



**British Columbia
Broader Public Sector
Identity Management Architecture Project
Architecture Document**

Author(s):	Michael DeSandoli Sergio Fiszman Alan Harbitter Dick Hardt Don Schmidt Stephen Skordinski
Creation Date:	April 12, 2007
Last Updated:	August 10, 2007
Version:	FINAL Draft 3.0

Table of Contents

1	Introduction.....	1
1.1	Background.....	1
1.2	Methodology.....	2
1.3	Principles.....	3
1.4	Scope.....	5
2	Architecture.....	6
2.1	Core Architecture.....	8
2.2	IA Possesses Claims Prior to Resource Request.....	11
2.3	Multiple Relying Parties.....	12
2.3.1	Multiple Relying Parties; AP-asserted Claims.....	13
2.3.2	Multiple Relying Parties; Self-assertion.....	15
2.4	Late Claim Validation.....	16
2.5	Delegation.....	18
2.6	Offline Requests.....	19
3	Lexicon.....	19
4	Standards and Architectural Recommendations.....	20
4.1	Core Architecture Interactions.....	20
4.2	IA Possesses Claims Prior to Resource Request.....	21
4.3	Multiple Relying Parties.....	22
4.4	Multiple Relying Parties; Self-assertion.....	24
4.5	Late Claim Validation.....	25
4.6	Delegation.....	25

List of Exhibits

Exhibit 1 - Architecture Components 8

Exhibit 2 - Core Architecture Interactions 9

Exhibit 3 - IA Possesses Claims Prior to Resource Request..... 11

Exhibit 4 - Multiple Relying Parties; AP-asserted Claims..... 13

Exhibit 5 - Multiple relying parties; Self-assertion..... 15

Exhibit 6 - Late Claim Validation 17

Exhibit 7 - Delegation..... 18

Exhibit 8 - Offline Requests 19

1 Introduction

This document describes the result of collaboration between government and industry to develop an identity management architecture. A fundamental goal of this architecture is to facilitate the delivery of online governmental and educational services to the people of British Columbia (B.C.). This first section provides a project background, an overview of the scope of the architecture, and a summary of the methodology used to define the architecture. The second section presents the architecture itself by describing its components and illustrating how they interact to support typical service delivery.

1.1 Background

Over the past three decades, the Provincial Government and Broader Public Sector (BPS) organizations have invested heavily in the automation of business processes. Much of this investment has taken place only to meet a single organization's unique local needs. It was usually done with limited consideration towards building interoperable cross-organizational information architecture.

To achieve the broader goals of the Province and improve service delivery, a mechanism must be created to securely share information between organizations and systems. An important piece of this mechanism is the development of common cross-organizational standards for interoperable identity management.

This is a complex issue to resolve when one considers the spectrum of public and private sector stakeholders involved: policy, management and administrative issues; privacy and security requirements for the management of access to information; and the various technologies in place. Compounding this architectural challenge is the fact that the Information Technology industry does not have an inclusive "off the shelf" solution. This project will require the British Columbia public sector to work with industry to build upon existing international standards in the development of a business and technology architecture to meet the secure information sharing needs of the BPS.

In January 2006, the Secure Identity Aware Network (SIAN) framework project was initiated by the Government of British Columbia Office of the Chief Information Officer. The goal of SIAN is to meet the government's "need to share information and better integrate its services." This framework document, "Secure Identity Aware Network, Component Model and Business Case V1.0," was released in July 2006 as a "vision of a framework for supporting information sharing."

SIAN suggests some principles and minimal requirements that, if followed, will make it easier to access information across organizational lines while improving the privacy and security of that information. This ability to share and integrate information has an eye to a future where interoperability exists not only between the provincial government and the BPS but also with their private sector partners and ultimately, the citizens and businesses that interact with government. If government is to implement this vision of an Identity Aware Network, then a rigorously defined identity management architecture is needed to ensure interoperability.

1.2 Methodology

The collaboration between government and industry is an important part of the methodology for developing this architecture. The collaborative approach helps produce an architecture that both meets the government's basic requirements for identity management and can be implemented with commercially available software tools and standards. The methodology follows a five step process:

1. Form a working group with government and industry participants. This working group includes lead government representatives from the Office of the Chief Information Officer, Ministry of Education, Ministry of Health, WorkSafeBC, BC Hydro, Attorney General, Ministry of Employment and Income Assistance, and leading universities, along with subject matter experts from Bell, CA, Deloitte, IBM, Microsoft, Nortel, Novell, Oracle, Siemens, Sun Microsystems, Sxip, and Telus. The government agencies will be users of the implemented identity management system. The companies are leaders in this market and are actively developing products that will support the implementation of online identity management systems. Their participation will help ensure that the architecture can be implemented with commercially available technologies and products.
2. Develop and distribute use cases. Fifteen government and academic stakeholder organizations participated in the definition and documentation of use cases for identity management. The working group relied on these use cases as the starting point for defining requirements.
3. Conduct the first workshop to establish common objectives, and assign preliminary responsibilities for requirements definition. Sxip, the company responsible for managing industry participation in this program, conducted a workshop to introduce the government and industry participants and start the requirements definition phase. Following this workshop, a subgroup developed a preliminary technical requirements document.
4. Conduct a second workshop to review and update the requirements and initiate architecture analysis and discussion. The second workshop resulted in a preliminary architecture that was developed collaboratively by government and industry working group participants. After this workshop, a subgroup drafted this document which presents the architecture.
5. Conduct a third workshop to conclude discussions on the architecture and identify candidate standards and implementation strategies. The product of the third workshop and its follow-up activities was an updated draft of this architecture document as well as documented candidate implementation standards and strategies.

1.3 Principles

The description of the components and their interaction contained in this document should provide an understanding of the identity management architecture and how it works. However, there are important questions that lie between the lines of this description, such as: Why was this architecture selected? How is it different from other approaches? What are the advantages of this architecture? To address these questions, it may be more useful to review the underlying principles of the architecture. The government/industry working group established these principles early on to provide a basis for selecting among architectural alternatives. The principles complement the requirements analysis in guiding decisions as the architecture was developed and refined. Seven of these driving principles are described below:

The architecture must be user-centric.

The user, or more specifically the B.C. citizen, is at the center of this architecture. Wherever possible, identity information flows through the user at the request of the user. This allows the B.C. citizens to control how much information they reveal about themselves in exchange for access to services and information.

The architecture must assist users in protecting privacy, and limit the amount of exposed personal information to the minimum required.

User centricity certainly helps the citizen protect his or her privacy. In addition, the architecture provides mechanisms to allow the user to determine how much information is required to access a government service or resource. The user may then decide if s/he is willing to provide that information in exchange for services.

The architecture must be forward-looking, providing a target for existing systems to migrate towards.

We do not believe that this architecture can be implemented in its entirety with present day standards and products. However, we have designed it so that the majority of its functions can be implemented. Furthermore, the full architecture aligns with the direction in which the industry and standards committees are headed. As a result, the architecture provides a long term vision that can be implemented, in part, with current technology tools.

The components of the architecture must be as loosely coupled as possible to allow the identity management system to scale.

The target community for the identity management architecture will potentially include millions of users and hundreds of government agencies. There is a loose coupling between the major components of this architecture. The number of each component can be increased without limit as the system expands to include more systems and users. If there is less dependency among components, each component can also be changed without creating big impacts on the system as a whole. So loose coupling also has benefits in enabling the gradual implementation of the components, as well as allowing a graceful convergence toward evolving standards.

The architecture must support and promote implementations based on open standards.

