Integrating the Healthcare Enterprise

IHE IT Infrastructure
White Paper

A Service-Oriented Architecture (SOA) View of IHE Profiles

Public Comment

Date: September 28, 2009

Author: Joshua Painter (Intel Corporation), Alean Kirnak (Software Partners LLC), John Moehrke (GE Healthcare)

Email: ihe@himss.org

CONTENTS

1 Introduction 2
2 Comparing the Approaches 3
  2.1 Background on IHE 3
    2.1.1 Mission 3
    2.1.2 Approach 3
  2.2 Background on SOA 4
    2.2.1 Definition of SOA 4
    2.2.2 Services 5
  2.3 Mapping Between SOA and IHE Concepts 6
    2.3.1 Mapping of IHE and SOA Concepts 7
    2.3.2 Mapping of SOA and IHE Concepts 10
  2.4 Introduction to SOA Design Practices 12
    2.4.1 Meet in the Middle 12
    2.4.2 An Example Service Model 13
3 Service Modeling By Example 15
  3.1 Document Sharing Example 15
    3.1.1 IHE “Wrapper” Service Design 19
    3.1.2 Service Composition 21
    3.1.3 Example of a Deployment View 23
  3.2 Example Service Definition for Identity Resolution 24
    3.2.1 Abstract Service Definition 25
    3.2.2 Operations 26
    3.2.3 Concrete Service Definitions 28
    3.2.4 Other aspects of service definitions 30
  3.3 Value 30
4 Further Issues for Exploration 32
5 Conclusion 33

1 Introduction

This white paper is targeted specifically at SOA practitioners who want to take advantage of the IHE profiles in their implementations. It expands SOA in the healthcare domain with IHE's concrete approach to interoperability. It establishes a vernacular for comparing SOA and IHE approaches, and illustrates the use of IHE profiles in a hypothetical SOA design.

The focus of the paper is to: (1) illustrate how IHE profiles can be leveraged in a SOA design; and (2) begin to explore the issues, challenges and benefits of a closer alignment of the IHE Technical Framework with SOA approaches in the future.

This paper examines the issues from both an IHE and SOA perspective, to relate the different nomenclatures, goals and approach to gain a perspective that is useful to both SOA and IHE implementers alike. It makes frequent use of realistic but hypothetical examples, which are intended to provide concrete, practical use cases for IHE-based interoperability in an SOA.

It assumes the reader has a fair grasp of the fundamental principles of service-orientation and SOA design. Despite this, there are a series of broader topics addressed in the paper that may be of interest to readers looking to further their understanding of best practice for designing cost-effective, flexible and interoperable healthcare architectures.

2 Comparing the Approaches

2.1 Background on IHE

2.1.1 Mission

IHE is an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information. IHE promotes the coordinated use of established standards such as HL7, DICOM, OASIS, and W3C to address specific clinical need in support of optimal patient care. Systems developed in accordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively.

2.1.2 Approach

The IHE process is defined in ISO 28380-1. It starts with proposals that describe an existing interoperability problem. These proposals are discussed and prioritized by a planning committee. Given that IHE works on a strict time based deadline, only a defined amount of work can be accomplished each year, thus focusing the efforts on the most urgent issues. The chosen work items are developed in open, consensus based meetings utilizing existing standards (typically from HL7, DICOM, OASIS, W3C, etc) and vocabulary, balloted in a public comment, and published as ‘trial implementation’.

These work items are organized as a set of domain specific “Profiles”, which are detailed specifications for communication among systems to address key clinical use cases, all based on established standards. IHE Profiles address critical interoperability issues related to information access for care providers and patients, clinical workflow, security, administration and information infrastructure. Each profile defines the actors, transactions and information content required to address the clinical use case by referencing appropriate standards.

IHE Profiles are documented in the IHE Technical Frameworks — detailed technical documents that serve as implementation guides. For each domain, the Technical Frameworks identify a subset of the healthcare enterprise, called IHE Actors, and specify their interactions in terms of a set of coordinated, standards-based transactions. They describe this body of transactions in progressively greater depth. Volume I provides a high-level view of IHE functionality, showing the transactions organized into functional units (Integration Profiles) that highlight their capacity to address specific clinical needs. Volume II provides detailed technical descriptions of each IHE transaction.

A profile is not moved to a ‘final text’ unless three independent vendors/implementers prove interoperability at an international IHE Connectathon. In this way only implementable profiles that are clearly documented get final recognition.

2.2 Background on SOA

This section provides some background on SOA concepts. Where applicable, it will provide base definitions of concepts and design principles to which it refers, since discussion of SOA often suffers from wide-spread ambiguity in the technology industry.

2.2.1 Definition of SOA

SOA definitions vary widely. A sampling from various standards bodies is given in Table 2.2.1-1: SOA Definitions.

Table 2.2.1-1: SOA Definitions

Organization	SOA Definition
OASIS	Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.
W3C	An SOA is "a set of components which can be invoked, and whose interface descriptions can be published and discovered."
Open Group	Service-Oriented Architecture (SOA) is an architectural style that supports service orientation. Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services.
CMU/SEI	We define SOA as an architectural style where a system consists of service users and service providers.
OMG	Service Oriented Architecture is an architectural style for a community of providers and consumers of services to achieve mutual value, that: Allows participants in the communities to work together with minimal co-dependence or technology dependence Specifies the contracts to which organizations, people and technologies must adhere in order to participate in the community Provides for business value and business processes to be realized by the community Allows for a variety of technology to be used to facilitate interactions within the community

Common elements of the sampled definitions are:

- SOA is an architectural style
- SOA defines services
- Services are invoked, requiring a provider of the service and a consumer of the service

Whether SOA includes a concept of interoperability is subject to debate. Generally, SOA is thought not to necessarily imply interoperability[1].

[1] Practical Guide to SOA in Healthcare. Healthcare Services Specification Project, Health Level 7/Object Management Group, 2008.

2.2.2 Services

The most fundamental building block of an SOA is the *service*. A service implements one or more related capabilities. It is exposed by a service provider, to be accessed by a service consumer. The roles of *service consumer* and *service provider* are introduced in 2.3.2 Mapping of SOA and IHE Concepts.

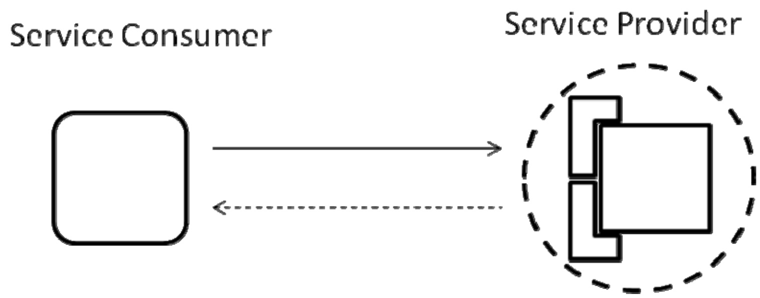


Figure 2.2.2-1 – Service Consumer to Service Provider Interaction

Two elements – the *service definition* and the *service implementation* - comprise a service, as represented in Figure 2.2.2-1.

2.2.2.1 Service Definition

The *service definition* contains the terms for information exchange, providing the service's technical constraints and requirements as well as any semantic information needed to consume the service. It is comprised of two parts: (1) an *abstract portion*; and (2) a *concrete portion*. The abstract portion describes the functionality of the service. It includes a series of technology-independent elements: the interface, operations, operation semantics, and data structure definitions. The concrete portion describes how to access the service. It effectively designates how the abstract interface connects to technology that implements the service. Note that there can be more than one concrete portion corresponding to a single abstract portion. This ensures that the means used to access the service – such as web services, messaging or direct invocation – can be independent of the abstract service definition.

2.2.2.2 Service Implementation

The Service Implementation is the core logic in support of the service definition. It is essentially the code behind the service, often written in an application language like Java, .Net or C++.

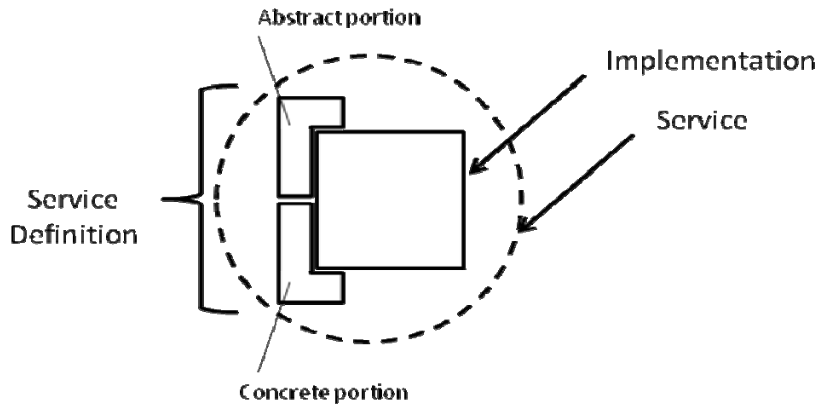


Figure 2.2.2.2-1: Diagram of a Service

2.3 Mapping Between SOA and IHE Concepts

The comparisons between IHE concepts and SOA concepts are broad ones; strict mappings do not exist. Nevertheless, providing a sense of comparison between the two provides a frame of reference that helps tremendously in evaluating how the interoperability offered by IHE can be leveraged in SOA.

