



Reconciliation of FHIR Data With Incoming Data

Roman Polyanovsky - Google Cloud



HL7 FHIR DevDays 2020, Virtual Edition, November 17–20, 2020 | @FirelyTeam | #fhirdevdays | www.devdays.com/november-2020

ORGANIZED BY **firely**

Who am I?

- Roman Polyanovsky
- Software Developer



Google Cloud



What is this talk about?

- This talk is intended as a **high level discussion**
 - Primarily of challenges of **merging streams of data** into FHIR **at scale**
 - We'll also discuss **ideas for solutions** to these challenges as well as their **gaps and pitfalls**

The world in FHIR

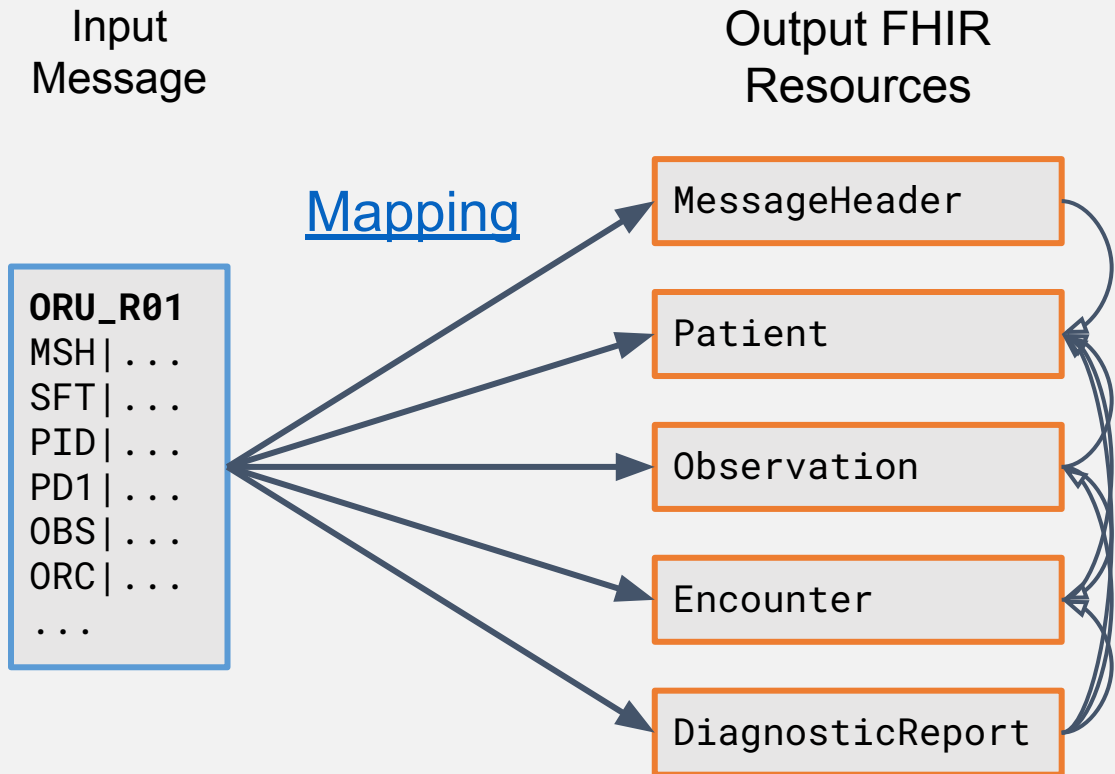
- FHIR represents a complete state of the world
- Other formats (**HL7v2** for example) **are not a complete state**
 - They are **transactional/event based**
 - They represent a **continuous stream** of updates

