

# IMPLEMENTING PART IN CPR BROKER

## Designspecification

**MAGENTA<sup>aps</sup>**

© Copyright 2011

# TABLE OF CONTENTS

<b>1 Introduction</b>	<b>1</b>		
1.1 PART and Event Interface	1		
1.2 CPR Data Provider Interfaces	1		
1.2.1 Using PersonMaster to create UUID's for persons	2		
1.3 Administration Interface	2		
<b>2 Overview</b>	<b>3</b>		
<b>3 Use cases</b>	<b>4</b>		
<b>4 Request processing layers</b>	<b>5</b>		
<b>5 Data Providers</b>	<b>8</b>		
<b>6 Registrations</b>	<b>10</b>		
6.1 Database structure	10		
6.2 Time perspectives	10		
6.3 Mapping	10		
6.4 Registration object creation process			
			11
		6.4.1 Values	11
		6.5 Local database update	12
<b>7 Data structures</b>	<b>13</b>		
<b>8 Read &amp; List operations</b>	<b>15</b>		
<b>9 Search operation</b>	<b>16</b>		
			16
<b>10 Events</b>	<b>17</b>		
10.1 Client interface	17		
10.2 Overall Event Architecture	17		
10.3 Subscription flow	17		
10.4 Notifications	18		
10.4.1 Birthday Notifications	18		
10.4.2 Data Changed Notifications	19		
10.5 Notification channels	20		

# 1 INTRODUCTION

The CPR Broker is an implementation of the PART Standard.

The CPR Broker can be employed in organizations that make use of CPR.

The interface defined in the PART standard enable the organizations to use the same interface to access CPR from several sources. I allow suppliers of solutions to base their products on a single interface that can be used by many organizations.

The CPR Broker holds CPR data. This ensures better performance, less latency, and more reliability. I many cases It will also results in cost-savings for the organizations, depending of the contracts they have with providers of CPR data.

The CPR broker is also able to provide more information than is available from existing providers because it keeps more historical data, such as bi-temporal registrations and because organizations will be able to provide CPR data directly to the broker without being restricted to policies and technical limitations of external providers.

There are three kinds of interfaces

## 1.1 PART and Event Interface

This interface is specified by the PART standard.

It is used by applications inside organizations that need CPR data.

It also covers subscriptions to events and notification of events. I.e BirthdayRegistration allows a service to register itself to receive notifications when all or a specified subset of person reach a given age.

This interface is read-only for CPR data, but it does allow applications to change the state of the broker to e.g. maintain the list of notification registrations.

## 1.2 CPR Data Provider Interfaces

The interfaces towards the existing external providers of CPR data are not standardized yet. That is one of the motivations to use the CPR Broker in an organization. Therefore the CPR Broker have implemented adapters for some existing providers. In this version DPR and KMD is supported. Internally in the CPR Broker, Data Providers are represented by adapter objects, providing a common interface, to simplify addition of more data providers.

In addition to the interfaces implemented by existing data providers, the CPR Brokers also have a ReadRefresh service, that allow data providers to signal to the CPR Broker that data for specified persons have been updated. Currently no data providers have implemented such event signaling and because the events originate outside the organizations using the CPR

Broker, the broker itself can not do it.

However Gentofte Municipality have developed a service that simulated propagation of events by polling DPR and on update invoking ReadRefresh on the CPR Broker.

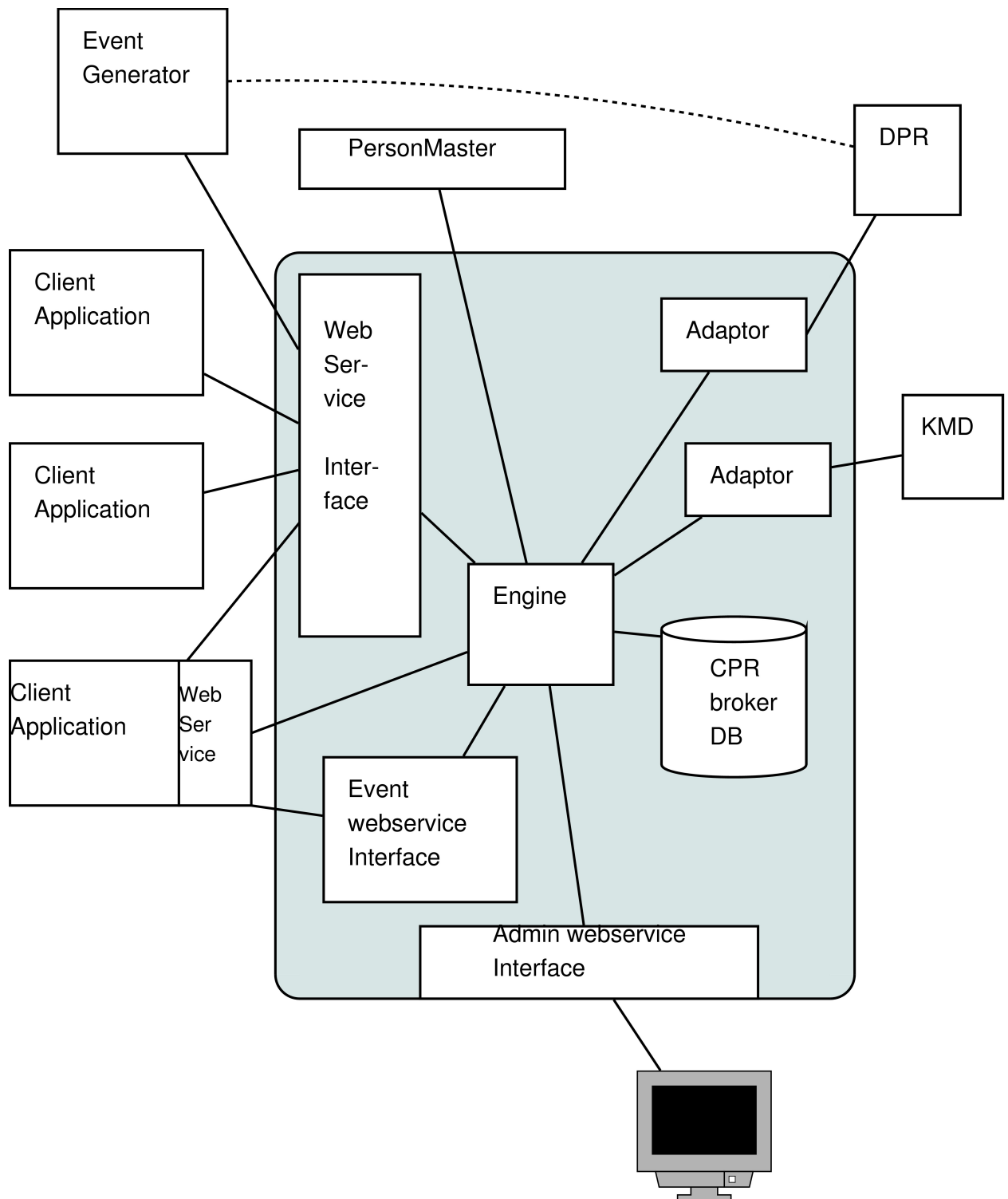
### **1.2.1 Using PersonMaster to create UUID's for persons**

