uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Chintan Doshi

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (E-Business Technologies)

GRADE / DEGREE

System Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

An Integrated Trusted Processes Framework for Consumer-facing B2B Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Dr. Liam Peyton

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Dr. Daniel Amyot

Dr. Thomas Tran

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

An Integrated Trusted Processes Framework for Consumer-facing B2B Networks

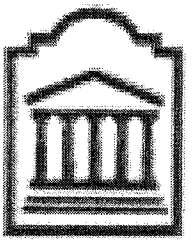
Chintan Doshi

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the M. Sc. degree in E-business Technologies



University of Ottawa

Ottawa, Ontario, Canada

May 2008

Your file Votre référence
ISBN: 978-0-494-50873-2
Our file Notre référence
ISBN: 978-0-494-50873-2

Abstract

With the advent of the Internet, co-operating enterprises are increasingly sharing consumer data to deliver them information-rich online experiences with value-added services. At the same time, technological advances in web services standards based on a Service Oriented Architecture (SOA) design have enabled heterogeneous Business-to-Business (B2B) integration between enterprises. End-to-End business processes that span across multiple enterprises can be developed using the Business Process Execution Language (BPEL). However, such processes must address issues that normally do not arise for processes within a single enterprise. A framework is needed which supports a SOA-enabled business process management approach but also has technical infrastructure to address issues related to identity, privacy, compliance, monitoring and people interaction. In our thesis, we accomplish this using a framework that supports the design, implementation and management of trusted B2B processes defined using the BPEL standard and deployed into Circle of Trust architecture as specified by the Liberty Alliance federated identity standards. A key contribution of our thesis was to extend the Circle of Trust architecture with a new entity that introduces components to help businesses manage their B2B processes. Two use-case scenarios involving information-rich B2B processes were implemented to evaluate our framework.

Acknowledgment

I would like to specially thank my supervisor Dr. Liam Peyton for his guidance, encouragement, feedback and patience throughout this Master's thesis. His advice and insights have been extremely valuable in helping me conduct my research and improve the quality of this thesis. I am fortunate to be one of his students and learnt a lot under his supervision throughout my Masters program.

Thanks to Pierre Seguin and Jun Hu with whom I enjoyed collaborating with on our early research publications. Also, thanks Benjamin Eze for his collaboration on scenarios related to people interaction. Finally, I would like to thank my family for their encouragement and support through my academic studies.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
API	Application Programming Interface
ATS	Audit Trail Service
B2B	Business-to-Business
BAM	Business Activity Monitoring
BPA	Business Process Automation
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
CoT	Circle of Trust
CTR	Click-through rate
DS	Discovery Service
EAI	Enterprise Application Integration
EPR	End-point reference
ESS	Event Stat Service
FIM	Federated Identity Management
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
ID-FF	Identity Federation Framework from Liberty Alliance
IDP	Identity Provider

IM	Identity Management
ID-SIS	Identity Service Interface specifications
ID-WSF	Identity Web Service Framework from Liberty Alliance
IRC	Information Rich Commerce
IS	Interaction Service
IT	Information Technology
ITPF	Integrated Trusted Processes Framework
J2EE	Java 2 Platform Enterprise Edition
KPI	Key performance indicator
OASIS	Organization for Advancement of Structured Information Standards
Oracle PM	Oracle Process Manager (BPEL engine)
PDP	Policy Decision Point
PIPEDA	Personal Information Protection and Electronic Documents Act
SAML	Security Assertion Markup Language
SSL	Secure Socket Layer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SP	Service Provider
SSO	Single-sign on
TP	Task Portal
TPP	Trusted Process Provider
TS	Task Service
UDDI	University Description, Discovery and Integration

URI	Uniform Resource Indicator
W3C	World Wide Web Consortium
WFMS	Workflow Management System
WS-Addressing	Web Service Addressing Standard from W3C
WS-BPEL	Business Process Execution Language for web services.
WSDL	Web Service Description Language
WS-Security	Web Service Security Standard from OASIS
XML	Extensible Markup Language
XPATH	XML Path Language

Chapter 1. Introduction

1.1. Thesis Motivation

The rapid proliferation of the Internet as an open, universal and dynamic communications platform has made it easier for organizations to share data. B2B networks of cooperating enterprises have emerged to support cross-organizational business processes that collect and share detailed information about consumer's preferences, historical transactions, browsing patterns etc to provide them value-added services. For example, consumers can benefit from personalized product suggestions, customized marketing offers and convenience of information-rich transactions. However, such collaborative processes face unique issues that normally do not arise in processes that run within an organization.

Consumer's have privacy concerns over the sharing of their data. Identity management techniques are required to manage consumer's identities across a heterogeneous B2B environment and protect it while conducting cross-organizational transactions. [Fichman2003] has highlighted the role of consumer privacy in building a trusted infrastructure for the successful adoption of information-rich e-commerce services. Accordingly, processes must safeguard consumer identities and restrict their data-sharing activities in accordance with consumer's preferences.

Governments have enacted legislations such as the Personal Information Protection and Electronic Documents Act [PIPEDA2000] in Canada that limit an organizations use of personal

information and requires them to comply with these rules. Proper means must exist to ensure compliance of processes with privacy legislations.

Infrastructure support is required to allow organizations to monitor events and activities of a process and to measure the overall performance of processes in order to bring end-to-end visibility to business operations and ensure that business level objectives are being met.

Interaction with people is another key requirement of any business process. Techniques must exist that facilitate interactive participation of employees belonging to multiple organizations as part of the management and automation of processes across B2B networks.

Issues related to identity and privacy are driven by business needs to comply with legislations and build consumer confidence. On the other hand, monitoring and human interactions are issues driven by business needs to manage, improve and coordinate the process workflow.

1.2. Thesis Problem and Objectives

There is a set of standards from W3C [WC3] based on a Service Oriented Architecture (SOA) [Papazoglou2003, Curbera2002] that has emerged as a key driver for heterogeneous B2B integration. In addition, OASIS [OASIS] has been developing standards related to business process management and automation. These standards promote a new technological opportunity for consumer-facing B2B networks to rapidly build agile information-rich business processes that deliver value to consumers and create new revenue opportunities for businesses. However, an architecture or framework based on these standards must also address issues related to identity, privacy, compliance and monitoring discussed above that normally do not arise for

business processes that operate solely within an organization. The Liberty Alliance Project [Liberty] has proposed some initial standards in this area.

The objective of our thesis framework is to enhance traditional SOA-enabled business process management with additional technical infrastructure that enables consumer-driven collaborative processes to conduct transactions and share data in a trusted and seamless fashion. Our framework should support the design, implementation and management of trusted collaborative processes defined using Business Process Execution Language (BPEL) standard [Arkin2005] and deployed into a Circle of Trust architecture[IDFF2007,IDWSF2004] as specified by the Liberty Alliance federated identity standards. The key goals of our framework are:

- Leverage federated identity standards to securely manage consumer identities and authentication in B2B network.
- Protect consumer data through a system of permission-based consent for data sharing between businesses in a B2B network.
- Help collaborating businesses manage their processes through a system of event monitoring and audit trails for documentation of privacy compliance and performance management.
- Provide transparency through privacy auditing reports.
- Support both fully automated processes and interactive processes that involve human participation.

1.3. Thesis Contributions

The thesis proposes and evaluates a new approach for developing collaborative B2B processes that deliver information-rich services to consumers. In doing so, the key contributions made by our thesis were:

- Identification of the key requirements for trusted processes in consumer-facing B2B Networks.
- Proposed an integrated framework for trusted collaborative processes that combines BPEL-related standards for business process automation and Liberty Alliance standards for federated identity management in a B2B SOA.
- Extensions to the Liberty Alliance defined Circle of Trust to support trusted business process management consisting of a Trusted Process Provider, an Audit Trail Service (and privacy portal), Event Stat Service and Task Service (and task portal).
- Identified gaps and conflicts between the WS-BPEL 2.0 standards and Liberty Alliance specifications that are problematic for integration and suggested fixes.
- Comparative survey of existing tool and vendor support for BPEL related and Circle of Trust related standards.

Some of the above contributions have been published in the following papers:

- L. Peyton, **C. Doshi**, J. Hu, P. Seguin, "Information Rich Monitoring of Interoperating Services in Privacy Enabled B2B Networks", *International Journal of Advanced Media and Communication*, Inderscience Publishers, Geneva, Switzerland. Vol. 2, No.2, 2008.

- **C. Doshi**, L. Peyton, "Trusted Information Processes in B2B Networks", To appear in the *Proceedings of the 10th International Conference on Enterprise Information Systems*, Barcelona, Spain, June, 2008
- L. Peyton, **C. Doshi**, P. Seguin, " An Audit Trail Service to Enhance Privacy Compliance in Federated Identity Management ", *Proceedings of CASCON 2007*, ACM, Toronto, October, 2007.
- L. Peyton, J. Hu, **C. Doshi**, P. Seguin, "Addressing Privacy in a Federated Identity Management Network for E-Health", *8th World Congress on the Management of eBusiness*, Toronto, July, 2007.

1.4. Thesis Organization

The thesis is organized as follows. In Chapter 2, key standards from OASIS and WC3 that are used in our framework are briefly explained. A background on federated identity management (including the Liberty Alliance Project) standards as well as business process automation (including the BPEL standard) is presented. Related work in trust, consumer privacy and legislations is briefly discussed. Chapter 3 describes our thesis approach. The requirements of a trusted process are first established before our Integrated Trusted Processes framework (ITPF) is explained in detail. In Chapter 4, we look at the implementation of two use-case scenarios that were used to evaluate our framework approach. An overview of the design tools, Circle of Trust prototype and runtime environment in our implementation is also presented. Chapter 5 evaluates our framework alongside different dimensions such as the feature set provided, efforts to implement, debug, maintain trusted processes and also the complexity involved in scripting trusted processes. The Liberty Alliance specifications and BPEL standards

are analyzed from the point of view of our framework and some of the existing supporting tools tried out in lab are compared. Chapter 6 summarizes our contributions, conclusions and proposes future work.

Chapter 2. Background

2.1. Trust in E-business processes

2.1.1 Growth of E-business and Information-Rich Commerce

The rapid proliferation of the Internet and support for E-business has transformed the way organizations do business. New technologies have evolved that help organizations streamline their existing business processes and collaborate with their business partners to share knowledge and become more responsive to changing market needs.

The idea of information-rich commerce is presented in [Fichman2003]. The new digital environment online has made it easier for organizations to collect and track vast amounts of data about consumers such as their preferences, interests, demographics, opinions, browsing behaviors and so on. B2B integration has made it possible for co-operating enterprises to build B2B business processes to share and leverage such data. Data mining techniques are being utilized that enable organizations to analyze such data to offer personalized services tailored to consumer's individual needs. Similarly, recommendation systems and collaborative filtering techniques have matured to help consumer find new interesting services and products. Businesses benefit from the opportunity to engage in one-to-one marketing with consumers, increase customer loyalty and customer satisfaction [Peppers1999]. Consumers benefit from convenience of personalized internet experiences that suits their preferences and value-added product offerings customized to better meet their needs.

2.1.2 Trust

The European Dependability initiative [Wilikens1998] identified the four new drivers for trust in e-business systems as globalization, complexity of large-scale open systems, transition to virtual digital environments and rapidly evolving technologies.

[Chellappa2005] concludes that consumer usage of personalized services is positively influenced if they trust the organizations they are dealing with. Moreover, it suggests that organizations should engage in trust-building activities to improve their ability to acquire and share consumer data. Such activities may require them to deal with third-parties intermediaries.

Trust has multiple dimensions having been studied in a wide variety of academic disciplines. [Boon1991] presents a general definition for trust as "a state involving confident positive expectations about another's motives with respect to oneself in situations entailing risk". [Rantnasingam2002a] analyze the importance of technology trust in e-commerce and define technological trust as "the subjective probability by which organizations believe that the underlying technological infrastructure is capable of facilitating transactions according to their confident expectations". There are has been extensive research in literature that model or measure trust for entities participating in a system. For example: [Tran2004] presents a strategy to model trust for agents in electronic marketplaces based on their reputation. [Xong2003] uses community-based reputation model of peers to quantify and compare trustworthiness in peer-to-peer e-commerce. In our thesis, we do not address trust infrastructure issues in terms of quantifying or modeling the trust that users have in the business processes. Rather, we make an assumption that a well designed support infrastructure for consumer-oriented B2B process must have certain requirements and we examine how our framework architecture can be engineered to

address those requirements. Our rationale is that if a framework meets those requirements, the users are more likely to trust the system, but it is beyond the scope of our thesis to measure or model the level of their trust.

In investigating the benefits and challenges of organizations collaborating in a B2B network to provide value-added services to customers, [Fichman2003] highlighted the challenge of developing an IT infrastructure that can be trusted to ensure transactions are conducted privately, securely, and in accordance with consumer preferences. The author identifies its presence as one of the key factors to ensure successful adoption of information-rich commerce. Its suggested that such an infrastructure must provide transparency to what data gets shared. Further, the author points out that government legislation on consumer privacy may also drive the need for it.

A framework for eliciting high level trust requirements in e-business is presented in [Jones2000]. It looks at trust from the point of view of its key stakeholders, namely the consumers, businesses and administrators and tries to deduce a set of generic trust requirements for each of them. Another conceptual framework based on a stakeholder perspective is presented in a survey of trust online in [Shanker2002]. The framework notes that trust for stockholders or business partners can be related to availability of information on firm's strategy and performance while trust for a regulator may depend on transparency with respect to a firm's compliance of regulations.

2.1.3 Consumer Privacy

Consumer privacy is generally defined as a consumer's ability to protect access to their identity information by others and revealing it selectively only in accordance with their preferences.

The first point requires that consumer's information should be collected by organizations only with their proper knowledge. Further, data that's collected must be protected from unauthorized access. [Shanker2002] suggests that consumer's privacy online is often tied to degree of control they have over their personal information. This includes environment control (control over the use of personal information while conducting an online transaction) and secondary control (control over any subsequent use of collected information once the transaction is complete).
To engage in information-rich commerce, the consumer is required to disclose information about them which may be shared between cooperating enterprises. This could include their personally identifiable information. Thus, the benefits derived from value-added services come at the loss of privacy. In investigating customers adoption of personalization, [Chellappa2005] concludes that consumers usage of personalization is a tradeoff between their perceived value from personalization and their concern for privacy. Further, such concern is not just restricted to their personally identifiable information but also about non-identifiable information in the fear that it may be possible to deduce identifiable information from it sometime in future.

2.1.4 Privacy Legislation

In response to consumer's privacy concerns online, governments have passed legislation and laws that require businesses to limit the sharing of personal information and protect identity of consumers. Examples include European Union Prime Directive on Privacy [EUPrivacy2002], the Health Insurance Portability and Privacy Act [HIPAA1996] in the United States, and the Privacy Act and the Personal Information Protection and Electronic Documents Act [PIPEDA2000] in Canada.

PIPEDA establishes a number of basic principles for organizations to comply with the legislation. A key principle states that businesses must obtain consumer consent before collection, use and disclosure of their information with third-party businesses. Another principle requires businesses to clearly identify their purpose while collecting consumer information and later on limit its use of data to that purpose.

Early work on addressing the relationship between privacy enhancing technology and privacy legislation was done in [Peyton2004], which proposed an Information Transfer Registry (ITR) to support the logging and auditing of information transfers between businesses in B2B networks. Key principles for compliance with privacy legislation were identified as:

- Organizations must identify how they intend to use personal data and receive consent from the individual.
- Organizations must establish internal procedures to document and safeguard their use of data.
- Individuals must be given access to their data and have recourse to challenge its accuracy and use.

Audit trails that record details of user activity are relevant to privacy compliance. In [Yip2006], an extensible information security specification format acts as a compliance audit mechanism for enforcing business rules and information security policies based on audit trails.

2.2. Business Process Automation

2.2.1 Business Process Management

[Hammer1994] defines a *Business process* as "a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer". In our thesis, we focus on business processes that are sequences of automated or manual tasks that create, deploy and exchange electronic artifacts with information systems and people.

[Casati2000] identifies business process automation as a key-enabler for e-business. The enterprise must integrate and automate the operations of its backend systems that handle order fulfillment, procuring, billing, distribution and so on with the information systems of its suppliers, distributors and outsourcing partners to collaborate, share knowledge and align their business processes with changing market conditions. Automation of business processes can help eliminate error-prone manual human efforts, duplication of work and remove process inefficiencies or bottlenecks. This can help them cut down operating and transactions costs, speed up process execution and gain a competitive advantage in the market.

Business Process Management (BPM) systems have emerged to support business processes as information-processing. A Garner report [Laurence2007] has identified BPM suites as one of the fastest growing software markets and predicts the worldwide market for BPM suites to exceed $2.6 billion in 2011 driven by an increase in globalization, importance of consumer and rise of internet. [Wil2003] defines BPM as "Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, Organizations, applications, documents and other sources of information".

BPM marks a shift from data-oriented to process-oriented thinking in information systems. BPM have their roots in *workflow management systems (WFMS).* The workflow management coalition group defines WFMS as "a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications" [Lawrence1997]. Another definition describes WFMS as a "middleware system that provides a central point of control for defining business processes and orchestrating their execution" [Jablonski1996]. WFMS provides a process-aware information system to assemble and execute business processes out of existing artifacts.

While workflow management systems focus on the enactment or *automation* of business processes, BPM is wider in scope and encompasses the design, modeling, enactment, diagnosis and monitoring aspects of business processes. For example, BPM may include performance management or *business activity monitoring* (BAM) to enable organizations use metrics gathered by information systems for performance measurement and diagnosis of business processes. A Garner report [Gassman2006] describes Business Activity Monitoring as "a technology and a technique that provides real-time access to key business metrics. The reasons for deploying BAM are to monitor key business objectives, anticipate operational risks and reduce the time between a material event and taking effective action." It may include techniques for business process modeling. For example, there is Business Process Modeling Notation BPMN [White2004] which is a graphical notation to depict the end-to-end steps of the workflow in a business process that can be easily understood by business analysts, developers and managers. Similarly, [Weiss2005] shows how the User Requirements Notation (URN) can be applied to

business process modeling. It consists of two complementary graphical notations - a Goal Requirement Language (GRL) to capture business goals and Use Case Maps to depict business processes as scenario consisting of flows of responsibilities or activities. In [Pourshahid2008], the URN notation was extended to provide business process monitoring and performance management capabilities.

Another point of view is that BPM is a management discipline that applies IT to represent and improve business processes of the enterprise at a strategic-level, while WFMS is a technology that enables process automation. BPM may or may not use a WFMS for automation of its processes.

2.2.2 Service Oriented Architecture (SOA)

[Lublinsky2007] describes a SOA as an architectural style "promoting the concept of business-aligned enterprise service as the fundamental unit of designing, building, and composing enterprise business solutions". Thus, the key abstraction in a SOA is a service that is loosely-coupled and reusable. From a business perspective, SOA helps align IT with business by developing a core set of reusable business services to rapidly develop new solutions and expose them to their business partners. From an IT perspective, SOA presents a flexible architecture to design applications as a set of services that are loosely coupled to their underlying application's programming languages and operating systems. A SOA can be built using any technology that supports the creation of services. Recently standards-driven web services [SOAP, WSDL, UDDI] from W3C [WC3] and OASIS [OASIS] have emerged as the key implementation technology for SOA.

SOA is primarily an IT initiative while BPM is business-driven. However, they both have increasingly converged. Research from Forrester [Ken2006] highlights the new trend of using a SOA in BPM suites. A Garner report [Jim2006] predicts organizations must combine BPM with technology initiatives in SOA to remain competitive in the future. [Jasmine2005] explores how a SOA enabled BPM helps achieve business agility by creating a flexible integration platform for process automation. The key characteristic of such a flexible platform is that the logic of a business process is separated from the applications and resources that it accesses. [Woodley 2005] investigates the potential and challenges for convergence of SOA and BPM. They look at both how a BPM can be used to improve the execution of an organization's SOA initiatives and how SOA can be used to build a BPM solution.

2.2.3 Web Services

Recently, Web services have emerged as widely accepted industry standard for building a SOA. [WC3] defines web service as "a software system designed to support interoperable machine to machine interaction over a network". Web services allow organizations to publish an API over the network (usually over HTTP or HTTPS) and make it accessible using any programming platform. Web Services helps reduce the complexity of application integration through loosely-coupled and implementation-neutral services. The core Web Service technology stack consists of three XML-based standards namely SOAP, WSDL and UDDI [SOAP, WSDL, UDDI, Curbera2002]. Figure 1 shows how they work together.

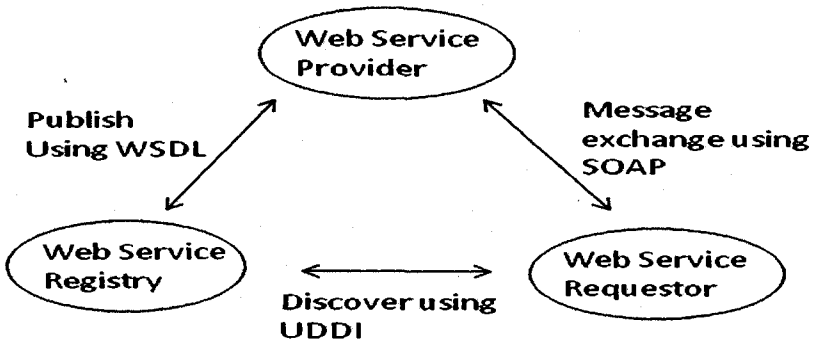


Figure 1 Web Services Model

In the diagram above, service providers advertise their web service by publishing its interface in the Web Service Description Language (WSDL) to a service registry. Service requestors locate new services from the Service registry using Universal Description, Discovery and Integration protocol (UDDI). Finally, Service requestors communicate with service providers using the Simple Object Access Protocol (SOAP) messages.

WC3 defines WSDL as a "An XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information" . WSDL serves as a contract between service providers and service requestors describing the capabilities of a web service and information required to locate it. It includes abstract details such as the web service data types (*types*), messages formats (*message*), operations performed by web service (*portType*) and concrete details such as address of service endpoint and binding information (transport protocols used by the web service).

SOAP is a W3C standard that defines a simple communications protocol for exchanging messages over a computer network. SOAP messages are XML documents that have a single *Envelope* element as its root element. Inside the *Envelope* element, SOAP schema defines an

optional *Header* element and a required *Body* element. The *Header* element can be used to send extra payload about the message such as security information, addressing information etc. The *Body* element consists of the application logic data such as the operation being invoked, input parameters being sent etc. Thus, SOAP by itself is a very simple protocol that simply defines an envelope containing two extensible containers for transmitting XML-based data. SOAP messages are sent over some underlying transport protocol, usually HTTP. This information is usually specified in the binding details of WSDL interface of a web service.

UDDI is an OASIS standard that describes an XML-based registry for storing information about web services. It allows businesses to publish their web service listings over the internet. It allows service requestors to query the registry for WSDL interfaces of web services. It's designed to communicate using SOAP messages.

A number of other web services and XML related specifications have been developed by various organizations and industry vendors. They address issues related to web service security, message addressing, message reliability, interoperability and are often complementary to each other. One such standard for web service security is WS-Security [Nadalin2004] from OASIS. WS-Security defines header extensions for SOAP messages to embed information for authentication, end-to-end confidentiality and end-to-end integrity. It is a flexible specification that provides mechanisms to reuse existing security token formats such as X.509, Kerberos, and SAML in SOAP messages. It also uses WC3 recommended XML-encryption [Imamura2002] specification for confidentiality. XML-encryption defines syntax and processing guidelines for encrypting XML text and specifying its cipher text, algorithm and keying information. It provides the ability to encrypt whole or just parts of a message. WS-Security also uses XML-

signature [Bartel2002], another recommendation from WC3 for representing cryptographic digital signatures inside XML documents. It defines procedures for computing and verifying XML signatures. Output of both XML encryption and XML signatures is well-formed XML syntax consisting of signature hash values, cipher text, key info etc. WS-Security simply combines the security token information, XML encryption and XML signatures into a WS-Security header element for SOAP messages. Another important WC3 standard is WS-Addressing [Box2004] that describes two constructs. First, it contains an XML-based structure called end-point reference (EPRs) to convey the information needed to address a Web service endpoint. Second, it describes SOAP header extensions for a set of transport-protocol independent message addressing properties to describe end-to-end addressing characteristics of a message. These include a *ReplyTo* that indicates where a response for an asynchronous web service request should be sent and a *MessageId* to uniquely identify a message. In order to promote interoperability amongst the different web service specifications, an industry consortium called Web Service Interoperability Organization [WS-I] has been established that provides different set of *profiles.* Each such profile provides implementation guidelines to developers for using a group of web service specifications at specific revision levels to build interoperable web services.

2.2.4 Business Process Execution Language (BPEL)

WS-BPEL [Arkin2005] from OASIS is an XML-based standard to define executable business processes exclusively out of web service interactions. BPEL evolved out of independent research efforts from IBM and Microsoft in web service composition, later combined and

submitted to OASIS for standardization. BPEL complements web services to implement a SOA. While web services provide the fundamental SOA abstraction of a *service*, BPEL provides abstraction of a *business process* by composing web services. BPEL supports automation of business processes both within an enterprise and across multiple organizations. However strictly speaking, BPEL is not a true workflow language since it does not have abstractions to support people and tasks. But such capabilities can be provided in a service-oriented fashion (such as a Task Service used in our thesis framework).

BPEL uses an *orchestration* paradigm as opposed to *choreography*. Under orchestration, the sequence of message exchanges between a process and other services is controlled by a central orchestrator (analogous to the conductor in a musical orchestration). On the other hand, choreography describes peer-to-peer interactions without centralized control (analogous to a dancing team). Moreover, orchestration defines executable processes whereas choreography defines abstract processes wherein each peer-to-peer interaction can be realized differently.

BPEL processes are defined using XML syntax. BPEL product vendors such as Oracle, Active endpoints, Sun (open-ESB project) and so on provide tools ranging from designer, management console and an execution engine that orchestrates the defined processes. Some vendors supply their own proprietary extensions for BPEL syntax and/or execution engine. BPEL has no standard graphical notation (different vendors provide their own notations). However, there has been previous work to map BPMN models of a process into BPEL code such as in [Ouyang2006]. BPEL was designed to conform to web services standards and thus uses WSDL interfaces, SOAP messages etc. A BPEL process itself is exposed as a web service endpoint, typically with an operation to initiate execution of a new instance.

