HL7® FHIR® DevDays

JUNE | 2023

How to integrate FHIR with your analytics pipeline

Piotr Szul, CSIRO (Australia)

HL7 FHIR DevDays 2023 | Hybrid Edition, Amsterdam | June 6–9, 2023 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely | HL7 International

Who am I?

- Piotr Szul
- Senior Engineer @ CSIRO
  - AEHRC (Australian e-Health Research Centre)
  - FHIR Terminology and Tooling Team
- Three years of experience developing tools enabling FHIR based analytics
- Many more years of developing tools for analyzing research data

Learning Objectives

- Learn how to
  - build a simple, automated, batch analytics pipeline for FHIR data using open-source software, including Apache Spark and Pathling.
  - use FHIRPath and SQL to transform complex FHIR data into simpler, use-case centric tabular views
  - to tackle some of the complexities of working with FHIR data, including: codes and terminology, Questionnaires, and extensions

Agenda

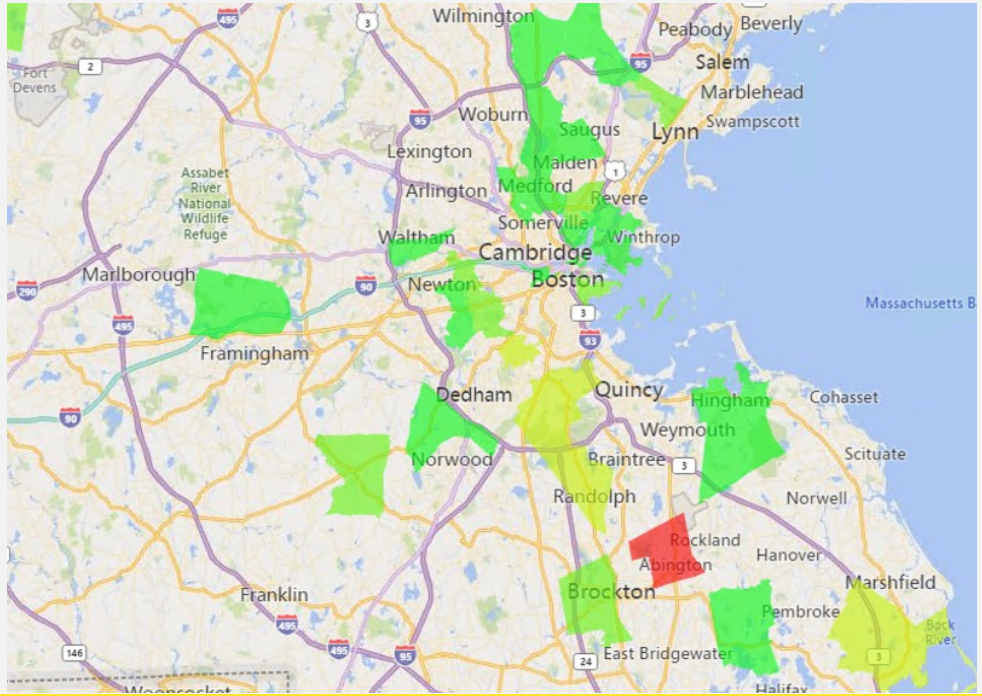
- FHIR analytics pipeline
  - Introduce the analytics use case
  - Discuss the solution pipeline and the technologies used
  - LAB: Demonstrate the implementation of the pipeline
- Additional FHIR specific topics
  - Extensions and recursive structures
  - LAB: Demonstrate how to incorporate data above

