

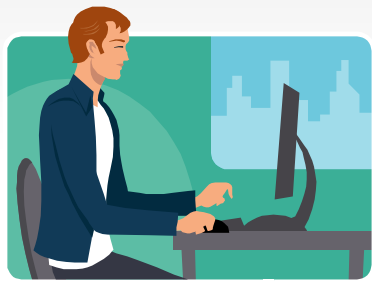
Java EE Security

Java EE 8

The New Security API

Overview Series

■ Java Enterprise



Browser



*Web Server or
Java EE
Application Server*



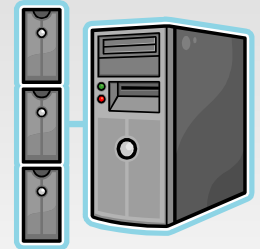
*Java EE
Application Server*



*Java EE
Application Server*



*Java EE
Application Server*

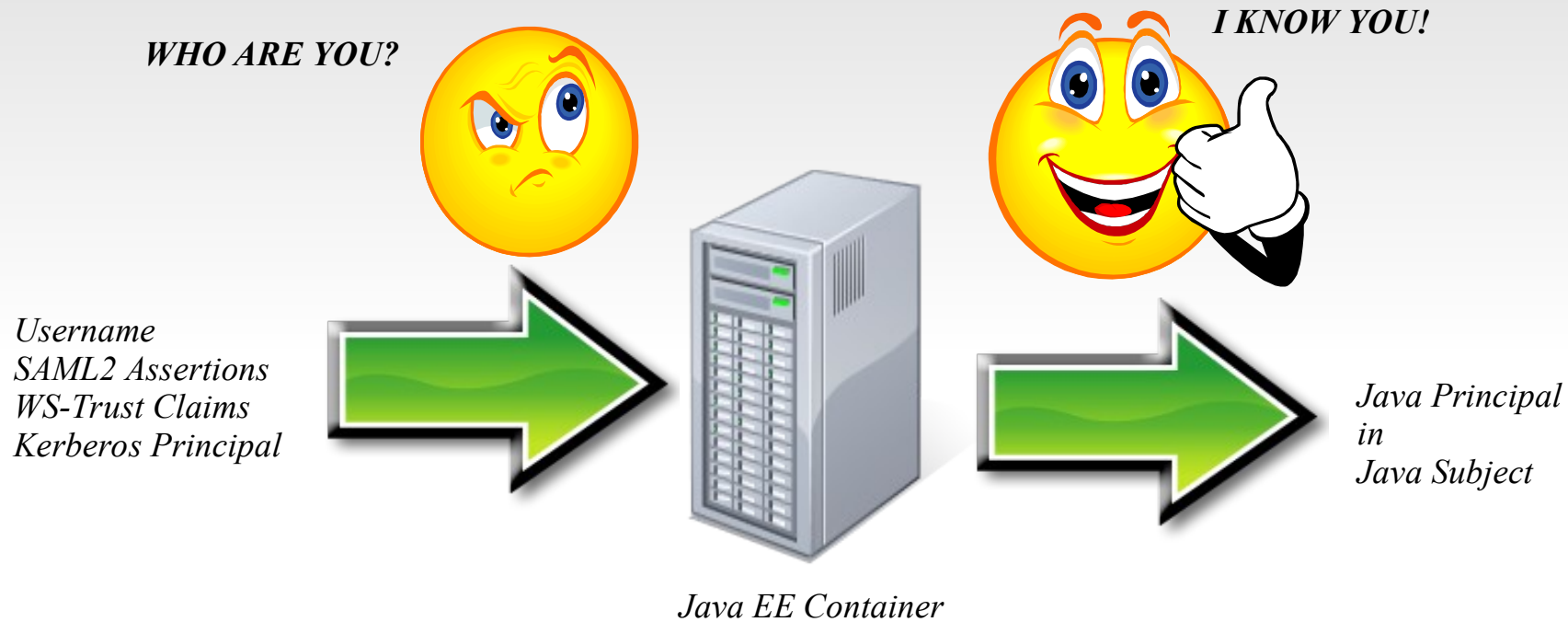


*Legacy
Infrastructure*



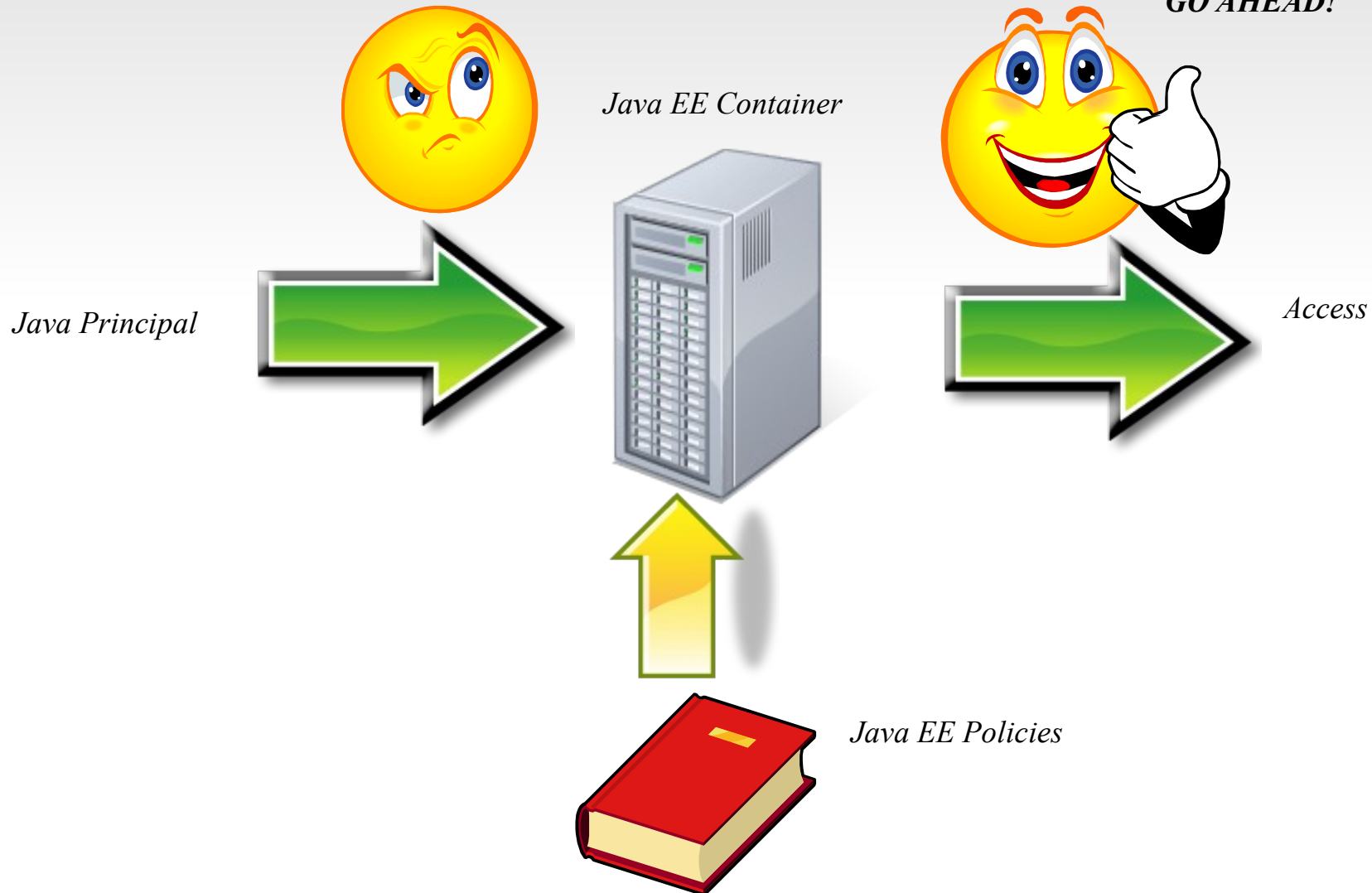
*Database/
Messaging/
LDAP*

■ Java EE Containers Authentication



■ Java EE Containers Authorization

WHAT ROLES DO YOU HAVE?