PART make use of UUID's to identify persons. The CPR Broker as well as other applications based on PART will be deployed in many organizations and must be able to exchange information about persons. Therefore a CPR broker installation cannot assign its own UUID's to persons. Gentofte Municipality have developed a service, PersonMaster that can assign UUIDs to persons and provide UUIDs of persons based on current CPR numbers.

## **1.3 Administration Interface**

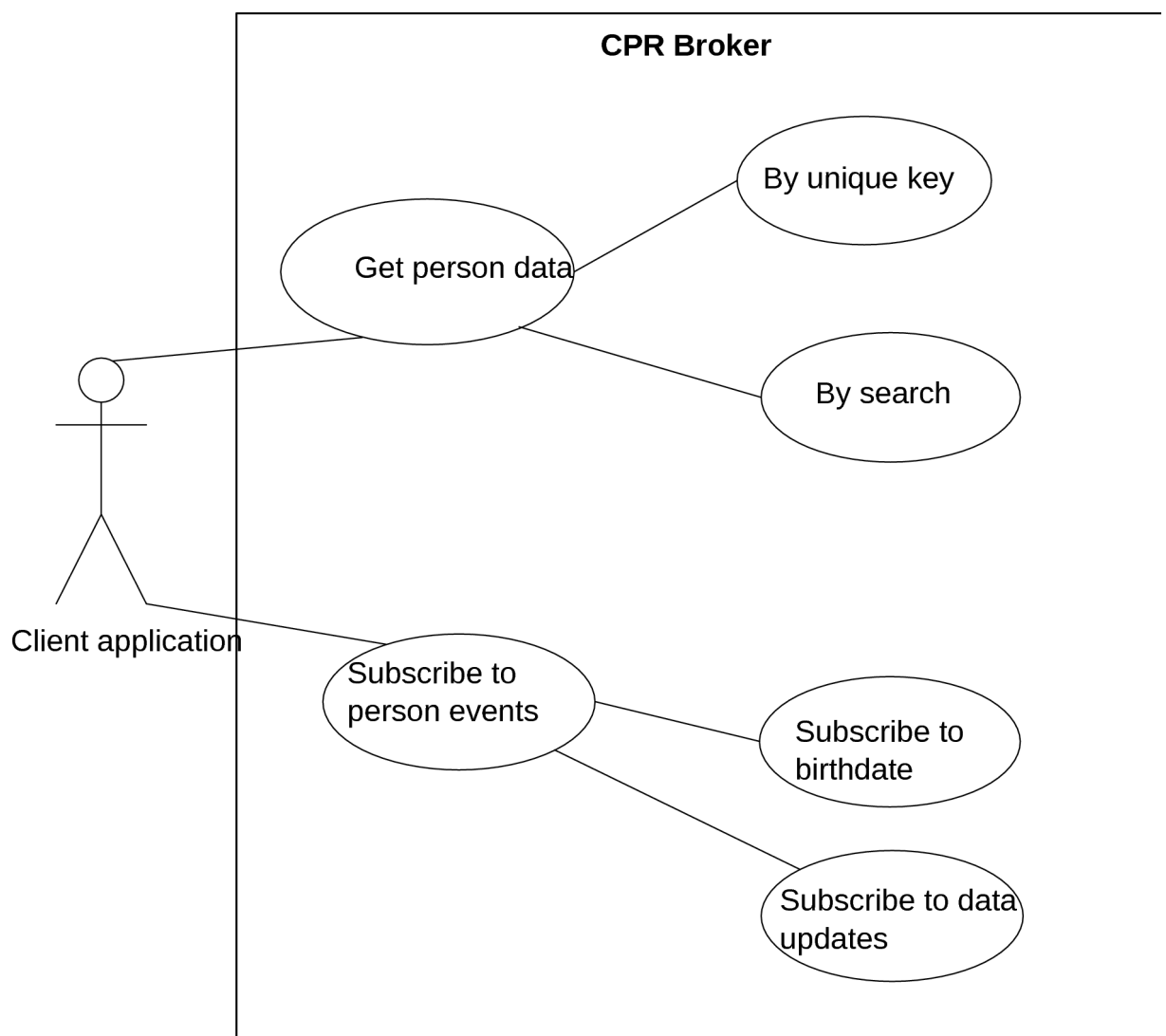
This interface is for administering the CPR Broker. i.e., Configuring, authorizing applications, logging etc.

## 2 OVERVIEW



## 3 USE CASES

### Use Case



The use case for a client application is to look up person data and to subscribe to person events.

## 4 REQUEST PROCESSING LAYERS

Clients make request through the web service implementing the PART standard. The CPR Broker extracts values from the SOAP headers, and invokes the broker engine.

