Endeavour Health

Endeavour Health Information Exchange

Leeds CF Project - Hosting Requirements

Background to the Leeds Project

The overall aim of the Leeds Patient Orientated IT Benefit Evaluation Project is to provide a communication link between the clinician providing care and to digitise the workflow of medication delivered to the patients' home via the Homecare delivery service. This will enable an automated data flow from the CF pharmacist, hospital pharmacist, to the patient, with data flows back to the pharmacist.

This project is funded and supported by Endeavour Health Charitable Trust, working in partnership with Leeds Teaching Hospital Trust and Leeds University. Dr Daniel Peckham is project lead and owner responsible for informing the design, functional requirements and project delivery. Endeavour is responsible for providing development resource to support the successful delivery of the project.

The initial scope of the project is intended as a pilot, using pre-selected patients currently receiving treatment from within the Adult Cystic Fibrosis (CF) unit at Leeds. The patient recruitment process will be managed collaboratively between the Trust and University.

Initial scope will be limited to a simple digital workflow between the hospital pharmacists, and the patient. It is expected that future phases will see the inclusion of the Homecare delivery suppliers integrating with the JAC prescribing management solution in use at the Trust, as well as a link to the GP records within primary care.

Technical Scope

Initial scope of this project from the perspective of the Endeavour Data Service is as follows:

1. The integration of the Care Management System through the use of the EMIS Web Patient Access API. Changes made within the EMIS Web system should be made available to the Data Service as close to real time as the implemented solution permits. It is envisaged that an initial bulking phase will be required to seed the remote repositories.

2. Data entered in either the patient or pharmacy facing systems that is deemed significant to the ongoing care of the patient will be transferred and written back into the EMIS Web clinical system via the Patient Access API in the form of patient notes.

3. Initial population and ongoing maintenance of a central data repository within the Data Service.

4. The management of data publications and subscriptions in the form of Data

Distribution Protocols.

5. The integration of VitruCare clinical system including bi-directional message flows.

6. The integration of the Prescribing Workflow Application including bi-directional message flows.

7. The initial population and ongoing maintenance of a data repository within the Leeds Teaching Hospitals Data Analytics Service.

High Level Design

The Endeavour Data Service implementation routes healthcare messages securely and reliably from the data publisher to the appropriate subscribers.

Publisher and Subscriber contracts are configured using Data Distribution Protocols which underpin the patient consent and data sharing rules within the Data Service.

The Data Service is also responsible for message validation, message transformation and acknowledgement management.

Data received by the Data Service will be persisted in a data repository in internal format.

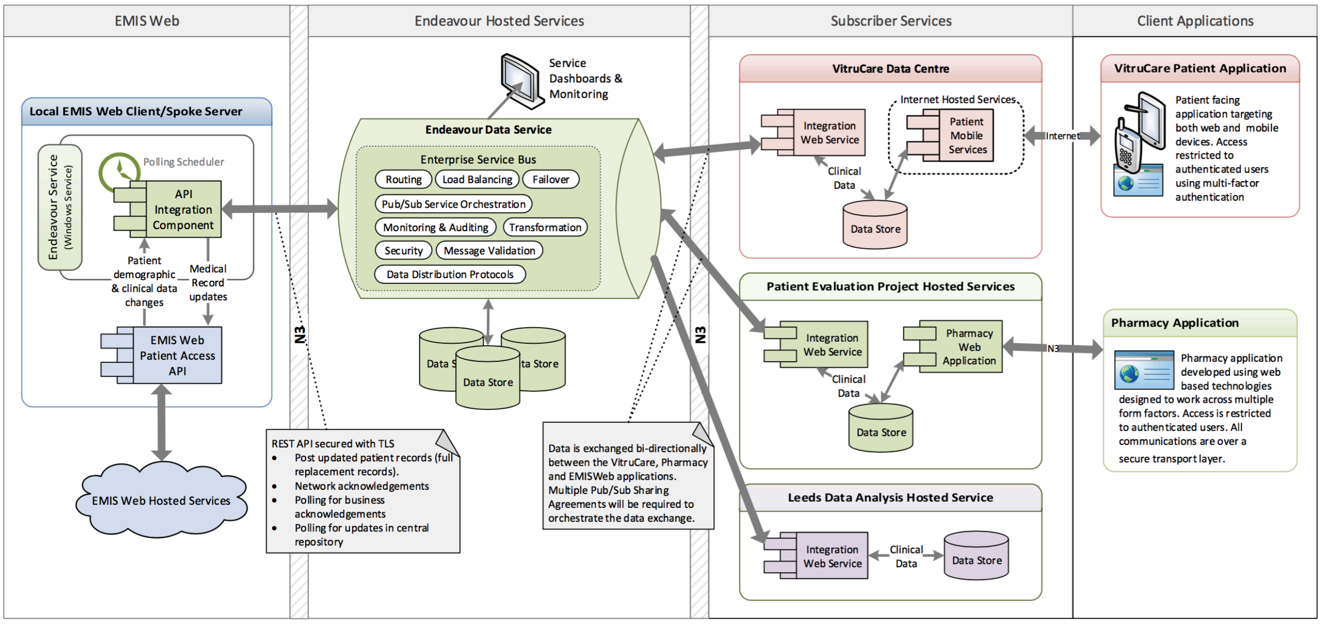
The service has been designed with scalability and resilience at its core, eliminating any single points of failure and encompassing automatic failover. It is built using the following open source technologies:

