InterSystems
DACH SYMPOSIUM 2023

HEALTHCARE FORUM

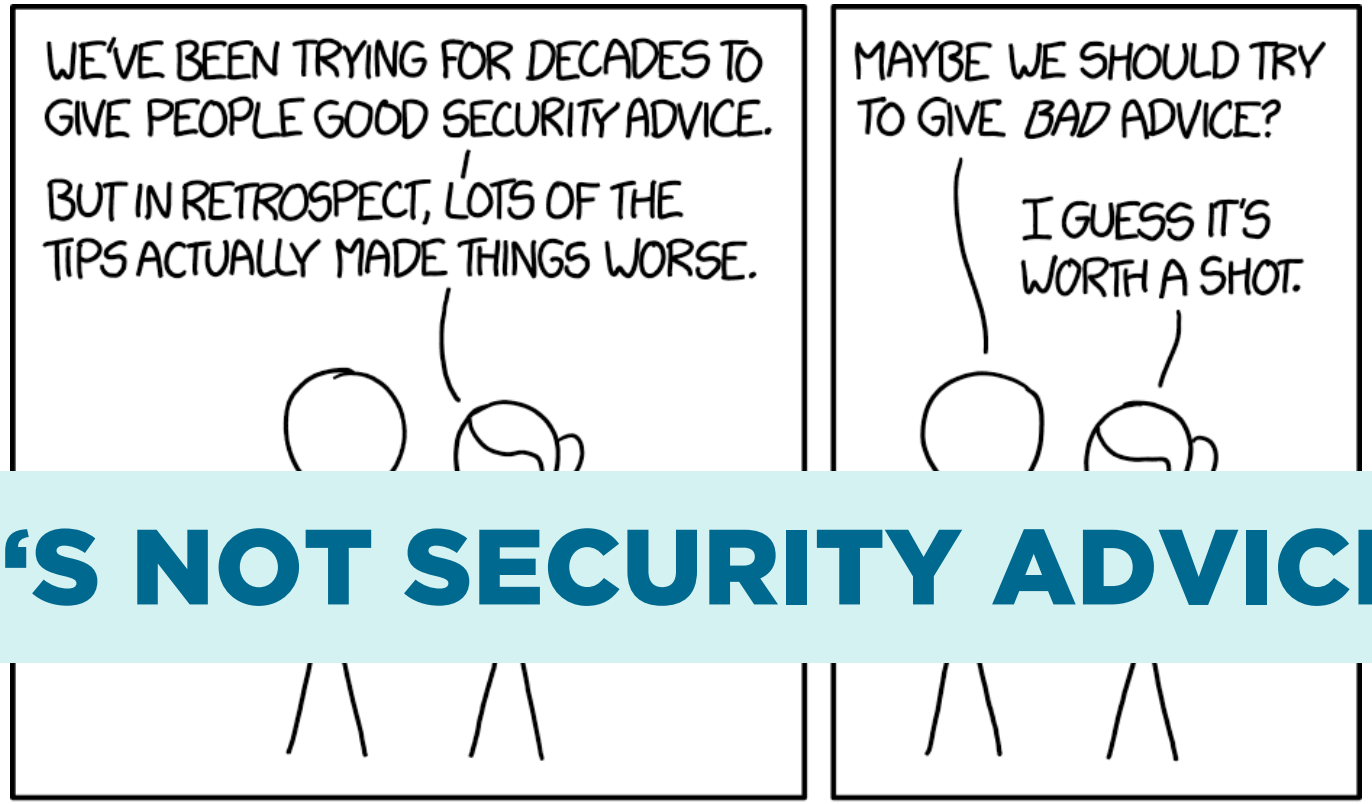
Meine Daten, deine Daten - FHIR-Ressourcen gut behütet

Nils Dittberner | InterSystems

Agenda

- Hintergrund und Zielsetzung
- FHIR Security
  - Spezifikationen
- OAuth2
  - Grundlagen
  - Scopes
- Richtig Behüten?!
- Fazit

Disclaimer



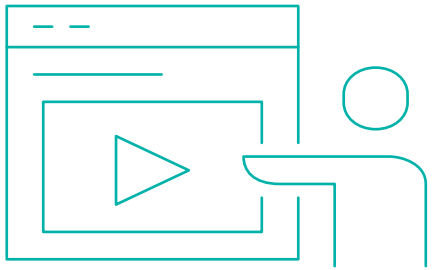
IT'S NOT SECURITY ADVICE!

Hintergrund und Zielsetzung

Meine Daten, deine Daten – FHIR-Ressourcen gut behütet

Demo I

- FHIR Server Configuration anpassen
  - Allow Unauthenticated Access
  - New Service Instance



FHIR Security

6.1.0 FHIR Security

Security Work Group	Maturity Level: 4	Standards Status: Trial Use

Fast Healthcare Interoperability Resources (FHIR) is not a security protocol, nor does it define any security related functionality. However, FHIR does define exchange protocols and content models that need to be used with various security protocols defined elsewhere. This section gathers all information about security in one section. A summary:

1. Time Keeping - all clocks should be synchronized using NTP/SNTP, and the design of the system should be robust against a system clock with the wrong value
2. Communications Security - all exchange of production data should be secured using TLS (e.g., https).
3. Authentication - Users/Clients must be authenticated. For web-centric, OAuth is recommended, consider use of HL7 SMART App Launch Implementation Guide where appropriate.
4. Authorization/Access Control - FHIR defines a Security Label infrastructure to support access control management. FHIR may also define a set of resources to administer access control management, but does not define any at present
5. Audit - FHIR defines provenance and audit event resources suitable for tracking the origins, authorship, history, status, and access of resources
6. Digital Signatures - FHIR includes several specifically reserved locations for digital signatures
7. Attachments - FHIR allows for binary resources and attachments. These have their own concerns
8. Labels - FHIR allows for set of security related tags that affect the way resources are handled
9. **Data Management Policies** - FHIR defines a set of capabilities to support data exchange. Not all the capabilities that FHIR enables may be appropriate or legal for use in some combinations of context and jurisdiction (e.g. HIPAA, GDPR). It is the responsibility of implementers to ensure that relevant regulations and other requirements are met.
10. Narrative - Care must be taken when displaying the narrative from FHIR resources
11. Input Validation - Validate all input received from other actors to assure the data is well formed and does not contain content that would cause unwanted system behavior. Testing ensures that the input is not susceptible to data input validation errors by using techniques such as fuzzing, invalid input attacks, and injection attacks.
12. When using OAuth - Consider the draft Best-Current-Practice for OAuth
13. **Security / Privacy Event Reporting** - Consider legal obligations and ethical obligations to provide a means for Security and/or Privacy Event Reporting such as security vulnerability, or breach.

