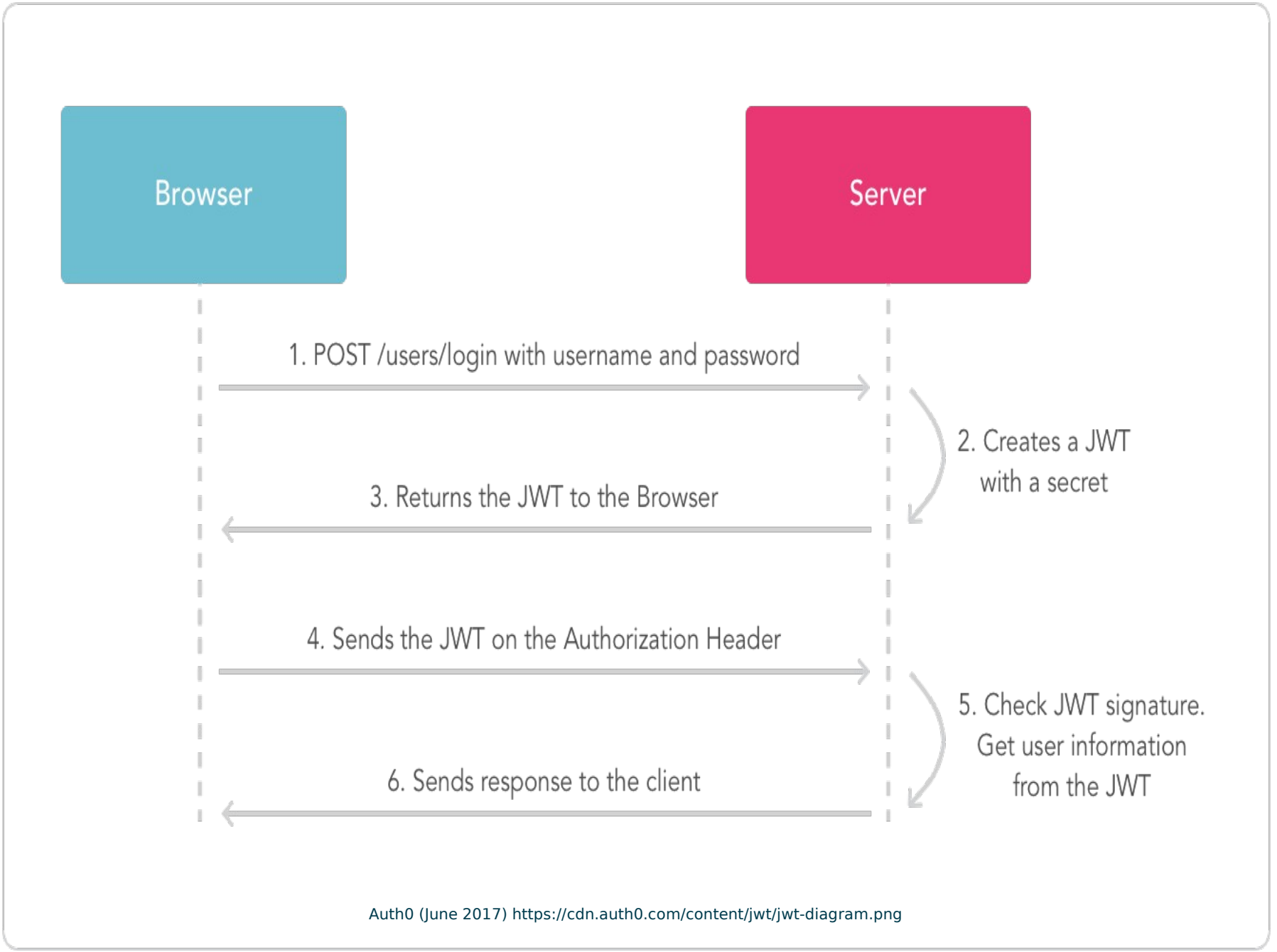
JAVA EE 8 SECURITY and JSON WEB TOKENS

Popularly called jots

What is JWT ?