BPEL syntax aims to support *programming in the large* which allows process designers to script a process with high-level programming abstractions for sending a message, waiting for a message and so on instead of writing its low-level implementation code. A BPEL process definition typically has three main parts described below.

First, there are declarations for *partner links*. A BPEL process communicates with external web services through partner link abstractions. A partner link is created for every web service the BPEL process interacts with. Every partner link maps to a *PortType* declared in the WSDL interface of either the BPEL process or partner web service. Further, each partner link has roles defined (*myRole* and/or *partnerRole*) that indicates which entity (BPEL process or partner service) implements and which entity calls the operations defined in that *PortType*.

Second, it consists of variable declarations. BPEL process uses variables to store the messages for every interaction with a partner link. These variables typically map to the input parameters and output parameters for invoking a specified operation in the partner link. These variables can be thought of as XML documents that are manipulated inside a process.

Third, BPEL process definition consists of an ordered sequence of BPEL *activities* that describe how the process should be orchestrated. An `<invoke>` activity with a specified partner link and operation is used to interact with web services. A `<receive>` waits for an incoming message from a specified partner link (it blocks process). A `<assign>` manipulates the XML documents contained in a variable using XPATH queries [XPATH1999]. It's commonly used to initialize the input for a partner link operation or copy contents from one variable to another. BPEL also supports a `<flow>` activity for concurrent processing, `<scope>` and `<sequence>` activities to structure the process into smaller logical units and

<faultHandlers> to handle exceptions. In addition, <case>,<switch>, <if>, <else>, and <while> activities are also supported to control the flow of process execution.

2.2.5 Business Process Execution Language (BPEL) Extensions

There has been related work in BPEL to add extensions for human interactivity, business activity monitoring and Service Level Agreements (SLAs) monitoring. In [Kloppmann2005], a BPEL4People extension layered on top of WS-BPEL is proposed. It allows means to assign generic roles to people and defines a new BPEL People activity to delegate tasks to a person. The tasks are based on accompanying WS-Human task specifications for BPEL4People. In [Holmes2008], architecture of VieBOP based on BPEL4People specifications is proposed to illustrate the integration of people activity into an arbitrary BPEL engine. Similarly, [Thomas2007] proposes a language to specify BPEL4extensions within a BPEL script and evaluates architecture to implement them within an existing BPEL engine. In [Bertino2006], the authors extend the BPEL specifications with RBAC-WS-BPEL and BPCL languages to associate users and their authorization constraints with activities during the execution of a BPEL process. In [Baresi2005], a business activity monitoring (BAM) extension for BPEL running on an underlying service-oriented middleware platform SOPware of Deutsche Post AG is discussed. Business process events as well as events from underlying middleware platform are collected to build "event clouds" to provide inputs for business activity monitoring where enterprise performance dashboards are used to indicate the performance of the processes. In [Barbon2006], an event monitoring architecture for BPEL processes is described. It supports "event monitors"

for both single instances of BPEL processes and aggregate information for multiple instances of a BPEL process. In [Rud2006], mathematical model is used to estimate the run-time quality of service metrics for BPEL orchestrations of a process and accordingly manage the Service Level Agreements between involved parties.

2.3. Federated Identity Management

2.3.1 Identity Management

An *entity* is a uniquely identifiable object in the real-world (such as a person) or digital world (such as a machine).Work on identity management [Parr2001,Casassa2003] generally defines *identity attributes* as characteristics associated to an entity and *identity* as a view or a subset of those attributes at a specific point of time for a specific context. These sets of attributes can encompass information about an individual's preferences, habits, interests, hobbies, public keys, homepage, and even context related information such as current geographic location. An entity can have several identities for different contexts. For example, a person "Bob" (an entity) can have several identities as a consumer on an ecommerce website, as an employee at work, as a citizen on a government portal and so on. An entity's identities and the associated contexts can change over time. In the context of privacy, three different levels of identities are explained in [Koch2005]. With a *veronymous* level of identity, it is possible to deduce the entity from the attributes of an identity. With a *pseudonymous* level of identity, a persistent identifier is used to reference an identity across different sessions but it is not possible to derive the entity from the identity attributes. Finally, an *anonymous* level of identity is valid for only a single session or single transaction with no prior knowledge about its identifiers or entity.

Identity management (IM) is concerned with the representation, provisioning and life-cycle management of identities for entities such as a user or an object in an information system.

In [Casassa2003], the scope of identity management solutions is categorized into three distinct areas. (a) Data representation of identity and a directory service for storage of identity data. (b) Control functions for authentication (verifying owner of an identity), authorization (allowing identity owners to define permissions for controlling access to their identity attributes by others) and auditing. (c) Management functions for provisioning and life-cycle management of user accounts associated with identities.

[Koch2005] argues the need for identity management in e-commerce and collaborative applications. [Casassa2002] identifies identity management as a key e-business enabler in B2C, B2B and G2C commerce:

a) B2C (*Consumer facing businesses):* Identity Management solutions enhance consumer's online experience through a single-sign on service (SSO) that enables them to access multiple participating merchant websites with a single set of login credentials. In addition, they can enhance consumer's privacy with secure storage and access of their personal information by online merchants.
b) B2B (*Enterprise):* The focus of traditional identity management solutions was the enterprise. Management of employee's identity such as new account provisioning and its administration were the key tasks of such a solution. With time, the IM solutions have matured to support identity management across cross-organizational B2B networks.
c) G2C (*Government to Citizens):* Identity management solutions can help government provide secure digital identification mechanisms for its citizens and clients. This is part of their wider goal of rationalization of all their services online.

2.3.2 Federated Identity

The problem of federated identity management is well studied in [Libera2003]. Historically, identity management solutions focused on the tightly controlled enterprise and used a centralized approach to store, authenticate and manage identities. With the emergence of new B2B networks linking the enterprise over the Internet to its suppliers, distributors, collaborators, content providers and outsourcing partners, the enterprise was required to authenticate and manage identities for users from other organizations and vice-versa. In addition, with the growing popularity of B2B integration driven by SOA, new techniques were required for identity management across organizational boundaries.

Federated Identity Management (FIM) introduces the notions of Identity Providers (IDP) and Service Providers (SP). Identity providers are trusted third-parties that link and manage an entity's multiple identities at different service provider's websites to form their network or "federated" identity. Once the identities are federated, IDP transmits the entity's authentication and authorization states to Service Providers who can verify the identity of the user and make access-control decisions about them. Clearly, the SP must "trust" the IDP for this to happen. The establishment of trust between the participating parties in FIM is a precursor to federated identities. A major FIM function is Single-sign on (SSO), wherein an IDP authenticates the user and transmits his identity to any Service Providers who wants to identify the user. Users get the convenience of having to login only once (with IDP) during a given online session. Some FIM solutions also allow users to securely store their personal information with the IDP and grant access to service providers if required. FIM may also provide a framework for the sharing of identity attributes between participating service providers in the federation. A key federated

identity technology is SAML and some popular FIM solutions are Liberty Alliance Project, WS-Federation, OpenId and Windows Live ID. They are discussed next.

2.3.3 Security Assertion Markup Language (SAML)

Interoperability is a critical requirement for any FIM solution since it operates across a heterogeneous cross organization domain. SAML [Hughes2004] is one such interoperable standard developed by OASIS for the exchange of authentication & authorization data between different businesses in heterogeneous environments. The basic idea is that different parties exchange XML messages as defined by SAML schema elements and in a specific fashion as defined by SAML profiles.

A key concept is that of SAML "assertions". Assertions are statements issued by Identity Providers that "attest" facts about the user such as:

- *Authentication Statement:* For example, Bob has logged in with IDP using plain text password.
- *Authorization statement*: For example, Bob has permission to access John's photographs.
- *Attribute statement*: For example, Bob's zip code is K1N 6l6. Service providers trust assertions issued by IDPs to make access control decisions. IDPs are also called *issuing party* while service providers are called *relying party.*

The SAML specifications define several "*Bindings*" and "*Profiles*". Profiles explain how to use SAML with web browsers, web services, smart clients and so on. Bindings explain how to transport SAML messages over different transport protocols such as HTTP, SOAP and so on.

A major use-case for SAML is Single-sign on (SSO). Another use-case is federated

identity management where service providers use SAML assertions to link or de-federate user's identity with IDP. Finally, SAML tokens can be used in WS-Security SOAP header extensions to transmit security information for web services. SAML uses other standards like XML encryption and XML signature for confidentiality and integrity of SAML assertions exchanged between IDPs and SPs.

2.3.4 Windows Live ID

Microsoft's Windows Live ID [Live2006] evolved out of its earlier Passport Service. The Passport Service was positioned to be a single sign-on service for all internet commerce but did not gain much adoption. It was criticized for its proprietary approach wherein Microsoft was the sole authority that acts as the Identity provider. Privacy concerns were also raised about Microsoft having full access to consumer identity and usage information.
Subsequently, it was revamped to develop Windows Live ID that is primarily used by Microsoft sites and applications such as Hotmail, MSN subscriptions services, Xbox, .Net Messenger services and so on. A Windows Live ID Web Authentication SDK was also released to enable third-party providers to leverage the large Windows Live ID user base into their web sites and rich-client applications.

2.3.5 OpenID

OpenID [Recordon2006] is a lightweight, decentralized, free single-sign on solution that emerged out of open source community efforts. It aims to simplify the online login experience for internet users. The basic rational is that anyone can choose to become an OpenID user or

OpenID provider for free without approval from any organization. The users can register with any OpenID provider of his choice that they trust. Recently, OpenID has become increasingly popular with Google, Yahoo, Microsoft, AOL and other leading vendors acting as OpenID providers and more than ten thousand OpenID enabled websites.

OpenID leverages the already established internet technologies such as URI, HTTP, SSL and Diffie-Hellman for single-sign on. URIs is used as a user's identifier. For example, Bob may have an OpenID Identifier `bob.openid.example.org` with his provider "example.org". The user submits his OpenID Identifier to a relying party's website. The relying party transforms it into a URL such as http://bob.openid.example.org to discover the user's OpenID Provider. Once the OpenID Provider is located, the relying party can use one of the two modes, namely "check_immediate" and "check_setup" to communicate with the Identity Provider for user authentication. In the former mode, the user submits his login credentials to the relying party which relays it to OpenID provider (i.e. user does not directly communicate with Identity Provider). In the latter mode, the user is simply redirected to Identity Provider for direct authentication. The latter mode is typically used.

While OpenId is lightweight, technology neutral and easy-to-use, it has been criticised for being susceptible to phishing attacks. Moreover, it can violate consumer privacy because a consumer's OpenId identifier is known to every website he visits making it possible to track his behaviour across multiple websites.

2.3.6 WS-Federation

WS-Federation [Libera2003] is a federated identity management standard jointly developed by IBM, Microsoft, Verisign, RSA Security and BEA systems. It provides a WS-Federation language that describes how different security domains broker authentication, security tokens and identity attributes. The major components of specifications include the IDP that acts as a Security Token Service (STS), an Attribute service (for sharing attributes) and a Pseudonym service (for preserving user's privacy by masking the real identity). In addition, it provides two profiles.

- A Passive Requestor Profile that describes how WS-Federation identity services can be used with HTTP web browsers and web-enabled mobile devices.
- An Active Requestor Profile that describes how WS-Federation identity services can be used with web services and other clients communicating via SOAP messages.

2.3.7 Liberty Alliance Project

The Liberty Alliance Project [Liberty] was established in 2001, by a leading consortium of 150+ organizations, to establish open standards, guidelines and best practices for federated identity management. A central concept to Liberty Alliance is a Circle of Trust (CoT) defined as "a federation of service providers and identity providers that have business relationships based on Liberty architecture and operational agreements, and with whom users can transact business in a secure and apparently seamless environment". In Liberty Alliance terminology Principals, Identity Providers (IDP) and Service Providers (SP) participate in a CoT. A Principal is an entity which can assume a federated identity that is managed by the IDP in the CoT. Liberty Alliance FIM consists of 3 key classes of specifications:

(A) ID-FF (Identity federation framework): [IDFF2003] defines specifications for *identity federation, defederation and single-sign on* using the Security Assertion Markup Language (SAML). Earlier versions of ID-FF extended SAML 1.1. More recently, SAML 2.0 and ID-FF have converged together. Using ID-FF, a principal can "federate" or link any local identities he maintains at a SP with his IDP. Once federated, a principal single-sign on and visit service providers in the CoT without having to login again during same session. This works because the principal's IDP is able to transmit their authentication state to the SP by means of SAML assertions.

(B) ID-WSF (Identity Web-service framework): This framework [IDWSF2004] consists of specifications for *creation, discovery, invocation* of inter-operable web-services *and permission based identity attribute sharing.* The focus of ID-WSF framework is on special web services (known as Liberty services) that share user data in a privacy-preserving fashion. These web services extend the SOAP header element with additional information as specified by the Liberty SOAP binding [LibertySOAPBinding2006]. The Liberty SOAP binding describes the purpose for the different headers and explains how they need to be processed in a Circle of Trust environment. A web service consumer must first discover a Liberty Service using the Discovery Service [LibertyDisco2006] before it is able to share user's identity attributes with it. In addition, an Interaction Service [LibertyInteract2006] is specified that communicates with principal on behalf of a service provider. This can be used for example to obtain consumer's consent. In addition, it describes a People Service [LibertyPeople2006] that enables secure access by a principal to another principal's identity attributes and a Data Services Template Specification [LibertyDST2006] that describes a standard protocol for querying and modifying principal's identity attributes with a Liberty Service. Note that ID-WSF is designed as a SOA

comprising of Liberty Services with Discovery, People and Interaction Services being special Liberty Services.

(C) ID-SIS: Specifications that describe data models for standard *interoperable identity services* such as personal profile, employee profile, contact book, geo-location service, etc.

A key feature of Liberty Alliance specifications is that it builds on already established industry standards. For example, it uses SOAP messages for communication between different services. WSDL is used to describe a Liberty Service interface. XML schemas are used to describe the data type elements. The Liberty SOAP binding extensively borrows headers from WS-Security, WS-Addressing and so on. The liberty security mechanisms [LibertySechMech2006] accommodate a wide variety of security technologies for use in a Circle of Trust. Typically, SAML assertions are extensively used to exchange security information. To do so, Liberty provides a SAML security token profile [LibertySecMechSAML2006] that describes how SAML assertions can be packaged as security tokens inside the Circle of Trust.

Finally, there are specifications, guidelines and research that investigate security and privacy issues in Liberty Alliance standards. A Liberty Alliance specification [LibertyPrivacy2004] outlines the security and privacy issues in ID-WSF framework and explains the potential security and privacy ramifications of the technology used in ID-WSF. [Varney2005] provides guidelines to address some privacy issues arising in deployment of Liberty enabled solutions in business-to-consumer context. [Alsaleh2006a, Alsaleh2006b] identifies and analyzes possible privacy breaches in Liberty Alliance ID-FF and ID-WSF frameworks.

2.4. Summary

In section 2.1, we analyzed the importance of building trust in B2B processes that deliver information-rich experiences to the consumers. In this thesis, we don't model or measure the level of trust users have in a B2B process. Rather, we describe a framework that addresses certain requirements in B2B processes which are assumed to help gain trust amongst its users. Consumer Privacy is about protecting access to their information from others and giving them control over its sharing. Legislations have been enacted to protect consumer privacy. In section 2.2, we differentiate between business process automation and business process management which is wider in scope. BPM includes Performance Management or Business Activity Monitoring and may or may not use business process automation. For our thesis, we use a SOA based on web services and BPEL for automation of the process and enhance it with additional infrastructure for business process management. In section 2.3, federated identity management is discussed as a key enabler in heterogeneous B2B networks and we introduced the Liberty Alliance federated identity standards.

Chapter 3. Integrated Trusted Processes Framework

In this chapter, we will describe our thesis approach to build a framework that will support business process automation in consumer-facing B2B networks. In section 3.1, we introduce our basic approach which is to integrate BPEL processes into a Liberty Alliance CoT. In section 3.2, we identify the requirements of a consumer-facing trusted B2B process that our framework must address. Finally, in section 3.3 we present our framework and briefly describe how its components work together to address the requirements we have identified.

3.1. Basic Approach

BPEL automation of a process in a SOA helps organizations closely align their information systems with their business processes using a standards-driven approach. BPEL has traditionally been focused on addressing business processes that operate within a single organization or enterprise boundary. We will be looking to extend BPEL to support business processes that operate across organizational boundaries in a B2B network. Within a single organization, processes can be built that connect isolated heterogeneous systems spread across different departments of an enterprise and thus facilitate EAI (enterprise application integration). Beyond the boundaries of a single organization, BPEL can help an organization collaborate with its business partners through B2B process automation that cuts across enterprise boundaries. In both cases, BPEL leverages the power of web-services driven SOA that enables organizations to

package their already existing application functionality as loosely coupled and platform-agnostic services. In addition, BPEL's XML syntax simplifies the task of business process development for process designers who work at a high level of abstraction by using a "programming in the large" model. The benefits of a SOA leveraging BPEL's XML-based process definition capabilities allows organizations to rapidly build and deploy agile processes that can react to changing business needs and environments.

Within the organization, BPEL processes operate in a single trust domain (Figure 2) where users are identified, authenticated, and authorized by services under the control of a single organization. Such processes serve users who belong to the same organization (example: an organizational process that offers payroll related services to its employees). The web services (that are part of process orchestration), the BPEL process definitions and BPEL Engine (that enact defined processes) are all managed by the same organization. It is also easier for the organization to monitor processes over which they have full control.

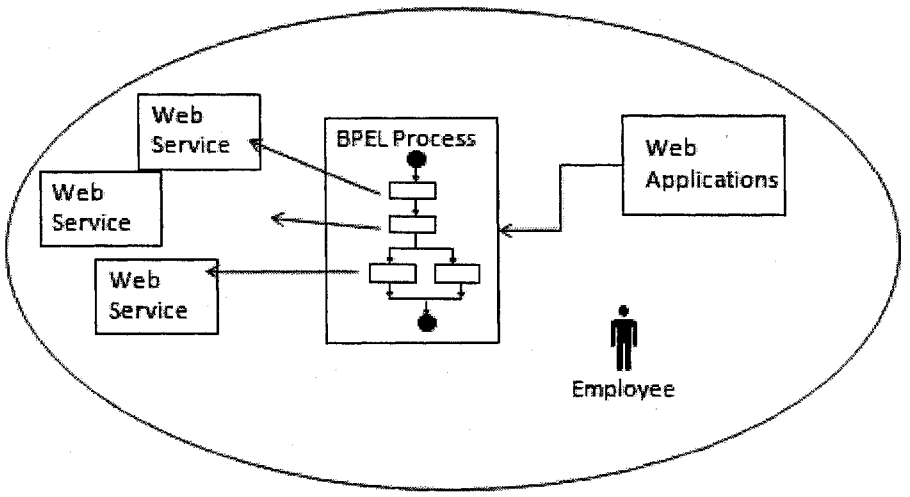


Figure 2 SOA using BPEL within organization

However, consumer-oriented collaborative B2B processes cut across multiple trust domains. They execute on behalf of consumers who exist outside any particular organization in the B2B network. Similarly, organizations that run the web services in SOA, design the BPEL process, administer the BPEL engine and so on could be different.

In such a cross-organizational environment, several unique issues arise which need to be addressed in the BPEL definition of such processes. Each organization may have its own proprietary and autonomous identity management infrastructure. Thus, this imposes challenges to manage a consumer's multiple identities in a consistent fashion across such a heterogeneous environment. Consumers have privacy concerns over the usage and sharing of their identity data and are less likely to adopt processes that share such data in an ad hoc fashion between organizations. Sharing of consumer data between B2B businesses is subject to privacy legislations. Hence, the process is required to document its compliance with consumer privacy legislations. In addition, mechanisms must exist that allow measurement of key performance indicators in process that operate in a distributed B2B environment. Similarly, the process needs a standardized approach to interact with employees in different B2B organizations. However, the WS-BPEL standard currently does not address the issue of modeling human participation into a BPEL process definition.

Thus while the use of SOA-based BPEL automation promises potential for consumer-facing collaborative B2B processes, it also poses several new challenges that must be resolved before they can be fully trusted by consumers and managed by the B2B businesses.

As a first step to address such issues, we leverage federated identity management inside our B2B network. We look at how consumers and collaborating businesses inside our B2B

network can be reorganized into a Liberty Alliance Circle of Trust. As introduced in section 2.3.7, Liberty Alliance is a leading consortium of leading industry vendors and consumer-facing enterprises that have established open standards and set of specifications for federated identity management. A key concept in Liberty Alliance is a Circle of Trust (CoT) which is a B2B network of cooperating enterprises that provide integrated services to consumers based on the Liberty Alliance Architecture. The main goals of a CoT are to allow businesses to share data while protecting identities and give consumers greater control over the sharing of their data. Figure 3 illustrates a Liberty Alliance CoT.

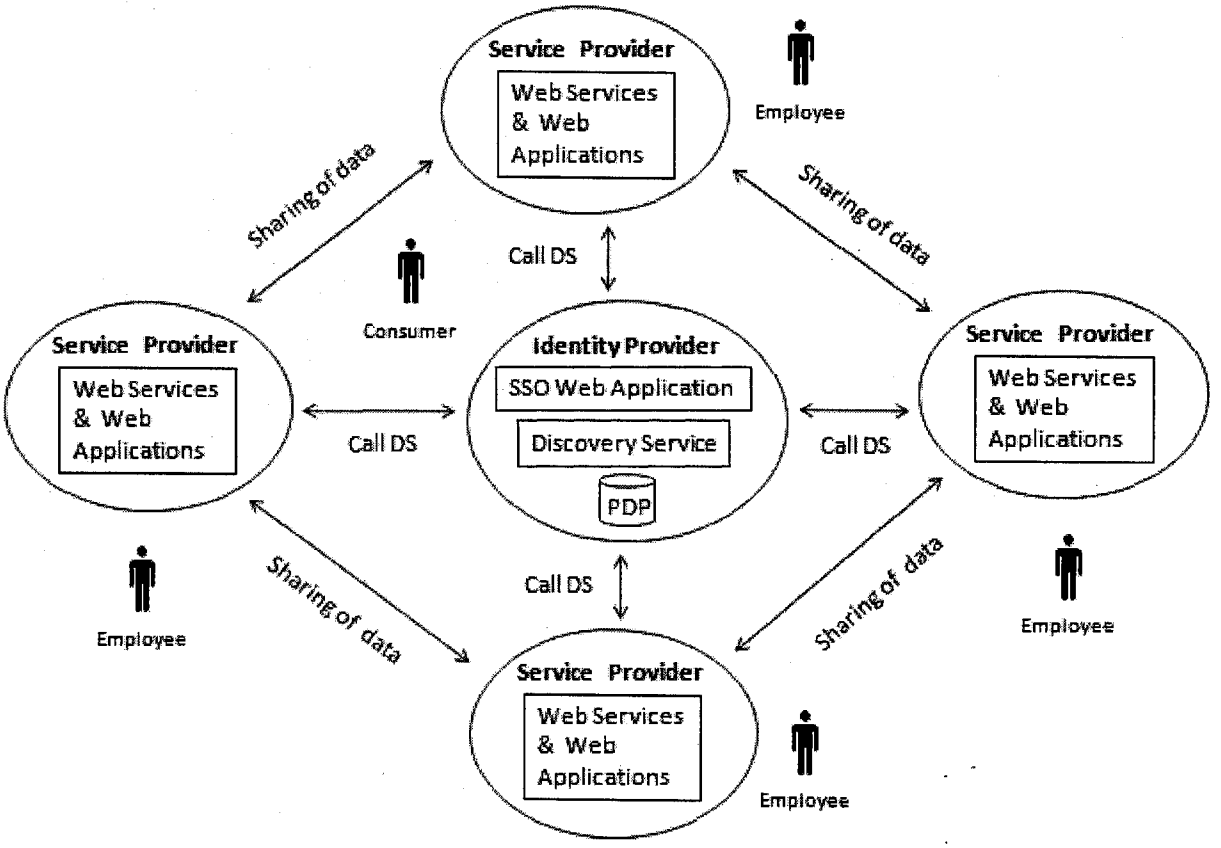


Figure 3 Liberty Alliance Circle of Trust

In the above diagram, the *Identity Provider (IDP)* is a special trusted authority that manages a consumer's federated identity and single-sign on across the CoT. The *Service Providers* are B2B businesses that deliver services to consumers on their websites. IDP broker's sharing of consumer data between service providers and ensures that proper consent from consumer was obtained. As shown above, the Service provider first calls an IDP component *Discovery Service* [LibertyDisco2006] in order to share consumer data with other service providers. All these interactions happen in a service-oriented fashion, i.e., service provider and Discovery Service leverage web services standards. Liberty Alliance standards [LibertySOAPBinding2006] specify the structure of the SOAP messages used in the CoT architecture. It defines extra header blocks that must be processed by a Liberty Service Provider to work in a CoT.

In the CoT, it is assumed that Service Providers have business relationships and operational agreements established between them. A special trust relationship exists between Service Providers and the IDP such that Service Providers trust or rely on claims made by IDP about a consumer's identities, their authentication and authorization state. IDP makes such claims by issuing SAML assertions to Service Providers.

Inside the CoT, only the IDP knows the authenticated identity of a consumer. In other words, the real-world entity that owns a CoT identity is known to the IDP. Each Service Provider is assigned a persistent pseudonym identifier through which they uniquely recognize a consumer and store their identity attributes. Based on this pseudonym alone, the service provider cannot dereference the entity that it represents but can still differentiate between different consumers who visit their website. Thus, two different levels of identities are used for consumers in the CoT (veronymous at IDP and pseudonymous at Service Provider). This makes sense because a

consumer is often not required to reveal any personal information to obtain services from a Service Provider. This is analogous to real-world situations wherein the same person implicitly switches between different levels of identities depending on the people he is dealing with.

