InterSystems
Creative data technology

Deep Dive into SMART on FHIR®

Tani Frankel
Sales Engineer Manager

SMART DAY

SMART ON FHIR TRAINING
& WORKSHOP DAY

08:30 - 09:00	Registration and Welcome Coffee
09:00 - 09:20	Welcome by Community (Adi / Ronen)
09:20 - 09:50	Introduction to SMART on FHIR
09:50 - 10:20	Understanding FHIR
10:20 - 10:50	Understanding OAuth 2.0 and Identity Providers
10:50 - 11:10	Coffee Break
11:10 - 12:10	Deep Dive into SMART
12:10 - 13:00	Lunch Break
13:00 - 14:30	Hands-on Session: Building a SMART App & Setting-up OAuth
14:30 - 15:00	Coffee Break
15:00 - 16:30	Hands-on Session: Integrating with FHIR Server
16:30 - 17:00	Q&A and Closing Remarks

FHIR® IL by 8400 aws InterSystems Creative data technology

The Lupa Equivalent

The

Authorizing requests with OAuth 2.0

All requests to the Google Photos APIs must be authorized by an authenticated user.

The details of the authorization process, or "flow," for OAuth 2.0 vary somewhat depending on what kind of application you're writing. The following general process applies to all application types:

1. When you create your application, you register it using the Google API Console. Google then provides information you'll need later, such as a client ID and a client secret.
2. Activate the Google Photos APIs in the Google API Console. (If the API isn't listed in the API Console, then skip this step.)
3. When your application needs access to user data, it asks Google for a particular **scope** of access.
4. Google displays a **consent screen** to the user, asking them to authorize your application to request some of their data.
5. If the user approves, then Google gives your application a short-lived **access token**.
6. Your application requests user data, attaching the access token to the request.
7. If Google determines that your request and the token are valid, it returns the requested data.

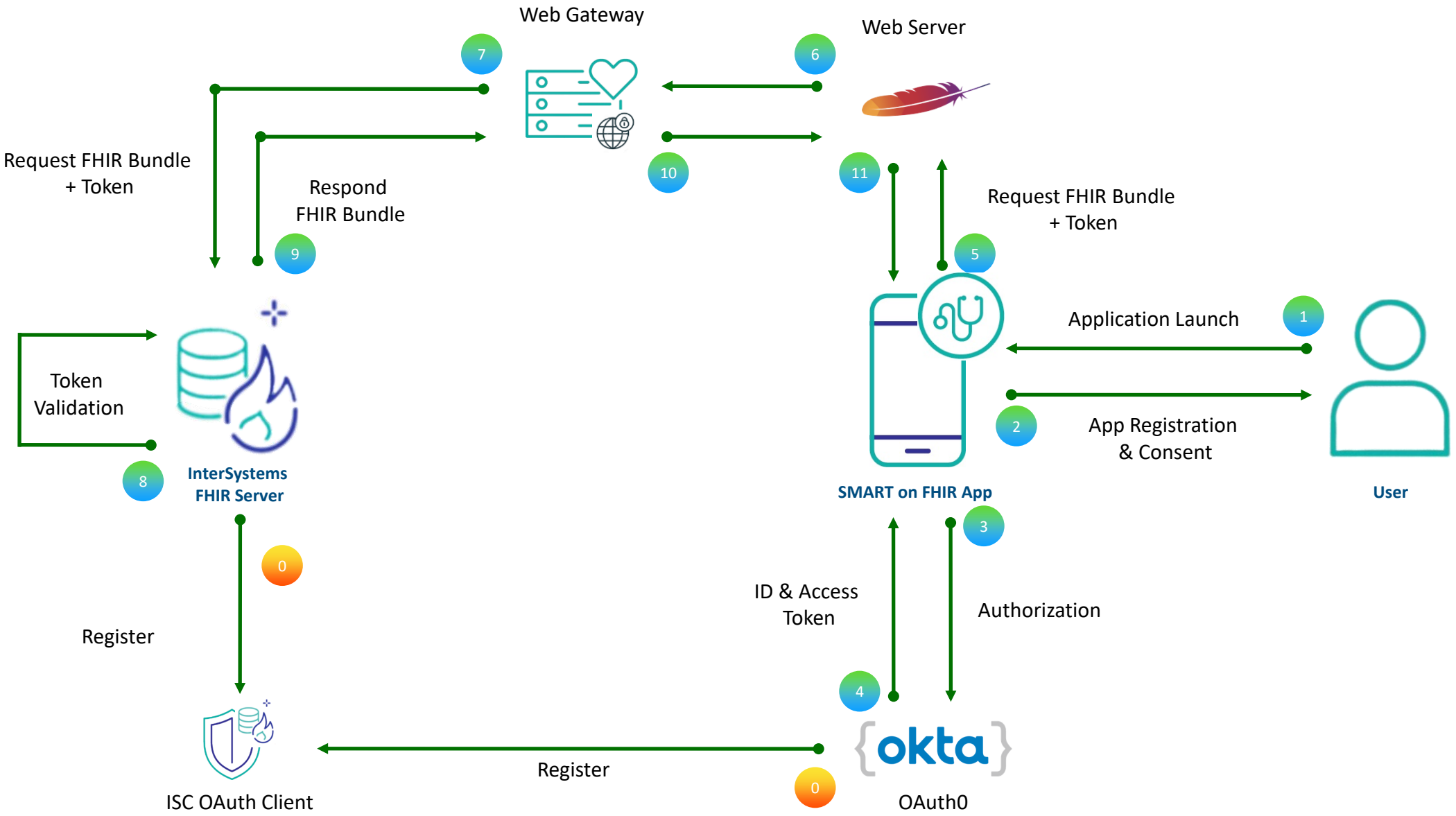
Some flows include additional steps, such as using **refresh tokens** to acquire new access tokens. For detailed information about flows for various types of applications, see Google's OAuth 2.0 documentation.

