dhaca Interoperability Profile

dhaca Reference Architecture

Profile document

v1.0

18 June 2014

© Digital Health and Care Alliance 2014

Document Details

Subject	Reference Architecture
Category	Profile document
Title	dhaca Reference Architecture
First Publication Date	18 June 2014
Last Revision date	18 June 2014
Revision	v1.0

Revision History

1.0	18 June 2014	First release

© 2014 Digital Health and Care Alliance

This document is made available on a royalty-free basis to members of the Alliance for their own use. Please obtain permission before distributing it further. The document was produced using the reasonable skills, knowledge and experience of the authors as they relate to the subject matter at the time of publication. As such, the Alliance accepts no liability to any member or third party for their reliance on any of its contents for any purposes whatsoever. It is subject to revision at any time, with any revised document being published on the Alliance's website, www.dhaca.org.uk.

Table of Contents

1 Introduction

1.2 Purpose

The purpose of this **dhaca** reference architecture profile is to describe a reference technical architecture for digital health and care ecosystems that meets both the stakeholder requirements that originated from the TSB's *dallas* programme (1) and provides the necessary technical infrastructure that facilitates the economic and business models described in the **dhaca** Economics and Business Modelling profile (2).

The challenge is not necessarily one requiring new technological advances, but rather one of deploying tools and methods that are already feasible or demonstrated in other scenarios into the digital health and care environment.

This document specifically addresses the high-level interoperability concepts, identifying the types of interactions and interfaces that will be required in virtually any digital health and care ecosystem.

1.3 Drivers for Interoperability

1.3.1 Low cost

To reach large numbers of people it is important to remove the barrier of price from potential purchasers, be they private individuals, insurance companies, charities or statutory health care bodies.

One of the ways to achieve lower cost is by deploying services on platforms and devices that already benefit from mass-market economies of scale, with the low costs already achieved in the wider market. A further way to do this is by deploying on devices that people may already have, thereby removing the barrier of hardware purchase from their decision-making process.

1.3.2 Compatibility

One of the barriers to adoption is incompatibility between devices and systems. Ecosystems must be based on open standards that allow the maximum interoperability between disparate devices and systems.

1.3.3 Acceptability

People who wish or need to access digital health and care services also have normal lives to lead, and they therefore have many other reasons to adopt technological products. It is today quite normal for people to use these products in public and in their own homes without a second thought. There are, therefore, significant benefits to digital health and care services being available on devices that the general population already use, and using business and use models with which they are already familiar and accepting of.

2 Abstract View of the Architecture

2.1 Top-level architecture

Figure 1 below shows the top-level abstract view of the architecture.

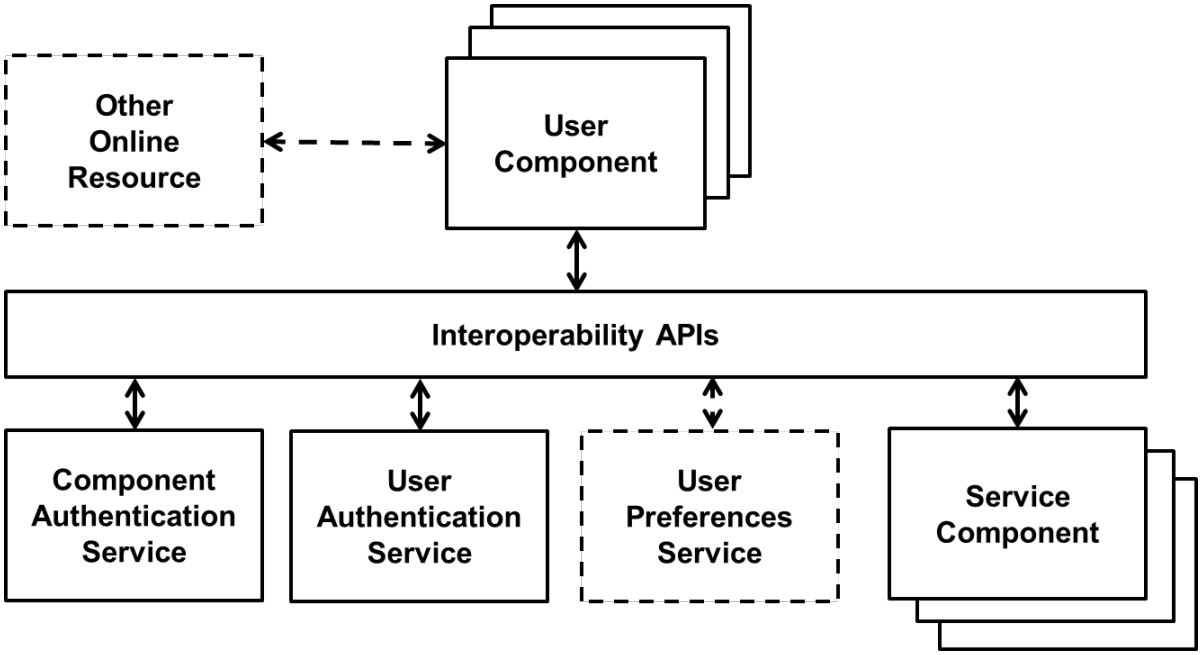

Figure 1: Abstract view of top-level architecture

This abstract view of the Reference Architecture contains two significant operating elements, the User Components and the Service Components.