The government/industry working group constructed the architecture with an eye toward what can be implemented with current open standards. Because of the prevalence of the Internet and widespread adoption of Internet protocols, the working group considered operation over the Internet to be a minimum mandatory requirement. This focus on open identity standards-based implementations will make interoperability and integration among online government applications easier to accomplish. In addition, adherence to this principle will result in an architecture that is vendor-agnostic and eliminates reliance on a single source for products and technology services.

The architecture must provide the flexibility to meet a broad range of identity-aware applications.

The working group recognized early on that the architecture would have to support a broad array of applications with an equally broad array of identity requirements. Some applications would need only minimal information about a user to provide a service. Perhaps the user would only need to establish an identity that was unique for that application without further third party proofs. Other applications may require more rigor with multiple pieces of corroborating identity information to definitively prove that the individual requesting services was authentic. Consequently, the architecture is claims-based. In other words, users present service providers with validated claims concerning identity. The identity claims can vary in subject and content, and can be validated by a third party with varying degrees of rigor. This permits a wide range of identity-aware applications to be supported.

The architecture must address high priority issues such as identity theft, identity proliferation, and inconsistent representation of identity across multiple government services.

In addition to the more lofty principles, there are current, real problems that motivate the need for identity management architecture. One of the most publicized of these is the growing problem of identity theft. The architecture includes several concepts such as an overarching “trust model” and “authoritative parties” that thwart identity theft and allow stolen identities to be more easily discredited. It also includes the concept of an “identity agent” as a tool for a user to help manage multiple identities, among other things. Finally, because of its scalability and flexibility, the architecture can provide the framework to build consistent identity management procedures across the many government services that require reliable identity information to function.

This architecture was selected to meet the above principles and satisfy the identity requirements of B.C. government services. Its focus on user-centricity, while not an entirely new concept, differentiates it from many other identity architectures which focus on large central software applications or administration bureaucracies. This identity management architecture has the advantage that it meets the identity requirements of B.C. government services and also helps safeguard the privacy of its citizens.

1.4 Scope

This architecture defines the terminology, principles, roles, and interactions among the key participants in online government services as they exchange the necessary identity claims and related information. Subsequent analysis will also identify standards and implementation strategies for those who wish to build operational systems that adhere to the guidelines of the architecture.

The scope of the identity management architecture does not include:

- Specifics of the government or private sector software applications that use identity management services,
- Specifics of the access control schemes used by government or private sector software applications,
- Support for all legacy applications. The architecture should be inclusive of legacy applications where possible, but is not required to be compatible with all existing applications.

Trust Model

As the architecture is described, it will become clear that there is the need for an electronic trust model. There are several options for implementing such electronic trust. In defining the architecture, we assume that such a trust model exists, but do not assume a specific implementation.

2 Architecture

The architecture is constructed around the concept of claims-based authentication. It is designed to support a loose coupling of the components, which allows a scalable number of organizations and principals to leverage the architecture for their identity management needs. As illustrated in Exhibit 2-1, the architecture relies on the interaction of five base components:

1. Authoritative Party (AP): An entity whose authority to make claims is recognized by one or more relying parties. Claims made by recognized authoritative parties are used by relying parties to make access control decisions.

2. Relying Party (RP): An entity that consumes claims to satisfy policy. The RP has control of information services and resources in which the principal is interested. A government ministry might be an RP.

3. Identity Agent (IA): An entity that facilitates the distribution of claims. The IA serves a principal (e.g., a citizen of B.C.) within the larger system by interacting with the AP and RP and managing claims so that the principal may satisfy information and service requests. Since it is likely that digital signature will play a role in the establishment of electronic trust, the IA may make the principal able to digitally sign information.

4. Root Authorities/Trust Model: The structure and mechanisms used to establish electronic trust among entities that require it. The trust model within the architecture is capable of supporting a number of functions. While the explicit implementation of the model is out of scope, the trust model implementation will enable a) IAs, RPs, and APs to validate claims, b) requesting parties to look up APs, and c) validating parties to ensure the claim is anchored to a root authority in the trust chain.

5. Claim: An attribute regarding a principal or transactional context. Claim types are unlimited, but are typically associated with: identification of principal, organization, user role and/or request context (network, device, location, and connection type). Claims are self describing and contain attributes such as originating AP. The architecture enables two methods for passing claims:

- The AP passes claims to the IA. Subsequently, the IA passes claims to RPs as required by the principal. The AP has no record of the RPs consuming the claims and can not collect any data on the principal's claim usage. This method is envisioned as the primary method for the passing of claims.
- The AP passes a claim location pointer to the IA. Then, the IA passes the pointer to the RP. The RP is then required to make a direct request to the AP for the claims. This method will only be used by the AP when there is a policy preventing claims from being released directly to the IA. This method allows the AP to strictly enforce where claims are passed. This also enables the AP to control how the principal uses claims.

The remainder of this section describes how claim requests are conceptually handled in the proposed identity management architecture. Each subsection includes an exhibit and a detailed interaction table explaining how the architecture supports the passing of identity claims in a given scenario. These scenarios are expected to occur frequently and are intended to satisfy the requirements derived and captured in the “BC IdM Requirements” document. Six scenarios are described:

1. **Core architecture:** This scenario describes the most typical interaction of architectural components in which a principal needs to pass identity claims to an RP to obtain access to services or information. For example, a B.C. citizen presents a valid drivers’ license to pay the fine for a traffic citation.
2. **IA possesses claims prior to resource request:** This scenario is similar to the core architecture interaction but reflects a situation where the principal may be using cached validated claims or un-validated claims and does not need to interact with the AP. For example, a B.C. citizen needs a validated birth certificate to access many different types of government services and decides to keep a copy stored on his/her laptop for convenience.
3. **Multiple relying parties:** This scenario describes a more complex interaction in which information or services are required from more than one RP. For example, a B.C. citizen may wish to look at a unified educational record that involves courses at several universities and requires retrieval of records held by those universities.
4. **Late claim validation:** The scenario describes a case in which a claim may be validated at a point in time after which it is first submitted. For example, a law enforcement officer subscribes to a service that provides information on who is being released from prison. The application that broadcasts this information will need to check that the principal is still a law enforcement officer before periodic updates are distributed.
5. **Delegation:** This scenario describes the situation in which a principal authorizes another entity to act on his/her behalf. For example, a doctor who wishes to allow their assistant to view and update patient records on his/her behalf. The doctor would need to delegate authority to the assistant in order for the assistant to access the appropriate patient records.
6. **Offline requests:** This scenario reflects a situation in which, for some reason, it is not possible for an IA to communicate with some or all of the other entities in the identify management system. For example, a field worker has information stored on his/her laptop that is sensitive and is controlled through the identity management system to prevent inappropriate access. However, this field worker does not have network connectivity and the identity management system must operate without the ability to communicate with an AP or an RP.