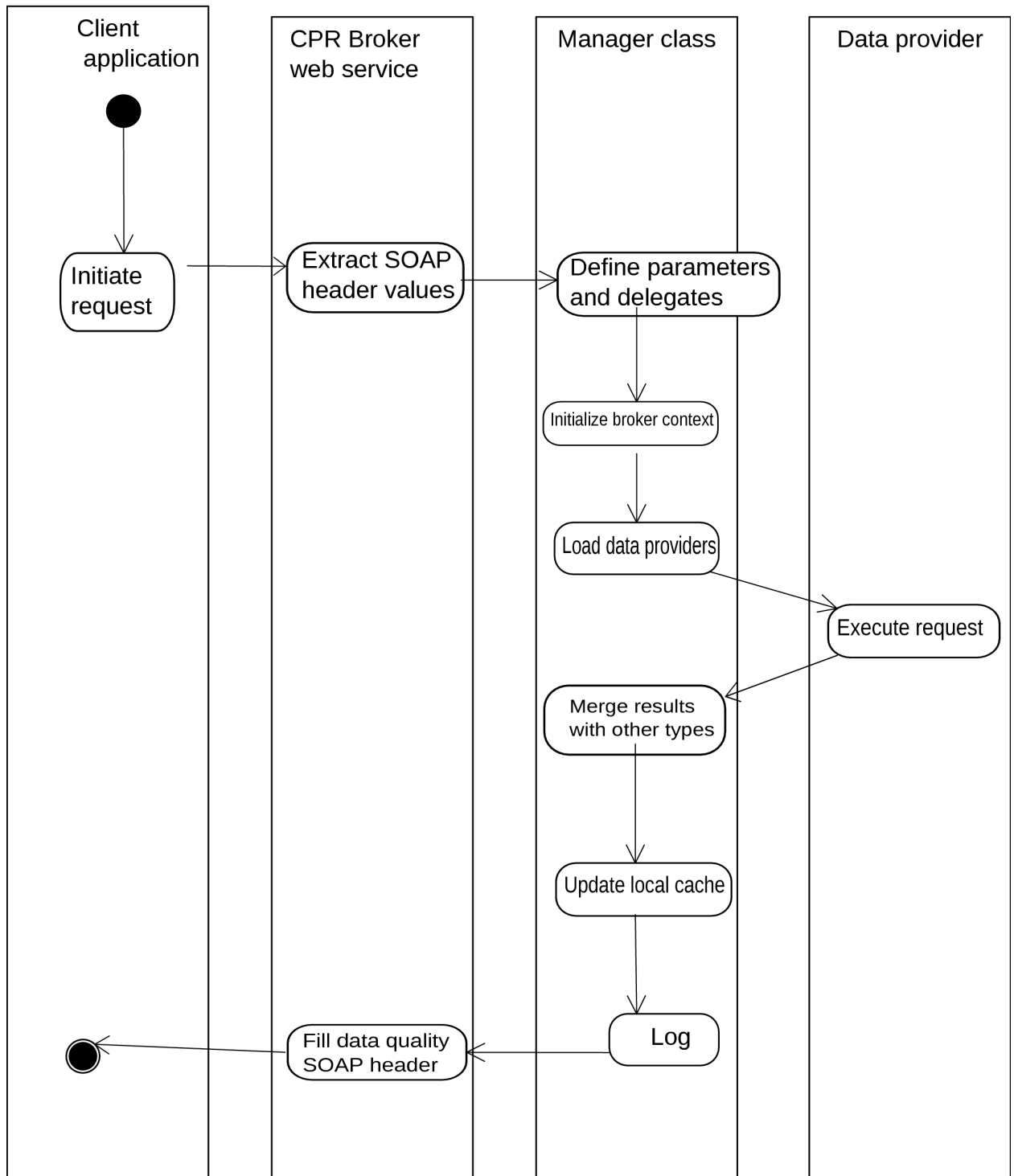
The engine contacts external data providers to ensure that the client will get fresh data. If new data is obtained the local broker database is updated with the new data as well as registration of the source of the data.

If the client request contains CPR numbers, not already registered in the broker, the broker will request the mapping from CPR numbers to UUID from the PersonMaster service. The broker registers the client request in its own log, generates a response according to the PART standard and delivers it to the client as a SOAP response.





## Activity



## 5 DATA PROVIDERS

Client requests require access to at least one data provider to ensure that the answer is up-to-date. If no data providers are accessible an error is returned to the client.

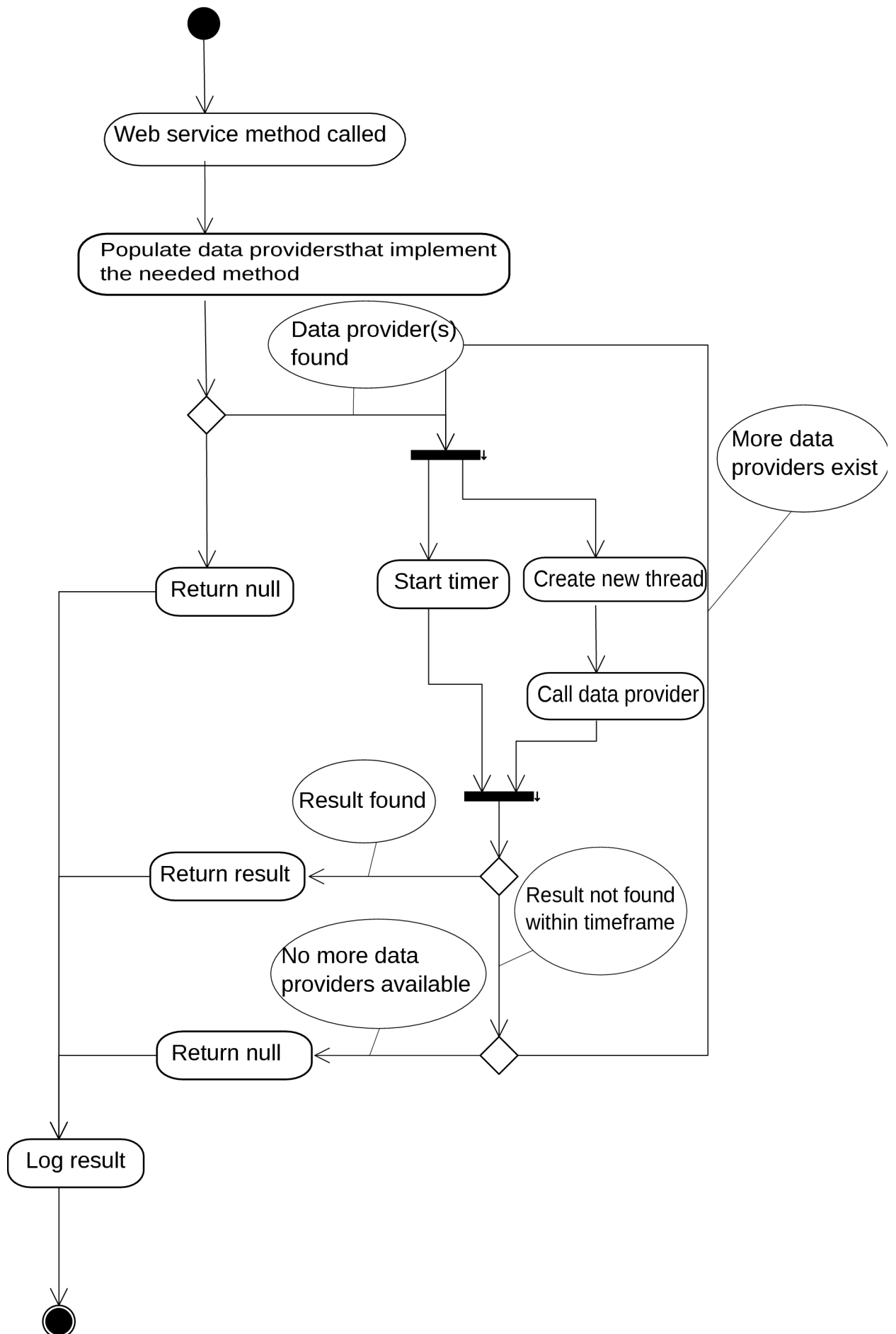
A request is handled by a method implemented in the corresponding web interface.