In addition, a Service provider's has no knowledge about the consumer pseudonyms with other service providers in the CoT. This implies that no two service providers can directly share consumer data since they cannot cross-reference their local pseudonyms. Instead, a service provider must first obtain an End-point reference (EPR) [LibertyDisco2006] from the Discovery Service (DS) before calling another service provider. This EPR contains a security token that conveys the consumer's relevant pseudonym to the service provider who is being called. Such a token is encrypted by DS so that the pseudonym is not revealed to the service provider who is making the call. Thus, EPRs enable the sharing of data between service providers without allowing them to cross-reference or match their pseudonyms. This prevents any collusion attempts by fraudulent service providers who may otherwise try to share unauthorized consumer data. In addition, DS contains a Policy Decision Point (PDP) where consumers can express their privacy preferences and restrict the sharing of their data through access-control policies. The DS first consults the PDP to check for proper permissions before it issues any EPR to a service provider.

By leveraging the Liberty Alliance CoT architecture in our approach, we address consumer's privacy concerns regarding the sharing of their data. The system of pseudonymous identifiers helps protect a consumer's identity at a Service Provider's website. Discovery Service along with PDP enables the sharing of consumer data between service providers without allowing cross-referencing of local pseudonyms. This helps build consumer's confidence and

gain their trust. In a trusted environment, Service Providers are more likely to acquire consumer data and participation into services that require sharing of their data.

Our approach is to add business process support into such a Liberty Alliance CoT. We investigate how a BPEL definition of a process can be integrated into a CoT to handle Liberty SOAP binding, the Discovery Service interactions, EPRs and security tokens so that it can conform to the Circle of Trust mechanisms for managing identities and sharing data. A new organization called Trusted Process Provider is introduced into CoT to help service providers manage their BPEL process automations. It provides an Audit Trail Service, Event Stat Service and a Task service to address the previously discussed issues that arise in consumer-oriented B2B networks that operate across multiple trust domains.

Section 3.2 refines our concept of a trusted process by eliciting different requirements that our framework must address. Finally in section 3.3, we introduce our integrated trusted processes framework and describe how it meets those requirements.

3.2. Requirements for Consumer-oriented B2B Trusted Processes

Consumers are unlikely to adopt information-rich services unless they believe service providers are respecting their privacy while sharing their data and obtaining their consent. The consumer should be allowed to control which identity attributes can be shared and with whom. Consumer's identity should be protected when their data is being shared between B2B businesses. In situations where the user has not specified any explicit access-control policy, the process must first obtain the consumer's consent before sharing their data. Additionally, consumers may like to have the convenience of single-sign on (SSO) wherein once authenticated

they can access different service providers in the B2B without having to login again. This can simplify the login credentials they have to remember.

In addition, service providers must ensure that their processes are in compliance with consumer privacy legislations. To do so, they must document their use of consumer's data and allow privacy officers to investigate any potential privacy breaches. Additionally, service providers need means to provide transparency and build accountability into their data-sharing activities beyond the minimum requirements of law in order to gain consumer's confidence.

Also since a business process closely reflects the strategy of the organizations, it is important to monitor and measure the key performance indicators of a process in order to improve and manage it. Finally, a process workflow must be able to interact and coordinate with people during the course of its execution. Such interactions can be with employees of a service provider or even consumers themselves. Table 1 summarizes the different requirements.

Table 1 Requirements of a Trusted Process

Requirements	Description
Consumers	
Privacy	Protect Consumer Identity, Control data-sharing, Obtain consents
Convenience	Simplified set of login credentials and single-sign on authentication.
Service Providers	
Privacy Compliance	Document compliance with consumer privacy legislations
Transparency	Build consumer confidence by showing them how their data is being shared beyond the minimum requirements of law
Monitoring	Measurement of key performance indicators for process and bring end-to-end visibility to business activity events
Interactivity	Support human interactions in a process

3.3. Integrated Trusted Processes Framework

Figure 4 illustrates our integrated trusted processes framework that uses the basic approach described in section 3.2. We will first explain the different components shown in that diagram. After that, we illustrate how a BPEL process would interact with each of these components.

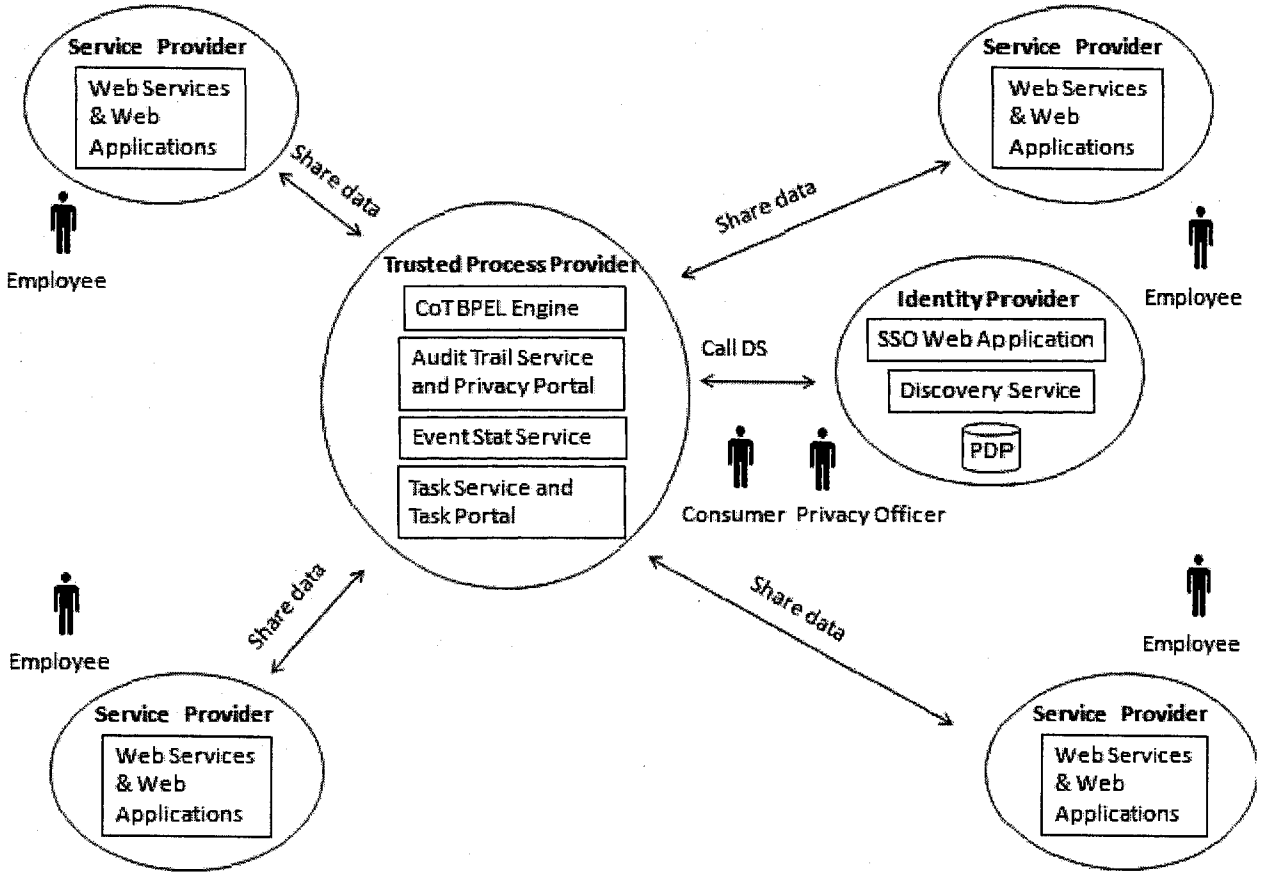


Figure 4 Integrated Trusted Processes Framework

As introduced before, the Identity Provider is a trusted organization that manages the federated identity of the consumer by authenticating them and federating (or linking) CoT services on their behalf, with their permission, through a system of pseudonymous identifiers.

The Discovery Service (DS) enables any two service providers to share data on consumer's behalf without allowing them to cross-reference the consumer's local pseudonyms. The consumer can limit such data-sharing by specifying access-control policies at the DS Policy Decision Point.

The framework introduces a new organization called "Trusted Process Provider" (TPP) to support business processes inside the CoT. It is another special trusted authority inside the CoT (but operates independently of the IDP). While an IDP helps manage a consumer's federated identity, TPP helps service providers manage business process automation in a CoT. It runs a CoT BPEL engine which is a trusted environment where service providers can orchestrate their process on behalf of consumers with whom they interact. TPP also offers common infrastructure services that help service providers manage their process:

- Audit Trail Service and Privacy Portal
- EventStat Service
- Task Service and Task Portal

An Audit Trail Service is provided to log the data-shared by the B2B process for documentation of privacy compliance. Every time service providers exchange consumer's identity attributes in the B2B process, a new event is logged into the Audit Trail Service. By doing so, a consolidated historical log or an "audit trail" of all data-sharing activities across the CoT can be built. Privacy officers can use these logs to verify privacy compliance of any organization and investigate any potential privacy breaches. In addition, these logs also provide a platform for service providers to provide transparency to consumers by showing them how their

data is being used or shared. TPP runs a privacy portal where privacy officers or consumers can view their audit trail records from ATS.

The ATS recognizes the consumer through its own unique pseudonym (just like any other Service Provider). Thus, the audit event logs only contain references to consumer's pseudonyms. In addition, the *values* of identity attributes that are shared are not logged. Instead, only the *field names* of the attributes are stored. This ensures that no personally identifiable information is recorded in an audit event and thus prevents the ATS from de-referencing any pseudonymous identities stored in audit trail records. This is required to make sure that ATS is not a privacy breach itself.

The Event Stat Service is another common service that helps service providers monitor and measure the performance of a cross-organizational process. It uses a simple event logging mechanism wherein business activity events are logged to ESS during the runtime of process. Thus, in a way the ESS acts as a global store for business events across the CoT. Service Providers can run reports against ESS to calculate key performance indicators of their processes. No generic Performance Management Portal is provided, as it is expected that the reports and measures used to manage performance will vary from process to process and organization and organization and so there will be a variety of custom made portals. The ESS will provide basic access to the raw events that any authorized performance management portal can access. ESS can also be used to record events pertaining to Service Level Agreements (SLAs) established between Liberty Service providers. For example: events can be logged to keep track of the number of times a request is made to a Liberty Service by a business process during a given timeframe. Note that while the audit trail service logs events pertaining to data-sharing activities of a consumer, ESS logs events for business activity monitoring.

Finally, the TPP provides a Task service (TS) to coordinate process steps that require human interactions. Human activity is integrated into a process by assigning tasks to different people such as employees, consumers and so on. Using a complementary Task portal (TP), people can perform actions on tasks assigned to them. For example: a Loan Application process could assign an "Approval Task" to a Bank Manager that requires him to review and sign customer's loan offer.

Table 2 Framework Components

Component	Based on	Function
IDP (Organization)	Liberty Alliance ID-FF specifications	Federated Identity Management and single-sign on
Discovery Service	Liberty Alliance ID-WSF specifications	Enable system of pseudonyms to support trusted data-sharing between services and ensure user consent through a policy decision point
Trusted Process Provider (Organization)	Added by our framework (uses WS-BPEL process automation)	Manage execution and monitoring of trusted processes defined in BPEL
Audit Trail Service	Added by our framework (implemented as a Liberty web service)	Audit Trail to document Privacy Compliance
Privacy Portal	Added by our framework(implemented as a Liberty web application)	Interface to view Audit Trail for Privacy Compliance
Event Stat Service	Added by our framework(implemented as a Liberty web service)	Log and aggregate events to support performance management
Task Service	Added by our framework(implemented as a Liberty web service)	Manage process steps which require user interaction.
Task Portal	Added by our framework (implemented as a Liberty web application)	Interface to view and update the status of tasks assigned to users

The task service can also be used to dynamically obtain the consumer's consent for sharing his data. In such a situation, the process can assign a task to the consumer that requires him to

explicitly respond whether or not his data should be shared. The Task Service also maintains its own unique pseudonym to recognize the user (separate from Audit Trail Service). Table 2 summarizes the different components of our framework and where they come from.

Note that the CoT BPEL Engine does not automatically start the execution of a new process instance on a scheduled basis. Instead, only service providers can trigger the initiation of a new process instance in response to a consumer's request or action made on their website. All trusted process *instances* are thus effectively "owned" by the service provider that initiates it. While the same trusted process can have multiple instances started by different service providers, each instance is owned only by its calling service provider. During process orchestration, the pseudonym of the consumer used by the CoT BPEL Engine is determined by the Service Provider that owns the process instance (i.e. it uses owner's pseudonym while interacting with other Service Providers, Discovery Service etc in CoT). For example, if consumer "Bob" visits Service Provider "A" website which starts a trusted process "X" and later visits Service Provider "B" website which starts the same trusted process "X", the CoT BPEL Engine orchestrates those two instances using their respective owner's pseudonym for Bob. No pseudonyms are assigned to the CoT BPEL Engine itself.

On the other hand, Audit Trail Service and Task Service are both assigned unique pseudonyms for each consumer since both store data related to them (i.e. audit trails and list of tasks).It is important that the Trusted Process Provider be kept separate from the Identity Provider. IDP knows the consumer's real identity for any given pseudonym. TPP runs consumer centric processes that share their information (ATS especially has complete audit trail records). Together, the IDP and TPP can track consumer's data and behavior across CoT tied to their real

identity which is a privacy breach. Thus, the organization that runs TPP should be independent of the organization that runs the IDP.

We will now show how a trusted BPEL process interacts with the different components of IDP and TPP inside a CoT.

3.3.1 Single-sign on with IDP

Before the service provider can start a trusted process on the CoT BPEL Engine, it needs to identify who the consumer is. Figure 5 below shows how the consumer is authenticated using a single-sign on mechanism inside a CoT.

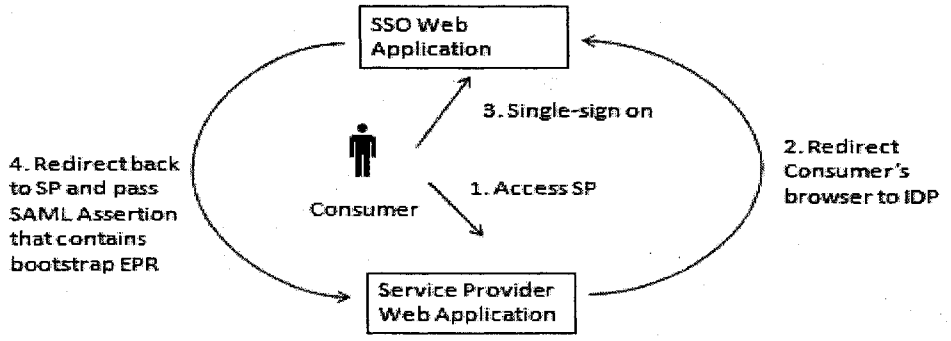


Figure 5 Single-sign on in CoT

The above interactions are based on Liberty Alliance ID-FF specifications which use the SAML 2.0 Web Browser Profile [Hughes2004]. The Service Provider first redirects the consumer's browser to IDP for authentication. IDP checks if the user has already authenticated before. If not, he is asked to login with his single-sign on credentials at IDP. It then redirects the consumer's browser back to the service provider website and passes a SAML Assertion as part of SSO process. Using this SAML Assertion, the service provider decrypts its unique pseudonym for the

consumer to identify them. The assertion also contains a special End-point reference called "Bootstrap EPR" that enables the service provider to call DS on consumer's behalf.

Once SSO is complete, service provider can start a new trusted process instance at the CoT BPEL engine on behalf of the consumer as shown in Figure 6.

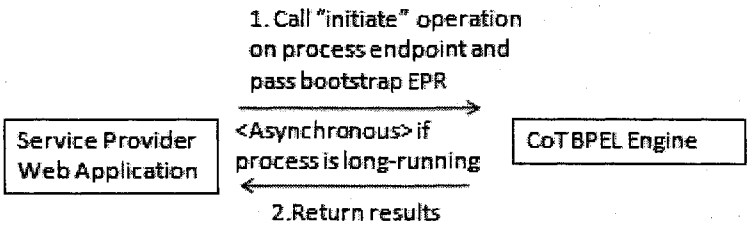


Figure 6 Service Provider calling a Trusted Process

The Bootstrap EPR that was obtained during SSO is passed as input. The process can utilize it to call DS and obtain EPRs for communicating with other service providers on consumer's behalf. In cases where the process execution may take a long time (such as interactive processes with human tasks), the process endpoint is called asynchronously. Note that while interactions in Figure 5 are HTTP interactions, Figure 6 uses SOAP messages (typically over HTTP binding).

3.3.2 Discovery Service

Figure 7 below illustrates how a trusted process calls other service providers to obtain consumer data. First, the process calls the DS to request an EPR for the service provider. As part of the request message, it passes a security token (issued by IDP) contained in the Bootstrap EPR to Discovery Service. DS then consults with PDP to ensure that the user has granted permission for

sharing of his data between the service provider (that initiated the process) and the service provider whose EPR is being requested .Finally, DS returns back a new EPR to the process.

The process can now call the SP to obtain consumer's data. Again, the security token contained inside the EPR issued by DS is passed off to SP. SP decrypts the token to extract its local pseudonym for the consumer and accordingly return his data. Notice that DS encrypts the EPR token issued to process to prevent it from dereferencing the consumer's pseudonym at the Service provider.

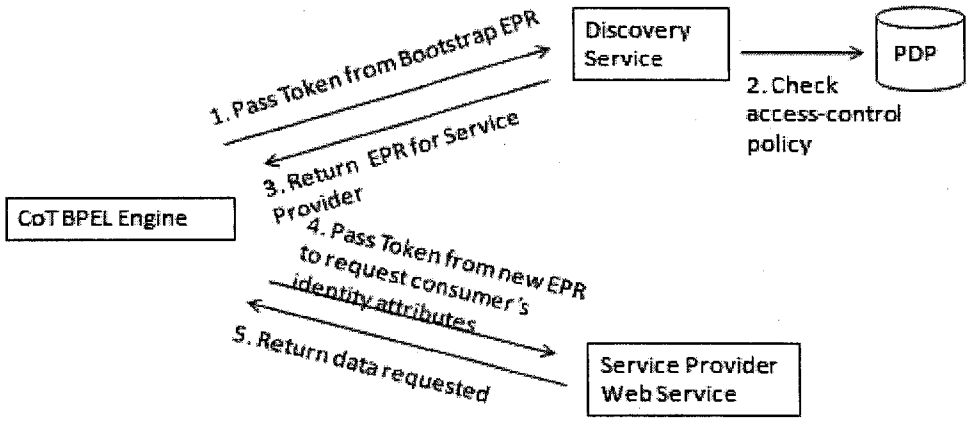


Figure 7 Sharing consumer data using Discovery Service

3.3.3 Audit Trail Service and privacy portal

Figure 8 illustrates how a trusted process utilizes the Audit Trail Service (ATS) for compliance.

The process cannot directly log an audit event to ATS. This is because even ATS has its own unique pseudonym for the consumer using which it associates the audit event with that consumer. Accordingly, the process first obtains an EPR for ATS from Discovery Service. It then logs the event to ATS by passing the token contained inside the ATS EPR along with the

names of the service providers who shared the data, the field names of the identity attributes that were exchanged and the timestamp. As emphasized earlier, identity attributes values that were shared are not passed to ATS for privacy concerns. The ATS decrypts its local pseudonym for the consumer and creates and persists the audit event to its data store.

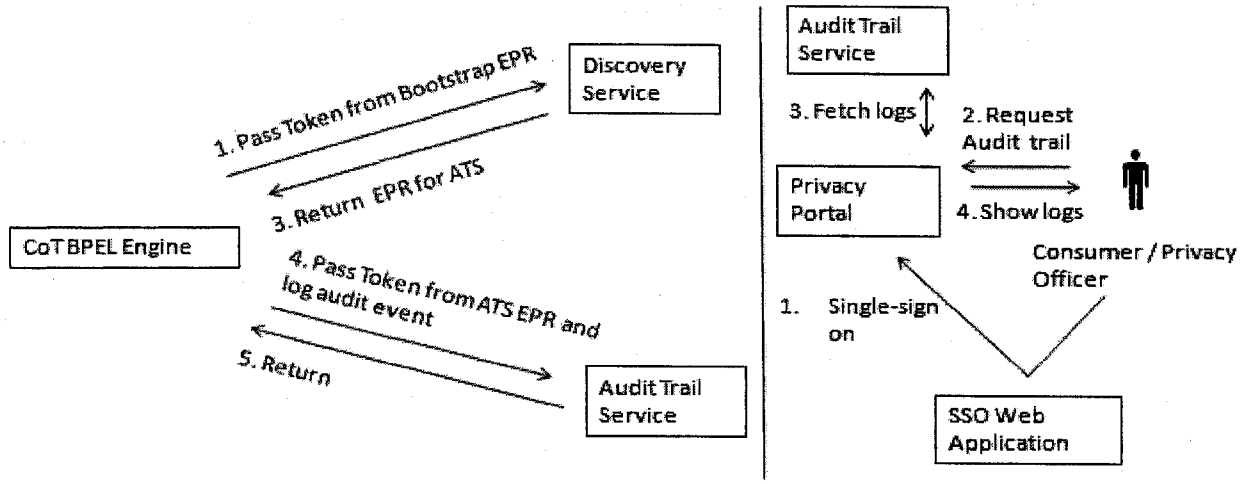


Figure 8 Documenting and verifying privacy compliance

The right hand side of Figure 8 shows how a consumer (or privacy officer) accesses the privacy portal to run reports on the audit trail records. The consumer (or privacy officer) must first single-sign on with IDP before requesting any information from the Privacy Portal. Note that both ATS and privacy portal use the same pseudonym to identify the consumer (or privacy officer). A proper access-control mechanism is required at Privacy Portal to ensure that consumers can view only their own audit trail records. On the other hand, the access control mechanism should allow the privacy officer to access audit trail records for any consumer.

3.3.4 Event Stat Service

Figure 9 below illustrates the use of Event Stat Service. In this thesis, we assume that business activity events are not associated with any particular identity. This holds true if businesses are measuring KPIs at an aggregate level such as click-through rates or number of loan applications broken down by specific consumer segments. Hence, business activity events are logged directly to ESS on a non-identifying basis. However, care should be taken that no personally identifiable information is logged as part of a business activity event.

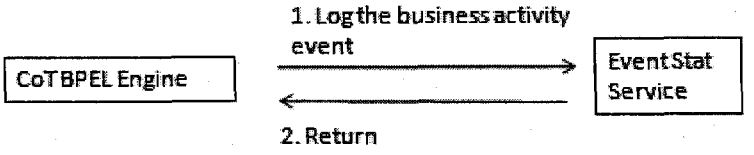


Figure 9 Logging events to ESS for process monitoring

3.3.5 Task Service and task portal

Next we show how a trusted process can manage steps that require interaction with a user. Figure 10 shows one such example where a process dynamically obtains consumer's consent for the sharing of his data.

The process first gets an EPR for the Task Service through the DS. Using this EPR, it calls the Task Service to assign a new "consent" task to the consumer. As usual, token contained inside the EPR is passed to the Task Service. In addition, information about consumer's identity attributes that the process wishes to share with a Service Provider is also passed as input to the Task Service. The Task Service resolves its unique pseudonym for the consumer through the token and accordingly creates a new task for him.

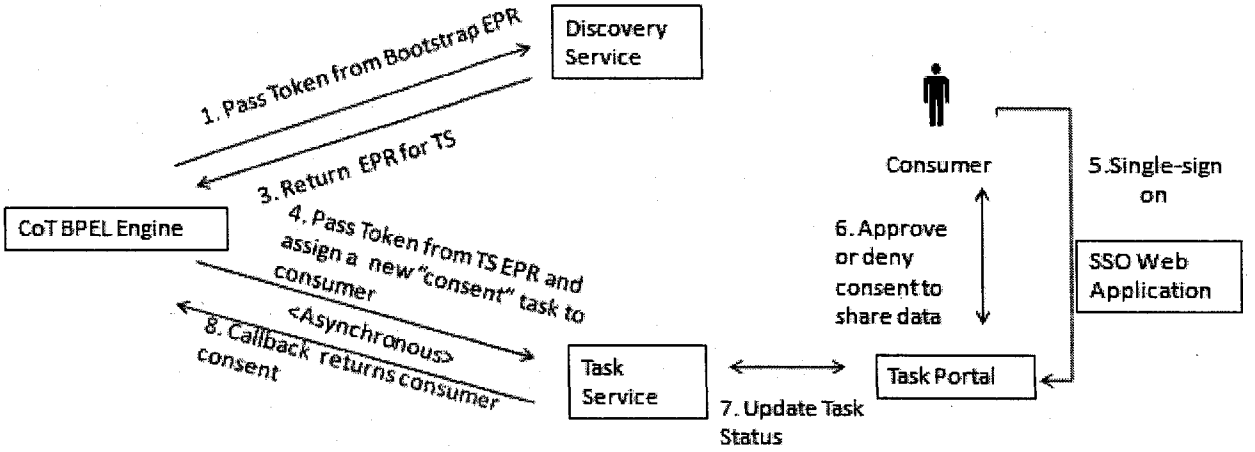


Figure 10 Managing human interactions in trusted process

The consumer visits the task portal to act upon this task. The task portal provides employees of service providers and consumers with a standardized visual interface to view and process their pending tasks. Accordingly, the consumer uses it to complete the "consent" task that was assigned to him by granting or denying consent to share the requested identity attributes. Next, the Task portal notifies TS about the completion of task. TS returns back the user's decision back to the BPEL process. Note that both TS and task portal share common pseudonyms to identify the users. In addition, asynchronous interaction was used for communication between the trusted process and TS since a completion of a task can take several hours. The asynchronous message exchange pattern is one of the features supported by BPEL standards. Using it, the process shown above makes a non-blocking request to the Task Service. It then waits for a response from task service. Between the asynchronous request and response, the process is free to execute any other activity (however the process that's shown simply waits).

