PART Standard

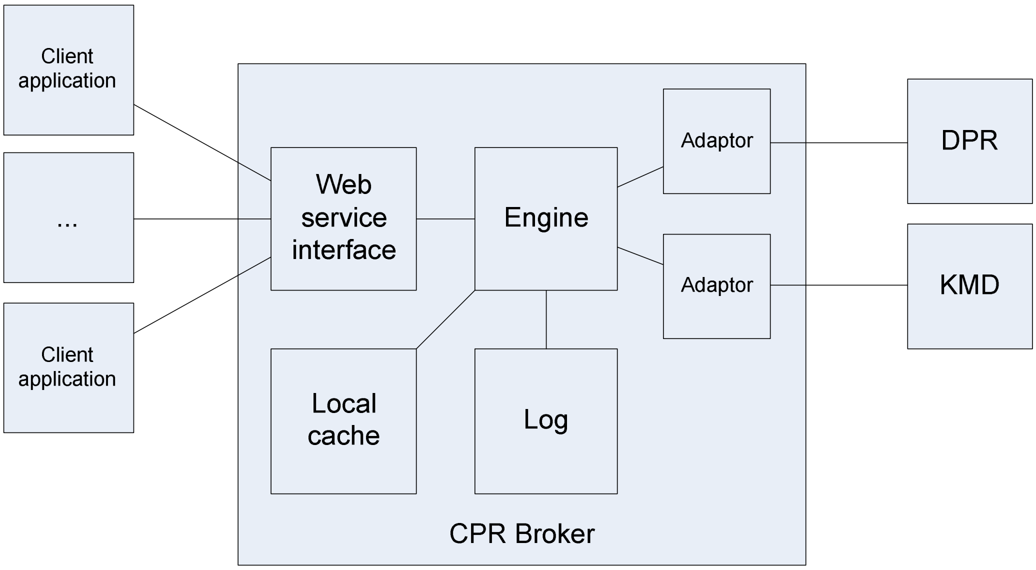
for

CPR Broker

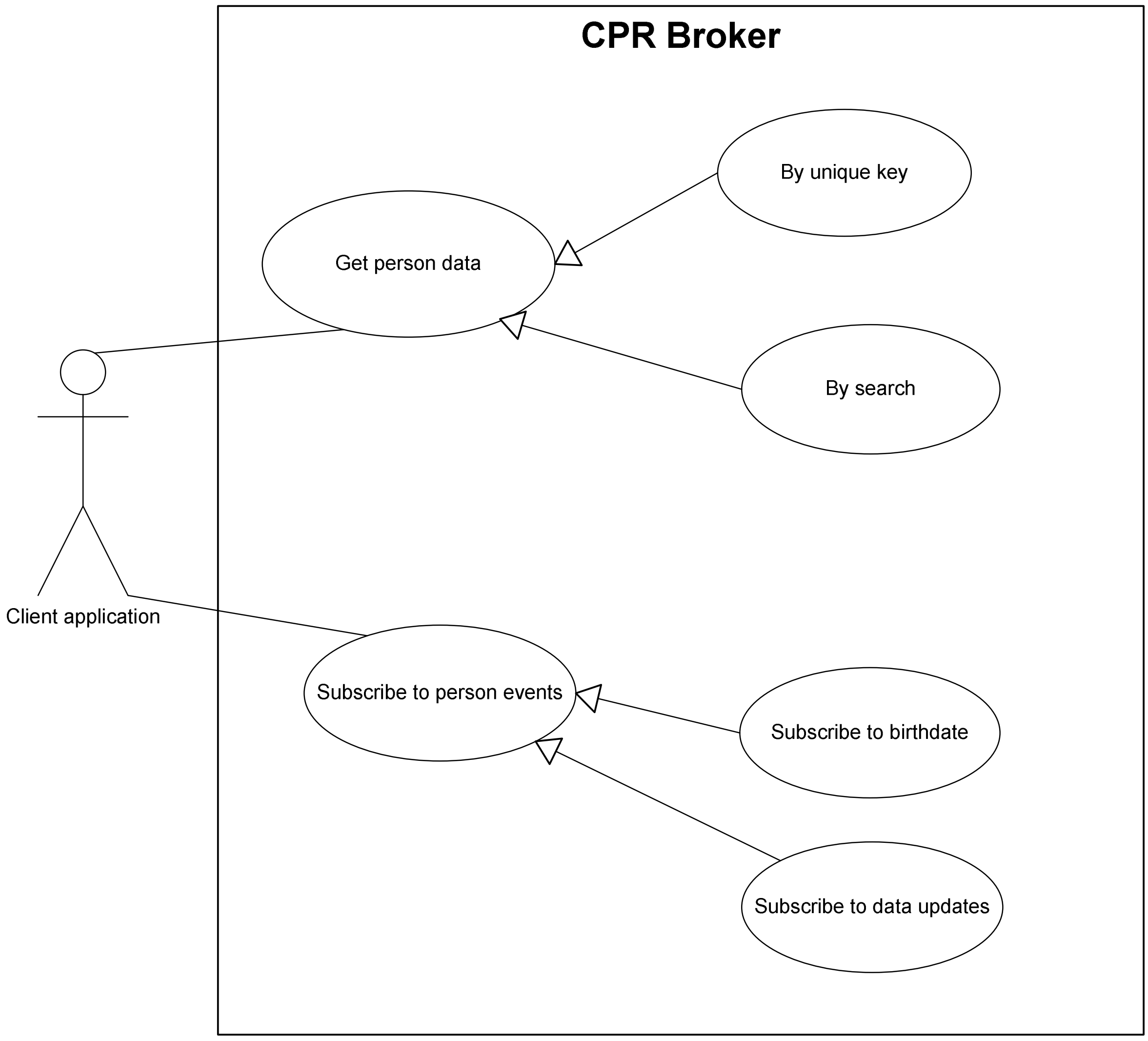
Workshop topics

Contents