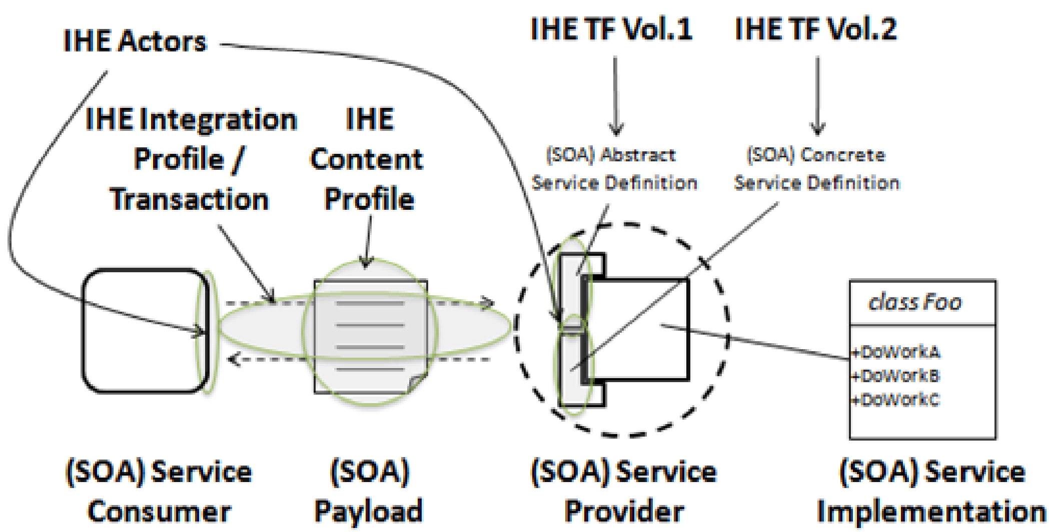


Figure 2.3-1: IHE and SOA

Figure**Error! Reference source not found.** 2.3-1 builds on the previous diagram by illustrating a simple request to our service by a service consumer, and by adding references to key IHE concepts introduced in Section 2.1.2. When a service is invoked, IHE *actors* may be classified as either the *service consumers* or *service provider*. IHE *transactions* are initiated by service consumer actors, and are often defined and named from the perspective of the service consumer e.g. *Patient Identity Feed, Patient Demographics Query*. The IHE *Content Profile* specifies the information content that may be exchanged between consumer and provider, which corresponds to the service *payload*. Note that the SOA service payload is not restricted to IHE defined Content Profiles.

The SOA *Service Definition* is defined by the IHE *Technical Framework (TF)* specifications. The *abstract portion* is largely addressed by the detail contained in *TF Volume I*, while the concrete portion is addressed by specification found in *TF Volume II*. Lastly, the IHE *Integration Profil*e it not explicitly referenced in the figure below, because a typical profile encompasses multiple transactions between actors.

2.3.1 Mapping of IHE and SOA Concepts

The following table starts with the IHE concept and provides a mapping to the closest SOA concept or concepts. These relationships are never a perfect match.

IHE Concept	Formal IHE Definition*	Corresponding SOA Concept
Actor	Essential component of an IHE Integration Profile that is an abstraction of the endpoint responsible for the initiation or response to a Transaction. Systems implement one or more Actors (Grouped) as declared in the systems Integration Statement. 1) A functional component of a communicating healthcare IT system and device. 2) Actors are information systems or components of information systems that produce, manage, or act on information associated with operational activities in the enterprise.	Related to SOA concept of *role,* e.g. service provider, service consumer
Integration Profile	An IHE Integration Profile specifies a coordinated set of interactions exchanged between the functional components of communicating healthcare IT systems and devices. These functional components are called IHE actors. An IHE Integration Profile specifies their interactions in terms of a set of coordinated, standards-based transactions. An IHE Integration Profile is a reusable specification that defines the Interoperability solution to a healthcare workflow that requires two or more systems to work together.	An integration profile might correspond to a service definition, collaboration or a capability, depending upon the scope of the integration profile.
Transaction	Essential component of an IHE Integration Profile that is the pre-defined interaction between Actors. IHE Transaction defines the network semantics, trigger events, and expected actions 1) Transactions are interactions between actors that communicate the required information through standards-based messages. 2) Transactions are interactions between actors that transfer the required information through standards-based messages.	Most similar to SOA concept of a concrete service-definition. Sometimes similar to a capability or operation.

IHE Concept	Formal IHE Definition*	Corresponding SOA Concept
Connectathon	A testing event to which developers have registered their implementations for supervised interoperability testing with other implementations. Each participating system is tested for each registered combination of IHE Actor and IHE Integration or Content Profile.	Loosely related to conformance; SOA specifies no specific testing and validation mechanism.
Technical Framework	A collection of Profile Specifications related to an IHE Domain and its specific clinical or technological focus. Profiles within a Technical Framework and across Technical Frameworks may be combined.	None. This isn't a necessary concept in SOA as SOA is an open method and not an organization publishing their specifications.
Message Semantics	Encoding rules for the message as communicated.	Related to the detail found in a concrete service definition.
Option	Named variance in the Integration Profile.	Related to the SOA concept of a profile when used to identify different grouping of capabilities.
Use Case	The defined healthcare workflow that outlines the interoperability problem that is the focus of an Integration Profile A textual and graphical depiction of the actors and operations that addresses information exchange in the context of a set of specific tasks performed by different systems or devices.	Related to SOA use case. Use cases are often included in the SOA capabilities documents.
Integration Statement	IHE Integration Statements are documents prepared and published by vendors to describe the conformance of their products with the IHE Technical Framework. They identify the specific IHE capabilities a given product supports in terms of IHE actors and integration profiles	Related to the SOA idea of conformance.
Process Flow Diagram	A graphical illustration of the flow of processes and interactions among the actors involved in a particular example.	Might be included in documentation of capabilities. The flow of processes between actors is somewhat implicit in SOA because all actors are either service consumers or service providers. Generally IHE profiles and transactions are oriented from the point of view of service consumer, for example "Patient Demographics Query". This is logical since by definition, the service consumer initiates the interaction.

IHE Concept	Formal IHE Definition*	Corresponding SOA Concept
Content Profile	An IHE Content Profile specifies a coordinated set of standards-based information content exchanged between the functional components of communicating healthcare IT systems and devices. An IHE Content Profile specifies a specific element of content (e.g. a document) that may be conveyed through the transactions of one or more associated Integration Profile(s).	Most similar to SOA concept of payload, where the payload is not defined in the service because the service is agnostic to the payload content.
Grouping	Set of related and interdependent profiles driven from requirements that an actor supporting one profile be grouped with one or more actors supporting other integration profiles. Grouping includes specific rules regarding mutual behavior, such as shared data requirements.	Related to SOA concept of service composition.

2.3.2 Mapping of SOA and IHE Concepts

The following table starts with the SOA concept and provides a mapping to the closest IHE concept or concepts. These relationships are never a perfect match.

SOA Concept	SOA Definition	Corresponding IHE Concept
Service Provider [a]	An entity (person or organization) that offers the use of capabilities by means of a service.	Service Provider is similar to IHE Actors that are recipients of transactions.
Service Consumer [a]	An entity which seeks to satisfy a particular need through the use capabilities offered by means of a service	Service Consumer is similar to IHE Actors that are initiators of transactions.
Capability[2] [a]	A real-world effect that a service provider is able to provide to a service consumer. Embodies the the functional requirements or behavior of a service, described in a verbal or high level way.	TF Vol. 1 Transaction diagram and Transaction descriptions
Service Definition - abstract portion[3] [b]	The service definition contains metadata which describes the terms for information exchange with the service, describing the service's technical constraints and requirements as well as any semantic information needed to consume the service. The abstract portion describes the functionality of the service using a series of technology-independent elements, including interface, operations and operation semantics, and data structures.	Generally speaking, the content in Technical Framework Volume I most closely matches the abstract portion of the service definition.
Service Definition - concrete portion[4] [b]	The service definition contains metadata which describes the terms for information exchange with the service, describing the service's technical constraints and requirements as well as any semantic information needed to consume the service. The concrete portion of the service definition describes how to access the service. It effectively designates how the abstract interface connects to technology that implements it (service implementation)	Generally speaking, the content in Technical Framework Volume II most closely matches the concrete portion of the service definition.
Service Implementation [b]	The core logic in support of the Service Definition. A service's implementation is basically the code behind the service, often written in an application language like Java, .Net or C++.	Maps to vendor products which implement IHE profiles.

