

# E-enablement of the Common Assessment Framework

eCAF Interfaces View

Version 1.0

PHOTO REDACTED DUE TO THIRD PARTY RIGHTS OR OTHER LEGAL ISSUES



Every Child Matters  
Change For Children

## Document Control

### *Revision History*

Issue date	Version	Summary of Changes
08/06/2006	1.0	Initial publication

### *Purpose of this Document*

This document extends the eCAF Requirements Catalogue with additional information and guidance on interfacing requirements for an eCAF system.

## Contents

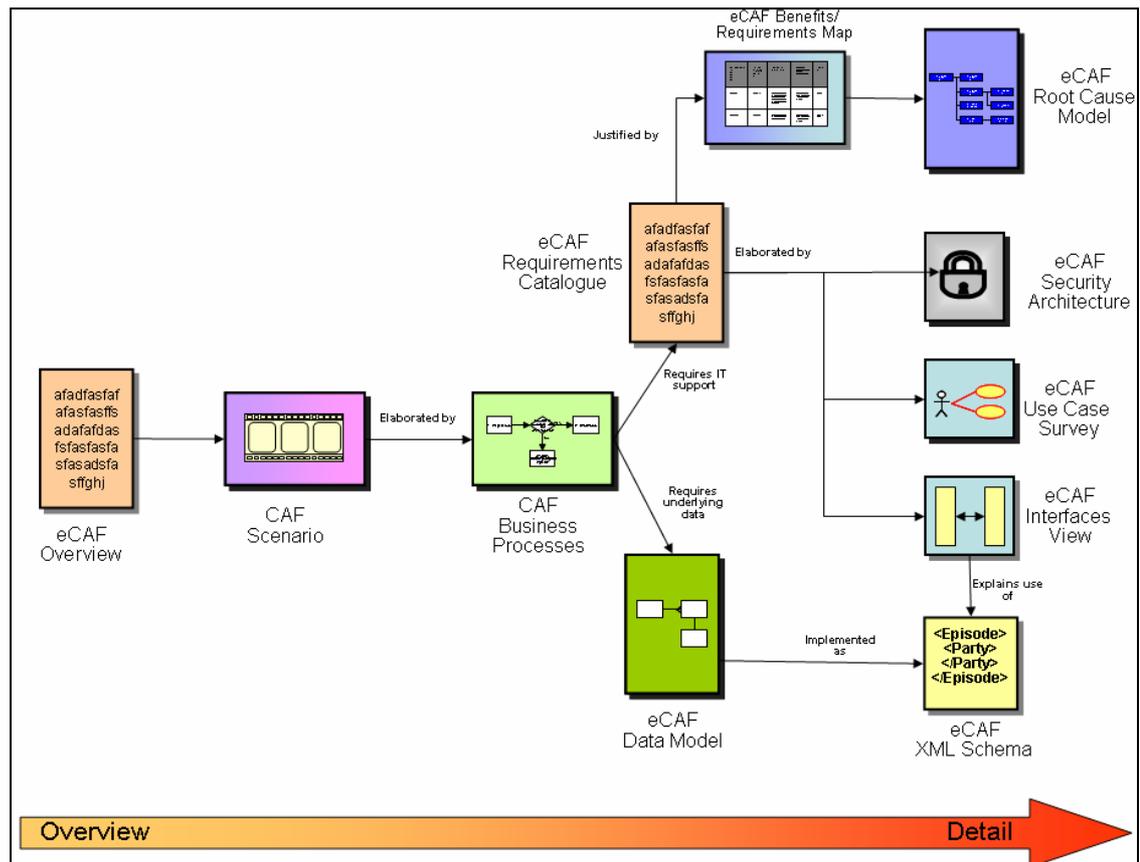
Document Control	2
Revision History	2
Purpose of this Document	2
1. eCAF Documentation Reader's Guide	5
1.1 Diagram	5
1.2 Description of documents	5
2. Introduction	7
2.1 Purpose	7
2.2 Document scope	8
3. Interfaces Overview	10
3.1 Systems overview	10
3.2 Interfaces overview	11
3.3 Interfacing mechanisms and evolution	12
4. File import/export	13
4.1 Introduction	13
4.2 File transfer scenarios	13
<b>4.2.1 Transfer CA Episodes to another Local Authority (1)</b>	13
<b>4.2.2 Download referral to Case Management System (2)</b>	14
<b>4.2.3 Upload/download from/to Offline Devices (3)</b>	15
4.3 Reference Specifications	15
5. Web Services	16
5.1 Introduction	16
5.2 Logical interface	17
<b>5.2.1 Web Service principles and patterns</b>	17
<b>5.2.2 Web Service functions</b>	17
<b>5.2.3 Fault codes</b>	19
<b>5.2.4 Interactions</b>	20
5.3 Physical interface	23
<b>5.3.1 Recommended standards</b>	23
<b>5.3.2 WS-Security recommendations</b>	24
<b>5.3.3 WS-Policy recommendations</b>	24
5.4 Reference Specifications	26
6. IS Index Integration	27
6.1 Introduction	27
6.2 Interactions	27
<b>6.2.1 The CAF Flag (1)</b>	28
<b>6.2.2 Update child details (2)</b>	28
<b>6.2.3 View child details (3)</b>	28
6.3 Reference Specifications	28
7. Appendix A - Infrastructure Considerations	29
7.1 Authentication / Single Sign On	29
<b>7.1.1 Relevance to eCAF</b>	29

<b>7.1.2</b>	<b>Current situation</b>	29
7.2	Messaging Hubs	30
<b>7.2.1</b>	<b>Relevance to eCAF</b>	30
<b>7.2.2</b>	<b>Current situation</b>	30
7.3	Service Registry	31
<b>7.3.1</b>	<b>Relevance to eCAF</b>	31
<b>7.3.2</b>	<b>Current situation</b>	31
7.4	Secure email	31
<b>7.4.1</b>	<b>Relevance to eCAF</b>	31
<b>7.4.2</b>	<b>Current situation</b>	31
8.	Appendix B – Web Service Patterns	32
8.1	Request – response pattern	32
8.2	Stateless services	32
8.3	No transactions	32
8.4	No reliable delivery requirement	32

## 1. eCAF Documentation Reader's Guide

### 1.1 Diagram

The diagram below shows the documents in the set, and each one is briefly described in the following text.



### 1.2 Description of documents

- **eCAF Overview** – Essential starting point and executive summary. Introduces the other documents in the set.
- **The CAF Scenario** – This document walks through a “story”, showing an example of how the CAF Business Processes might work in practice. Useful for all readers, to gain a basic familiarity with CAF process.
- **The CAF Business Processes** – This document describes the people and business activities that are required to complete a Common Assessment and the subsequent actions arising out of that Assessment. It also indicates where IT support from an eCAF system will assist these activities.
- **The Requirements Catalogue** – This document defines what system support is required by practitioners using the Common Assessment Framework (CAF). It contains categorised listings of functional and non-functional requirements.

