HEART Overview

• Why HEART?
• What is HEART?
• Value of HEART
Healthcare Challenges/Gaps (1 of 2)

- Needs to see a specialist outside of her healthcare system
- Share health data with a spouse or adult child
- Share health data with a research organization
- A new provider does not have access to a patient’s record
Healthcare Challenges/Gaps (2 of 2)

- Ability to share relevant device data
- Needs to keep some aspects of their data private
- Patients travel or relocate seasonally
- Decision making by an advocate or medical power of attorney
- Emergency responder access
Pain Points – the human perspective

- Frustration
- Waste of time
- Negative impact of care
Why HEART?

- Created to address these challenges and gaps
- Enables the patient to safely share her health records with users of her choice, in an interoperable way that respects and honors patient security and privacy
- Enables patient directed sharing of their clinical data
HEART (HEAIlth Relationship Trust) is a set of profiles that enable patients to control how, when, and with whom their clinical data is shared.
What is HEART?

- Leverages existing open standards
  - FHIR / SMART on FHIR
  - OAuth 2
  - OpenID Connect
  - User Managed Access

- Best practice security standards

- Adds additional security features

- Gives patients control over how their data is shared

- Defines interoperable process for patient directed clinical data sharing
Building the Bridge to Trust
Providers can access patient health data within their health care system

Using FHIR, innovative clinical functionality can be integrated with clinical data and made available to providers, all within their health care system
Where Is the Industry Now?

- The industry is rolling out systems where the patient can safely access her health records from her provider’s EMR/portal
- This enables patient-focused innovations
Empower the patient to safely share her health records, with users of her choice, in an interoperable way that respects and honor patient security and privacy.
Clinical data needs to be exchanged across health care systems
Background: Terminology

Users authenticate within one physical office

This is a ‘Narrow Ecosystem’
In a larger integrated facility, data access from multiple resources may authenticate with one server. This is still a ‘Narrow Ecosystem’
Patients need to exchange clinical data across many health care systems.

Now we have a ‘Wide Ecosystem’
1. HEART enables patient directed sharing across a wide ecosystem
Patient Directed Sharing

1. Gives patients control over how their data is shared
2. Electronic consents define patient’s sharing wishes
3. Authorization is based on patient-specified policy
4. Enables multi-party sharing
5. Authorization is provided asynchronously
6. The patient makes the decision on who has access to their data
Patient Directed Sharing
Patient Directed Sharing

- The general population is becoming more aware of cybersecurity and privacy concerns
- Greater awareness of privacy concerns
- Realization of privacy rights and options
- Increased patient demand to exercise those rights.
HEART Overview

1. HEART enables patient directed sharing across a wide ecosystem

2. The patient controls who has access to their data (Patient Directed)
HEART works in conjunction with Best Practice Security Standards

- We want to know that our patient Alice is really Alice
  - The patient is identified through identity assurance
  - The patient is authenticated through trusted authentication systems
- We want to know that the user requesting information is who he says he is
  - The user is identified through identity assurance
  - The user is authenticated through trusted authentication systems
True secure delegation; no password sharing

Foster compliance through standards

KEEP CALM AND PREPARE FOR THE GDPR
HEART Overview

1. HEART enables patient directed **sharing** across a wide ecosystem

2. The patient controls who has access to their data (Patient Directed)

3. HEART works in conjunction with Best Practice Security Standards
More Granular Data Management

Confidential App
is requesting permission to access:
- Access and change your email contacts

Allow Access  No thanks
More Granular Data Management

• Which Resource?
• What Scopes?
• What sensitive data?
• The options vary per data source
More Granular Data Management

Example A

- A portal supports reading a patient’s common clinical data set
- That same portal may allow users to both read and update a care plan
- The patient may choose to authorize a new specialist to read some subset of her clinical data set and update her care plan
More Granular Data Management

Example B
Consent 2 Share

Create Consent

I, Sally Share, hereby authorize...

The following individual or organization

JENKIN ROSEMARY

To disclose my information to

MARYLAND CVS PHARMACY, LLC.

Medical Information

Select how you would like to share your medical information.

- SHARE ALL information in my medical record.
- SHARE my medical record WITH EXCEPTION of specific information.

Purpose of Use

Choose for what purposes your medical information may be used.

- SHARE my medical record ONLY for the selected purposes of use.

Mental health information
Drug use information

Privacy Settings

Select the medical information that you DO NOT wish to share.

