Intelligent Client
Marketing Requirements Document

Version: 1.0

Abstract:

There are a number of industry organizations producing authentication, privacy, and payment standards for the enterprise, mobile, financial, and services industries. This has resulted in overlapping and disconnected standards that have slowed the adoption of Web-based services. Liberty proposes an interoperable framework that supports these multiple standards in both online (connected) and offline (standalone) environments.

To address these issues, a new type of Liberty-enabled client is needed. This new client will be referred to as an Intelligent Client, or iClient. The iClient must be able to run on existing devices and be able to function in online and offline states. The Liberty-Enabled Client and Proxy, together with the enhancements proposed herein and the Trusted Module, are collectively known as the iClient.

Filename: liberty-iClient-mrd-v1.0.pdf

This Market Requirements Document (MRD) has been developed by the Business and Marketing Expert Group of Liberty Alliance to capture the business requirements for an identity governance framework. Liberty Alliance is making this MRD publicly available to the industry at large for review and consideration. In addition, this MRD is being provided to the appropriate technical standards development group within Liberty Alliance for consideration of new technical work to address the requirements identified herein. This publication does not constitute a commitment by Liberty Alliance, explicit or implied, to develop technical specifications in full compliance with the requirements herein, now or in the future.

Notice:

Implementation of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. **This Specification is provided "AS IS," and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementers of this Specification are advised to review the Liberty Alliance Project's website (http://www.projectliberty.org/) for information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance Management Board.

Contents

1 Introduction

There are a number of industry organizations producing authentication, privacy, and payment standards for the enterprise, mobile, financial, and services industries. This has resulted in overlapping and disconnected standards that have slowed the adoption of Web-based services. Liberty has the opportunity to propose an interoperable framework that supports these multiple standards in both online (connected) and offline (standalone) environments.

To address these issues, a new type of Liberty-enabled client is needed. This new client will be referred to as an Intelligent Client, or iClient. The iClient must be able to run on existing devices and be able to function in online and offline states.

Three groups of requirements exist for the iClient:

1. Enhancements to the LUAD specifications.
2. Defining a "Trusted Module" (TM) that can be associated with a Liberty-Enabled Client to allow consistently accessible Liberty functionality on new or existing clients.
3. Sharing an Intelligent Client to advance vertical market interoperability and reduce deployment costs.

The Liberty-Enabled Client and Proxy, together with the enhancements proposed herein, and the Trusted Module, are collectively known as the iClient.

Therefore, as a logical entity, the iClient can be, between others, a LUAD implementation, a LUAD plus enhancements, or a LUAD plus TM. IClient describes the client part of the interaction in the Liberty framework. Existing entities are encompassed in this definition.

The LUAD is defined in the Liberty "ID-WSF Identity Profiles" document, with LEC/LEP discussions in the Liberty "ID-FF Arch Overview" document. The Mobile Implementation Guidelines, referenced later in the document, can be found in the "Liberty ID-FF Guidelines" and the "Liberty ID-WSF Guidelines" documents.

1.1 The iClient Problem Statement

The impetus for an iClient definition derives from three (3) broad problem areas: authentication, connectivity variances, and local attribute access and control. Some of the joint requirements brought by this problem formulation are already met by the existing LUAD specification. However, to meet the needs completely, more comprehensive client definitions are required and will be covered in the subsequent sections.

1.1.1 Authentication

IdPs need a range of authentication options to meet existing and anticipated deployment needs. Client-side storage and processing could enhance these options and mitigate risk.

1.1.2 Connectivity Variances

Even when continual connection between a Principal and an IdP or AP cannot be maintained, the client may require their functionality. This functionality may be provided via client-side storage and processing.

Three categories of scenarios drive the requirements for this behavior:

- Offline examples – A Principal may desire or may be required to operate offline, independent of a Liberty-enabled infrastructure. For example, secure environments may limit Internet connections, including government agencies that have restrictions against exchanging data directly.
- Continuity examples – the mobility of a Principal introduces coverage dynamics and exchanges that can affect communication, for instance, technical inefficiencies of moving between connection technologies (e.g., PAN, WiFi).
- Multi-mode examples – the IdP or AP may prefer to accomplish certain functions over multiple communication paths.

1.1.3 Local Attribute Access and Control

APs and Principals may want to store some attributes on the client for improved Principal satisfaction and adoption. The following concerns drive this need for local storage and access:

- Ensuring consistency of availability, regardless of connection state.
- Offering the Principal a stronger feeling of user control. In some scenarios (e.g., Healthcare and Government), a feeling of "possessing" or "controlling" very sensitive data is essential to adoption.
- Assuring dynamic attributes can be more accurately collected. Attributes like Geolocation may also be particularly sensitive since they extend traditional Internet functionality.

Such local attribute access may also require particular confidentiality guarantees. No attempt is made in this document to require a given level of "trustworthiness" associated with such an attribute store. However, it is worth noting that various levels of trust can be established for such a well-defined local store, and it is anticipated that the specific market needs of the applications will determine what those acceptable levels are.

It is also worth noting, as an example, that the currently-defined LUAD does, in fact, answer the above problem of accessible attribute storage in certain situations, namely that in which the AP resides on the client and the local store is accessible to that AP. However, the intent of the problem description above is to also cover situations in which the AP is not on the client, or in which the client-resident AP does not have a priori access to the store (e.g., the AP may reside in a differently-secured client partition from that of the store).