- Java – Core language
- Apache Cassandra – Database
- RabbitMQ – Message Broker
- Apache Camel – Routing Framework

A comprehensive service dashboard is available to configure and monitor the health status of the Data Service.

All messages are sent to the Data Service over a secure transport layer on the N3 network.

The following diagram shows a high-level project architectural overview. The Endeavour Hosted Services section is the focus of our Hosting Requirements.



REST API secured with TLS

- Post updated patient records (full replacement records).
- Network acknowledgements
- Polling for business acknowledgements
- Polling for updates in central repository

Data is exchanged bi-directionally between the VitruCare, Pharmacy and EMISWeb applications. Multiple Pub/Sub Sharing Agreements will be required to orchestrate the data exchange.

Patient facing application targeting both web and mobile devices. Access restricted to authenticated users using multi-factor authentication

Pharmacy application developed using web based technologies designed to work across multiple form factors. Access is restricted to authenticated users. All communications are over a secure transport layer.

High Level Overview of Services

Over and above the current operational care management systems, three additional things are needed:

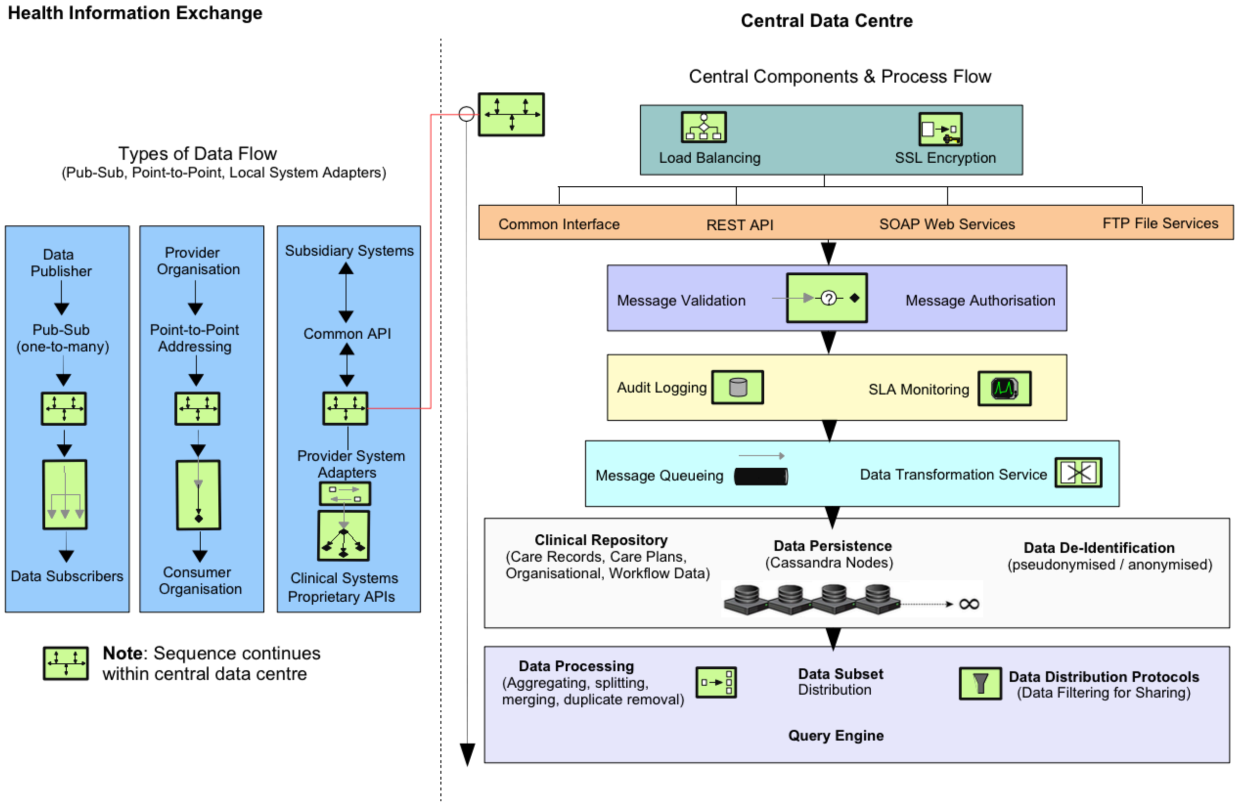
1. An N3 compliant hosting environment to operate the information exchange service
2. The data service (or information exchange) itself, to join the systems together
3. Additional functionality, additional modules and/or additional applications embedded within the care management systems themselves for the user to be able to take advantage of the new information flows

Single Platform, Multi-Vendor Components

In order to be able to make sense across a patch there does need to be a single software and hardware platform at the core. Ideally this platform will have three characteristics:

1. Hosted using a cloud computing model (or as close to that as the NHS allows) as the small number of interactions initially will rise exponentially.