FHIR Security

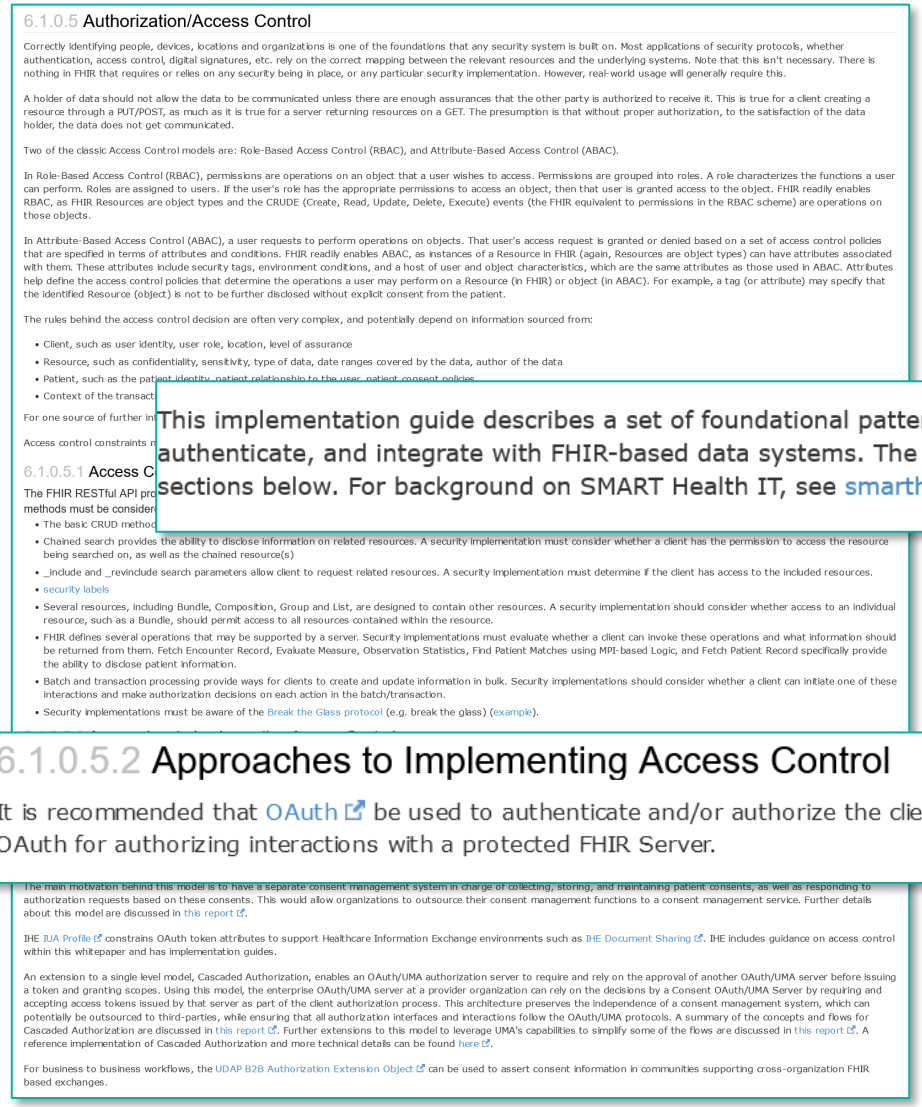
6.1.0 FHIR Security

Security Work Group	Maturity Level: 4	Standards Status: Trial Use

Fast Healthcare Interoperability Resources (FHIR) is not a security protocol, nor does it define any security related functionality. However, FHIR does define exchange protocols and content models that need to be used with various security protocols defined elsewhere. This section gathers all information about security in one section. A summary:

1. Time Keeping - all clocks should be synchronized using NTP/SNTP, and the design of the system should be robust against a system clock with the wrong value
2. Communications Security - all exchange of production data should be secured using TLS (e.g., https).
3. Authentication - Users/Clients must be authenticated. For web-centric, OAuth is recommended, consider use of HL7 SMART App Launch Implementation Guide where appropriate.
4. Authorization/Access Control - FHIR defines a Security Label infrastructure to support access control management. FHIR may also define a set of resources to administer access control management, but does not define any at present
5. Audit - FHIR defines provenance and audit event resources suitable for tracking the origins, authorship, history, status, and access of resources
6. Digital Signatures - FHIR includes several specifically reserved locations for digital signatures
7. Attachments - FHIR allows for binary resources and attachments. These have their own concerns
8. Labels - FHIR allows for set of security related tags that affect the way resources are handled
9. **Data Management Policies** - FHIR defines a set of capabilities to support data exchange. Not all the capabilities that FHIR enables may be appropriate or legal for use in some combinations of context and jurisdiction (e.g. HIPAA, GDPR). It is the responsibility of implementers to ensure that relevant regulations and other requirements are met.
10. Narrative - Care must be taken when displaying the narrative from FHIR resources
11. Input Validation - Validate all input received from other actors to assure the data is well formed and does not contain content that would cause unwanted system behavior. Testing ensures that the input is not susceptible to data input validation errors by using techniques such as fuzzing, invalid input attacks, and injection attacks.
12. When using OAuth - Consider the draft Best-Current-Practice for OAuth
13. **Security / Privacy Event Reporting** - Consider legal obligations and ethical obligations to provide a means for Security and/or Privacy Event Reporting such as security vulnerability, or breach.

FHIR Security



6.1.0.5 Authorization/Access Control

Correctly identifying people, devices, locations and organizations is one of the foundations that any security system is built on. Most applications of security protocols, whether authentication, access control, digital signatures, etc. rely on the correct mapping between the relevant resources and the underlying systems. Note that this isn't necessary. There is nothing in FHIR that requires or relies on any security being in place, or any particular security implementation. However, real-world usage will generally require this.

A holder of data should not allow the data to be communicated unless there are enough assurances that the other party is authorized to receive it. This is true for a client creating a resource through a PUT/POST, as much as it is true for a server returning resources on a GET. The presumption is that without proper authorization, to the satisfaction of the data holder, the data does not get communicated.

Two of the classic Access Control models are: Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC).

In Role-Based Access Control (RBAC), permissions are operations on an object that a user wishes to access. Permissions are grouped into roles. A role characterizes the functions a user can perform. Roles are assigned to users. If the user's role has the appropriate permissions to access an object, then that user is granted access to the object. FHIR readily enables RBAC, as FHIR Resources are object types and the CRUDE (Create, Read, Update, Delete, Execute) events (the FHIR equivalent to permissions in the RBAC scheme) are operations on those objects.

In Attribute-Based Access Control (ABAC), a user requests to perform operations on objects. That user's access request is granted or denied based on a set of access control policies that are specified in terms of attributes and conditions. FHIR readily enables ABAC, as instances of a Resource in FHIR (again, Resources are object types) can have attributes associated with them. These attributes include security tags, environment conditions, and a host of user and object characteristics, which are the same attributes as those used in ABAC. Attributes help define the access control policies that determine the operations a user may perform on a Resource (in FHIR) or object (in ABAC). For example, a tag (or attribute) may specify that the identified Resource (object) is not to be further disclosed without explicit consent from the patient.

The rules behind the access control decision are often very complex, and potentially depend on information sourced from:

- Client, such as user identity, user role, location, level of assurance
- Resource, such as confidentiality, sensitivity, type of data, date ranges covered by the data, author of the data
- Patient, such as the patient identity, patient relationship to the user, patient consent policies
- Context of the transact