User Components are intended to be the main user-operated components, typically client apps with graphical interfaces, keyboard, camera, voice or other actuators. They can be implemented on a variety of platforms including PCs, tablets and smartphones. They can operate online or offline, and could be connected to "external" resources.

Service Components do not contain user interfaces. These are the elements that typically reside remotely from the user, and will either be databases or servers in their own right, or gateways to external servers operated by statutory or commercial bodies.

The User and Service Components are connected together via the **Interoperability APIs**. These mediate between Components providing a common paradigm of

interconnectivity, which if conformed to, will ensure interoperability between all elements of the system.

Specialist services – the **User Authentication Service** and **Component Authentication Service** – authenticate users attempting to access the system, and Components (both User and Service) that connect to the system. A further possible special service, the **User Preference Service** stores information about individual users (such as their preferred form of address, other preferences, consents, etc.) and makes these available to all User and Service Components.

2.2 User Components

The User Component presents a user interface. In its generic form, as illustrated in Figure 2 below, it contains a User Interface Layer, and a Interoperability Layer Interface. It will also contain some internal functionality according to the nature of the component and the service being accessed.

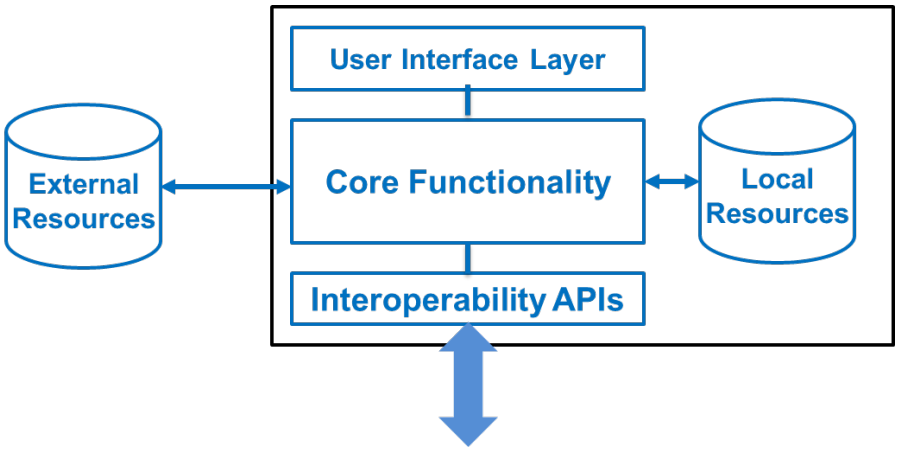

Figure 2: User Component general form

2.2.1 User Component as a native app

When implemented as a native app, the User Interface is implemented using native device UI API, or API provided by application development environment and may be well-tuned to the target device, by making best use of screen size and resolution, and input mechanisms.

2.2.2 User Component as a Web Application

The User Component may also be implemented as a Web Application, for example delivering using HTML5 to the device's web browser.

2.2.3 User Component as an interface to an Enterprise IT System

Figure 3 below shows a variant of the User Component which allows a user (for example. a health care professional) to access services from within their Enterprise IT system (e.g. an electronic patient records system or GP Practice system). This variant effectively acts as a communications gateway, with the Enterprise IT system being responsible for the end user interface.

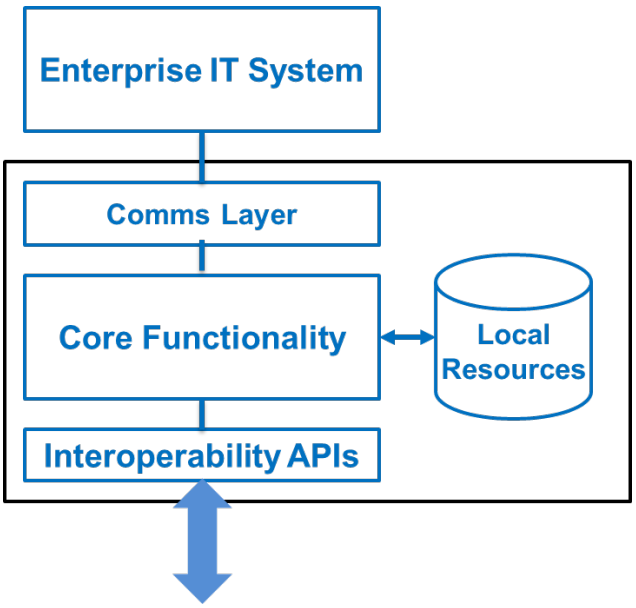

Figure 3: Enterprise IT system gateway User Component

2.2.4 Examples of User Components

Examples of systems that would be User Components in the logical architecture include:

- Personal life record apps;
- Telehealth monitoring apps;
- Long-term condition-specific support and advice apps;
- Medical records viewing apps;
- GP appointment-booking and repeat prescriptions apps;
- Informal care-givers' apps;
- Support group apps;

and also user interfaces/gateway apps to:

- GP systems;
- Hospital consultant systems;
- Social Services systems; and
- Voluntary-sector service provider systems.