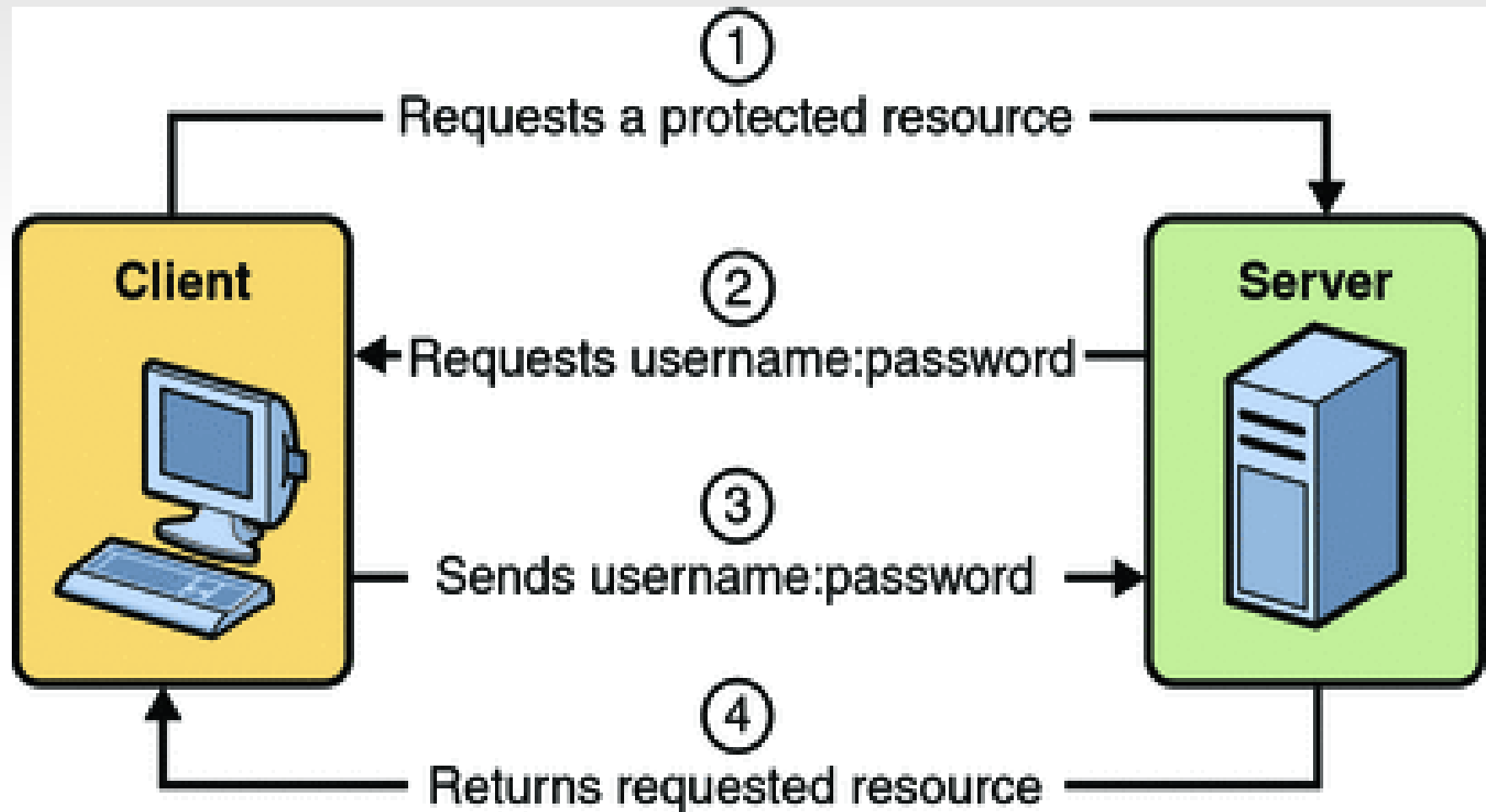
AUTHENTICATION TYPES

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

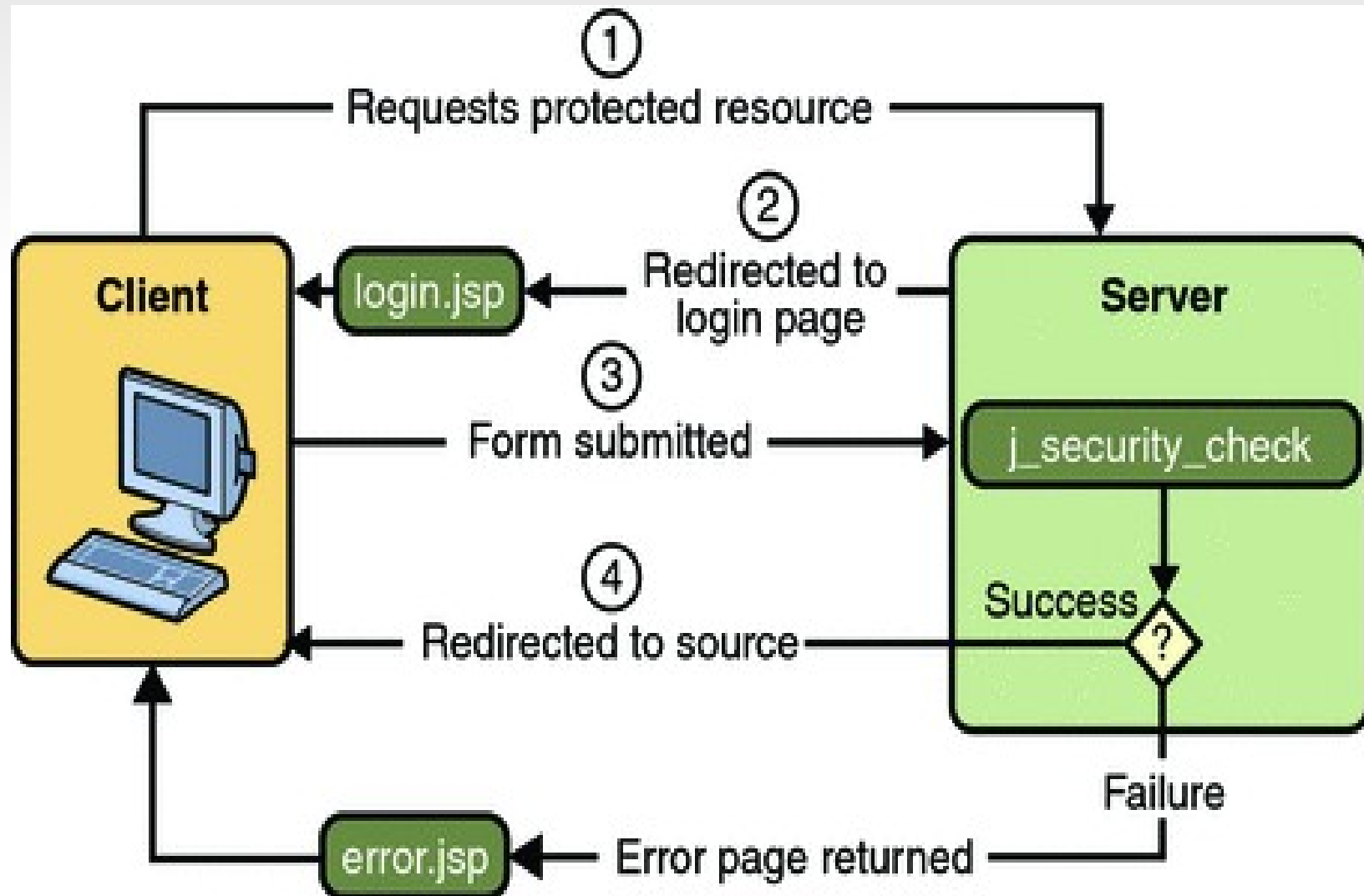
- Important Terms in Java EE Security
- Realm
- Groups
- Users
- Principal
- Role