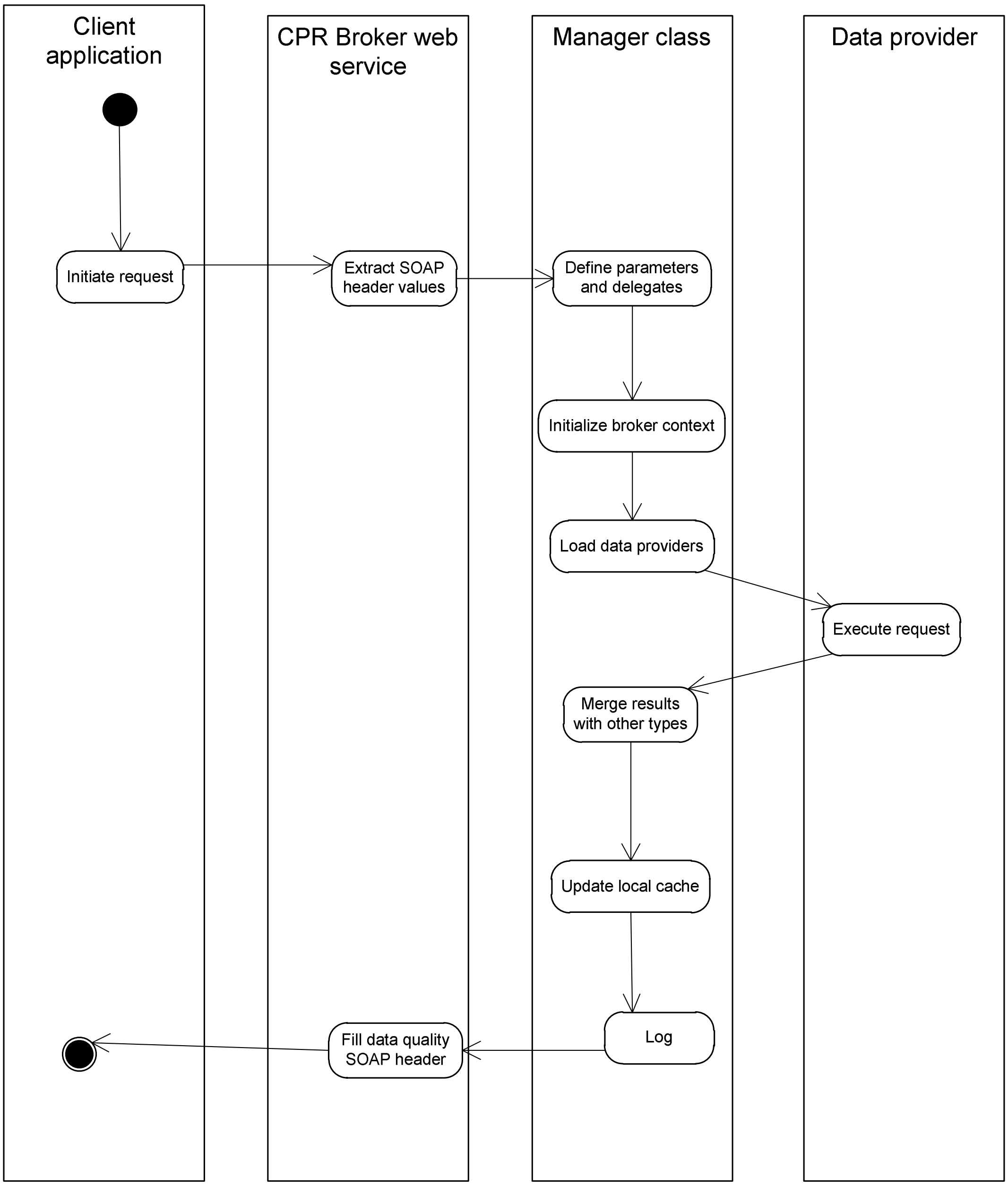
1 Overview


Client application
...
Client application
Web service interface
Engine
Adaptor
Adaptor
DPR
KMD
Local cache
Log
CPR Broker