2.3 Service Components

Service Components represent the server-end of the user's experience. As illustrated in Figure 4, below, they may either provide access to information, or transact a service using local resources, or they may act as a gateway to external services that perform these tasks.

The Service Component does not contain a User Interface itself, but is accessed through one or more User Components. In common with the User Component, there is an Interoperability APIs interface, core functionality and access to unspecified local resources.

To provide the Gateway functionality, there is a Communications Layer that communicates with external resources as required. This layer does not need to comply with any standardised APIs and may contain whatever code required to access the external resource.

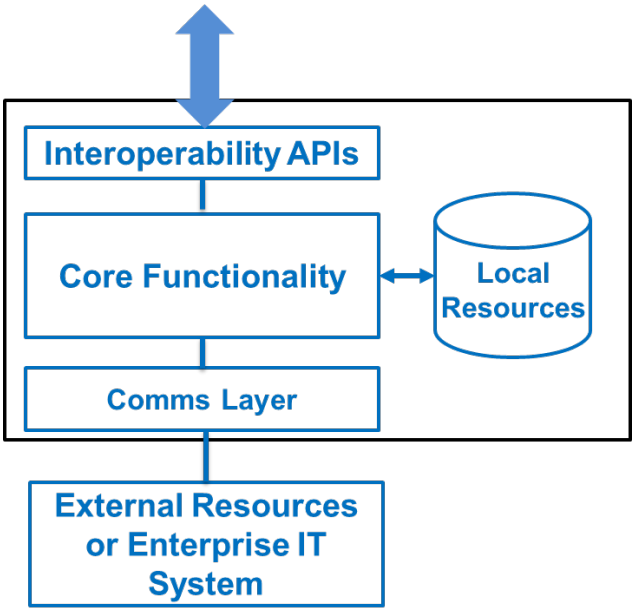

Figure 4: Service component general form

2.3.1 Stand-alone Service Component

An example stand-alone Service Component, in the form of a Personal Health Record system, is illustrated in Figure 5 below. In this case, all access to the information in the PHR is via the Interoperability APIs.

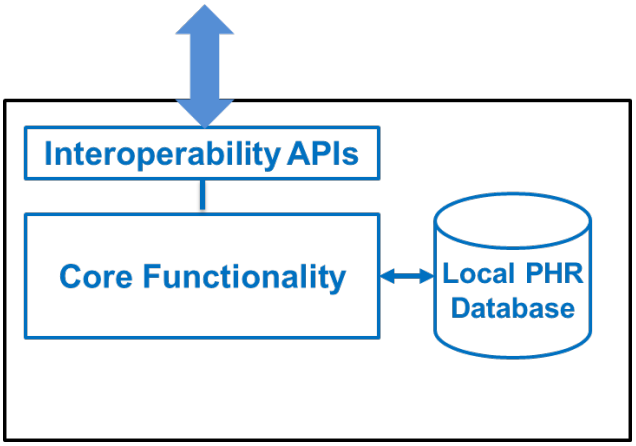

Figure 5: Stand-alone PHR Service Component

2.3.2 Gateway Service Component

An example of a Service Component acting as a gateway to external resources, in this case an external health record system, is shown in Figure 6 below. In this case data can written to and/or retrieved from the external system via the gateway Service Component (which is acting as a protocol convertor between the Interoperability APIs and the proprietary APIs of the external system), and also directly by other systems.

If the external system is a GP records system, for example, it can be updated directly by GPs in their surgery, but also made accessible to their patients via the interoperability APIs.

If the external system was a Personal Health Record, as well as allowing access via the Interoperability APIs, it might also allow other websites and "native" apps direct access, using its proprietary APIs.

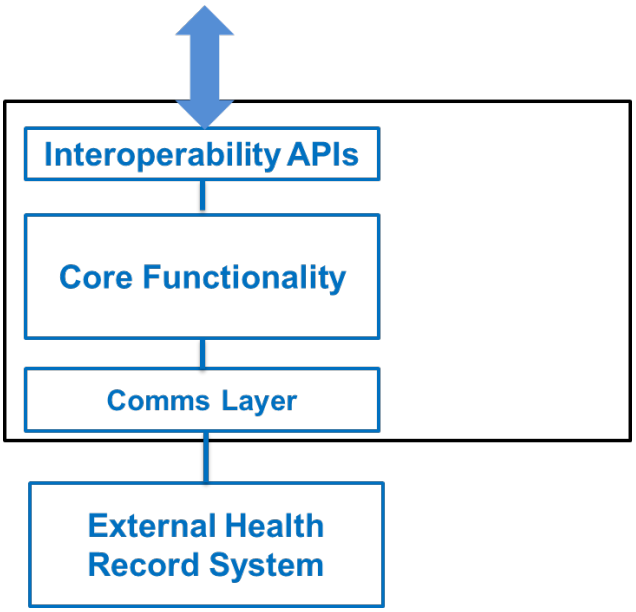

Figure 6: Service Component Gateway to Health Record System

2.3.3 Examples of Service Components

Examples of systems that would be Service Components in the logical architecture include:

- Shared calendars;
- Health insurance provider systems;
- Health information/education service;
- Peer-group support forums;
- Third-sector support services;

and gateways to:

- GP repeat prescriptions systems;
- Adult Social Services records systems;
- Telehealth and telecare systems;
- PHR systems;
- GP Records systems;
- Hospital electronic records systems;
- GP appointment-booking systems; and
- NHS eReferrals system.

