経済産業省委託調査

ＳＡＭＬ利用検討報告書

電子商取引推進協議会
財団法人日本情報処理開発協会 電子商取引推進センター

←（背表紙）

（表紙）

経済産業省委託調査

平成１５年度ＥＣ技術基盤の相互運用性に関する調査研究事業
（取引相手先の属性認証技術等の調査）

ＳＡＭＬ利用検討報告書

平成１６年３月

電子商取引推進協議会
財団法人日本情報処理開発協会
電子商取引推進センター

（表紙裏）

この報告書は、平成１５年度受託事業として（財）日本情報処理開発協会電子商取引推進センターが経済産業省から委託を受けて、電子商取引推進協議会（ ECOM）の協力を得て実施した「平成１５年度ＥＣ技術基盤の相互運用性に関する調査研究事業（取引相手先の属性認証技術等の調査）」の成果を取りまとめたものです。

序文

認証・公証ＳＷＧ１では、属性認証を実現化するモデルシステムについて検討してきた。
昨年度は、属性認証書を利用した適用モデルについて、サービス提供者、運用者等の立場から検討し、「属性認証の適用ガイドライン」および「属性認証の利用モデル」を作成した。
その検討過程で、属性認証を実現する新たな技法として、XML 技術の一つである SAML(Security Assertion Markup Language)が注目されつつあることが判明し、本年度の検討の方向として、システム検討の対象を SAML に移し、これに関する技術動向を調査し、電子商取引におけるユースケースを想定し、本報告書としてまとめた。

なお本検討は、(財)日本情報処理開発協会電子商取引推進センターが事務局業務を行う電子商取引推進協議会（ECOM）の場において、新たに SAML 利用検討 TF(TF7)を募集し、そのメンバーが主体となって推進した。

また、 検討作業は以下のような手順で実施した。

ⅰ）SAML 関連の技術動向調査

OASIS の文献を中心に、SAML や XML 署名などの関連文献を調査し、その技術動向や課題、将来性などについて、共通認識を持った。

ⅱ）SAML によるユースケースの検討

SAML によるユースケースを３分野８ケースについて想定し、それぞれに関して、システムに求められる機能要件、セキュリティ要件等を議論した。

ⅲ）SAML 利用検討報告書の作成

ⅰ)、ⅱ)の検討結果に関連情報を加え、本報告書としてまとめた。

本報告書が今後予想される SAML を活用した属性認証の発展に寄与することを確信する。

平成１６年３月

財団法人日本情報処理開発協会
電子商取引推進センター
電子商取引推進協議会

SAML の利用モデル

目次

1 はじめに 9

2 SAML の概要 11

2.1 SAML の概要 11

2.1.1 SAML (Security Assertion Markup Language) の特徴 11

2.1.2 SAML のモデル 13

2.1.3 SAML の利用例 14

2.1.4 SAML プロトコルと SAML アサーション 18

2.1.5 SOAP バインディングとプロファイル 23

2.2 Liberty Alliance Project の概要および SAML との関係 27

2.2.1 Liberty Alliance Project について 27

2.2.2 Liberty Alliance と SAML の関係 29

3 SAML のユースケース 31

3.1 電子申請のワンストップサービス（行政） 31

3.1.1 背景 31

3.1.2 登場人物 31

3.1.3 シナリオ 33

3.1.4 特徴 35

3.1.5 考察 36

3.2 電子申請のワンストップサービス（民間） 37

3.2.1 背景 37

3.2.2 登場人物 37

3.2.3 シナリオ 38

3.2.4 特徴 41

3.2.5 考察 41

3.3 インターネットオークションの起業申請 43

3.3.1 背景 43

3.3.2 登場人物 43

3.3.3 シナリオ 45

3.3.4 特徴 47
3.3.5 考察 47
3.4 電子カルテ 49
3.4.1 背景 49
3.4.2 登場人物 49
3.4.3 シナリオ 51
3.4.4 特徴 53
3.4.5 考察 54
3.5 電子紹介状 55
3.5.1 背景 55
3.5.2 登場人物 55
3.5.3 シナリオ 57
3.5.4 特徴 60
3.5.5 考察 61
3.6 在宅介護における病院と自治体の連携 62
3.6.1 背景 62
3.6.2 登場人物 62
3.6.3 シナリオ 64
3.6.4 特徴 66
3.6.5 考察 66
3.7 ネット投票/アンケート 68
3.7.1 背景 68
3.7.2 登場人物 68
3.7.3 シナリオ 70
3.7.4 特徴 74
3.7.5 考察 74
3.8 決済 76
3.8.1 背景 76
3.8.2 登場人物 76
3.8.3 シナリオ 77

3.8.4 特徴 80
3.8.5 考察 82
4 Liberty と WS-Federation について 83
4.1 Liberty Alliance アーキテクチャの概要 83
4.1.1 Web リダイレクトのアーキテクチャコンポーネント 84
4.1.2 Web サービスのアーキテクチャコンポーネント 89
4.1.3 メタデータとスキーマのアーキテクチャコンポーネント 89
4.1.4 シングルサインオンとアイデンティティ連携 90
4.1.5 IdP イントロダクション 95
4.1.6 シングルログアウト 97
4.2 WS-Federation の概要 99
4.2.1 連携シングルサインオンと ID 管理 100
4.2.2 連携 Identity モデル 102
4.3 Liberty Alliance と WS-Federation の相違の概要 109
4.3.1 仕様項目の相違 109
4.3.2 プロトコル上の差異の例 112
5 今後の課題 119
付録 D：用語集 121
参考文献 146
メンバーリスト 149

図目次

図 2-1 SAML モデル 14
図 2-2 シングルサイオン･モデル 15
図 2-3 SSO Pull モデル 16
図 2-4 SSO Push モデル 17
図 2-5 アクセス制御サービス 17
図 2-6 アクセス制御の手順 18
図 2-7 SAML プロトコル 19
図 2-8 要求プロトコル 19
図 2-9 応答プロトコル 20
図 2-10 認証問い合わせ 20
図 2-11 認証アサーション 21
図 2-12 属性問い合わせ 21
図 2-13 属性アサーション 22
図 2-14 認可決定問い合わせ 23
図 2-15 認可決定アサーション 23
図 2-16 SAML 要求(SOAP over HTTP)バインディング 24
図 2-17 SAML 応答(SOAP over HTTP) 25
図 2-1 連携ネットワーク ID とトラストサークル (引用元：Liberty アーキテクチャ概要 バージョン 1.1) 28
図 2-2 Liberty Alliance Project のロードマップ 28
図 2-3 SAML と Liberty Alliance の関係 29
図 2-4 Liberty Alliance の構成例 29
図 2-5 Liberty 仕様における SAML の利用例 30
図 3-1 電子申請（行政） 32
図 3-2 電子申請（行政）のユースケース図 33
図 3-3 電子申請（行政）のシーケンス図 35
図 3-4 電子申請（民間） 38
図 3-5 電子申請（民間）のユースケース図 38

図 3-6 電子申請（民間）のシーケンス図 41
図 3-7 インターネットオークションの起業申請 44
図 3-8 インターネットオークションの起業申請のユースケース図 45
図 3-9 インターネットオークションの起業申請のシーケンス図 47
図 3-10 電子カルテ構成図 50
図 3-11 電子カルテのユースケース図 51
図 3-12 電子カルテのシーケンス図 53
図 3-13 電子紹介状サービス 56
図 3-14 電子紹介状サービスのユースケース図 57
図 3-15 電子紹介状サービスのシーケンス図 60
図 3-16 在宅介護連絡システム 63
図 3-17 在宅介護連絡システムのユースケース図 63
図 3-18 在宅介護連絡システムのシーケンス図 66
図 3-19 ネット投票 69
図 3-20 ネット投票のユースケース図 70
図 3-21 ネット投票のシーケンス図 73
図 3-22 決済サービスのユースケース図 77
図 3-23 決済サービスのシーケンス図 80
図 4-1 Liberty の全体的なアーキテクチャ 83
図 4-2 ユーザエージェントを経由する SP と IdP 間の Web リダイレクト 84
図 4-3 HTTP リダイレクト機能による認証処理（Liberty Browser Artifact Profile） 86
図 4-4 FORM-POST ベースの認証処理（Liberty Browser POST Profile） 87
図 4-5 アイデンティティ連携 90
図 4-6 シングルサインオン 91
図 4-7 シングルサインオンプロトコルのフローの例 (ログオン) 94
図 4-8 シングルサインオンプロトコルのフローの例 (セッション確立) 95
図 4-9 IdP イントロダクションのフローの例 96
図 4-10 IdP イントロダクションのフローの例 97
図 4-11 シングルログアウト（IdP でログアウト） 98
図 4-12 シングルログアウト（SP でログアウト） 98

図 4-13 Web Service Security Roadmap 99
図 4-14 連携シングルサインオンと連携 ID の概念の例 （引用元：Federation of Identities in a Web Services World (white paper)） 100
図 4-15 IP と STS の組合せ例 104
図 4-16 仮名サービス（サイン・イン） 107
図 4-17 仮名サービス（サービス要求） 108

表目次

表 3-1 登場するアクター収集情報 76
表 4-1 Liberty アーキテクチャのコンポーネント 84
表 4-2 IdP と SP のパターンについて 91
表 4-3 4つのプロファイル 92
表 4-4 仕様項目の相違 110

1 はじめに

本 TF では平成 15 年度の認証・公証 WG の活動の一つとして OASIS（Organization for the Advancement of Structured Information Standards）で標準化が行われている SAML（Security Assertion Markup Language）[SAMLCore]について、その利用方法を検討することになった。SAML は新しい認証・認可の仕組みとして標準化されたシングルサインオン（SSO）のメカニズムを提供すると同時に、プライバシーに考慮したアイデンティティの管理方法を提供するものとして注目されている技術である。

SAML はセキュリティのベースメカニズムは特に規定せず、ID/パスワード、共通鍵ベースの Kerberos や公開鍵の PKI などの基盤の上で稼動させることができる。ECOM 認証・公証 WG では PKI の応用として長期署名保存、タイムスタンプ、属性証明書の利用、証明書の利用モデルなど PKI を中心に検討してきたが､本 TF では新しい XML ベース仕様で、PKI に必ずしもとらわれない認証・認可の利用検討を図ることになった。

SAML は、認証、認可のオーソリティが標準としての認証、属性、認可決定のアサーションを発行し、このセキュリティトークンをアプリケーションが消費することで、オンラインのアプリケーションのセキュリティを強化する。SAML はまた標準仕様の確立でマルチベンダーのセキュリティアプリケーションの相互運用性を確保するものとして期待されている。

SAML は、またアイデンティティの連携（Identity Federation）を強化した Liberty Alliance Project[LibArch11]のベース技術となっており、SSO や認可サービスが Web サービスとして提供される。Liberty は多くの参加団体で検討されてきており、Liberty の成果は SAML 2.0 に反映されることになった。

本 TF7 の活動中の 2003 年 7 月、IBM、Microsoft などから同様のアイデンティティの連携サービスの仕様 WS-Federation[WSF-WP]が公開された。今後どの技術が主流になっていくかは市場の動向を見ていくしかないが、プライバシーの保護が重要な時代となってきている中で、アイデンティティ・マネージメントの一つの重要な要素技術として SAML は多くの支持を集めていくものと思われる。

電子政府のサービスも 2003 年度で準備が完了し、公的個人認証サービスが開始され、次年度から本格的な展開がなされていくと期待される。今後電子政府のサービスも Web のポータルを介して様々な行政機関が連携するようになっていくと思われ、その際の Web サービス連携が重要な役割を果たすようになると思われる。このようなオンラインの連携のサ

ービスをインターネットで提供するとなると、サービスのフロントエンドでの認証が必須であり、それらは利便性のために SSO サービスでなければならない。

米国政府は今後の電子政府サービスのアーキテクチャを策定するために 2003 年 10 月に e-Authentication Initiative[eAuth][NISTeAuth]を立ち上げ 2004 年 6 月までにその基本仕様をまとめようとしている。ここでは市民サービスのフロントとしてアイデンティティ連携が可能な SSO を前提に考えられており、ユーザ認証の方法も、幾つかのセキュリティレベルに応じて ID/パスワードから連邦政府 PKI（FPKI）における証明書等の利用を可能としている。

本 TF ではこれらの動向を探りながら、本年度の前半では膨大にある SAML、Liberty、WS Federation などの仕様の資料を勉強することから始めた。これらの理解をベースに本年度の主要な成果物として、第 3 章に SAML を幾つかの分野に応用したユースケースをまとめた。それは電子政府分野、e コマース分野、医療分野から複数のテーマを取り出し、SAML を適用した場合のユースケースとそのシナリオを作成した。これらのテーマは SAML の適用事例を明確にするために、現実のモデルをシンプル化した仮想のものである。しかし、これらのユースケースは、実際に SAML をベースにしたオンラインアプリケーションのイメージを提供することで、SAML の有用性を理解するのに役立ててもらえると期待している。

ユースケースとシナリオおよび手順は UML（Unified Modeling Language）の手法を適用して、UML のユースケース図、シナリオ、シーケンス図として記述した。

また新しい SSO の理解のために、第 2 章に SAML の概要をまとめ、第 4 章では Liberty 仕様と WS Federation の比較を行った。

SAML のセキュリティとプライバシー保護については十分に理解しておく必要があると思われるので、セキュリティとプライバシーの考察を行っている SAML の付属文書、Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1 の主要部分の翻訳を付録に付けた。

2 SAML の概要

2.1 SAML の概要

SAML は OASIS Security Service TC で策定された認証、認可の要求／応答のプロトコルと、セキュリティトークンとしてのアサーションを定めた標準である。2002 年 11 月に SAML v1.0 が OASIS 標準となり、また 2003 年 9 月に v1.１が OASIS 標準として発行された。現在（2004 年 2 月）、Liberty Alliance の SAML 拡張を取り入れ SAML v2.0 ドラフトが検討されている。本解説は、SAML の基本概念を解説するもので、正確には付録の SAML 概要および OASIS (Organization for the Advancement of Structured Information Standards)の最新版を参照のこと。

本解説では、SAML の概要とその利用法について述べるとともに、SAML 要求／応答プロトコルとアサーションの構文およびメッセージングプロトコル SOAP や HTTP へのバインディングについて述べることにする。

本稿は@IT の Security Forum に Web サービスのセキュリティ[1]について鈴木が解説した SAML の記事をベースに書き直したものである。

2.1.1 SAML (Security Assertion Markup Language) の特徴

多くの関連する Web サイトやページにアクセスする時、それぞれから独立した認証を求められるのでは極めて面倒であるばかりではなく、それぞれの ID やパスワードを個別に覚えておくことは難しい。シングルサインオン（SSO）は Web ページやサイトの連携をはかるために必須な機能といえよう。

従来このような SSO は認証クッキーを使って実現してきた。しかし、認証クッキーは同じクッキードメイン内での SSO に制限されている。またクッキーは伝達する認証クッキーをブラウザにキャッシュすることが出来るため、この認証クッキーを第三者が不正使用してなりすましを許す可能性があることからセキュリティ上の問題が指摘されてきた。

SAML はクッキーを多用せず、クッキーの柔軟性を継承し、クッキーのもつスケーラビリティの制限とセキュリティ問題を解決することを目指して設計された。SAML の第一の目的は柔軟でかつ強化されたセキュリティの SSO を実現することである。しかし、実際のセキュリティアプリケーションでは、認証の後に利用者の資格などの属性によってアクセ

スできるページや Web サイトを制限したり、また与えられたアクセス権限によりリソースへのアクセスを制御したりする、いわゆる認可サービスが必要である。SAML は認証情報伝達サービス（Authentication Assertion）に加えて２つの認可サービスが加えられている。１つは属性情報の伝達（Authorization Assertion）であり、もう１つはアクセス制御情報の伝達（Authorization Decision Assertion）である。

SAML 仕様は PKI のセキュリティ環境の利用のみを前提にしたものではなく、セキュリティを強化した ID/パスワードから対称鍵による認証も含む幅の広いものとなっている。しかし、SAML に PKI を導入することによって強力なセキュリティを持つ SSO と柔軟で強力な属性制御を実現できるのである。

SAML 標準の 1 つの目的は、異なるベンダのアクセス制御製品間の相互運用性を推進することである。現状では異なるベンダ製品で既に大規模に展開されたサイト間での SSO を実現することが難しい。しかし、SAML 標準を実装したアクセス制御製品間では連携 SSO が可能となる。Liberty Alliance 仕様もこのような環境の実現を目指している。

SAML 1.1 は OASIS によって 2003 年 9 月に以下のような標準仕様書群が発表されている。SAML スキーマ仕様や関連する仕様の正確な情報は以下の一連の仕様書を参照してほしい。

SAML 1.1（Security Assertion Markup Language）

http://www.oasis-open.org/committees/security/#documents

・Assertions and Protocol [SAMLCore]

－SAML の Assertions のスキーマと要求／応答プロトコルを定めた仕様

・Assertion Schema [SAMLXsd]

・Protocol Schema [SAMLPXsd]

・Bindings and Profiles [SAMLBind]

－SAML を SOAP と HTTP にバインドする仕組みと SSO プロファイルの規定

・Security and Privacy Considerations [SAMLSec]

－SAML のセキュリティ要件を考察したもの

・Conformance Program Specification [SAMLConform]

－SAML の相互運用性を確保するために適合性要件をまとめたもの

---

[1] @IT Security http://www.atmarkit.co.jp/fsecurity/webserv04/webserv01.html

・Glossary [SAMLGloss]

－SAML で用いる用語の定義

2.1.2 SAML のモデル

SAML はセキュリティ情報交換のための XML ベースのフレームワークである。SAML は要求と応答のプロトコルと応答に含まれるアサーションの構文仕様を定めたものである。このセキュリティ情報は対象とする主体（Subject：人またはコンピュータ）のあるセキュリティドメインにおける認証情報や属性情報や認可情報をアサーションの形式で表現する。

SAML オーソリティには以下の３つのオーソリティがあり、要求者の問合わせに対して、

- 認証オーソリティは認証情報の再利用（SSO）を可能にする「認証アサーション」を提供する
- 属性オーソリティはポリシーに定めた資格や役職などの「属性アサーション」提供する
- 認可決定オーソリティはポリシーで定めた規則に従った「認可決定アサーション」提供する

リソースへのアクセスを要求するシステムエンティティはオーソリティからアサーションの提供を受け、ポリシーを実行するアプリケーション（PEP：Policy Enforcement Point）に渡し要求を実行する。すなわち、

- 本人性（Identity）を認証し、指定した Web サイトにアクセスさせる
- 本人の資格属性によって特定のページにアクセスを許可する
- 本人の認可権限によって特定のリソースへのアクセス（読み、書き、実行など）させる

図 2-1に SAML のフレームワークを示す。図に示すように System Entity（主体のクライアントまたはクライアントからアクセス要求を受けたサーバ）は、まず SAML 認証オーソリティに SAML 認証要求としてクレデンシャル（パスワードや鍵情報など）を示す。認証オーソリティは認証ポリシや外部の認証環境（PKI など）を検証して、主体の本人性（Identity）を証明する認証アサーション（Authentication Assertion）を発行する。次にこの認証アサーションを属性オーソリティまたは認可決定オーソリティ（PDP: Policy

Decision Point）に示して認可権限を問合わせる。属性オーソリティはポリシーに沿った主体の資格などの属性アサーションを発行し、認可決定オーソリティに主体のアクセス権限を問合わせる。認可決定オーソリティは予め登録された認可権限をポリシデータベースから検索し認可決定アサーションを発行し、実際にアクセス制御を行う PEP（Policy Enforcement Point）に渡しアプリケーション要求に沿った Web ページへのアクセスやリソースへのアクセスを実行させるのである。

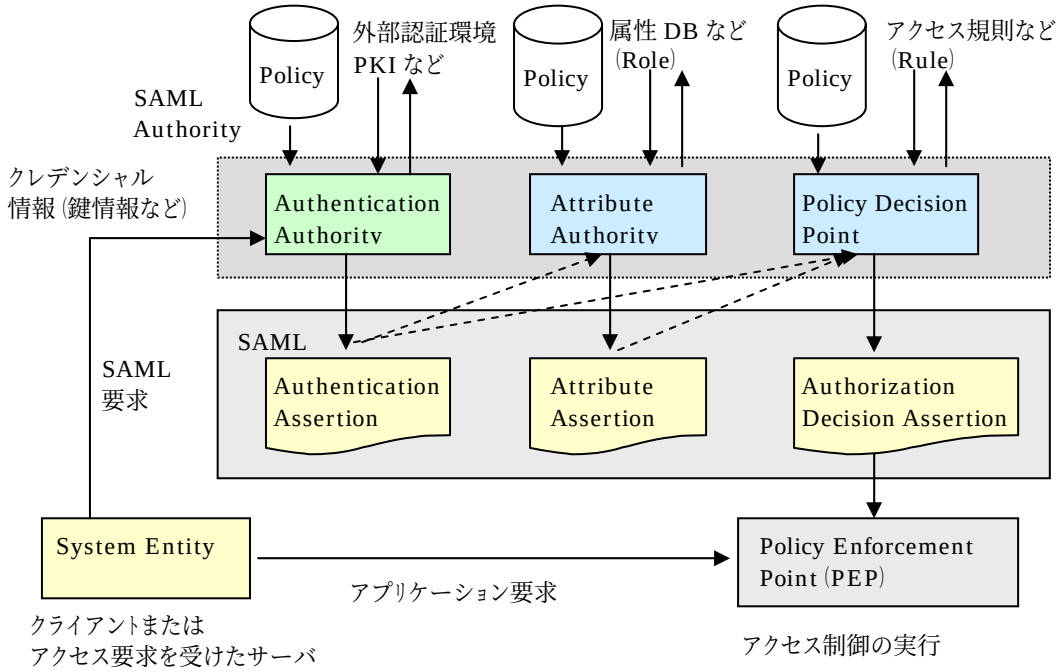


図 2-1 SAML モデル

図 2-1に示した３つの SAML オーソリティは必ず必要ではない。SSO を実現するだけなら、認証オーソリティだけを用いるだけでよい。しかし、属性や認可制御を木目細かに制御するためには、認証オーソリティの認証アサーションを属性オーソリティやポリシー決定点に提示し認可決定のアサーションを受ける必要がある。

2.1.3 SAML の利用例

SAML の主要な利用法はシングルサインオン（SSO）である。また属性や認可情報を使った Role ベースや Rule ベースのアクセス制御への利用が可能である。ここでは幾つかの

典型的な利用例についてのべる。この利用例は SAML 仕様の補助ドキュメントとして OASIS Security Service Use Case and Requirements に述べられている。

2.1.3.1 シングルサインオン(SSO)

SSO の概念は図 2-2に示すように、（Step 1）Web ユーザは最初にソース Web サイト（認証オーソリティ）で認証を受ける。（Step 2）その後この認証情報を用いて他の Web サイトに再度認証することなくアクセスできる。

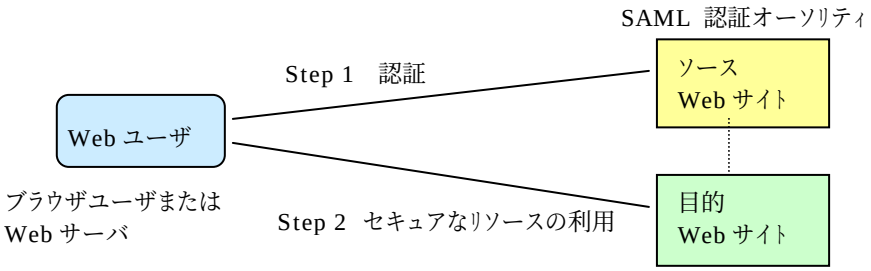


図 2-2 シングルサイオン･モデル

SSO には SAML を用いて以下の２つのモデル（Pull モデルと Push モデル）が考えられる。Pull モデルはアクセス要求を受けたサイトが Web ユーザの認証情報をオーソリティに問合わせるモデルであり、Push モデルは要求者の依頼でオーソリティがアクセス予定のサイトに認証情報を事前に伝えるモデルである。

[SSO Pull モデル]

Pull モデルでは、図 2-3に示すような SSO 認証の手順をとる。

① Web ユーザがクレデンシャル（パスワードまたは公開鍵または認証の参照先）を示す
② ソース Web サイトで認証を受けた後に、
③ 目的 Web サイトにアクセスすると、
④ 目的 Web サイトは Web ユーザの認証情報をソース Web サイトに問合わせ、
⑤ 認証アサーションを引き出す（Pull）。
⑥ その後 Web ユーザは目的 Web サイトのリソースを利用出来るようになる。

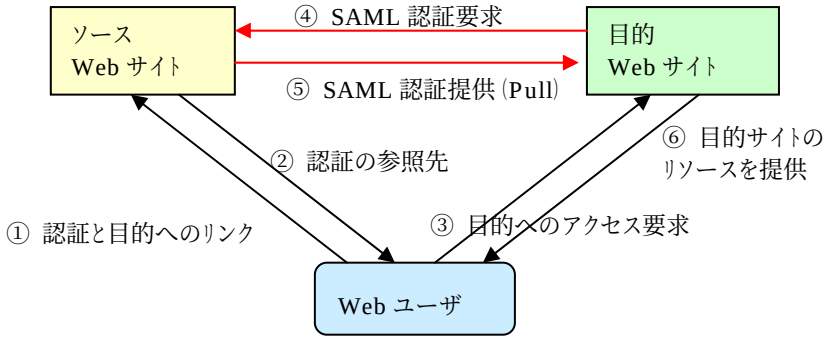


図 2-3 SSO Pull モデル

[SSO Push モデル]

SSO Push モデルでは Pull モデルと違って、図 2-4に示すような手順をとる。

① Web ユーザがソース Web サイトへ認証と目的 Web サイトへのリンク要求を出すと、
② ソース Web サイトは Web ユーザのために目的 Web サイトへアクセス権限を証明するアサーションをだす（Push）。
③ 目的 Web サイトはアクセス許可をしたソース Web サイトへアクセス決定の参照先を提供する。
④ ソース Web サイトは Web ユーザに決定の参照先を提供し、目的 Web サイトへのリダイレクトを促す。
⑤ Web ユーザは認証決定の参照を提示し目的 Web サイトにアクセス要求をだす。
⑥ 目的 Web サイトは Web サイトへのリソースへのアクセスを提供する。

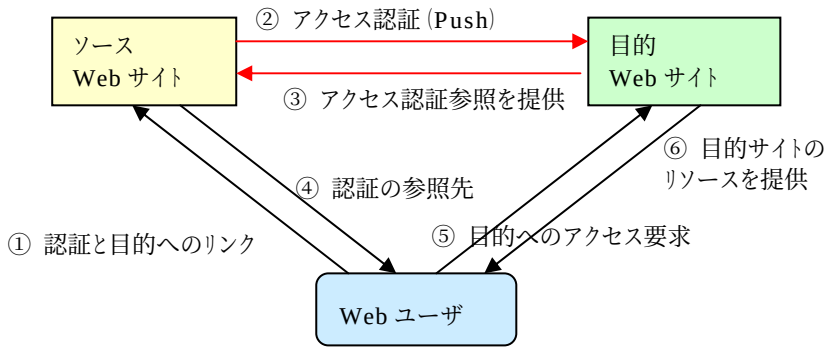


図 2-4 SSO Push モデル

2.1.3.2 認可サービス

アクセス制御サービスのモデルを図 2-5に示す。基本的に 2 つの Step を必要とする。

Step1：Web ユーザはアクセス制御を実行する Web サイト PEP（Policy Enforcement Point）へアクセスを要求する。

Step2：PEP は認可決定オーソリティ（PDP: Policy Decision Point）に問合わせ、Web ユーザのアクセス権限をチェックし、リソースへのアクセス制御を実行する

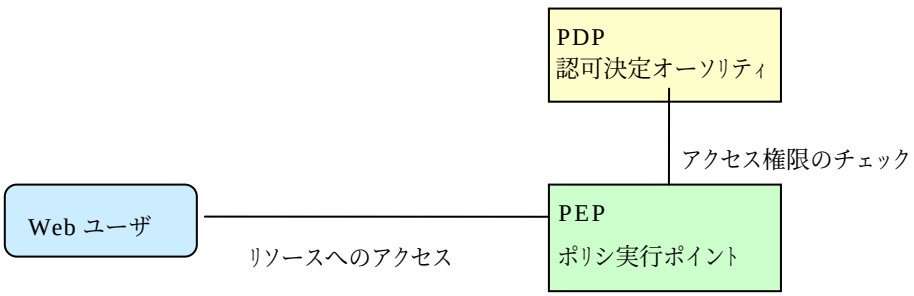


図 2-5 アクセス制御サービス

[認可モデル]

このモデルは図 2-6に示すような 1 つのセキュリティゾーン内の認可手順を示す。Web ユーザはセキュリティシステム（SAML オーソリティ）で認証した後に、Web サーバへ動的なリソースを要求する。Web サーバはアプリケーションにユーザの認可の権限をチェック出来るように認証情報を提供する。

このモデルではセキュリティシステムはユーザのクレデンシャルを収集し、SAML オーソリティ（認証オーソリティや属性オーソリティおよびポリシ決定点）として機能する。アプリケーションは PEP としての機能を持つ。

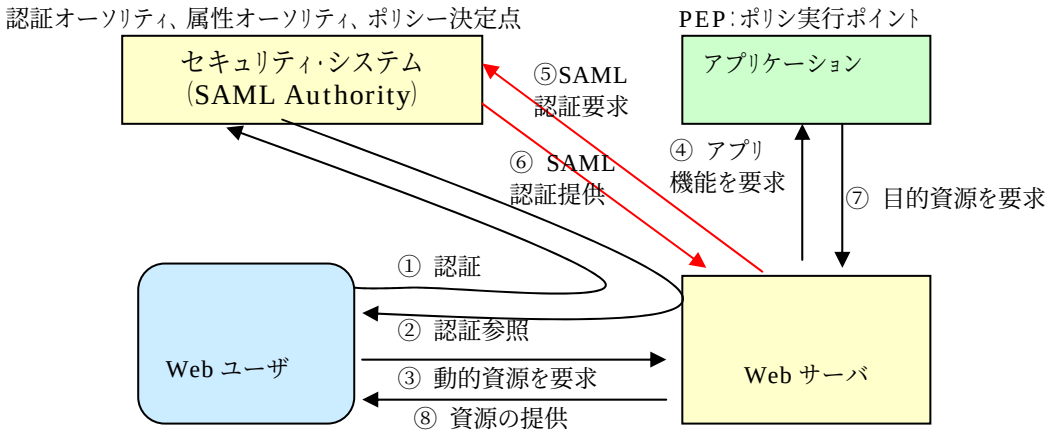


図 2-6 アクセス制御の手順

2.1.4 SAML プロトコルと SAML アサーション

SAML プロトコルは要求と応答の手順を定めたものである。実際のトランスポートのプロトコルは色々なものが使えるように SAML は下位のトランスポートプロトコルに独立に定めているが、１つのバインディングとして SAML over SOAP over HTTP の仕様を OASIS で定めている[SAML Bind]。

SAML プロトコルを図 2-7に示す。SAML クライアント（Requester）は要求プロトコル<Request>要素の子要素に各種の<Query>要素を指定して要求を送信する。SAML レスポンダ（SAML オーソリティ）は応答プロトコル<Response>要素の子要素に<Status>要素と<Assertion>要素を含めた応答を返す。<Response>処理で何らかのエラーが生じた場合、<Response>要素は<Status>要素にエラーコードを返すだけで<Assertion>要素は含まない。

SAML の<Request>、<Response>および<Assertion>にはデジタル署名（XML 署名）を付けることができる。デジタル署名はオプションであるが、デジタル署名を付けた場合、送信者の識別や否認防止を強力にサポートする(後述する SAML のセキュリティの項参照)。

SAML ではアサーション(saml)とプロトコル(samlp)のスキーマで定めた２つの名前空間を用いる。xmlns:saml と xmlns:samlp である。

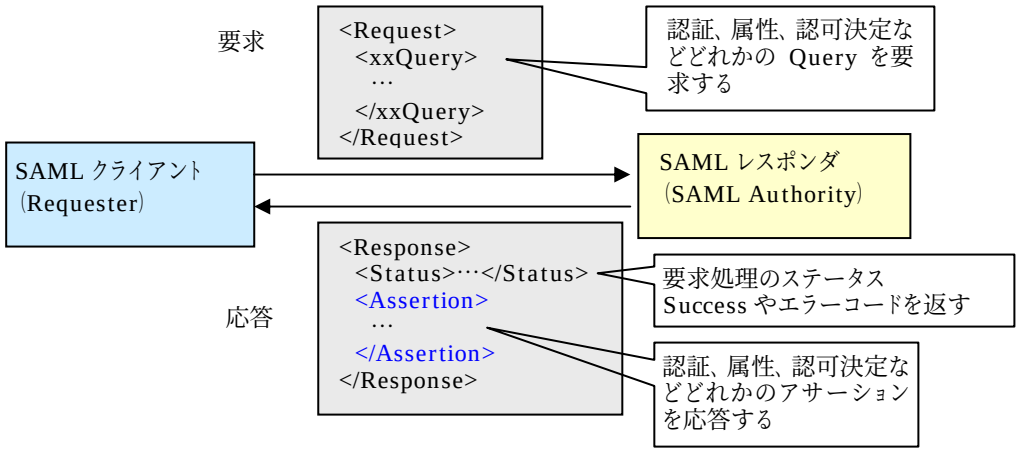


図 2-7 SAML プロトコル

2.1.4.1 SAML **要求プロトコル** <Request>

SAML<Request>要素は図 2-8に示す構文で構成される。

```
<samlp:Request
        RequestID="IE2002001"                          //リクエストID
        MajorVersion="1" MinorVersion="0"              //Requestバージョン番号
        IssueInstant="2002-08-03T10:02:00Z"/>          //発行日時（UTC）
        <samlp:RespondWith URI="    " />              //応答すべきタイプ
        <samlp:xxxQuery>
            //各種のxxxQueryの構文（以下に示す）
        </saml:xxxQuery>
       <ds:Signature> …. </ds:Signature>               //XMLデジタル署名：オプション
</samlp:Requst>
```

図 2-8 要求プロトコル

SAML <Request>の Query 要素は要求の種類（認証、属性、認可など）に応じて以下に示すような各種の Query 要素のどれか１つを指定することになる。

- 認証問合わせ要素 <AuthenticationQuery>
- 属性問合わせ要素 <AttributeQuery>
- 認可決定問合わせ要素 <AuthorizationDecisionQuery>

2.1.4.2 SAML **応答プロトコル** <Response>

SAML の応答である<Response>要素は、図 2-8に示した認証、属性、認可の要求に対して図 2-9のような構文となる。

```
<samlp:Response
        ResponseID="123456789"                                    //レスポンスID
        InResponseTo="IE2002001"                                  //リクエストID
        MajorVersion="1" MinorVersion="0"                         //Requestバージョン番号
        IssueInstant="2002-08-03T10:02:00Z"/>          //発行日時（UTC）
        <samlp:Status>
          <StatusCode>Success</Statuscode>                        //応答ステータスコード
        </StatusCode>
        <saml:Assertion>
            //SAMLアサーションの構文（以下に示す）
        </saml:Asertion>
       <ds:Signature> …. </ds:Signature>              //XMLデジタル署名：オプション
</samlp:Response>
```

図 2-9 応答プロトコル

応答プロトコルに含まれる<saml:Assertion>には認証、属性、認可決定の<Query>に対応する<Assertion>には、以下のような子要素が含まれる。

- 認証　<AuthenticationQuery> → <AuthenticationStatement>
- 属性　<AttributeQuery> → <AttributeStatement>
- 認可決定 <AuthorizationQuery> → <AuthorizationStatement>

2.1.4.3 各種問合わせ<Query>と対応する SAML アサーション<Assertion>の例

(1) 認証問合わせ要求 <AuthenticationQuery>と対応する認証アサーションの例

図 2-10は X.509v3 の証明書を使って認証アサーションを要求する例である。

```
<samlp:AuthenticationQuery>
   <saml:Subject>
      <saml:NameIdentifier>
         <SecurityDomain>abc.com</SecurityDomain>
         <Name>Suzuki@abc.com</Name>
      </saml:NameIdentifier>
      <saml:ConfirmationMethod>
         X.509v3
      </saml:ConfirmationMethod>
   </saml:Subject>
</samlp:AuthenticationQuery>
```

主体の認証方法（PKI）

図 2-10 認証問い合わせ

図 2-11に示す認証アサーションの例は図 2-10の認証要求<Request>に対するもので、PKI（X.509v3 の証明書）で主体を認証し、有効期間が 20 分であることを示している。

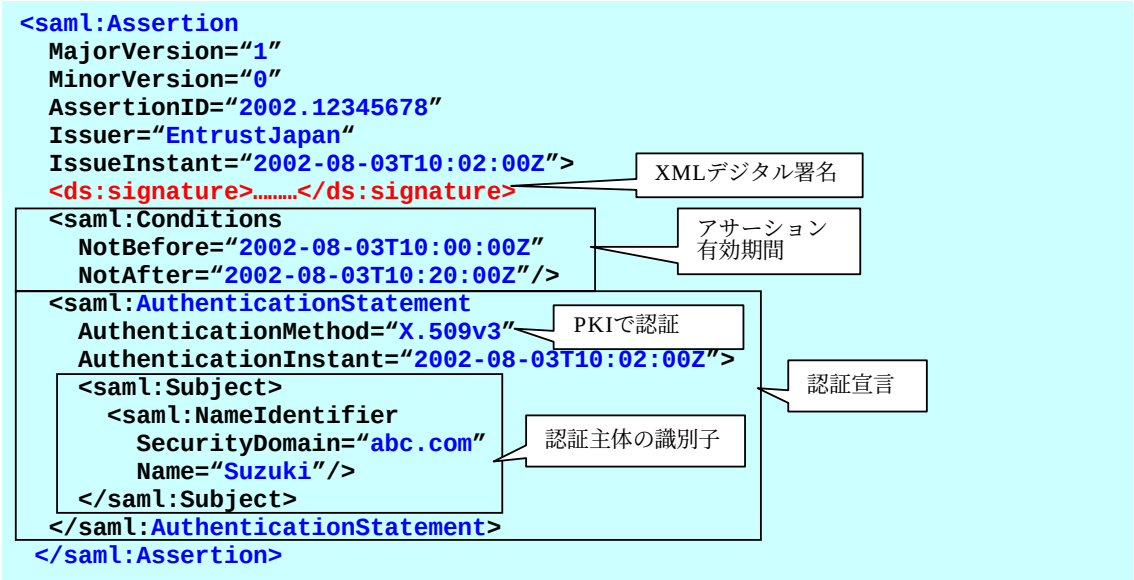


図 2-11 認証アサーション

(2)属性問合わせ要求 <AttributeQuery>と対応する属性アサーションの例

図 2-12に示す例では主体の認証について認証アサーションを添付し、主体の資格を問合わせている。

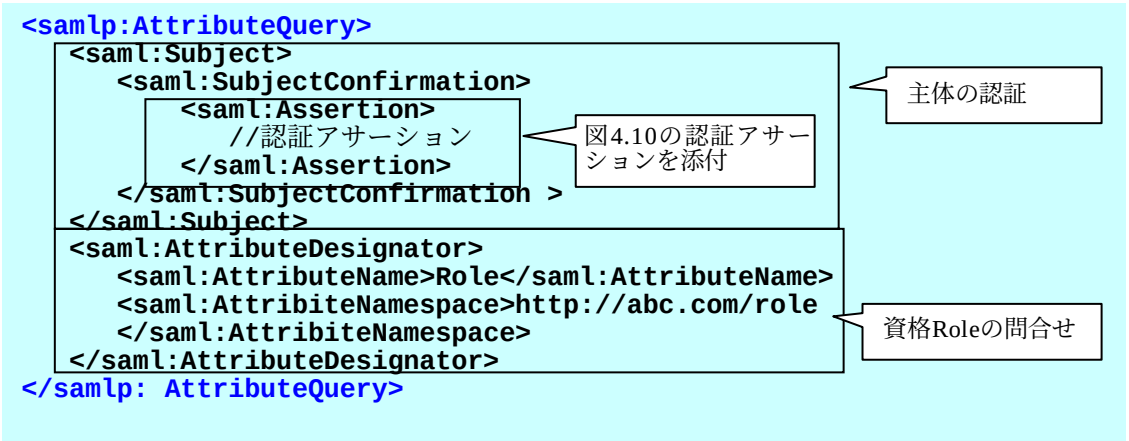


図 2-12 属性問い合わせ

図 2-13に示す例は、図 2-13に示した主体 "Suzuki" の資格の問合わせ要求に対して属性値 "Manager" を含む属性アサーションである。

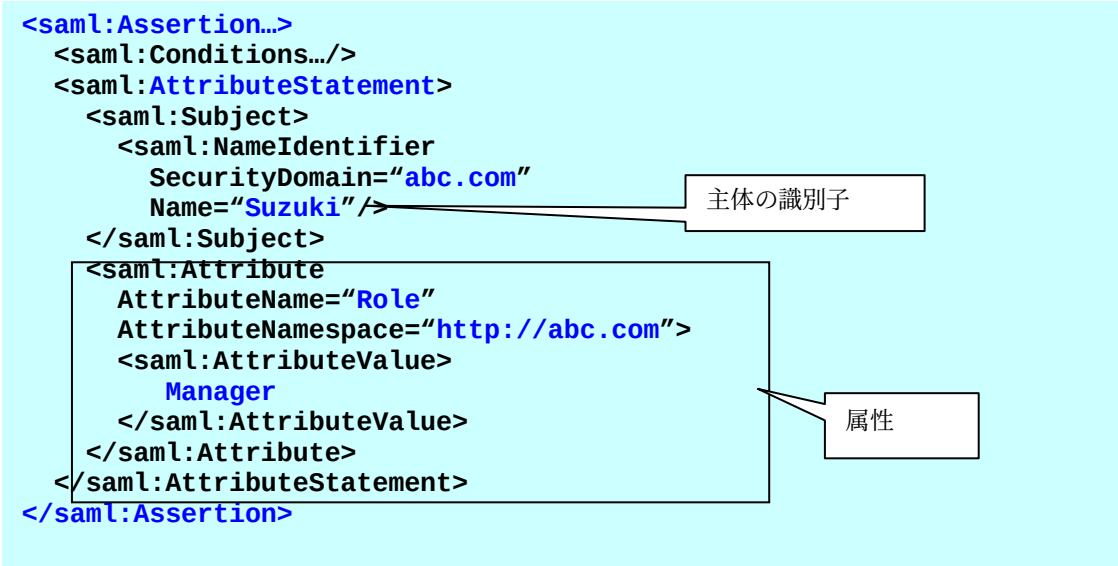


図 2-13 属性アサーション

(3) 認可決定問合わせ要求<AuthorizationDecisionQuery>と対応する認可決定アサーションの例

図 2-14に示す例では認証アサーションと属性アサーションを添付してリソースへのアクセスを要求している。この認可要求に対して図 2-15の認可決定アサーションが返される。

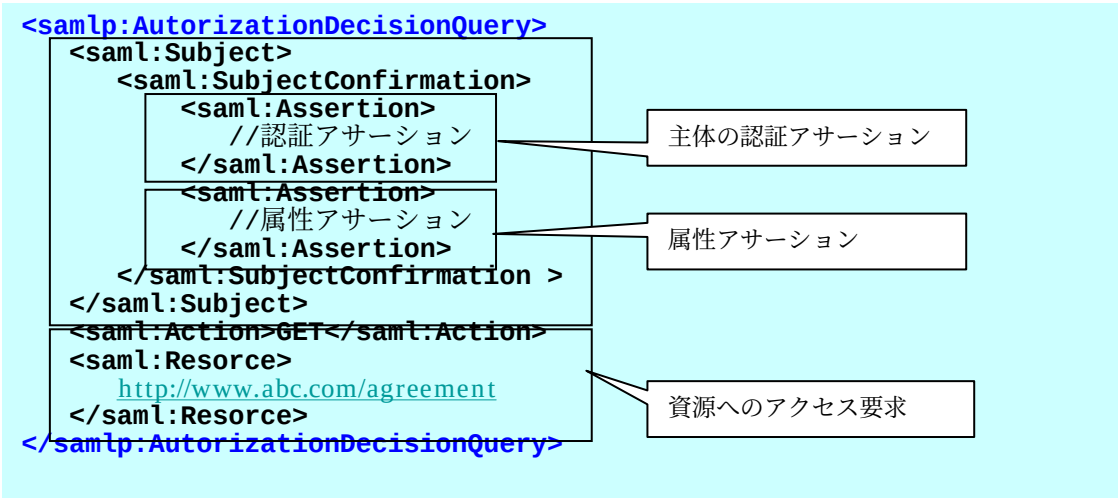


図 2-14 認可決定問い合わせ

図 2-15に示す例は、図 2-14で示した資格属性を指定した認可要求に対して、アクセスを許可（Permit）する認可決定アサーションの例である。この例ではリソースへのアクセスを許可（Permit）している。要求者に権限がなければ認可決定は不許可（Deny）となる。

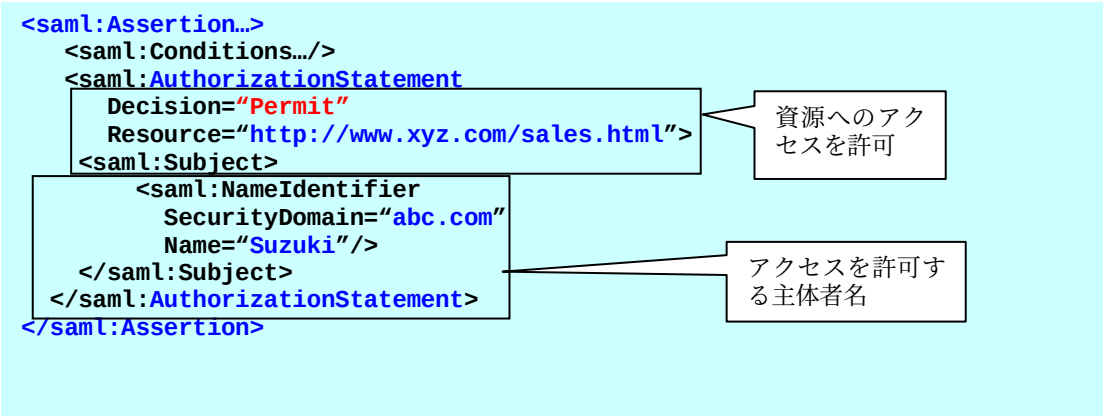


図 2-15 認可決定アサーション

2.1.5 SOAP バインディングとプロファイル

SAML 仕様は下位のトランスポートプロトコルとして特定のプロトコルのみを想定していないが、SAML over SOAP over HTTP を必須のプロトコルバインディングとして規定している[SAML Bind]。このバインディングでは要求／応答はいわゆる Web サービスとして

行われる。SAML を SOAP にバインドする際の基本的方針は以下の通りである。

- SOAP バインディングでは SOAP の Header は使わず、Body 内に SAML プロトコルを記述する。
- １つの SOAP メッセージには SAML<Request>は１つだけにする。
- １つの SOAP メッセージには SAML<Response>は１つだけにする。

2.1.5.1 SOAP over HTTP を使った SAML 要求／応答

図 2-16は SAML 認証を SAML 認証オーソリティに要求する場合の例である。

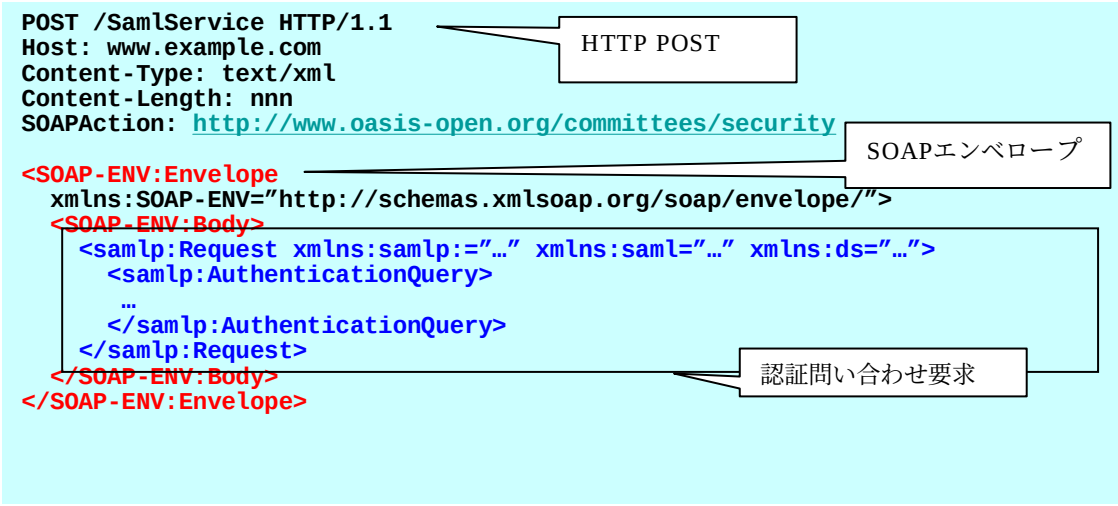


図 2-16 SAML 要求(SOAP over HTTP)バインディング

図 2-17は図 2-16の SAML 認証要求に対する応答の例で、認証アサーションが返される。

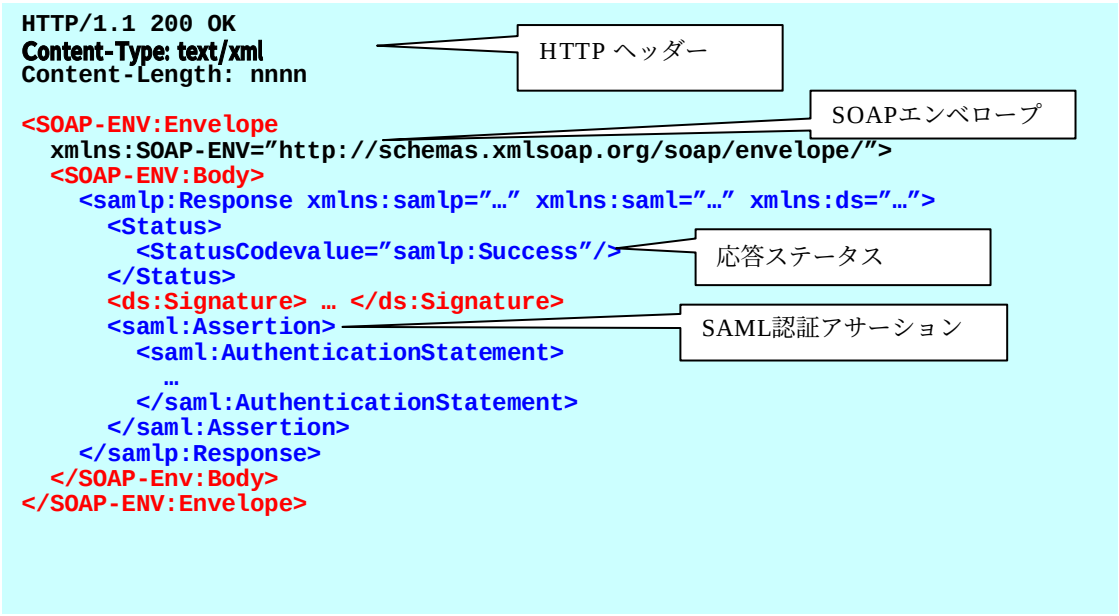


図 2-17 SAML 応答(SOAP over HTTP)

2.1.5.2 SAML SSO プロファイル

SAML のプロトコルはサーバ間での要求／応答を定めたものだが、ユーザインタフェースとして Web ブラウザを用いる場合 SAML プロトコルのプロファイルが必要になる。

Binding and Profiles for the OASIS Security SAML の仕様書ではシングルサインオン（SSO）を実現するに当っての Web ブラウザベースの SAML のプロファイルを規定している。ここでは Web ブラウザの SAML プロファイルとして、ブラウザ/アーティファクトプロファイルとブラウザ/POST プロファイルを定めている[SAMLBind]。

- ブラウザ/アーティファクトプロファイル

  パスワードベースの認証を想定している。ブラウザは URI に認証要求を Base64 でエンコードして認証オーソリティに訊ねる。認証オーソリティはパスワードのハンドルをブラウザに返し、目的 Web サイトにリダイレクトさせる。目的 Web サイトはブラウザユーザのパスワードのハンドルの真偽を認証オーソリティに問合わせてユーザを認証する。正しいユーザなら指定したページをアクセスさせる。これは図 2.3 の Pull モデルに相当する。
- ブラウザ/POST プロファイル