- The application Server will provide the facility to create realm, users and groups

- 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

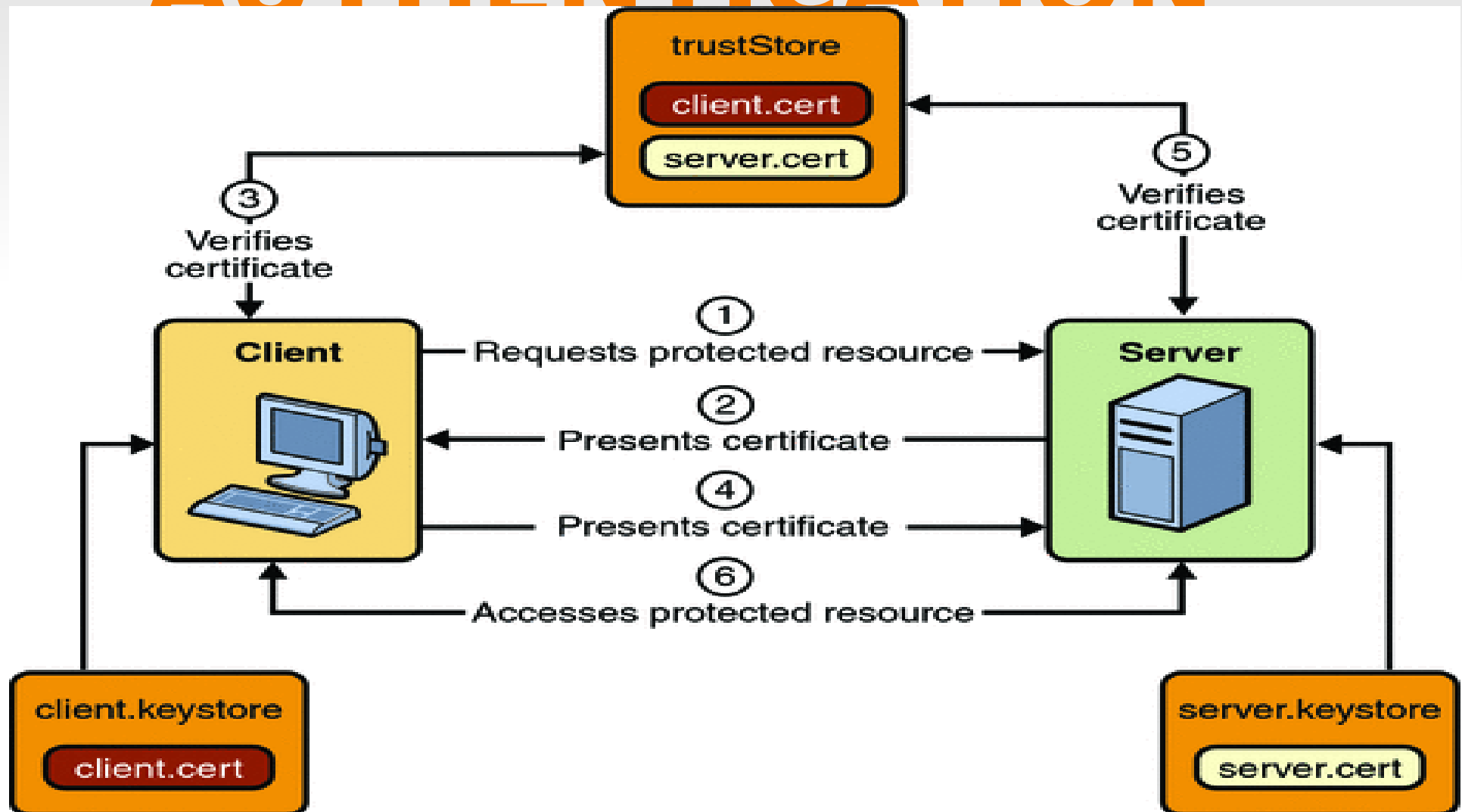
BASIC AUTHENTICATION



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.