- **The Security Architecture** – This document defines in more detail the security requirements for an eCAF system. This is a critical aspect, and thus worthy of specific consideration.
- **The Use Case Survey** – This document presents the requirements as Unified Modelling Language (UML) Use Case diagrams. This may be useful for more technical readers, for example to inform the Inception and Elaboration stages of a Rational Unified Process (RUP) development project.
- **The Interfaces View** – This document provides more information about the interfacing requirements for an eCAF system. Interfacing is important but potentially complex, so this document provides additional guidance.
- **The Data Model** – This document contains a high-level diagram of the information that will be required in the context of CAF. It provides a more detailed view of information requirements in the form of an Entity Relationship Diagram that defines the essential eCAF data items and their relationships. It also includes a set of Data Classifications which summarise the types of data used in CAF, such as Name and Contact Details. It provides standard names and definitions that will be used by an eCAF system.
- **The XML Schema** – This is a technical schema specification (plus example xml file), providing a standard representation of the Data Model as an XML (GovTalk) message. XML is a widely accepted data format used for information exchange between systems.
- **The Root Cause Model** – This document describes the root causes of the main issues which prevent the delivery of the targeted outcomes of the 'Every Child Matters: Change for Children' Programme (relevant to initial assessments). It states both the business challenges faced (the issues and their root causes) and the business need to be addressed.
- **The Benefits/Requirements Map** – This document provides the linkage between the root causes eCAF looks to address and the solution components (requirements) designed to address them.

## 2. Introduction

### 2.1 Purpose

This document extends the eCAF Requirements Catalogue with additional information about interfacing requirements for an eCAF system. It expands on the “Convert and Share” aspect of the functional requirements, and is intended for technical readers.

While the Requirements Catalogue provides a formal listing of requirement statements, its format makes it difficult to explain the underlying concepts for eCAF interfacing. This document helps by providing, in a less formal style, additional background information on the principles, concepts and practical details of interfacing.

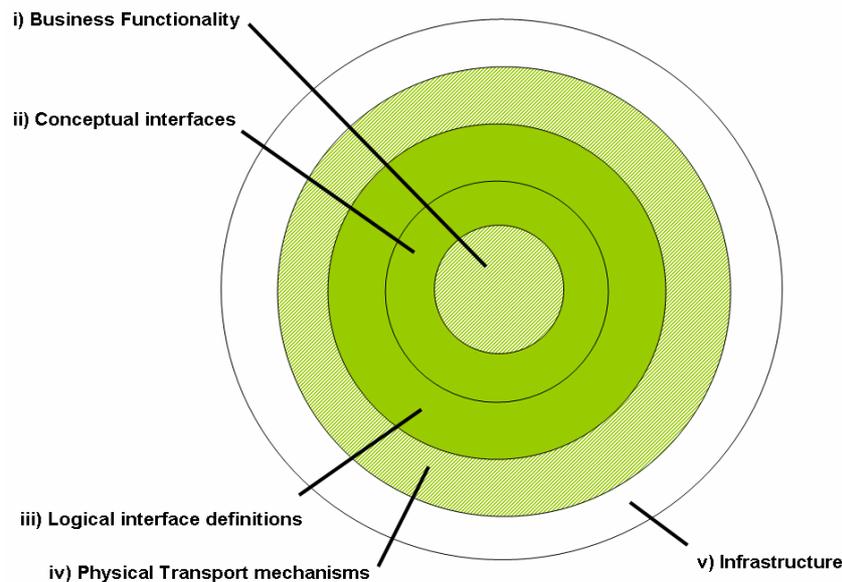
In addition it provides pointers, where appropriate, to detailed technical specifications, such as eCAF XML schema files.

It is worth noting that the interfaces differ from the rest of the eCAF requirements in that they deviate from the principle of defining “what not how”. Consistent interfaces are key to the cross-border interoperability of eCAF systems, and the intention is to define their implementation in as much detail as possible.

NB: eCAF is one of a number of Every Child Matters projects (including for example the IS Index), which are currently under development. This document provides provisional guidance based on best-information at the current time (June 2006). Both the overall picture of the “fit” between the systems, and the physical transport and infrastructure details will continue to evolve as these other projects progress.

## 2.2 Document scope

The diagram below illustrates the “layers” involved in constructing eCAF system interfaces, and helps to explain the scope of this document. The notes following the diagram considering each layer in turn:



### i) **Business Functionality (Partly covered)**

Business Functionality is covered in the *eCAF Requirements Catalogue* – please see that document for more details. However, for ease of reference, business functionality is mentioned here where it is relevant to operation of the interfaces. This applies especially, for example, to items such as logging and access control.

### ii) **Conceptual interfaces (In Scope)**

This document explains the interfaces of an eCAF system on a conceptual level. “Conceptual” means “what” – what systems are involved, in what circumstances they need to talk to each other, and what kind of messages are transferred. See **Section 3** for more about this.

### iii) **Logical interface definitions (In Scope)**

Logical interface definitions are the main focus of this document.

“Logical” means “how” – how the interfaces work and what messages are exchanged.

- o **Section 4** explains how the interfacing requirements can be met by using the eCAF XML Schema as the basis of a simple file import/export facility
- o **Section 5** is again based on the eCAF XML Schema, but this time describes the functions required for a Web Services Interface. It is intended as a more readable introduction to the formal eCAF WSDL definitions.

- **Section 6** explains the information that will be exchanged between eCAF and the IS Index.

(Note that although the eCAF XML schema is referenced in all of these sections, the contents of the CA Episode dataset are not discussed in detail. Please see the **eCAF Logical Data Model** for more about this).

**iv) Physical transport mechanisms (Discussed but Out of Scope)**

This document discusses possibilities for physical transport mechanisms when describing each interface. However it does not mandate specific solutions. It explains the qualities required for each interface and explores some likely options.

*Any decisions on physical transport mechanisms between systems should be reviewed for compatibility with other Every Child Matters projects (eg IS Index) as more details on these become available.*