- Federal Categories
- Drug use information
- Alcohol use and Alcoholism information
- State Categories
- Mental health information
- HIV/AIDS information
- Communicable disease information
- Sexuality and reproductive health information

Cancel
Save changes
HEART Overview

1. HEART enables patient directed sharing across a wide ecosystem
2. The patient controls who has access to their data
3. HEART works in conjunction with Best Practice Security Standards
4. HEART provides more granular management over protected resources
Leverages Open Standards

- Leverages existing open standards
- FHIR/ SMART on FHIR
- OAuth 2
- OpenID Connect
- User Managed Access
HEAIlth Relationship Trust

- OAuth2
- FHIR
- OpenID Connect (OIDC)
- Electronic Consent
- User Managed Access (UMA)
- SMART on FHIR
HEART Overview

1. HEART enables patient directed **sharing** across a wide ecosystem
2. The patient controls who has access to their data
3. HEART works in conjunction with Best Practice Security Standards
4. HEART provides more granular management over protected resources
5. Leverages existing open standards
Ease of Use

- HEART has addressed pesky use case challenges
- The more difficult issues are addressed by HEART
- The patient interface is easy to use
- The provider interface is easy to use
- As this new paradigm is adopted and trust increases, sharing private clinical data will become seamless
  - Ultimately this improves health and reduces the cost of healthcare.
Ease of Use

- Patient Alice creates a policy to share with Dr. Erica, she selects her sharing preferences, and presses SHARE.

- Patient sharing is easy!
Ease of Use

Provider wishes to view clinical data

Provider usage is also easy. The power is in what happens behind the scenes!
1. HEART enables patient directed **sharing** across a wide ecosystem
2. The patient controls who has access to their data
3. HEART works in conjunction with Best Practice Security Standards
4. HEART provides more granular management over protected resources
5. Leverages existing open standards
6. HEART Patient and Provider clients are intended to be EASY to use
HEART Implementations

- EMR Direct/HealthToGo
- HIE of One/Trustee
- HealthyMePHR/ShareMedData

Reference implementers drafts at openid.net/wg/heart
Latest specs approved March 12, 2019
HEART Use Cases - #1 Portal
HEART Use Cases - #2 Shared from EMR (1 of 2)
HEART Use Cases - #2 Shared from EMR (2 of 2)

You granted access to your data by:
- Dr. Henry Seven
- Jeremy Bates
- Rebecca Larson

Actions:
- Stop Sharing

Share data with another (enter OpenID, email, or Direct Address):

By clicking "Accept & Grant" below, you attest that you have the legal authority to grant access to this information and you authorize the bearer of the digital identity above to obtain the appropriate password or alternative authorization to authenticate.

Transmit Summary
Accept & Grant

Terms of Use Privacy Policy
© 2019 EMR Direct
HEART Use Cases - #3 Device Data Sharing

1. Certified device identity
2. Strongly authenticated patient identity
3. Patient/device association
4. Consented device data sharing with clinicians and others
5. Strongly authenticated third-party identity

Standards:
- FIDO Alliance
- UMA
- Health Information Technicians

Cryptographic auditability
Why is HEART good for organizations?

- Leverages existing standards
- Empowers the patient
- Delivers patient-mediated sharing to a wide ecosystem
- Meets goal of seamless clinical data availability
Benefits to Providers

- Accurate data
- Adequate data
- Innovation
Benefits to Patients

- Control over access
- Transparency over who has accessed
- Empowerment
- Ability to share and consult
- Better Care
Call to Action

- [openid.net/wg/heart/](openid.net/wg/heart/)
- Refer to the HEART profiles and use cases for more information
- Reach out to the HEART WG to learn more and get involved
HEART Overview

CONTACT INFORMATION

Nancy Lush, Lush Group, Inc.
Nancy.lush@lgisoftware.com    401-423-9111

ref: http://openid.net/wg/heart/
User-Managed Access (UMA) 2.0 Overview

kantarainitiative.org/confluence/display/uma/Home

Eve Maler, UMA Work Group chair and HEART Working Group co-chair | @xmlgrrl | @UMAWG
OAuth enables constrained delegation of access to apps

Benefits:
• Flexible, clever API security framework
• Alice can agree to app connections and also revoke them
UMA adds cross-party sharing...