  デジタル署名の利用を前提にしている。ブラウザは SAML<Request>で認証オーソ

リティに認証要求<AuthenticationQuery>を HTTP の POST で問合わす。認証オーソリティは認証アサーションにデジタル署名をしてブラウザに返す。ブラウザはこの認証アサーションを目的 Web サーバに送る。目的 Web サーバは署名を検証し、アサーションの完全性とブラウザユーザ認証を行い、正しければ指定したページをアクセスさせる。これは Push モデルの一形態である。

SAML は Web サービスに PKI と PMI すなわち、認証や認可のセキュリティを与えることができるフレームワークとして注目されているサービスである。最近 Liberty Alliance が発表した Liberty 仕様は企業間連携サービスのためのプライバシを強化した SSO（Federated SSO）を実現するものとして期待を集めている。Liberty 仕様は SAML 認証に基づた実際のサービスを目指して、SAML に若干の拡張と曖昧性のないプロファイルを定義したものである。

2.2 Liberty Alliance Project の概要および SAML との関係

本節は以下の文献を元に記述した。

- Liberty アーキテクチャ概要 バージョン 1.1 [LibArch11]
- ID-FF アーキテクチャ概要 [LibArch12]

2.2.1 Liberty Alliance Project について

Liberty Alliance Project は、インターネット上での新しい水準の信頼、商取引、通信を推進するために結成された幅広い産業団体である。Liberty は、ID(アイデンティティ)情報のプライバシーやセキュリティを考慮した仮想的に取引参加可能なネットワーク化された世界を目指し、その実現化のために広域なネットワーク ID ベースのやりとりをオープンな技術仕様で策定するとされている。

ネットワーク ID とは、アカウント、名前、メールアドレス等の属性のグローバルセットである。Liberty ではネットワーク ID に次のような目標を掲げている。

利用者がネットワーク ID 情報のプライバシーとセキュリティを保護できること

企業が第三者の介入を受けることなく、顧客との関係を維持し、管理できること

複数ベンダからの分散型の認証と許可を行えるオープンなシングルサインオンの規格を提供する

現行および新規のすべてのネットワークアクセスデバイスをサポートするネットワーク ID インフラを構築する。

これらを実現するのがトラストサークルである。トラストサークルは、”Service Provider”(以下、”SP”と略)と”Identity Provider”(以降、”IDP”と略)が、Liberty 対応技術と企業間のビジネス上の運用協定に基づいて接続されたものである。その上でユーザが企業ごとのアカウントを連携することで、安全かつシームレスな環境で取引することが可能となる。

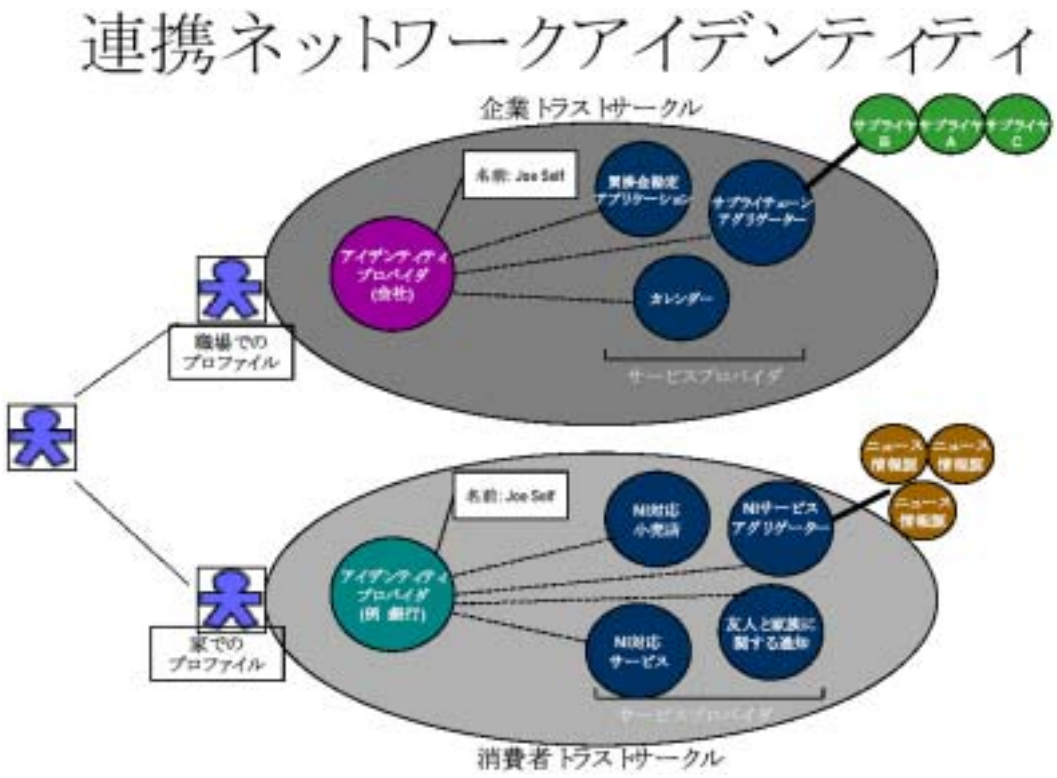


図 2-18 連携ネットワーク ID とトラストサークル

(引用元:Liberty アーキテクチャ概要 バージョン 1.1)

Liberty Alliance Project では、下記のように Phase に分けて検討を展開しており、現在、Phase2 が公開された段階である。

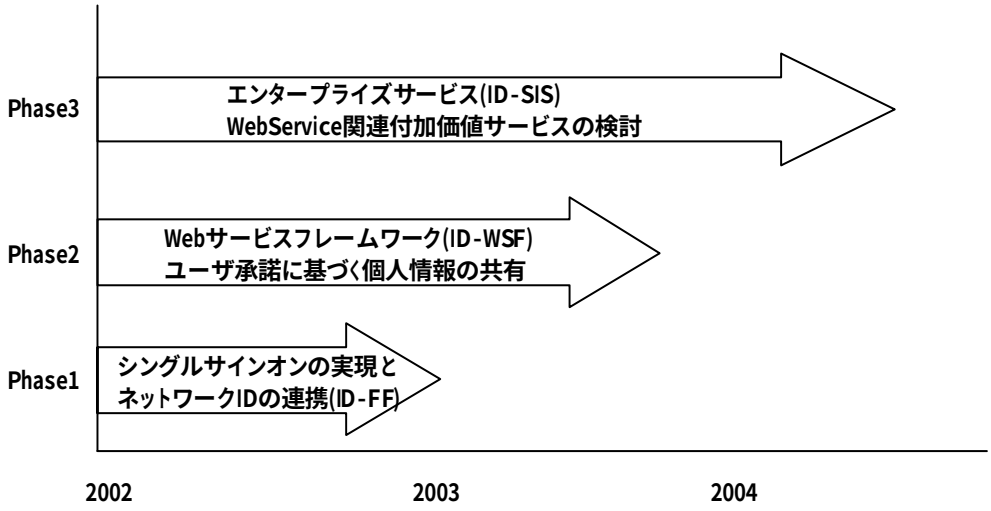


図 2-19 Liberty Alliance Project のロードマップ

2.2.2 Liberty Alliance と SAML の関係

SAML は Web サービスへのサイン・インにおける属性情報やアクセス制御情報を伝達するプロトコルとして用いられる。Liberty Alliance Project が公開した Liberty 仕様は、連携した企業間の Web サービスのシングルサインオンを実現するため、この SAML 認証に基づき、拡張やプロファイル定義を行ったものである。

つまり、Liberty 仕様は、SAML 認証を拡張した仕様ということができる。

図 2-20 SAML と Liberty Alliance の関係

以下に、Liberty 仕様における SAML の利用例を挙げる。

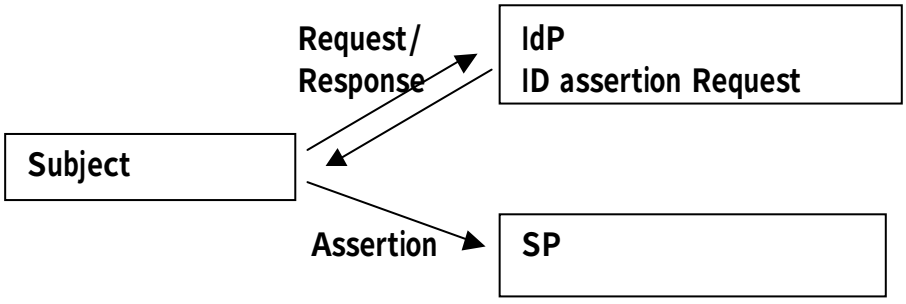


図 2-21 Liberty Alliance の構成例

IdPにAuthenticate対する要求

```
<samlp: Request ...>
  <samlp: AttributeQuery>
    <saml: Subject>
      <saml: NameIdentifier
        SecurityDomain="ecom.jp"
        Name="rimap"/>
    </ saml: Subject>

    <saml: AttributeDesignator
      AttributeName="Employee_ ID"
      AttributeNamespace="ecom.jp">
    </ saml: AttributeDesignator>
  </ samlp: AttributeQuery>
</ samlp: Request>
```

SAMLAuthenticationレスポンス

```
<samlp: Response
  MajorVersion="1" MinorVersion="0"
  RequestID="128.14.234.20.90123456"
  InResponseTo="123.45.678.90.12345678"
  StatusCode="Success">
```

SAMLAuthenticationアサーション

```
<saml: Assertion
    MajorVersion="1" MinorVersion="0"
    AssertionID="123.45.678.90.12345678"
    Issuer="Ecom.jp"
    IssueInstant="2002- 01- 14T10: 00: 23Z">

    <saml: Conditions
      NotBefore="2002- 01- 14T10: 00: 30Z"
      NotAfter="2002- 01- 14T10: 15: 00Z" />

    <saml: AuthenticationStatement
    AuthenticationMethod="Password"
    AuthenticationInstant="2001- 01- 14T10: 00: 20Z">

      <saml: Subject>
        <saml: NameIdentifier
          SecurityDomain="ecom.jp"
          Name="rimap" />
      </ saml: Subject>
    </ saml: AuthenticationStatement>
  </ saml: Assertion>
</ samlp: Response>
```

図 2-22 Liberty 仕様における SAML の利用例

3 SAML のユースケース

3.1 電子申請のワンストップサービス（行政）

SAML により行政機関の窓口の連携を行い、引越しのための電子申請手続きを一度のログインで可能にし、住民へのサービス向上を図る。プロファイルとしてアーティファクトを用いる。

3.1.1 背景

住民が A 市から B 市に引越しするケースを想定し、このときの行政手続を例とする。住民の家族構成は３人家族で小学生１名を含んでいる。住民は引越しポータル窓口で引越しのための電子申請を行うことで、市役所 A から市役所 B への住民票の異動手続きと、教育委員会 A から教育委員会 B への転校手続きがオンラインで行うことが可能である。ただし本ケースでは行政機関の窓口が連携するユースケースであるため、ユースケース図やシーケンス図において住民を登場させていない。

行政機関は、手続きが信頼できる行政機関からの申請要求かどうかを確認するために、SAML オーソリティが発行したアサーションを利用する。アサーションの署名を検証するための SAML オーソリティの証明書は、LGPKI の認証局から発行されているものとする。SAML オーソリティは引越しポータル窓口や行政機関を PKI により認証することとする。引越しポータル窓口は民間の認証局から証明書の発行を受けており、行政機関は LGPKI の認証局から証明書の発行を受けているものとする。

3.1.2 登場人物

SAML オーソリティ	引越しポータルと行政機関の ID を管理している。
引越しポータル窓口	住民からの引越し申請をとりまとめ、行政機関に申請手続きする。
市役所 A	A 市の市役所で住民票を管理している。同市の教育委員会と関係を持っている。(引越し元)
市役所 B	B 市の市役所で住民票を管理している。同市の教育委員会と関係を持っている。(引越し先)
教育委員会 A	A 市の教育委員会で市内の学生を管理している。(転校元)。
教育委員会 B	B 市の教育委員会で市内の学生を管理している。(転校先)。

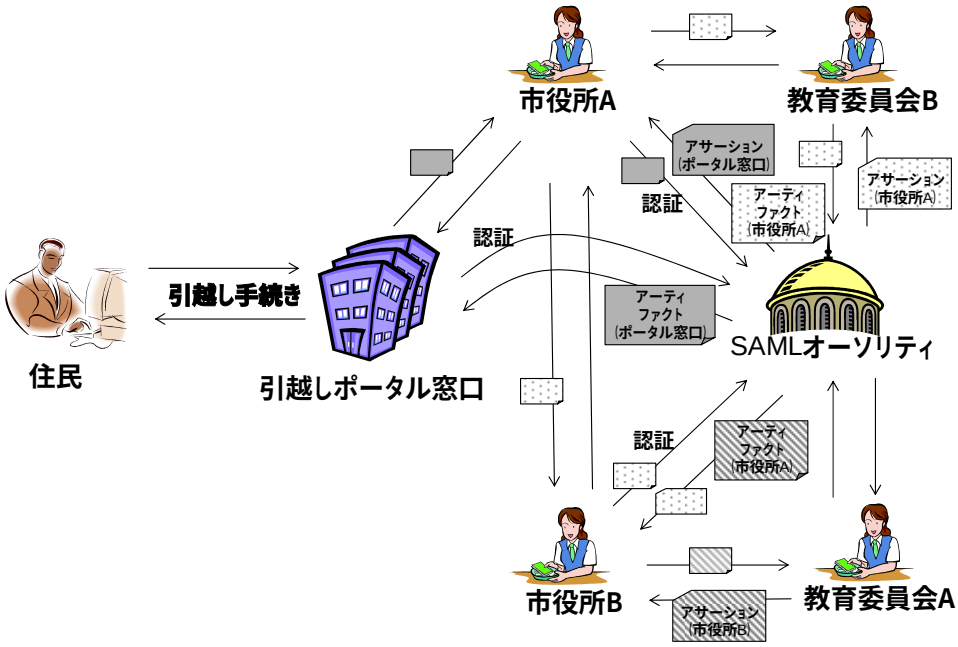

市役所A
教育委員会B
アサーション
(ポータル窓口)
認証
アーティ
ファクト
(市役所A)
アサーション
(市役所A)
認証
引越し手続き
アーティ
ファクト
(ポータル窓口)
SAMLオーソリティ
住民
引越しポータル窓口
認証
アーティ
ファクト
(市役所A)
市役所B
アサーション
(市役所B)
教育委員会A

図 3-1 電子申請(行政)

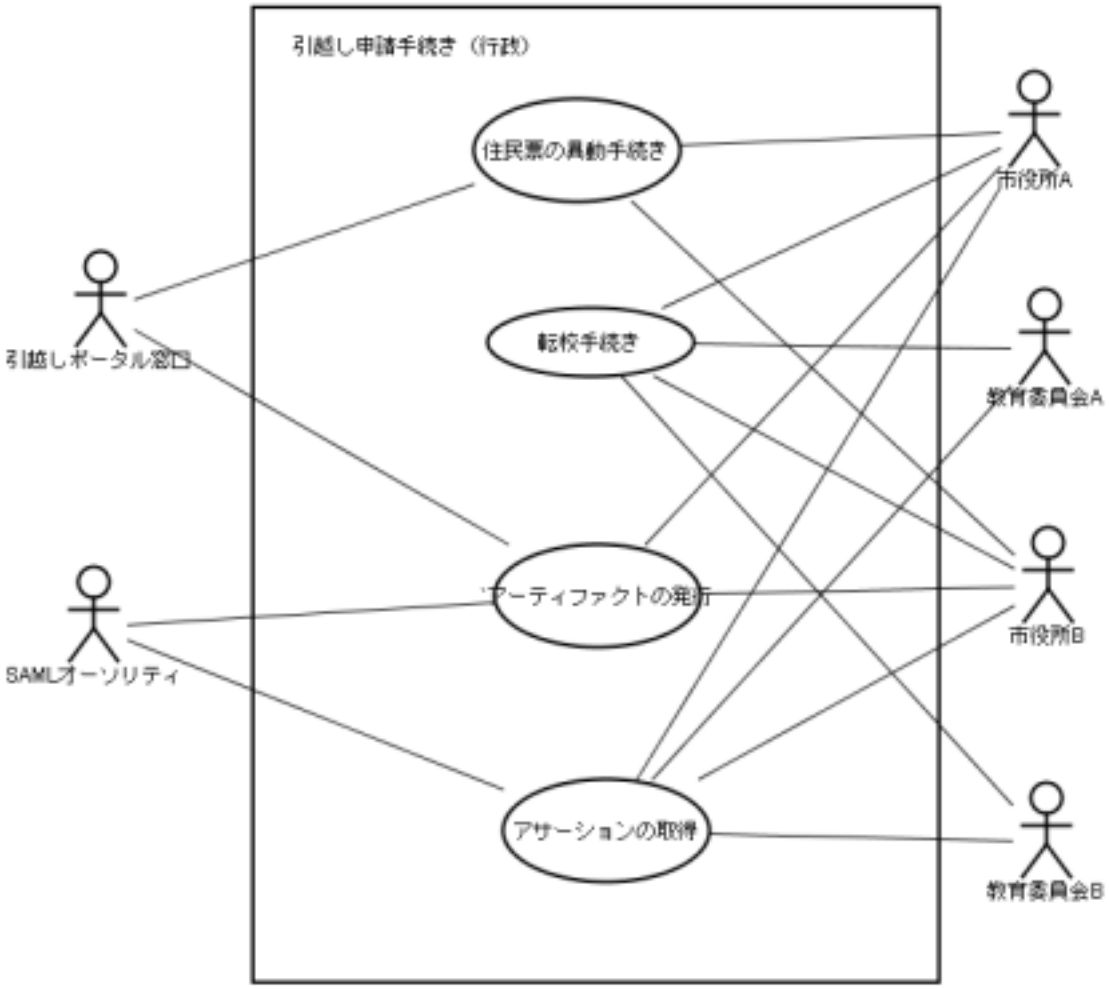


図 3-2 電子申請(行政)のユースケース図

3.1.3 シナリオ

- 手順1

  引越しポータル窓口は、住民から引越し申請を受けると、SAML オーソリティにログインする。SAML オーソリティは、引越しポータル窓口の認証に成功すると、アサーションとアーティファクトを生成する。そしてアサーションを SAML オーソリティ内に格納し、引越しポータルにアーティファクトを送信する。

- 手順2

  引越しポータルは、市役所 A に引越し手続きの申請を行う。

- 手順3

  市役所 A は、SAML オーソリティに引越しポータルの認証情報の取得を行う。SAML

オーソリティは、市役所 A に認証アサーションを送信する。市役所 A は、アサーションの署名を検証して SAML オーソリティが発行したことを確認する。

- 手順４

  市役所 A は引越し申請の内容から家族に小学生１名が存在することを知ると、小学生の転校手続きを行うために教育委員会 A に転校手続きを行う。転校手続きのために市役所 A は SAML オーソリティにログインする。SAML オーソリティは、市役所 A の認証に成功すると、アサーションとアーティファクトを生成する。そしてアサーションを SAML オーソリティ内に格納し、市役所 A にアーティファクトを送信する。

- 手順５

  市役所 A は教育委員会 A に転校手続きを行う。

- 手順６

  教育委員会 A は、SAML オーソリティに市役所 A の認証情報の取得を行う。SAML オーソリティは、教育委員会 A に認証アサーションを送信する。教育委員会 A は、アサーションの署名を検証して SAML オーソリティが発行したことを確認する。転校手続きのための在学証明書を発行して、市役所 A に在学証明書を送信する。

- 手順７

  市役所 A は、市役所 B に引越し手続きを行うために住民票と在学証明書をそろえて市役所 B に引越し手続きを行う。

- 手順８

  市役所 B は、SAML オーソリティから市役所 A の認証情報の取得を行う。SAML オーソリティは市役所 B に認証アサーションを送信する。市役所 B は、アサーションの署名を検証して SAML オーソリティが発行したことを確認する。市役所 B では小学生の転校手続きのために教育委員会 B に転校手続きを行う。

- 手順９

  転校手続きのために市役所 B は SAML オーソリティにログインする。SAML オーソリティは、市役所 B の認証に成功すると、アサーションとアーティファクトを生成する。そしてアサーションを SAML オーソリティ内に格納し、市役所 B にアーティファクトを送信する。

- 手順１０

  市役所 B は、教育委員会 B に転校の申請手続きを行う。

- 手順１１

  教育委員会 B は、SAML オーソリティから市役所 B の認証情報の取得を行う。SAML オーソリティは、教育委員会 B に認証アサーションを送信する。教育委員会 B は、アサーションの署名を確認して SAML オーソリティが発行したことを確認する。教育委員会 B は転校手続きを行い、市役所 B に転校手続き完了の通知を行う。市役所 B は住民票の異動と転校手続きが完了したことから、市役所 A に引越し手続き完了の通知を行う。市役所 A は引越しポータル窓口に引越し手続き完了の通知を行い、引越しポータル窓口から住民に引越し手続き完了の通知が行われる。

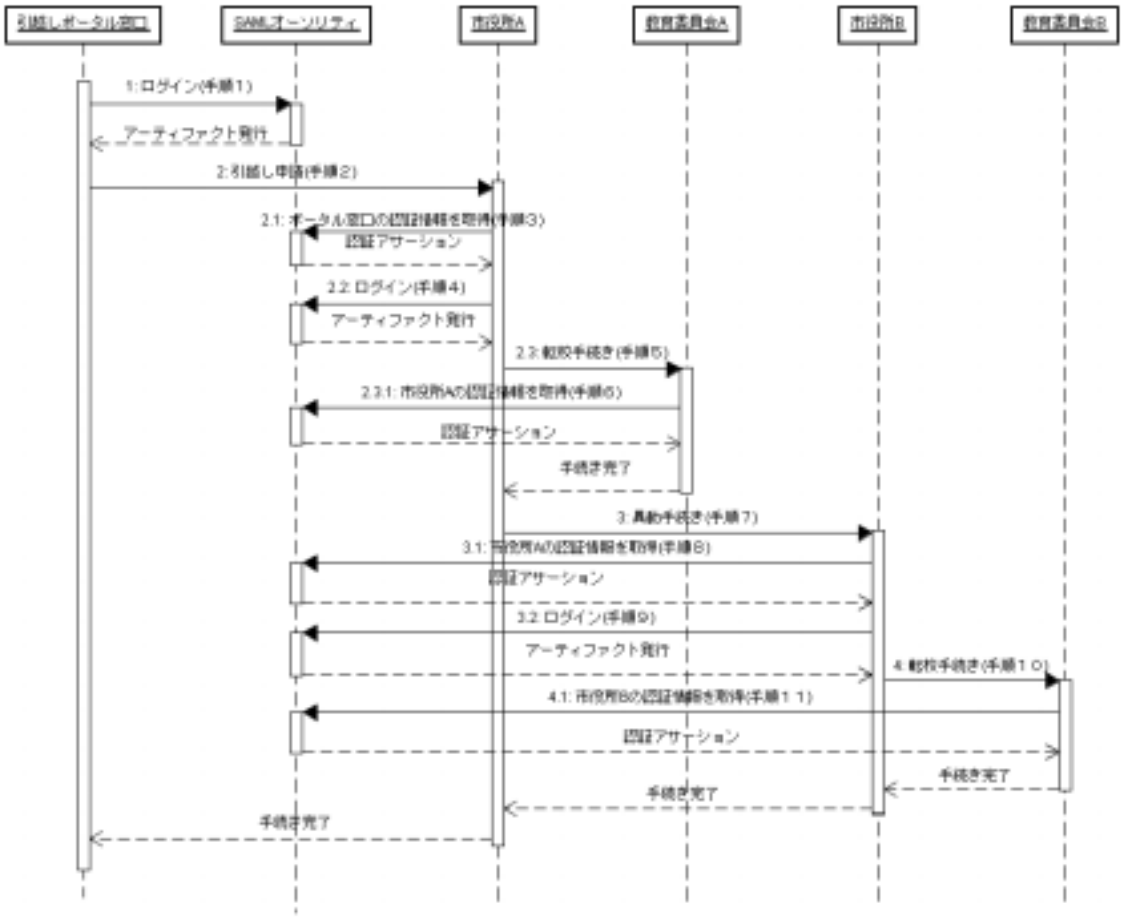


図 3-3 電子申請(行政)のシーケンス図

3.1.4 特徴

SAML オーソリティは、市役所 A の認証アサーションをキャッシュすることで、認証アサーションの有効期間内で市役所 A を再認証しない。市役所 A は教育委員会 A との通信後、

市役所 B と通信するときキャッシュしたアーティファクトを送信する。SAML オーソリティは、認証アサーションをキャッシュすることで処理負荷を低くすることが可能である。

3.1.5　考察

本ユースケースでは、市役所や教育委員会のように互いの組織が連携して手続き業務を進めるような比較的信頼関係がある場合のモデルある。このモデルは信頼関係を持つ組織以外の第三者になりすましを行われなければよいケースであるため、アサーションやアーティファクトのキャッシュを可能としている。そのため本ユースケースでは教育委員会 A や市役所 B がアーティファクトを再利用して市役所 A になりすますことができる可能性がある。しかし教育委員会 A や市役所 B が市役所 A になりすましたとしても、教育委員会 A や市役所 B は手続き業務上自らの業務が支障をきたす恐れがあり意味がないと思われる。

このようなモデルでは、アーティファクトやアサーションをキャッシュすることで SAML オーソリティの負荷が低くなるため、本モデルの検討を考えてみてもよいと考えられる。

本ユースケースでは、引越しの際の手続きとして、住民票の異動と学生の転校を例に挙げたが、他に国民年金や、国民健康保険、福祉関連、印鑑登録などの手続きも存在する。

3.2　電子申請のワンストップサービス（民間）

SAML により民間企業の連携を行い、引越しのための電子申請手続きを一度のログインで可能にすることで、利用者へのサービス向上を図る。プロファイルとしてアーティファクトを用いる。

3.2.1　背景

利用者は A 市内で引越しするケースを想定し、このときの電力会社 A および電話会社 A 内での変更手続きを例とする。利用者は電力会社窓口で引越しのための電子申請を行うことで、電力会社 A への利用停止手続きと利用開始手続き、電話会社 A への利用停止手続きと利用開始手続き、および、金融機関の振込み口座の住所変更手続きを行うことが可能である。

民間企業は、手続きが信頼できる企業からの申請要求かどうかを確認するために、SAML オーソリティが発行したアサーションを利用する。アサーションの署名を検証するための SAML オーソリティの証明書は、民間の認証局から発行されているものとする。SAML オーソリティは電力会社 A、電話会社 A および金融機関を PKI により認証することとする。証明書は民間の認証局から発行を受けているものとする。

3.2.2　登場人物

SAML オーソリティ	民間企業の ID を管理している。
金融機関	利用者が契約している金融機関である。金融機関は電力会社 A と電話会社 A の振込み口座を管理している。
電力会社 A	A 市の電力会社
電話会社 A	A 市の電話会社

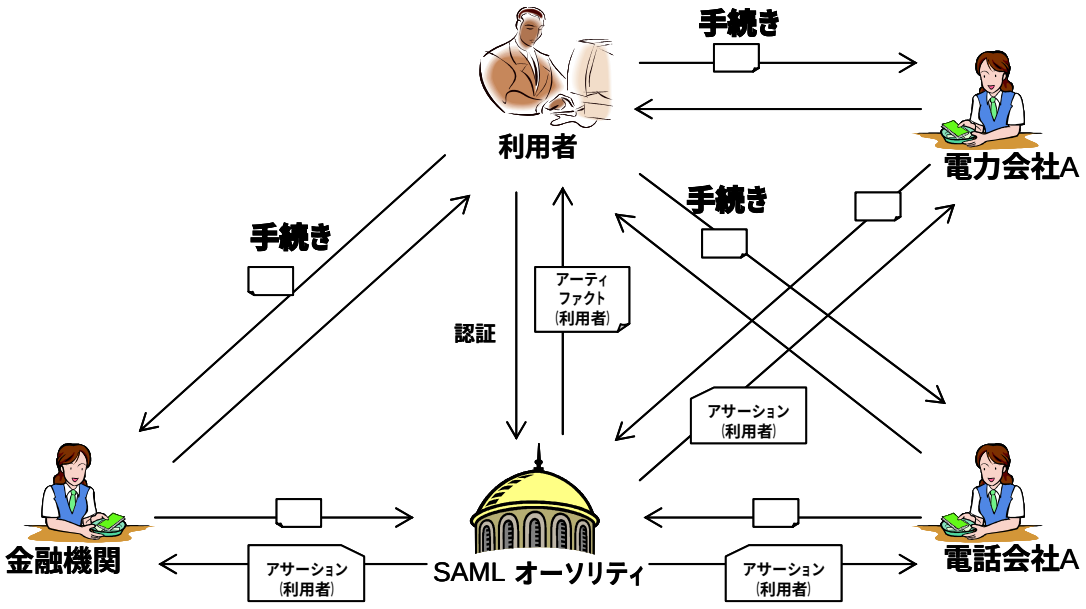


図 3-4 電子申請(民間)

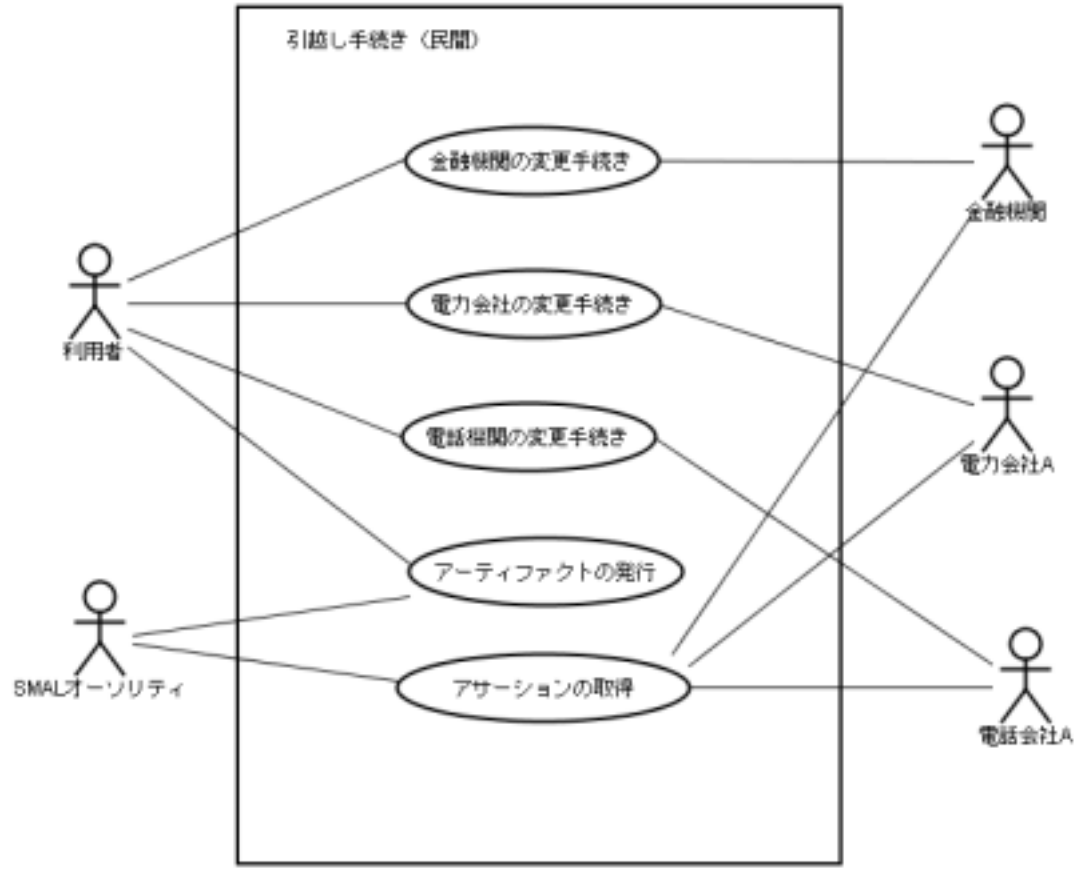


図 3-5 電子申請(民間)のユースケース図

3.2.3 シナリオ

- 手順1

  利用者は、金融機関に住所変更の手続きを申請する。金融機関 A は利用者が SAML オーソリティで認証完了していないため、利用者を SAML オーソリティにリダイレクションする。

- 手順2

  利用者は SAML オーソリティにログインする。SAML オーソリティは、利用者の認証に成功すると、アサーションとアーティファクトを発行する。そして SAML オーソリティはアサーションを内部に格納し、アーティファクトを利用者に送信する。

- 手順3

利用者は金融機関に再び住所変更の手続きを行う。

- 手順４

  金融機関は SAML オーソリティに利用者の認証情報の取得を行う。SAML オーソリティは金融機関に認証アサーションを送信する。金融機関は認証アサーションの署名を検証して SAML オーソリティが発行したことを確認する。金融機関は、利用者の自動振込み先の電力会社 A を確認し、電力会社 A に利用停止手続きと利用開始手続きを行うために電力会社 A にリダイレクションする。

- 手順５

  利用者は、電力会社 A に利用停止手続きと利用開始手続きを行う。電力会社 A は、利用者が SAML オーソリティで認証完了していないため、利用者を SAML オーソリティにリダイレクションする。

- 手順６

  利用者は SAML オーソリティにログインする。SAML オーソリティは、利用者の認証に成功すると、アサーションとアーティファクトを発行する。そして SAML オーソリティはアサーションを内部に格納し、アーティファクトを利用者に送信する。

- 手順７

  利用者は電力会社 A に再び利用停止手続きと利用開始手続きを行う。

- 手順８

  電力会社 A は SAML オーソリティに利用者の認証情報の取得を行う。電力会社 A の申請窓口は、認証アサーションの署名を検証して SAML オーソリティが発行したことを確認する。そして電力会社 A は利用停止手続きと利用開始手続きを行い、利用者に手続きの結果を返す。

- 手順９

  利用者は、利用停止手続きと利用開始手続きの結果を金融機関に送信する。金融機関は、利用者の自動振込み先の電話会社 B を確認し、電話会社 B に利用停止手続きと利用開始手続きを行うために電話会社 A にリダイレクションする

- 手順１０

  利用者は、電話会社 A に利用停止手続きと利用開始手続きを行う。電話会社 A は、利用者が SAML オーソリティで認証完了していないため、利用者を SAML オーソリティにリダイレクションする。

- 手順１１

  利用者は SAML オーソリティにログインする。SAML オーソリティは、利用者の認証に成功すると、アサーションとアーティファクトを発行する。そして SAML オーソリティはアサーションを内部に格納し、アーティファクトを利用者に送信する。

- 手順１２

  利用者は電話会社 A に再び利用停止手続きと利用開始手続きを行う。

- 手順１３

  電話会社 A は SAML オーソリティに利用者の認証情報の取得を行う。電話会社 A の申請窓口は、認証アサーションの署名を検証して SAML オーソリティが発行したことを確認する。そして電話会社 B は利用停止手続きと利用開始手続きを行う。利用者に手続きの結果を返す。

- 手順１４

  利用者は、利用停止手続きと利用開始手続きの結果を金融機関に送信する。金融機関は、住所変更手続きを行い利用者に手続きの結果を返す。

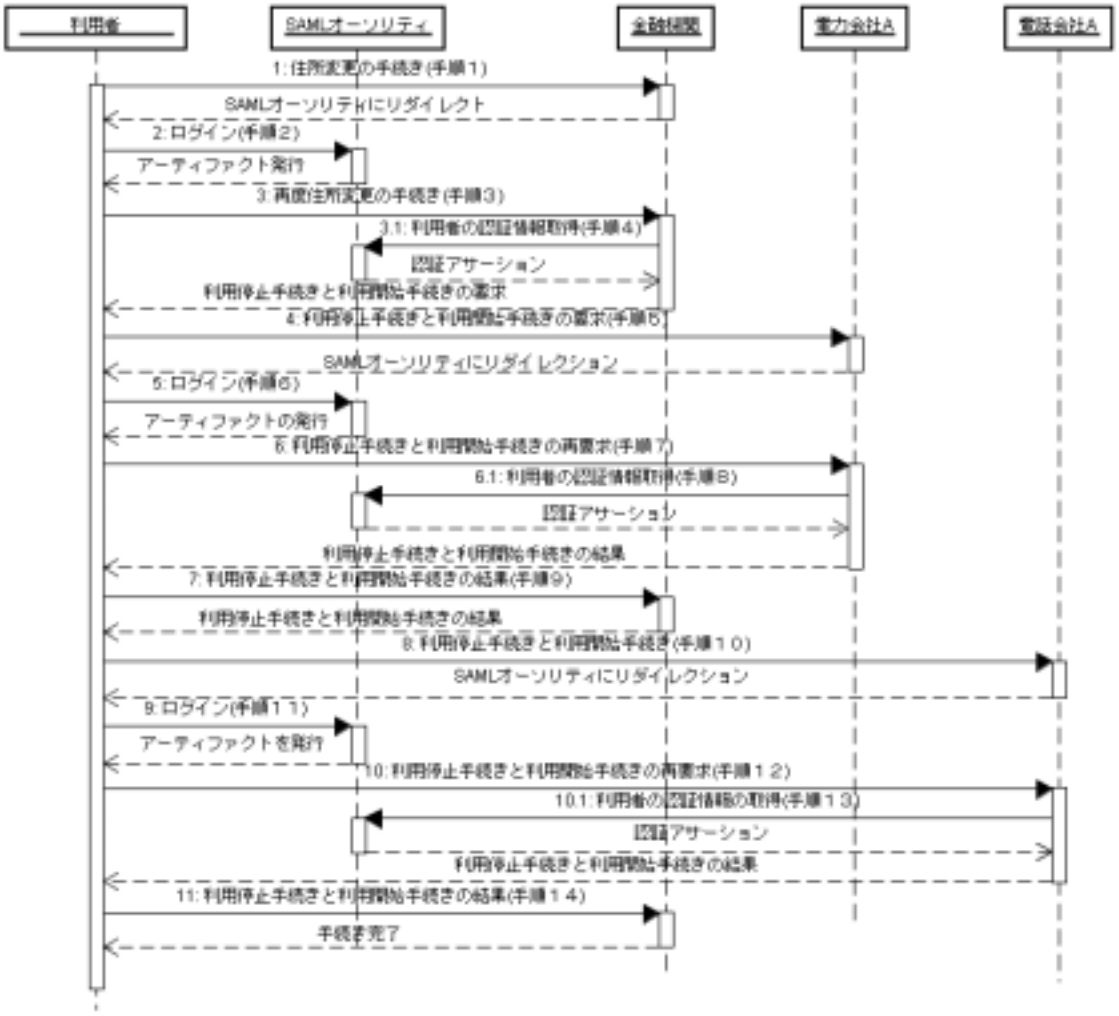


図 3-6 電子申請 (民間) のシーケンス図

3.2.4 特徴

利用者は、SAML オーソリティにログインする前に金融機関からアクセスを行う場合の例である。金融機関は事業者が SAML オーソリティで認証完了していないため、SAML オーソリティにリダイレクションする。利用者が SAML オーソリティや金融機関、電力会社および電話会社へのアクセスを意識することなく手続きを進めることが可能となる。

3.2.5 考察

本ユースケースにおいて、金融機関、電力会社、電話会社の関係は、電子申請のワンストップサービス（行政）の市役所、教育委員会のように相互に手続き業務が密接に関連しているわけでない。手続き業務として関係はあるが独立であるため、アーティファクトやアサーションをキャッシュしないモデルである。本モデルでは他の組織にリダイレクショ

ンするごとに SAML オーソリティがアーティファクトとアサーションを発行する。そのため SAML オーソリティに負荷がかかることが考えられるためにネットワークや機器選定が必要となる。

本ユースケースでは、利用者が金融機関、電力会社、電話会社のどこからでも申請手続きが可能であり、申請後に利用者に意識させることなく関連手続きにリダイレクションする。

3.3　インターネットオークションの起業申請

SAML により行政機関で連携を行い、起業のおける電子申請を一度のログインで可能にすることで、事業者へのサービス向上を図る。プロファイルとしてアーティファクトを用いる。

3.3.1　背景

＜対象手続の概要＞

根拠法令：古物営業法第十条の二

手続内容：古物競りあっせん業の営業開始届出

添付書類：定款、登記簿謄本、ホームページのＵＲＬに関する資料など。

※　　法律上、申請と届出は異なるものであるが（許認可等の申請と異なり、届出は、形式的要件を満たしたデータが行政側のシステムに到達した時点で手続完了となる）、ここでは「申請」の用語で統一することとする。

事業者（法人）はインターネットオークションサービス（古物競りあっせん業）を行う場合、営業開始の日から２週間以内に、営業所所在地を管轄する公安委員会（所在地を管轄する警察署を経由して行う）に申請手続を行う。申請を受付けた警察署は、法務局（オンライン登記情報提供制度）に対して事業者の登記情報を確認する。登記情報を確認した警察署は、事業者に対して手続完了の通知を行う。

事業者の認証に使用する証明書は、商業登記認証局から発行されているものとし、SAML オーソリティのアサーションの署名検証の証明書は、民間の認証局から発行されているものとする。

3.3.2　登場人物

SAML オーソリティ	事業者と行政機関の ID を管理している。
事業者	インターネットオークションのサービスを開始する。
警察署	インターネットオークションの申請手続を受付ける。
法務局	インターネットオークションの申請手続に必要な登記簿謄本を管理している。

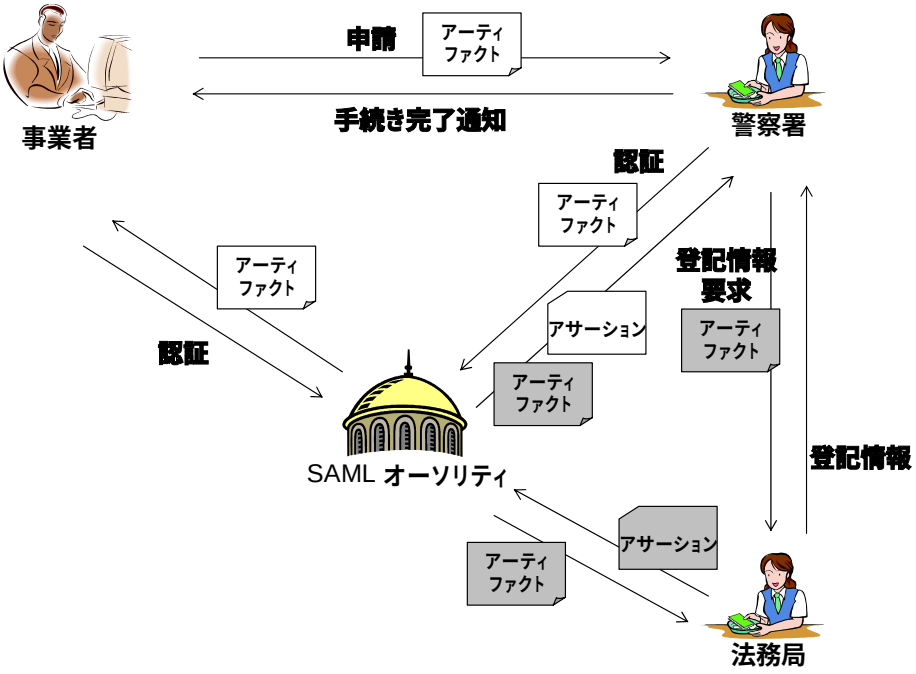


図 3-7 インターネットオークションの起業申請

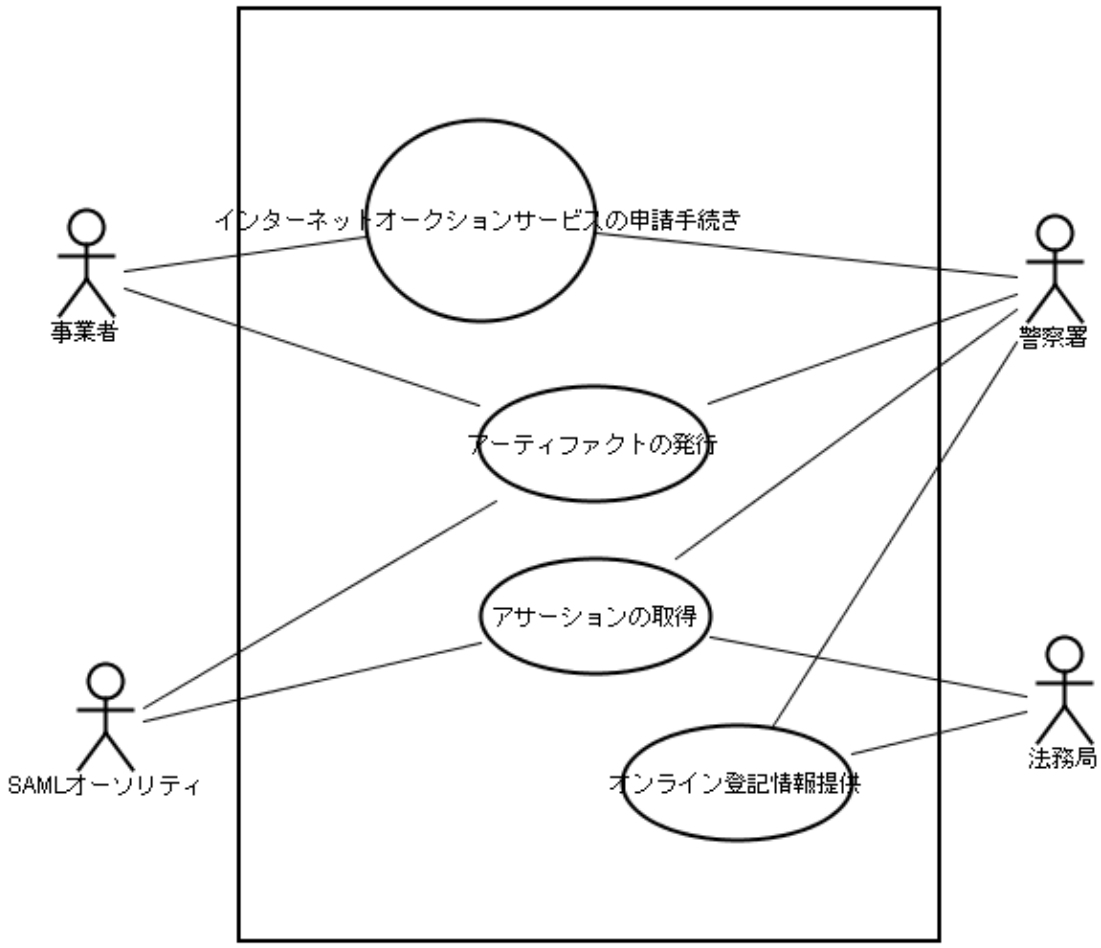


図 3-8 インターネットオークションの起業申請のユースケース図

3.3.3 シナリオ

- 手順1

  事業者は警察署にインターネットオークションのサービスを開始するための申請手続きを行う。警察署は事業者が SAML オーソリティで認証完了していないため、事業者を SAML オーソリティにリダイレクションする。

- 手順2

  事業者は SAML オーソリティに認証を要求する。SAML オーソリティは、事業者の認証に成功すると、アサーションとアーティファクトを発行する。そして SAML オーソリティはアサーションを内部に格納し、アーティファクトを事業者に送信する。

- 手順3

  事業者は警察署に再び申請を行う。このとき事業者はアーティファクトを警察署に送信

する。

- 手順４

  警察署はアーティファクトによって SAML オーソリティに事業者の認証情報を要求する。SAML オーソリティは警察署に認証アサーションを送信する。警察署は、認証アサーションの署名を検証して SAML オーソリティが発行したことを確認する。

- 手順５

  警察署は SAML オーソリティに認証を要求する。SAML オーソリティは、警察署の認証に成功すると、アサーションとアーティファクトを発行する。そして SAML オーソリティはアサーションを内部に格納し、アーティファクトを事業者に送信する。

- 手順６

  警察署は法務局に登記情報の提供を要求する。このとき警察署はアーティファクトを送信する。

- 手順７

  法務局はアーティファクトによって SAML オーソリティに事業者の認証情報を要求する。SAML オーソリティは警察署に認証アサーションを送信する。法務局は、認証アサーションの署名を検証して SAML オーソリティが発行したことを確認する。確認後、法務局は警察署に事業者の登記情報を提供する。警察署はインターネットオークションサービスの手続きを行い、申請処理が完了すると事業者にその結果を返す。

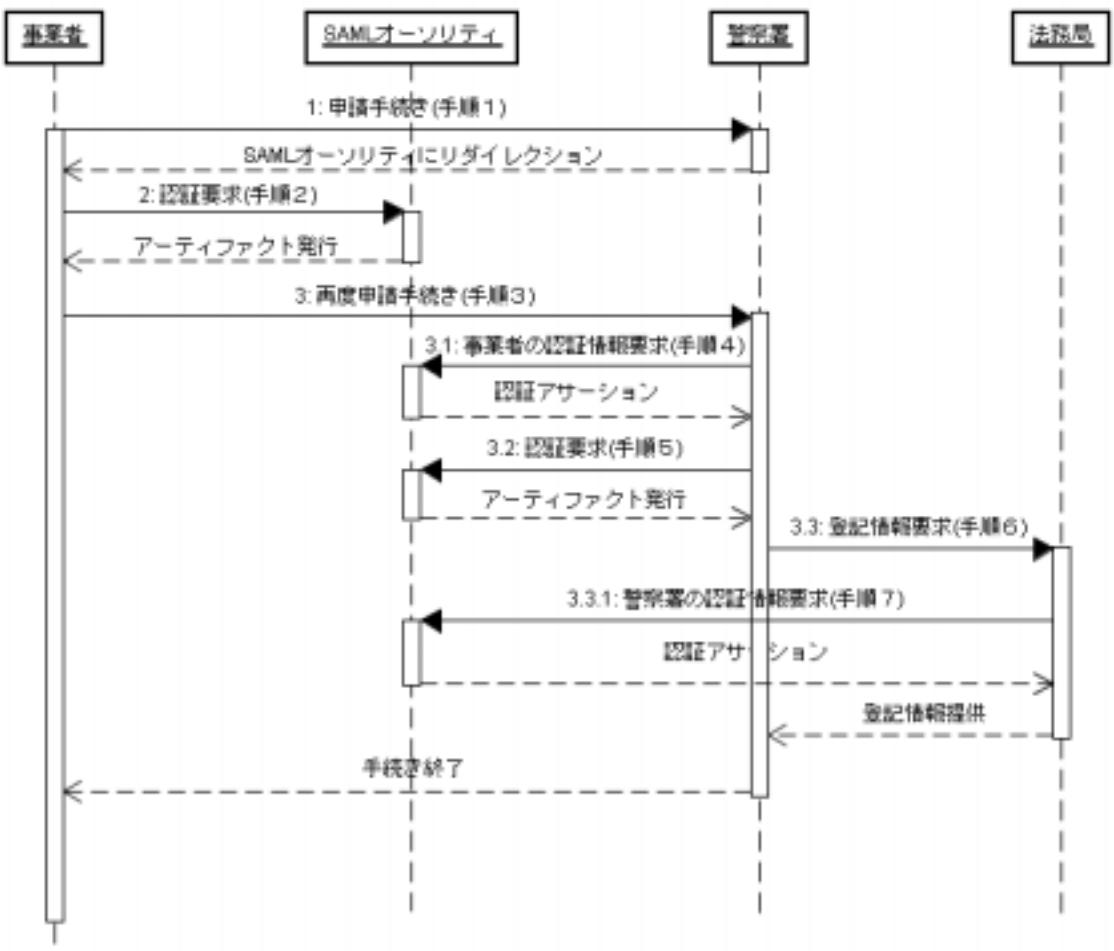


図 3-9 インターネットオークションの起業申請のシーケンス図

3.3.4 特徴

事業者は行政機関にアクセスして行政機関が信頼する SAML オーソリティに事業者をリダイレクションする。そして行政機関は申請手続きを受けると手続きに関連する他の行政機関から情報を取得することで申請手続きを処理する。

事業者は申請手続きを行う行政機関のみ意識し、SAML オーソリティや関連する他の行政機関を意識する必要がない。

3.3.5 考察

本ユースケースは、電子申請のワンストップサービス（行政）と比較して、手続き業務が比較的独立な場合のモデルである。そのため各行政機関で必要な行政機関のみに情報を提供できるように、アーティファクトやアサーションを再利用して情報漏洩を防止している。手続き業務がそれほど密接に関連しない場合に本モデルを検討してみるとよいと思わ

