PART Standard

for

CPR Broker

Workshop topics

Contents

1 Overview

2   Use cases

CPR Broker

Client application

Get person data

By unique key

By search

Subscribe to person events

Subscribe to birthdate

Subscribe to data updates

3   Request processing layers

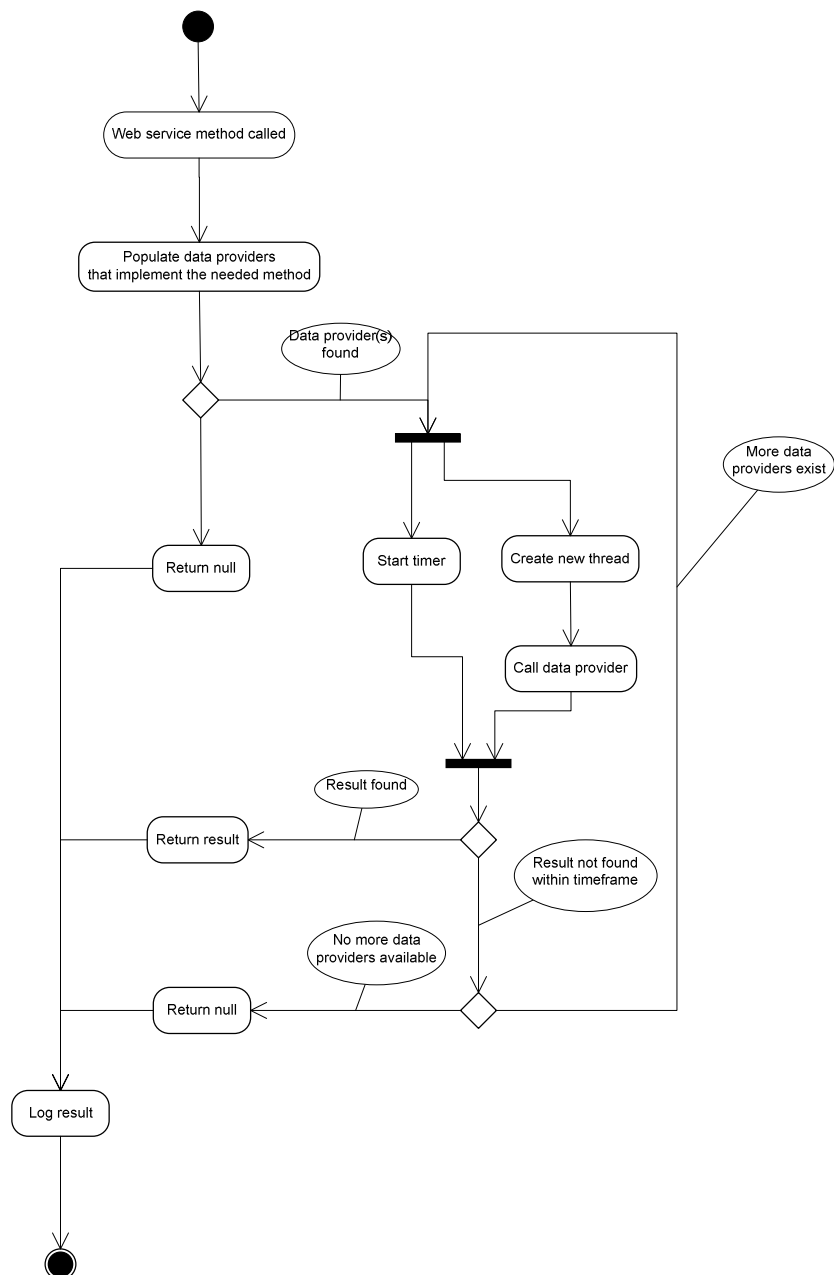| Client application	CPR Broker web service	Manager class	Data provider
●			
Initiate request	Extract SOAP header values	Define parameters and delegates	
		Initialize broker context	
		Load data providers	
			Execute request
		Update local cache	
		Log	
◉	Fill data quality SOAP header		

4 Data Providers

- Requests can only be handled with a data provider.

- A request maps to a method that is defined in a certain interface.

