READ FIRST: ETL Design Decisions

Baseline spec:

- HL7’s Genomics Reporting Implementation Guide: Home Page - Genomics Reporting Implementation Guide v4.0.0-ballot
  - FHIR design to support genomics reporting
  - Provides profiles = flavors of FHIR resources to account for additional datafields and connections between resources + better semantics for communication
  - The guide is not stringent, it leaves a lot of flexibility around the design of the FHIR data model
  - Based on the Profile descriptions, it advocates for data redundancy or perhaps it is telling us all the different resources we can store a data field.
  - It doesn’t cover the mapping of all the ZD data fields
    - FHIR’s Observation resource uses components to represent each data field.
      - This provides us the flexibility of having as many datafields as we want. This potentially means we can store all datafields as a single resource. The only caveat, is that all parties must agree to System and code for each Component for interoperability.

Mapping ZD fields to FHIR

To see what ZD fields were mapped to FHIR: FHIR Molecular Design Decisions
To see all ZD data fields with all the system:code details:
https://childrenscancerinstitute.sharepoint.com/:x:/r/sites/CompBio/_layouts/15/Doc.aspx?sourced[...]ion=default&mobileredirect=true&DefaultItemOpen=1

Problem: implementing MolecularSequence to represent Gene

- MolecularSequence Entry = Gene - can only reference one Variant observation. This is bad because we want a gene to link to many variants.
- However, Variant Observation can link to many genes. This is helpful for things like SV, because the SV start and end involves two genes. However, there is no way to determine

which gene is first and which is last.

- Hence FHIR’s implementation of molecularSequence Resource is insufficient for our use case.

Resolution: add all molecularSequence(Gene) info into each Variant Observation (Observation of a Variant)

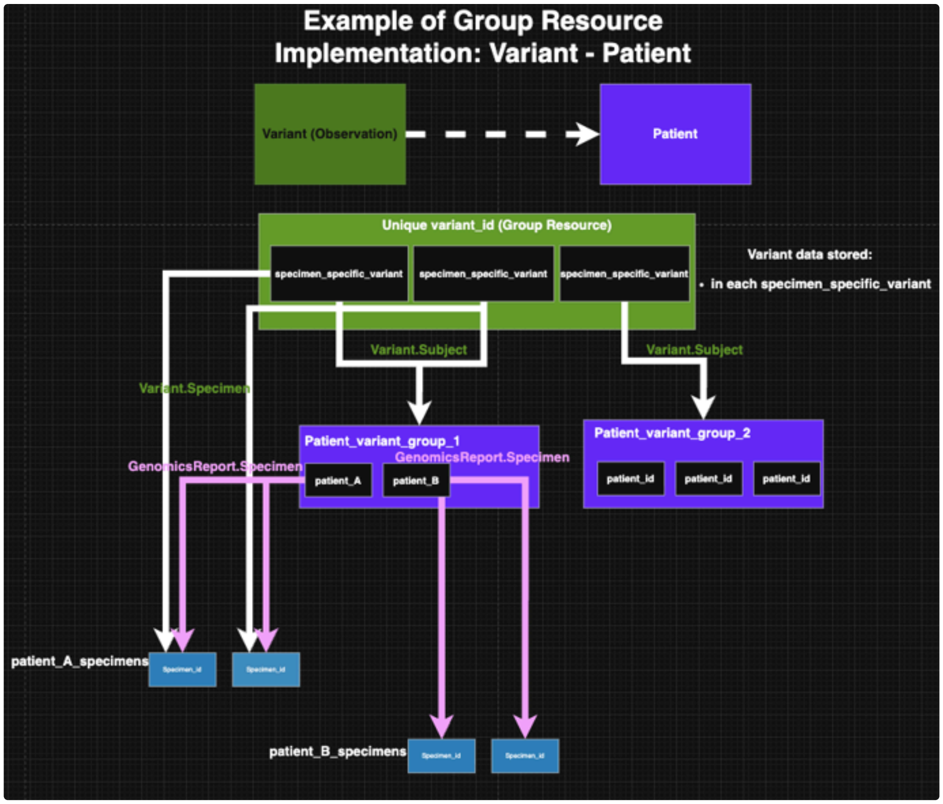
- This is bad because of data redundancy = pagination performance and future data change implementations
- We cannot make Observation Resource become a unique Variant because it cannot be linked to multiple biosamples and patients.
- see FHIR design diagram: https://app.diagrams.net/#G1moMtAxJczICbXLpJiGuIt5HqTa1NMFHu#{"pageId"%3A"4XJpWMBPGZUvkoF6CKvT"}