For one source of further in

Access control constraints r

6.1.0.5.1 Access C

The FHIR RESTful API pro
methods must be consider

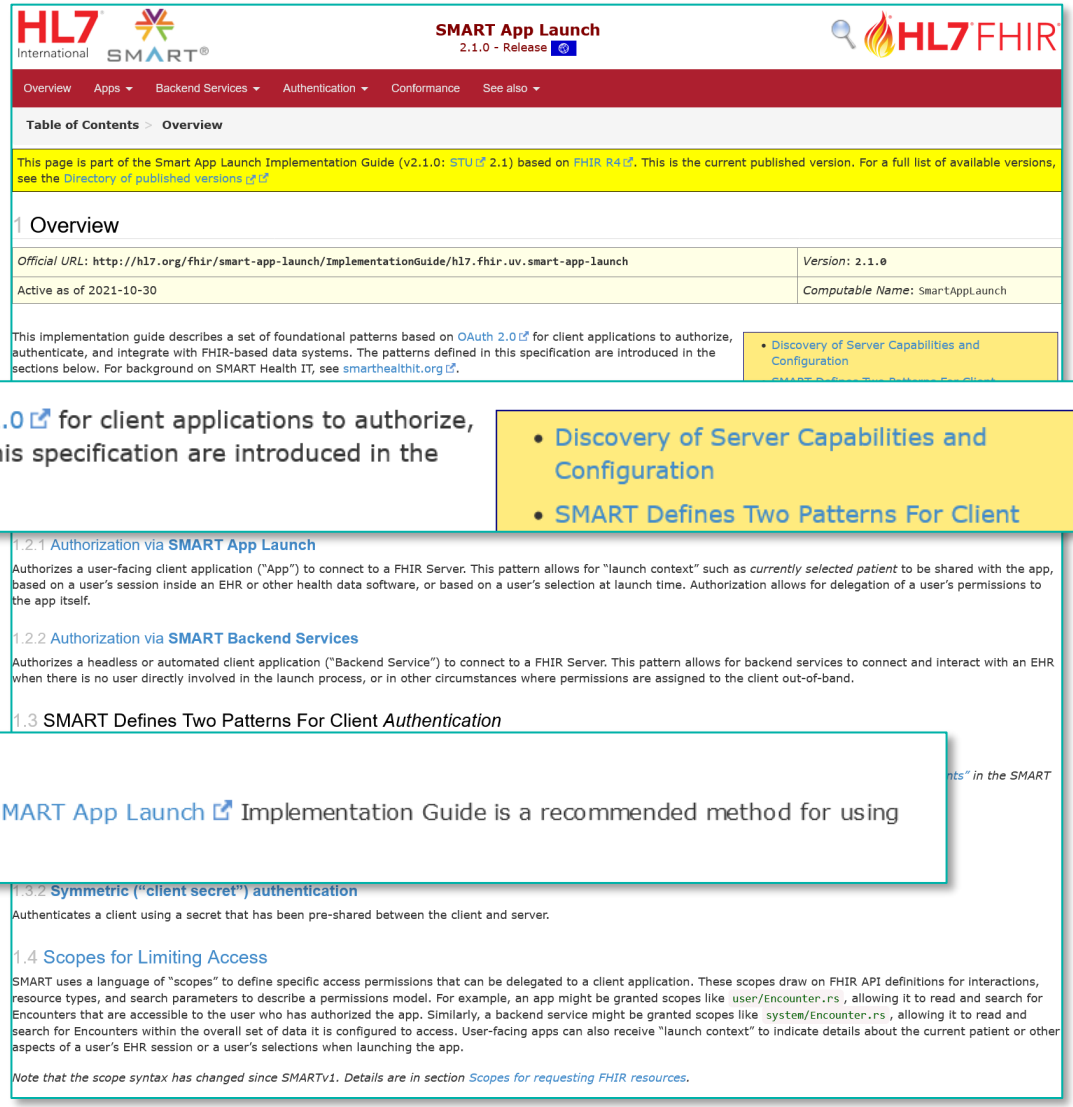
- The basic CRUD metho
- Chained search provides the ability to disclose information on related resources. A security implementation must consider whether a client has the permission to access the resource being searched on, as well as the chained resource(s)
- _include and _revinclude search parameters allow client to request related resources. A security implementation must determine if the client has access to the included resources.
- security labels
- Several resources, including Bundle, Composition, Group and List, are designed to contain other resources. A security implementation should consider whether access to an individual resource, such as a Bundle, should permit access to all resources contained within the resource.
- FHIR defines several operations that may be supported by a server. Security implementations must evaluate whether a client can invoke these operations and what information should be returned from them. Fetch Encounter Record, Evaluate Measure, Observation Statistics, Find Patient Matches using MPI-based Logic, and Fetch Patient Record specifically provide the ability to disclose patient information.
- Batch and transaction processing provide ways for clients to create and update information in bulk. Security implementations should consider whether a client can initiate one of these interactions and make authorization decisions on each action in the batch/transaction.
- Security implementations must be aware of the Break the Glass protocol (e.g. break the glass) (example).

The main motivation behind this model is to have a separate consent management system in charge of collecting, storing, and maintaining patient consents, as well as responding to authorization requests based on these consents. This would allow organizations to outsource their consent management functions to a consent management service. Further details about this model are discussed in this report.

IHE IUA Profile constrains OAuth token attributes to support Healthcare Information Exchange environments such as IHE Document Sharing. IHE includes guidance on access control within this whitepaper and has implementation guides.

An extension to a single level model, Cascaded Authorization, enables an OAuth/UMA authorization server to require and rely on the approval of another OAuth/UMA server before issuing a token and granting scopes. Using this model, the enterprise OAuth/UMA server at a provider organization can rely on the decisions by a Consent OAuth/UMA Server by requiring and accepting access tokens issued by that server as part of the client authorization process. This architecture preserves the independence of a consent management system, which can potentially be outsourced to third-parties, while ensuring that all authorization interfaces and interactions follow the OAuth/UMA protocols. A summary of the concepts and flows for Cascaded Authorization are discussed in this report. Further extensions to this model to leverage UMA's capabilities to simplify some of the flows are discussed in this report. A reference implementation of Cascaded Authorization and more technical details can be found here.

For business to business workflows, the UDAP B2B Authorization Extension Object can be used to assert consent information in communities supporting cross-organization FHIR based exchanges.



HL7 International SMART®

SMART App Launch
2.1.0 - Release

HL7 FHIR

Overview | Apps | Backend Services | Authentication | Conformance | See also

Table of Contents > **Overview**

This page is part of the Smart App Launch Implementation Guide (v2.1.0: STU 2.1) based on FHIR R4. This is the current published version. For a full list of available versions, see the Directory of published versions

1 Overview

Official URL: http://hl7.org/fhir/smart-app-launch/ImplementationGuide/hl7.fhir.uv.smart-app-launch	*Version*: 2.1.0
Active as of 2021-10-30	*Computable Name*: SmartAppLaunch

This implementation guide describes a set of foundational patterns based on OAuth 2.0 for client applications to authorize, authenticate, and integrate with FHIR-based data systems. The patterns defined in this specification are introduced in the sections below. For background on SMART Health IT, see smarthealthit.org.

- Discovery of Server Capabilities and Configuration

1.2.1 Authorization via **SMART App Launch**

Authorizes a user-facing client application ("App") to connect to a FHIR Server. This pattern allows for "launch context" such as *currently selected patient* to be shared with the app, based on a user's session inside an EHR or other health data software, or based on a user's selection at launch time. Authorization allows for delegation of a user's permissions to the app itself.

1.2.2 Authorization via **SMART Backend Services**

Authorizes a headless or automated client application ("Backend Service") to connect to a FHIR Server. This pattern allows for backend services to connect and interact with an EHR when there is no user directly involved in the launch process, or in other circumstances where permissions are assigned to the client out-of-band.

1.3 SMART Defines Two Patterns For Client *Authentication*

1.3.2 **Symmetric ("client secret") authentication**

Authenticates a client using a secret that has been pre-shared between the client and server.

1.4 Scopes for Limiting Access

SMART uses a language of "scopes" to define specific access permissions that can be delegated to a client application. These scopes draw on FHIR API definitions for interactions, resource types, and search parameters to describe a permissions model. For example, an app might be granted scopes like `user/Encounter.rs`, allowing it to read and search for Encounters that are accessible to the user who has authorized the app. Similarly, a backend service might be granted scopes like `system/Encounter.rs`, allowing it to read and search for Encounters within the overall set of data it is configured to access. User-facing apps can also receive "launch context" to indicate details about the current patient or other aspects of a user's EHR session or a user's selections when launching the app.

Note that the scope syntax has changed since SMARTv1. Details are in section Scopes for requesting FHIR resources.