- A data provider implements one or more interface(s)

- An entry is added to the log in all cases

```
                            ●
                            │
                            ▼
              ┌─────────────────────────┐
              │ Web service method called│
              └─────────────────────────┘
                            │
                            ▼
           ┌──────────────────────────────────┐
           │   Populate data providers        │
           │ that implement the needed method │
           └──────────────────────────────────┘
                            │         Data provider(s)
                            ▼            found
                            ◇─────────────────────────────┐
                            │                              │
                            │              ████████        │
                            │              │      │        │
                            │              ▼      ▼        │  More data
                            ▼         Start timer  Create new thread   providers exist
                     ┌──────────┐                 │
                     │Return null│                ▼
                     └──────────┘          Call data provider
                            │                     │
                            │              ████████
                            │                     │
                     Result found ◇───────────────┘
            ┌──────────┐         │
            │Return result│◄─────┤
            └──────────┘          │ Result not found
                            │     within timeframe
           No more data     │
          providers available◇───────────────┘
            ┌──────────┐
            │Return null│◄─────
            └──────────┘
                 │
                 ▼
           ┌──────────┐
           │Log result│
           └──────────┘
                 │
                 ▼
                 ●
```

5 Actors

Actors will be supported to a limited extent. In the broker's context, an actor is a data provider used by the broker to get data.

When a data provider is added, there will be a need to supply an object ID that equals the UUID of the IT system represented by the data provider.
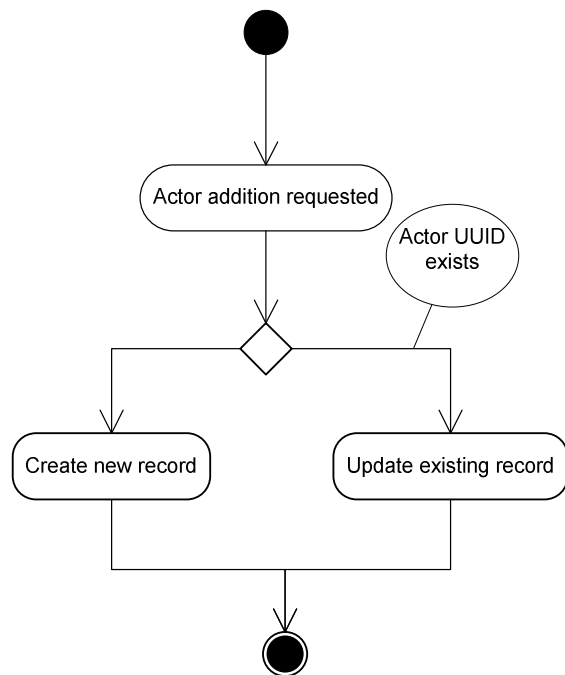
5.1 Questions

- Shall an actor be the actual authority that updated the person data? This can be the 'Authority code' in a person's data.

6 Registrations

There seems to be a need to have triple history rather than dual. The three time perspectives are

- Effect date: the date when the action took effect(getting married)

- Registration date: the date when the action was registered (registration of marriage in CPR)

- Broker update date: the date when CPR broker created the local registration.

The last perspective is not necessarily visible to the broker's clients.

6.1 Mapping

The concept of a UUID is not known to the system's data providers. To solve this, the system will have a mapping object (database table) that maps a person's physical identifier (UUID) to his logical identifier. The logical identifier can be the CPR number plus any other field. The candidate so far is the birth date.

6.1.1 Question

- Is the birth date a good identifier here? DPR can only update a person's data using the CPR number alone.

6.2 Registration operations

The following discussion of PART operations is only for logical understanding. They are not a part of the proposed solution.