Benefits:
- **Secure** delegation
- Alice can be absent when Bob attempts access
- Helpful **error handling** for client applications
Benefits:
• Alice controls trust between a service that hosts her resources and a service that authorizes access to them.
...of resource hosts

Benefits:
• Resource hosts can outsource authorization management – and liability – to a specialist service
• Alice can manage sharing at a centralizable service
• Bob can revoke his access to Alice’s resources
UMA user experience opportunities

Resource owner

UX
Share
Monitor
Withdraw
Opt in
Approve

Ahead of time
Anytime
Anytime
At run time
After the fact

Confidential App
is requesting permission to access:
- Access and change your email contacts

Allow Access
No thanks

The Office of the National Coordinator for Health Information Technology
## Benefits for service providers: a summary

<table>
<thead>
<tr>
<th>True secure delegation; no password sharing</th>
<th>Scale permissioning through self-service</th>
<th>API-first protection strategy</th>
<th>Foster compliance through standards</th>
</tr>
</thead>
</table>

- **True secure delegation; no password sharing**: Requires authentication and verification for access, ensuring data security and privacy.
- **Scale permissioning through self-service**: Enables on-demand resource allocation and manages access rights based on user roles and needs.
- **API-first protection strategy**: Focuses on creating secure APIs that abstract security from the underlying infrastructure, promoting a more granular and controlled approach to access management.
- **Foster compliance through standards**: Adheres to regulatory standards and guidelines to ensure data protection and privacy compliance, including practices that align with international frameworks like GDPR.
### Benefits for patients and consumers: a summary

<table>
<thead>
<tr>
<th>Choice in sharing with other parties</th>
<th>Convenient sharing/approval with no outside influence</th>
<th>Centralizable monitoring and management</th>
<th>Control of who/what/how at a fine grain</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Choice in sharing" /></td>
<td><img src="image2" alt="Convenient sharing" /></td>
<td><img src="image3" alt="Centralizable monitoring" /></td>
<td><img src="image4" alt="Control of who/what/how at a fine grain" /></td>
</tr>
</tbody>
</table>

**Benefits for patients and consumers:**

- **Choice in sharing with other parties:**
  - Selection of sharing partners without external influence.

- **Convenient sharing/approval with no outside influence:**
  - Centralized monitoring and management.
  - Control at a fine grain.

- **Centralizable monitoring and management:**
  - Customizable monitoring and approval.

- **Control of who/what/how at a fine grain:**
  - Fine-grained control over data access and use.
UMA in a nutshell

- Developed at Kantara Initiative; V2.0 complete in Jan 2018
- Leverages existing open standards:
  - OAuth2
  - OpenID Connect and SAML (optional but popular)
- Contributed to IETF OAuth WG in Feb ‘19
- Profiled by multiple industry sectors (financial, healthcare)
- UMA business model effort (“BLT”) supports legal licensing for personal digital assets
  - Example: Mother (legal guardian) manages sharing for child (data subject); child becomes old enough and starts to manage sharing herself
User-Managed Access (UMA) 2.0 Overview

CONTACT INFORMATION

Eve Maler, ForgeRock
UMA Work Group chair
HEART Working Group co-chair
eve.maler@forgerock.com | @xmlgrrl | @UMAWG
kantarainitiative.org/confluence/display/uma/Home
UMA 2.0 Deep Dive

kantarainitiative.org/confluence/display/uma/Home

Eve Maler, UMA Work Group chair and HEART Working Group co-chair | @xmlgrrl | @UMAWG
1. The **UMA grant of OAuth** enables Alice-to-Bob delegation

2. UMA standardized an **API for federated authorization** at the AS to make it centralizable

3. There are **nicknames** for enhanced and new tokens to keep them straight
The UMA extension grant adds...

docs.kantarainitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html

- **Party-to-party:** Resource owner authorizes protected-resource access to clients used by requesting parties
- **Asynchronous:** Resource owner interactions are asynchronous with respect to the authorization grant
- **Policies:** Resource owner can configure an AS with rules (policy conditions) for the grant of access, vs. just authorize/deny
  - Such configurations are outside UMA’s scope
UMA federated authorization adds...

docs.kantarainitiative.org/uma/wg/rec-oauth-uma-federated-authz-2.0.html

• **1-to-n:** Multiple RS’s in different domains can use an AS in another domain
  - “Protection API” automates resource protection
  - Enables resource owner to monitor and control grant rules from one place

• **Scope-grained control:** Grants can increase/decrease by resource and scope

• **Resources and scopes:** RS registers resource details at the AS to manage their protection
The UMA Grant
The UMA extension grant flow and its options