For example: HL7v2 → FHIR

Input
Message

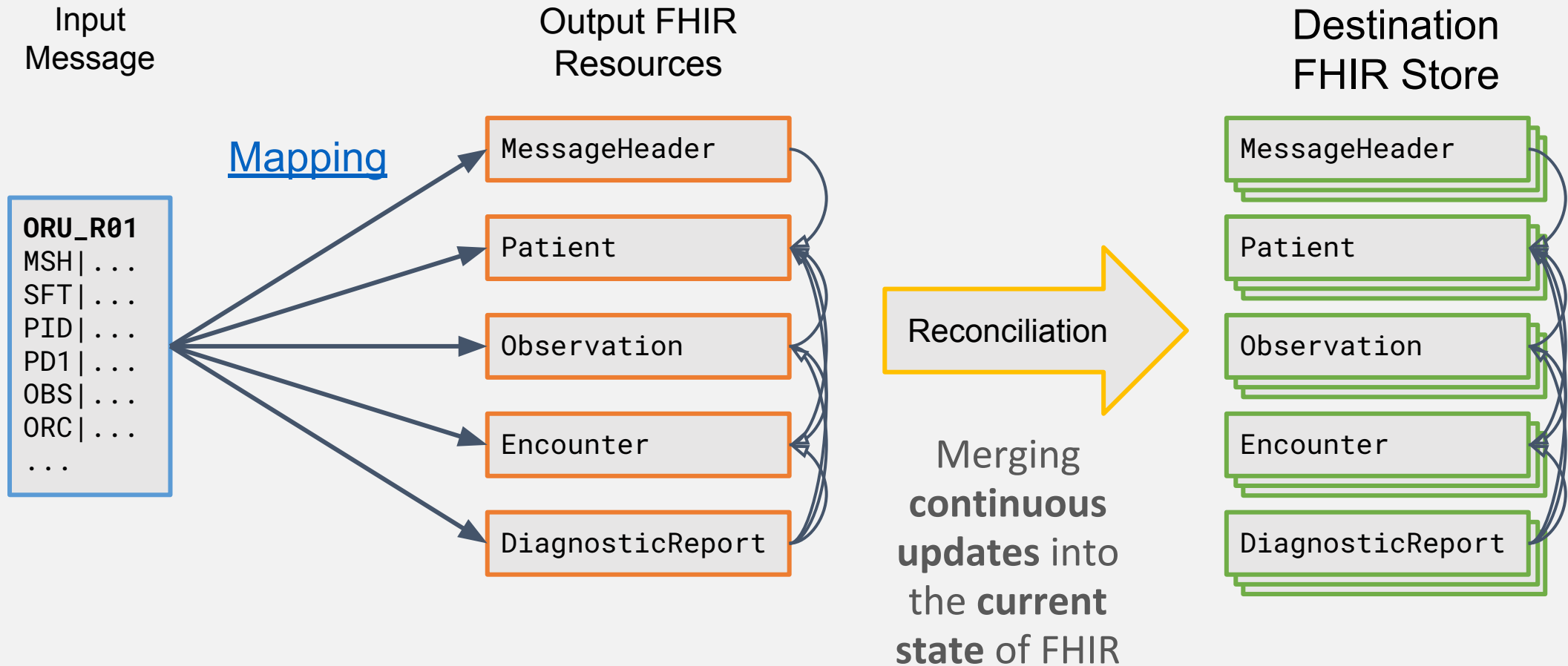
```
ORU_R01  
MSH|...  
SFT|...  
PID|...  
PD1|...  
OBS|...  
ORC|...  
...
```