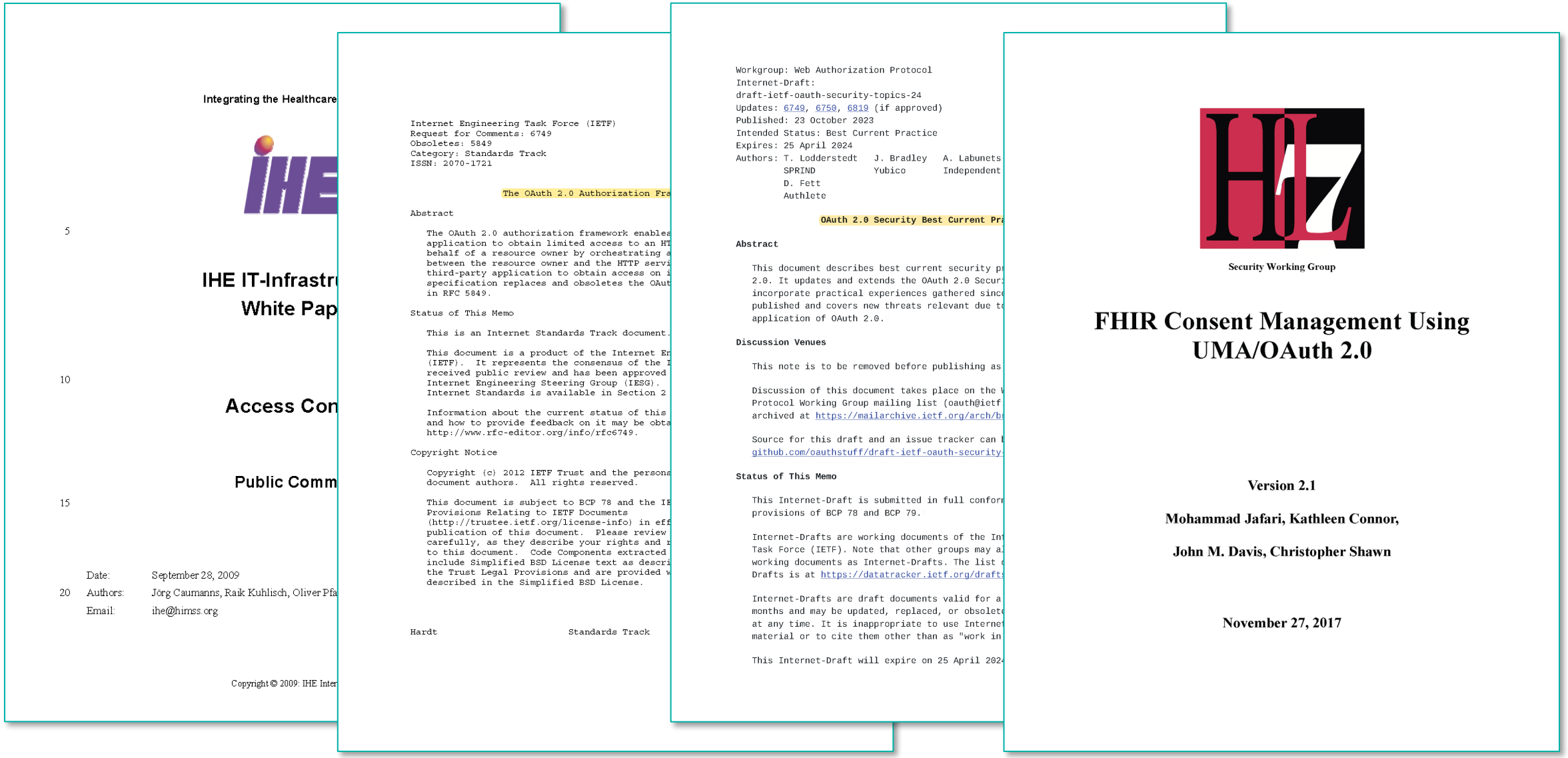
This implementation guide describes a set of foundational patterns based on OAuth 2.0 for client applications to authorize, authenticate, and integrate with FHIR-based data systems. The patterns defined in this specification are introduced in the sections below. For background on SMART Health IT, see smarthealthit.org.

- Discovery of Server Capabilities and Configuration
- SMART Defines Two Patterns For Client

6.1.0.5.2 Approaches to Implementing Access Control

It is recommended that OAuth be used to authenticate and/or authorize the client and user. The HL7 SMART App Launch Implementation Guide is a recommended method for using OAuth for authorizing interactions with a protected FHIR Server.

FHIR Security



Integrating the Healthcare

IHE IT-Infrastr
White Pap

Access Con

Public Comm

Date: September 28, 2009
Authors: Jörg Caumanns, Raik Kuhlisch, Oliver Pfa
Email: ihe@himss.org

Copyright © 2009: IHE Inter

Internet Engineering Task Force (IETF)
Request for Comments: 6749
Obsoletes: 5849
Category: Standards Track
ISSN: 2070-1721

The OAuth 2.0 Authorization Fra

Abstract

The OAuth 2.0 authorization framework enables
application to obtain limited access to an HT
behalf of a resource owner by orchestrating a
between the resource owner and the HTTP servi
third-party application to obtain access on i
specification replaces and obsoletes the OAut
in RFC 5849.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet En
(IETF). It represents the consensus of the I
received public review and has been approved
Internet Engineering Steering Group (IESG).
Internet Standards is available in Section 2

Information about the current status of this
and how to provide feedback on it may be obta
http://www.rfc-editor.org/info/rfc6749.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons
document authors. All rights reserved.

This document is subject to BCP 78 and the IE
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in eff
publication of this document. Please review
carefully, as they describe your rights and r
to this document. Code Components extracted
include Simplified BSD License text as descri
the Trust Legal Provisions and are provided w
described in the Simplified BSD License.

Hardt Standards Track

Workgroup: Web Authorization Protocol
Internet-Draft:
draft-ietf-oauth-security-topics-24
Updates: 6749, 6750, 6819 (if approved)
Published: 23 October 2023
Intended Status: Best Current Practice
Expires: 25 April 2024
Authors: T. Lodderstedt J. Bradley A. Labunets
SPRIND Yubico Independent
D. Fett
Authlete

OAuth 2.0 Security Best Current Pra

Abstract

This document describes best current security pr
2.0. It updates and extends the OAuth 2.0 Securi
incorporate practical experiences gathered since
published and covers new threats relevant due to
application of OAuth 2.0.

Discussion Venues

This note is to be removed before publishing as

Discussion of this document takes place on the W
Protocol Working Group mailing list (oauth@ietf
archived at https://mailarchive.ietf.org/arch/br

Source for this draft and an issue tracker can b
github.com/oauthstuff/draft-ietf-oauth-security-

Status of This Memo

This Internet-Draft is submitted in full conform
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Int
Task Force (IETF). Note that other groups may al
working documents as Internet-Drafts. The list o
Drafts is at https://datatracker.ietf.org/drafts

Internet-Drafts are draft documents valid for a
months and may be updated, replaced, or obsolete
at any time. It is inappropriate to use Internet
material or to cite them other than as "work in

This Internet-Draft will expire on 25 April 2024

Security Working Group

FHIR Consent Management Using UMA/OAuth 2.0

Version 2.1

Mohammad Jafari, Kathleen Connor,

John M. Davis, Christopher Shawn

November 27, 2017

FHIR Security

- OAuth + Scopes
- SMART on FHIR

OAuth2

- The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.