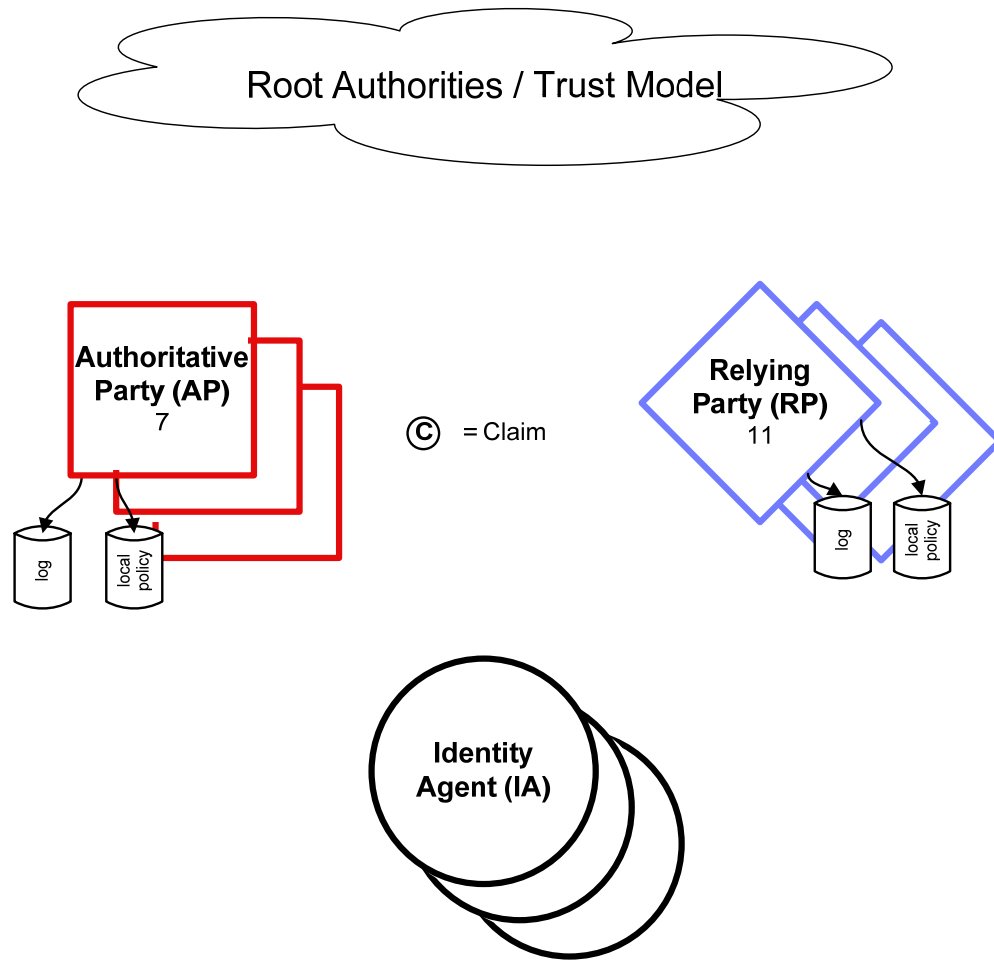


Exhibit 1 - Architecture Components

2.1 Core Architecture

Scenario: A principal attempts to access the services or resources of an RP. The principal does not possess the claims needed for access at the time of the request. A single AP and RP are required to process the request.

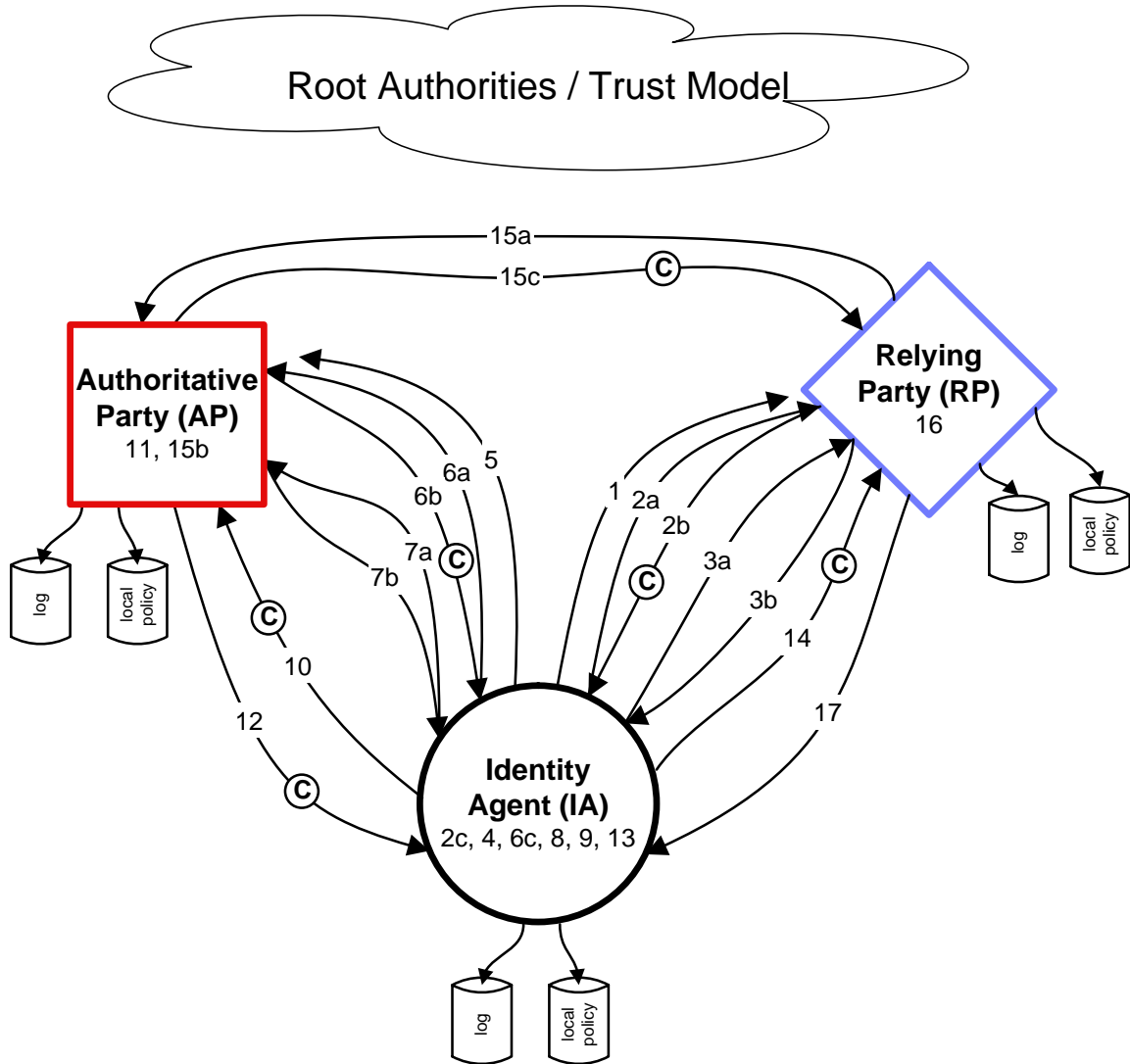


Exhibit 2 - Core Architecture Interactions

Core Architecture Interaction Descriptions		
Step #	Title	Description
1	IA identifies the RP	
2a	(Optional) IA requests RP claims	IA requests claims from the RP, in order to verify properties of the RP.
2b	(Optional) RP sends claims to IA	RP provides claims about itself.
2c	(Optional) IA validates RP claims	The IA validates the claims of the RP.

Core Architecture Interaction Descriptions		
Step #	Title	Description
3a	IA requests advertised policies from the RP	The IA sends a request to the RP for the advertised policies (such as required claims, optional claims, acceptable APs and T&Cs) required to access a resource.
3b	RP provides advertised policies	The RP responds by sending advertised policies required for access to the requested resource.
4	IA determines where to get claims required by the RP	The IA determines which APs can provide the claims requested by the RP from the RP itself or the discovery mechanism.
5	IA identifies AP	
6a	(Optional) IA requests AP claims	IA requests claims from the AP, in order to verify the properties of the AP.
6b	(Optional) AP sends claims to IA	AP provides claims about itself.
6c	(Optional) IA validates AP claims	The IA validates the claims of the AP.
7a	IA requests advertised policies from the AP	The IA requests advertised policies required by the AP, prior to the AP releasing claims about the principal.
7b	AP provides advertised policies	The AP responds by sending advertised policies required in order to make a policy decision.
8	IA obtains the claims required by AP policy for issuing tokens	IA may already possess the claims or obtain them from the principal or another AP.
9	Principal approves claims to be sent to AP	The IA must determine if it wants to present the claims to the AP.
10	IA provides claims AP needs & requests token/claims RP needs	The IA provides the claims to the AP.
11	AP evaluates claims from IA	The AP evaluates the claims provided by the IA, to determine if they are valid and meet local policy requirements.
12	AP provides claims RP needs in token issued to IA	The AP provides the requested claims to the IA.
13	Principal approves claims to be sent to RP	The IA must determine if it wants to present the generated claims to the RP.
14	IA provides claims to RP	The IA provides the claims to the RP. Claim could be the data itself or a reference to the data.
15a	(Optional) RP requests claims from AP	Claims passed by reference require a connection to the AP. RP validates the identity of the AP. The RP initiates the request using the data provided by the IA in the claim.
15b	(Optional) AP validates RP request	AP identifies and validates the RP. AP evaluates the RP's request.
15c	(Optional) AP provides claims to the RP	AP passes the requested claims to the RP.
16	RP determines access	The RP validates claims and applies local policy to determine if the principal should be granted access to the resource.
17	Access granted	The RP grants the principal access to the requested resource.

