**Department of Veterans Affairs**

**Veterans Health Administration**

**Office of Information**

Security Architecture, Framework Engineering

**Use Cases for FHIR Security Authorization with Patient Consent**

The intellectual property in this document is made available by the   
Veterans Health Administration, US Department of Veterans Affairs

DRAFT

Version 0.6

Dec 13, 2016

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Reason for Changes** | **Author** |
| 31 Oct 2016 | 0.1 | Initial Draft | Tony Mallia |
| 1 Dec 2016 | 0.2 | Remove extraneous content | Tony Mallia |
| 6 Dec 2016 | 0.3 | Minor corrections | Tony Mallia |
| 8 Dec 2016 | 0.4 | Detailed review adjustments | Tony Mallia |
| 11 Dec 2016 | 0.5 | Final review adjustments | Tony Mallia |
| 13 Dec 2016 | 0.6 | Move standards to appendix | Tony Mallia |

Contents

[1 Introduction 1](#_Toc469406260)

[2 Composite Framework 2](#_Toc469406348)

[2.1 Overview 2](#_Toc469406349)

[2.2 Custodian Organization Authorization Server (CO AS) 2](#_Toc469406350)

[2.3 Custodian Organization Patient Consent Authorization Server (CPC AS) 3](#_Toc469406351)

[2.4 Third Party Patient Consent Authorization Server (TPPC AS) 3](#_Toc469406352)

[2.5 Trust Chain 4](#_Toc469406353)

[2.5.1 Trust Description 4](#_Toc469406354)

[2.5.2 Access Token Validation 5](#_Toc469406355)

[3 Use Cases 6](#_Toc469406356)

[3.1 Patient Manages Consent 6](#_Toc469406357)

[3.1.1 Actors 6](#_Toc469406358)

[3.1.2 Sub Use cases 7](#_Toc469406359)

[3.1.3 Choreography for Patient Consent Submission 8](#_Toc469406360)

[3.1.4 Patient Identity Authentication and Matching 8](#_Toc469406361)

[3.2 Consent is considered in Access Decision 8](#_Toc469406362)

[3.3 Composite Use cases extension to Authorize Access 9](#_Toc469406363)

[3.3.1 Actors 9](#_Toc469406364)

[3.3.2 Sub Use Cases 9](#_Toc469406365)

[3.4 Access Control Choreography (Grant flow) 11](#_Toc469406366)

[4 Aspects for Profile 12](#_Toc469406367)

[5 Appendix Existing specifications 13](#_Toc469406368)

[5.1 FHIR Specification 13](#_Toc469406369)

[5.1.1 6.1.0.1 General Considerations 13](#_Toc469406370)

[5.1.2 6.1.0.4 Authorization/Access Control 14](#_Toc469406371)

[5.2 User Managed Access (UMA) 15](#_Toc469406372)

[5.2.1 UMA Description 15](#_Toc469406373)

[5.2.2 The Three Phases of the UMA Profile of OAuth 16](#_Toc469406374)

[5.3 US DAF FHIR Profile (Now US Core Profile) 16](#_Toc469406375)

[5.4 Specification Conclusion 17](#_Toc469406376)

[5.4.1 Compliance requirement 17](#_Toc469406377)

[5.4.2 Resource Ownership 17](#_Toc469406378)

[6 Acronyms 18](#_Toc469406379)

# Introduction

This is an exploration into a Cascading approach for access decision making based on OAuth 2.0 with an ability to perform multiple redirects to achieve a decision based on both Custodian and Patient policies or directives .A decision is based on first, the Custodian’s privacy policies, which may then require that the decision consider the patient’s consent directive.

A previous approach known as “Privacy on FHIR” demonstrated cascading decisions with partial use of OAuth 2.0 and the UMA profile.

The approach in this paper provides a complete OAuth 2.0 environment where the Client always gets back an Access Token which can be used for its lifetime thereby reducing repetitive triggering of the whole Access Control Service.

The paper discusses the various current standards involved in Access Decision making for the HL7 FHIR environment and particularly those that require Patient Consent. These standards are then harmonized in a preliminary way to provide a set of use cases which describe both the submission of Patient Consent by the Patient and the consumption of that Patient Consent during the Access Decision flow when the user is granted or denied access to the Patient health resources.

The intention of the paper is as a prelude to the formation of an OAuth 2.0 profile which supports the multiple jurisdictions of Health Record Custodian and Patient where the decision is made based on multiple Authorization Servers acting collaboratively where each acts on behalf of the Custodian or Patient jurisdiction.