2.4 Component Authentication Service

The Component Authentication Service, illustrated in Figure 7 below, provides the means by which the ecosystem operator ensures that only approved User Component applications can use the Interoperability APIs to access Service Components, and that only approved Service Components are permitted to join the ecosystem.

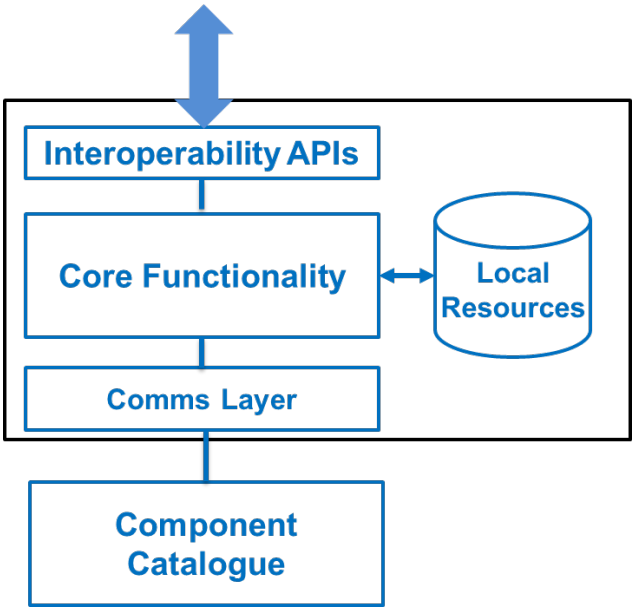

Figure 7: Component Authentication Service

Service and User Components use the interoperability APIs to register/authenticate with the Component Authentication Service, and to verify the authenticity of any other components with which they communicate.

The fact that the Component Authentication Service, of necessity, knows of the existence of every other component in the ecosystem makes it a logical for it to be the source of any catalogue of components made available to the end user. Provision of such a catalogue allows end users to confidently and securely select, purchase and/or install (as necessary) and use apps and services.

2.5 User Authentication Service

The User Authentication Service is the means by which User and Service Components authenticate the end-user. It provides a single sign-on facility for the user across the ecosystem, so as to prevent the user having to continually re-authenticate as the navigate apps and services within the ecosystem.

As shown in Figure 8 below, it is anticipated that the user will be given the option of creating an account with the ecosystem, or of using one of their existing digital identities within the ecosystem.

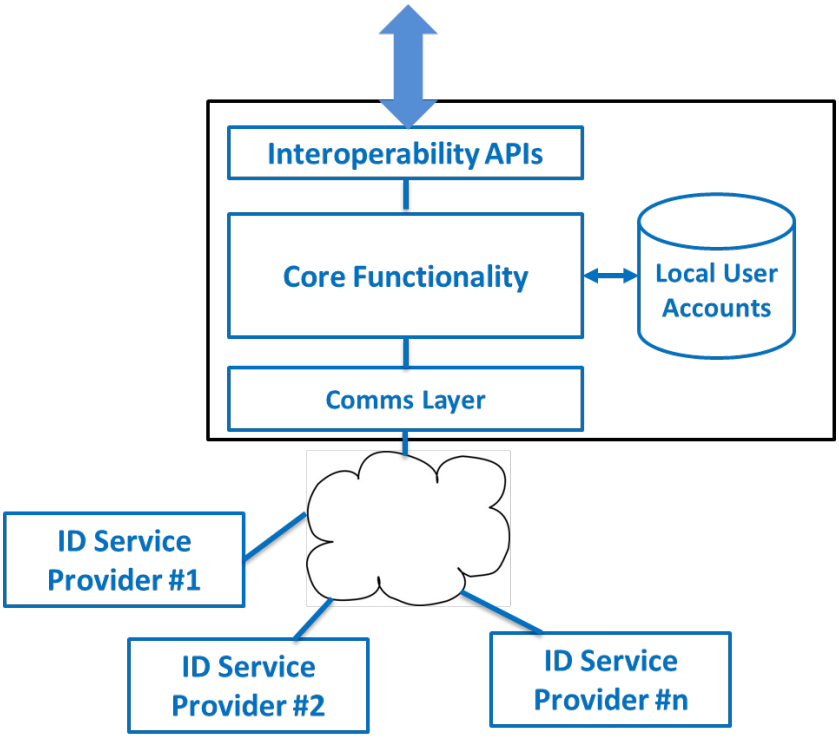

Figure 8: User Authentication Service

This external digital identity could be a social network identity (e.g. Facebook (3) or Google (4)), or it could be a formal government-issued identity, such as the Scottish Government's mygovscot myaccount (5), or from an identity service provider that is part of the UK Government's Identity Assurance Programme (6).