6.2.1 Create

This happens when the person is first introduced to IT systems. This is also the case for persons that may already have a UUID assigned to them, but whose UUIDs are not known for the moment

6.2.2 Import

This happens when a person with a known UUID is introduced into the current IT system, with data obtained from another IT system.
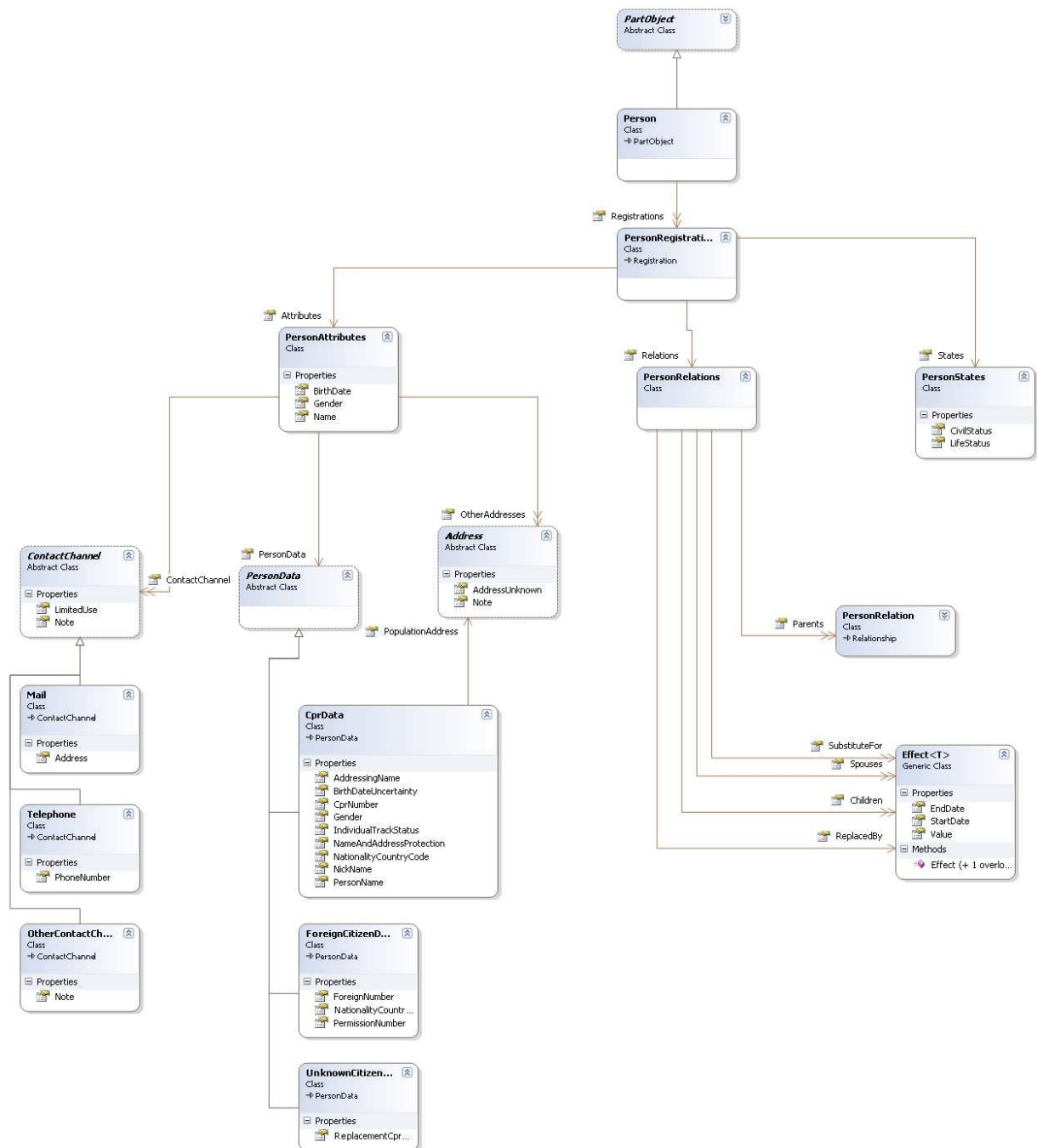
6.2.3 Edit