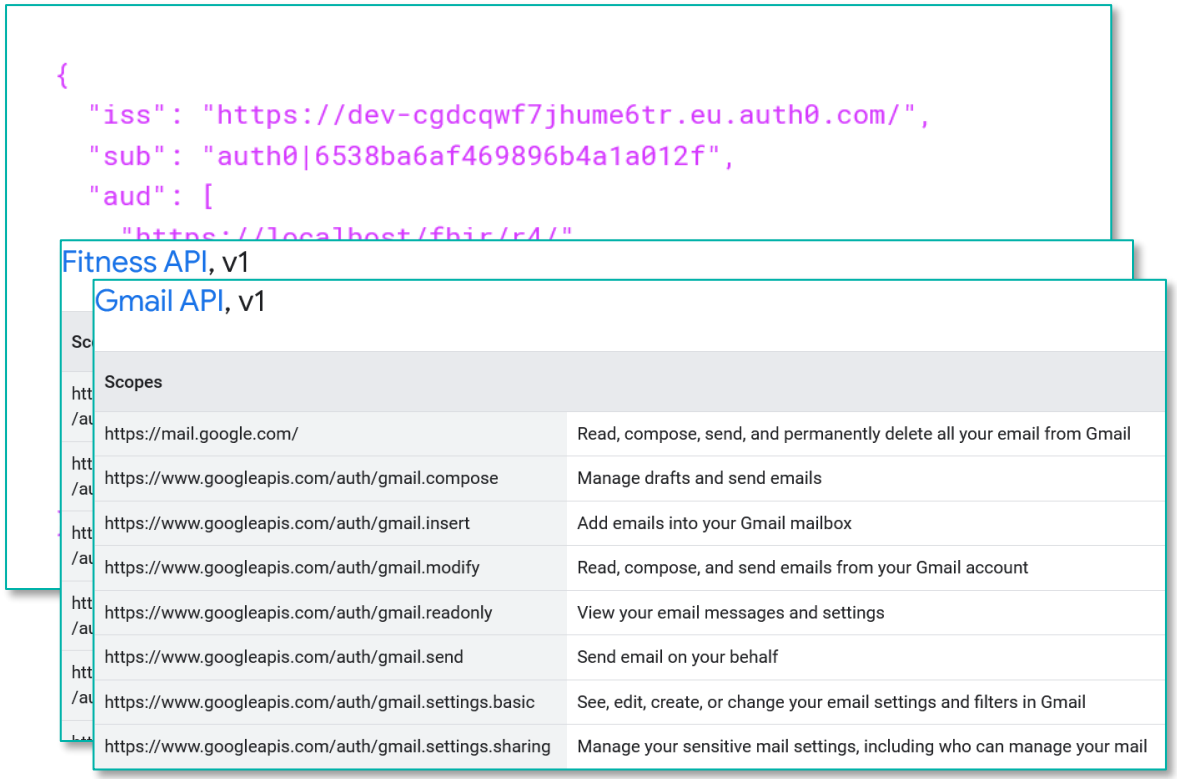
- OAuth 2.0 steht für „Open Authorization“ und ist ein Standard, mithilfe dessen eine Website oder Anwendung auf Ressourcen zugreifen kann, die von anderen Web-Apps für einen Nutzer gehostet werden.

https://auth0.com/de/intro-to-iam/what-is-oauth-2

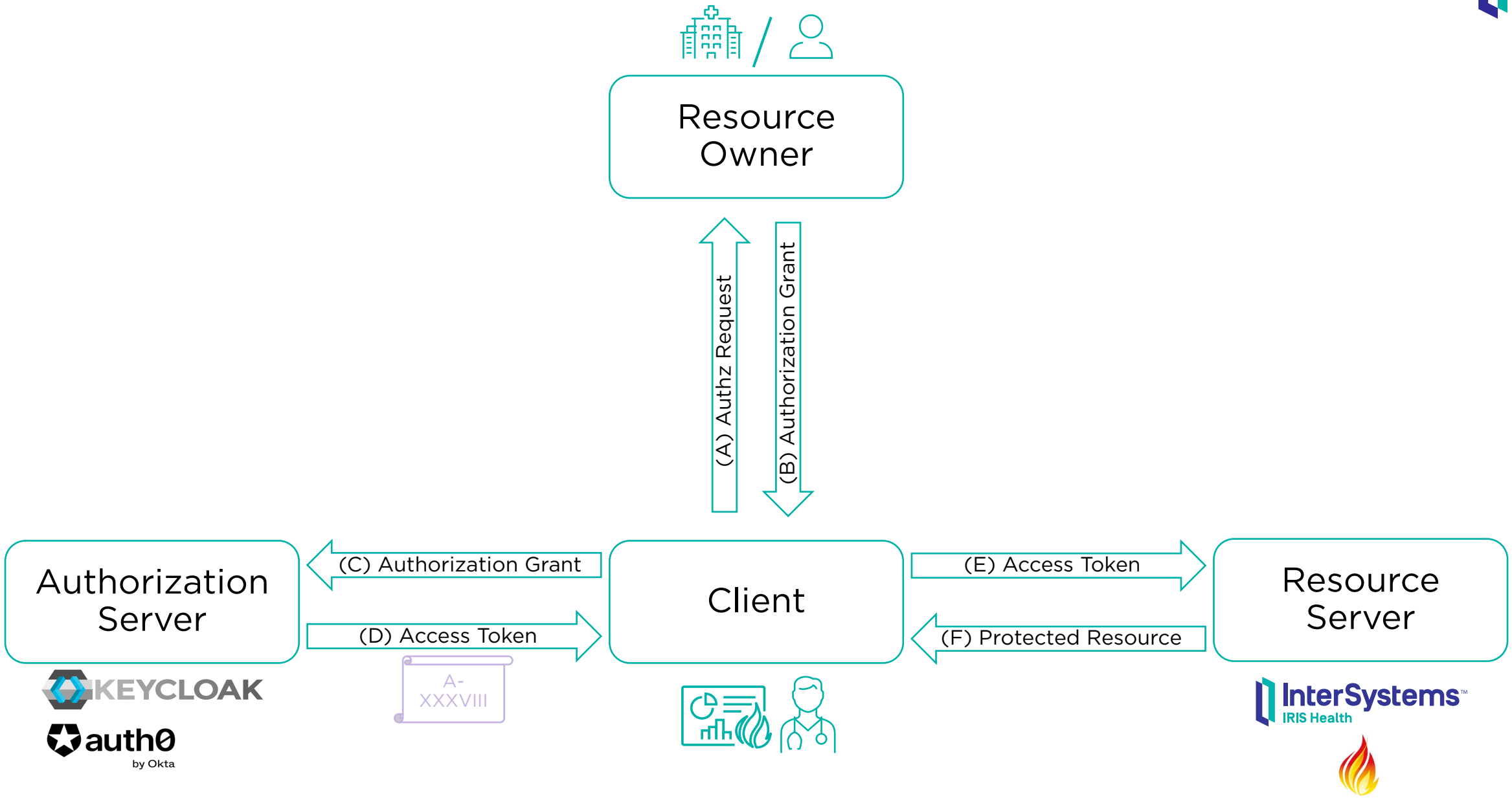
OAuth2

- Resource Owner
- Resource Server
- Client
- Authorization Server

- Access Token
- Scopes



OAuth2


Resource Owner
(A) Authz Request
(B) Authorization Grant
Authorization Server
(C) Authorization Grant
(D) Access Token
A-XXXVIII
Client
(E) Access Token
(F) Protected Resource
Resource Server
KEYCLOAK
auth0 by Okta
InterSystems IRIS Health

OAuth2

You

Was ist OAuth 2.0?

