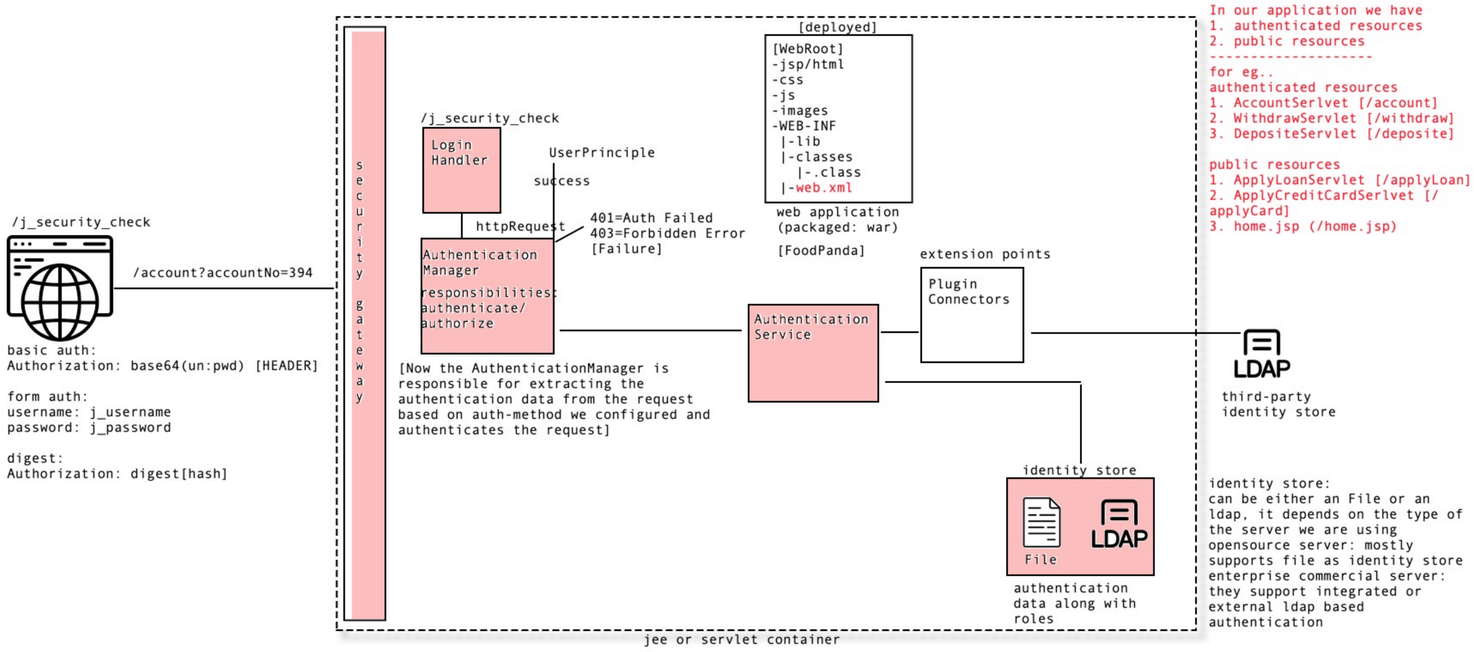

Few Points:
1. always the incoming request would be received by the ServletContainer or Jee container, so he is the one first-in place responsible for enforcing the security in granting the access to the underlying resources of the application, before forwarding the request to our application.

---

2. The enduser should send the authentication data (username/password) aspart of the HttpRequest, letting the server authenticate in granting the access to the authenticated resource.
The HTTP protocol has standardized the mechanism in passing the authentication data between the client and the server these are called Authentication Mechanisms. There are 3 ways the client/user can send the username/password aspart of the HttpRequest which are
1. Basic Authentication
2. Form-based Authentication
3. Digest Authentication

1. Basic Authentication = The client has to send the username & the password as part of the http request header as
"Authorization" : "Basic base64Encoded(username:password)"
so that server can read the username/password to authenticate the user

2. Form-based Authentication
The application can create an login form, letting the user to input the username/password to login, these values are sent as part of Request Body so that server can read these request parameters in authenticating the user

3. Digest Authentication
The server gives an unique code (nonce) that is one-time usable to the client. The client has to do an MD5Hash(nonce, un, pwd, requestURI) generates an hashstring out of it and sends it aspart of the request to the server.
The server upon receiving the request will takes the client hash and runs through the list of usernames/passwords it has by applying the same MD5 hash on each value and generated hash will be compared with client hash to authenticate and identify the user trying to login
The digest is more robust authentication mechanism that avoid stealing and replaying the username/password of the client

These are the standard security mechanims standardized by HTTP protocol and are adopted by all the servers of any language include jee servers and servlet containers also.

The developer while designing the application should choose which authentication mechanism he want to use in authenticating the user and he should let the servlet container understand/know about it. So that Servlet Container upon receiving the request, he will look for the authentication data in the request based on the mechanism we specified.

How to tell the servlet container, which authentication mechanism we want to use for our application?
The only way to define the information about our application to the servlet container or jee container is through web.xml, so in the web.xml we should specify the authentication mechanism as

```
<login-config>
  <auth-method>BASIC/FORM/DIGEST</auth-method>
</login-config>
```

---

3. In addition to the login-config, we need to define which resources are authenticated resources and which are public resources along with the roles of the users who are permitted to access, so that the ServletContainer upon receiving the request it can check before granting the access to our resources of our application
This can be done by writing security-constraints (security rules) in web.xml

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>account</web-resource-name>
    <url-pattern>/account</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>administrator</role-name>
    <role-name>clerk</role-name>
    <role-name>customer</role-name>
  </auth-constraint>
</security-constraint>
```

---

4. along with that who are the roles of users permitted to access our application in web.xml using

```
<security-role>
  <role-name>customer</role-name>
</security-role>

<security-role>
  <role-name>administrator</role-name>
</security-role>

<security-role>
  <role-name>clear</role-name>
</security-role>

<security-role>
  <role-name>manager</role-name>
</security-role>
```