

Snow agents

Simple Network of Work Agents

The Snow agents system Epidemiology Service

Design document

Authors:
Johan Gustav Bellika

Revision: 0.4

Revision history:

| Issue | Details | Who | Date |
|--------------|---|------------|--------------|
| 0.4 | Revised the design to enable use with the peer-to-peer organisation of the Snow Agent servers. | JG Bellika | 28 July 2005 |
| 0.3 | Added description of failed host problem to unresolved issues section. Synchronised several sections to current design and implementation. | JG Bellika | 26 Oct 2004 |
| 0.2 | Changed the use case to make it more convenient for testing and demonstration purpose. Mission result files are sent to the post office web server. | JG Bellika | 13 Aug 2004 |
| 0.1 | Initial version of Design document | JG Bellika | 06 Aug 2004 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Table of contents

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 4 |
| 1.1 | USE CASE | 4 |
| 1.2 | MISSION SPECIFICATION | 4 |
| 1.3 | SYSTEM CONFIGURATION ISSUES | 5 |
| 1.4 | THE MISSION AGENT LUGGAGE TAGS | 6 |
| 1.5 | TARGET SPECIFICATION | 7 |
| 1.5.1 | <i>The hierarchical organization.....</i> | <i>7</i> |
| 1.5.2 | <i>The peer-to-peer organization.....</i> | <i>8</i> |
| 1.6 | THE DATA CACHE..... | 8 |
| 1.7 | UNRESOLVED ISSUES | 9 |
| 2 | THE EPIDEMIOLOGY MISSION AGENTS..... | 10 |
| 2.1 | THE MAIN MISSION AGENT | 10 |
| 2.1.1 | <i>Configuration</i> | <i>11</i> |
| 2.1.2 | <i>The sub-mission mission specification</i> | <i>11</i> |
| 2.1.3 | <i>Mission results</i> | <i>13</i> |
| 2.2 | THE POST OFFICE AGENT | 13 |
| 2.2.1 | <i>Configuration</i> | <i>14</i> |
| 2.2.2 | <i>The sub-mission mission specification</i> | <i>14</i> |
| 2.2.3 | <i>Mission results</i> | <i>16</i> |
| 2.3 | THE EHR HOST AGENT | 16 |
| 2.3.1 | <i>Configuration</i> | <i>16</i> |
| 2.3.2 | <i>Mission results</i> | <i>17</i> |
| 3 | DISEASE PROFILES | 18 |

1 Introduction

The epidemiology mission collects information about occurrences of a disease in a specified geographic area and plots the occurrences on an interactive map. The map may be viewed and manipulated by the user using a web browser.

1.1 Use case

The user logs on to the Snow agent system using a jabber client. If the EHR system collects epidemiological information automatically, the EHR performs the logon on behalf of the user. Using the client, the user or the EHR system specifies the mission information: ICPC-2 code, current data, onset date and counties or municipalities to collect information from. The mission result file is available via the web server running on the same server as the main_epi mission agent runs and may be viewed using a web browser with a SVG plug-in installed. The mission requester is informed of available mission results using a normal IM message.

1.2 Mission specification

The client or the EHR system builds and sends the following mission specification to a mission controller:

```
<iq type="set" to="<mc jid>" id="epil">
  <query xmlns="jabber:iq:sa_mission">
    <spec>
      <agent missionMode='spread'
        type='main-epidemio'
        version='1.0'
        name='Main-Epidemiology'
        user='GP1-user'
        user_certificate='xxxxxxx'
        requesterjid='<requester JID>' />
      <mission submissions='yes' descendantsCount='3' />
      <visitDuration unit='hours' value='10' />
      <maxTimeDelta unit='hours' value='1' />
      <negotiationLimit unit='minute' value='10' />
      <missionDuration remove_ts='20040623T23:59:59'
        infinite='no' />
      <storage unit='mb' value='10' />
      <targets deviations='no'>
        <host name='<Snow Agent server jid>'
          seq_num='1'
          optional='no'
          status='unknown'
          retries_left='3' />
      </targets>
    </spec>
  </query>
</iq>
```

```
<luggage type='internal'>
  <icpc2>R71</icpc2>
  <dataSource type="cached" maxage="20030211060000"/>
  <cdate>2003-02-11</cdate>
  <odate>2003-01-17</odate>
  <geo level='county'>
    <item>FINNM</item>
    <item>TROMS</item>
    <item>NORDL</item>
  </geo>
</luggage>
<result recipient='<requester JID>' />
</spec>
</query>
</iq>
```

The value of the <Snow Agent server jid> specified as the target is dependent of the organization of the servers. See further details below. For further details about the mission specifications tags see the Snow Agent system mission specification.