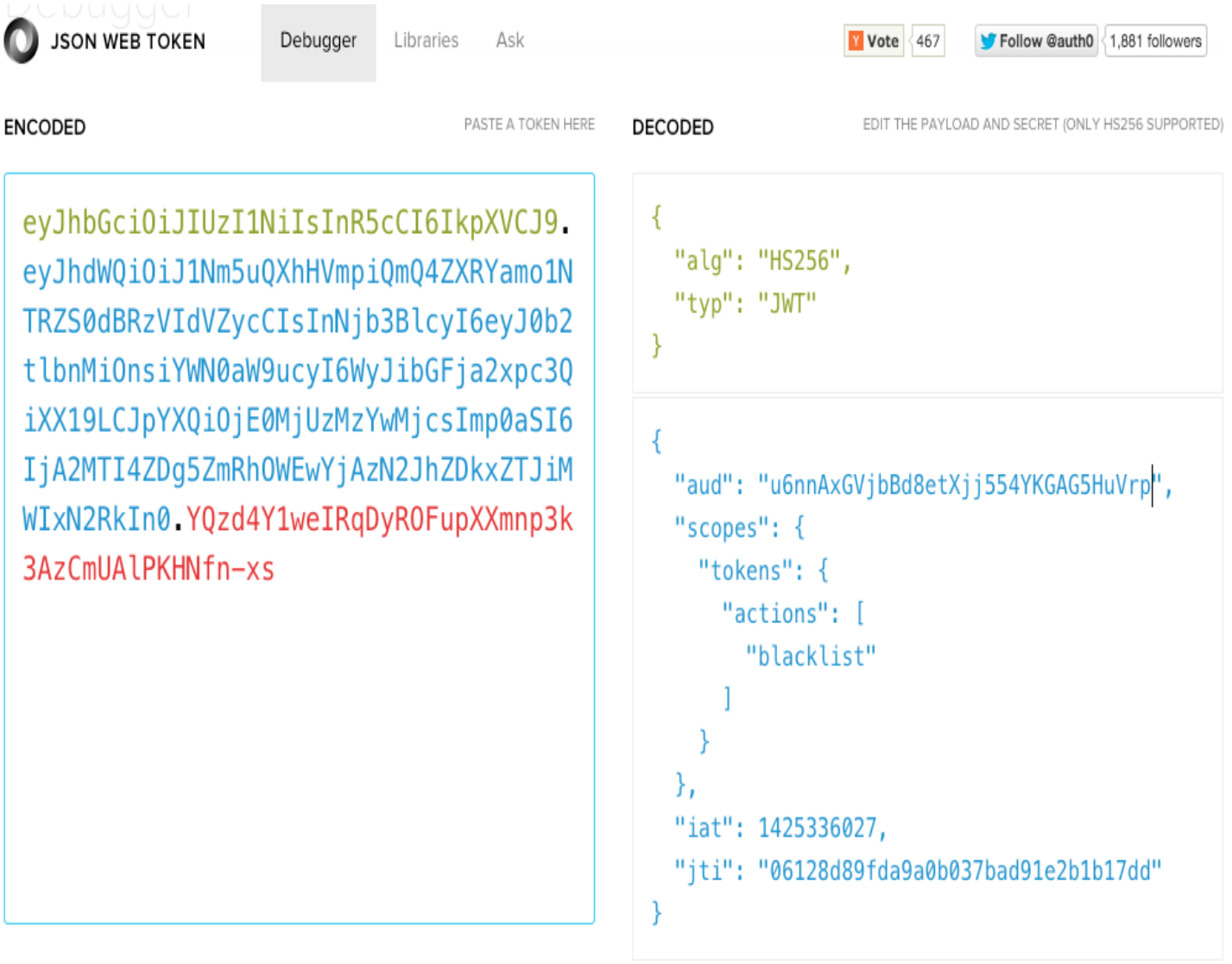
Is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

WORKING OF JWT (jots)