2.6 User Preferences Services

For similar reasons to providing a single sign-on facility for users within the ecosystem, an ecosystem operator is likely to want to offer its users the ability to store certain information about themselves and their preferences, securely and once only, with this information being made available to applications and services with the users' consent.

The User Preferences Service, therefore, stores information about individual users (such as their preferred form of address, other preferences, consents, etc.) and makes these available to all User and Service Components.

It is possible that having a separate component to handle this functionality may not be required; instead it may be possible for this information to be stored as "attributes" against the user's digital identity, and provided to apps and services (with the user's consent) along with the user's identity.

3 Reference Implementation

3.1 Overview

Figure 9 below shows a reference implementation of the architecture, built from generic versions of real-world system components. The main difference between this view of the architecture and the abstract view presented earlier is that components can be hybrid User and Service Components. For example, the GP Surgery System can both act as a User Component (for example, allowing GPs with the necessary consent from their patients to access the patients' PHR), and as a Service Component (for example, allowing patients access to their GP medical records, to order repeat prescriptions, and to book appointments).

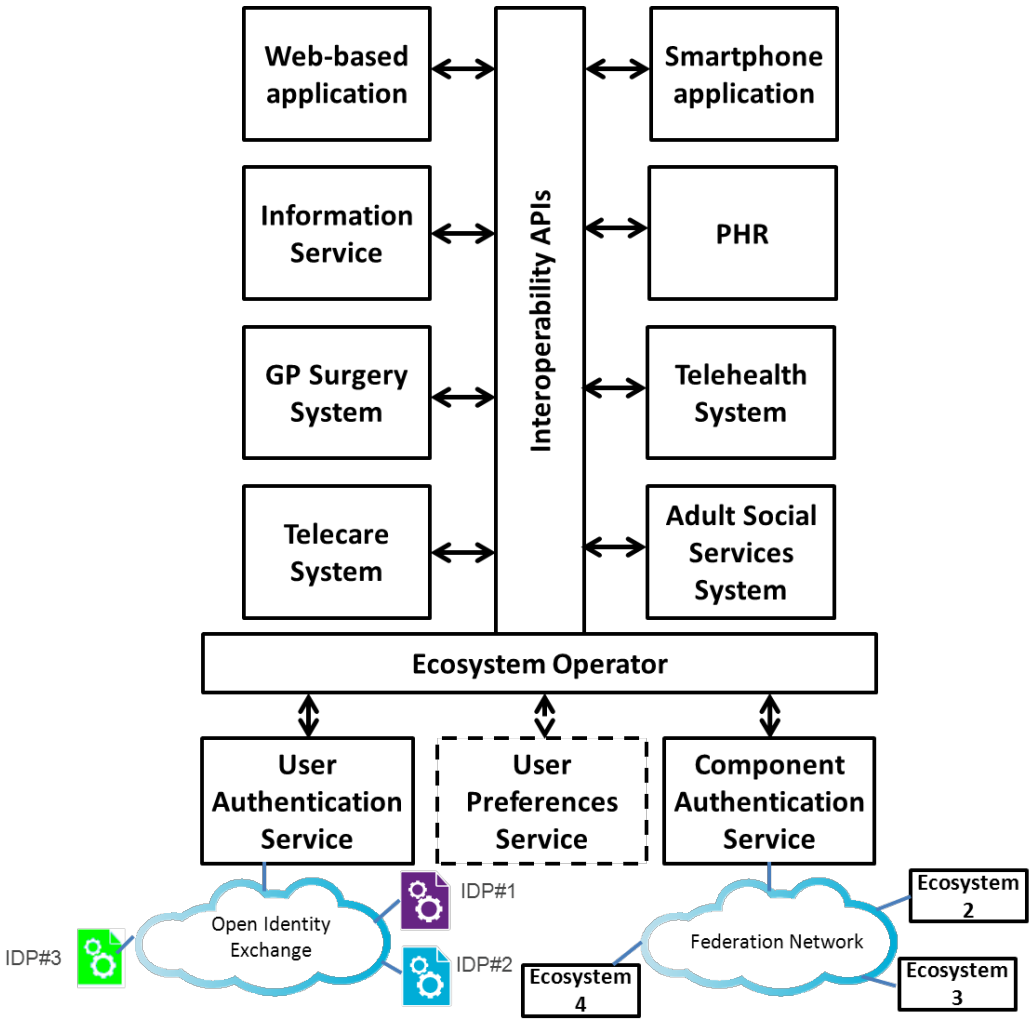

Figure 9: Reference Implementation of the architecture

Whilst the components use the Interoperability APIs to communicate securely amongst themselves, there is nothing within the reference architecture to stop any component accessing content, services or other applications from outside the ecosystem, or to stop components allowing access to their data and services from other applications using other proprietary APIs; the architecture is designed to be inclusive, rather than exclusive, in nature.

3.2 Interoperability APIs

It is not the intention of the reference architecture to define in great detail every API for every User and Service Component within the ecosystem; in most cases, especially at this early stage in the development of digital health and care ecosystems, it is better to allow the developers the freedom to specify their own APIs, within a common framework.