**v) Infrastructure (Out of Scope)**

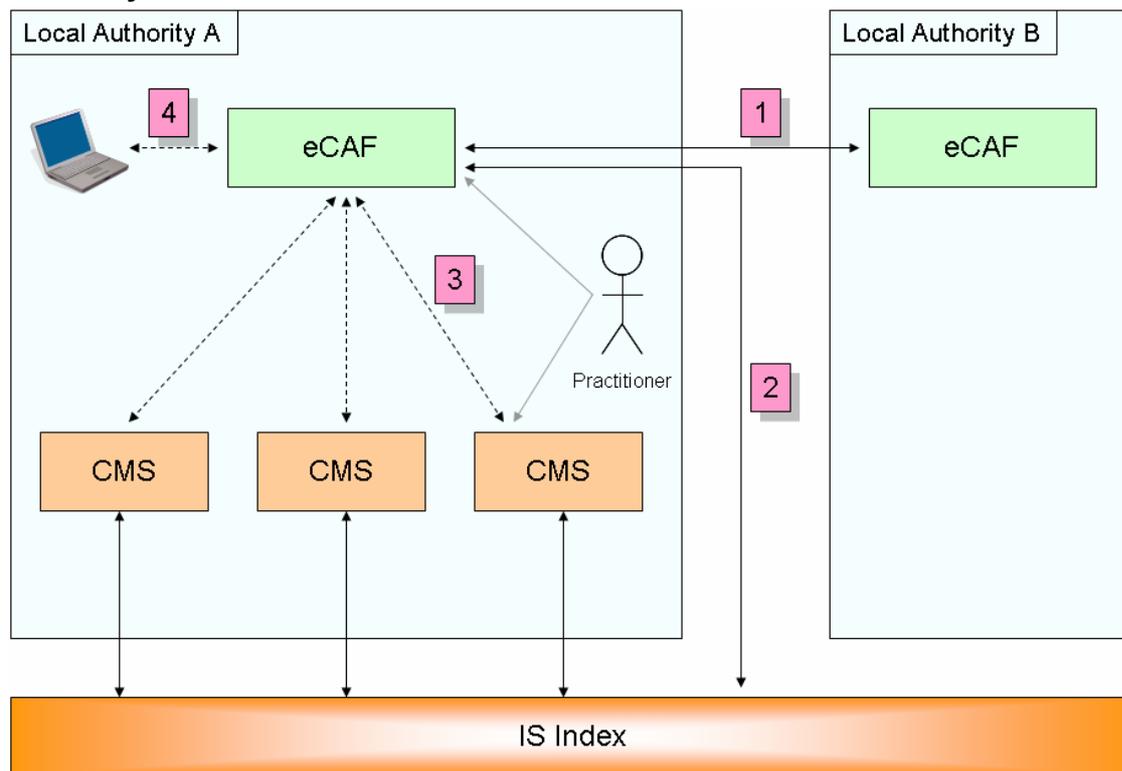
An eCAF system does not exist in isolation – it will be installed into an existing IT infrastructure. The surrounding infrastructure will have an important effect on its operation, especially when it comes to interfaces which communicate with the outside world. However eCAF systems will be used across a wide range of environments and, while there would be significant benefits from a consistent UK Public Sector infrastructure, it is beyond the scope of this document to specify this.

*Any decisions on system and security infrastructure should be reviewed for compatibility with other Every Child Matters projects (eg IS Index) as more details on these become available.*

### 3. Interfaces Overview

This section outlines at a high level the landscape of IT systems that interoperate with eCAF, and the nature of those interactions.

#### 3.1 Systems overview



The diagram above shows three major building blocks of IT support for Every Child Matters. Each of the three types of system fulfils an important and specific role in the overall picture:

- **eCAF**  
eCAF systems sit at Local Authority level and provide a forum for shared working. All CA Episode data is stored in the Local Authority-based eCAF system, where it can be accessed (subject to consent) by all practitioners. eCAF brings a thin layer of needs-based coordination on top of the detailed activities in Case Management Systems.
- **Case Management Systems**  
Case management Systems (CMS) continue to be used by practitioners to record specialist assessments and detailed (private) casework.
- **IS Index**  
The IS Index sits at a National level.

It supplies basic information about the child and any practitioners working with them. It also provides information about any CAFs that are in existence.

The diagram shows a practitioner using both eCAF and a Case Management System. In fact this view is somewhat simplified - as practitioners may also use the IS Index, and there are options for access to eCAF via Case Management Systems. (This is discussed further below). However the point of the diagram is to illustrate that eCAF and CMS are distinct systems, each being the correct “tool” for a specific job. In many cases practitioners will see them in this way, and will require access to both.

## 3.2 Interfaces overview

Interoperability between the systems discussed above can help practitioners with their work, particularly in the area of information sharing. Common standards, as defined in this document set, make this possible. The numbers on the diagram highlight the key interfaces that an eCAF system must (solid line) or may (dotted line) provide to support links between systems:

### 1. eCAF – eCAF

At a minimum, eCAF systems must be able to transfer CA Episode data between different Local Authorities when a child moves house. (More advanced scenarios might also see eCAF systems “talking to each other” in response to cross-border enquiries)

### 2. eCAF – IS Index

eCAF behaves like Case Management Systems in this respect. It uses the IS Index as a source of definitive basic information about the child, and passes on updates to child data that it receives.

However the IS Index also maintains a specific “CAF Flag” to track the existence and location of a CA Episode.

### 3. eCAF – Case Management Systems

Three workable levels of integration between eCAF and Case Management Systems have been identified.

**i) No integration** – the systems remain separate, each fulfilling their own role. The practitioner does private Casework in their CMS, and shared working on CAFs in the Local Authority eCAF system.

**ii) Integrated referrals** – this allows CAF data to be passed through to a CMS when a child is referred. The Case Management System must map the data items as necessary and use them to pre-populate a specialist assessment

**iii) Full integration** – The CMS acts as a “front end” to eCAF, and the practitioner accesses eCAF data from within their existing Case Management System. It is important that this provides true online access to the eCAF database, and does not encourage siloed working on a private copy of the data.

It is important to note that these three levels are not mutually exclusive, and that different Case Management Systems may co-exist at different levels, or progress through the levels over time. It is also important to note that this section provides a provisional view and it will be important to monitor developments on other ECM projects (such as IS Index) and maintain a compatible approach.

#### **4. eCAF – Offline devices**

This link shows the possibility for practitioners to complete Common Assessments “in the field”, using laptops, digital pens, PDAs, and so on. Interfaces allow for connecting and uploading the results on return to base.

### **3.3 Interfacing mechanisms and evolution**

Having outlined the interfaces at a conceptual level, the rest of this document looks at the mechanisms that can be used to implement them.

Two mechanisms are described in detail:

- File import/export
- Web Services

Once again, these are not envisaged as mutually exclusive, and a progression is expected over time:

- In the early stages, eCAF systems are likely to be largely self-contained - with the focus being on getting core functionality working and used by practitioners. Interfacing might be by simple file import/export.
- The next step might involve greater use of Web Services to further automate the links between systems and, for example, to allow connection to eCAF from Case Management Systems
- The final goal is an extremely dynamic environment - with local eCAF systems, Case Management Systems, and the IS Index all interoperating and exchanging data to provide a seamless experience for practitioners.