A data provider implements one or more interfaces.

An entry is added to the log in all cases.

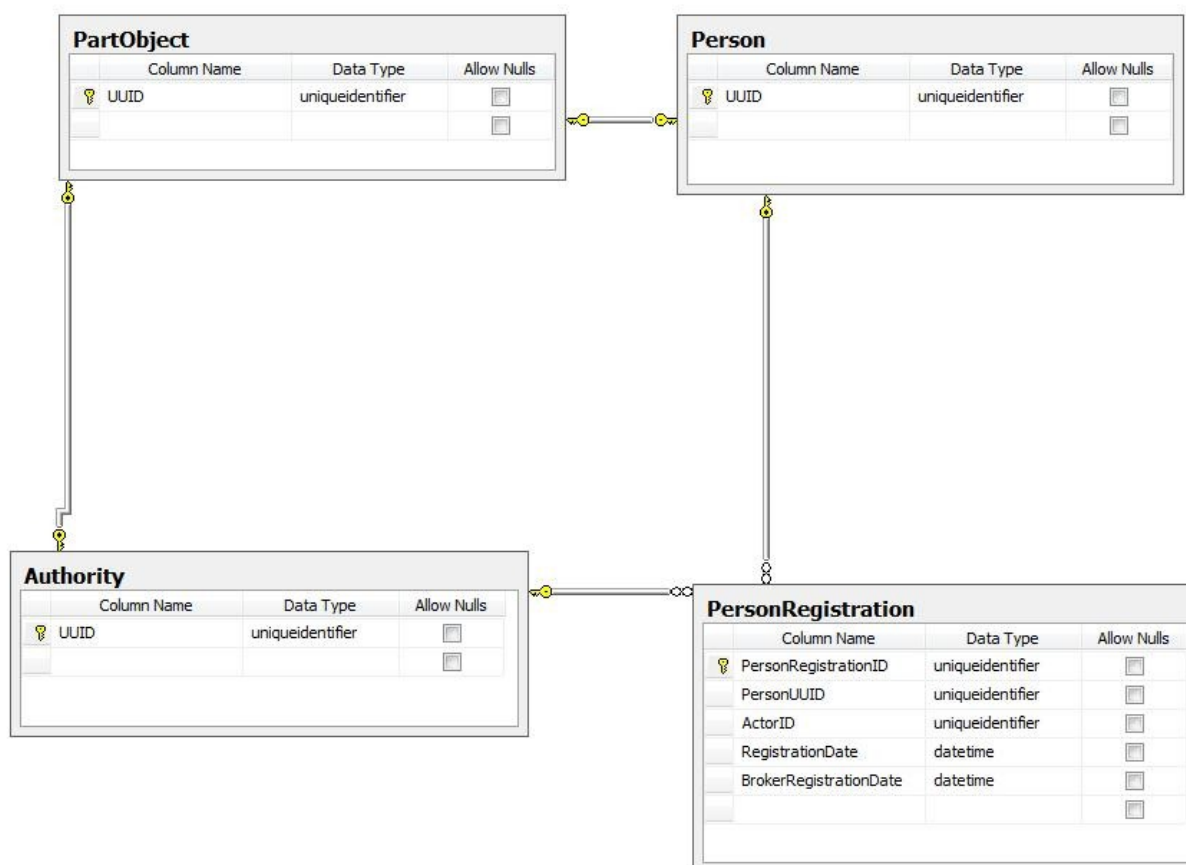
There are support for addition of data providers, e.g., for contact info and medical info.

## Activity



## 6 REGISTRATIONS

### 6.1 Database structure



### 6.2 Time perspectives

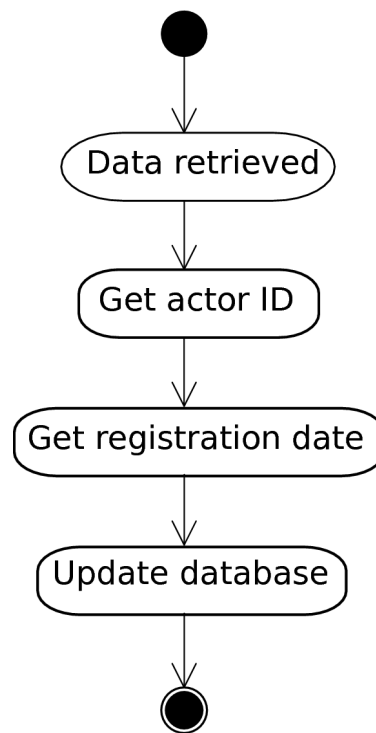
The most important time perspective is the registration time. Effect time will be specified whenever available. An unknown effect start and/or end date is represented using a special value.