Chapter 4. Case Study

Two different use-case scenarios involving consumer-oriented B2B processes were used to validate our integrated trusted processes framework. For both the scenarios, a "strawman" BPEL implementation was used to compare it with our integrated trusted processes framework implementation. In the first "MusicWeb" scenario, a B2B process is used to personalize a catalog webpage displayed to users based on their music preferences. In section 4.1, we introduce this use case scenario. In section 4.2, we show our strawman implementation of MusicWeb scenario and analyze its shortcomings. This is followed by integrated trusted process framework implementation of MusicWeb Scenario which is described in Section 4.3. The second "Car Loan" scenario looks at a more complex workflow involving steps that require human interactions in order to further validate our framework (section 4.4). In section 4.5, we describe the CoT prototype, design tools and runtime environment used in our implementation.

4.1. Music Web Scenario

We will now introduce the MusicWeb scenario. Figure 11 illustrates the high-level interactions. MusicWeb is a B2B network of businesses offering music-related services to end consumers. There are 4 main service providers in this network. **eShare** is an online social networking website where consumers can upload and share music content. The consumers are able to "tag" such content with special keywords like genre, artists, album and other such information that describe the music. Over time, eShare is able to accumulate detailed consumer

profiles that keep track of their music preferences based on the tags of content they frequently access. **eTunes** and **iMusic** are online music stores affiliated to MusicWeb network.

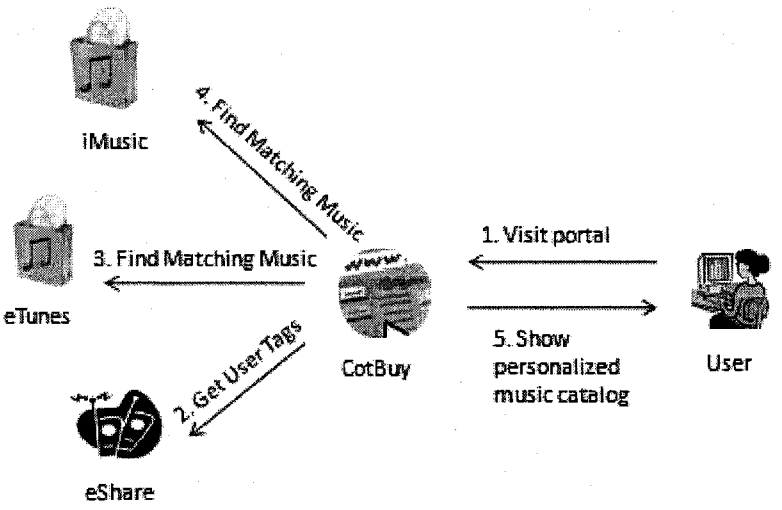


Figure 11 MusicWeb Scenario

Finally **CoTBuy**, another service provider runs an online shopping portal that acts as a centralized gateway for consumers who want to buy music from MusicWeb B2B network. The consumer can log into the portal and browse through a catalog of music items from MusicWeb stores such as eTunes and iMusic. CoTBuy uses a B2B process to personalize the music catalog shown to the consumers based on their data found at eShare. The B2B process works as follows:

(1) The consumer logs into CotBuy portal and clicks on a link to access music catalog

(2) The process first obtains the most popular music tags for that consumer from eShare.

(3) and (4) searches for music items in eTunes and iMusic catalogs that closely match those tags.

(5) A personalized music catalog webpage is returned to the consumer based on the information obtained in steps 3 and 4.

4.2. A Strawman BPEL Implementation of the MusicWeb Scenario

In this section, we present a first-cut BPEL implementation of the MusicWeb Scenario without the use of any Liberty Alliance Circle of Trust components. This is a Strawman implementation that reflects the BPEL automation of the MusicWeb process in a "plain-vanilla" SOA. We use it as a starting point for the integrated trusted processes framework implementation of the same process. An overview of the BPEL architecture is first presented in section 4.2.1. This is followed by a discussion of the BPEL process definition that was built in section 4.2.2. In section 4.2.3, we critique it to illustrate the issues that arise with BPEL processes in consumer-facing B2B networks.

4.2.1 BPEL Architecture

Figure 12 illustrates the high-level architecture. In our Strawman implementation, we do not use any federated identity management techniques. Each consumer maintains separate local user accounts with different MusicWeb Service Providers and each service provider locally authenticates the consumer. A SOA exposes the functionality of MusicWeb B2B partners. Three web services were created as follows:

- **eShare Service:** It has an operation "*getUserTags*" that can be used to obtain the consumer's most popular music tags. The consumer's userid at CoTBuy is passed as input. (eShare uses this information to cross-reference the consumer's corresponding eShare userid)
- **eTunes Service:** It has an operation "*getMusicByTags*" that takes music tags as input and returns eTunes music items that match those tags.

- **iMusic Service:** It has an operation "*getMusicByTags*" that takes music tags as input and returns iMusic music items that match those tags..

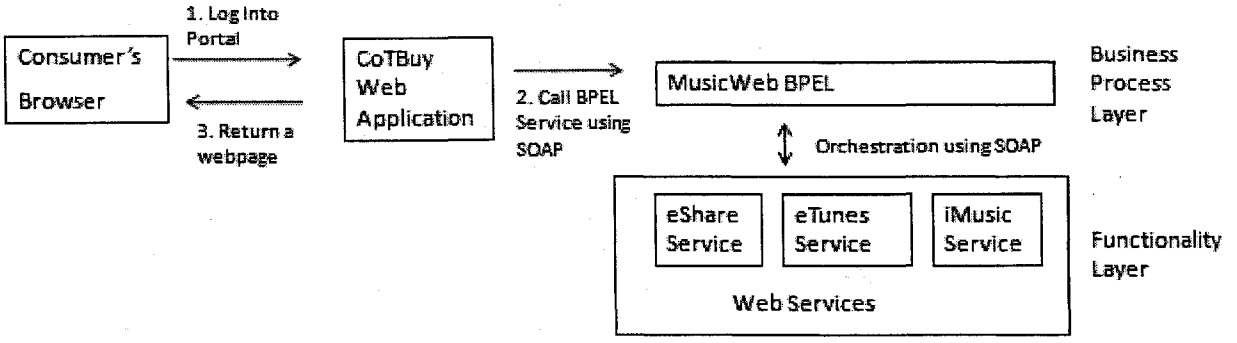


Figure 12 BPEL Architecture for MusicWeb Scenario (Strawman)

The MusicWeb process was implemented as a BPEL orchestration over these services. It has a single operation defined in its SOAP endpoint. The CotBuy portal invokes this operation to initiate the execution of a new process instance. The consumer's CoTBuy userid is passed as input and a list of music items is returned back. Section 4.2.2 below briefly describes this BPEL process.

4.2.2 BPEL Process Definition

Figure 13 illustrates a visual model of the BPEL process that was defined. The process has 4 partner links (one for the client i.e. CotBuy portal and three for the MusicWeb network services). Variables are declared to store the input and output messages for each partner link service. The variables are defined in the same XML Schema namespace as that of the corresponding input or output message defined in WSDL interface for a partner service. The sequential steps in this process are as follows:

(a) *<<Receive>>:* The process execution starts when it obtains the consumer's CoTBuy userid as input. (b) *<<Assign>>*: The process stores this id into the input variable for eShare service (c) *<<Invoke>>*: eShare service "*getUserTags*" operation is called to obtain the consumer's popular music tags (d) In the left branch, the user's tags obtained in step c are placed into input variable for iMusic. The iMusic Service "getMusicByTags" operation is then called to fetch matching music items. (e) In the right branch, the user's tags obtained in step c are placed into input variable for eTunes. The eTunes Service "getMusicByTags" operation is then called to fetch matching music items (f) *<<Assign>>:* combines the list of music items obtained in last 2 steps. (g) *<<Reply>>:* returns back the results to the client.

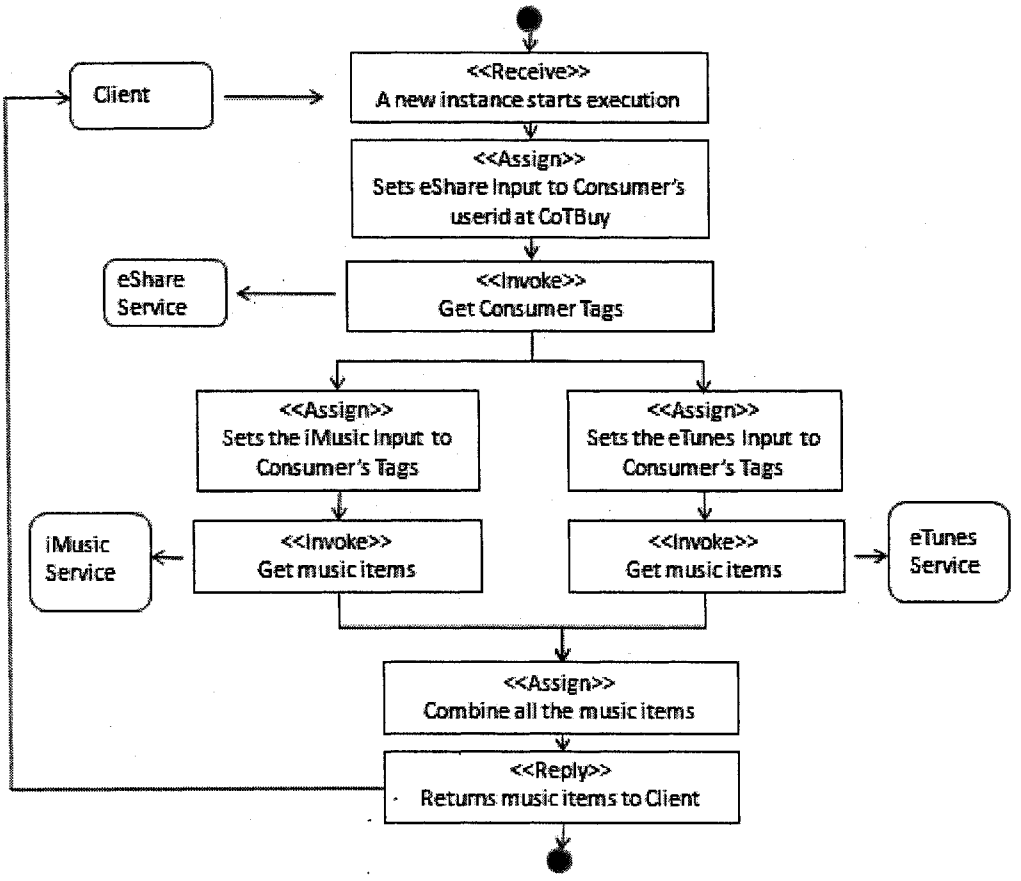


Figure 13 BPEL definition for MusicWeb Process (Strawman)

4.2.3 Analysis

Our implementation supports MusicWeb process automation as a composition of services. The functional requirements of the scenario were met. However, we notice that:

- The consumer has to remember multiple login credentials for the different MusicWeb service provider websites. A consumer logged in at eShare would need to authenticate again if he were to visit CoTBuy.
- The consumer's identity at CoTBuy is revealed to eShare for the sharing of his data. The two service providers are required to cross-reference the consumer's local identities with each other in order to facilitate data-sharing.
- This also raises a possibility of collusion between service providers. For example, eShare and CoTBuy can build consolidated consumer profiles that track their behavior on the eShare social networking website tied to their home address information revealed to CoTBuy website during an order shipment process.
- Consumer Privacy legislations would apply to MusicWeb Service Providers. Our above architecture and process definition does not document compliance.
- There is no support to measure key performance indicators of a business process with the current infrastructure. For example, assume that MusicWeb service providers want to keep track of "Click-through rates" for the different music items returned by MusicWeb process and displayed on catalog webpage broken down by vendor information (eTunes or iMusic).The above architecture lacks support to monitor the events required to calculate such process metrics.

4.3. ITPF Implementation for MusicWeb Scenario

We now use our integrated framework to implement the MusicWeb Scenario. In section 4.3.1, we will show how the MusicWeb B2B network works in a Liberty Alliance Circle of Trust. Section 4.3.2 describes the single-sign on interactions. This is followed by an in-depth discussion of the trusted BPEL automation of the MusicWeb process (section 4.3.3). Finally, we summarize the improvements in MusicWeb process obtained using our framework (section 4.3.4).

4.3.1 ITPF Architecture

Figure 14 presents a high-level overview of the architecture. MusicWeb B2B network now functions as a Liberty Alliance Circle of Trust. New trust relationships are introduced between MusicWeb service providers and IDP.

Liberty Federated Identity standards are leveraged to manage consumer's identity in MusicWeb B2B network. IDP assigns unique pseudonyms for each consumer to every service provider. For example, a consumer "Bob" would be known as "Bob_eShare" to eShare, "Bob_CotBuy" to CotBuy and so on (In real-world Liberty deployments, the pseudonym could be a more obscure numeric values. For the sake of simplicity we use easy-to-understand textual pseudonyms identifiers here). Discovery Service enables sharing of consumer's data between eShare and CoTBuy while protecting his identity. The consumer can control the data-sharing between eShare and CoTBuy at PDP.

In addition, the Trusted Process Provider and its services are introduced to manage the MusicWeb process. Audit Trail Service allows the MusicWeb process to document the sharing

of consumer data between eShare and CoTBuy for privacy compliance. Event Stat Service is added to measure "Click Through rates by Vendor Store", which we assume to be one of the key performance indicators that CotBuy, eTunes and iMusic are interested in.

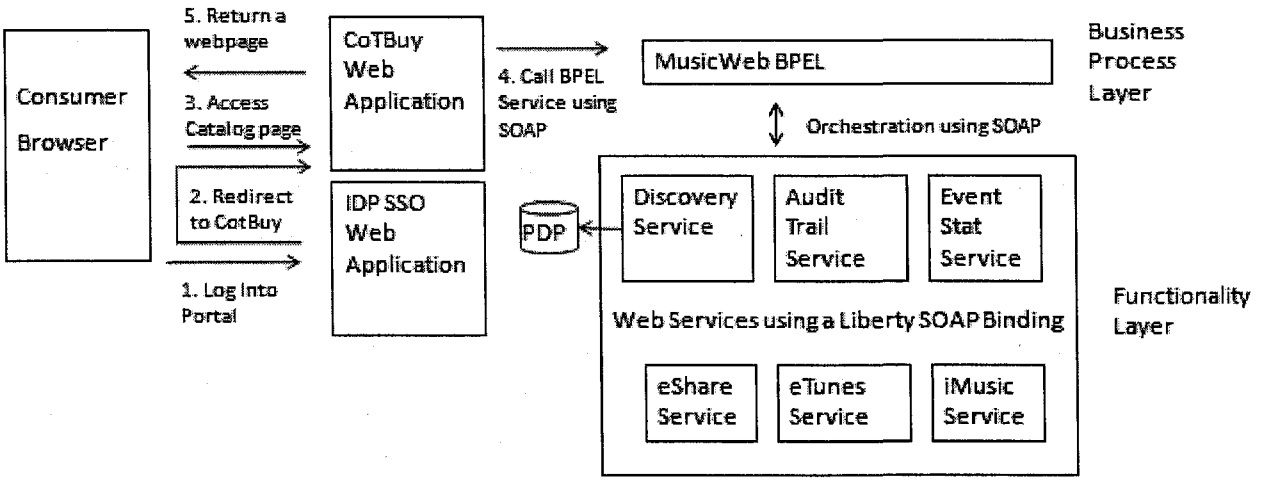


Figure 14 Integrated Framework Architecture for MusicWeb Scenario

The three MusicWeb web services introduced in section 4.2.1 are modified to process SOAP headers defined in Liberty SOAP Binding. This is required to integrate them into CoT. eShare service no longer requires the consumer's CoTBuy userid as an input parameter. Instead, it recognizes the consumer by de-referencing his pseudonym from the encrypted security token that is passed to it.

The consumer only needs to remember his login credentials for IDP. The IDP SSO web application acts as a single point of authentication for all MusicWeb consumers. As shown above, the consumers single-signs on with IDP and is redirected back to CotBuy portal. As part of this SSO process, CotBuy obtains the bootstrap EPR which is passed on to a new MusicWeb BPEL instance it initiates. These SSO details are explained next (section 4.3.2) and the new trusted BPEL process that was built is shown later (section 4.3.3).

4.3.2 Single-sign on interactions

Figure 15 shows the detailed SSO interactions for MusicWeb scenario.

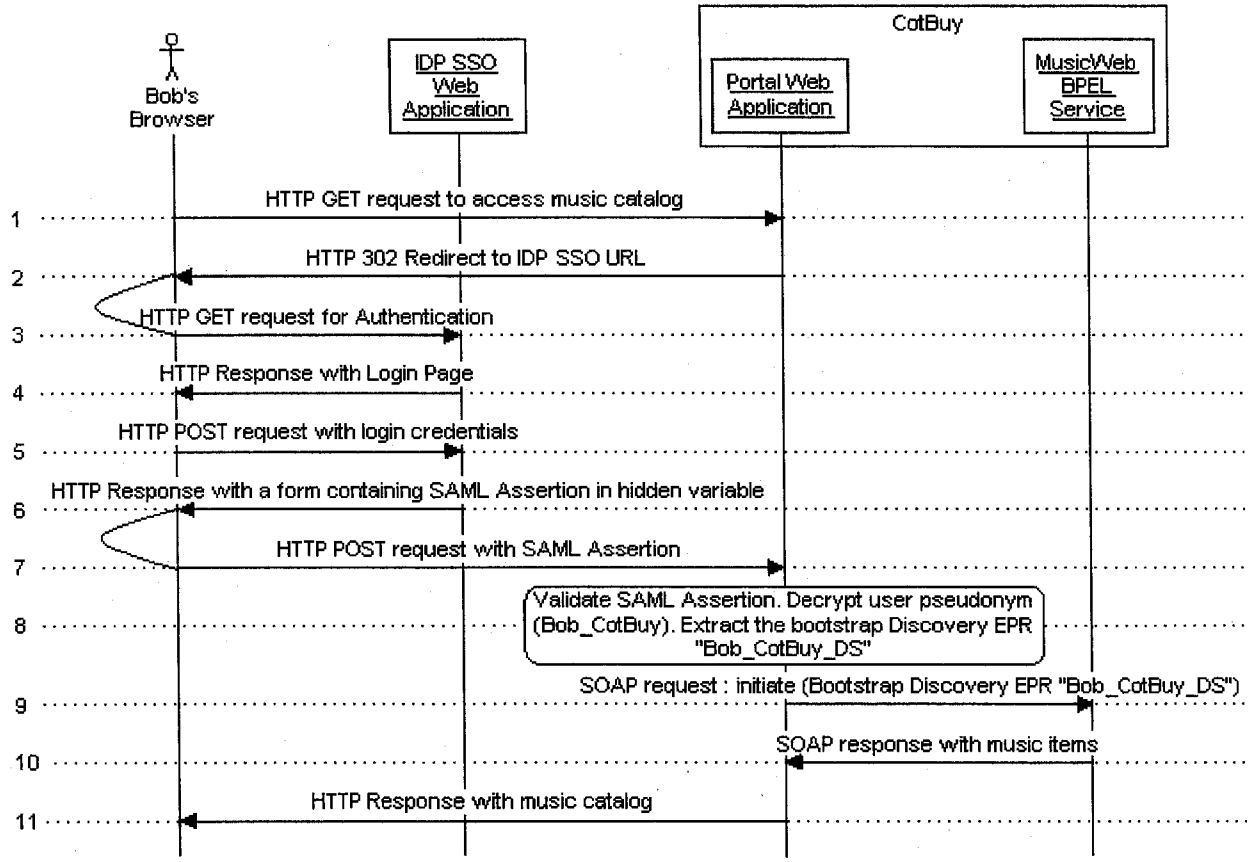


Figure 15 Single-sign on Interactions with CotBuy Portal

Here, a user "Bob" is redirected to IDP SSO web application when he visits the CotBuy portal. The IDP checks whether or not the user has already authenticated. Since Bob has not yet SSO, he is prompted to login. Bob fills out his IDP login credentials and submits the form. IDP establishes a user session and returns an HTML page containing a form with its "Action" value set to CotBuy portal and a hidden parameter containing a SAML Assertion.

The form gets auto-submitted (using javascript) to CotBuy portal. CotBuy thus obtains the SAML Assertion from IDP. Notice that all interactions so far are sent as HTTP messages. Figure Figure 16 below illustrates an example SAML Assertion issued by IDP to CotBuy.

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Version="2.0"  ID="sxJu9g">

<saml:Issuer>http://musicweb.idp.com<saml:Issuer>
<saml:Signature> <!-- signature by the issuer over the assertion --></saml:Signature>

<saml:Conditions>
      <saml:Audience>http://cotbuy.com</saml2:Audience>
</saml:Conditions>

<saml:Subject>
      <saml:EncryptedID>
      <xenc:EncryptedData>U2XTCNvRX7Bl1N</xenc:Encrypted Data>
      <xenc:EncryptedKey> <!-- CotBuy's Public Key --> </xenc:EncryptedKey>
      </saml:EncryptedID>
</saml:Subject>

<!-- describes the authentication event -->
<saml:AuthnStatement AuthnInstant="2008-03-20" SessionIndex="6345789">
      <saml:AuthnContext>
      PasswordProtectedTransport
      </saml:AuthnContext>
</saml:AuthnStatement>

<saml:AttributeStatement>
      <saml:Attribute>
      <!-- Bootstrap Discovery EPR issued goes here-->
      <!-- Used by CotBuy to access Discovery Service on subject's behalf-->
      </saml:Attribute>
</saml:AttributeStatement>

</saml:Assertion>
```

CotBuy decrypts it to find out user's pseudoynm (Bob_CotBuy)

CotBuy passes this EPR "Bob_CotBuy_DS" to BPEL process

Figure 16 Example SAML Assertion issued by IDP to CoTBuy

The assertion shown above uses a simplified version of the OASIS schema for SAML 2.0 assertion. The *Issuer* and *digital signature* values are validated by CotBuy to ensure the authenticity of the assertion. It also contains an *Authentication Statement* that asserts details about Bob's authentication event with IDP. *EncryptedID* contains Bob' unique pseudonym at CotBuy ("Bob_CotBuy"). The assertion also has an *Attribute Statement* that contains a

Bootstrap EPR (let's call it "Bob_CotBuy_DS"). We will explain the details of this EPR in the next section.

4.3.3 BPEL Process Definition

Now we describe the trusted MusicWeb BPEL process that was defined for the MusicWeb scenario. Three new partner links are introduced for Discovery Service, Audit Trail Service and Event Stat Service. Additional data manipulation is required in <*Assign*> blocks to set the SOAP headers required to call a Liberty Service Provider. In order to handle SOAP headers in a BPEL script, we used extensions from the Oracle's Process Manager (PM) BPEL Engine that was used in our implementation. A detailed analysis of WS-BPEL standards with regards to our framework is presented later in Chapter 5. We will sequentially go through the different parts of the orchestration logic in the process. The shaded steps in the diagrams indicate the extra steps inserted into BPEL definition illustrated earlier in Figure 13 to make it work in our integrated framework.

(A) Starting the execution of a new process instance:

A <<*Receive*>> Activity kicks off the process execution as shown in Figure 17. The Bootstrap EPR ("Bob_CotBuy_DS") is passed as an input parameter by the Client (i.e. CotBuy web application) and stored into the input variable for BPEL process.

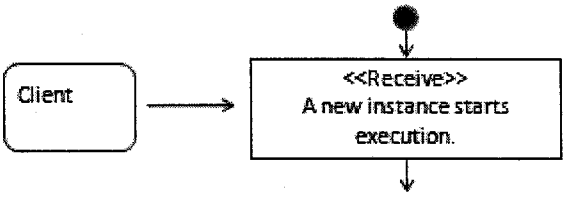


Figure 17 Starting the MusicWeb Process

An example bootstrap EPR 'Bob_CotBuy_DS' is shown in Figure 18. EPR's issued by DS are similar, except that they contain SAML Assertions issued and signed by DS.