Problem: single source of truth vs data redundancy ❓

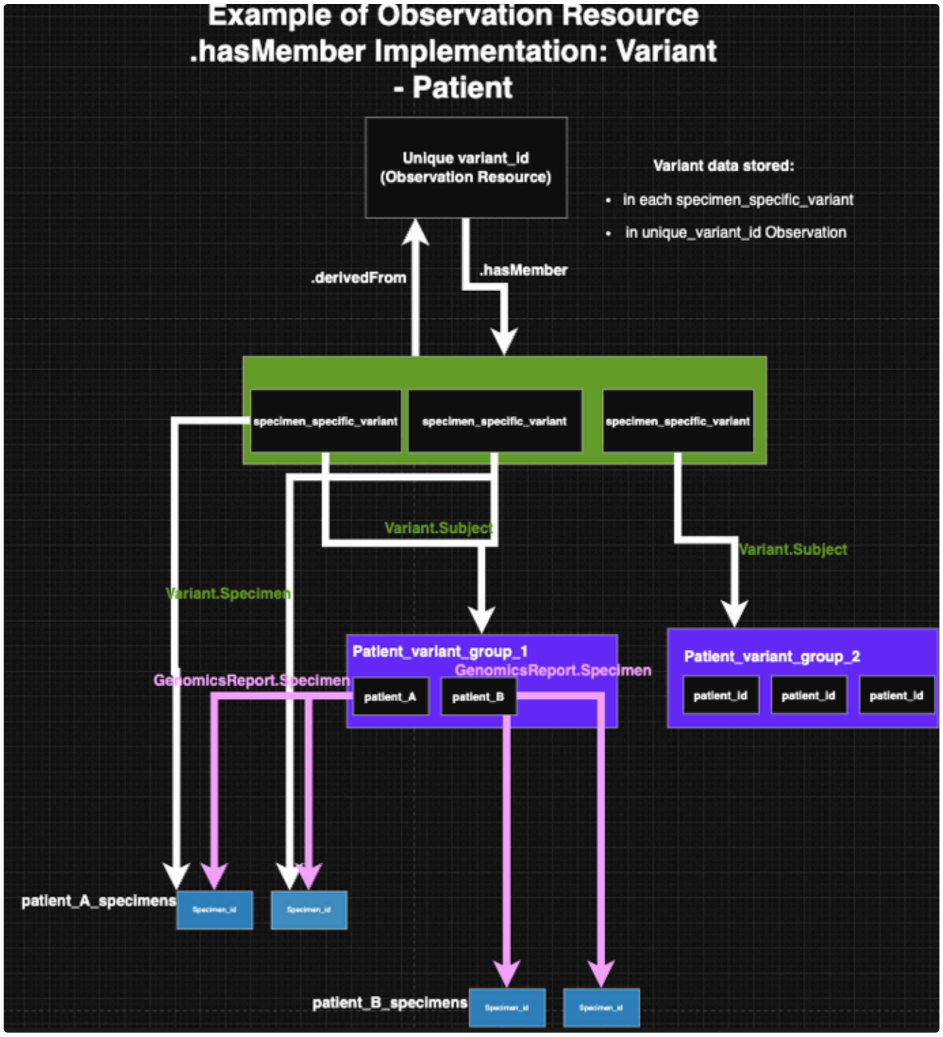
context: implementation of Gene and Variant and how they reference patients and biosamples.

Current solution is data redundancy.

- a single source of truth is a Resource that holds all the info on an object. It is unique and if referenced or references other Resources. E.g. Gene or Variant.
    - Problematic:
        - FHIR cardinality does not support this design
            - Observation of a variant cannot be linked to multiple biosamples and patients.
        - Forcing resources to adhere to this design makes:
            - references messy (Observation.focus field) = reference field contains all types of resources (against the implementation guide and FHIR documentation), rather than one or two resource types
            - Or makes the FHIR design very messy (Implementation of Group Resources or Observation.hasMember data field, check the top right corner of the design diagram) and difficult to navigate in the data aggregation layer
                - Diagram of Group FHIR designs:



- Diagram of .hasMember Implementation as alternative to Group:



- data redundancy - the same info about an entity is duplicated across all resources that mention the entity. Each Variant will have a unique uuid, but share a ZD variant_id.
  - e.g. All variants contain the gene name and gene details, rather than referencing a single gene Resource
    - This could make updating info on genes or variants difficult
    - increases paging of FHIR results
    - keeps the design simple

Problem: Handling the storage of gene information in SV variant. ❓

- context = data redundancy approach to FHIR design
- status quo: SNV and CNV variants currently hold data on a single gene as identifiers (alias, names) and components (position)

Solutions

- maintain status quo: ✅

    - store 2 genes in SV as identifiers and components.
        - problems:
            - representing start and end gene using system(URI) means that SV will have its own gene URI that is separate from the other variants (using LOINC URI+code in accordance to implementation guide) = data aggregation layer problem
- reimplement design approach to single source of truth
    - variant can reference to more than one gene (MolecularSequence Resource)
    - otherwise, store start_gene_id and end_gene_id as identifiers and use that info to infer the connection the Gene resource = 2 queries for linking SV to genes

Problem: codeableConcepts use system(URI) ✅

- used FHIR documented URI whenever possible, otherwise we just call our URI= ZD

Problem: lack of FHIR fields to support our fields ✅

- FHIR supports < 50% of our fields in Observation Resources = Variants
- Current Solution:
    - Observation resources are made up of components = fields that represent observed data
    - We just made many components in the Observation resource to support the ETL

Problem: handling ZD Null data ✅

FHIR Observation components do not support Null data fields

- Potential solutions:
    - Not create the component
- Current solution
    - create the component but set it with data-absent-reason = unknown
        - Valueset-data-absent-reason - FHIR v6.0.0-ballot3

FHIR Resources that have no data to store

- It is possible for a resource to exist but not have any non-null data in it.
- Current solution: just don’t make the resource and reference to that resource