https://github.com/aehrc/fhir-analytics-pipeline

https://tinyurl.com/ydnax4as

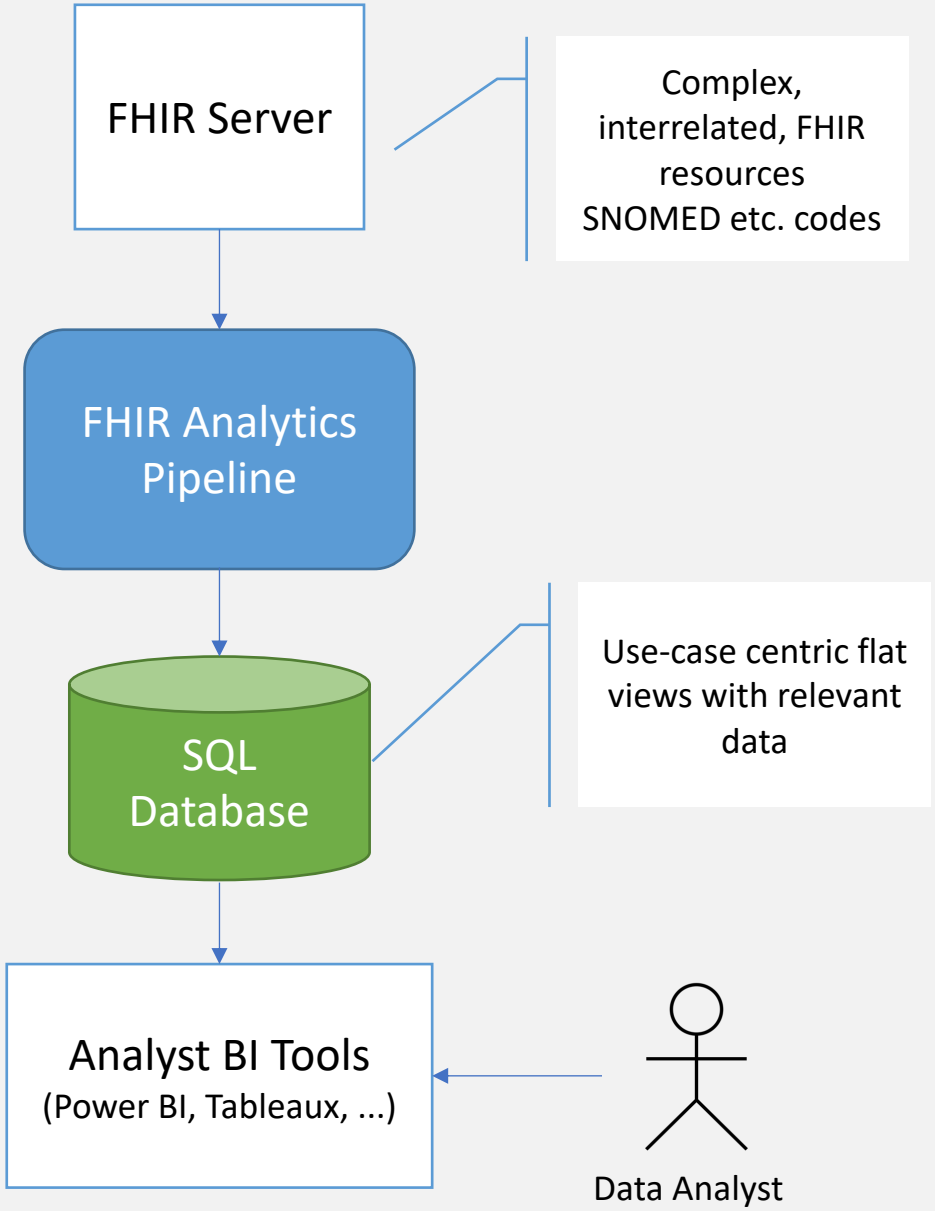
Self-service analytics for COVID-19 management

- The objective is to allow data analysts with BI tools leverage FHIR data for analytics
- COVID-19 management, e.g.: the number of unvaccinated patients by location stratified by a customizable risk score
  - Based on ”Pathling: analytics on FHIR” DOI:10.1186/s13326-022-00277-1