```
<wsa:EndpointReference
notOnOrAfter="2005-08-15T23:18:56Z" >

<wsa:Address>http://localhost:8080/CotBuy-DS-Context/DiscoveryService</wsa:Address>

<ds:ServiceType>DiscoveryService</ds:ServiceType>

<sec:Token usage="urn:liberty:security:tokenusage:2006-08:SecurityToken">

      <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      Version="2.0" ID="re4xt2">
            <!-- SAML Assertion was here . Not shown to save space -->
            <!-- It is issued and signed by IDP -->
            <!-- It contains Encrypted Subject Id that resolves to Bob's psuedoynm at DS -->
            <!-- Also has an authentication statement from IDP -->
      </saml:Assertion>

</sec:Token>

</wsa:EndpointReference>
```

Figure 18 Bootstrap EPR "Bob_CotBuy_DS"

The *Address* value can be used to dynamically resolve the location of Discovery Service in the CoT. However, for sake of simplicity we don't use this feature (we assume static endpoint addresses that are already known beforehand) .The *Security token* contains a SAML assertion iss-ued by IDP. MusicWeb process presents this token to DS while calling it. In a way, the token *authorizes* CotBuy (owner of process instance it started) to call DS on consumer's (Bob) behalf.

(B) Obtaining the consumer's music tags from eShare Service:

Next, we fetch consumer (Bob's) tags from the eShare service. Figure 19 illustrates how it's done. An EPR obtained from DS is utilized to pass security tokens to eShare while calling it.

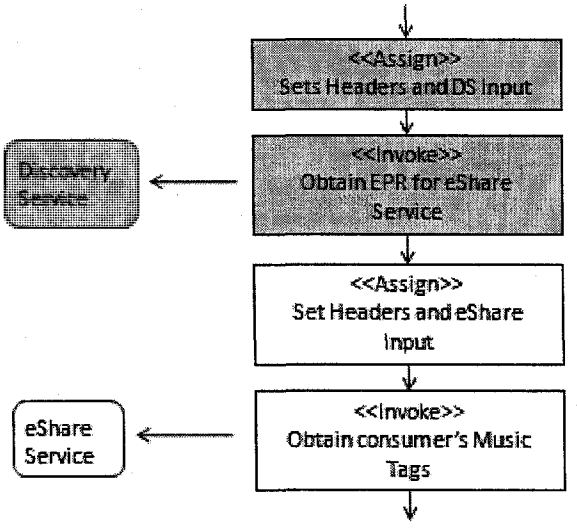


Figure 19 Obtaining consumer's tags from eShare service

The different steps are as follows:

(a) <<*Assign*>>: Sets the DS input variable along with Liberty SOAP headers. These header values include *WS-Security token* (SAML assertion inside Bootstrap EPR 'Bob_CotBuy_DS'), *sbf:framework* (indicates the version of Liberty Alliance ID-WSF specification) and *sbf:sender* (set to CotBuy). DS input variable values correspond to different parameters in the DS *Query* operation described in [LibertyDisco2006]. For sake of simplicity, we just use a single parameter *ServiceType* ('eShare') that is sufficient to indicate a request for eShare Service EPR.

(b) <<*Invoke*>>: Calls the DS *Query* operation to request an EPR ("Bob_CotBuy_eShare") for calling eShare service on consumer (Bob's) behalf. Before issuing the EPR, the DS consults the PDP to check for consumer (Bob's) consent.

(c) <<*Assign*>>: Sets the eShare input variable along with Liberty SOAP headers. SOAP headers values are similar to those shown in step a. *WS-Security token* header is set to SAML Assertion contained in the EPR "Bob_CotBuy_eShare" that was obtained in step b. eShare input variable is empty as *getUserTags* operation has no input parameters (since we no longer pass in userids).

(d) <<*Invoke*>>: Calls the eShare *getUserTags* operation to request the consumer (Bob's) music tags. Tags returned back are stored into eShare Output variable.

(C) Logging the sharing of consumer tags to Audit Trail Service

The sharing of consumer (Bob's) data between eShare and CotBuy in the MusicWeb process is documented for privacy compliance. Figure 20 illustrates the additional steps inserted into our process to log a new audit event with the Audit Trail Service. The steps are explained below.

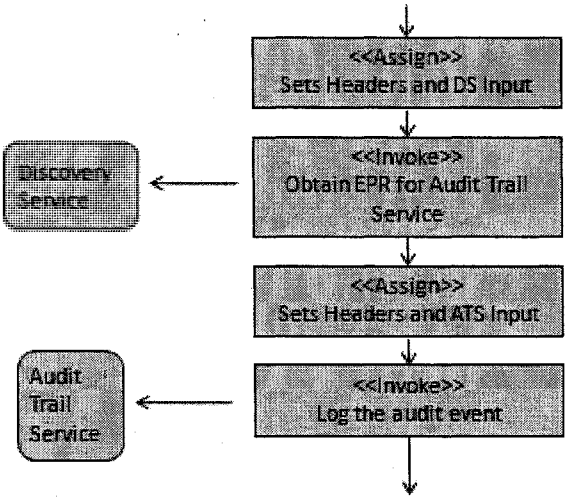


Figure 20 Logging the sharing of consumer tags to Audit Trail Service

(a) <<*Assign*>>: Fills in the DS Input variable and its SOAP headers. The SOAP Header values are similar to those set in our pervious call to DS (to obtain eShare EPR). The *ServiceType* parameter in DS Input Variable is set to "AuditTrailService" to request an EPR for ATS.

(b) <<*Invoke*>>: Calls the DS *Query* operation to obtain the EPR ("Bob_CotBuy_ATS") and stores it into DS Output Variable.

(c) <<*Assign*>>: Fills in the ATS Input Variable and its SOAP headers. *WS-Security token* header is set to the SAML token contained in EPR "Bob_CotBuy_ATS" from DS output variable. Other headers are also set as shown earlier. The ATS input variable describes the audit event that needs to be logged. Accordingly, it sets the 3 parameters for *logEvent* operation – *AttributeProvider* ("eShare") , *Timestamp* (set by Oracle BPEL in-built function) and *AttributeName* ("MusicTags"). The Attribute Requester and consumer's identity are not passed as input. This is because the *sbf:sender* header already has information about the service provider ("CotBuy") who requested the attributes. Similarly, the SAML Assertion passed as *WS-Security* header contains the pseudonym for the consumer whose data was shared.

(d) <<*Invoke*>>: Calls the ATS *logEvent* operation to create & log the new audit event.

The ATS has a standard event structure as shown in step c. The value for *WS-Security* header is extracted from the End-point reference that was obtained for ATS from the Discovery Service. The values for *Attribute Provider* and *sbf:sender* correspond to the names of the service providers who own the process and with whom the process shared consumer data respectively. *Attribute Name* is derived from the field names of the consumer identity attributes that were exchanged. The *Timestamp* value is dynamically set by Oracle BPEL engine to indicate when the

data-sharing occurred. Together, these event fields describe what data was shared, when and with whom.

(D) Measuring click-through rates (CTR) using Event Stat Service.

We now demonstrate how the Event Stat Service can be used to capture business activity events in a trusted B2B process to help service provider's measure performance. In MusicWeb scenario, we will employ it to calculate click-through rates (CTRs) for music items displayed to user on catalog webpage and broken down into vendor segment (eTunes or iMusic in our scenario). In order to do so, two CTR events are logged into ESS - item impression and item click. Each such event would record the music item that was returned and its vendor store information. CoTBuy can then generate reports on ESS to calculate CTRs for any given pair of music item and vendor store as the number of item clicks divided by number of item impressions.

The ESS can also log events related to tags obtained from eRadio to build a complete performance management picture of CTR broken down by vendor store and by Tag. To illustrate logging to the ESS, however, we will focus simply on the CTR events by vendor store.

First we log an event for each music item recommended by MusicWeb music vendors (based on a consumer's tags) inside MusicWeb process. This will help us determine how many times a music item from a particular vendor store was shown to MusicWeb consumers on catalog webpage (i.e. no. of item Impressions). The shaded steps in the Figure 21 were added to our process after invoking the iMusic and eTunes services.

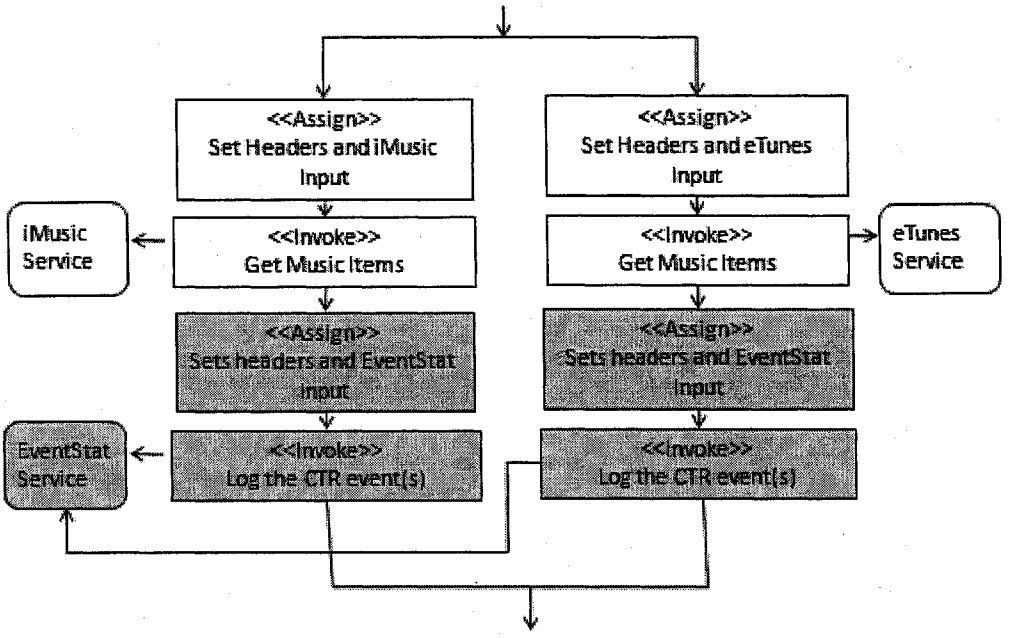


Figure 21 Logging CTR event (item impression) to Event Stat Service

Notice that iMusic and eTunes services can recommend music items that match music tags from eShare without having to know the identity of the consumer to whom those tags belong. Hence, the process is not required to obtain any EPRs from DS to call them. Similarly, no EPRs are required to log events to ESS as no identities are stored inside a CTR event. The four steps in Figure 21 are as follows (c and d are the shaded steps):

(a) <<*Assign*>>: Consumer's tags from eShare output variable are copied into the input variables for iMusic and eTunes Services. The *WS-Security token* value in the header is simply left empty. Other header blocks are set as shown earlier.

(b) <<*Invoke*>>: Calls the iMusic and eTunes services and stores the music items returned into their respective output variables.

(c) <<*Assign*>>: The *WS-Security token* value in the header is left empty. Other header blocks like are set as shown earlier. The EventStat *logEvent* operation accepts as input the list of music items (for each of whom an event is logged), type of event (CTR impression or CTR click) and the *music vendor* (eTunes or iMusic). Accordingly, we copy the list of music items from eTunes or iMusic output variable (depending on the flow branch) into the Input Variable for EventStat Service. In addition, we set *type* as 'CTR_impression' and *vendor* as 'iMusic' or 'eTunes' (depending on the flow branch).

(d) <<*Invoke*>>: The EventStat Service *logEvent* operation is invoked to store one or more 'CTR_impression' events (depending on the number of music items passed).

Next we log a CTR click event every time a music item is clicked by a consumer browsing his personalized catalog webpage This will help us determine how many times a music item from a particular vendor store that was shown to MusicWeb consumers on catalog webpage was clicked (i.e. no. of item Clicks). Figure 22 below shows a separate process used to capture these events from the catalog webpage when the consumer clicks on a music item. The music item that was clicked and its vendor information are passed as input parameters.

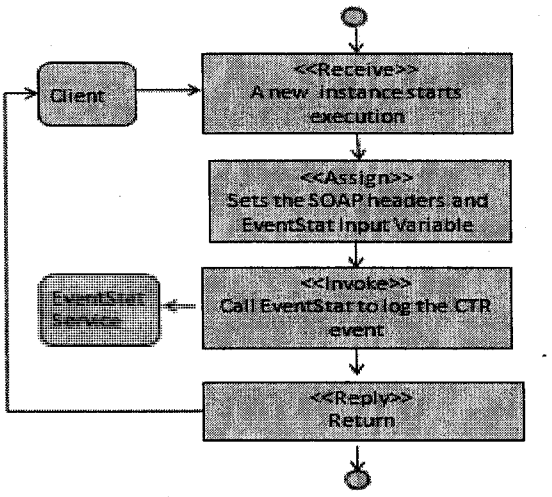


Figure 22 Logging CTR event (Item clicks) to Event Stat Service

Steps are as follows:

(a) <<*Receive*>> kick-starts the process and stores the music item and vendor information into input variable for BEL.

(b) <<*Assign*>> The *WS-Security token* value in the header is left empty. Other header are set as shown earlier. The *music item* and *vendor* from BPEL input variable is copied into the ESS Input Variable. In addition, we set *type* as 'CTR_click'.

(c) <<*Invoke*>> Calls the logEvent operation on EventStat Service to log a new CTR click event.

(d) <<*Reply*>> ends the process.

Unlike ATS, ESS doesn't have a standard event structure. This is a natural consequence of the fact that each service provider may have distinct requirements for the information that needs to be logged for different types of business activity events. However, the ESS can still act as a single global event repository for the entire CoT by providing different operations in its interface to capture different types of events (For example, an operation logCTRevent , logTagEvent, logSearchEvent and so on)

(E) Returning back the music items found to CotBuy Portal

Finally once the music items are obtained from the eTunes and iMusic services and CTR(s) event is logged, we return back the results to the client. Figure 23 illustrates the final steps of our trusted process. An <*Assign*> first copies the music items from eMusic and iMusic output variable into the output variable for BPEL. A <*Reply*> sends back those values back to the client (This is return value for the *initiate* operation that is called by CoTBuy website)

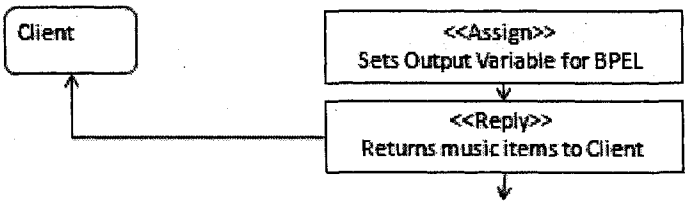


Figure 23 Ending the MusicWeb process

4.3.4 Analysis

MusicWeb B2B network now works as a Liberty Alliance Circle of Trust. The existing webservices of MusicWeb service providers need to process Liberty SOAP headers in order to integrate into a Liberty Alliance Circle of Trust environment. For example: eShare needs to process the *WS-Security token* header to identify the consumer whose music tags are being requested.

The use of single-sign on simplifies authentication for both service providers and MusicWeb consumer. eShare and CoTBuy are able to share consumer's data without revealing their local identity for the consumer. Since they cannot cross-reference their local pseudonymous identifiers, it prevents any collusion attempts. An Audit Trail Service was shown to help MusicWeb process document the sharing of consumer data between CoTBuy and eShare. With the Event Stat Service, we looked at how an event-logging mechanism can be used to help eTunes,iMusic and CoTBuy measure a key performance indicator.

The basic mechanism was to propagate the security tokens (in form of SAML assertions) from the BPEL process to Circle of Trust services to conduct transactions on behalf of the

consumer (whose pseudonym is contained in the token). Such tokens are obtained through EPRs from Discovery Service during runtime and also during SSO as a bootstrap EPR from IDP.

4.4. Support for Trusted Processes with Human interaction

The MusicWeb BPEL process had a fully automated workflow with only service-to-service interactions. In this section, we look at another scenario to evaluate how our framework helps trusted processes manage workflow steps that require human interactions.

A process may need such interaction to dynamically obtain consent from consumer before trying to access his data. In addition, the process may need inputs from service provider employees (such as Loan Officer) during process runtime as part of workflow logic. However, the current OASIS WS-BPEL specifications are service-centric and lack the ability to integrate people into processes. In our framework, we use the Task Service and a Task portal to resolve this problem. The TS allows BPEL processes to assign task activities to people as part of workflow orchestration. The Task portal allows people to act upon the tasks assigned to them.

A Car Loan Scenario is introduced in section 4.4.1. The scenario involves a process that dynamically interacts with a Loan Officer to obtain his approval for a loan offer. A straw man BPEL implementation is briefly discussed in section 4.4.2 to illustrate the problem in combining human interaction with a BPEL process. Section 4.3.2 presents our integrated framework implementation in which we adds steps to interact with Loan Officer into BPEL process definition. Finally, section 4.4.4 wraps up with an analysis of our approach.

4.4.1 Car Loan Scenario

In this scenario, a car dealer runs an online portal for prospective car buyers. On its website, the consumer can request a quote for car loan. In order to process this request, the Car Dealer interacts with other service providers offering loan-related services in the B2B network. These include a Credit Rating Agency that provides access to consumer's credit rating score and a Loan Provider that calculates loan interest rates.

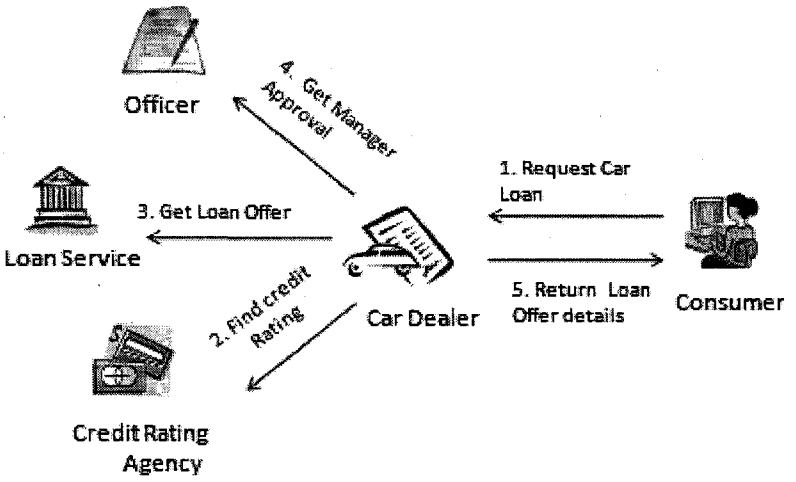


Figure 24 Car Loan Scenario

Figure 24 illustrates the different interactions in the scenario: (1) The consumer submits a loan application on the car dealer's website specifying the car details, loan duration and their Social Security Number (SSN). We assume that by making this request, the consumer gives his explicit consent to allow car dealer to access his credit rating information available inside the B2B network (for example, he is made to accept terms of conditions before submitting a loan request). (2) The Car Dealer uses the consumer's SSN to obtain his credit rating score from the Credit Rating Agency. (3) The Car Dealer sends the loan application details along with the

consumer's credit rating to the Loan Provider. Based on information provided, the Loan provider returns a potential loan offer according to current market interest rates. (4) The Car Dealer sends this offer to a Loan Officer for approval. (5) Once approved by the Loan Officer, the loan offer is emailed to the consumer.

Notice that step 4 introduces human interaction into the business process. This in turn makes the process long-running since the Loan officer may take several hours to approve the offer.

4.4.2 Strawman BPEL Implementation of Car Loan Scenario

In our first-cut Strawman implementation, the loan approval step of the Car Loan Process is handled separately by an approval application outside the BPEL process. Figure 25 below presents a high-level overview of the architecture.

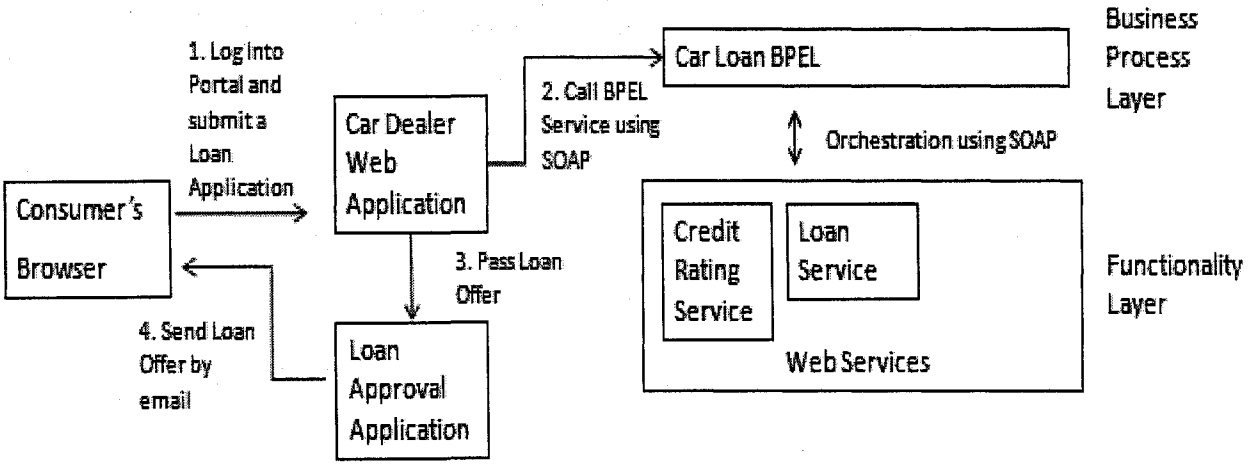


Figure 25 BPEL Architecture for CarLoan Scenario (Strawman)

A SOA is used to expose the functionality of Credit Rating Agency and Loan Provider as web services. The credit rating service takes consumer's SSN as input and returns their credit rating. Note that we assume that consumer has already granted permission when he made a loan application request on car dealer website (hence we do not show PDP here for simplicity). The Loan Service takes car, loan duration, and credit rating score details as input and returns a loan offer.

The Car Dealer web application locally authenticates the consumer. Upon the submission of a new car loan application by a consumer on its website, the Car Dealer web application starts the execution of the Car Loan BPEL process. A visual model of the process that was defined is shown in Figure 26. The process first obtains the consumer's credit rating and uses it to obtain a Loan offer based on his credit rating. Once the process finishes execution, the loan offer is sent back to the car dealer web application. This offer is then passed over to a Loan Approval Application where Loan Officer reviews and approves the offer. Finally, it is sent over to the consumer by email.

Notice that the BPEL process only uses service-to-service interactions. As noted earlier, the Loan Approval step is processed separately through an Approval application. There are proposed extensions such as a BPEL4People specification [Kloppmann2005] that add support for human interactions in BPEL. However, we do not use them here and instead focus on a straw man implementation that uses the BPEL standards specified in [Arkin2005].

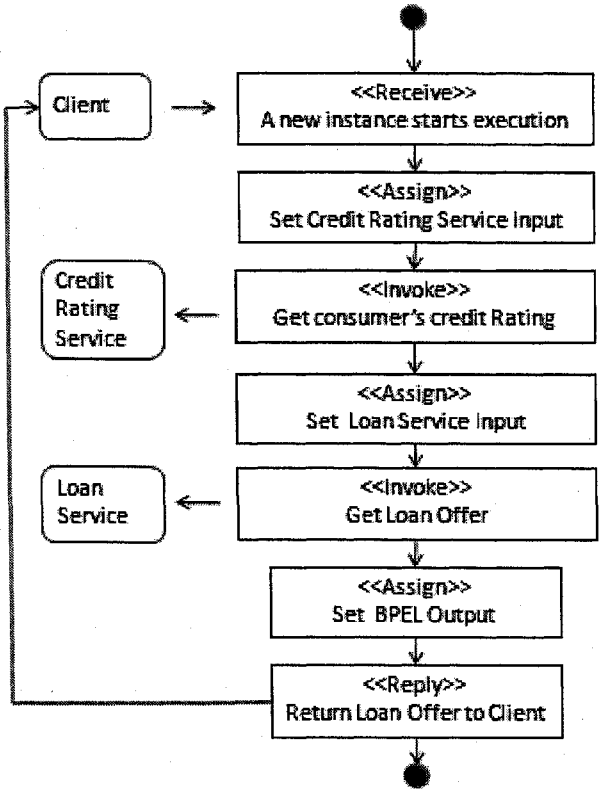


Figure 26 BPEL definition for the Car Loan Process

4.4.3 Integrated Framework Implementation of Car Loan Scenario

We will now explain how the same Car Loan scenario was implemented as a trusted process in our framework. The B2B service providers are integrated into a Liberty Alliance CoT as shown in Figure 27. IDP introduces Single-sign on and federated identity into the network. Thus, Car Dealer and Credit Rating Service each recognize the user (say Bob) through their own unique pseudonyms (Lets say "Bob_CarDealer" and "Bob_CreditRating"). Discovery Service enables the sharing of consumer's credit rating score between Credit Rating Service and Car Dealer. The consumer is no longer required to divulge his SSN to request a Car Loan quote. Instead, the Credit Rating Agency identifies consumer through EPR token that is passed to it.

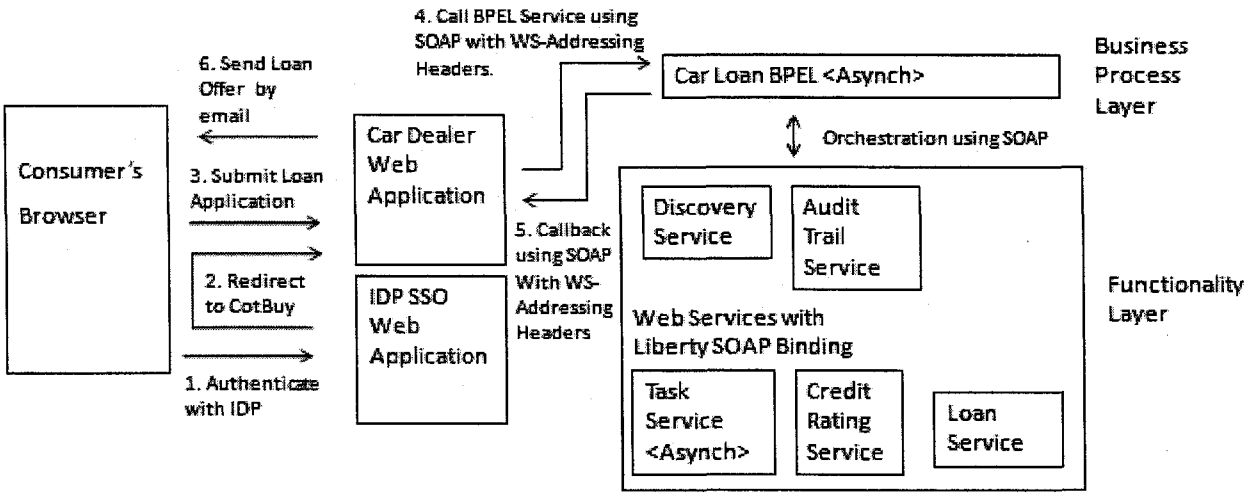


Figure 27 Integrated Framework Architecture for Car Loan Scenario

Two additional TPP components are used to support the human interaction in the Car Loan Scenario. First, a *Task Service* is used to assign the loan approval task to the Loan Officer and then wait for its completion inside the BPEL definition for Car Loan process. Second, a *Task portal* has a web interface for Loan Officers to approve the loan offers assigned to them. Note that both these components are designed to be generic i.e. they can be used by any BPEL process to manage different human task activities for employees/consumers in B2B network. (For example: to obtain consent from the consumer).

Since it could take several hours for the loan offer to be approved, Car Loan trusted process is designed to be asynchronous. To do so, its SOAP endpoint was defined to have two ports each containing a *one-way* operation – *initiate* (implemented by BPEL) and *onCallback* (implemented by the client). This is different from MusicWeb process which had a single two-way *initiate* operation. When the consumer submits the Loan Application, Car Dealer invokes

the *initiate* operation to start the execution of a new process instance. The loan application along with the bootstrap EPR obtained during SSO ("Bob_CarDealer_DS") gets passed as input to the process. Car Dealer also needs to set the WS-Addressing Headers *ReplyTo* and *ConversationId* while calling the *initiate* operation. They are required because asynchronous request and response messages are sent on different transport channels. Hence, these headers assist the process in sending back the response to Car Dealer. *ReplyTo* is the car dealer's callback endpoint address where the *onCallback* operation is invoked. *ConversationId* is passed back and forth between car dealer web and BPEL process so that car dealer can correlate an incoming response (i.e. *onCallback operation*) with its correct corresponding request (*initiate operation).*