The specific content of the tokens is expected to be determined during the profile development and the parties to construct the profile are not identified at this time.

# Composite Framework

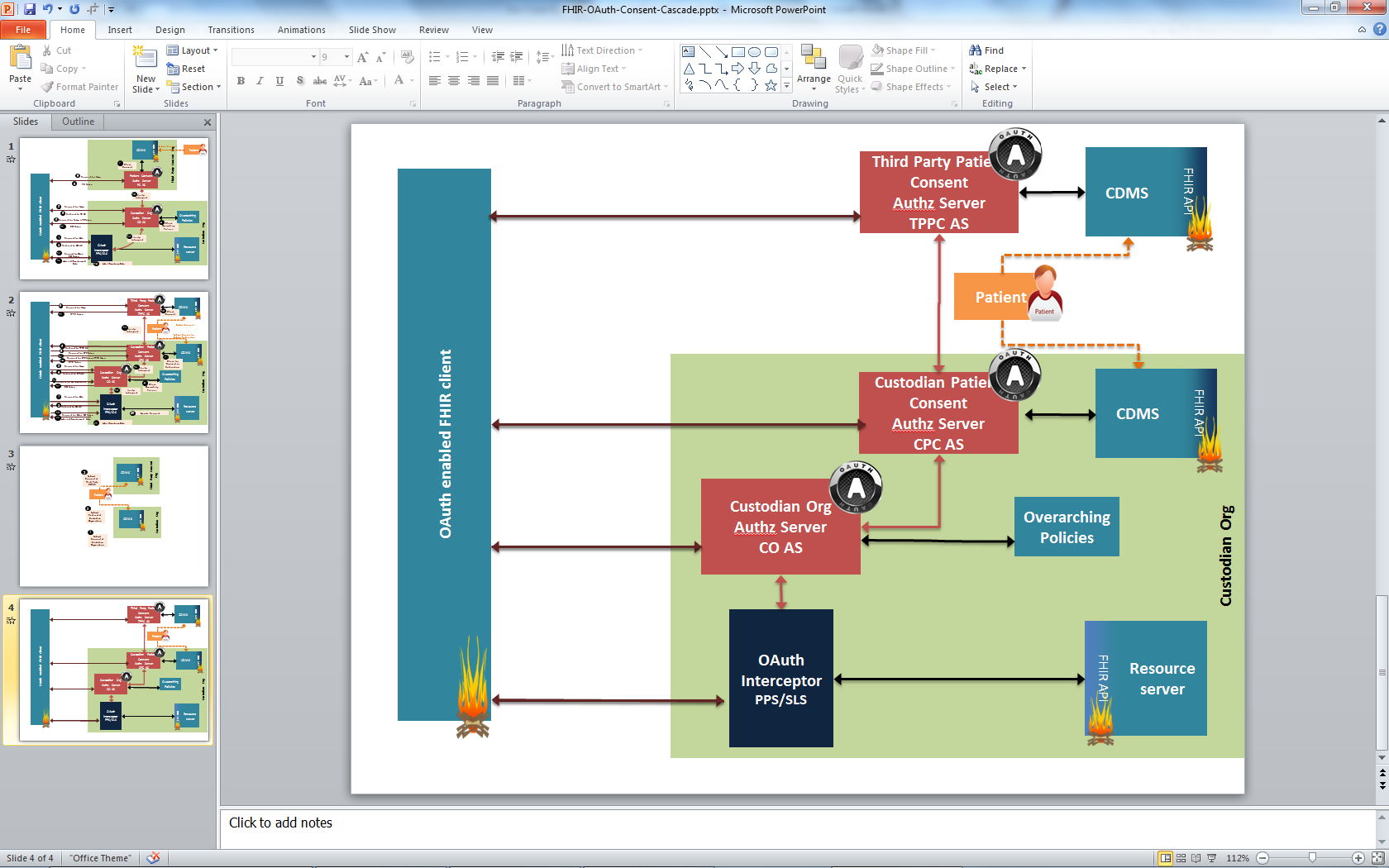
## Overview

The concept for this Composite Approach is that the Patient’s Consent Directive may be administered by the Custodian Organization or by a Third Party Patient Consent Directive Management Service

The Custodian Organization makes Access Decisions based on privacy/security policies and the Patient Consent wherever it is.

