HL7® FHIR® DevDays

JUNE | 2022

Let's Build: HAPI FHIR MDM

Ken Stevens, Smile CDR

HL7 FHIR DevDays International 2022 | Hybrid Edition, Cleveland, OH | June 6–9, 2022 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely | HL7 International

Creative Commons

- This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
- All code samples are available on GitHub in fully working form

smile CDR

Presenter

Ken Stevens

Director Core Development, Smile CDR

Let's Build: Agenda

1. Introduction to HAPI FHIR MDM
2. Let's Build: EID Matching
3. Let's Build: Score-based Matching

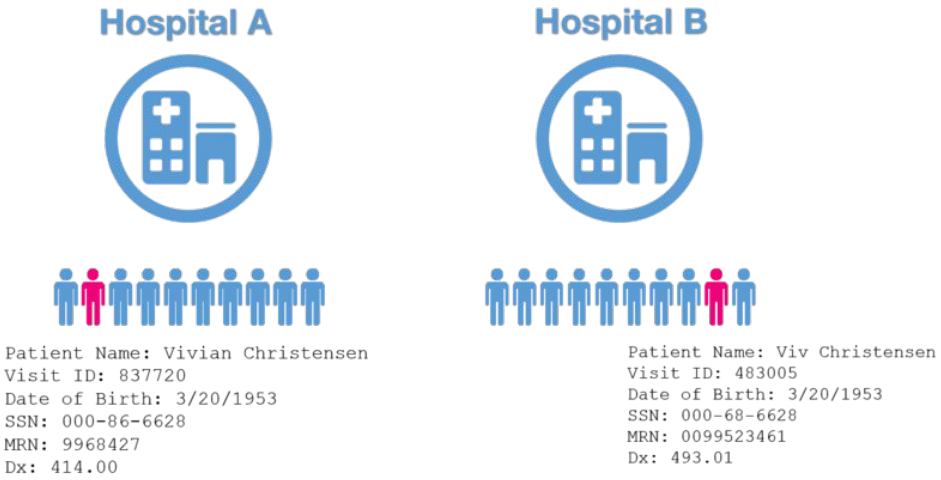
HAPI FHIR MDM

- Started as HAPI FHIR EMPI in 2019 as a module of HAPI FHIR
- Goal: Build an open-source EMPI solution as an extension of a FHIR repository

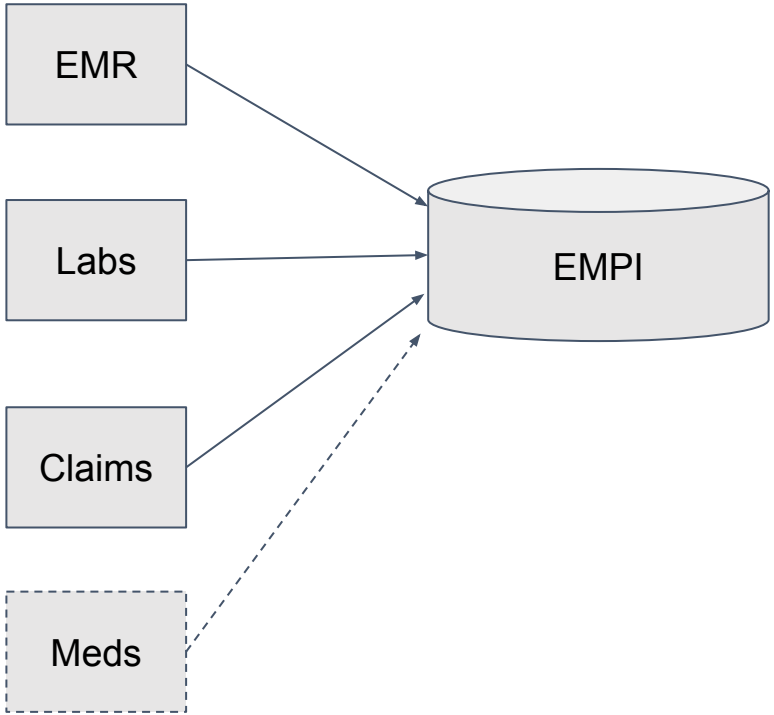
smile CDR

What is EMPI?

