Federated Identity in Cloud Ecosystem

USE CASE PROCESS FLOW
OpenStack Use Case
Request Flow for Provisioning Instance in OpenStack
Open Stack – Authentication Process Flow

1. User Initiate Service Request
2a. Route User to Keystone (HTTP 301 Redirect)
2b. SAML2 or OpenID Connect Request
3a. Route User to Appropriate IdP
3b. Authenticate to IdP with SAML / OpenID Request and User ID / Password
4a. Provide SAML Assertion / OpenID Claim
4b. Request Service with SAML Assertion / OpenID Claim
5a. Java Script code that includes Federated Token
5b. Service Request with Federated Token
SDP Use Case
SDP Client On-boarding (IdP scenario)

Identity Provider

Controller Machine or Network

Credential Maker

Controller (separate, restful API)

Client Machine

SDP Client App

User

1. Admin creates new User-Machine ID via controller API. (see Note 2)

5. SDP client requests identity token from IdP

6. User authentication exchange

8. Client sends request to be onboarded, includes identity token

9. Credential Request

10. New Credentials

11. New Credentials

12. New Credentials

13. Credential ACK

14. Set “on-boarded” flag

15. Notify all relevant gateways

7. IdP sends user identity token

2. User installs client software

3. User starts client software

4. User enters controller address

Credentials per User-Device Pair:
1. TLS Encryption Key
2. TLS Certificate
3. SPA Encryption Key
4. SPA HMAC Key
SDP Client On-boarding (without IdP)

1. Admin enters new user data via controller API.
   Data can come from Active Directory, PKI server, MDM.
   (see Note 2)

2. Controller stores user data in database

3. User installs client software

4. User starts client software

5. User enters controller address

6. Client sends https request to be on-boarded

7. Credential Request

8. New Credentials

9. New Credentials

10. New Credentials

11. Client sets “on-boarded” flag

12. Credential ACK

13. Set “on-boarded” flag

14. Notify all relevant gateways

Credentials per User-Device Pair:
1. TLS Encryption Key
2. TLS Certificate
3. SPA Encryption Key
4. SPA HMAC Key
## SPA Packet

<table>
<thead>
<tr>
<th>Clear Text</th>
<th>Client ID</th>
<th>32-bit numeric identifier, assigned per user-device pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher Text</td>
<td>Nonce</td>
<td>16-byte random data field, prevents replay attack</td>
</tr>
<tr>
<td></td>
<td>Timestamp</td>
<td>Prevents servicing outdated SPA packets</td>
</tr>
<tr>
<td></td>
<td>Message Type</td>
<td>Basic access request or NAT access request. Can also enable gateway command message type.</td>
</tr>
</tbody>
</table>
| | Message String | Source IP address to allow, protocol/port number to open  
NOTE: Instead of port, client may specify service ID. Gateway would know which port to open as well as whether and where to forward the connection. |
| | Optional Fields | NAT request details – internal destination IP address, internal destination port.  
NOTE: This field is likely to become obsolete in favor of using service ID. |
| | Digest | Before encryption, this SHA256 hash is calculated over the cipher text portion of the message and then used by the server to verify message integrity after a successful message decrypt. |
| Clear Text | HMAC | Calculated over all fields above, including Client ID. Algorithm choices are MD5, SHA1, SHA256 (default), SHA384, and SHA512. |
### SDP Timing Diagram (Service Gateway Focus)

**Gateway Credentials:**
1. TLS Encryption Key
2. TLS Certificate
3. SPA Encryption Key
4. SPA HMAC Key

**Controller Machine or Network**
- **Credential Maker**
- **Controller**
- **SDP Gateway**

**Service Machine or Network**
- **SDP Gateway**
- **ctrl client**
- **Service**

#### SSL Diagram
1. SPA
2. mTLS Connection

**Steps:**
1. SPA
2. mTLS Connection
3. Credential Request (See note 2.)
4. New Credentials
5. New Credentials
6. Credential ACK
7. New Credentials
8. Notify all relevant gateways
9. Access Refresh Request
10. Access Refresh
1. SPA
User then free to access service

2. User accesses service
Includes any required authentication steps specific to the service
SDP Timing Diagram (Client Credential Refresh)

User accesses service first, credential update occurs in background.

Controller Machine or Network

<table>
<thead>
<tr>
<th>Credential Maker</th>
<th>Controller</th>
<th>fwknop gate</th>
</tr>
</thead>
</table>

Client Machine

| SDP Client App | 1. SPA | User now free to access service |

Service Machine or Network

| ctrl client | Service |

Credentials per User-Device Pair:
1. TLS Encryption Key
2. TLS Certificate
3. SPA Encryption Key
4. SPA HMAC Key

Flow:
1. SPA
2. SPA
3. mTLS Connection
4. Credential Request (See note 2.)
5. New Credentials
6. New Credentials
7. Credential ACK
8. New Credentials
9. Notify all relevant gateways
SSL Timing Diagram

Notes:
- Key Length = 2048 bit RSA or 256 bit ECC
- Hash algo – strong SHA-2 for cert
- Guard root certificate
- Monitor private SSL keys
- Use TLS 1.2, latest OpenSSL, & AES encryption
- No ShellShock on Linux or RC4

Has public key + private key

Creates digital signature by taking data, creating a hash and encrypting hash with private key

Send public key to establish an encryption method

Send encryption method

Send session + unique session key (?), + digital signature

Decrypts hash, checks data with public key for tempering, & trusts originator

Unique session key protects message privacy, message integrity & server security
AWS Use Case
AWS (Flood Ins. Service/Site Access)

1. User Logs in to Insurer System
2. SAML Authentication Statement Request Access To Flood Ins. System
3a. Error Returned to Caller
3b. User is redirected to Flood Ins. Authenticated pages
4. Authenticated Session Token

Business Agreement