caBIG® Clinical Information Suite
Deployment Team

caBIG® Clinical Information Suite

Architecture Team

SOFTWARE ARCHITECTURE DOCUMENT

TABLE OF CONTENTS

77 # DOCUMENT CHANGE HISTORY

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
0.1	Raghu Chintalapati, Lloyd McKenzie, David Bass, George De La Torre, Jingdong Li, Satish Patel, Kunal Modi, Harsh Marwaha	4/14/2011			Initial draft
0.2					
0.3					
0.4					

78

79

1 Introduction

The Software Architecture Document (SAD) is intended to provide an overview of the major software components of the system. It is intended to be used by the software architects, software developers, product managers, and anyone who is interested in the high level design/architecture of the software being built. The document provides

- The organization of the software system
- The selection of structural elements and their interfaces by which the system is composed
- Their behavior, as specified in the collaboration among those elements
- The composition of these elements into progressively larger subsystems
- The architectural style embraced by the software architect that guides the project

In addition, the document is also concerned with usage, functionality, performance, reuse, comprehensibility, technological constraints, and trade-offs.

2 System Objectives and Considerations

The proposed architecture has been developed with the aim of most efficiently achieving the following objectives and taking into account the outlined considerations.

1. Provide a standardized mechanism to allow TRANSCEND information to be shared with external EHR applications, including the caEHR PCO application.
2. Create a caEHR PCO application to allow gathering, editing, authoring and distributing information about patient-centric outcomes
3. Maximize the chances for uptake and re-use of the solution by EHR and other clinical application vendors and developers working in the oncology space. (E.g. use freely distributable or open-source components, leverage well-adopted standard interfaces, etc.)
4. Allow extensibility to support additional data elements and additional exchange formats as new needs arise.
5. Do not build components from scratch if off-the-shelf solutions already exist.
6. Ensure that a useful, production ready system is in place and tested by the Sept. 30 deadline
7. Leverage design and development artifacts created as part of the prior caEHR project phases where appropriate.

3 Approach

The technical approach in accomplishing the system objectives includes three main strategies.

Strategy 1: Create only what will be tested by the current implementation

Creating detailed designs or writing code that is not actually required for the current scope carries several risks. The requirements are not known, the content cannot be tested in a real-world environment, and investing resources in non-needed content diverts them from the artifacts that are required to achieve success in our limited timeframe.
Examples of this approach include:

- Limiting our focus to data elements available in TRANSCEND or required in the CCD or other publication formats rather than attempting to construct a "best practice" data model whose data elements will not be fully populated by the initial implementation
- Not introducing registry or other service functions not required to deliver in-scope functionality

Strategy 2: Design for anticipated changes and reuse

While it does not make sense to invest resources in creating functionality that will not yet be used, an implementation that is only capable of satisfying the current project scope and cannot be expanded would defeat most of the project objectives. The solution will therefore be designed to accommodate expected future requirements making it easy to add additional features as needed.
Examples include:

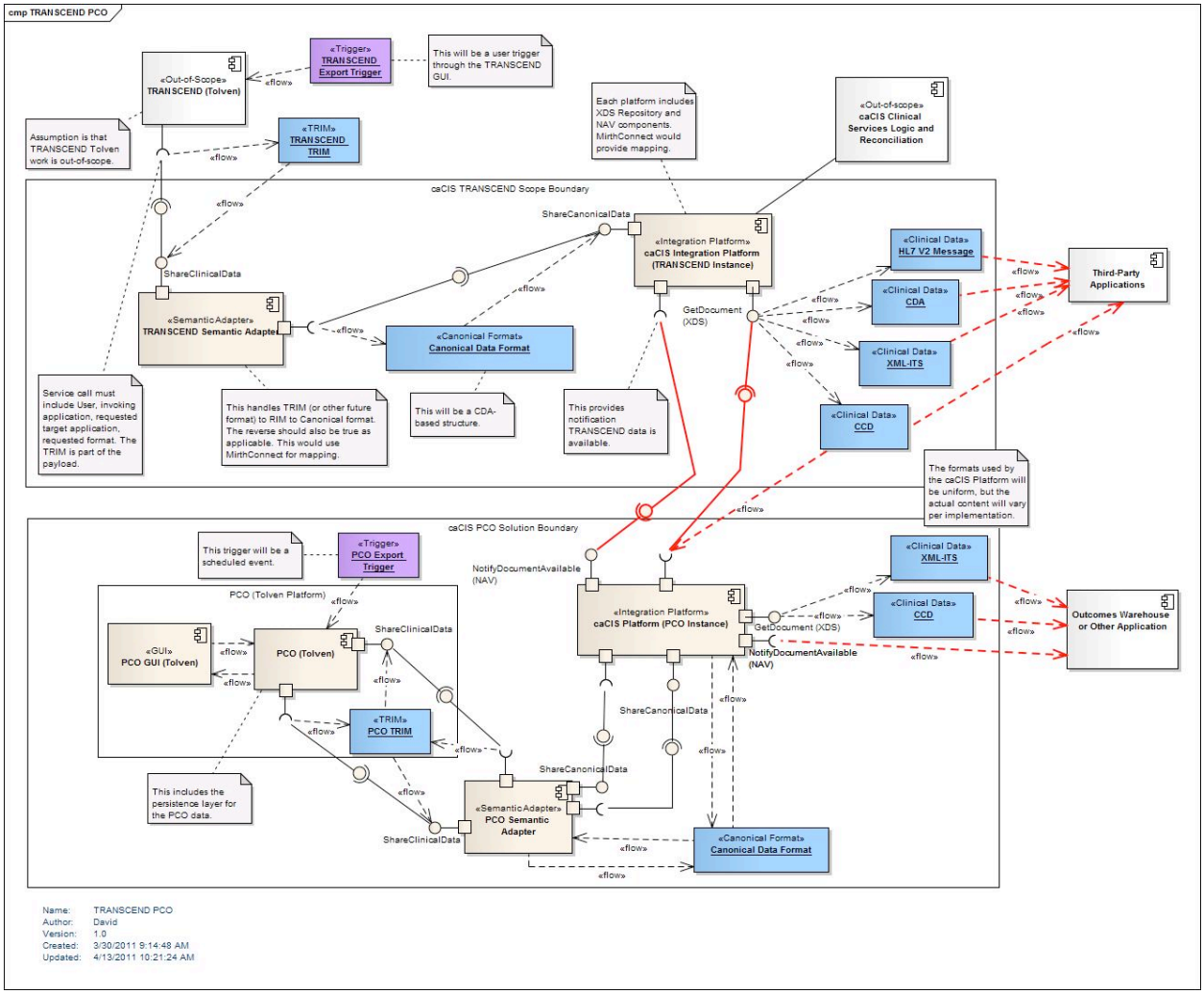
- Allowing easy expansion of the set of supported data elements in subsequent iterations
- Address the reuse objectives of the project, making it easy to adapt the proposed solution to other sites, besides Transcend
- Defining interface points to allow data integration from multiple sources
- Ensuring technology choices will allow easy introduction of new exchange formats and integration of new clinical systems
-

Strategy 3: Leverage existing standards and off-the-shelf components where possible

Using standards assists several objectives: it reduces project risk by reducing the amount of custom development and testing; it encourages adoption by clinical system vendors and developers; and it

4 Proposed Architecture

4.1 Overview



4.2 Components

The solution to meet in-scope requirements includes the creation of a Tolven instance to support capture and persistence of patient-centric outcome (PCO) data and a set of components to be used with both the PCO and TRANSCEND Tolven instances to allow the standardized sharing of data both with each other as well as with other applications to be defined. Because the architecture of the PCO Tolven instance is essentially fixed by the design of the Tolven application, this document will focus on the architecture of the communication/integration components.