The common framework eases development by requiring that all APIs use the same standards and work in the same way. This reduces the burden of learning on developers and enables them to use a single code library to implement the lower layers of the APIs.

The common frame work consists of:

- a RESTful interface (7);
- with data serialised in JSON (8);
- carried over secure HTTP connections (9); and
- with access authenticated using OAuth2.0 (10) and identities verified using OpenID Connect 1.0 (11).

Based on this common framework, the developers of any Service Component commit to expose all significant functionalities through “open” APIs (APIs which are publicly documented, fit-for-purpose, and reusable – see the NHS England Open API policy (12) for an example), and the developers of any User Component commit to use those APIs to enable person-centric digital health and care applications. Popular APIs will be adopted, both by client developers and other server developers. As the market matures, it might be appropriate for some of these APIs to then be standardised.

Where there are existing standard APIs, developers should give proper consideration to their adoption. For example, health records systems should, where appropriate, provide

APIs that comply with the HL7 "Fast Healthcare Interoperability Resources" (FHIR) standard (13).

3.3 Open Identity Exchange

The reference architecture uses an Open Identity Exchange (14) compatible Trust Framework with a number of Identity Service Providers (IDPs) to allow users to use an existing digital identity within the ecosystem. The IDPs should include social network identity providers, such as Google (4) and Facebook (3), and also IDPs offering more robust identity services, such as those accredited as part of the UK Government's Identity Assurance programme (6).

The User Authentication Service acts as a hub, or proxy, which all components within the ecosystem use to authenticate users. The User Authentication Service then uses the OpenID Connect 1.0 standard (11), which in turn is built on the OAuth 2.0 standard (10), to interact with the IDPs.

3.4 Inter-Ecosystem Interoperability

3.4.1 Federation network

In Figure 9, the Component Authentication Service is shown as being part of a Federation Network with four other ecosystems. Such a federation network would be particularly useful if a number of regionally-based digital health and care ecosystems were created; it would allow, for example, national services, that were members of a national ecosystem, to be available to all users of the regional ecosystems, without having to separately join each regional ecosystem. Such a federation, of course, requires that a suitable trust framework is developed between the various ecosystem operators, so that all stakeholders can have confidence in the operation, security and privacy of the overall network.

3.4.2 Multi-ecosystem deployment

An alternative scenario is where a component developer wishes to deploy his/her product into two ecosystems that are not part of a federation network. An example of this is shown in Figure 10 below. The ecosystem from Figure 9 is shown on the right, with a web-based application and two services working both as part of this ecosystem and a parallel ecosystem on the left.

The information and telecare services expose the same APIs to both ecosystems, but they use and present different authentication credentials, depending which ecosystem they are operating with. In fact, there would be nothing to stop the services exposing different subsets of their API to the different ecosystems, if there was good reason for them to do this.

