Spine 101

Technical Connection Quick-Start

Published October 2016

Information and technology
for better health and care

Contents

Contents

Introduction

This document is intended to provide a sufficient but brief introduction to the various technical steps that a supplier needs to perform in order to connect to the Spine2 instance running at a connect-a-thon.

For all new Spine2 interfaces short implementation guides are being created which contain the majority of information a consuming system will require to utilise the service. At this time the follow implementation guides are available:

- EPS_prescription_tracker_implimentation_guide
- FGM_implimentation_guide
- Spine mini services 1.2 (itk_smsp_0.1.0_20140724000001)[1]
- The MESH API specification
- The SSP implementation

In some areas this guide refers out to the formal “External Interface Specification” (EIS) document originally from the Spine1 programme. Currently the latest version of the EIS is 12.2. For new interfaces introduced since August 2014 the EIS is not required. However, for those wishing to utilise historic services which have not yet been refreshed (such as EPS or SCR retrievals) communications based on the EIS will be required.

Locating and connecting to the Spine2

Certificates

In order to connect to Spine services you will need to obtain a Spine certificate. Typically this forms part of a registration process on the Endpoint Registration System. For the purposes of a connectathon you will be provided with temporary certificates, and the CA certificate for establishing a trust chain.

These are available from NHS Digital staff on the day.

You will not be able to use these certificates except for connecting to Spine running on laptops provided at the connectathon. They are not valid for the NHS Spine or the Open Test environment.

All connections to Spine must be encrypted with a certificate, otherwise they will be rejected.

Spine URLs

Once you have a certificate installed you will typically need to connect to the URL of a spine service. Specific Spine service URLs provided in the relevant sections below.

There are a number of different interfaces that you may need to use to access and update data on the Spine. The Spine2 service supports all of the original messaging interfaces from Spine1 as well as an increasing number of new, simpler services interfaces.

You should add each server name you require to your hosts file, with the IP address you are given on the day for the local spine instance running on a NHS Digitial laptop.

[1] This is a previously existing ITK specification describing the SMSP interface which Spine2 has implimented

GUIDs

Globally Unique Identifiers (GUIDs) are used in messages, and should be generated in accordance with RFC4122.

An example in python to generate a new GUID:

```
#!/usr/bin/python
import uuid
print str(uuid.uuid4()).upper()
```

Output:

```
A3A4C78B-02C1-4432-A224-FAF854F867A7
```

Spine1 interfaces

Spine1 supported a number of synchronous and asynchronous transport formats both for accessing Spine data and for forwarding data via the Spine to other parties. Chapter 3 of the EIS, the “Message Interaction Map[2]” describes which messages use which possible Spine1 formats.

These formats are generally ebXML + HL7, or SOAP, and are either synchronous or asynchronous. Asynchronous messaging formats require an endpoint registration for the asynchronous response to be sent back to.

Some examples are provided in the next sections, and will be available on the day in electronic form.

PDS Retrieval

Demographic data such as addresses, date of birth, telephone numbers, are stored in the demographic record. These can be retrieved simply using a PDS retrieval if the NHS number is known. An example retrieval is shown below (for the connect-a-thon, use 990101234567 as the to ASID and 230811201097 as the from ASID):

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:hl7="urn:hl7-
org:v3">
      <SOAP-ENV:Header>
            <wsa:MessageID>uuid:{{uuid}}</wsa:MessageID>
      <wsa:Action>urn:nhs:names:services:pdsquery/QUPA_IN000008UK05</wsa:Action>
            <wsa:To>https://pds-sync.national.ncrs.nhs.uk/syncservice-
pds/pds</wsa:To>
            <wsa:From>
                  <wsa:Address></wsa:Address>
            </wsa:From>
            <hl7:communicationFunctionRcv>
                  <hl7:device>
```

[2] See for example document “2087 EIS11.8--Part 3--MessageInteractionMap.doc”

```
                        <hl7:id root="1.2.826.0.1285.0.2.0.107"
                                                    extension="{{to ASID}}"/>
                  </hl7:device>
            </hl7:communicationFunctionRcv>
            <hl7:communicationFunctionSnd>
```

Replace to and from ASIDs with those given to you.

