1. RunTime Security:
   1. Java Sandbox Environment. Need policies.conf to access Files, Socket etc…
2. Authentication:
   1. In a simple web-app, username and password is encrypted (base64, sha hashing.) and sent (without https). Then server gives a token and then flow goes.
3. Authentication and Authorization:
   1. Using JAAS for roles based authN and authZ. Your code will be coupled with server.
   2. Using JWT, OAUTH to offer a client valet key to access a set of services. Logic session handling is separated from Application Server. Auth-Server takes care of creating token for a user, validating the token when a user asks for a service. Token is valid for a 15 or 30 minutes. Then Use Refresh\_Token to get updated token.
   3. This type of valet key is used mostly in client-server application.
   4. This doesn’t talks about how service data is managed. Ex: get Employee data. Confidentiality and Integrity of application is dependent on TLS or HTTPS.
4. Sometimes we want to send data safely across network. Ex: Trusteer scans the browser and sends result to Trusteer network. RBS after verifying the UserName and Password calls Trusteer network to get the Brower threat level. Here Browser to Trusteer network info is encrypted and sent using JavaScript.

From RBS Data Center to Trusteer, a secure https communication is made using certificates.