1.3 System configuration issues

Dependence of user knowledge or access to a system administrator for configuration of agent missions is a strict requirement for small GP offices. All efforts must therefore be used to avoid using human resources on such tasks. For this mission there are several issues that potentially require end user attention.

The mission specification requires that the client or the user knows which target to run the main mission agent. For the end user to make this decision may be problematic. Such knowledge should be provided by the mechanism. We choose to use pre-configuration of the client in a mission configuration file. The appropriate server to run is also dependent of the organization of the Snow Agent servers.

Peer-to-peer organization: The appropriate server to run the main agent is the local GP office Snow Agent server in the peer-to-peer organization. Dedicated servers may also be used for this purpose.

Hierarchical organization: If the receiving Snow Agent system is the local GP installation, then it knows of the post office server providing relay of messages and may use this as a main mission target. If the receiving system is the post office server then it can run the main mission agent itself. So both the post office server and the EHR host installation may have the same main mission target specification.

The mission specification above don't mention the disease profiles that needs to be applied by the mission agents on the target EHR hosts providing information about infected persons. Which diagnosis codes that are supported must be establish by the system owners and specified in the system configuration. It is of no use to specify an epidemiology mission for a disease that is not supported by the EHR hosts in the specified geographical area.

Pre-configuration of the client may be used to provide information about supported disease

profiles. Specialized agent missions can distribute updated configuration information for mission specification. Use of newly deployed configuration files must however be based on user consent. The actual GP consent, which is stored within the disease profile configuration file, is checked before a mission is allowed to use a disease profile. The GP that produce the EHR data needs to have the final decision about which disease profiles he is willing to support. The consent may be collected using a web page that is able to change the consent field of the disease profiles. The web pages for doing this may be provided by a specialized epidemiology configuration agent.

1.4 The mission agent luggage tags

```
<luggage type='internal'>
  <icpc2>R71</icpc2>
  <dataSource type="cached" maxage="20030211060000"/>
  <cdate>2003-02-11</cdate>
  <odate>2003-01-17</odate>
  <geo level='county'>
    <item>FINNM</item>
    <item>TROMS</item>
    <item>NORDL</item>
  </geo>
</luggage>
```

The icpc2 tag specifies which disease to generate the mission results for. This tag will be enhanced in future versions of the epidemiology mission. As used it identifies which disease profile to use for the epidemiology mission.

The dataSource tag in the luggage specification determines whether to use cached data or data from the source systems (the EHR systems). The type attribute value of “production” means that only data from the EHR systems may be used. If the type attribute contains the value “cached” the maxage attribute is also required. The maxage attribute specifies the maximum age of the data in the cache using the timestamp format “YYYYMMDDHHMMSS”. The data produced by the EHR mission agent is therefore time stamped when extracted from the EHR system. This timestamp is used when data is inserted into the cache and compared to the maxage attribute in the mission specification. If some of the data in the cache is older than the maxage attribute requirement, data is collected from the EHR systems.

The cdate tag specifies the current date. The current date specifies the end of the period data is collected for.

The odate tag specifies the onset date of the disease. It is an important input parameter to the epidemiology system. Combined with the knowledge in the disease profile the start date, infection period, incubation time, and generation time is calculated.

The geo tag specifies the geographical area the user wants information from. There are currently only two levels, “county” level and “municipal” level. The tag must contain a list of item tags listing all codes that specifies the geographical area. For the county level we only support “FINNM”, “TROMS” and “NORDL”, representing respectively Finnmark, Troms and

Nordland counties of northern Norway. For the municipal level we support all municipalities within the above counties. The municipality is specified using the municipality codes which range (not continuously) from 1804 (Bodø) to 2030 (Sør-Varanger). Even if the query specifies which geographical area to search for occurrences, cases outside this area may be reported. This is the case because a patient may consult a GP while traveling or visiting a geographical area other than his/her home. This represents a visualization problem because such cases may have home address outside the visualization area. Such information may be important because an infected person has been in the geographical area and may have infected local patients.