```
                  <hl7:device>
                        <hl7:id root="1.2.826.0.1285.0.2.0.107"
                                                    extension="{{from ASID}}"/>
                  </hl7:device>
            </hl7:communicationFunctionSnd>
      </SOAP-ENV:Header>
      <SOAP-ENV:Body>
```

HL7 payload begins here

```
            <QUPA_IN000008UK05 xmlns="urn:hl7-org:v3">
                  <id root="7101DEAD-9F4E-468B-B8D3-1CB4B483DEDD"/>
                  <creationTime value="20101125153000"/>
                  <versionCode code="3NPfIT7.2.02"/>
                  <interactionId root="2.16.840.1.113883.2.1.3.2.4.12"
extension="QUPA_IN000008UK05"/>
                  <processingCode code="P"/>
                  <processingModeCode code="T"/>
                  <acceptAckCode code="NE"/>
                  <communicationFunctionRcv>
                        <device classCode="DEV" determinerCode="INSTANCE">
                              <id root="1.2.826.0.1285.0.2.0.107"
                                                      extension="{{to ASID}}"/>
                        </device>
                  </communicationFunctionRcv>
                  <communicationFunctionSnd>
                        <device classCode="DEV" determinerCode="INSTANCE">
                              <id root="1.2.826.0.1285.0.2.0.107"
                                                      extension="{{from ASID}}"/>
                        </device>
                  </communicationFunctionSnd>
                  <ControlActEvent classCode="CACT" moodCode="EVN">
                        <author1 typeCode="AUT">
                              <AgentSystemSDS classCode="AGNT">
                                    <agentSystemSDS classCode="DEV"
determinerCode="INSTANCE">
                                          <id root="1.2.826.0.1285.0.2.0.107"
                                                       extension="{{from ASID}}"/>
                                    </agentSystemSDS>
                              </AgentSystemSDS>
                        </author1>
                        <query xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-
org:v3 ../../Schemas/QUPA_MT000003UK02.xsd">
                    <historicDataIndicator>
                        <value code="0"
codeSystem="2.16.840.1.113883.2.1.3.2.4.17.36"/>
                        <semanticsText>historicDataIndicator</semanticsText>
                    </historicDataIndicator>
                    <person.id>
```

Replace with NHS Number to be retrieved

```
                        <value root="2.16.840.1.113883.2.1.4.1"
                                                      extension="{{nhsNumber}}"/>
                        <semanticsText>person.id</semanticsText>
                    </person.id>
                    <retrievalItem>
                        <semanticsText>person.allData</semanticsText>
                    </retrievalItem>
                </query>
```

```
                  </ControlActEvent>
            </QUPA_IN000008UK05>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

To send the command, you need to make a HTTP POST command to https://SERVER.cfh.nhs.uk/smsp/pds. It requires the following headers:

```
SOAPAction: urn:nhs:names:services:pdsquery/QUPA_IN000008UK05
content-type: text/xml
```

Add the client cert and key to your keystore. Depending on the programming language (e.g. Java) you may have to add the the ca.cert in a separate keystore.

The following curl command (from Linux) will send the file through and receive the response:

```
curl -XPOST -H "SOAPAction: urn:nhs:names:services:pdsquery/QUPA_IN000008UK05" -H
"content-type: text/xml" --cert ehitestClient.crt --key ehitestClient.key --cacert
ca.crt -d @retrievalquery.xml https://SERVER.cfh.nhs.uk/sync-service
```

Simple Trace

Simple trace is used when the NHS Number is unknown, but you have the name, gender, DOB and address. It will only return one result.

An Example will be provided.

Advanced Trace (Async)

Advanced Trace is more complex as it is an asynchronous interaction. You will need a listener (see python listener example below) and you will need to register an ASID. See a member of the NHS Digital team for help.

```
from tornado.httpserver import HTTPServer
import tornado.ioloop
import tornado.web
import ssl

class PositiveHandler(tornado.web.RequestHandler):
    def initialize(self, servicesDict):
        pass

    def post(self):
        print self.request.body

servicesDict = {}

application = tornado.web.Application([(r"/spine", PositiveHandler,
dict(servicesDict=servicesDict))])
crt = 'server.crt'
key = 'server.key'
ca = 'ca.crt'
httpsServer = HTTPServer(application,
                         ssl_options=dict(certfile=crt,
                         keyfile=key,
                         ca_certs=ca,
                         cert_reqs=ssl.CERT_REQUIRED))
```

```
httpsServer.listen(443)
tornado.ioloop.IOLoop.instance().start()
```

An example message will be provided.

New Spine2 Interfaces

SMSP

Spine Mini Services Provider (SMSP) provides access to directly query a subset of Spine services for demographic information (PDS). However, all SMSP interfaces are synchronous, removing the requirement for endpoint registration and making development simpler.