2. Select a platform that can operate independently of the components i.e. select a platform that is not only care system independent but is also independent of the content which passes through it.
3. Select a platform that supports the emerging standards (such as REST and FHIR messaging) out of the box but enables legacy standards to co-exist.

The following diagram illustrates a logical separation of the main hosted service components:


Health Information Exchange
Central Data Centre
Types of Data Flow
(Pub-Sub, Point-to-Point, Local System Adapters)
Data Publisher
Pub-Sub (one-to-many)
Data Subscribers
Provider Organisation
Point-to-Point Addressing
Consumer Organisation
Subsidiary Systems
Common API
Provider System Adapters
Clinical Systems Proprietary APIs
Note: Sequence continues within central data centre
Central Components & Process Flow
Load Balancing
SSL Encryption
Common Interface
REST API
SOAP Web Services
FTP File Services
Message Validation
Message Authorisation
Audit Logging
SLA Monitoring
Message Queueing
Data Transformation Service
Clinical Repository
(Care Records, Care Plans, Organisational, Workflow Data)
Data Persistence
(Cassandra Nodes)
∞
Data De-Identification
(pseudonymised / anonymised)
Data Processing
(Aggregating, splitting, merging, duplicate removal)
Data Subset Distribution
Data Distribution Protocols
(Data Filtering for Sharing)
Query Engine

Technology Stack (for this project)

Development Tools

- Java8
- Javascript/JQuery
- Maven (build automation tool)
- Spring 4 (Core support for dependency injection, transaction management, web applications, data access, messaging, testing)
- IntelliJ (Development IDE)
- GitHub (web-based repository for revision control and source code management)

Database

Apache Cassandra - version 2.2.2

Key Features of Cassandra:

- Elastic scalability
- Always on architecture
- Fast linear-scale performance
- Flexible data storage
- Easy data distribution
- Operational simplicity
- Transaction support