The communication/integration portion of the architecture consists of two major modules – the *Semantic Adaptor* and the *Integration Platform*. There will be separate instances of these – one for each transcend instance. They will share information using a standardized *Canonical Data Format*. In a future implementation involving additional EHR applications, additional instances of the *Semantic Adaptor* and integration platform would exist for each participating EHR system.

4.2.1 Semantic Adaptor

The *Semantic Adaptor* is responsible for converting data exposed by the EHR application being integrated (for this project, the TRANSCEND or PCO Tolven instance) to and from the internal format used by that EHR system to expose its data. This component functions as the "transformation" part of an interface engine. It is responsible for handling missing data, generating human-readable renderings, performing vocabulary translations and any other clean-up that is needed to expose the EHR application's data in and consume data from a standardized form.

Each instance of the *Semantic Adaptor* will require significant configuration to tune it to the local format supported by the EHR application being integrated. The development of the mapping configuration for each adaptor will generally need to be the responsibility of the vendor of that EHR system. However, the project will create the mapping configuration for the two in-scope Tolven instances. A key consideration in the selection of the underlying interface engine used by the *Semantic Adaptor* and in the design of the *Canonical Format* mapped to and from will be making the mapping process as easy as possible for other EHR implementers.

The *Semantic Adaptor* will make use of off-the-shelf components for both the transformation functionality and for the vocabulary mapping functionality. If possible, a single component will be used for both. However, if the selected interface engine's vocabulary translation capability is not sufficiently sophisticated, LexEVS may be used for terminology service functionality. The interface engine under consideration for use in the *Semantic Adaptor* is Mirth Connect. It meets the "free" and "cross-platform" requirements, provides a GUI interface for constructing mappings and is familiar to both the development team and the Tolven vendor. (See 6.1.1 – Mirth Connect.)

The *Semantic Adaptor* will expose a simple service interface to allow EHR applications to send data for a given patient to the *Integration Platform* as well as, when necessary[1], to request data from the *Integration Platform*. However, because of the simple nature of the services provided,

[1] In most cases, EHR applications will load data using existing application capabilities by retrieving one of the standardized CCD, CDA or v2 representations using the XDS interface. However, if the application does not have the necessary parsing capabilities and is comfortable with direct data import, down-converting from the canonical format to internal format using the *Semantic Adaptor* and importing the data will be a supported alternative.

a simple sockets based interface or other interface mechanism could easily be added if more appropriate for interoperating with EHR applications that do not support services.

4.2.2 Canonical Data Format

While not a true "component", the *Canonical Data Format* is still a key piece of the architecture. The format provides a common, extensible baseline into which data from all EHR data sources will be converted and from which the various standard exchange formats will be generated. The use of this common framework is what distinguishes the architecture from being a simple point-to-point integration engine solution. Mapping EHR data into a common set of data structures helps ensure consistency of data representation and improves data quality for NCI uses.

The *Canonical Data Format* will be based on a CDA RIM structure of a header together with a number of sections, some required, most optional. These sections will be of three types: CCD, regular CDA and "extended" CDA.

CCD: These sections will be used to represent information that can be represented within the CCD structure. This is the data that is most likely to be understood by other EHR applications at a discrete level rather than just a human-readable level. As much information as possible will be exposed at this level.

CDA: Some of the data elements which need to be shared will not exist in the standard CDA syntax. However, they will still be expressible as part of regular CDA. I.e. they can be expressed as part of the clinical statement pattern used for the right-hand-side of the CDA model. While few EHR systems can read arbitrary discrete data from a CDA instance, most can at least understand the human-readable component out-of-the-box. And making custom changes to support data extraction from a particular CDA instance is a process many EHR vendors are familiar with and are willing to undertake if the discrete data is useful enough. To increase re-use, any CDA sections introduced that are not already tightly defined by CCD will align with other common industry templates such as those created by the American Society of Clinical Oncology (ASCO) and the College of American Pathologists (CAP).

"Extended" CDA: Because CDA R2 limits itself to a constrained model (an older version of Clinical Statement) and a fixed version of the RIM and structural vocabulary, there are some clinical data elements that cannot be expressed within CDA proper. Additional RIM classes or attributes and/or newer structural vocabulary is required. The canonical model will capture these within the CDA-like structure, but will add in the additional data elements required.

There are several benefits to using a CDA-based structure:

- The primary exchange syntax being generated (the one most likely to be understood and supported out-of-the-box by EHR vendors) is CCD
- For all other data elements, CDA's "human readable" portion provides a base level of interoperability for all data elements
- CDA's approach of one document with multiple "sections" makes it easy to extend the canonical format in the future by introducing new sections
- CDA uses the RIM as its underlying structure enabling the use of existing templates and data model structures to represent information in a semantically rigorous manner

To reflect the variability of EHR application capabilities and allow for maximum uptake, the canonical data format will treat most data elements as optional unless they are critical for safe data interpretation or minimal delivery of adequate clinical care.

The proposed solution makes "meaningful use requirements" a top priority throughout the mapping and modeling process. The meaningful use final rule requirement (§170.304(i)) requires that CCD and C32 be used to exchange patient summary reports. Whenever a CCD/C32 template exists for mappable TRIM data elements, the team intends to use the template for mapping. For TRIM data element that have no corresponding CCD/C32 template, we will look into the HL7 published CDA IGs (Historical and Physical report, Consultation note, Operative note, MDS questionnaire IG etc.) and reuse the precedent HL7 CDA IG based template whenever possible.

For the TRIM data elements that have no corresponding standard CDA IG based template, we will create new CDA templates by reusing precedent CDA template structure and by using meaningful use final rule designated standard vocabularies, ensuring that the template instance will be valid against the normative CDA r2 XSD. When NCI templates are finalized, NCI can promote the template definition to HL7 for ballot, and promote the cancer related templates as HL7 standard template set for nationwide cancer patient data exchange.

Data exports from clinical applications such as TRANSCEND Tolven will always be performed using the complete set of data available, aiming to populate the Canonical format as completely as possible. While slightly increasing bandwidth considerations, this approach has a number of advantages:

- It eliminates any need for the clinical application to track "state" – knowing what data it has shared previously and what it has not.
- It ensures that only one transformation process from native to canonical representation is required
- It avoids dealing with variations in the granularity of data from different clinical applications
- It reduces the need to merge data or to include metadata identifying whether missing data is to be removed or merely omitted due to bandwidth reasons.