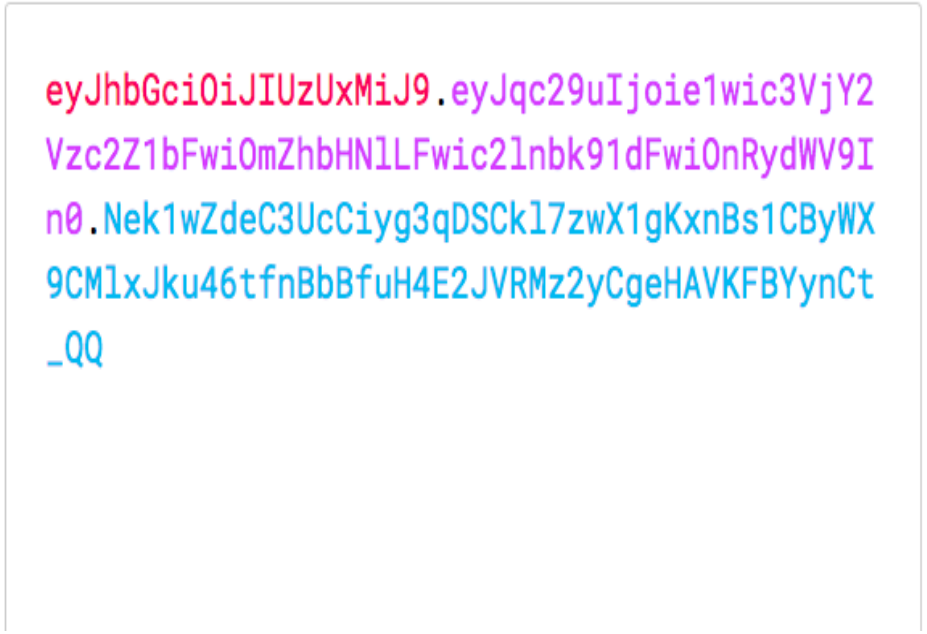


Auth0 (June 2017) https://cdn.auth0.com/content/jwt/jwt-diagram.png