[2] Also known as "Service Functional Model".

[3] Also known as "Platform Independent Model".

[4] Also known as "Platform Specific Model".

Service [a]	The means by which the needs of a consumer are brought together with the capabilities of a provider. Represented by the Service Definition and Service Implementation taken together.	IHE Technical Framework (Volumes I and II) combined with vendor products
Binding [c]	In the context of WSDL, a binding is the association of protocol and message format information to a service operation	The association between the use case and the transactions is found in Volume I; the expected actions are found in Volume II Transactions
Service Operation [c]	A service function or method.	Loosely corresponds to transaction or message. Sometimes service operations correspond directly to IHE transactions or message, but not always.
Payload[5] [c]	Data structures shared between a service consumer and service provider. Also called a Message in SOA literature. The payload is defined in the Service Definition: the "logical" payload is described in the abstract portion of the Service Definition, while the "concrete" part describes how the payload format is mapped into concrete on-the-wire data formats	Relates to an IHE profile section called Message Semantics which may or may not reference a Content Profile.
Composition [a]	Ability of a service to be composed with another service to create a new service. Composition is a core principle of SOA, which permits business logic to be represented at different levels of granularity, and promotes reuse and abstraction (also core principles of SOA)	The way IHE uses UML in Volume I and Volume II to show interactions or implement a workflow is close to the idea of composition. IHE does not have a concept as complete as composition, but some of the concepts come through in the requirements IHE uses around Grouping. It is also related to Dependency, which is a term used in IHE to point to Grouping. For example, a Document Consumer Actor may also have to play a security role.
Orchestration [d]	A late-binding, stringing together of services to implement a workflow. Rules that govern behavioral characteristics of how a group of services interact, usually in the context of a business process	IHE does not have a concept that corresponds well to orchestration. However, sometimes the stringing together of profiles is either implied or suggested in a white paper, appendix, or informal notes.
Taxonomy[6] [d]	A classification of services for the purpose of identifying reuse of services. An SOA solution is often analyzed and depicted as layers of services corresponding to these classifications.	The first level of taxonomy is in the breakup of profiles into IHE domains. IT Infrastructure and Patient Care Coordination domains were explicitly created to foster reuse of profiles. Within domains, IHE has begun developing lower level taxonomy.
Conformance [d]	Ability tof a service implementation to fulfill the requirements of a Service Definition.	Testability of vendor implementations at Connectathon.

[a] OASIS Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard, Oct. 12, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html

[b] Erl,T.; SOA: Principles of Service Design. Boston, MA: Prentice Hall, 2008

[c] Erl, T: A W3C Web Services Glossary. http://www.ws-standards.com/glossary.asp

[d] Definition generalized from multiple sources, as no single authoritative source determined.

[5] Also known as "Message Body"

[6] Also known as "Service Model".

2.4 Introduction to SOA Design Practices

2.4.1 Meet in the Middle

Frequently, SOA design is a combination of bottom-up and top-down approaches. A “bottom-up” service modeling approach is based upon “wrapping” existing application functionality to create one or more services. By contrast, a “top-down” approach creates “new” services through requirements analysis, use case definition and business process modeling. Often, a combination of both approaches is most suitable for the design of the SOA. This is referred to as the “meet-in-the-middle” approach, where the bottom-up view derives a set of services that expose existing application infrastructure, while the top-down view specifies new services to meet desired new capabilities. It’s compelling in that it addresses the shortcomings of these other modeling approach: the bottom-up approach tends to overlook behavior and be too instance focused, while the top-down can be a bit like an ivory tower, disconnected from real implementation.

As mentioned in the introduction, it is possible to simply re-factor an IHE Profile into an SOA service. In fact, that can be a sensible starting point for leveraging IHE profiles in an SOA: treating those profiles as an interoperable “black box” and deriving a set of services using a “bottom-up” approach. Although this provides limited value on its own, it provides a foundation upon which higher-level, more purposeful services can be assembled to gain the business benefits associated with SOA whilst reaping the interoperability promise of IHE.

The development of our examples will use meet-in-the-middle, as illustrated in Figure 2.4.1-1.

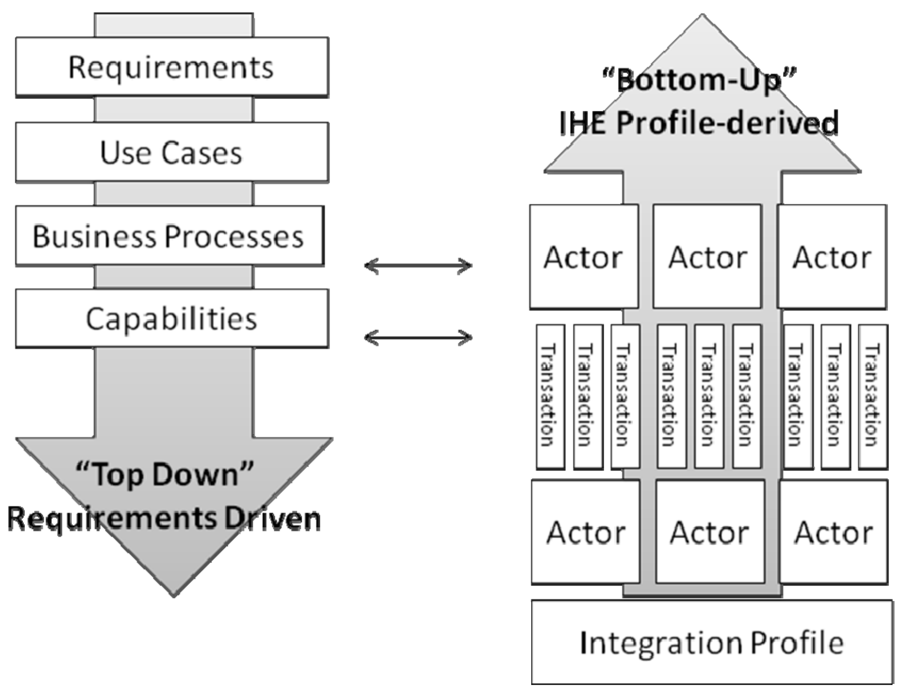


Figure 2.4.1-1: Meet-in-the middle methodology for service modeling

2.4.2 An Example Service Model

The *service model* is common to well-designed SOAs, and provides a means to classify services. It is also sometimes referred to as a “taxonomy”. The service model is organized as a set of logical abstraction layers which categorize services “by the type of logic they encapsulate, the extent of reuse potential this logic has, and how this logic relates to domains within the enterprise” (Erl, 2008). This promotes the development of well-defined interfaces and provides a foundation for service reuse in enterprise-level deployments.

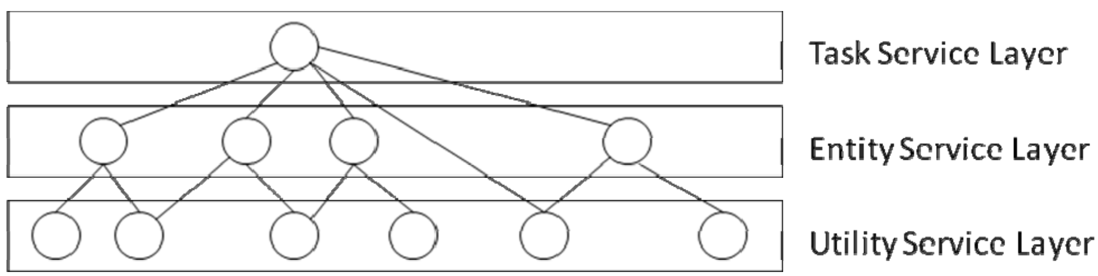


Figure 2.4.2-1: Service Model

For consistency, the examples in this white paper will leverage the service model shown in Figure 2.4.2-1, which is the model described in the book *SOA Design Patterns* by Thomas Erl. This service model has three distinct types of services – task services, entity services, and utility services - each with increasing attention to reuse. Although this model seems to imply strict hierarchy, there are no hard and fast rules for how services are composed. However, it is good design practice in an SOA to constrain the dependencies between services in a manner consistent with the service model hierarchy. For example, task services may depend on entity or utility services, but entity and utility services would never depend on task services.

Note that there is no right or wrong service model, particularly as there is not yet industry consensus around an authoritative or de-facto accepted classification scheme for services. As such, attention should be given to ensuring a service model is appropriate for context in which it’s used -- whether for an application, integration solution or enterprise architecture -- and that is it applied consistently in that context. The service model referenced in this white paper was defined to support the examples, and is not intended to be viewed as a normative service model for IHE-derived services in an SOA.