The AS is acting as an **agent** for an absent RO

The client’s first resource request is **tokenless**

The RS provides a **permission ticket** and allows **AS discovery**

There are two **claims collection options** for meeting policy

Authorization assessment and token issuance has **guardrails**

RPTs can be **upgraded**, **revoked**, **introspected**, and **refreshed**

---

The client’s first resource request is **tokenless**

The AS is acting as an **agent** for an absent RO

The RS provides a **permission ticket** and allows **AS discovery**

There are two **claims collection options** for meeting policy

Authorization assessment and token issuance has **guardrails**

RPTs can be **upgraded**, **revoked**, **introspected**, and **refreshed**
The permission ticket: how you *start* building a bridge of trust

- **Binds client, RS, and AS:** Every entity may be *loosely coupled*; the whole flow needs to be bound
  - It’s like an overarching state parameter or “ticket-getting ticket”
  - Or maybe even a bit like an authorization code
- **Refreshed for security:** The client can retry RPT requests after non-fatal AS errors, using either claims collection option of the grant flow
  - The AS *refreshes* the permission ticket when responding with such errors
Pushed claims scenario:
for wide-ish ecosystems

The AS is the requesting party’s IdP and the client is the RP

More detail on the RS’s initial response to the client

The client pushes its existing ID token to the token endpoint

The AS is in the primary audience for this token

Somewhat resembles SSO or the OAuth assertion grant, where a token of expected type and contents is “turned in”
Interactive claims gathering scenario: for wide ecosystems

A claims interaction endpoint must have been declared in the discovery document to allow this flow.

The AS mediates gathering of claims from any source.

A key “metaclaim” to think about: consent to persist claims.

A PCT potentially enables a better RqP experience next time; the AS can then re-assess using claims on hand.

Resembles the authorization code grant, but can apply to non-unique identities and is repeatable and “buildable”.

Gather claims interactively:

- Request resource with no access token
  - Determine request requires more permissions than available
  - Return 401, providing UMA auth scheme as _uri, ticket
  
  Redirect RqP to...

  - claims interaction endpoint with permission ticket

  AS redirects RqP back, providing rotated permission ticket...

  ...to claims redirect URI to finish interaction

  - Request RPT, providing grant_type, permission ticket

  Perform authorization assessment

  - Return 200 OK: Return RPT, optionally a PCT

  Request resource with RPT

  - Assess resource request against RPT

  Return protected resource
Federated Authorization
A new perspective on the UMA grant

How does the AS know when to start protecting resources?

How does the RS know what ticket the AS is associating with the RS’s recommended permissions?

Is there anything special about token introspection?

Let’s standardize an interface at the AS for these jobs

Federated authorization perspective

Request resource with no access token

Determine request requires more permissions than available

Return 401, providing UMA auth scheme, as-uri, ticket

Request RPT

Perform authorization assessment

Return 200 OK: Return RPT

Request resource, providing RPT

Assess resource request against RPT

Return protected resource
The protection API: how you *federate* authorization

• **RS registers resources:** This is required for an AS to be “on the job”
  » Scopes can differ per resource
  » Resource and scope metadata assist with policy setting interfaces

• **RS chooses permissions:** The **RS interprets** the client’s tokenless resource request and **requests** permissions from the AS
  » The AS then issues the initial permission ticket

• **RS can introspect the RPT:** UMA **enhances** the token introspection response object

• **RO controls AS-RS trust:** The protection API is **OAuth-protected**
  » The resource owner authorizes the scope **uma_protection**
  » The issued token is called the **PAT**
The resource registration endpoint

Registering a resource puts it under protection

Setting policies can be done anytime after creation

Deregistering a resource removes it from protection

UMA Federated Authorization Resource Registration Endpoint

resource owner (RO) → resource server (RS) → authorization server (AS) → resource reg at AS

Create resource (POST resource description document)

201 Created with resource ID

Set policy conditions

Read (GET) with resource ID

200 OK with resource description document

Update (PUT resource description document) with resource ID

200 OK with resource ID

List (GET)

200 OK with list of resource IDs

Delete (DELETE) with resource ID

200 OK or 204 with No Content
Resource and scope registration

• The RS is authoritative for what its resource boundaries are
  » It registers them as JSON-based descriptions
  » There is a resource “type” parameter

• Scopes can be simple strings or URIs that point to description documents

• The HEART profiles spell out familiar FHIR resource types and FHIR/SMART on FHIR/HL7 scope values
The permission endpoint

The RS interprets the client’s tokenless (or insufficient-token) resource request

The RS must be able to tell from the client’s request context which RO and AS were meant

Request:
POST /perm/ HTTP/1.1
Content-Type: application/json
Host: as.example.com
Authorization: Bearer MHg3OUZEQkZBMjcx
...
{  
  "resource_id":"rsrl",
  "resource_scopes":[
  "patient/*.*read"
  ]
}