ChatGPT

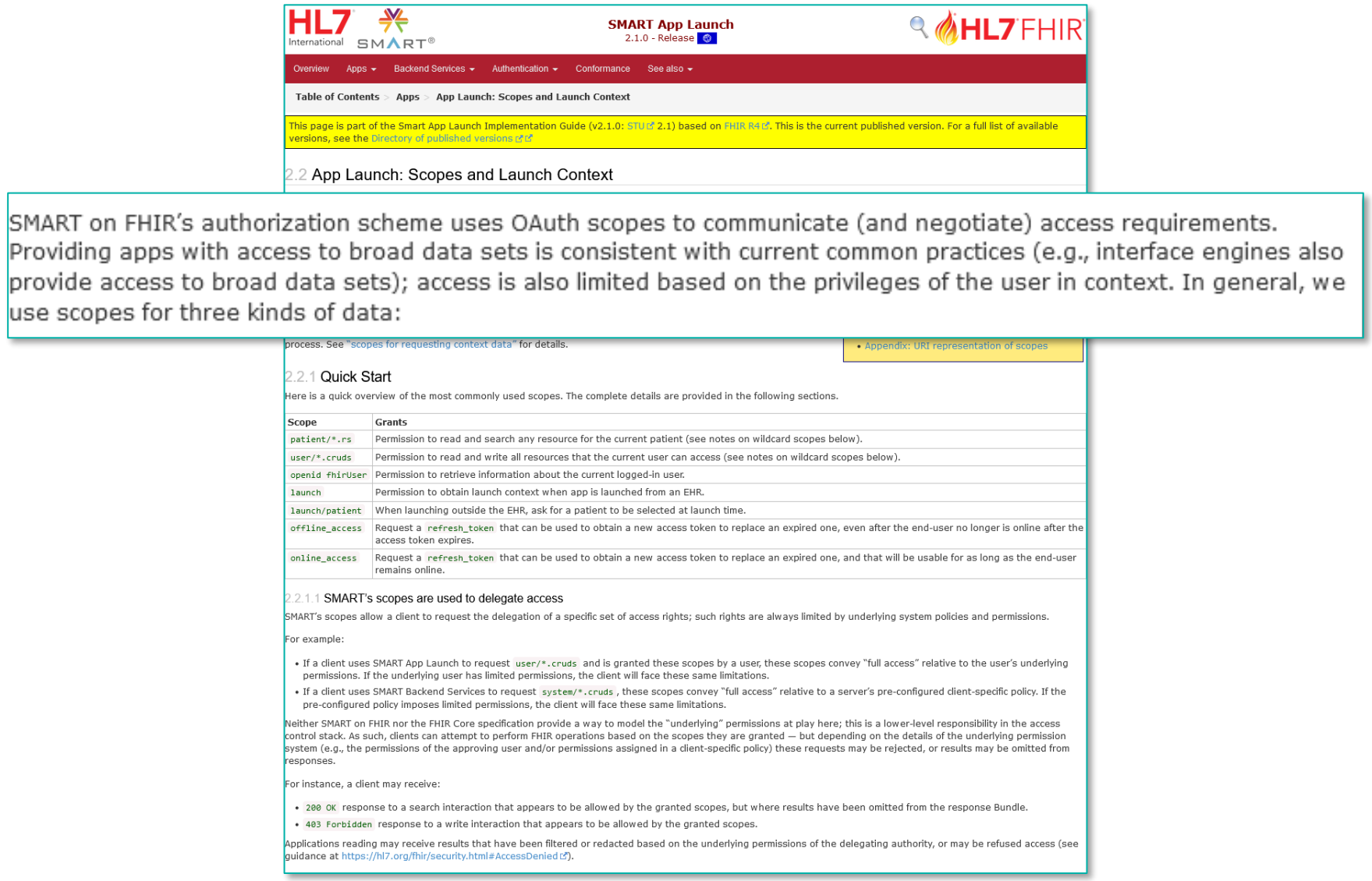
OAuth 2.0 ist ein Autorisierungsprotokoll, das es ermöglicht, den Zugriff auf Ressourcen eines Benutzers durch Dritte zu steuern, ohne die sensiblen Anmeldedaten weiterzugeben. Es wurde entwickelt, um die Sicherheit bei der Authentifizierung und Autorisierung von Anwendungen zu verbessern, die auf Benutzerdaten zugreifen müssen, ohne diese Daten direkt preiszugeben.

Im Wesentlichen erlaubt OAuth 2.0 einer Anwendung (dem sogenannten "Client") den Zugriff auf geschützte Ressourcen, die sich auf einem anderen Dienst oder Server befinden (dem "Authorization Server"), im Namen des Benutzers, ohne dass die Anwendung die tatsächlichen Anmeldeinformationen des Benutzers kennt oder speichert.

OAuth 2.0 arbeitet mit verschiedenen Autorisierungsflüssen, die je nach Anwendungsfall variieren können. Hier sind einige Schlüsselelemente von OAuth 2.0:

1. **Autorisierung:** Der Prozess, durch den ein Benutzer einer Anwendung oder einem Dienst Zugriff auf seine Ressourcen gewährt, ohne sensible Anmeldeinformationen preiszugeben.
2. **Autorisierungsserver:** Der Server, der die Autorisierung des Benutzers durchführt und die Zugriffstoken für den Zugriff auf Ressourcen ausstellt.

SMART App Launch



HL7 International SMART®

SMART App Launch
2.1.0 - Release

HL7 FHIR

Overview | Apps | Backend Services | Authentication | Conformance | See also

Table of Contents > Apps > App Launch: Scopes and Launch Context

This page is part of the Smart App Launch Implementation Guide (v2.1.0: STU 2.1) based on FHIR R4. This is the current published version. For a full list of available versions, see the Directory of published versions

2.2 App Launch: Scopes and Launch Context

SMART on FHIR's authorization scheme uses OAuth scopes to communicate (and negotiate) access requirements. Providing apps with access to broad data sets is consistent with current common practices (e.g., interface engines also provide access to broad data sets); access is also limited based on the privileges of the user in context. In general, we use scopes for three kinds of data:

process. See "scopes for requesting context data" for details.

- Appendix: URI representation of scopes

2.2.1 Quick Start

Here is a quick overview of the most commonly used scopes. The complete details are provided in the following sections.

Scope	Grants
`patient/*.rs`	Permission to read and search any resource for the current patient (see notes on wildcard scopes below).
`user/*.cruds`	Permission to read and write all resources that the current user can access (see notes on wildcard scopes below).
`openid fhirUser`	Permission to retrieve information about the current logged-in user.
`launch`	Permission to obtain launch context when app is launched from an EHR.
`launch/patient`	When launching outside the EHR, ask for a patient to be selected at launch time.
`offline_access`	Request a `refresh_token` that can be used to obtain a new access token to replace an expired one, even after the end-user no longer is online after the access token expires.
`online_access`	Request a `refresh_token` that can be used to obtain a new access token to replace an expired one, and that will be usable for as long as the end-user remains online.

2.2.1.1 SMART's scopes are used to delegate access

SMART's scopes allow a client to request the delegation of a specific set of access rights; such rights are always limited by underlying system policies and permissions.

For example:

- If a client uses SMART App Launch to request `user/*.cruds` and is granted these scopes by a user, these scopes convey "full access" relative to the user's underlying permissions. If the underlying user has limited permissions, the client will face these same limitations.
- If a client uses SMART Backend Services to request `system/*.cruds`, these scopes convey "full access" relative to a server's pre-configured client-specific policy. If the pre-configured policy imposes limited permissions, the client will face these same limitations.

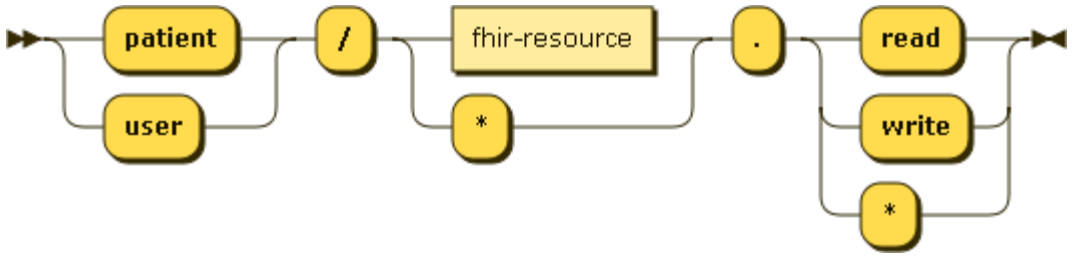
Neither SMART on FHIR nor the FHIR Core specification provide a way to model the "underlying" permissions at play here; this is a lower-level responsibility in the access control stack. As such, clients can attempt to perform FHIR operations based on the scopes they are granted — but depending on the details of the underlying permission system (e.g., the permissions of the approving user and/or permissions assigned in a client-specific policy) these requests may be rejected, or results may be omitted from responses.

For instance, a client may receive:

- `200 OK` response to a search interaction that appears to be allowed by the granted scopes, but where results have been omitted from the response Bundle.
- `403 Forbidden` response to a write interaction that appears to be allowed by the granted scopes.

Applications reading may receive results that have been filtered or redacted based on the underlying permissions of the delegating authority, or may be refused access (see guidance at https://hl7.org/fhir/security.html#AccessDenied).