The Client makes an access request to the Resource Server protected by the interceptor which redirects to the Custodian Organization OAuth Authorization Server (CO AS). An access decision is cascaded between the CO AS supporting the Provider Organization’s policies with an OAuth Authorization Server still in the **Provider Organization** representing the Patient Consent Directives (CPC AS) and optionally cascading further to a UMA AS Third Party Consent Repository (TPCR) (also an OAuth Authorization Server) if the patient has given a directive to the Provider Organization to do so.

The patient Consent Authorization Servers are supported by Consent Directive Management Systems which are the repositories for Patient Directives.



## Custodian Organization Authorization Server (CO AS)

It can also be considered the same as SMART on FHIR Server where the jurisdiction is the Custodian.

*Authorization server – server used by EHR system to authenticate the clinician and to authorize the application to access EHR data on behalf of the clinician, in accordance with legal and institutional policies,*

## Custodian Organization Patient Consent Authorization Server (CPC AS)

This is an authorization Server managed by the Custodian Organization and is equivalent to the SMART on FHIR AS. It makes access control decisions based on Resource Owner (patient) consent stored in the CDMS (Consent Directive Management System)

## Third Party Patient Consent Authorization Server (TPPC AS)

This is an authorization Server managed by a Third Party Patient Consent Directive Management Service. It makes access control decisions based on Resource Owner (patient) consent stored in the CDMS (Consent Directive Management System). There can be many TPPC AS and the patient may choose which one to use.

The TPPC AS used by the patient is known to the Custodian Organization through the entry in the CPC AS denoting the TPPC AS to use for that patient.

## Trust Chain

The Composite framework has a chain of trust:

Trusts

Trusts

Trusts

Trusts

Trusts

Trusts

Trusts

### Trust Description

The Custodian Resource Server trusts the CO AS.

The CO AS trusts and delegates the determination of Patient Consent to the CPC AS. All validation of directives is done by the CPC AS.

The CPC AS trusts the patient Consent Directive in that there is reliable correlation of the patient identity and that the consent has been verified to be originated from that patient (electronic signature).

Similarly the CPC AS trusts a redirection address coming from the patient if the resolution of this address to a TPPC AS is to a trusted one.

The CPC AS trusts the TPPC AS based on some certification or accreditation of that organization.

Lack of accreditation of the TPPC AS would lead to rejection of the submission of the Redirection Address by the Patient.

### Access Token Validation

The verification that the OAuth token comes from the trusted party is key to the acceptance of the token by recipient.

* The Resource Server only accepts OAuth Access Tokens signed by the CO AS.
* The CO AS only accepts OAuth Consent Access Tokens signed by the CPC AS.
* The CPC AS only accepts OAuth Consent Access Tokens signed by an accredited TPPC AS which has been redirected by the Patient in the CPC CDMS.

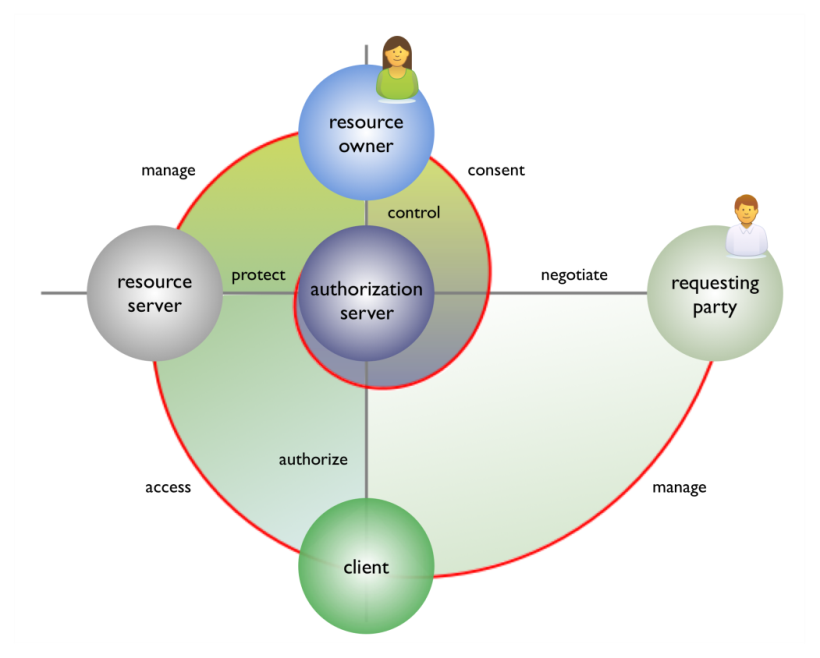
Accreditation is performed when a Consent Redirection from a Patient to a TPPC AS is not on the accreditation list and is a workflow triggered to make this happen. The trigger is an error on the CPC CDMS when the redirection is attempted.