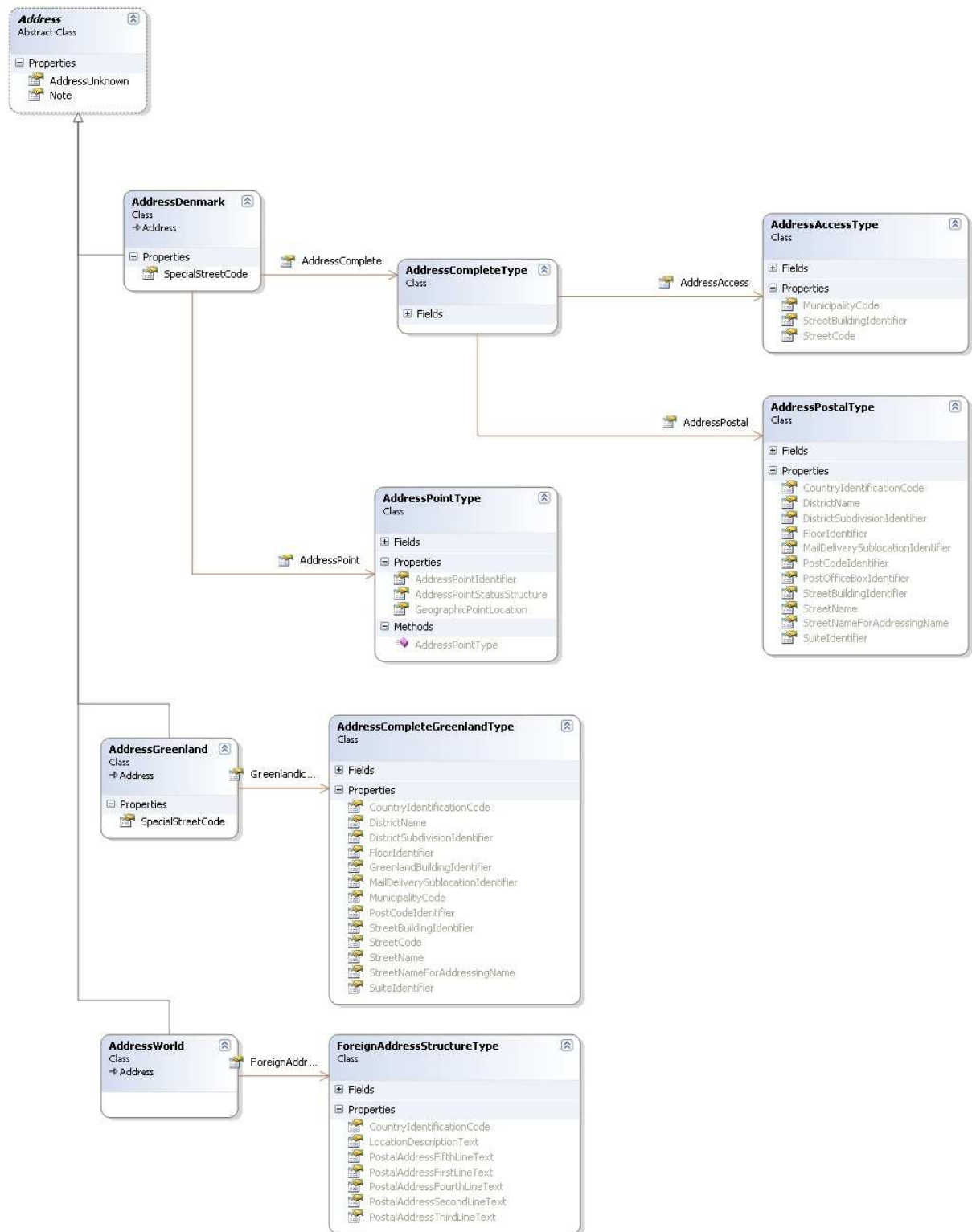This is where most of problems arise. See questions section.

6.2.3.1 Questions

- What real life actions are considered an 'Edit' operation?

- How can these actions be detected in DPR?

- How can these actions be detected in KMD?

Can an action be 'Apply for a new ID card' or 'apply for a passport'?