2.2 IA Possesses Claims Prior to Resource Request

Scenario: A principal attempts to access the services or resources of a RP. The principal already possesses the claims needed for access at the time of the request. A single AP and RP are required to process the request.

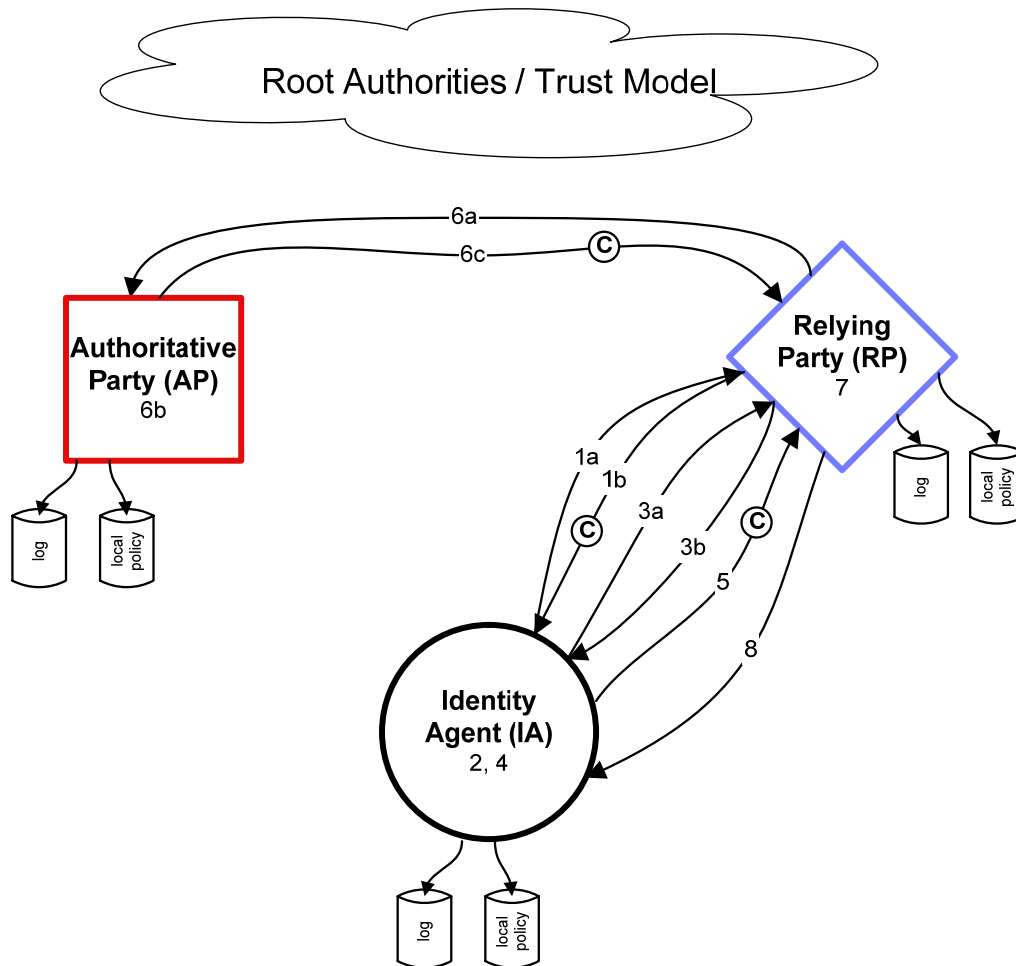


Exhibit 3 - IA Possesses Claims Prior to Resource Request

IA Possesses Claims Prior to Resource Request		
Step #	Title	Description
1	IA identifies the RP	
2a	(Optional) IA requests RP claims	IA requests claims from the RP, in order to verify properties of the RP.
2b	(Optional) RP sends claims to IA	RP provides claims about itself.
2c	(Optional) IA validates RP claims	The IA validates the claims of the RP.
3a	IA requests advertised policies from	The IA sends a request to the RP for the advertised policies (such

IA Possesses Claims Prior to Resource Request		
Step #	Title	Description
	the RP	as required claims, optional claims, acceptable APs and T&Cs) required to access a resource.
3b	RP provides advertised policies	The RP responds by sending advertised policies required for access to the requested resource.
4	Principal approves claims to be sent to RP	The IA must determine if it wants to present the requested claims to the RP.
5	IA provides claims to RP	The IA provides the required principal claims, or claim pointer to the RP.
6a	(Optional) RP requests claims from AP	Claim pointers require a connection to the AP. The RP initiates the request using the data provided by the IA in the claim pointer.
6b	(Optional) AP validates RP request	AP evaluates the RP's request.
6c	(Optional) AP provides claims to the RP	AP passes the requested claims to the RP.
7	RP determines access	The RP validates claims and applies local policy to determine if the principal should be granted access to the resource.
8	Access granted	The RP grants the principal access to the requested resource.

2.3 Multiple Relying Parties

Scenario: A principal attempts to access the services or resources of an RP. The principal does not possess the claims needed for access at the time of the request, and must use an AP to acquire the appropriate claims. RP1 receives the claims, but must accumulate additional information to satisfy the request. RP1 assumes the role of an identity agent, to acquire additional data on behalf the principal from RP2. A single AP and two RPs are required to process the request. It is assumed that RP1 and RP2 both require the same claims and RP1 maintains a list of claims required by RP2.

A necessary condition here is that the principal has to have a relationship with all RPs involved.