# Use Cases

## Patient Manages Consent



### Actors



* Patient - This is the same as a specialized Resource Owner in UMA.
* Custodian Patient Consent Authorization Server
* Third Party Patient Consent Authorization Server

### Sub Use cases

#### Submit Consent at Custodian

This sub-use case is the submission of the patient consent (or redirection address) to the Custodian Organization CDMS with any of the three options:

##### Paper submission

Paper form is submitted and an authorized user acts for the Patient to submit the consent to the CDMS

##### On-line access to Health Enterprise

The Patient accesses the CDMS directly via a Web Portal

##### Mobile access to Health Enterprise

The Patient accesses the CDMS directly via a Mobile device application

#### Submit Consent to Third Party

Step 2 is the submission of the patient consent to the Third Party Consent Repository – CDMS.

Step 3 is the submission of a redirection to the Health Enterprise to instruct that the Patient’s consents are at that address – see Submit Consent at Custodian.

##### Paper submission

Paper form is submitted and an authorized user acts for the Patient to submit the consent to the Third Party Consent Repository – CDMS

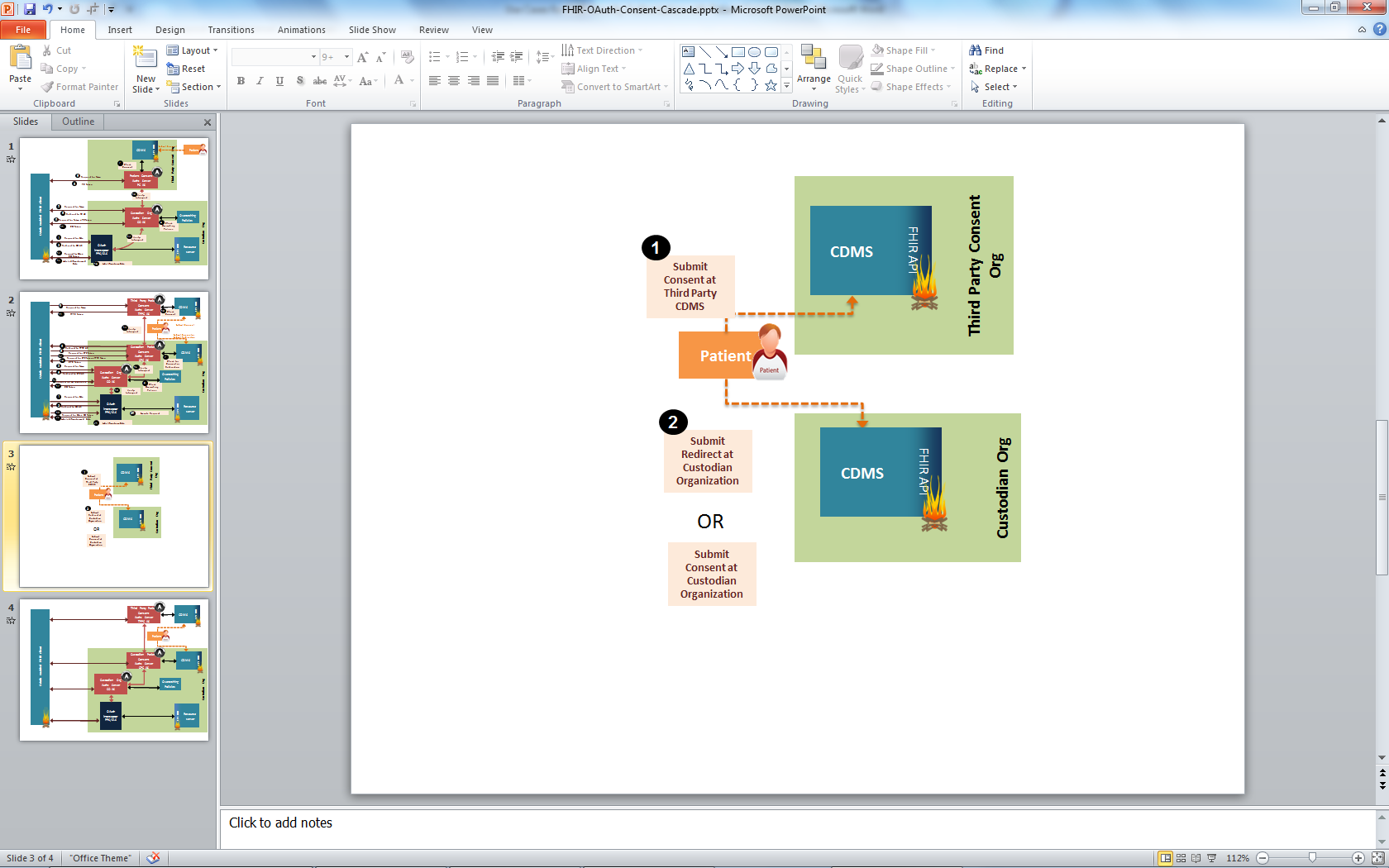
##### On-line access to Third Party Consent Repository (TPCR)