4.2.3 Integration Platform

The *Integration Platform* provides the secure interface that exposes EHR patient data consolidated from one or more sources to interested EHRs. It is responsible for integrating data from all sources, transforming the data to a format that can be understood by a receiver, identifying what content it has available to interested receivers and actually managing the exchange of information in a secure form, ensuring authentication of the systems involved. The *Integration Platform* will behave in the same way for each EHR system it is installed with. Configuration information will only be required to identify communication partners and ensure secure exchange.

Like the *Semantic Adaptor*, there will be a simple service interface that permits the *Semantic Adaptor* to pass data to the *Integration Platform*. The external interface by which that data is accessed by other EHR applications is described below in section 4.2.3.4.

The *Integration Platform* includes a number of functions, including validating data extracted from EHR systems, assembly of information into standardized formats (CCD, CDA, HL7 v2, etc.) for exchange and managing the exchange of that content with other systems including authenticating the communication ends and ensuring appropriate authorization for the exchange. Details on how each of these functions will be provided by the architecture is covered in the following sections.

4.2.3.1 Validation

Some degree of validation of the data provided by an EHR application to the *Integration Platform* is essential to ensure that shared information is "safe" for clinical use. Because the data will be transformed into a variety of formats, it is essential that certain minimal requirements be met by the data content before undergoing transformation. If the data is not valid (unexpected information is present, essential information is missing, or the data is not coded correctly), the resulting transformed data could cause the receiving application to fail or result in inappropriate clinical decisions being made.

Because the *Canonical Data Format* is CDA with extensions, schema validation will be of limited usefulness. In addition to XML schema, expectations for document content will be enforced through the use of Schematron rules validation. These rules can be easily enforced using XSLT in a cross-platform manner. The rules will be produced in an automated fashion through the use of Templated CDA tools that will also be used to create the CDA implementation guides.

4.2.3.2 Data integration

Because the scope of this project does not require integration of data from multiple sources, the data integration component will not be included in the initial release of the *Integration Platform*. However, it is noted in the architecture diagram to make it clear where this function would be performed and to ensure that the software design provides appropriate extension points for this functionality to be introduced in future versions.

4.2.3.3 Publication Assembly

This is the process by which the *Canonical Data Format* is transformed into one of the public exchange formats – CCD, CDA, RIM XML or HL7 v2. This function shares the same requirements as the *Semantic Adaptor* component – it needs to be able to transform and filter data and possibly translate coded data. The main difference is that the transformations are stable because the use of the *Canonical Data Format* insulates the publication assembly process from the variations in the data organization of the various participating EHR applications.

Because of the similarity of requirements, the architecture will re-use the same integration engine component selected for use in the *Semantic Adaptor*. This simplifies deployment and minimizes learning curve while delivering all needed functionality. For simple transformations (such as filtering the canonical form to just CCD or just CDA, the architecture team will examine the possibility of just using hand-coded XSLT, as this may be even faster than using the mapping functions of the interface engine.

Introducing new publication formats will be a simple matter of developing the necessary mapping profile and adding it to the configuration information for the tool.

4.2.3.4 Content exchange

This is the process by which EHR applications request delivery of a particular formatted publication of a particular patient's data. It also represents the process by which data can be "pushed" to other EHR applications in situations where that's the desired application behavior.
For EHR-initiated data retrieval from the *Integration Platform*, there's already a widely adopted IHE Cross-enterprise Document Sharing (XDS) standard that allows retrieval and storage of documents, v2 message structures and similar data constructs. This standard is web services based and meets the requirements for data exchange. Because it is already widely adopted, it increases the likelihood of broader uptake for the artifacts for this project. There are already open-source XDS solutions available that we can leverage rather than building our own secure communication mechanism.

For "pushed" data, the project will leverage IHE's Notification of Document Availability (NAV) standard to inform a recipient that a document exists. The recipient will then execute a standard XDS Get to retrieve the document and will then initiate their local import process. (In the case of the caEHR PCO environment, this will mean transforming to canonical and invoking the Semantic Adaptor to convert the data to PCO TRIM and invoke the Transcend import process.) The rationale for this two-phase "push" is that it avoids any assumptions about the recipient application's network location or availability and gives the recipient application the ability to process received documents as appropriate for its own workflow (via user intervention, off-hours batch load or some other mechanism). It also leverages industry standards and provides for consistent behavior in both "push" (notification) and "pull" (query) modes of operation, should the latter mode be desired in the future.

4.2.3.5 Authentication and Audit

IHE provides a companion standard for XDS called ATNA (Audit Trail and Node Authentication). This profile is also widely supported and meets the project requirements for authentication of the two communicating nodes, secure communication and auditing of the exchanges that have taken place. Rather than introducing protocols that are unfamiliar to EHR vendors, it makes sense to leverage existing standards that have already been incorporated into off-the-shelf standards.

Because the NAV notification doesn't contain any patient-specific information (only a document id, the source application and the target application), authentication and encryption are unnecessary for this portion.

4.2.3.6 Patient Registry

A patient registry is not strictly required for the in-scope functionality for this project. However, patient registry services may become important in future versions of the application where data integration will be required or where there will be multiple clients for the *Integration Platform*. For this reason, the existing Client Registry functionality will be retained as part of the *Integration Platform* functionality. Time allowing, the Client Registry function could be upgraded to expose[2] the PIX/PDQ (Patient Identifier Cross-reference and Patient Demographic Query) interfaces to allow broader and easier uptake by the EHR vendor community.

4.3 Orchestration

There are two communication patterns that need to be supported by the architecture solution:

[2] These interfaces are already available in the open source components being evaluated for inclusion in the solution.

1. Source EHR export pushed to consumer EHR for direct import
2. Consumer EHR querying data from one or more source EHRs

Within both of these communication patterns, there are two communication steps. The first is between the source EHR application and the *Integration Platform* via the *Semantic Adaptor*. The second is between the *Integration Platform* and the consuming EHR application.

4.3.1 Source EHR to Integration Platform

The first step – source EHR to *Integration Platform* – will always happen in the same way. Communication will be initiated from the source EHR for the data it wishes to share. This data will be pushed across the *Semantic Adaptor* into the *Integration Platform* where it will be stored in its canonical form. The specific timing of the push of data may vary from application to application. It could be done real-time as data is updated in a patient record, or it could be handled as an off-hours upload.

There are several reasons for adopting a "push" model from the source EHR application and introducing a data storage requirement for the *Integration Platform*:

1. There is no standard mechanism for EHR vendors to process real-time requests for data and few EHR applications have that capability. Introducing such a mechanism adds considerable complexity to the EHR application that vendors may be reluctant to undertake
2. The performance characteristics of data retrieval from some EHR applications is likely to be poor. When the time needed to transform the data to canonical form is added, not to mention the time to retrieve and integrate data from multiple data sources, a real-time retrieval approach becomes extremely hard to do in the time demanded by an inbound query with a clinician waiting on the other end.
3. Having a data store within the *Integration Platform* removes the "load" of needing to respond to real-time query requests from the EHR application. All the EHR application needs to do is push out the data once each time it changes. It doesn't need to deal with the 10s or 100s of times that particular record might subsequently be queried.
4. Not all EHR applications have 24x7x365 availability. Offloading the data sharing responsibility to a separate component reduces risks of needed data being temporarily unavailable.
5. The approach of data sources pushing information into a data repository is aligned with the architecture supported by the XDS specification.