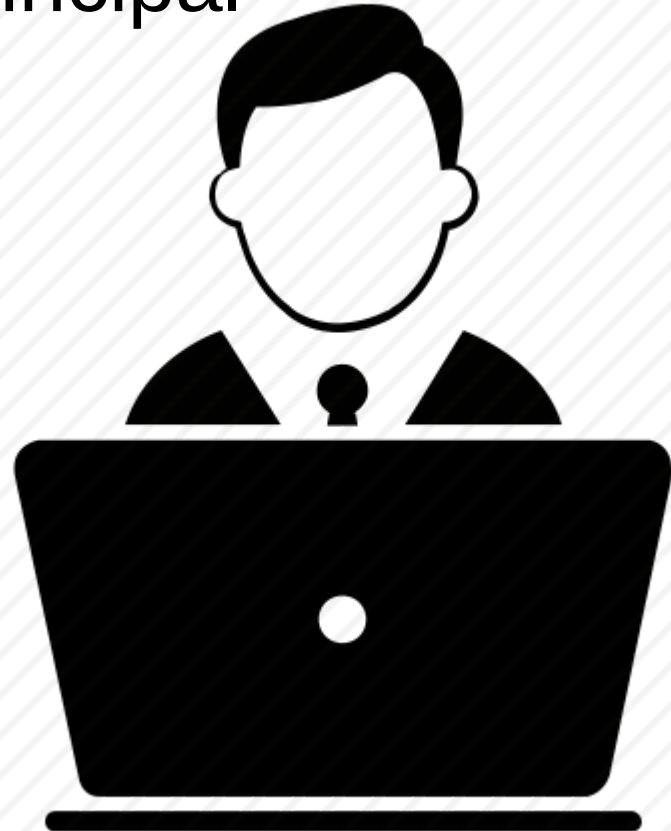
Components of Java EE 8 security

- *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

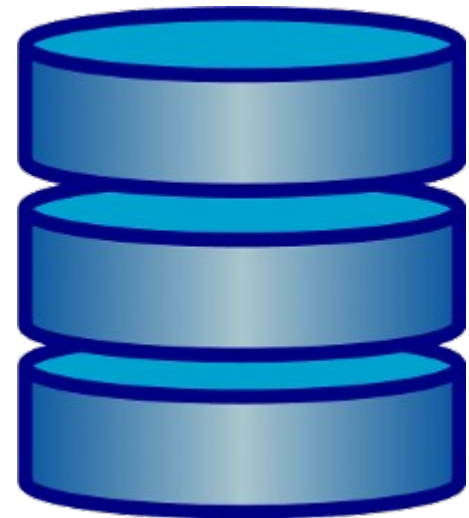
Following Credentials are Supported in Java EE 8

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

Identity Stores

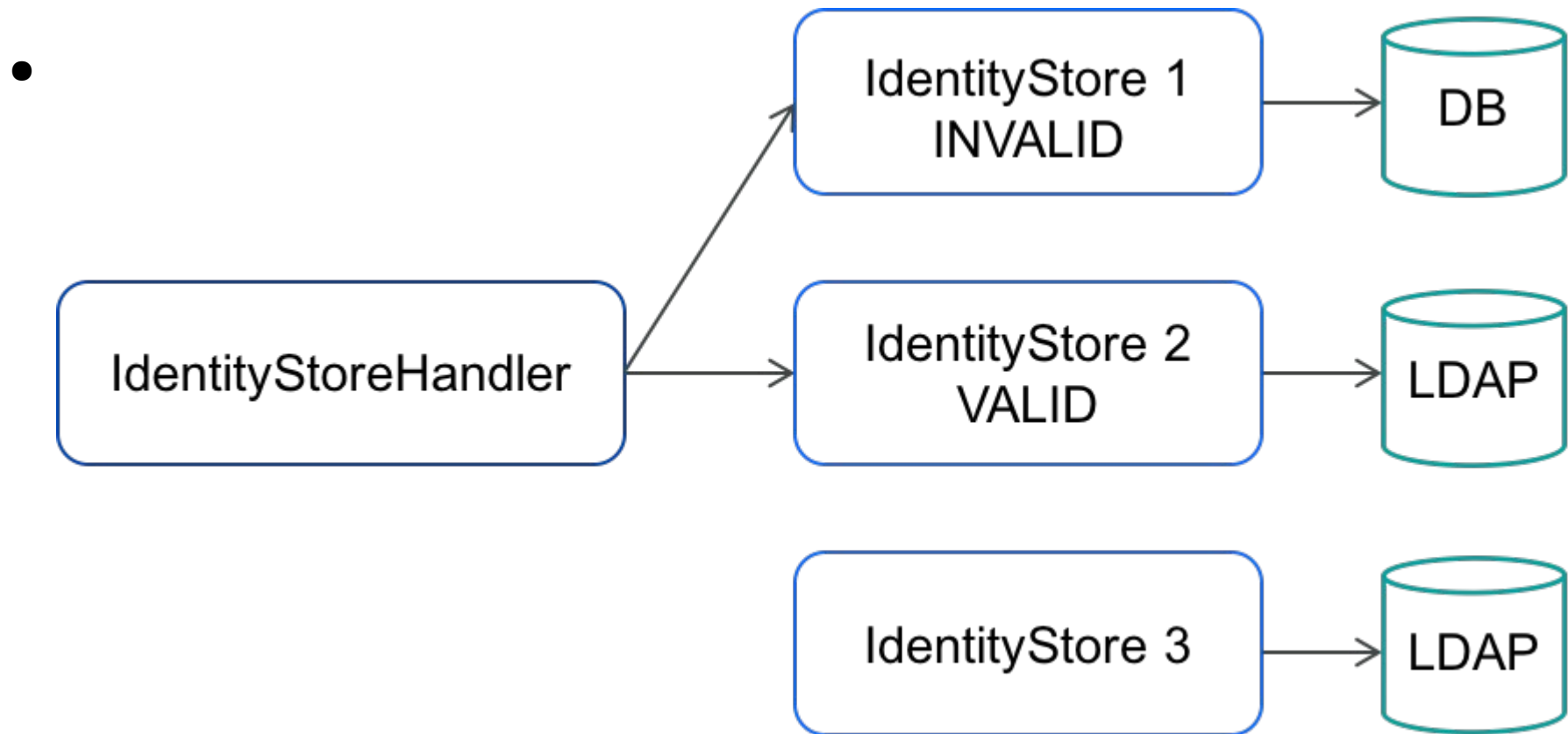
Identity Stores are the Objects encapsulating the repository of registered user credentials. The Identity Stores can represent data in