For example: HL7v2 → FHIR



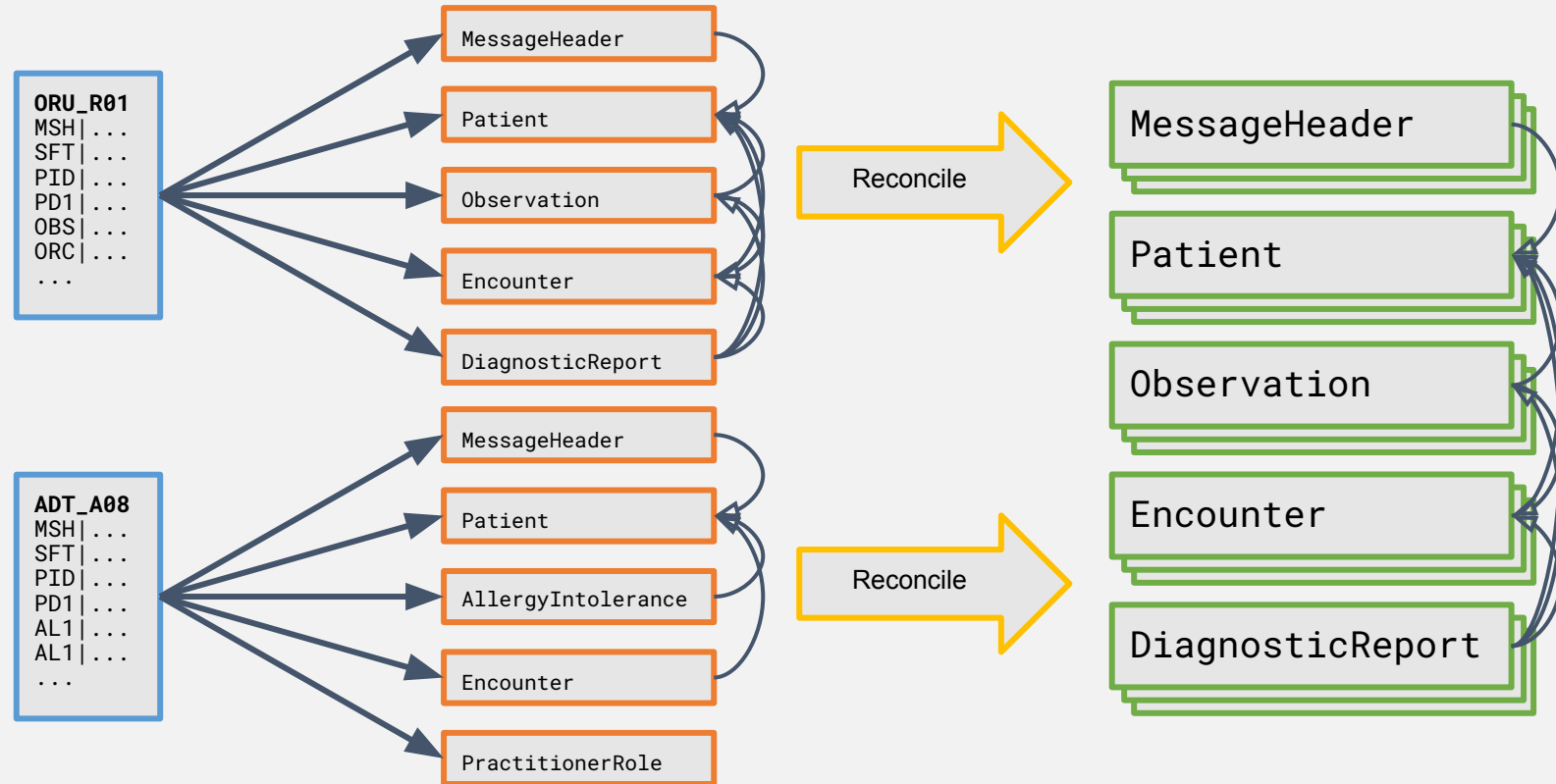
github.com/GoogleCloudPlatform/healthcare-data-harmonization

For example: HL7v2 → FHIR



Challenges

- **Concurrency** - many streams (maybe different formats), no synchronicity



Challenges

- **Concurrency** - many streams (maybe different formats), no synchronicity
- **Out of Order Updates** - (maybe severely) delayed updates or backfills



Challenges

- **Concurrency** - many streams (maybe different formats), no synchronicity
- **Out of Order Updates** - (maybe severely) delayed updates or backfills
- **Maintaining accurate history** - must avoid polluting FHIR store

Challenges

- **Concurrency** - many streams (maybe different formats), no synchronicity
- **Out of Order Updates** - (maybe severely) delayed updates or backfills
- **Maintaining accurate history** - must avoid polluting FHIR store
- **Arbitrary merge logic** - Merges spanning many resources with bespoke merging requirements