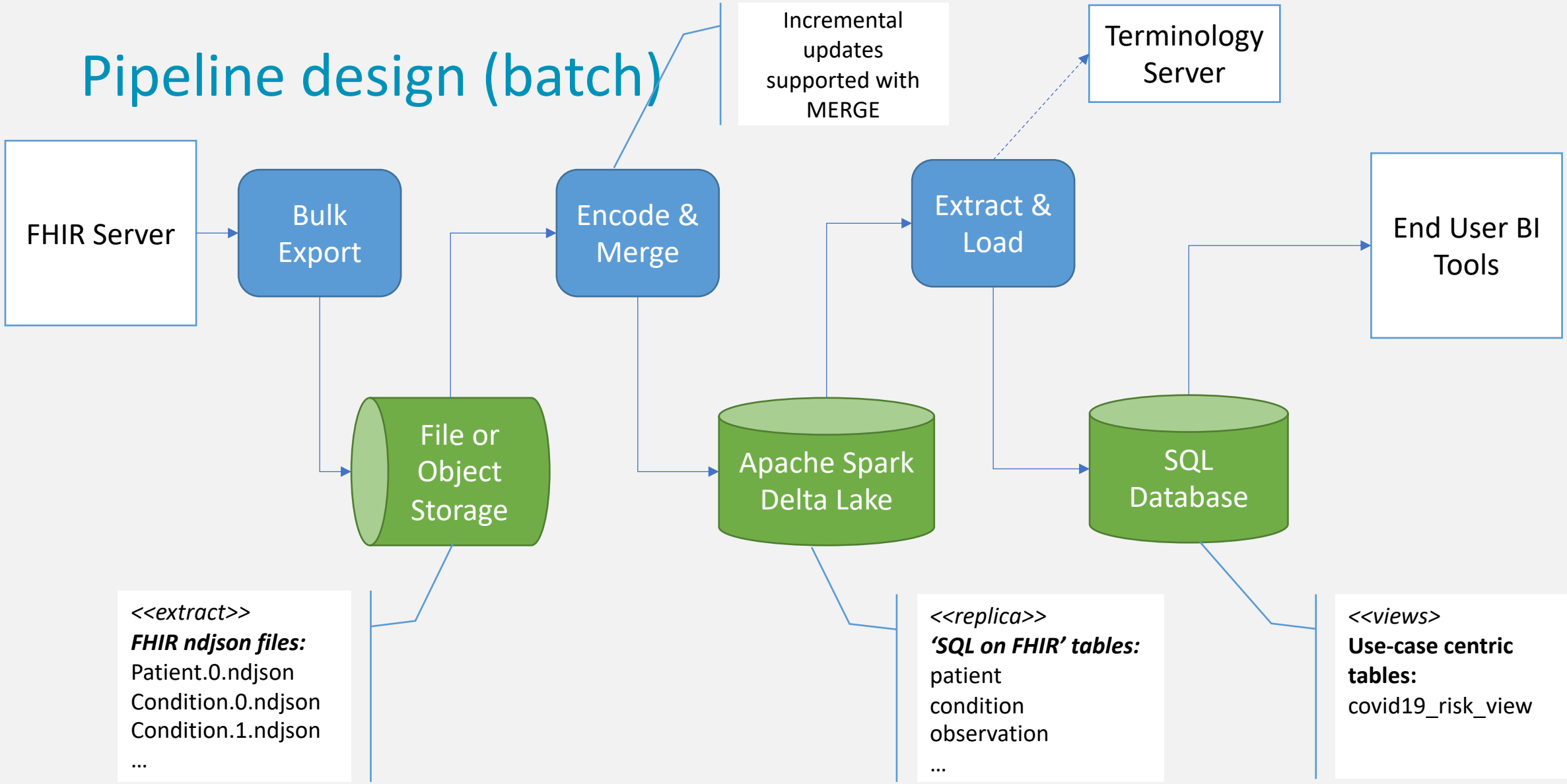


The scenario: Details

- COIVD-19 centric data view
  - Patient data
    - DOB
    - Postal Code
  - COVID-19 Immunization status
  - Risk factors:
    - Diagnosed with chronic kidney disease
    - Diagnosed with heart disease
    - Recorded body mass index (BMI) greater than 30
- Flexibility (extend to other use-cases)



Pipeline design (batch)