Response:
HTTP/1.1 201 Created
Content-Type: application/json
...
{  
  "Ticket":"016f84e8-f9b9-11e0-bd6f-0021cc6004de"
}
The token introspection endpoint

UMA enhances the token introspection response object

A permissions claim is added, with resource ID-bound scopes

UMA Federated Authorization Token Introspection Endpoint

Response:
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

```json
{
  "active":true,
  "exp":1256953732,
  "iat":1256912345,
  "permissions":[
    {
      "resource_id":"rsrcl",
      "resource_scopes":[
        "patient/*/read"
      ],
      "exp":1256953732
    }
  ]
}
```

Request:
POST /introspect HTTP/1.1
Host: as.example.com
Authorization: Bearer MBg3OUZEQkZBMjcx

```text
token=mF_9.B5f-4.1JgM
```
Authorization assessment: how the AS adheres to the RO’s wishes in the larger context

The client can request scopes at the token endpoint, but must have pre-registered them with the AS for it to work.

The AS treats the scopes in this intersection as matching any available scope associated with a resource in the ticket.

Permissions associated with the ticket can add to total requested scopes.

If authorization assessment results in only a subset of client-desired scopes, the AS can choose to error.

RequestedScopes = C ∪ (A ∩ B)
UMA 2.0 Deep Dive

CONTACT INFORMATION

Eve Maler, ForgeRock
UMA Work Group chair
HEART Working Group co-chair
eve.maler@forgerock.com | @xmlgrrl | @UMAWG
kantarainitiative.org/confluence/display/uma/Home
Specifications provide options and extensions
Options aren’t always compatible
Profiles select specific options
What is a profile?

• “A conformant subset of a specification”
• Make optional things mandatory
• Remove problematic options
Choose compatible options
Choose secure options
Things work together
• **Health Relationship Trust**

• Suite of profiles from OpenID Foundation
  » First set of vertical-specific profiles from OIDF

• User-centered access of healthcare data APIs
The HEART approach

**Mechanical Profiles**
- OAuth
- OpenID Connect
- UMA

**Semantic Profiles**
- FHIR over OAuth
- FHIR over UMA
Mechanical Profiles

- Not healthcare specific
- Focus on underlying security layer
- Build interoperability and security
- Connectivity between all components
Semantic Profiles

- Healthcare specific
- Focus on healthcare data access
- Security for FHIR protocol

FHIR over OAuth

FHIR over UMA
HEART mechanical profiles

- All clients have asymmetrical keys
- Servers must support discovery
  - Including all key publication
- Servers must allow dynamic registration
- Servers must enable introspection
- Access tokens are always JWTs
- Only certain kinds of OAuth grants allowed
- All clients are required to register
- Redirect URIs must match exactly
- UMA must support OpenID Connect ID Token claims
- Recommended token lifetimes
HEART OAuth Connections

Client → AS → RS
HEART OAuth Connections

- Discovery
- Keys
- Registration
- URL matching
- Grant types
- Client types
HEART client types

- Full client with user delegation
  » Traditional web application
- In-browser client with user delegation
  » Self-contained single-page-application
- Native client with user delegation
  » Mobile or desktop software
- Direct access client
  » Bulk or batch access, not on behalf of a single user
HEART OAuth Connections

- Token format
- Presentation method
HEART OAuth Connections

- Introspection
- Token format
- Keys
Resource server connections

• Connection between RS and AS is out of scope for OAuth
  » Several options exist but aren’t mandatory

• Specify token format and content
  » JSON Web Token (JWT), signed by AS
  » Include issuer and key pointer, don’t include PII

• Introspection available at AS
Why both JWT and Introspection?

- Signed JWTs give a fast first check
  - Is this from a server that I trust? Has it been modified? Is it expired?

- Introspection gives detailed and real-time information
  - What’s this token actually good for? Has it been revoked?

- An RS can talk to multiple AS
  - Parse the JWT to see which AS to introspect the token at
HEART OpenID Connections
HEART OpenID Connections

- ID Token claims
- Signature methods
- ... plus everything from OAuth profile
HEART UMA Connections

Client → AS → RS
HEART UMA Connections

- ID Token claims
- UMA Discovery
- Token format
- ... plus everything from OAuth
HEART UMA Connections

- Token format
- ... plus everything from OAuth
HEART UMA Connections

- Federated authorization
- Token format
- Introspection
- ... plus everything from OAuth
HEART Semantic Profiles

• How to access FHIR APIs
• Which scopes to ask for as a client
• How to interpret scopes as a resource
patient/Condition.read
patient/Condition.read

