

# oAuth & JWT

Hybrid Integration Enablement

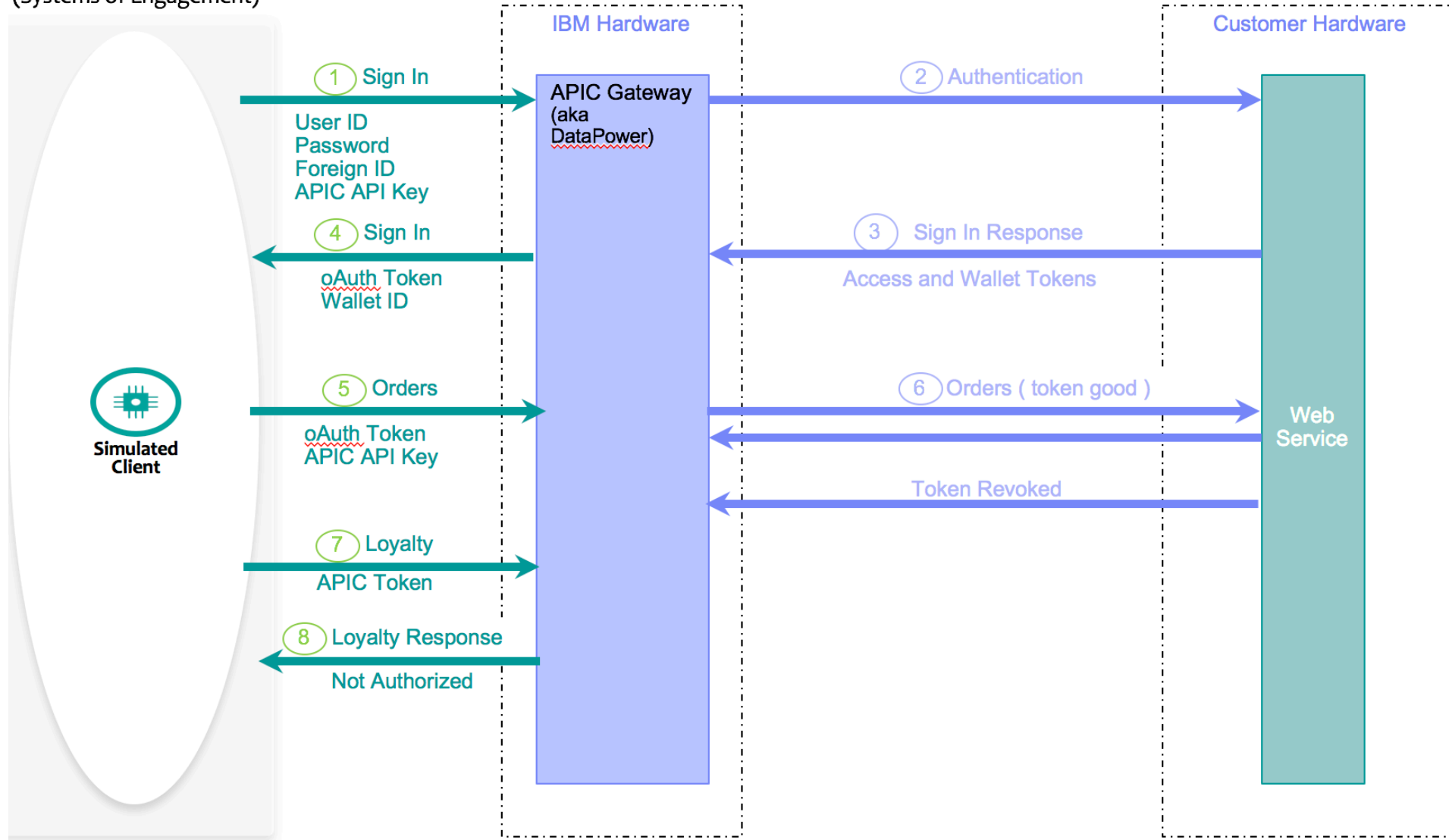
October 2017

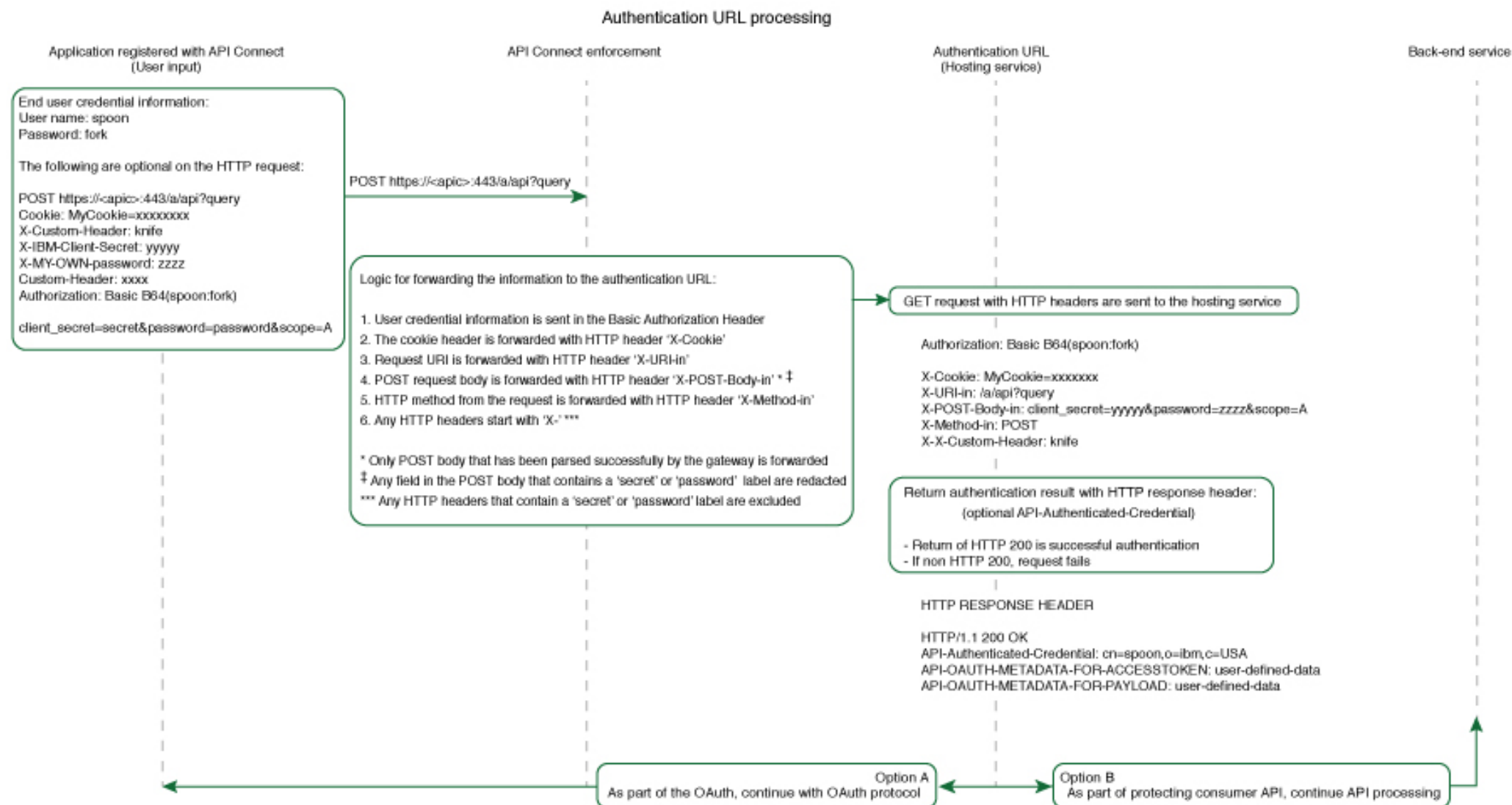


# oAuth 2 Legged

Use Case: I have a mobile app and I want to oAuth protect my api so API Gateway can handle session management?


## Consumer (Systems of Engagement)





# APIC Conducting Token Management

- Provides endpoints to generate tokens



```
GET /oauth2/authorize
POST /oauth2/authorize
POST /oauth2/token
```

- Provides endpoints to manage tokens



```
POST /oauth2/introspect
GET /oauth2/issued
DELETE /oauth2/issued
```

# JSON Web Token

Use Case: Do you have support for JWT

JSON Web Token (JWT) is:

- A compact, URL-safe way of representing claims that are to be transferred between two parties.
- Based on RFC 7519 that defines a self-contained way for securely transmitting information between parties as a **JSON** object.
- A policy that can be verified and trusted because it is digitally signed.
- Composed of a header, a payload, and a signature

# Good Use Cases for JWT

---

## When should you use JSON Web Tokens?

Here are some scenarios where JSON Web Tokens are useful:

- **Authentication:** This is the most common scenario for using JWT. Once the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token. Single Sign On is a feature that widely uses JWT nowadays, because of its small overhead and its ability to be easily used across different domains.
- **Information Exchange:** JSON Web Tokens are a good way of securely transmitting information between parties. Because JWTs can be signed—for example, using public/private key pairs—you can be sure the senders are who they say they are. Additionally, as the signature is calculated using the header and the payload, you can also verify that the content hasn't been tampered with.