Concept Architecture

Using an intermediate store or "staging" area

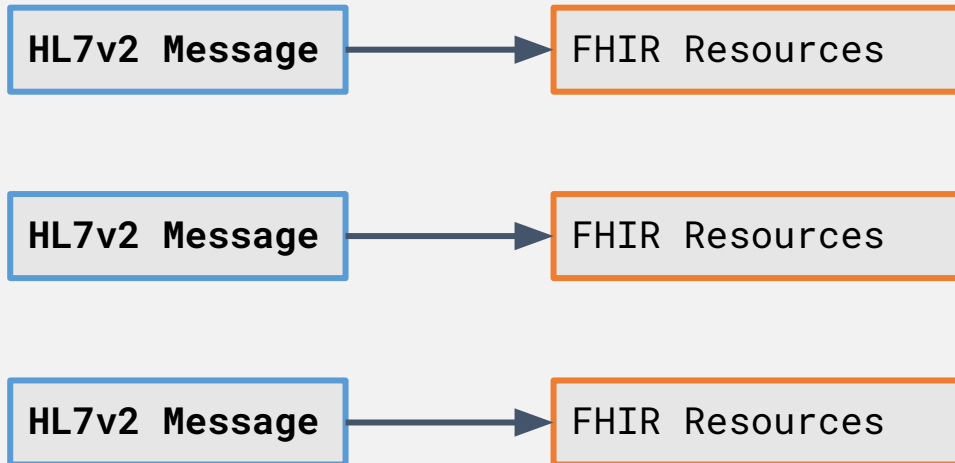
HL7v2 Message

HL7v2 Message

HL7v2 Message

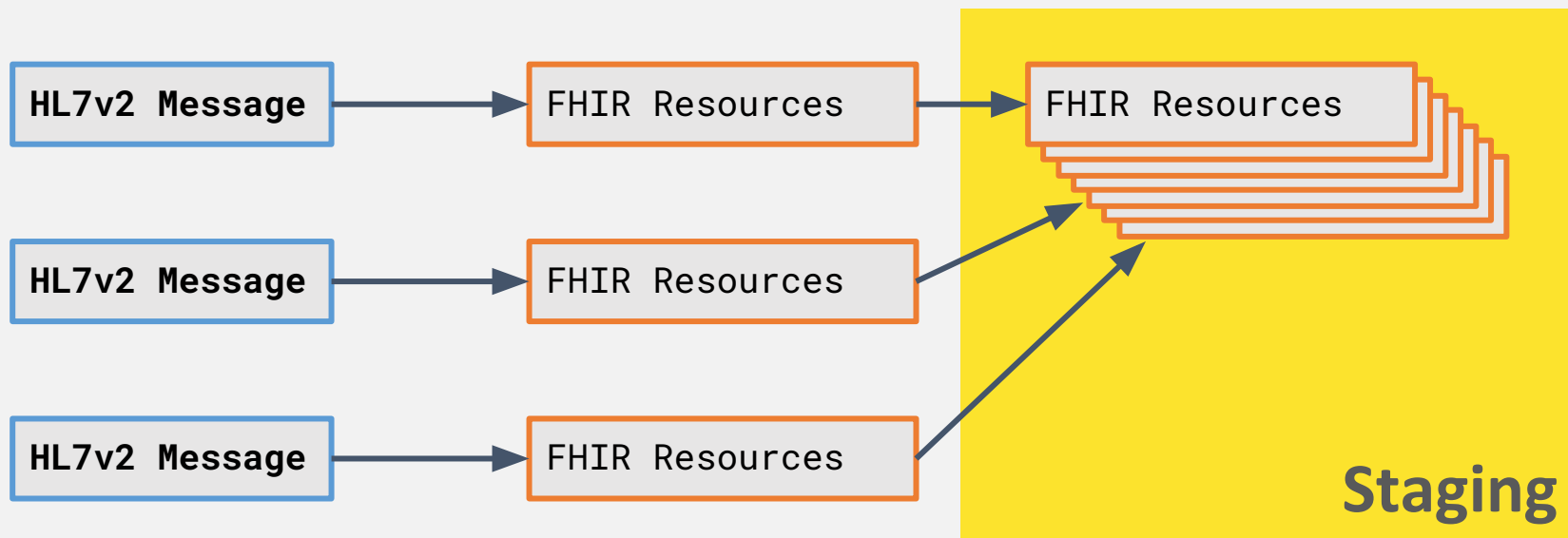
Concept Architecture

Using an intermediate store or "staging" area



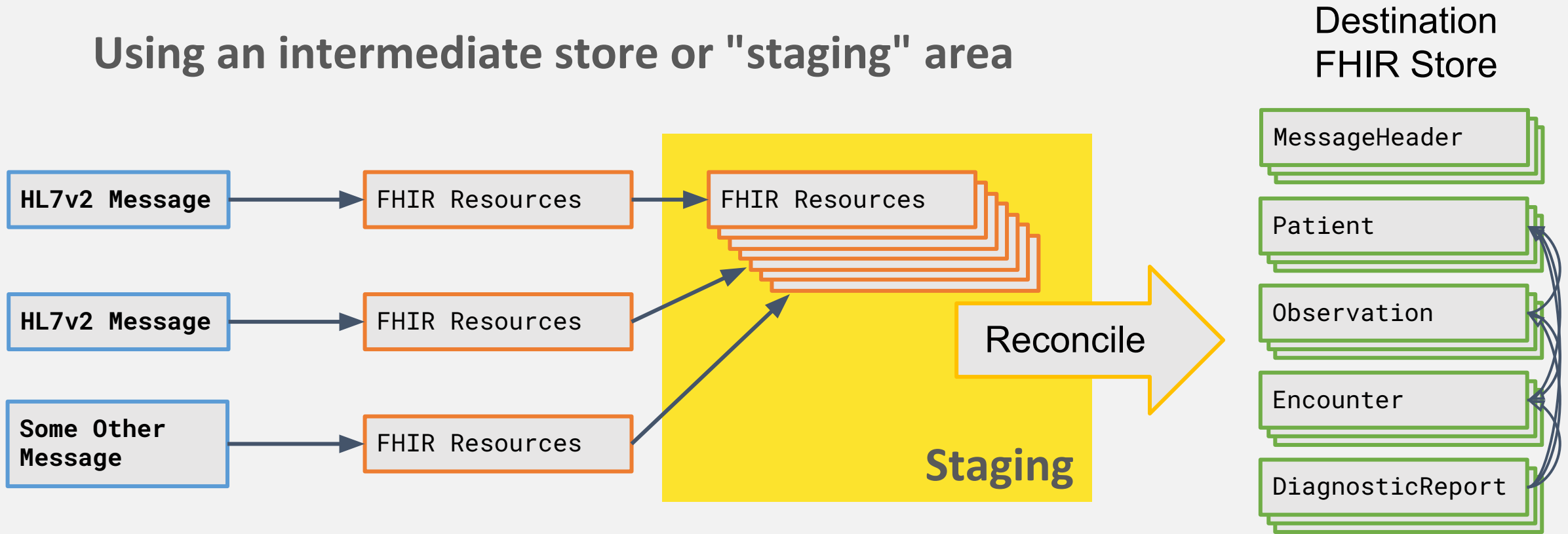