- “patient” – individual accessing a specific record
- “user” – bulk access to a set of records
HEART core scope

patient/Condition.read

• Name of FHIR resource to access
  » Any FHIR resource type can be used

• Wildcard allowed for “all resources”: *

The Office of the National Coordinator for Health Information Technology
HEART core scope

patient/Condition.read

- “read”: I can download information from the API
- “write”: I can upload information to the API
- “*”: I can do any available action including “read” and “write”
HEART confidentiality scope

conf/R

• For information tagged with confidentiality markers
• Tokens with this scope are allowed access to this kind of information
• Three basic levels, plus not-specified
HEART sensitivity scope

sens/SOC

- For information tagged with sensitivity markers
- Tokens with this scope are allowed to access this kind of information
- Standard set of sensitivity markers
HEART emergency scope

btg

- “break the glass”

- This client is allowed to access information in an emergency situation
  - Potentially because of who the resource owner is

- Triggers additional audit and notification requirements
Other profiles

- SMART
- OpenBanking UK
- FAPI
- iGov
SMART

- Deployed healthcare project for user-controlled applications
- Targets application portals and bundled applications
  - Integration for healthcare providers
  - Adds a “launch” context
- HEART semantic profiles are based on SMART scopes
  - Aligned but not built on
OpenBanking UK

- Financial industry consortium profile for UK banks
- Allow user-controlled apps access to account info and transfer functions
  - Account management
  - Transfer money (electronic payment)
- Government-led mandate to drive industry forward
FAPI

- OpenID Foundation profile for finance and high-value APIs
  - Focus on financial APIs
- Parent specification of OpenBanking UK
- Source of general-purpose extensions
  - CIBA
  - JARM
iGov

• OpenID Foundation profile for international government use
• Similar technical profiles
• Extended profiling of OpenID Connect claims
  » Government identification numbers
  » Proofing documents
  » Vectors of Trust integration
## Comparing Profiles

<table>
<thead>
<tr>
<th>Feature</th>
<th>HEART</th>
<th>SMART</th>
<th>OB</th>
<th>FAPI</th>
<th>iGov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit Grant</td>
<td>Restricted</td>
<td>Forbidden</td>
<td>Required (hybrid)</td>
<td>Required (hybrid)</td>
<td>Restricted</td>
</tr>
<tr>
<td>Mobile</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PKCE</td>
<td>Mobile</td>
<td>Optional</td>
<td>Optional</td>
<td>All</td>
<td>Mobile</td>
</tr>
<tr>
<td>Identity</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>URI Match</td>
<td>Exact</td>
<td>Exact</td>
<td>Exact</td>
<td>Exact</td>
<td>Exact</td>
</tr>
<tr>
<td>Shared Secrets</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>DynReg</td>
<td>Y</td>
<td>N</td>
<td>Y (specialized)</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>OAuth 2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>OIDC</td>
<td>Some</td>
<td>Some</td>
<td>Y</td>
<td>Some</td>
<td>Y</td>
</tr>
<tr>
<td>UMA 2</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Ecosystem</td>
<td>Wide</td>
<td>Narrow</td>
<td>Narrow</td>
<td>Narrow</td>
<td>Wide</td>
</tr>
</tbody>
</table>
Justin Richer, Bespoke Engineering

justin@bspk.io

https://bspk.io/
Using HEART OAuth 2.0 Scopes with UMA 2.0

David Staggs, JD, CISSP, Subject Matter Expert, Security Risk Solutions, Inc.
HEART FHIR OAuth 2.0 Token Scopes

- Scopes define individual pieces of authority that can be requested by clients, granted by resource owners, and enforced by protected resources.

- In the HEART specification, scopes are described as:
  \[
  \text{scope} := \text{permission/resource.access}
  \]

- Permission can be “patient” (single patient) or “user” (bulk).

- Resource can be any FHIR resource.

- Access can be “read” or “write.”

- Additional access scopes, e.g. confidentiality and sensitivity, are supported.
Token Scope Illustration

```plaintext
patient / fhir-resource . read
  * write
```
HEART Security Labels: Confidentiality Codes

- Confidentiality codes describe the sensitivity of the information associated with the resource.

  » Considered the “high water mark” across a collection of data.

- Confidentiality code vocabulary supported by HEART: N, R, and V

- Example token scope using a confidentiality code:

  "scope": "patient/\.*\.* conf/R"

  This request has permission to access data labels as restricted (e.g. data concerning HIV status).
HEART Security Labels: Sensitivity Labels

• Sensitivity labels represent the sensitive nature of the data.
  » Allows data segmentation of data based on privacy policy and patient consent.