2 Use cases


CPR Broker
By unique key
Get person data
By search
Client application
Subscribe to person events
Subscribe to birthdate
Subscribe to data updates

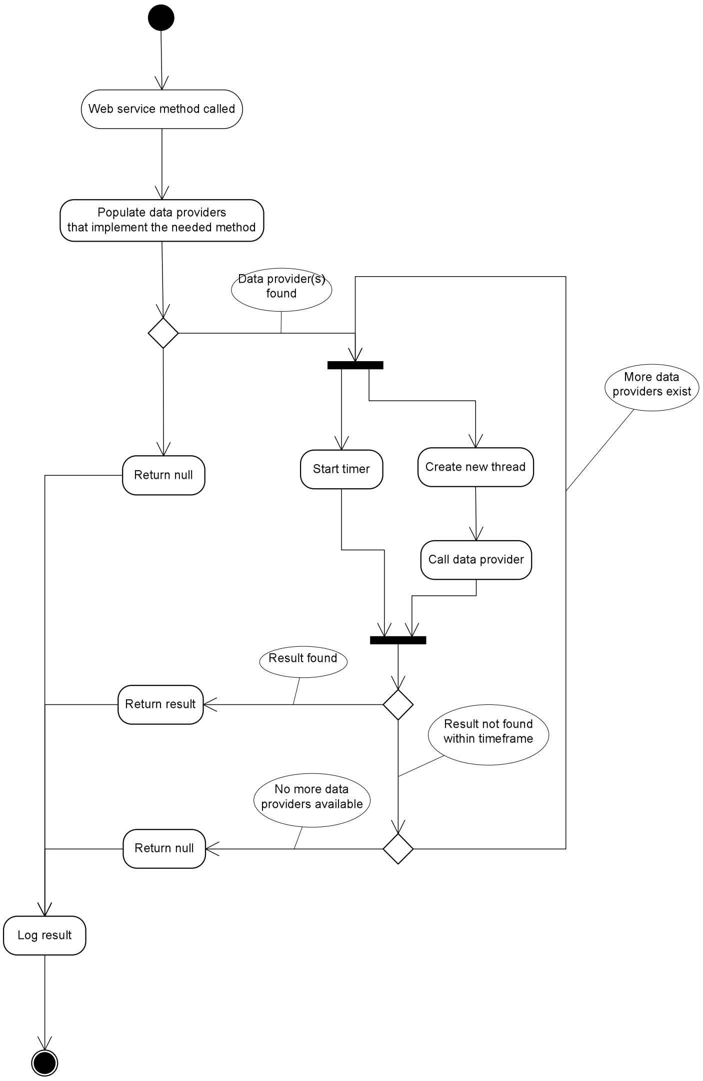
3 Request processing layers



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
Merge results with other types
Update local cache
Log
Fill data quality SOAP header