7 Data structures

Address
Abstract Class

☐ Properties
- 🔲 AddressUnknown
- 🔲 Note

AddressDenmark
Class
→ Address

☐ Properties
- 🔲 SpecialStreetCode

🔲 AddressComplete →

AddressCompleteType
Class

⊞ Fields

🔲 AddressAccess →

AddressAccessType
Class

⊞ Fields

☐ Properties
- 🔲 MunicipalityCode
- 🔲 StreetBuildingIdentifier
- 🔲 StreetCode

🔲 AddressPostal →

AddressPostalType
Class

⊞ Fields

☐ Properties
- 🔲 CountryIdentificationCode
- 🔲 DistrictName
- 🔲 DistrictSubdivisionIdentifier
- 🔲 FloorIdentifier
- 🔲 MailDeliverySublocationIdentifier
- 🔲 PostCodeIdentifier
- 🔲 PostOfficeBoxIdentifier
- 🔲 StreetBuildingIdentifier
- 🔲 StreetName
- 🔲 StreetNameForAddressingName
- 🔲 SuiteIdentifier

🔲 AddressPoint →

AddressPointType
Class

⊞ Fields

☐ Properties
- 🔲 AddressPointIdentifier
- 🔲 AddressPointStatusStructure
- 🔲 GeographicPointLocation

☐ Methods
- 🔍 AddressPointType

AddressGreenland
Class
→ Address

☐ Properties
- 🔲 SpecialStreetCode

🔲 Greenlandic ... →

AddressCompleteGreenlandType
Class

⊞ Fields

☐ Properties
- 🔲 CountryIdentificationCode
- 🔲 DistrictName
- 🔲 DistrictSubdivisionIdentifier
- 🔲 FloorIdentifier
- 🔲 GreenlandBuildingIdentifier
- 🔲 MailDeliverySublocationIdentifier
- 🔲 MunicipalityCode
- 🔲 PostCodeIdentifier
- 🔲 StreetBuildingIdentifier
- 🔲 StreetCode
- 🔲 StreetName
- 🔲 StreetNameForAddressingName
- 🔲 SuiteIdentifier