The following functionality is possible via SMSP:

- Verify NHS Number using a set of demographic details
- Search for patient details using the NHS Number and DoB
- Search for patient details based upon Name, DoB, and Gender
- Search for a patient NHS number using demographic details
- Search for demographic details based upon a search using NHS Number or demographic information

Example SMSP message (Get patient details based on Name, DoB, and Gender)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:wsa="http://www.w3.org/2005/08/addressing"
               xmlns:itk="urn:nhs-itk:ns:201005">
    <soap:Header>
        <wsa:MessageID>B9A2F260-0590-11E6-B62A-0800270849FF</wsa:MessageID>
        <wsa:Action>urn:nhs-itk:services:201005:getPatientDetails-v1-
0</wsa:Action>

<wsa:To>http://SERVER.cfh.nhs.uk:8080/spinemessagehandler/itk/getPatientDetails-
v1-0</wsa:To>
        <wsa:From>
            <wsa:Address>http://www.cloud-harness.co.uk</wsa:Address>
        </wsa:From>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
            <wsu:Timestamp xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="D6CD5232-14CF-11DF-9423-1F9A910D4703">
                <wsu:Created>2016-04-19T08:55:42</wsu:Created>
                <wsu:Expires>2016-04-19T18:24:42</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken>
                <wsse:Username>www.cloudharness.co.uk</wsse:Username>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <itk:DistributionEnvelope>
```

```
            <itk:header service="urn:nhs-itk:services:201005:getPatientDetails-v1-
0" trackingid="B9A85AF2-0590-11E6-B62A-0800270849FF">
                <itk:auditIdentity>
                    <itk:id type="2.16.840.1.113883.2.1.3.2.4.18.27" uri="urn:nhs-
uk:identity:ods:rhm:team1:C"/>
                </itk:auditIdentity>
                <itk:manifest count="1">
                    <itk:manifestitem id="uuid_B9A45D80-0590-11E6-B62A-
0800270849FF" mimetype="text/xml"
                                profileid="urn:nhs-
en:profile:getPatientDetailsRequest-v1-0"
                                base64="false" compressed="false"  encrypted="false"/>
                </itk:manifest>
                <itk:senderAddress uri="urn:nhs-uk:addressing:ods:rhm:team1:C"/>
            </itk:header>
            <itk:payloads count="1">
                <itk:payload id="uuid_B9A45D80-0590-11E6-B62A-0800270849FF" >
                    <getPatientDetails-v1-0  xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" moodCode="EVN"
classCode="CACT">
                        <id root="FED2C1D0-02D6-450D-88CB-616D56AE2CD6"/>
                        <code codeSystem="2.16.840.1.113883.2.1.3.2.4.17.284"
                            code="getPatientDetailsRequest-v1-0"/>
                        <queryEvent>
                            <Person.DateOfBirth>
                                <value value="20111226"/>
                                <semanticsText>Person.DateOfBirth</semanticsText>
                            </Person.DateOfBirth>
                            <Person.Gender>
                                <value code="1"
                                  codeSystem="2.16.840.1.113883.2.1.3.2.4.16.25"/>
                                <semanticsText>Person.Gender</semanticsText>
                            </Person.Gender>
                            <Person.NHSNumber>
                                <value root="2.16.840.1.113883.2.1.4.1"
                                                    extension="9450827834"/>
                                <semanticsText>Person.NHSNumber</semanticsText>
                            </Person.NHSNumber>
                            <Person.Name>
                                <value>
                                    <given>RIA</given>
                                    <family>Smith</family>
                                </value>
                                <semanticsText>Person.Name</semanticsText>
                            </Person.Name>
                        </queryEvent>
                    </getPatientDetails-v1-0>
                </itk:payload>
            </itk:payloads>
        </itk:DistributionEnvelope>
    </soap:Body>
</soap:Envelope>
```

Ensure you update the Created and Expires times highlighted above, or the message will return an error instead of the correct data.

To send the command, you need to make a HTTP POST command to https://SERVER.cfh.nhs.uk/smsp/pds. The IP address of this will be available on the day. It requires the following headers:

```
SOAPAction: SOAPAction: urn:nhs-itk:services:201005:getNHSNumber-v1-0
```

```
content-type: text/xml
```

Add the client cert and key to your keystore. Depending on the programming language (e.g. Java) you may have to add the the ca.cert in a separate keystore.

The following curl command (from Linux) will send the file through and receive the response:

