

Challenges with FHIR server as a Façade

Lukasz Nosol – Sr. Director Enterprise Clinical Integration and Interoperability at Optum



HL7 FHIR DevDays 2021, Virtual Edition, June 7–10, 2021 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY



PARTNER



Who am I?

- Lukasz Nosol
- Sr. Director - Enterprise Clinical Integration and Interoperability at Optum
- Leading development FHIR based solutions:
 - POCA – Point of Care Assist[®] – payer agnostic EHR point of care services
 - Optum[®] Interoperability and Patient Access Services – CMS Compliance Platform for payers



Challenges with FHIR server as a Façade in the context of CMS Patient Access Rule

FHIR server Façade vs Repository

FHIR Façade based server acts as a gateway to the data. Data is stored in non-FHIR, frequently proprietary format, often in many systems downstream from the gateway. FHIR Façade receives REST based FHIR requests and translates them to downstream native call, fetches data from an appropriate system, converts it to FHIR resource model on the fly and returns it to the client. Typically, best suited for read-only (i.e. query) use cases.

FHIR Repository based server stores data internally, often directly as FHIR resources. It receives REST based FHIR request and retrieves the FHIR resource directly from its repository returning it to the client. Typically, best suited for read and write use cases.

1. Ever Changing Implementation Guides
2. Logical IDs
3. Member Identity
4. Data sensitivity
5. Data Access via Façade Pattern

1. Ever Changing Implementation Guides

1 Ever changing Implementation Guides

- **Challenge** - [CMS Recommended](#) Implementation Guides were being update throughout the year
 - US Core was the most stable one
 - Significant changes as late as October 2020
 - CARIN BB profiles changes
 - DaVinci Plan Net implementation clarification
 - Every Implementation Guide had different style
- **Resolution** – Intermediate releases of the platform had to incorporate partially complete Implementation Guide

2. Logical IDs

2. Logical IDs

➤ Problem:

- Logical IDs are easy with Repository Pattern
- Logical IDs can be very tricky with Façade Pattern

Each resource has an `id` element which contains the "logical id" of the resource assigned by the server responsible for storing it. Resources always have a known logical id except for a few special cases (e.g. when a new resource is being sent to a server to assign a logical id in the `create` interaction). The logical id is unique within the space of all resources of the same type on the same server. **Once assigned by the server, the id is never changed.**

- Surrogate key may not be feasible
- Composite key may not be FHIR friendly

Logical ids (and therefore locations) are case sensitive. Logical Ids are always opaque, and external systems need not and should not attempt to determine their internal structure. A logical id SHALL always be represented in the same way in resource references and URLs. **Ids can be up to 64 characters long,** and contain any combination of upper and lowercase ASCII letters, numerals, "-" and ".".

- **Resolution** – A mechanism to track composite IDs to logical ID mapping is needed

3. Member Identity

3. Member Identity

- **Problem:** Member mediated access requires member's identity to be verified which can be tricky with Façade Pattern
 - Different plans within payer may have different IDPs, but want to share data via the same set of FHIR server endpoints
- **Resolution:** Let member pick a medical plan
 - a) In 3rd party App before the launch
 - b) During authorization after the launch

4. Data sensitivity

4. Data Sensitivity

- **Problem:** Certain data attributes may be required by Implementation Guides, but may be considered sensitive from payer's perspective, for example
 - (Patient) Logical IDs
 - (Patient) Identifiers
- **Resolution**
 - Obfuscate – Hash vs Encrypt vs. ???
 - If Hashed, needs to be tracked in the FHIR Façade
 - If Encrypted, watch out for the size of data or allowed characters

5. Data Access via Façade Pattern

5 Data Access via Façade Pattern

- **Problem:** Every Implementation Guide has a list of SHALL query parameters
 - Downstream system need to support it, but what if it doesn't?
- **Resolution:**
 - Implement required searches in downstream system
 - Trade off: Time to delivery i.e. cost
 - Get all data THEN filter locally
 - Trade off: Performance implications

Contact

- During DevDays, you can find / reach me here:
 - Via Whova App – Speaker’s Gallery
 - Email: Lukasz.Nosol@UHC.com
 - <https://www.linkedin.com/in/lukasznosol/>

Q&A

ORGANIZED BY



PARTNER