- The Generate JWT policy enables you to generate claims and configure whether they are to be used as either:
  - the payload of a JSON Web Signature (JWS) structure
  - the plain text of a JSON Web Encryption (JWE) structure.
  - specifying the cryptographic material for both the JWS and the JWE produces a nested JWT that is both digitally signed and encrypted.
- The JWT is then assigned to the Authorization header as either:
  - a Bearer token (the default option)
  - the runtime variable in the JSON Web Token (JWT) property, if specified.
- You can attach this policy to the following API flows:
  - REST
  - SOAP

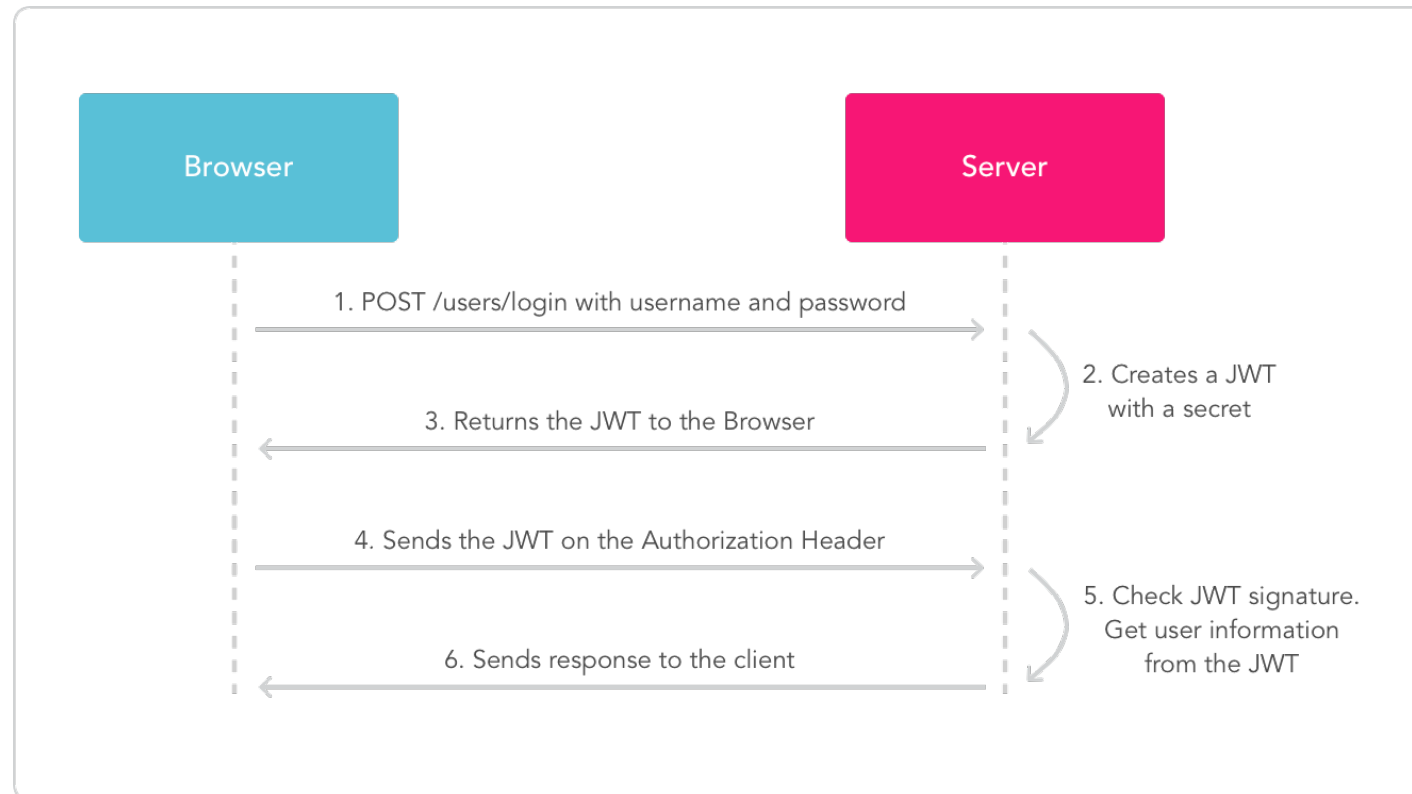
## Pre requisites

---

- IBM API Connect Version 5.0.1 or later.
- IBM DataPower V7.5 with the Application Optimization (AO) option – not supported in Micro Gateway
- If you are using one or more cryptographic objects, they must be located in the IBM API Connect domain on the DataPower appliance. The cryptographic objects must reference the Shared Secret Key or certificate that is needed to encrypt or sign the JWT contents.
- If a JSON Web Key (JWK) is being used, it must be referenced by a runtime variable.

# How it works

- User signs into an authentication server
- JWT returned to authenticated user
- User passes JWT when making API calls
- Application verifies and API call allowed to proceed



# oAuth/ JWT Questions

---

- Open Conversation / Questions?

Back to [Presentation](#) Topics