Scopes



Scopes

user/Patient.read

- „patient“ – Zugriff auf spezifische Daten eines (Patienten-)Records
- „user“ – Zugriff auf alle einem Benutzer zugänglichen Daten

Scopes

user/Patient.read

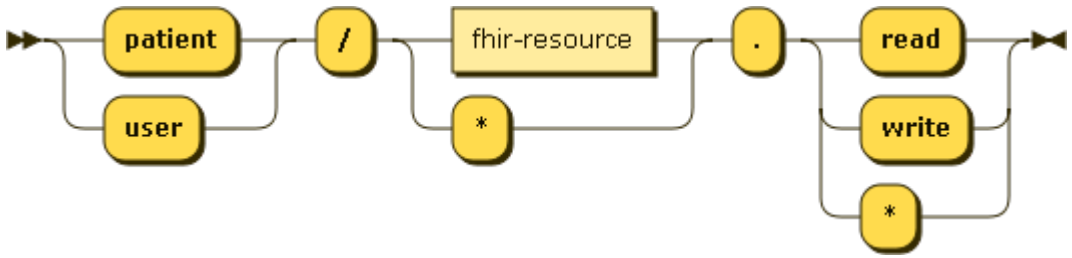
- resourceType / Name der Resource auf die zugegriffen werden darf
- Wildcard für *alle* Resourcen: *

Scopes

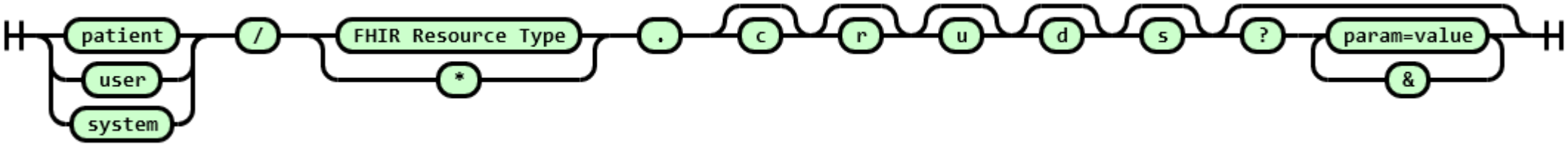
user/Patient.read

- „read“ – erlaube Interaktion „lesen“
- „write“ – erlaube Interaktion „schreiben“
- * - erlaube alle Interaktionen, inkl. „lesen“ und „schreiben“

Scopes



SMART App Launch 1.0

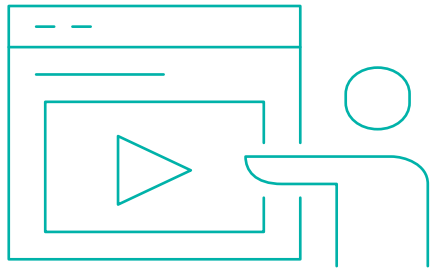


SMART App Launch 2.0

Demo II

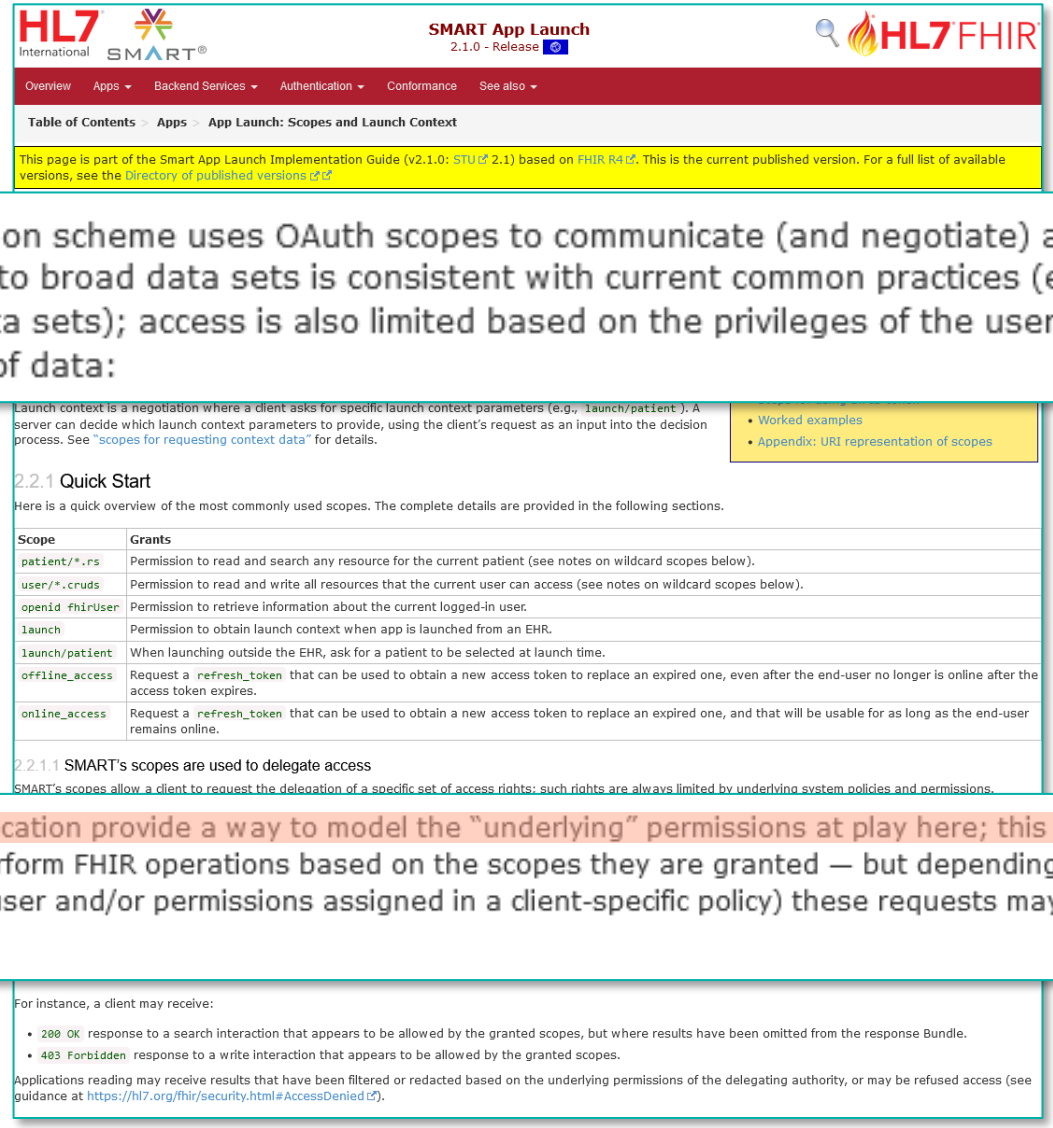
- SSL Config anlegen
  - System > Security Management > SSL/TLS Configurations > New SSL/TLS Configuration

- OAuth2 Client (und Server) konfigurieren
  - System > Security Management > OAuth 2.0 Client
    - Create Server Description
    - Create Client Description



- Auth2 Client Name in FHIR Server Configuration eintragen

SMART App Launch



HL7 International SMART®

SMART App Launch
2.1.0 - Release

HL7 FHIR

Overview | Apps | Backend Services | Authentication | Conformance | See also

Table of Contents > Apps > App Launch: Scopes and Launch Context

This page is part of the Smart App Launch Implementation Guide (v2.1.0: STU 2.1) based on FHIR R4. This is the current published version. For a full list of available versions, see the Directory of published versions

SMART on FHIR's authorization scheme uses OAuth scopes to communicate (and negotiate) access requirements. Providing apps with access to broad data sets is consistent with current common practices (e.g., interface engines also provide access to broad data sets); access is also limited based on the privileges of the user in context. In general, we use scopes for three kinds of data:

Launch context is a negotiation where a client asks for specific launch context parameters (e.g., `launch/patient`). A server can decide which launch context parameters to provide, using the client's request as an input into the decision process. See "scopes for requesting context data" for details.