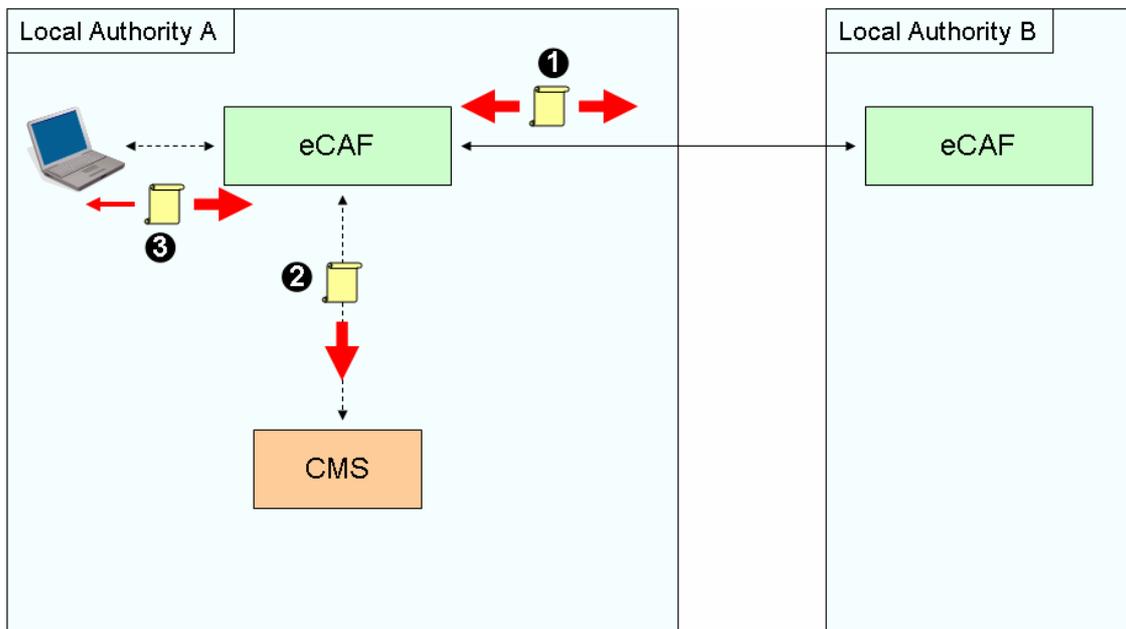
This last stage is the ultimate aspiration and may take some time to evolve. However it will quickly become possible – with the IS Index in particular helping to promote interoperability and a coordinated National approach. The intention in this document, then, is to define the basic interfacing facilities for an eCAF system that can be used as a foundation for this vision as it develops.

## 4. File import/export

### 4.1 Introduction

File import/export is the basic level of interfacing that all eCAF systems must support. While at times cumbersome, file transfer is a simple and well-proven technology that most computer systems can use. The data in the file will be in XML format, as specified by the eCAF XML schema for a CA Episode. XML is, again, a widely accepted standard that almost all computer systems support.

### 4.2 File transfer scenarios



The diagram shows three scenarios where file import/export is able to provide a solution. These are described further in the paragraphs below (the numbers in brackets refer to numbers in the diagram).

In all cases the basic mechanism to import/export an XML file containing CA Episode data is the same - but the surrounding business logic and transfer mechanism vary:

#### 4.2.1 Transfer CA Episodes to another Local Authority (1)

- **Background**  
This will be required when a child moves house. The “old” Local Authority will export an XML file and send it to the “new” Local Authority to import
- **Business Logic**  
This interaction involves the actual movement of the child’s data.

The **export routine** must prompt the user to search for ALL CA Episodes for the child and

export them to a single file. After exporting the file, the child's CA Episode data must be removed from the "old" system.

The **import routine** imports the new CA Episode data. The import routine will then prompt for administrative action to ensure that correct ownership and access permissions of the CA Episode are assigned for the "new" eCAF system.

- **Transfer mechanism**

As the child's data is actually being moved then the transport mechanism must be "**reliable**". This means that it must include backup, confirmation and retry mechanisms to ensure that the file actually arrives and cannot be "lost". (This might be achieved by sophisticated automatic protocols or by careful manual procedures).

The transport mechanism needs to be secure and suitable for transferring **Level 3 security** rated data.

For a fully automated solution that meets these criteria then **secure messaging systems** such as those provided by GovConnect and CJIT should be considered.

For a more manual solution then **secure email** is a possibility – although care is needed that this does not provide system administrators with inappropriate access to child data. (Encrypting/decrypting the data on import/export is a possible solution to this problem).

**"Ordinary" email and internet ftp** are not suitable for Level 3 data. However they too could become options if the data was **strongly encrypted**.

#### 4.2.2 Download referral to Case Management System (2)

- **Background**

This corresponds to "Level 2" integration with a Case Management System. When the child is referred (as a result of CA Action Planning) then the CA Episode data is exported so it can be used by the CMS as the basis of a specialist assessment.

- **Business Logic**

The Business Logic of the **export routine** is simple. The user selects a CA Episode to which they have access, and the export routine exports the CA Episode data to an XML file.

The **import routine** is the responsibility of the Case Management System. It may have to map the data items into the format expected for its own type of specialist assessment. It should delete the XML file after import, to avoid the danger of the data falling into the wrong hands.

- **Transfer mechanism**

One likely mechanism might be file download over a secure internet connection (SSL) to the practitioner's desktop. The practitioner could then import the file into their Case Management System.

### 4.2.3 Upload/download from/to Offline Devices (3)

- **Background**

“Offline devices” is a general term for laptops, PDAs, digital pens, and other similar devices that practitioners may use to work on CAF data “in the field”.

- An upload facility (import) allows Common Assessments to be created remotely.
- A download facility (export) allows practitioners to take a copy of the CA Episode data out with them.
- A synchronisation facility (import) allows practitioners to take away a copy to update (for example with a new CA Action Plan) and upload the changes on returning to base.

- **Business Logic**

The Business Logic of the **export routine** is simple. The user selects a CA Episode to which they have access, and the export routine exports the CA Episode data to an XML file.

A basic **import routine** may also be simple – it needs to check that there is no existing CA Episode with the same Episode Id and then import and save the data. Note however that the import routine must write to the audit log, and perform validation checks rather than relying on the integrity of the imported data.

A **synchronisation facility** adds to this the ability to import and merge updates into existing CA Episodes. The CA Episode includes a version number which must be checked for concurrency clashes. The data structures are designed so that records are always added and never changed – this is intended to help facilitate the merging process.

- **Transfer mechanism**

The exact details of transfer from offline devices are too varied to speculate on. However it is anticipated that the end result could be an XML file a computer connected to the network. This can then be uploaded to eCAF - for example over a secure internet connection (SSL).

## 4.3 Reference Specifications

- **eCAF XML Schema**

The eCAF XML schema defines the format for an XML file of CA Episode data

## 5. Web Services

### 5.1 Introduction

By implementing Web Service interfaces, an eCAF system can provide an enhanced level of integration support.

Web Services are now a well-established standard for enabling interoperability between computer systems across the internet. They allow for a more seamless user experience than can be achieved using file transfers.

For example, Web Services allow for “Level 3” integration with Case Management Systems. “Level 3” is defined as allowing the CMS to act as a “front end” to the eCAF database, enabling the practitioner to access and work on CAFs from their familiar Case Management System user interface. This level of online access would not be feasible to implement using file transfers.

In summary, several benefits of Web Services make them a good choice for eCAF:

- Slicker user experience (compared to, for example, file transfers)
- Standards based
- Interoperable (independent of underlying platform)
- Future proof
- Service Oriented Architecture (SOA) compatible
- Good tool support

However while the basics of Web Services are well established and straightforward, implementing a secure Web Service interface over the internet is still challenging. Success is dependant on facilities provided by the surrounding infrastructure, and Appendix A introduces some of the considerations around this.