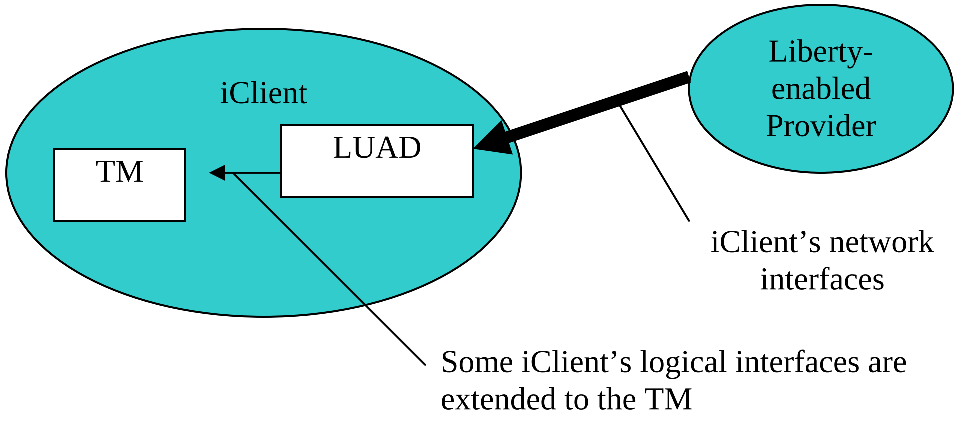
1.2 iClient Module Definition

The problem statement above outlines the need for an accessible storage and processing module available in association with a LUAD. Such a module will subsequently be referred to as a "Trusted Module" or TM.

The formulation of a TM presents certain basic requirements. Its abstraction must be such that:

- It can be made accessible to given IdPs and APs, and not to other IdPs and APs.
- A particular implementation can be derived that would allow a TM to be added as a module to existing clients.
- A particular implementation can be derived that would allow a TM to be instantiated as a portable module.
- When a TM is embodied in a portable container, a particular interconnect technology is not required.
- No particular partitioning of the implementation between hardware and software components is required. In fact, if the business requirements (including trust) of the IdP, AP, and the Principal are met satisfactorily, a TM may be entirely implemented within the client, using software-only mechanisms.
- Multiple TM’s may be associated with a particular LUAD. A one-to-one relationship is not mandated between the TM, the LUAD, and the IdP or AP.

These top-level requirements are called out for elucidation in this section, and then used as a basis for the more detailed requirements in Section 3.


iClient
TM
LUAD
Liberty-enabled Provider
iClient's network interfaces
Some iClient's logical interfaces are extended to the TM

This diagram shows the different potential interactions in the iClient. The bold arrows capture the existing protocol definitions as extended to meet the problem statements outlined above. Additionally, the existence of an accessible, optional "Trusted Module," or TM, is postulated to provide local processing and storage in a manner accessible to APs and IdPs, regardless of their location. This level of accessibility requires the exposure of a protocol to address the TM. This requirements specification sets forth the requirements on the TM, its protocol, and the extensions to the LUAD.

Without such a protocol, communication with a TM could be driven by various industry verticals that might not ensure interoperability. This could lead to complaints about the number and cost of TMs for a Principal. Overall, a Liberty TM:

- Reduces cost by sharing infrastructure, leveraging previous TM distribution, and reducing support costs.
- Allows modules to be combined into a single module for the Principal.
  - Data stores for various IdPs and APs.
  - Loadable into various terminals.

1.3 TM Interactions with LUAD

The following sections give more particular requirements on the TM and how it is integrated into an iClient. Adding an accessible store in the form of a TM must be done in a manner that does not disrupt the existing LUAD definition. The TM must also fit well within existing infrastructure implementation and usage models for similarly accessible storage and processing modules (e.g., SIM cards, Smart Cards, TPMs).

1.3.1 Trustworthy Entity Managed by the IdP and/or AP

The TM must be defined so that the IdP and/or AP can directly manage it. Through this link, the different Liberty components are able to optimize the federation process on the client. This ability to interact with a well-defined, well-bounded local module is a model that is widespread today throughout the industry (mobile communication, financial services, etc.).

1.3.2 Principal Bonded

While the TM must be able to be directly managed by the IdP and/or AP in order to mitigate risk and to fit well within the IdP's and/or AP's infrastructure, the fact that TMs, especially hardware TMs, are physically controlled by the Principal addresses concerns the Principal may have regarding Principal control policy and provable consent. Ultimately, this local control will be clearly perceived as privacy-friendly as opposed to infrastructure-hidden solutions.

Many existing TM-like modules have achieved an end-user familiarity status equal to the home/car key. The Principal perceives they "own" the module and, if leveraged, this will reduce a Principal's fears in trusting and adopting an identity framework.

When deriving the TM's requirements, then, it will be important to balance the need for direct IdP and/or AP management with the requirement that the Principal maintains both the perception and the reality of local control.

1.3.3 Movable, Modular

When we consider the direction the IT industry is moving today, it is clearly becoming increasingly important to address portability issues. Some TMs may be used as portable extensions of the infrastructure that will allow for the Principal to connect on any Liberty-enabled terminal. In support of this portability, the specifications for any movable TM must allow a compact implementation.

1.3.4 Deployable on Existing Client's Install Base

The clients deployed on the market today are not Liberty ready, and they will not be for a while. By taking advantage of the existing client-installed base, the TM allows to see massive Liberty deployments on existing technologies. As a compact entity, it is aimed at being available on any existing devices, including resource-constrained environments.

The Liberty specifications, such as the LUAD, PAOS, and WSF, permit constructs, such as the AP, to run equally well on the client or server domains of the network. However, since the LUAD specification alone exceeds the current hardware and/or software capabilities of many clients in use today, the Liberty-Enabled Proxy was introduced in Phase 1 as requirement 6.2.15.