JWT Structure

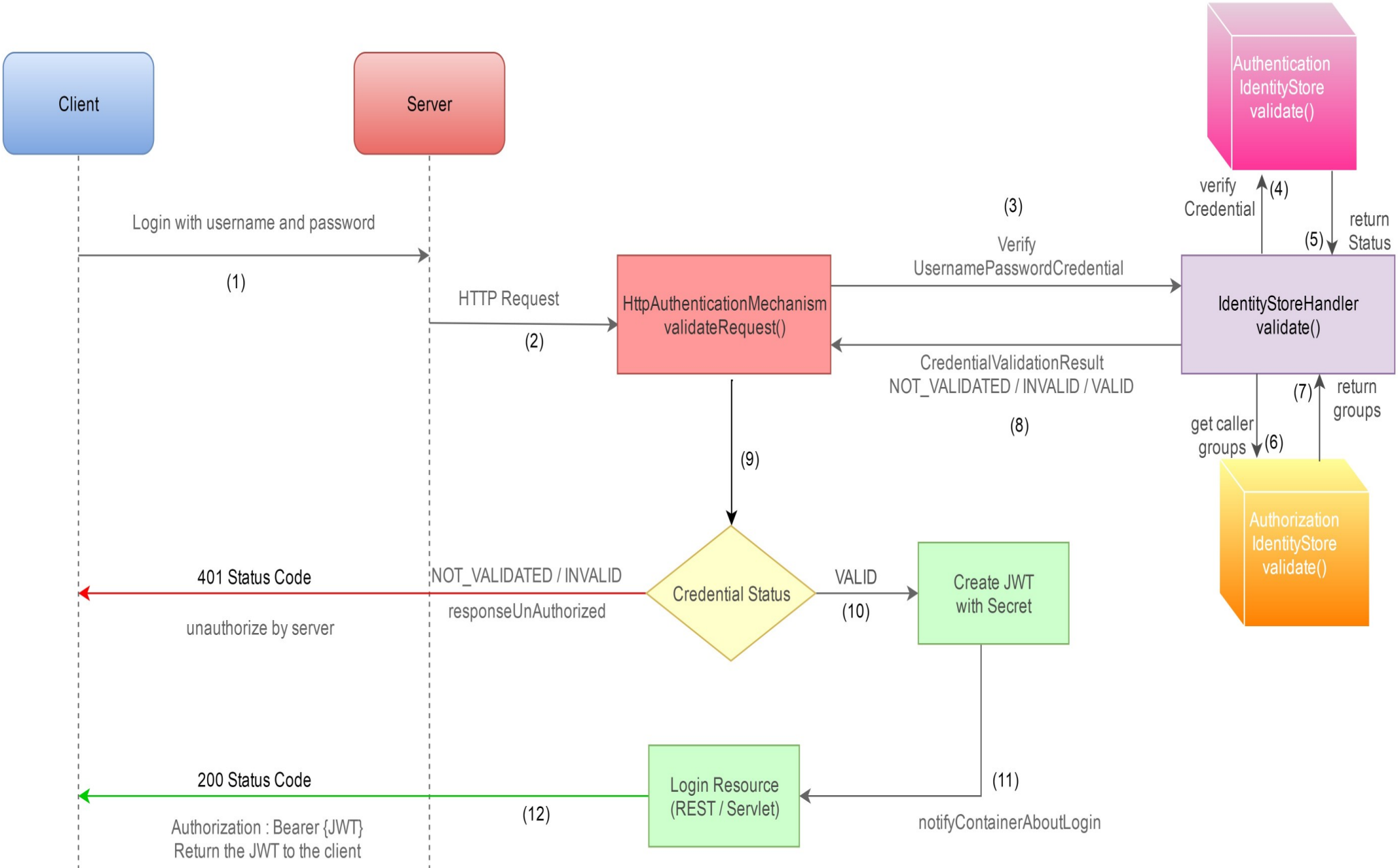


JWT Structure