### 6.3 Mapping

The concept of a UUID is not known to the system's data providers. Therefore the system use a database table to map persons physical identifiers (UUIDs) to logical identifiers (the CPR number).

The values for the mappings are obtained from a municipal API. If this is not possible, an error is raised.

## 6.4 Registration object creation process



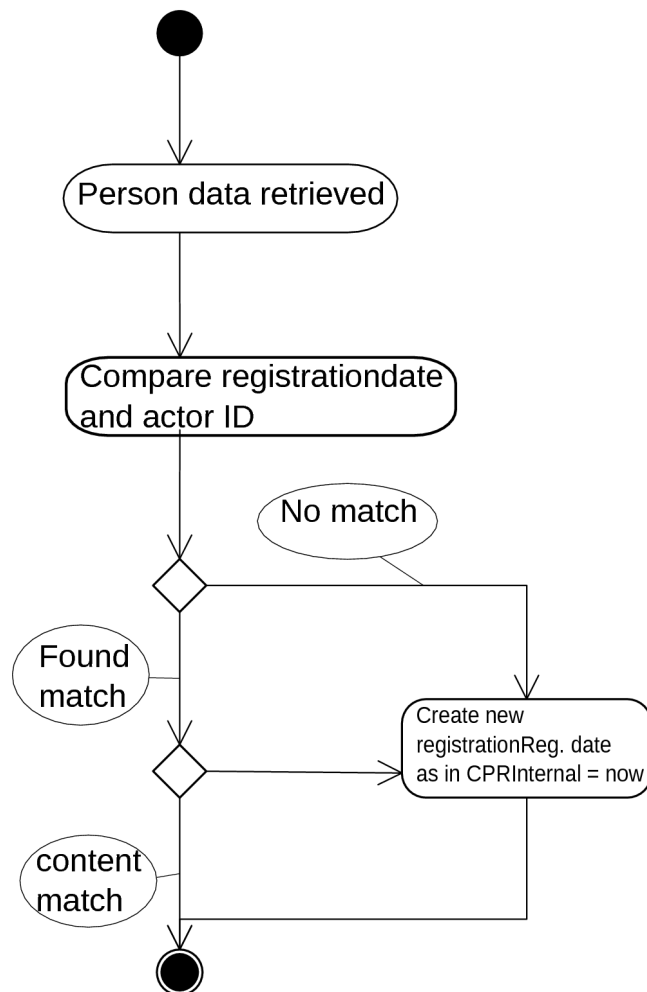
### 6.4.1 Values

Actors can be data providers, e.g., DPR or KMD. Each actor is assigned a fixed UUID. The registration date will be the latest date in the available data. If not possible, then the date when the broker got the data.

Registrations are identified by a 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.