It is a requirement, then, that the specification of the TM allow it to be added to the installed base of existing clients in such a way that the ability of these less-capable clients to interact with a Liberty-enabled infrastructure is enhanced.

1.3.5 TM Client Integration Requirements

Aside from the need for a particular TM implementation to be portable, no assumptions are made as to the physical embodiment of the TM. A TM may be physically joined to a LUAD using any number of appropriate interconnect technologies, including internal and external buses, as well as direct implementation on the client. Indeed, the TM that is presented as an accessible entity by the client may be supported by an implementation of any combination of software and hardware.

The logical implementation of any TM, regardless of hardware/software partitioning, must, with the exception of portability requirements for specific implementations, follow the hardware implementation principles in order to keep interoperability between Liberty entities.

1.4 Targeted Scenarios

Since the intelligent client is under the end-user's control, it applies to B2C, B2E, and C2C scenarios. In the following scenarios, specific use models for the TM's capabilities are described. Note that, except where explicitly called out, no assumption is made as to the physical encapsulation of the TM. For example, an RFID system can receive information directly from a portable TM or from a client fully enclosing a TM bonded to that particular client.

1.5 Business Scenarios

1.5.1 Scenario for B2C

Ticketing systems. The TM provides a single use "ticket" to the event barrier (cinema, theatre, concert, etc.) over a local link – RFID, infrared, scanned screen, etc. The barrier system can validate and retrieve attributes in real-time (film, time of show, etc.), either from a back-end or from the TM directly, without relying on network connection between the iClient device and the IdP.

Proximity payment. The TM provides a single-user "token" to the till over a local link – RFID, infrared, scanned screen, etc. The merchant system can validate this and retrieve attributes in real time (real card number, expiration date, etc.), either from the TM or from a back-end, without relying on a network connection between the iClient device and the IdP.

Separate iClient. In the same scenarios, the iClient is independent of the device used to access the service. In proximity services, the access device could be a simple swipe-card or static RFID token (badge, etc.). Strong authentication is gained by getting the Principal to confirm the access event, payment, etc., on the separate iClient. The advantage here is that the access device and iClient device can be legacy systems. The disadvantage is that real-time network access to the iClient is required.

1.5.2 Scenario for B2E

Corporate remote access. The TM provides a single-use "password" to the LUAD which then passes it on to the SP. The SP validates (possibly with IdP server to server). This can use the same basic system as for corporate access with one-time passwords, but without the Principal having to read and enter one-time passwords.

Corporate physical access. The TM provides a single-use "token" to the door system over infrared, RFID, etc. The door system acts as an SP to validate, and also retrieves employee privileges (are they allowed in this area, at this time?) as attributes. These can be retrieved from the TM (where door system is not networked) or from a back-end (where it has real-time access to the AP). There is no reliance on a network connection between the iClient device and the IdP.

Separate iClient. In the same scenarios, the iClient is independent from the device used to access the service. With remote access, the access device could be a zero-install browser in a cybercafé or even a non-IP channel (email over IVR, etc.) with advantages and disadvantages as before.

Disconnected terminal application. There are a number of applications today relying on offline permission checking, such as access to equipment or secure areas. (e.g., right-to-drive vehicles).

1.5.3 Scenario for C2C

Peer to peer payment. The TM of the sending peer provides a single use "coin" to the receiving peer, who quickly exchanges this at their bank. The receiving bank validates the coin with IdP (server to server) and retrieves the necessary attributes (e.g., coin value). The bank then credits the receiving peer's account.

Separate iClient. The sending peer sends a simple, unauthenticated payment instruction to the receiving peer and the receiving peer presents this to their bank. The sending peer is asked to confirm the instruction on the separate iClient with advantages and disadvantages as before.

1.5.4 Scenario for G2C (government to citizen)

There are an increasing number of devices issued by Government or government agencies for Identification and authentication. Examples of these are the Macao National Identity card, the United Emirate Identity card, and the Common Access Card issued by the US Department of Defense. All these devices are provisioned with numerous attributes dealing with personal information or personal permissions. Most of these attributes are field-checked with disconnected terminals (e.g., blood type information, citizenship information).

For privacy and confidentiality purposes, portable certified devices are the optimal repository for such information since they are provisioned by the issuer with the end user's consent and the end user maintains control of private attribute information by allowing release or not. These types of intelligent devices can also associate attribute release with the end user's provable consent.

As a significant example of a recent development in this field, the Government of Taiwan has recently announced a program to issue 22 million Java-card-based healthcare identification cards. These cards will store patients-critical information, precisely to avoid exposing it through high-risk central storage.

1.6 High-Level Dependencies

The iClient and TM definitions rely on and affect the definitions of LUAD, AP, IdP, and SP.

In addition, the Mobile Implementation Guidelines and Mobile Business Guideline documents may require the iClient and TM specifications.