Concept Architecture

Using an intermediate store or "staging" area



Concept Architecture

Using an intermediate store or "staging" area

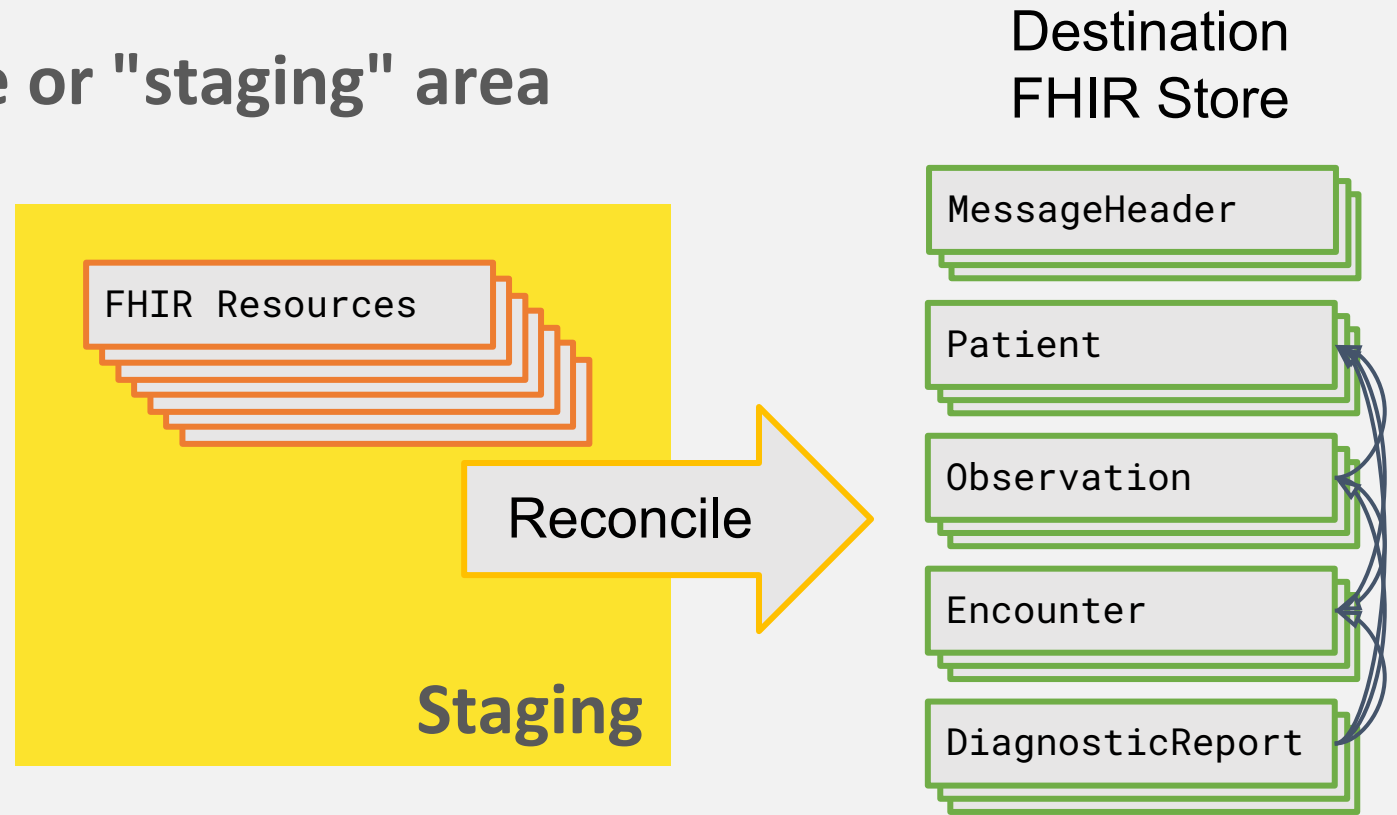


Concept Architecture

Using an intermediate store or "staging" area

- **Concurrency**

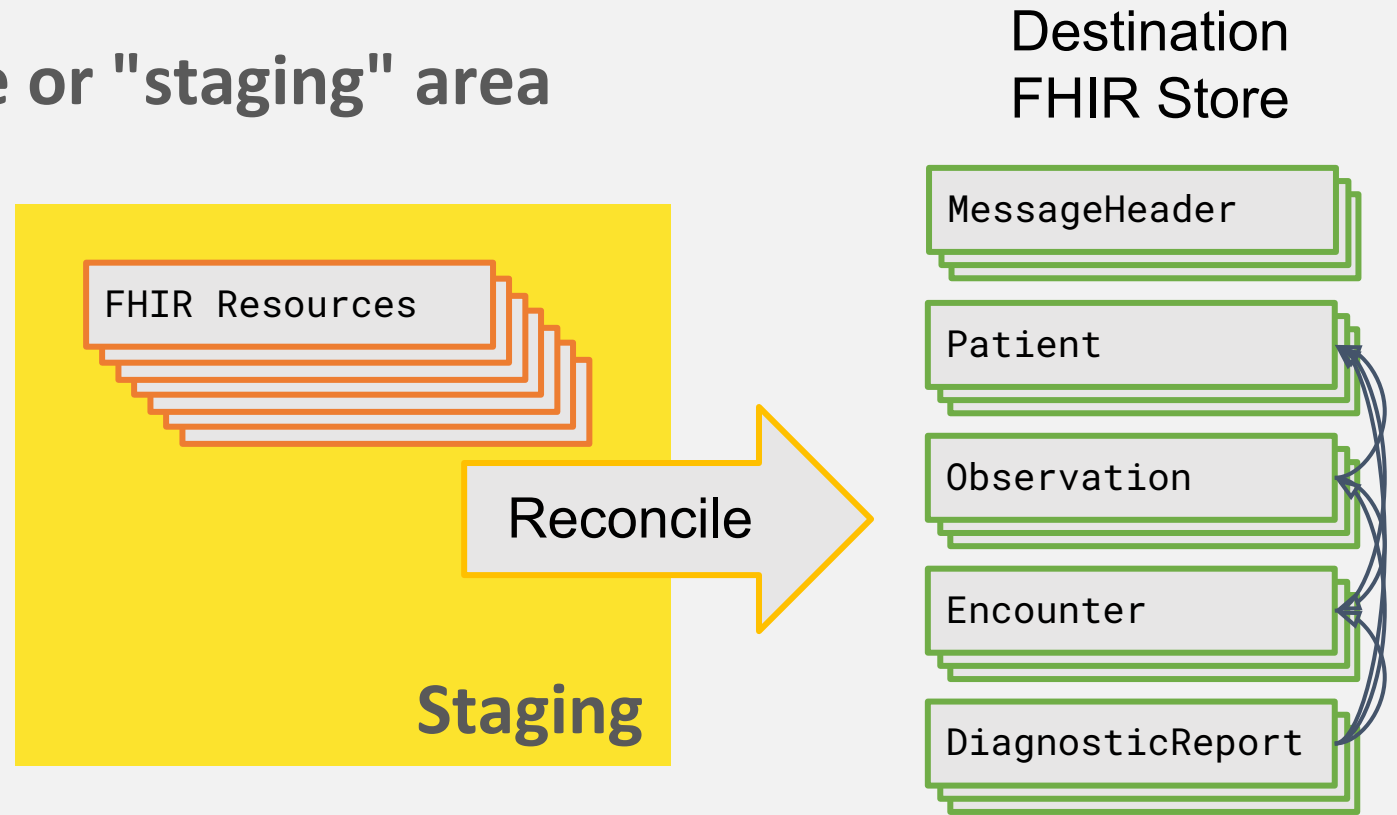
- Staging area is just full of immutable resources
- Mapped messages are just loosely dropped in
- **Gap: Need to execute searches/several fetches to find groups of resources for reconciliation**
 - Reverse linking can help with this, as well as maybe clever labelling
 - Search by extension would also help