れる。

3.4 電子カルテ

電子カルテサーバに保管された電子カルテのアクセス制御を行う。通常は電子カルテサーバを運用する病院の医師にだけ電子カルテへのアクセスを許可するが、非常時には他の病院の医師に対して電子カルテへのアクセスを許可できるようにする。

3.4.1 背景

生活習慣病などで医師の治療を受け、継続的に処方された薬を服用しているような患者が外出先で倒れたような場合、通常倒れた場所の近くの救急病院などに運ばれることになるが、救急病院で応急処置を行うに際して、患者が日ごろ服用している薬の種類や量といった情報が必要な場合がある。

本ユースケースは、救急病院の当直医が、他の病院の電子カルテシステムに蓄積されている電子カルテに、救急病院の当直医であるという属性を提示することによってアクセスする仕組みをＳＡＭＬを利用して提供する。

3.4.2 登場人物

患者	診療の対象
病院Aに所属する医師 a	患者の主治医。PKI 認証局から公開鍵証明書を発行されている。公開鍵証明書の subjectDirectorAttribute エクステンションに、ISO TS 17090 で定義されている hcRole ｱﾄﾘﾋﾞｭｰﾄで、医師資格があることを示す属性が設定されている[医療 PKIGL]。
病院 B に所属する医師 b	救急病院 B の当直医。PKI 認証局から公開鍵証明書を発行されている。公開鍵証明書の subjectDirectorAttribute エクステンションには、ISO TS 17090 で定義されている hcRole ｱﾄﾘﾋﾞｭｰﾄで、医師資格があることを示す属性が設定されている [医療 PKIGL]。
病院毎の SAML オーソリティ	自病院に所属する医師に対して SAML アサーションを発行する。PKI 認証局からアサーションに署名するための公開鍵証明書を発行されている。
病院毎の電子カルテサーバ	自病院における診療記録（電子カルテ）を保管するサーバ。電子カルテへのアクセス要求に対して、添付された SAML アサーションを使用してアクセスの可否を判断する SAML の policy decision points の機能を兼ねる。ただし、自分自身が保持するデータに対してアクセス可否を判断しているため、実際に認可決定アサーションの発行を行うことはしない。
公的機関もしくは MEDIS や医師会などが運営する PKI 認証局	医師や SAML オーソリティに公開鍵証明書を発行する認証局。

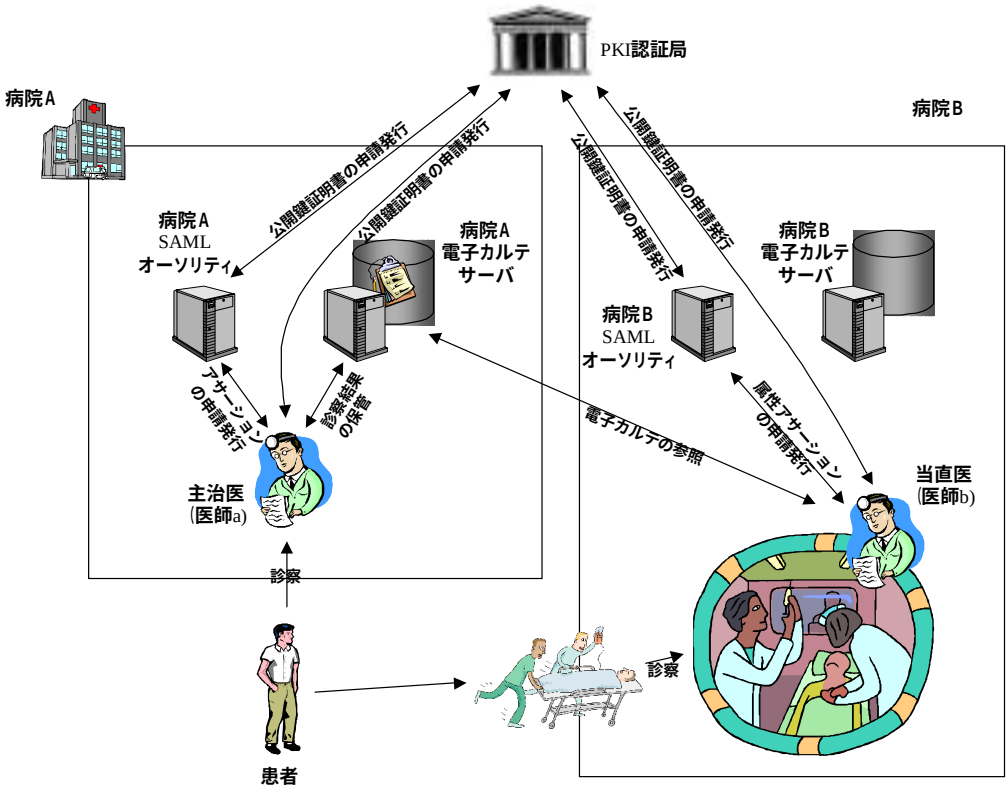


図 3-10 電子カルテ構成図

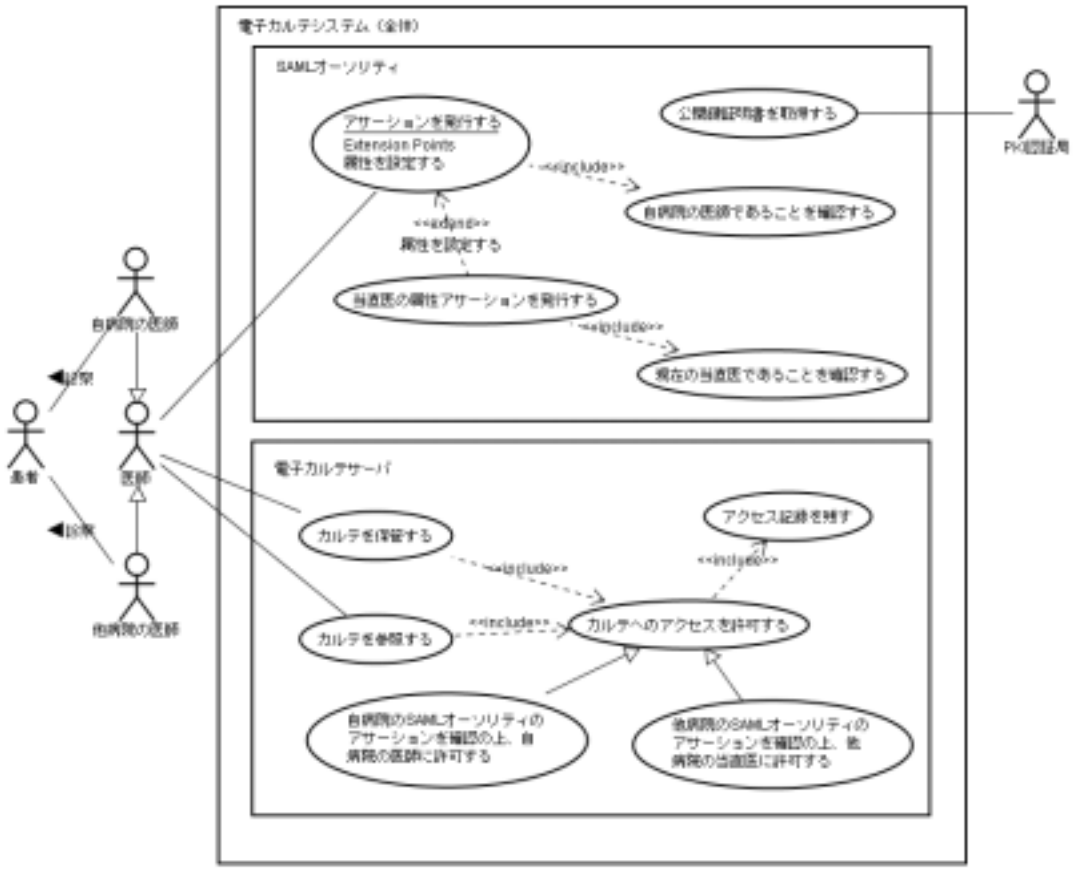


図 3-11 電子カルテのユースケース図

3.4.3 シナリオ

(前提)

各病院間では、救急病院の当直医に対しては電子カルテの内容をお互いに開示することが合意されている。他病院の当直医にカルテの内容を開示した場合には、事後に患者に対してその旨が通知されるようにすることによって、当直医による不正な電子カルテへのアクセスが牽制される仕組みになっている。

- 手順１

  医師 a は PKI 認証局に公開鍵証明書の発行を申請し発行される。公開鍵証明書の subjectDirectorAttribute エクステンションには医師資格があることを示す hcRole アトリビュートが設定されている。

- 手順２

医師 b は PKI 認証局に公開鍵証明書の発行を申請し発行される。公開鍵証明書の subjectDirectorAttribute エクステンションには医師資格があることを示す hcRole アトリビュートが設定されている。

- 手順３

  病院 A の SAML オーソリティは PKI 認証局に公開鍵証明書の発行を申請し発行される。

- 手順４

  病院 B の SAML オーソリティは PKI 認証局に公開鍵証明書の発行を申請し発行される。

- 手順５

  患者は主治医である医師 a の診察を受ける。

- 手順６

  医師 a は患者のカルテにアクセスするため、病院 A の SAML オーソリティに、医師 a が病院 A に所属する医師であることを示すアサーションの発行を要求し、発行される。

- 手順７

  医師 a は、病院 A の電子カルテサーバに対して、病院 A の SAML オーソリティに発行されたアサーションを提示して、患者のカルテに診察結果を保管する。電子カルテサーバは医師ａが病院 A の医師であることをアサーションによって確認し、電子カルテへの診察結果の保管を許可する。

- 手順８

  出先で倒れた患者が病院 B に運ばれ、医師ｂの診察を受ける。

- 手順９

  患者が持っていた診察券や健康保険証カードなどから病院 A で診療を受けていたことがわかり、患者の症状から患者の病歴などを確認する必要があると考えた医師ｂは、病院Ｂの SAML オーソリティに、医師 b が病院 B の当直医であることを示す属性アサーションの発行を要請し、アサーションを発行される。

- 手順１０

  医師ｂは、病院 A の電子カルテサーバに対して、病院 B の SAML オーソリティが発行した属性アサーションを提示し、健康保険証番号と氏名を指定して患者の電子カルテへの参照を要求する。病院Ａの電子カルテシステムは、アサーションの署名によって、アサーションが確かに病院Ｂの SAML オーソリティが発行した属性アサーションであることを確認し、さらに医師 b の証明書で医師ｂが確かに医師の資格を持っていることを

確認した結果、病院Ｂの当直医が患者の電子カルテに対するアクセスを求めていることから、患者に緊急事態が発生しているものと判断して電子カルテに対するアクセスを許可する。病院Ａの電子カルテシステムは、病院Ｂの医師 b に対してカルテの参照を許可したことを、医師 b が提示したアサーションと共にログとして記録する。

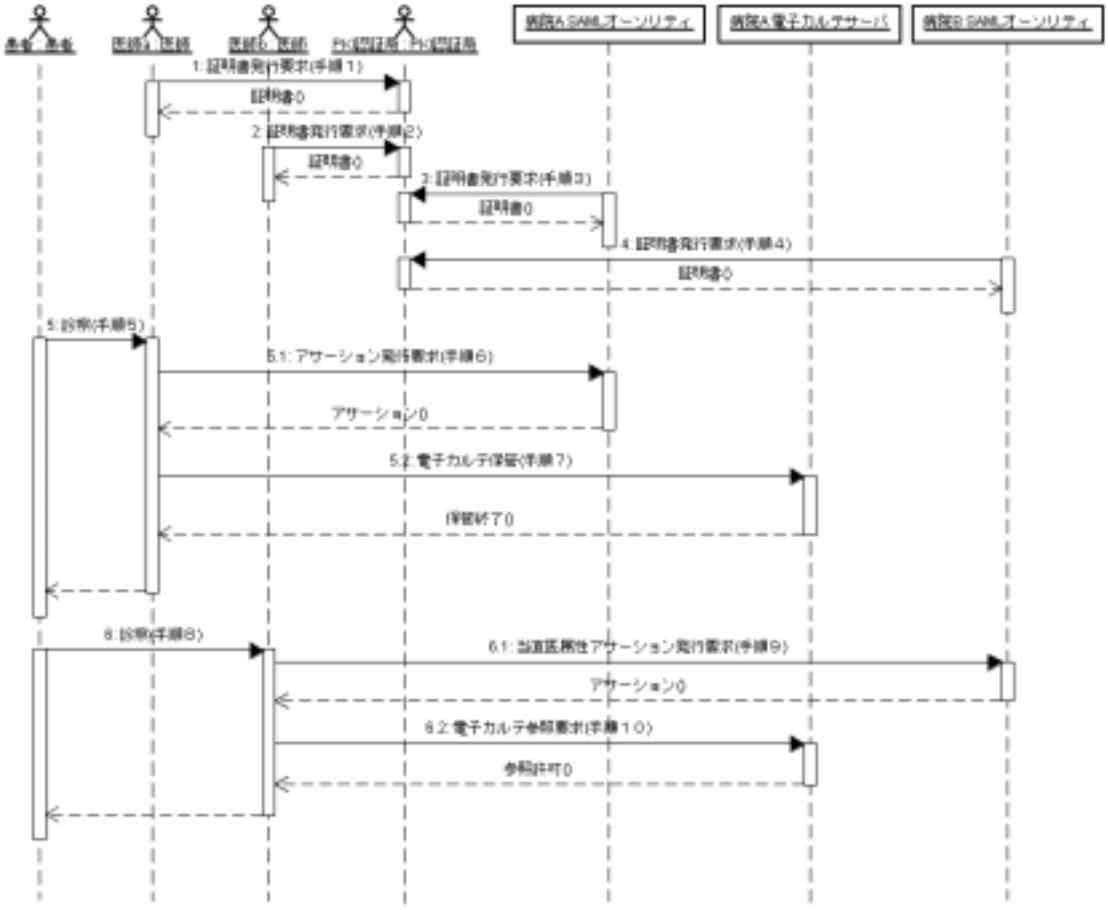


図 3-12 電子カルテのシーケンス図

3.4.4 特徴

あらかじめ病院間で取り決めておくことにより、他の病院の当直医に対して電子カルテへのアクセスを許可する。当直医であることを示すアサーションの有効期間を短くしておくことにより、当直を解かれた医師に対して不用意に電子カルテへのアクセスを許可してしまうことを防ぐ。また、電子カルテサーバ側ではアクセス許可をした場合に、許可の根拠となったアサーションを保管しておくことにより、誰に対して電子カルテへのアクセスを許可したのかを示すエビデンスとすることができる。

3.4.5　考察

緊急時に他病院の電子カルテの内容が参照できれば、当直医だけでなく患者にとってもメリットがあると考える。当直医だけでなく、患者の許可の下で複数の病院間で電子カルテの共有ができれば（内科と耳鼻科など）、さらにメリットがあるが、電子カルテへの参照を患者が許可した、ということをシステム的に確認する方法についてさらに検討する必要がある。

このユースケースにおいて考えられる脅威とその対策としては以下のものがあげられる。

（脅威１）

病院 A の主治医（医師 a）以外の医師が患者の不必要にカルテを覗いてしまう。

（対策１）

患者に対して主治医（主治医に対して患者）を結びつける属性情報を新たに増やし、属性認証を行う。通常時、主治医以外にカルテを参照する必要がある場合は電子紹介状を活用し、不必要にアクセスができないようにする。

（脅威２）

本システムに参加するすべての病院の当直医が、オーソリティを悪用すれば誰のカルテでも見ることができてしまう。

（対策 2-①）当直医であることを示すアサーションの有効期間を短く設定しておく

（対策 2-②）運用方法(単独または組み合わせ)により牽制を行う。

例1）主治医以外で、各病院の当直医にカルテの内容が開示された場合、事後に患者に対してその旨を通知する。

例2）アクセスログを監査証跡として、各医療機関（及び患者）に開示を行う。

3.5 電子紹介状

病院間での患者の紹介を電子紹介状を用いて実現する。電子紹介状サーバが一括して電子紹介状を保管し、電子紹介状へのアクセスは SAML を利用して医師資格と医師が勤務している病院名という属性を確認することで制御する。これにより、正当な権限を持たない者による、電子紹介状の作成・参照を防ぐ。

3.5.1 背景

特定療養費制度により、大学病院等の床数の多い病院への紹介状の発行数が増加していく傾向にある。紙による紹介状の発行に比較して、電子紹介状ではデータの受け渡しが簡単になり、電子カルテやレントゲン写真等を添付することも容易になる。

電子紹介状の発行は患者が許可した正当な医師によりなされる必要がある。また、電子紹介状には個人情報が含まれるため、患者は紹介された病院の医師以外には電子紹介状を参照されたくない。

本ユースケースでは、電子紹介状は医師会や MEDIS[2]などが管理する電子紹介状サーバに安全に保管されることを想定し、電子紹介状の作成・参照を医師資格と医師の勤務する病院名という属性を確認することによって許可する仕組みを SAML を利用して実現する。

3.5.2 登場人物

患者	他の病院への電子紹介状を発行される者
病院 A、医師 a	かかりつけの病院（診療所など）と、その病院に所属する医師。医師には PKI 認証局から subjectDirectoryAttributes エクステンションに、ISO TS 17090 で定義されている hcRole アトリビュートで医師資格があることを示す属性が設定された鍵証明書が発行されている[医療 PKIGL]。
病院 B、医師 b	大きな病院（大学病院など）と、その病院に所属する医師。医師には PKI 認証局から subjectDirectoryAttributes エクステンションに、ISO TS17090 で定義されている hcRole アトリビュートで、医師資格があることを示す公開鍵証明書が発行されている[医療 PKIGL]。
SAML オーソリティ	医師の勤務する病院名や専門とする科などの属性情報を管理している。SAML Authentication オーソリティと SAML Attribute オーソリティの役割を果たす。PKI 認証局により公開鍵証明書が発行されている。
電子紹介状サーバ	電子紹介状 DB を管理する。電子紹介状の作成･参照を要求された際に、SAML アサーションにより電子紹介状 DB へのアクセス権

[2] MEDIS 財団法人 医療情報システム開発センター http://www.medis.or.jp/

	限を判定する SAML の policy decision points の機能を兼ねる。ただし、自分自身が保持するデータに対してアクセス可否を判断しているため、実際に認可決定アサーションの発行を行うことはしない。
PKI 認証局	公的機関もしくは医師会･MEDIS などが運営する PKI 認証局。医師と SAML オーソリティに対して公開鍵証明書を発行する。

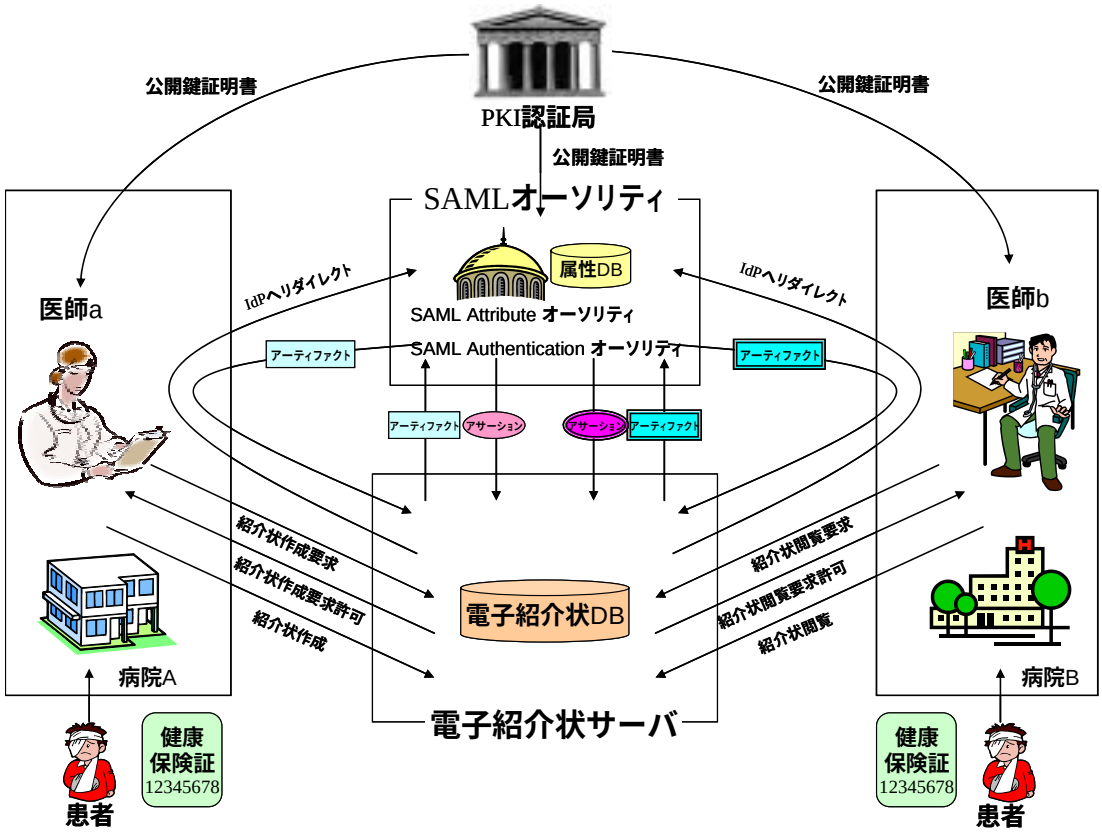


図 3-13 電子紹介状サービス

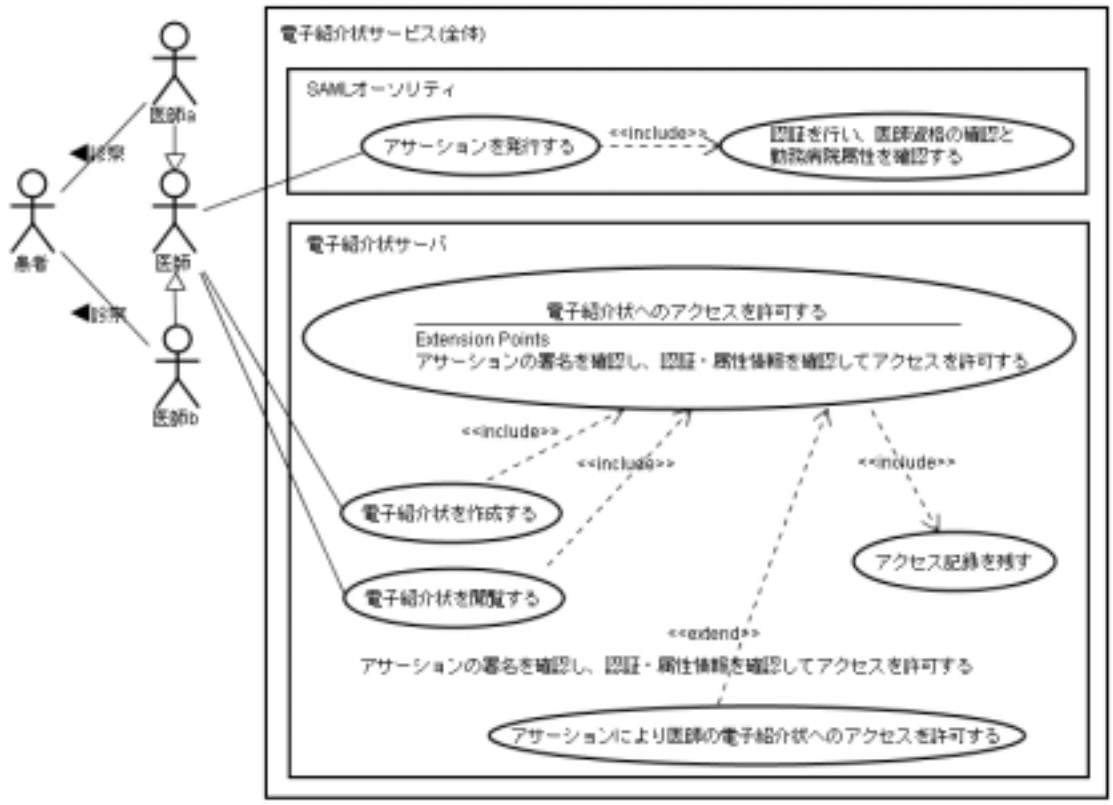


図 3-14 電子紹介状サービスのユースケース図

3.5.3 シナリオ

医師 a が電子紹介状を作成

（前提）

患者が病院 A で医師 a の診断を受けたところ、他の病院に紹介する必要がでてきた。

医師 a は患者を病院 B に紹介することにし、患者からの同意を得ている。

- 手順１

  医師 a は電子紹介状システムにアクセスし、電子紹介状の作成を要求する。電子紹介状システムは、医師 a を、医師 a のアイデンティティを管理している SAML オーソリティへリダイレクトする。

- 手順２

  医師 a は SAML オーソリティにリダイレクトされ、SAML オーソリティは医師 a の公開鍵証明書（医師資格は subjectDirectoryAttributes＝hcRole で表示）により、医師 a

の本人認証と、医師資格の検証を行う。さらに、属性 DB で"医師 a が確かに病院 A に勤務している"ことを確認し、AuthenticationStatement と AttributeStatement を含むアサーションを作成する。SAML オーソリティは医師 a にそのアサーションに対応付けられる SAML アーティファクトを発行し、医師 a を再び電子紹介状サーバへリダイレクトする。

- 手順３

  医師 a は電子紹介状サーバにリダイレクトされ、電子紹介状サーバは SAML オーソリティに対して SAML アーティファクトを提出して、医師 a の認証情報、属性情報を要求する。SAML オーソリティは、アサーション（AuthenticationStatement、AttributeStatement）を含む応答を電子紹介状サーバに返す。電子紹介状サーバは、SAML オーソリティから受けとったアサーションの署名を検証し、アサーションの内容により医師 a が確かに医師資格を持ち病院 A に勤務していることを確かめて、医師 a に電子紹介状の作成を許可する。

- 手順４

  医師 a は電子紹介状を作成し、電子紹介状 DB に病院 B の医師だけがアクセスできるように、アクセス権限を設定して登録する。その際、患者の健康保険被保険者証番号も合わせて登録しておく。

医師 b が電子紹介状を閲覧

（前提）

後日、患者が病院 B を訪れる。病院 B の医師ｂは、患者から病院Ａからの電子紹介状があることを伝えられる。

- 手順５

  医師 b は電子紹介状システムにアクセスし、患者の氏名と健康保険被保険者証の番号を入力して、患者の電子紹介状の閲覧を要求する。電子紹介状システムは、医師 b を、医師 b のアイデンティティを管理している SAML オーソリティへリダイレクトする。

- 手順６

  医師 b は SAML オーソリティにリダイレクトされ、SAML オーソリティは医師 b の公開鍵証明書（医師資格は subjectDirectoryAttributes＝hcRole で表示）により、医師 b

の本人認証と、医師資格の検証を行う。さらに、属性 DB で"医師 b が確かに病院 B に勤務している"ことを確認し、AuthenticationStatement と AttributeStatement を含むアサーションを作成する。SAML オーソリティは医師 b にそのアサーションに対応付けられる SAML アーティファクトを発行し、医師 b を再び電子紹介状サーバへリダイレクトする。

- 手順7

  医師 b は電子紹介状サーバにリダイレクトされ、電子紹介状サーバは、SAML オーソリティに対して、SAML アーティファクトを提出して、医師 b の認証情報、属性情報を要求する。SAML オーソリティは、アサーション（AuthenticationStatement、AttributeStatement）を含む応答を電子紹介状サーバに返す。電子紹介状サーバは、SAML オーソリティから受けとったアサーションの署名を検証し、アサーションの内容により医師 b に患者の電子紹介状の閲覧を許可する。

- 手順8

  医師 b は電子紹介状を閲覧する。

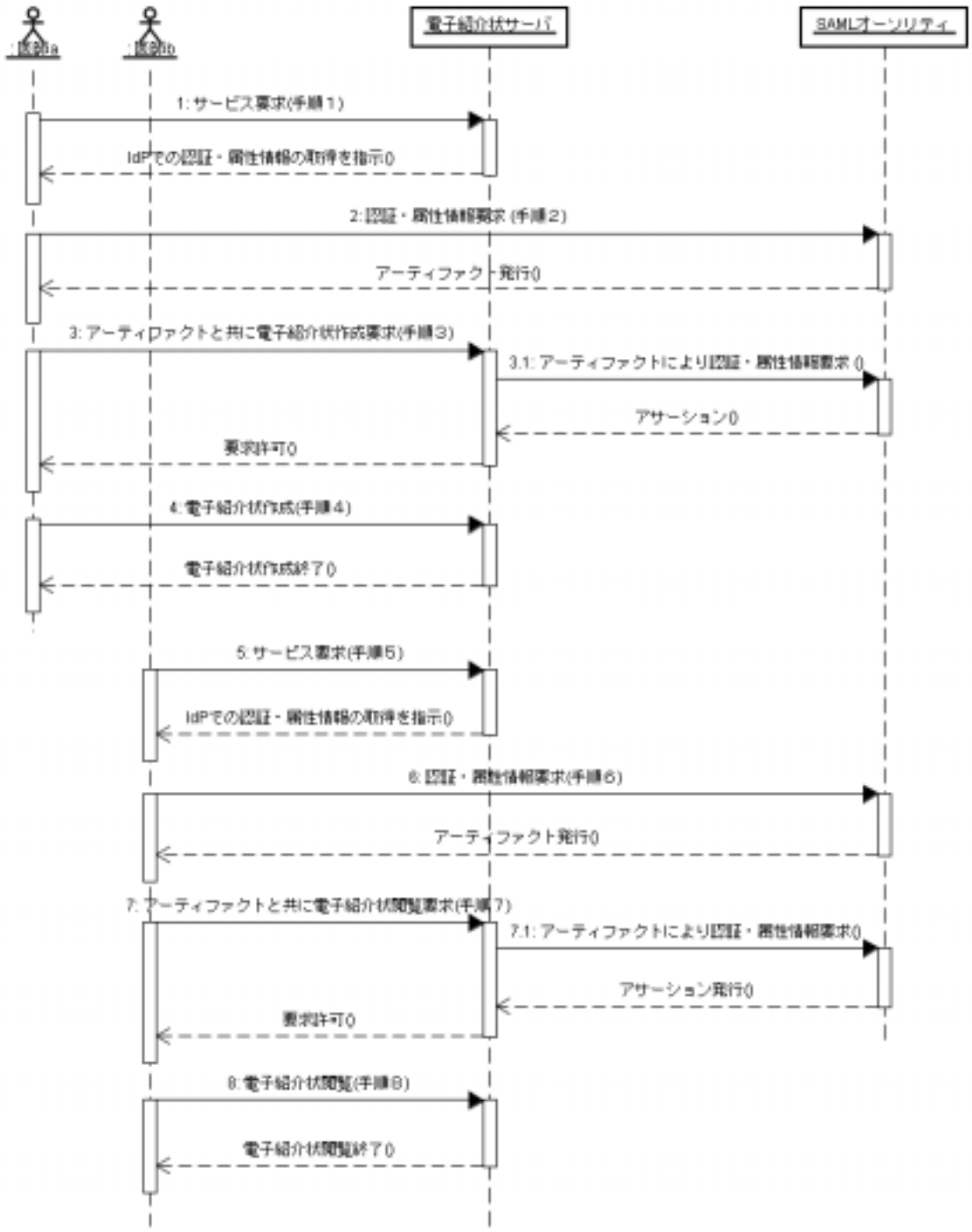


図 3-15 電子紹介状サービスのシーケンス図

3.5.4 特徴

SAML を利用したアクセス制御により、電子紹介状サービスへの不正アクセスを防ぎ, 個人情報漏えいによるリスクを回避することができる。

また、各病院の医師の属性情報の管理を一つの SAML オーソリティが一括して行うことにより、各病院毎に SAML オーソリティを持つ必要がない。また、電子紹介状も一つの電子紹介状サーバが一括して管理するため、各病院でサーバを運営・管理する必要がない。よって、病院側に負担がかからないサービスを実現できる。

3.5.5 考察

電子紹介状サーバに電子紹介状を保管することで、システム上に記録を残すことができ、署名機能やタイムスタンプ機能を用いることで、電子紹介状を証拠として用いることも可能となると考えられる。さらに、電子紹介状のテンプレートを用意しておくことで、容易に作成でき、読みやすい電子紹介状を実現できる。

また、電子カルテサービスと連携させると、電子カルテからコピーして貼り付けるといったことを行うことができるようになり利便性が高くなるため、統合的な医療情報の電子化管理を検討していく必要がある。

このユースケースにおいて考えられる脅威とその対策としては以下のものがあげられる。

（脅威１）医師 b が紹介状データベースに不正にアクセスし、病院 B 向けの紹介状で医師 b が担当した患者以外の紹介状を覗き見てしまう。

（対策１）紹介された患者の何らかのアクション（承認）が無ければ、該当するカルテにはアクセスできないような仕組みを導入する。

3.6 在宅介護における病院と自治体の連携

本ユースケースは、医師が、自治体が管轄する在宅介護連絡システムと病院が管轄する電子カルテシステムとを、シングルサインオンで利用可能にする仕組みを提供するものである。シングルサインオンには SAML を利用する。

3.6.1 背景

介護保険制度を導入しているある自治体が、介護保険利用者の状況を把握するため、在宅介護連絡システムを導入しているものと仮定する。在宅介護連絡システムとは、自治体と契約して患者の在宅介護を行なうホームヘルパー等が、日々の患者の健康状態や介護状況を報告する(書き込む)ものとする。また、介護保険利用者のかかりつけの病院は、患者の病状や経過、処置、処方箋等の医療情報を記した電子カルテを電子カルテシステムに保管しているものと仮定する。これらのカルテは、その病院の医療従事者が参照できるものとする。

介護保険利用者(患者)の主治医である医師は、患者が外来に来たときに診察しているが、患者の日常の健康状況等をより頻繁にチェックすることで、より良い処方ができるため、ヘルパーによる患者の日々の介護報告を参照しようとする。

電子カルテシステムも、在宅介護連絡システムも、個人情報を扱っているため、権限のある人にしか参照を許可しない仕組みをとっているが、医師はそれぞれに別のアカウントでログインし直さずに、両方のシステムに 1 度のログインでアクセスできるようにしたい。

3.6.2 登場人物

患者 A	介護保険利用者
医師 B	病院 H の勤務医で、患者 A の主治医
ホームヘルパーC	自治体との契約で、患者 A の在宅介護を実際に担当する者。日々の介護状況を自治体の在宅介護連絡システムに書き込んでいる。
病院 H の電子カルテシステム	自病院における診療記録(電子カルテ)を保管するサーバ。
自治体 G の在宅介護連絡システム	ホームヘルパー等による、介護保険利用者の健康状態や介護状況の報告を管理している。医師による SAML アサーションを使用した介護情報へのアクセス要求に対して、アクセスの可否を判断する SAML の policy decision points の機能を兼ねる。ただし、自分自身が保持するデータに対してアクセス可否を判断しているため、実際に認可決定アサーションの発行を行うことはしない。

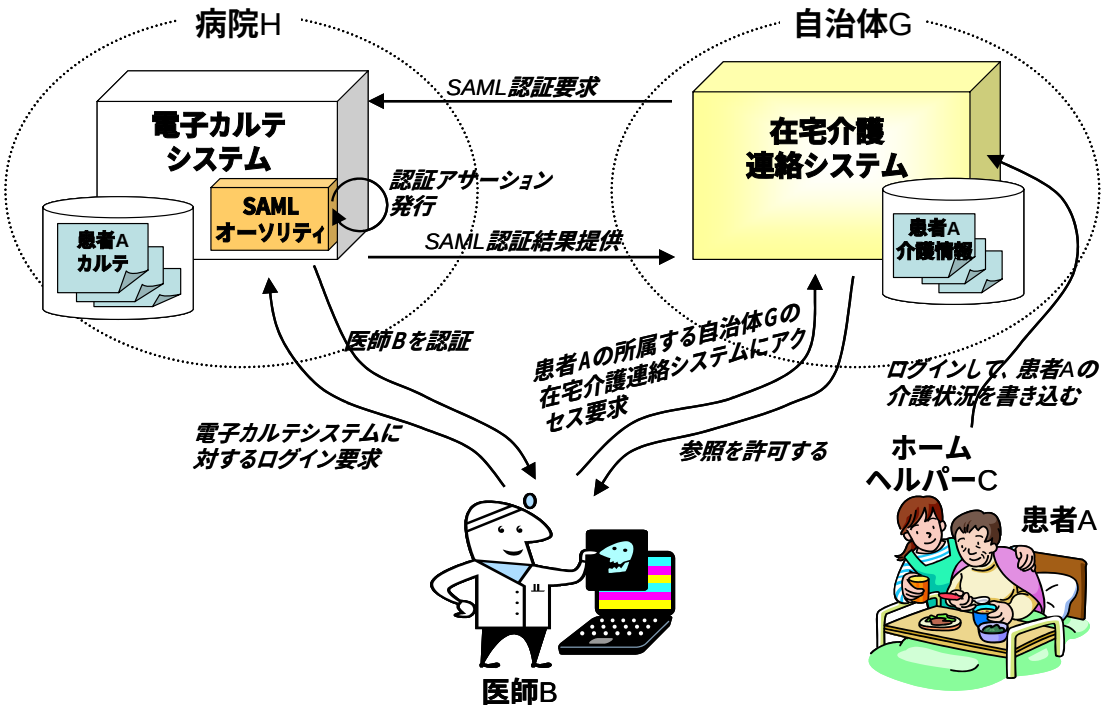


図 3-16 在宅介護連絡システム

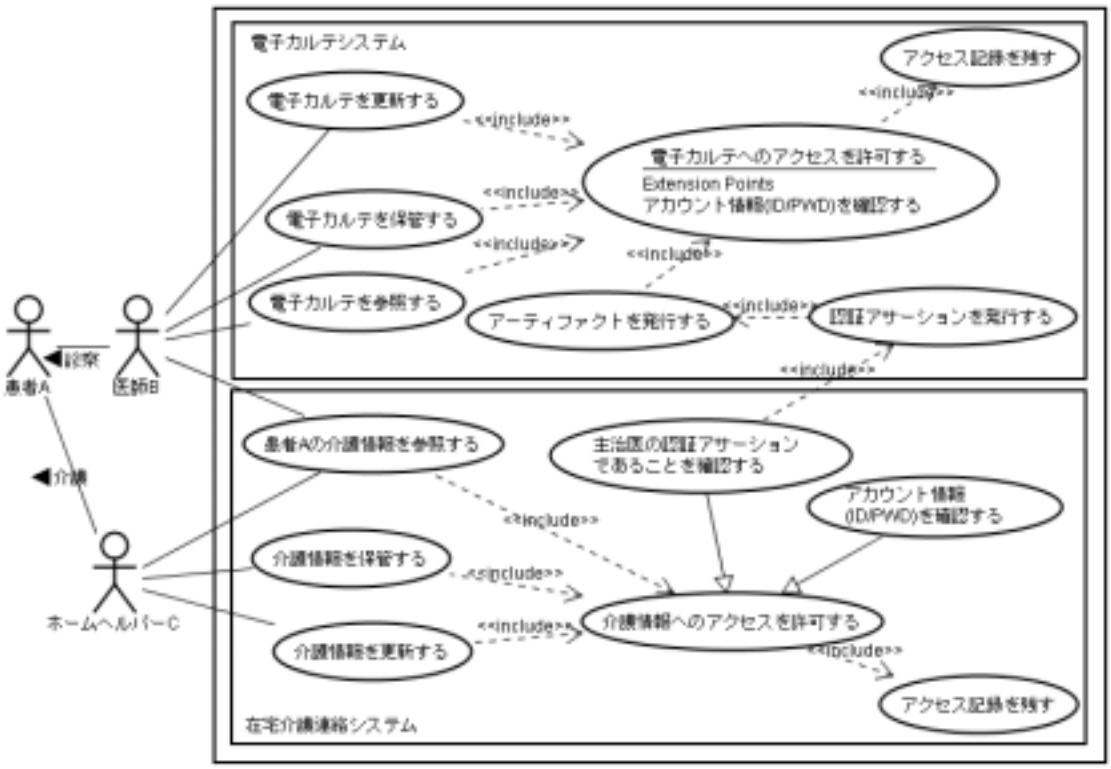


図 3-17 在宅介護連絡システムのユースケース図

3.6.3 シナリオ

（前提および事前準備）

■病院 H の電子カルテサーバに関する前提と事前準備

- 病院 H の電子カルテシステムの SAML オーソリティは、医師の認証アサーションに署名するための公開鍵証明書を信頼できる認証局から発行されている。
- 患者 A のカルテは、病院 H の電子カルテシステムに保管されている。
- 医師 B は、勤務先の病院 H から、電子カルテシステムにアクセスするためのアカウント(ID とパスワード)を付与されている。
- 病院 H の電子カルテシステムは、通院中または退院後の患者が、自治体の介護保険制度を利用して在宅介護を受けている場合、患者と所属自治体の対応付けを管理している。医師の認証を行なうと、その医師専用のページを表示し、ログイン後のページには、その医師がアクセス可能な SSO 先へのリンクが動的に表示されるものとする。
- SSO 先のリンク URL には、認証時に電子カルテシステムの SAML オーソリティが発行したアーティファクトが埋め込まれているものとする。

■自治体 G の在宅介護連絡システムに関する前提と事前準備

- ホームヘルパーC は、契約先の自治体 G から、在宅介護連絡システムにアクセスするためのアカウント(ID とパスワード)を付与されている。
- 在宅介護連絡システムは、ホームヘルパーC が患者 A の担当者であることを事前に認識しており、ホームヘルパーC の認証を行なうと、そのヘルパーの担当患者 A の介護情報のページを表示し、ヘルパーC による参照、更新、保管を許可している。
- 在宅介護連絡システムは、医師 B が患者 A の主治医であることを事前に認識しており、医師 B であることが確認できれば、在宅介護連絡システムの患者 A の領域へのアクセス（参照）を許可するものとする。
- 病院 H と自治体 G は、両者間におけるシングルサインオンの実現について合意しているものとする。

（シナリオ）

- 手順１

  ホームヘルパーC は、ID とパスワードで自治体 G の在宅介護連絡システムにログイン

する。在宅介護連絡システムは、ホームヘルパーC を認証し、そのヘルパー用の画面を提示する。

- 手順２

  ホームヘルパーC は、患者 A の介護情報の参照、更新、保管を行なう。

- 手順３

  医師Ｂは、自宅 PC から、病院 H の電子カルテシステムにアクセスし、ID とパスワードでログインする。電子カルテシステムは、医師 B を認証して、アーティファクトを発行する。また、医師 B が担当する患者の所属する自治体の在宅介護連絡システムの URL にそのアーティファクトを埋め込み、医師 B 用の画面を提示する。

- 手順４

  医師 B は、担当する患者 A のカルテを参照し、さらに日々の健康状況を見るため、患者 A のカルテ画面に表示されている在宅介護連絡システムの URL にアクセスする。

  在宅介護連絡システムは、受け取ったアーティファクトをもとに、電子カルテシステムの SAML オーソリティに対して、SAML Request プロトコルを利用して、アクセス者（医師 B）の認証を要求する。電子カルテシステムの SAML オーソリティは、在宅介護連絡システムの依頼を受けて、医師 B の認証アサーションを発行する。

  電子カルテシステムの SAML オーソリティは、 SAML Response プロトコルを利用して認証アサーションを含む応答を在宅介護連絡システムに返す。在宅介護連絡システムは、電子カルテシステムの SAML オーソリティから受けた認証結果に基づいて、医師 B のアクセスを許可する。

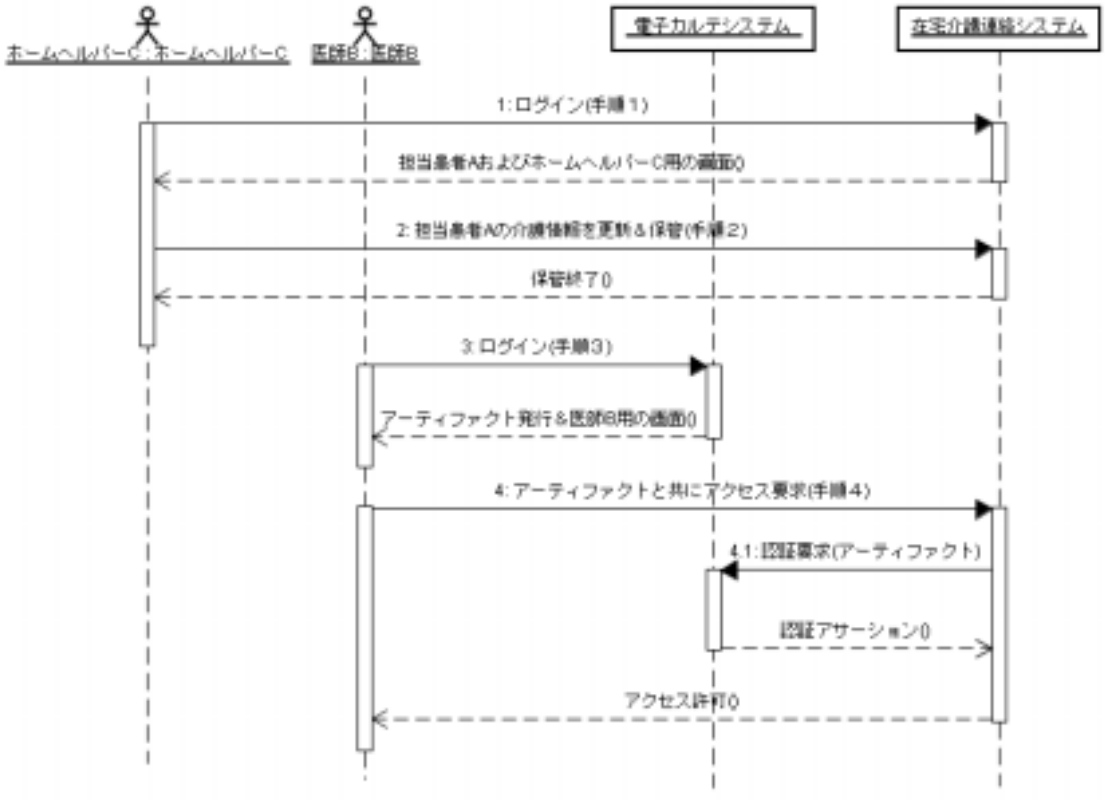


図 3-18 在宅介護連絡システムのシーケンス図

3.6.4 特徴

本ユースケースの特徴は、アクセス制御が必要な個人情報を保持するシステムがまったく別の組織によって管轄されている場合でも、SAML のアサーションを利用することで、１度のログインでアクセスを可能にしているところである。

また、医師による電子カルテサーバへのログインは、事前に医師に付与されたアカウント情報(ID とパスワード)を利用するため、SAML のアサーションを必要としないが、医師が SSO の連携先である在宅介護連絡システムへアクセスすると、在宅介護連絡システムから電子カルテサーバに対して認証要求がなされ、それに基づき認証アサーションが発行されるという流れになっている。医師にとっての病院など自システムへのログイン時には事前に取り決めたアカウントで認証を行ない、外部(SSO 連携先)からの認証要求時にはアサーションを発行して対応するという方法をとることで、既存の認証システムを生かしつつ、外部と SSO を行なうことができると考えられる。

3.6.5 考察

あるエンティティが、異なる組織によって管理・運営される複数のシステムに連続して

アクセスすることは非常に良く起こり得る話であり、さらにそれらの個々のシステム内で管理される情報の間に密接な関連がある場合には、いちいち別のアカウントでログインせずに利用したいというニーズは多いと思われる。本ケースに限らず、ある病院に勤務する医師が、別の病院から転院してきた患者の診療記録を参照したい場合などにも本ケースと同様の仕組みが適用できる。

また、医師の本人性だけでなく、勤務先病院や専門科目などの医師の属性によってアクセス制御を行なったり、薬剤師などの別の資格を取り扱ったりする要求条件もあると考えられ、認証アサーションと属性アサーションを組合せれば、さらに用途が広がると考えられる。

本ケースにおいては、まず自治体側と病院側が患者 1 人 1 人について「どの患者の介護情報に、どの医師がアクセスできるか」を取り決めたうえで、SSO を可能にしているため、事前の取り決め部分(provisioning)を如何に効率よく、作業負担を軽くできるかというところが実際上の課題であると考えられる。

このユースケースにおいて考えられる脅威とその対策としては以下のものがあげられる。

（脅威１）医師の認証ＩＤ／パスワードが漏れてしまうことにより、電子カルテシステムの内容が漏洩してしまう。

（対策１-①）機密性が高い転送手段を利用する。（SSL、IP-VPN 等）

（対策１-②）パスワードを定期的に変更する等の運用ルールを定め、確実に運用を行う。

（対策１-③）IdP による公開鍵証明を活用した認証方法を導入し、本人認証、属性認証を実施する。

3.7 ネット投票/アンケート

SAML により無記名で投票でき、また、不正操作を防止できる投票サイトをつくる。別組織の投票を一度のログインで可能にする。プロファイルとしてアーティファクトを用いる。

3.7.1 背景

ネット投票を行ったり、世論調査やアンケートなどを集計するサイトを、ここでは投票サイトと呼ぶ。投票サイトは悪意のあるユーザが自動投票のプログラムなどを使用して大量の票を投じるなどの不正な操作が行われる可能性がある。投票サイトが公正な投票を実施するためには、同一ユーザによる複数票を排除するなどの仕組みを持たなければならない。この仕組みの一つとして IP アドレスによるアクセス制御があるが、プロキシーサーバや NAT などのネットワーク環境を考えると有効な手段であるとはいい難い。また、投票者を特定する情報により制限をかける方法では、投票の匿名性が損なわれてしまう。

本ユースケースでは、投票サイトの仕組みに SAML を用いることで、投票サイトに対する匿名性と不正投票を防止する仕組みを提供する。投票者の本人確認と有効票の判定を IdP が行い、投票サイトには集計に必要な必要最低限の情報しか与えないように投票者がコントロールすることができる。

3.7.2 登場人物

投票者	ネット投票/アンケートを行う者
アイデンティティプロバイダ(IdP)	投票者の個人情報を管理し、また、有効票であるかどうかの判定も行う。SAML オーソリティとしての役割をもつ。
投票サイト 1,2	投票やアンケートを実施し、集計する。個人情報の収集は行わない。

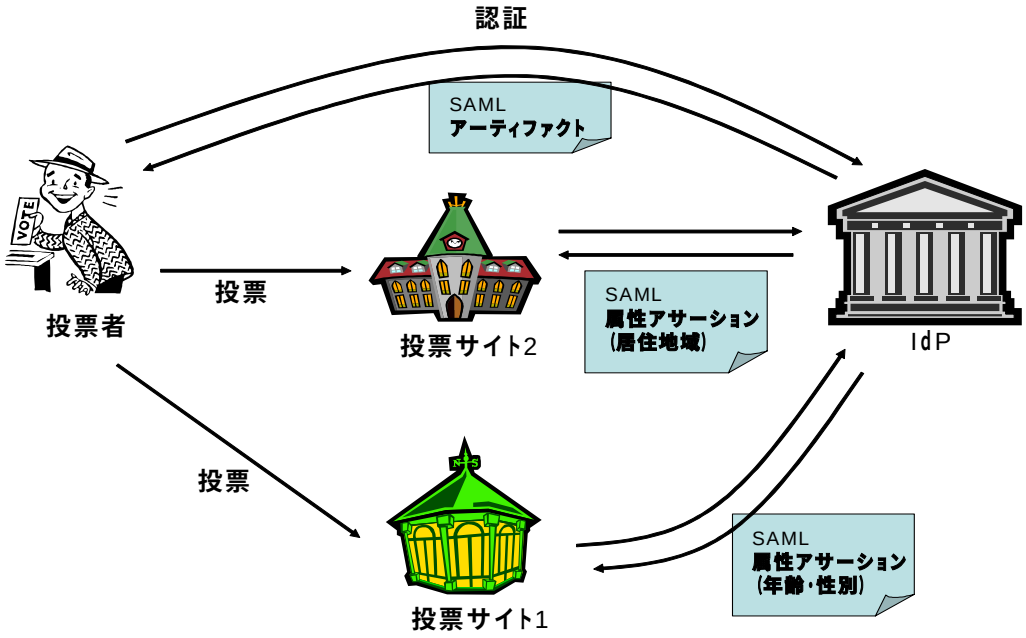

認証
SAML
アーティファクト
VOTE
投票者
投票
投票サイト2
SAML
属性アサーション
(居住地域)
IdP
投票
投票サイト1
SAML
属性アサーション
(年齢・性別)