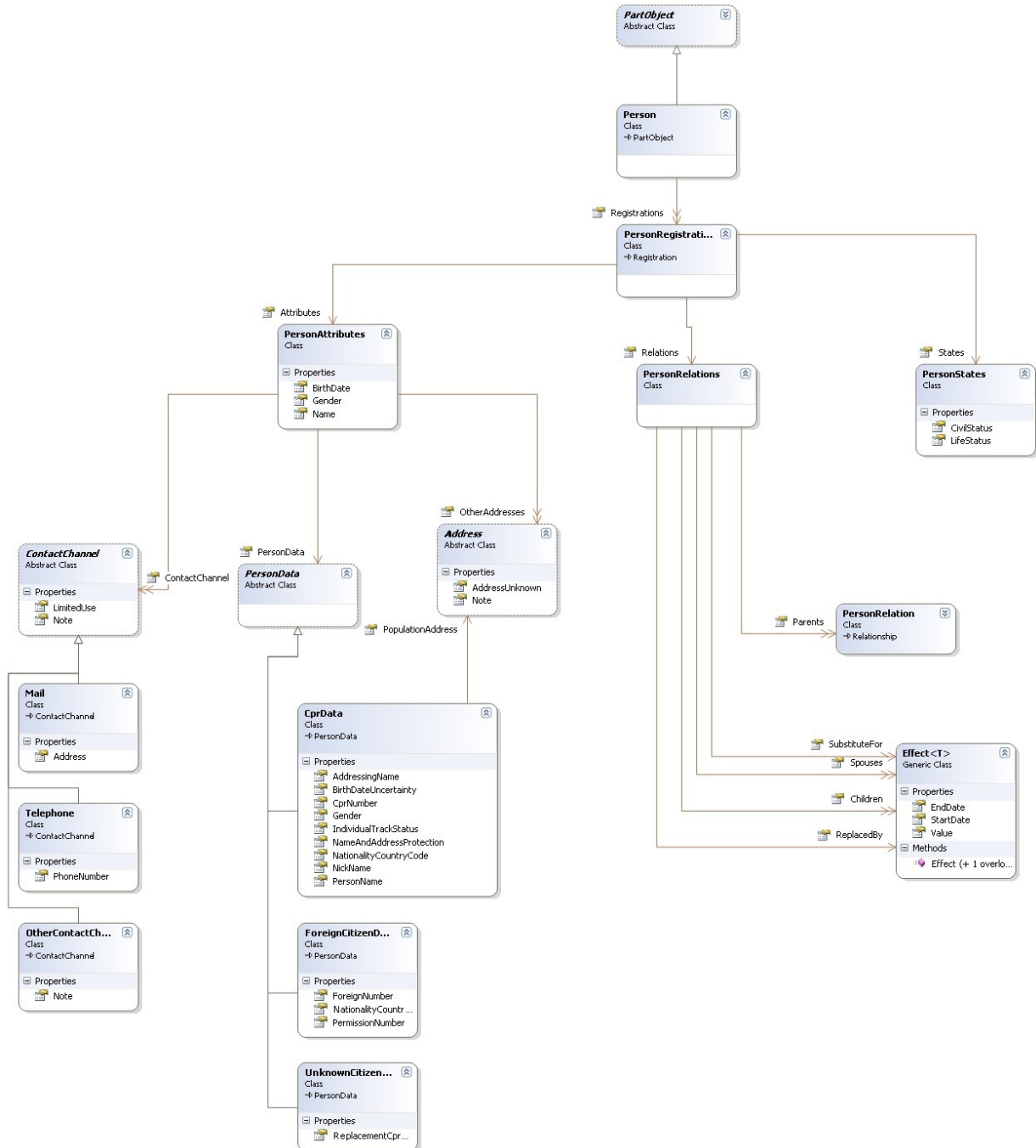
## 6.5 Local database update

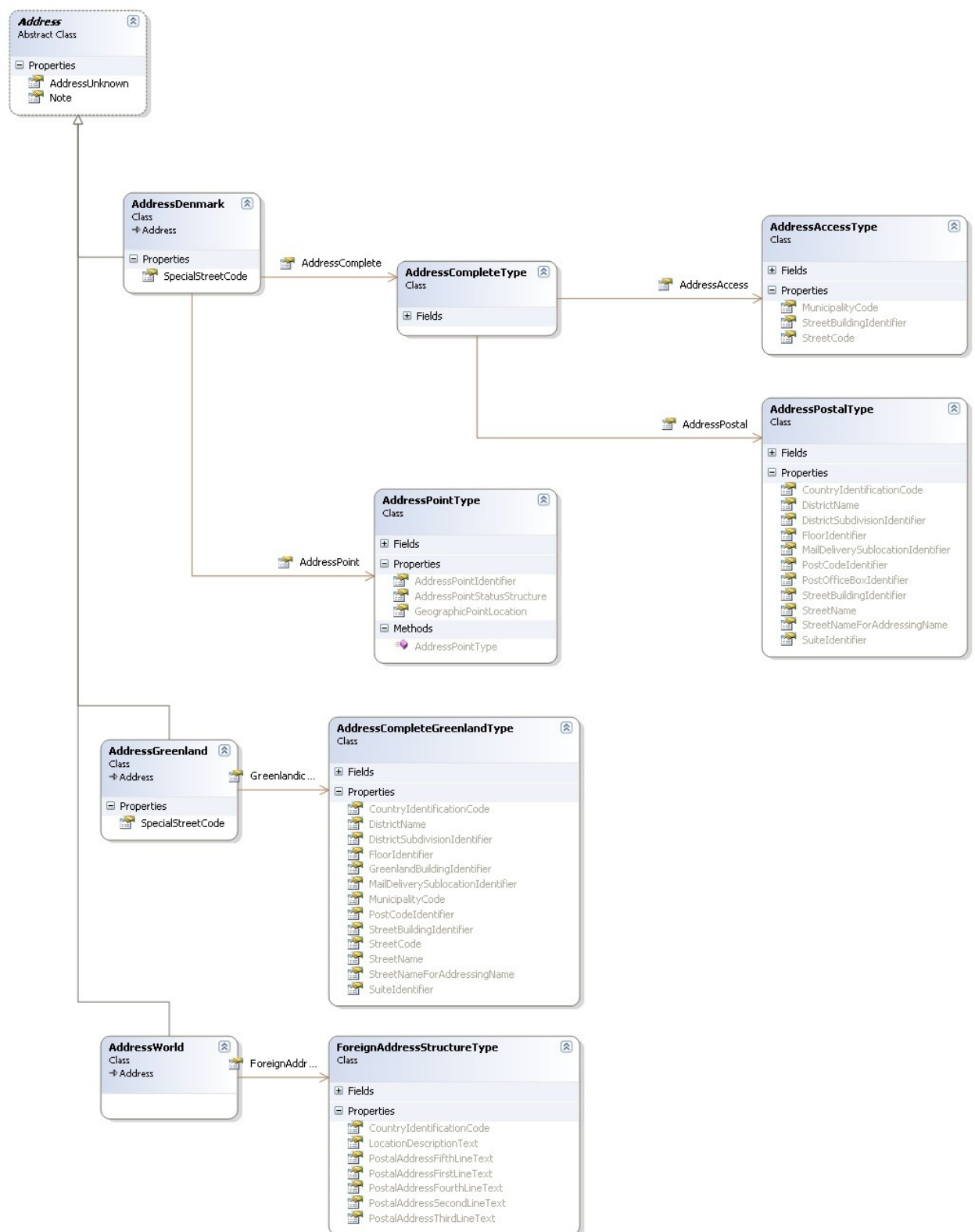
### Activity



# 7 DATA STRUCTURES

The following diagrams describe the details of the registration object.