2.4.2.1 Task Services

Task services are based on a specific business process, and typically act as an entry point and controller for a service composition. As a result, task services generally have less reuse potential than the other services types. An example of a task service is a *RunAuditReport* service which retrieves, aggregates and displays audit record details for a clinical system.

2.4.2.2 Entity Services

Entity services are derived from one or more related business entities. They are considered highly reusable because they minimize dependencies to parent business processes. Examples of healthcare-specific business entities include patient, lab order and medical summary.

2.4.2.3 Utility Services

Utility services encapsulate common, cross-cutting functionality that is useful in many contexts but is not derived from the business architecture. They are also highly reusable services due to minimal dependencies on business as well as application context. Examples of typical utility services include notification, logging, and messaging.

3 Service Modeling By Example

The approach to service design is influenced by a multitude of factors, including organizational goals and objectives, choice of technologies and standards, as well as constraints imposed by existing IT systems and infrastructure, such as the adherence to IHE Integration Profiles. In order to illustrate how to leverage IHE profiles in an SOA, this whitepaper will illustrate modeling by example. Note, the examples that follow are appropriate for the context of this white paper, and are not intended to be prescriptive for every SOA. SOA practitioners considering IHE profiles will need to determine for themselves what the most efficient means are to deliver services across their enterprise.

3.1 Document Sharing Example

Our first example is for simple document sharing. A distributed community of providers is looking to publish, locate and retrieve clinical documents across their community of care. The key capability is the real-time assembly of a longitudinal health record for each patient within each of the practices. In addition, patient identities across different clinical and ancillary systems must be resolved. Treatment of security and access control is intentionally omitted from this example to reduce complexity and facilitate better understanding. This very important topic – including methods for secure design of security in an SOA – are the subject of the 2009 IHE IT-Infrastructure white paper entitled *Access Control*.

Figure 3.1-1: Document Sharing via Health Information Exchange

In our example, a provider community decides to put in place a basic Health Information Exchange (HIE), using infrastructure hosted in a secure, central location. The HIE will provide document sharing services and an Enterprise Master Person Index (EMPI) to facilitate identity resolution. The team tasked with implementing the solution has decided to adopt an SOA design and to leverage relevant IHE integration profiles.

The key considerations are:

- Heterogeneity: each of the providers have medical records systems and ancillary systems from a variety of vendors
- Portability: the providers would like to extend document sharing to other clinical and administrative systems in the future
- Extensibility: permit customization of the "provider-specific" aspects of the document sharing process i.e. local identity resolution, custom audit logging
- Legacy integration: accommodate both legacy and modern systems and applications
- Cost: minimize changes to provider systems to keep costs manageable
- Standards: use standards where possible. This includes the IHE integration profile Cross-Enterprise Document Sharing (XDS), and related profiles.

Service modeling for this example will follow a "meet-in-the-middle" approach. The first step is the top down analysis. For the document sharing solution, the key capabilities include:

- Submit document
- Locate document
- Retrieve document
- Resolve identity
- Transform legacy patient identifier
- Record audit record
- Retrieve patient's longitudinal health record

These capabilities are modeled as four separate services, comprised of two entity services, one utility and one task service:

- Document Sharing (entity service)
- Identity Resolution (entity service)
- Audit (utility service)
- GetPatientLHR (task service)

The Document Sharing service will implement the capabilities for document submission, location and retrieval. The Identity Resolution service will handle identity resolution and any transformation of legacy patient identifiers, and the Audit service will handle recording of audit records. Lastly, the GetPatientLHR service is responsible for orchestrating a number of other capabilities to perform distributed search and retrieval of multiple documents to assemble a patient's longitudinal health record.

The next step is to perform the bottom-up analysis. As mentioned above, the IHE profiles will be treated as an interoperable "black box" in the same manner as an existing application or legacy system. Analyzing the set of actors and transactions from the IHE *TF Volume I* and *Volume II* yields a number of wrapper services based on the profiles XDS (Cross Enterprise Document Sharing), PIX (Patient Identifier Cross Referencing), PDQ (Patient Demographics Query) and ATNA (Audit Train and Node Authentication). These services should be designed with interfaces and behaviors very true to the IHE technical framework specification, thus ensuring interoperable document sharing, identity resolution and audit recording in this SOA.

The required IHE profiles are modeled as five separate services:

- Patient Identifier Cross-Referencing (PIX) Manager (utility)
- Patient Demographics Query (PDQ) Manager (utility)
- Registry (utility)
- Repository (utility)
- Audit (utility)

Because they implement IHE profile actors, the IHE XDS and other integration profiles are being treated here as interoperability infrastructure, which exposes a set of capabilities not derived from the business architecture and which is useful in many contexts. These services are therefore classified as *utility* services as per the service model classification introduced in Section 2.4.2. Analysis of design alternatives is outside the scope of this white paper.

At this point, there are nine total service candidates, which is reduced to eight since Audit is a duplicate that was defined by both in the top-down and bottom-up methodology. The table in 3.1-2 illustrates a functional model of the services mapped to the specific IHE XDS transactions.

Use Case Description	Task Service	Entity Service	Utility Service	Capability Implemented	#	IHE Transactions
Primary Care Physician requests Patient's Medical History	Applied to all IHE-compliant services			Provide Assertion	1	XUA-ITI-40
				Query Time	2	CT-ITI-1
				Record Audit	3	ATNA-ITI-20
	Get Patient LHR (Longitudinal Health Record)	Identity Resolution	Patient Demographics Query (Patient Demographics Supplier)	Demographics Query	4	PDQ-ITI-21
			Patient Identifier Cross-Referencing Manager (PIX Manager)	Update Patient Identity, Notify	5	PIX-ITI-8, PIX-ITI-10, PIX-ITI-30
				Query Patient Identity	6	PIX-ITI-9
			Registry (Document Registry)	Update Patient Identity	7	XDS-ITI-8 XDS-ITI-44
			Audit	Record Audit	8	ATNA-ITI-20
		Document Sharing	Repository (Document Repository)	Submit Document	9	XDS-ITI-41
				Retrieve Document	11	XDS-ITI-43
			Registry (Document Registry)	Locate Document	12	XDS-ITI-18,
				Register Document	10	XDS-ITI-42
			Audit	Record Audit	13	ATNA-ITI-20

Figure 3.1-2: Functional Service Model Mapped to IHE XDS Profile

Next, the service dependencies and compositions are modeled. Note that in this example, the dependencies and compositions happen only in a single direction, top-down from the task service layer to the entity service layer and down to the utility service layer. This is critical to ensuring adherence to core SOA principles of loose coupling and autonomy, which fosters flexibility and reuse. It is also important to emphasize that this hierarchical composition relationship does not preclude service as at the lower levels from being accessed directly by layers not directly above, or even by service consumers.