図 3-19 ネット投票

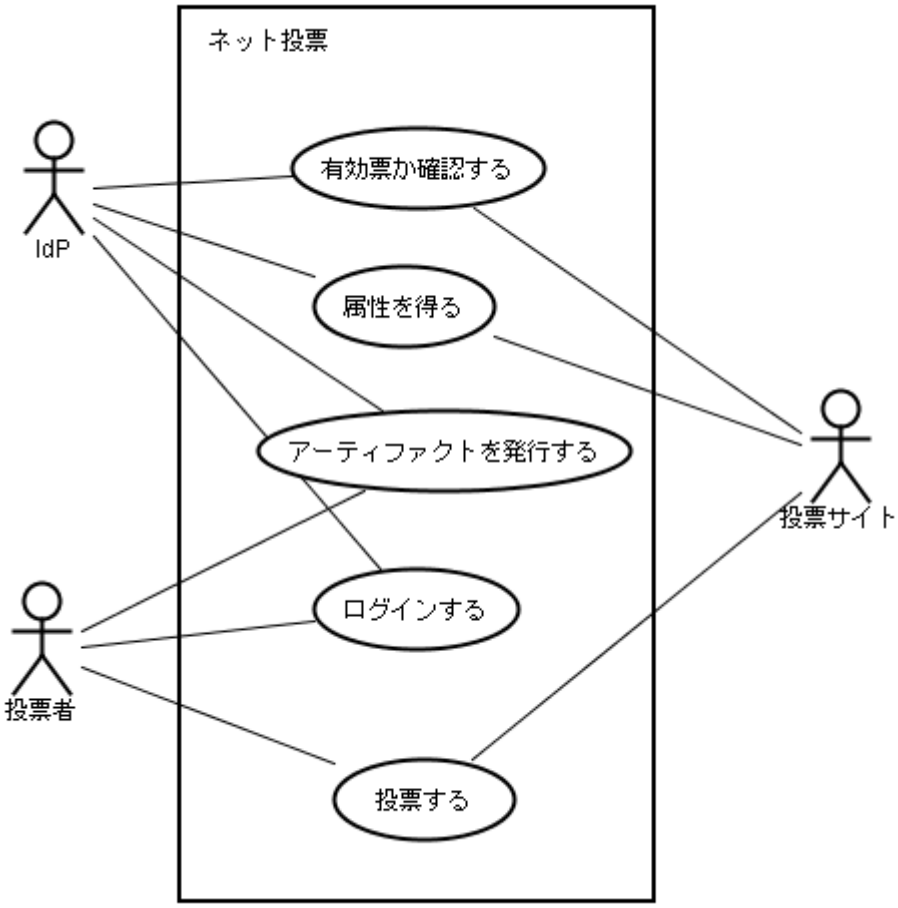


図 3-20 ネット投票のユースケース図

ここで、以下の用語を定義する。

a) **ネット投票**

選挙や採決のためにオンラインで行う投票。

b) **世論調査型アンケート**

ある母集団の中から選出された標本に対するアンケート調査。

c) **モニター型アンケート**

商品の購入者など特定の人に対するアンケート調査。

3.7.3 シナリオ

ネット投票、世論調査型アンケートやモニター型アンケートは同様のメカニズムで実施することができる。

- 手順1

投票サイト 1 は投票者へのネット投票やアンケートの告知を依頼するために、IdP へア

ンケート開始の要求を送信する。投票サイトの目的により手続きが異なる。

a) **ネット投票の場合**

投票サイトは IdP **へ投票者の条件を指定し、ネット投票の告知を行うよう** IdP **へ要求する。**IdP **以外からの告知（例えば投票サイトのページから告知）が行われる場合には、この手続きは不要となる。**

b) **世論調査型アンケートの場合**

投票サイトは IdP **へ投票者の条件を指定し、アンケートの告知を行うよう** IdP **へ要求する。**

c) **モニター型アンケートの場合（商品購入者に対するアンケートの例）**

事前に投票サイトはショッピングサイトと契約しておく。ショッピングサイトは商品を購入した消費者（投票者）のアーティファクトを投票サイトへ送信する。投票サイトはそのアーティファクトを元に、その消費者に対してアンケート協力を告知するよう IdP **へ要求する。**

- 手順２

  IdP は手順１の要求に従い、適切な投票者を選定し、投票サイト 1 のネット投票やアンケートを促すメッセージを電子メール等を用いて送信する。

- 手順３

  投票者は IdP へログインする。

- 手順４

  IdP は投票者に SAML アーティファクトを発行する。

- 手順５

  投票者は投票サイト 1 へ接続し SAML アーティファクトを送信する。

- 手順６

  投票サイト 1 は年齢層や性別、居住地域などの情報が必要であれば、取得した SAML アーティファクトを元に、IdP へ問い合わせる。(SAML リクエスト)

- 手順７

  IdP は投票サイト 1 に対し、要求された属性を与えてよいか、投票者に確認する。

- 手順８

  投票者が投票サイト 1 への属性公開を許可する。許可しない場合には以降の処理は中断される。

- 手順９

  IdP は投票者が事前に許可した属性についてのみ、投票サイト 1 へ属性アサーションとして送信する。（属性アサーション）
- 手順１０

  投票サイト 1 はアーティファクトの保持者である投票者の投票が有効かを IdP に問い合わせる。
- 手順１１

  IdP はアーティファクトの保持者である投票者の情報を調べ、その投票者はまだ投票を行っていないことを確認し、投票サイト 1 へ有効な票であることを伝える。(認可決定アサーション）
- 手順１２

  投票サイト 1 は投票画面を表示する。
- 手順１３

  投票者は投票を許可され、投票あるいはアンケート記入を行う。
- 手順１４

  投票者の操作が終了したら、投票サイト 1 は IdP へアーティファクトに基づき投票終了を伝える。
- 手順１５

  IdP はアーティファクトを元に投票者の投票履歴情報を更新する。
- 手順１６

  IdP は投票履歴情報を更新したのち、登録完了のメッセージを投票サイトに伝える。
- 手順１７

  投票サイト 1 は投票者へ投票完了の画面を表示する。
- 手順１８～手順３０

  投票サイト 2 のネット投票やアンケートが実施されている場合には、手順からまでと同様の処理を行う。すでに投票がログイン済みである場合にはログイン手続きは不要。

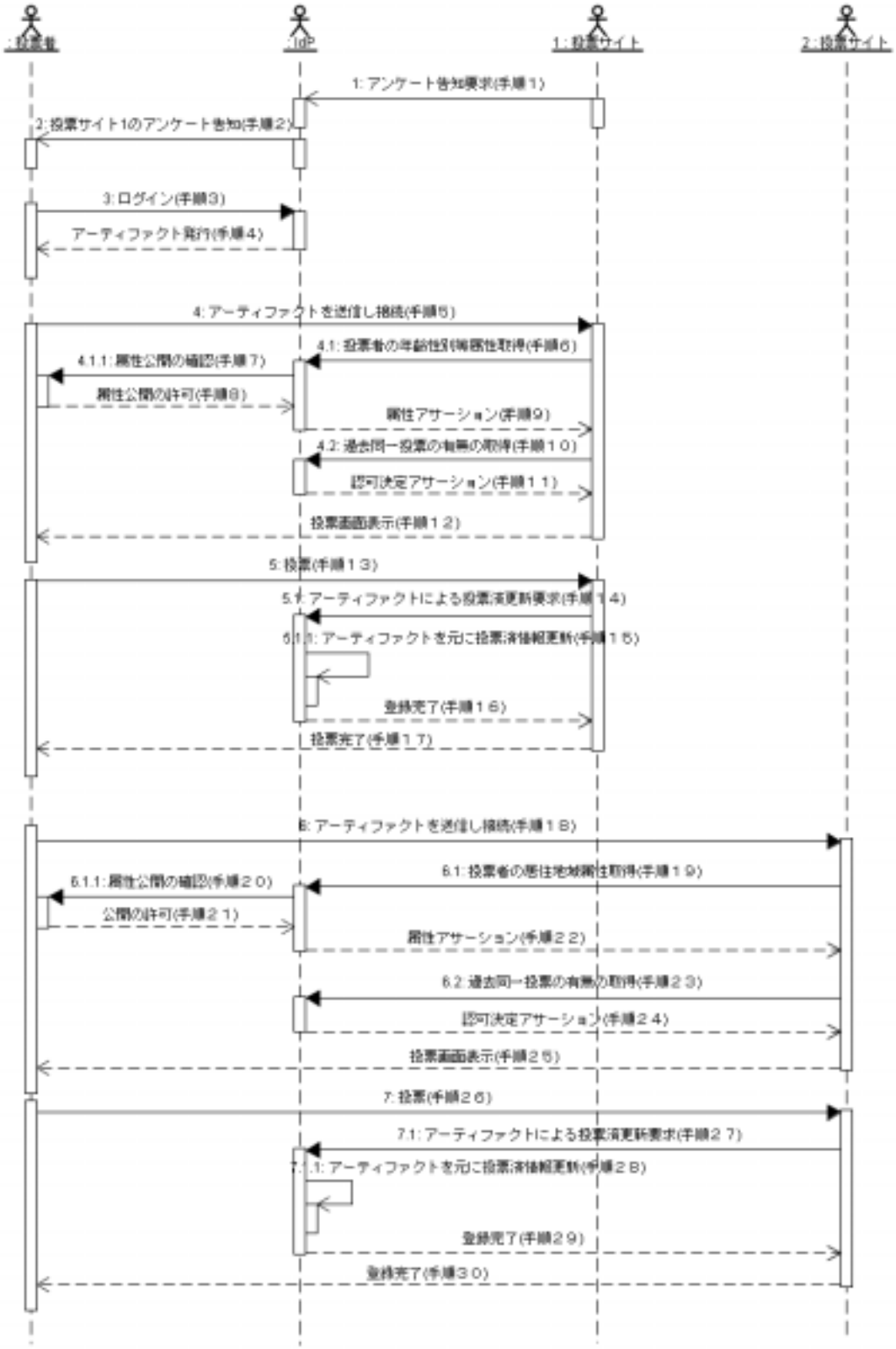


図 3-21 ネット投票のシーケンス図

3.7.4 特徴

投票者の個人情報の管理と有効票の判定を IdP が行うことで、投票サイトは不正な投票の防止と個人情報漏えいによるリスクを回避することができる。

IdP と投票サイトの間で行われる情報交換に SAML のアーティファクトを使用することで、投票者は各投票サイトへ不必要な情報を公開することなく投票することができる。

3.7.5 考察

3.7.5.1 既存サービスとの比較

既存のサービスにアンケート代行サービスがある。アンケート代行サービスでは、会員企業の依頼を受け会員ユーザに対しアンケートを実施する。アンケートは電子メールなどを用いて行い、集計結果を会員企業へ提供する。このようなアンケート代行サービスは本ユースケースでの IdP に相当するが、本ユースケースでは投票の実施方法や集計は投票を行う企業の責任になっている点が異なる。

アンケート代行サービスでは、会員がアンケートに答えるたびにポイントを与え、ある一定のポイントを取得したユーザに景品を与えるというサービスを行っている。本ユースケースにおいても、IdP が投票サイトから投票済情報更新の要求を受けたとき（手順 14）に、対応するユーザのポイントを追加することで、同様のサービスを行うことができる。

3.7.5.2 セキュリティ考察

このユースケースではアーティファクトを使用しているため、**エラー! 参照元が見つかりません。**節に記述されている脅威と対策（特にアーティファクトに対する脅威と対策）が当てはまる。IdP から投票サイトへ流れる属性アサーションは年齢層や居住地域など比較的、秘匿性の低いものを想定しているが、アーティファクトが悪意ある者に入手されたり、偽造された場合には、それをもとに投票者のより秘匿性の高い個人情報を引き出されてしまう恐れがある。アーティファクトに関する脅威への対策を以下にまとめる。

（アーティファクトの偽造への対策）

- アーティファクトの値を予測困難なものにする。

（アーティファクトの不正利用への対策）

- （対策 1）機密性の高い転送手段を用いて盗聴を防止する。
- （対策２）アーティファクトを投票サイトごとに発行し、発行対象である投票サイトとは異なるサイトからリクエストを受けても、アサーションを発行しないようにする。

- （対策３）アーティファクトの有効期限（短時間）を設定する。アーティファクトの発行時刻と、投票サイトからのリクエスト受信時刻の差分が有効期限内にあることを確認する。

3.8 決済

プライバシーへの意識の高まりを背景として、オンラインショッピングにおいては個人情報の保護が課題となっている。オンラインショッピングの決済、配送に SAML を適用することで、匿名性を維持したオンラインショッピングが可能となる。

3.8.1 背景

インターネットの普及により、オンライン上で商品やサービスを購入するオンラインショッピングが利用されるようになっている。反面、相次ぐ個人情報の漏えい事件やプライバシー意識の高まりにより、利用者がショッピングサイトへのクレジットカード情報を含む個人情報の提供に心理的不安を覚えるケースが増加している。特に、ネットワーク経由の情報漏えいは、従来のリアル社会での情報漏えいと比較して、情報の伝播速度が速く、影響が大きいといえる。

他方、本年の個人情報保護法の成立により、個人情報を収集する側にも厳格な管理責任が問われるようになった。不必要な個人情報の収集はショッピングサイトにとってもデメリットとなっている。

このような背景から、匿名性を確保しつつオンラインショッピングを実現できる環境の必要性が高まっている。このためのアプローチとしては、電子マネー、SET、決済代行サービスなどが存在するが、本 TF では決済代行サービスへの SAML の適用に焦点を当てて検討を実施した。

3.8.2 登場人物

オンラインショッピングに登場するアクターとそれらが収集する個人情報を表 3-1に、ユースケース図を図 3-22にてあらわす。ショッピングサイトが提供する商品が、電子データなど配送不要のサービスである場合は、配送サービスは登場しない。

表 3-1 登場するアクター収集情報

アクター	役割	収集する個人情報
ユーザ	ショッピングサイトにて商品を購入する	—
ショッピングサイト	ユーザに商品を提供する	連絡先（メールアドレス）
決済サービス	ショッピングサイトに代わり商品購入代金の決済手段を提供する	決済に必要な情報（クレジットカード番号など）
配送サービス	ユーザが購入した商品を配送する	配送先（住所、氏名、電話番号）

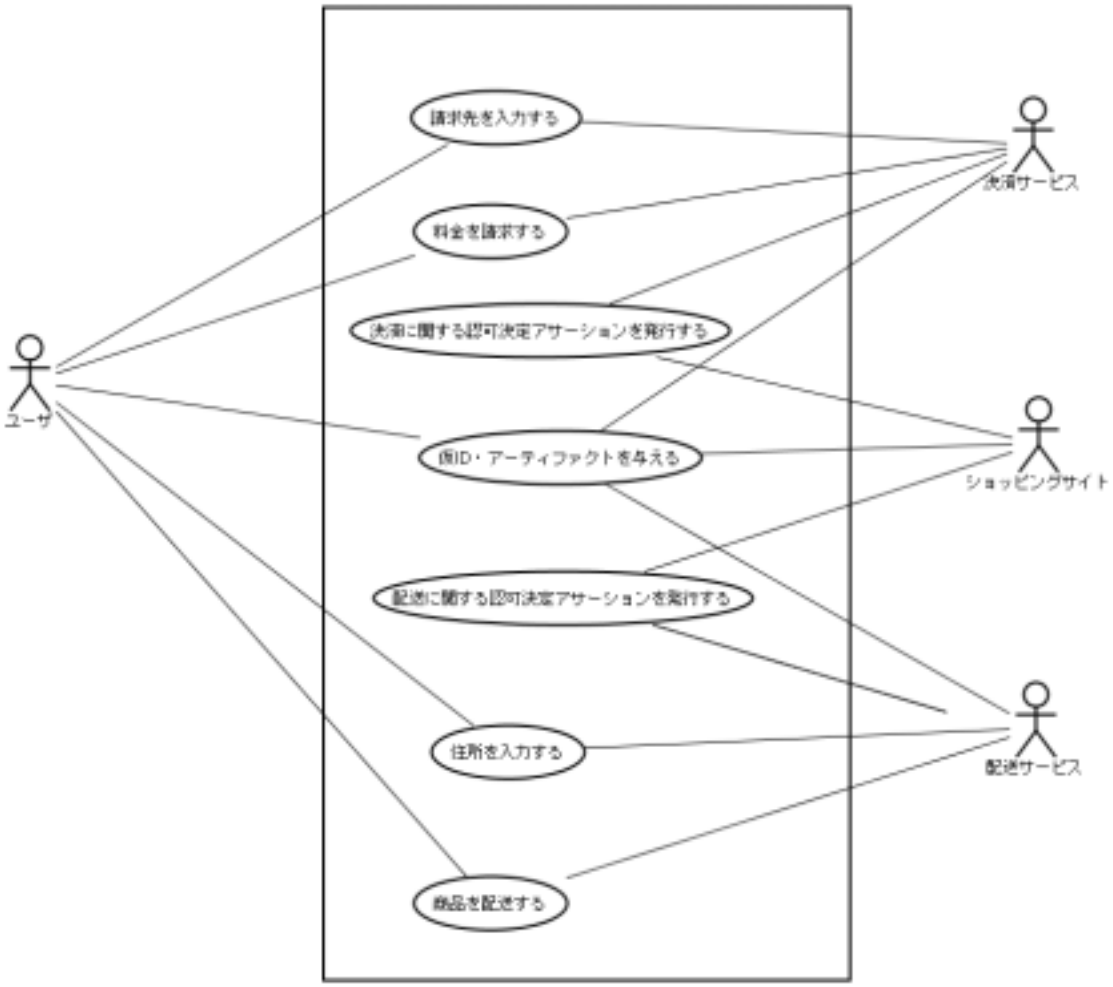


図 3-22 決済サービスのユースケース図

3.8.3 シナリオ

- 手順1

  ユーザはショッピングサイトにアクセスし、購入する商品を決定する。

- 手順2

  ショッピングサイトは一時的なユーザ ID(仮 ID)を生成する。仮 ID は認証アサーション A として生成し、対応するアーティファクト 1 を生成する。

- 手順3

  ショッピングサイトはアーティファクト 1 とともに配送サービスへのリダイレクトをユーザへ指示する 。

- 手順４

  ユーザはアーティファクト 1 とともに配送サービスへアクセスする。
- 手順５

  配送サービスはユーザから取得したアーティファクト 1 を元にショッピングサイトへ認証アサーション A を要求する。
- 手順６

  配送サービスは認証アサーション A と購入物品の情報を得る。
- 手順７

  配送サービスは配送先の入力画面を表示する。ユーザは配送先の情報を入力し、配送サービスへ送信する。
- 手順８

  配送サービスは配送先情報に誤りがないことを確認し、認可決定アサーション X と対応するアーティファクト 2 を生成する。認可決定アサーション 2 には、認証アサーション A の情報が付与されている。
- 手順９

  配送サービスはアーティファクト 2 とともにショッピングサイトへのリダイレクトをユーザへ指示する。
- 手順１０

  ユーザはアーティファクト 2 とともにショッピングサイトへアクセスする。
- 手順１１

  ショッピングサイトはユーザから取得したアーティファクト 2 を元に配送サービスへ認可決定アサーション X を要求する。
- 手順１２

  認可決定アサーション X と配送金額を取得する。
- 手順１３

  ショッピングサイトは決済サービスのためにユーザに新しい仮 ID を割り当てる。新しい仮 ID は認証アサーション B として生成し、対応するアーティファクト 3 を生成する。
- 手順１４

  ショッピングサイトはアーティファクト 3 とともに決済サービスへのリダイレクトをユーザへ指示する。

- 手順１５

  ユーザはアーティファクト 3 とともに決済サービスへアクセスする。

- 手順１６

  決済サービスはユーザから取得したアーティファクト 3 を元に、ショッピングサイトへ認証アサーション B を要求する。

- 手順１７

  決済サービスは認証アサーション B と購入物品および決済金額を取得する。

- 手順１８

  決済サービスは、購入物品と決済金額を画面に表示し、ユーザの購入意思確認を促す。

- 手順１９

  ユーザは決済サービスに決済情報を送信する。決済サービスは決済情報を元に決済処理を行う。

- 手順２０

  決済サービスは決済可否の情報を持つ認可決定アサーション Y と対応するアーティファクト 4 を生成する。

- 手順２１

  決済サービスはアーティファクト 4 とともに、ショッピングサイトへのリダイレクトをユーザへ指示する。

- 手順２２

  ユーザはアーティファクト 4 とともにショッピングサイトへアクセスする。

- 手順２３

  ショッピングサイトはユーザから取得したアーティファクト 4 を元に、決済サービスへ認可決定アサーション Y を要求する。

- 手順２４

  ショッピングサイトは認可決定アサーション Y を取得する。

- 手順２５

  ショッピングサイトは認可決定アサーション Y により決済が完了していることを確認し、アーティファクト 2 とともに配送サービスへ商品の配送を依頼する。

- 手順２６

  ショッピングサイトは決済および配送処理が完了したことをユーザへ通知する

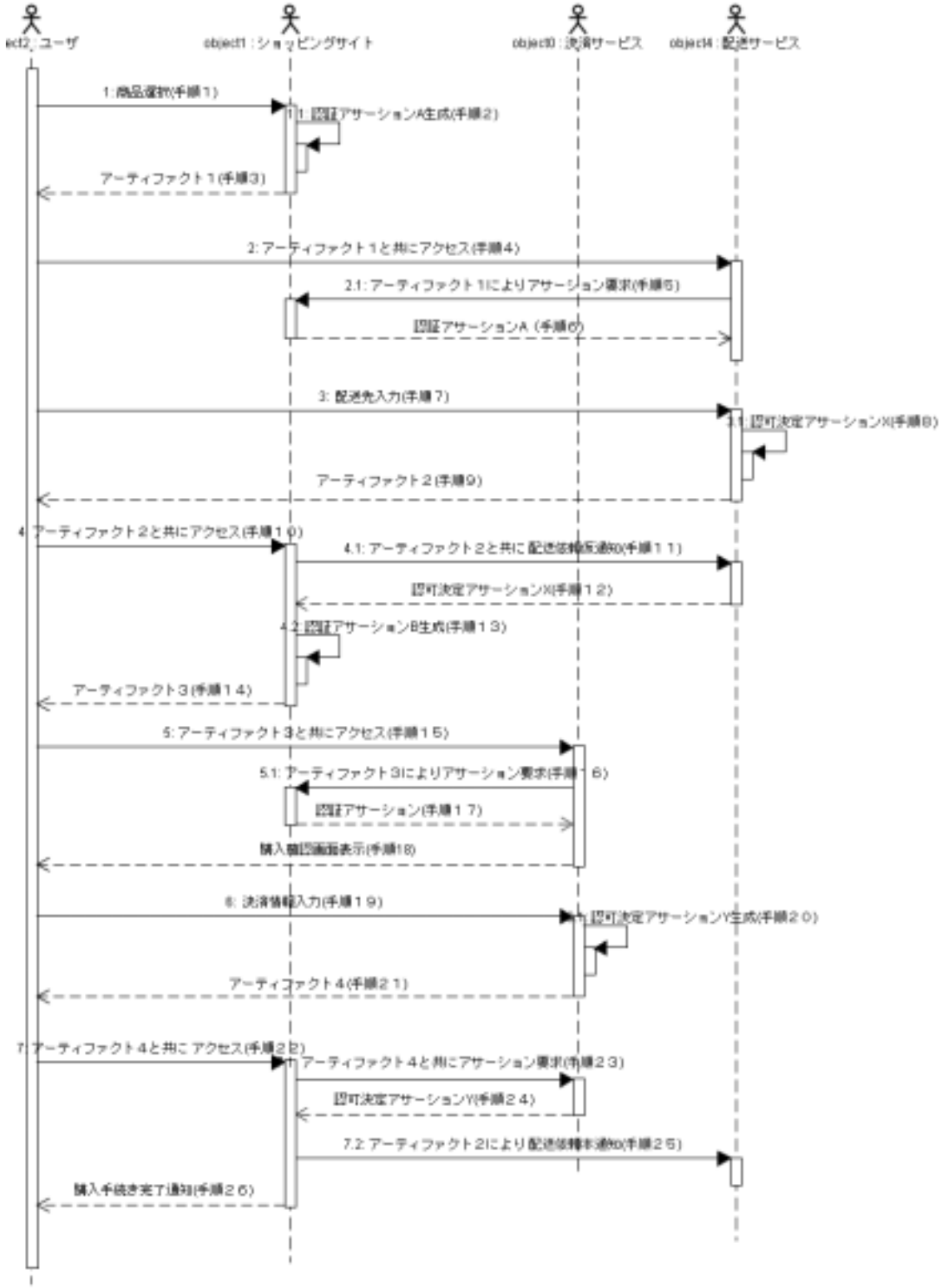


図 3-23 決済サービスのシーケンス図

3.8.4 特徴

ユーザは属性情報（クレジットカード番号、配送先住所、氏名）をショッピングサイトに知らせることなく、商品、サービスの購入が可能である。IdP から決済代行サービスと配送サービスへの情報提供時に SAML アーティファクトを利用することで、ユーザの匿名性を保ったままオンラインショッピングに必要な各種情報を提供することができる。

各サイトが生成するアサーションの内訳は下記の通りとなる。

アサーション	発行者	記載内容
認証アサーション A	ショッピングサイト	ショッピングサイトが一時的に生成した仮ユーザ ID を記載し、AudienceRestrictionCondition には配送サービスを指定する。
認証アサーション B	ショッピングサイト	ショッピングサイトが一時的に生成した仮ユーザ ID を記載し、AudienceRestrictionCondition には決済サービスを指定する。仮ユーザ ID には認証アサーション A と異なる ID を使用すること。
認可決定アサーション X	配送サービス	AuthorizationDecisionStatement に認証アサーション A の仮ユーザ ID を記載し、配送の可否を Decision 属性にて通知する。 AudienceRestrictionCondition にはショッピングサイトを指定する。
認可決定アサーション Y	決済サービス	AuthorizationDecisionStatement に認証アサーション B の仮ユーザ ID を記載し、決済の可否を Decision 属性にて通知する。 AudienceRestrictionCondition にはショッピングサイトを指定する。

その他の特徴は以下の通りである

配送サービスと決済サービスの間の名寄せを防ぐため、認証アサーション A と認証アサーション B には異なる仮ユーザ ID を用いる。

悪意あるユーザによる決済金額の改変を防ぐため、決済金額や購入物品などの情報はユーザの Web ブラウザを経由せずに、アサーションと同時にサイト間で交換する。アサーションの拡張により、アサーション内にこれら情報を記述することもできる。

ユーザによるリプレイアタックを防ぐため、認可決定アサーションには AudienceRestrictionCondition を指定する。（たとえば、認可決定アサーション Y はレシートと同様の意味合いを持つため、リプレイが可能であると一度の決済で何度も物品購入が可能になってしまう）

3.8.5　考察

近年では、サイト上で重要な操作（パスワードの変更、購入意思の伝達）をユーザが行う場合には、ログイン時の認証に加えもう一度認証を行うことが主流となっている。本モデルにおいても決済時には決済サービスが何らかの手段によってユーザに決済の意思表示を確認する手段が必要になるであろう。

4 Liberty と WS-Federation について

4.1 Liberty Alliance アーキテクチャの概要

現在、Liberty Alliance アーキテクチャについては、下記の文書が公開されている。

（http://www.projectliberty.org/jp/resources/index.html）

「Liberty 仕様フェーズ１」

- Architecture Overview v1.1

  Liberty・アーキテクチャ概要 v1.1

「Liberty 仕様フェーズ２」

- Liberty ID-FF Architecture Overview

  Liberty ID-FF アーキテクチャ概要 v1.2

- Liberty ID-WSF Overview (Draft v1.0-07)

  Liberty ID-WSF アーキテクチャ概要 Draft v.1.0-07 [

上記資料によると、Liberty の全体的なアーキテクチャは、以下の 3 つの相互に関連するコンポーネントから構成される。

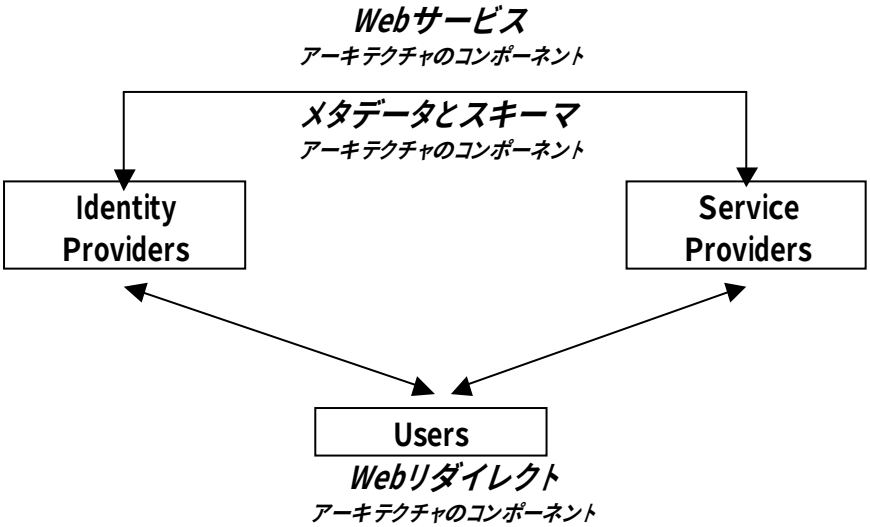


図 4-1 Liberty の全体的なアーキテクチャ

以下に、アーキテクチャの各コンポーネントの役割をまとめる。

表 4-1　Liberty アーキテクチャのコンポーネント

Web リダイレクト	既に導入されているユーザエージェントによって Liberty 対応エンティティがサービスを提供できるようにする操作
Web サービス	Liberty 対応エンティティが直接通信できるようにするためのプロトコルのプロファイル
メタデータとスキーマ	さまざまなプロバイダ固有およびその他の情報を伝えられるようにするために Liberty 対応サイトで使用される一般的なメタデータとフォーマットの組合せ

以降の「4.1.1」～「4.1.6」では、「Liberty アーキテクチャ概要 バージョン 1.1」を用い、アーキテクチャの概要を説明する。

4.1.1　Web リダイレクトのアーキテクチャコンポーネント

Web リダイレクトのアーキテクチャコンポーネントは、以下の 2 種類のリダイレクトがある。

- HTTP リダイレクトによるリダイレクト
- フォーム POST によるリダイレクト

これらの 2 種類のリダイレクトでは、ユーザエージェントを介して IdP と SP の間に通信チャネルが作成される。

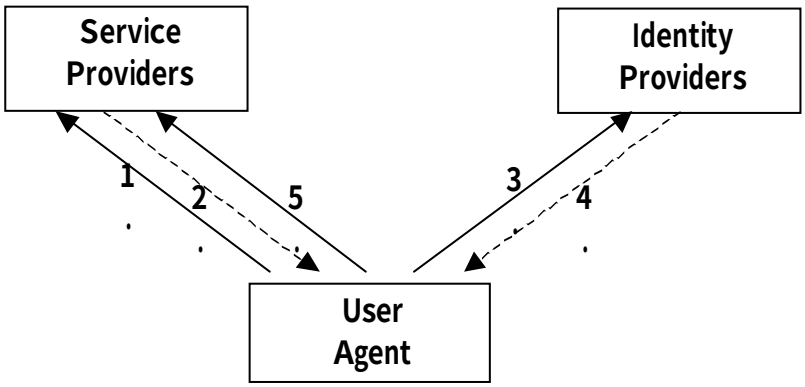


図 4-2 ユーザエージェントを経由する SP と IdP 間の Web リダイレクト

(1) HTTP リダイレクトによるリダイレクト

HTTP リダイレクトによるリダイレクトは、HTTP プロトコルの HTTP リダイレクトクラスの応答（リダイレクト）と URI の文法を使用して、IdP と SP 間に通信チャネルを提供する。そのため、上記図に示された手順では、以下のように SP と IdP 間に通信チャネルが作成される。

1. ユーザエージェントが SP に HTTP リクエスト（通常は GET）を送信する。通常はここでユーザがユーザエージェントに表示された Web ページのリンクをクリックする。
2. SP が状態コード 302（リダイレクト）の HTTP 応答を返し、Location ヘッダフィールドの URI を変更する。この例の場合、Location URI は IdP をポイントし、その URI には SP をポイントするために埋め込まれた別の URI が含まれる。
3. ユーザエージェントが IdP に HTTP リクエスト（通常は GET）を送信し、GET の引数として手順 2 で返された応答の Location フィールドから取得した URI 全体を指定する。注：この URI には、SP をポイントする 2 つ目の埋め込まれた URI が含まれる。
4. IdP が Location ヘッダフィールドに（手順 3 で提供された GET 引数の URI から抽出した）SP をポイントする URI を含むリダイレクトで応答し、自分自身をポイントする 2 つ目の埋め込まれた URI を任意で含める。
5. ユーザエージェントが SP に HTTP リクエスト（通常は GET）を送信し、GET の引数として手順 4 で返された応答の Location フィールドから取得した完全な URI を指定する。

   注：この URI には、IdP をポイントする 2 つ目の埋め込まれた URI が含まれる場合がある。

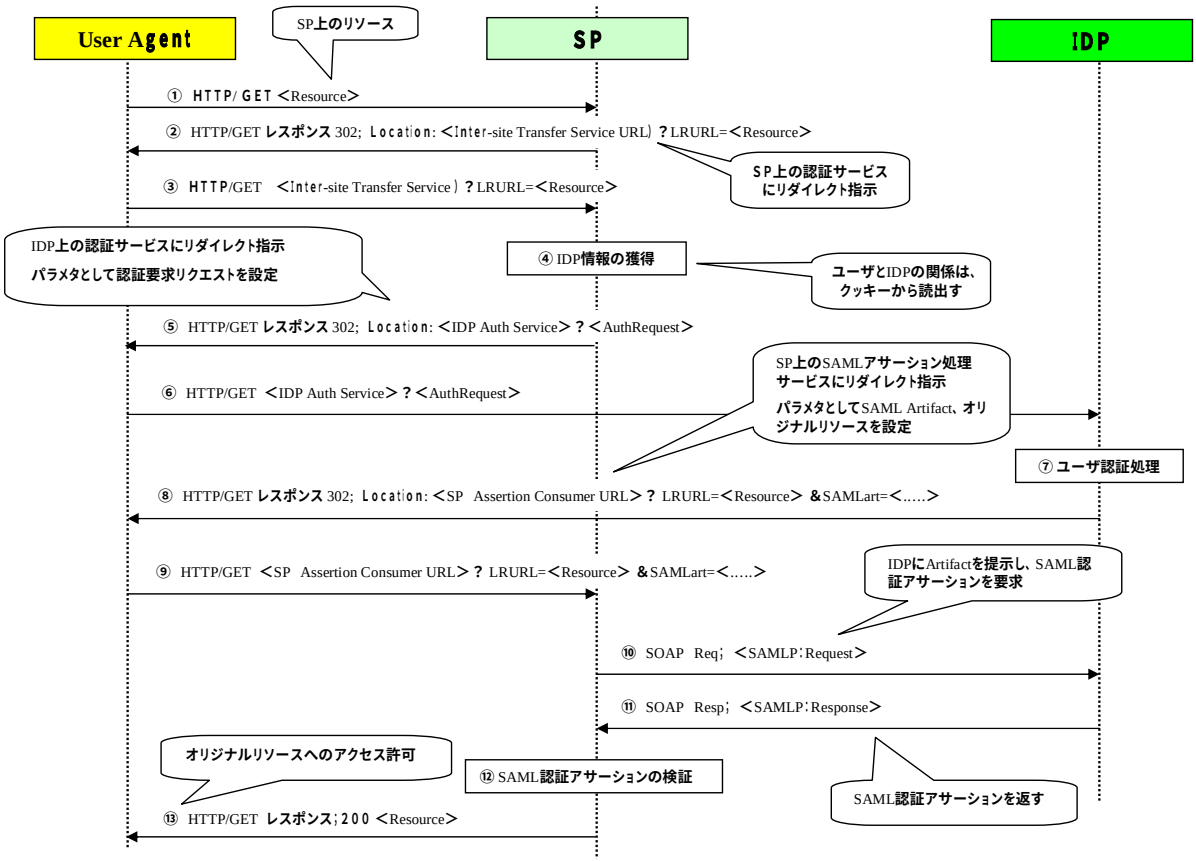


図 4-3 HTTP リダイレクト機能による認証処理（Liberty Browser Artifact Profile）

(2) フォーム POST によるリダイレクト

フォーム POST によるリダイレクトでは、上記、図の手順の手順 2、手順 3 が、以下のように変更される。

1. ユーザエージェントが SP に HTTP リクエスト（通常は GET）を送信する。通常はここでユーザがユーザエージェントに表示された Web ページのリンクをクリックする。
2. SP がユーザエージェントに対し、HTML フォームを返して応答する。この際 HTML フォームには IdP をポイントする action パラメータと POST 値を持つ method パラメータが含まれる。他のフォームフィールドに任意のデータが含まれる場合もある。またフォームには、ユーザの操作なしで次の手順を実行するような JavaScript や ECMAscript フラグメントが含まれる場合もある。

3. ユーザが Submit ボタンをクリックするか、JavaScript または ECMAscript が実行される。どちらの場合も、フォームとその任意データの内容が HTTP POST メソッドによって IdP に送信される。
4. IdP が Location ヘッダフィールドに（手順 3 で提供された GET 引数の URI から抽出した）SP をポイントする URI を含むリダイレクトで応答し、自分自身をポイントする 2 つ目の埋め込まれた URI を任意で含める。
5. ユーザエージェントが SP に HTTP リクエスト（通常は GET）を送信し、GET の引数として手順 4 で返された応答の Location フィールドから取得した完全な URI を指定する。

   注：この URI には、IdP をポイントする 2 つ目の埋め込まれた URI が含まれる場合がある。

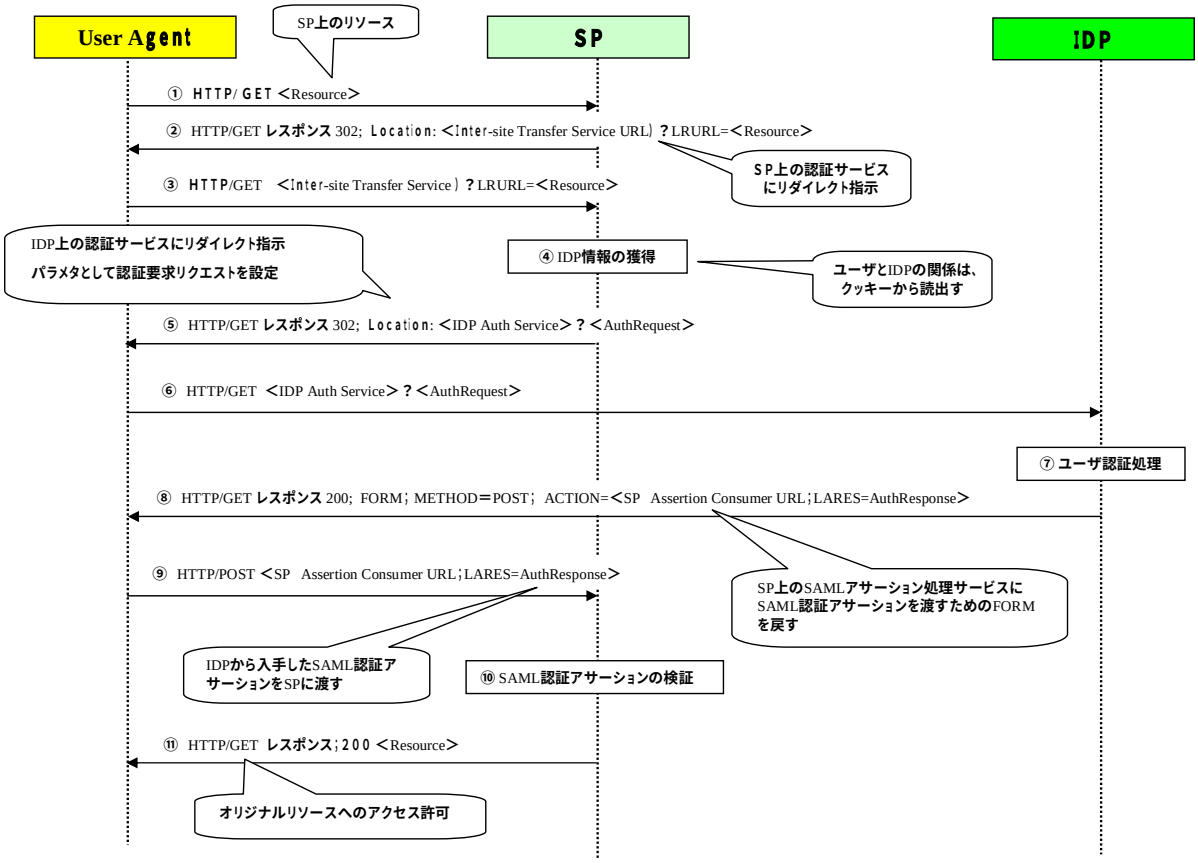


図 4-4　FORM-POST ベースの認証処理 (Liberty Browser POST Profile)

(3) クッキー

クッキーは、[RFC2965]に規定された HTTP 状態管理メカニズムで、ユーザエージェントに状態を維持する。異なる複数の IdP および SP でクッキーを使用して通信する場合、ユーザはセキュリティ設定を低く設定する必要があり、これは受け入れにくい条件である。ポリシーによりでクッキーをオフにする組織もある。

Liberty では、ユーザの同意のもと、ユーザがどの IDP を使用しているのかを SP が確認するための手段としてクッキーを利用できるようにも仕様化されている。

- 個人識別情報と認証情報がクッキーに含まれる場合は、実装者および導入者のクッキー使用には十分な検討を行う必要がある
- ディスクに書き込まれる永続的なクッキーは、ユーザエージェントの呼び出しによって持続されるため、特に注意を要する
- 永続的なクッキーにセッションの認証トークンがキャッシュされる場合、ユーザがブラウザを終了した後、認証メカニズムによって課された認証期限が切れていない場合、2 人目のユーザが 1 人目のユーザになりすましが可能である
- 永続的なクッキーは、ユーザの承諾が得られた場合に限って使用すべきである

(4) Web リダイレクトのまとめ

Web リダイレクトは、理想的な分散システムアーキテクチャではない。

Web リダイレクトチャンネルによる通信には、十分に検証された多くのセキュリティの脆弱性が存在する。そのため Web リダイレクトを使用するプロトコルを設計する際には注意が必要である。そのような注意点は[LibertyBindProf]に規定し、プロファイルの設計に組みこまれている。

① 盗聴

SSL または TLS セッション、あるいは IPSec による VPN などで保護しないと、通信は平文で流れる。

② ユーザエージェントの漏洩

チャンネルはユーザエージェントを経由してリダイレクトされるため、ユーザエージェントに情報がキャッシュされ、後で漏れる可能性がある。情報は平文でブラウザに保持されるため、通信路の保護では保護できない。そのため機密情報は送信する前に暗号化する必要がある。

4.1.2 Web サービスのアーキテクチャコンポーネント

Liberty のプロトコルは、Web リダイレクト以外のシステムエンティティ間でも直接実行され、SOAP を利用する。

4.1.3 メタデータとスキーマのアーキテクチャコンポーネント

メタデータとスキーマは、SP と IdP 間でプロトコルまたはプロトコル以外の方法で交換されるさまざまな情報のサブクラスとそのフォーマット全般を示す包括的な用語である。

(1) **アカウント/アイデンティティ**

Liberty Version 1.0 でのアカウント/アイデンティティは、単に SP と IdP が通信時にユーザを参照するための名前として使用するユーザハンドルである。今後の Liberty フェーズで、さまざまな属性が含まれる予定である。

(2) **認証コンテキスト：**

Liberty は、IdP による任意の認証メカニズムおよび技術に適応し、IdP が認証技術を選択することで、その手順や認証方法により、法的に責任範囲が異なる。選択する認証技術は、主に SP の要件によって決定する。

認証コンテキストスキーマによって、SP と IdP に要件情報を伝える手段が提供されている（[LibertyAuthnContext]を参照）。

(3) **プロバイダのメタデータ：**

IdP および SP が相互に情報を伝えるには、各々に関するメタデータを概念的に持っていなければならない。プロバイダのメタデータには、X.509 証明書やサービスエンドポイントなどが含まれる。[LibertyProtSchema]には、プロバイダのメタデータ交換

に使用される IdP と SP のメタデータスキーマが定義されている。ただし、プロバイダのメタデータ交換プロトコルは、Liberty Version 1.0 仕様の範囲に含んでいない。

4.1.4 シングルサインオンとアイデンティティ連携

Liberty のシングルサインオンとアイデンティティ連携については、シングルサインオンおよび連携プロトコル（Single Sign-On and Federation Protocol）によって進められる。一連のプロトコルの流れで、アイデンティティ連携およびシングルサインオンの両方が進められる。

(1) アイデンティティ連携

Liberty では、複数の IdP と SP が含まれる環境下で、プロバイダ間において、ユーザが（自分の判断で）一意に識別される機構を提供する。

図 4-5 アイデンティティ連携

ユーザが IdP を使用して初めて SP にログインした際にアイデンティティ連携を選択すると、明示的なトラストチェーンが作成される。トラストサークルのメンバーは、ユーザに実際のアカウント識別子を提供する必要はなく、その代わりにあるユーザにハンドルを提供できる。このような連携に含まれる IdP と SP は他のユーザハンドルを記憶する必要があるため、ユーザディレクトリにエントリを相互に作成し、互いのユーザハンドルを記録する。

二者相互間のアイデンティティ連携、すなわち SP と IdP の両者がユーザのためにハンドルを交換する機構は、IdP の数と SP の数のパターンによって、以下の 4 種類に分類される。

表 4-2　IdP と SP のパターンについて

№	IdP の数	SP の数	IdP 間の連携	ケース	ハンドル交換方法
1.	単一	単一	―	ケース1	SP と IdP 両者のハンドル（あるいは IdP のみのハンドル）が交換される
2.	単一	複数	―	ケース2	SP 間で情報が直接やりとりされることはなく、IdP と個別に情報をやりとりすることしかできない。
3.	複数	単一	無し	ケース3	複数の IdP を特定の SP にリンク可能 IdP は相互の情報にアクセスできない
4.	複数	単一	有り	ケース4	新しい SP を登録する際に両方の IdP に登録するのは煩わしいので、複数の IdP を連携させ、IdP が相互の情報にアクセスできるようなポリシーを設定可能。

(2) シングルサインオン

シングルサインオンは、ユーザの IdP および SP のアイデンティティが連携されると有効になる。ユーザから見ると、シングルサインオンは、ユーザが IdP にログインし、再度サインオンすることなく連携する複数の SP を使用した場合に認識される。

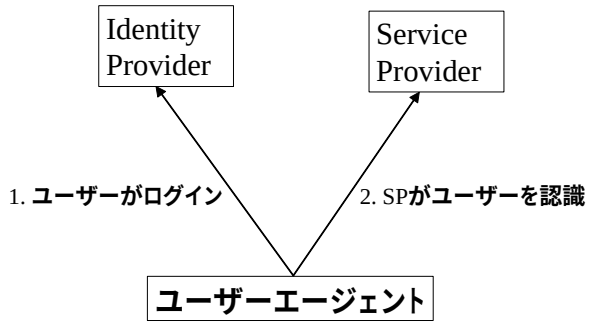


図 4-6 シングルサインオン

(3) シングルサインオンおよび連携プロトコルのプロファイル

シングルサインオンおよび連携プロトコルには、SP と IdP 間で交換されるメッセージが定義されている。これらのメッセージの特定の転送（HTTP など）とメッセージ（SOAP）プロトコルへの具体的なマッピングは、プロファイルと呼ばれる。シングルサインオンおよび連携プロトコルには、４つのプロファイルが規定されている。

４つのプロファイルを通信方式、SP と IdP 間の直接通信、メッセージのサイズ制限という観点から表にまとめた。

表 4-3 4つのプロファイル

プロファイル名	通信方式	SP と IdP 間の直接通信	メッセージのサイズ制限
Liberty ブラウザのアーティファクトプロファイル	Web リダイレクトによって IdP と SP 間で交換される URI へのアーティファクトの埋め込みが規定される	必要	アーティファクトが URL エンコードされた形で小さくなるため、サイズ制限を心配することなく URI に適合する。
Liberty ブラウザの POST プロファイル	ブラウザ機能によって、自動的にフォームをポストする JavaScript や ECMAscript データを含むフォーム要素が存在する HTML ページを送信して、リダイレクトを実行可能。	不要	HTML フォームを使用することで、URL に比べて、サイズ制限が大きくなるため、完全な認証アサーションを含めることが可能
Liberty WML の POST プロファイル	WML イベントに依存して、WML ブラウザに HTTP フォームの送信を命令する。WML ブラウザは、携帯端末で一般的であり、専用のプロキシである WAP ゲートウェイを経由して通信する。このプロキシは、端末のワイヤレスセッションプロトコル（Wireless Session Protocol）を HTTP に変換する。	不要	同上
Liberty 対応クライアントおよびプロキシのプロファイル	ユーザが SP で使用する IdP を認識しているか、または認識する方法を知っているクライアント。 Liberty 対応プロキシは、Liberty 対応クライアントをエミュレートする HTTP プロキシ（通常は WAP ゲートウェイ）である。	不要	Liberty 対応クライアントは、POST を使用して HTTP リクエストおよび応答の本体で Liberty メッセージを送受信する。そのため、Liberty 対応クライアントには、Liberty プロトコルメッセージのサイズ制限が存在しない。

① Liberty ブラウザのアーティファクトプロファイル

Liberty ブラウザのアーティファクトプロファイルでは、Web リダイレクトによって IdP と SP 間で交換される URI へのアーティファクトの埋め込みが規定され、さらに SP と IdP 間の直接通信が必要となる。この方式の目的は、アーティファクトが URI エンコードされたフォーム内で小さくなるため、サイズ制限を心配することなく URI に適合することである。

② Liberty ブラウザの POST プロファイル

JavaScript や ECMAscript をサポートする最近のブラウザは、自動的にフォームをポストする JavaScript や ECMAscript データを含むフォーム要素が存在する HTML ページを送信して、リダイレクトを実行することができる。アサーションを取得するために SP と IdP 間で直接通信する必要がない。HTML フォームを使用することで、URL に比べて、サイズ制限が大きくなるため、完全な認証アサーションを含めることが可能。

③ Liberty WML の POST プロファイル

Liberty WML の POST プロファイルは、WML イベントに依存して、WML ブラウザに HTTP フォームの送信を命令する。WML ブラウザは、携帯端末で一般的であり、専用のプロキシである WAP ゲートウェイを経由して通信する。このプロキシは、端末のワイヤレスセッションプロトコル（Wireless Session Protocol）を HTTP に変換する。

④ Liberty 対応クライアントおよびプロキシのプロファイル

Liberty 対応クライアントは、ユーザが SP で使用する IdP を認識しているか、または認識する方法を知っているクライアントである。また、Liberty 対応クライアントは、HTTP リダイレクトに依存したり、URL にプロトコルパラメータをエンコードしたりするのではなく、POST を使用して HTTP リクエストおよび応答の本体で Liberty メッセージを送受信する。そのため、Liberty 対応クライアントには、Liberty プロトコルメッセージのサイズ制限が存在しない。

Liberty 対応プロキシは、Liberty 対応クライアントをエミュレートする HTTP

プロキシ（通常は WAP ゲートウェイ）である。

⑤ シングルサインオンプロトコルのフローの例：Liberty ブラウザのアーティファクトプロファイル

Liberty ブラウザのアーティファクトプロファイルにおけるシングルサインオンの第 1 段階は、ユーザが SP を訪問して、ユーザの好みの IdP を選択してログインすることである。このログインは、SP のログインページに表示される一覧から好みの IdP を選択することで実行される。

ユーザが IdP を選択すると、発信元の SP を示すパラメータが埋め込まれた状態で、ユーザのブラウザが IdP にリダイレクトされる。ユーザは、通常どおりの方法で IdP にログインできる。