FHIR mapping to SQL

”SQL on FHIR”
work in progress

Patient	N		DomainResource
identifier	Σ	0..*	Identifier
active	?! Σ	0..1	boolean
name	Σ	0..*	HumanName
telecom	Σ	0..*	ContactPoint
gender	Σ	0..1	code
birthDate	Σ	0..1	date
deceased[x]	?! Σ	0..1	
deceasedBoolean			boolean
deceasedDateTime			dateTime
address	Σ	0..*	Address
maritalStatus		0..1	CodeableConcept

FHIR Structure

```
root
 |-- id: string

 |-- active: boolean
 |-- name: array
 |     |-- element: struct
 |     |     |-- id: string
 |     |     |-- use: string
 |     |     |-- family: string
 |     |     |-- given: array
 |     |     |     |-- element: string

 |-- gender: string
 |-- birthDate: string
 |-- deceasedBoolean: boolean
 |-- deceasedDateTime: string
 |-- address: array
 |     |-- element: struct
 |     |     |-- id: string
```

(Spark) SQL Schema

The implementation

- Technologies
  - Apache Spark, Spark SQL, PySpark,
  - Pathling and Ontoserver (for terminology queries)
  - ndjson, parquet/delta
  - Python, SQL, FHIRPath
- Databricks (for convenience)
  - Spark cluster for execution of the pipeline code
  - Notebooks for exploratory examples
  - Data flow orchestration + SQL Analytics and reporting

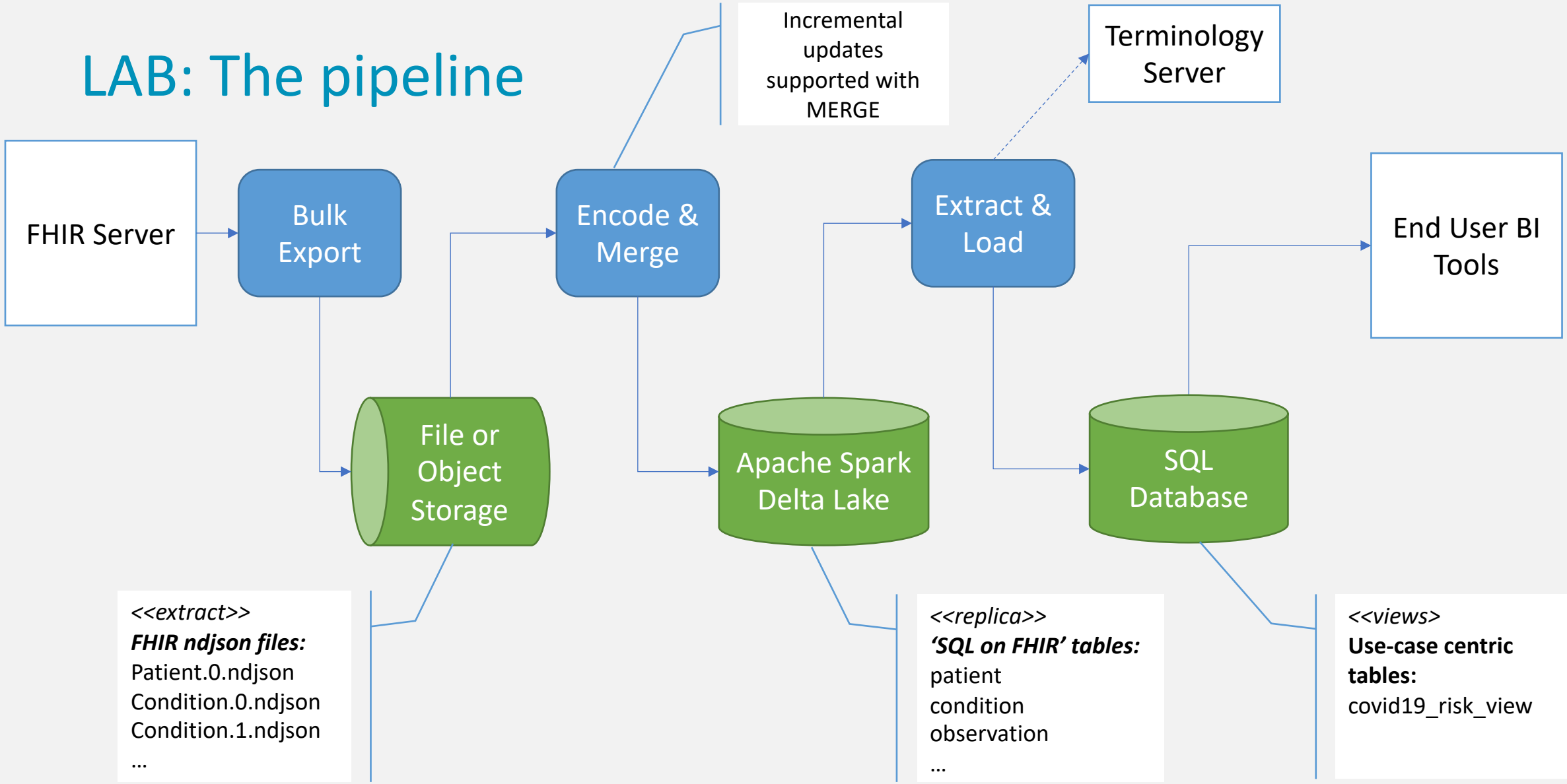
The pipeline can be deployed on any Apache Spark enabled platform (AWS, Azure, …)