```
curl -XPOST -H "SOAPAction: urn:nhs-itk:services:201005:getNHSNumber-v1-0" -H
"content-type: text/xml" --cert ehitestClient.crt --key ehitestClient.key --cacert
ca.crt -d @smspgetnhsnumber.txt https://SERVER.cfh.nhs.uk/smsp/pds
```

More information is available here:

http://developer.nhs.uk/learn/demographics-itk-spine-mini-services-developers-guide

SMSP will also be available via the Internet Gateway.

EPS Tracker API

The prescription tracker provides a REST interface with two queries – prescription search and prescription retrieve - allowing querying to take place utilising a simple syntax. Both queries utilise a GET request which is made to the service and which returns a json result.

Search

To search for a list of prescriptions the external system will make an HTTP request which should include, as a minimum, the following parameters:

- NHS Number
- Format (this is a fixed value of ‘trace-summary’. Introduced for forwards compatibility)

In addition, the external system may also provide the following optional parameters:-

- Prescription date range (earliest and/or latest)[3]
- Prescription status
- Prescription Version
- Message version[4]

Thus the structure of the query will be:

[3] If earliestDate is not supplied it will default to 28 days ago. Note that this means that if no earliest date is supplied and the latest date is more than 28 days ago the earliest date will be after the latest date and no prescriptions will be found.

If latestDate is supplied it will include all of that date i.e. up until midnight of that date
If latestDate is not supplied it will default to the current date/time
If both earliestDate and latestDate are supplied and latestDate is earlier than earliestDate no prescriptions will be found.

[4] message version indicates the version of the message response and is not used as part of the search
N.B. The additional “version” parameter which is not required in this release.

```
https://mm-sync.national.ncrs.nhs.uk/mm/prescriptions?nhsNumber = {NHS
Number}&format= trace-summary&earliestDate={yyyymmdd}&latestDate
={yyyymmdd}&prescriptionStatus={prescription status}&
prescriptionVersion={prescription version}&version={version number}
```

The following mandatory HTTP headers are also required for authorisation and auditing purposes:

```
Accept: application/json
Spine-From-Asid: {senders AS ID}
```

The following optional HTTP headers can be added and will be logged:

```
Eps-TraceId              (optional)
Spine-UserId             (optional)
Spine-RoleProfileId      (optional)
```

Retrieve

To retrieve a particular prescription the external system will make an HTTP request which should include, as a minimum, the following parameters:

- Prescription Id (this is part of the URL in RESTful manner)
- Format (this is a fixed value of ‘trace’. Introduced for forwards compatibility)
- issueNumber (optional)
- Message version[5] (optional)

Thus the structure of the query will be:

```
https://mm-
sync.national.ncrs.nhs.uk/mm/prescriptions/{prescriptionId}?format=trace&version={
version number}&issueNumber{issue number}
```

With HTTP headers:

```
Accept: application/json
Spine-From-Asid: {senders AS ID}
```

The following optional HTTP headers can be added and will be logged:

```
Eps-TraceId              (optional)
Spine-UserId             (optional)
Spine-RoleProfileId      (optional)
```

[5] message version indicates the version of the message response and is not used as part of the search
N.B. The additional “version” parameter which is not required in this release.

The json response will follow a simple four part structure. Note that these sections are not ordered so may appear in any order:

- reason
- version
- statusCode
- prescription or prescriptionList

Only authorised external systems will have access to this service. This will be enforced by checking that :

a) the ASID of the sender matches that of the certificate
b) the ASID of the sender is included in the list of allowed ASIDs for this interaction

Refer to the EPS Prescription Tracker implementation guide for more detail.

Forward messaging

MESH

Message Exchange for Social care and Health (MESH) is an overall national solution for asynchronous messaging communications built on a 'store and poll' pattern. It builds upon the old DTS service. It maintains all the benefits of DTS but also introduces better performance, higher scalability (to support ever greater numbers of messages) and multiple comms channels.

Organisations interact with MESH by uploading files for other recipients to the recipient's virtual inbox and by polling their own virtual inbox for messages intended for them. Private suppliers typically support a MESH endpoint, integrating their systems with it on behalf of their NHS customer organisations. But NHS Organisations are equally able to apply for MESH endpoints directly.

Multi-channel MESH

Organisations can interact with the MESH by using any one of:

1) A free MESH downloadable client, available in Java or Perl. This is most suitable for previous DTS endpoints and endpoints of medium technical capability.
2) Direct interaction with the MESH API. This is most suitable for endpoints that want greater control over the pushing and the polling of their data;
3) A MESH GUI. This is an online application for accessing and uploading MESH messages. Users log onto the MESH GUI using two factor authentication either with an NHS smart card or alternative authentication mechanism. Once logged in they can see the messages intended for their mailbox and upload and address messages to other endpoints.

Installation of MESH Client

This is useful from a testing standpoint. You can setup a test client to check if your messages sent from your application via the API are working.

The MESH Client will be provided electronically.

The installer is a GUI installer.

Installation path, in Linux enter