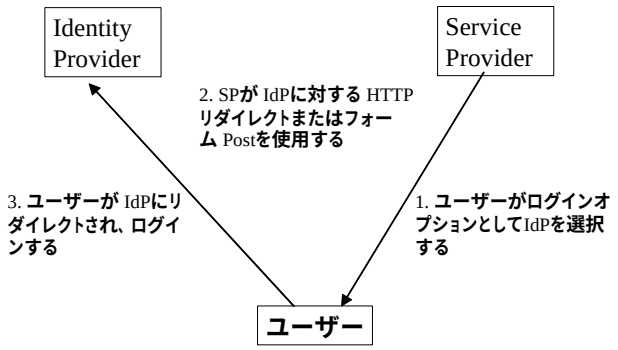


図 4-7 シングルサインオンプロトコルのフローの例 (ログオン)

IdP は通常どおりログインを処理し、ログインが終了すると、アーティファクトと呼ばれる暗号化された一時的なクレデンシャルを URI に埋め込んで、ユーザのブラウザを発信元の SP にリダイレクトする。SP は URI のアーティファクトを解析し、直接使用して、IdP にユーザを照会する。応答で IdP がユーザを保証し、SP がローカルな意味でのセッション状態を確立する。

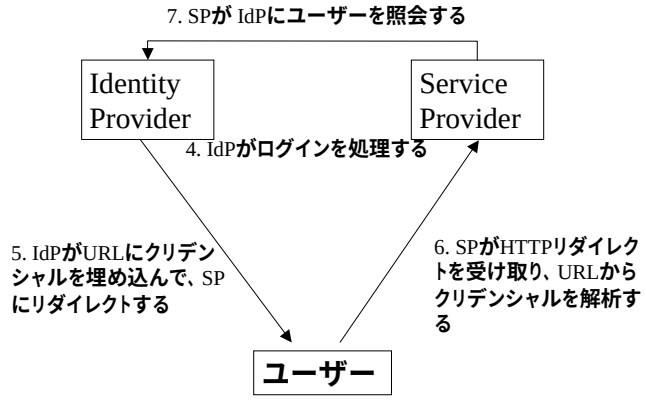


図 4-8 シングルサインオンプロトコルのフローの例 (セッション確立)

4.1.5 IdP イントロダクション

複数の IdP が存在するトラストサークルでは、ユーザがどの IdP を使用しているかを SP が確認するための手段が必要となる。本来は、SP が読み取れるクッキーを IdP が書き込めることが理想的である。Liberty では、ユーザの同意のもと、ユーザがどの IDP を使用しているのかを SP が確認するための手段としてクッキーを利用できるようにも仕様化されている

クッキーは、[RFC2965]に規定された HTTP 状態管理メカニズムであり、Web サーバが情報を保存する、つまりユーザエージェントに状態を維持させるための手段である。ただし、普及率の高いユーザエージェントのデフォルトのセキュリティ設定では、書き込んだ Web サイトだけがクッキーを読み込めるようになっている。この時のサイトの識別は、読み込みと書き込みを行うサイトの DNS ドメインに基づいて行われる。DNS ドメインの異なる複数の IdP および SP でクッキーを使用して通信できるようにするには、ユーザがユーザエージェントのデフォルトのセキュリティ設定を低く設定しなければならない。このオプションは、たいてい受け入れられない条件となる。また、ユーザおよび/または組織にとって、ユーザエージェントでクッキーをオフにすることは珍しくない。

クッキーには 以上のような制約が存在するため、ある DNS ドメインの IdP が別の

DNS ドメインの SP で読み取り可能なクッキーを書き込むための標準的な手段が存在しない。

このイントロダクションの問題の解決策は、AirlineAffinityGroup.inc や AAG.inc のような、該当するトラストサークルに共通するドメインを使用して、すべての参加者がアクセスできるようにすることである。この DNS ドメイン内のエントリは、系列グループのそれぞれのメンバーが指定する IP アドレスをポイントする。たとえば、SP である CarRental.inc は、自身の指定する IP アドレスをポイントするサードレベルドメインの「CarRental.AAG.inc」を取得する。この共通ドメインサービスをホスティングするコンピュータは、状態を保持しない。それらは、リダイレクト URL 内で渡されるパラメータに基づいて、クッキーの読み取りと書き込みだけを行う。

ユーザが IdP に認証を行うと、IdP はユーザがその IdP を使用していることを示すパラメータを付けて、ユーザのブラウザを IdP の共通ドメインサービスのインスタンスにリダイレクトする。共通ドメインサービスは、クッキーにその設定を書き込み、ユーザのブラウザを再び IdP にリダイレクトする。その後、ユーザはトラストサークル内の SP にサイトを変更することができる。

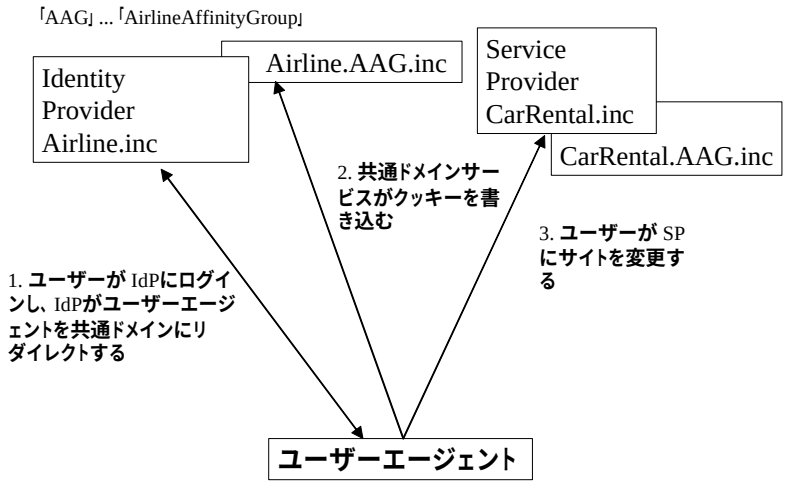


図 4-9 IdP イントロダクションのフローの例

ユーザがトラストサークル内の SP にサイトを変更すると、SP はユーザのブラウザを共

通ドメインサービスのインスタンスにリダイレクトする。共通ドメインサービスがクッキーを読み取って、ユーザの IdP を URL に埋め込んだ状態でユーザのブラウザを再び SP にリダイレクトし、そして SP の DNS ドメイン内で SP のシステムが稼働できるようになる。

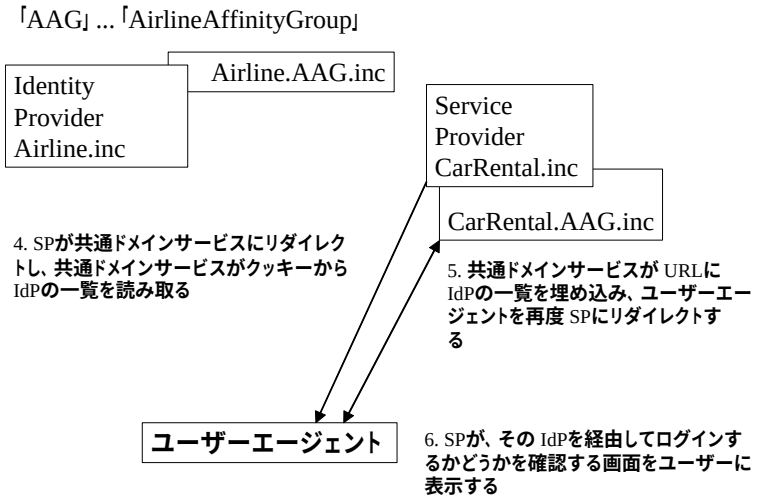


図 4-10 IdP イントロダクションのフローの例

この時点で、SP はユーザがトラストサークル内のどの IdP に認証を行ったかを把握し、ユーザの代理を務めるその IdP と共に、シングルサインオンなどの Liberty プロトコルの運用を進めることができる。

4.1.6 シングルログアウト

シングルログアウトは SP、IdP のいずれかで開始される。その後、IdP はセッションを確立した SP 全てにログアウトを通知する。

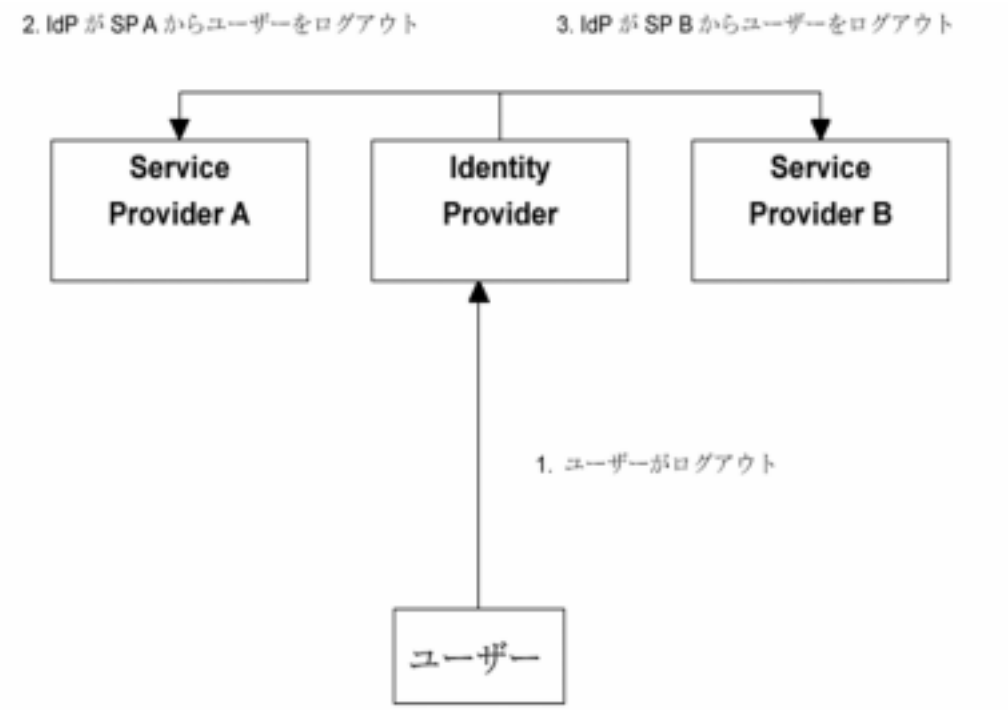


図 4-11 シングルログアウト (IdP でログアウト)

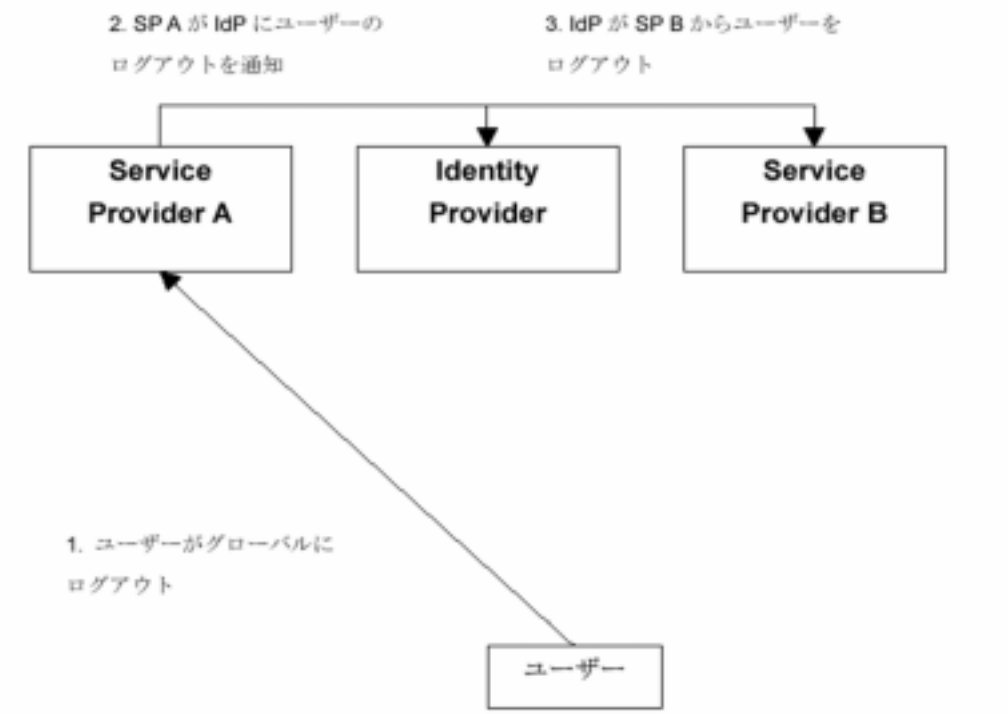


図 4-12 シングルログアウト (SP でログアウト)

4.2 WS-Federation の概要

2003 年 7 月 8 日 に IBM, Microsoft, VeriSign, BEA Systems, RSA Security 等のベンダが、「WS-Federation」を発表し、仕様を公開した。この「WS-Federation」は 2002 年 4 月に IBM と Microsoft が示した Web サービスセキュリティのロードマップの中で「WS-Policy」「WS-Trust」「WS-SecureConversation」の仕様に続くものである。

なお IBM, Microsoft は、「WS-Federation」仕様を近い将来、標準化団体に提出することを表明している。

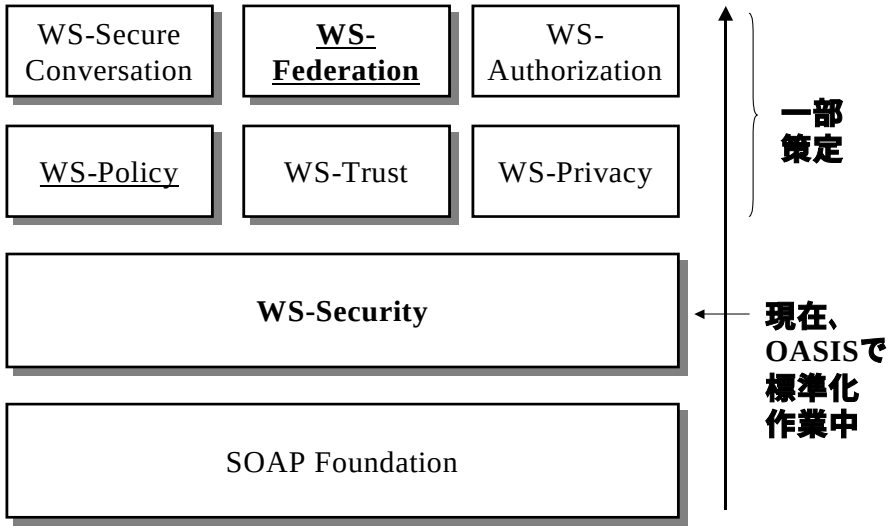


図 4-13 Web Service Security Roadmap

現在 WS-Federation 関連のドキュメントは、以下の４文書となっている。

- Federation of Identities in a Web Services World (white paper)
- WS-Federation: Passive Requestor Profile (Passive Requestor Profile の規約)
- WS-Federation: Active Requestor Profile (Active Requestor Profile の規約)
- Web Service Federation Language (WS-Federation に関する XML 規約)

以降の「4.2.1」～「4.2.2」では、「Federation of Identities in a Web Services World (white paper)」を用い、WS-Federation における連携シングルサインオンと連携 ID について概要を例を挙げて説明する。

4.2.1 連携シングルサインオンと ID 管理

連携シングルサインオンと連携 ID の概念を簡単な事例で示すと下図のように図示することができる。

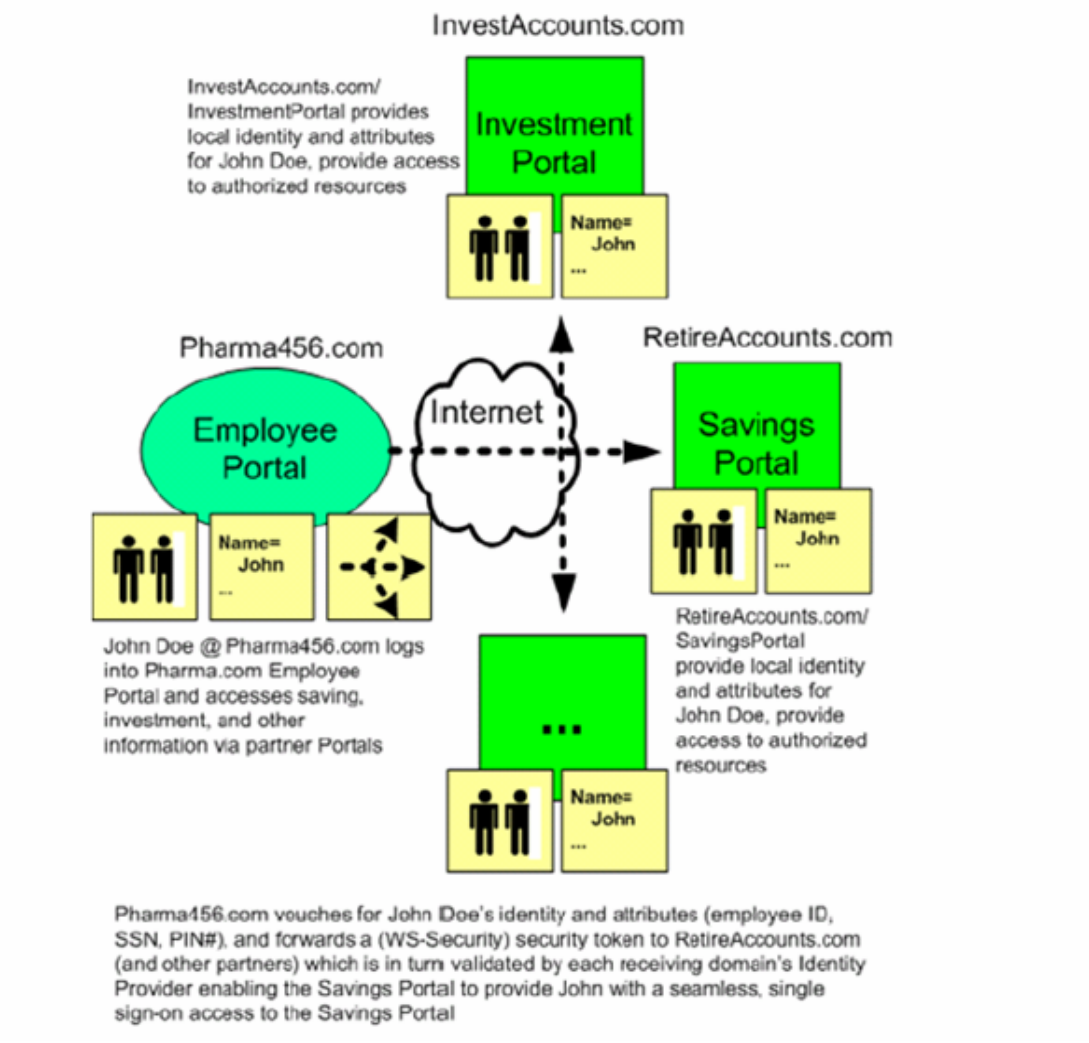


図 4-14 連携シングルサインオンと連携 ID の概念の例

(引用元:Federation of Identities in a Web Services World (white paper))

ユーザ John Doe は、Pharma456.com と呼ばれる製薬会社で働いている。

John は、Pharma456.com にアカウントを持っており、Pharma456.com 内のリソースにアクセスしようとする場合には、認証される必要がある。John は、従業員としての福利厚生的なパートナー企業のサービスを受けることができるようにその企業のアカウントも持っている。この事例では、会社の貯蓄サービスが、RetireAccounts.com と呼ばれる金融サービス会社(サービスプロバイダ)によって管理されている。また、投資サービスは、

InvestAccounts.com(サービスプロバイダ)と呼ばれる金融会社により管理されている。

例えば、RetireAccounts.com によって管理される貯蓄ポータルのようにパートナー企業のサイトにアクセスするためには、ユーザは RetireAccounts.com において認証されなければならない。その際、RetireAccounts.com では、社会保険番号（SSN）と従業員番号と個人の PIN コードを入力が必要となる。

連携が行われていない場合には、John は、既に Pharma456.com で認証済みであったとしても Pharma456.com 経由で RetireAccounts.com の自分の貯蓄口座にアクセスするためには、明に RetireAccounts.com のサイトで再度認証されなければならない。

けれども彼の従業員としてのポータルサイト Pharma456.com に連携シングルサインオンを行うことで、パートナーサイトでの再認証や追加情報の送信を行うことなく、ポータルサイト Pharma456.com 上のメニューをクリックするだけで、自分の貯蓄口座にアクセスすることができるのである。

連携は、また、連携管理の技術を利用することで、二つの会社間で同一のユーザであることが明確に判っている ID を安全に関係付けることができる。連携への参加企業は、信頼された企業が、あたかも自サイトで独立にドメイン内のユーザの ID を検証することができるかのように ID 情報を交換することができる。

（セキュリティ・トークンの）発行ドメインである Pharma456.com と信頼を受けるドメイン（連携サービスプロバイダ RetireAccounts.com）間で連携シングルサインオンを行うことによりユーザ ID と関連する他の属性情報（ロールやグループメンバシップのような認可情報、従業員番号、社会保険番号、PIN コードのような資格情報）のセキュアで信頼性のある転送が可能となる。

連携 ID 管理は（セキュリティ・トークンの）発行企業から受信した（信頼できる）情報に基づいて、Relying party（RetireAccounts.com）ローカルに有効な ID かを決定できる

処理プロセスを規定している。転送内容の詳細は、当該ビジネス（サービスプロバイダ）とユーザの所属する企業ポータル（ID プロバイダ）間で複数のメッセージにより処理されるかもしれないが、それらの処理は、ユーザ（は意識する必要なく）透過に行われる。

(1) **属性サービス** (Attribute Service)

属性サービスは、連携 ID に関する認可属性に対する統一したアクセス手段を提供する。

上図の例では、John Doe が、各パートナーのポータルサイトにアクセスした場合、取引内容に基づいたパーソナライズ化を行ったり、必要とされる情報を獲得する目的で、John の属性情報にアクセスしてもよい。プライバシー保護を確保するために、各サービスに対して提供する認可情報に関しては、John がすべてを制御する機能を持つことが要件となる。Pharma456.com での運用において規定されるプライバシポリシーを遵守していることを保証するために、これらの属性に対するアクセスは、すべてモニターされることができるしログも残す実装にする必要があるだろう。WS-Federation では、特定の属性サービスの提供方式を必須として規定することはせず、代わりに UDDI のようなサービスを（属性サービスとして）利用することができる。

(2) **仮名サービス** (Pseudonyms Service)

プライバシー保護の立場からターゲットサービスに対して ID のマッピングを隠蔽しておきたいという要件がある。すなわち対象となるサービスにおいては、John のローカルな個人情報により”John Doe”を知っているが、グローバルな ID は知らないという運用をしたいとする。

仮名サービスとは、そのような場合にプライバシーと（グローバル）ID を保護するために信頼された ID と仮名間のマッピングを行う機能である。

（連携シングルサインオンを含む）連携 ID 管理は、Web シングルサインオンよりも広い概念である。

4.2.2 連携 Identity モデル

連携 Identity モデルは、Web サービス基盤とセキュリティ仕様をインテグレートして構築することで、一貫性があって、拡張性のある一つのセキュリティモデルを形成している。

(1) **プロファイル**

連携(federation)仕様は、一般的なモデルとフレームワークを提供する。

プロファイルは、異なった環境下で、どのようにモデルが適用されるかを記述したものである。将来追加されるプロファイルは、他の環境に適用する連携モデルを規定するかもしれない。

各々のプロファイル仕様は、passive 又は、active requestors のように与えられた環境下で WS-Federation における機構が、どのように適用されるかを定義する。

(2) Identity Providers （ID **プロバイダ** ）

リクエスタ又はリクエスタの代行者（identity データの所有者であることを認可された ID プロバイダ）が、identity の存在を示し、（SAML におけるアサーションする）ID プロバイダが、そのアサーションが有効であることを検証する。そして、連携（Federation）は、ID プロバイダと ID プロバイダの Identity の決定を信頼するサービスブロバイダ間の信頼を確立するための機能となる。

(3) STS（Security Token Service）

セキュリティ・トークン・サービス（STS）は、共通のモデルとメッセージセットを利用してセキュリティのトークンを発行したり、交換したりする一般的なサービスである。

いかなる Web サービスでも WS-Trust 仕様をサポートすることによって、その Web サービス自体が、STS となることができる。

ある一つの identity provider (IP)は、一つの特別な STS でもある。

連携モデルは、Identities の発行と連携を含むトークンの発行及び交換の機構を利用して、

WS-Trust 仕様で規定されるフレームワーク上に構築されている。本モデルでは、以下のように処理される。

プリンシパルの ID プロバイダ(又は STS:)でのサインーイン。(最後にｻｲﾝ-ｱｳﾄ）。

プリンシパルが、リソース／サービスに対してトークンを要求すると、発行されたトークンは、プリンシパルの主利用 ID か、又は複数存在するかもしれない仮名のどちらかを適切な範囲で表すトークンとなる。

ID プロバイダ（又は STS）は、要求元であり（かつ認可された）受信側に対してメッセージを発行する。

プリンシパルが属性／仮名サービスに登録されることで、属性や仮名が追加されたり、利用されたりするように設定される。

サービスは、提供された ID 情報を使って、ID に関する認可情報を獲得するために属性／仮名サービスに問い合わせすることができる。

下図は､サービスにアクセスするために可能な IP と STS の組合せを図示している。

この例では、

リクエスタが IP(Business456.com)から identity セキュリティ・トークンを獲得

STS(Fabrikam123.com)に対してアサーション要求を送信。Fabrikam123.com が、Business456.com を信頼していれば、認可は承認されてリクエスタに対して、アクセストークンが返される。

そして、リクエスタは、このアクセストークンを利用して、Fablikam123.com に対してサービスを要求する。

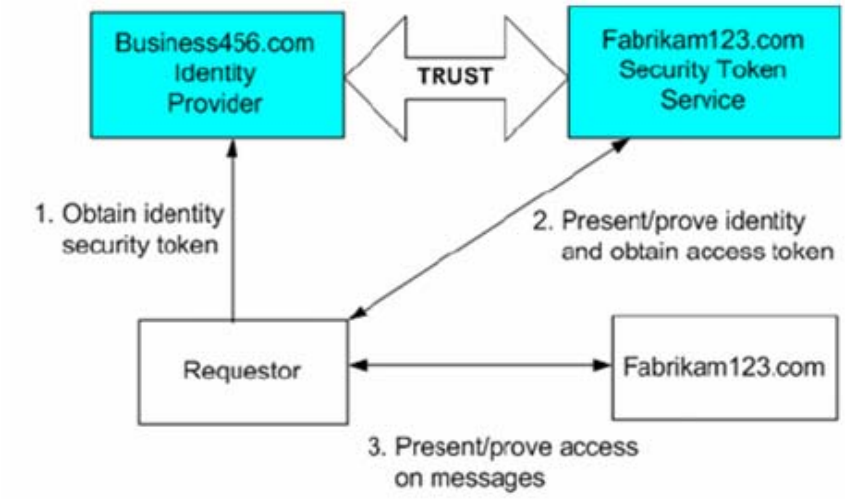


図 4-15 IP と STS の組合せ例

(4) シングルサインオンとサインアウト

WS-Federation 仕様で述べるモデルは、サービス側の管理 (service-managed) とリクエスタ側の管理(requestor-managed)のようにユーザのセションにおいて異なる概念を認めている。

サービス側管理 (service-managed)のセションの一例としては、Web ブラウザセションの中でクッキーの生成と管理を行うケースがあげられる。

リクエスタ側の管理 (requestor-managed)のセションの一例としては、トークンを獲得したWebサービスのリクエスタが、そのトークンをある一定の時間内に利用して、失効する前にそのトークンを無効化するケースがあげられる。

サインアウトの概念は、異なったプロファイルが、どのように各々のプロファイルに対してこれらのトランザクションを適用するかを指定することを可能にする。

シングルサインオンの目的は、連携されたドメイン／領界(realm)内でリソースへアクセスするために必要なセキュリティ・トークンを確立させることである。

同様に連携サインアウトは、連携されたドメイン／領界(realm)内に存在するかもしれない、いかなる（メモリ上に）キャッシュされたセション状態情報やセキュリティ・トークンを消去させる場合に利用する。これらの処理を可能にするためには、WS-Federation の機構において、IP が発行した federated sign-in と sign-out メッセージを sign-in/sign-out の処理を認可しているドメイン／領界(realm)内の各 party に対して、送信する必要がある。

（その方法によって、ドメイン／領界(realm)内の IP やサービスプロバイダは、必要ないかなるセットアップやクリーンナップ処理も可能となる。）

補足） realm は、Kerberos 用語で、Kerberos が発行したチケットが有効なドメインを指す用語である。

(5) **属性と仮名**(Pseudonyms)

本仕様では、UDDI のような異なったタイプの属性サービスを利用することができる。

例えば、エンドユーザ（人間）は、以下のようにセットアップするかもしれない。

“ イントラネットにおける私が利用するサービスは、私の last name(姓)にアクセスしてもよい。その他のサービスは、私からの許可無しに利用することはできない。”

属性サービスは、既存のリポジトリを利用してもよい。また、いくつかの組織のレベルやコンテキストを提供するかもしれない。組織の名前空間内において、個人のプリンシパルは、登録されており、XML 中で表現される属性のプロパテイ（本質的には、名前／値の組。名前は、文字列のプロパテイの名前で、値は、XML のエレメントである。）の中では、プリンシパルと関係付けられる。

プリンシパルが、誰に対してどのように情報を公開するかを管理するために、各々の属性に対して、固有のセキュリティ認可の規約やプライバシーに関するポリシーを設定してもよいことに留意することは非常に重要である。

異なった属性サービスは、その属性サービスのポリシードキュメントの中で表現される、異なった機能を持つ。

(6) **仮名サービス**

仮名サービスは、プリンシパルが異なるリソース・サービス又は、別のドメイン／領界(realm)内において、仮名を持つ機能である。

例えば、リクエスタが、Business456.com で彼の主要な ID である “Fred.Jones” として認証されたとする。けれども Fabrikam123.com では、彼は “Freddo”として知られている。

仮名性を保護するために Business456.com は、Fred.Jones がサイン・インしたと

きに異なる ID を発行することができる。その結果、下図の Step3 に図示されるように”anonymous(別名) “ が示される。

Fabrikam123.com は、“ Freddo”に仮名をパッピングすることのできる仮名サービスを提供することによって、Fabrikam123.com(Step4)のサービスを利用しようとするリクエスタのためにローカル名として “Freddo”を設定することができる。

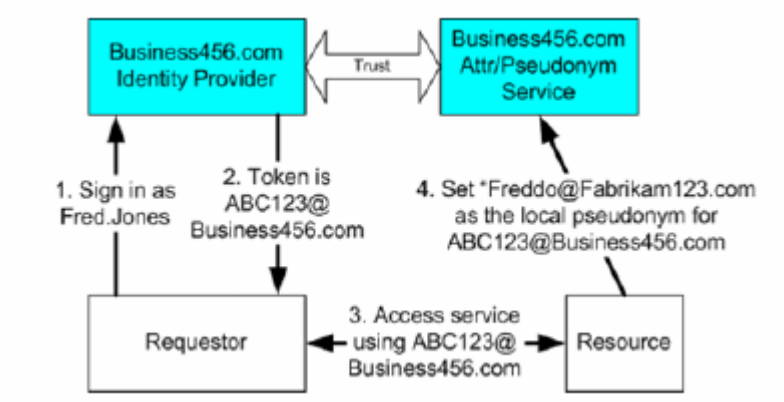


図 4-16 仮名サービス (サイン･イン)

次にリクエスタが、Business456.com IP に対して、サイン・インした場合（次の図の Step1）彼は, XYZ321@Business456.com (下図 Step2)のように新しい ID を割り付けられるかもしれない。Business456.com IP は、前述したようにマッピングする機能を持っているので、

Fabrikam.com における Web サービスは、Fabrikam.com にサービス要求があった場合の

XYZ@321Business456.comが、以前、どのように設定されていたかを遡って認識する。

このケースでは、Freddo@Fabrikam123.com” (下図の Step5)である。

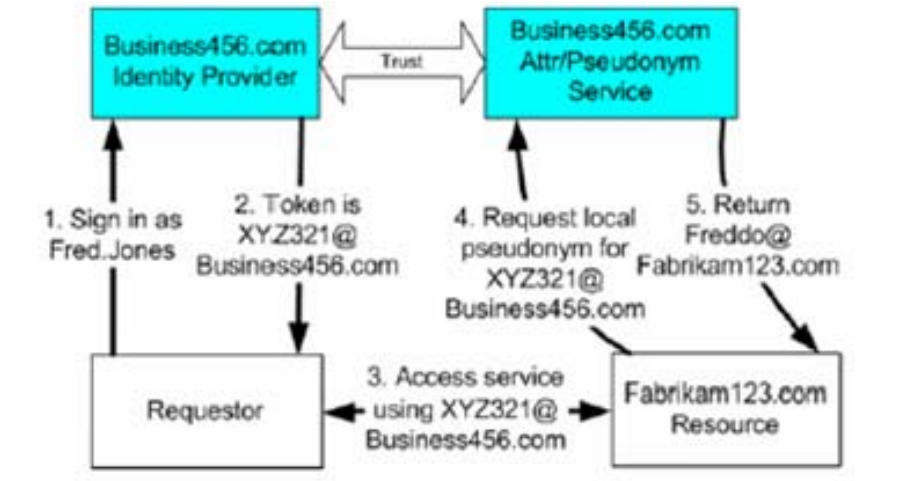


図 4-17 仮名サービス(サービス要求)

仮名サービスは、このようなサービスを可能にする。なぜならそれは、異なったドメイン／領界(realm)のための仮名”XYZ321@Business456.com”を Business456.com で知られる ID に関連付ける back-map 機能を持っている。この back-mapping には、IP と仮名サービス間の信頼関係が確立していることが基本となる。同様にリソースとおそらく IP によって起動される）仮名サービス間にも信頼関係は確立している。

4.3 Liberty Alliance と WS-Federation の相違の概要

本節では以下の文献を参考に、Liberty Alliance と WS-Federation の相違について概要をまとめた。

- Liberty Alliance & WS-Federation: A Comparative Overview
  October 2003
  (Liberty Alliance Project)

4.3.1 仕様項目の相違

仕様項目レベルでは、Liberty の連携フレーム(ID-FF、ID-WSF および ID-SIS)および WS-Federation とでは以下の点は同様のものとなっている

- 基本的なメッセージング・プロトコルのプロファイルを通じて、(これらのクライアントの性質は異なりそうだが)ブラウザおよびスマート・クライアントを識別すること
- セキュリティ・トークンの発行による信頼の仲介業務
- プライバシーにコントロールされる属性共有
- 連携されたサインアウトによる基本的なセッション管理の動作

しかしながら、これらの原理を適用する方法は、アプローチおよび基礎技術において大部分が異なる。下記のマトリックスは、これらの違いをより詳細に示す。

表 4-4 仕様項目の相違

Category	Component	Liberty	WS-Federation +
Linkage	Account Linking	ID-Federation Framework	Yes, through Set messages of Pseudonym Service
Single Sign On	Authentication Request	SAML Request	WS-Trust Token Issuance Request
	Authentication Response	SAML Response	WS-Trust Token Issuance Response
	Assertion	SAML Authentication Statement	Arbitrary tokens
	Authentication Details	Authentication Context can be specified on request and response	
	Profiles	Browser Artifact, Form POST, LEC	Passive and Active with variations
Session	Single Logout initiated by IDP	Yes	Yes
	Single Logout initiated by SP	Yes	Yes
	Session Credentials		WS-SecureConversation
Privacy	Opaque Identifiers	Yes	Optional (can use non-opaque & persistent identifiers)
	Management	NameRegistration protocol	Yes, through Set messages of Pseudonym Service
	Policy for released attributes	Usage Directives	
	Encrypted identifiers and URIs	Yes	
Authorization	Authorization Request	Implicit	WS-Trust
	Authorization Response	Implicit	WS-Trust
	Attributes (roles)		Yes
Trust	Legal Agreements	Can be referenced from Authentication Context	
	Business agreements	Can be referenced from Authentication Context	
	Affiliations	Yes	
	Introduction	See below	
	Token Exchange/Mapping		WS-Trust

（続く）

（続き）

Category	Component	Liberty	WS-Federation +
Security	Message Security	XML Signature/XML Encryption/WS-Security protected messages	XML Signature/XML Encryption/WS-Security protected messages
Metadata	Publishing	DNS & Known location; Can be published in UDDI directory	
	Retrieval	DNS & Known location	
	Schema	ID-WSF Metadata	WS-MetadataExchange
Discovery	Principal IDP	Common Domain cookie	
	Publishing	ID-WSF DiscoveryLockupUpdate	UDDI
	Query	ID-WSF DiscoveryLockupRequest	UDDI
	Security Policy		WS-SecurityPolicy
Introduction	Trust brokering	Yes	WS-Trust
	Notification of Principal Federation	Yes	
	Notification of trust termination	Yes	
Information Sharing	Access	Yes, Identity Service	Attribute Service
	Store		UDDI
	Privacy policy	Privacy Policy Expression Language	WS-Privacy?
	Data manipulation	WSF Data Service Template	Not in WS-Federation, perhaps .Net My Services HSDL
	Data interface	ID-Personal Profile	Not in WS-Federation, perhaps .Net My Services?
	Brokering	Yes	
User Interaction	User Consent	ID-WSF Interaction Service	
	Federation termination	Yes	

（参照：Liberty Alliance & WS-Federation: A Comparative Overview

Appendix: Detailed technical analysis between Liberty Alliance and WS-Federation）

4.3.2 プロトコル上の差異の例

(1) Linkage

- Liberty Alliance Project：

  ID-FF に基づく。

  認証方式は限定せず、アカウント連携、仮名、匿名のアイデンティティ連携を行う。

- WS-Federation：

  (WSF) Pseudonym Service(仮名サービス)

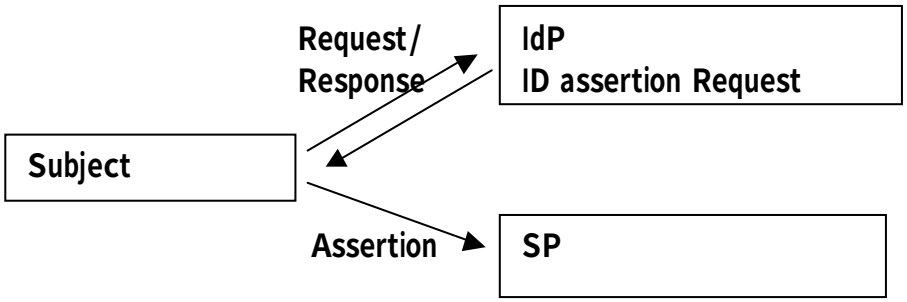
  仮名サービス取得プロトコルメッセージにより、トラストサークルや仲介を経由したアイデンティティ連携

(2) Request & response &assertion

両者ともよく似た方式であるが、Liberty Alliance では SAML のみをベースにしており、WS-Federation では、WS-Trust (Kerberos, SAML, X509v3, XrML)を利用している。

認証サイトが IdP(LAP)サイトであるのに対して、SecurityTokenService を動作させている Policy サーバでのサービスレベルである点が異なる。

- Liberty Alliance Project：



IdPにAuthenticate対する要求

```
<samlp: Request ...>
  <samlp: AttributeQuery>
    <saml: Subject>
      <saml: NameIdentifier
        SecurityDomain="ecom.jp"
        Name="rimap"/>
    </ saml: Subject>

    <saml: AttributeDesignator
      AttributeName="Employee_ ID"
      AttributeNamespace="ecom.jp">
    </ saml: AttributeDesignator>
  </ samlp: AttributeQuery>
</ samlp: Request>
```

SAMLAuthenticationレスポンス

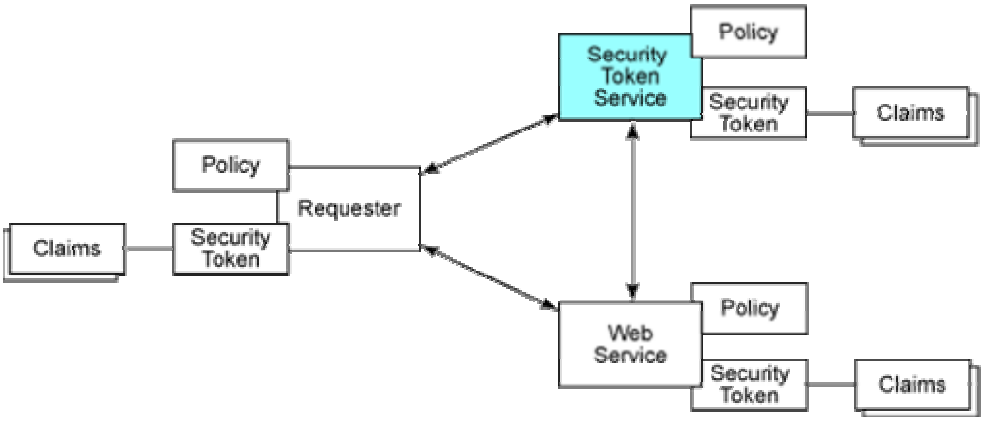
```
<samlp: Response
  MajorVersion="1" MinorVersion="0"
  RequestID="128.14.234.20.90123456"
  InResponseTo="123.45.678.90.12345678"
  StatusCode="Success">
```

SAMLAuthenticationアサーション

```
<saml: Assertion
    MajorVersion="1" MinorVersion="0"
    AssertionID="123.45.678.90.12345678"
    Issuer="Ecom.jp"
    IssueInstant="2002- 01- 14T10: 00: 23Z">

    <saml: Conditions
      NotBefore="2002- 01- 14T10: 00: 30Z"
      NotAfter="2002- 01- 14T10: 15: 00Z" />

    <saml: AuthenticationStatement
    AuthenticationMethod="Password"
    AuthenticationInstant="2001- 01- 14T10: 00: 20Z">

      <saml: Subject>
        <saml: NameIdentifier
          SecurityDomain="ecom.jp"
          Name="rimap" />
      </ saml: Subject>
    </ saml: AuthenticationStatement>
  </ saml: Assertion>
</ samlp: Response>
```

- WS-Federation (WS-Trust Token Issuance Request)

（参照元：Web Services Trust Language (WS-Trust) Version 1.0）



Security Tokenサービスに対する要求

```
<S:Envelope xmlns:S="..." xmlns=".../secext" xmlns:wsu=".../utility>
  <S:Header>
    ...
    <Security>
      <UsernameToken wsu:Id="myToken">
        <Username>NNK</Username>
        <Nonce>FKJh...</Nonce>
        <wsu:Created>2001-10-13T09:00:00Z </wsu:Created>
      </UsernameToken>
      <ds:Signature xmlns:ds="...">
        ...
      </ds:Signature>
    </Security>
    ...
  </S:Header>
  <S:Body wsu:Id="req">
    <RequestSecurityToken>
      <TokenType>wsse:X509v3</TokenType>
      <RequestType>wsse:ReqIssue</RequestType>
      <Base>
        <Reference URI="#myToken"/>
      </Base>
    </RequestSecurityToken>
  </S:Body>
</S:Envelope>
```

セキュリティ・トークンResponseメッセージ

```
<S:Envelope xmlns:S="..." xmlns=".../secext" xmlns:wsu=".../utility">
  <S:Header>
    ...
    <Security>
      <BinarySecurityToken
       wsu:Id="myToken2"
       ValueType="wsse:X509v3"
       EncodingType="wsse:Base64Binary">
       DFJHuedsujfnrnv45JZc0...
      </BinarySecurityToken>
      <ds:Signature xmlns:ds="...">
        ...
      </ds:Signature>
    </Security>
    ...
  </S:Header>
  <S:Body>
  <RequestSecurityTokenResponse>
    <RequestedSecurityToken>
      <BinarySecurityToken
       ValueType="wsse:X509v3"
       EncodingType="wsse:Base64Binary">
       MIIEZzCCA9CgAwIBAgIQEmtJZc0...
      </BinarySecurityToken>
    </RequestedSecurityToken>
    <RequestedProofToken>
      <ds:KeyInfo xmlns:ds="...">
        <ds:KeyValue>
          ...
        </ds:KeyValue>
      </ds:KeyInfo>
    </RequestedProofToken>
  </RequestSecurityTokenResponse>
  </S:Body>
</S:Envelope>
```