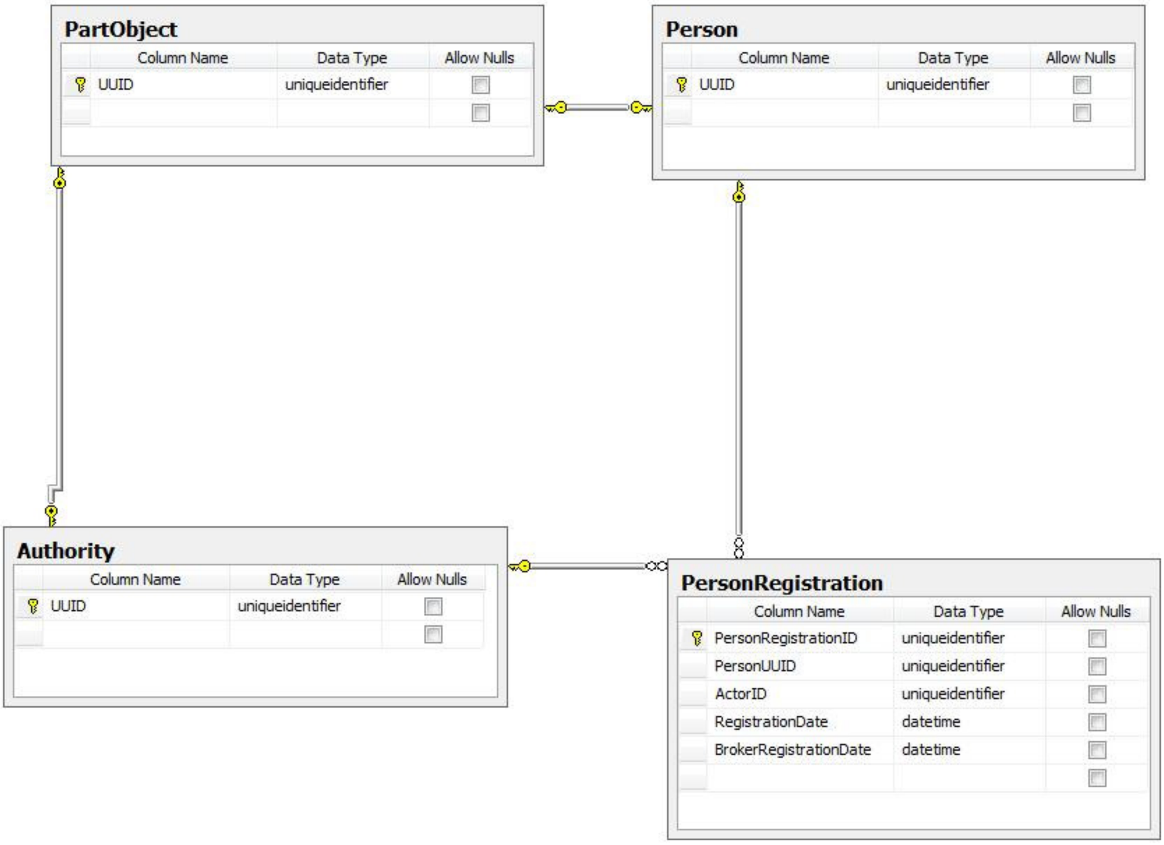
4 Data Providers

- 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
- There will be support for addition of data providers for contact info and medical info. However, no real data providers wll be implemented.


Web service method called
Populate data providers that implement the needed method
Data provider(s) found
More data providers exist
Return null
Start timer
Create new thread
Call data provider
Result found
Return result
Result not found within timeframe
No more data providers available
Return null
Log result

5 Registrations

5.1 Database structure



PartObject

Column Name	Data Type	Allow Nulls
UUID	uniqueidentifier	□
		□

Person

Column Name	Data Type	Allow Nulls
UUID	uniqueidentifier	□
		□

Authority

Column Name	Data Type	Allow Nulls
UUID	uniqueidentifier	□
		□

PersonRegistration

Column Name	Data Type	Allow Nulls
PersonRegistrationID	uniqueidentifier	□
PersonUUID	uniqueidentifier	□
ActorID	uniqueidentifier	□
RegistrationDate	datetime	□
BrokerRegistrationDate	datetime	□
		□

5.2 Time perspetives

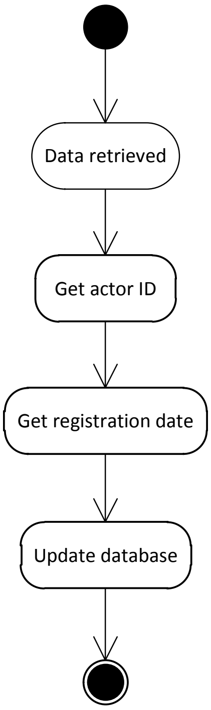
The most important time perspective is the registration time. Effect time will be specified whenever available. An unknown effect start and/or end date will be expressed with a special value (possibly null).

5.3 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 would be the CPR number.

The values for mapping will be obtained from a municipal API. If this is not possible, an error will be raised.

5.4 Registration object creation process

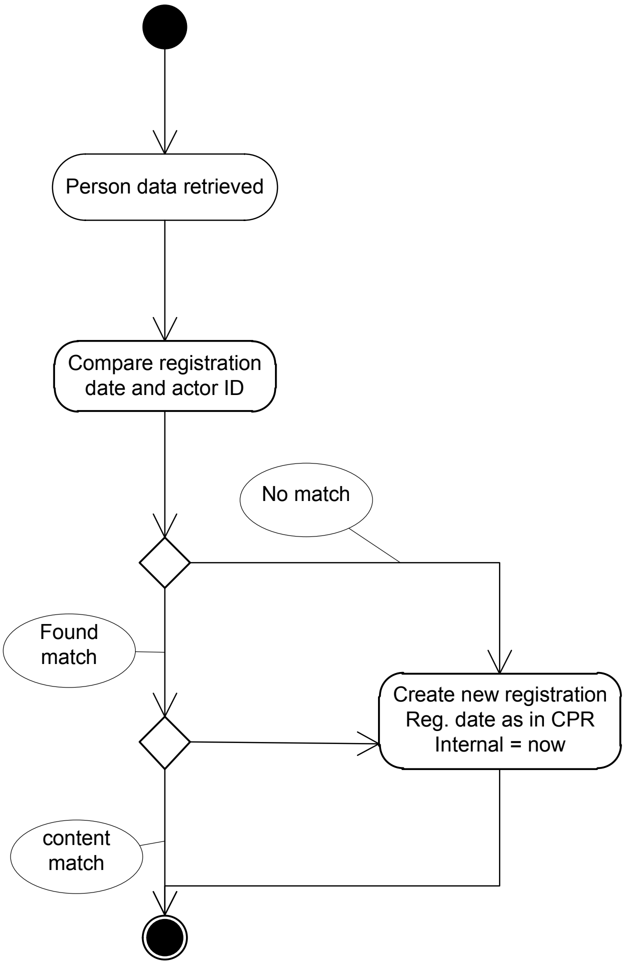


5.4.1 Values

The actor will be considered either DPR or KMD. Each will be assigned a fixed UUID. The registration date will be the latest date in the available data. If not possible, that would be the date when the broker got the data.