2.3.1 Multiple Relying Parties; AP-asserted Claims

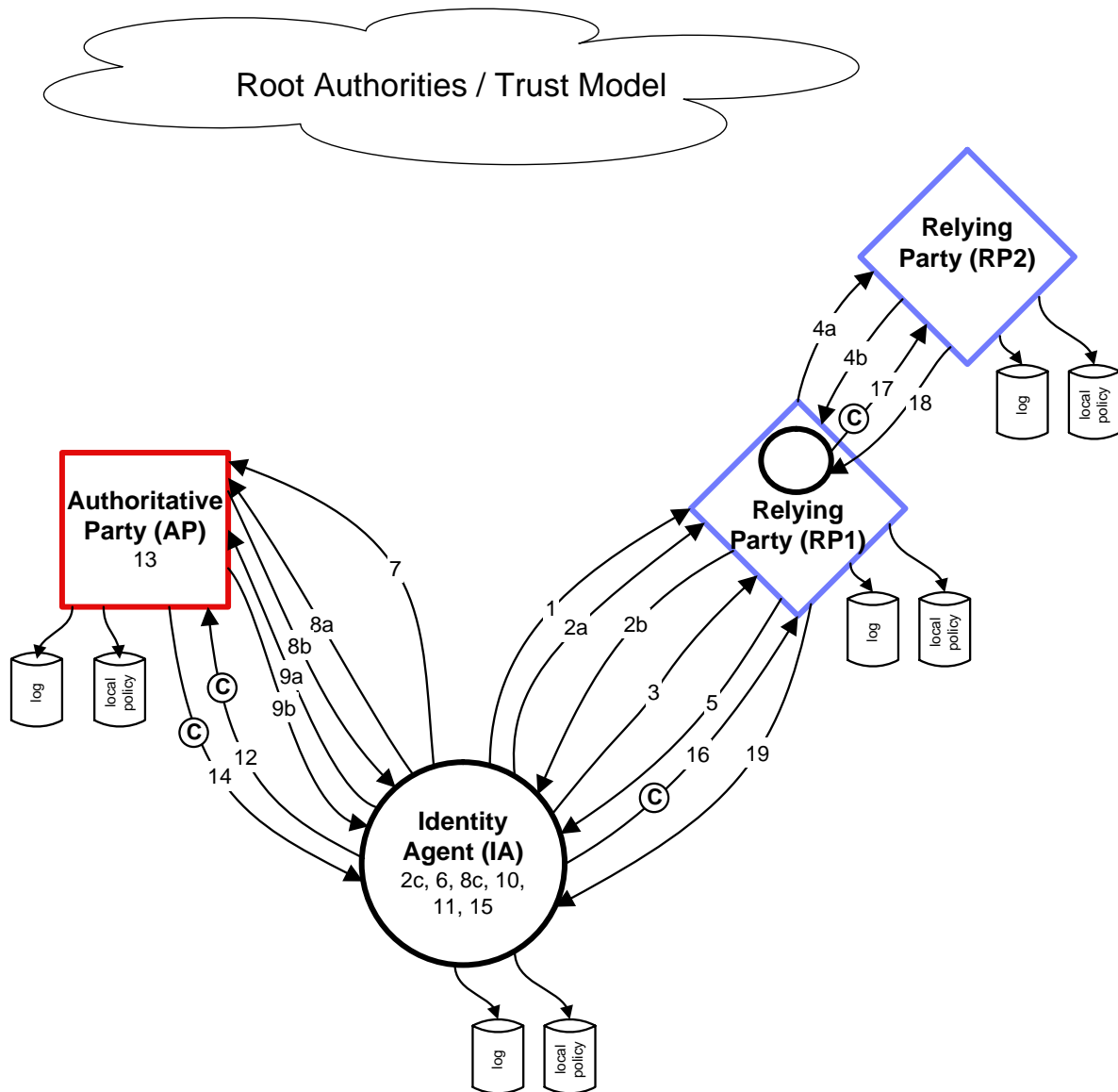


Exhibit 4 - Multiple Relying Parties; AP-asserted Claims

Multiple Relying Parties – AP-asserted Claims		
Step #	Title	Description
1	IA identifies the RP	
2a	(Optional) IA requests RP claims	IA requests claims from the RP, in order to verify properties of the RP.
2b	(Optional) RP sends claims to IA	RP provides claims about itself.
2c	(Optional) IA validates RP claims	The IA validates the claims of the RP.

Multiple Relying Parties – AP-asserted Claims		
Step #	Title	Description
3	IA requests advertised policies from the RP	The IA sends a request to the RP for the advertised policies (such as required claims, optional claims, acceptable APs and T&Cs) required to access a RP1 and RP2 resources.
4a & 4b	RP1 queries & receives advertised policies from RP2	
5	RP1 provides its own advertised policies and those of RP2	The RP responds by sending advertised policies required for access to the requested resource.
6	IA determines where to get claims required by the RP	The IA must determine if it wants to present the requested claims to the RP.
7	IA identifies AP	
8a	(Optional) IA requests AP claims	IA requests claims from the AP, in order to verify properties of the AP.
8b	(Optional) AP sends claims to IA	AP provides claims about itself.
8c	(Optional) IA validates AP claims	The IA validates the claims of the AP.
9a	IA requests advertised policies from the AP	The IA requests advertised policies required by the AP, prior to the AP releasing claims about the principal.
9b	AP provides advertised policies	The AP responds by sending advertised policies required in order to make a policy decision.
10	IA obtains the claims required by AP policy for issuing claims	IA may already possess the claims or obtain them from the principal or another AP.
11	Principal approves claims to be sent to AP	The IA must determine if it wants to present the claims to the AP.
12	IA provides claims AP needs & requests token/claims RP needs	The IA provides the claims to the AP.
13	AP evaluates claims from IA	The AP evaluates the claims provided by the IA, to determine if they are valid and meet local policy requirements.
14	AP provides claims RP needs in claim issued to IA	The AP provides the requested claims to the IA.
15	Principal approves claims to be sent to RP1/RP2	The IA must determine if it wants to present the generated claims to RP1 & RP2
16	IA provides claims to RP1	The IA provides the claims to RP1. Claim could be the data itself or a reference to the data. Claims are those required by RP1 and RP2.
17	RP1 passes claim to RP2	
18	RP2 provides access to RP1	
19	RP1 provides combined resource access to IA	

Note: AP may or may not have a relationship with the RP2. AP also may be part of the RP2 or be a generic, more broadly available AP.