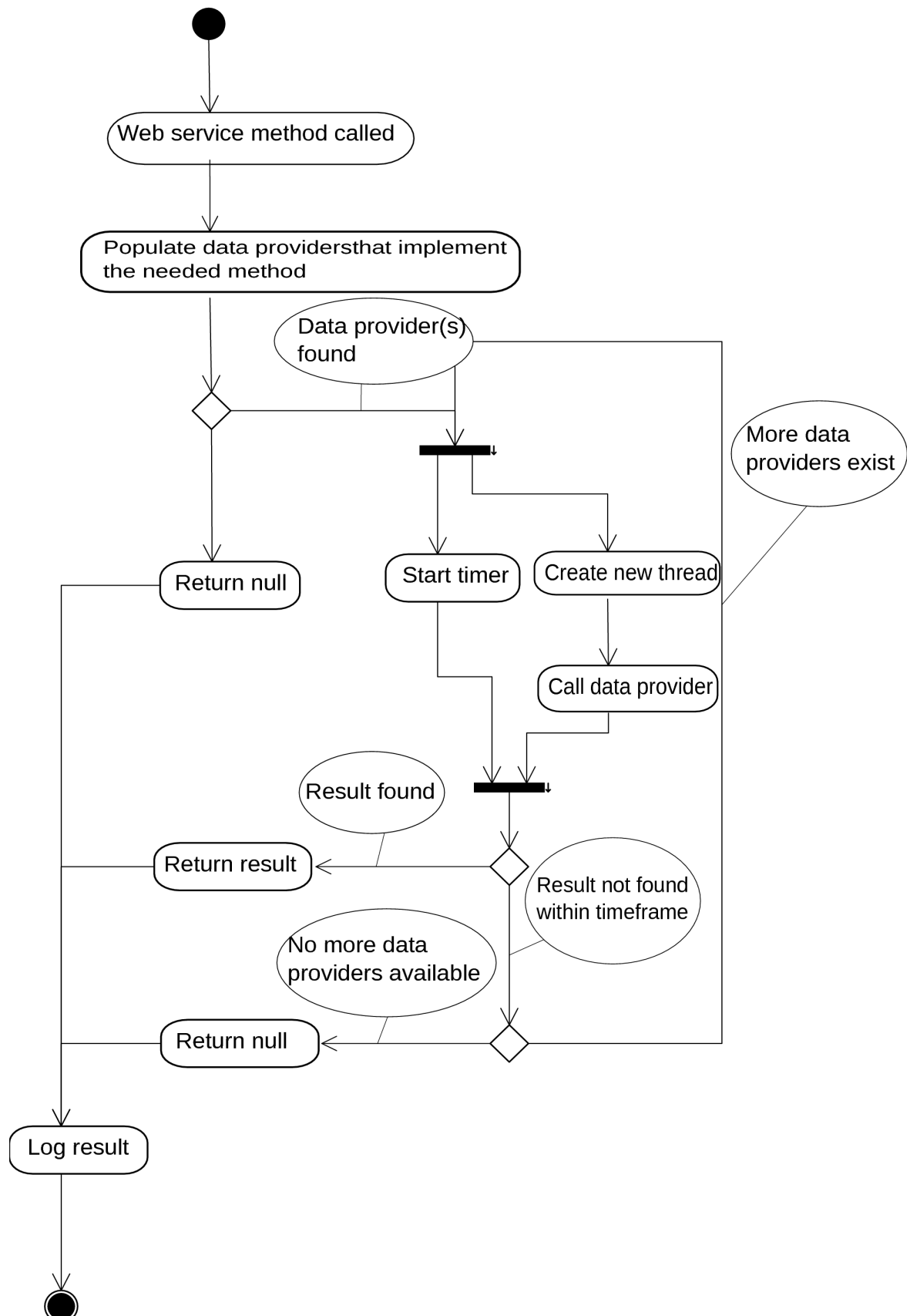




## 8 READ & LIST OPERATIONS

The refresh read operation forces the broker to get new data from data providers.

### Activity

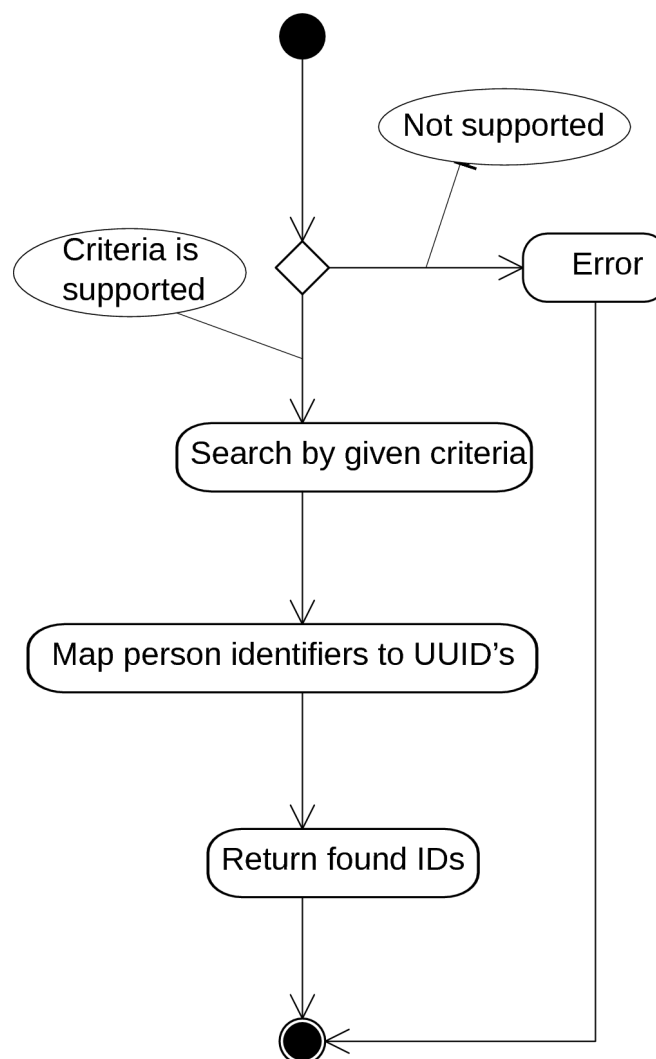


## 9 SEARCH OPERATION

The Search operation allows searching the local CPR Broker Database, i.e., not the external data providers.

It is possible to search for people by their CPR number, Age or Name.

### Activity



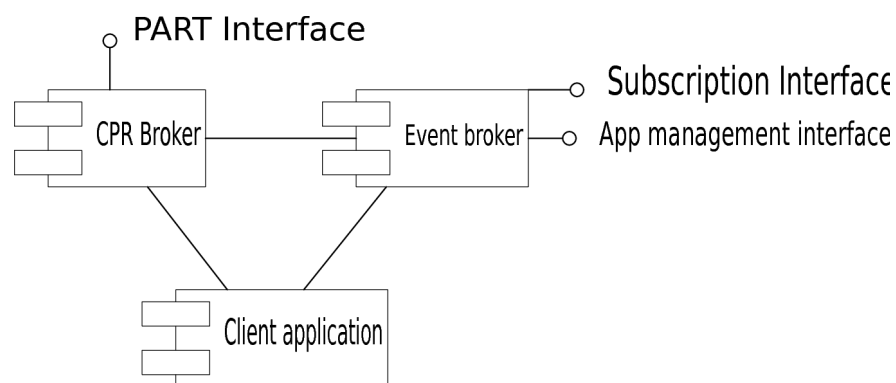
## 10 EVENTS

### 10.1 Client interface

Clients use the event broker web interface to register for events. When registering the client specify both the events it is subscribing to and the channel it want to receive notifications.

A channel can be a web service or a file name on a file system accessible to the broker.