Here's the OAuth 2.0 scope information for the Google Photos APIs:

Picker API scopes

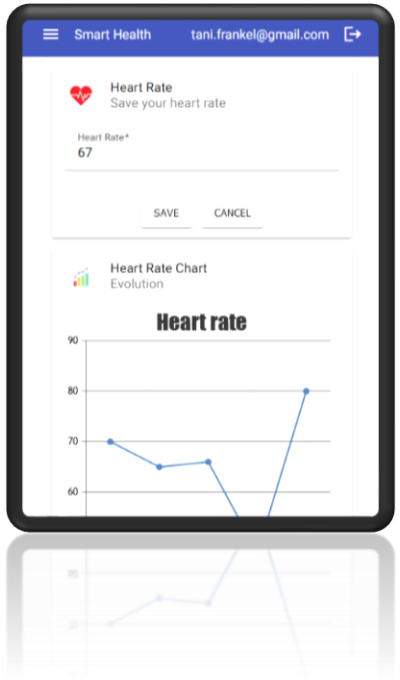
Scope	Meaning
`https://www.googleapis.com/auth/photospicker.mediaitems.readonly`	Access to create, get, and delete sessions, and to list media items for sessions.

SMART App Sample Flow

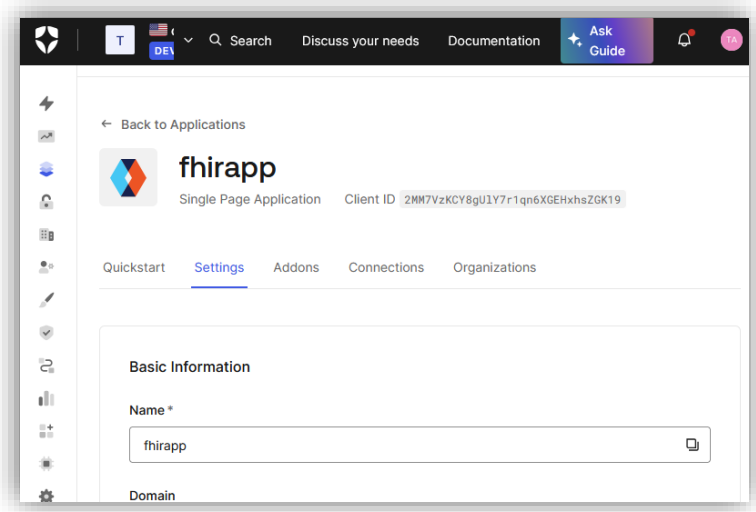


Hands-on Exercise

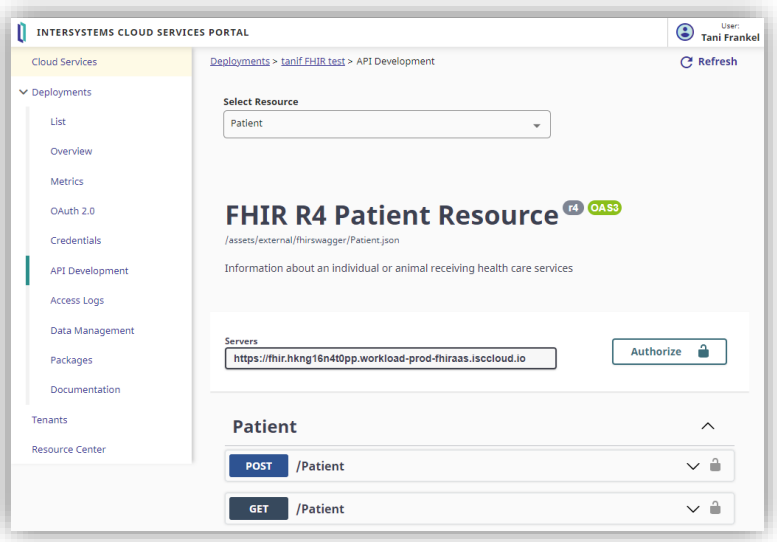
Application



OAuth Server



FHIR Server



Understanding OAuth2 Scopes in SMART on FHIR

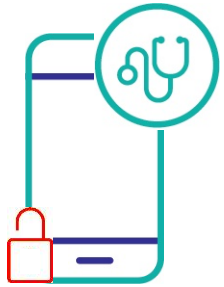
- **Key Concepts**
  - Oauth2 for authorization
  - **Scopes** define the **level of access** granted to the SMART app
  - Examples of common scopes:
    - `patient/*.read`, `user/*.write`, `launch`, `openid`,
- **Types of Scopes**
  - User-Level: Authenticated user-based (e.g. `user/*.read`)
  - Patient-Level: Patient-specific data (e.g. `patient/*.read`)
  - System-Level: System access for server-to-server communication (e.g. `system/*.read`)

Common SMART on FHIR Scopes

Scope	Explanation
openid	Enables OpenID Connect (OIDC) for identifying users
launch/patient	App launched in context of a patient and need the Patient Resource
Launch/encounter	App launched in context of an encounter and need the Encounter Resource
Offline_access	Enables long-lived access even when users are no longer online

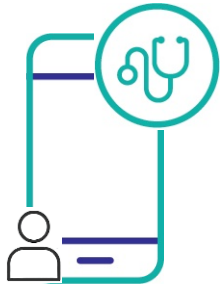
Client Overview (App)