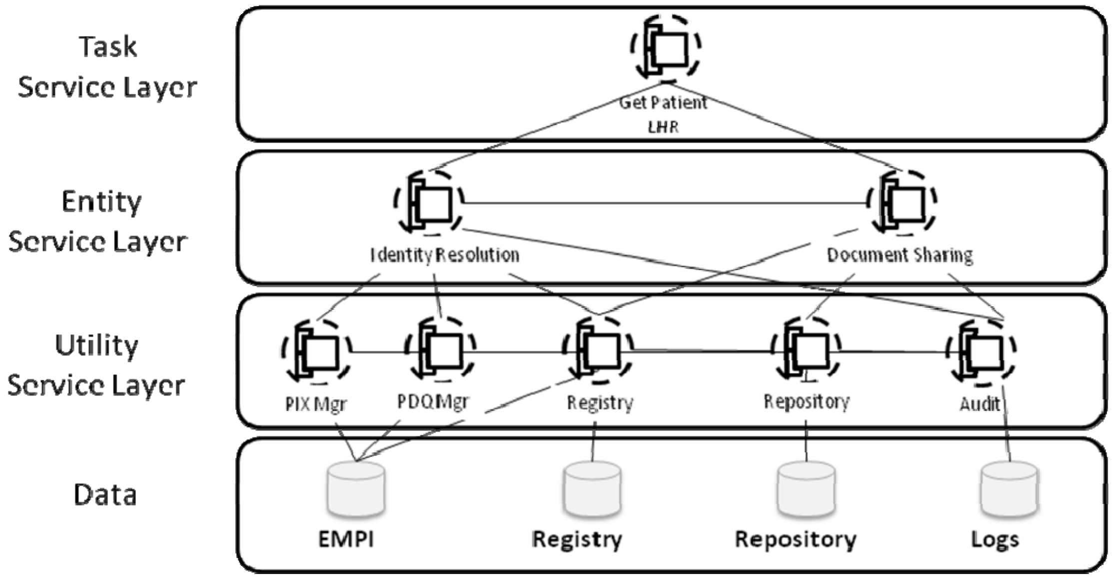


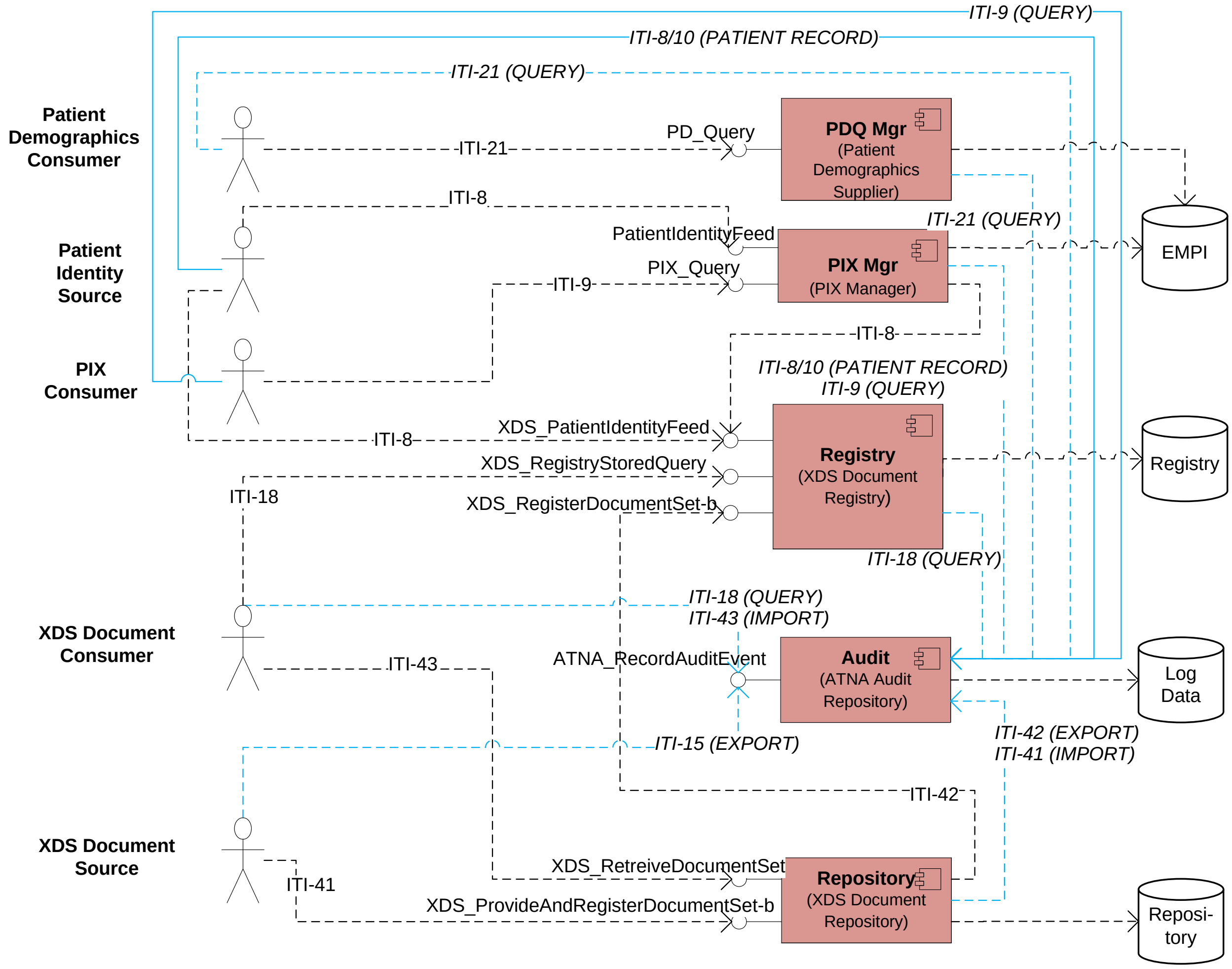
Figure 3.1-3 - Service Dependencies and Composition

3.1.1 IHE “Wrapper” Service Design

In the previous section, we introduced a basic “meet-in-the-middle” methodology for service analysis and design, including definition of the service model, service candidates and their dependencies. Up to this point a series of assumptions were made that treated IHE profiles as an interoperable “black box”. This section explores the design of the five IHE “wrapper” services in greater detail.

Figure 3.2-1 models the IHE profiles selected for the document sharing example. The IHE actors which are fulfilling a service provider role are represented by the newly defined utility services, designated by the UML component symbol (shaded). The specific IHE actor name which the service implements is labeled on each service, in parentheses. Each of the services exposes one or more interfaces and implements the core logic that permits it to fulfill its required role as outlined in the IHE Technical Framework specifications. This includes responding to named trigger events, supporting required IHE content profiles, and implementing the appropriate behaviors to ensure inputs to the service and outputs from the services conform to the profile.

Those IHE actors which are acting as service consumers are represented as non-service actors in Figure 3.2-1, designated by the UML Actor symbol. Data stores are depicted for information only, designated as canisters e.g. EMPI, Registry, etc. The data stores are part of the profile implementation and but are not part of the profile itself.



360 **Figure 3.1.1-1: Some IHE Profiles Depicted in SOA Terms**

The service consumers (Patient Demographics Consumer, Patient Identity Source) are each *initiators* of transactions. The Service Providers (Patient Demographics Supplier, etc.) are *recipients* of transactions. Notice that whether an actor behaves as a service consumer or a service provider is a feature of the specific transaction in question. For example, when the Repository service (XDS Document Repository actor) responds to requests for document retrieval (ITI-43), it is acting as a service provider in this transaction. That same Repository service acts as a service consumer when initiating a request to the Registry Service (XDS Document Registry actor) to Register a Document Set (ITI-42). Acting in "dual roles" is a common characteristic of services in an SOA, and is fundamental to understanding the concept of service composition discussed in the next section.

3.1.2 Service Composition

Figure 3.3-1 introduces two new entity services, Identity Resolution and Document Sharing, designated by the leftmost shaded UML component symbols. These services act in the dual-role described above, implementing the service consumer behaviors for XDS, PIX, PDQ and ATNA transactions, but also acting as a service provider to other service consumers desiring simple document sharing.