1.5 Target specification

The mission agents need information about which host that provides epidemiology service for a given geographical region. This can easily be provided by a LDAP server. For this limited experiment we do however use a table in the pertussis database to search for which hosts that provides the Epidemiology service. The mission agents query this table to build the target lists for the distributed epidemiology service. The definition of this table is as follows:

```
CREATE TABLE targets (
  Level          CHAR(15),
  Countycode     CHAR(7),
  Municode       CHAR(7),
  AreaDesc       VARCHAR(30),
  Hostname       VARCHAR(40),
  Optional       CHAR(4) DEFAULT 'no',
  Retries        CHAR(3) DEFAULT '3'
);

/* County servers */
insert into targets values ("county","FINNM","FINNM","Finmark
fylke","po.finnmark.health.net","no","3");
....

/* Municipality server */
insert into targets values ("municipal","FINNM","2002","Vardø
kommune","po.finnmark.health.net","no","3");
....

/* GP Offices servers with pertussis in Troms */
insert into targets values ("gp","TROMS","1923","Salangen
legekontor","gp.salangen.health.net","no","3");
....
```

The Snow Agent servers may be organized in at least two different organizations: the hierarchical organization and the peer-to-peer organization. Combinations of these may also be possible. However, we have only considered organizations following one of the two approaches.

1.5.1 The hierarchical organization

In the hierarchical organization the EHR hosts may connect periodically to a post office Snow Agent server. The post office server provides offline storage of messages and processing requests (mission specifications) while the EHR host is offline. When the EHR host connects these messages and requests are forwarded to the EHR host Snow Agent server. This configuration is enabled by the jabber extensions named PO (post office) and POLL (poller). The post office

also provide relay of messages on behalf of the EHR hosts as an external server-to-server (s2s) component. The EHR host participating in the organization may be protected by a firewall that block all incoming connections to the EHR installation.

1.5.2 The peer-to-peer organization

In the peer-to-peer (p2p) organization the EHR host Snow Agent servers must be constantly connected to a common network as the Internet or a dedicated health network. The EHR host must allow incoming connections from Snow Agent servers in remote health institutions. If the GP office Snow Agent server hosts the main mission agent itself, it must be support the network load generated by potentially a large number of host sending sub-mission results to the main epidemiological agent running on the EHR host. The main epidemiology agent may also be hosted on dedicated servers where the users in the GP office have authorization to run agent missions. Dedicated processing servers may inserted into the target table with the level attribute set to “county” or “municipal”. These servers will act as post office servers within the hierarchical organization for all “gp” level servers covered by that geographical area. If no dedicated processing servers are available, the main epidemiology agent must act as the post office server for all EHR hosts covered by the mission. All EHR host servers must be added to the targets table with the level attribute set to “gp”.

1.6 The data cache

The post office and the ehr host agents stores mission results in a data cache to enable fast response when fresh data exists. The mission results are stored in the `epi_cache` table defined by the following SQL statement:

```
CREATE TABLE epi_cache (
  hostname          VARCHAR(20),
  zipcode           CHAR(4),
  icpc2Code         CHAR(4),
  ts                DATETIME,
  date              DATE,
  count             INT
);
```

In addition the agents stores key data about every mission performed in the `mission_result` table defined by the following SQL statement:

```
CREATE TABLE mission_result (
  mission_id        VARCHAR(41),
  hostname          VARCHAR(20),
  icpc2Code         CHAR(4),
  ts                DATETIME,
  currentDate       DATE,
  onsetDate         DATE,
  startDate         DATE
);
```

To be able to determine whether the cache is valid, every date in the query period need to be checked for validity. If one date is missing in the cache we cannot use the cached data. To

support this need we use the following table definition.

```
CREATE TABLE cache_meta (  
    icpc2Code          CHAR(4),  
    ts                 DATETIME,  
    date               DATE  
);
```

The cache is checked for validity using the following SQL statement for every date in the query period.

```
SELECT MIN(ts) > "<max age>" FROM cache_meta WHERE date =  
"<qdate>" AND icpc2Code = "<icpc2code>";
```

The <max age> and <icpc2code> parameters is provided by the mission specification. If the returned value of `MIN(ts) > "<max age>"` is "1", then the cache is valid for that particular <qdate>.