We will now describe the trusted version of Car Loan process that was designed. The first task of our process is to obtain the consumer's credit rating information. The shaded steps in Figure 28 indicate the extra steps that were inserted into our straw man BPEL definition in order to accomplish this task. The <*Receive*> Activity starts the execution of a new process instance. Next, an EPR ("Bob_CarDealer_CreditRating") is obtained from DS. This EPR provides Car Loan process with tokens required to invoke the Credit Rating Service on behalf of consumer ("Bob"). The input variable for Credit Rating Service is left blank (Consumer's SSN is no longer required since Credit Rating Service identifies the consumer through the EPR token).

Next task of our process is to log this data-access to the Audit Trail Service for documenting compliance. To accomplish this, we query the DS for an Audit Trail Service EPR ("Bob_CarDealer_ATS") and use it to log the event to ATS. The input variables for DS and ATS along with Liberty SOAP headers are set in a similar fashion to the MusicWeb trusted process that was shown before in section 4.3.3.

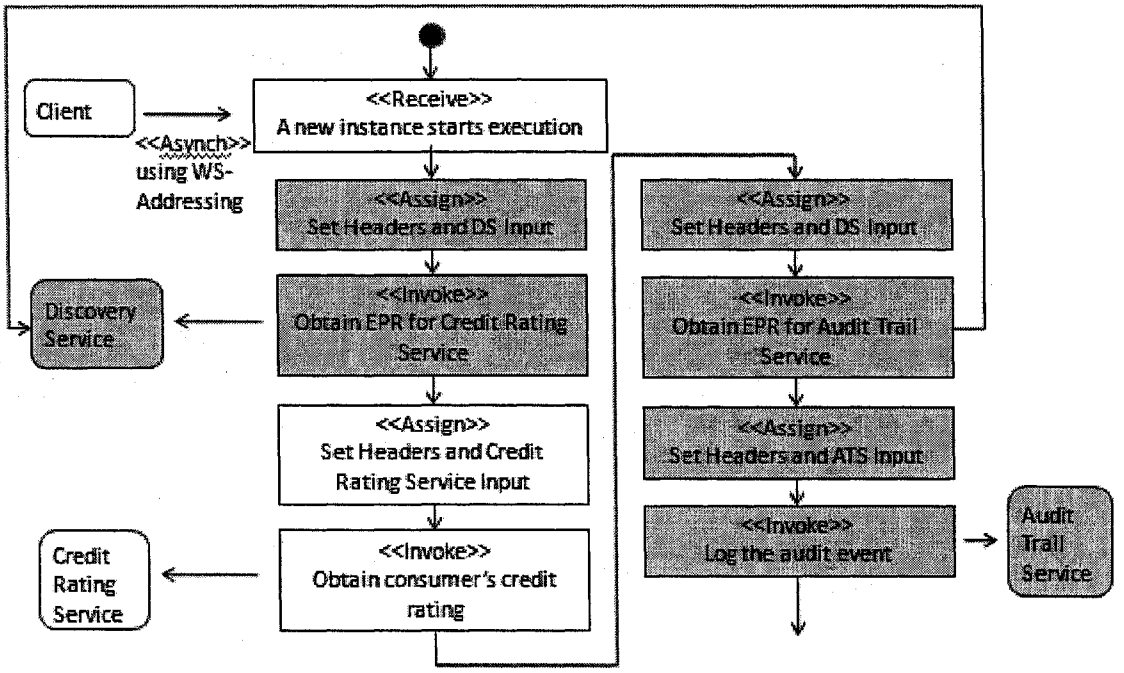


Figure 28 Sharing of credit rating information

The remaining part of our BPEL definition is illustrated in Figure 29. The first two steps are used to obtain a Loan Offer from the Loan Service. The input variable for Loan Service is set to car, loan duration and credit rating details. Notice that the process is not required to obtain EPRs to call Loan Service since it does not need to identify the consumer (Loan Service does not need to know *whose* credit rating information is being passed).

To integrate the loan approval task into our process, we inserted the five shaded steps on the right in Figure 29. The Task service is also designed to be asynchronous. Accordingly, its SOAP endpoint exposes two ports such that each of them define a single one-way operation, namely *addTask* (implemented by TS) and *onCompletion* (implemented by BPEL) respectively. Inside the partner link definition for Task Service both *myrole* and *partnerrole* values are set to

employ the asynchronous message exchange pattern while communicating with Task Service. By doing so, BPEL Engine automatically implements the port with *onCompletion* operation (which is the callback endpoint where TS notifies the process about the completion of a task). Note that for synchronous services only *partnerrole* value in a partner link definition needs to be set.

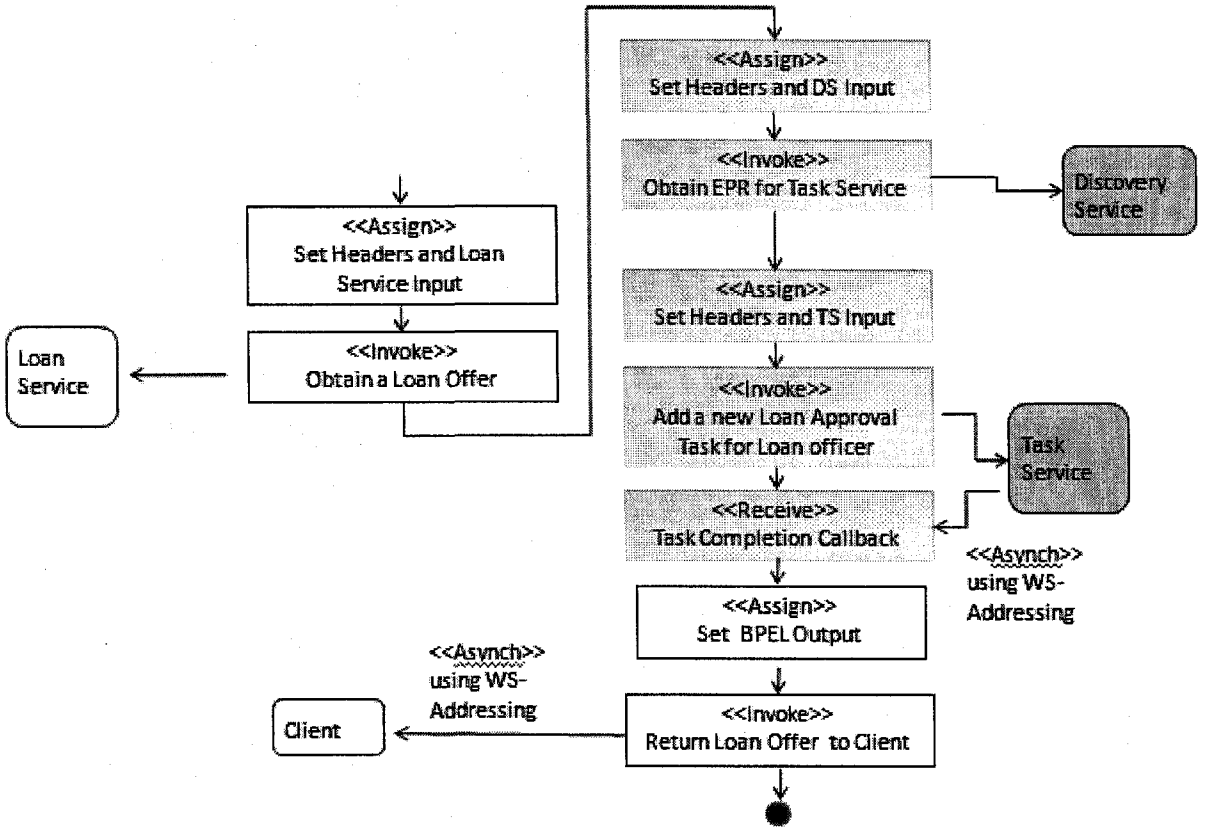


Figure 29 Obtaining Loan Offer Approval

The five steps added to manage Loan Approval Task with TS are as follows:

(a) *<<Assign>>*: sets the Discovery Service Input Variable and the associated Liberty SOAP headers to request an EPR that provides tokens to call Task Service on a Loan Officer's behalf (say "Fred"). Instead of obtaining a TS EPR on consumer's (Bob) behalf as usual, we do so on Loan Officer's (Fred) behalf to assign him a new task.

(b) <<*Invoke*>>: Calls the DS *Query* operation to request the EPR ("Fred_CarDealer_TS") and stores it into DS output variable.

(c)<<*Assign*>>: Sets the TS Input variable and the associated Liberty headers. The security token contained inside EPR ("Fred_CarDealer_TS") is copied into *WS-Security token* header. It is also imperative to set the *WS-Addressing* headers so that TS can perform the callback on process once the loan is approved. These values are automatically handled by the Oracle Process Manager (PM) BPEL engine we used. The TS Input Variable sets two parameters for the *addTask* operation. The Loan Offer from Loan Service output variable is copied into *taskData* part. The *taskType* part is initialized to "Loan_Approval" value. Together these values assign a new Loan Approval Task for the loan offer contained in the *taskData* for Loan Officer "Fred_TS" (Fred's pseudonym at TS extracted from *WS-Security* token header).

(d) <<*Invoke*>>: Calls the one-way operation *addTask* on Task Service to add a new Loan Approval task for Loan Officer (Fred). This is the asynchronous request to TS.

(e)<<*Receive*>>*:* Makes the process instance wait for the Task Service to invoke the callback operation *onCompletion*. In other words, it blocks the process until the asynchronous response from Interaction Service is received (upon the completion of a task). Oracle PM BPEL engine suspends the process while it is waiting for response and activates it again once the callback operation is invoked. BPEL provides an <<onAlarm>> activity that can be used to set a timeout for the response message, if required.

Note that for steps (d) and (e) to work, it is important to use WS-Addressing Headers. They are used to ensure that an asynchronous response is routed back to the correct process instance running on CoT BPEL Engine. With Oracle PM BPEL engine, the WS-Addressing

correlation is handled implicitly at BPEL engine level hidden from Car Loan BPEL process definition.

When the request is sent to IS, Oracle PM BPEL engine automatically sets WS-Addressing *ReplyTo* and *ConversationId* headers in the outgoing message. Interaction service uses the *ReplyTo* value to dynamically locate the BPEL process instance's endpoint for the callback operation. It also echoes back the *ConversationId* in its response to BPEL. Oracle BPEL Engine intercepts this value in the incoming response to route it to the correct BPEL process instance which made the corresponding outgoing request.

The steps above described how Car Loan trusted process works with the Task Service. Figure 30 below illustrates how the Task Portal helps the Loan Officer process the task assigned to him. In the diagram, Step 1 and 8 corresponds to the steps in Car Loan BPEL where it asynchronously interacts with TS (Figure 29). Task Service updates the Loan Officer's task list when it receives a new *addTask* request from the BPEL process instance. The Task Portal works just like any other web application in the CoT. Hence, the Loan Officer is first made to SSO with IDP before he can access his task list.

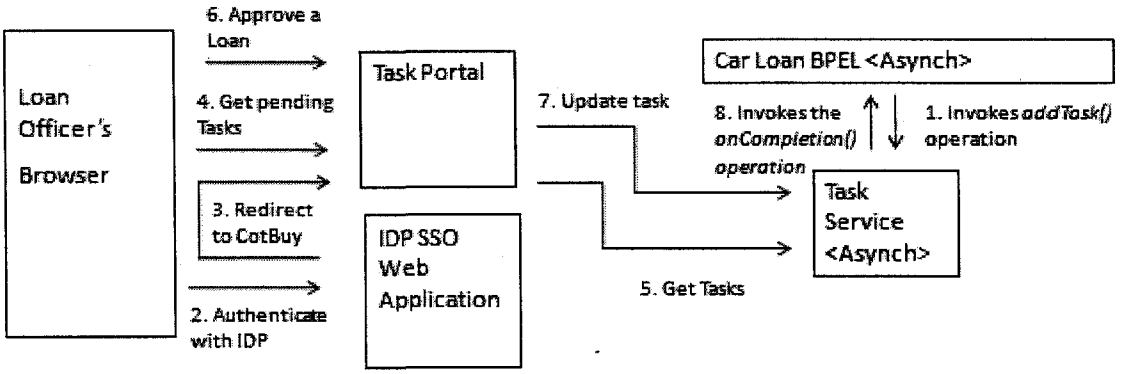


Figure 30 Approving a Loan Offer at Task Portal

Note that both CoT Task Portal and Task Service recognize the users through the same set of pseudonyms assigned by IDP. For example, Loan Officer Fred is known as "Fred_TS" to both the task portal and TS. The Task Portal simply provides a visual interface for a Loan Officer to view and perform action on pending tasks in his list. Once a loan offer is approved on the task portal, it notifies the Task Service which in turn notifies the BPEL through the callback operation.

4.4.4 Analysis

As business processes expand beyond organizational boundaries, integrating human actors into the process workflows becomes a challenge. Through a use-case scenario we demonstrated how a Task Service enables us to insert human interaction across different points of our BPEL process definition. A complementary CoT Task portal provides a centralized web based interface for any human actors inside the B2B network (including consumers) to view and process any tasks that have been assigned to them.

We exploited the asynchronous message patterns supported by BPEL standards and the in-built support for WS-Addressing processing by Oracle BPEL Engine to successfully implement such a Task Service. In our simple design of *addTask* operation, *taskData* and *taskType* parameters make it possible to extend TS with new types of tasks. For example, in our Car Loan process we could have used TS to obtain the consumer's consent before calling the credit rating service. In this case, we would obtain an EPR for Task Service on consumer's behalf and set *taskData* to the identity attributes being accessed (credit rating) and *taskType* to the "Consent". The consumer then logs into CoT task portal and authorizes their consent.

(However, in our scenario we assumed that at the time of making a car loan application, consumer explicitly opts-in to allow Car Dealer to check their credit rating).

4.5. Implementation Setup

In this section, we will describe the tools and the environment used in the implementation of our use-case scenarios. We first describe a Circle of Trust prototype that was built to simulate the Liberty Alliance Circle of Trust architecture in section 4.5.1. This is followed by a brief look at the Oracle BPEL designer tool that was used to script the definitions of our trusted processes in section 4.5.2. Finally, section 4.5.3 describes the run-time deployment.

4.5.1 Circle of Trust Prototype

To realize our use-case scenarios, we first implemented the IDP and TPP components to provide the basic infrastructure services of a Circle of Trust environment. Since Liberty Alliance is a vast set of specifications that in turn builds on several other industry standards from OASIS and WC3 , we were constrained by the lack of time and resources to implement all the details of the Liberty specifications. Instead, we built a CoT prototype that simplified the core mechanisms and features of Liberty Specifications (such as single-sign on, SAML token structure, EPR discovery, Liberty SOAP binding processing) and eliminated Liberty components that are not relevant to our framework. The prototype helped us abstract the complexity of the Liberty Specifications and analyze the integration of BPEL processes into CoT.

The IDE used for prototype implementation was Oracle Jdeveloper 10.1.3g. The J2EE 1.4 web components were used to create the SSO application, Task portal and privacy portal. JAX-RPC 1.1 web service toolkit from SUN was used for creating Liberty Services. It provides tools such

as WSDL2Java generation, XML-to-Object mappings and run-time libraries for SOAP message processing in J2EE application servers to simplify the web services development. Figure 31 illustrates a UML package diagram of our CoT prototype.

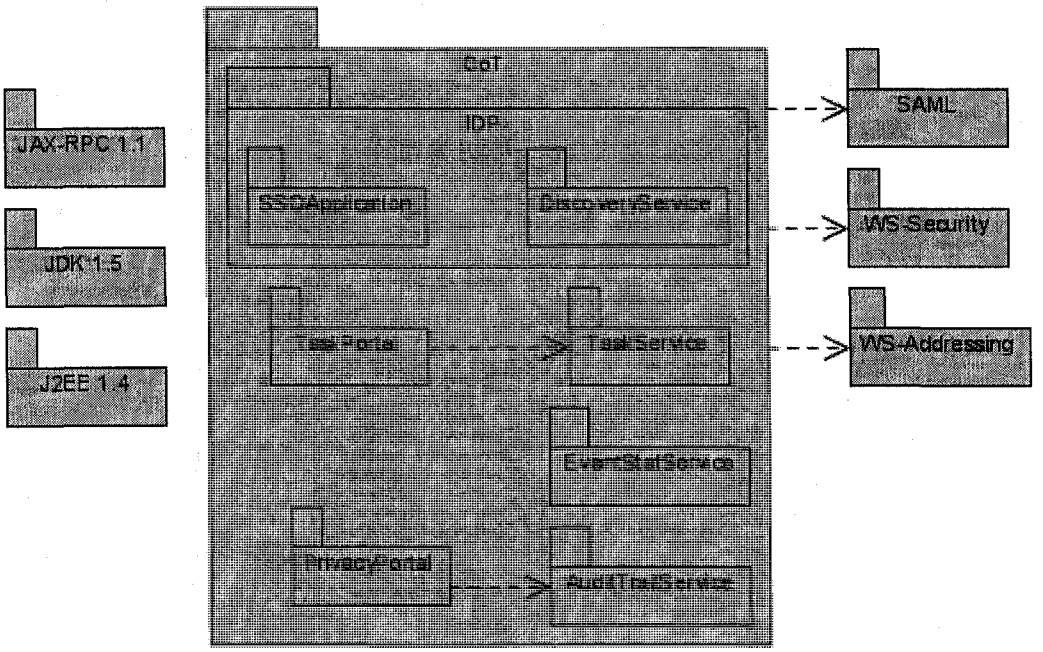


Figure 31 CoT Prototype package diagram

In the above diagram, the IDP package consists of the two core Liberty Alliance components, i.e., a SSO application and Discovery Service. DS implements an operation to obtain End-point references. Both these components have access to consumer's federated identities and are able to translate consumer's pseudonyms between different service provider namespaces. The packages for SAML, WS-Security and WS-Addressing are object mappings generated by JAX-RPC 1.1 tools from their XML schema descriptions. SAML schema is used to describe the structure of SAML assertion, WS-Security schema describes the structure of a WS-Security token and WS-Addressing schema describes the structure of an End-point reference and WS-Addressing *ReplyTo* and *ConversationId* headers. In the prototype, we used simplifications of the actual

schemas from OASIS and WC3. The other packages for Task Service, Audit Trail Service and Event Stat service are the new (Liberty) web services added into a CoT by our integrated framework approach. Section 5.3.1 analyzes the steps to create a new Liberty web service.
We also did a brief survey of existing implementations for Liberty Alliance specifications from industry vendors. Most implementations are in the form of library modules that can be leveraged by organizations that act as an Identity Provider or Service Provider. However at the time of our investigation during September 2007, most of these libraries were geared towards single-sign on and federated identity/SAML (i.e. Liberty ID-FF), with little or no support / documentation for the Liberty ID-WSF framework comprising of Discovery Service[LibertyDisco2006], Liberty SOAP binding [LibertySOAPBinding2006], Liberty SAML token profile [LibertySechMechSAML2006] and so on. A detailed comparison of these tools with respect to our CoT prototype is discussed in section 5.3.

4.5.2 BPEL Designer Tool

A very handy tool for our framework is a BPEL designer. While BPEL is purely an XML based language, BPEL designer tools facilitate rapid creation of BPEL processes through visual drag-and-drop of BPEL activities from a palette. As noted earlier, BPEL itself describes no standard graphical notation. We looked at BPEL designer tools from Netbeans 5.5, Eclipse 3.3 (as a plug-in) and Oracle Jdeveloper 10.1.3 IDE. Section 5.4.4 briefly compares them. Oracle's BPEL designer was by far the most feature-rich tool with extensions for handling SOAP headers. More over, it easily integrates with our CoT prototype code (also under Jdeveloper IDE) and Oracle Process Manager Engine (used to execute trusted processes). Figure 32 shows a screenshot of the Oracle BPEL designer.

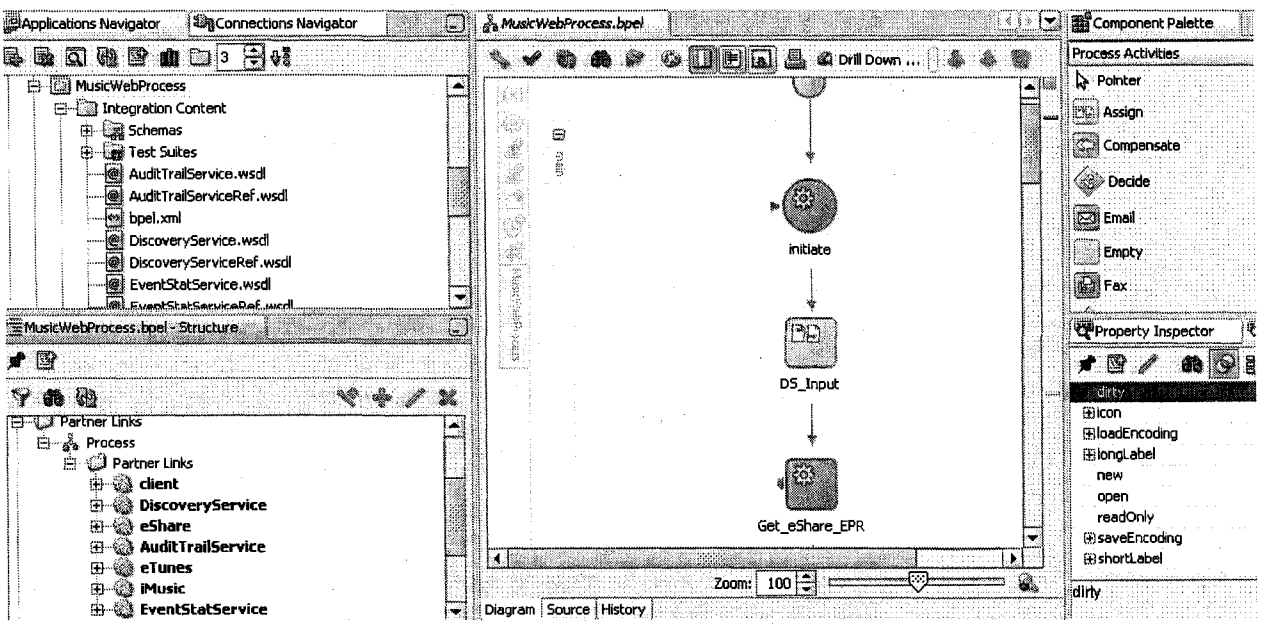


Figure 32 Oracle JDeveloper BPEL Designer

Useful features of the designer include an advanced XPATH editor to manipulate variable data, an editor to import WSDLs for CoT web services, automatic generation of partner link definitions, rendering of a visual model to illustrate the process workflow and support for several extensions provided by the Oracle PM BPEL engine. In addition, it facilitates easy deployment of the process definitions to any standards compliant BPEL engine.

4.5.3 Runtime environment

The deployment diagram shown in Figure 33 illustrates the runtime environment of our implementation. Two instances of Oracle Application Server (also known as Oc4j container) were used to test the execution of the use-case scenarios. The first container ran all the web applications and services that are part of IDP and TPP in our CoT prototype. In addition, it also deploys the web applications and services implemented for the different service providers in our

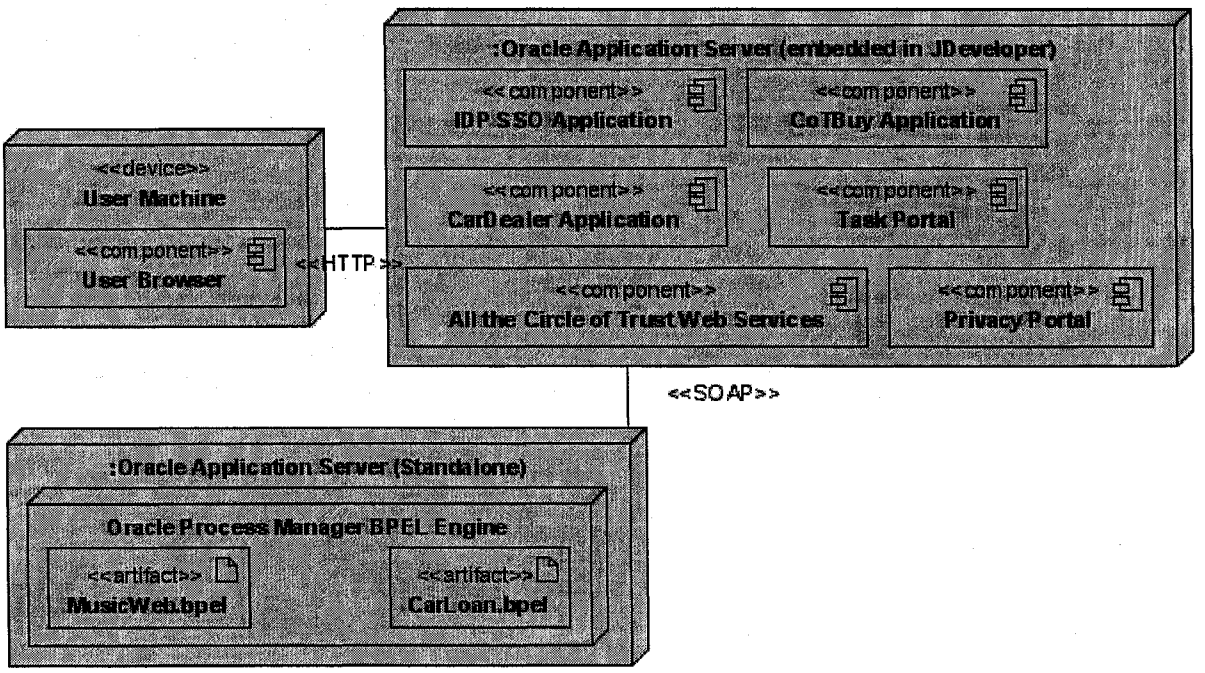


Figure 33 Deployment Diagram

two use-case scenarios (i.e. the eShare service, iTunes service, Credit Rating Service, CarDealer application and so on). This container came embedded with Oracle Jdeveloper IDE. The second container ran the Oracle Process Manager BPEL Engine. It implements WS-BPEL 2.0 standards but also comes with several extensions for XML manipulation, SOAP headers and implicit WS-addressing support. The MusicWeb and CarLoan trusted processes were deployed to this container from the Oracle BPEL designer.

The user's browser interacts directly with the web applications deployed inside embedded container. These web applications in turn start the execution of new process instances by making a SOAP request to Oracle PM Engine deployed inside the standalone container. The engine executes the orchestration logic and interacts with the Circle of Trust services by exchanging SOAP messages with the embedded container.

Chapter 5. Evaluation

The results of our case study are used to evaluate our thesis contributions along a number of different dimensions. First we analyze the feature set provided and the effort to implement trusted processes in our framework. Then we evaluate the tools and standards used in our case study to realize our framework.

5.1. Feature Comparison

In this section, we evaluate the extra value added by our integrated trusted processes framework over the service-oriented WS-BPEL and the Liberty Alliance frameworks it leverages. In order to do so, we analyze each of them for their support of consumer-facing B2B business processes in terms of the features they provide.

5.1.1 Liberty Alliance Framework

Liberty Alliance Framework comprising of ID-FF, ID-WSF and ID-SIS set of specifications addresses issues of identity and privacy through its core IDP, Discovery Service and PDP components. It also describes an interaction service [LibertyInteract2006] to enable human interaction into Liberty Services.

While Liberty Alliance uses a SOA design in ID-WSF specifications for a data exchange framework, there are no specifications for business process automation and management. Currently, applications in the Circle of Trust need to hard code the automation of a process. This

makes it difficult to design and deploy processes. Tracking Ad-hoc process instances would be challenging in the absence of a process management console facility usually provided by business process automation engines to monitor lifecycle of process instances.

Finally, Liberty Alliance specifications were designed with interoperability as one of their key goals. It extensively uses other established standards from WC3 consortium and OASIS including SAML, WS-Security, WS-Addressing and so on.

5.1.2 Service-Oriented WS-BPEL standard

Here, we refer to the use of WC3 web service technology stack comprising of SOAP, WSDL, UDDI with the OASIS WS-BPEL standards in a SOA. Web services provide the fundamental abstraction of a service while BPEL provides the abstraction of a business process. BPEL is a business process automation tool that is an important point of intersection between SOA and BPM.

However, qualities of service requirements like security, privacy, compliance issues are not addressed as part of WS-BPEL standard. Further, BPEL is not a true workflow engine lacking in-built integration with people. However, there is BPEL4People **[Kloppmann2005]** extension that is being pushed for standardization and it introduces the concept of people activity and human task to integrate people into processes. Similarly, BPEL is not equivalent to BPM and does not address issues related to performance monitoring. However, as described in section 2.2 Business Activity monitoring extensions have been investigated for BPEL.

5.1.3 Integrated Trusted Processes Framework

Our trusted processes framework builds on the SOA described by a Liberty Alliance Circle of Trust with business automation and management support. The addition of a Trusted Process provider helps manage BPEL automation of processes inside CoT. It specifically caters to the issues that arise in cross-organizational processes in B2B networks. The Audit Trail Service was integrated into the CoT to fill in the gaps for privacy compliance. An event stat service in our scenario illustrated how an event logging mechanism across the CoT can be used to monitor performance. Human interaction support was added to BPEL processes through a service-oriented Task Service. However, as we will see in section 0, there are certain incompatibilities that exist in WS-BPEL standards to fully comply with Circle of Trust Services.

Table 3 Comparison of features for building consumer-facing B2B processes

Approach / Feature	Service-Oriented WS-BPEL	Liberty Alliance Framework	Integrated Trusted Processes Framework
Process Definition	Service Composition	Hard-coded applications	Service Composition (using WS-BPEL)
SSO and Federated Identity	-	Identity Provider	Identity Provider
Controlling sharing of data while protecting identity	-	Discovery Service with PDP	Discovery Service with PDP
Privacy Compliance	-	-	Audit Trail Service
Transparency		-	Privacy Portal
Performance Monitoring	No, but BAM extensions in literature	-	Event Stat Service
Human Interaction	No, but BPEL4People is a proposed standard extension	Interaction Service	Task Service and Task Portal
Standards-driven approach to ensure B2B interoperability	Yes	Yes	Yes, but currently extensions required for WS-BPEL processes to work with Liberty web services

Table 3 above summarizes the features supported by the integrated trusted processes framework in comparison to the Service-Oriented WS-BPEL and Liberty Alliance framework.

5.2. Evaluation of Efforts and Complexity

We have shown that our framework provides an enhanced set of features for trusted processes. In this section, we evaluate the effort and skills required to define and deploy trusted processes as well as the complexity of the definitions created. We will show that the use of BPEL scripting greatly reduces the effort to create, debug, and maintain a trusted process in a CoT, compared to hard-coding of business process into composite applications in a SOA as is traditionally done for a Liberty Alliance CoT. These advantages are obtained through the simple fact of integrating a BPEL engine into our framework. The critical element in analyzing our approach is then to understand how much complex the BPEL scripting needs to be in our framework in order to leverage the additional CoT services for Identity Management, Privacy Compliance Transparency, and Performance Management.