The data flow for this communication will be as follows:

1. The Source EHR application sends the data to the *Semantic Adaptor* by invoking the "new/updated client data" service
2. The adaptor converts the data to the canonical format and passes it to the *Integration Platform* using an equivalent services interface.

3. The *Integration Platform* validates the data and if it finds any issues, rejects the request logging an error to a file for subsequent analysis as well as passing an error back[3] in the service call response.
4. The *Integration Platform* integrates the data into its persisted record for that patient. (For the purposes of this project, integration will a simple "copy and replace" of any previously submitted canonical instance for the patient.)
5. If necessary, the *Integration Platform* updates the XDS Index indicating that it has data available for the specified patient in the formats supported by the *Integration Platform*.
6. If the XDS Index encounters an issue, that issue is recorded in the log and returned back in the service response, otherwise a success is returned
7. The service response is propagated back to the source EHR application via the *Semantic Adaptor*.

4.3.2 Integration Platform to Client EHR

This "push" mode will be used for passing data from the TRANSCEND Tolven instance to the POC Tolven instance. It may be used for communication to additional future consumers such as Epic as well. In this mode, the steps will be as follows:

1. This process is triggered when a canonical document is registered in the document repository
2. A list of target recipients and required output format(s) for each recipient is determined from the recipients identified as part of the EHR to *Integration Platform* service call.[4]
3. The *Integration Platform* sends a NAV "Notification of Document Availability" to each target application with a document id keyed to the desired format.
4. When convenient, the consumer application invokes the "Get Document" service on the *Integration Platform* to retrieve the identified document.
5. The *Integration Platform* retrieves the *Canonical Data Format* document from the repository and transforms it using the format identified by the document id and returns the resulting document to the requesting application.

4.3.3 Client EHR to Integration Platform

This "pull" mode is likely to be the most frequent data access mechanism for client EHR applications as it is most common in existing software. It is included in this architecture

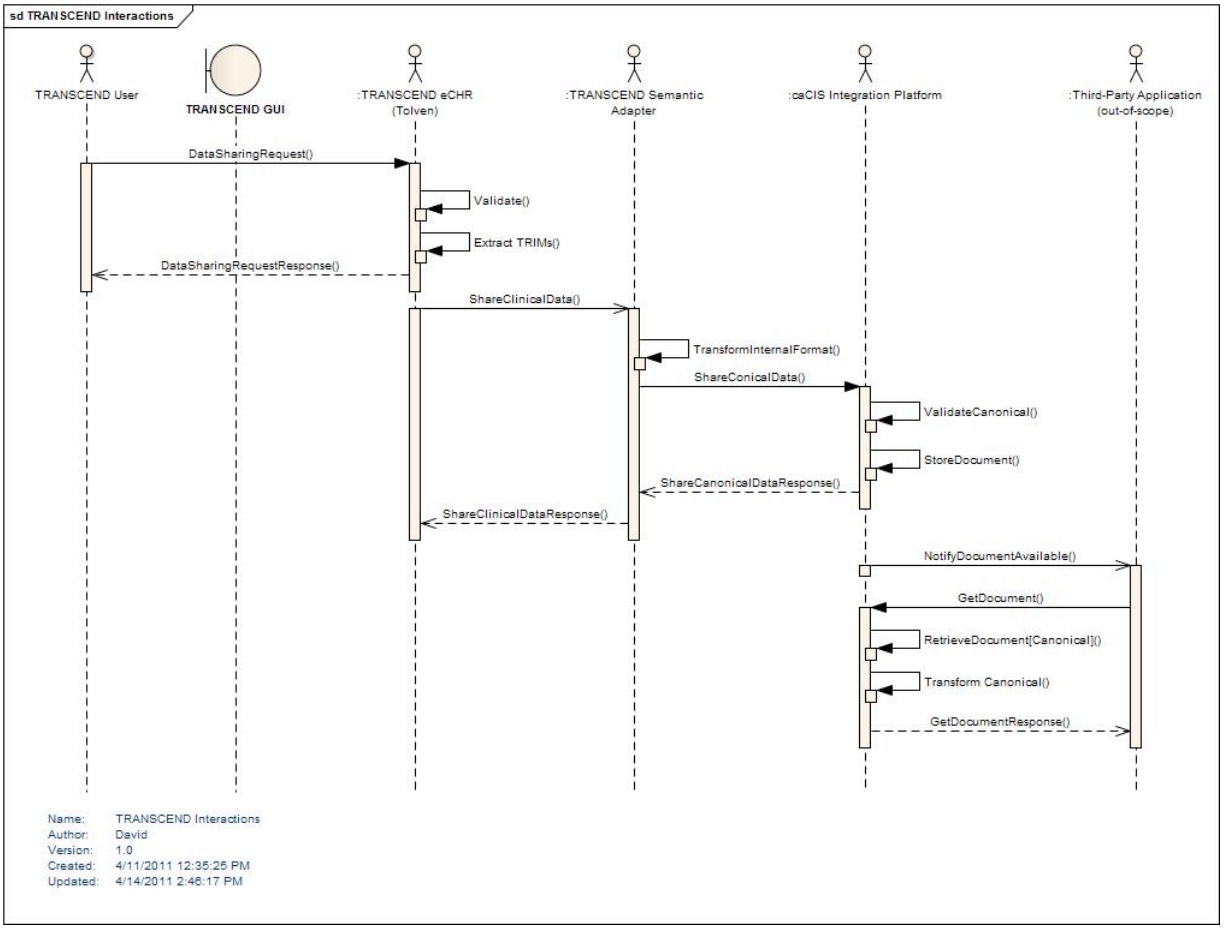
[3] Both mechanisms are supported to accommodate the possibility that some EHR applications may not have a mechanism for handling error responses for exported data, meaning the log is the only way of alerting someone in the source organization that there was a problem.

[4] In the future, recipients and formats could potentially also be determined by pre-set configuration parameters within the Integration Platform, though this would eliminate individual user responsibility for the transmission of a particular patient record to a particular receiver.

document for completeness, as a "pull" or "query" mode is out of scope for this project. In this mode, the steps will be as follows:

1. A client EHR application initiates a query to the XDS Index service interface requesting a list of available documents for a specified patient id[5].
2. The XDS Index service returns a list of the supported document formats
3. The client EHR application initiates a query to the XDS Repository service interface to retrieve the desired document
4. The XDS Repository retrieves the stored *Canonical Data Format* version of the document and transforms it to the requested exchange format and returns it to the requesting application.
5. The client "imports" the information in the retrieved document into its local repository.
6. If there are issues with the import, the application alerts the user who can initiate a manual exception process.