This requires that all EHR hosts covered by the post office are queried and has provided data to the cache when the cache is invalid. If only parts of the geographical area covered by the post office can be queried, then it is necessary to store spatial information in the cache_meta table, and check the validity of the cache for the spatial dimension as well. This can be done by building the host list for a potential mission and check the validity of the cache for each host in the mission itinerary. We did not implement support for checking the validity of the cache for the spatial dimension in the experimental service.

The mission results may be produced from the cache by the following SQL statement.

```
SELECT zipcode, date, SUM(count) FROM epi_cache  
WHERE date >= "<start date>" AND date <= "<end date>" AND  
icpc2Code = "<icpc2code>" GROUP BY zipcode,date ORDER BY date  
ASC;
```

1.7 Unresolved issues

The client does not know the coverage of the mission results. This data must be produced by the agent mission.

If one of the hosts providing data from a geographical area is down or unavailable it will be described in the final mission specification held by the main mission controller. This data should therefore be made available to the mission requester, in this case the main EHR agent. Based on this data, the agent could provide information about the coverage of the data provided.

An epidemiology service should provide both incident and prevalence rates. To be able to produce this kind of data we need access to demographic data which we did not have during construction of the system. In future versions these rates should be provided. This kind of data may be built directly from the patient records on the EHR hosts and may therefore support a resolution down to zip code level.

2 The epidemiology mission agents

The epidemiology service consists of three types of agents.

- 1.The main mission agent: main_epi
- 2.The post office agent: po_epi
- 3.The EHR host agent: ehr_epi

Each mission agent type has different responsibilities as outlined in the following sub sections.

2.1 The main mission agent

The responsibility of the main_epi mission agent is to distribute po_epi and ehr_epi mission agents to all the post office and EHR host Snow Agent servers necessary to cover the geographical area requested by the user. It is also responsible for merging all partial results together to produce the final SVG map result file. This SVG file is available via the local web server running on the Snow Agent server. The mission requester is informed of available mission result using a normal IM message providing a link to the result SVG file.

Because the main_epi mission agent may be employed on a post office server or an EHR host in the p2p organization, the main_epi mission agent may need to perform the operation of a post office agent (po_epi), if this host is part of the epidemiological query. If only a single post office server is necessary to cover the geographical area, the main mission agent will perform the task of the post office agent. If no post office servers are involved (p2p) the main_epi agent will perform the operations of the po_epi agent.

Information about which post office server or EHR host servers that provides service for a geographical area is currently stored in the local database for each post office server. See section 1.5 above.

2.1.1 Configuration

The main_epi epidemiology agent configuration is listed below.

```
<!-- configuration of the main epi agent -->
<service id="ma">
  <load>
    <epi_main>../bin/epi.so</epi_main>
  </load>
  <main_epi xmlns="jabber:config:main_epi">
    <!-- include mission agent configuration contents -->
    <agd_jid>
      <jabberd:cmdline flag="a">agd.dipato.stud.cs.uit.no</jabberd:cmdline>
    </agd_jid>
    <local_mc>
      <jabberd:cmdline flag="m">mc.dipato.stud.cs.uit.no</jabberd:cmdline>
    </local_mc>
    <serverhost>
      <jabberd:cmdline flag="h">dipato.stud.cs.uit.no</jabberd:cmdline>
    </serverhost>
    <user><jabberd:cmdline flag="U">aguser</jabberd:cmdline></user>
    <birthid><jabberd:cmdline flag="b">agbirthid</jabberd:cmdline></birthid>
    <home_path><jabberd:cmdline flag="H">/usr/tmp/SAS/agd/</jabberd:cmdline></home_path>
    <timeout_value>360</timeout_value>
    <svg_js_file>../epi/map-scripts.js</svg_js_file>
    <result_path>/users/snow/result/</result_path>
    <result_url_prefix>http://dipato.stud.cs.uit.no/epi/</result_url_prefix>
    <disease_profile_fn>../conf/disease_profiles.xml</disease_profile_fn>
    <db_host>localhost</db_host>
    <db_user>gp_user</db_user>
    <db_passwd>isasecret</db_passwd>
    <db_dbname>pertussis</db_dbname>
  </main_epi>
</service>
```

The following configurations options are provided by the agent daemon on instantiation of the agents. These configuration options are available to all agents instantiated by the agent daemon.

- a agd_jid
- m local mc jid
- h server hostname
- U jabber username
- b agent birthid
- H home path for the agent

More details are available in the example configuration files in the /conf directory of the source distribution.