- Worked examples
- Appendix: URI representation of scopes

2.2.1 Quick Start

Here is a quick overview of the most commonly used scopes. The complete details are provided in the following sections.

Scope	Grants
`patient/*.rs`	Permission to read and search any resource for the current patient (see notes on wildcard scopes below).
`user/*.cruds`	Permission to read and write all resources that the current user can access (see notes on wildcard scopes below).
`openid fhirUser`	Permission to retrieve information about the current logged-in user.
`launch`	Permission to obtain launch context when app is launched from an EHR.
`launch/patient`	When launching outside the EHR, ask for a patient to be selected at launch time.
`offline_access`	Request a `refresh_token` that can be used to obtain a new access token to replace an expired one, even after the end-user no longer is online after the access token expires.
`online_access`	Request a `refresh_token` that can be used to obtain a new access token to replace an expired one, and that will be usable for as long as the end-user remains online.

2.2.1.1 SMART's scopes are used to delegate access

SMART's scopes allow a client to request the delegation of a specific set of access rights; such rights are always limited by underlying system policies and permissions.

Neither SMART on FHIR nor the FHIR Core specification provide a way to model the "underlying" permissions at play here; this is a lower-level responsibility in the access control stack. As such, clients can attempt to perform FHIR operations based on the scopes they are granted — but depending on the details of the underlying permission system (e.g., the permissions of the approving user and/or permissions assigned in a client-specific policy) these requests may be rejected, or results may be omitted from responses.

For instance, a client may receive:

- `200 OK` response to a search interaction that appears to be allowed by the granted scopes, but where results have been omitted from the response Bundle.
- `403 Forbidden` response to a write interaction that appears to be allowed by the granted scopes.

Applications reading may receive results that have been filtered or redacted based on the underlying permissions of the delegating authority, or may be refused access (see guidance at https://hl7.org/fhir/security.html#AccessDenied).

Scopes

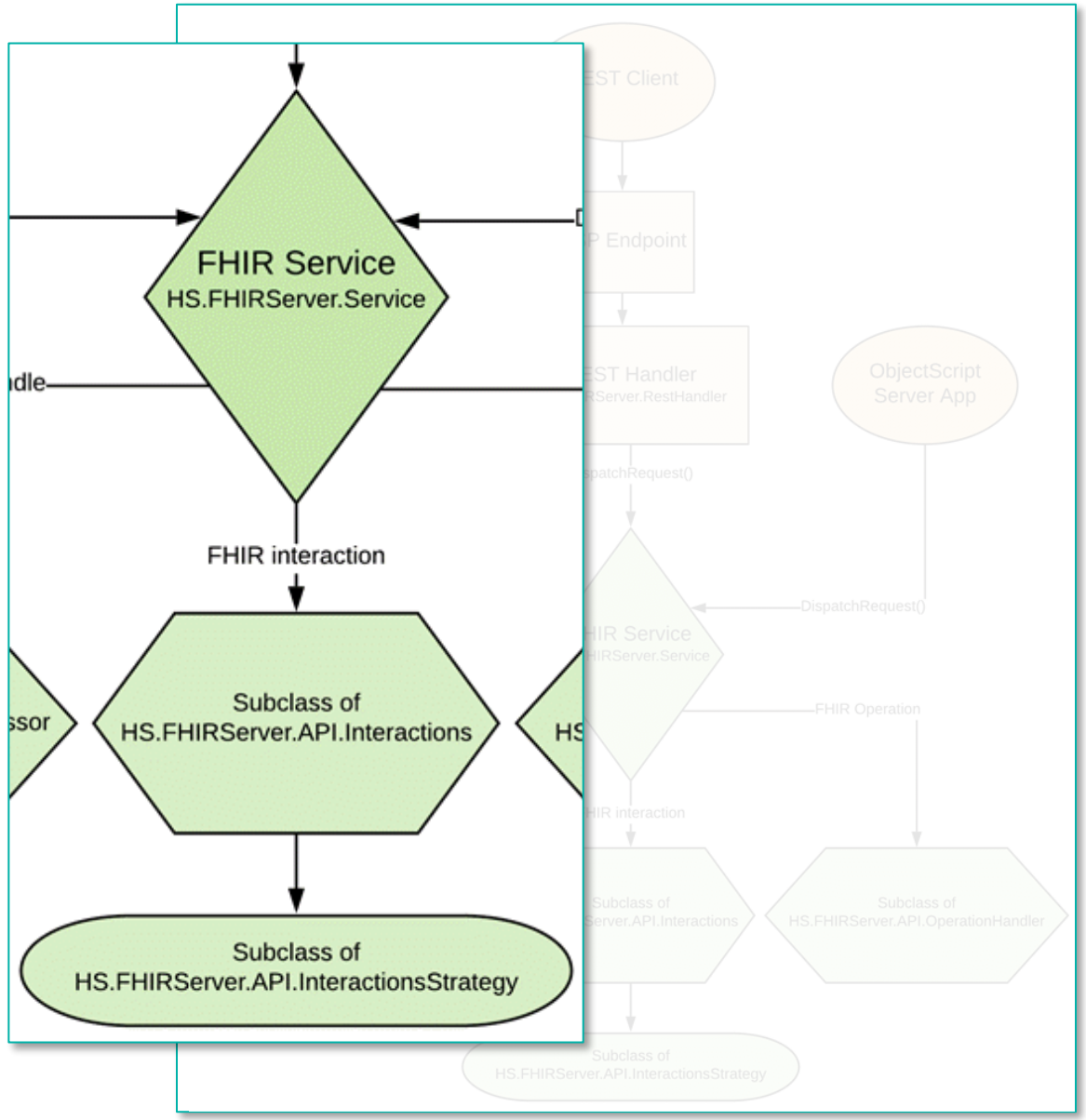
- Benutzer, der eine Ressource anfragt, ist nicht der eigentliche Eigentümer dieser Ressource
- Nicht granular genug, um spezifische Operationen oder Zugriffe auf FHIR-Ressourcen zu steuern
- Komplexe Anforderungen an den Datenschutz und die Zugriffskontrolle, dynamische Zugriffskontrolle
  - Rollenbasierte Zugriffskontrolle (RBAC) und Attributbasierte Zugriffskontrolle (ABAC)
  - Patienteneinwilligung
- Authorization Server hat nicht alle Informationen über Resourcen und „Consent“

https://auth0.com/blog/on-the-nature-of-oauth2-scopes/

Richtig Behüten?!

- OAuth + Scopes
- SMART on FHIR
- Lower level access control stack ?

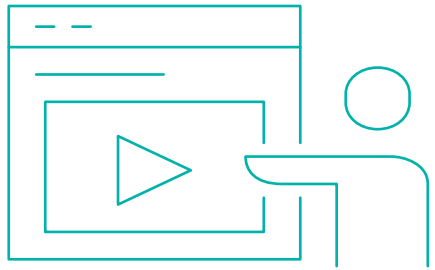
Richtig Behüten?!



Demo III

- Customizing a FHIR Server
  - HS.FHIRServer.Storage.Json.Interactions
  - HS.FHIRServer.Storage.Json.InteractionsStrategy
  - HS.FHIRServer.Storage.Json.RepoManager
  - HS.FHIRServer.Util.OAuth2Token

- Lookup Tabelle → *Consent*



Fazit - Richtig Behüten!

- OAuth + Scopes
- SMART on FHIR
- Lower level access control stack

Fazit - Richtig Behüten!

- OAuth + Scopes
- SMART on FHIR
- RBAC + ABAC
- Patienteneinwilligungen (z.B. FHIR Consent) und weitere Erlaubnistatbestände
- Data Masking
- ...

Fragen

InterSystems®
DACH SYMPOSIUM 2023

Vielen Dank