Client Types



Confidential Clients

Can include secrets in the app, likely does not need consent



Public Clients

Cannot include a secrets in the app, probably require consent

Elements of Application Registration

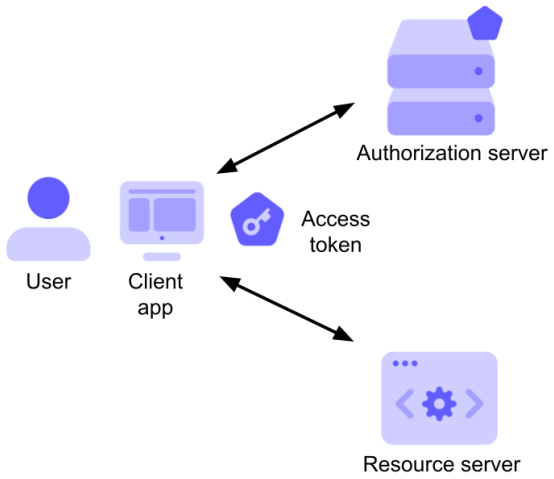
1. Takes place on the Authorization Server
2. A unique client ID is created
3. Usually will get a client secret (the app's password).
   Note: should only be used in backend applications
4. Application Name – possibly shown to users
5. Application redirect – extremely important, means that attackers can not steal credentials and direct to their website
6. Application Type – affects token lifetimes

Tokens Overview

Tokens are Fundamental in OAuth

- **Access**

1. **Header: Data about the token's type and the algorithm used to make it are included here.**
2. **Payload: Information about the user, including permissions and expirations, is included here.**
3. **Signature: Verification data, so the recipient can ensure the authenticity of the token, is included here. This signature is typically hashed, so it's difficult to hack and replicate.**



- **ID**

What Can You Get in an ID Token?

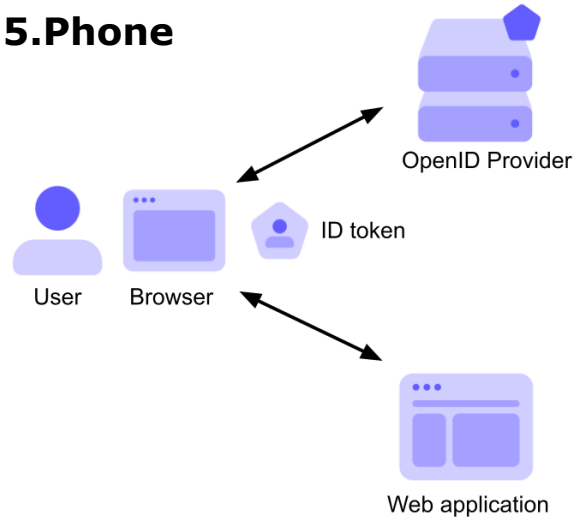
1.OpenId

2.Profile

3.Email

4.Address

5.Phone



https://auth0.com/blog/id-token-access-token-what-is-the-difference/

Access Token vs ID Token

Access Token

- Not read by the application
- Used by the application to get access to a given API
- “aud” - is the api or resource server