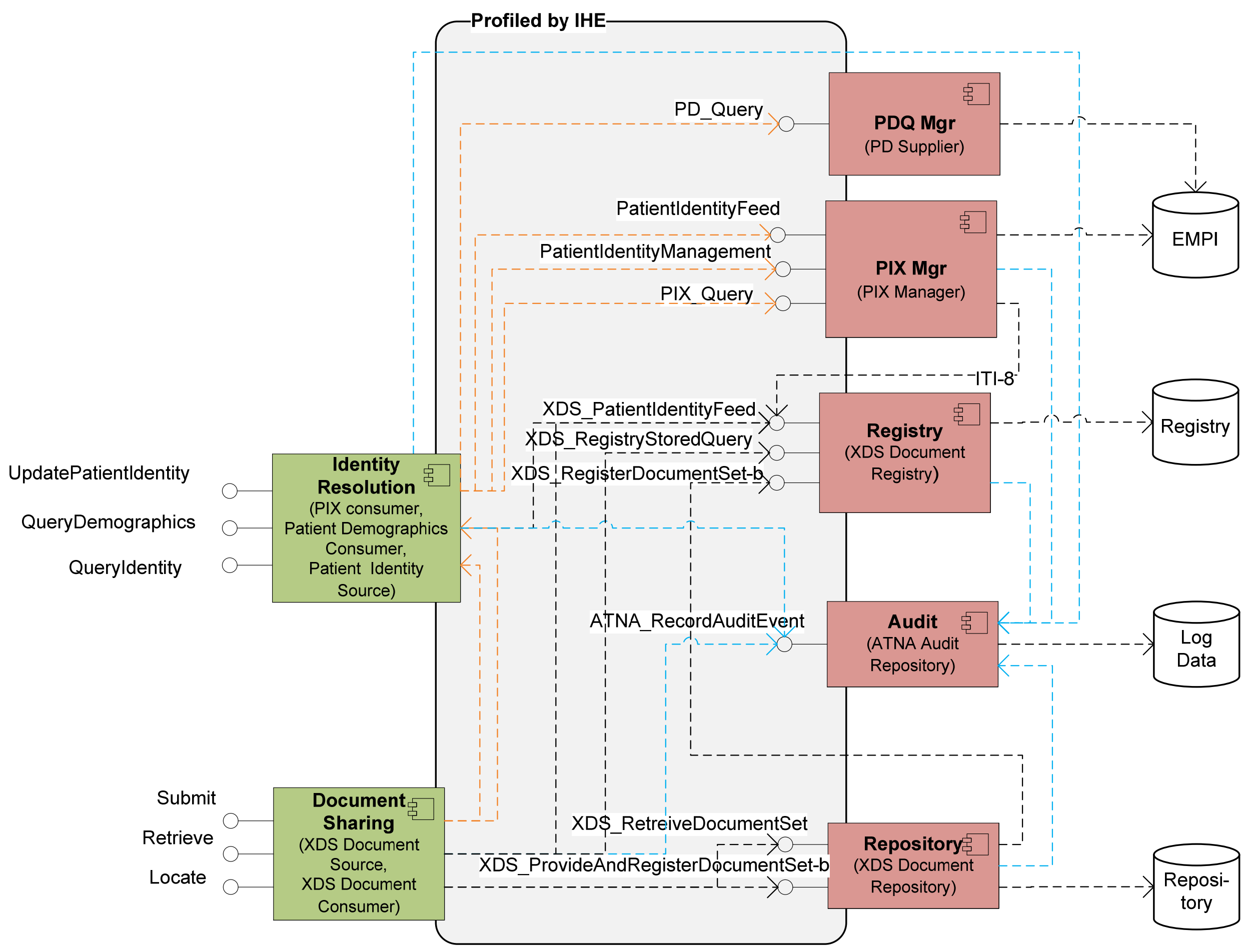


Figure 3.1.2-1: Service Definitions by Composition of IHE Profiles

The gray-shaded boundary designates the scope of functionality defined by IHE integration profiles, which extends from the IHE consumer actors implemented within the entity layer services to the interfaces exposes by the utility layer services. It includes all the transactions in between, which are IHE compliant. This means that Identity Resolution service logic implementing the IHE actors for PIX Consumer, PD Consumer, and Patient Identity Source conforms to the appropriate IHE profiles, thus enabling interoperable access to the relevant utility layer services (PDQ Mgr, PIX Mgr, and Registry and Audit in our example) or any other endpoint that can demonstrate such compliance. Similar holds true for the Document Sharing service, which is interoperates with any IHE-complaint XDS Document Registry, XDS Document Repository, or ATNA Audit Repository actor. Note that the service endpoints for Document Sharing service and Identity Resolution service are outside this boundary, since existing IHE profiles don't cover this.

From the perspective of the utility service layer, the IHE wrapper services expose capabilities that are compliant with IHE integration profiles for XDS.b, PIX, PDQ and ATNA. This means that any IHE-compliant consumer actor – not just the ones implemented in the Document Sharing and Identity Resolution service – can interoperate with those services. Note that the service endpoints for the utility services (PDQ Mgr, PIX Mgr, Registry, Repository and Audit service) are inside the scope of this IHE profile boundary.

3.1.2.1 Identity Resolution

In the terminology of the concept mapping table in section 2.3.2, Identity *composes* PIX, PDQ and Audit. It acts as a service provider to expose a set of identity management *operations* to outside systems. In turn, it offers to its service consumers capabilities that include those of its composed services. Identity also acts as a service consumer. In IHE terms, Identity performs grouping of the PIX, PDQ and Audit consumer actors, encapsulating those actors into a single service.

Service composition abstracts the set of dependent operations from consumers of the new, composed service. For example, the Identity Resolution service fulfills the audit requirements for both the PIX and PDQ consumers. This is useful to deliver a simplified view of PIX and PDQ services to the participants in the HIE network, who would otherwise have to each become familiar with the implementation details of the relevant IHE profiles in order to issue patient identity or demographics queries. Furthermore, by encapsulating this functionality in a single Identity Resolution service and implementing it as a web service, for example, the HIE participants can access the service via a vendor-neutral communications framework, providing immediate utility to the network, potentially leading to faster uptake and fostering greater reuse.

Composition also provides a means to address the legacy integration requirement introduced in Section 3.1. For example, a legacy MPI system may be required to provide the Patient Identity Feed to the PIX Manager service, but is not able to accommodate it for reasons of cost or inadequate technology. The Identity Resolution service could be flexibly adapted to handle the receipt of this data via a legacy protocol, and could be re-factored to delegate the legacy payload to a translation service responsible for transforming the feed into a HL7 v2 or HL7v3 complaint feed required by the PIX Integration Profile.

Some further examples of the value of service composition include the following:

- Developing a stable service interface and exercising appropriate governance permits greater control in making changes to the Identity Resolution service without impacting existing service consumers. This is an important consideration for service lifecycle management, particularly service versioning. Also, because the service implementation is independent from the interface, the core service logic can be re-factored or even replaced with no impact to existing consumers of the service.

- This refactoring of the service implementation may include altering the composition, or building an entirely new composition. The organization responsible for identity management may require additional service logic to accommodate new requirements, such as compliance legislation. Again, this can be implemented without impact to existing service consumers through close attention to interface stability.

3.1.2.2 Document Sharing

The Document Sharing service performs a very similar role to that of the Identity Resolution service, except that it delegates to the XDS-related services rather than the PIX and PDQ services. It also composes the Audit service and the Identity Resolution service as well. As a composed service, Document provides a very similar set of advantages over direct use of XDS that Identity provides over direct use of PIX and PDQ, the most powerful being a single, simplified entry point which abstracts the implementation complexities of XDS to service consumers. This enables the delivery of a service that is simpler to understand, easier to consume, and more flexible to change than the alternative of having each participant become familiar with the implementation details of the relevant IHE profiles in order to submit, query and retrieve documents.

There is an additional level of composition at play in the Document Sharing service, since it composes another entity service, the Identity Resolution service. This is the first example of reuse in our document sharing example, since the Identity Resolution service is fulfilling a role as an independent provider of an identity resolution service for the HIE participants. This composition of Identity by the Document Sharing service has other advantages as well. It permits a clean separation of concerns, which is a fundamental principle of service-orientation. This means that the Document Sharing service need only be concerned with providing document-related capabilities; it delegates everything else to other services. In this case, the Document Sharing service is able to effectively offer identity resolution as an integrated capability to its service consumers, which permits implementers of the service some flexibility in determining how and when identity resolution should happen during document sharing. An example of this would be permitting the Document Sharing service to accept organization-specific patient identifiers, which the Document Sharing service would resolve to common names before issuing a PDQ query to get the correct identifier and value for the Assigning Authority.

3.1.3 Example of a Deployment View

Figure 3.4-1 illustrates an example deployment view of the services designed in the document sharing example. One additional task-layer service, GetPatientLHR, has been created to handle the creation of a Longitudinal Health Record (LHR).