## 5.2 Logical interface

This section provides details of the standard Web Service functions for an eCAF system. It is at a “logical” level, in that it explains in detail how the interfaces work – but without describing the physical standards and transport mechanisms used. The Logical Interface is likely to be relatively stable over time, changing only if new business functionality is added to eCAF.

### 5.2.1 Web Service principles and patterns

The design of the logical Web Service interface has been influenced by the guiding principles of:

- ***Simplicity***
- ***Interoperability.***

These principles are important, because the eCAF Web Service interfaces may need to be implemented by many different developers on different platforms. This means that the simplest possible approach has been used, avoiding immature standards and “advanced” techniques. The intention is to ensure that the interfaces can be implemented using widely available and proven tools and techniques.

As consequence of accepting only established standards, the following simplifications have been assumed:

- Only the request-response synchronous messaging pattern will be used.
- Only stateless Web Services will be used
- Web Services will not be transactional i.e. the consumer and provider will not participate in transactions.
- The interfaces will be designed so as not to require reliable delivery.

Further detailed explanation of these points is contained in Appendix B.

### 5.2.2 Web Service functions

The tables below outline the Web Service functions that a Web-Service-enabled eCAF system must provide. They are intended as a brief introduction to the formal **eCAF WSDL specification**.

<b>Get Episode</b>	
<b>Input parameters</b>	<ul style="list-style-type: none"> <li>• Episode Id (guid)</li> </ul>
<b>Output parameters</b>	<ul style="list-style-type: none"> <li>• CA Episode (as defined in eCAF XML schema)</li> </ul>
<b>Operation</b>	<ul style="list-style-type: none"> <li>• Finds and returns the requested Episode</li> <li>• Must check access rights to ensure that the user is able to access this Episode</li> </ul>

<b>Save Episode</b>	
<b>Input parameters</b>	<ul style="list-style-type: none"> <li>CA Episode (as defined in eCAF XML schema)</li> </ul>
<b>Output parameters</b>	<ul style="list-style-type: none"> <li>Acknowledgement return code</li> </ul>
<b>Operation</b>	<ul style="list-style-type: none"> <li>Saves the Episode data in the eCAF database</li> <li>Creates the Episode if it is a new Episode Id</li> <li>Checks for concurrency clashes and merges the data if it is an update</li> <li>Must check access rights, validate data, and update audit log</li> </ul>

<b>Search Episode</b>	
<b>Input parameters</b>	<ul style="list-style-type: none"> <li>Search Request (as defined in eCAF XML schema)</li> <li>This includes the child's name, address and date of birth</li> <li>The search can be limited by status of the CA Episode (eg to show only open Episodes)</li> <li>The search can be across the whole system, or restricted to only those Episodes to which the practitioner already has access</li> </ul>
<b>Output parameters</b>	<ul style="list-style-type: none"> <li>Search Results (as defined in eCAF XML schema)</li> <li>A list of matching Episode Ids</li> <li>Also includes basic information about each Episode – for example version number, status, start date, and child name, address and date of birth</li> </ul>
<b>Operation</b>	<ul style="list-style-type: none"> <li>Searches for Episodes matching the requested criteria, and returns a list of results</li> </ul>

<b>Notify Transfer</b>	
<b>Input parameters</b>	<ul style="list-style-type: none"> <li>CA Episode (as defined in eCAF XML schema)</li> <li>Confirmation Endpoint (the address to call back and confirm the transfer once it is successfully received and processed)</li> </ul>
<b>Output parameters</b>	<ul style="list-style-type: none"> <li>Acknowledgement return code</li> </ul>
<b>Operation</b>	<ul style="list-style-type: none"> <li>Similar to Save Episode</li> <li>It is different from Save Episode in that this operation is part the business process to transfer a child between Local Authorities. Therefore after performing the save it may need to initiate further administrative actions</li> </ul>

<b>Confirm Transfer</b>	
<b>Input parameters</b>	<ul style="list-style-type: none"> <li>Episode Id (guid)</li> </ul>
<b>Output parameters</b>	<ul style="list-style-type: none"> <li>Acknowledgement return code</li> </ul>
<b>Operation</b>	<ul style="list-style-type: none"> <li>This message indicates confirmation that the receiving system has successfully received and processed the transfer. The sender can now remove the record from its database.</li> </ul>

### 5.2.3 Fault codes

The logical interface uses the standard SOAP fault handling mechanism to notify of errors. The following fault categories are defined:

<b>Code</b>	<b>Meaning</b>
<b>101</b>	Service not available
<b>102</b>	Timeout
<b>103</b>	Permission denied
<b>199</b>	Miscellaneous – any other infrastructure level error
<b>201</b>	Not authorised – the user requested a record or operation that they do not have access to
<b>202</b>	Not found – the requested record was not found
<b>203</b>	Concurrency error – the user's attempt to save has clashed with updates by another user
<b>299</b>	Miscellaneous – any other application level error

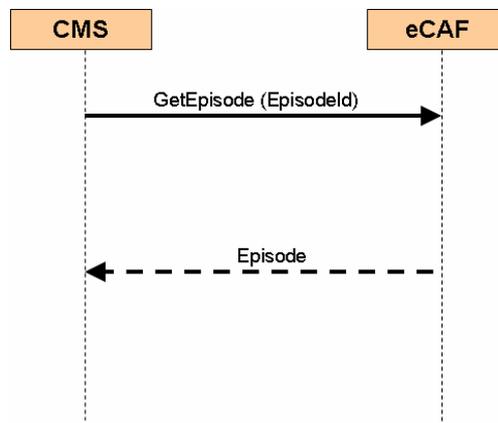
The fault structure also allows for further detailed coding and free text description of the error.

## 5.2.4 Interactions

The diagrams below indicate how the Web Service functions are used in interaction scenarios. (NB: Note that the background information and business logic in the Section 4.2 “File transfer scenarios” is still relevant here and will not be repeated. Also note that, for simplicity, the diagrams show eCAF interacting with a Case Management System (CMS). However this could in fact represent any kind of other system, portable device, another eCAF system, etc)

### Get Episode

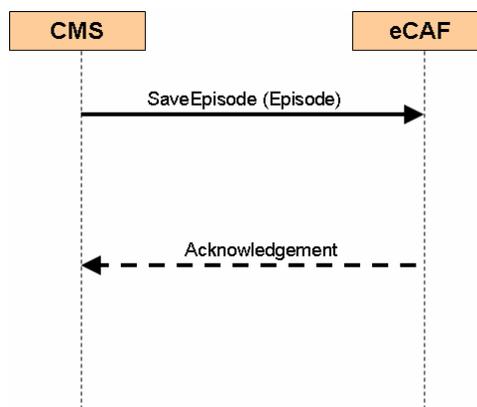
In this example the CMS already knows the Episode Id – maybe due to storing it as an additional data field about the child. It calls the eCAF system and receives the full Episode details in return.