[5] This query may actually be preceded by a PDQ query to determine the appropriate patient identifier to use when querying the XDS index



469

470 **Figure 1 - TRANSCEND interactions**

471

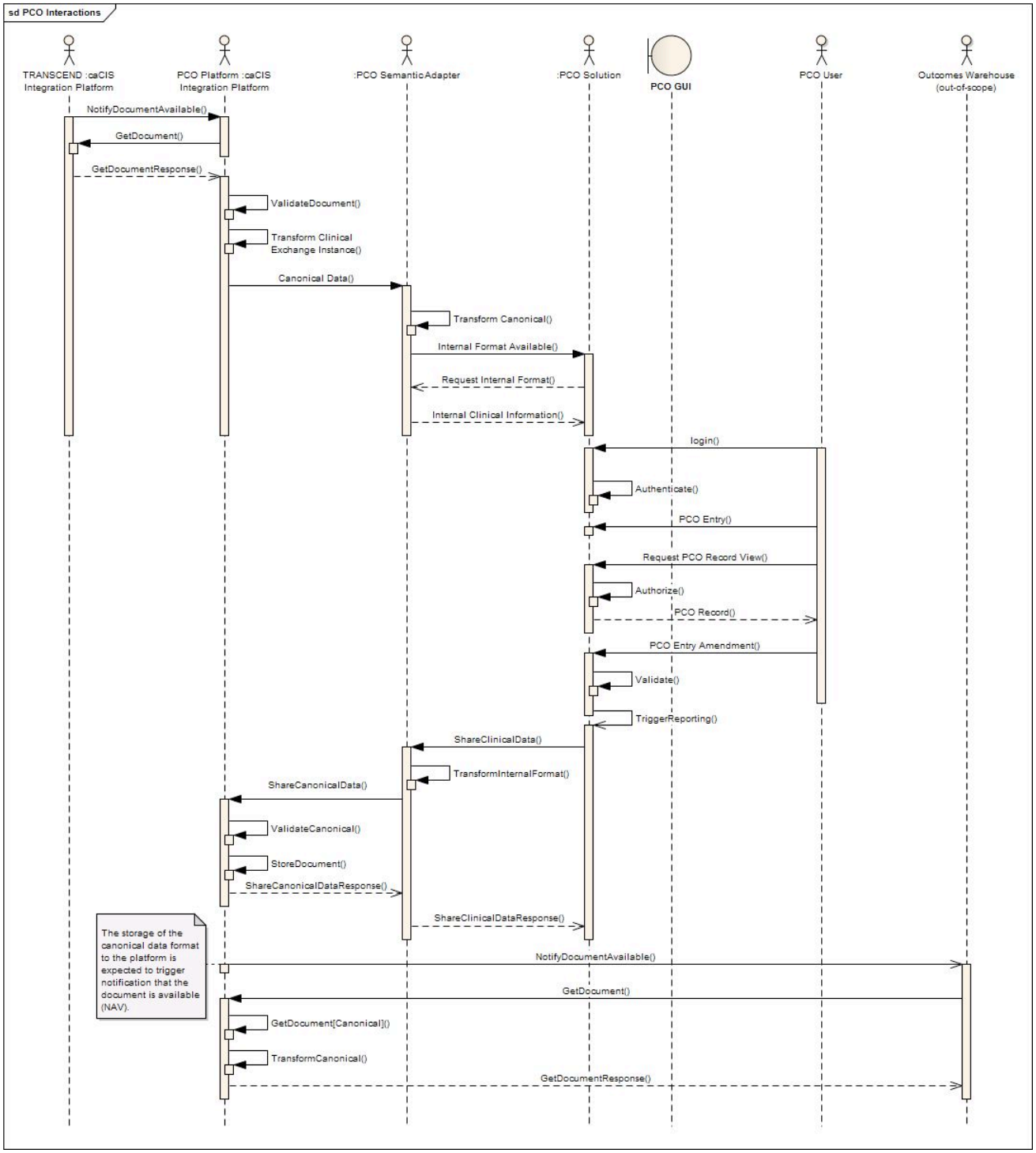


Figure 2 - PCO Interactions

4.4 Security

4.4.1 Security Considerations

Since this integration effort deals with Patient Identifiable Information (PII) and Protected Health Information (PHI), security of the data becomes of prime importance. Also, due to the nature of the data, it falls under various federal regulations and policies such as HIPAA (Health Insurance Portability and Accountability Act), HITECH (Health Information Technology for Economic and Clinical Health) Act.

Security in this context encompasses concerns such as:

4.4.1.1 Patient Consent

HIPAA requires a medical provider to obtain a patient's consent before their medical information can be shared with another medical provider. This is generally obtained before beginning of any treatment or, in case of a clinical trial, at the time of enrollment into a study. This is a patient centric activity and is generally carried out in field by the medical provider or their assistants.

As part of this integration architecture, obtaining or ensuring a patient's consent is out of scope. Front end applications such as TRANSCEND's Tolven which is used to register subject onto the I-SPY2 trial will be responsible for obtaining all the necessary consent as part of the registration process. The integration architecture does not perform any sort of checks to verify if the consent is obtained or present. This is mainly for the reason that each of the front end clinical application (in TRANSCEND's case Tolven) handle and store this consent in a different manner.

4.4.1.2 Restricting Access

The HIPAA privacy law specifies that access to a patient's medical record should be restricted and be available only to authorized users. This implies that the systems which store and present patient data to users should have security mechanism to ensure that only authorized users are allowed access to patient medical records. This is generally achieved by the combination of Authentication and Authorization mechanisms which not only establish who the user is but also control their access privileges.

Since authenticating an end user and checking their access privileges is a function of the front end systems (in this case TRANSCEND's Tolven eCHR), they are out of scope of the current integration work.

4.4.1.3 Securing Transmission between Systems

As this architecture deals with effective transmission of Patient Centric Outcomes as well as Clinical Notes data across the enterprise, securing transmission of PII and PHI is of prime concern from a security perspective. Security in this context includes ensuring that only the intended recipient receives data, that the source of the data is a trusted source, that no third party has access to the transmitted data, and that the data cannot be manipulated between sender and receiver without detection.

There are three points of communication in the proposed architecture:

1. Between a clinical application and the *Semantic Adaptor*
2. Between the *Semantic Adaptor* and the *Integration Platform*
3. Between the *Integration Platform* and other *Integration Platforms* or client clinical applications

The first two of these communication paths are expected to occur in a secure network space. As such, these communications do not require securing. However the third

4.4.1.4 Audit

A key aspect of data security and of some privacy legislation is the ability to audit what information was shared and with whom. This is also a required aspect for this project

4.4.2 Security Approach

To ease integration with existing clinical applications, security requirements will be met by leveraging IHE (Integrating the Healthcare Enterprise) profiles that have been widely adopted in this space. These profiles provide details about how security can be achieved in an integrated health enterprise, in particular when using other IHE standards such as XDS.

4.4.2.1 Audit Trail and Node Authentication (ATNA) Profile

The **Audit Trail and Node Authentication (ATNA)** Integration Profile establishes security measures which, together with the security policy and procedures, provide patient information confidentiality, data integrity and user accountability when data is transmitted between systems. ATNA contributes to access control by limiting network access between nodes and limiting access to each node to authorized users. Network communications between secure nodes in a

secure domain are restricted to only other secure nodes in that domain. Secure nodes limit access to authorized users as specified by the local authentication and access control policies.