5.2.1 Creating a new trusted process

Table 4 summarizes the efforts required to create a new trusted process in the CoT.

Methodology:

Without service composition, the automation of a process has to be hard-coded in an ad-hoc fashion. Business Process activities are intertwined with the low-level programming details. Extensive knowledge of a programming language and web service framework API is required. In

addition, the programmer may have to worry about details like XML-to-object mapping, marshalling and de-marshalling of SOAP message and handling of asynchronous service interactions.

Table 4 Efforts to create a Trusted Process

Criteria	Hard-coding of process automation in a Circle of Trust	With BPEL using our integrated trusted processes framework
Methodology	Enterprise Java or .NET coding using Web Service Frameworks (like JAX-RPC)	Writing BPEL script
Artifacts	Classes and Objects including client stubs/proxies , Ant build scripts ,deployment descriptors	XML file (BPEL script), schemas, WSDLs and deployment descriptors
Use of Visual Tools	IDE editors, wizards to import WSDLs/schemas and generate client stubs/proxies	Graphical BPEL designer with XPATH editor

BPEL helps design trusted processes at higher level of abstraction. The implementation details for enacting the workflow are left to the BPEL engine. This is well suited for business analysts who can focus on process workflow and business logic. Knowledge about BPEL syntax and a conceptual understanding of web services (including SOAP and WSDL) is required.

Apart from benefits of providing a high-level abstraction for process developers, WS-BPEL supports advanced concepts and infrastructure support that can help someone engineer a process using best-practices with limited development experience. For example, BPEL supports asynchronous message-exchange patterns which were used to integrate a Task Service into our framework. In order to have stateful conversations with partner services, BPEL uses the concept of correlation variable sets to pass explicit conversation identifiers. Furthermore, BPEL Engine can offer additional runtime services such as automatic persistence of long-running processes

(Oracle BPEL PM does). All this could be done manually using programming language but it would be more time-consuming and complex.

Artifacts:

Hard-coding composite applications for trusted processes involves generation of various artifacts including client stubs/proxies generated by web-service frameworks, a set of core business process objects that make use of them and various other deployment descriptors/build scripts depending on the programming platform that's used. On the other hand, only one XML script containing the BPEL syntax comprised the source code for the entire process in each of our scenarios. In addition a deployment descriptor was used by Oracle BPEL to deploy the process. Also, since a BPEL process is itself exposed as a web service endpoint, both MusicWeb Process and Car Loan Process have their own WSDL and XML schema files.

Use of Visual Tools:

A lot of the artifacts listed above are typically auto-generated by IDE tools/wizards in both cases. However with our framework, a Visual BPEL designer can substantially eliminate the need to write any code by hand at all. In our case study, Oracle JDeveloper BPEL designer came with wizards to import WSDLs and setup partner links, palettes to drag-and-drop BPEL activity syntax and property editors to customize them. Similarly, a grapahical XPATH editor simplified manipulation of data to copy variable values. The IDE automatically generated deployment descriptors, WSDL and Schema files for the process. On the other hand, data manipulation with the hard-coding of process automation may require use of complex DOM/ SAX XML processing APIs.

5.2.2 Debugging a Trusted Process

Table 5 below summarizes the efforts required to debug a trusted process in the CoT.

Table 5 Efforts to Debug a Trusted Process

Criteria	Hard-coding of process automation in a Circle of Trust	With BPEL using our integrated trusted processes framework
Understanding the Process	Complex as process logic & workflow is intertwined with implementation details.	Easier with a Graphical notation rendered in a BPEL designer
Process Debugging	At code level	At BPEL Script Level.

Understanding the process workflow

With hard-coding of process automation into applications, it can be a tedious task to comprehend the process by directly analyzing the code. With BPEL, it is relatively easier to read the human-friendly XML in a BPEL script. BPEL constructs like Assign, Invoke, and Receive are simple to understand and directly map to a workflow step. In addition, all BPEL designer tools we evaluated came with 2 different views of a process – graphical model and XML source code. The Graphical Model illustrates the flow of process execution and any links to Partner Services that makes it easier for anyone even without full knowledge of BPEL syntax to comprehend the trusted process. However with both approaches, presence of design documents (such as graphical notations using a business process modeling tool) would assist in understanding the process.

Business Process Debugging

Here, we are concerned with the ability to debug at the business process level. A BPEL script consists of partner link definitions, variable declarations and the sequence of activities for

orchestration logic (analogous to a main method). Stepping through his sequence block makes it possible to debug at the business process level. Oracle BPEL Process Manager came with a management console to visually debug processes. Just like debugging in a programming language, it was possible to watch the contents of BPEL variables (that correspond to the body of input and output SOAP messages) at every step of activity sequence. In case of an error, they could be examined for SOAP fault messages.

5.2.3 Maintaining a trusted process

Table 6 below summarizes the efforts to maintain a trusted process.

Changes to Process

A change at the business process level could translate into extensive modifications in the code to fix it with hard-coded approach. In our framework, all changes to process workflow happen in BPEL script file. If external web services change their interfaces, far fewer artifacts need to be repackaged or redeployed compared to hard-coded approach.

In BPEL process definition, partner links are tied to portTypes which are abstract concepts in a WSDL. Thus it is agnostic to concrete details such as service endpoint location or SOAP binding (such as SOAP/HTTP) that is being used. If there are any changes in a binding protocol or the endpoint address, only the partner WSDL interface needs to be re-imported. No changes are required to process definition in BPEL script. The use of BPEL provides high-level abstractions and infrastructure services that can enable the service provider to adapt quickly to changes in a CoT architecture or business needs.

Table 6 Efforts to maintain a trusted process

Criteria	Hard-coding of process automation in a Circle of Trust	With BPEL using our integrated trusted processes framework
Changes to Process	Changes to code (possibly multiple objects), recompilation of client stubs/proxies, importing/updating partner WSDL & schemas	All Changes happen in a single BPEL script, importing/updating partner WSDL & schemas.

5.2.4 Complexity in scripting a Trusted Process in our framework

In this section, we analyze the complexity of a BPEL script to design a trusted process. As seen in our case study the number of BPEL activity steps and complexity of data manipulation (i.e. <assign> activity) increases substantially as we add various trusted features to our BPEL process.

Partner Link Definitions

As with any BPEL process, a partner link is added for every B2B web service that the process interacts with. Four additional partner links need to be defined for the Circle of Trust services used in our framework: (a) Discovery Service (b) Audit Trail Service (c) Event Stat Service (d) Task Service.

Variable Definitions and BPEL Activities

Every interaction with a Circle of Trust service adds extra activities and variables to our BPEL process. The complexity of data manipulation involved also increases as additional <copy> blocks are introduced into <Assign> activity to use the Liberty SOAP headers.

We will briefly analyze the complexity for each major CoT function:

(A) *Sharing consumer data:* An <Assign> followed by <Invoke> is first used to obtain an EPR from DS for the Service Provider. This is followed by another set of <Assign>, <Invoke> to call the Service Provider. Four variables are used in all to store the input and output for DS and Liberty Identity Service. For both the <Assign> activities, at least three copy blocks are used to set *WS-Security*, *sbf:framework* and *sbf:sender* headers. DS *queryDS* operation requires at least one parameter to request an EPR. The number of copy blocks varies while calling a service provider depending on the operation being called.

(B)*Document Compliance:* An <Assign> followed by <Invoke> obtains an EPR from DS for the Audit Trail Service which is stored in DS output variable. This ATS EPR value can be reused again whenever a data-sharing event needs to be logged. Hence, unlike (A) obtaining an EPR is only a one time operation. After that, an <Assign> creates the audit instance event in ATS input variable. This requires three copy blocks to initialize the input parameters for the Audit Trail Service *logEvent* operation. Finally, <Invoke> logs the event to ATS.

(C)*Capture Business Activity Events:* No identity related information is stored in ESS. Hence discovery EPR mechanisms are not used. Logging events to ESS only requires two steps. <Assign> initializes the ESS input variable with the event details that needs to be logged (Click through rates in our case study). <Invoke> logs the event to ESS and stores the status message (indicating success or failure) in ESS output variable. At least two copy blocks are required for <Assign> activities (*WS-Security token* header can be left empty).

(D) *Human Interaction:* As <Assign> followed by <Invoke> obtains an EPR from DS for Task Service. Two one-way operations are used to asynchronously communicate with Task Service. <Invoke> adds human task to TS without any output variable. <Receive> is asynchronous

callback from TS upon completion of task and has no input variable. Two additional copy blocks are used to set *WS-addressing* headers. Table 7 above summarizes the complexity involved to support the different trusted components of our integrated framework.

Table 7 Complexity of BPEL Process Definition

Function	Complexity				
	BPEL Activity	Purpose	<copy> Blocks for Liberty SOAP headers	<copy>Blocks for SOAP Body	Variables Used
(A) Sharing Consumer Data	<Assign>	Get EPR from DS for Service Provider	At least 3	At least 1	
	<Invoke>		-	-	2
	<Assign>	Share Identity Attributes	At least 3	Varies	
	<Invoke>		-	-	2
(B) Document Compliance	<Assign>*	Get EPR from DS for ATS	At least 3	At least 1	
	<Invoke>*		-	-	2
	<Assign>	Log Data-sharing to ATS	At least 3	3	
	<Invoke>		-	-	2
(C) Capture a business activity event	<Assign>	Log Business Activity Event to ESS	At least 2	Varies	
	<Invoke>		-	-	
(D) Human Interaction	<Assign>	Get EPR from DS for TS	At least 3	At least 1	
	<Invoke>		-	-	2
	<Assign>	Add Human Task	At least 5	2	
	<Invoke>		-	-	1
	<Receive>	Wait till Task is complete	-	-	1

*** Required only once in entire script**

5.3. Analysis of Liberty Alliance Tool Support and Specifications

In this section, we will first compare the different Liberty Alliance implementation tools we investigated and look at efforts to integrate a web service into CoT (Table 8). We then look at the different steps involved in creating a Liberty Web Service.

Table 8 Comparison of LA Tool Support and Efforts to Integrate a web service into CoT

Liberty Alliance – Related Implementations	Usage	ID-FF	ID-WSF	Platform	Effort to integrate Web Service into CoT
Sun Access Manager 7.0	For Identity Provider functionality	1.1	-	Java	Not possible
Sun Federation Manager 7.0	For Service Provider Functionality	1.2 (SAML 2.0)	1.0	Java	A Library supports it but I couldn't get it to work
Open SSO	Open source spin-off from Sun products	1.2 (SAML 2.0)	1.1	Java	No documentation for creating new Liberty Services
Lasso	Primarily supports Single-sign on and federated identity	1.2 (SAML 2.0)	Incomplete	C with bindings for multiple languages	Not possible
OpenLiberty	Client library to call Liberty services	N/A	Incomplete (in progress)	C++,Java	Not possible
Prototype CoT	Used to analyze and evaluate our ITPF	Simplified subset of 1.2 (SAML 2.0)	Simplified 2.0 subset	Java	Straightforward but manually done

Access Manager and Federation Manager from Sun included libraries with an implementation of Liberty specs and a deployable J2EE web application for the admin console. These two products complement each other. Access Manager offers the functionality at Identity

Providers site, while federation manager offers the capabilities at Service Providers website for Single-sign on and Identity federation. Sun Federation Manager however supports latter version of SAML and ID-FF specs. Access Manager does not support integration of new Liberty Services while Federation Manager does. OpenSSO is open-source library based on Sun's Access manager & federation manager products. Future versions of Access Manager and Federation Manager are planned to be released upon open-source OpenSSO builds.

Lasso is a free software C library aiming to implement the Liberty Alliance standards. The library has bindings for multiple languages and operating systems. The initial support was only for ID-FF specifications (mostly SSO), but an August 2007 release had some preliminary support for ID-WSF.

OpenLiberty (http://www.openliberty.org) is a recent open source initiative. Recently in Feb 2008, they released a client library for calling Liberty Services including the Discovery Service. However, there is no library support to convert an existing web service into a Liberty Service.

In the lab, we tried out a few of these libraries around September 2007.Most of these libraries worked well for single-sign on and identity federation. However at that time, only OpenSSO and Sun federation manager supported the ID-WSF standards. OpenSSO came with no samples or documentation for creation and integration of Liberty services. The initial builds, samples & related documentation of OpenSSO were primarily targeted towards ID-FF set of specifications. Sun federation manager came with some documentation for the creation of new ID-WSF identity services. However, even though it was successfully installed, we were not able to get the web services working.

5.3.1 Steps for Creating a Liberty Service

In order to integrate an existing web service into Circle of Trust architecture, a web service needs to process information as defined in [LibertySOAPBinding2006]. Table 9 presents an overview of the key steps involved in creating a Liberty Service that is relevant to our framework.

Table 9 Conversion and Integration steps for a Liberty Service

Extra Steps	Purpose
Processing SAML Token in WS-Security Token Header	Identify the user
Processing WS-Addressing Headers	Stateful Message conversations
Required and Optional Headers in [LibertySOAPBinding2006]	Various purposes such as version of liberty ID-WSF specs, sender's identifier, conveying target identity, consent usage field and so on.
Standard Data Template defined in [LibertyDST2006]	Standard Interface and data-model to a Liberty Service for adding, modifying and querying data. (Optional)

The extra steps are:

(1) The WS-Security Token header is required for all liberty services that store, modify and share user identity attributes. The Liberty Service must extract this token and decrypt it using its key to obtain the user's unique local pseudonym. This step is not required if the service does not store identity related data (such as Event Stat service).

(2) WS-Addressing headers are only required if the Liberty Service interacts asynchronously (like Task Service). The Service must process the *ReplyTo* and *ConversationId* value to successfully send a response back to its caller.

(3) In addition to (1) and (2), [LibertySOAPBinding2006] defines several other required and optional headers. The sbf:sender can be used to find out the service provider who is requesting data. The sbf:framework is used to indicate the version of ID-WSF specs being used.

(4) The Liberty Standard data service template is optional and it defines a standard interface to add, modify and query user data.

5.4. Analysis of WS-BPEL standards and BPEL Tool Support

In this section, we briefly analyze how the WS-BPEL standards support our trusted processes framework and propose extensions to standards to resolve some incompatibilities. Finally, we present a comparison of 3 BPEL tools that were tried out in our lab and see how well they support our framework on two separate dimensions: (a) BPEL designer features (b) BPEL specifications and extensions support.

BPEL models business processes as compositions of web service interactions and it fits well in a CoT that is architected as a SOA. However, we found gaps in WS-BPEL standards that prevent them from working seamlessly with our framework.

5.4.1 WS-Addressing correlation mechanism for Asynchronous Task Service

As described earlier, BPEL standards support asynchronous message exchange patterns whereby request-response messages are sent on different transport channels. BPEL Engine must be able to route an incoming asynchronous response message to the correct process instance that made the corresponding request. To do so, a *correlation* mechanism is used. As illustrated in section 4.4.3,

WS-addressing headers are used to correlate the messages with task service in the car loan process. The advantage of using WS-addressing is that Task Service is dynamically able to locate and invoke the callback endpoint on CarLoan BPEL process instance. In fact, it can do so with any BPEL process instance it interacts with (since callback endpoint address is always passed in header). This keeps the design of task service generic and loosely-coupled to the BPEL process it communicates with. However, WS-Addressing correlation is an extended feature supported by Oracle BPEL engine. It is not part of BPEL standards.

WS-BPEL standards specify the use of "correlation variable sets" as the default correlation mechanism. When this mechanism is used, it is not possible for Task Service to dynamically callback a BPEL process instance. The Task Service has to be tightly coupled with a static BPEL callback endpoint for it to send a response upon completion of a task

In order to design a true generic task service that can be reused by any trusted process in our framework, the use of WS-addressing is required. Thus, the WS-Addressing correlation mechanism needs to be adopted into the BPEL standards to ensure greater inter-operability.

5.4.2 Support for handling SOAP headers defined in WSDL binding

While WS-BPEL standard provides an Assign activity to manipulate data, it is only restricted to data contained inside the body element of SOAP message. It does not provide any way to manipulate information in the SOAP header. This is a stumbling block for our framework since ability to handle SOAP headers was required to set the header information as defined in [LibertySOAPBinding2006] while interacting with a Circle of Trust service.