2 Requirements

Req#	UC #	Use Category	Requirements (for Version 3 only)
1	3.4	Offline SSO	iClient can obtain and store in a efficient manner from an IdP the credentials and information necessary to authenticate at a later time to a given SP from a list of potential SPs
2	3.5	Offline SSO	iClient can obtain from an IdP authentication assertions targeted at a given SP
3	3.5	Offline SSO	iClient can proactively include authentication assertions in its service requests to an SP
4	3.6	Offline Accounting	IdP can request from the iClient a record of the SSO transaction that took place while the iClient was offline
5	3.6	Offline Accounting	iClient can securely report SSO transactions to the IdP
6	3.7	Offline SSO	TM can obtain and store in a efficient manner from an IdP the credentials and information necessary to authenticate at a later time to a given SP from a list of potential SPs without relying on the security characteristics of any network intermediate
7	3.7	Offline SSO	TM can obtain from an IdP authentication assertions targeted at a given SP without relying on the security characteristics of any network intermediate
8	3.7	Offline SSO	TM can authenticate an SP without relying on the security characteristics of any network intermediate
9	3.7	Offline SSO	IdP can request from the TM a record of the SSO transaction that took place while the TM was offline without relying on the security characteristics of any network intermediate
10	3.7	Offline SSO	TM can securely report SSO transactions to the IdP without relying on the security characteristics of any network intermediate
11	3.8	Lifecycle Support	iClient can announce and negotiate its authentication capabilities to IdP in the course of an iClient authentication
12	3.9	Convergence	IdP can provision into a pre-deployed TM (for example credentials)
13	3.1	Offline Attribute	TM can securely convey attributes to an SP without relying on the security characteristics of any network intermediate

Req#	UC #	Use Category	Requirements (for Version 3 only)
14	3.1	Offline Attribute	TM can securely express the attribute's source AP to an SP without relying on the security characteristics of any network intermediate
15	3.2	Offline Attribute	AP can securely store attributes and associated sharing policies in a TM without relying on the security characteristics of any network intermediate
16	3.2	Offline Attribute	AP can securely store credentials associated with shared attribute in a TM without relying on the security characteristics of any network intermediate
17	3.2	Offline Attribute	AP can request from the TM a record of attribute sharing transactions that tool place while the TM was offline without relying on the security characteristics of any network intermediate
18	3.3	Offline Attribute	TM can securely report attribute sharing transaction to an AP without relying on the security characteristics of any network intermediate
19	3.10	Dyn Att Privacy	iClient can request dynamic data sharing permission from an AP
20	3.10	Dyn Att Privacy	AP can establish dynamic data sharing policies at iClient
21	3.11	0 Wt Client	Ability to route access request, server to server; from SP to correct TM agent for given userid (possibly via intermediaries)
22	3.11	0 Wt Client	TM agent can grant external party (e.g., IdP) a session with iClient to confirm access request
23	3.11	0 Wt Client	Ability to route authentication assertion, server to server; for given userid, e.g., from IdP to SP (possibly via intermediaries)
24	3.12	0 Wt Client	Ability to route attribute request, server to server; from SP to correct TM agent for given userid (possibly via intermediaries)
25	3.12	0 Wt Client	TM agent can grant external party (i.e., SP) a session with TM to retrieve user attributes (possibly via server intermediaries)
26	3.13	Single Logout	iClient can securely request IdP a SLO
27	3.13	Single Logout	iClient can securely notify SP of a SLO

Req#	UC #	Use Category	Requirements (for Version 3 only)
28	3.13	Single Logout	iClient can securely report SLO transactions to the IdP
29	All	All	LUAD protocols must be extended sufficiently to support new requirements of the TM
30	3.14	Continuity	IdP can reliably determine the continuity of the TM's binding to the rest of the iClient.
31	3.14	Continuity	TM can notify the IdP that its binding with the rest of the iClient should be broken
32	3.14	Continuity	IdP can notify SPs of a frozen SSO session (or conditional SLO)
33	3.14	Continuity	IdP can perform pro-active SSO while reestablishing a frozen SSO session
34	3.14	Continuity	TM can securely request an IdP a SLO without relying on the security characteristics of any network intermediate
35	3.14	Continuity	TM can securely notify SP of a SLO without relying on the security characteristics of any network intermediate
36	3.14	Continuity	TM can securely report SLO transactions to the IdP without relying on the security characteristics of any network intermediate

3 Use Cases

3.1 TM-Enabled Offline Attribute Exchange

Enables a Principal to service an SP attribute request directly from the iClient so that an SP can retrieve them when the other source AP is not available for direct connection or due to legal reasons, for example, the US law prohibiting agencies from directly exchanging data. To limit risk exposure and follow industry-standard practices, these attributes shall be stored and its transmission secured in the TM component of the iClient.

3.1.1 Benefit

When the AP is not available, Principals can rely on identity attributes to simplify the service enrollment and/or fulfillment from the SPs.

Principals also may feel more comfortable releasing attribute data from a personal device than directly leveraging some "intangible" upstream server.

3.1.2 Dependencies With Other Use Cases

Some SPs may impose an off-line SSO (use case 3.7) pre-requisite.

Attributes must be securely stored in the TM following use case 3.2

3.1.3 Details

Title/ID	TM-enabled Offline Attribute Exchange
Pre-Conditions	1. Principal has federated IdP with SP. 2. Principal has been authenticated to both IdP and SP. 3. AP has stored attributes an associated credentials in the TM.
Constituents	Principal, TM, iClient (that is, the iClient components other than the TM), IdP, SP
Use Case	1. Principal request service to the SP. 2. SP requests attributes from the iClient. 3. TM authenticates SP and evaluates privacy policy. 4. TM secure conveys attributes to SP together with their AP-representing credentials and bounding privacy policy. 5. SP performs service fulfillment.
Post Conditions	SP performs services fulfillment as if Attributes where directly provided by an upstream AP.
Alternate	After step 3, the iClient could poll the Principal for data sharing

Courses of Action	consent. After step 5, the iClient could notify an AP of the attribute sharing operation.

3.2 AP to TM Attribute Downloading

Enables an AP or Principal to request the iClient to store AP-sources attributes so an SP can retrieve them when the other source AP is not available for connection or due to legal reasons, for example, the US law prohibiting agencies from directly exchanging data. To limit risk exposure, these attributes shall be stored and its transmission secured in the TM component of the iClient.