The Patient accesses the TPCR directly via a Web Portal

##### Mobile access to Third Party Consent Repository (TPCR)

The Patient accesses the TPCR directly via a Mobile application

### Choreography for Patient Consent Submission



The flow does not show the access control which is managed by the respective Authorization Servers. These authorization servers allow the patient to access and manage their own Consent Directives. In this respect the access mechanism is Identity Based Access Control (IBAC) and relies on the authenticated identity of the patient to match the identity assigned to the Consent Directive.

### Patient Identity Authentication and Matching

#### CPC Matching

Identity matching in the Custodian Patient Consent is performed by controlling the provisioning of Patient access to CPC such that the identity is known from authentication and matches that used to access the Consent Directive.

#### TPPC AS Matching

TBD

## Consent is considered in Access Decision

This is the implementation of Consent requirement in US Core FHIR Profile.

## Composite Use cases extension to Authorize Access



### Actors

* Client Application -   
  *Application – requests, and (on approval) accesses EHR data on the clinician’s behalf*
* Custodian Organization Authorization Server
* Custodian Patient Consent Authorization Server
* Third Party Patient Consent Authorization Server

### Sub Use Cases

#### Authorize Access

*Authorization server authorizes the application to retrieve the requested data within a limited period of time.*

It is extended to consider consent in this step.

#### Check Overarching Policies

The Custodian Authorization Server evaluates the request for access and determines whether Patient Consent is required. If not required it can authorize or deny the request. If Patient Consent is needed it checks for consent.

#### Check for Consent

The Custodian Authorization Server delegates to the Custodian Consent Authorization Server to make an authorization based on the Patient Consent. This delegation is through an OAuth redirection.

#### Authorize for Consent at Custodian

The Custodian Consent Authorization Server checks whether there is consent for the patient.

If there is Consent then it makes a decision whether to authorize the request based on that consent.

If there is a Redirection Address for the patient then it delegates to the Third Party Patient Consent Authorization Server. This delegation is through an OAuth redirection.