ATNA profile addresses three areas of security in an enterprise:

- **User Authentication:** The ATNA profile requires only local user authentication, thereby allowing the each of the secured nodes (e.g. TRANSCEND's eCHR, institute's EPIC system) to use the access control technology of its choice to authenticate users.

- **Connection Authentication:** The ATNA profile requires the use of bi-directional certificate-based node authentication for connections to and from each node. The DICOM, HL7, and HTML protocols all have certificate-based authentication mechanisms defined. These authenticate the nodes, rather than the user. This profile relies on the WS-I Basic Security Profile 1.1 for Web Service security.

- **Audit Trail:** The ATNA profile ensures user accountability via audit trails. It allows administrators to:
    - Assess compliance with a secure domain's policies
    - Detect instances of non-compliant behavior
    - Facilitate the detection of improper creation, access, modification and deletion of Protected Health Information (PHI)

ATNA not only deals with creation and storage of Audit records but also transmission of Audit records across the enterprise using the Syslog Protocol. Since the integration architecture has no requirement of making audit trail available across the enterprise, the caEHR solution will not implement the Audit Trail sharing portion of the ATNA profile.

Connection Authentication implies establishing mutual authentication between various components of the integration architecture. This implies securing the connection between the EHR system and the semantic adapter using HTTPS as well as between the semantic adapter and the integration platform. On the other end, the outbound channels from the integration platform will be secured using HTTPS as well. Securing these interfaces will require use of digital certificates.

4.4.2.2 Consistent Time (CT) Profile

The **Consistent Time Integration** Profile (CT) provides a means to ensure that the system clocks and time stamps of the many computers in a network are well synchronized. This profile specifies synchronization with a median error less than 1 second. This is sufficient for most purposes. Various infrastructure, security, and other profiles require use of a consistent time

base on multiple computers, to synchronize logs, authenticate users or nodes (using their digital certificates), digitally sign documents, etc. The Consistent Time profile requires the use of the Network Time Protocol (NTP) which is supported by most of the operating systems.

Since the integration architecture relies on use of digital certificates to mutually authenticate the nodes, it will require that all the systems that participate in the integration be synched with a common time service such as time.nist.gov. A detailed list of servers can be obtained from http://tf.nist.gov/tf-cgi/servers.cgi.

4.4.2.3 Other Profiles

IHE provides two additional profiles which deal with User Authentication and Authorization:

- **Enterprise User Authentication (EUA)** Integration Profile defines a means to establish one name per user that can then be used on all of the devices and software that participate in this integration profile.
- **Cross-Enterprise User Assertion Profile (XUA)** - provides a means to communicate claims about the identity of an authenticated principal (user, application, system...) in transactions that cross enterprise boundaries.

However, since the end user authentication is out of the scope, these profiles are no relevance to the integration architecture.

4.4.3 Digital Certificates and Level of Assurance (LOA)

NIST's E-Authentication Guidelines on Level of Assurance (LOA) for a user's (or a system's) identity provides guidelines on the minimal LOA that needs to be established depending upon the type of data access or transmitted. Since we are dealing with PHI/PII as part of this integration, the minimal level of assurance should be LOA2 or 3. The use of digital certificates satisfies the two-factor authentication requirement of the LOA3 guideline.

Even though the integration architecture does not deal with end users' credentials, it still requires obtaining credentials for nodes which participate in the integration. This will require obtaining digital certificates for these nodes.

NCI's caGrid runs its own Certificate Authority (CA) called Dorian which is capable of issuing long term (1 year) certificates. However this CA is not certified and operates at a lower level of assurance. This means that anyone can obtain certificates from this CA without any elaborate vetting process and participate in the architecture. This is not acceptable for the proposed caEHR solution because the architecture deals with transmission of PHI.

The other options are to obtain certificates from third party entities such as Verisign, ETrust etc. They have a defined identity vetting process which can map to a higher LOA such as LOA-3. Alternatively, the system administrators can choose to generate their own certificates on the servers. In either case, trust needs to be established between the participating nodes. This is achieved by adding the public certificate of the other party (or if the certificates are obtained from a CA, then CA's public certificate) in the node's trust store.

5 Alternatives

This section explores some of the alternative architectural approaches that could have been adopted and discusses the reasons the architecture team chose to make alternate choices.

5.1 Clinical Services

The architecture team seriously considered the prior design's plan to make use of clinical domain-specific services within the *Integration Platform*. The initial vision had been to use these domain-specific services for several functions, including more fine-grained extraction and capture of data, integration of data from multiple sources and exposure of data to EHR applications at a services level. However, upon review of the project objectives for both short term implementation and longer term extensibility and given the scope of the project, the complexity of such an architecture was not justified.

5.1.1 Project Timeline and dependencies

We believe the use of the previously designed clinical services would introduce a significant risk to the project due to the aggressive timeline and project dependencies with an external customer. The project cycle will require specification development, specification QA, development, software QA and Transcend deployment in the next five and half months. Given the external dependency of deployment in the Transcend environment we anticipate that the project will have to factor time for unanticipated issues at deployment/integration test. In our experience integrating systems both at caBIG and Transcend, integration testing and deployment take up a significant number of cycles. Factoring these risks, we believe the project has less than five months to complete all the project activities.

The current project teams are not staffed to implement clinical services. The project is operating with a fraction of the resources available to the original team. Using historical metrics for service specification and development on caCIS and other CBIIT projects, the team does not anticipate any scenario in which these specifications can be completed in time.

5.1.2 Unbounded Scope

In the absence of direct interaction with external stakeholders the models and specifications developed by the team will capture our "view of the world". Modeling any specific domain requires the input of many subject matter experts, review of multiple domains and an active requirements gathering cycle. It is not clear when the model/specifications would be complete

and the success criteria for this effort are very ambiguous. Prior efforts at creating specifications ran into similar issues, and we anticipate that the current team will run into scoping issues as well. In the absence of a formal requirements gathering cycle, the models will risk reflecting the Transcend requirements. This will limit reusability of clinical services interfaces outside Transcend.

5.1.3 Fine-grained data extraction and retrieval

In terms of fine-grained data extraction and capture, EHR applications have a variety of internal data architectures with significant discrepancy in terms of how data is stored. For example, are drug and non-drug allergies kept separate? Are they mixed together with adverse reactions? Are they all lumped together with health conditions? Requiring EHR vendors to export and potentially import data with a granularity that did not match their expected internal granularity would add significant complexity to the import/export process. A data granularity based on a "full patient record" is a level supported by all EHR applications.

The proposed approach addresses this constraint by supporting interfaces at the right level of granularity, using accepted standards. Any changes to the current model will require active participation and engagement of the EHR vendor community.

5.1.4 Multiple source data integration

A second use-case for the use of domain-specific clinical services was the potential need to integrate data from a number of sources before generating an output artifact such as a CCD, CDA or v2 message. Such an architecture would position the *Integration Platform* as a central mediator between a number of EHR systems. While setting up such an architecture is more complex in the negotiations that must occur between the communicating partners, it is foreseeable that such functionality would be useful.