Concept Architecture

Using an intermediate store or "staging" area

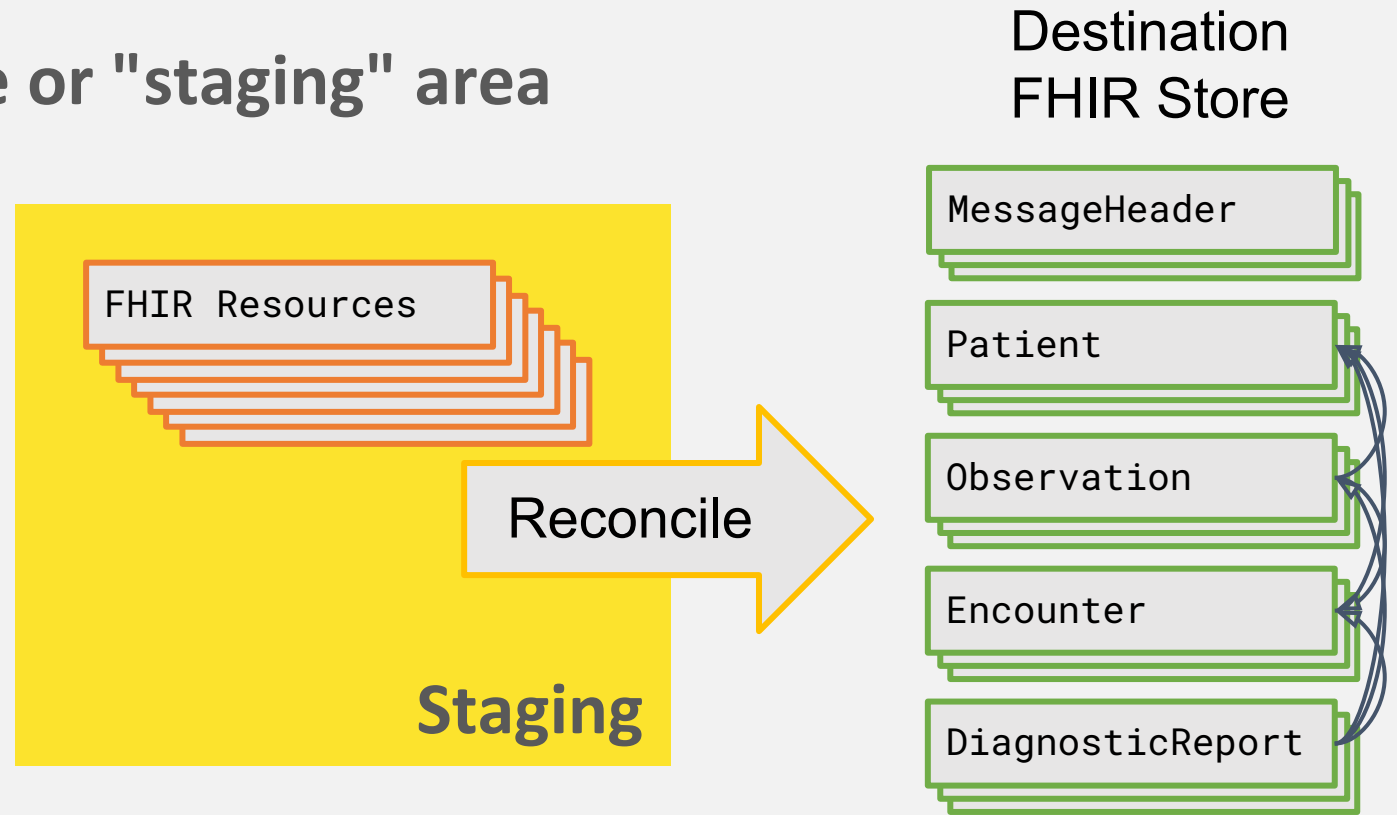
- **Out of Order Updates**
 - Resources can be (re)sorted upon reconciliation and (re)merged
 - **Pitfall: Need to be careful about picking the ordinal to sort by**
 - Links to a timestamped MessageHeader (can even duplicate the timestamp across all resources during mapping)