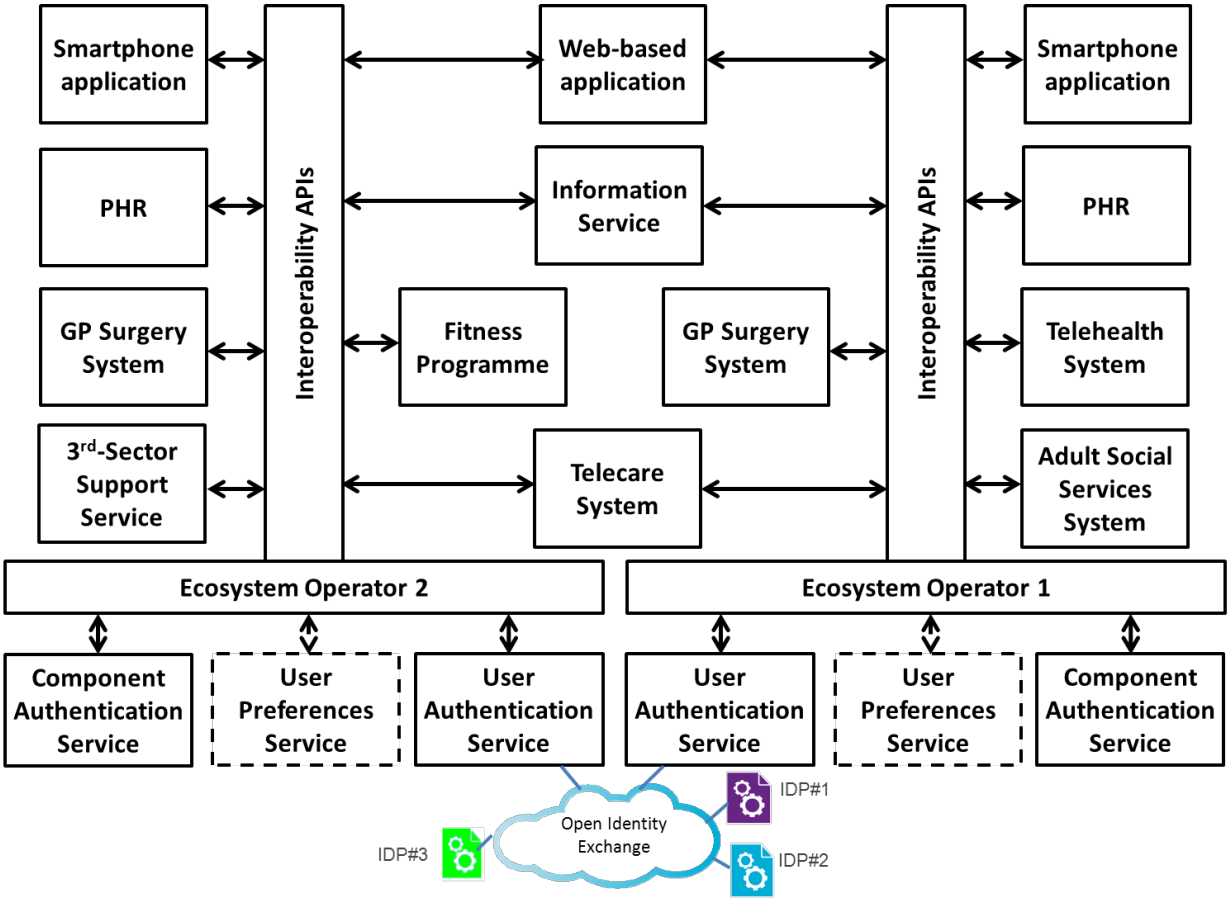

Figure 10: Parallel ecosystems sharing components

The web-based application's ability to work within both ecosystems is facilitated by the common API framework. Building on this framework, its developers use the APIs of the services available in either ecosystem. At some point during the configuration of the user's account, he/she decides which ecosystem they wish to work with, and the application uses the appropriate authentication credentials for that ecosystem.

Figure 10 also shows the two ecosystems using Open Identity Exchange-compliant Identity Service Providers (IDPs) to authenticate their users. The ecosystems may use the same or different IDPs, depending on their business models and requirements as to the level of trust they need in the authenticity of users' identity.

An alternative view of these parallel ecosystems is shown in Figure 11. This view attempts to highlight the importance of the common API framework to the inter-ecosystem interoperability and the development of the wider digital health and care market.

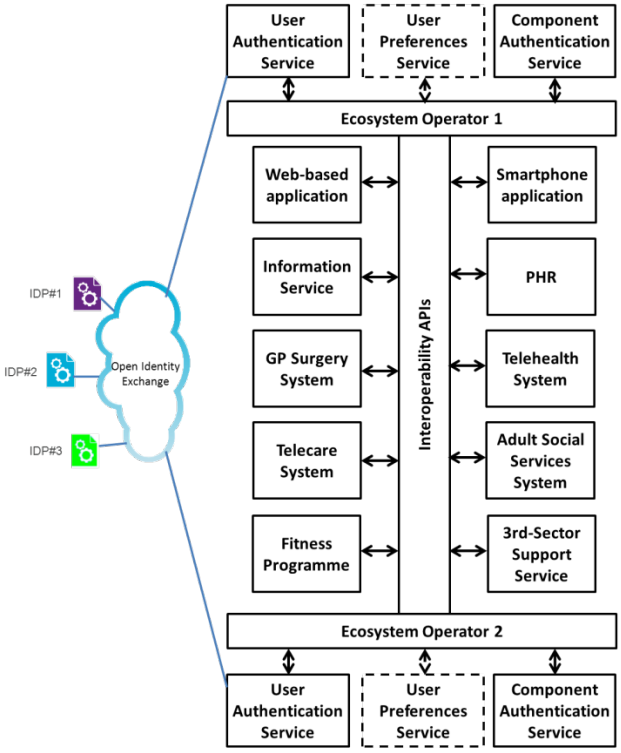

Figure 11: Alternative view of parallel ecosystems

Figure 12 below shows the extent of Ecosystem 1 within the wider environment.

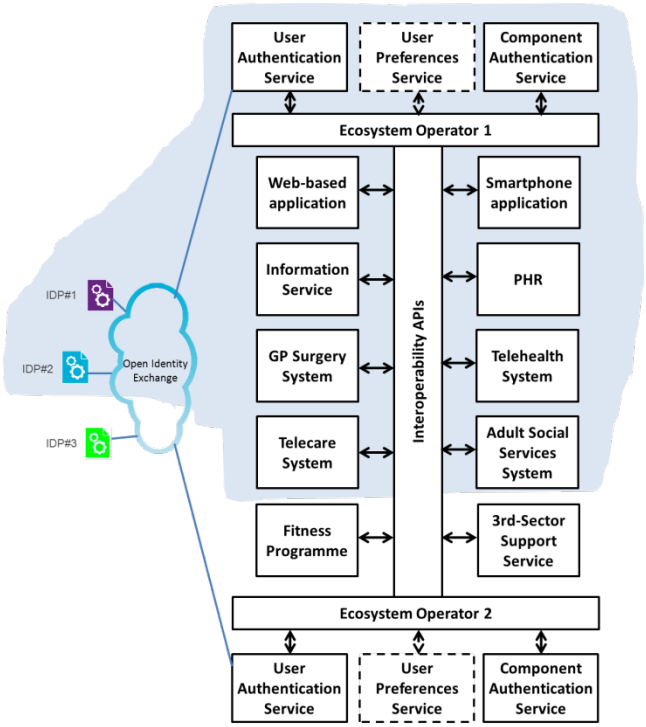

Figure 12: Ecosystem 1 within the wider environment

It encompasses the six services, two applications and two IDPs, highlighted in blue. Figure 13 below overlays the extent of Ecosystem 2, showing it sharing the web-based app, two services and one IDP with Ecosystem 1, and also encompassing two further services and an alternative IDP.