The justification for introducing clinical domain-specific services in such an environment is that the integration requirements for data would vary by clinical domain. For example, the rules for integrating multiple-source data related to patient allergies would differ from that for drugs, for labs or surgical history. Encapsulating these rules in clinical-domain specific services would result in a cleaner, more extensible architecture.

While this functionality may be desirable eventually, the scope of the current project only allows for single-sourced data rendering. Furthermore, there is only one EHR application system in use by the project. Any development of domain-specific clinical services providing a data integration capability would therefore need to be developed on a purely theoretical basis without any real-world samples to verify requirements or to test the resulting solution. The probability of these services working in a real-world environment without significant redevelopment is low.

Therefore, it makes sense to defer the development of such services until such time as requirements can be gathered and testing performed in a more realistic multi-EHR environment. The proposed architecture would allow for clinical domain services to be integrated in a future release to provide any needed multi-source integration.

5.1.5 Direct EHR access to discrete clinical services

One of the visions of the original caEHR project was that EHR vendors would modify their applications to communicate directly with discrete, clinical domain-specific, RIM-based service interfaces. This would "move the ball forward" in terms of EHR capabilities and provide a platform for improved integration, decision support and other functionality. The underlying belief seemed to be one of "build it and they will come". If appropriate services interfaces were designed and published, it would be relatively easy to convince EHR vendors to modify their applications to take advantage of them.

EHR vendors have historically been reluctant to embrace tight integration with other systems, to support real-time data exchange, and to support RIM-based data structures. Where such interfaces are being supported, it tends to be around legacy protocols such as v2 or flexible document-centric specifications like CDA. It also helps to have the backing of legislative or regulatory requirements, significant funding or both. Most importantly, successful introduction of a new interfaces requires direct engagement with the EHR vendor community to come to agreement on requirements to ensure that the interface will meet the needs of and fall within the capabilities of the majority of EHR application solutions.

The scope of the current project allows no such engagement with the EHR vendor community. Existing services-based standard interfaces such as XDS already exist and are enjoying widespread market penetration. To maximize chances of successful adoption and realize NCI's objectives of improved data exchange, leveraging these existing standards seems a wiser course.

5.2 Canonical format

There are two obvious alternatives to the use of a CDA-based canonical format: Use of the RMIM models designed for the clinical domain services as part of the previous project or using the TRIM format exported by the Tolven application

5.2.1 Clinical Service Model-based

There were several disadvantages to using the models developed as part of the Clinical Service definitions:

- The data models were constructed in a manner that was independent of both the target formats and the capabilities of the source EHR applications, instead focused on idealized "best practices". This created significant risk of mapping issues on both the source EHR to canonical and the canonical to publication format sides.
- The clinical service models did not include a focus on human readability. CDA and CCD are the most critical of the exchange formats and both have mandatory requirements for human readability
- The disparate formats didn't have a unifying architecture to bring them all together
- EHR Vendors, who will need to perform the mapping from their internal data structures to the canonical format, have more familiarity with CDA than they do with arbitrary RIM-based models.
- The primary objective for this project is to ensure Tolven data can be successfully communicated. We do not have the scope or expertise to determine or validate "ideal" data models
- Using an arbitrary RIM format would increase the complexity of transformation for both the CCD and CDA publication formats and would make no significant difference to the other formats.

5.2.2 TRIM based

Using the TRIM format as the canonical source would minimize project effort, but would significantly limit the re-usability of the solution with other non-Tolven EHR applications. In addition, the semantic clarity that should ideally be a benefit of the canonical format would not be achieved due to significant modeling issues in the Tolven TRIM data representations.

5.3 Publication Assembly

The previous architecture had called for the publication formats to be assembled by making calls to the various clinical services. This approach was rejected as part of the new project for a few reasons:

- The majority of the clinical services were in early stages of design and development and it would be challenging to get them all complete within the time available
- The services approach added considerable complexity to the process of what is really just a simple transformation step
- The use of the services for other parts of the architecture had already been excluded.

5.4 Data exchange and security

Instead of using the proposed standards-based exchange and security approach of XDS, NAV and ATNA, it would be possible to use a variety of custom exchange and security protocols.

Because the stability, reliability and extendability of the open-source IHE protocol implementations have not yet been fully evaluated, it may be necessary to pursue a non-standard interface approach due to the tight time and resource constraints on the project.

If an alternative exchange mechanism were required, it would make sense to leverage the existing integration engine used within TRANSCEND's ESB to manage the exchange process. As described in section 6.1.1, Mirth Connect provides support for a number of exchange and security protocols. A similar solution to XDS of TLS with dual authentication using SOAP over HTTP could also be implemented with the Mirth Connect engine.

6 External Components

This section describes the open-source components presently being considered for inclusion as part of the solution. A final decision on components will be made after further evaluation and in consultation with the development team.

6.1 Transformation solutions

6.1.1 Mirth Connect

Mirth Connect is an open source standards based healthcare interface engine. Mirth Connect facilitates routing, filtering and transformation of messages between health information systems. Mirth Connect provides an easy to use graphical administrative user interface that can be used to define routing, filtering and transformation of messages. Mirth Connect integration interfaces and transformation capabilities can potentially be used for integration with Transcend and PCO Tolven instances.

Mirth Connect supports a variety of protocols and integration interfaces including: LLP (Lower Layered Protocol), SOAP, JMS, FTP, File, HTTP, TCP and Database. It provides transformation support for messages based on a number standards including: HL7 v2, HL7 v3, XML, DICOM (Digital Imaging and Communication in Medicine), NCPDP (National Council for Prescription Drug Programs), EDI (Electronic Data Interchange), X12 and Delimited Text

Within the scope of this project, Mirth Connect is the probable candidate for authoring mappings and executing the transformation of instances between Canonical and native formats and between Canonical and exchange formats. The transport capabilities are a fall-back consideration in the unlikely event of issues with the planned IHE XDS, NAV and ATNA strategy.

6.2 XDS solutions

The IHE profiles required for the external interface of the *Integration Platform Component* are:

- Enterprise Document Retrieval and Storage (XDS.d),
- Notification of Document Availability (NAV), and
- Audit Trail and Node Authentication (ATNA)

As well, support for future Patient Registry support (PIX/PDQ) would be useful, though not a strict requirement.

An evaluation for available XDS open source implementations was performed based on the above outlined criteria. The evaluation criteria added Non-Functional requirements for software maturity, active open source community adoption; TRANSCEND platform support and IHE Connectathon verification for XDS implementation requirements. The identified candidates follow:

6.2.1 OpenExchange

Provided by Misys Open Source Solutions (MOSS) initiative. MOSS has implemented the complete required TRANSCEND evaluation criteria as described. An active forum fulfills the Non-Functional requirements with access to technical documentation.

6.2.2 Open eHealth Integration Platform (IPF)

Provided by InterComponentWare AG, ICW. The IPF initiative extends the open source Apache Camel open source mediation and open source engine. The Open eHealth Integration Platform fulfills the TRANSCEND evaluation as described, however provides additional support for Document building and mediation.