3.2.1 Benefit

When the AP is not available, Principals can rely on identity attributes to simplify service enrollment and/or fulfillment from SPs.

Principals may also feel more comfortable releasing attribute data from a personal device than directly leveraging some "intangible" upstream server.

3.2.2 Dependencies with Other Use Cases

TM-enabled SSO (use case 3.7)

3.2.3 Details

Title/ID	AP to TM attribute downloading
Pre-Conditions	1. Principal has federated IdP with AP.
Constituents	Principal, TM, iClient (that is, the iClient components other than the TM), AP
Use Case	1. Principal or iClient browses to AP. 2. AP authenticates Principal and TM via IdP following SSO. 3. AP requests TM to store Attributes. 4. TM authenticates AP. 5. AP securely downloads attributes together associated authentication credentials and sharing conditions into TM.
Post Conditions	TM can service Attribute requests from SP as if it was the AP
Alternate Courses of Action	After step 3, iClient may request the Principal for consent. In step 5, AP could request the TM for post-attribute sharing notification

3.3 TM to AP Attribute Sharing Notification

Enables a TM to notify an AP about off-line attribute sharing with an SP.

3.3.1 Benefit

Enables APs to perform service accounting and limit risk exposure.

3.3.2 Dependencies with Other Use Cases

The notification request must be delivered following use case 3.2.

AP may request TM authentication following use case 3.7.

Note that notifications are considered an integral part of attribute management and therefore do not require an independent end-user consent from the one obtained in use case 3.2.

3.3.3 Details

Title/ID	TM to AP Attribute Sharing Notification
Pre-Conditions	1. Principal has federated IdP with AP 2. AP has stored attributes in TM 3. AP has stored attribute sharing notification request in the TM 4. TM has engaged in a sharing transaction with an SP
Constituents	Principal, TM, iClient (that is, the iClient components other than the TM), AP
Use Case	1. Principal or iClient browses to AP. 2. AP authenticates Principal and TM via IdP following SSO. 3. TM authenticates AP. 4. TM delivers to AP attribute sharing notifications.
Post Conditions	APs can account for all attribute sharing interactions.
Alternate Courses of Action	

3.4 Off-line SSO

Enables Single-Sign-On functionality when a connection to the IdP is not available, such as in a security zone where communication is controlled or when the Principal is a guest user on the network

Note that when the iClient interacts initially with the IdP, neither itself, the IdP, nor the Principal may know exactly which SP will be targeted subsequently.

3.4.1 Benefit

SPs can rely on IdP authentication services when the Principal is not connected to the IdP.

3.4.2 Dependencies With Other Use Cases

May require post-execution of use case 3.6.

3.4.3 Details

Title/ID	Offline SSO
Pre-Conditions	Principal has federated IdP and several SPs
Constituents	iClient, IdP, SPs, Principal
Use Case	1. Principal requests pre-authorized authentication assertions from the IdP. 2. IdP and iClient establish Principal authentication as necessary. 3. IdP generates credentials and Principal-name mapping information and stores them in the iClient. 4. Principal enters a controlled zone. 5. Principal request a services from one of the zone's SPs. 6. SP requests authentication from the iClient. 7. The iClient cannot reasonably communicate with the IdP because of zone restrictions. 8. iClient processes request and creates authentication assertion from the materials received in step 3. 9. iClient sends authentication assertion to SP. 10. SP processes authentication assertion and verifies genuine IdP's involvement. 11. SP performs service fulfillment.
Post Conditions	Principal authenticated at SP.

Alternate Courses of Action	At step 3. The IdP may indicate to the iClient its desire to be notified upon a successful SSO operation.

3.5 Proactive SSO

Enables efficient Single-Sign-On functionality when the iClient has behavioral knowledge that an SP will require authentication in the near future.

In a standard SSO flow, the Principal requests a service at the SP that triggers it to route an authentication request through the client up to the IdP, which processes the request and routes the response back to the SP in a similar fashion. These message interchanges may introduce a non-negligible delay between the service request and its fulfillment.

Instead, the iClient will pre-issue the authentication request to the IdP and retain the resulting authentication assertion (or some other suitable artifact). Upon the Principal's service request at the SP, the iClient will proactively include the authentication assertion, resulting in an immediate service fulfillment. These two steps could also be performed over networks differing in significant characteristics, such as bandwidth or spatial availability, attaining further operational cost and efficiency gains over the normal SSO flows.

3.5.1 Benefit

For the Principal and the SP: Optimized authenticated service fulfillment.

3.5.2 Dependencies with Other Use Cases

May share technological mechanisms with use case 3.1

May require post-execution of use case 3.6

3.5.3 Details

Title/ID	Proactive SSO
Pre-Conditions	Principal has federated IdP and SP
Constituents	iClient, IdP, SP, Principal
Use Case	1. iClient requests authentication assertions from the IdP targeted at a given SP. 2. IdP and the iClient establish iClient and Principal authentication as necessary. 3. IdP generates authentication assertion(s) and stores them in the iClient. 4. Principal performs an action that triggers a service request at

	SP. 5. iClient generates service request including cached authentication assertion. 6. SP processes service request and Authentication Assertion. 7. SP performs service fulfillment.
Post Conditions	Principal authenticated at SP
Alternate Courses of Action	At step 3. The IdP may indicate to the iClient its desire to be notified upon a successful SSO operation.

3.6 Post-SSO Reporting

Use cases 3.1 and 3.5 enable the iClient to undertake SSO without requiring a synchronous IdP involvement. For security and accounting reasons, it may be necessary for an IdP to obtain a notification from the iClient that the SSO has been performed at a given SP.