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



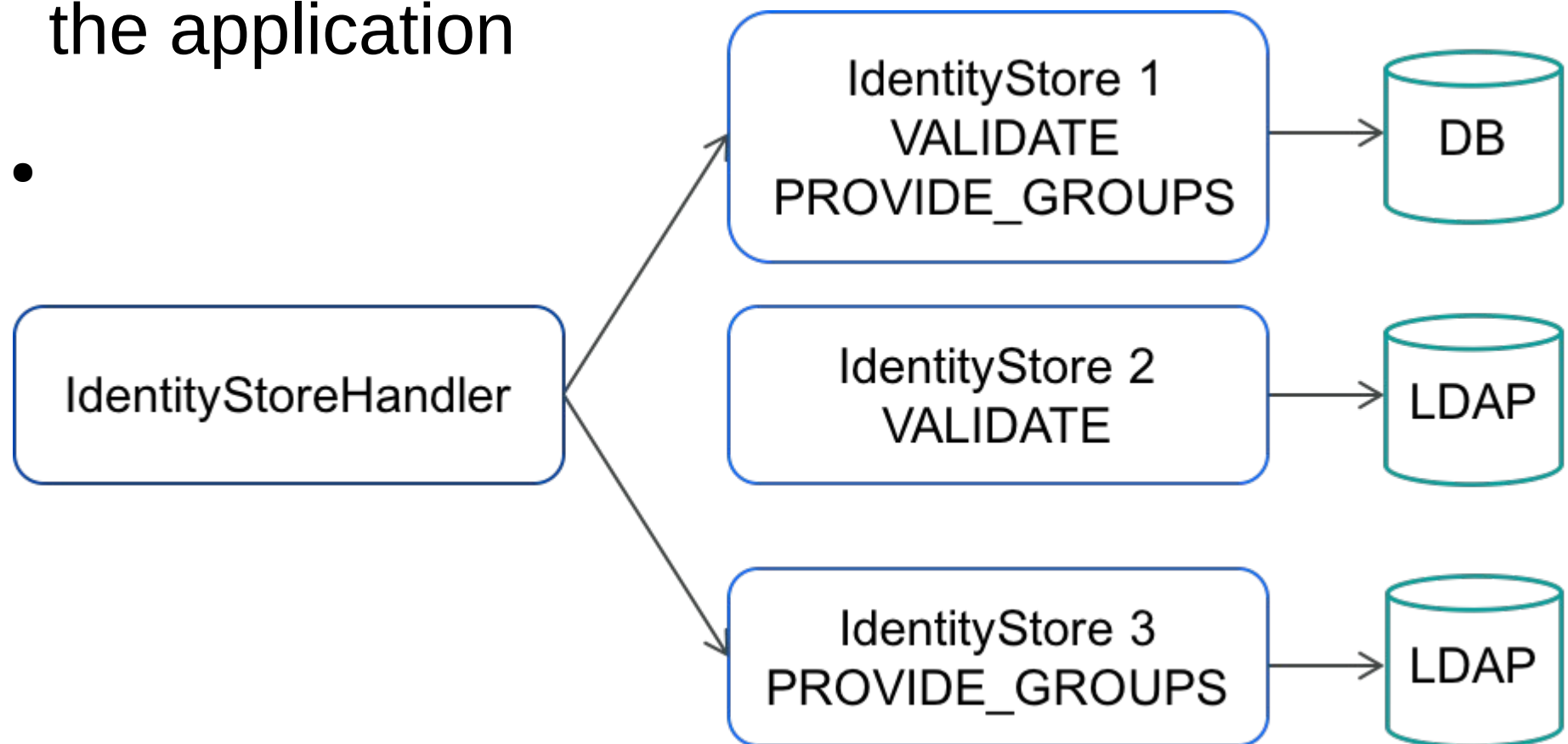
Identity Store Handler

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

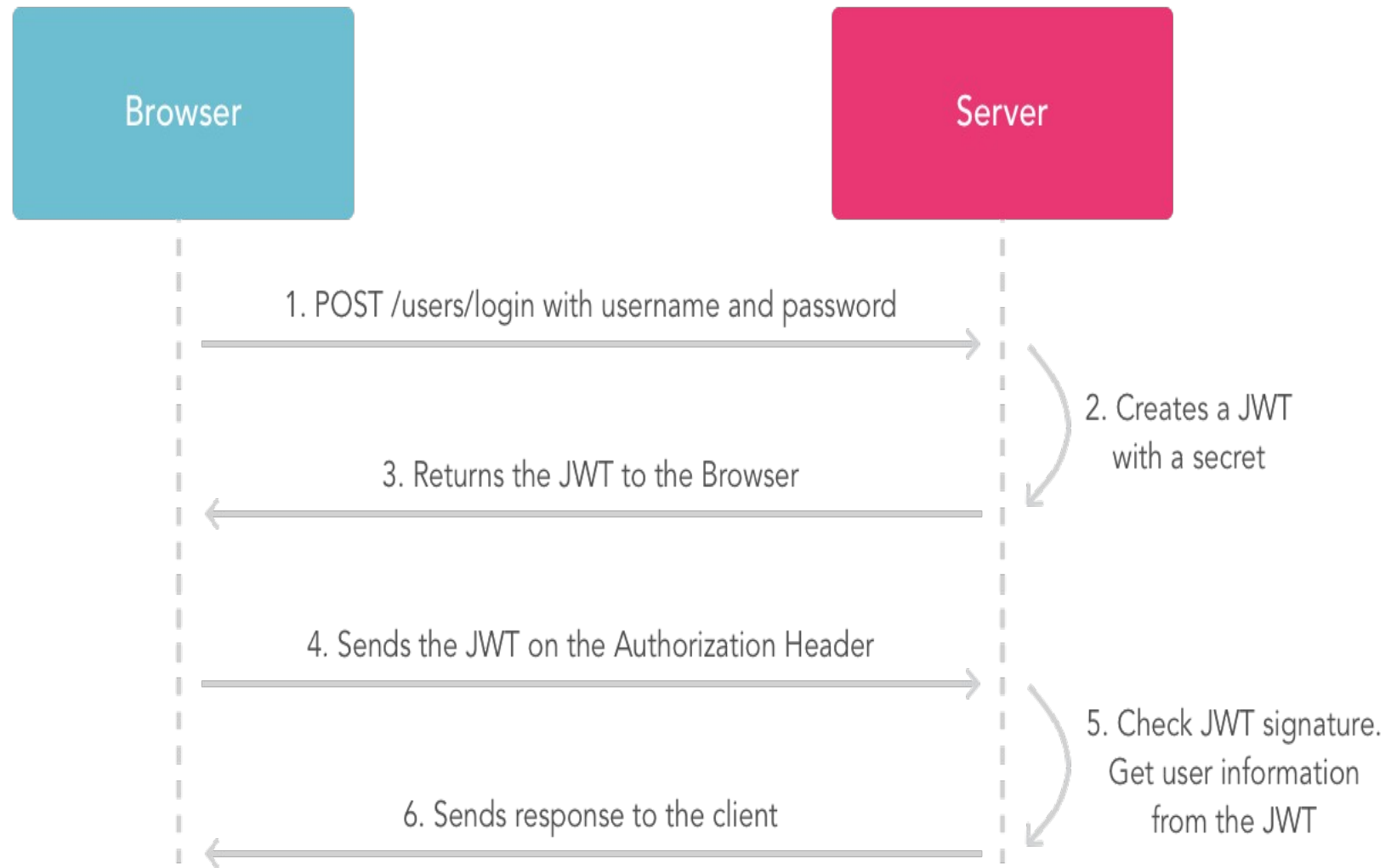


*Popularly called **jots***

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)



JWT Structure



JSON WEB TOKEN

Debugger

Libraries

Ask

Vote

467

Follow @auth0

1,881 followers

ENCODED

PASTE A TOKEN HERE

DECODED

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJhdWQiOiJ1Nm5uQXhHVmпиQmQ4ZXRYamo1N  
TRZS0dBRzVIdVZycCI6InNjb3BlcyI6eyJ0b2  
tlnMi0nsiYWN0aW9ucyI6WyJibGFja2xpc3Q  
iXX19LCJpYXQiOiJlMjUzZDg5ZmRhOWEwYjAzN2JhZDkxZTJiM  
WlxN2RkIn0.YQzd4Y1weIRqDyR0FupXXmnp3k  
3AzCmUALPKHNfn-xs
```

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