2.3.2 Multiple Relying Parties; Self-assertion

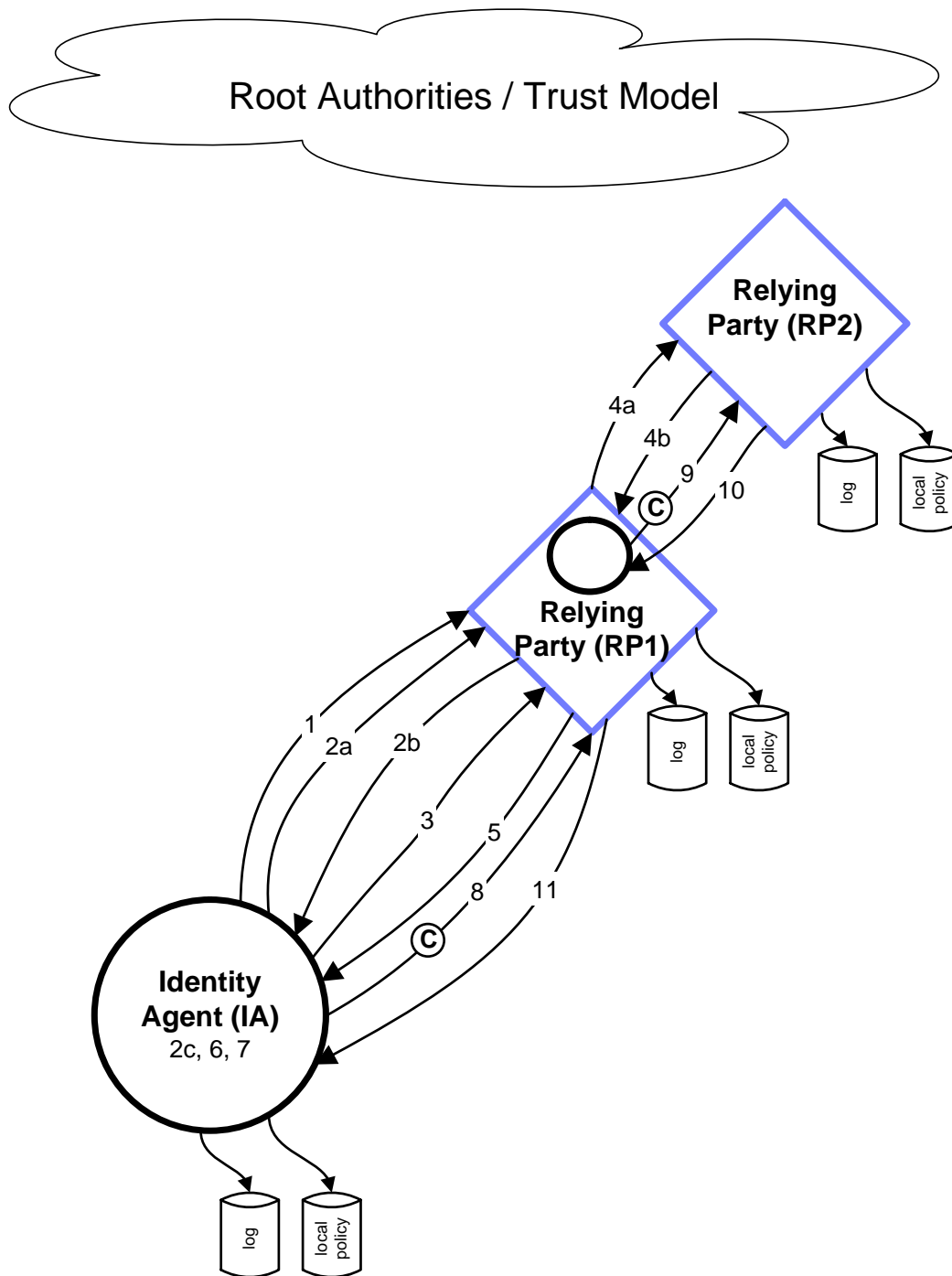


Exhibit 5 - Multiple relying parties; Self-assertion

Multiple Relying Parties – Self-assertion		
Step #	Title	Description
1	IA identifies the RP	
2a	(Optional) IA requests RP claims	IA requests claims from the RP, in order to verify properties of the RP.
2b	(Optional) RP sends claims to IA	RP provides claims about itself.
2c	(Optional) IA validates RP claims	The IA validates the claims of the RP.
3	IA requests advertised policies from the RP	The IA sends a request to the RP for the advertised policies (such as required claims, optional claims, acceptable APs and T&Cs) required to access a RP1 and RP2 resources.
4a & 4b	RP1 queries & receives advertised policies from RP2	
5	RP1 provides its own advertised policies and those of RP2	The RP responds by sending advertised policies required for access to the requested resource.
6	Principal approves claims to be sent to RP1/RP2	The IA must determine if it wants to present the requested claims to the RP.
7	IA creates its own claims	These claims permit RP1 to contact RP2
8	IA passes own claims and others required by RP1 & RP2 to RP1	The IA provides the claims to RP1. Claim could be the data itself or a reference to the data. Claims are those required by RP1 and RP2.
9	RP1 passes claim to RP2	
10	RP2 provides access to RP1	
11	RP1 provides combined resource access to IA	

2.4 Late Claim Validation

Scenario: A principal initiates a request with the RP. The RP requires an extended processing time, in order to complete the request. The RP must be able to validate the claim at multiple points in time. It is assumed the principal has utilized the core architecture (2.1) in order to initially pass the claim(s) to the RP. A single AP and RP are required to process the request.

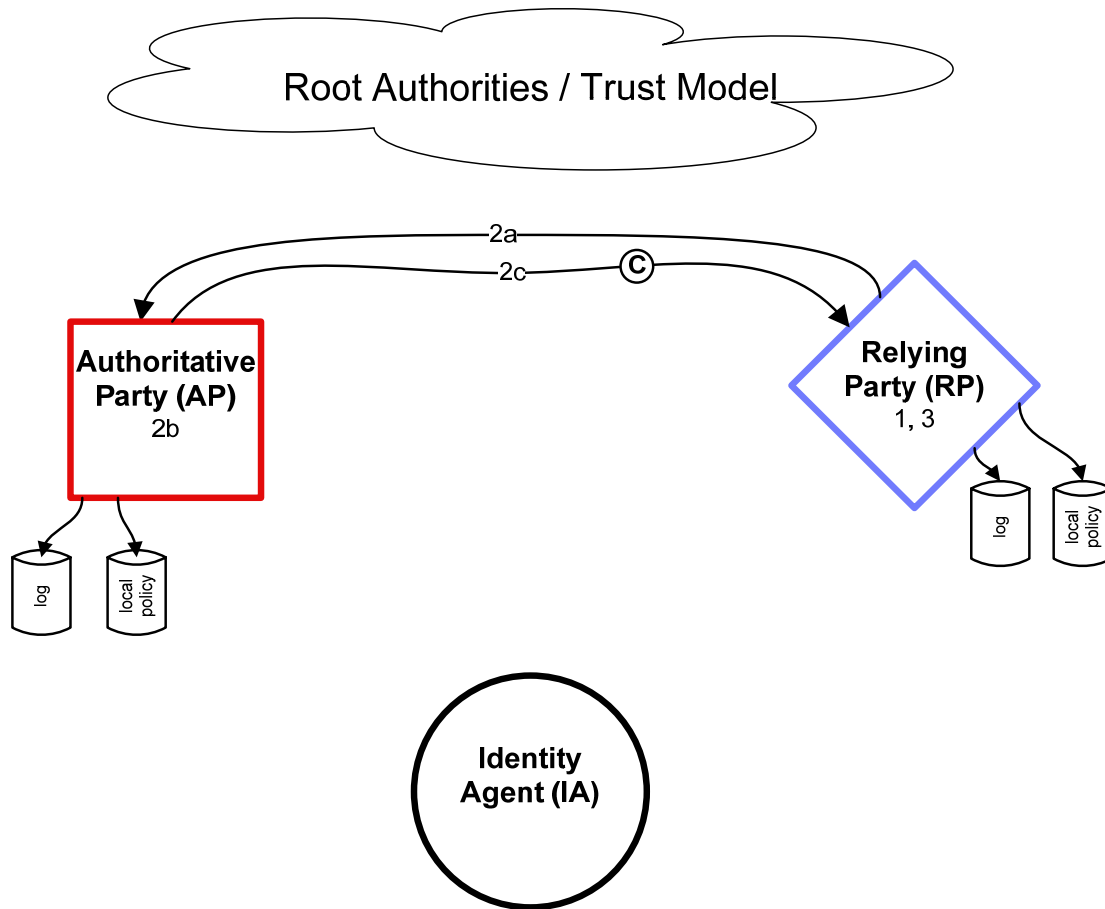


Exhibit 6 - Late Claim Validation

Late Claim Validation		
Step #	Title	Description
1	RP identifies need to validate existing claims	In order to complete a request, the RP must validate existing claims.
2a	RP requests claims validation from AP	AP requests claims from the RP, in order to verify the identity of the RP.
2b	AP validates RP request	AP evaluates the RP's request, to determine if they are valid and meet local policy requirements.
2c	AP provides claims validation to the RP	AP passes the requested claims validation to the RP.
3	RP processes results	The RP evaluates the validation results, and completes the initial request.

Late claim validation could also be handled by assigning a “time to live” (TTL) to each claim. The RP would use the TTL to determine if a claim is valid after a given period of time. If the claim has expired, the principal can be notified to provide new claims.

2.5 Delegation

Scenario: A principal wishes to delegate a portion of their role(s) to another principal.