2.1.2 The sub-mission mission specification

The main_epi mission agent sends the following mission specification to the local mission controller.

```

<iq type="get" to="<mc jid>" id="epil">
  <query xmlns="jabber:iq:sa_submission">
    <spec>
      <agent missionMode='spread'
        type='po-epidemiology'
        version='1.0'
        name='PO-Epidemiology'
        user='GP1-user'
        user_certificate='xxxxxxx'
        requesterjid='<main mission agent JID>' />
      <mission submissions='yes' descendantsCount='2' />
      <visitDuration unit='hours' value='10' />
      <maxTimeDelta unit='hours' value='1' />
      <negotiationLimit unit='minute' value='10' />
      <missionDuration remove_ts='20040623T23:59:59'
        infinite='no' />
      <storage unit='mb' value='10' />
      <targets deviations='no'>
        <host name='po.finnmark.health.net'
          optional='no'
          status='unknown'
          retries_left='3' />
        <host name='po.nordland.health.net'
          optional='no'
          status='unknown'
          retries_left='3' />
      </targets>
      <luggage type='internal'>
        <icpc2>R71</icpc2>
        <dataSource type="cached" maxage="20030211060000" />
        <cdate>2003-02-11</cdate>
        <odate>2003-01-17</odate>
        <geo level='county'>
          <item>FINNM</item>
          <item>TROMS</item>
          <item>NORDL</item>
        </geo>
      </luggage>
      <result recipient='<main mission agent JID>' />
    </spec>
  </query>
</iq>

```

The luggage of the mission specification has now been translated into two mission targets, one snow agent post office server for each of the two remaining target counties.

The main mission agent needs to be operational until all post office agents have completed their tasks and reported their results. In the example above the main mission agent is run from po.troms.health.net post office server.

2.1.3 Mission results

The result produced by the main_epi mission agent is a SVG file which shows a map of the geographical region specified by the user. The file is given the name: “<main agent birthid>.svg”. The map is interactive and can show the occurrences of the disease specified in the mission specification. The user can click on zip codes or polygons representing municipalities to view a graph of the disease occurrences. See Figure 1. The SVG file is built from the mission results (XML encoded) produced by the post office or EHR host agents and potentially also the main mission agent. See mission results below. The SVG file is stored on host where the main_epi mission agent runs and is available to the user as a url to the local web server. The SVG file is based on the Vienna example provided by the carto.net web site. Andreas Neumann has also given his consent for use of the Vienna example JavaScript code which is embedded in the SVG file.