```
{  
  "aud": "u6nnAxGVjbBd8etXjj554YKGAG5HuVrp",  
  "scopes": {  
    "tokens": {  
      "actions": [  
        "blacklist"  
      ]  
    }  
  },  
  "iat": 1425336027,  
  "jti": "06128d89fda9a0b037bad91e2b1b17dd"  
}
```

JWT Structure

```
eyJhbGciOiJIUzUxMiJ9.eyJqc29uIjoie1wic3VjY2Vzc2Z1bFwiOmZhbHN1LFwic2lnbk91dFwiOnRydWV9In0.Nek1wZdeC3UcCiyg3qDSCk17zwX1gKxnBs1CByWX9CM1xJku46tfnBbBfuH4E2JVRMz2yCgeHAVKFBYynCt_QQ
```

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS512"  
}
```

PAYLOAD: DATA

```
{  
  "json": "{\\"successful\\":false,\\"signOut\\":true}"  
}
```

JWT Structure

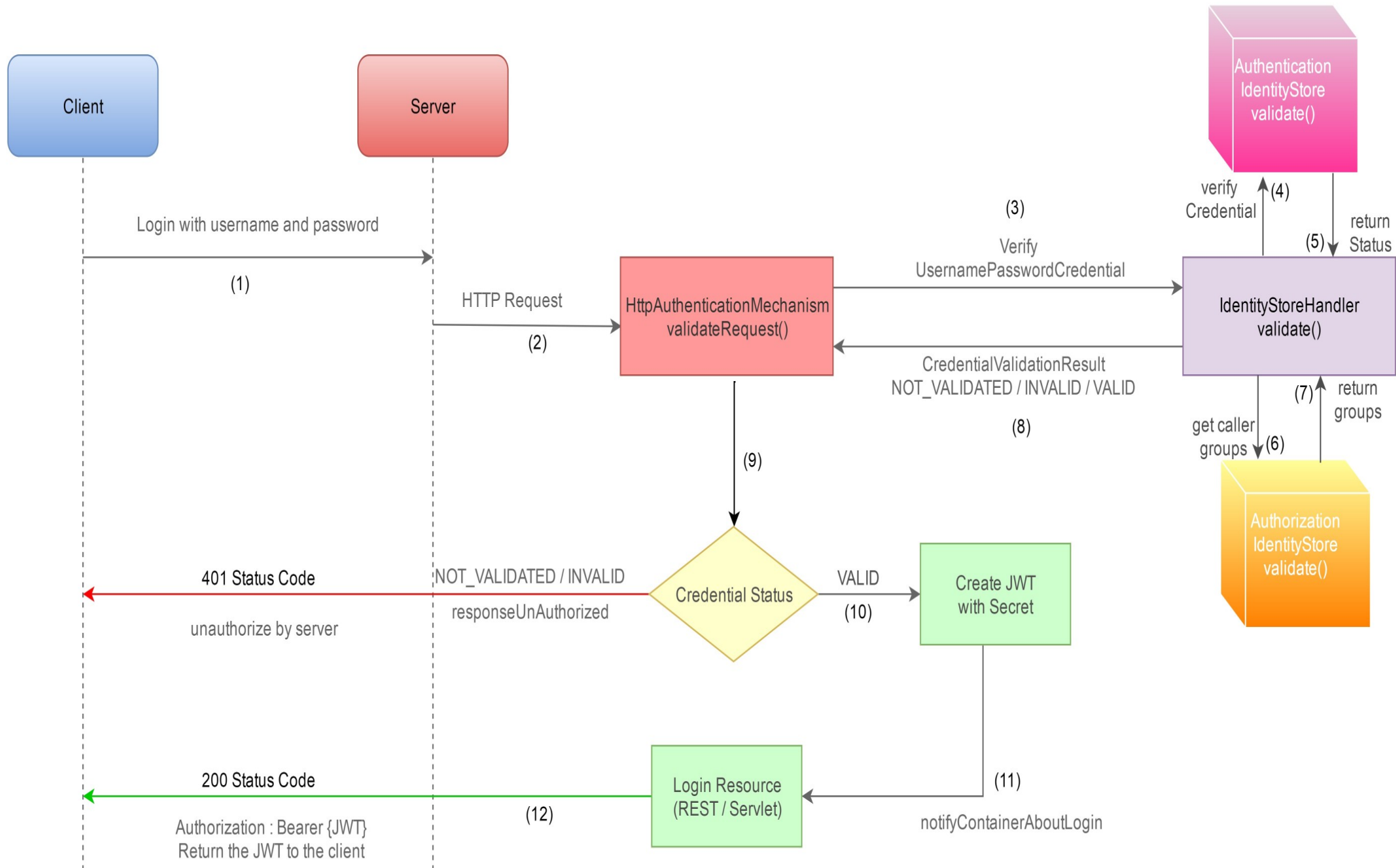
Header

Claims

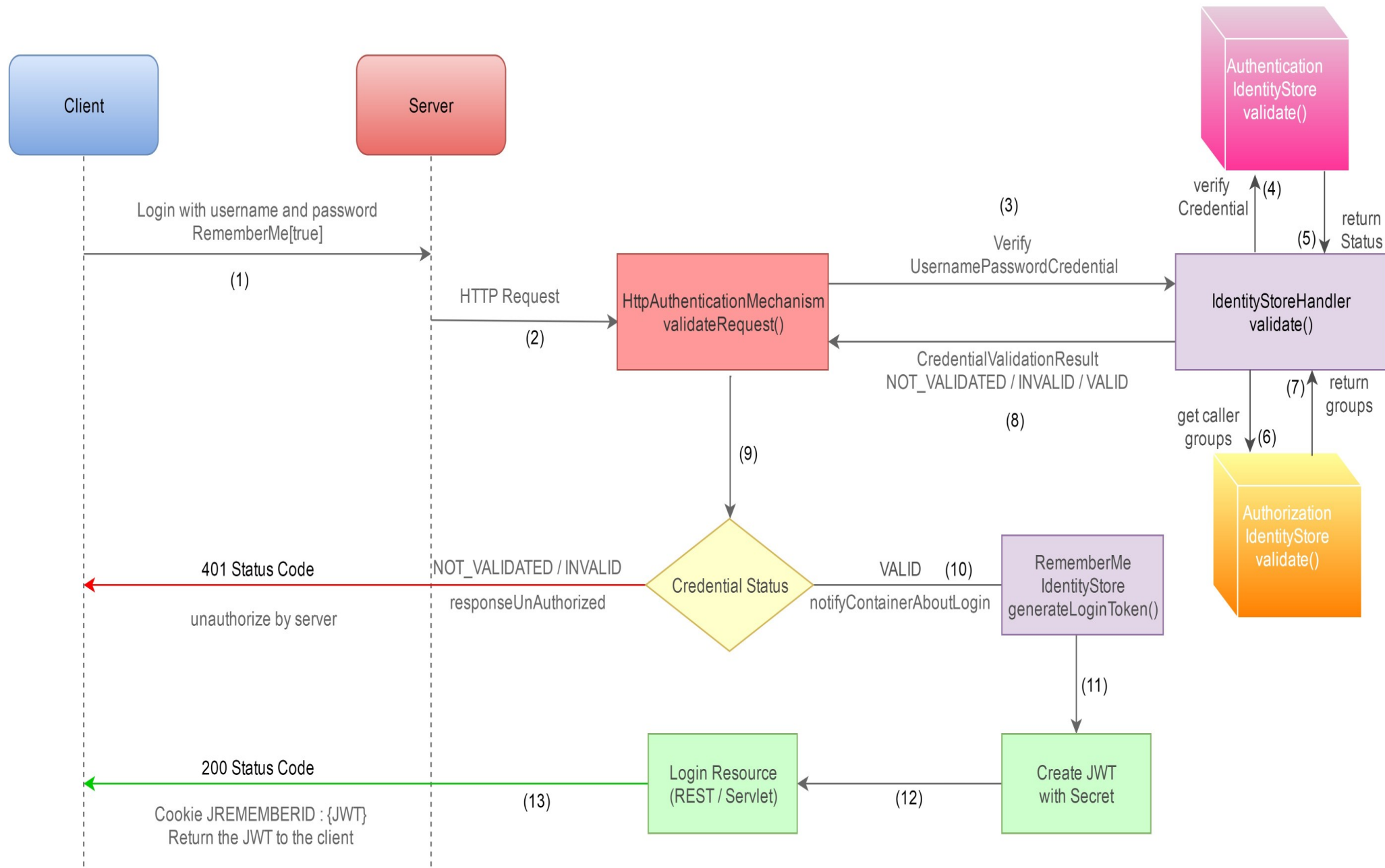
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleH
AiOjE0MTY0NzE5MzQsInVzZXJfbmFtZSI6InVzZXIiL
CJzY29wZSI6WyJyZWFKIiwid3JpdGUiXSwiYXV0aG9y
aXRpZXMioIlsiUk9MRV9BRE1JTjIsIlJPTEVfVFNFUjJ
dLCJqdGkiOiI5YmM5MmE0NC0wYjFhLTRjNWUtYmU3MC
1kYTUyMDc1YjlnODQiLCJjbGllbnRfaWQiOiJteS1jb
GllbnQtd2l0aC1zZWNyZXQifQ.AZCTD_fiCcnrQR5X7
rJBQ5r0-2Qedc5_3qJJf-ZCvVY