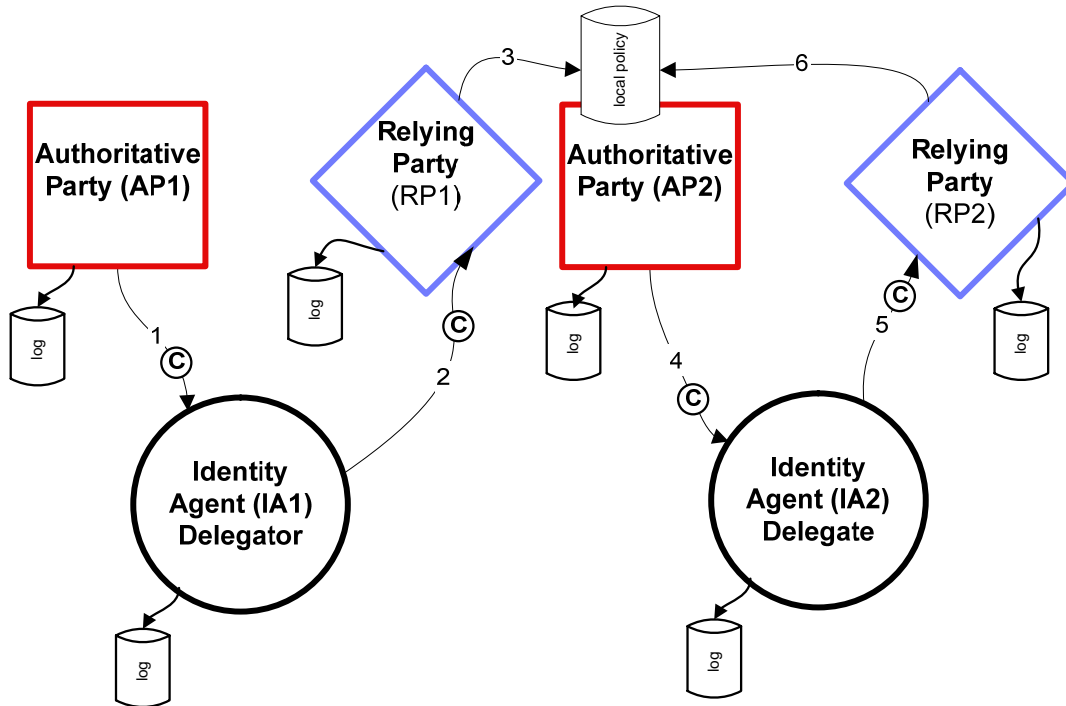


Exhibit 7 - Delegation

Delegation		
Step #	Title	Description
1	Establishing claims of the delegator	IA1 (delegator) is passed claims used to establish their role and permission to delegate.
2	Principal delegates activities	IA1 passes claims to RP1. RP1 then authorizes IA1 to delegate.
3	Delegation role(s) is stored in local policy	Permissions, valid time period, and other appropriate delegation details are stored in the local policy as part of AP2.
4	AP2 passes delegated claims to IA2	IA2 (delegate) is passed claims, used to access RP2. The claims are specific to the function assigned to the delegate.
5	Delegate claims passed to RP2	The claims of IA2 are passed to RP2.
6	RP2 determines delegate's access	RP2 validates the claims, and then references the local policy to determine the level of access to be granted to the IA2. The roles include those delegated.

Note that this is a specialized case of any other kind of administration of an AP. Also, RP1 and AP2 would usually be part of the same system and closely bound. In addition, AP1 will often be the same entity as AP2.

2.6 Offline Requests

Scenario: User of the system must be able to access local resources, even when conditions do not allow access to the Internet.

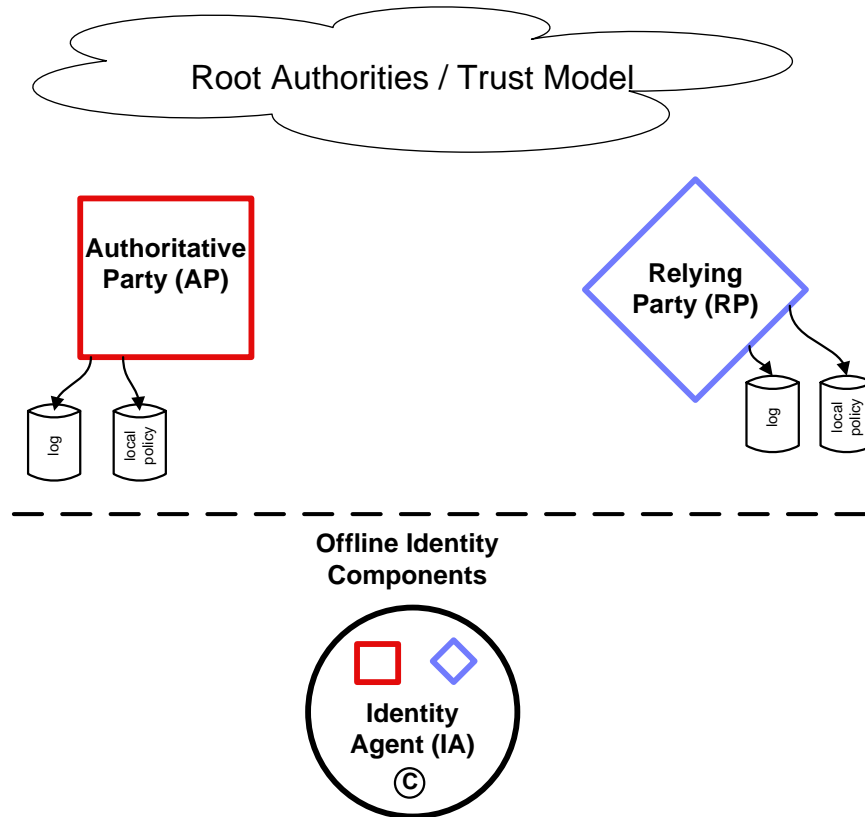


Exhibit 8 - Offline Requests

In order to handle the offline scenario, a self contained subset of the architecture must be enabled to allow access to local resources. While under offline conditions, the local architecture will function as the IA, AP, and RP.

3 Lexicon

Please see the lexicon included in "British Columbia Broader Public Sector, Identity Management Architecture Project, Requirements Document."

4 Standards and Architectural Recommendations

This section lists protocol standards appropriate to each step of the interaction diagrams above. In the final column, related standards and architectural recommendations are provided.

4.1 Core Architecture Interactions

Core Architecture Interactions			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
1	IA identifies the RP	HTTPS GET	Server authN with EA X.509 certificate
2a	(Optional) IA requests RP claims		Part of step 1 , or Part of step 3a
2b	(Optional) RP sends claims to IA		Embedded in cert or as OIDs, or Returned in metadata
2c	(Optional) IA validates RP claims		Cert validation (dereference), or Lookup in “Authorized Services” directory
3a	IA requests advertised policies from the RP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy
3b	RP provides advertised policies	WS-Transfer GetResponse	
4	IA determines where to get claims required by the RP		Issuer specified in RP policy, or Identity selector matches claim types in RP policy & AP metadata
5	IA identifies AP	HTTPS GET	Server authN with EA X.509 certificate
6a	(Optional) IA requests AP claims		See 2a
6b	(Optional) AP sends claims to IA		See 2b
6c	(Optional) IA validates AP claims		See 2c
7a	IA requests advertised policies from the AP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy
7b	AP provides advertised policies	WS-Transfer GetResponse	
8	IA obtains the claims required by AP policy for issuing tokens	Local operation or WS-Trust RST/RSTR	Get credentials from cache or user, or Get token & claims from another AP
9	Principal approves claims to be sent to AP		Credentials UI (eg password, PIN), or Identity Selector claims preview

Core Architecture Interactions			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
10	IA provides claims AP needs & requests token/claims RP needs	WS-Trust or WS-Federation	RequestSecurityToken, or wsignon1.0 request
11	AP evaluates claims from IA		Validate user's credentials (eg X.509 cert, Kerberos ticket, Password, OTP), or Validate token & claims (from Self-issued InfoCard or another AP)
12	AP provides claims RP needs in token issued to IA	WS-Trust or WS-Federation	RequestSecurityTokenResponse, or wsignon1.0 response (SAML 1.1, SAML 2.0 or other token)
13	Principal approves claims to be sent to RP		Identity Selector claims preview
14	IA provides claims to RP	WS-Security, WS-Federation or HTTPS POST	Use token(s) (or reference) from step 12
15a	(Optional) RP requests claims from AP	WS-Transfer GET or HTTPS GET	Reference token from step 14 , or WS-Federation wresultptr from step 14
15b	(Optional) AP validates RP request	WS-Security or HTTPS	SOAP message security, or Client authN with X.509 certificate
15c	(Optional) AP provides claims to the RP	WS-Transfer GetResponse or HTTPS response	See 15a
16	RP determines access		RP verifies digital signature of token RP verifies AP trusted for claim value namespaces Optional WS-Federation SOAP fault to request application specific claims
17	Access granted		