### Save Episode

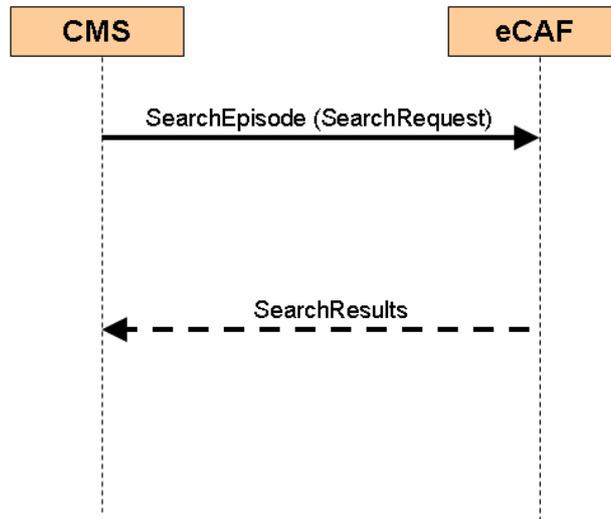
This example shows an Episode being created in the CMS and then saved to back to the eCAF system.

The Save function should be idempotent (more than one invocation produces the same result as a single invocation and does not put data integrity at risk) - so that it may be safely retried in the case of failure.



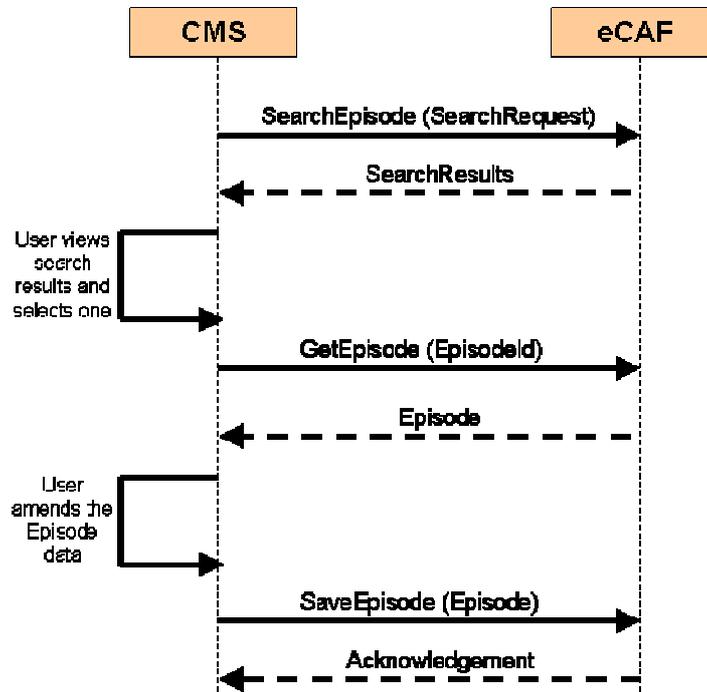
### Search Episode

This example shows the CMS user performing a search and receiving back a list of basic information about Episodes that match the search criteria.



### Search, Get, and Save Episode

This example shows several of the functions combined in a realistic scenario. It shows the user searching for an Episode, selecting one to view and getting the full details, making some changes, and then saving the new version.



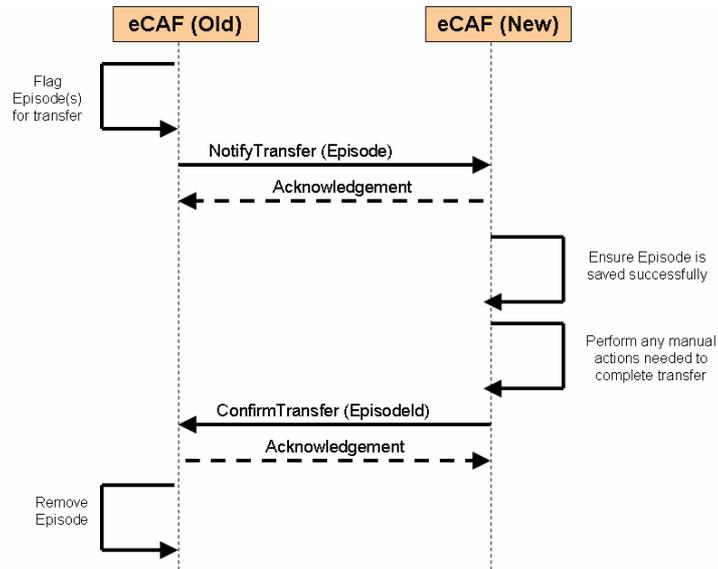
### Transfer Episode

This diagram shows the transfer process between eCAF systems when a child moves house. In order to ensure reliable delivery it is a two step process. This is done at the application level –

both to keep the infrastructure requirements simple, and because of the possible time delay before confirmation.

The “transfer” functions should be idempotent (more than one invocation produces the same result as a single invocation and does not put data integrity at risk) so that they may be safely retried in the case of failure

(Note also that the two “transfer” messages are in reality one-way asynchronous calls. However they are implemented as a request-response with a dummy “acknowledgement” return code. This is for practical interoperability reasons as WDSL 1.1 does not support a one-way message exchange pattern).



### 5.3 Physical interface

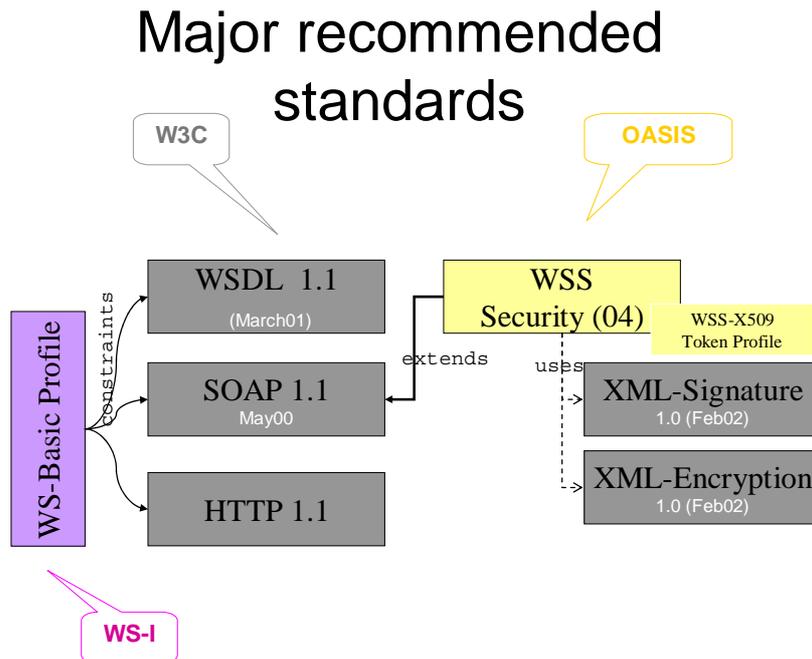
This section contains recommendations for the Web Service physical transport layer. As stated in Section 2.2, it is beyond the scope of this document to mandate the physical implementation layer. However by applying the principles of simplicity and interoperability it is possible to recommend widely available and well established standards.