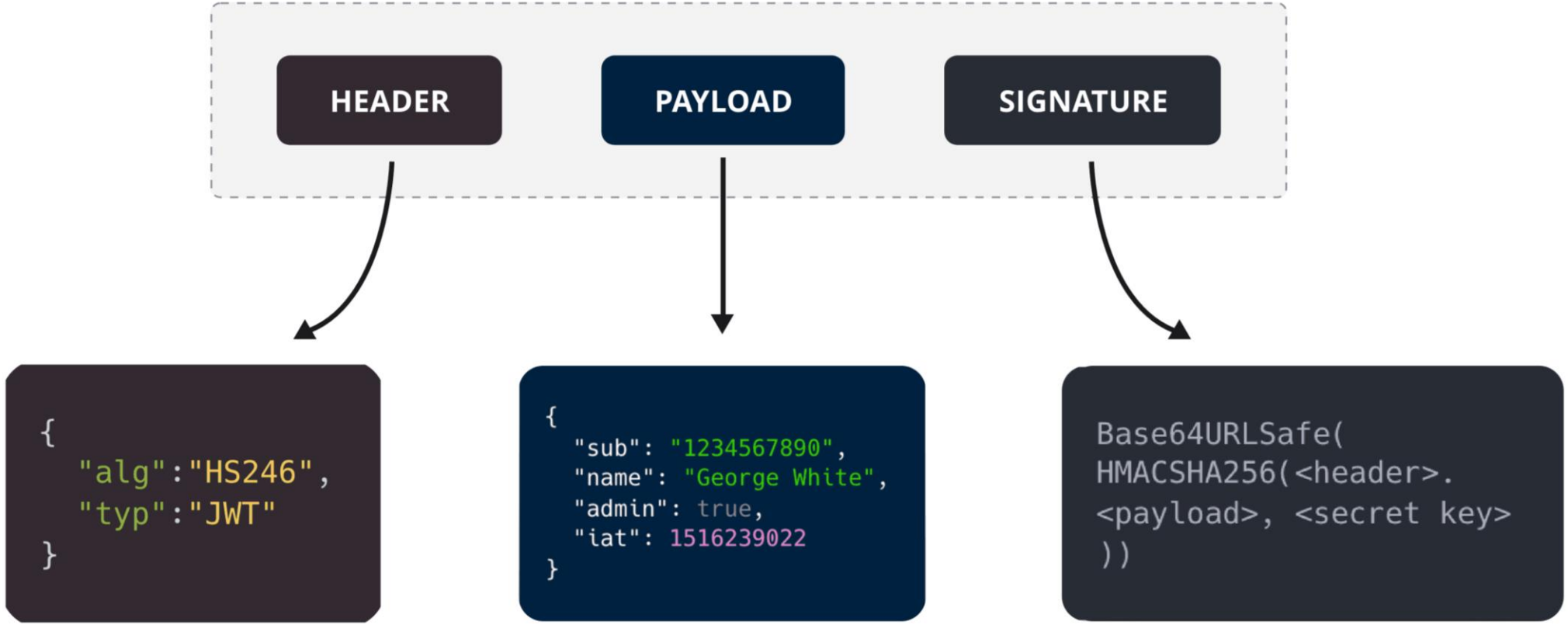

```
{
  "iss": "https://dev-1h5yru1mv5rucu2k.us.auth0.com/",
  "sub": "auth0|63dbb2bb9911bfb5e7935d9a",
  "aud": "https://fhir.intersystems.internal.",
  "iat": 1675365807,
  "exp": 1675452207,
  "azp": "Mvfyl9FrJVahOl4r46Yf12FFP2zu010Z"
}
```


ID Token

- Unpacked by the application
- Provides information about the user
- “aud” – is the id of the application


```
{
  "iss": "https://dev-1h5yru1mv5rucu2k.us.auth0.com/",
  "sub": "auth0|63dbb2bb9911bfb5e7935d9a",
  "aud": 45672345,
  "iat": 1675365807,
  "exp": 1675452207,
  "azp": "Mvfyl9FrJVahOl4r46Yf12FFP2zu010Z"
}
```

JSON JWT Token Structure



Local Token Introspection

Local introspection means the token is unpacked and validated locally, without a request to a remote server.

The Token Introspection extension defines a mechanism for resource servers to obtain information about access tokens.

With this spec, Resource Servers can check the validity of access tokens, and find out other information such as which user and which scopes are associated with the token.

Token Lifetimes

1. If you want to increase the security of your APIs, use access token lifetimes that are extremely short (< 10 minutes)
2. Limits the risks of leaked tokens
3. Refresh token lessen the hit in user experience from short tokens

Authorization Server Overview

Authorization Server

{okta}

- **Protect the API from unauthorized access**
- **Logon occurs through re-direction at the authorization server endpoint**
- **Tokens are given to applications (token factory)**