SAMLを使ったAssertionセキュリティ・トークンの例

```
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <saml:Assertion
            MajorVersion="1"
            MinorVersion="0"
            AssertionID="SecurityToken-ef375268"
            Issuer="elliotw1"
            IssueInstant="2002-07-23T11:32:05.6228146-07:00"
           xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
        ...
      </saml:Assertion>
      ...
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

(3) Authentication Details

- Liberty Alliance Project：

  Liberty の認証オーソリティは、追加の Authentication Context 情報を含めることができる。

  それらには、identification, technical Protection, Operational Protection (security audit, records archival), authentication Method (smartcard ..), Governing Agreement などに分類できる項目が含まれる。

- WS-Federation

  この項目は規定されていない。

(4) Profile

- Liberty Alliance Project：

  ブラウザのアーティファクト、フォームベースの POST、WML(ECMAScript) ベースの LEC プロファイル。（Bindings Profiles）

- WS-Federation

  バインディングプロファイルとして、Passive 要求者 Profile と Active Profile とがある。

(5) Session

SingleLogout については IDP 主導、SP 主導のいずれにも LAP、WSF ともに存在する。

Session　Credential については WSF のみに定義がある。

WS-SecurityConversation これは複数のメッセージでセッションを確立、維持するために当事者間でセキュリティコンテキスト、セキュリティ・トークン（秘密鍵など）を交換または、作成し共有する。これにしたがって署名されたメッセージをおのおの検証し、メッセージの確からしさと通信の安全性を保証する仕組みとなる。

なお、Liberty Alliance Project ではメッセージでなく HTML/SSL などによるセッション維持によって安全性を保証している。

「引用元」

本章の引用元および参照資料を下記に記す。

「4.1」

- Liberty アーキテクチャ概要 バージョン 1.1 [LibArch11]
- ID-FF アーキテクチャ概要 [LibArch12]

「4.2」

- Federation of Identities in a Web Services World [WSF-WP]

「4.3」

- Liberty Alliance & WS-Federation: A Comparative Overview [LibWsfComp]

- Web Services Trust Language (WS-Trust) Version 1.0 [WST]

5 今後の課題

分散システムの認証の問題は SAML の登場によって新たな関心が寄せられるようになって来た。それは従来の SSO が独自仕様で実現されてきたのに対して標準仕様の確立によって相互運用性に大きな進展を見せているからである。しかし SAML の重要性は、単に相互運用性を進展させただけでなく、プライバシーの保護の視点を強化しアイデンティティの分散管理と連携の新しい仕組みを導入したことにある。ビジネスにおけるアイデンティティの確認は、オンラインにおける本人認証だけではなく本人の役割や資格などの属性情報を伴ってはじめて資源などへのアクセスの認可につながってくる。SAML は認証、属性、認可の仕組みを柔軟に連携させるものである。このようなセキュアーで安心できる分散環境における認証の仕組みは、新しいアプリケーションの発展を促す可能性を秘めている。

Globus Alliance[3]のグリッドコンピューティングの認証セキュリティにも SAML が導入され、科学技術グリッドコンピューティングからビジネスグリッドコンピューティングの可能性も開けてきた。SAML をベースにアイデンティティの連携を強化した Liberty Alliance Project は、Phase 2 では Web サービス環境のフレームワーク（ID-WSF）や ID サービスインタフェース（ID-SIS）の仕様を策定し、より広範なアプリケーションの利用環境を整備してきている。また IBM、Microsoft らによる同様な Web サービス環境での ID 連携 WS-Federation も提案されている。今後どの仕様が市場を獲得していくのかは分からないが、共にアイデンティティ保護を強化した新しい認証の仕組みとアプリケーションが主流になっていくのは間違いない。

現在の日本の電子政府の電子申請アプリケーションでは電子署名を基本として作られている。しかし、これらのアプリケーションでは必ずしも電子署名を必須としなくても良い場合が多いとも考えられる。電子申請や民間の e-ビジネスでは利用者がサービス提供者のシステムにログインし、各種のサービスを受ければよいからである。ただしこのとき相手認証は必須のものとなる。また電子申請などをワンストップサービスで行うためには SSO が必須の提供技術となる。このときプライバシー保護の観点から利用者のアイデンティティを一元管理するのではなく各省や各部門で分散管理し、アイデンティティの連携機能を設ければ安全で便利なサービスが可能となる。

米国政府は最近 e-Authentication Initiative を作り国民サービスのために SAML や Liberty を想定したアイデンティティ連携の e-Authentication 基盤整備の準備を始めてい

る[eAuth][NISTeAuth]。ここは用途に応じてポリシーに応じて Authentication Context によって各種の認証技術を選択できるようにとしている。リスクの高いサービスには認証は PKI で行いリスクの低いサービスには ID,パスワードでもよいというように各種の認証レベルを分けて考えることをこの e-Authentication 基盤の基本に据えようとしている。

ここでも考え方は 1 つの手法にロックインするのではなく、ポリシーに応じて多様性を基本とする考え方をとっている。この方法によれば結果的に迅速な展開と総合的なコストを削減できるように思われる。日本の電子政府アプリケーションでもシステムは用途によって選択するという多様性を身に着けるべきである。

今回 TF7 の作業として電子政府や医療分野および EC 分野についての SAML のユースケースを検討してきた。これらのユースケースで新たなアプリケーションに SAML が適用できることの例を示すことができたのではないかと思う。また本作業では新しい流れとしての Liberty や WS-Federation の比較紹介も行った。

SAML などの新しい ID 連携の仕組みやそのアプリケーションは始まったばかりである。OASIS では SAML 1.0 仕様の実際の展開や Liberty などの活動の経験を踏まえて現在 SAML 2.0 の仕様を策定中である。今後これらの技術をどのように実際の電子政府に適用していくのか、あるいは民間の電子商取引に適用していくのかについてのガイドラインを策定していくことが必要になろう。次年度の課題として SAML やその関連仕様について、その利用環境、セキュリティレベル、PKI との関連、認証・属性情報の管理などの利用ガイドラインをまとめていきたい。

[3] http://www.globus.org/

付録A　Assertions and Protocol for the OASIS
Security Assertion Markup Language(SAML) V1.1

1. Introduction

1.1. Schema Organization and Namespaces

(1) OASIS SAML の名前空間は以下のとおりである。

アサーション定義の名前空間

urn:oasis:names:tc:SAML:1.0:assertion

プロトコル定義の名前空間

urn:oasis:names:tc:SAML:1.0:protocol

(2) 文字列と URI の値はスキーマの既定義型である xsd:string と xsd:anyURI を使用する。文字列には最低一つの非空白文字を含まなくてはいけない。また、特に指定しない限り URI にも最低一つの非空白文字を含まなくてはいけない。URI については絶対パス指定をすることを強く推奨する。
(3) 時刻にはスキーマの既定義型である xsd:dateTime を使用し、UTC で表現しなくてはいけない。
(4) アサーション、リクエスト、レスポンスに xsd:ID 型の識別子が使用されているが、その値は以下の特性を満たさなければならない。

- 識別子を割り当てる者は、自分自身や他の当事者が、異なるデータに誤って同じ識別子を割り当ててしまう可能性が無視できるほど小さいことを保障しなくてならない。
- 識別子が定義されているデータには、必ず一つ識別子が宣言されていなくてはならない。

識別子の参照としては xsd:IDREF 型ではなく、xsd:NCNAME 型を使用する。なぜなら、参照される識別子の定義が、そのドキュメントには存在しない場合があるからである。たとえば、リクエストに指定されている識別子を、レスポンスで参照するような場合が該当する。
(5) SAML の文字列値の比較はバイナリで行われる。特に、大文字小文字や空白文字の正規化タイミング、数字や通貨単位の地域化などに依存してはならない。

1.2. SAML Concepts

SAML アサーションは XML を使って記述されており、以下の情報を運ぶことができる。

① 過去にそのサブジェクトによって実行された認証の結果(Authentication Assertion)
② サブジェクトの属性(Attribute Assertion)
③ サブジェクトが特定のリソースにアクセス可能であるかどうかについての認証の結果(Authorization Decision Assertion)

アサーションは SAML オーソリティによって発行される。SAML オーソリティとは、以下の 3 つのことである。

① 認証オーソリティ(Authentication Authority)
② 属性オーソリティ(Attribute Authority)
③ 認可決定オーソリティ(Policy Decision Point)

SAML ではアサーションのフォーマットだけでなく、クライアントが SAML オーソリティにアサーションをリクエストしレスポンスをもらうための、ＸＭＬで記述されたデータフォーマットも規定している。実際に通信する際には、上記データをいろいろな通信手段に乗せて通信する。現在規定している通信手段は SOAP over HTTP であるが、その他の手段も可能な仕組みになっている。
SAML オーソリティはアサーションを作成するために、リクエストの中のオーソリティへの入力として渡されるアサーションなど、いろいろなリソースを利用することができる。したがって、SAML クライアントは常にアサーションを消費するが、SAML オーソリティはアサーションを生成する場合もあるし、消費する場合も

ある。

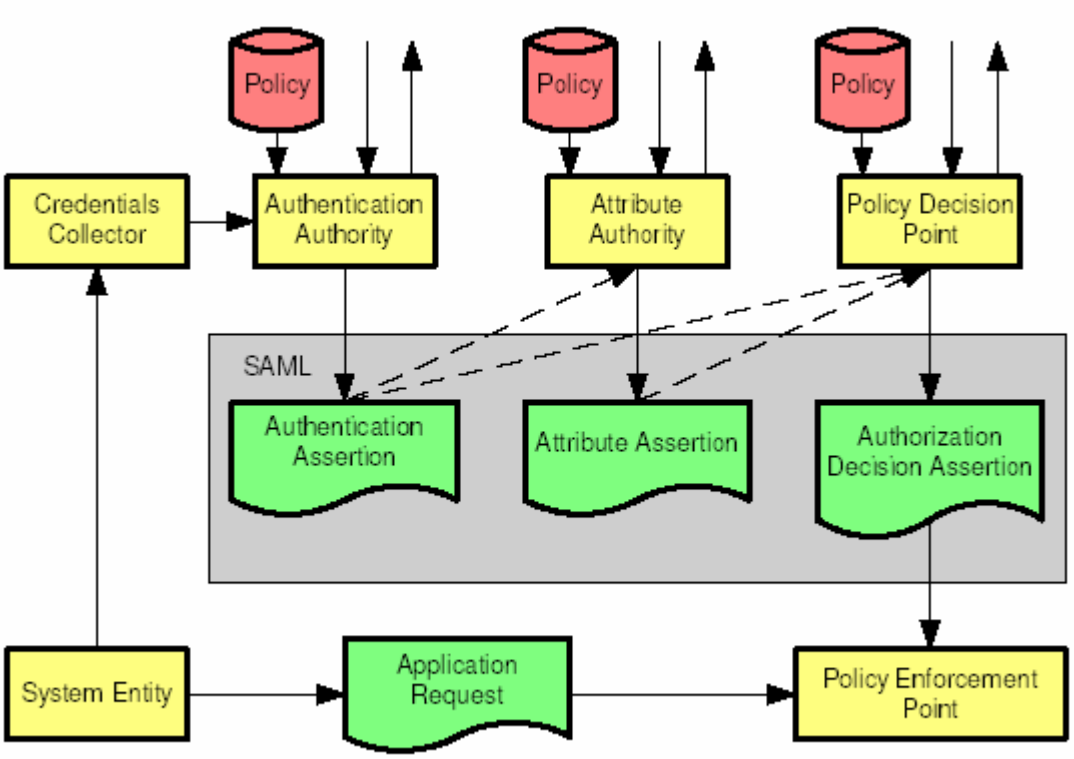


Figure 1 The SAML Domain Model

(出典："Assertions and Protocol for the OASIS Security Assertion Markup Language(SAML) V1.1 Committee Specification, 1 July 2003")

2. SAML Assertions

Assertion(saml:AssertionType 型)

要素			型	内容
sequence	Conditions ?		saml:ConditionsType	アサーションの有効性を評価するときに考慮する条件。
	Advice ?		saml:AdviceType	アサーションに関連する付加情報。
	choice	Statement	saml:StatementAbstractType	ステートメント（拡張スキーマで定義）。
		SubjectStatement	saml:SubjectStatementAbstractType	サブジェクトステートメント（拡張スキーマで定義）。
		AuthenticationStatement	saml:AuthenticationStatementType	認証ステートメント。
		AuthorizationDecisionStatement	saml:AuthorizationDecisionStatementType	認可決定ステートメント。
		AttributeStatement	saml:AttributeStatementType	属性ステートメント。
	ds:Signature ?		ds:Signature	アサーションを認証（authenticates）する XML 署名。

属性	型	内容
MajorVersion	xsd:integer	メジャーバージョン。SAML1.1 では「1」。
MinorVersion	xsd:integer	マイナーバージョン。SAML1.1 では「1」。
AssertionID	xsd:ID	アサーションの識別子。一意性をもつこと。
Issuer	xsd:string	アサーションを生成した SAML オーソリティ。URI 参照を使う場合あり。
IssueInstant	xsd:dateTime	アサーションの発行日時。

Conditions(saml:ConditionsType 型)

要素		型	内容
choice	AudienceRestrictionCondition *	saml:AudienceRestrictionConditionType	アサーションが特定の受信者（audience）に受信（addressed）されることを指定する。
	DoNotCacheCondition *	saml:DoNotCacheConditionType	アサーションがすぐに使われる必要があり（SHOULD）、将来の利用のために保有してはならない（MUST NOT）ことを指定する。
	Condition *	saml:ConditionAbstractType	拡張スキーマによる新しい条件の定義を可能にするための拡張点を提供する。

属性	型	内容
NotBefore ?	xsd:dateTime	アサーションが有効になる日時。
NotOnOrAfter ?	xsd:dateTime	アサーションが無効になる日時。省略した場合は無期限。

AudienceRestrictionCondition(saml:AudienceRestrictionConditionType 型)

要素		型	内容
seq	Audience *	xsd:anyURI	アサーションが特定の受信者（audience）に受信（addressed）されることを指定する。

DoNotCacheConditionType(saml:DoNotCacheConditionType 型)

この型は空内容モデルである。

Condition(saml:ConditionAbstractType 型)

この型は空内容モデルである。

Adivice(saml:AdviceType 型)

要素		型	内容
choice	AssertionIDReference *	xsd:NCName	付加情報としてのアサーションの参照。
	Assertion *	saml:AssertionType	付加情報としてのアサーションの値。
	（要素名なし） *	xsd:any	付加情報としての他の名前空間の要素。

Statement(saml:StatementAbstractType 型)

この型は空内容モデルである。

SubjectStatement(saml:SubjectStatementAbstractType 型)

要素	型	内容
Subject	saml:SubjectType	対象となるサブジェクト。

Subject(saml:SubjectType 型)

要素			型	内容
choice	seq	NameIdentifier	saml:NameIdentifierType	名前とセキュリティドメインによるサブジェクトの識別。
		SubjectConfirmation ?	saml:SubjectConfirmationType	サブジェクトが認証されるための情報。
	SubjectConfirmation		saml:SubjectConfirmationType	同上。

NameIdentifier(NameIdentiferType 型)

この型は単純内容モデル（xsd:string 型）である。

属性	型	内容
NameQualifier ?	xsd:string	サブジェクトの名前の修飾。セキュリティドメインもしくは管理ドメインが使われる。アカウントの連携時に名前衝突を防ぐために使われ

		る。
Format ?	xsd:anyURI	<NameIdentifier>のフォーマットを表す URI 参照。

SubjectConfirmation(SubjectConfirmationType 型)

要素		型	内容
bas	ConfirmationMethod +	xsd:anyURI	認証に使われたプロトコル。[SAMLBind] にて定義している。
	SubjectConfirmationData ?	xsd:anyType	認証プロトコルで使われる付加的な認証情報。
	ds:KeyInfo ?	ds:KeyInfo	サブジェクトの暗号鍵へのアクセス方法。

AuthenticationStatement(AuthenticationStatementType 型)

要素		型	内容
bas	Subject	saml:SubjectType	対象となるサブジェクト。
	SubjectLocality ?	saml:SubjectLocalityType	認証されたシステム自体の DNS ドメイン名および IP アドレス。
	AuthorityBinding *	saml:AuthorityBindingType	AuthenticationStatement を生成する SAML 検証者を示す。将来のバージョンでは削除される予定。

属性	型	内容
AuthenticationMethod	xsd:anyURI	認証タイプを指定する URI 参照。セクション 7.1 を参照。
AuthenticationInstant	xsd:dateTime	認証された時刻。

SubjectLocality(SubjectLocalityType 型)

この型は空内容モデルである。

属性	型	内容
IPAddress ?	xsd:string	認証されたエンティティの IP アドレス。
DNSAddress ?	xsd:string	認証されたエンティティの DNS ドメイン名。

AuthorityBinding(AuthorityBindingType 型)

この型は空内容モデルである。

属性	型	内容
AuthorityKind	xsd:QName	AuthorityBinding 要素が指定する SAML オーソリティへの問い合わせにおける SAML プロトコルの型。
Location	xsd:anyURI	SAML オーソリティへ通信する方法（URI 参照）。HTTP ベースの場合は Web の URL となり、SMTP をベースとする場合は「mailto:」スキーマを使用する。
Binding	xsd:anyURI	SAML オーソリティとの通信に使用する SAML プロトコル結合を識別す

		る URI 参照。

AttributeStatement(AttributeStatementType 型)

要素		型	内容
seq	Subject	saml:SubjectType	対象となるサブジェクト。
	Attribute *	saml:AttributeType	サブジェクトの属性。

AttributeDesignator(AttributeDesignatorType 型)

この型は空内容モデルである。

属性	型	内容
AttributeName	xsd:string	属性名。
AttributeNamespace	xsd:anyURI	属性の名前空間。

Attribute(AttributeType 型)

要素	型	内容
AttributeValue +	xsd:anyType	属性値。

属性	型	内容
AttributeName	xsd:string	属性名。
AttributeNamespace	xsd:anyURI	属性の名前空間。

AuthorizationDecisionStatement(AuthorizationDecisionStatementType 型)

要素		型	内容
seq	Subject	saml:SubjectType	対象となるサブジェクト。
	Action +	saml:ActionType	認可されたアクション。
	Evidence ?	saml:EvidenceType	認可の判断の元となったアサーション。

属性	型	内容
Resouce	xsd:anyURI	アクセス認可の対象リソース。
Decision	saml:DecisionType	リソースへのアクセス可否。

Action(ActionType 型)

この型は単純内容モデル（xsd:string 型）である。

属性	型	内容

Namespace ?	xsd:anyURI	アクション名の名前空間。

Evidence(EvidenceType 型)

要素		型	内容
choice	AssertionIDReference	xsd:NCName	証拠として参照するアサーションの ID。
	Assertion	saml:AssertionType	証拠とするアサーションそのもの。

Decision(DecisionType 型)

この型は単純内容モデル（xsd:string 型）である。

値(enumeration)	内容
Permit	アクションは許可される。
Deny	アクションは禁止される。
Indeterminate	不定。アクションの許可／禁止を決定できない。

3. SAML Protocol

3.1. Requests

RequestAbstractType

要素		型	内容
seq	RespondWith *	xsd:QName	リクエスタが利用可能なレスポンスの型。
	ds:Signature ?	ds:Signature	リクエストを認証する XML 署名。

属性	型	内容
RequestID	xsd:ID	リクエストの識別子。
MajorVersion	xsd:integer	メジャーバージョン。SAML1.1 では「1」。
MinorVersion	xsd:integer	マイナーバージョン。SAML1.1 では「1」。
IssueInstant	xsd:dateTime	リクエストの発行時刻。

Request(RequestType 型)

要素			型	内容
sequence	seq	RespondWith *	xsd:QName	リクエスタが利用可能なレスポンスの型。
		ds:Signature ?	ds:Signature	リクエストを認証する XML 署名。
	... q	Query	samlp:QueryAbstractType	拡張スキーマが照会の新しい型を定義できる拡張点。

		SubjectQuery	samlp:SubjectQueryAbstractType	拡張スキーマが1つのSAMLサブジェクトを指定する照会の新しい型を定義できる拡張点。
		AuthenticationQuery	samlp:AuthenticationQueryType	認証情報を照会する。
		AttributeQuery	samlp:AttributeQueryType	属性情報を照会する。
		AuthorizationDecisionQuery	samlp:AuthorizationDecisionQueryType	認証決定の照会を行う。
		AssertionIDReference +	xsd:NCName	その AssertionID 属性の値を参照することによってアサーションをリクエストする。
		AssertionArtifact +	xsd:string	それを表すアサーションアーティファクトを与えることによってアサーションをリクエストする。

属性	型	内容
RequestID	xsd:ID	リクエストの識別子。
MajorVersion	xsd:integer	メジャーバージョン。SAML1.1では「1」。
MinorVersion	xsd:integer	マイナーバージョン。SAML1.1では「1」。
IssueInstant	xsd:dateTime	リクエストの発行時刻。

Query(QueryAbstractType 型)

この型は空内容モデルである。

SubjectQuery(SubjectQueryAbstractType 型)

要素	型	内容
Subject	saml:SubjectType	対象となるサブジェクト。

AuthenticationQuery(AuthenticationQueryType 型)

要素	型	内容
Subject	saml:SubjectType	対象となるサブジェクト。

属性	型	内容
AuthenticationMethod ?	xsd:anyURI	リクエストの識別子。

AttributeQuery(AttributeQueryType 型)

要素		型	内容
seq	Subject	saml:SubjectType	対象となるサブジェクト。
	AttributeDesignator *	saml:AttributeDesignator	値を返されるべき属性。

属性	型	内容
Resource ?	xsd:anyURI	属性の照会がリソースに関連する特定のアクセスリクエストを評価するために行われることを指定する。

AuthorizationDecisionQuery(AuthorizationDecisionQueryType 型)

	要素	型	内容
sequence	Subject	saml:SubjectType	対象となるサブジェクト。
	Action +	saml:Action	認証がリクエストされる動作。
	Evidence	saml:Evidence	SAML オーソリティがその認証決定において依存することができるアサーションの組。

属性	型	内容
Resource	xsd:anyURI	認証がリクエストされるリソースを示す URI 参照。

3.2. Responses

ResponseAbstractType

要素	型	内容
ds:Signature ?	ds:Signature	レスポンスを認証する XML 署名。

属性	型	内容
ResponseID	xsd:ID	レスポンスの識別子。
InResponseTo ?	xsd:NCName	リクエスト識別子の参照。
MajorVersion	xsd:integer	メジャーバージョン。SAML1.1 では「1」。
MinorVersion	xsd:integer	マイナーバージョン。SAML1.1 では「1」。
IssueInstant	xsd:dateTime	レスポンスの発行時刻。
Recipient ?	xsd:anyURI	本レスポンスの所期の受信者。

Response(ResponseType 型)

	要素	型	内容
seq	ds:Signature ?	ds:Signature	レスポンスを認証する XML 署名。
	Status	samlp:StatusType	対応するリクエストのステータスを表すコード。
	Assertion *	saml:Assertion	アサーションを値によって指定。

属性	型	内容

ResponseID	xsd:ID	レスポンスの識別子。
InResponseTo ?	xsd:NCName	リクエスト識別子の参照。
MajorVersion	xsd:integer	メジャーバージョン。SAML1.1では「1」。
MinorVersion	xsd:integer	マイナーバージョン。SAML1.1では「1」。
IssueInstant	xsd:dateTime	レスポンスの発行時刻。
Recipient ?	xsd:anyURI	本レスポンスの所期の受信者。

Status(StatusType型)

要素		型	内容
seq	StatusCode	samlp:StatusCodeType	対応するリクエストのステータスを表すコード。
	StatusMessage ?	xsd:string	オペレータに返されることのあるメッセージ。
	StatusDetail ?	samlp:StatusDetailType	エラー状況に関する追加情報。

StatusCode(StatusCodeType型)

要素	型	内容
StatusCode ?	samlp:StatusCodeType	対応するリクエストのステータスを表すコード。

属性	型	内容
Value	xsd:QName	ステータスコードの値。

StatusDetail(StatusDetailType型)

要素	型	内容
any	namespace=”##any”	どのようなＸＭＬ形式のデータでも設定可能である。

4. SAML Versioning

SAML のバージョンには Major Version と Minor version があり、*Major.Minor* という形式で表される。

4.1. SAML Specification Set Version

SAML の規格に変更が生じた場合、バージョンが変更されることになるが、一般的には、変更後に以前の版に対して互換性がある場合はマイナーバージョンアップ、そうでない場合はメジャーバージョンアップとなる。以前の版に対して互換性があるとは、以前の版で正しいメッセージが新しい版でも正しいとみなされる、ということである。ただし、SAML V1.1 は一箇所 SAML V1.0 に対して非互換な修正がなされていることに注意してほしい。

(1) スキーマには<schema>要素の version 属性で SAML のバージョンが明示されている。
(2) SAML アサーションのバージョンは<Assertion>要素に属性でバージョンを設定できるようになっている。アサーションのバージョンには以下のルールが適用される。

- SAML オーソリティは自身がサポートしていないバージョン番号が設定されているアサーションを発行してはいけない。
- SAML 署名検証者は自身がサポートしていないメジャーバージョン番号を持ったアサーションを処理してはいけない。
- SAML 署名検証者は自身がサポートしているより大きなマイナーバージョン番号を持ったアサーションを処理してもよいし、拒否してもよい。しかし、同じメジャーバージョン番号を持った全てのアサーションは共通の処理規則と意味を持っているので、実装はそれを同じように取り扱うことができる。

(3) SAML プロトコルの<Request>要素と<Response>要素には属性でバージョンを設定できるようになっている。リクエストのバージョンには以下のルールが適用される。

- SAML リクエスタはリクエスタとレスポンダの両方がサポートしている最も高いバージョンでリクエストを出すべきである。
- もし SAML リクエスタが、レスポンダがサポートしているバージョンを知らなかった場合は、レスポンダはリクエスタがサポートしている最高のバージョンのリクエストを処理できると仮定するべきである。
- SAML リクエスタはレスポンダがサポートしていないバージョンのリクエストを出すべきではない。
- SAML レスポンダはレスポンダがサポートしていないメジャーバージョンのリクエストを拒否しなくてはならない。
- SAML レスポンダは自身がサポートしているより大きなマイナーバージョン番号を持ったリクエストを処理してもよいし、拒否してもよい。しかし、同じメジャーバージョン番号を持った全てのリクエストは共通の処理規則と意味を持っているので、実装はそれを同じように取り扱うことができる。

レスポンスのバージョンには以下のルールが適用される。

- SAML レスポンダは対応するリクエストに設定されたバージョン番号よりも大きなバージョン番号を持ったレスポンスを出してはならない。
- SAML レスポンダは RequestVersionTooHigh のエラーを通知するのではない限り、対応するリクエストに設定されたメジャーバージョン番号よりも小さなメジャーバージョン番号を持ったレスポンスを出してはならない。

(4) 一般的には、あるメジャーバージョンのアサーションは同じメジャーバージョンのレスポンスに入る。そのことは SAML プロトコルスキーマへの SAML アサーション名前空間のインポートで許されている。将来的には、メジャーバージョン間の許される組み合わせについて明示することになる。

4.2. SAML NamespaceVersion

SAMLの名前空間を表すURIにはどこかに*Major.Minor*の形式のバージョン情報を含んでいる。このURIの中のバージョン情報は、その名前空間が最初に導入されたときの規格の番号と同じである。一般的にはメジャーバージョン番号が同一の名前空間は、異なるマイナーバージョン番号の規格にまたがって有効であると考えてよい。新しい名前空間を導入するときに、古い名前空間の定義を置換してもよいが、そういったケースはまれであり、そのような場合には、メジャーバージョンが変わらない限り、古い名前空間の定義も有効のままであると考えてよい。

5. SAML and XML Signature Syntax and Processing

SAMLアサーション、リクエスト、レスポンスはデジタル署名をつけることができる。デジタル署名には以下の利点がある。

- SAMLオーソリティが署名したアサーションは以下をサポートする
  - アサーションの完全性
  - SAML署名検証者に対するSAMLオーソリティの認証
  - もし署名がSAMLオーソリティの公開鍵ペアで行われているのであれば、SAMLオーソリティによる否認不可
- メッセージの送信者が署名したSAMLリクエスト／レスポンスメッセージは以下をサポートする。
  - メッセージの完全性
  - 受信者に対するメッセージ送信者の認証
  - もし署名が送信者の公開鍵ペアで行われているのであれば、送信者による否認不可

SAMLにおいてはデジタル署名が常に求められているわけではない。たとえば、以下のような場合には求められていない。

- 署名されていないアサーションが、署名されているレスポンスに入っているような場合。ただし、この場合は、アサーション部分の署名の検証にレスポンス全体の署名の検証が必要になるので、注意が必要である。
- SAML署名検証者やSAMLリクエスタが、アサーションやメッセージをSAMLオーソリティやSAMLレスポンダから直接セキュアな通信路を通して受け取る場合。アサーションやメッセージを送信するSAMLオーソリティやSAMLレスポンダはデジタル署名以外の方法で署名検証者に認証されている必要がある。

デジタル署名は特に以下の場合には使用することを推奨する。

- SAML署名検証者がSAMLオーソリティ以外のエンティティからSAMLアサーションを受け取る場合。
- 受信者が送信者以外のエンティティからSAMLプロトコルメッセージを受け取る場合。

以下にSAMLアサーションやプロトコルメッセージにデジタル署名をする際の一般的な決まりごとを述べる。

(1) 特に断りがない限り、署名の際にはEnveloped XML Digital Signatureを使用する。
(2) SAMLを処理するソフトウェアは、"http://www.w3.org/2000/09/xmldsig#rsa-sha1"で識別される、RSAによる署名と署名検証をサポートしなくてはならない。
(3) SAMLアサーションやプロトコルメッセージが署名される場合には、そのルート要素（<Assertion>,<Request>,<Response>）に識別子を割り当てなくてはならない。また、署名には署名対象のルート要素の識別を設定した<ds:Reference>要素が一つだけ設定されていなくてはならない。
(4) <ds:SignedInfo>の<ds:CononicalizationMethod>要素と<ds:Transform>アルゴリズムにはExclusive Canonicalizationを使用するべきである。
(5) Transformsには"enveloped signature transforms(http://www.w3.org/2000/09/xmldsig#enveloped-signature)"か、"exclusive canonicalization transforms(http://www.w3.org/2001/10/xml-exc-c14n#またはhttp://www.w3.org/2001/10/xml-exc-c14n#WithComments)"のいずれかを使用するべきである。署名の検証の際、上記以外のtransformアルゴリズムが使われていた場合には、不正であるとして拒絶してもよい。もし、検証者が上記以外のtransformアルゴリズムを受け入れるのであれば、

検証者はSAMLメッセージの全ての内容が署名に含まれていることを保証しなくてはならない。
(6) <ds:KeyInfo>はあってもなくてもよい。
(7) 署名がされていることによってアサーションのステートメントの意味に影響は及ぼさない。
(8) 署名フォーマットはSAML V1.0と互換性はない。これは、V1.0では署名対象のSAML contentを指定するためにURI referenceが使用できない、という制限があったが、V1.1ではその制限が取り除かれた、という点である。これは、V1.0では、署名対象の識別子として、xsd:ID型の属性ではなく、xsd:string型からSAMLの名前空間内で派生させたsaml:IDType型の値を使用していたためである。SAML V1.1ではsaml:IDType型は廃止され、xsd:ID型を使用している。

6. SAML Extensions

SAMLのスキーマ定義には拡張性がある。SAMLアサーションを拡張している例として、Liberty Protocols and Schema Specificationがある。
SAMLのスキーマでは代替の使用を制限してはいないので、全てのSAMLの要素を代替グループのヘッド要素として使用することができる。また、全てのSAMLのデータ型は"final"と指定されていないので、拡張したり制限したりして新しい型を派生させることが可能である。

6.1. Assertion Schema Extension

以下のSAMLアサーションの要素は抽象型として定義されているため派生型の基底型としてしか使用できず、意図的に拡張スキーマの拡張点として使えるようになっている。

- <Condition>
- <Statement>
- <SubjectStatement>

以下の要素はSAMLの一部として直接使用することができるが、拡張することもできる。

- <AuthenticationStatement>
- <AuthorizationDecisionStatement>
- <AttributeStatement>
- <AudienceRestrictionCondition>

以下の要素はSAMLの要素を拡張するのではなく、任意の名前空間の要素を設定できる拡張として使用することができる。

- <AttributeValue>
- <Advice>

6.2. Protocol Schema Extension

以下のSAMLプロトコルの要素は抽象型として定義されているため派生型の基底型としてしか使用できず、意図的に拡張スキーマの拡張点として使えるようになっている。

- <Query>
- <SubjectQuery>

以下の要素はSAMLの一部として直接使用できるが、拡張することもできる。

- <Request>
- <AuthenticationQuery>
- <AuthorizationDecisionQuery>
- <AttributeQuery>
- <Response>

7. SAML-Defined Identifiers

7.1. Authentication Method Identifiers

<AuthenticationStatement>の AuthenticationMethod 属性や、<SubjectConfirmationMethod>要素に使用できる認証メソッドの識別子を示す。

メソッド	概要	識別子
Password	認証をパスワードで行う。	urn:oasis:names:tc:SAML:1.0:am:password
Kerberos	認証を Kerberos プロトコルで行う。	urn:ietf:rfc:1510
SRP	認証を RFC2945 の Secure Remote Password protocol で行う。	urn:ietf:rfc:2945
Hardware Token	認証を何らかのハードウェアトークンを使用して行う。	urn:oasis:names:tc:SAML:1.0:am:HardwareToken
SSL/TLS クライアント認証	認証を SSL または TLS の証明書ベースのクライアント認証で行う。	urn:ietf:rfc:2246
X.509 公開鍵	認証を X.509 の PKI で認証された公開鍵ベースの方法で行う。	urn:oasis:names:tc:SAML:1.0:am:X509-PKI
PGP 公開鍵	認証を PGP web of trust で認証された公開鍵ベースの方法で行う。	urn:oasis:names:tc:SAML:1.0:am:PGP
SPKI 公開鍵	認証を SPKI PKI で認証された公開鍵ベースの方法で行う。	urn:oasis:names:tc:SAML:1.0:am:SPKI
XKMS 公開鍵	認証を XKMS trust service で認証された公開鍵ベースの方法で行う。	urn:oasis:names:tc:SAML:1.0:am:XKMS
XML 署名	認証を XML デジタル署名で行う。	urn:ietf:rfc:3075
未定義	認証を未定義の方法で行う。	urn:oasis:names:tc:SAML:1.0:am:unspecified

7.2. Action Namespace Identifiers

以下にリソースに対して実行できる、<Action>要素の Namespace 属性に使用可能なアクションの共通セットの識別子を示す。

セット	アクション	概要	識別子
Read/Write/Execute/Delete/Control	Read	サブジェクトはリソースを読み出すことができる。	urn:oasis:names:tc:SAML:1.0:action:rwedc
	Write	サブジェクトはリソースを変更することができる。	
	Execute	サブジェクトとはリソースを実行することができる。	
	Delete	サブジェクトはリソースを削除することができる。	
	Control	サブジェクトはリソースにアクセス制御ポリシーを設定することができる。	
否定付き Read/Write/Execute/Delet	Read	上記と同じ	urn:oasis:names:tc:SAML:1.0:action:rwedc-negation
	Write		
	Execute		

e/Control	Delete		
	Control		
	~Read	上記の反対	
	~Write		
	~Execute		
	~Delete		
	~Control		
Get/Head/Put/Post	GET	リソースに対してHTTPのGETメソッドを実行できる。	urn:oasis:names:tc:SAML:1.0:action:ghpp
	HEAD	リソースに対してHTTPのHEADメソッドを実行できる。	
	PUT	リソースに対してHTTPのPUTメソッドを実行できる。	
	POST	リソースに対してHTTPのPOSTメソッドを実行できる。	
UNIX File Permissions	数値化された値	*extended user group world*	urn:oasis:names:tc:SAML:1.0:action:unix

7.3. NameIdnetifier Format Identifiers

以下に<NameIdentifier>要素のFormat属性に使用可能な、<NameIdentifier>要素の内容の共通フォーマット識別子を示す。

フォーマット	概要	識別子
未定義	実装に任される	urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
電子メールアドレス	RFC2822で定義された電子メールアドレスのフォーマット	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
X.509サブジェクト名	XML署名の<ds:X509SubjectName>要素のフォーマット	urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
Windowsドメイン名	Windowsドメイン名で修飾された名前	urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName

付録: SAML アサーションの例

以下に SAML アサーションの例を示す。

<saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="20030708.0000" Issuer="Sample Authority" IssueInstant="2003-07-08T02:30:00Z">	サブジェクトに対する認証情報（アサーション）。 メジャーバージョンおよびマイナーバージョン。 アサーションを一意に識別する ID。 アサーションの発行者。 アサーションの発行時刻。
<saml:Conditions NotBefore="2003-07-08T02:30:00Z" NotAfter="2003-07-18T02:30:00Z"> <DoNotCacheCondition /> </saml:Conditions>	[オプション] 有効性を評価する際の前提条件。 有効期限（開始） 有効期限（終了） アサーションのキャッシュを禁止
<saml:Advice> <Assertion>...</Assertion> </saml:Advice>	[オプション] アサーションに関連する付加情報。
<saml:AuthenticationStatement> ... </saml:AuthenticationStatement>	アサーションの内容（ステートメント）。 AuthenticationStatement, AttributeStatement, AuthorizationDecisionStatement のいずれか。 詳細は下記を参照。
<ds:Signature>...</ds:Signature>	[オプション] SAML オーソリティの電子署名。
</saml:Assertion>	

アサーションの内容（ステートメント）は、認証情報をあらわす AuthenticationStatement、属性情報をあらわす AttributeStatement、アクセス許可情報をあらわす AuthorizationDecisionStatement のいずれかである。

以下に AuthenticationStatement の例を示す。

<saml:AuthenticationStatement AuthenticationMethod="urn:oasis:names:tc:SAML: 1.0:am:password" AuthenticationInstant="2003-07-08T02:30:00Z">	認証に関するステートメント。 認証手段。この例はパスワード認証。 認証した時刻。
<saml:Subject> <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1: nameid-format:emailAddress"> user@example.com </saml:NameIdentifier> </saml:Subject>	サブジェクト（認証主体）に関する情報。 名前の種別。この例はメールアドレス。他に X.500 識別名や Windows ドメイン識別名などがある。 サブジェクトの名前。
</saml:AuthenticationStatement>	

以下に AttributeStatement の例を示す。

<saml:AttributeStatement>	属性に関するステートメント。
<saml:Subject>...</saml:Subject>	サブジェクトに関する情報。
<saml:Attribute AttributeName="Role" AttributeNamespace="http://example.com"> <saml:AttributeValue> Manager </saml:AttributeValue> </saml:Attribute>	サブジェクトの属性。 属性の名前。 属性の名前空間。 属性の値。
</saml:AttributeStatement>	

以下に AuthorizationDecisionStatement の例を示す。

<saml:AuthorizationDecisionStatement Resource="http://example.com/test.html" Decision="Permit">	アクセス許可に関するステートメント。 対象となるリソース。 アクセス許可（Permit, Deny, Indeterminate のいずれか）。
<saml:Subject>...</saml:Subject>	サブジェクトに関する情報。
<saml:Action NameSpace="urn:oasis:names:tc:SAML:1.0: action:ghpp"> GET </saml:Action>	許可されるサブジェクトのアクション。 アクション名を識別する名前空間。 アクションの種別（GET, HEAD, PUT, POST や Read, Write, Execute など）。
</saml:AuthorizationDecisionStatement>	

付録B Bindings and Profile for the OASIS
Security Assertion Markup Language(SAML) V1.1

1. Introductiion

1.1. Protocol Bindings and Profile Concepts

SAMLのリクエスト／レスポンスメッセージの交換から、標準メッセージまたは通信プロトコルへのマッピングは、SAML プロトコルバインディング(または単にバインディング)と呼ばれる。例えば、SAML SOAPバインディングは、SAMLのリクエスト／レスポンスメッセージの交換が、どのようにしてSOAPメッセージ交換にマッピングされているかを記述している。

SAMLアサーションをフレームワークやプロトコルへ埋め込んだり、取り出したりする方法を記述したルールセットは、SAMLプロファイルと呼ばれる。例えば、SAML SOAPプロファイルは、SAMLアサーションがどのようにしてSOAPメッセージに加わるか､SAMLアサーションによってSOAPヘッダがどの様な影響を受けるか、SAML依存エラー状況をどのようにSOAPメッセージに反映すべきかを記述している。

2. Specification of Additional Protocol Bindings and Profiles

以下の節はバインディングとプロファイルの仕様を定めるためのガイドラインと、バインディングとプロファイルを記述し登録するためのプロセス・フレームワークを提供する。

2.1. Guidelines for Specifying Protocol Bindings and Profiles

各バインディングとプロファイルを決めるにあたり取り組むべき項目の一覧を示す。

1. バインディングやプロファイルの中に現れる当事者間の相互作用を記述する。各当事者によって使用されるアプリケーション、および各相互作用の中で使用されるプロトコルのどのような制約も明示的に示されなくてはならない。
2. 当事者がいくつ現れるのか、中間者が存在するのかどうかを含めて、各相互作用に関係する当事者を明らかにする。
3. 認証が要求されるかどうか、許容する認証の形式は何か、ということを含めて、各相互作用に関係する当事者間の認証方法を指定する。
4. メッセージの完全性を保証するために使われるメカニズムを含むメッセージの完全性のためのサポートレベルを明らかにする。
5. 第三者がSAMLメッセージおよびアサーションの内容を見てもよいかどうか､バインディングあるいはプロファイルが機密性を要求するかどうか、および機密性を達成するために推奨されるメカニズムを含めて、機密性のためのサポートレベルを明らかにする。
6. 各参加者、特にSAMLアサーションあるいはメッセージの受信と処理をする参加者のエラー状態を含めて、エラー状態を明らかにする。
7. 脅威の分析および対抗策の説明を含むセキュリティ上考慮すべき事柄を明らかにする。
8. バインディングあるいはプロファイルによって定義され、かつ/または、利用されるSAML confirmation method識別子を明らかにする。

2.2. Process Framework for Describing and Registering Protocol Bindings and Profiles

新しいバインディングやプロファイルの相互運用性を確保するため OASIS Security Service Technical Committee がバインディングとプロファイルの登録、リポジトリの維持、提案に関する指示を行う。バインディングやプロファイルを登録する際には以下の情報が提供されなくてはならない。

1. Identification: プロトコルバインディングあるいはプロファイルを一意に識別するURIを規定する。
2. Contact information: プロトコルバインディングあるいはプロファイルの作者への郵便あるいは電子的なコンタクト情報を明記する。
3. Description: プロトコルバインディングあるいはプロファイルのテキストによる記述を提供する｡記述は2.1節にて記述されたガイドラインに従う必要がある。
4. Updates: 今回のバージョンが改良あるいは置き換えようとする､以前に登録されたプロトコルバインデ

ィングあるいはプロファイルへの参照を提供する。

3. Protocol Bindings

以下のセクションで、現在唯一認定されている SAML SOAP バインディングについて定義する。

3.1. SAML SOAP Binding

SAML SOAP バインディングは SAML リクエストおよびレスポンスの送信および受信のための SOAP の使い方を定義する。

3.1.1. Required Information

Identification: urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding
Contact information: security-services-comment@lists.oasis-open.org
Description: 下記
Updates: なし

3.1.2. Protocol-Independent Aspects of the SAML SOAP Binding

以下の節では、SAML SOAP バインディングの特徴を、SOAP メッセージを運ぶために使用される HTTP のような SOAP の下位プロトコルからは独立な形で定義する。

3.1.2.1 Basic Operation

SAML 通信を SOAP 上で行うために使用されるシステムモデルは単純なリクエストレスポンスモデルである。

1. SAML リクエスタの役割をするシステムエンティティは、SAML レスポンダの役割をするシステムエンティティへ、SAML <Request>要素を SOAP メッセージのボディに入れて送信する。SAML リクエスタは、一つの SOAP メッセージに複数の SAML リクエストを入れたり、SOAP ボディ内に追加の XML 要素を入れたりしてはならない。
2. SAML レスポンダは、<Response>要素を別の SOAP メッセージのボディに入れて返すか、あるいは SOAP フォルトコードを返さなければならない。SAML レスポンダは、一つの SOAP メッセージに複数の SAML レスポンスを入れたり、SOAP ボディ内に追加の XML 要素を入れたりしてはならない。もし、何らかの理由で SAML レスポンダが SAML リクエストを処理できないならば、SOAP フォルトコードを返さなければならない。SOAP フォルトコードは、例えば、拡張スキーマを見つけることができないとか、認証クエリーにおいてサブジェクトがリソースにアクセスすることを許可されないことを示すシグナル、といった SAML の領域内のエラーのために送られてはならない。

3.1.2.2 SOAP Headers

SAML 通信 over SOAP での SAML リクエスタは SOAP メッセージに任意のヘッダを追加してよい。このバインディングは追加 SOAP ヘッダを定義しない。SAML レスポンダは SOAP メッセージのためのヘッダを必要としてはならない。

3.1.2.3 Authentication

SAML リクエスタおよび SAML レスポンダ相方の認証はオプションであり使用環境に依存する。下位プロトコルで利用可能な認証プロトコルを、認証を提供するために使用してよい。

3.1.2.4 Message Integrity

SAML リクエストと SAML レスポンスのメッセージの完全性はオプションであり使用環境に依存する。下位プロトコルのセキュリティレイヤーを、メッセージの完全性を保証するために使用してよい。

3.1.2.5 Confidentiality

SAML リクエストと SAML レスポンスのメッセージの機密性はオプションであり使用環境に依存する。下位プロトコルのセキュリティレイヤーを、メッセージの機密性を保証するために使用してよい。

3.1.3. Use of SOAP over HTTP

SAML SOAP バインディングへの適合を主張する SAML 処理系は SAML over SOAP over HTTP を実装しなければならない。SOAP HTTP バインディングでは SOAP HTTP リクエストの一部として SOAPAction ヘッダの使用を必要とする。SAML レスポンダはこのヘッダの値に依存してはならない。SAML リクエスタは SOAPAction ヘッダの値として次のような値をセットできる。

```
http://www.oasis-open.org/committees/security
```

3.1.3.1 HTTP Headers

HTTP プロキシは SAML アサーションを運ぶレスポンスをキャッシュしてはならない。HTTP1.1 を使うときは次の条件を二つとも適用する。

- Cache-Control ヘッダフィールドの値が no-store に指定されない限り、SAML レスポンダはレスポンスに Cache-Control ヘッダフィールドを入れてはならない。
- Expires レスポンスヘッダフィールドは値が no-store に指定された Cache-Control ヘッダフィールドによって無効にされない限り、Expires フィールドを入れるべきではない。

3.1.3.2 Authentication

SAML リクエスタおよびレスポンダは次の認証方法を実装しなければならない。

1. クライアントあるいはサーバ認証なし。
2. SSL3.0 あるいは TLS1.0 を伴う、または伴わない HTTP ベーシッククライアント認証[RFC2617]
3. サーバサイド証明書を伴う HTTP over SSL3.0 あるいは HTTP over TLS1.0(6 節参照)サーバ認証
4. サーバサイドおよびクライアントサイド証明書の両方を伴う HTTP over SSL3.0 あるいは HTTP over TLS1.0 相互認証

SAML レスポンダが SSL3.0 あるいは TLS1.0 を使うならば、サーバサイド証明書を使わなければならない。

3.1.3.3 Message Integrity

メッセージの完全性が保証される必要があるとき、SAML レスポンダはサーバサイド証明書を伴う HTTP over SSL3.0 あるいは HTTP over TLS1.0(6 節参照)を使わなければならない。

3.1.3.4 Message Confidentiality

メッセージの機密性が要求されるとき、SAML レスポンダはサーバサイド証明書を伴う HTTP over SSL3.0 あるいは HTTP over TLS1.0(6 節参照)を使わなければならない。

3.1.3.5 Security Considerations

認証、メッセージの完全性、および機密性のメカニズムの各組み合わせは実際に適用される前に、適用される環境における脆弱性に関して分析されるべきである。

3.1.3.6 Error Reporting

SAML レスポンダがリクエスタにエラーが発生したことを報告する方法について規定する。

1. SAML リクエスタとのメッセージ交換の実行を拒否する場合は「403 Forbidden」レスポンスを返す必要がある。
2. SOAP リクエストを処理中（制御が SAML プロセッサに渡される前に SOAP に関するエラーが検出された場合、もしくは SOAP プロセッサで内部エラーが発生した場合）の SOAP エラーの場合、SOAP HTTP サーバは SOAP の fault エレメントが設定された SOAP メッセージを含む「500 Internal Server Error」レスポンスを返さなくてはならない。
3. SAML 処理中のエラーの場合、SOAP HTTP サーバは「200 OK」の応答の中に、<SOAP-ENV:Body>要素の唯一の子か、SOAP Body 内の SAML<Response>要素の唯一の子として SAML の<Status>要素を入れて応答しなくてはならない。

4. Profiles

以下のセクションにて、OASIS Security Service Technical Committee によって認められた SAML のプロファイルを定義する。ブラウザベースの SSO をサポートするプロファイルが２つ定義されている。

- SAML ブラウザ／アーティファクトプロファイル
- SAML ブラウザ／POST プロファイル

Security Services Technical Committee 以外からいくつかのプロファイルが発行されている。

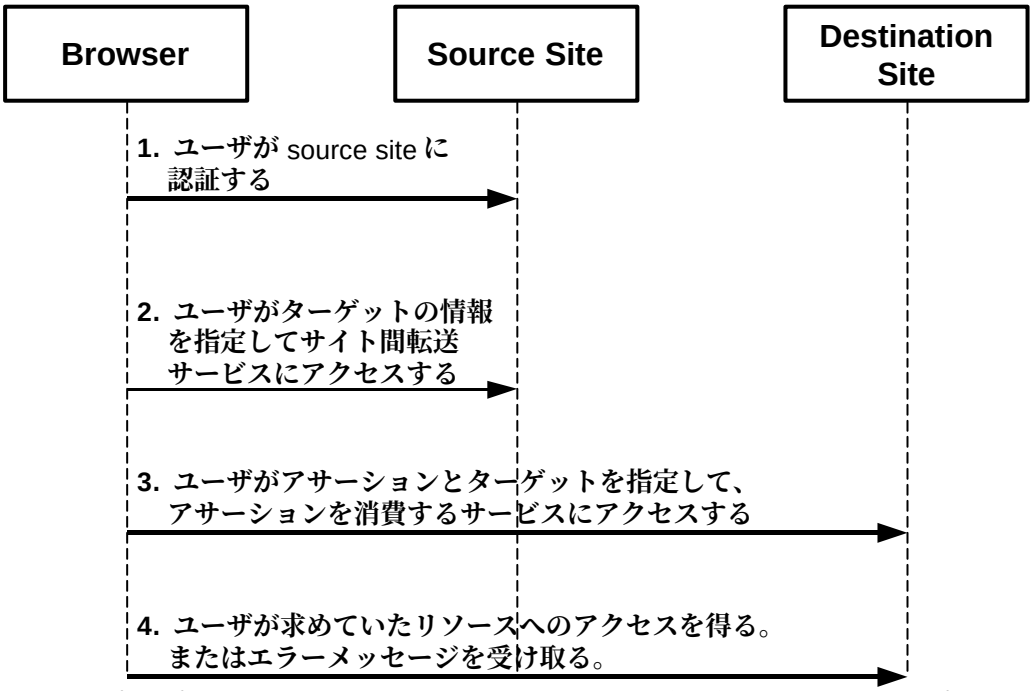
- OASIS Web Services Security Technical Committee は WSS 規格の"SAML トークンプロファイル"のドラフトを作成している。この規格では、web サービスのメッセージで SAML アサーションをセキュアに使用する方法について述べている。
- Liberty Alliance Project が、彼らの SAML の拡張版で使用するためにプロファイルを作成している。

4.1. Web Browser SSO Profiles of SAML

web ブラウザを使用したこのシナリオでは、ユーザはソース Web サイトに認証をおこない、それから目的 Web サイトにある守られたリソースを、直接目的 Web サイトに認証することなく利用する。
このプロファイルの目的のため、以下の仮定がなされている。

- ユーザは標準的な商用のブラウザを使用し、ソース Web サイトに対しては SAML のスコープ外の何らかの方法で認証されている。
- ソース Web サイトは、ローカルに認証したユーザを追跡できる適切な何らかのセキュリティエンジンを持つ。通常、これは暗号化されたクッキーやエンコードされた URL、もしくは何らかのそのほかの技術によって表現されるセッションの形をとる。

以下の図は SSO を実現する基本的なテンプレートを描いたものである。



(出典:"Bindings and Profiles for the OASIS Security Assertion Markup Language(SAML) V1.1Committee Specification, 27 May 2003)

web ブラウザを使用した SSO プロファイルでは、あるサイトから他のサイトへ標準的な商用のブラウザを使用して情報を運ぶために、HTTP ベースの方法が２種類使用されている。

- **SAML アーティファクト:** SAML アーティファクトは URL のクエリーストリングの一部として目的 Web サイ

トに運ばれる小さなサイズのデータで、後でソースWebサイトに戻されると、アサーションに対する参照として利用できるようなものである。アーティファクトはリダイレクションを使って目的Webサイトに運ばれ、目的Webサイトはその後のステップでソースWebサイトから参照先のアサーションを得る。この方法はブラウザ／アーティファクトプロファイルで使用される。

- **Form POST:** SAMLアサーションはHTMLのフォームの中に入れられてブラウザに渡され、ユーザがフォームのサブミットを行ったときにHTTP POSTの一部として目的Webサイトに運ばれる。この方法はブラウザ／POSTプロファイルで使用される。

これらのプロファイルではソースWebサイトと目的Webサイトが同じ"cookie domain"に属さなくてはならないと言う制限が発生するため、クッキーは使用されない。
以下の議論で *SSO アサーション*という用語が使用されるが、それは NotBefore と NotOnOrAfter 属性がある <saml:Conditions>要素を持ち、サブジェクトに関する少なくとも一つの authentication statement を含んでいるアサーションのことである。

4.1.1. Browser/Artifact Profile of SAML

4.1.1.1 Required Information

Identification: urn:oasis:names:tc:SAML:1.0:profiles:artifact-01
Contact information: security-services-comment@lists.oasis-open.org
SAML Confirmation Method Identifiers: このプロファイルでは"SAML アーティファクト" confirmation method識別子が使用される。このconfirmation methodに割り当てられている識別子として以下のものが推奨されている。

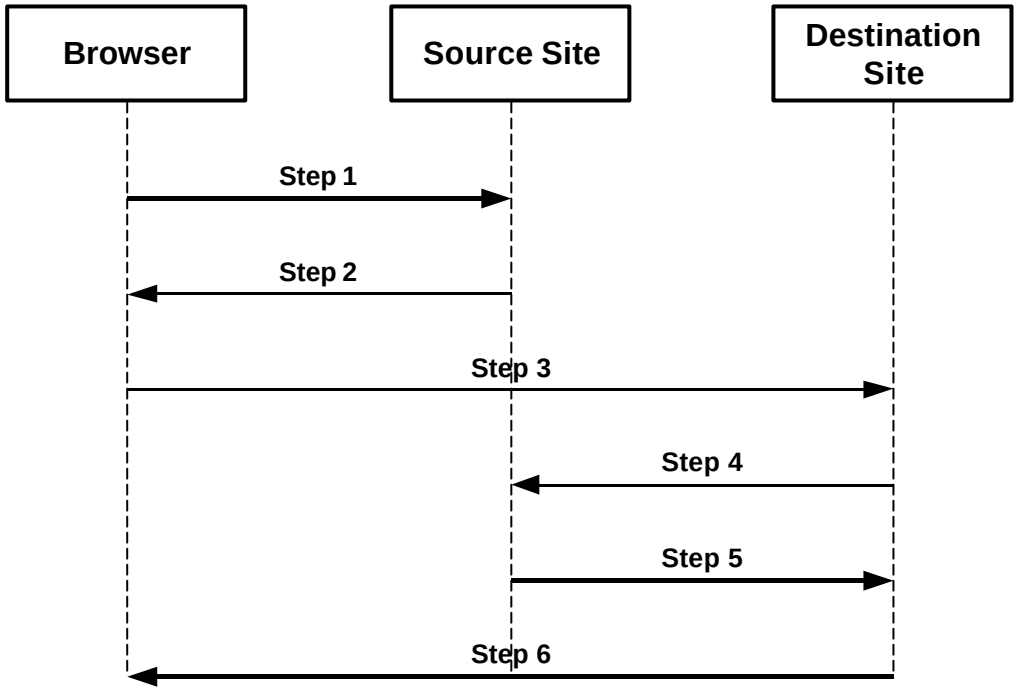
urn:oasis:names:tc:SAML:1.0:cm:artifact

以下の識別子はSAMLの次版では削除されることが計画されており、推奨されていない。

urn:oasis:names:tc:SAML:1.0:cm:artifact-01

Description: 下記
Updates: なし

4.1.1.2 Preliminaries



(出典:"Bindings and Profiles for the OASIS Security Assertion Markup Language(SAML) V1.1 Committee Specification, 27 May 2003)

ステップ	通信内容	説明
Step 1	GET <path>?...TARGET=<Target>...<HTTP-Version> <other HTTP 1.0 or 1.1 components>	ユーザのブラウザが、https://<inter-site transfer host name>のサイト間転送サービスにアクセスする。そのとき、実際に利用したい目的 Web サイトのターゲットの情報を URL に付けて送る。機密性とメッセージの完全性が保証されなくてはならない。
Step 2	<HTTP-Version> 302 <Reason Phrase> <other headers> Location : https://<artifact receiver host name and path>?<SAML searchpart> <other HTTP 1.0 or 1.1 components>	サイト間転送サービスがユーザのブラウザを目的 Web サイト上のアサーションを使用するサービスにリダイレクトする。<SAML searchpart>には一つのターゲットの情報と、最低一つの SAML アーティファクトを含む。機密性とメッセージの完全性が保証されなくてはなら

	<SAML searchpart>=...TARGET=<Target>...SAMLart=<SAML artifact>...	ない。
Step 3	GET <path>?...<SAML searchpart>...<HTTP-Version> <other HTTP 1.0 or 1,1 request components>	ユーザのブラウザが、リダイレクトされた目的 Web サイト上のアサーションを使用するサービスにアクセスする。機密性とメッセージの完全性が保証されなくてはならない。
Step 4	<samlp:Request> including <samlp:AssertionArtifact> プロトコルは SAML over SOAP over HTTPS などを利用	目的Webサイトが SAML プロトコルバインディングで規定された方法で SAML リクエストをソース Web サイトに送る。機密性とメッセージの完全性が保証されなくてはならない。
Step 5	<samlp:Response> including or not including assertion プロトコルは SAML over SOAP over HTTPS などを利用	ソース Web サイトが目的 Web サイトに SAML レスポンスを返す。リクエストに指定されたアーティファクトでアサーションに対する参照が解決できれば、レスポンスにはアサーションが含まれる。機密性とメッセージの完全性が保証されなくてはならない。
Step 6	規定なし	ユーザのブラウザは元々求めていたリソースへのアクセスを許可または拒絶される。

4.1.1.8 Artifact Format

アーティファクトのフォーマットを下記に示す。