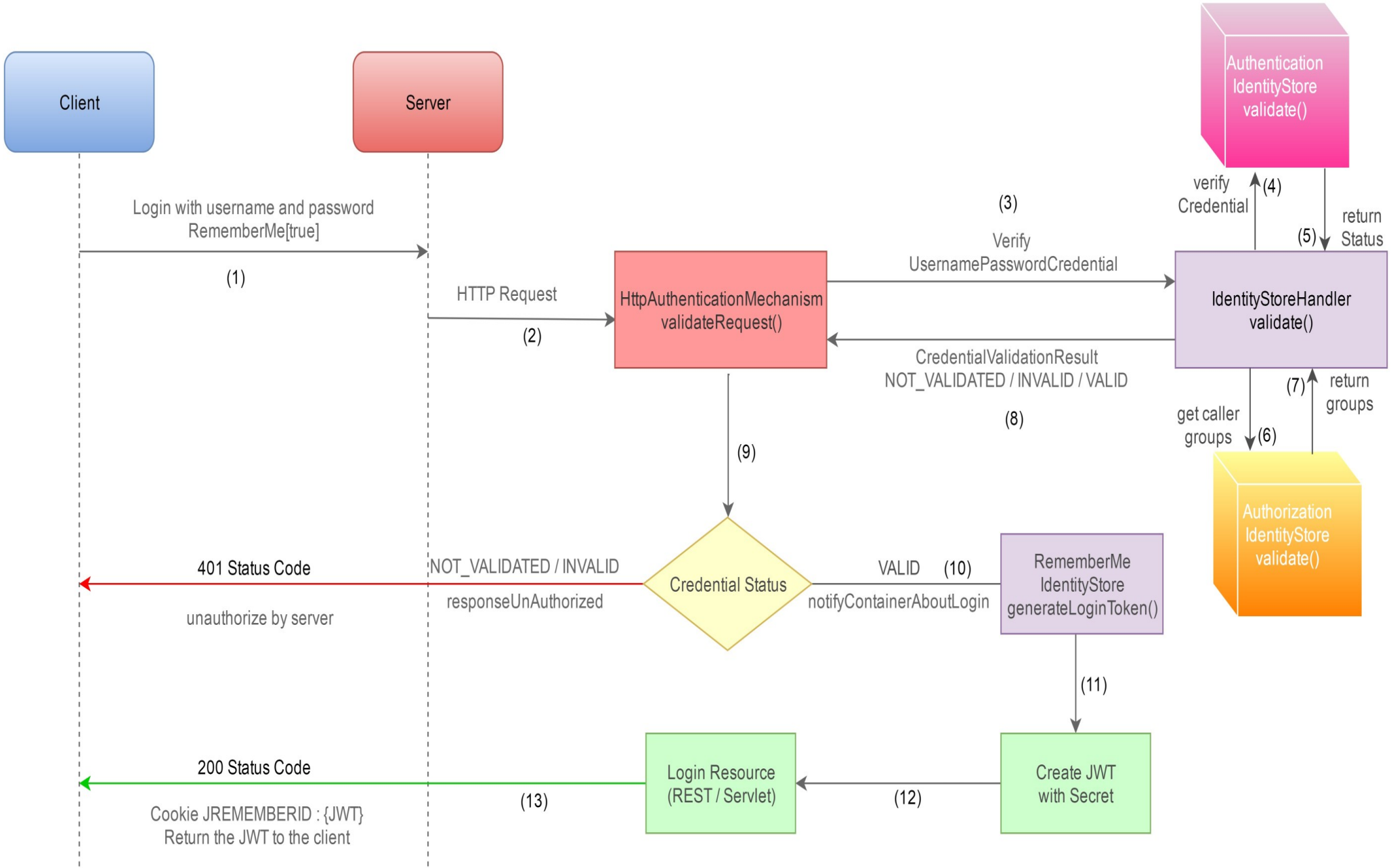


JWT Structure

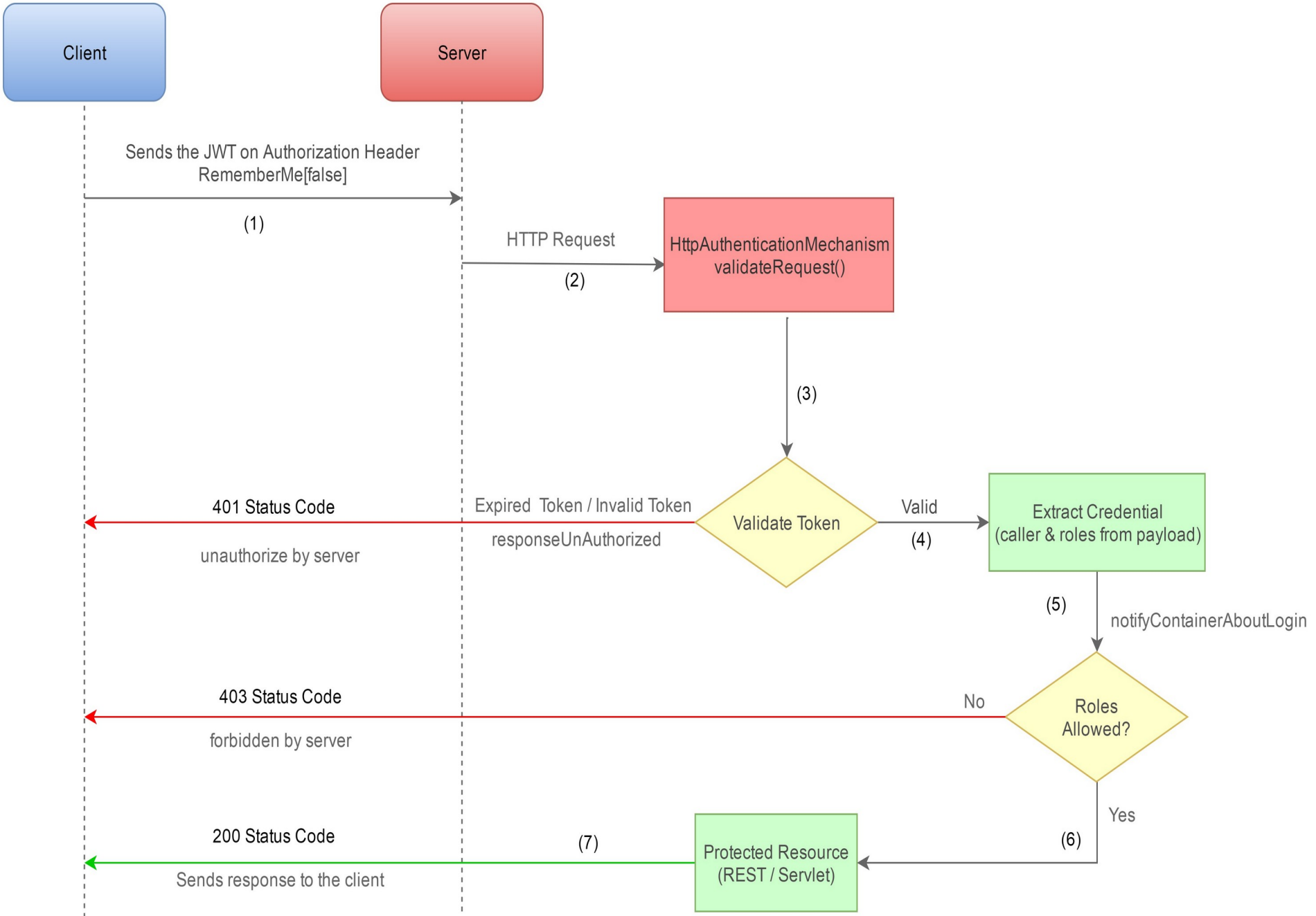
PAYARA – Authentication with JWT