• Example token scope using sensitivity scopes:
  "scope": "patient/\*.* sens/ETH sens/PSY"

full access to this patient’s data including substance abuse information and psychiatry disorder information.
In general, POU involves the reason for, or context of, the request (used to determine appropriateness of allowing access).

- General categories: marketing, operations, payment, research, patient requested, public health, and treatment

- POU security label vocabulary includes: emergency access, break the glass, research, etc.

- Example request using the break the glass scope:
  
  "scope": "patient/*.* btg"
  
  full access to this patient’s data even if patient consent is not available.
UMA 2.0 Entities

Diagram showing the relationships between resource servers, authorization servers, clients, requesting parties, and UMA grants.
UMA 2: Resource Owner Authorizes Resource Server

- Resource Owner
  - Defines desired scopes and resources
  - Authorizes PAT for use by resource server
- Resource Server
- AuthZ Server
  - Later, sets policy, e.g., consent required for sensitive data
  - Endpoints:
    - Resource Registration
    - Permission Introspection
- Identity Provider
- Client
- Requesting Party

Red font = UMAGrant
Green Font = UMAFedAuthz
UMA 2: Resource Registration Request and Response

Resource Owner

Resource Server

(using PAT, RS Client Credentials)

Endpoint:
Resource Registration
Permission
Introspection

(_id)

AuthZ Server

Returns a value corresponding to the registered resource

Identity Provider

Client

Requesting Party

RS registers RO’s resource, includes scopes

Red font = UMAGrant
Green Font = UMAFedAuthz

The Office of the National Coordinator for Health Information Technology
UMA 2: Protected Resource Request without RPT

RS identifies AS and selects permissions; requests ticket from AS

Endpoint
Protected Resource

Returns Permission Ticket + AuthZ Server Location

Resource Request
(No Requesting Party Token)

Resource Server

Endpoints:
Resource Registration
Permission
Introspection

Ticket

@endpoint

AuthZ Server

Returns a permission ticket representing requested permissions

Identity Provider

Client

Requesting Party

Resource Owner
UMA 2: Client Seeks RPT for the Requesting Party

Resource Owner

Resource Server

Endpoint Protected Resource

Endpoints:
- Resource Registration
- Permission
- Introspection

AuthZ Server

Endpoint Token

Claims Interaction

Client ID & Permission Ticket

Client

Error code & Permission Ticket

Requesting Party

Identity Provider

AS may use pushed claims or require interactive claims gathering

Red font = UMAGrant
Green Font = UMAFedAuthz
UMA 2: Access Request with RPT

Resource Owner

Requests what permissions are in RPT

Resource Server

(Using PAT, RS Client Credentials)

Endpoint Protected Resource

Returns permissions

Endpoints:
Resource Registration
Permission Introspection

AuthZ Server

Endpoint Token
Claims Interaction

Client

Resource Request with RPT

Allows request

Identity Provider

Red font = UMAGrant
Green font = UMAFedAuthz
The Permission Concept

- A permission (requested or granted) represents authorized access to a particular resource with some number of scopes bound to that resource.
- A permission ticket represents some number of requested permissions.
- An RPT represents some number of granted permissions.
- Requesting a permission with no scopes might be when an API call is ambiguous without further context – a request for a particular scope at the token endpoint later can clarify the desired access. (UMAFedAuthZ pp. 19-20)
  » As we did with BTG scope in the previous example.
Permissions Parameter (From Introspection Example)

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
...
{
  "active":true,
  "exp":1256953732,
  "iat":1256912345,
  "permissions":[
    {
      "resource_id":"112210f47de98100",
      "resource_scopes":[
        "view",
        "http://photoz.example.com/dev/actions/print"
      ],
      "exp":1256953732
    }
  ]
}

The parameter named "permissions" contains an array of objects, each one represents a single permission.

The parameter named "resource_scopes" contains an array of strings representing scopes to which access was granted for the associated resource.
Requesting an RPT: the Authorization Assessment

• For each resource in the permission ticket, the final set of requested scopes are the combination of 1) scopes found in the permission ticket and 2) any requested scopes that are also pre-registered by the client.

• AS then applies claims and policies to each set of final requested scopes and determines an authorization decision.

• Each requested scope allowed on a resource is collected in the CandidateGrantedScopes(resource) array.