```
SAML_artifact       := B64(TypeCode RemainingArtifact)
TypeCode            := Byte1Byte2
RemainingArtifact   := SourceID AssertionHandle
TypeCode            := 0x0001
SourceID            := 20-byte_sequence
AssertionHandle     := 20-byte_sequence
```

SAML_artifact	TypeCode と RemainingArtifact を結合させて Base64 変換を行ったもの
TypeCode	2 バイトのタイプコード。このプロファイルでは 0x0001 固定
RemainingArtifact	SourceID と AssertionHandle を結合させたもの
SourceID	ソース Web サイトの身元とロケーションの識別子として目的 Web サイトが使用する 20 バイトのデータ。この情報はソース Web サイトと目的 Web サイトの間で SAML の規格外の方法で通信される。
AssertionHandle	アサーションの内容や、有効な既存の AssertionHandle の値を推測したり作成できないよう、ソース Web サイトによって管理される 20 バイトのデータ。

ソース Web サイトが SAML アーティファクトを生成するにあたって以下の慣習に従うことが推奨される。

- 各々のソース Web サイトは単一の識別 URL を選択する。この URL にて使われるドメイン名は適切な機関 (authority) に登録され、ソース Web サイトによって管理される。
- ソース Web サイトは、識別 URL の SHA-1 ハッシュをとることにより、アーティファクトの SourceID コンポーネントを生成する。
- AssertionHandle 値は、暗号的に強固な乱数もしくはソース Web サイトで生成した擬似乱数シーケンス [RFC1750]によって構築する。シーケンスは 8 バイト以上のサイズをもつ値で構成される。これらの値は合計が 20 バイトの長さになるように詰められるべきである。

4.1.1.9 Threat Model and Countermeasures

4.1.1.9.1 Stolen Artifact

脅威: 盗聴者が正規ユーザの SAML アーティファクトをコピー可能ならば、盗聴者は正規ユーザの SAML アーティファクトを付与した URL を生成でき、目的 Web サイトにおいてそのユーザになりすますことが可能である。

対策: ステップ 2, 3, 4, 5 で示したように、サイトと利用書のブラウザ間でアーティファクトが交換されているあいだは、機密性が確保されなければならない (MUST) 。機密性は、正規ユーザの SAML アーティファクトへのアクセスを得ようとする盗聴者に対する保護を提供する。

機密性を確保するために使われる手段を盗聴者が破る可能性があるならば、さらに以下のような対策をとることができる。

- ソース Web サイトおよび目的 Web サイトは、両サイトにおける時刻設定が最大でも数分しか違わないことを保証するために、合理的な努力をするべきである (SHOULD) 。様々な時刻同期サービスがインターネット上および専用のソースから利用可能である。
- ステップ 5 にて通信する SAML アサーションは、SSO アサーションを含まなければならない (MUST) 。
- ソース Web サイトは、SAML アーティファクトが生成され URL に置かれた時刻と、そのアーティファクトを運ぶ <samlp:Request> メッセージが目的 Web サイトによって受信された時刻の差分を確認するべきである(SHOULD) 。タイムリミットは最大でも数分にすることが推奨される。万が一、このタイムリミットをこえて目的 Web サイトからの問い合わせによってアサーションが要求された場合は、ソース Web サイトは目的 Web サイトにアサーションを供給してはならない (MUST) 。
- ソース Web サイトは、対応するアーティファクトが生成される時か、アーティファクトを運ぶ

<samlp:Request> メッセージが受信された時のどちらかの時点で、SSO アサーションを生成することができる。どちらのケースにおいても、アサーションの有効期間は、前者のケースではより長く、後者のケースではより短くなるように適切に設定されるべきである（SHOULD）。。

- SSO アサーションにおける NotBefore および NotOnOrAfter 属性の値は、ソース Web サイトから目的 Web サイトへのアサーションの伝達が成功する、最短の有効期間とするべきである（SHOULD）。これは通常数分のオーダである。これにより、盗まれたアーティファクトが小さなタイムウィンドウ内にのみ使うことが可能であることが保証できる。
- 目的 Web サイトは、ソース Web サイトから獲得したすべてのアサーションの有効期間を確認し、期限切れのアサーションは拒否しなければならない（MUST）。目的 Web サイトは SSO アサーションに対して厳密な検証検査を実装することを選んでもよい（MAY）。これは、アサーションの IssueInstant や AuthenticationInstant 属性値が、目的 Web サイトがアサーションを受信した時点から数分以内になることを要求するような検査である。
- 受信したアサーションステートメントに利用者の IP アドレスをもつ <saml:SubjectLocality> 要素が含まれているならば、目的 Web サイトは認証ステートメント内の IP アドレスとブラウザの IP アドレスを照合してもよい（MAY）。

4.1.1.9.2 Attacks on the SAML Protocol Message Exchange

脅威: ステップ 4 と 5 におけるメッセージ交換は、アーティファクトやアサーションの盗難、リプレイ、メッセージの挿入や改変、そして中間者攻撃を含む様々な方法で攻撃される。
対策: SAML プロトコル結合を利用するときに相互認証、メッセージの完全性、および機密性を要求することが、これらの攻撃に対する防御となる。

4.1.1.9.3 Malicious Destination Site

脅威: 悪意のある目的 Web サイトが利用者からのアーティファクトを獲得すると、他の目的 Web サイトにおいて利用者になりすますことが可能になる。その目的 Web サイトはソース Web サイトからアサーションを獲得し、悪意のあるサイトが利用者であると信じてしまうだろう。
対策: ソース Web サイトはアサーションのターゲットとなったサイト以外からアサーションを要求されても、アサーションを発行しないようにする。

4.1.1.9.4 Forged SAML Artifact

脅威: 悪意のあるユーザが SAML アーティファクトを偽造できる。
対策: 悪意あるユーザは推測したアーティファクトの値を何度も試すと考えられるので、ソース Web サイトは存在しないアーティファクトを繰り返し問い合わせられた場合に警告を発する仕組みを実装するべきである。

4.1.1.9.5 Browser State Exposure

脅威: SAML ブラウザ／アーティファクトプロファイルは、ソース Web サイトからブラウザに対する SAML アーティファクトの「ダウンロード」を含んでいる。この情報は Web ブラウザの状態の一部として存在し、通常はまったくセキュアでない方法で、利用者のシステムの固定記憶装置に格納される。ここでの脅威は、アーティファクトが格納後に再利用されるかもしれないということである。
対策: SAML アーティファクトの「one-use」特性は、ブラウザからそれらを再使用できないことを保証する。アーティファクトおよび必須 SSO アサーションにおける推奨された短いライフタイムによって、アーティファクトを盗んで、後ほど他のブラウザから再利用することが困難になる。

4.1.2. Browser/POST Profile of SAML

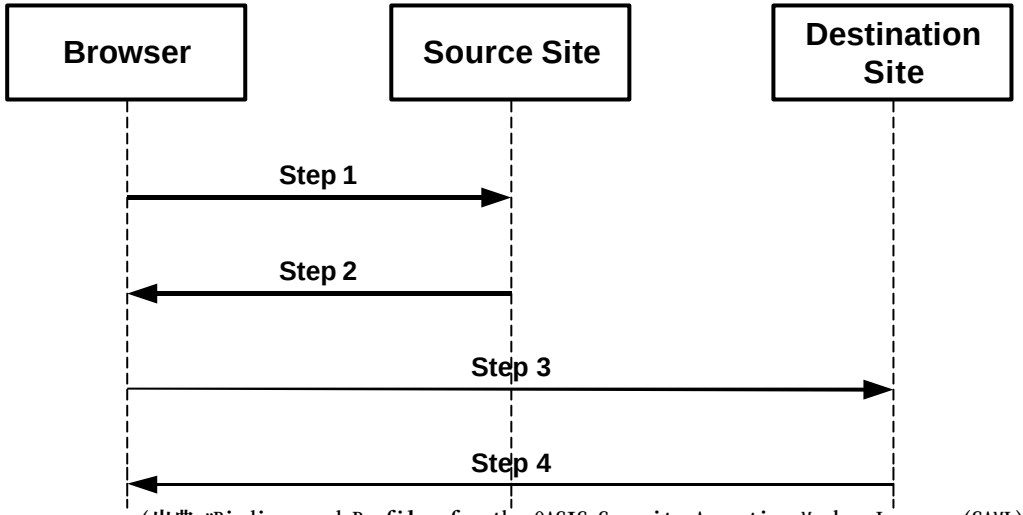
4.1.2.1 Required Information

Identification： urn:oasis:names:tc:SAML:1.0:profiles:browser-post
Contact information: security-services-common@lists.oasis-open.org
SAML Confirmation Method Identifiers: このプロファイルでは、”Bearer”式の confirmation method の識別子が使用される。この confirmation method には以下の識別子が割り付けられている。

urn:oasis:names:tc:SAML:1.0:cm:bearer

Description: 下記
Updates: なし

4.1.2.2 Preliminaries



(出典:"Bindings and Profiles for the OASIS Security Assertion Markup Language(SAML) V1.1 Committee Specification, 27 May 2003)

ステップ	通信内容	説明
Step 1	GET <path>?...TARGET=<Target>...<HTTP-Version> <other HTTP 1.0 or 1.1 components>	ユーザのブラウザが、https://<inter-site transfer host name>のサイト間転送サービスにアクセスする。そのとき、実際に利用したい目的 Web サイトのターゲットの情報を URL に付けて送る。機密性とメッセージの完全性が保証されなくてはならない。
Step 2	<HTTP-Version> 200 <Reason Phrase> <other HTTP 1.0 or 1.1 components> <Body> <FORM Method="Post" Action="https://<assertion consumer host name and path>"...> <INPUT TYPE="hidden" NAME="SAMLResponse" Value="B64(<response>)"> ... <INPUT TYPE="hidden" NAME="TARGET" Value="<Target>"> </Body>	ソース Web サイトが SSO アサーションを含む SAML レスポンスを入れた HTML のフォームデータを生成して応答する。このフォームを POST する先が目的 Web サイトのアサーションを受け取るサービスになるようにする。この URL は SSL3.0 または TLS1.0 で守られていることが推奨されている。フォームの SAMLResponse に SAML レスポンスを Base64 エンコードした結果を設定する。ターゲットについての記述が TARGET という名前で一つ設定されなくてはならない。機密性とメッセー

		ジの完全性が保たれなくてはならない。
Step 3	POST <path> <HTTP-Version> <other HTTP 1.0 or 1.1 request components>	ブラウザがSAMLレスポンスを含むフォームを目的Webサイトのアサーションを受け取るサービスにサブミットする。フォームデータのセットには、SAMLResponseという名前で SAML レスポンスが一つだけ設定されていなくてはならない。ターゲットについての記述がTARGETという名前で一つだけ設定されていなくてはならない。HTTPリクエストの機密性とメッセージの完全性が維持されなくてはならない。
Step 4	規定なし	ユーザのブラウザは元々求めていたリソースへのアクセスを許可または拒絶される。

4.1.2.7 Threat Model and Countermeasures

4.1.2.7.1 Stolen Assertion

脅威：盗聴者が正規ユーザの SAML レスポンスと含まれていたアサーションをコピー可能ならば、盗聴者は正しい POST の内容を造ることができ、目的 Web サイトにおいてユーザになりすますことができる。
対策：ステップ 2 とステップ 3 に示されているように、サイトとユーザのブラウザの間でレスポンスが通信される場合には、必ず機密性が保たれなくてはならない。これによって、盗聴者が本当のユーザ向けのレスポンスやアサーションを獲得することを防ぐことができる。

機密性を確保するために使われる手段を盗聴者が破る可能性があるならば、さらに以下のような対策をとることができる。

- ソース Web サイトおよび目的 Web サイトは、両サイトにおける時刻設定が最大でも数分しか違わないことを保証するために、合理的な努力をするべきである（SHOULD）。様々な時刻同期サービスがインターネット上および専用のソースから利用可能である。
- ステップ 3 で通信される SAML アサーションは、SSO アサーションを含まなくてはならない。
- SSO アサーションにおける NotBefore および NotOnOrAfter 属性の値は、ソース Web サイトから目的 Web サイトへのアサーションの伝達が成功する、最短の有効期間とするべきである（SHOULD）。これは通常数分のオーダである。これにより、盗まれたアサーションが小さなタイムウィンドウ内にのみ使うことが可能であることが保証できる。
- 目的 Web サイトは、ソース Web サイトから獲得したすべてのアサーションの有効期間を確認し、期限切れのアサーションは拒否しなければならない（MUST）。目的 Web サイトは SSO アサーションに対して厳密な検証検査を実装することを選んでもよい（MAY）。これは、アサーションの IssueInstant や AuthenticationInstant 属性値が、目的 Web サイトがアサーションを受信した時点から数分以内になることを要求するような検査である。
- 受信したアサーションステートメントに利用者の IP アドレスをもつ <saml:SubjectLocality> 要素が含まれているならば、目的 Web サイトは認証ステートメント内の IP アドレスとブラウザの IP アドレスを照合してもよい（MAY）。

4.1.2.7.2 MITM Attack

脅威：目的 Web サイトがユーザから SAML アサーションを HTML のフォームで受け取ると、悪意のあるサイトは他の新しい目的 Web サイトに対してユーザに成り済ますことができるようになる。その目的 Web サイトは悪意のあるサイトをアサーションのサブジェクトであると信じてしまうだろう。
対策：目的 Web サイトは SAML レスポンスの Recipient 属性の値が https://<assertion consumer host name and path>になっていることを確認しなくてはならない。レスポンスがデジタル署名されていれば、悪意のあるサイトによって Recipient 属性の値が変更されてしまうことは無い。

4.1.2.7.3 Forged Assertion

脅威：悪意のあるユーザやブラウザのユーザが、SAML アサーションを作ったり改変したりする。
対策：ブラウザ／POST プロファイルでは、メッセージの完全性と認証を実現するために、SAML アサーションを運ぶ SAML レスポンスには署名が必要である。目的 Web サイトは署名の検証と issuer の認証を行わなければならない。

4.1.2.7.4 Browser State Exposure

脅威：ブラウザ／POST プロファイルでは Web ブラウザからソース Web サイトへのアサーションのアップロードがある。この情報は Web ブラウザの状態の一部として存在し、通常はまったくセキュアでない方法で、利用者のシステムの固定記憶装置に格納される。ここでの脅威は、アサーションが格納後に再利用されるかもしれないということである。
対策：このプロファイルを使用して通信されるアサーションは、必ず SSO アサーションを含まなくてはならない。SSO アサーションは寿命が短く、目的 Web サイトは SSO アサーションが再提出されたのでないことを保障すると考えられる。

5. Confirmation Method Identifiers

<SubjectConfirmation>要素は、署名検証者によってリクエスト又はメッセージが文書中のサブジェクトと一致するシステムエンティティから来たことを確認するために使われるべきである。<ConfirmationMethod>要素は署名検証者がこの判断のために使用するべき手段を示す。プロファイルやバインディングが、それぞれ異なる SAML の使用方法のシナリオに関連した<ConfirmationMethod>の値を定義することになると予想される。

5.1. Holder of Key

URI: urn:oasis:names:tc:SAML:1.0:cm:holder-of-key

<SubjectConfirmation>要素に<ds:KeyInfo>要素が存在しなくてはならない。

[XMLSig]で規定されているように、<ds:KeyInfo>要素には、鍵かアプリケーションが鍵を得るための情報が格納される。アサーションの statement(s)のサブジェクトは、自分が鍵の持ち主であることを示すことができる。

5.2. Sender Vouches

URI: urn:oasis:names:tc:SAML:1.0:cm:sender-vouches

アサーションの使用される状況についてその他の情報が無いことを示す。署名検証者はアサーションをさらに処理するべきかどうかを判断するために、他の方法を利用しなくてはならない。

5.3. SAML Artifact

Recommended URI: urn:oasis:names:tc:SAML:1.0:cm:artifact
Deprecated URI: urn:oasis:names:tc:SAML:1.0:cm:artifact-01

アサーションのサブジェクトは、SAML アーティファクトを提出した party である。署名検証者はそのアーティファクトをアサーションを得るために使用する。セクション 4.1.1.1 を参照。

5.4. Bearer

URI: urn:oasis:names:tc:SAML:1.0:cm:bearer

アサーションのサブジェクトはアサーションを提出したものである。セクション 4.1.2.1 を参照。

6. Use of SSL 3.0 or TLS 1.0

SAML で SSL3.0**[SSL3]**や TLS1.0**[RFC2246]**を使用する場合は、サーバは X.509 v3 証明書を使用してクライアントを認証しなくてはならない。クライアントは証明書の内容に基づいて（通常は証明書のサブジェクト DN を確認することによって）サーバのアイデンティティを確認しなくてはならない。

6.1. SAML SOAP Binding

TLS を使用する実装では、TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suite を実装しなくてはならず、TLS_RSA_AES_128_CBC_SHA cipher suite を実装してもよい。

6.2. Web Browser Profiles of SAML

SSL が使用可能な SAML のブラウザ／アーティファクトプロファイルやブラウザ／POST プロファイルの実装では、SSL_RSA_WITH_3DES_EDE_CBC_SHA cipher suit を実装しなくてはならない。

TLS が使用可能な実装では、TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suit を実装しなくてはならない。

7. Alternative SAML Artifact Format

7.1. Required Information

Identification: urn:oasis:names:tc:SAML:1.0:profiles:artifact-02
Contact information: security-services-comment@lists.oasis-open.org
Description: 下記
Updates: なし

7.2. Format Details