4.2 IA Possesses Claims Prior to Resource Request

IA Possesses Claims Prior to Resource Request			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
1	IA identifies the RP	HTTPS GET	Server authN with EA X.509 certificate
2a	(Optional) IA requests RP claims		Part of step 1 , or Part of step 3a
2b	(Optional) RP sends claims to IA		Embedded in cert or as OIDs, or Returned in metadata
2c	(Optional) IA validates RP claims		Cert validation (dereference), or Lookup in "Authorized Services" directory
3a	IA requests advertised policies from the RP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element),

IA Possesses Claims Prior to Resource Request			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
			WSDL, WS-Policy, WS-SecurityPolicy
3b	RP provides advertised policies	WS-Transfer GetResponse	
4	Principal approves claims to be sent to RP		Identity Selector claims preview
5	IA provides claims to RP	WS-Security, WS-Federation or HTTPS POST	Use token(s) (or reference) previously obtained for RP
6a	(Optional) RP requests claims from AP	WS-Transfer GET or HTTPS GET	Reference token from step 5, or WS-Federation wresultptr from step 5
6b	(Optional) AP validates RP request	WS-Security or HTTPS	SOAP message security, or Client authN with X.509 certificate
6c	(Optional) AP provides claims to the RP	WS-Transfer GetResponse or HTTPS response	See 6a
7	RP determines access		RP verifies digital signature of token RP verifies AP trusted for claim value namespaces Optional WS-Federation SOAP fault to request application specific claims
8	Access granted		

4.3 Multiple Relying Parties

Multiple Relying Parties			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
1	IA identifies the RP	HTTPS GET	Server authN with EA X.509 certificate
2a	(Optional) IA requests RP claims		Part of step 1, or Part of step 3
2b	(Optional) RP sends claims to IA		Embedded in cert or as OIDs, or Returned in metadata
2c	(Optional) IA validates RP claims		Cert validation (dereference), or Lookup in “Authorized Services” directory
3	IA requests advertised policies from the RP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy
4a & 4b	RP1 queries & receives advertised policies from RP2	WS-Transfer Get WS-Transfer GetResponse	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy

Multiple Relying Parties			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
5	RP1 provides its own advertised policies and those of RP2	WS-Transfer GetResponse	
6	IA determines where to get claims required by the RP		Issuer specified in RP policy, or Identity Selector matches claim types in RP policy & AP metadata
7	IA identifies AP	HTTPS GET	Server authN with EA X.509 certificate
8a	(Optional) IA requests AP claims		See 2a
8b	(Optional) AP sends claims to IA		See 2b
8c	(Optional) IA validates AP claims		See 2c
9a	IA requests advertised policies from the AP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy
9b	AP provides advertised policies	WS-Transfer GetResponse	
10	IA obtains the claims required by AP policy for issuing claims	Local operation or WS-Trust RST/RSTR	Get credentials from cache or user, or Get token & claims from another AP
11	Principal approves claims to be sent to AP		Credentials UI (eg password, PIN), or InfoCard selection & claims preview
12	IA provides claims AP needs & requests token/claims RP needs	WS-Trust RST or WS-Federation	RequestSecurityToken, or wsignon1.0 request
13	AP evaluates claims from IA		Validate user's credentials (eg X.509 cert, Kerberos ticket, Password, OTP), or Validate token & claims (from Self-issued InfoCard or another AP)
14	AP provides claims RP needs in claim issued to IA	WS-Trust or WS-Federation	RequestSecurityTokenResponse, or Wsignon1.0 response (eg SAML 1.1, SAML 2.0 or other token)
15	Principal approves claims to be sent to RP1/RP2		Identity Selector claims preview
16	IA provides claims to RP1	WS-Security, WS-Federation or HTTPS POST	Use token(s) from step 14
17	RP1 passes claim to RP2	WS-Security or HTTPS POST	

Multiple Relying Parties			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
18	RP2 provides access to RP1		
19	RP1 provides combined resource access to IA		RP verifies digital signature of token RP verifies AP trusted for claim value namespaces Optional WS-Federation SOAP fault to request application specific claims

4.4 Multiple Relying Parties; Self-assertion

Multiple Relying Parties; Self-assertion			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
1	IA identifies the RP	HTTPS GET	Server authN with EA X.509 certificate
2a	(Optional) IA requests RP claims		Part of step 1, or Part of step 3
2b	(Optional) RP sends claims to IA		Embedded in cert or as OIDs Returned in metadata
2c	(Optional) IA validates RP claims		Cert validation (dereference) Lookup in “Authorized Services” directory
3	IA requests advertised policies from the RP	WS-Transfer Get	WS-Addressing (EndPointReference), WS-MEX (Metadata element), WSDL, WS-Policy, WS-SecurityPolicy
4a & 4b	RP1 queries & receives advertised policies from RP2	WS-Transfer Get & WS-Transfer GetResponse	
5	RP1 provides its own advertised policies and those of RP2	WS-Transfer GetResponse	
6	Principal approves claims to be sent to RP1/RP2		Identity Selector claims preview
7	IA creates its own claims		SAML 1.1, SAML 2.0 or other token
8	IA passes own claims and others required by RP1 & RP2 to RP1	WS-Security or HTTPS POST	Use token(s) from step 7
9	RP1 passes claim to RP2	WS-Security or HTTPS POST	RP verifies digital signature of token RP verifies issuer(s) trusted for claim value namespaces
10	RP2 provides access to RP1		
11	RP1 provides combined resource access to IA		

4.5 Late Claim Validation

Late Claim Validation			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
1	RP identifies need to validate existing claims		
2a	RP requests claims validation from AP	WS-Transfer GET or HTTPS GET	Reference token supplied by IA in earlier interaction, or WS-Federation wresultptr supplied by IA
2b	AP validates RP request	WS-Security or HTTPS	SOAP message security, or Client authN with X.509 certificate
2c	AP provides claims validation to the RP	WS-Transfer GetResponse or HTTPS response	See 2a
3	RP processes results		

4.6 Delegation

Delegation			
Step #	Title	Wire Protocol Standards	Related Standards and Architectural Recommendations
0		Similar to steps 1-9 in table 4.1	
1	Establishing claims of the delegator	WS-Trust RST/RSTR	IA1(delegator) obtains tokens from AP1 that authorize access to RP1 and specify principal1's claims that can be delegated to AP2 principals
2	Principal delegates activities	WS-Security	(1) IA1 authenticates to delegation admin app at RP1 (2) principal1 configures claims to be delegated
3	Delegation role(s) is stored in local policy		(1) Delegation admin app verifies that claims configured by principal1 are authorized by token from AP1 (2) Claims, valid time period, and other delegation details are stored in policy db of AP2.
4	AP2 passes delegated claims to IA2	WS-Trust RST/RSTR	IA2 sends WS-Trust RST for Principal2 to AP2 on behalf of principal1
5	Delegate claims passed to RP2	WS-Security	SOAP message security and optionally supporting token in body with delegated claims
6	RP2 determines delegate's access		RP2 validates claims from tokens (and optionally references the AP2 policy db) to determine level of access to be granted to Principal2