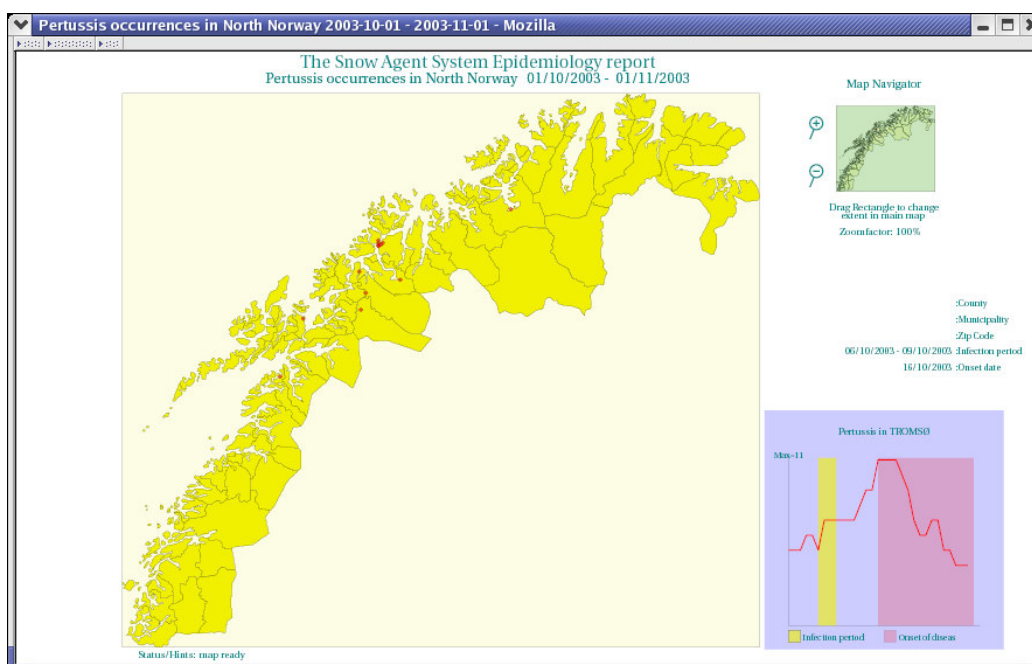


Figure 1 Epidemiology map for Northern Norway

2.2 The post office agent

The post office agent, po_epi, is responsible for spreading mission agents to all EHR hosts attached to the post office in the hierarchical organization, or the EHR hosts it covers in the p2p organization. In both organizations the po_epi agent covers parts or the whole area of the geographical area specified in the mission specification. The targets are identified by doing a lookup into the targets table. See section 1.5 above. The post office agents merge the results from the EHR hosts into a final list which is sent to the main mission agent.

The post office agent is responsible for caching the mission results if several requests arrive close

in time. In the mission specification above the source is specified as cached data. In such a case the post office agent don't need to send mission agents to the EHR host. It can choose to provide the data directly from its data cache.

As mentioned in section 1.6 above, the client cannot know the coverage of the data reported in the final result file. In future versions, the post office agent may be assigned the responsibility of producing coverage information by requesting information about the amount of people serviced by each of the GP offices serviced by the post office server.

2.2.1 Configuration

The post office epidemiology agent configuration is listed below. It is similar to the main mission agent described above. See section 2.1.1.

```
<!-- configuration of the po epi agent -->
<service id="po_epi">
  <load>
    <po_epi>../bin/epi.so</po_epi>
  </load>
  <po_epi xmlns="jabber:config:po_epi">
    <!-- include mission agent configuration contents -->
    <agd_jid><jabberd:cmdline flag="a">agd.dipato.stud.cs.uit.no</jabberd:cmdline></agd_jid>
    <local_mc><jabberd:cmdline flag="m">mc.dipato.stud.cs.uit.no</jabberd:cmdline></local_mc>
    <serverhost><jabberd:cmdline flag="h">dipato.stud.cs.uit.no</jabberd:cmdline></serverhost>
    <user><jabberd:cmdline flag="U">aguser</jabberd:cmdline></user>
    <birthid><jabberd:cmdline flag="b">agbirthid</jabberd:cmdline></birthid>
    <home_path><jabberd:cmdline flag="H">/usr/tmp/SAS/agd/</jabberd:cmdline></home_path>
    <disease_profile_fn>../conf/disease_profiles.xml</disease_profile_fn>
    <timeout_value>5</timeout_value>
    <db_host>localhost</db_host>
    <db_user>gp_user</db_user>
    <db_passwd>isasecret</db_passwd>
    <db_dbname>pertussis</db_dbname>
  </po_epi>
</service>
```

2.2.2 The sub-mission mission specification

The Post office agent is responsible for sending the following mission specification to the local mission controller. The mission specification is inherited from the mission specification for the po_epi agent. Only the following tags and attributes are altered by the post office epidemiology agent.

- Agent.type attribute, Agent.name attribute, Agent.requesterjid.
- The target list is replaced by a hosts list produced by querying the targets table in the local database.
- The result recipient attribute is set to the jid of the post office epidemiology agent.

The altered attributes and tags are marked in the example listing below.

```
<iq type="get" to="<mc jid>" id="epil">
  <query xmlns="jabber:iq:sa_submission">
    <spec>
      <agent missionMode='spread'
        type='EHR-epidemio'
        version='1.0'
        name='GP-Epidemiology'
        user='GP1-user'
        user_certificate='xxxxxxx'
        requesterjid='<post office agent JID>' />
      <mission submissions='no' descendantsCount='1' />
      <visitDuration unit='hours' value='10' />
      <maxTimeDelta unit='hours' value='1' />
      <negotiationLimit unit='minute' value='10' />
      <missionDuration remove_ts='20040623T23:59:59'
        infinite='no' />
      <storage unit='mb' value='10' />
      <targets deviations='no'>
        <host name='<GP office server 1>'
          optional='no'
          status='unknown'
          retries_left='3' />
        <host name='<GP office server 2>'
          optional='no'
          status='unknown'
          retries_left='3' />
      </targets>
      <luggage type='internal'>
        <icpc2>R71</icpc2>
        <dataSource type="cached" maxage="20030211060000" />
        <cdate>2003-02-11</cdate>
        <odate>2003-01-17</odate>
        <geo level='county'>
          <item>FINNM</item>
          <item>TROMS</item>
          <item>NORDL</item>
        </geo>
      </luggage>
      <result recipient='<post office agent JID>' />
    </spec>
  </query>
</iq>
```