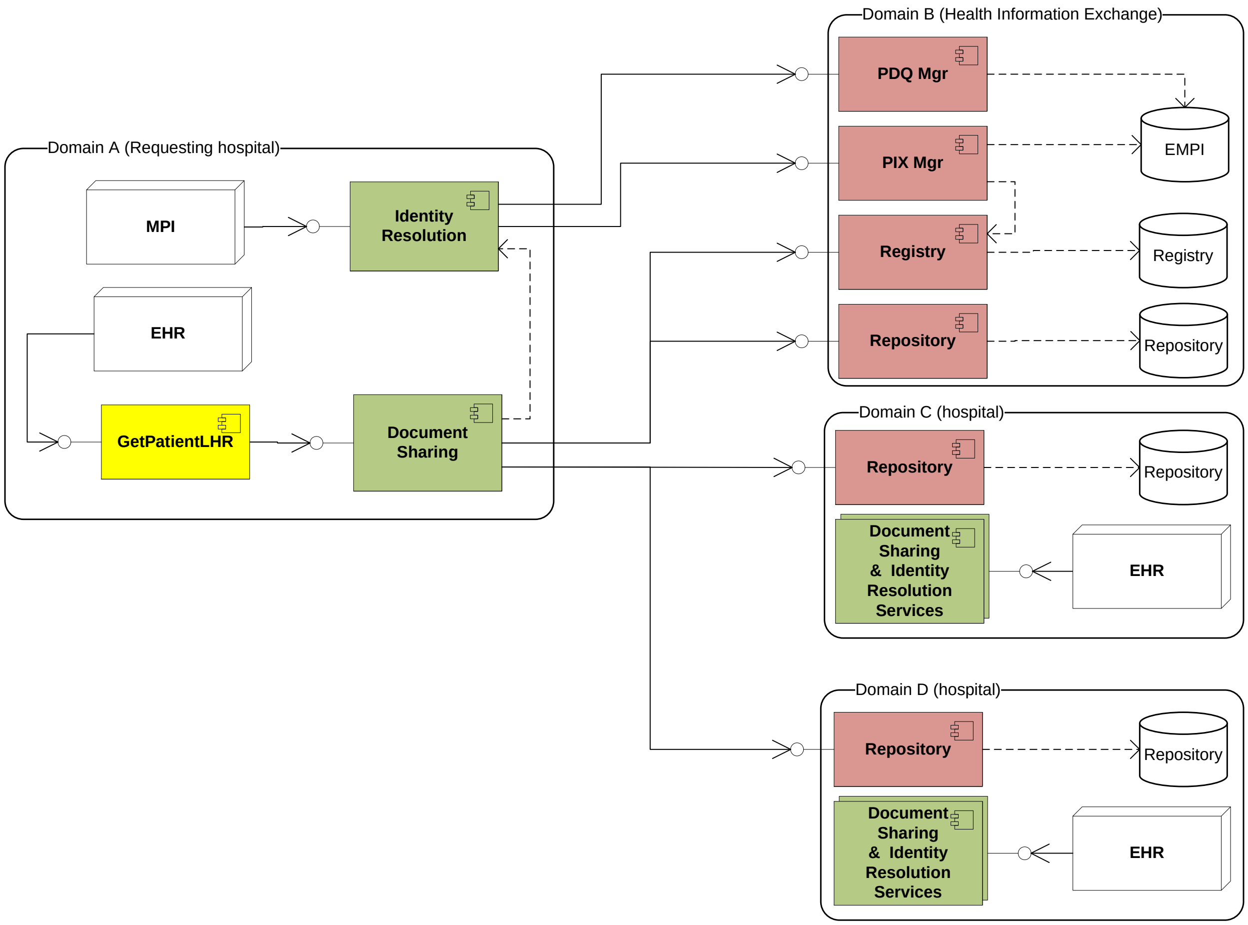


Figure 3.1.3-1: Document Sharing Example Deployment

In this topology, The Document and Identity Resolution services are deployed as federated services at each of the HIE participant domains (A, C and D). This provides a set of local services for the LHR service to perform a federated query and retrieval of the set of documents required to establish a patient longitudinal health record. It also enables a simple set of interfaces to permit each of the HIE participants to engage in interoperable document sharing leveraging the IHE XDS integration profile.

Note that the Document and Identity Resolution services could have been deployed centrally in Domain B. In such a model, Identity Resolution and Document Sharing services need to be standardized, at least across all participating domains in our example. In order for such standardization to take place, common service definitions, leveraging existing IHE profiles, are needed. In the next section, we describe in detail on how one such service definition, for our Identity Resolution service example, could be approached.

3.2 Example Service Definition for Identity Resolution

In this section, we build an example service, Identity Resolution, leveraging existng IHE profiles and the SOA concepts and definitions of ***Error! Reference source not found.*** Identity

Resolution illustrates one approach to defining a service that provides the capabilities of Identity in our document sharing example.

In building our example, we draw heavily from concepts embodied the Entity Identification Service (EIS) specification[7], published by the Object Management Group (OMG). The similarities are only general; Identity Resolution is not a line-by-line implementation of EIS.

3.2.1 Abstract Service Definition

First, we provide an example abstract service definition for Identity Resolution. Our abstract service definition begins with a list of concept definitions and illustrations that are used in a precise way to describe the behaviour of the service operations. A sample list of concept definitions is given in Table 3.2.1-1.

Table 3.2.1-1: Concept Definitions in an Abstract Service Definition

Real World Entity (RWE)	Represents the actual physical thing itself, e.g. the actual Person, the actual Device etc.
Entity	The software representation of a RWE - a record.
Source	A system generating an Entity.
Domain	A set of values in which each is unique.
Identifier (ID)	A value within a Domain that is associated with an object – an Entity, a Source, etc. - and uniquely identifies it within the scope of the Domain.
Entity ID	An Identifier associated with an Entity.
Source ID	An Identifier associated with a Source.
Trait	A data element used by an EIS matching algorithm to link a particular (Source ID, Entity ID) pair with an existing Identity, based upon a conclusion that the Entities they are identifying represent the same RWE.

Our sample abstract service definition then goes on to use the defined terms to describe our service capabilities.

> Identity Resolution creates and maintains of an index consisting of a linked set of Source ID/Entity ID pairs representing the same Real World Entity (RWE). … A Source ID and Entity ID are supplied in pairs in order that they may uniquely identify an Entity with *the Domain of the index.* (An Entity ID alone uniquely identifies an Entity *within the Domain of the Source*).

[7] Entity Identification Service (EIS). Object Management Group 2008.

Thus, an index is associated with a set of Source IDs in which each is unique. Each Source ID is associated with a set of Entity IDs in which each unique. Since an Entity ID within the Domain of one Source can be repeated within the Domain of another Source, the Source ID/Entity ID pair is required to uniquely identify an Entity within the index. Entities within the index are linked when they are determined to be associated with the same Real World Entity. An RWE ID which uniquely identifies the RWE within the index is assigned to the linked set of Entities.

Source ID	Entity ID	RWE ID	Trait (First Name)	Trait (Last Name)
A	1001	10000	John	Jones
A	1002	10001	Michael	Jackson
B	2001	10000	John	Jones
B	2002	10002	Michael	Tyson
C	1001	10000	John	Jones
C	2001	10003	Michael	Flynn

Figure 3.2.1-1: Example index

These definitions and descriptive text lay the foundation for defining what work is performed by our operations.

3.2.2 Operations

Our abstract service definition goes on to define the service operations:

Table 3.2.2-1: Identity Resolution Operations

Operation	Description
Register	This operation inserts a Source ID/Entity ID pair and supplied Traits into the index with implicit linking to other matching Source ID/ Entity ID pairs, based on the configured internal matching algorithm.
Update	This operation updates the Traits stored in the index for the Entity identified by the supplied Source ID/Entity ID pair.
List	This operation retrieves all the Source ID/Entity ID pairs that are linked to the supplied Source ID/Entity ID pair. The operation can be filtered to only return entities within specified Source domains.
Query	This operation provides the means to perform a broad search of all records in the index whose traits match some criteria in the supplied search criteria.
Link	This operation provides the means to create an explicit (as opposed to automatic) linking between two Source ID/Entity ID pairs in the index.
Unlink	This operation provides the means to create an explicit (as opposed to automatic) linking between two Source ID/Entity ID pairs in the index.
Merge	This operation provides the means to explicitly consolidate index Source ID/Entity ID pairs in the index.

Operations are further defined by the start state and end state of the index for each operation.

For example, the Register operation may be defined in terms of what happens if the operation is called passing each of the following payloads:

Case: input RWE is…	Source ID	Entity ID	Trait (First Name)	Trait (Last Name)	Outcome
Unknown to index	C	2090	Michelle	Piper	Insert with new RWE ID
Known to index	C	2090	Michael	Jackson	Insert and link to existing RWE ID
Known and previously registered by this Source	C	2090	Michael	Flynn	Insert and link to existing RWE ID; indicate possible duplicate in return values
Can't tell if known or unknown	C	2090	Mike	Tyson	Insert but do not assign RWE ID pending link operation

Figure 3.2.2-2: Further Definition of Register Operation

The operation abstract definition needs to make sure that a precise outcome is defined for all possible cases of input combinations. Such terms as "known" and "unknown" may be defined in terms of whether a matching algorithm determines a definite match; whether the input Source ID matches the Source ID of a previously registered matching RWE (suggesting a duplicate); and so forth.

3.2.2.1 Relation to the IHE Technical Framework

We state in Section 2.3.2**Error! Reference source not found.**:

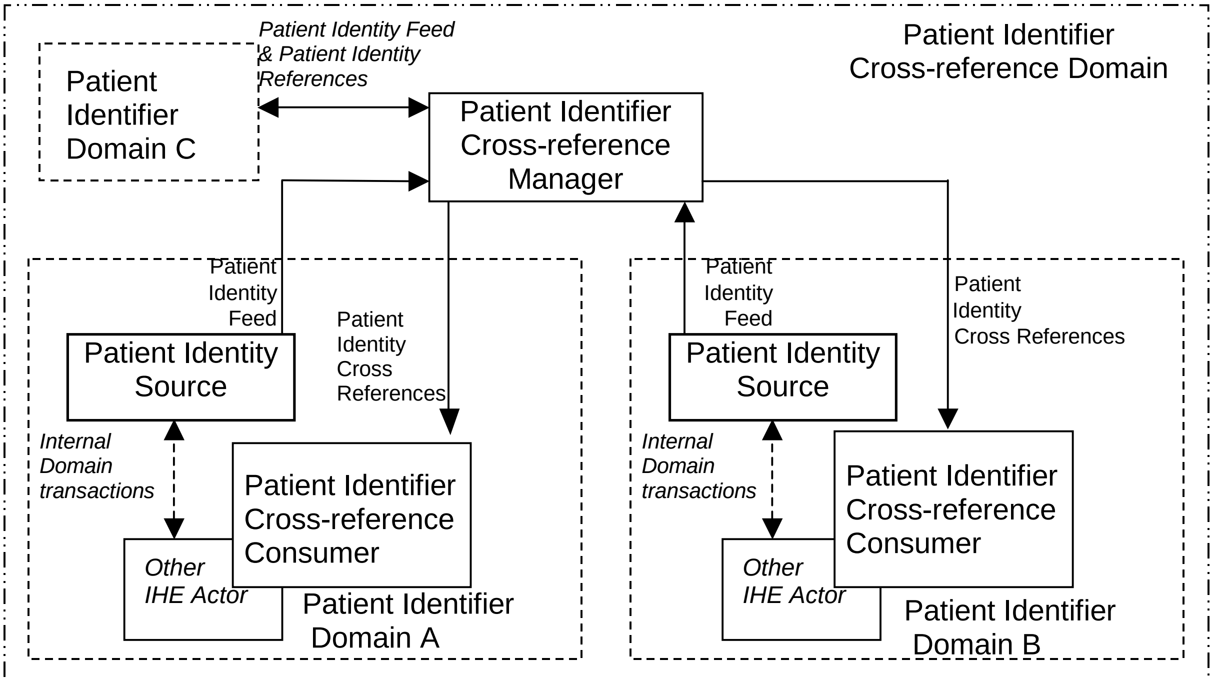
> Generally speaking, the content in Technical Framework Volume I most closely matches the (service definition) abstract portion.

We also say:

> (An operation) loosely corresponds to transaction or message. Sometimes service operations correspond directly to IHE transactions or message, but not always.

The IHE IT Infrastructure (ITI) Technical Framework Volume I (ITI-TF-1) Integration Profiles defines similar concepts. From *Section 5, Patient Identifier Cross-Referencing (PIX):*

> "The ***Patient Identifier Cross-referencing Integration Profile (PIX)*** … supports the cross-referencing of patient identifiers from multiple Patient Identifier Domains…. The following diagram shows the intended scope of this profile….



We see the following similarities with our Identity Resolution abstract definition:

- Both support the concept of domains of identifiers.
- Both support cross-referencing of identifiers from multiple domains.
- The Identity Resolution operation Register loosely corresponds to the IHE transaction Patient Identity Feed.

To further determine whether the relationship between our abstract service definition and IHE patient identification profiles, we would need to carefully check all aspects of our service definition against Volume I content to determine the relationship.

3.2.3 Concrete Service Definitions

In Section 2.3.2 we state:

> Generally speaking, the content in Technical Framework Volume II most closely matches the concrete portion of the service defition.

One or more concrete service definitions may correspond to a single abstract service definition. In our example Identity Resolution, we create multiple concrete service definitions based upon passing different but equivalent IHE messages as payloads to the same Identity Resolution operation. Messages are taken from the transactions of IHE IT Infrastructure (ITI) Technical Framework Volume II.

PIX, and the Patient Demographics Query (PDQ) profile that is grouped with it in our example, may be implemented using either HL7 Version 2 or HL7 Version 3. , Therefore, we create two concrete service definitions corresponding to our single abstract service definition: one using HL7 V2.5 messages as operation payloads, and one using HL7 V3.0.

3.2.3.1 HL7 V2.5 Concrete Service Definition

The payloads passed to Identity Resolution operations under the HL7 V2.5 concrete definition are given as:

Operation	IHE Transaction	HL7 V2 Message	HL7 V2 Event	Trigger Event
Register	ITI-30 Patient Identity Management	ADT	Create new patient	A28
Update		ADT	Update patient information	A31
List	ITI-9 PIX Query	QBP	Get Corresponding Identifiers	Q23
Query	ITI-21 Patient Demographics Query	QBP	Find Candidates	Q22
Link	ITI-30 Patient Identity Management	ADT	Link Patient Information	A24
Unlink		ADT	Unlink Patient Information	A37
Merge		ADT	Merge two patients	A40

Figure 3.2.3.1-1: HL7 V2.5 Operation Payloads

3.2.3.2 HL7 V3.0 Concrete Service Definition

The payloads passed to Identity Resolution operations under the HL7 V3.0 concrete definition are given as:

Operation	IHE Transaction	HL7 V3.0 Message	HL7 V3.0 Event	Trigger Event
Register	ITI-44 Patient Identity Feed V3	PRPA_RM201301UV02	Patient Registry Record Added	PRPA_TE201309UV02
Update		PRPA_RM201302UV02	Patient Registry Record Revised	PRPA_TE201309UV02
List	ITI-45 PIXV3 Query	PRPA_RM201303UV02	Patient Registry Get Identifiers Query	PRPA_TE201309UV02
Query	ITI-47 Patient Demographics Query V3	PRPA_RM201302UV02	Find Candidates Query	PRPA_TE201305UV02
Link	none			
Unlink	none			

Merge	ITI-44 Patient Identity Feed V3	PRPA_RM201302UV02	Patient Registry Duplicates Resolved	PRPA_TE201304UV02

Figure 3.2.3.2-1: HL7 V3.0 Operation Payloads

Regardless of which concrete service definition is implemented, the way the service functions will be the same. The actions described above in the abstract service definition are performed regardless of the format in which the payload is passed to the service.

3.2.4 Other aspects of service definitions

If we have done our job, this example has given a feel for how an organization could set up a service interface that leverages IHE profiles. There are many details of our Identity Resolution example that are not elaborated here. For example, operations may be structured into one or more service interfaces so that some implementations may include a minimal set of capabilities while others include more extensive ones. An example for this exists with the IHE PIX HL7V3 profile and the IHE PIX (based on HL7V2) profile with the lack of an HL7V3 equivalent of the link and unlink operations of HL7 V2.5. A required interface could exclude these operations, which would appear in a second, optional interface.

We have not touched upon precisely how the detail contained in the IHE Technical Framework would be carried forward into the service defintion. For instance, in HL7 V2, the information contained in a transaction trigger event may no longer be needed when conveyed by the service operation itself.

3.3 Value

In summary, leveraging IHE profiles in a SOA provides the following value:

1. Reduced complexity. Document Sharing and Identity Resolution services provide a single, simplified entry point which abstracts the implementation complexities of the service and its dependencies
2. Flexible deployment. All hospitals can choose to host the Document Sharing and Identity Resolution services in their domain , or consume a centrally hosted service
3. Increase agility i.e. adaptable to change. Techniques for service composition and design patterns like service mediation enable flexible customization of services while ensuring existing interfaces remain stable. This provides a means to adapt to changing requirements in a highly cost effective manner.
4. Phased approach to modernization. Services support multiple service bindings, enabling a single service definition to support multiple service instantiations that may differ widely in terms of application protocol and operation semantics. This permits hospitals to phase in their legacy infrastructure to take advantage of document sharing capabilities.
5. The possibility of a new level of interoperability at the service interface level. Our Identity Resolution example defines testable service functionality that is common to a number of specific messaging standards. It is interoperable in its service provider role;

the extra step of passing the payload on to an existing PIX or PDQ manager using existing IHE transactions may, but does not need to, be taken. It may simply be a new or modified interface to an existing PIX or PDQ manager.

4 Further Issues for Exploration

This white paper is intended to open the doors to the discussion of issues raised here but not brought to conclusion. Here are examples of such topics for further discussion:

1. Does the introduction of a SOA approach imply the selection of a particular architectural approach? Specific implementation choices and architectural approaches are generally avoided by IHE because they may limit competition, favor certain vendors or implementations, and because such choices are not required to achieve interoperability.
2. On the other hand, does the real value of a SOA approach lie in the ability to specify things in an abstract way? Perhaps there are opportunities for new methods of specifying profiles in the Technical Framework along the lines of the examples given here. This may create further adaptability as standards change, since abstract service defintions can be reused, even as "on the wire" standards evolve.
3. Expressing certain IHE transactions as services may require establishing specific conventions related to the specific standards used on the wire; for example making consistent choices in the the handling of HL7 V2 or V3 messages trigger events in a concrete service definition.
4. As services are defined with alternative concrete service definitions, the support of different protocols on the wire does not always allow for the coverage of the entire range of operations. A consistent approach to address the service "gaps", such as the use of interfaces, may be of value to IHE.

5 Conclusion

Interoperability refers to a set of mechanisms in place that guarantee the sharing of data. SOA and service-oriented design principles can foster the design of interoperability in support of services, but do not intrinsically guarantee it. The paper has demonstrated that IHE with its profiles can bring proven interoperability to healthcare SOA solutions.

This paper proposes a bridge between the SOA and IHE terminology to provide an integrated language for discussing IHE concepts in the context of SOA. It also looks into the future for further discussion of the challenges and opportunities as further efforts are made to combine SOA and IHE approaches.