Pathling@AeHRC

Pathling is a set of open source tools facilitating the use clinical terminology and FHIR® within health data analytics.

- Built on Apache Spark for scalability
- Python, Java/Scala libraries and FHIR REST API
- Relevant features
  - Encoding of FHIR ndjson resources to Apache Spark delta tables
  - Extracting tabular views from FHIR data with FHIRPath expressions
  - Support for terminology queries against coded fields within the FHIR data from SQL and FHIRPath
- See: https://pathling.csiro.au/

LAB: The pipeline



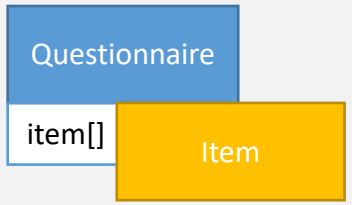
Recursive nested types

- Recursive nested structures
  - E.g. `item` in Questionnaire
  - In principle allows for unlimited and undetermined levels of nesting
- Pathling solution
  - Allow for limited predetermined level of nesting
  - ***max_nesting_level*** parameter applied globally to encoding of all recursive nested structures (except for extensions)

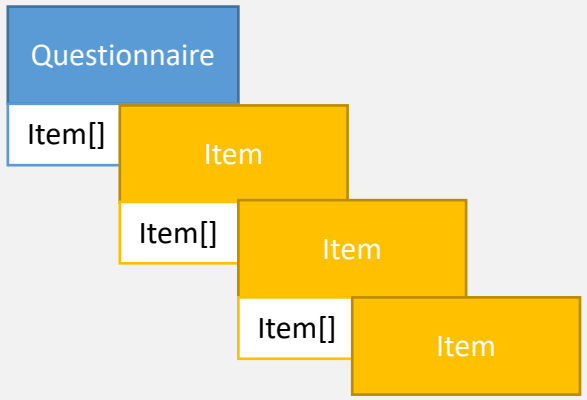
Structure

Name	Flags	Card.	Type
Questionnaire	TU		DomainResource
url	Σ C	0..1	uri
item	C	0..*	BackboneElement
linkId	C	1..1	string
valueReference			Reference(Any)
item		0..*	see item