```
/home/<username>/MESH-APP-HOME
```

Or in Windows

```
C:\MESH-APP-HOME
```

For ‘Legacy DTS Client’ Select ‘No’ unless you know you have the DTS client installed.

For Data files, in Linux enter:

```
/home/<username>/MESH-DATA-HOME
```

Or for windows:

```
C:\MESH-DATA-HOME
```

Finally, ‘Allow Automatic updates’ can be set as you prefer.

In your MESH-APP-HOME directory, edit the meshclient.cfg file to add the keystore password (Spine2 for the example given).

Configure your meshclient.cfg for your assigned mailbox (likely one of CONNECTATHONXXX).

```
<?xml version="1.0"?>
<MESHConfig>
    <PrimaryURL>https://mesh-sync.national.ncrs.nhs.uk</PrimaryURL>
    <LogPath>/home/<username>/MESH-APP-HOME/log</LogPath>
    <KeyStorePath>/home/<username>/MESH-APP-
HOME/KEYSTORE/mesh.keystore</KeyStorePath>
    <KeyStorePassword>Spine2</KeyStorePassword>
    <WorkPath>/home/<username>/MESH-APP-HOME/working</WorkPath>
    <ProxyPath>/home/<username>/MESH-APP-HOME/proxy</ProxyPath>
    <SignalPath>/home/<username>/MESH-APP-HOME/sig</SignalPath>
    <PollPeriod>30</PollPeriod>
    <ServerRetry>3</ServerRetry>
    <MaxMessages>100</MaxMessages>
    <MaxConnections>4</MaxConnections>
    <AutoUpgrade>Y</AutoUpgrade>
    <Client>
        <ClientIdentity>CONNECTATHONXXX</ClientIdentity>
        <ClientAuthentication>connectathon</ClientAuthentication>
        <InterfaceRoot>/home/spineii-user/MESH-DATA-
HOME/CONNECTATHON001</InterfaceRoot>
        <CollectReport>Y</CollectReport>
        <TransferReport>N</TransferReport>
        <PollReport>N</PollReport>
        <SaveSent>Y</SaveSent>
    </Client>
</MESHConfig>
```

Ensure you have mesh-sync.national.ncrs.nhs.uk set to the Spine IP address in your hosts (Linux) or lmhosts (Windows) file.

Finally, to run in Linux:

```
runMeshClient.sh
```

In windows, the Mesh client can be installed as a service and runs in the background. You will have to restart the service after changing the config.

Use of the MESH Client

Using the client is simple. Once installed as above, create a control file as shown:

```
<DTSControl>
  <Version>1.0</Version>
  <AddressType>DTS</AddressType>
  <MessageType>Data</MessageType>
  <From_DTS>CONNECTATHON001</From_DTS>
  <To_DTS>CONNECTATHON003</To_DTS>
  <Subject>Heart rate data</Subject>
  <LocalId>CONNECTATHON001</LocalId>
  <Compress>N</Compress>
  <Encrypted>N</Encrypted>
  <WorkflowId>MEDRPT_V3</WorkflowId>
</DTSControl>
```

Enter your mailbox id in ‘From_DTS’ and ‘LocalId’, and the target mailbox in ‘To_DTS’. Change the subject line to one relevant to you and your recipient.

You should name the control file and the data file with the same name, but a different extension (.ctl for the control file, .dat for the data file).

The content of your data file can be anything, eg some heart rate data:

```
20160416,91
20160417,91
20160418,101
20160419,67
20160420,91
20160421,78
```

To send, simply copy both the payload and control file into the OUT directory (payload first, then control file).

Any messages to be downloaded will be polled while the application is running at the rate shown in the config file, and will be available in the IN directory.

MESH API

The MESH API is available here: http://docs.meshapi.apiary.io/