AddressWorld
Class
→ Address

🔲 ForeignAddr ... →

ForeignAddressStructureType
Class

⊞ Fields

☐ Properties
- 🔲 CountryIdentificationCode
- 🔲 LocationDescriptionText
- 🔲 PostalAddressFifthLineText
- 🔲 PostalAddressFirstLineText
- 🔲 PostalAddressFourthLineText
- 🔲 PostalAddressSecondLineText
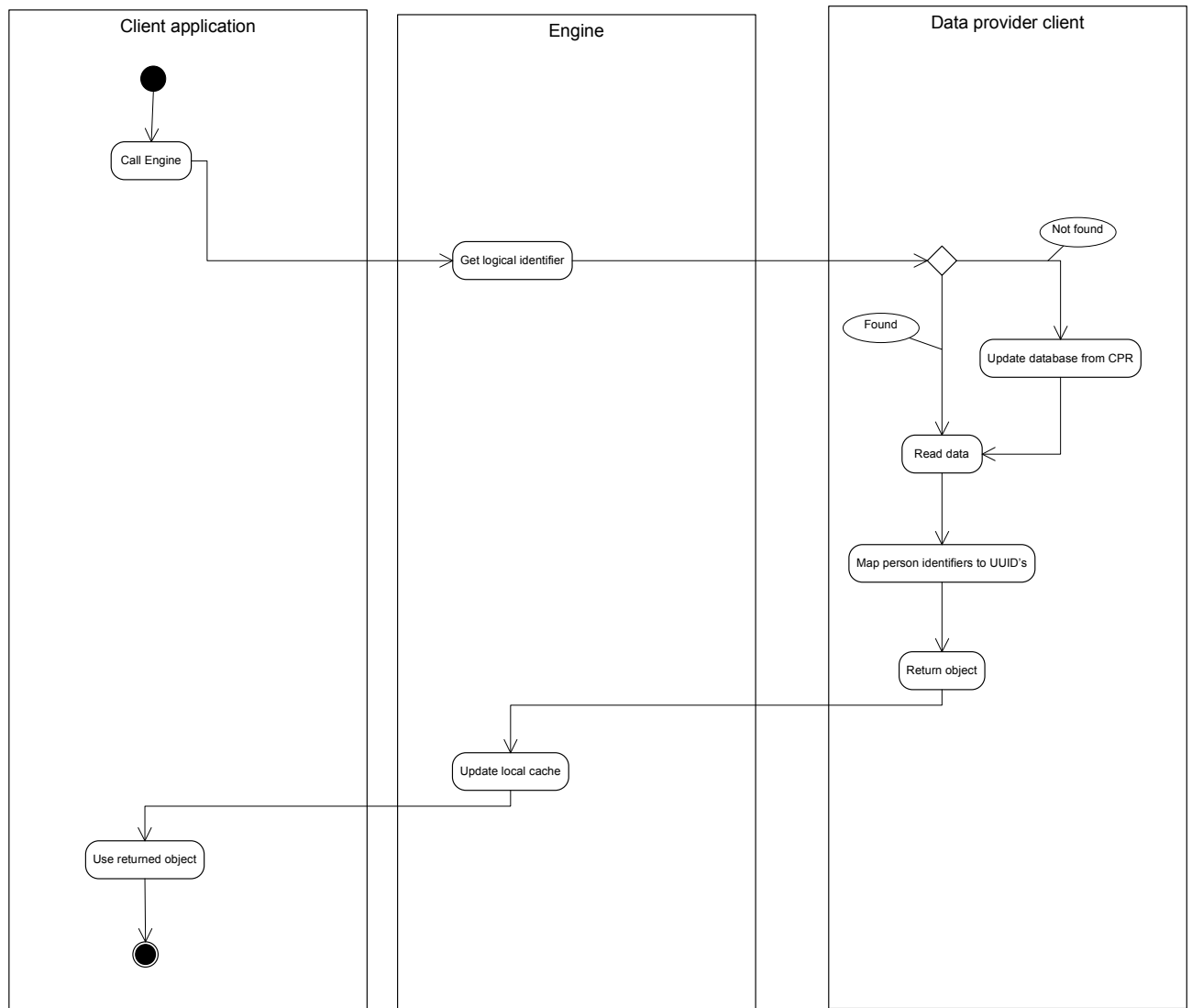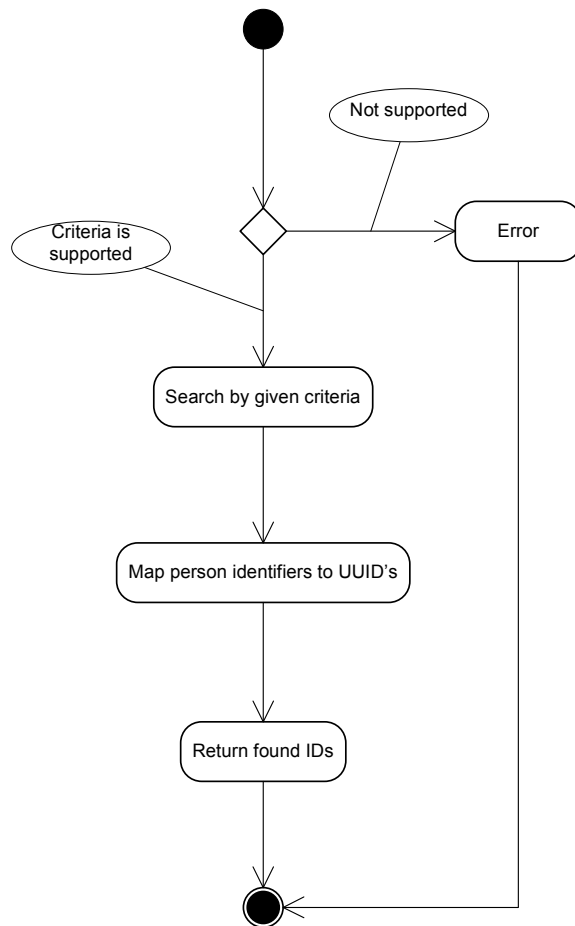- 🔲 PostalAddressThirdLineText

7.1 Questions

- Marital status 'deceased' is not in the PART standard. Is this a just mistake or made on purpose? Shall it be left the same as the status just before the person had died?