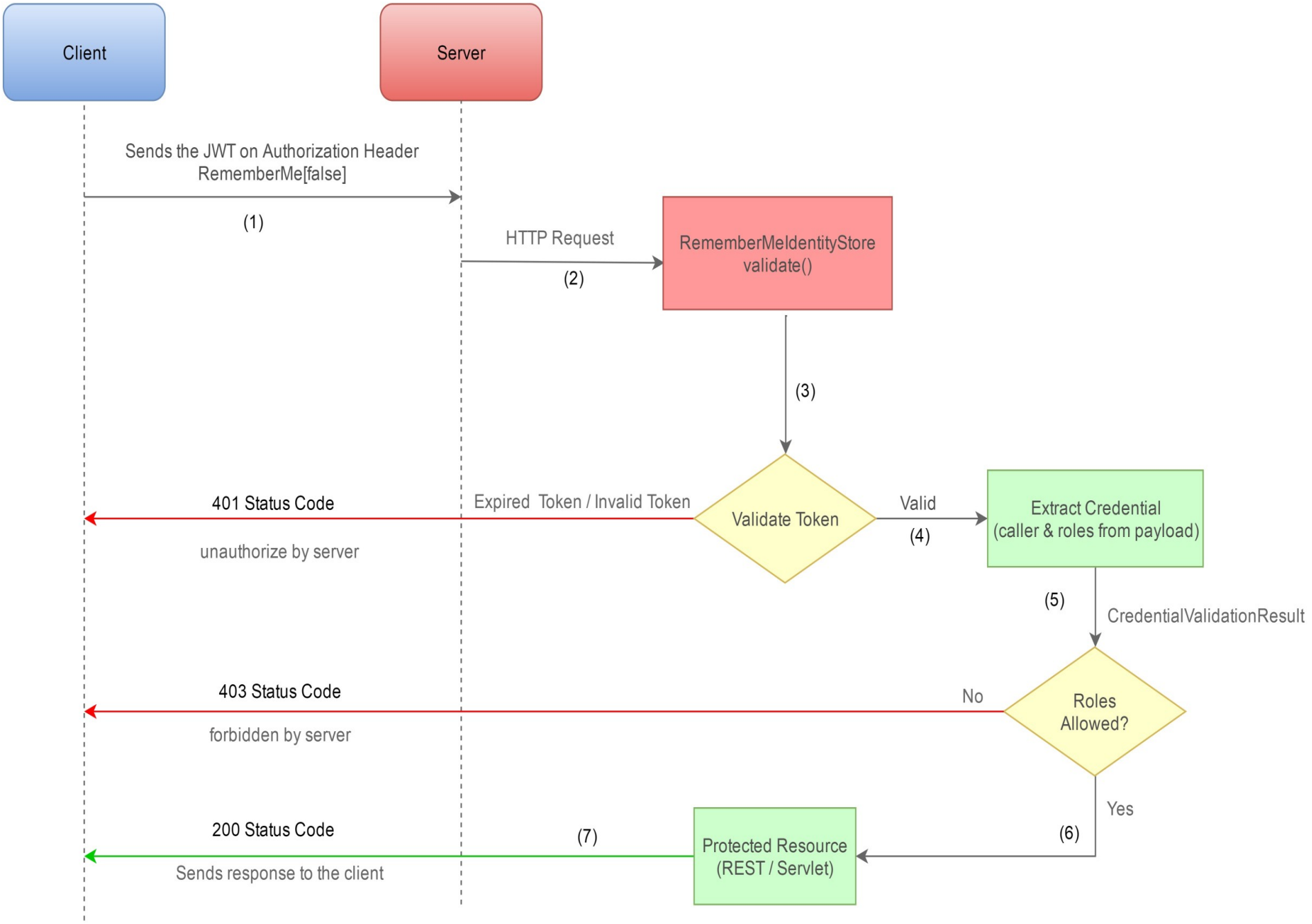
PAYARA – Remember Me Authentication with JWT