6.3 Vocabulary Mapping

As part of the mapping exercise from native formats to the *Canonical Data Format* and from the canonical format to exchange formats, it will sometimes be necessary to perform mapping of vocabulary values. Re-usable mappings and an easy-to-use maintenance interface will be essential here as vocabularies often change much more quickly the data structures that use them.

Ideally, the mapping tools for structures will be usable for terminology as well, however, a terminology server such as LexEVS or even a simple spreadsheet format are alternatives.

7 Reusability

One of the primary objectives of this project is to ensure that the solution will be reusable with other applications and in other environments. This section talks about some of the reusability considerations with the proposed solution.

7.1 Semantic Adaptor

The *Semantic Adaptor* qualifies as a "partially" re-usable component. The underlying technology of Mirth Connect will work with any application capable of exporting (and possibly importing) data in XML, HL7 v2, fixed length, or character-delimited formats. This should cover the vast majority if not all clinical systems. The mapping technology is robust and, provided the application being integrated has relevant data, it should be possible to convert that data to and from the canonical format in most circumstances. However, because each integrated application will contain different data elements with differing granularities and drawn from different vocabularies, a significant amount of effort will be required to develop the necessary mappings for each system. This effort will be minimized as much as possible through the selection of appropriate mapping tools. However, the tasking of mapping data elements is unavoidably time-consuming and will thus have some degree of impact on uptake.

7.2 Canonical Data Format

The *Canonical Data Format* will have moderate re-usability. Its basis on CCD and other broadly used CDA templates will enhance the degree of re-usability. However, the initial version is being created based solely on the data extractable from the TRANSCEND and PCO Tolven instances. As additional clinical systems are brought into the interoperability environment, the canonical format will need to expand to accommodate additional data elements. However, significant refactoring is unlikely due to the reliance on existing public data standards such as CCD and other standard templates. As well, the section-based approach of CDA will make expansion to include additional types of data relatively transparent for previous implementations.

7.3 Integration Platform

The *Integration Platform* will be highly reusable. The platform is isolated from the variations of participating clinical applications by the *Semantic Adaptor*. The use of an integration engine to manage the transformation from Canonical to published exchange format will make the introduction of additional exchange formats quite straight-forward. The primary change in adopting the Integration Platform to accommodate new clinical applications will be modifying existing transforms between the *Canonical Data Format* and the various publication formats to

account for additional data elements introduced to the canonical format as additional applications are included.

7.4 Transport and Security Layer

The XDS and NAV interfaces will be 100% re-usable in future implementations. Many clinical applications support these interfaces out of the box. The open source solution will allow easy introduction of query, patient lookup and patient identity resolution services when and if they are needed as part of NCI integration requirements.

7.5 Exchange Formats

The reusability of the exchange formats will vary depending on the format. The portion of the CCD syntax that is based on CCD and other common industry templates will be highly re-usable due to widespread use. The remainder of the CCD will be reusable primarily at the human-readable level, though it is possible that if NCI introduces one or two new CDA section types that meet a clear industry need those could also see broad adoption. The HL7 v2 syntax may have moderate re-use depending on how well the HL7 profiles created by the design team align with interfaces already present in existing systems. The XML ITS syntax is extremely unlikely to see re-use due to the lack of uptake of the syntax by industry and due to the custom nature of the content. Reusability of the PCO CCD syntax will depend on industry acceptance of the PCO model developed as part of the project.

8 Assumptions

1. Any patient-specific consent-based constraints on what data may be shared with other systems or imported from other systems will be enforced by the EHR systems being integrated (in this case the Tolven instances). The Semantic Adaptor and Integration Platform will have no responsibility in this area. Any changes needed to the Tolven instance to support consent policies fall outside the scope of this project.
2. The creation of the business associate agreements required by legislation (e.g. HIPPA) to allow for the exchange and storage of personally-identifiable healthcare information is outside the scope of this project.
3. Automated configuration of communication paths between EHR applications via publish/subscribe or similar mechanism is out of scope.
4. HL7 v2 exchanges will be treated as "documents" that just happen to be expressed in v2 syntax. i.e. There will be no HL7 workflow functionality required in terms of real-time transmission in response to trigger events or to deal with acknowledgement messages.
5. While audit log sharing is supported by the underlying technology, audit logs will be stored locally by each Integration Platform instance and will not be shared.
6. User authentication and permissions, including permissions related to the sharing and/or importing of data, will remain the responsibility of the communicating applications and will not be part of the functionality of the Semantic Adaptor or Integration Platform components.
7. The TRANSCEND Tolven instance will be modified as necessary to allow the user to initiate the export of data and invoke the service call on the *Semantic Adaptor*. The making of these modifications is out of scope for the project team.
8. All exports will be treated as "snapshot", conveying the complete set of data needed to populate the Canonical format. There will be no support for "partial" exports.
9. The XDS persistent store and the ATNA log files will be stored within the linked clinical application's "trusted" data area. Similarly, communication between the clinical applications, the

9 Risks

1. The initial set of data elements expressed in the *Canonical Data Format* and the transforms mapping to and from the canonical format will be driven by the data available from TRANSCEND. Revision will be required for future integrated applications and could be substantial if TRANSCEND is not a representative application in terms of data requirements and capabilities.
2. There is only one client and one target system to test mapping and data exchange. This is not very rigorous for a system that is intended to support a wide variety of source and client

applications. Integration issues are therefore likely to surface when introducing new clinical applications.

3. The proposed solution is based on our current understanding of requirements which may be dated or incomplete.
4. The solution relies on the use of several external off-the-shelf components. This may complicate maintenance due to version management issues as new releases of the off-the-shelf components are produced.
5. TRANSCEND may not be able to make the required changes to integrate with the proposed solution.

10 Appendix A – Service interfaces

The following is a list of the expected service interfaces to be exposed by the components developed:

10.1 Semantic Adaptor

ShareClinicalInformation: Passes a set of clinical information, submitting user, and possibly target applications and desired format.

LocalizeCanonicalInformation: Takes canonical information from the *Integration Platform* and converts it to the localized format needed by a clinical application.

10.2 Integration Platform

ShareCanonicalInformation: Takes canonical information from the *Semantic Adaptor* and persists it into the XDS repository after validating the content. If routing information was provided, will also cause the invocation of the NAV Notification of Document Availability for any intended recipients.

GetDocument: XDS interface that allows retrieval of a specifically identified document.

NotificationOfAvailability: Receives a notification from another *Integration Platform* that a document in a particular format is available for retrieval and conversion. Invocation of this service will trigger execution of Get Document on the remote system and then Localize Canonical Information on the Semantic Adapter with the retrieved document.

Note: Additional services such as PIX, PDQ and Query may be added in future releases

11 References

Reference	Link
XDS	http://wiki.ihe.net/index.php?title=Cross-Enterprise_Document_Sharing
NAV	http://wiki.ihe.net/index.php?title=Notification_of_Document_Availability
ATNA	http://wiki.ihe.net/index.php?title=Audit_Trail_and_Node_Authentication
Mirth Connect	http://www.mirthcorp.com/community/mirth-connect
OpenExchange	https://www.projects.openhealthtools.org/sf/projects/openexchange
Open eHealth IPF	http://repo.openehealth.org/confluence/display/ipf2/Home