max_nesting_level = 0



max_nesting_level = 2

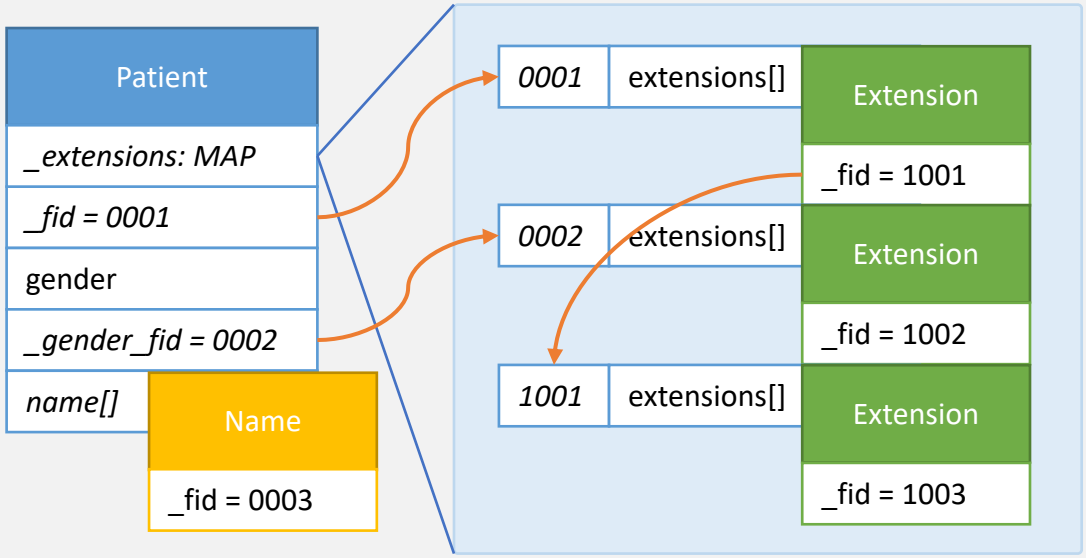


Extensions

- Challenges
  - Every resource or datatype element may include one or more "extensions"
  - Complex extensions contain one or more nested extensions
  - value[x] can be of 50+ types
- Pathling solution
  - Annotate each field and and element with unique id and store extensions in a single MAP column indexed by the id
  - Constraint the types allowed as values (***enabled_open_types***)

Structure

Name	Flags	Card.	Type
Extension	N		Element
url		1..1	uri
value[x]	C	0..1	*



LAB: Questionnaires and Extensions

- Accessing Questionnaire using FHIRPath (and SQL)
- Accessing extensions FHIRPath (and SQL)

Pathling `extract()` operation

Create tabular views/extracts from FHIR data:

- define the columns as FHIRPath expressions
- filter resources with FHIRPath

```
{
    'id':'1',
    'name'[
        { 'given': 'Benjamin', 'family':'Franklin'},
        { 'given': 'Silence', 'family':'Dodood'}
    ]
},
{
    'id':'2',
    'name'[
        { 'given': 'Isaac', 'family':'Asimov'},
        { 'given': 'Paul', 'family':'French'}
    ]
}
```

```
extract('Patient',
  columns = [
    'name.given',
    'name.family'
  ],
  filters = [])
```

Given name	Family name
Benjamin	Franklin
Silence	Dogood
Isaac	Asimov
Paul	French

Smart Unnesting
only the combinations that exists in the FHIR data are created

Given name	Family name
Benjamin	Franklin
Benjamin	Dogood
Silence	Franklin
Silence	Dogood
Isaac	Asimov
Isaac	French
Paul	Asimov
Paul	French

What did you learn?

- How to
  - use Apache Spark and Pathing to encode FHIR data into delta tables optimized for large scale SQL queries
  - use FHIRPath and SQL to transform FHIR data into tabular views :
    - simple, complex data types, Quantities
    - data in related resources
    - codes and terminology functions
    - extensions and nested structures.
  - create an batch ETL pipeline that automates the extraction of data from a FHIR server and loading of the relevant data to the end user analytical database.

Databricks is just one of many possible deployment platforms.

Contact

- During DevDays, you can find / reach me here:
  - Via Whova App – Speaker's Gallery
  - Email: piotr.szul@csiro.au
- Github:
  - The Pipeline: https://github.com/aehrc/fhir-analytics-pipeline
  - Pathling: https://github.com/aehrc/pathling
- Related sessions:
  - Wed 10:15 – 11:00: The FHIR Bulk Data API and what's new in v2!
  - Wed 15:45 – 16:30: SQL on FHIR 2.0 BOF
  - Wed 15:45 – 16:30: Let's Build! Hands-on FHIRPath

Q&A

Feedback?

devdays.com/feedback

https://www.devdays.com/feedback/

ORGANIZED BY

firely | HL7® International