Hi patrick.jamieson@intersystems.com,

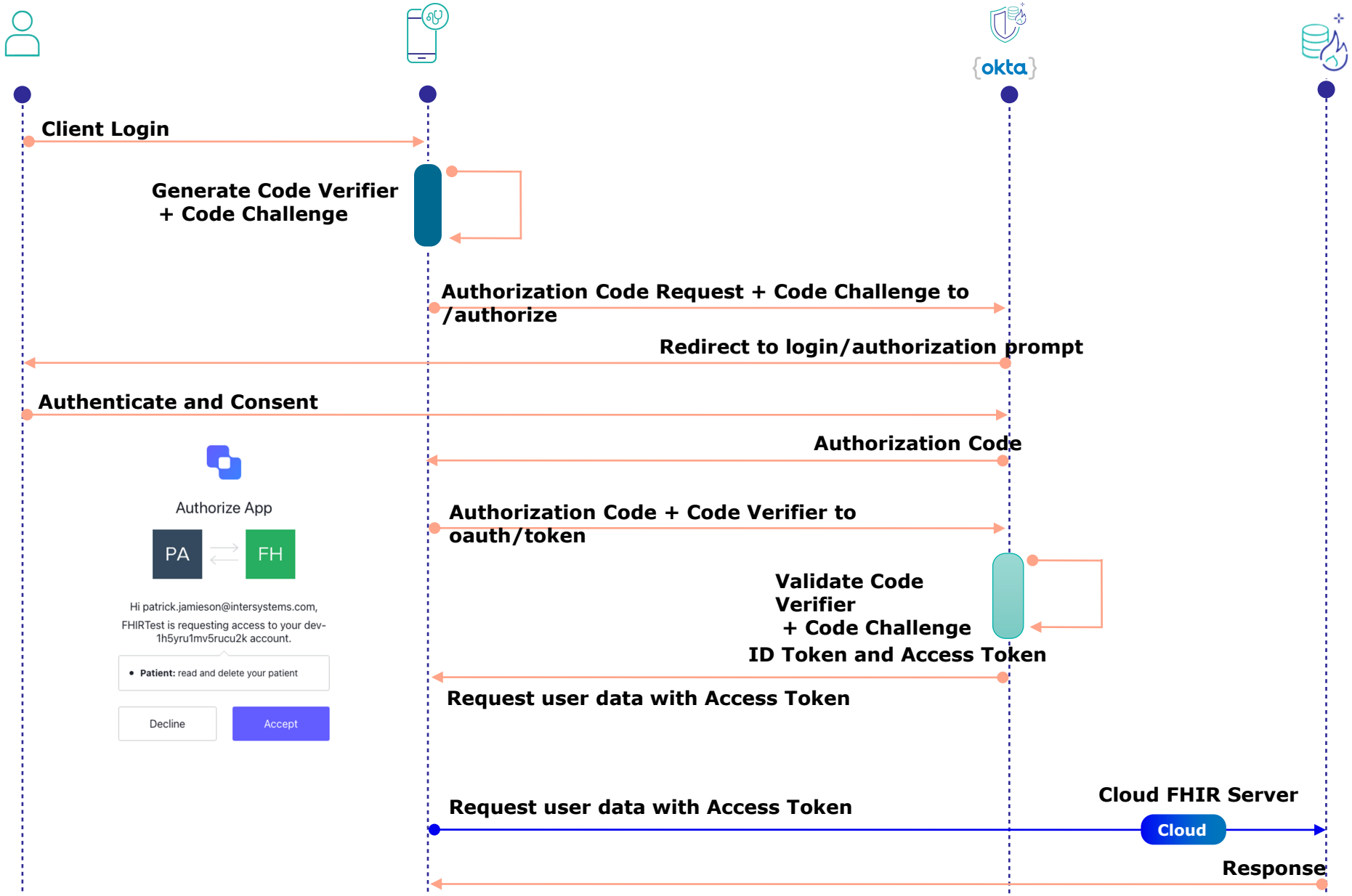
FHIRTest is requesting access to your dev-1h5yru1mv5rucu2k account.