It is also worth noting that the physical layer may need to be reconsidered over time as Web Service technology develops. An important advantage of Web Services is that the details of this physical layer can evolve independently from the logical interface definitions.

***This section must be regarded as provisional. It is anticipated that projects such as IS Index will also define physical Web Service standards. These physical interface recommendations must be reviewed for compatibility at that stage.***

#### 5.3.1 Recommended standards

The diagram and list below summarise the recommended Web Service standards. These have been selected as “lowest common denominator”, well supported standards.

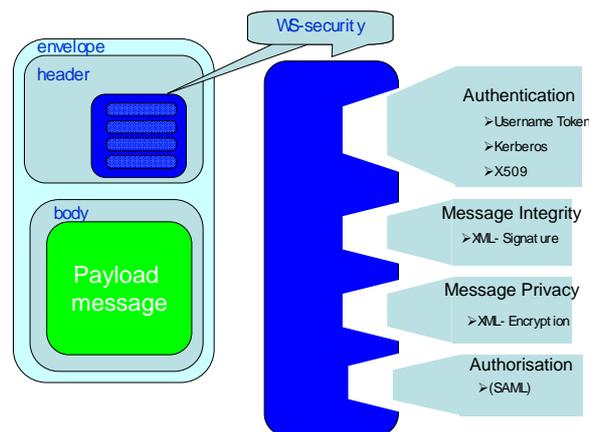


- WS-specific W3C standards
  - WSDL [1.1] Web Service Description Language
    - This older standard is recommended as per WS-Basic Profile to improve interoperability
- WS-I (Web Services Interoperability Organization)
  - WS-Basic Profile [1.1], as restricting and clarifying the usage of other specs
- W3C standards:

- XML [1.0]
- XML Schema
- Protocols
  - SOAP [1.1] with restrictions/clarifications as per WS-Basic Profile
  - HTTP [1.1] as above
- OASIS Open
  - WS-Security [1.0]
- W3C – security related
  - XML Encryption [1.0]
  - XML Signature [1.0]

### 5.3.2 WS-Security recommendations

Security infrastructure is again beyond the scope of eCAF to mandate. However the WS-Security standard is well established and the diagram below shows a schematic of the facilities it can provide:



WS-Security allows for message-level encryption and signing. This is in many ways preferable to transport-level security (eg SSL) - as it ensures end-to-end security for the message. WS-Security also allows for exchange of security tokens. There are various mechanisms for establishing trust and acquiring tokens, with the detailed approach being highly dependant on the surrounding infrastructure. (For example facilities for federation and single-sign-on)

### 5.3.3 WS-Policy recommendations

WS-Policy is intended as an automated mechanism for web services to declare policies in areas such as security. It has the potential to allow Web Services to state their physical interface requirements and thus interoperate across diverse infrastructures. However its use can NOT be recommended at present because:

- It is not recommended by (i.e. outside) BasicProfile 1.1 (could be by the next level of this standard)
- An attachment to WSDL not formally available in WSLD 1.1

- Standards related to the WS Policy (i.e. WS Policy Framework, 2004) are emerging but do not have as yet, widely available implementations (WS-SecurityPolicy, 2002 is obsolete)
- Older implementations of app servers have often proprietary elements

The standard should be monitored and reconsidered when/if this position changes

## 5.4 Reference Specifications

- **eCAF XML Schema**  
The eCAF XML schema defines the XML format to be used by Web Services for passing CA Episode data, and other web service function parameters
- **eCAF WSDL specification**  
The eCAF WSDL specification describes the web service functions in a standard, formal, and machine-readable format
- **Web Service standards**  
See Section 5.3.1 above for a reference list of Web Service standards and protocols
- **IS Index interface specifications**  
It is anticipated that the IS Index will expose Web Service functions. When available, these specifications should be examined in order to maintain a compatible approach.

## 6. IS Index Integration

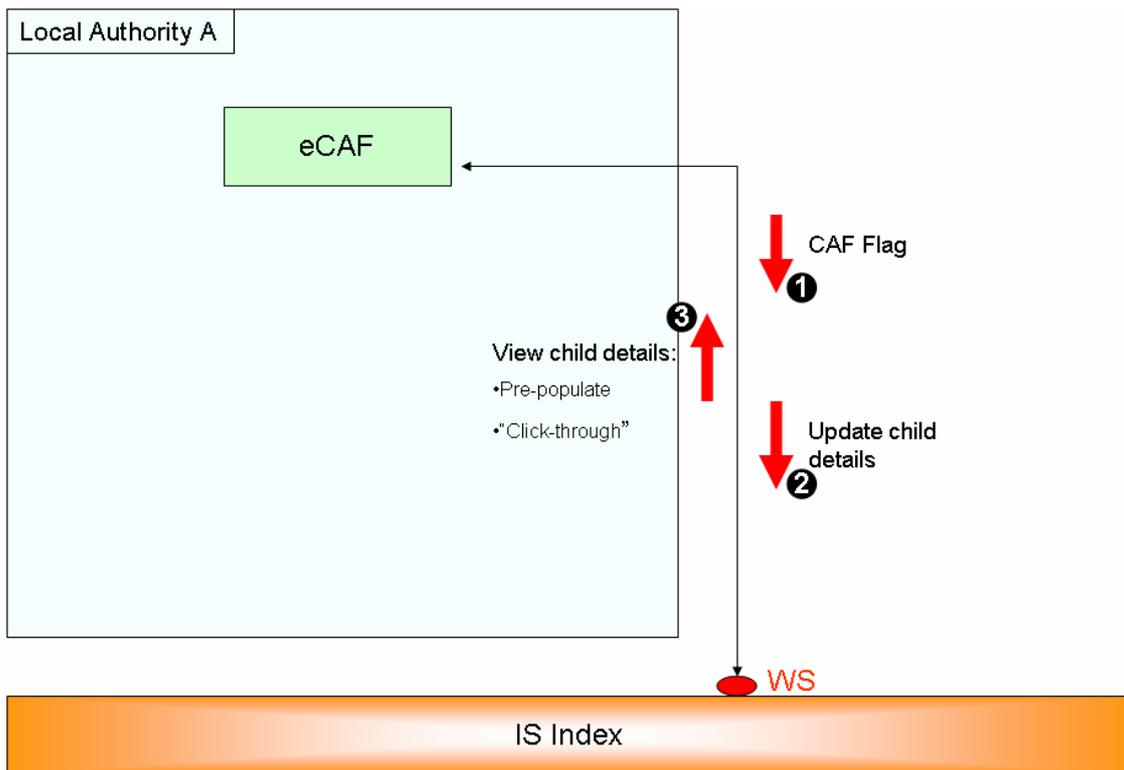
### 6.1 Introduction

This section outlines at a “conceptual” level the integration touch-points between eCAF and the IS Index.