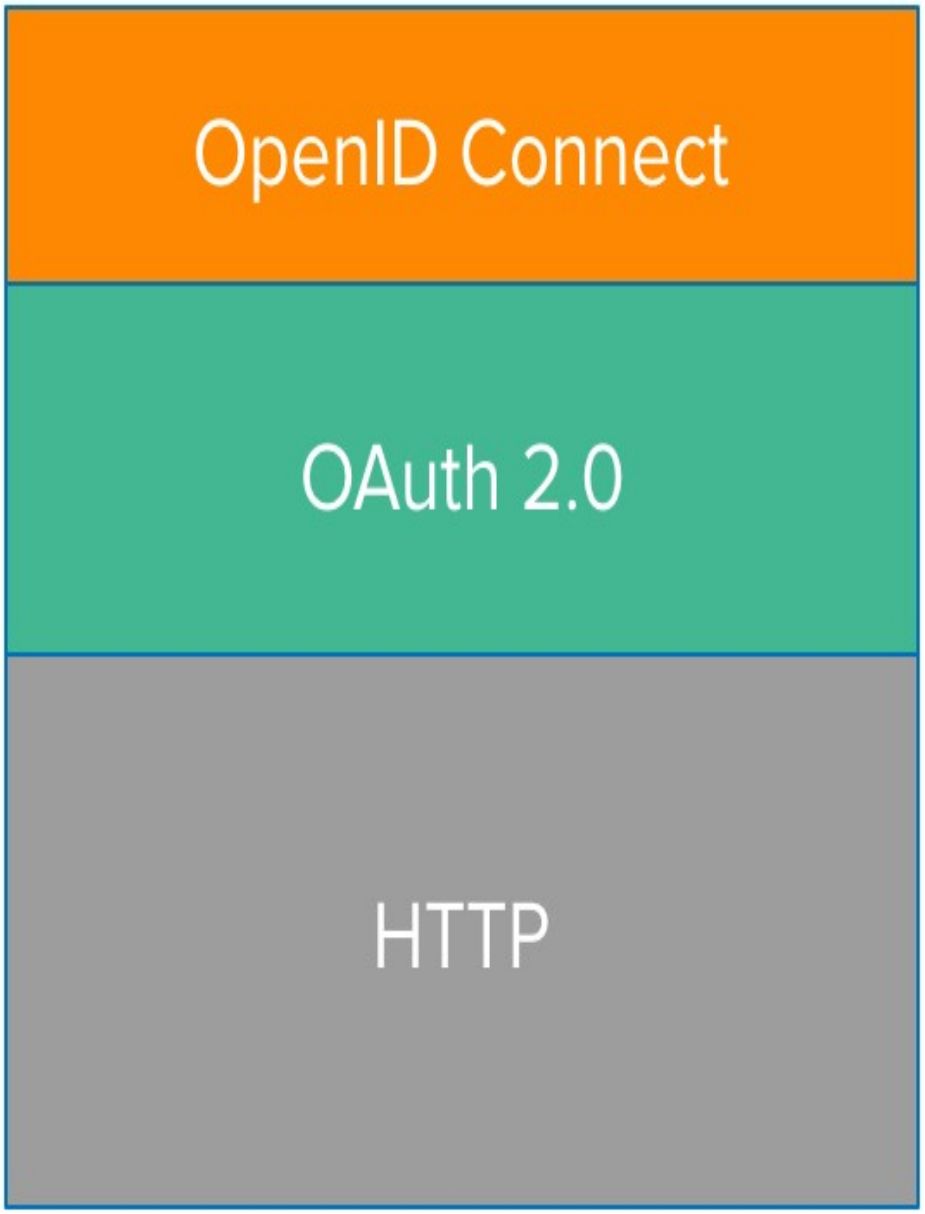
PAYARA – Authorization with JWT



PAYARA – Authorization with JWT (JREMEMBERID Cookie)



JAVA EE 8 SECURITY



OpenID Connect is for **authentication**

OAuth 2.0 is for **authorization**

Lets Have some workable examples