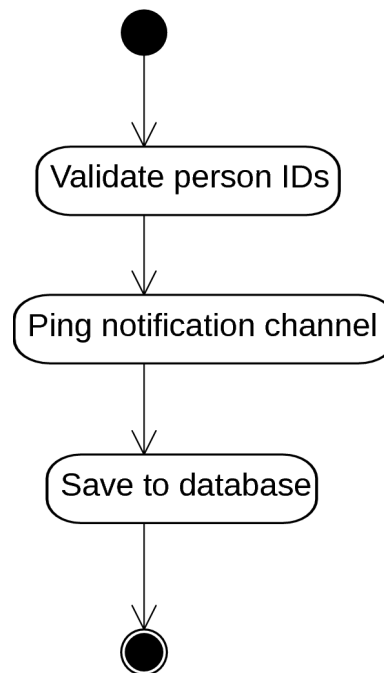
### 10.2 Overall Event Architecture



### 10.3 Subscription flow

## 10.4 Notifications

### Activity



Data change events are detected as soon as they occur.

External events are detected when the PART web interfaces receives refresh calls from external event generators and the refresh call subsequent obtains new data from a data provider. The external event generators can be services that polls data providers.

Birth-date events are detected daily.

The Event Broker service delivers the notifications to subscribers.

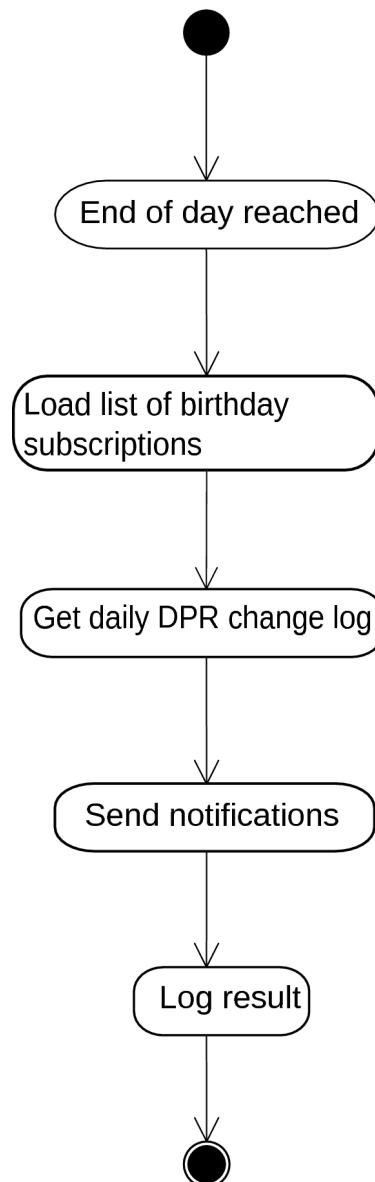
### 10.4.1 Birthday Notifications

Birthday notifications allows a subscriber to specify an age as a number of years and a advance period as a number of days for a list of one or more persons, using their current CPR numbers. Subscribers will receive a notification the advance period before each person on the list reach the specified age.

Birthday notification are generated and delivered once a day. If delivery fails for an event subscriber, it is written as an error the log systems. Client application subscribers that for

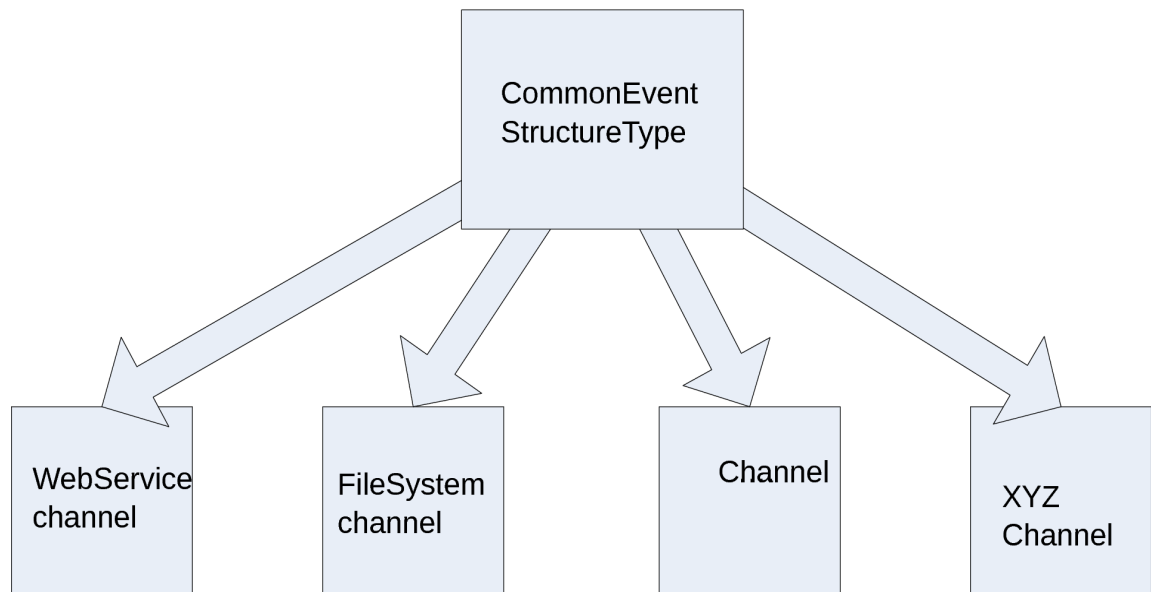
technical reasons did not receive subscriptions have the option of resubscribing with a longer advance period.

### 10.4.2 Data Changed Notifications



Data Changed notifications are generated when the CPR brokers is updated with new information about persons. This can only happen when data is received from external data providers. Data is received from external data providers either through normal client Read operations or through Read Refresh operations. Read Refresh operations can be performed by external services to propagate updates through the broker to broker clients subscribing to the relevant person.

## 10.5 Notification channels



**MAGENTA<sup>aps</sup>**

**adresse**

Stuðiestræde 14, 1.  
1455 Københavk K

**email**

info@magenta-aps.dk

**telefon**

(+45) 33 36 96 96