2.2.3 Mission results

The mission results produced by the `po_epi` agent looks like the XML listing below. This requires that each post office has the sole responsibility to produce epidemiology data from an unique set of EHR hosts in the geographical area. If not, the final occurrence numbers will contain duplicates. If several post office servers uses data from the same EHR system, then disguised identification of patients must be used to ensure that occurrences are only counted once.

```
<result>
  <po_epi host='po.troms.health.net'>
    <epi_data startdate='2003-01-19'
      currentdate='2003-02-11'
      onsetdate='2003-02-09'
      ts='2003-02-11 01:26:54'
      icpc2code='R71'
      icpc2desc='Pertussis'>
      <zip code="9012">0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1</zip>
      <zip code="9022">1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,1,1,2</zip>
      <zip code="9024">0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0</zip>
      ....
    </epi_data>
  </po_epi>
</result>
```

2.3 The EHR host agent

The EHR host agent is responsible for extracting statistical data from the local EHR database or the local cache, produce the local mission results, store the result in the local cache and send it to the post office server. The EHR host agent uses its local disease profiles to produce the data requested.

2.3.1 Configuration

```
<!-- configuration of the ehr epi agent -->
<service id="ma">
  <load>
    <epi_ehr>../bin/epi.so</epi_ehr>
  </load>
  <ehr_epi xmlns="jabber:config:ehr_epi">
    <!-- include mission agent configuration contents -->
    <agd_jid><jabberd:cmdline flag="a">agd.dipato.stud.cs.uit.no</jabberd:cmdline></agd_jid>
    <local_mc><jabberd:cmdline flag="m">mc.dipato.stud.cs.uit.no</jabberd:cmdline></local_mc>
    <serverhost><jabberd:cmdline flag="h">dipato.stud.cs.uit.no</jabberd:cmdline></serverhost>
    <user><jabberd:cmdline flag="U">aguser</jabberd:cmdline></user>
    <birthid><jabberd:cmdline flag="b">agbirthid</jabberd:cmdline></birthid>
    <home_path><jabberd:cmdline flag="H">/usr/tmp/SAS/agd/</jabberd:cmdline></home_path>
    <timeout_value>360</timeout_value>
    <disease_profile_fn>../conf/disease_profiles.xml</disease_profile_fn>
    <db_host>localhost</db_host>
    <db_user>gp_user</db_user>
    <db_passwd>isasecret</db_passwd>
    <db_dbname>pertussis</db_dbname>
  </ehr_epi>
</service>
```


3 Disease profiles

The disease profiles are necessary to make the epidemiology service generic and support GP consent. The default content of a disease profile must always contain “not allowed” in the consent tag. The GP have to update this value to the contents “allowed”. The time values is used by the service to calculate start and end date for the query period.

```
<profiles>
  <disease code="R71">
    <name>Pertussis</name>
    <minIncubation>7</minIncubation>
    <maxIncubation>10</maxIncubation>
    <minGeneration>7</minGeneration>
    <maxGeneration>10</maxGeneration>
    <contagiousWithMed>5</contagiousWithMed>
    <contagiousWithoutMed>21</contagiousWithoutMed>
    <consent>allowed</consent>
  </disease>
  <disease code="R80">
    <name>Influenza</name>
    <minIncubation>2</minIncubation>
    <maxIncubation>3</maxIncubation>
    <minGeneration>1</minGeneration>
    <maxGeneration>2</maxGeneration>
    <contagiousWithMed>7</contagiousWithMed>
    <contagiousWithoutMed>7</contagiousWithoutMed>
    <consent>allowed</consent>
  </disease>
  <disease code="N71">
    <name>Meningitis</name>
    <minIncubation>2</minIncubation>
    <maxIncubation>10</maxIncubation>
    <minGeneration>3</minGeneration>
    <maxGeneration>3</maxGeneration>
    <contagiousWithMed>?</contagiousWithMed>
    <contagiousWithoutMed>?</contagiousWithoutMed>
    <consent>not allowed</consent>
  </disease>
</profiles>
```