If everything goes OK, then a registration is uniquely identified by person's UUID, an actor ID and a registration date. In the very rare cases when this is not unique (such as in race conditions), there might be more than one registration with the same date. They will all be saved in the local database.

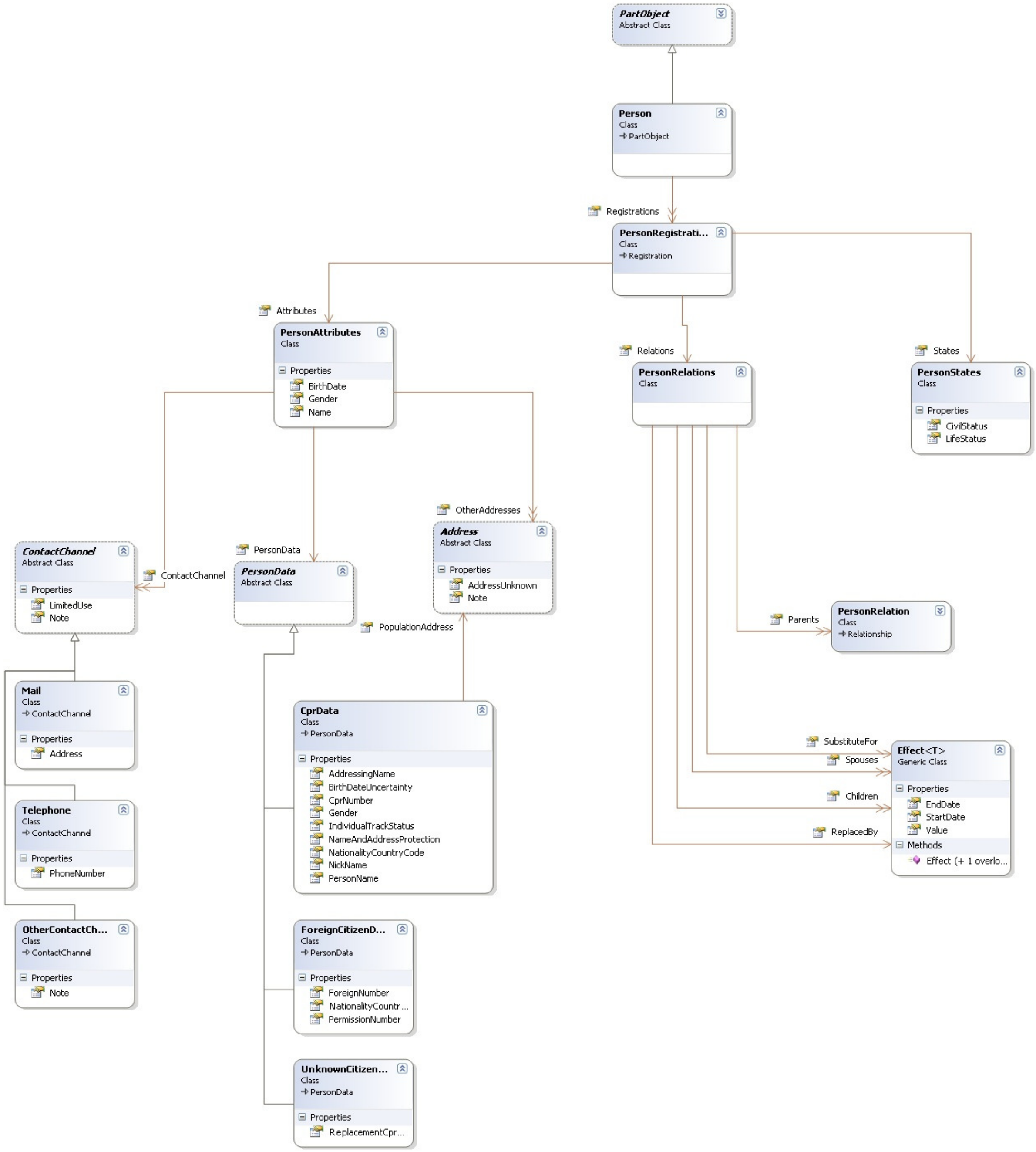
5.5 Local database update

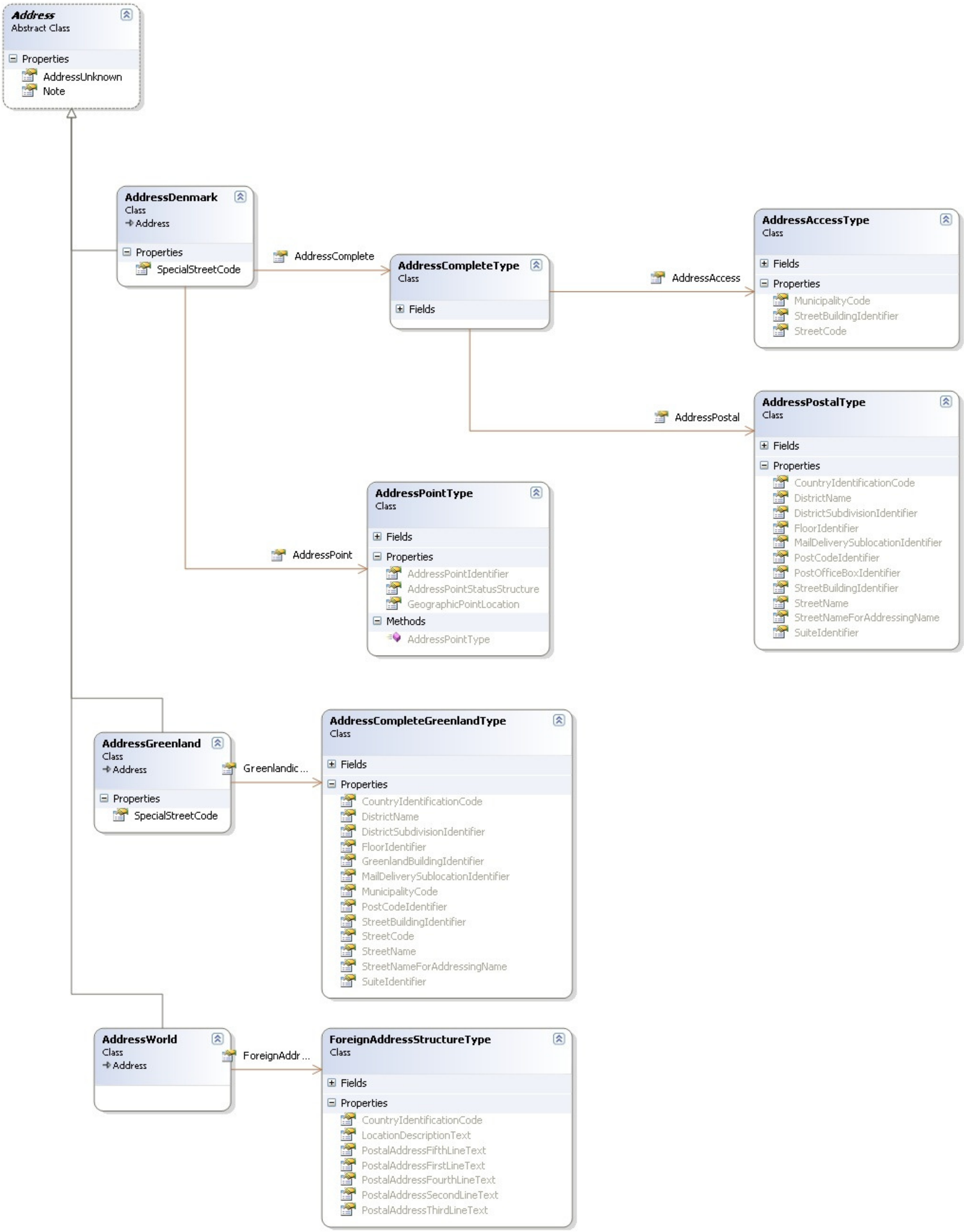


6 Data structures

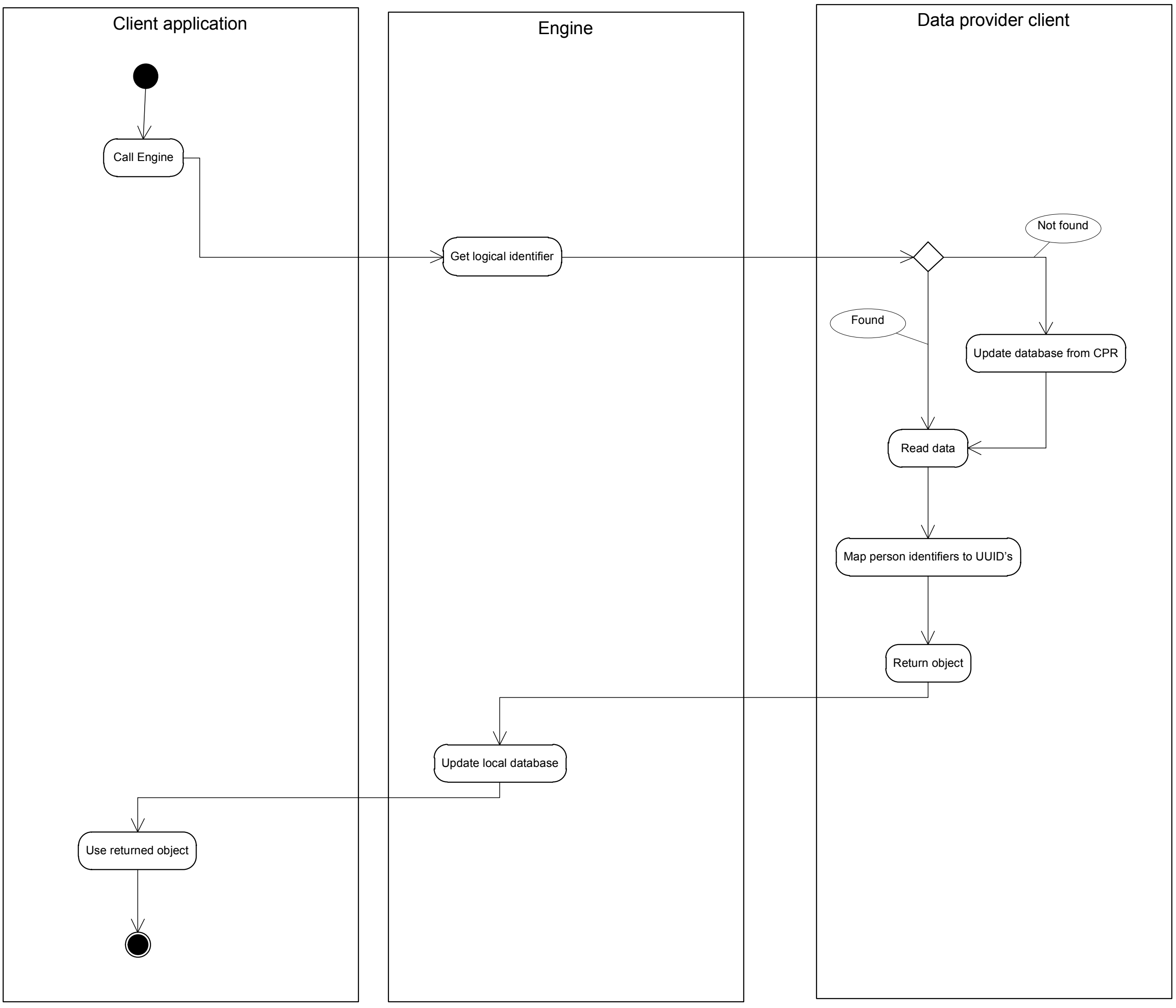
The folowing diagrams describe the details of the registration object, apart from its actor and date.