- Enterprise Master Patient Index
- Data in FHIR repositories often comes from multiple sources
- EMPI links the patient records from different source systems



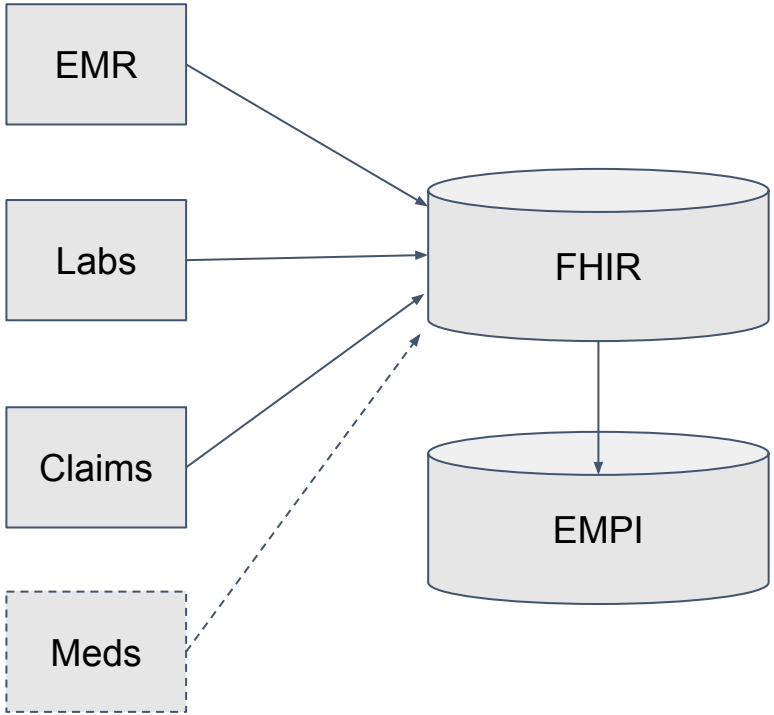
Advantages of EMPI on FHIR



Standalone EMPI

- It's a lot of work to configure EMPI to digest data from different data sources.
- You need to have EMPI configuration expertise on hand for every new data source you want to add in the future.

Advantages of EMPI ON FHIR: 1 Leverage Mappings

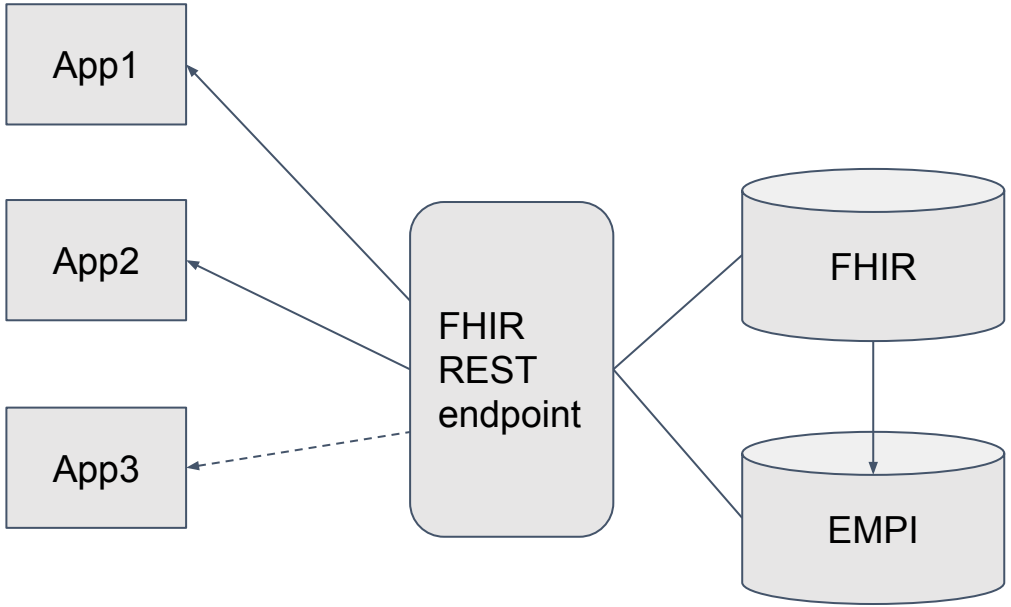


EMPI ON FHIR

- If you already have a FHIR repository, you can leverage existing system integrations.
- As new data sources are added to the FHIR repository, they are automatically available to EMPI at no incremental cost.
- Data is already mapped, normalized, and indexed for EMPI.

smile CDR

Advantages of EMPI on FHIR: 2 Easy Access

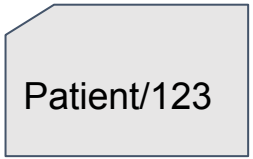


FHIR EMPI

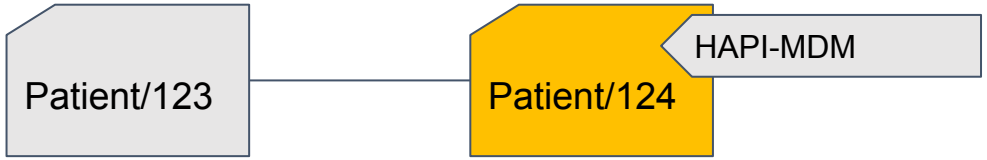
- Consumers of EMPI data can leverage the existing FHIR api to consume EMPI data, rather than having to implement a proprietary legacy EMPI api.