Server Technology

- Apache Tomcat 8 (open-source web server)
- Apache Camel 2.15 (rule-based routing and mediation engine that provides a Java object-based implementation of the Enterprise Integration Patterns using an API)
- Java SE 8
- Linux Ubuntu 12.04 or above

Messaging Queuing

RabbitMQ – version 3.5.6

Key Features of RabbitMQ:

- Reliability
- Flexible Routing
- Clustering
- High availability queues

- Multi-protocol
- Management UI
- Tracing

APIs and Protocols

- REST
- SOAP
- SFTP

Data Standards and Transforms

- IHTSDO SMOMED CT
- FHIR
- ITK
- XML
- JSON
- EMISOPEN

Server Specification Requirements

The "scale-out" architecture of the data service is best suited to the use of "commodity spec" servers, using inexpensive, homogeneous servers that can be easily replaced, with software that can handle losing a few servers at a time.

Preference is to use Virtual Machines running Linux Ubuntu 12.04 or above.

We require two specifications of server:

1. For running our Cassandra database cluster.
2. For running our Java based component implementations (APIs, routes, business logic, message transformations etc.), which are orchestrated under the Apache Camel and Tomcat services, together with our RabbitMQ queueing technology.

It is assumed that the hosting environment will support a multiple datacenter fail-over replicated facility.

Server specification 1 (each Cassandra node)

We will use a Cassandra replication factor of 3 meaning at a minimum 3 nodes will be required.

Per Node	Requirement	Notes
Operating System:	Linux Ubuntu 12.04 or above	64-bit kernel. Either XFS or ext4.
Disk Storage:	1 Tb per node	SSDs are recommended for

		Cassandra. The NAND Flash chips that power SSDs provide extremely low-latency response times for random reads while supplying ample sequential write performance for compaction operations. Cassandra does not need RAID.
Memory:	16Gb per node	More RAM allows for larger cache sizes and reduces disk I/O for reads.
CPU:	4-core CPU processors (if virtual environment) For dedicated hardware, 8-core CPU processors	Cassandra is highly concurrent and uses as many CPU cores as available, For virtual environments, consider using CPU bursting, such as Rackspace Cloud Servers.
Network:	Recommended bandwidth is 1000 Mbit/s (gigabit) or greater.	Since Cassandra is a distributed data store, it puts load on the network to handle read/write requests and replication of data across nodes. Interfaces should be bound to separate Network Interface Cards (NIC). Cassandra efficiently routes requests to replicas that are geographically closest to the coordinator node and chooses a replica in the same rack if possible; it always chooses replicas located in the same data center over replicas in a remote data center.
Firewall	Generally, when you have firewalls between machines, it is difficult to run JMX across a network and maintain security. This is because JMX connects on port 7199, handshakes, and then uses any port within the	If using a firewall, make sure that nodes within a cluster can reach each other.

	1024+ range.	

Server specification 2 (each Tomcat node)

We will need a minimum of 3 nodes.

Per Node	Requirement	Notes
Operating System:	Linux Ubuntu 12.04 or above	64-bit kernel. Either XFS or ext4.
Disk Storage:	500 Mb per node	SSDs are recommended for RabbitMQ.
Memory:	8Gb per node	More RAM allows for larger cache sizes and reduces disk I/O for reads.
CPU:	4-core CPU processors	For virtual environments, consider using CPU bursting, such as Rackspace Cloud Servers.
Network:	Recommended bandwidth is 1000 Mbit/s (gigabit) or greater.	
Firewall	Standard SSL port Standard SSH port For RabbitMQ 5672	

Support Requirements

Endeavour contracted authorized support and development staff will need access to both the live and demo environments in order to provide administration, deployment, debugging and testing functions.

Remote Access for Administration

Remote access will be requirements to all server nodes and the components running therein. Standard private key SSH console based access will be sufficient.

Endeavour will produce administration and monitoring tools to assist the on-going support of the products.

Patching and Deployment

All component patching and deployment will be planned and scheduled, and wherever possible automated deployment scripts will be utilised to avoid human error.

The system will be configured in such a way that multiple versions of software will be supported to provide maximum disruption to the interoperability of components.

Testing and Scaling

Endeavour contractors will require access to both live production and test environment versions of the above platforms.