It should be possible to perform the notification at a completely different time from the SSO operation itself.

3.6.1 Benefit

For the IdP: higher control of the iClient's behavior and the ability to consolidate SSO notifications with partner SPs.

3.6.2 Dependencies with Other Use Cases

Must report results from use cases 3.1 and 3.5.

3.6.3 Details

Title/ID	Post-SSO Reporting
Pre-Conditions	1. Principal has federated IdP and SP. 2. Principal has authenticated at SP thanks to the IdP's vouching.
Constituents	IdP, iClient
Use Case	1. The iClient "discovers" that the IdP is within reach. 2. The iClient inspects the SSO notification request list. 3. iClient forwards SPs identifier together with the Authentication Assertion leveraged to login at that SP or a suitable representation of it. 4. The IdP replies with a reception confirmation. 5. The iClient removes the notification request.

Post Conditions	The IdP can account for all SSO transactions as if they were directly performed by itself.
Alternate Courses of Action	None

3.7 TM-Enabled SSO

An iClient may be comprised of several components, some of them considerably more trustworthy to the IdPs, SPs, and Principals than others. Furthermore, it is common in some wide-spread transactional settings that the only trustworthy client-side component is a highly-compact secured computing module such as a Smart Card or a SIM, one where the Principal has physical possession of the component.

For these use cases that carry a significant iClient trust-requirement such as 3.1, 3.5, and 3.6, it is important to IdPs, Principals, and SPs that SSO operations can be enabled and controlled unilaterally by a TM.

3.7.1 Benefit

Enables efficient high-value service fulfillment by leveraging the TM's trustworthy characteristics.

3.7.2 Dependencies with Other Use Cases

Implements use cases 3.1, 3.5, and 3.6

3.7.3 Details

Title/ID	TM-enabled SSO
Pre-Conditions	1. Principal has federated IdP and SP. 2. iClient contains TM.
Constituents	TM, iClient (that is, the iClient components other than the TM), IdP, SP, Principal
Use Case	Ex: TM-Enabled Proactive SSO 1. iClient requests authentication assertions from the IdP targeted at a given SP. 2. IdP and the TM establish TM and Principal authentication as necessary. 3. IdP generates authentication assertion(s) and stores them in the TM. 4. Principal performs an action that triggers a service request at SP.

	5. TM validates SP as a suitable target for the authentication assertions and conveys them to the iClient. 6. iClient generates service request including the authentication assertion from the TM. 7. SP processes service request together with Authentication Assertion. 8. SP performs service fulfillment.
Post Conditions	Successful SSO operation is tightly controlled by the TM involvement.
Alternate Courses of Action	Ex: TM-Enabled Proactive SSO At step 3. The IdP may indicate to the TM its desire to be notified upon a successful SSO operation.

3.8 iClient Announcement of Authentication Capabilities

The Principals of an IdP have various version levels of clients which are allowed to access the federated services from various channels. Therefore, the IdP must support several Authentication Context Classes (ACCs) to authenticate the Principal. At login time, the IdP would want to know the client characteristics to determine which authentication options are available. See the Phase 1 MRD, Requirement 6.2.19, Use Case 5.3.1 for more data on ACCs.

3.8.1 Benefit

Ability for the IdP to better support a heterogeneous iClient deployment.

Ability for the IdP to insure that it is always leveraging the most secure authentication mechanism available resulting in a higher service fulfillment to the Principal.

3.8.2 Dependencies with Other Use Cases

None.

3.8.3 Details

Title/ID	iClient Announcement of Authentication Capabilities
Pre-Conditions	
Constituents	iClient, IdP
Use Case	1. iClient request service at SP (because of Principal…). 2. SP requires Principal authentication. 3. SP generates authentication request for IdP and hands it to iClient. 4. iClient forwards authentication request to IdP.

	5. iClient announces authentication capabilities to IdP. 6. IdP request Principal authentication to iClient using preferred method. 7. iClient authenticates Principal to IdP. 8. IdP generates authentication assertion for SP and hands it to iClient. 9. iClient forwards authentication assertion to SP. 10. SP performs service fulfillment.
Post Conditions	Principal authenticated at IdP with most secure method available. Authentication Assertions expresses most secure authentication context available.
Alternate Courses of Action	Other SSO flows may be impacted accordingly.

3.9 Multi-Credential TMs

Share TMs between non-federating IdPs to reduce deployment costs and improve authentication and consent for online services.

Example 1: An IdP in the payment industry, such as an issuing bank, adds their credential to an available Liberty-Enabled Module (TM), such as an operator's SIM in a mobile phone, to reduce fraud in online purchases and offer guaranteed payment to merchants.

Example 2: The Principal wants some of its authentication modules to be combined with another module like its physical access device or phone. The iClient supports the addition of multiple credentials so that the module is able to undertake the union of the functions performed by combined authentication modules.

3.9.1 Benefit

Enhanced Principal authentication for all services while leveraging an available TM to reduce deployment and support costs.

Combining credentials into a Liberty module will also benefit the Principal because he will be able to carry fewer devices.

3.9.2 Details

Title/ID	Multi-Credential TMs
Pre-Conditions	TMa deployed TM.
Constituents	TM, iClient (that is, the iClient components other than the TM), TM Agent (TMa) (typically an IdP), IdP, Principal

Use Case	1. Principal navigates to IdP. 2. IdP authenticates Principal leveraging some ad-hoc mechanism. 3. IdP conveys "authenticators" to TMa. 4. TMa downloads authenticators into TM.
Post Conditions	IdP can authenticate Principal through TM as if the TM was deployed by itself.
Alternate Courses of Action	Flow above is just a high-level example. Bindings between steps 1, 2, and 3 are not defined.