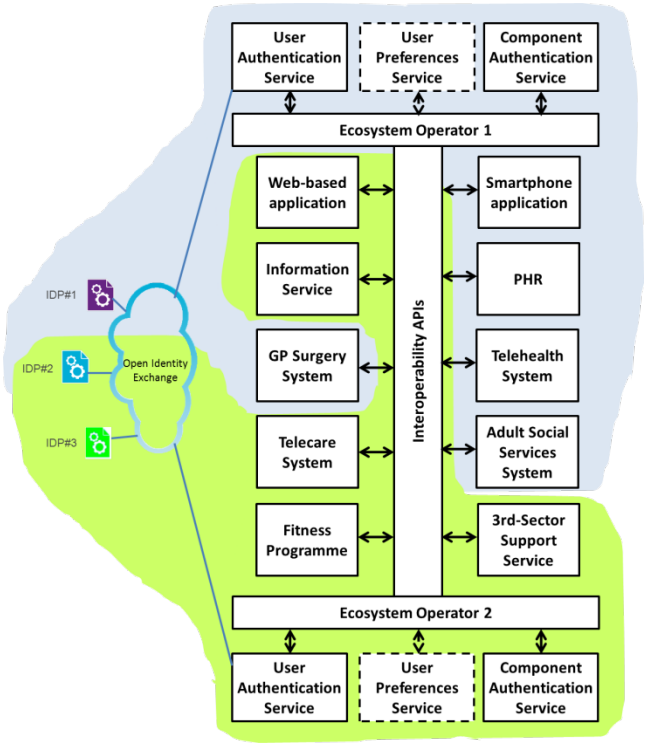

Figure 13: Ecosystem 2 within the wider environment

This shows that the common API framework set out in Section 3.2, together with the use of the use of Open Identity Exchange trust frameworks and protocols, as set out in Section 3.3 provide a firm technical framework upon which the business models described in the Economic and Business Modelling Profile can be deployed to develop a thriving, innovative digital health and care system.

4 Bibliography

1. **i-focus.** Stakeholder Requirements for Reference Architecture. [Online] 17 October 2013. [Cited: 18 June 2014.] http://ifocus-dallas.com/pub/wp-content/uploads/2013/03/D3.2.1-Stakeholder-Requirements-for-dallas-Reference-Architecture_v0.1.pdf.

2. **Digital Health and Care Alliance.** Interoperability Profile: Economics and Business Modelling. [Online] 18 June 2014. [Cited: 18 June 2014.] !!!!INSERT URL HERE!!!!.

3. **Facebook Inc.** Facebook Login. *Facebook Developers Site.* [Online] [Cited: 18 June 2014.] https://developers.facebook.com/docs/facebook-login/v2.0.

4. **Google Inc.** Google Accounts Authentication and Authorization. *Google Developers.* [Online] [Cited: 18 June 2014.] https://developers.google.com/accounts/.

5. **Scottish Government.** myaccount for Scotland. *mygovscot.* [Online] [Cited: 18 June 2014.] https://signin.mygovscot.org/home/.

6. **UK Government Cabinet Office.** Identity assurance: enabling trusted transactions. *GOV.UK.* [Online] [Cited: 18 June 2014.] https://www.gov.uk/government/publications/identity-assurance-enabling-trusted-transactions.

7. **Wikipedia.** Representational state transfer. [Online] [Cited: 18 June 2014.] http://en.wikipedia.org/wiki/Representational_state_transfer.

8. **Crockford, Douglas.** The application/json Media Type for JavaScript Object Notation (JSON). [Online] July 2006. [Cited: 18 June 2014.] http://tools.ietf.org/html/rfc4627.

9. **Rescorla, Eric.** HTTP Over TLS. [Online] May 2000. [Cited: 18 June 2014.] https://tools.ietf.org/html/rfc2818.

10. **Hardt, Dick (editor).** The OAuth 2.0 Authorization Framework. [Online] October 2012. [Cited: 18 June 2014.] http://tools.ietf.org/html/rfc6749.

11. **OpenID Foundation.** OpenID Specifications. [Online] [Cited: 18 June 2014.] http://openid.net/developers/specs/.

12. **NHS England.** Open API Architecture Policy. *NHS England.* [Online] May 2014. [Cited: 18 June 2014.] http://www.england.nhs.uk/wp-content/uploads/2014/05/open-api-policy.pdf.

13. **Health Level 7 Inc.** FHIR Specification Home Page. [Online] [Cited: 18 June 2014.] http://www.hl7.org/implement/standards/fhir/.

14. **Open Identity Exchange.** *OIX website.* [Online] [Cited: 18 June 2014.] http://openidentityexchange.org/.