Magenta will work based on this structure until the finalized schema is received from ITST.




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

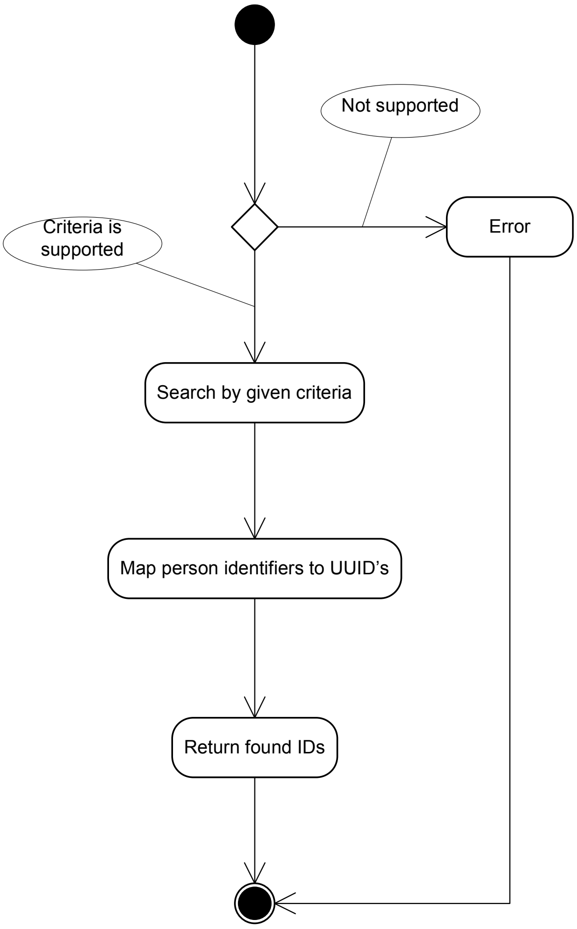
7 Read & List operations



There will be a refresh read operation which forces the broker to get new data rom data providers.