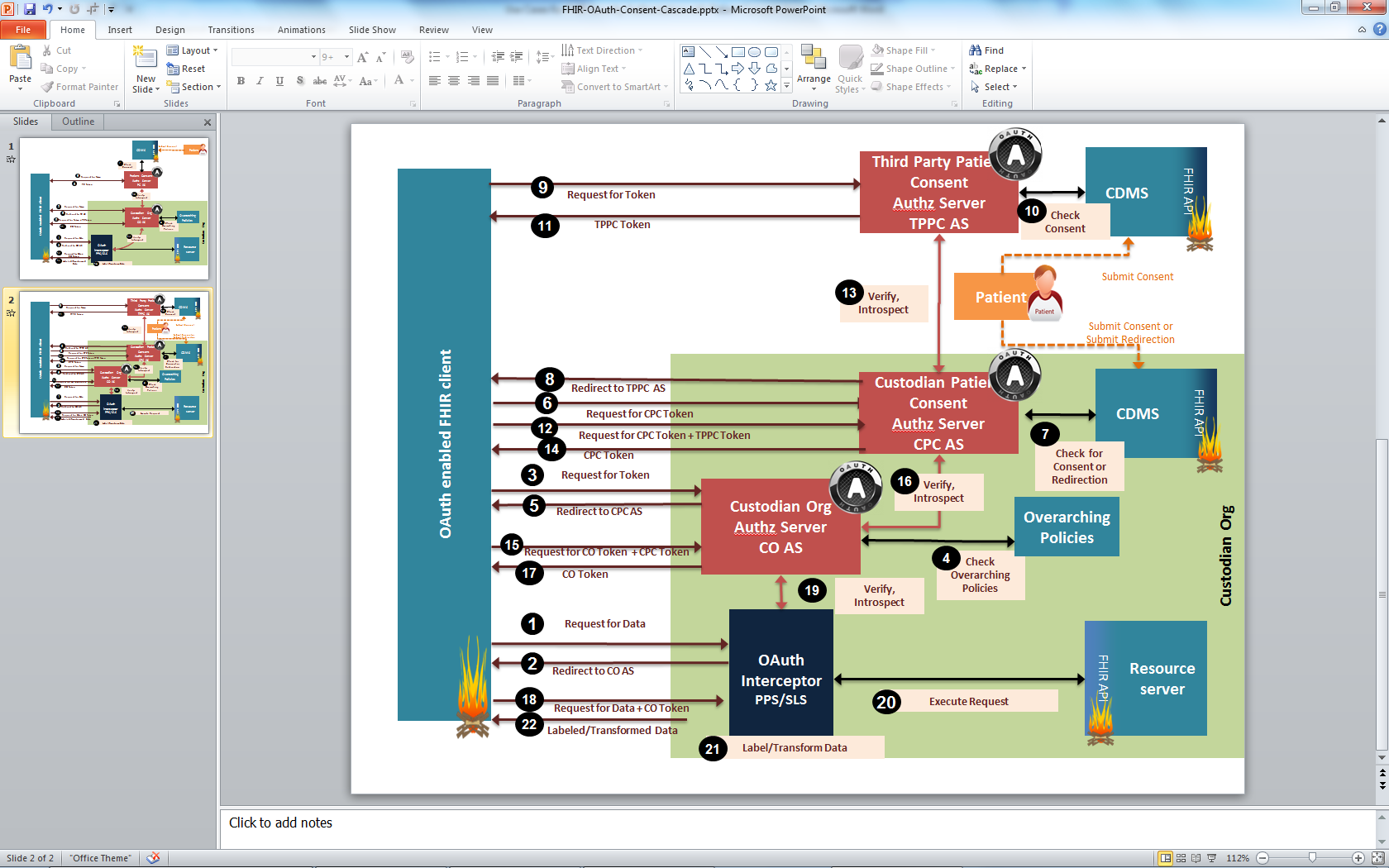
#### Authorize for Consent at Third Party

The TPPC AS makes a decision whether to authorize the request based on the Patient Consent stored at the TPPC CDMS.

The logic of that decision is opaque to the Custodian but known to the Patient thus there is no opportunity for the Custodian to accept or reject any restriction there.

The decision to accept authorization at the TPPC AS is made by the Custodian when accepting the Redirection Address from the Patient in step 3 of the use Case “Submit Consent to Third party”. A custodian may refuse to accept the redirect based on its policies.

## Access Control Choreography (Grant flow)



Sequence Description

* 1. Client makes a FHIR request to the Resource Server which is intercepted.
  2. The interceptor redirects the client to the Custodian Org Authz Server (CO AS)
  3. Client makes a request for an Access Token and may be challenged for user credentials (not shown)
  4. CO AS Checks overarching policies to determine if Consent is needed (assumed yes here)  
     If not needed then go to step 17
  5. CO AS redirects Client to Custodian Patient Consent Authz Server (CPC AS)
  6. Client makes a request for a CPC Token
  7. CPC AS checks its CDMS for either a Consent or Redirection (assumed here)  
     If Consent is present go to step 14
  8. CPC AS Redirects Client to Third Party Patient Consent Authz Server (TPPC AS)
  9. Client makes a request for a TPPC Token
  10. TPPC AS checks its CDMS for a Consent (Assumed permitted here)
  11. TPPC Token returned to Client
  12. Client repeats the request for a CPC Token submitting the TPPC Token
  13. CPC AS inspects the Token from TPPC AS
  14. CPC Token returned to Client
  15. Client repeats the request for an Access Token to the CO AS
  16. COAS inspects the token from CPC AS
  17. COAS Returns Access Token to Client
  18. Client repeats Request for Data submitting an Access Token
  19. The Resource Interceptor inspects the Access Token from the CO AS and validates it
  20. If valid the Interceptor Executes the data access request on the Resource Server
  21. The Privacy Protective Service (PPS) and Secure labeling Service (SLS) label and redact data (not technically part of this use case
  22. The data is returned to the client.