It is rather different to the other sections of this document, as the IS Index interfaces will be specified and implemented as part of the IS Index project, with eCAF acting as the interface consumer. Therefore technical details are not given here. (Although Web Service interfaces are anticipated).

Rather, this section describes – based on provisional plans for IS Index– the anticipated interactions between eCAF and IS Index. It outlines some business scenarios where these interactions will be beneficial.

### 6.2 Interactions



The diagram shows three scenarios where interaction with the IS Index is envisaged. These are described further in the paragraphs below (the numbers in brackets refer to numbers in the diagram).

### 6.2.1 The CAF Flag (1)

The most important interaction with the IS Index is notification of the “CAF Flag”. When a CA Episode is created (or has a significant status change) then the eCAF system must notify the IS Index of its existence. This ensures that it is always possible to be aware of the existence of a CAF, even if it is in a different Local Authority eCAF system.

### 6.2.2 Update child details (2)

An eCAF system can act as a valuable source of information about child details – helping to keep the IS Index up-to-date. In this respect it is the same as Case Management Systems – providing a feed of data to the IS Index when a practitioner makes updates to eCAF.

### 6.2.3 View child details (3)

The IS Index will be useful as a definitive source of up-to-date information about the child when working on a CAF. Two examples would be:

- **Pre-populating a new CA Episode**  
If a practitioner is creating a new CA Episode, it will be useful to access the IS Index and view details of the child. (This will include other practitioners already working with the child). An eCAF system should allow the data from the IS Index to be automatically mapped into CA Episode fields – thus saving typing by pre-populating the CAF.  
***NB: This scenario is dependant on legal advice as to whether this is an appropriate use of IS Index data. As of this writing (June 2006) the situation is not yet clear.***
- **“Click through” to latest child details**  
When working on an existing CA Episode, it will be useful to allow the practitioner to easily “click through” to the IS Index to view latest details of the child. This would allow changes and new practitioners to be picked up. An eCAF system should allow the practitioner to automatically “refresh” the child’s details with the latest information from IS Index on request.

## 6.3 Reference Specifications

- **IS Index interface specifications**  
The details of interfaces to IS Index will be defined by the IS Index project as part of the forthcoming technical design phase. Specifications will be published in due course. (A Web Services approach is anticipated).

## 7. Appendix A - Infrastructure Considerations

As explained in the introduction and eCAF system will be installed into the IT infrastructure of the UK public sector – and this infrastructure is beyond the scope of eCAF and this document. Nonetheless, the capabilities of the infrastructure are developing rapidly, and hold the promise of significant benefits for practitioners using eCAF. This section briefly outlines some considerations to be aware of.

### 7.1 Authentication / Single Sign On

#### 7.1.1 Relevance to eCAF

eCAF is a secure system requiring the highest security level (Level 3) of authentication for practitioners wishing to access it. (See the **eCAF Security Architecture** for more details) This is potentially burdensome as, in the worst case, practitioners will require a separate, dedicated, secure logon token for every Local Authority eCAF system that they need to use.

On the other hand, widespread implementation of Single Sign On (SSO) might mean that many practitioners need no security tokens at all. They would log in to their local network as usual and be able to pass through to “just use” any eCAF system.

Benefits of Single Sign On therefore include:

- Ease of use for practitioners
- Reduced need for security tokens
- Delegated user administration
- Improved cross-boundary working

In addition to user logon, authentication is important for Web Service interfaces - as these also need to be secure. There are additional options available for this, including the passing of credentials between trusted applications. However, one way or another, the authentication issue needs to be solved before Web Services can be used.

#### 7.1.2 Current situation

Using Single-Sign-On terminology, an eCAF system needs to be able to function as an **Identity Consumer**, with the **Identify Provider** being one (or more) of the several competing federations springing up in the UK Public Sector (eg GovConnect, CJIT, Athens). Unfortunately much of this infrastructure is still under development and the different initiatives are not necessarily compatible with each other. Commercial “identity management” products may be able to help to some extent, and the situation requires continued monitoring.

## **7.2 Messaging Hubs**

### **7.2.1 Relevance to eCAF**

Messaging hubs are sometimes known as “email for computers”. They allow one computer to post a message containing data, which is later delivered to another computer for processing. Typically messaging hub products contain functionality with “just takes care of” all the problems associated with transport mechanisms, connectivity and security.

When considering the options for file import/export in Section 4, a messaging hub provides a solution to the question of how the file is actually moved around between systems.

Some of the interfacing needs of eCAF involve synchronous (online) access by a Case Management System user, and these are not a good fit with a messaging hub solution. However the transmitting of referral data, and the moving of a child’s CA Episodes between Local Authorities are more likely candidates. (In the last case, reliable delivery is required – and this is often provided by messaging hubs).

### **7.2.2 Current situation**

GovConnect provides a messaging hub, as does Criminal Justice IT (CJIT)

## **7.3 Service Registry**

### **7.3.1 Relevance to eCAF**

In some of the more sophisticated integration scenarios, different applications will need to find out about each other. For example, a Case Management System may need to locate the eCAF Web Service for a number of Local Authorities. In addition, trust relationships need to be recorded, allowing the agreed relationships between multiple practitioner groups and Local Authorities to be securely checked before allowing access.

Service Registries act as a “Yellow Pages for computers” to store this kind of information.

### **7.3.2 Current situation**

No known initiatives in this area at present

## **7.4 Secure email**

### **7.4.1 Relevance to eCAF**

Secure email may provide a way of moving files of CA Episode data around. The most obvious possible application would be transferring a child’s data between Local Authorities when they move house.

### **7.4.2 Current situation**

GovConnect is providing a secure email service

## **8. Appendix B – Web Service Patterns**

This Appendix provides more detailed reasoning behind some of the Web Service usage patterns selected

### **8.1 Request – response pattern**

The interface functions are likely to be invoked by a Case Management System (CMS) user on an item-by-item basis as they work with a single Episode. The synchronous request - response pattern therefore, fits well with the service requirements of the CMS interface.

The Transfer functions are a little different and a more sophisticated, asynchronous/reliable pattern would be better suited to this interface. However, it was felt that the eCAF business requirements can be met without putting this additional requirement on the infrastructure.

### **8.2 Stateless services**

The limitation to stateless services is mandated for simplicity. The data sets should be small and the functions are simple – therefore there is no need to complicate implementations with additional state management.

### **8.3 No transactions**

Standards related to transactions are in preparation however, implementations are not readily available. Arguably, functions such as “save” or “transfer” could be modeled as transactional between eCAF and the Case Management System. However, it is felt that the eCAF requirements can be met without transactional services and the additional interoperability complications entailed. The Web Service functions are all simple - and specifically either read-only or idempotent. This makes retries easy, and mitigates the lack of transactional control.

### **8.4 No reliable delivery requirement**

The lack of reliable delivery has no impact on most of the eCAF functions. For the “transfer” function, while reliable delivery might be appropriate it is felt that the business requirements can be satisfied without imposing this condition on the infrastructure. The client, using a definition-driven process, will handle the cases of failed invocations.