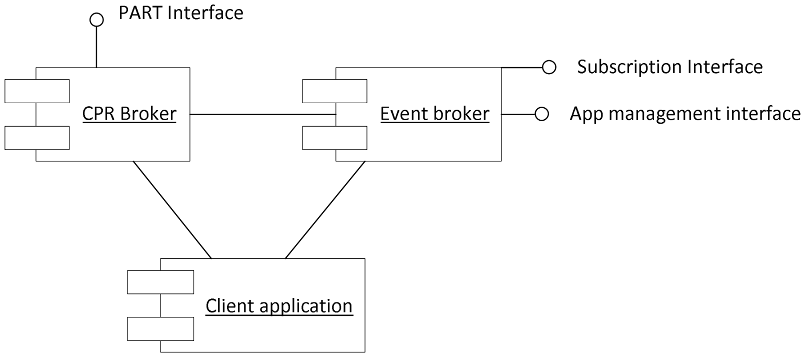
8 Search operation

Search will be implemented for th local database only. It will contain the posibility to search for people by their CPR number, Age or Name.

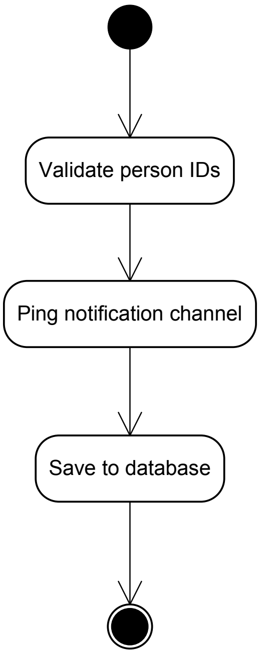


9 Events

9.1 Overall architecture



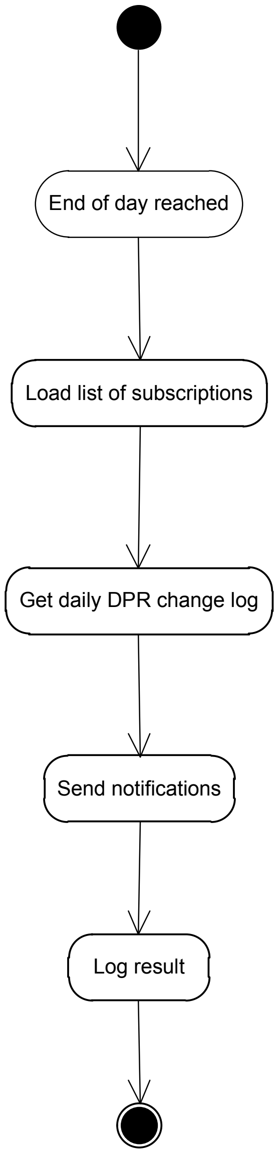
9.2 Subscription flow



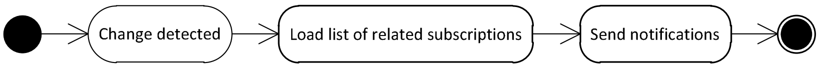
9.3 Notifications

- Data change events are detected as soon as they occur.
- An extract from DPR might be parsed daily to find which people have changed.
- Birthdate events are detected daily.
- A separate windows service handles the notifications.

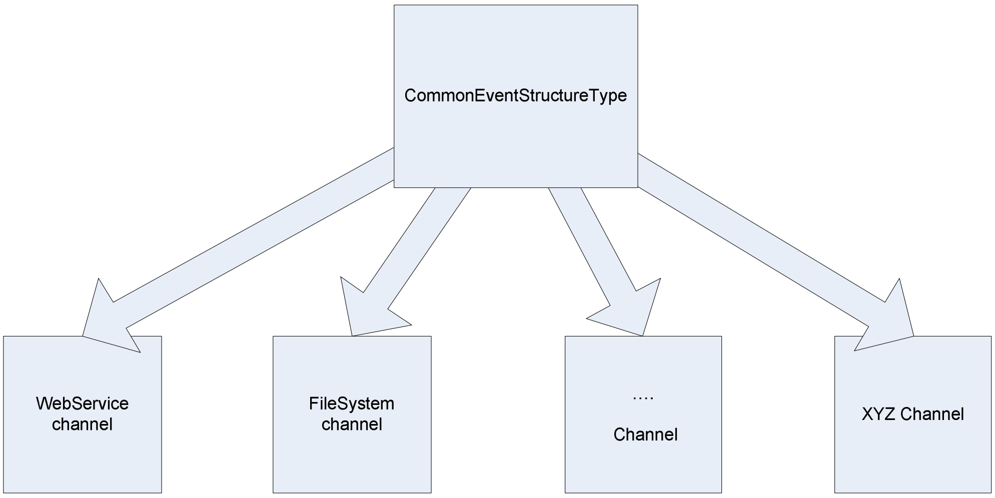
9.3.1 Daily notifications



9.3.2 Instantaneous data changed (caused by Read operations)



9.4 Notification channels


CommonEventStructureType
WebService channel
FileSystem channel
….
Channel
XYZ Channel