Signature

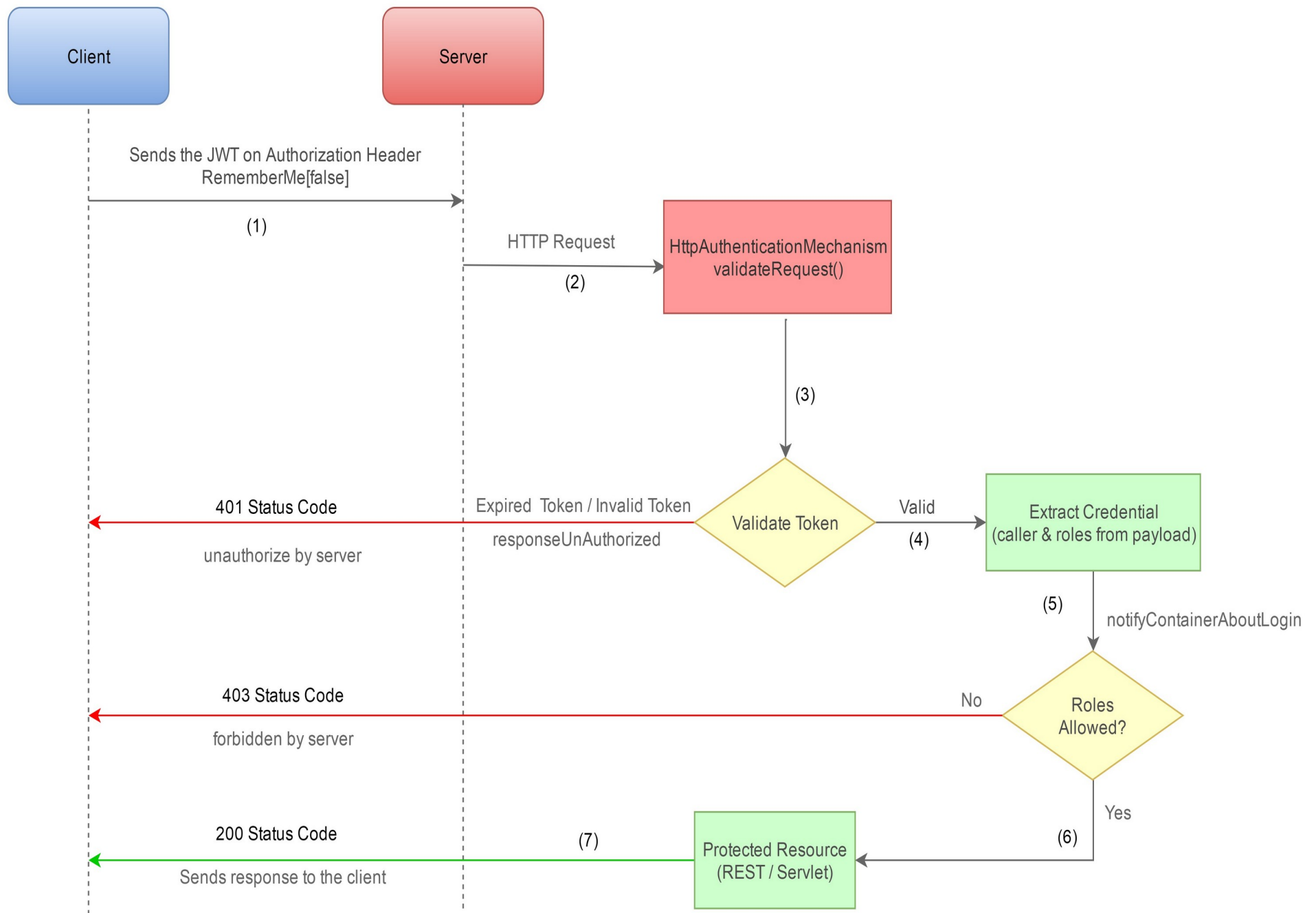
PAYARA – Authentication with JWT



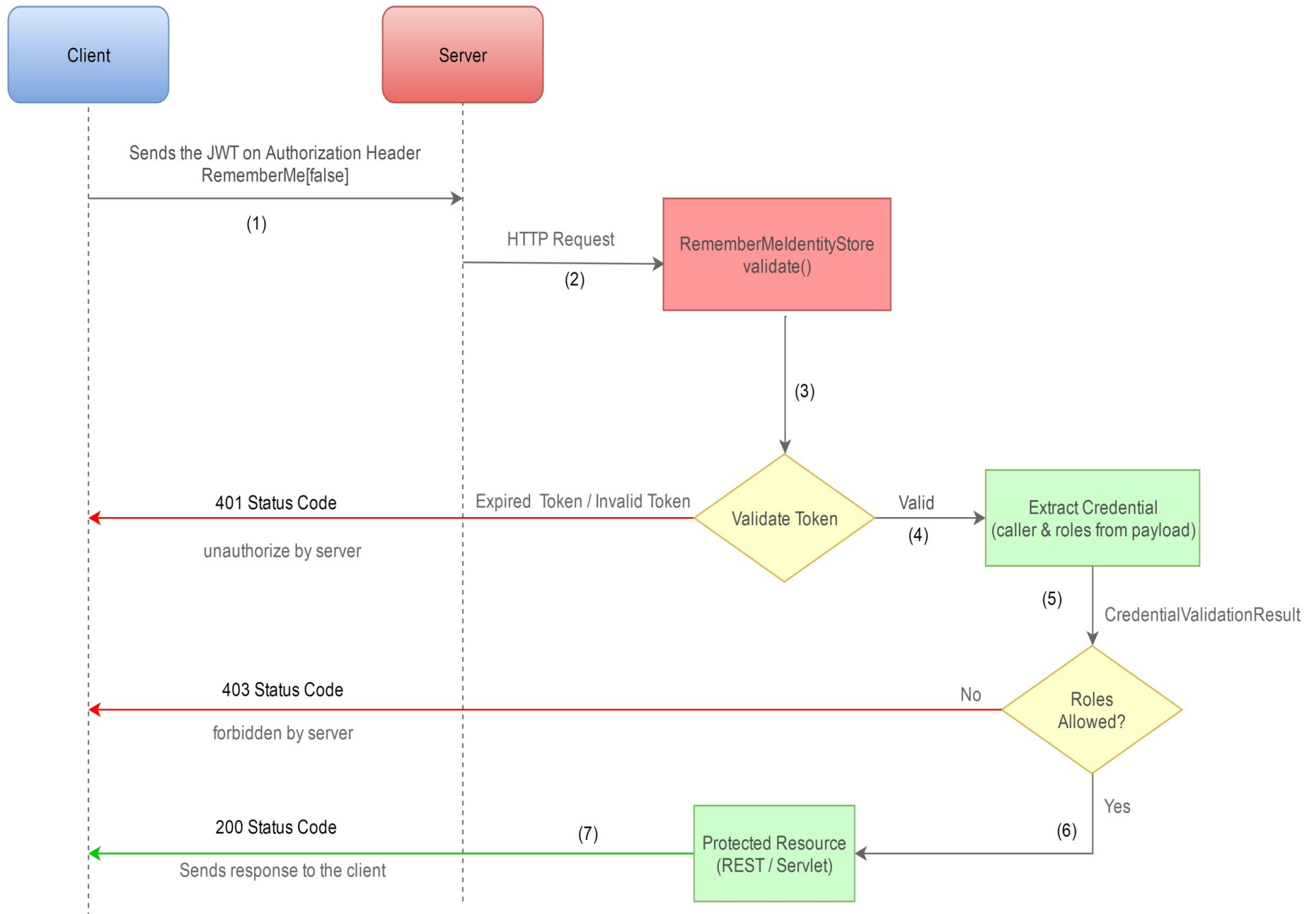
PAYARA – Remember Me Authentication with JWT



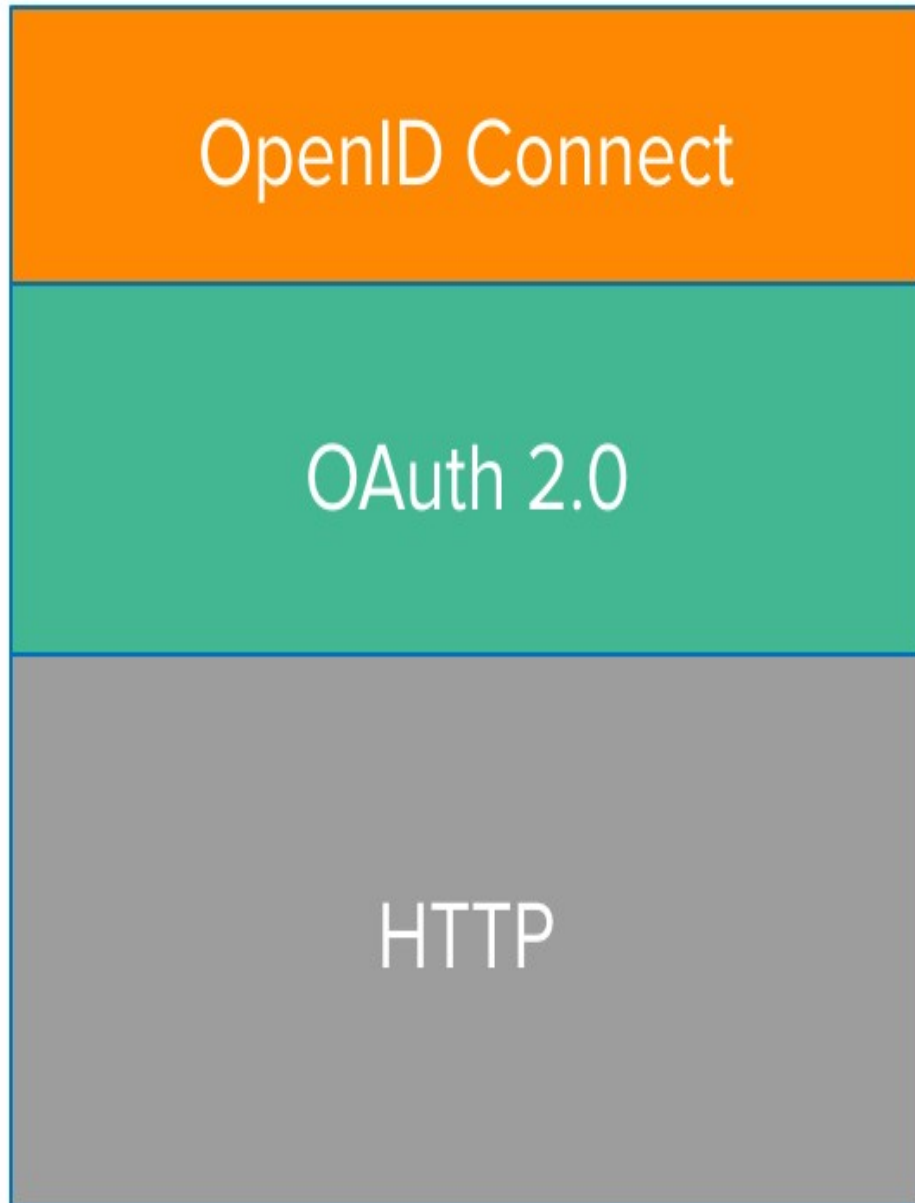
PAYARA – Authorization with JWT



PAYARA – Authorization with JWT (JREMEMBERID Cookie)



JAVA EE 8 SECURITY



OpenID Connect is for
authentication

OAuth 2.0 is for
authorization

Thank You

Lets Have some workable examples

shutterstock.com • 780491263