3.10 Filter for Dynamic Data Collection

Principals experience different environments and constraints that impact their willingness to allow others to access their information (voluntarily or not, i.e., Kids, Gamblers, Job interviews, Adult entertainment, etc.).

By providing a filtering capability through the iClient, its administrator and the Principal will be able to modify attribute access, broadcast, and attribute collection. These situations can be very brief and heterogeneous and are dependent on the Principal's local variables which are mostly inaccessible to the IdP and other network entities.

Note that Emergency services (e.g., e911) and public safety officials may require a method to "override" iClient settings in emergencies or in response to court orders.

3.10.1 Benefit

Enhanced Principal-related dynamic data sharing between providers by enabling the Principal to sustain continuous control on when and how this data is shared.

3.10.2 Details

Title/ID	Filter for Dynamic Attribute Data Collection
Pre-Conditions	1. Principal federated IdP with SP and IdP with AP. 2. Principal authenticated at SP via SSO with IdP. 3. Principal authenticated at AP via SSO with IdP.
Constituents	Principal, iClient, AP, SP, IdP
Use Case	1. Principal navigates at SP. 2. SP request dynamic attribute collection from iClient. 3. iClient request AP for dynamic attribute provisioning at SP. 4. AP authorizes iClient to perform dynamic attribute provision to SP. 5. iClient notifies the Principal of the new condition. 6. Principal sets filter for dynamic data collection.

	7. iClient starts fulfilling SPs request accordingly to the filter settings.
Post Conditions	Principal is in direct control of the dynamic Attribute sharing process.
Alternate Courses of Action	After step 3, AP may request Principal SSO.

3.11 iClient SSO with Zero Install Service Channel

In this use case, the iClient is a portable device with its own communication channel. It is used alongside any browser or any piece of equipment, in any primary access channel, to provide single-sign-on based on any user identity to any Service Provider.

The Principal carries the portable device, but nothing needs to be installed into the browser or terminal used to access the service.

The TM agent is responsible for establishing sessions with the iClient over the additional communications channel. To avoid each SP having to have a direct relationship with each TM agent, or each IdP, the role of an "Acquiring Entity" is introduced; it effectively acts as a proxy for the SP in server-to-server communications with the IdP or TMa.

3.11.1 Benefit

Gives the benefit of portability of TM (TM inside iClient), but without the requirement to customize the service access channel (either at the browser or the service front end). Almost all traffic is server-to-server, so there is no need to do https re-directs, etc., in the browser. Gives a single solution that works across access channels.

3.11.2 Dependencies with Other Use Cases

Will generally depend on 3.9

3.11.3 Details

Title/ID	iClient SSO with Zero Install Service Channel
Pre-Conditions	1. Principal has portable iClient containing TM. 2. Principal’s IdP has provisioned the TM. 3. SP accepts sign-ins based on userid from that IdP.
Constituents	TM, iClient (that is, the iClient components other than the TM), IdP, SP, Principal, Acquiring Entity
Use Case	1. Principal browses to SP and provides a userid, which indicates that this use-case will be used. (Could be done manually or by browser cookie, and userid could be a temporary handle etc.).

	2. SP sends an access request for this userid to its Acquiring Entity. 3. Acquiring Entity (e.g., with help of IdP) uses userid to route the access request to the correct TM agent. 4. TM agent sets up session to iClient (e.g., end to end between IdP and iClient) to send access request. 5. iClient asks Principal to confirm access request. 6. Principal confirms access request to iClient. 7. iClient uses TM to generate credential and passes it back over the session established by TM agent. 8. IdP receives and verifies the credential, and generates an authentication assertion. 9. IdP returns authentication assertion to SP via Acquiring Entity. 10. SP processes and validates authentication assertion. 11. SP performs service fulfillment.
Post Conditions	Successful SSO operation is controlled by the iClient involvement.
Alternate Courses of Action	In Step 4, the session could be end to end to the TM itself. Step 5 could generally be more complex on first access (use enters PIN) than on subsequent accesses in a session (user just clicks OK). At step 8, there are several branching cases depending on who verifies the credential; could be IdP, but could also be TM agent, Acquiring Entity, or SP itself. After step 11, a fresh handle generated by IdP could be written to cookie (keeps the userid dynamic).

3.12 iClient Attribute Exchange with Zero Install Service Channel

As with 3.11, the iClient is a portable device with its own communication channel. It is used alongside any browser or any piece of equipment, in any primary access channel, to provide attributes associated with any identity to any Service Provider.

The Principal carries the portable device, but nothing needs to be installed into the browser or terminal used to access the service.

The TM agent is responsible for establishing sessions with the iClient over the additional communications channel. To avoid each SP having to have a direct relationship with each TM agent, or each IdP, the role of an "Acquiring Entity" is introduced; it effectively acts as a proxy for the SP in server-to-server communications with the IdP or TMa.

3.12.1 Benefit

Gives the benefit of portability of TM (TM inside iClient), but without the requirement to customize the service access channel (either at the browser or the service front end). Almost all traffic is server-to-server, so no need to do https re-directs, etc., in the browser. Gives a single solution that works across access channels.

