

example

Contents

1	Summary	1
2	Architecture	2
3	Patient Registration Logic	2
3.1	Handling the Source Identifiers and Aggregate ID	2
3.2	Custom Properties	2
4	Custom OID Resolution	2
5	Deploy Configuration	3
5.1	Empi Service Configuration File	3
5.2	ECR Information	3
6	Source Mapping	3
7	Delete Identity	5
7.1	Logic Flow Diagram	5
8	NativeID	5
8.1	Original Native ID Logic Text	5
8.2	NativeID Logic Flow Diagram	5
9	Terminology	6
9.1	nativeId	6
9.2	linkId	6

1 Summary

This deployment consists of the EMPI Identity Service and EMPI Consumer. The identity resolution uses an external provider Verato. The api documentation is provided separately. The customizations mentioned in the patient registration section are in this project. The core Verato implementation is a module within `empi-core`.

2 Architecture

TODO: need to add an overall description of services. Only mention deviations from norms and reference core documentation

3 Patient Registration Logic

3.1 Handling the Source Identifiers and Aggregate ID

Because Contexture deals with mrn re-use we are using Verato's overlay detection to alter the default `aggregate_id` behavior. Patient registration will use the fields from `nexus::PatientDemographics` along with `Contexture Master Feed List`. See the section on OID resolution.

1. register patient with Verato
 - lookup "Verato Source" from the "master feed list"
 - "Native Id" constructed as: "[Verato Source]::MRN"
2. if an overlay is detected append an increment
 - format: "[Verato Source]::MRN::[index]"
3. The response will include:
 - `aggregateId`: [Verato Source]::[MRN] or [Verato Source]::[MRN]::[index]
 - `empiId`: the value of "Verato's Link Id"
 - `patientIdRoot`: OID
 - `patientIdExt`: MRN

3.2 Custom Properties

This is done at patient upsert

4 Custom OID Resolution

TODO: Describe the OID resolution logic and data structures

5 Deploy Configuration

5.1 Empi Service Configuration File

```
apiUrl: https://cust0033-dev.verato-connect.com
apiUrlIp: 104.16.96.56:443
# credentials used for basic authentication on top of
# mutual tls
basicUsername: contexture.umpi.cust0033-dev.2023.04
basicPw: "...
# name of the filenames for mutual tls within the "config" folder
# on s3
veratoKeyLoc: verato.key
veratoPemLoc: verato.pem
clientConfig:
  maxOverlayRetries: 20000
  # this is where the verato responses are archived
  veratoArchivedResponses:
    locationType: s3
    bucket: "[env]-contexture-empisvc"
    subFolder: "archive-output"
  # the current state of polled identity updates
  notificationsState:
    location:
      type: s3
      location: "[env]-contexture-empisvc"
    path: "/config/notifications-search-state.json"
  # oid to verato source name resolution
  # provided mappings by Contexture
  veratoSourceMappings:
    location:
      type: s3
      location: "dev-contexture-empisvc"
    path: "/config/verato_source_mappings.csv"
```

5.2 ECR Information

Need to test this out TODO

Harsh Kharate

October 26, 2023 at 3:37 AM

Edited

161674638527.dkr.ecr.us-east-1.amazonaws.com/ninjacat/empi-test

161674638527.dkr.ecr.us-east-1.amazonaws.com/ninjacat/empi-prod

@Manideep Senagapalli As per our discussion over teams we have created 2 ECR repos in AWS ROOT Account. permissi

Ohip : 392240231723

sdhl : 870745327828

Contexture: 014355562867

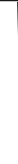
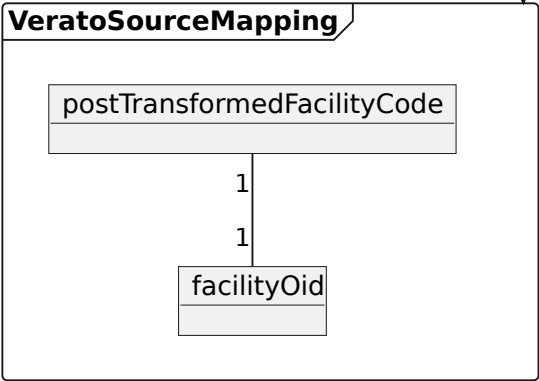
wyfi: 851153337270

CC: @Mohan Gadige

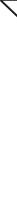
6 Source Mapping

High level view of matching Verato source name and facility codes.