• AS then issues either an RPT containing CandidateGrantedScopes for each resource, or an error codes, as appropriate.
Using HEART OAuth 2.0 Scopes with UMA 2.0

CONTACT INFORMATION
David Staggs, JD, CISSP,
Subject Matter Expert, Security Risk Solutions, Inc.

drs@securityrs.com
Trusted Dynamic Client Registration
Tools to increase scalability and confidence in the HL7 FHIR® ecosystem

Luis Maas, MD, PhD | CTO, EMR Direct
Dynamic Client Registration

• Client app registration today is typically a manual process

• And we are only getting started -- client app proliferation expected

• Automation needed in order to scale the process of enabling trust between the growing number of client apps, servers, users, and to appropriately authorize data access according to one or more community standards or common agreements

• Support for DCR is required by HEART for native client apps & OAuth servers
Dynamic Client Registration Benefits

- Consume FHIR resources using an app, the same way you would a browser...

App registration details are clearly indicated.
• Client app endorsements & certifications
  » Expands Dynamic Client Registration into a framework that can combine some vetting on an endorser’s side, informing an endpoint’s registration decisions
  » Also increases an end user’s confidence in the application
  » Uses digital signatures for authenticity and integrity
    – Can be packaged as signed JWTs for distribution and integrity protection
    – Can use X.509 tools to facilitate key distribution
    – Active work on harmonizing current initiatives in the field
• Client app identity

  » Opportunity to go beyond self-assertions by clients to validated information about identity and other attributes like privacy policy

  » Can extend to FHIR endpoints, increasing confidence in server identity during exchange and informing directory resources
Ecosystem Components & the OAuth Sign In Page

User Authorization: Pre-registered or Discoverable Identity; Client may access Health Data & Consents

Trusted FHIR Server: Discovery, Validation, Provenance

Dynamic Client Registration (DCR) & Trusted DCR: App Discovery & Validation, Reusable Client Vetting

By clicking Authorize, you agree to the Interoperability Engine Open API Terms of Use and request that ABC Hospital share with HealthToGo the following health information accessible using your credentials:

- Personal information, such as name, birthdate, gender, and other demographics
- Observations, such as lab results, vital signs, imaging, and social history
- Conditions, such as medical problems, diagnoses, and health concerns
- Documents, such as summaries of care and discharge summaries
- Records relating to medications, allergies, immunizations, surgeries or other procedures, implanted devices, care plans, care teams, and goals
- Any other categories of health information or other data, including categories that become accessible in the future

The client application is also requesting:

- Personal information about you, such as your name
- Information about health data you have shared with others

Afterwards, you’ll be automatically redirected back to HealthToGo.

Contact ABC Hospital directly regarding credentials, or with other questions about application access APIs.

*About the app you are using to access this data:

HealthToGo completed an automated dynamic client registration process to identify itself. The developer of HealthToGo provided the following website during the registration process:

http://www.smindirect.com

The information above was provided by the app developer. This app also presented a trusted digital certificate containing the following verified information:

Developer Organization: EMR Direct

Privacy Policy: https://www.emrdirect.com/privacy

You assume all responsibility and liability for any apps you authorize. Apps may use their data use policies and may not be subject to the same privacy and security laws that healthcare providers are, so refer to the app developer’s privacy policy before processing.
Trusted Identity Networks

• Cross-organizational reciprocity of user credentials issued by trusted Identity Providers
  » Tiered OAuth
  » Increases the usefulness and scalability of sharing when data grantees do not need to have local credentials
  » Reusable digital identities
Ecosystem Components
Adding Federated Identities

Trusted 3\textsuperscript{rd} Party OAuth Servers:
Leverage Reusable Identities Issued by Others through Tiered OAuth

User Authorization—
Pre-registered or Discoverable Identity;
Client may access Health Data & Consents

Trusted FHIR Server:
Discovery, Validation, Provenance

Dynamic Client Registration (DCR) &
Trusted DCR:
App Discovery & Validation, Reusable Client Vetting
**Ecosystem Summary**

- Pre-register user, client, and/or server
- Can be automated, with optional vetting

**Endpoint certificate or OpenID**

**Query**
- Each participant can dynamically register and be reliably identified

**Response**
- Participants can dynamically discover attributes about other participants to inform policy decisions
What’s Next?

• Implementation Guide for Trusted Dynamic Client Registration

• HL7 May FHIR Connectathon Track in Montreal, Canada

• Continued development of Unified Data Access Profiles (UDAP) to scale trusted networks (www.udap.org)

• Develop participation agreements and baseline criteria
Getting In Touch

CONTACT INFORMATION

Luis Maas, CTO, EMR Direct
lcmaas@emrdirect.com, 858 367 0770