# Aspects for Profile

It is expected that a profile will be developed based on the Composite Framework which will identify the specific OAuth interactions for the Client interactions with the Cascading Authorization Servers.

In general, the focus of the profile will be to structure the tokens the same way since the client is seeing all of them. In addition the structure of the tokens must be sufficient so that scopes are passed correctly to match consent scopes.

Additional structures need to be defined for all the interactions.

# Appendix Existing specifications

This section explores current specifications and their relationships.

## FHIR Specification

<http://hl7-fhir.github.io/security.html>

The current build FHIR specification is not precise about the method by which implementation of security and decisions would be made on a client / server pair agreement. This is not sufficient when the client and server are in different organizations.

The FHIR specification does not even specify the use of OAuth.

### 6.1.0.1 General Considerations

A production FHIR system will need some kind of security sub-system that administers users, user authentication, and user authorization. Where this subsystem fits into the deployment architecture is a matter for system design:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| http://hl7-fhir.github.io/security-layout.png |  | |  |  | | --- | --- | | http://hl7-fhir.github.io/security-icon-user.png | The consumer that is using a healthcare related system | | http://hl7-fhir.github.io/security-icon-app.png | The client application the user is using (application, mobile app, website, etc.) | | http://hl7-fhir.github.io/security-icon-sec.png | The security system (authentication and access control) | | http://hl7-fhir.github.io/security-icon-fhir.png | The clinical/healthcare repository | |

In this diagram, the red lines represent FHIR interfaces. From the perspective of the FHIR API, the client (consumer of FHIR services) may either interact with a security system that manifests as a FHIR server, and which depends on a subsequent FHIR interface to provide the actual storage, or either the client or server interacts with the security system independently. In each of these 3 scenarios, the different components may be assembled into applications or network components differently, but the same logical layout applies. The FHIR specification assumes that a security system exists, and that it may be deployed in front of or behind the FHIR API.

The security system includes the following subsystems:

* Authentication: identifies and authenticates the user
* Access Control decision engine: decides whether FHIR operations are allowed
* Audit Log: records actions to allow for subsequent review and detection of intrusion or inappropriate usage

Because there are a plethora of standards relating to the administration and functionality of the security system, FHIR does not provide user, profile, or other such administration resources. Instead, the FHIR resources are the targets of the policies expressed in these other approaches. What FHIR does specify is a way to apply [security labels](http://hl7-fhir.github.io/security-labels.html) to resources so that a security system may use these (along with the contents of the resources if appropriate) to determine whether a user is authorized to perform a particular FHIR operation or not.

### 6.1.0.4 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 implementation. However, real world usage will generally require this.

A holder of data should not allow the data to be communicated unless there are sufficient 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 is allowed to 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 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 depends 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 transaction, system identity, time-of-day, purpose of use, workflow state, and transport security

For one source of further information, see the [IHE Access Control white paper [http://hl7-fhir.github.io/external.png](http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_WhitePaper_AccessControl_2009-09-28.pdf)](http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_WhitePaper_AccessControl_2009-09-28.pdf)

Access control constraints may result in data returned in a read or search being redacted or otherwise restricted. See [Variations between Submitted data and Retrieved data](http://hl7-fhir.github.io/updates.html).

## User Managed Access (UMA)

### UMA Description

User-Managed Access (UMA) is a profile of OAuth 2.0 [[RFC6749]](https://docs.kantarainitiative.org/uma/draft-uma-core.html#RFC6749). UMA defines how resource owners can control protected-resource access by clients operated by arbitrary requesting parties, where the resources reside on any number of resource servers, and where a centralized authorization server governs access based on resource owner policies. Resource owners configure authorization servers with access policies that serve as asynchronous authorization grants.

UMA serves numerous use cases where a resource owner uses a dedicated service to manage authorization for access to their resources, potentially even without the run-time presence of the resource owner. A typical example is the following: a web user (an end-user resource owner) can authorize a web or native app (a client) to gain one-time or ongoing access to a protected resource containing his home address stored at a "personal data store" service (a resource server), by telling the resource server to respect access entitlements issued by his chosen cloud-based authorization service (an authorization server). The requesting party operating the client might be the resource owner, where the app is run by an e-commerce company that needs to know where to ship a purchased item, or the requesting party might be resource owner's friend who is using an online address book service to collect contact information, or the requesting party might be a survey company that uses an autonomous web service to compile population demographics. A variety of use cases can be found in [[UMA-usecases]](https://docs.kantarainitiative.org/uma/draft-uma-core.html#UMA-usecases) and [[UMA-casestudies]](https://docs.kantarainitiative.org/uma/draft-uma-core.html#UMA-casestudies).