3.12.2 Dependencies with Other Use Cases

Will generally depend on 3.9 and 3.11

3.12.3 Details

Title/ID	IClient Attribute Exchange with Zero Install Service Channel
Pre-Conditions	1. Principal has portable iClient containing TM. 2. Principal’s IdP and AP have provisioned the TM. 3. Principal has signed in to SP with userid from IdP.
Constituents	TM, iClient (that is, the iClient components other than the TM), IdP, SP, Principal, Acquiring Entity
Use Case	1. SP requests attributes for logged in userid via Acquiring Entity. 2. Acquiring Entity (e.g., with help of IdP) uses userid to route the attribute request to the correct TM agent. 3. TM agent sets up session to iClient (e.g., end to end between TM and SP) to request user attributes. 4. iClient asks Principal to confirm release of attributes. 5. Principal confirms request to iClient 6. iClient uses TM to retrieve desired attributes and pass back over the session established by TM agent.

	7. SP receives and processes the attributes. 8. SP performs service fulfillment.
Post Conditions	Successful attribute exchange is controlled by iClient involvement.
Alternate Courses of Action	After step 3, the TM may determine that the attributes are not available locally. But iClient is used (3.5) to get user permission to release attributes, and this permission is communicated back to IdP to be passed to an AP. At step 7, there may be intermediaries involved (e.g., Acquiring Entity) but these do not necessarily have visibility of the attributes.

3.13 iClient-Driven Single Logout

The iClient, thanks to its proximity to the Principal, can more easily determine which environmental conditions signal the Principal's willingness to terminate SSO-enabled sessions.

Also, from the Principal's perspective, a single iClient can be configured more easily than a set of IdPs with the policy that determines when a session should be terminated.

Finally, in off-line SSO settings, the iClient remains the only convenient repository to host and drive the global session termination policy.

3.13.1 Benefit

Diminishing the risk associated with unnecessarily keeping sessions alive. Ability to address the "forgot to logout" problem more effectively.

3.13.2 Dependencies with Other Use Cases

Dependencies with use cases 3.1, 3.5, and 3.6.

3.13.3 Details

Title/ID	iClient-driven Single Logout
Pre-Conditions	1. Principal has federated IdP and SP. 2. Principal has authenticated to IdP. 3. Principal has SSO to SP.
Constituents	iClient, IdP, SP, Principal
Use Case	1. iClient determines the Principal's intention to drop the session. 2. iClient notifies IdP. 3. IdP performs SLO.
Post Conditions	Principal has SLO at all SPs involved in the current session.

Alternate Courses of Action	If Session has been initiated through off-line SSO: 1. iClient notifies each SP to which it has provided an authentication assertion of the SLO event. 2. If requested to do so, the iClient later notifies the IdP of the SLO event.

3.14 TM-Driven Dynamic Session Management and Transfer

The TM is the iClient's component that is more tightly coupled with the Principal. In some settings, the Principal will carry the TM with him while he physically roams through iClient-enabled terminals. One such example would be the Principal browsing the Internet with a PC while the authenticated session is initiated and managed by a TM-bearing handset connected through a Bluetooth interface.

By binding the SSO sessions with the TM, rather than the iClient as a whole, this use case would allow to:

- Automatically perform SLO whenever the TM is disconnected or out or reach from the rest of the iClient.
- Transfer an existing session to another iClient terminal whenever the Principal connects and/or acknowledges the new connection with the TM.

3.14.1 Benefit

Increased ability to control SSO sessions and enabling of powerful dynamic scenarios.

3.14.2 Dependencies with Other Use Cases

Dependencies with use cases 3.1, 3.5, 3.7, and 3.13.

3.14.3 Details

Title/ID	TM-driven dynamic session management and transfer
Pre-Conditions	1. Principal has federated IdP and SP. 2. TM has authenticated Principal has authenticated to IdP. 3. Principal has SSO to SP.
Constituents	TM, iClient(that is, the iClient components other than the TM) , IdP, SP, and Principal
Use Case	1. IdP monitors if the TM is in reach with the iClient. 2. TM moves out of reach. 3. IdP performs "conditional SLO" with all SPs. 4. TM moves within reach of another iClient terminal. 5. Principal confirms to TM the new iClient binding.

	6. TM notifies IdP. 7. IdP performs a proactive-SSO with SPs with a statement notifying these as a continuation previous session.
Post Conditions	Principal has moved iClient terminal while keeping his previous session state.
Alternate Courses of Action	- If after step 4, the TM never enters within reach of another iClient. 1. IdP performs standard SSO. - If at step 2 ,the TM is able to determine that connection is going to lost before it actually happens. 1. TM notifies to IdP initiate conditional SLO. - If off-line SSO is used, the TM has to drive the SLO process instead of the IdP (probably it is too risky to do conditional SLO in this setting). 1. TM determines that the session should be terminated. 2. TM notifies each SP to which it has provided an authentication assertion of the SLO event. 3. If requested to do so, the TM later notifies the IdP of the SLO event.

4 New Glossary Terms

4.1 Acquiring Entity

A business organization or a linked chain of organizations (typically involving IdPs) that connects a Service Provider to a TM agent through contractual agreements. Its main function is to enable an SP with a Zero Install Service Channel to set up an independent communication channel to an iClient.

4.2 iClient

The union of the LUAD, the TM, and other network client-side components required to implement the resulting protocols of this proposal.

4.3 Liberty Trusted Module (TM)

A new Liberty infrastructure component that accelerates the adoption of Liberty through the use of deployed client infrastructure. TMs are compact, trustworthy computing modules, which include smart cards for the mobile (e.g., SIM), financial (e.g., EMV), and ID markets, as well as security chips available on some PC motherboards and some mobile phones and PDAs.

4.4 Liberty Trusted Module Agent (TMa)

An entity that deploys and administrates Liberty Trusted Modules, typically an IdP.