- **Patient:** read and delete your patient

Decline

Accept

Authorization Flow + PKCE - Workflow Review



Scopes Overview

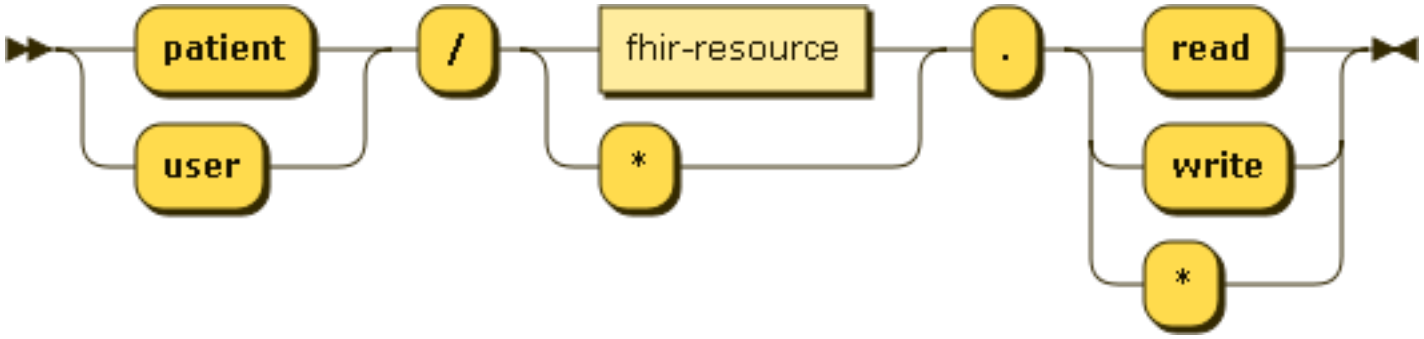
Scopes Characteristics

1. Not well defined in the OAuth.
2. Limit abilities of application.
3. Applications will “request” access.
4. OAuth does not define permissions for different groups.
5. Limit what an access token can do within the context of what a user can do.
6. Just strings, only used by the API.
7. Could use : to separate string.
8. Read, Write access very common.
9. Consent is typically added for third-party APIs.

SMART v1 scopes

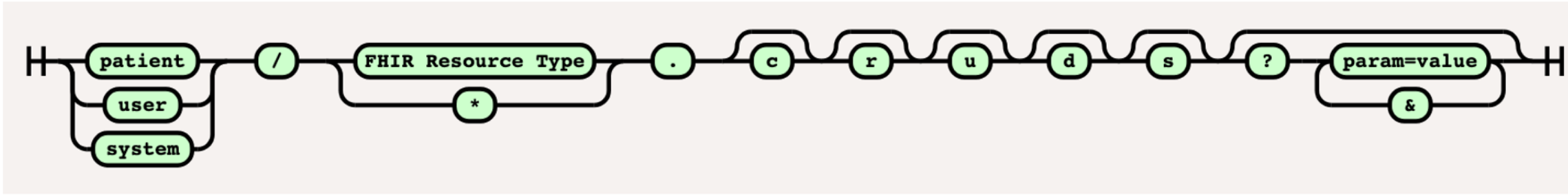
Patient-specific scopes allow access to specific data about a single patient. (You'll notice that we don't need to say which patient here: clinical data scopes are all about "what" and not "who.").
Patient-specific scopes take the form: patient/resourceType.(read|write|*).

('patient' | 'user') '/' (fhir-resource | '*') '.' ('read' | 'write' | '*')



SMART v2 scopes

('patient' | 'user' | 'system') '/' (fhir-resource | '*') '.' ('create' | 'read' | 'update' | 'delete' | 'search') '.' (param= `value'| &)



Examples

Goal	Scope	Notes
Read and search for all observations about a patient	patient/Observation.rs	
Read demographics about a patient	patient/Patient.r	Note the difference in capitalization between “patient” the permission type and “Patient” the resource.
Add new blood pressure readings for a patient	patient/Observation.c	Note that the permission is broader than the goal: with this scope, an app can add not only blood pressures, but other observations as well. Note also that write access does not imply read access.
Read all available data about a patient	patient/*.cruds	See notes on wildcard scopes below.

Access scopes using SMART v2

- **scope=user/Observation.r?category=laboratory**
  - **User is allowed to read Observation resources with a category element containing the code “laboratory”**

- **scope=user/Observation.rs?encounter.id=Encounter/456**
  - **User is allowed to see all Observation resources linked to the Encounter with id “456”.**

Q & A

Thank you

InterSystems®
Creative data technology