ExtendedPatientMapping	
facilityOid	id.RootOid
mrn	id.extension



VeratoSourceMappingRow
postTransformedFacilityCode
veratoSourceName
facilityOid



VeratoRegisterPatient
veratoSourceName

veratoSourceName
extracted from
the matched facility
code and oid pair

ContextureNativeId
postTransformedFacilityCode
mrn
Option<Index>



facilityCode::mrn::index

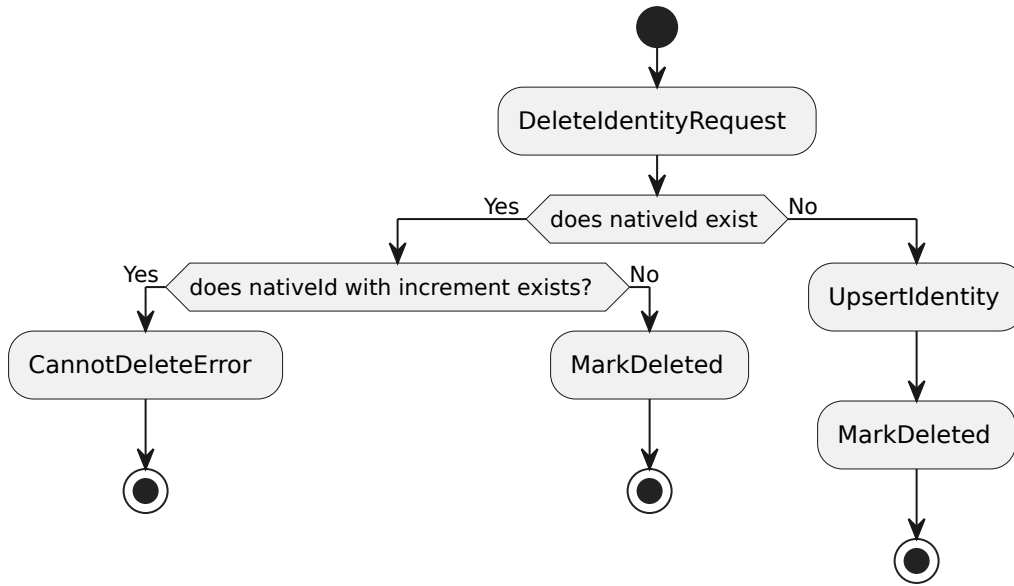
AggregateId



RegisteredPatient
empId
aggregateId
patientIdExt
patientIdRoot

7 Delete Identity

7.1 Logic Flow Diagram



8 NativeID

8.1 Original Native ID Logic Text

Original notes given before the project started:

Nexus/InterChange will ping PReg API with Demo + OID + MRN

EMPI Service (PtReg) - Demo + OID + MRN + VeratoSrcID::MRN

Invoke Verato API

200 - Success

Conflict

Demo + OID + MRN + VeratoSrcID::MRN::1

Conflict

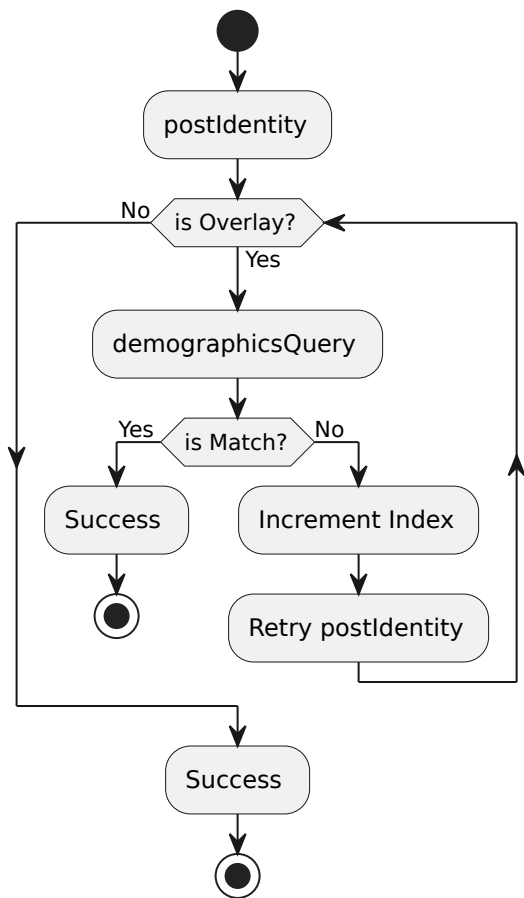
Demo + OID + MRN + VeratoSrcID::MRN::2

200 - Success

Response to Nexus/InterChange team - OID + MRN + VeratoSrcID::MRN::2 + EMPIID (LinkID)

8.2 NativeID Logic Flow Diagram

See <https://crashedmind.github.io/PlantUMLHitchhikersGuide/> on usage



9 Terminology

9.1 nativeId

Verato's source id. Otherwise known as the MRN. Sometimes used interchangeably in the documentation

9.2 linkId

Verato's empi id