smile CDR

FHIR Model



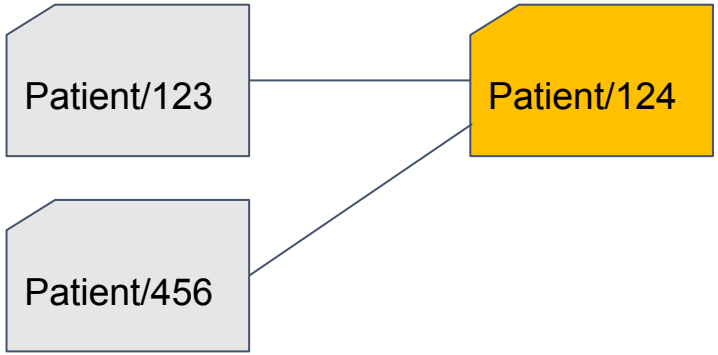
FHIR Model



MPI_LINK Table

Source Resource	Golden Resource
Patient/123	Patient/124

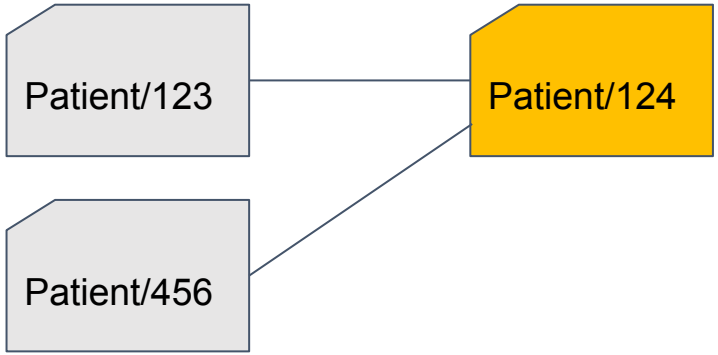
FHIR Model



MPI_LINK Table

Source Resource	Golden Resource
Patient/123	Patient/124
Patient/456	Patient/124

FHIR Model



MPI_LINK Table

Source Resource	Golden Resource
Patient/123	Patient/124
Patient/456	Patient/124

EMPI vs MDM

- Initial EMPI implementation was Patient & Practitioner only
- Since all the mechanisms were FHIRPath based, the mechanism can apply to all resource types
- Look forward to seeing what people might do with this
  - Best Possible Medication History
  - Allergy, Immunization reconciliation
  - …?

HAPI FHIR MDM

Two matching modes

Core MDM Question

Are two different people actually the same person?

MDM Matching Mode 1: EID

- Simplest and most common type of EMPI matching
- Configure an eid system in HAPI FHIR MDM. If two resources have the same identifier with that system, they are linked.
- A couple of config values affect the logic. Works best in “strict mode” with both of these enabled:
  - Prevent EID Updates
  - Prevent multiple EIDs

EID Matching Minimal Config

```
{
  "version": "1",
  "mdmTypes": ["Patient"],
  "candidateSearchParams": [{
      "resourceType": "Patient",
      "searchParams": ["identifier"]
    }],
  "eidSystem": "http://example.com/fhir/identifier/my-eid-system"
}
```

Let's Build!

git clone git@github.com:hapifhir/hapi-fhir-jpaserver-starter.git

git checkout 2022-devdays-mdm

>>> git compare with branch origin/master

- Edit MdmConfig.class:
  resourceLoader.getResource("mdm-rules-eid.json");
- rm -fr target
- Run Demo in IntelliJ

Let's Build! ctd…

In IntelliJ, run:

1. Patient A1, view Demo logs
2. MdmQueryLinks
3. Patient A2, view Demo logs
4. MdmQueryLinks
5. Patient B, view Demo logs
6. MdmQueryLinks