Concept Architecture

Using an intermediate store or "staging" area

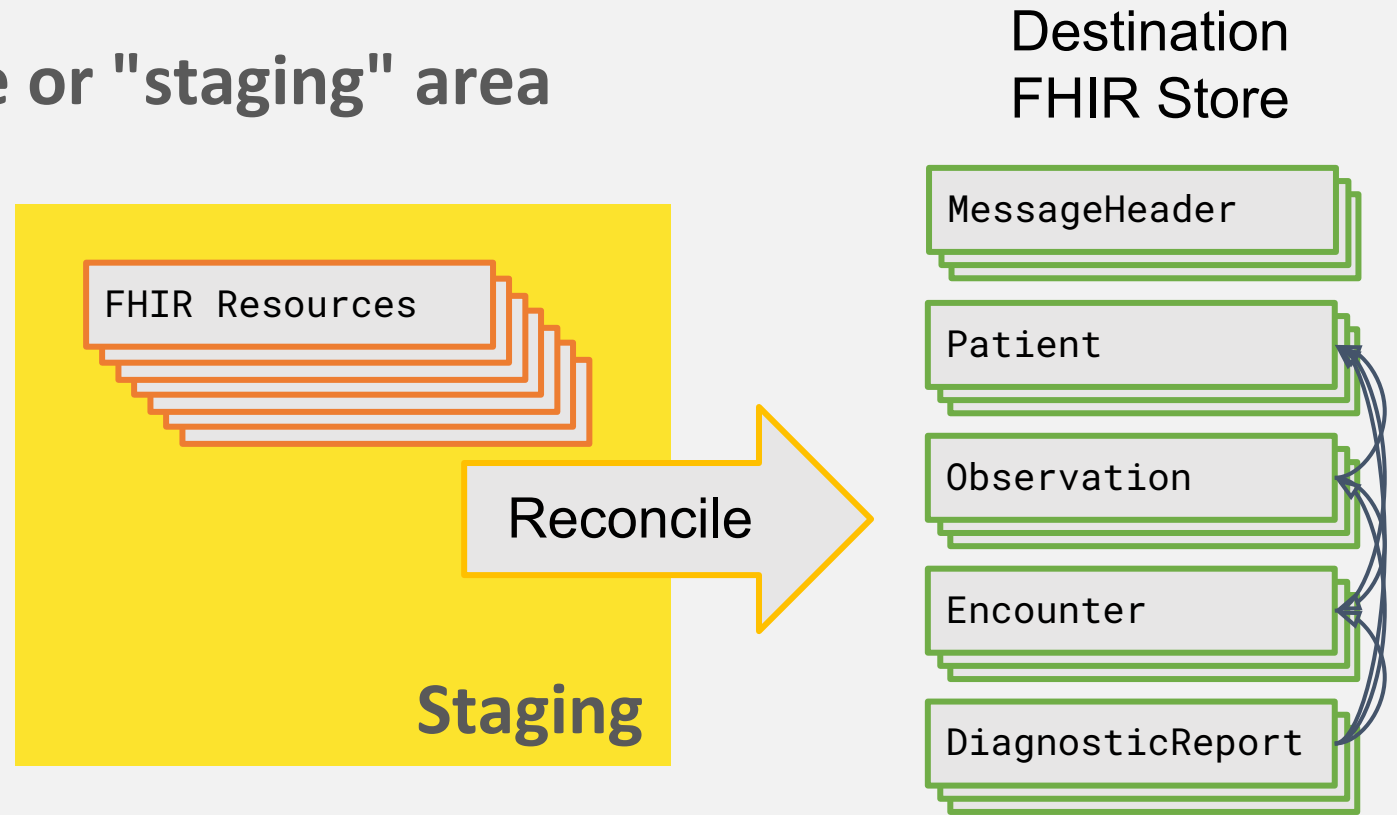
- **Maintaining History**
 - Sorted + grouped resources from staging can paint an accurate history
 - **Gap: FHIR history is immutable in the Destination Store***
 - Make it mutable or use Staging/Other store for historical queries.



Concept Architecture

Using an intermediate store or "staging" area

- **Arbitrary merge logic**
 - The Reconciliation service can pull all necessary resources from the intermediate store, and implement merge logic bespoke to each group
- **Gap: May need to fetch (too) many resources from staging**
 - Remedy with "snapshots" and Container resources



Conclusions

- Reconciling FHIR resources is complicated by
 - Concurrent streams/race conditions
 - Out of order messages
 - Merging instead of overwriting (e.x. HL7v2 omitted fields: . . . | | . . .)
 - Searching/querying limitations
- An intermediate staging area/store simplifies some of these problems by
 - Introducing immutability
 - Allowing sorting and resorting of messages
 - Maintaining a complete history of individual events

The world in FHIR

-

Contact

- During DevDays, you can find / reach me here:
 - Via Whova App – Speaker's Gallery
 - If I can figure out what this is or how it works
 - **Email: romanpoly@google.com**

Q&A