- I cannot see the difference between ContactAddress and PopulationAddress (PersonData)

8 Read & List operations

Client application	Engine	Data provider client

Client application
- Call Engine
- Use returned object

Engine
- Get logical identifier
- Update local cache

Data provider client
- Not found
- Found
- Update database from CPR
- Read data
- Map person identifiers to UUID's
- Return object

9   Search operation

9.1 Questions

1- When calling 'Search', what should be done if there is one or more criterion that cannot be searched by the data source? Possible options are:

- Raise an error

- Ignore that specific criterion

- Temporarily ignore the criterion and do an in memory filter

- Permanently ignore the criterion and show a warning message

2- On the other hand, what if the search criteria is empty or too wide for a reasonable (or possible) search

- Return error and ask the caller for a more strict criteria

- Return all matches (could be very large result)

- Return first N matches and a warning message (where N is a setting)
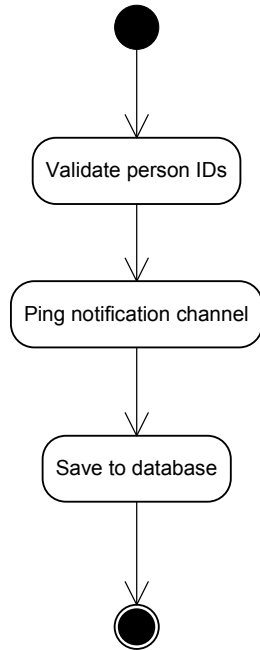
10 Local cache update
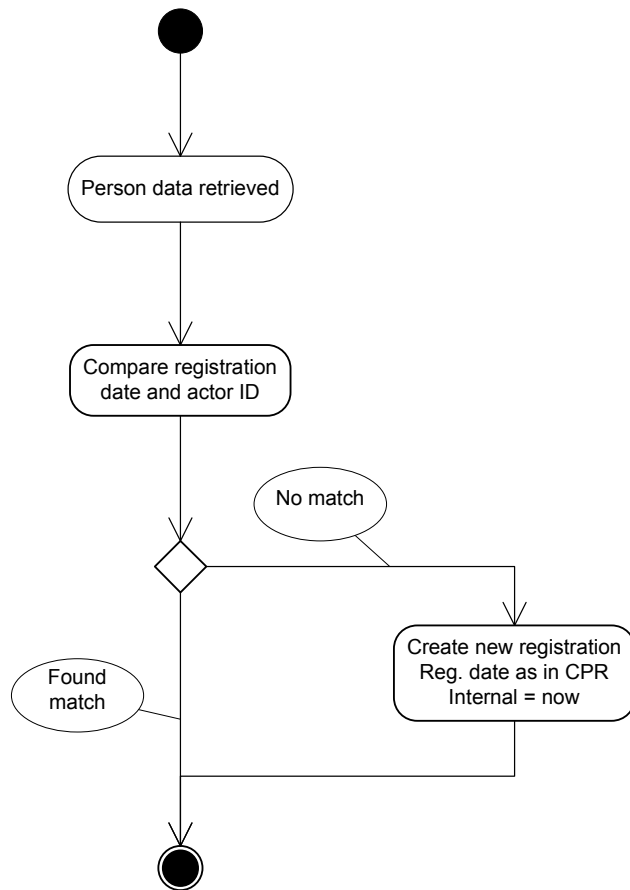
11 UUID Uniqueness & Import

Uniqueness can be achieved by a partial or full implementation of 'Import' operation. A draft can be like this
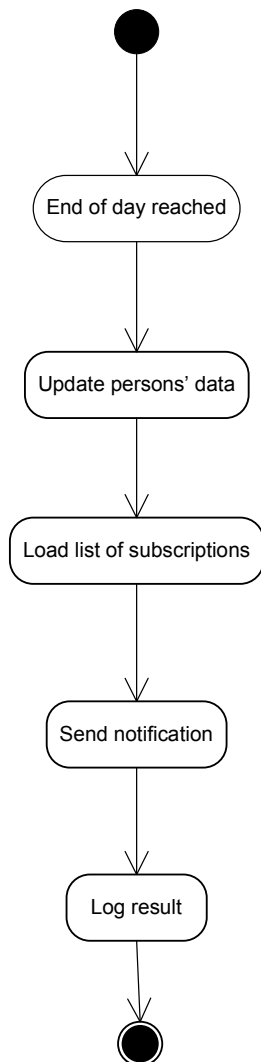
12 Subscription flow