### The Three Phases of the UMA Profile of OAuth

+--------------+

| resource |

+---------manage (A)------------ | owner |

| +--------------+

| Phase 1: |

| protect a control (C)

| resource |

v v

+------------+ +----------+--------------+

| | |protection| |

| resource | | API | authorization|

| server |<-protect (B)--| (needs | server |

| | | PAT) | |

+------------+ +----------+--------------+

| protected | | authorization|

| resource | | API |

|(needs RPT) | | (needs AAT) |

+------------+ +--------------+

^ |

| Phases 2 and 3: authorize (D)

| get authorization, |

| access a resource v

| +--------------+

+---------access (E)-------------| client |

+--------------+

requesting party

UMA does not specify that the Owner of the Resource is the Patient. However this has been widely interpreted as the case.

## US DAF FHIR Profile (Now US Core Profile)

DAF transactions often make use of patient-specific information which could be exploited by malicious actors resulting in exposure of patient data. For this reason, all DAF transactions must be secured appropriately with access to limited authorized individuals, data protected in transit, and appropriate audit measures taken.

For Authentication and Authorization, Systems SHALL use the [Smart On FHIR [http://hl7.org/fhir/external.png](http://fhir-docs.smarthealthit.org/argonaut-dev/authorization/backend-services/)](http://fhir-docs.smarthealthit.org/argonaut-dev/authorization/backend-services/)OAuth2 profiles[[1]](#footnote-1). NOTE: The Smart On FHIR specifications include the required OAuth2 scopes for enabling security decisions.

Systems SHOULD implement requirements from [FHIR Security Labels](http://hl7.org/fhir/security-labels.html) to provide information on the type and sensitivity of data.

Systems SHALL implement consent requirements per their state, local, and institutional policies. The Business Associate Agreements SHOULD document systems mutual consent requirements. DAF actors SHALL ensure that any necessary consent records exist and are reviewed prior to each exchange of patient-identifiable healthcare information. This verification should be logged in the same manner as other transactions, as discussed above under [General Security Considerations](http://hl7.org/fhir/daf/daf-security.html#general).

## Specification Conclusion

### Compliance requirement

The only mandatory requirement for consent is in US Core Profile.

It is concluded that the overarching conformance should be to the US Realm Core Profile for FHIR which points to SMART on FHIR.

### Resource Ownership

The real difference is in the assumption that the Resource Owner is either the Custodian Organization (sometimes referred to as the Holder of Data) or the Patient. The analysis shows that there are merits to both but that the Custodian Organization is the one liable for breach.

# Acronyms

|  |  |  |
| --- | --- | --- |
| Term | Acronym | Description |
| Custodian Organization Authorization Server | CO AS |  |
| Custodian Patient Consent Authorization Server | CPC AS |  |
| Privacy Protective Service | PPS | Capability to apply obligations resulting from an access control decision to content to modify information to be disclosed. Possible modifications include inserting privacy marks, redaction, masking, and de-identification. |
| Redirection address |  |  |
| Re-directive |  | Patient's directive to Custodian Organization that the Patient's consents are at a Third Party CDMS address. Used by CPC AS to redirect authorization. |
| Security Labeling Service | SLS | Capability to execute rules for assigning security labels to content by applying classification, sensitivity, integrity, compartment, provenance, and handling instructions to disclosed information. |
| Third Party Consent Repository | TPCR |  |
| Third Party Patient Consent Authorization Server | TPPC AS |  |
| Custodian Organization Authorization Server | CO AS |  |
| Security Labeling Service | SLS | Capability to execute rules for assigning security labels to content by applying classification, sensitivity, integrity, compartment, provenance, and handling instructions to disclosed information. |
| Custodian Organization Authorization Server | CO AS |  |
| Custodian Patient Consent Authorization Server | CPC AS |  |
| Third Party Patient Consent Authorization Server | TPPC AS |  |
| Third Party Consent Repository | TPCR |  |

1. Note that this points to the Backend Services Profile [↑](#footnote-ref-1)