```
<invoke name="Get_eShare_EPR" partnerLink="DiscoveryService"
        portType="ns1:DiscoveryService" operation="queryDS"
        inputVariable="Call_DS_queryDS_InputVariable"
        outputVariable="Call_DS_queryDS_OutputVariable"/>

<assign name="eShare_Input">
  <copy>
      <!-- Assign Information to Call_eShare_getUserTags_InputVariable -->
  </copy>
  <copy>
    <from variable="Call_DS_queryDS_OutputVariable" part="parameters"
          query="/ns1:queryDSResponseElement/wsa:EndpointReference/saml:token"/>
    <to variable="SecurityHeader" part="wssecurity"
        query="/wsse:security"/>
  </copy>
 </assign>

<invoke name="Fetch_User_Tags" partnerLink="eShare"
        portType="ns5:FetchTagsService" operation="getUserTags"
        bpelx:inputHeaderVariable="SecurityHeader"
        inputVariable="Call_eShare_getUserTags_InputVariable"
        outputVariable="Call_eShare_getUserTags_OutputVariable"/>
```

Figure 34 Oracle BPEL Extensions to handle SOAP headers

In order to address this problem, some BPEL tool vendors include their own extensions. For our case study, we used one such extension supported by Oracle's BPEL tools that were used in our implementation. Figure 34 above illustrates how SAML tokens contained in an EPR are passed off to eShare service as WS-Security headers in Oracle JDeveloper BPEL Designer. The first <Invoke> activity calls the Discovery Service and stores returned EPR in `"Call_DS_queryDS_OutputVariable"`. The <Assign> activity extracts the SAML token from that variable using XPATH query and stores it in `"SecurityHeader"` variable. Finally while invoking eShare service, `bpelx:inputHeaderVariable="SecurityHeader"` sets the header for the outgoing SOAP envelope.

5.4.3 Support for Digital signatures

As per liberty security mechanisms for SAML tokens as described in LibertySecMechSAML2006], service providers must digitally sign the header information (including *WS-Security token)* using XML signature while communicating with other Circle of Trust services. However, this is neither supported by BPEL standards nor the Oracle PM BPEL engine. One possible solution is to have BPEL engine configured to automatically sign headers while interacting with a liberty service.

5.4.4 Comparison of BPEL Tool Support

In this section, we evaluate the BPEL tools from 3 different vendors – (1) Sun Netbeans which uses the BPEL implementation from Open-ESB Project (2) Eclipse Project and (3) Oracle. The criteria listed in Table 10 below reflect some of the features and capabilities that were found to be useful and/or essential for building trusted processes using our framework.

All 3 vendors provide a graphical BPEL designer and they either come bundled with IDE or as a separate plug-in. Similarly, they include a graphical XPATH editor to manipulate process data. It helps to visually copy variable value (or parts of it) into another variable (or parts of it) without having to write complex XPATH queries. Oracle's JDeveloper BPEL designer came with an advanced XPATH editor with several proprietary extensions to copy XML data.

Table 10 BPEL Tool Support Comparison

	Netbeans 5.5	Eclipse 3.0	Oracle Jdeveloper & Process Manager BPEL Engine
BPEL Designer and Runtime Environment Features			
Graphical Designer	Yes (Integrated with IDE)	Yes (as a plugin)	Ye s (Integrated with IDE)
Graphical XPATH editor	Yes	Yes	Yes
BPEL Engine to deploy processes	Yes comes bundled with an Open-ESB BPEL Service Engine	Currently BPEL plug-in project has stopped at milestone 3 and doesn't support execution of BPEL processes. Future milestones plan to add support to deploy processes into other BPEL runtime engines like Oracle Process Manager	Yes (Oracle BPEL Process Manager) with special optimizations for long-running asynchronous processes.
Process Management console	No but comes with support for debugging of processes	N / A	Yes . Supports managing the entire lifecycle of deployed processes, visual debugging of process and a visual audit log of completed process instances
WS-BPEL Specifications and recommended Extensions Support			
Compatibility	WS-BPEL 2.0 (Some constructs are not fully supported)	WS-BPEL 2.0	WSBPEL 1.1 with partial support for WSBPEL 2.0
Asynchronous service correlation mechanism	correlation variable sets	Correlation variable sets	WS-Addressing headers implicitly (also supports correlation variable sets)
Human Task Service	No	No	Yes (This is different from task service we built in our framework)
Handle SOAP headers	No	No	Yes through proprietary extensions for advanced XML manipulation

Eclipse Project does not come with a BPEL engine to orchestrate the process definition. Netbeans comes bundled with a BPEL Service engine from OpenESB project. Oracle comes with its own Process Manager BPEL engine. It has a web-based management console to manage the deployment and execution of new process instances and also debug them visually which was a very helpful feature during the implementation of our scenarios.

OpenESB BPEL SE implements WS-BPEL 2.0 (not fully supported yet), while Oracle BPEL PM supports WS-BPEL 1.1. In terms of their support for extensions discussed above, Oracle BPEL PM has better support. It uses a WS-addressing correlation mechanism by default and if required correlation variable sets could also be used. Similarly, it came with an out-of-box proprietary human task service (different from task service we implemented).

5.5. Summary of results

The integrated trusted processes framework meets the requirements we have stated for a trusted process in consumer-facing B2B network in section 3.2. The Trusted Process Provider extended the Circle of Trust architecture for business process automation and management. Our approach has advantages over traditional hard-coded business process applications in the Circle of Trust both in terms of functionality and efforts required for development lifecycle of trusted processes. From the functionality viewpoint, we notice that a central infrastructure that can be trusted by all the stakeholders to support privacy compliance and event monitoring are missing in the current Liberty Alliance specifications. In terms of efforts to develop, BPEL vendor tools can substantially eliminate the writing of BPEL code and help debug processes at a business process level. BPEL processes provide more agility and flexibility to changes over hard-coded processes

in CoT. The complexity of scripting increases as we add trusted features both in terms of BPEL activity steps and data manipulation involved.

However, there are certain shortcomings in BPEL standard to fully integrate with CoT. We saw that Oracle Process Manager supports our framework well with its extensions to support SOAP handling, WS-addressing along with the feature rich designer and management console tools. Finally, we note that Liberty Alliance is complex set of specifications, but for our framework only a core subset was required.

Chapter 6. Conclusions

6.1. Objectives Achievement Verifications

In section 1.2, we presented the key objectives of our research. We verify the accomplishment of these objectives in this section.

Objective 1: Leverage federated identity standards to securely manage consumer identities and authentication in B2B network.

Liberty Alliance federated identity standards were leveraged into the ITPF. The Liberty Alliance Identity Provider protects a consumer's identity at service provider's site through a system of pseudonymous identifiers. The IDP SSO service securely authenticates the consumers in a CoT.

Objective 2: Protect consumer data through a system of permission-based consent for data sharing between businesses in a B2B network.

The ITPF uses Liberty Alliance Discovery Service to control data-sharing between service providers while protecting consumer's identity. The DS checks for consumer's consent at a PDP before allowing the sharing of data. Our framework also has a Task Service to dynamically obtain consumer consent before sharing their data.

Objective 3: Help collaborating businesses manage their processes through a system of event monitoring and audit trails for documentation of privacy compliance and performance management.

The Event Stat Service in ITPF helps tracking business activity events during run-time of a process for performance management. The use-case scenario in section 4.1 demonstrated the use of Event Stat Service to monitor events for the measurement of a key performance indicator.

Also as outlined in section 2.1.4 , to comply with privacy legislations, organizations must document their use of personal data and provide individuals access to their data to challenge its accuracy and use. In our framework, the Audit Trail Service and Privacy Portal help address these requirements respectively. Our compliance mechanism focused on events related to the sharing of consumer data between distinct organizations in B2B processes. However, we didn't document information about consumer consent and access control policies that are being used by the Discovery Service / B2B process to enforce the sharing of data.

Objective 4: Provide transparency through privacy auditing reports.

The Privacy Portal in our framework enables consumers and privacy officers to run audit trail reports on the Audit Trail Service and gain insights to data-sharing activities of service provides.

Objective 5: Support both fully automated processes and interactive processes that involve human participation.

The Task Service helps support BPEL process steps that required human interactivity. The two use-case scenarios in section 4.1 and 4.4 were used to demonstrate and analyze the implementation of fully automated and interactive processes using our framework.

6.2. Summary of Contributions

In this thesis, we presented an integrated trusted processes framework for managing trusted B2B processes that provide information-rich services to consumers. The basic approach was to integrate and extend BPEL related standards for business process automation with Liberty Alliance standards that created a SOA-based Circle of Trust for B2B networks.

In section 3.1 and 3.2, we first identified issues that typically arise in cross-organizational processes and accordingly categorized the requirements of a B2B process from the point of view of its two main stakeholders – the consumer and the business. By identifying these requirements, we were able to conceptualize our notion of a trusted B2B process.

The Federated Identity Standards from Liberty Alliance was recognized as the key building block in our framework. The Circle of Trust architecture was leveraged for the B2B network with an IDP to protect federated identity of a consumer through pseudonymous identifiers, and the Discovery Service to enable data-sharing between service providers on a non-identifying basis in accordance with consumer's wishes.

We introduced several new components as part of a new "Trusted Process Provider" into this Circle of Trust architecture in order to provide support for business process management and automation (section 3.3). An Audit Trail Service (and privacy portal) was used to track privacy compliance. An Event Stat Service acted as a global event repository for business process monitoring where service providers can run performance management reports. Finally, a Task Service (and task portal) helped business process manage steps that required participation of consumer or employees in B2B network.

In sections 4.3.3 and 4.4.3, we described the BPEL definitions of trusted processes to

illustrate the mechanisms for sharing of data, documenting compliance, logging events and human interaction. Both synchronous and long-running asynchronous processes were shown using two different use-case scenarios. Details about setting Liberty SOAP headers, obtaining EPRs from DS, passing security tokens were explained to fully understand how BPEL integrates with the Circle of Trust services.

As part of our research, we investigated existing implementations of Liberty Alliance specifications and compared their capabilities and ease of use in Section 5.3. Section 4.5 described the CoT prototype which was built for the implementation of our use-case scenarios. The prototype is helpful in identifying the features of Liberty (and other related) specifications that are most relevant to our framework objectives. In Section 5.3.1, we highlighted the parts of Liberty SOAP binding that are important in our framework.

Section 0 identified three gaps with WS-BPEL standards for integration with Liberty specifications. First, BPEL lacks the ability to manipulate SOAP header information and thus cannot fully conform to Liberty SOAP binding. Next, we saw that support for WS-Addressing is required to create a loosely-coupled Task service. Finally, there is no option to digitally sign SOAP header information in outgoing messages.

6.3. Conclusions

As shown in section 5.1, the integrated framework (with the addition of Trusted Processes Provider) provides a richer feature set than the Liberty Alliance framework for addressing the requirements of collaborative trusted processes. The use of a business process automation language such as BPEL eliminates the need to write hard-coded composite applications for developing information-rich services inside CoT. Section 5.2 demonstrates its

advantages in terms of efforts to create, debug and maintain trusted processes. Service Providers can utilize the BPEL designer tools to rapidly build new process workflows without writing any implementation code. Another advantage is the possibility to debug at the business process level.

By leveraging the WS-BPEL standards with Liberty Alliance specifications, our integrated framework builds upon several leading industry standards from OASIS and WC3. For example: SAML (from OASIS) is used to exchange security information between IDP and Service Providers, WS-Security tokens (from OASIS) are transmitted in message headers, Liberty Services have a WSDL interface (from WC3) and so on. The use of standards when adopted ensures interoperability which is especially important in B2B networks to facilitate partner integration between heterogeneous systems.

However, as identified in section 0 there were gaps in the WS-BPEL standards that required proprietary extensions from Oracle PM Engine for our use-case scenario implementations. These extensions or similar workarounds need to be incorporated into WS-BPEL standards to ensure full interoperability of our trusted BPEL processes between BPEL engines from different vendors. Also as noted in section 5.3, the tool support for Liberty Alliance needs to further mature to help service providers leverage the CoT architecture. Libraries could be provided that allow them to seamlessly process messages as defined by Liberty SOAP binding to integrate their web services into CoT architecture. Such a library in turn would need to be built upon other toolkits developed for WS-* standards, SAML and so on which are used by Liberty SOAP binding.

Finally, we realize the limitations of the scope of our research based on a CoT prototype. As with any initial research, we did not have the time or resources to use the entire spectrum of

specifications from Liberty Alliance and instead decided to abstract its complexity through a prototype. The goal of the prototype was to use the core subset of Liberty standards for proof-of-concept code to validate and analyze our framework. Accordingly, some details and mechanisms that are part of Liberty Specifications and related standards were simplified or not included in our prototype implementation. For example, we simplified the SAML schema from OASIS to represent a SAML assertion and the EPR schema from W3C for end-point references in our prototype. However in spite of these simplifications, we believe that the complexity of scripting trusted processes as shown in section 5.2.4 should remain relatively unaffected. For example: whether or not SAML tokens conform to exact schema, the BPEL syntax for extracting SAML tokens from EPR and copying them into the WS-Security header remains the same (i.e. copying a chunk of XML data from one variable to another). The complexity increases only if any additional optional headers defined in Liberty SOAP binding are utilized.

6.4. Future work

Scripting a long trusted process involving repetitive calls to Discovery Service, Audit Trail Service and so on can get quite tedious. More work is required to analyze if these common CoT services can be handled automatically. Additional configuration such as a common data-sharing and event model can be provided. Indeed Liberty Alliance provides a specification [LibertyMetaData2006] for describing and discovering such common meta-data in a Circle of Trust.

By leveraging the common data-sharing and event model, the CoT BPEL engine can be configured to automatically obtain EPRs from Discovery Service before sharing data with a Liberty Service. Similarly, invocations to ATS for logging audit events can be automated.

Common events for performance monitoring could possibly be defined in a similar approach, but this would require more shared analysis between organizations to define the events for monitoring shared processes. In addition, the CoT BPEL engine can implicitly set the headers defined by Liberty SOAP binding. For example, tokens in an EPR can be automatically set as a WS-Security header while invoking the Liberty Service. Oracle PM Engine already handles WS-addressing headers in such a transparent fashion.

Automation for calls to DS, ATS, ESS and the implicit handling of header information can considerably ease the task of scripting a trusted process. However, such proprietary extensions to the CoT BPEL engine can affect the interoperability of trusted process. Alternatively, a tool can be built that helps transforms a "straw man" version of the process into its trusted version. In order to do so, the process designers could be allowed to insert extra annotations alongside the XML syntax in the process definitions that can be translated into BPEL code for making calls to DS, ATS and ESS. Prior research done in [Charfi2005, Charfi2004] has investigated the use of an aspect-oriented approach in a BPEL process to provide separation of concerns between business process logic and quality of service issues. It can be leveraged to design a BPEL engine that automates the handling of our framework components.

The Task Service can be enhanced to directly notify users through email and so on upon the assignment of a new task to their task list. Liberty Alliance describes a similar Interaction Service [LibertyInteract2006] that sends notifications to users. However, privacy concerns must be considered since Task Service would then require access to user's personally identifiable information.

Our framework included a policy decision point where users can limit data-sharing between service providers. A web portal that is part of IDP can be built to allow users to specify access-

control policies using a privacy preference language. Related work done earlier in [Alsaleh2006a] can be leveraged into our framework.

Finally, further work needs to be done to evaluate quality of service issues related to performance and scalability of our framework in a real-world Liberty Alliance deployment.

References

[1] **[Alsaleh2006a]** M. Alsaleh, *Enhancing Consumer Privacy in Identity Federation Architectures*, Thesis, University of Ottawa, September 2006.

[2] **[Alsaleh2006b]** M. Alsaleh, C. Adams, "Enhancing Consumer Privacy in the Liberty Alliance Identity Federation and Web Services Frameworks". In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, United Kingdom, June 2006

[3] **[Arkin2005]** A. Arkin et al. Web Services Business Process Execution Language Version 2.0. Technical report, Oct 2005. http://www.oasis-open.org/apps/org/workgroup/wsbpel/.

[4] **[Bartel2002]** M. Bartel,J. Boyer, B. Fox, B. LaMacchia, and E. Simon. XML-Signature Syntax and Processing World Wide Web Consortium (W3C), February 2002. http://www.w3.org/TR/xmldsig-core.

[5] **[Baresi2005]** L. Baresi and S. Guinea. Towards Dynamic Monitoring of WS-BPEL Processes. In 5th International Conference on Service Oriented Computing, pages 269--282, 2005.

[6] **[Barbon2006]** Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti, "Run-Time Monitoring of Instances and Classes of Web Service Compositions," icws, pp. 63-71, IEEE International Conference on Web Services (ICWS'06), 2006.

[7] **[Bertino2006]** Elisa Bertino, Jason Crampton, Federica Paci, "Access Control and Authorization Constraints for WS-BPEL," icws, pp. 275-284, IEEE International Conference on Web Services (ICWS'06), 2006

[8] **[Boon1991]** Boon, S. and Holmes, J. The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. In R. Hinde and J. Groebel (Eds.). Cooperation and Prosocial Behavior. Cambridge University Press, Cambridge, UK: 1991, 190--211.

[9] **[Box2004]** D. Box, F. Curbera, et al. Web Services Addressing (WS-Addressing), Aug. 2004. W3C Member Submission, at http: //www.w3.org/Submission/ws-addressing/.

[10] **[Casassa2002]** M. Casassa Mont, P. Bramhall, M. Gittler, J. Pato, O. Rees, "Identity Management: a key e-business enabler", PL-2002-164, SSGRR2002s, L'Aquila, Italy, 2002

[11] **[Casassa2003]** M. Casassa Mont, P. Bramhall, J. Pato, On Adaptive Identity Management: The next generation of Identity Management Technolgies, HP Labs Technical Report, HPL-2003-149, 2003

[12] **[Casati2000]** Casati, F. and Shan, M. 2000. Process Automation as the Foundation for E-Business. In Proceedings of the 26th international Conference on Very Large Data Bases (September 10 - 14, 2000).

[13] **[Charfi2005]** Charfi, A. and Mezini, M. 2005. An aspect-based process container for BPEL. In Proceedings of the 1st Workshop on Aspect Oriented Middleware

Development (Grenoble, France, November 28 - December 02, 2005). AOMD '05, vol. 118. ACM, New York, NY DOI= http://doi.acm.org/10.1145/1101560.1101564

[14] **[Charfi2004]** A. Charfi and M. Mezini. Aspect-oriented web service composition with AO4BPEL. In Proceedings of the European Conference on Web Services (ECOWS), volume 3250 of LNCS. Springer, 2004.

[15] **[Chellappa2005]** Chellappa, R. K. and Sin, R. G. 2005. Personalization versus Privacy: An Empirical Examination of the Online Consumer's Dilemma. *Inf. Technol. and Management* 6, 2-3 (Apr. 2005), 181-202.

[16] **[Curbera2002]** Curbera, F., M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana (2002). Unraveling the web services web: an introduction to soap, wsdl, and uddi. Internet Computing, IEEE 6 (2), 86-93.

[17] **[EUPrivacy2002]** European Union Directive on Privacy and Electronic Communications. European Parliament,Brussels, Belgium, 2002. http://register.consilium.eu.int/pdf/en/02/st03/03636en2.pdf, Accessed 2008/04.

[18] **[Fichman2003]** Fichman, R. G. and Cronin, M. J. 2003. Information-rich commerce at a crossroads: business and technology adoption requirements. *Commun. ACM* 46, 9 (Sep. 2003), 96-102.

[19] **[Gassman2006]** B. Gassman, Who's Who in Business Activity Monitoring,4Q05, Gartner Research, 12 April 2006, p. 3.

[20] **[Hammer1994]** Hammer, M. and J. Champy (1994, April). *Reengineering the Corporation: A Manifesto for Business Revolution.* HarperBusiness.

[21] **[HIPAA1996]** Health Insurance Portability and Accountability Act (HIPAA), United States Congress, United States, 1996. http://aspe.hhs.gov/admnsimp/pl104191.htm,Accessed 2007/08.

[22] **[Holmes2008]** Ta'id Holmes, Martin Vasko, Schahram Dustdar, "VieBOP: Extending BPEL Engines with BPEL4People," pdp, pp. 547-555, 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), 2008

[23] **[Hughes2004]** J. Hughes et al., Technical Overview for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS SSTC, August 2004.

[24] overview-2.0-draft-01.

[25] **[IDFF2003]** Wason, T., eds., "Liberty ID-FF Architecture Overview"; version 1.2Liberty Alliance Project, March 2003.

[26] **[IDWSF2004]**Kemp, Y., eds., "Liberty ID-WSF Web Services Framework Overview", 2.0, Liberty Alliance Project, 2004.

[27] **[Imamura2002]** T. Imamura, B. Dillaway, and E. Simon. XML Encryption Syntax and Processing. World Wide Web Consortium (W3C), August 2002. http://www.w3.org/TR/xmlenc-core.

[28] **[Jablonski1996]** S. Jablonski and C. Bussler. WorkflowManagement: Modeling, Concepts, Architecture, and Implementation. International Thomson Computer Press,1996.

[29] **[Jasmine2005]** N. Jasmine, "BPM and SOA: Better Together", IBM White Paper, 2005, pp. 1-12.

[30] **[Jim2006]** Jim, H.B. Janelle, "Gartner Predicts 2007: Align BPM and SOA Initiative Now to Increase Chances of Becoming a Leader in 2010", Gartner Report, Nov 2006, pp. 1-4.

[31] **[Jones2000]** Jones, S., Wilikens, M., Morris, P., and Masera, M. 2000. Trust requirements in e-business. Commun. ACM 43, 12 (Dec. 2000), 81-87.

[32] **[Ken2006]** V. Ken, P. Henry, "The Forrester Wave: Integration-Centric Business Process Management Suites", Forrester Research Inc. Report, Q4, 2006, pp. 1-16.

[33] **[Kloppmann2005]** M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic. WS-BPEL Extension for People - BPEL4People. A Joint White Paper by IBM and SAP. ftp://www6.software.ibm.com/software/developer/library/ws-bpel4people.pdf. 2005

[34] **[Koch2005]** Koch, M., and Möslein, K.M. (2005) "Identity Management for Ecommerce and Collaborative Applications", *International Journal of Electronic Commerce / Spring 2005*, Vol. 9, No. 3, pp. 11–29.M.E. Sharpe Inc., 2005.

[35] **[Laurence2007]** Laurence Goasduff,Carina Forsling , Garner Press Release, 2007 . http://www.gartner.com/it/page.jsp?id=502645, Accessed 2008/04.

[36] **[Lawrence1997]** P. Lawrence, editor. Workflow Handbook 1997, Workflow Management Coalition. John Wiley and Sons, New York, 1997. (www.wfmc.org)

[37] **[Libera2003]** Della-Libera, Giovanni et al., "Federation of Identities in a Web Services World", Jul. 2003, IBM Corporation and Microsoft Corporation (p. 1-15).

[38] **[Liberty]** Liberty Alliance project. http://www.projectliberty.org/. Last visited: May 2008.

[LibertyDisco2006] Hodges, Jeff, Cahill, Conor, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). http://www.projectliberty.org/specs , Accessed 2007/08

[39] **[LibertyDST2006]** Kellomäki, Sampo, Kainulainen, Jukka, eds. "Liberty ID-WSF Data Services Template," Version 2.1,Liberty Alliance Project (30 July, 2006). http://www.projectliberty.org/specs Accessed 2007/08

[40] **[LibertyInteract2006]** Aarts, Robert, Madsen, Paul, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0, Liberty Alliance Project (30 July, 2006). http://www.projectliberty.org/specs, Accessed 2007/08.

[41] **[LibertyMetaData2006]** Davis Peter, ed. "Liberty Metadata description and discovery Specification," Version 1.1,Liberty Alliance Project (2006). http://www.projectliberty.org/specs, Accessed 2007/08.

[42] **[LibertyPeople2006]** Koga, Yuzo, Madsen, Paul, eds. "Liberty ID-WSF People Service Specification," Version 1.0,Liberty Alliance Project (30 July, 2006). http://www.projectliberty.org/specs Accessed 2007/08.

[43] **[LibertyPrivacy2004]**Landau, S., eds., "Liberty ID-WSF Security & Privacy Overview"; version 1.0, Liberty Alliance Project, 2003, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications. Accessed February 2007.

[LibertySechMech2006] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version v2.0, Liberty Alliance Project (30 July, 2006). http://www.projectliberty.org/specs , Accessed 2007/08

[44] **[LibertySecMechSAML2006]** Hirsch, Frederick, eds. "ID-WSF 2.0 SecMech SAML Profile," Version 2.0-errata-v1.0, Liberty Alliance Project (08 November, 2006). http://www.projectliberty.org/specs, Accessed 2007/08

[45] **[LibertySOAPBinding2006]** Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Liberty ID-WSF SOAP Binding Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007).http://www.projectliberty.org/specs, Accessed 2007/08.

[46] **[Live2006]** "Introduction to Windows Live ID", April 2006, Windows Live Development Center. http://msdn.microsoft.com/en-us/library/bb288408.aspx . Accessed 2008/04

[47] **[Lublinsky2007]** Boris Lublinsky,"Defining SOA as an architectural style",IBM article,

[48] http://www-128.ibm.com/developerworks/architecture/library/ar-soastyle/, 2007. Accessed 2008/04.

[49] **[Nadalin2004]** A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo. OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), Mar. 2004. OASIS Standard 200401, at http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

[50] **[OASIS]** Organization for the Advancement of Structured Information Standards.

http://www.oasis-open.org

[51] **[Ouyang2006]** Chun Ouyang, Marlon Dumas, Arthur H.M. ter Hofstede, Wil M.P. van der Aalst, "From BPMN Process Models to BPEL Web Services," *icws*, pp. 285-292, IEEE International Conference on Web Services (ICWS'06), 2006.

[52] **[Papazoglou2003]** Papazoglou, M. P. and Georgakopoulos, D. 2003. Service Oriented Computing : Introduction. *Commun. ACM* 46, 10 (Oct. 2003), 24-28.

[53] **[Parr2001]** B. Parr, R. Villars, "Digital Identities: The Coming Struggle for the Future of the net", IDC, 2001

[54] **[Peppers1999]**D. Peppers, M. Rogers and B. Dorf, Is your company ready for one-to-one marketing? Harvard Business Review (1999) 3–12.

[Peyton2004] L.Peyton, M. Nozin, "Tracking Privacy Compliance in B2B Networks", Sixth International Conference on Electronic Commerce,Delft, The Netherlands, October,2004.

[55] **[PIPEDA2000]** The Personal Information Protection and Electronic Documents Act (PIPEDA), Department of Justice, Canada, 2000.http://laws.justice.gc.ca/en/P-8.6/text.html,Accessed 2007/08

[56] **[Pourshahid2008]** Alireza Pourshahid, Daniel Amyot, Liam Peyton, Sepideh Ghanavati, Pengfei Chen, Michael Weiss, Alan J. Forster, "Toward an Integrated User Requirements Notation Framework and Tool forBusiness Process Management," *mcetech*, pp. 3-15, 2008 International MCETECH Conference on e-Technologies (mcetech 2008), 2008

[57] **[Rantnasingam2002a]** Ratnasingam, P., Pavlou, P. (2002a), "Technology trust: the next value creator in B2B electronic commerce", International Resources Management Association Conference – Washington, Seattle.

[58] **[Recordon2006]** Recordon, D. and Reed, D. 2006. OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the Second ACM Workshop on Digital Identity Management* (Alexandria, Virginia, USA, November 03 - 03, 2006). DIM '06. ACM, New York, NY, 11-16.

[59] **[Rud2006]** Dmytro Rud, Andreas Schmietendorf, Reiner Dumke, "Performance Modeling of WS-BPEL-Based Web Service Compositions," scw, pp. 140-147, IEEE Services Computing Workshops (SCW'06), 2006

[60] **[Shankar2002]** Venkatesh Shankar, Fareena Sultan and Glen L. Urban 2002. Online trust: a stakeholder perspective, concepts, implications, and future directions.,The Journal of Strategic Information Systems ,Volume 11, Issues 3-4, December 2002, Pages 325-344

[61] **[SOAP]**Simple Object Access Protocol (SOAP) 1.1, W3C Note, http://www.w3.org/TR/2000/NOTE-SOAP-20000508 . Accessed 2008/04.

[62] **[Thomas2007]** Jacques Thomas, Federica Paci, Elisa Bertino, Patrick Eugster, "User Tasks and Access Control overWeb Services," icws, pp. 60-69, IEEE International Conference on Web Services (ICWS 2007), 2007

[63] **[Tran2004]** Thomas Tran, Robin Cohen, "Improving User Satisfaction in Agent-Based Electronic Marketplaces by Reputation Modelling and Adjustable Product Quality," *aamas*, pp. 828-835, Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS'04), 2004

[64] **[UDDI]**Universal Description, Discovery, and Integration (UDDI), http://www.uddi.org/.

[65] **[Varney2005]** Varney, C. and Sheckler, V., "Deployment Guidelines for Policy Decision Makers". September 2005, Version 2.9, Liberty Alliance Project. Available from: http://www.projectliberty.org/about/whitepapers.php.

[66] **[W3C]** W3C World Wide Web Consortium, http://www.w3.org/

[67] **[Weiss2005]**Weiss, M. and Amyot, D. Business Process Modeling with URN. International Journal of E-Business Research, 1(3), 2005, pp. 63–90.

[68] **[White2004]** S. A. White. Business Process Modeling Notation (BPMN) Version 1.0. Business Process Management Initiative, BPMI.org, May 2004.

[69] **[Wil2003]** Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and Mathias Weske (2003), Business Process Management: A Survey,International Conference, BPM 2003 Eindhoven, The Netherlands, June 26–27, 2003 Proceedings.

[70] **[Wilikens1998]** Wilikens, M., Morris, P. and Masera, M., Eds. 1998. Defining the European.Dependability Initiative: A Strategy Document. European Communities.EUR Report, EUR 18139 EN, May 1998.

[71] **[Woodley2005]** Thomas Woodley and Stephane Gagnon,"BPM and SOA: Synergies and Challenges",Published in Springer Berlin / Heidelberg Web Information Systems Engineering – WISE 2005, Volume 3806, 2005

[72] **[WSDL]**Web Services Description Language (WSDL) 1.1, W3C Note, http://www.w3.org/TR/wsdl.html. Accessed 2008/04.

[WS-I]Web Services Interoperability Organization ,http:// www.ws-i.org

[73] **[Xong2003]** Li Xiong, Ling Liu, "A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities," *cec*, p. 275, 2003 IEEE International Conference on E-Commerce Technology (CEC'03), 2003

[74] **[XPATH1999]** J. Clark, S. DeRose, "XML Path Language (XPath) Version 1.0", W3C Recommendation, http://www.w3.org/TR/xpath, November, 1999.

[75] **[Yip2006]** Yip, F. Ray, P. Paramesh, N. (2006) 'Enforcing Business Rules and Information Security Policies through Compliance Audits;XISSF - A Compliance Specification Mechanism',Business-Driven IT Management,BDIM '06, The First IEEE/IFIP International,ISBN: 1-4244-0176-3, pp. 81- 90.