Read more

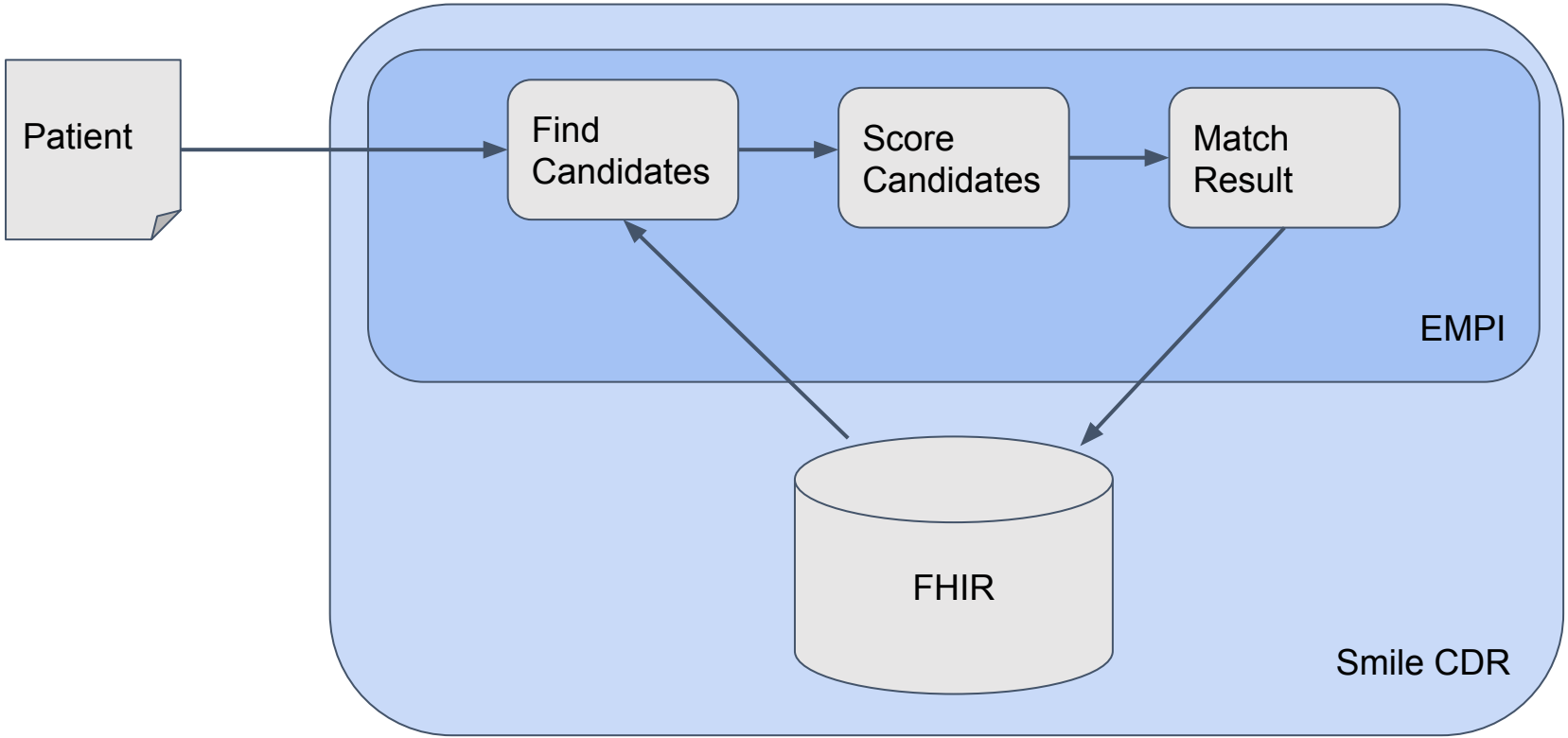
https://hapifhir.io/hapi-fhir/docs/server_jpa_mdm/mdm_eid.html

MDM Matching Mode 2: Score

Best match is determined in multiple steps:

1. Find candidates (“blocking” searches)
2. Score candidates in named matchers
3. Map matcher lists to POSSIBLE_MATCH or MATCH

EMPI Matching and Linking



Score Matching Simple Config

```
{
  "version": "1",
  "mdmTypes": ["Patient"],
  "candidateSearchParams": [
    {
      "resourceType": "Patient",
      "searchParams": ["birthdate",  "given:nickname",  "family"]
    },
    {
      "resourceType": "*",
      "searchParams": ["identifier"]
    }
  ]
```

Score Matching Simple Config ctd…

```
"matchFields": [
   {
     "name": "nickname",
     "resourceType": "*",
     "resourcePath": "name.given",
     "matcher": {
       "algorithm": "NICKNAME"
     }
   },
   {
     "name": "medicare-id",
     "resourceType": "Patient",
     "resourcePath": "identifier",
     "matcher": {
       "algorithm": "IDENTIFIER",
       "identifierSystem": "http://hl7.org/fhir/sid/us-medicare"
     }
   }
 ],
```

smile CDR

Score Matching Simple Config ctd…

```
"matchResultMap": {
  "nickname" : "POSSIBLE_MATCH",
  "nickname,medicare-id" : "MATCH"
}
```

Let's Build!

- Edit MdmConfig.class:
  resourceLoader.getResource("mdm-rules-score.json");
- rm -fr target
- Run Demo in IntelliJ

Let's Build! ctd…

In IntelliJ, run:

1. Patient C1
2. MdmQueryLinks
3. Patient C2
4. MdmQueryLinks
5. Patient C3
6. MdmQueryLinks
7. Patient D
8. MdmQueryLinks

MDM Link Expansion

```
http://example.com:8000/Observation?subject:mdm=Patient/123
```

```
http://example.com:8000/Observation?subject=Patient/123,Patient/456
```

Questions?