Validate person IDs

Ping notification channel

Save to database
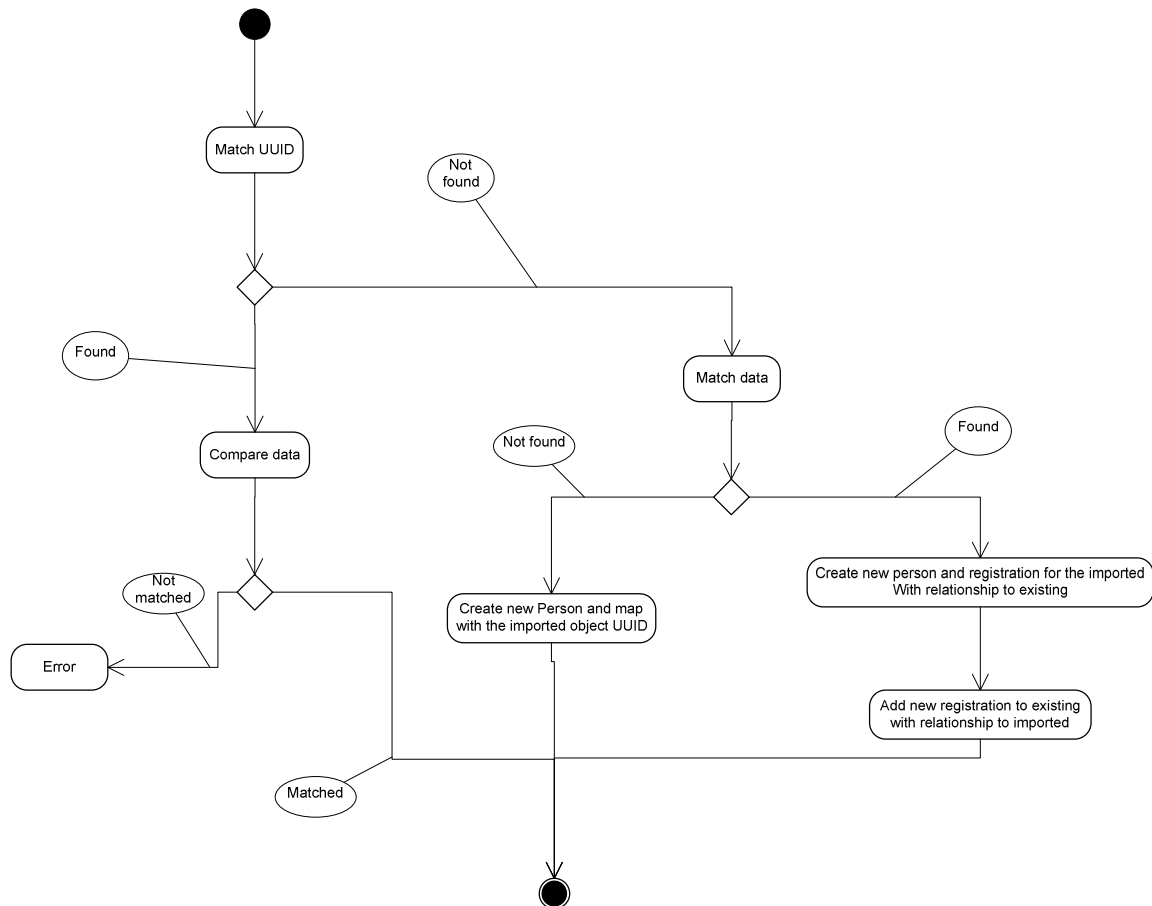
13 Events detection

- Events are detected daily

- A separate windows service handles the events

```
      ●
      │
      ▼
┌──────────────────┐
│ End of day reached │
└──────────────────┘
      │
      ▼
┌──────────────────┐
│ Update persons' data │
└──────────────────┘
      │
      ▼
┌──────────────────────┐
│ Load list of subscriptions │
└──────────────────────┘
      │
      ▼
┌──────────────────┐
│ Send notification │
└──────────────────┘
      │
      ▼
┌──────────────┐
│ Log result │
└──────────────┘
      │
      ▼
      ◉
```

14 Notification channels