```
TypeCode              := 0x0002
RemainingArtifact     := AssertionHandle SourceLocation
AssertionHandle       := 20-byte_sequence
SourceLocation        := URI
```

TypeCode	2 バイトのタイプコード。このプロファイルでは 0x0002 固定
RemainingArtifact	AssertionHandle と SourceLocation をくっつけたもの
AssertionHandle	セクション 4.1.1.8 に記述されたものと同じ
SourceLocation	UTF-8 にエンコードされた SAML レスポンダの URI

desitination site はこのアーティファクトを基本的にはセクション 4.1.1 の記述と同じように処理する。SourceID ではなく SourceLocation になっているので、ソース Web サイトの位置が直接得られる点だけが異なる。

8. URL Size Restriction (Non-Normative)

Microsoft:
IE ではＵＲＬは最大 2083 文字（ただしパス長は最大 2048 文字）。この制限は POST と GET の両方のＵＲＬに適用されるが、POST の場合は名前と値のペアがＵＲＬに含まれないため、名前と値に対してはこの制限は無関係。

Netscape Enterprise Server
最大 4096 文字

付録 C　Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1

(OASIS Standard, 2 September 2003)

Document identifier：oasis-sstc-saml-sec-consider-1.1
Location：http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

※この文書は、Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1 の主要部分を日本語に訳したものである。内容に隔たりがある場合は英語版が正しいものとする。

1. Introduction

この標準でない文書では、核となる SAML の仕様[SAMLCore]と SAML のバインディングとプロファイルに関する仕様[SAMLBind]が規定されている the OASIS Security Assertion Markup Language(SAML)のセキュリティとプライバシの性質を記述して分析している。この文書は SAML の設計に関してアドバイスを与え、設計者・開発者・SAML ベースのシステムの reviewers に以下に列挙する事項についての情報を提供することを目的としている。

- 脅威、SAML ベースのシステムが対象となるセキュリティリスク
- SAML アーキテクチャが取り組むセキュリティリスクと、その取り組み方
- SAML で扱わないセキュリティリスク
- それらのリスクを軽減する対応策の提案

この文書で用いられる用語は別の方法で言及されているものを除いて、SAML の用語集[SAMLGloss]によって定義されている。

このセクションの後の部分ではこの文書での分析の基となる背景と前提を記述している。セクション 4 は SAML で用いられるべきセキュリティの手法と技術のハイレベルな見解を示している。セクション 5 では SAML を利用する際に内在する固有のリスクを分析している。

2. Privacy

SAML は認証されたエンティティの属性と認可についてのステートメントを作成する能力を含む。これらのステートメントによって運ばれる情報が、その通信に参加しているパーティにとって、できるだけ限られたエンティティにだけしかアクセス可能でないようにすることを望むものであることは、よくあることである。医療もしくは金融上の属性のステートメントはそのようなケースのシンプルな例である。

ステートメントを作成する、アサーションを発行する、アサーションを運ぶ、アサーションを消費するパーティはこれらの潜在的なプライバシに対する懸念を認識しなければならない。そして、システムの実装においてそれらに取り組むよう努めるべきである。

2.1. Ensuring Confidentiality

おそらく SAML 対応のトランザクションにおいてパーティのプライバシを保証するために最も重要な点は、機密性の保証がなされたトランザクションを行えるかどうかということである。言い換えると、アサーションの情報が発行者から意図した受信者だけに運ばれ、他のパーティにアクセス可能にさせないことが可能であるかどうかということである。

情報を機密性を保護して運ぶことは技術的には可能である（セクション 4.2 で、機密性を提供する一般的な方法の議論がなされている）。SAML 対応のトランザクションへの全てのパーティは、機密性が保護されるべき情報が実際に保護されているかどうかを確かめるためにインタラクションのそれぞれのステップを解析すべきである。

また、単にアサーションの中身を隠すことでは、十分なプライバシ保護にならないかもしれないことに注意すべきである。与えられたユーザ(もしくは IP アドレス)が、与えられたサービスにアクセスしていたという情報を利用可能であるということだけでプライバシの侵害となるかもしれない多くのケースがある(例え

ば、ユーザがアサーションのために医療検査施設にアクセスしたという情報だけで、アサーションの内容は知られなくても十分プライバシ侵害となるかもしれない)。これらの問題に対する部分的な解決策は、下記に概説する anonymous なインタラクションのためのさまざまな技術によって提供される。

2.2. Notes on Anonymity

次のセクションでは anonymity の概念を議論する。

2.2.1. Definitions That Relate to Anonymity

全ての場合を満足させる anonymity の定義はない。多くの定義[Anonymity]では送信者とメッセージのシンプルな場合を扱っており、与えられた送信者と送信メッセージを結びつけることができない、またはもとの送信者へメッセージを結び付けることができないという観点から“ anonymity ”を議論している。
このような定義は一回限りのやり取りを扱うケースにおいては十分であるものの、識別子よりもむしろ時間をまたがった振る舞いに基づいて集めることができる情報の集合は無視している。
2 つの概念が一般的に有用なものであり、それらは互いに関連し、anonymity を定義する助けになる。
1 つめの概念は“ Anonymity、Unobservability、Pseudonymity ”[Anonymity]からの下記コメントのように、それらを一つとして anonymity を考えるべきである。

> : サブジェクトの anonymity を可能にするために常に、潜在的に同じ属性を持った適切なサブジェクトの集合でなければならない……anonymity が強くなればなるほど、それぞれの anonymity の集合はより大きくなり、より均等にその集合内のサブジェクトの送信、受信が各々頒布する。

【訳注：[Anonymity]の説明より抜粋】
Anonymity(無名) ： anonymity な集合であるサブジェクトの集合内で識別不可能である状態
Unobservability(不可観測性)：どんな対象事項ともまったく区別がつかない対象事項の状態
Pseudonymity(仮名)：ＩＤとしての仮名の利用

この概念はオーソリティの利用が理由で SAML に関係がある。たとえサブジェクトが anonymous であったとしても、関連するオーソリティのドメイン内の一連のサブジェクトのメンバであるということは特定できてしまう。
ユーザの統合的な属性が提供される場合、その集合はより小さくなる可能性がある、例えば、もしユーザが anonymous であるとしても Course 6@mit.edu の生徒という属性を持つという例がある。確かに、Course6 の生徒の数はどこにでもいるユーザの数をより限定した MIT に関係した人物の数よりもさらに少ない。
何故これが問題となるのであろうか？Non-anonymity は、Dingledine、Freedman、Molnar の Freehaven という文書[FreeHaven]で説明されているように、敵に攻撃する能力を与えてしまう。

> : anonymity と pseudonymity の両方がユーザの位置情報とトゥルーネーム（true name)というプライバシを保護する。位置情報はシステムへの実際の物理的な接続を参照する。“ true name ”という用語は個人の法律上のアイデンティティを参照するために Vinge によって紹介され、May によって一般化された。誰かの本当の名前もしくは位置情報を知ることは、彼もしくは彼女を傷つけることが可能になるということである。

これは、その同じソースからの集合と被害の可能性において、anonymity の概念の一体化を導く[FreeHaven]。

> : 我々は、もし敵がユーザの探索の範囲を被疑者集合のうちの一人であると限定できるならば、そのシステムは部分的に anonymous であると言うかもしれない。もしその集合が十分大きいならば、敵にとってその集合の中の全ての被疑者が疑わしいとして行動することは非現実的である。他方においては、被疑者の集合が小さい場合、敵に対してささいな疑いだけでそれらの全てに対しての行動を起こさせることになるかもしれない。

SAML 対応システムはオーソリティの利用のためにせいぜい部分的な anonymity に限定される。アサーションを作成されたエンティティはそれだけで、アサーションを発行しているオーソリティと関係を持つエンティティの集合のうちの一人であると識別される。
anonymity の制限は、識別子の使い方によっては、たとえば仮名識別子の再利用が情報を特定できる可能

性を増大させるように(セクション2.2.2を参照のこと)、単純なオーソリティとの関連性よりもより悪い状況になる可能性がある。さらに、SAML対応システムのユーザもまた、彼らの行動によってanonymityの侵害をひどくすることがあり得る(セクション2.2.3を参照)。

2.2.2. Pseudonymity and Anonymity

法的なアイデンティティは別として、サブジェクトに関する全ての識別子は仮名であるとみなされ得る。そして、鍵の所有者といった概念でさえサブジェクトに行動(もしくは行動の集合)を結びつけている仮名の同等なものとして同じように扱うことが認められる。オブジェクトXYZに23：34にアクセス要求をしただけのユーザのような定義でさえ、仮名の同等なものとして扱うことができる。

従って、害を加える能力を考慮すると、ユーザが識別子で表わされるのか、鍵の使用や行動といった振る舞いによって表されるのかには何の差異もない。

何が差異を生むかというと、特定の仮名と同等のものが使われる頻度である。

[Anonymity]はいつも使われるニックネームのような個人的な仮名から始まって、使い捨ての仮名としての様々なタイプの役割仮名（国防長官のような）まで、仮名の分類学を与えている。

使い捨ての仮名を用いないとanonymityを満たすことはできない(SAMLにおいて、これを“ある集合の範囲内でのanonymity”だと考慮せよ)。

与えられた仮名を何度も使えば使うほど、anonymityはなくなり、被害を受けやすくなる。言い換えると、仮名の再利用は仮名と関連するさらなる個人を特定する可能性がある情報を与えてしまう。長期にわたるとこれは、仮名と結びつくアイデンティティを一意に識別することができてしまうという結果につながる。

2.2.3. Behavior and Anonymity

Joe Kleinが証明しているとおりで、anonymityはいわれるほど良くはない。

Kleinは“Primary Colors”を書いた“Anonymous”である。彼の否認にも関わらず、Primary Colorの文書の法医学的な解析をしたVassarの教授であるDon Fosterによって彼が著者であるとして正体が暴かれた。Fosterは彼らの知識基盤に基づいて、書き方の傾向を考察した被疑者のリストからの文書とその文書を比較した。

それは、Kleinの特有の慣習であった(明らかに全ての著者がそれらを持つだろうけれども)。

SAMLに関係のあるポイントは、anonymousなユーザが(一度も名前がつけられてないものでさえ)、繰り返し行う一般的でない振る舞いによって、害を受けるのに十分な程度に識別され得るということである。いくつか例を示す。

- 毎週火曜の21時に、指の長さと寿命に関連するデータベースにアクセスするユーザはanonymousでなくなりはじめている。ユーザのほかの振舞いによって他の、本人を識別する情報が収集可能になるかもしれず、彼女もしくは彼を追跡可能になるかもしれない。
- 定期的に普通の商品をネットワーク上で販売している組織から買うユーザは、確実に彼ら自身が被害を受け得る状況にしている(ブービートラップの商品によって)。

2.2.4. Implications for Privacy

認証オーソリティと属性オーソリティのような基点サイトのオーソリティは使い捨ての識別子か鍵(holder of keyの場合)の採用によってある程度の部分的なanonymityを提供することが可能である。

サブジェクトはやむを得ずオーソリティと関係しているサブジェクトの集合へ限定されてしまうため、このanonymityはせいぜい部分的にしか満たされない。

基点となるサイトでさらに小さいユーザコミュニティで、集められた属性が使われるときにはこの集合はさらに縮小されるかもしれない(従ってさらにanonymityを低くする)。

真にanonymityを気にするユーザは、そのうちユーザのanonymityを侵害しかねない一般的でない行動パターンを隠すもしくは避けるように注意せねばならない。

3. Security

次のセクションではセキュリティに関して考慮すべき事柄を議論する。

3.1. Background

コンピュータベースのシステム間の通信はさまざまな脅威の対象となる。そしてこれらの脅威はいくつかの関連したリスクのレベルを持っている。リスクの本質は、通信の性質や通信システムの性質、通信媒体、通信環境、エンドシステムの環境などを含んだ多数の要素に依存する。RFC[Rescorla-Sec]のセキュリティの考慮すべき点を書いている IETF ガイドラインのセクション 3 ではインターネット(実装や、イントラネットによるものも)に内在する脅威の概要を提供している。

SAML はセキュリティドメイン内もしくは間においてアプリケーションレベルのコンピュータベースの通信のためのセキュリティコンテキストの構築で配置者(deployer)の役に立つことを目的としている。この役割を果たすことによって、SAML は通信セキュリティの(少なくとも部分的な)“エンドポイント認証”という側面と、システムセキュリティの“認可されていない利用”という側面の両方を解決する。通信セキュリティは SAML のデザインに直接適用できる。システムセキュリティは主に SAML の脅威モデルのコンテキストで対象となる。IETF ガイドラインのセクション 2 で通信セキュリティとシステムセキュリティの概要を提供している。

3.2. Scope

SAML を用いるシステムの全般的なセキュリティに広範に影響を与えるいくつかのエリアは明白に SAML の範囲外である。この文書はこれらの範囲を問題としないが、それらは、システムのセキュリティを概観する際には常に考慮されるべきである。特に以下の問題は重要であるが、現在は SAML の範囲を超えている。

- 最初の認証　：　SAML は認証がされたとして、作成されるステートメントを認めることになっているが、その認証に関しては何の要求も仕様も提供されていない。認証アサーションのコンシューマは認証アサーションが作成された根拠を知ることなしに、そして知るまでは、これらのアサーションを盲目的に信じることには慎重になるべきである。アサーションの確かさは、アサーティングパーティが主張している結論に正確に至っているという確かさを決して超えてはならない。
- トラストモデル　：　多くの場合、SAML 通信のセキュリティは典型的な鍵管理インフラストラクチャ(例えば PKI や秘密鍵)に基づく、基調を成すトラストモデルに依存している。例えば、XML 署名[XMLSig]によって安全にされている SOAP メッセージは、信頼され得る交換によって使われる鍵の場合にのみ保証される。気づかぬうちに危殆化した鍵や失効した証明書は、例えば、セキュリティの穴を作ることになり得る。証明書の要求をしなかったことでもなりすまし攻撃を許してしまうことになり得る。それ(例えば SAML の部分)のトップに構築されたレイヤが保証されるためには、PKI の構成が重要であり、正しく実装されなければならない。

3.3. SAML Threat Model

セキュリティに関する考慮点を述べた IETF のガイドライン[Rescorla-Sec]に記述されている、一般的なインターネットの脅威モデルが SAML の脅威モデルのベースとなる。我々は SAML トランザクションの 2 つかそれ以上のエンドポイントは危殆化していないが、攻撃者は通信チャンネルに完全なコントロールを持っているということをここに仮定する。

さらに、マルチパーティ認証・認可ステートメントプロトコルといった SAML の性質のため、そのトランザクションにおける役割内で合法的に機能する正規な SAML トランザクションでの一つまたは複数のパーティが次に起こるトランザクションに悪意を持って、前回のトランザクションから得た情報を使おうとすることを考慮しなければならない。

全ての場合において、システムがアサーションを作成するかしないかを決定するために用いるであろう局部的なメカニズムは SAML の範囲外である。したがって、例えば認証オーソリティでの最初のログインの詳細から生じる脅威は同様に SAML の範囲外である。もしもオーソリティが誤ったアサーションを発行するならば、そのとき下流部門のシステムによるそのアサーションの消費から生じる脅威は明らかに SAML の範囲外である。

そのようなスコーピングの直接の結果は、入力としてのアサーションに基づくシステムのセキュリティはそれらのアサーションを作成するために使われるシステムのセキュリティと同程度にだけよいということである。どの発行者を信じるかを決定しているとき、特にアサーションが認証もしくは認可決定への入力として使われるであろう場合において、正しくないが有効に発行されたアサーションの消費によって生じるセキュリティの危殆化のリスクは大きなものである。アサーティングパーティと信頼者の間の信頼ポリシは常に法的責任の重大な配慮を盛り込むために書かれるべきであり、実装は監査証跡を提供されなければならない。

4. Security Techniques

以下のセクションにて､SAMLの実装において利用可能なセキュリティの手法と様々な手持ちの技術を示す。

4.1. Authentication

ここでの認証とはトランザクションにおいて他のパーティのアイデンティティを確定するためのトランザクションをあるパーティが行えるということを意味する。この認証は一方的な場合もあるし、相互的なこともある。

4.1.1. Active Session

持続的でない認証はSAMLメッセージを運ぶために用いられる通信チャンネルによって提供される。この認証はセッションの開始者から受信者への一方向なもの、または双方向なものである。個別の方法は用いられている通信プロトコルによって決められるであろう。例えば、RFC2246[RFC2246]やIPセキュリティプロトコル[IPsec]のような安全なネットワークプロトコルの利用によってSAMLメッセージの送信者はTCP/IP環境のための送信先を認証することができる。

4.1.2. Message-Level

XML 署名[XMLSig]と the OASIS Web Services specifications[WSS]は、文書にしっかりと連結している普遍的な“認証”を行う方法を提供する。この方法はメッセージの送信者が実際にその署名者であることを独立的に保証はしない(実際に、仲介者を伴う多くの場合において、これは明白にそのケースではない)。
与えられたXMLメッセージの一部で一意的に決定できるエンティティの関与の持続的な確認を可能とするどんな方法もこの要求を満たすのに十分である。

4.2. Confidentiality

機密性とはメッセージの内容を意図した受信者のみが読むことができ、他の誰もそのメッセージに接触することがないという意味である。

4.2.1. In Transit

RFC2246[RFC2246]やIPセキュリティプロトコル[IPsec]のような安全なネットワークプロトコルを利用すると、2つのノード間での通信においてメッセージの一時的な機密性を実現することが可能である。

4.2.2. Message-Level

XML暗号[XMLEnc]によりXML文書の部分的な暗号化が可能である。この暗号メソッドは持続的に、XMLメッセージ内の要素の部分的な機密性の保護を可能とする。

4.3. Data Integrity

データの完全性とは受け取ったメッセージが、送られたメッセージのバージョンから変わっていないことを確かめることが可能なことである。

4.3.1. In Transit

RFC2246[RFC2246]やIPセキュリティプロトコル[IPsec]のような安全なネットワークプロトコルの利用はネットワーク接続を通して伝達されるパケットの完全性の検査CRCを提供できるように構成されるかもしれない。

4.3.2. Message-Level

XML 署名[XMLSig]はそのメッセージとしっかりと連結しているメッセージの不変の性質の持続的な保証を行う方法を提供する。
XMLメッセージの与えられた一部の不変な性質の持続的な確認を可能にするどんな方法もこの要求を満た

すのに十分である。

4.4. Notes on Key Management

この文書の多くの部分は電子署名と暗号化に関係するさまざまなスキームを通して認証、データの完全性、機密性を提供するためのシステムの能力について言及しているであろう。全てのこれらのスキームにとって、スキームによって提供されるセキュリティは適切な鍵管理システムに基礎を置いて限定される。いくつかの固有の制限の詳細を下記に示す。

4.4.1. Access to the Key

仮に鍵ベースのシステムが認証、データの完全性、否認防止のために用いられるならば、セキュリティは、適切でないパーティが鍵にアクセスすることができないことをきちんと保証すべき、ということが想定される。例えば、ボブの署名鍵で作成された電子署名は、ボブが鍵へのアクセスができる唯一の人である場合に限ったボブの関与の証明でしかない。

一般的に、鍵へのアクセスは最小限のエンティティの集合にだけ可能にされるべきであり(特に企業や組織に関する鍵にとって重要である)、パスワードもしくは他の手段で保護されるべきである。標準的なセキュリティ予防措置(パスワードを書き留めるな、コンピュータから離れる時は鍵を用いてアクセスしたウィンドウを開いたままにするな、など)が適用される。

4.4.2. Binding of Identity to Key

認証のために用いられる鍵ベースのシステムのために、アイデンティティの鍵への信頼された結びつきがなければならない。文書の電子署名を検証することによって、その文書が署名されて以来不変であるかどうか、そして与えられた鍵によって実際に署名されているということを判定することができる。しかしながら、これは用いられた鍵が実際に特定の個人の鍵であることを確かめようがない。

この鍵と個人の結びつけが達成されなければならない。一般的なソリューションとしては識別子と鍵の両方を保存しているローカルなディレクトリ(理解するにはシンプルであるが維持するのは難しい)、もしくは証明書の利用が挙げられる。

証明書(実質的にはアイデンティティと鍵の結びつきを署名したものである)は、殊更その問題に対しての強力な解決策であるが、それら自身の配慮を伴う。信頼されたルート認証局(CA)の集合はアイデンティティと鍵の結びつきの状態を作るために信頼するのは何か？という問い合わせに応答して、それぞれの署名のコンシューマに対して、本物であることを確認されなければならない。署名の検証はそれから、はじめにその署名を検証し(署名が問い合わせにおいてその鍵によって成されて、そのメッセージが変更されていないことを判定すること)、そしてそれから証明書チェーンの検証をする(その鍵が正しいアイデンティティに結びつくことを判定すること)という処理になる。

さらに、証明書を扱うステップにおいては、その結びつけが、現在有効であることを確かめなければならない。証明書は一般的に証明書に組み込まれた有効期間を持つが、もし鍵が証明書の有効期間の間に危殆化する場合は、証明書がそれの内容においてまだ有効であっても、証明書内の鍵とアイデンティティの結びつきは無効となる。また証明書はしばしば、有効期限が切れる前に終了するかもしれないアソシエーションに依存する(例えば、誰かが利用者を変えるとき、無効にするべき証明書など)。この問題は証明書失効リスト(CRL)によって解決される。証明書失効リストとは、それらの発行以来失効されている与えられた CA からの証明書のリストである。もう一つの解決策はオンライン証明書ステータスプロトコル(OCSP)である。それは与えられた証明書の現在の有効性を問い合わせるためにサーバを呼び出す方法を定義している。この同じ機能のいくつかは XML 鍵管理の仕様[XKMS]の上位レベルに組み込まれている。それは有効な鍵のために行われるためのリクエストを可能とする。

適切な鍵管理システムはしたがってかなり強力であるがとても複雑なものである。署名を検証することは結局、文書と鍵の結びつきの検証、それから鍵とアイデンティティの結びつきの検証、そして鍵と文書の結びつきの現在の有効性の検証という 3 ステージの処理をすることになる。

4.5. TLS/SSL Cipher Suites

HTTP での SSL3.0 や TLS1.0[RFC2246]の利用はこの文書の多くの場面で推奨されている。しかしながら TLS/SSL はたくさんの異なる cipher suites(それらの全てが最良のセキュリティの実践を提供するために適当であるわけではない)を用いるように設定されることができる。次のセクションで cipher suites の大まかな定義と cipher suite 選択の勧告を提供する。

4.5.1. What Is a Cipher Suite?

注：アメリカの輸出規制への参照が現在は廃止されているものの、cipher suites の普遍的につけられた名前は変わっていない。従って、SSL_DHE_DSS_EPORT_WITH_DES40_CBC_SHA（訳注：SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA の誤りと思われる）はいまだに有効な cipher suite の識別子であり、“EXPORT”の封入の歴史的な理由の説明は下記の概説に適切に任せられている。

cipher suite は 4 種のセキュリティ機能を組み合わせており、SSL プロトコル仕様に名前が与えられている。データが SSL 接続において流れる前に、両方のエンドポイントが cipher suite を取り決めようと試みる。これにより、利用可能である個々のメカニズムの組み合わせの制限内で、それらの通信の保護に適切な質で構築させる。cipher suite に関連した機能は下記のとおり。

1. 用いられている鍵交換アルゴリズムのタイプ。SSL はたくさん定義している、サーバ認証を提供するものはもっとも重要である、しかし anonymous な鍵交換がサポートされている(anonymous 鍵交換アルゴリズムは中間者攻撃の対象とされる、そして SAML コンテキストにおいて推奨されていないことに注意せよ)。“RSA”認証された鍵交換アルゴリズムは現在もっとも互換性のあるアルゴリズムである。もう一つの重要な鍵交換アルゴリズムは authenticated Diffie-Hellman “DHE_DSS”鍵交換で、特許関連の実装の制限がない。
2. 鍵交換アルゴリズムがアメリカから自由に輸出可能かどうか。輸出可能なアルゴリズムは鍵交換のために短い公開鍵(512 ビット)を、暗号化のために短い対称鍵(40 ビット)を使わなければならない。これらの鍵は現在、ほどほどに知識を持った攻撃者によって午後の時間内で解読できる対象である。
3. 用いられている暗号化アルゴリズム。最も速いオプションは RC4 ストリーム暗号で、DES と改良型(DES40、3DES-EDE)もまた null 暗号化(いくつかの suites で)として“cipher block chaining”(CBC)モードでサポートされている(Null 暗号化は何もしない、そのような場合 SSL は認証するためだけに使われ、完全性の保護を提供する。null 暗号化での cipher suites は機密性を提供しない、そして機密性が要求される場合には使うべきでない)。
4. メッセージ認証コードのために用いられるダイジェストアルゴリズム。選択肢は MD5 と SHA1 である。例えば、SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA と名づけられた cipher suite は SSL を用いて、authenticated Diffie-Hellman 鍵交換を用いて（DHE_DSS）、export grade であり(EXPORT)、輸出可能な DES 暗号の改良型を用いて(DES40_CBC)、それの MAC に SHA1 ダイジェストアルゴリズムを用いる(SHA)。

与えられた SSL の実装は特定の一組の cipher suites をサポートするであろう、そしてそれらのいくつかの部分集合はデフォルトで可能とされるであろう。アプリケーションはそれらの通信に用いられる cipher suites の制限されたある程度のコントロールを持つ。それらはサポートしている cipher suites のどれかを有効にすること、もしくは無効にすることができるが、利用可能である cipher suites を変更することはできない。

4.5.2. Cipher Suite Recommendations

下記に示す cipher suites は機密性とメッセージの完全性、という SAML の要求を十分に満たし、(X.509v3 証明書の存在を強いることによって)その上認証要求を満たすように構成されることが可能である。それらはまた、たくさんのクライアントアプリケーションによって十分サポートされている。これらの suites のサポートが推奨される。

- TLS_RSA_WITH_3DES_EDE_CBC_SHA(TLS を用いる際)
- SSL_RSA_WITH_3DES_EDE_CBC_SHA(SSL を用いる際)

しかしながら、IETF はスピードと強度の両方に優位点がある AES の利用を命じる方向に急速に動いている。将来的なシステムは下記のような AES cipher suites のサポートを実装したほうがよいであろう。

- TLS_RSA_WITH_AES_128_CBC_SHA

5. SAML-Specific Security Considerations

次のセクションではSAMLを利用して実装していく際のセキュリティリスクを分析し、そのリスクを軽減するための対応策を述べる。

5.1. SAML Assertions

SAMLアサーション自身のレベルではセキュリティに関して考慮すべきことはほとんどない。ほとんどの問題はリクエスト・レスポンスプロトコルでの通信においてや、バインディングの一つの手段によってSAMLを用いようと試みる間において発生する。もちろん、コンシューマにはアサーションの有効期間とアサーションに提示された<DoNotCacheCondition>要素を常にチェックすることが期待される。

しかしながら、アサーションレベルでの一つの問題として、アサーションが一度発行されると発行者にはコントロールできないという分析結果がある。この事実はたくさんの派生問題を抱えている。例えば発行者は、アサーションがどれくらい長くコンシューマのシステムに存続するかをコントロールできないし、コンシューマがアサーション情報をシェアするパーティのコントロールもできない。これらの懸念事項は、その上、暗号化されていない(もしくは暗号化が不十分な)通信路を通るアサーションの中身を見ることができる悪意のある攻撃者についての懸念もある。

SAML仕様内においてはこれらの問題点の多くを解決しようと努力がされているが、仕様に含まれるどんなことも、アサーションに入れるべき事柄の注意深い考慮の要求を消すことにはならない。アサーションの情報がリモートサイトに格納されるならば(そこでは直接悪用される、または潜在的なハッカーに公開される、またはより自分の都合のよいように詐欺に利用するための格納が可能である)、発行者は常に起こり得る影響を考慮すべきである。発行者もまた、アサーションの情報は故意もしくは不注意のどちらかにより他のパーティにシェアされるし、もしくは公開されさえする可能性を考慮すべきである。

5.2. SAML Protocol

次のセクションでは個別のプロトコルバインディングの利用から生じる脅威は除いて、SAML リクエスト・レスポンスプロトコル自身のセキュリティに関して考慮すべき事項を示す。

5.2.1. Denial of Service

SAMLプロトコルはDoS攻撃の影響を受けやすい。SAMLリクエストを処理するのは、リクエストメッセージの構文解析すること(一般的にDOM treeの構築を必要とする)やデータベース/アサーションの記録の参照(もしかすると索引のない鍵で)、レスポンスメッセージの作成、もしかすると一回かそれ以上の電子署名の演算といったことを含んで、とても高額な操作になりかねない。したがって、リクエストを生成する攻撃者によって要求される作業量はそれらのリクエストを処理するのに必要とされる作業量よりも非常に小さいものとなる。

5.2.1.1. Requiring Client Authentication at a Lower Level

クライアントに対して SAML プロトコルレベルより下のいくつかのレベルでの認証を要求することは(例えば、HTTP over TLS/SSLを用いてSOAP over HTTPバインディングを用いて、そして証明書のルートに信頼された認証局をもつクライアントサイド証明書を要求すること)、DoS攻撃の場合にトレーサビリティを提供するであろう。

もし認証がトレーサビリティを提供するためだけに用いられるのならば、そのときこれはそれ自身で攻撃が起こることを防ぐことはしないが、抑止力としての機能は果たす。

もし認証がいくつかのアクセスコントロールシステムと併用されるならば､そのとき部内者以外からのDoS攻撃は効果的にブロックされる(クライアント認証スキームの使い過ぎはいまだにSAMLサービスへのDoS攻撃として機能し得ることが可能であるが、この攻撃は選択されたクライアント認証スキームに照らして対応される必要があることに注意せよ)。

どんなクライアント認証システムが使われても、それぞれのリクエストの一意な作成者を特定する能力を提供すべきであり、サブジェクトは偽造されるべきでない(例えば、トレーサビリティのみの場合 IP アドレスのロギングはこの情報は容易になりすまされ得るため十分ではない)。

5.2.1.2. Requiring Signed Requests

セクション 5.2.1.1 で述べたクライアント認証から得られる利点に追加して、署名されたリクエストを要

求することもまた、リクエスタとレスポンダによってなされる作業の間の非対称性のオーダを減少させる。電子署名を計算する処理はリクエスタの作業の比較的大きな量に相当する一方で、署名を検証するためにレスポンダの要求される追加の作業はリクエスタの要求される全ての作業の中では比較的に小さな割合である。この非対称性を削減させることでDoS攻撃に関連するリスクが減少する。

しかしながら攻撃者は、理論上はこの要求された作業を回避するために、署名されたメッセージをキャプチャすることができ、さらにそれを継続的に再送できることに注意せよ。この状況は署名が新しいものであるかどうかを判定するために用いられることが可能なタイムスタンプを含んでいるXML署名要素<ds:SignatureProperties>の利用を要求することによって避けることが可能となる。この場合、署名が有効であるとして扱われる発行後の時間の幅を狭くするほど、DoS攻撃の再送に対してのセキュリティは強くなる。

5.2.1.3. Restricting Access to the Interaction URL

既知のパーティの集合がSAMLサービスへリクエストを発行するための能力をとても低いレベルに制限することはDoS攻撃のリスクを大幅に減らす。この場合、限定された既知のパーティの集合内から起こった攻撃しか不可能であるため、潜在的な悪意のあるクライアントへと、ゾンビとして危殆化されたマシンを用いたDoS攻撃への両方に対して障害を大いに減らすことになる。

SAMLレスポンダを保護されたイントラネット内に置き、ルータレベルでアクセスルールを実装するといったアクセスを制限可能な方法はたくさんある。

5.3. SAML Protocol Bindings

SAMLリクエスト・レスポンスプロトコルのデザインのセキュリティに関して考慮すべきことは大体において、用いられている固有のプロトコルバインディング(SAMLのバインディングの仕様[SAMLBind]に定義されているように)に依存する。現在のところ、the OASIS Security Services Technical Committeeによって認可されている唯一のバインディングはSOAPバインディングである。

5.3.1. SOAP Binding

SAML SOAPバインディングは通信中の機密性もしくはメッセージの完全性のために認証を要求せず、要求事項も持たないので、次のセクションで述べるような多種多様な一般的な攻撃に無防備である。一般的な考慮すべき事柄はSOAP-over-HTTPの場合に関係した考慮事項とは独立して議論される。

5.3.1.1. Eavesdropping

通信における機密性の要求がないため、盗聴するパーティが、リクエストを含んでいるSOAPメッセージと対応するレスポンスを含んでいるSOAPメッセージの両方を獲得することが可能である。この収集によりリクエストの性質とそのレスポンスの詳細（一つかそれ以上のアサーションを含んでいる可能性がある）の両方が露出してしまう。

リクエストの詳細の露出はいくつかの場合において、それが要求しているアサーションの種類の詳細を明らかにすることによって、もしくは誰にそれらのアサーションが要求されたかということから、要求しているパーティの安全性を弱めるであろう。例えばもしも盗聴者が、サイトXは頻繁にサイトYから与えられたconfirmation methodで認証アサーションを要求していることを判定できるなら、盗聴者はサイトXを危殆化させる目的でこの情報を使うことができるかもしれない。

同様に、一連の認可クエリの盗聴により、与えられた認可オーソリティの制御下にあるリソースの“map”を作成できてしまう。

さらに、いくつかの場合においては、リクエストそれ自身の露出によりプライバシの侵害となることがある。例えば、クエリとそれのレスポンスの盗聴は与えられたユーザがそのクエリをしているサイトにアクティブであるということを露出してしまうかもしれない。それは医療情報サイトや政治に関わるサイトなどのような場合には明かされるべきでない情報である。また、レスポンスで運ばれたアサーションの詳細も機密に保たれるべき情報であるかもしれない。これは特に属性アサーションを含んでいるレスポンスに対して言えることである。もしこれらの属性がトランザクションに対してのパーティでないエンティティには利用不可能であるべき情報を表している場合（クレジットの料金、医療属性など）、盗聴によるリスクは高い。

これらのリスクを考慮する場合、盗聴攻撃に対する対策は通信中のメッセージの機密性のなんらかの形式を与えることである。SOAPメッセージでは、この機密性はSOAPレベルかSOAPトランスポートレベル（もしくはそれより下のレベル）のどちらかで強化されることが可能である。

SOAPレベルでの通信中の機密性を付け足すことはSOAPメッセージをSOAP通信に関係なく、意図されたパ

ーティ以外は誰もそのメッセージにアクセスできないように構成することを意味する。この問題に対する一般的な解決策はXML 暗号化[XMLEnc]になるであろう。この仕様はSOAP メッセージそれ自身の暗号化、それにより暗号化に使われる鍵が危殆化していない限り盗聴のリスクを解消できるものを与えている。その代案として、配置者(deployer)は通信中の機密性を提供するために、SOAP トランスポートレイヤやその下のレイヤに依存することもできる。

この機密性の提供方法の詳細は選択された固有のSOAP トランスポートに依存する。HTTP over TLS/SSL（セクション 5.3.2 にさらに述べられている）を用いることも一つの方法である。他のトランスポートは他の通信中の機密性保持技術(例えばSMTP トランスポートはS/MIME を用いるかもしれない)を必要とするであろう。

いくつかの場合において、SOAP トランスポートの下位レイヤは要求された通信中の機密性を提供するかもしれない。例えば、もしもリクエスト・レスポンスのインタラクションが IPSec トンネルで行われれば、そのとき十分な通信中の機密性がトンネルそれ自身によって提供されるかもしれない。

5.3.1.2. Replay

SOAP バインディングのレベルにおいては再送攻撃への脆弱性はほとんどない。再送はさまざまなプロファイルにおいての方がよりいっそう問題となる。SOAP バインディングレベルでの再送についての主要な考慮すべき事柄はDoS 攻撃の手法としての再送の利用の可能性である。

一般的に、再送攻撃を防ぐ最良の方法はファーストプレイスでのメッセージキャプチャを防ぐことである。通信中の機密性を提供するために用いられるトランスポートレベルのスキームのいくつかはこの目的を達成するであろう。例えば、もしもSAML リクエスト・レスポンスのやり取りがHTTP/TLS でのSOAP 通信でなされるならば、サードパーティがそのメッセージをキャプチャすることを防げる。

再送攻撃を行う者にとってはメッセージを再送するためにそのメッセージを理解する必要がないため、XML 暗号化のようなスキームは再送に対しての保護は提供しない。もしも攻撃者が、リクエスタによって署名されてレスポンダ宛に暗号化してあるSAML リクエストをキャプチャできるならば、そのとき攻撃者はその暗号化をもう一度実行する必要性もなくいつでもそのリクエストを再送できてしまう。リクエストの発行時間の情報を含んだSAML リクエストは再送がされたのかどうかを判断することを可能とする。もう一つの方法として、リクエストの一意の鍵(RequestID)を、リクエストが再送リクエストであるかどうかを判断するために使われることが可能である。

さらに、再送攻撃からの脅威として“charge per request”モデルが実施されているケースがある。再送を、与えられたアカウントに対して大量にチャージをためるために用いることが可能である。

同様に、クライアントがシステムに固定数のインタラクションを割り当てられている(もしくは獲得する)モデルでは、発行者がそれぞれのリクエストの一意な鍵を把握するように注意しない限り、再送攻撃はこれらの利用を使い果たしてしまう可能性がある。

5.3.1.3. Message Insertion

SOAP バインディングへのメッセージ挿入攻撃は要するにリクエストの作成ということになる。リクエストを作成する能力はSOAP バインディングレベルでの脅威ではない。

5.3.1.4. Message Deletion

メッセージを削除する攻撃とは、リクエストが受信者に着かないようにするか、レスポンスがリクエスタに着かないようにするかのどちらかになるであろう。

どちらの場合でも SOAP バインディングはこの脅威を問題としない。SOAP プロトコルそれ自身とその下の通信はどのようにそのメッセージ削除が達成されたかに応じていくつかの情報を提供することができる。

このリスクを弱める信頼できるメッセージングシステムの例として、トランスポートレイヤで信頼できるHTTP(HTTPR)[HTTPR]と、マイクロソフトのSRMP for MSMQ[SRMPPres]のようなSOAP での信頼できるメッセージング拡張の利用が挙げられる。

5.3.1.5. Message Modification

メッセージの変更は両方の順路でSOAP バインディングへの脅威となる。

リクエストの詳細を改ざんするためのリクエストの変更は、返されるべき結果とは著しく異なる結果を引き起こす。言い換えると、賢い攻撃者によって、返されるアサーションに依存しているシステムを危殆化させるために利用され得るのである。例えば<AttributeDesignator>要素の要求された属性のリストを改ざんすると、レスポンダによるリクエストの危殆化もしくは拒否という結果につながり得る。

リクエストの明白な発行者を改ざんするためにリクエストを変更することはサービスの拒否もしくはレスポンスの誤った経路指定という結果になる可能性がある。この改ざんはSAML レベルの下で起こる必要がある

であろう、したがってSAMLの範囲外である。
アサーションの詳細を改ざんするためのレスポンスの変更はそこに膨大な危殆化という結果になる可能性がある。認証もしくは認可決定の詳細を改ざんするシンプルな例がとても重大なセキュリティの危殆化を導く可能性がある。
これらの起こり得る脅威を解決するために、通信中のメッセージの完全性を保証するシステムが使われなければならない。SAML プロトコルと SOAP バインディングは通信中のメッセージの完全性を保証するシステムの採用を要求も妨げもしない、しかしこの大きな脅威のためにそのようなシステムが使われることが強く推奨される。SOAPバインディングレベルで、これはXML署名[XMLSig]のようなシステムによって電子署名がなされたリクエストとレスポンスによって達成され得る。SAML仕様はそのような署名を認めている、さらなる情報はSAMLのアサーションとプロトコルの仕様[SAMLCore]を参照のこと。
もしもメッセージに電子署名がなされているならば(賢明な鍵管理インフラストラクチャで。セクション4.4参照)、そのとき受信者は用いられている鍵が危殆化していない限りそのメッセージは通信中に改ざんされていないという保証を持つ。
通信中のメッセージの完全性の目的は、保証された完全性の性質を提供するもしくはそのような性質を提供するプロトコルに基づいている SOAP 通信を用いることによって低いレベルで達成されることが可能である。SOAP over HTTP over TLS/SSLはそのような保証を提供する通信である。
暗号化だけではこの保護を提供しない。というのは、たとえ傍受したメッセージを改ざんすることは不可能だとしても、それを新しく作ったものと置き換えることは可能である。

5.3.1.6. Man-in-the-Middle

SOAPバインディングは中間者(MITM)攻撃の影響を受けやすい。悪意のあるエンティティが中間者として操作することを防ぐために(盗聴とメッセージ変更のセクションの両方で述べた全ての危険性によって)、なんらかの種類の相互認証が要求される。
相互認証システムは両方のパーティに、やり取りしているデータが実際にそのやり取りのもう一方のパーティから送られてきていることを確かめることを可能にする。
SOAPバインディングレベルで、この目的もまた電子署名がなされたリクエストとレスポンスによって達成されることが可能である(セクション 5.3.1.5 で述べた全ての警告でもって)。この方法は盗聴者が真ん中に割り込み、往復とも転送することを防ぐことはしない、しかし検知されることなしにどちらの道でもやり取りを改ざんされることは防げる。
SOAPのたくさんのアプリケーションはセッションを使わないので、このオーサーの認証のようなものは(送信者の認証に対抗するものとして)、“盗聴者としての中間者”の弱形を防ぐために、送信者とオーサーが同じパーティであることを確かめるためにトランスポートレイヤからの情報と組み合わさる必要があるかもしれない。
もう一つの実装は、相互認証を提供するSOAP通信に依存するか、もしくは相互認証を提供する低いレベルに実装されるかである。これの例は再び、サーバサイド、クライアントサイド証明書の両方が要求されるSOAP over HTTP over TLS/SSLである。
さらに、アサーションの有効期間は中間者攻撃からのリスクの度合いへの調整としての機能に反響する。アサーションの有効期間が短いほど、傍受された際のダメージはより小さくなる。

5.3.2. Specifics of SOAP over HTTP

SOAP バインディングは Basic over server-side SSL と certificate-backed authentication over server-side SSL のようなたくさんの様々な相互認証メソッドで適切なアプリケーションが HTTP over TLS/SSL をサポートすることを要求するので、これらのメソッドは、他の低いレベルのシステムが利用不可能で、これまでに挙げた攻撃が重大な脅威であると考慮される場合には脅威を軽減するために常に利用可能である。
これは相互認証のなんらかの方法を伴った HTTP over TLS の利用が義務であるという意味ではない。もしも様々なリスクからの保護の容認可能なレベルが他の手段を通して(例えば、IPSec トンネルによって)到達されることが可能ならば、full TLS with certificatesは要求されない。しかしながら、SOAP over HTTPの場合の大多数において、相互認証を伴ったHTTP over TLSを用いることは妥当な選択であるだろう。
しかしながら、トランスポートレベルのセキュリティの利用(SSLやTLSプロトコルunder HTTPのような)は、ワンホップに対しての機密性と/もしくは完全性と/もしくは認証だけを提供することに注意せよ。仲介者がいるかもしれないモデルや問い合わせでのアサーションがワンホップ以上存続する必要があるというモデルにおいて、HTTP with TLS/SSLの利用は十分なセキュリティを提供しない。

5.4. Profiles of SAML

SAML バインディングの仕様[SAMLBind]は加えて SAML のプロファイルを定義している。それはフレームワークやプロトコルへの SAML アサーションの埋め込み方とフレームワークやプロトコルからの SAML アサーションの抽出の仕方を記述しているルールセットである。現在、the OASIS Security Services Technical Committee によって認定された SAML のプロファイルは 2 つある。

- シングルサインオン(SSO)をサポートするウェブブラウザベースの 2 つのプロファイル
  - SAML ブラウザ／アーティファクトプロファイル
  - SAML ブラウザ／POST プロファイル

(The OASIS Web Services Security Technical Committee は SAML のもう一つ別のプロファイルを提供している。それは、ウェブサービスメッセージを保護するための SAML アサーションの使い方が記述してあるドラフト“SAML token profile”of the WSS specification[WSS-SAML]である。)

5.4.1. Web Browser-Based Profiles

次のセクションでは SAML のブラウザ／アーティファクトとブラウザ／POST プロファイルに共通するセキュリティに関して考慮すべき事柄を記述する。

ソース Web サイトでのユーザの認証は、それから生じる全ての問題のように、明白に SAML の範囲外であることに注意せよ。ポイントとなる概念はソースシステムエンティティが、それがインタラクションしている認証されたクライアントシステムエンティティが次のインタラクションステップでのそれと同じであるということを確かめることができなければならないということである。これを達成するための一つの方法は、これらの最初のステップのために、この最初のインタラクションのために用いられているプロトコルの下部のセッションレイヤとして TLS の利用が執行されることである（HTTP のように）。

5.4.1.1. Eavesdropping

盗聴の可能性は全てのウェブブラウザのケースに存在する。機密性が要求される場合においては(安全に送られていないアサーションはそれに関連したリクエストと同様に、悪意のある盗聴者に利用可能であるということを考慮した態度で)、HTTP 通信は機密性を保証する通信路上で行われる必要がある。HTTP over TLS/SSL[RFC2246]と IP セキュリティプロトコル[IPsec]がこの要求に合う。

次のセクションでは盗聴の脅威についてより詳細に述べる。

5.4.1.1.1. Theft of the User Authentication Information

サブジェクトが認証情報を示すことによってソース Web サイトに認証される場合、例えば、パスワードフォームでの、認証情報の盗難は敵にサブジェクトになりすますことを可能にするであろう。

この問題を避けるために、サブジェクトのブラウザとソース Web サイトの間の接続は機密性を保護する手段を実装しなければならない。加えて、そのような手段は、サブジェクトもしくは目的 Web サイトが、認証情報を示す前に、そのソース Web サイトが真に予期している信頼したソース Web サイトであることを確かめるために行われなければならない。HTTP over TLS を用いると、この考慮事項を解決できる。

5.4.1.1.2. Theft of the Bearer Token

認証アサーションがアサーションを運ぶ人の認証プロトコル識別子を含む場合、アーティファクトの盗難は敵に、サブジェクトになりすますことを可能としてしまうだろう。

次に示すそれぞれの方法はこれが起こる可能性を減らす。

- 目的 Web サイトがサブジェクトのブラウザとの接続において機密性を保護する手段を実装する。
- サブジェクトか目的 Web サイトが、ソース Web サイトがサブジェクトのブラウザとの接続において機密性を保護する手段を実装していることを確かめる。
- 目的 Web サイトが、サブジェクトのブラウザが直接サブジェクトを認証したソース Web サイトによって直接リダイレクトされていることを検証する。
- ソース Web サイトが同じアサーション ID に対応するアサーションの 2 回以上のリクエストへの応答を拒否する。
- アサーションが特定のドメインを識別する、AudienceRestrictionConditionType 型の condition 要素を

含むときは、目的Webサイトはそのドメインのメンバであることを検証する。
- 目的Webサイトとソース Webサイトの間のアサーション IDが渡される接続が、機密性を保護する手段で実装される。
- 目的Webサイトはアサーション IDが渡されるソース Webサイトとの通信において、ソース Webサイトが真に予期される信頼されたソース Webサイトであることを検証しなければならない。

5.4.1.2. Replay

この一連のプロファイルには、再送攻撃の可能性が存在する。再送攻撃はサービスを妨害しようとするためか、不正に情報を読み出すために用いられる。具体的な対抗手段は、どの特有のプロファイルが使われているかということに依存する。したがって、それらはセクション 5.4.2.1 と 5.4.3.1 で述べる。

5.4.1.3. Message Insertion

メッセージ挿入攻撃はこの一連のプロファイルでは一般的な脅威ではない。

5.4.1.4. Message Deletion

ブラウザと SAML アサーションの発行者と SAML アサーションのコンシューマの間でのインタラクションのいずれかのステップの間でのメッセージの消去は、インタラクションの失敗を引き起こす。それはいくつかのサービスの妨害を引き起こすが、何か情報の露出を増やしてしまうものではない。
SAML バインディングとプロファイルの仕様はメッセージ消去に対して何の対抗策も提供しない。

5.4.1.5. Message Modification

ストリームにおけるメッセージの改ざんの可能性はこの一連のプロファイルに対して存在する。いくつかの可能性のある好ましくない結果は下記のとおりである。

- 最初のリクエストの改ざんは、SAML 発行者で拒否される、もしくは要求されたリソースでない異なるリソースを目的としたアーティファクトを作成されるという結果をもたらし得る。
- アーティファクトの改ざんは SAML コンシューマにサービスの妨害という結果をもたらす。
- 通信中におけるアサーションそれ自身の改ざんは全ての種類の悪い結果をもたらす(もしもそれらが署名されていないならば)。もしくは、サービスの妨害という結果をもたらす(もしもそれらが署名されていてコンシューマがそれらを拒絶するならば)。

メッセージの変更を避けるために、データはエンドポイントからエンドポイントへのメッセージの完全性を保証するシステムの手段によって運ばれる必要がある。
ウェブブラウザベースのプロファイルでは、通信においてメッセージの完全性を提供する推奨手段はデータの完全性確認を提供する cipher suite を用いた HTTP over TLS/SSL を利用することである。

5.4.1.6. Man-in-the-Middle

中間者攻撃はこの一連のプロファイルにとっては特に致命的な攻撃である。中間者はリクエストを中継し、返ってきたアサーションやアーティファクトをキャプチャし、誤ったそれを返すことができる。その時、本来のユーザは問い合わせのリソースにアクセスすることができないが、中間者はキャプチャしたリソースを使ってそうすることができる。
この脅威を防ぐためにはたくさんの対抗手段が要求される。第一に、強い相互認証を提供するシステムを用いることで中間者にとって対話の中に入り込むことが非常に難しくなるであろう。
しかしながら中間者が存在する可能性はまだ残っている。中間者はまったく双方向のポートフォワーダーとして行動し、返されるアサーションをキャプチャする目的で情報を盗聴する、もしくはまったくハンドラーとして行動する(リクエスタへの最後の返信を改ざんすることが可能である)。機密性が保護されたシステムにすることで盗聴は防げるであろう。データの完全性を保護するシステムをおくことで、ポートフォワーディングの間でのメッセージの改ざんを防げるであろう。
もしも TLS/SSL レイヤが適切な cipher suite を用いて(機密性を提供するのに十分強い暗号で、データの完全性をサポートする)、認証に X509v3 証明書を要求するならば、この一連のプロファイルで、強い相互のセッション認証、機密性、データの完全性という全ての要求が HTTP over TLS/SSL の利用によって満たされ得る。

5.4.2. Browser/Artifact Profile

ブラウザ／アーティファクトプロファイルへのたくさんの固有の脅威と対応策は SAML バインディング仕様[SAMLBind]に標準として文書化されている。追加の非標準なコメントを下記に示す。

5.4.2.1. Replay

アーティファクトの再利用という再送の脅威はそれぞれのアーティファクトが使い捨てのものであるという要求によって解決される。システムは同じアーティファクトを参照している複数のリクエストが作成されているケースを、この状況は侵入を試みることに相当するものであるとして探知するべきである。
結果としてアサーションを作成する元のリクエストにおける再送の脅威はSAML によって解決されない。しかし、最初の認証プロセスによって軽減されるべきである。

5.4.3. Browser/POST Profile

ブラウザ／POST プロファイルへのたくさんの特有な脅威と対応策は SAML バインディング仕様[SAMLBind]に標準文書化されている。追加の非標準なコメントを下記に示す。

5.4.3.1. Replay

再送攻撃は保護されたリソースに不正にアクセスするためのフォームの再提出を意味する。プロファイルは、受け渡されたアサーションが再送攻撃の成功を防ぐために目的 Web サイトでの使い捨ての性質を持つことを命じている。

6. References

以下省略

付録 D：用語集

この用語集は以下の文献を元に作成した。

- Glossary for the OASIS Security Assertion Markup Language (SAML) v1.1
  sstc-saml-glossary-1.1-draft-05
- Liberty 技術用語集 バージョン 1.2
  liberty-glossary-v1.2-JP.pdf
- Web Services Federation Language (WS-Federation)
  http://www-106.ibm.com/developerworks/library/ws-fed/#onefiveterminology

A

Access / アクセス　{ SAML }
システムエンティティのリソースにおける一部もしくは全部についての情報を操作、利用、取得するため、またはその表示を獲得するため、あるいはその両方のために、システムエンティティとやりとりすること。

Access Control / アクセス制御　{ SAML, Liberty }
権限のないアクセスに対するシステムリソースの保護；システム資源を使用するプロセスは、セキュリティポリシーによってコントロールされ、権限を持った主体 (ユーザ、プログラム、プロセス、または他のシステム) によってのみセキュリティポリシーに基づいて許可される。[RFC2828]

Access Control Information / アクセス制御情報　{ SAML }
アクセスコントロール目的で用いられる任意の情報。文脈情報(contextual information) [X.812] を含む。文脈情報は、接続元 IP アドレス、暗号強度、要求されている操作タイプ、時刻などを含むかもしれない。アクセス制御情報の一部はリクエストそのものに特有かもしれない。あるものは、リクエストが送信される接続に関係しているかもしれないし、他のもの (例えば時刻) は「環境」に関係しているかもしれない。

Access Rights / アクセス権限　{ SAML)
主体者がリソースに対して所有可能な、認可された相互作用 (authorized interactions) タイプの記述。例えば、読み込み (read) 、書き込み (write) 、実行 (execute) 、追加 (add) 、変更 (modify) 、削除 (delete) を含む。

Account / アカウント　{ Liberty }
定期的な取引やサービスの提供に関して、主体者とサービスプロバイダ間で交わされた正式な契約。

Account Linkage / アカウントのリンク　{ Liberty }
「Identity Federation」を参照。

Active Requestors / アクティブな要求者　{ WSF }
WS-Security と WS-Trust で記述しているような Web サービスメッセージを発行することができるアプリケーション（Web ブラウザもありえる）

Active Role / 有効な権限　{ SAML }
ある操作（例えばリソースへのアクセス）を実行するときに、システムエンティティが取得したロール。

Administrative Domain / 管理ドメイン　{ SAML }
次に挙げるものの一つ以上の組合せによって定義される環境もしくは状況。管理ポリシー、インターネットドメイン名登録、市民の法律上の実体（例えば、個人、企業、他の正式に組織された実体）、加えて、ホストの接続、ネットワークデバイス、相互接続されたネットワーク（それに可能な他の特性）、加えて、（しばしば様々な）ネットワークサービスおよびこれらの上で実行されるアプリケーション。管理ドメインは、一つ以上のセキュリティドメインを含むか定義するかもしれない。管理ドメインは、単一サイトもしくは複数サイトを含むかもしれない。管理ドメインを定義するこれら特性は、時間とともに発展するかもしれない（そして多くの場合は発展するだろう）。管理ドメインは相互に作用し合意を結ぶかもしれない。それは、管理ドメインの境界を横切ってサービスを提供、および/または (and/or) 消費するためである。

Administrator / 管理者　{ SAML }
（例えば SAML セキュリティシステムをベースとする）システムをインストールもしくは維持する人。もしくは、システムエンティティ、ユーザ、かつ／またはコンテンツ（アプリケーション目的と相対するものとして; エンドユーザを参照。）を管理するためにシステムを使用する人。管理者は一般的に特定の管理ドメインに加わる。そして、一つ以上の管理ドメインに加わるかもしれない。

Affiliation (Affiliation Group)　/ アフィリエーション　{ Liberty }
アフィリエーションとは、プロバイダ ID で記述した 1 つまたは複数のエンティティの集まりであり、その中でメンバーとして Liberty の対話を行うことができる。アフィリエーションは、1 つのアフィリエーション ID で照会し、プロバイダ ID によって特定された 1 つのエンティティがこれを管理する。アフィリエーションに参加するメンバーは、アフィリエーションのメンバーとして（アフィリエーション ID を使用）または単独で（プロバイダ ID を使用）サービスを呼び出すことができる。「アフィリエーション」と「アフィリエーショングループ」は同義。

Anonymity / 匿名　{ SAML }
匿名である特性もしくは状態。これは、未知もしくは秘密にされている名前や識別情報 (identity) を持っている状態である。

APL　{ Liberty }
属性プロバイダ(AP)は、ID-PP 情報を提供する。ID-PP プロバイダとも言う。AP は、ID-PP をホスティングする ID-WSF Web サービスである。

Artifact / アーティファクト { SAML }
完全な SAML アサーションを指し示すためのランダムで小さな数値。SAML アーティファクトは、ブラウザの URL クエリ文字列によってサイト間で渡される。[SAMLBind11],[SAMLCore11]

Asserting Party / アサーティングパーティ { SAML }
正式には一つ以上の SAML オーソリティをホストする管理ドメイン。いわゆる SAML オーソリティのインスタンスのこと。

Assertion / アサーション { SAML, Liberty)
SAML オーソリティによって生成されるデータの一部。主体者に対する認証行為、主体者についての属性情報、指定されたリソースについてサブジェクトに適用される認可の許可 (authorization permissions) のいずれかである。

Association / 連合　{ WSF }
主体(principal)が信頼領域(trust realm)や連携(federation)と連合あるいは合併していくプロセス。

Attribute / 属性　{ SAML, Liberty }
(主体者の、SAML における)あるオブジェクトの明確な特徴。属性を定義することでオブジェクトを表す。実世界のオブジェクトにおける属性は、大きさ、形、重さ、色などの物理的な特徴をあらわす用語によってしばしば指定される。仮想空間におけるオブジェクトは、サイズ、エンコードの型、ネットワークアドレスなどを記述する属性をもつかもしれない。オブジェクトのどの属性が重要であるかは、オブザーバーによって決定される。「XML Attribute」も参照のこと。

Attribute Assertion / 属性アサーション　{ SAML }
主体者の属性に関する情報を運ぶアサーション。

Attribute Authority / 属性オーソリティ　{ SAML }
属性アサーションを生成するシステム。

Attribute Class / 属性クラス　{ Liberty }
あらかじめ定義された属性のセット。たとえば、主体者名の構成要素（プレフィクス、ファーストネーム、ミドルネーム、ラストネーム、およびサフィックス）などがある。Libertyのエンティティは、これらのクラスを標準化することができる。

Attribute Container / 属性コンテナ　{ Liberty }
使用目的に従ってグループ化した属性の集まりによって構成されるモジュールを指す。

Attribute Provider (AP) / 属性プロバイダ　{ Liberty }	
属性プロバイダ（AP）は、ID-PP（アイデンティティパーソナルプロファイル）情報を提供する。ID-PP プロバイダとも言う。AP は、ID-PP をホスティングする ID-WSF Web サービスである。	

Attribute Service / 属性サービス　{ WSF }
信頼領域やフェデレーション内にある主体者に関する属性情報を保持する Web サービス。ここでは、主体者という言葉は単なる人ではなく、任意のシステムの実体を指す。

Authenticated Principal / 認証された主体者　{ Liberty }
アイデンティティプロバイダからアイデンティティ認証を受けた主体者。

Authentication(AuthN) / 認証　{ SAML, Liberty }
指定もしくは理解された信頼レベルをもつ、システムの主張された主体識別を確認すること。

Authentication Assertion / 認証アサーション　{ SAML }
主体者が認証に成功したという情報を運ぶアサーション。

Authentication Assertion Context (AAC) / 認証アサーションコンテキスト(AAC)　{ Liberty }
資格付与の判断を下す前に、サービスプロバイダが認証アサーション自体に加えて要求する可能性のある情報。

Authentication Authority / 認証オーソリティ　{ Liberty }
認証アサーションを作成するシステムエンティティ。

Authentication Domain (AD) / 認証ドメイン　{ Liberty }
認証ドメイン（AD）とは、Liberty に対応するエンティティの公式コミュニティであり、既知の共通ルールセットに従って対話が行われる。

Authentication Quality / 認証クオリティ　{ Liberty }
サービスプロバイダがアイデンティティプロバイダから受け取った認証アサーションにおくことができる保証レベル。

Authentication Session / 認証セッション　{ Liberty }
A が B を認証した後から、A が B の ID アサーションの信頼を中止して再認証を要求するまでの時間。単に「セッション」と呼ばれることもあり、主体者による正常なログインからログアウトまでの状態を指す。

Authorization (AuthZ) / 認可　{ SAML, Liberty }
適用されるアクセス制御情報を査定することによって、対象ユーザーが特定のリソースに対する特定のタイプのアクセス権を有しているかどうかを判断するプロセス。通常、認証コンテキストに沿って認可を行う。認証されたユーザーに、異なるタイプのアクセス実行権限が付与される場合がある。

Authorization Decision / 認可決定　{ SAML }
認可行為の結果。この結果は否定かもしれない。すなわちそれは、リソースへのどのようなアクセスも主体者には許可されないことを示すかもしれない。

Authorization Decision Assertion / 認可決定アサーション　{ SAML }
認可決定についての情報を運ぶアサーション。

B

Binding, Protocol Binding / バインディング、結合プロトコル　{ SAML }
特定のプロトコルに対して SAML 要求･応答メッセージ交換をマッピングするインスタンス。おのおのの結合は、「SAML○○結合」という形式の名前を与えられる。

C

Certificate Management / 証明書管理　{ Liberty }
証明書のライフサイクルの間に、デジタル証明書の発行者が果たす以下のような機能。[RFC2828] ・ 証明書にバインドするデータ項目の取得と確認。 ・ 証明書の符号化と証明書への署名。 ・ ディレクトリまたはレポジトリでの証明書の保管。 ・ 証明書の再発行と鍵更新、内容更新。 ・ 証明書の失効と CRL の発行。

Certificate Policy (CP) / 証明書ポリシー (CP) { Liberty }
特定のコミュニティやアプリケーションのクラスに対する証明書の適用範囲を示す一連の規則。たとえば、証明書ポリシーでは、所定の価格帯の B2B 取引における参加者の認証には特定の種類の証明書が適切であると指示される場合がある。認証実施規定と証明書ポリシーの根本的な違いは、前者は証明書発行機関が「所有」するのに対し、後者は発行された証明書を使用するエンティティが所有することである。証明書ユーザーが証明書ポリシーを定義し、(異なる認証実施規定を保有する) 証明機関がその証明書ポリシーに特定の証明書が適切であることを証明する。

Certification Practice Statement (CPS) / 認証実施規定(CPS) { Liberty }
証明機関が証明書の発行時に適用する実施規定を記述したもの。認証実施規定は、信頼性の高いシステムの詳細と、証明書の発行時に採用する実施規定を証明機関が宣言する形で提示される。

Certificate Revocation List (CRL) / 証明書失効リスト (CRL) { Liberty }
発行者によって有効期限切れ以前に取り消されたデジタル証明書を列挙したデータ。[RFC2828]

CES { Liberty }
Case Exact String の略。ある属性を自由形式の文字列から成るものとして定義するために使用する用語。Exact Match は、この属性の比較で大文字と小文字を区別する。「CIS」も参照。

Circle of Trust (CoT), Trust Circle / トラストサークル { Liberty }
ユーザーが安全でシームレスな環境で取引できるような、Liberty アーキテクチャおよび運用協定に基づくビジネス関係を持っているサービスプロバイダとアイデンティティプロバイダとの連携。

CIS { Liberty }	
Case Inexact String の略。ある属性を自由形式の文字列から成るものとして定義するために使用する用語。Inexact Match、この属性の比較で大文字と小文字を区別しない。「CES」も参照。	

Claim / クレーム { WSF }
エンティティによってなされる宣言(declaration) (例えば、名前、ID、鍵、グループ、権限、能力、属性など)

Cookie / クッキー { Liberty }
通常 Web を利用した個人のローカルコンピュータに保存されるもので、ユーザー名や現在の日時を含む情報の集合を指す。主として以前にサイトを訪問したり、登録したことがある利用者を Web サイトが識別するために利用される。

Credentials / クレデンシャル { SAML, Liberty }
要求されたプリンシパル識別を確証するために転送されるデータ。

D

Data / データ　{ Liberty }
主体者がアイデンティティプロバイダやサービスプロバイダに提示するあらゆる情報。

Defederate Identity / アイデンティティ連携の解除　{ Liberty }
主体者がアイデンティティプロバイダやサービスプロバイダに提示するあらゆる情報。

Delegation / 委任　{ Liberty }
あるシステムエンティティに、主体者に代わってアイデンティティサービスにアクセスさせること。

Digest / ダイジェスト　{ WSF }
あるオクテットストリームに対する暗号的なチェックサム

Digital Certificate / デジタル証明書　{ Liberty }
電子的に署名されたアサーション。基礎となるアサーションの発行元と同じ主体者が証明書に署名しなければならない。

Digital Signature / デジタル署名　{ Liberty, WSF }
秘密鍵と署名対象のメッセージ内容に大きく依存するデータ構造。デジタル署名は、対応する公開鍵でしか確認できない。注：多くの点で、デジタル署名は手書きの署名と同義ではない。注：国際的な法律では、デジタル署名の定義はさまざまである。「公開鍵暗号方式」も参照。

Direct Brokered Trust / 直接に仲介された信頼　{ WSF }
ある当事者が、第三者(third party)のクレームを信頼/保証する別の当事者(second party)を信頼すること。

Direct Trust / 直接的な信頼,ダイレクトトラスト　{ WSF }
信頼当事者(relying party)が要求者より送られたトークン中のクレームの全て、または、いくつかのサブセットを正しいものとして受け取ること。

Discovery Service / ディスカバリサービス　{ Liberty }
属性プロバイダを特定するための Liberty のサービス。

DNS (Domain Name System)　{ Liberty }
主にインターネットでホスト名をインターネットアドレスに変換するために使用される、分散型の複製された汎用のデータ照会サービス。

E

Electronic Commerce Modeling Language (ECML) { Liberty }
複数ベンダーから提供される電子財布のような自動化ソフトウェアが、統一規格で必要データを提供できるようにするための階層的な支払い用のデータ構造の一式。

End Point / エンドポイント { Liberty }
「エントリポイント」の口語。

End User / エンドユーザ { SAML }
アプリケーション目的のためにリソースを使用する自然人（システム管理目的と相対するものとして; 管理者、利用者を参照）。

Entity-Provided Data / エンティティから提供されるデータ { Liberty }
エンティティから Liberty トラストサークルのメンバーに直接提供されるあらゆるデータ。

Entry Point / エントリポイント { Liberty }
サービスを利用するために使用できる、SOAP（RPC）アドレスと機能名。Liberty エントリポイントには、ディスカバリサービスにおいて検出される対象を指す。

F

Federate / 連携させる { Liberty }
2つ以上のエンティティをリンクまたはバインドすること。

Federated Architecture (Authentication) / 連携アーキテクチャ(認証) { Liberty }
Liberty トラストサークル内の加盟者間で主体者にサービスを提供する複数エンティティをサポートするアーキテクチャ。

Federation / 連携 { Liberty, WSF }	
任意の数のサービスプロバイダおよびアイデンティティプロバイダで構成される提携関係。	

H

HTTP (Hypertext Transport Protocol)　{ Liberty }
分散協調型ハイパーメディア情報システム用のアプリケーションレベルのプロトコル。[RFC2616]

I

Identifier　{ SAML }
一意に参照することができるシステムエンティティにマップされた表記（例えば文字列など）。

Identity / アイデンティティ　{ Liberty }
エンティティの本質のことであり、しばしばその特性によって表される。

Identity Federation / アイデンティティ連携 { Liberty }、 Identity Mapping / アイデンティティマッピング { WSF }
・ トラストサークル内のさまざまな LibertyAlliance エンティティにおいて、主体者の複数アカウントを関連付けたり、バインドすること。 (Liberty より) ・ アイデンティティプロパティの間の関係を築く方法。アイデンティティプロバイダはアイデンティティマッピングを利用してもよい。(WS-F より)

Identity Provider (IdP/IP) / アイデンティティプロバイダ (IdP/IP)　{ Liberty, WSF }
・ 主体者のアイデンティティ情報を生成、保管、管理し、主体者の認証をトラストサークル内の他のサービスプロバイダに提供する Liberty 対応のエンティティ。 (Liberty より) ・ アイデンティティプロパティの間の関係を築く方法。アイデンティティプロバイダはアイデンティティマッピングを利用してもよい。(WS-F より)

Identity Service　{ Liberty }
1つまたは複数のアイデンティティに関する情報を収集、更新し、それらアイデンティティのために何らかの行為を行うよう特定のリソースに作用する、Web サービスの抽象的観念。

ID-PP　{ Liberty }
ID 個人プロファイル（ID-PP）とは、プライベートと職場の領域を問わず、主体者に関するアイデンティティ情報のこと。

Indirect Brokered Trust / 間接的に仲介された信頼　{ WSF }
別の当事者(second party)がある当事者（first party)に対する第三者(third party)のクレームを即座に検証できず、クレームを検証し、第三者の信頼を評価するために、第三者、あるいは、さらに別の当事者に対して交渉をおこなう場合の、direct brokered trust の一つの形態。

Initial SOAP Sender / 初期 SOAP 送信者　{ SAML }
SOAP メッセージパスの開始点において SOAP メッセージを発生させる SOAP 送信者。

Invocation Identity / 呼び出しアイデンティティ　{ Liberty }
メッセージ処理時にサービスをリクエストする側であり、SAML アサーションの主体

IP/STS　{ WSF }
アイデンティティプロバイダ(IP)あるいはセキュリティトークンサービス(STS)のいずれかのサービスのこと。

IPsec (Internet Protocol Security) { Liberty }
パブリックなネットワークにおけるデータ通信の機密性、完全性、および真正性を確保するためのフレームワーク(RFC2401)

K

Kerberos　{ Liberty }
信頼できる第三者機関による認証プロトコル(RFC1510)。

L

Liberty Alliance guidelines / Liberty Alliance ガイドライン　{ Liberty }
LibertyAlliance によって定義され、Liberty 仕様を最大限に実装するために準拠することを推奨されているポリシー。

Liberty Alliance principles / Liberty Alliance 原則　{ Liberty }
アイデンティティプロバイダまたはサービスプロバイダが Liberty 規格に準拠であるために契約上合意しなければならないという約束事項。

Liberty architecture / Liberty アーキテクチャ　{ Liberty } 連携したアイデンティティによるシングルサインオンが可能なプログラムや仕様をサポートするアーキテクチャ。

Liberty-enabled client (LEC) / Liberty 対応クライアント(LEC)　{ Liberty } 主体者がサービスプロバイダで利用することを希望するアイデンティティプロバイダに関する情報を保有するか、または情報の取得方法を知っているエンティティを指す。

Liberty-Enabled Client and Proxy Profile / Liberty 対応クライアントとプロキシプロファイル { Liberty } このプロファイルでは、Liberty 対応クライアントまたはプロキシ（場合によってはその両方）、サービスプロバイダ、アイデンティティプロバイダの間のやり取りを指定します。

Liberty-enabled client or proxy (LECP) / Liberty 対応クライアントまたはプロキシ (LECP) { Liberty } Liberty 対応クライアントとは、主体者がサービスプロバイダで利用を希望するアイデンティティプロバイダに関する情報を保有するか、または情報の取得方法を知っているクライアントを指す。Liberty 対応プロキシとは、Liberty 対応クライアントをエミュレートする HTTP プロキシ（一般的には WAP ゲートウエイ）を指す。

Liberty-enabled Provider / Liberty 対応プロバイダ　{ Liberty } 本書においてのみ適用される定義として、Liberty 対応プロバイダとは、主体者の個人識別情報（PII）を収集、転送、または受信する属性プロバイダ（AP）、ディスカバリサービス（DS）、サービスプロバイダ（SP）、アイデンティティプロバイダ（IdP）のいずれかである。

Liberty-enabled Proxy (LEP) / Liberty 対応プロキシ (LEP)　{ Liberty } Liberty 対応プロキシとは、Liberty 対応クライアントをエミュレートする HTTP プロキシ（典型的には、WAP ゲートウェイ）を指す。

Liberty-enabled User Agent or Device (LUAD) / Liberty 対応ユーザーエージェントまたはデバイス(LUAD)　{ Liberty } Liberty 仕様の 1 つまたは複数のプロファイルに対して特有のサポートを有するユーザーエージェントまたはデバイス。標準の Web ブラウザは、Liberty で定義する多くのシナリオに用いることができるが、Liberty プロトコル特有のサポートがされないため、LUAD とはならないことに留意する必要がある。特有機能のいかなる要求も、LUAD の定義のみに基づくシステムエンティティの存在を意図するものではない。むしろ LUAD は、特有機能を実装する Liberty 仕様で定義される1つまたは複数の Liberty システムエンティティの役割を実行してもよい。たとえば、LUAD-LECP は、LibertyLECP プロファイルをサポートするユーザーエージェントまたはデバイスであり、LUAD-DS であれば、LibertyID-WSF ディスカバリサービスを提供するユーザーエージェントまたはデバイスが定義されることになる。

Login, Logon, Sign-On / ログイン,ログオン,サインオン　{ SAML, Liberty }
認証オーソリティに対して利用者がクレデンシャルを提示することによって、単純セッションの確立、および付加的にリッチセッションの確立を行う過程。

Logout, Logoff, Sign-Off, Sign-Out / ログアウト,ログオフ,サインオフ,サインアウト { SAML, Liberty, WSF }
単純セッションもしくはリッチセッションを終了することを利用者が要望する過程。

M

Markup Language / マークアップ言語　{ SAML }
特定の目的における XML 文書の構造へ適用する XML 要素と XML 属性の集合。マークアップ言語は典型的に、XML スキーマと添付の文書の集合という方法によって定義される。例えば、セキュリティアサーションマークアップ言語 (SAML) は二つのスキーマと標準の SAML 仕様テキストによって定義される。

MEP　{ Liberty }
MEP（Message Exchange Pattern:メッセージ交換パターン）は、SOAP ノード間で行われるメッセージ交換のパターンを設定するテンプレートを指す。[SOAPv1.2]

Metadata / メタデータ　{ Liberty }
アプリケーションまたは環境内で管理される他のデータに関する情報または記録を提供する定義データ。

minimum maximum / ミニマム最大値　{ Liberty }
特定フィールドの最大値またはサイズとしてサポートされるべき最小値。たとえば、URL のミニマム最大値は 256 文字であり、このフィールドに対応するすべてのシステムでは、少なくとも 256 文字がサポートされなければならないが、それ以上の文字数をサポートすることができる。

N

Namespace / 名前空間　{ Liberty }
すべての名前が一意となるような名前の集合。

Network Identity / ネットワークアイデンティティ　{ Liberty }
主体者のすべての既存アカウントから構成される属性のグローバルセットの抽象概念。

nonce / ノンス　{ Liberty }
ノンスとは、同じ目的のために 1 度しか使用されない値を指す。ノンスには、タイムスタンプ、Web ページの訪問カウンタ、ファイルの不正な再生や複製を制限または防止するための特殊なマーカーなどがある。

Nonrepudiation / 否認防止　{ Liberty }
ある行為や情報に関わったことを主体者が法的に否認できないようにすること。

Non-Transitive Proxy Capability / 非推移的プロキシ機能　{ Liberty }
信頼できる機関のポリシーに従い、別のエンティティのために行使できる機能。この機能は、移転しない。

O

Opaque Handle / オペイクハンドル　{ Liberty }
特定のアイデンティティプロバイダとサービスプロバイダ間でしか意味をなさない文字列。

P

PAOS　{ Liberty }
SOAP のための逆方向の HTTP バインディング[SOAPv1.2]。通常の HTTP バインディングとの主な相違点は、SOAP 要求が HTTP 応答にバインドされ、またその逆も同様にバインドされるという点である。

Party / パーティ　{ Liberty }
非公式なものとして、アサーションを受け取る、もしくはリソースにアクセスするといった、ある過程もしくはコミュニケーションに関係する 1 つ以上のプリンシパル。

Passive Requestors / パッシブな要求者　{ WSF }
広範にサポートされている HTTP(例えば HTTP/1.1)で有効な HTTP ブラウザ

Password / パスワード　{ Liberty }
認証情報として用いられる機密データ値（通常は文字列）。(RFC2828)

Permission / 許可　{ Liberty }
ユーザーがアクセスできるデータや使用できるメニューオプションおよびコマンドについて、各ユーザーに許諾された権限を指す。

Personally Identifiable Information (PII) / 個人識別情報(PII)　{ Liberty }
ある人物を特定したり、その位置を捜し出したりするためのデータを指し、主に名前、住所、電話番号、電子メールアドレス、銀行口座、または社会保障番号のようなその他の固有の識別子から構成される。

PIN (personal identification number) / 個人識別番号(PIN)　{ Liberty }
基本的には、パスワードと同じもの。一般に、サイズと内容は一部の文字や数字に限定される。(RFC2828)

policy / ポリシー　{ Liberty }
論理的に定義された、実行可能かつテスト可能な行動（ビヘイビア）ルールのセット。

Policy Decision Point / ポリシー決定点、認可決定オーソリティ (PDP)　{ SAML }
それ自身もしくはそのような決定を要求する他のシステムエンティティのために、認可決定を行うシステムエンティティ。例えば、SAML PDP は認可決定要求を消費し、応答において認可決定アサーションを生成する。PDP は「認可決定オーソリティ」である。

Policy Enforcement Point / ポリシー実行点 (PEP)　{ SAML, Liberty }
認可決定を要求し、続いてそれを実施するシステムエンティティ。例えば、SAML PEP は認可決定要求を PDP へ送信し、応答で送られてくる認可決定アサーションを消費する。

Principal / 主体者　{ SAML, Liberty }
・ 主体者とは、連携アイデンティティを取得でき、決定を下すことができ、認証された行為を自身のために行うことができるようなエンティティを指す。主体者には、個人ユーザー、個人のグループ、企業、その他の法人、Liberty アーキテクチャの構成要素などがある。(Liberty より) ・ 識別が認証可能であるようなシステム。(SAML より)

Principal Identity / プリンシパル識別子　{ SAML }
プリンシパルを識別する表現。一般的には識別子である。

Privacy / プライバシー { Liberty }
主体の希望に応じた、ライフサイクル全体にわたる個人情報の適切な取り扱い。

Profile / プロファイル { SAML, Liberty, WSF }
・ あるアイデンティティで認証を受けるために必要な識別とデータ以外に、そのアイデンティティのために保管される広範な属性から構成されるデータ。少なくとも、このような属性のいくつか（住所、嗜好、カード番号など）は主体者によって提示される。(Liberty より) ・ フレームワークかプロトコルからアサーションを埋め込むもしくは抽出する方法を記述した規則の集合。各プロファイルは「SAML の○○プロファイル」という形式の名前を与えられる。(SAML より) ・ このモデルがどのように要求者の特定のクラス（例えば、パッシブやアクティブなど）に適用されるかを記述したドキュメントのこと。(WS-F より)

Proof-of-Possession / 所有証明 { WSF }
主張されたアイデンティティによってメッセージが送られたこと、かつ（あるいは）、メッセージが作成されたことを証明するための認証データ

Proof-of-Possession Token / 所有証明トークン { WSF }
送信する当事者が所有の証明をするために用いるデータを含んでいるセキュリティトークン。概して、排他的ではないが、所有証明の情報は送信者と受信者だけが知っている鍵によって暗号化される。

Proprietary Data / 専有データ { Liberty }
ある組織に固有の保護データのこと。

proxy / プロキシ { SAML, Liberty }
他者を代理することを認定されたエンティティ。

Proxy Server / プロキシサーバ { SAML, Liberty }
クライアントとサーバのコンピュータシステム間でプロトコルを中継するコンピュータプロセス。これは、クライアントに対してはサーバのように見え、サーバに対してはクライアントのように見えることによる。

Pseudonym / 仮名 { Liberty }
所定の信頼当事者に対して主体者を確認するために、信頼当事者間でのみ有効となるようにアイデンティティプロバイダまたはサービスプロバイダによって割り当てられる任意の名前。

Pseudonym Service / 仮名サービス { WSF }
信頼領域やフェデレーション内にある主体者の代理アイデンティティの情報を保持する Web サービス。ここでは、主体者という言葉は単なる人ではなく、任意のシステムの実体を指す。

Public-Key Cryptography / 公開鍵暗号方式 { Liberty }
2 種類の鍵を使用する暗号方式。1 つ目の鍵は必ず秘密に管理され、それと一意に対応する 2 つ目の鍵は公開される。1 つ目の鍵（秘密鍵）を使用して作成されたメッセージは、2 つ目の鍵（公開鍵）を使用して「確実な」方法で一意に確認できる。確認の確実性が非常に高いため、このようなメッセージはデジタル署名と呼ばれる。最後に、公開鍵を使用して作成されたメッセージは、対応する秘密鍵でしか復号化することができない。「Digital Signature」を参照。

Public-Key Infrastructure (PKI) / 公開鍵基盤(PKI) { Liberty }
主体者を含むコミュニティのために、非対称暗号アプリケーションにおいて証明書管理、アーカイブ管理、鍵管理、およびトークン管理の機能を果たす証明機関（また、オプションとして登録機関やその他の支援サーバおよびエージェント）のシステム。(RFC2828)

Pull / プル(引き出す) { SAML }
システムエンティティから情報を能動的に要求すること。

Push / プッシュ(押し出す) { SAML }
能動的に情報を要求しないにシステムエンティティに対して、情報を供給すること。

R

Realm or Domain / 領域.,ドメイン { WSF }
セキュリティ管理あるいは信頼の単一のユニット

Recipient / 受信者 { Liberty }
メッセージを受信し、メッセージの最終的な処理者となるエンティティ。

Relying Party / 信頼当事者,信頼者 { SAML, Liberty }
要求されたサービスを提供するかどうかの判断に際して要求メッセージと関連するアサーションを信用する、メッセージの受信者。

Repudiation / 否認 { SAML }
義務や責任を拒絶したり、放棄したりすること。

Requestor / 要求者 { SAML, Liberty }
処理を行うために受信者にメッセージを送信するエンティティ。通常、要求者はメッセージの作成者でもある。

Resource / リソース　{ SAML, Liberty, WSF }
情報システム内（例えばファイルの形式や、メモリ内の情報）に含まれるデータ。システムによって提供されるサービス。システム機器の品目（言いかえれば、ハードウェア、ファームウェア、ソフトウェア、ドキュメントといったシステムの構成要素）システムオペレーションおよび機器を収容する設備。

Resource Offering / リソースオファリング { Liberty }
リソースとサービスインスタンスの関連付け。

Resource Owner Interaction (ROI)　{ Liberty }
Resource Owner Interaction の略。Resource Owner Interaction サービスは、リソース所有者とのやり取りを公開する Liberty アイデンティティサービスを指す。これにより、クライアント（通常は WSP。WSP が ROI サービスに対しては WSC として機能する）はリソース所有者に同意、認可決定などの問い合わせを行うことができる。

Responder, SAML Responder / 応答者,SAML 応答者　{ SAML }
他のシステムエンティティ（要求者）からのサービス要求に応答するために、SAML プロトコルを利用するシステムエンティティ。この概念において、「サーバ」という用語は使われない。なぜなら、多くのシステムエンティティが、同時もしくは連続的にクライアントとサーバの両方として動作するためである。SAML のために SOAP 結合が使用されている場合、SAML 応答者はアーキテクチャ的に最終 SOAP 受信者とは明確に性質が異なる。

Rights Expression Languages (REL) / 権利記述言語 (REL)　{ SAML }
使用方法の指示を伝えるための機械言語。REL の使用によって情報プロバイダは、情報交換の前に意図したとおりに情報が使用されることを要求し、特定のトランザクション中にやり取りされる情報について承認された使用方法を指示することができる。

Role / 役割、ロール　{ SAML }
辞書では「キャラクタあるいは実行者によって演じられた部分」や「職務や地位」と定義している。主体者は、連続的および/または同時に様々な種類のロールを身につける。例えば、能動ロールおよび受動ロールである。管理者の概念は、多くの場合においてロールの例である。

Remote Procedure Call Protocol (RPC)　{ Liberty }
プログラマーが明示的に動作を記述しなくても、あるホストで動作しているプログラムに別のホストで実行されるコードを生成させることが可能なプロトコル。

S

SAML (Security Assertion Markup Language)　{ Liberty }
セキュリティシステム間で認証および認可されたデータを交換するための XML 規格。(Liberty より) XML でエンコードされたセキュリティアサーション、様々なプロトコルとフレームワークにアサーションを添付するためのプロファイル、アサーションを得るために用いられる要求/応答プロトコル、および様々な伝送プロトコル（例えば SOAP や HTTP）へのこのプロトコルの結合について記述した仕様の集合。(SAML より)

SAML Authority / SAML オーソリティ　{ SAML, Liberty }
SAML ドメインモデルにおける、アサーションを発行する抽象的なシステムエンティティ。属性オーソリティ、認可オーソリティ、ポリシー決定点 (PDP) も参考のこと。

Security / セキュリティ { SAML }
情報の機密性、その過程で使われるシステムやネットワークの保護、そしてそれらへのアクセスの制御を確保する安全装置のコレクション。セキュリティは一般的に、秘密、機密性、完全性および可用性の概念を包括する。それは、システムが関連する攻撃から潜在的に耐えることを保証することを意図される。

Security Architecture / セキュリティ構成 { SAML }
管理ドメインおよびそのセキュリティドメインにおける計画および規則の集合。それらは、そのユーザの必要を満たすために提供することがシステムに要求されるようなセキュリティサービス、サービスを実装するために要求されるシステム要素、要素において脅威の環境に対処するために要求される実行レベル、について記述する。システムにおける完全なセキュリティ設計は、システムにおける完全なセキュリティ構成は、管理上のセキュリティ、通信のセキュリティ、コンピュータのセキュリティ、放射セキュリティ、人的セキュリティ、そして物理セキュリティに取り組み、各々のためにセキュリティポリシーを規定する。完全なセキュリティ構成は、故意で利口な脅威と偶然な脅威の両者に対処する必要がある。セキュリティ構成は、その管理ドメインの発展における不可欠な部分として、時間とともに明確に発展していくべきである。

Security Assertion / セキュリティアサーション　{ SAML }
セキュリティ構成の内容を綿密に調べることの表明。

Security Domain / セキュリティドメイン　{ SAML }
セキュリティモデルおよびセキュリティ設計で定義される環境や状況。リソースの集合およびリソースへのアクセスを認可するシステムエンティティの集合が含まれる。単一の管理ドメインにおいて 1 つ以上のセキュリティドメインが属することがある。与えられたセキュリティドメインで定義される特性は、典型的に時間とともに発展する。

Security Policy / セキュリティポリシー　{ SAML }
システムや組織がリソースを保護するためのセキュリティサービスをどのように提供するかを明示もしくは規定するルールとプラクティスの集合。セキュリティポリシーはセキュリティ設計の構成要素である。セキュリティポリシーの重要な部分は、セキュリティサービスを通して実装され、セキュリティポリシー表現を利用する。

Security Policy Expression / セキュリティポリシー表現　{ SAML }
プリンシパル識別および／または許可されるアクションを備えた属性のマッピング。多くの場合、セキュリティポリシー表現とは本質的にアクセス制御リストのことである。

Security Service / セキュリティサービス　{ SAML }
リソースへ特定の種類の保護を与えることを目的とする、システムが提供する処理または通信のサービス。前述のリソースは、前述のシステムに属するか、他のシステムに属するかもしれない。ここでのシステムとは例えば、認証サービスもしくは PKI ベース文書の属性および認証サービスである。セキュリティサービスはAAAサービスのスーパーセットである。セキュリティサービスは典型的にはセキュリティポリシーの一部を実装する。そして、セキュリティサービスはセキュリティメカニズムを通じて実装される。

Security Token / セキュリティトークン　{ WSF }
エンティティによってなされる宣言(declaration)（名前、ID、鍵、属性など）を表すクレームの集合

Security Token Service (STS) / セキュリティトークンサービス (STS)　{ WSF }
セキュリティトークンを発行する Web サービス(WS-Security を参照のこと)。STS は、STS を信頼する誰に対してでも、STS が信頼する証拠に基づきアサーションを生成する。信頼を伝えるために、サービスはひとつのセキュリティトークンやセキュリティトークンの集合のような証明を必要とし、そして、自身の信頼ステートメントを備えたセキュリティトークンを発行する（いくつかのセキュリティトークンのフォーマットでは再発行や連署するだけであることに注意）。信頼仲介(trust brokering)の基礎となっている。

Sender / 送信者　{ Liberty }
最初の SOAP 送信者。送信者のアイデンティティが呼び出しアイデンティティと異なる場合、その送信者はプロキシである。

Sender Authentication / 送信者認証　{ WSF }
送信者認証は Web サービスメッセージ（および関連するデータ）の送信者を指し示す Web サービスのアクター/役割を越えて確認された認証の証拠である。認証された仲介者が存在する場合には、ひとつのメッセージに対して複数の送信者がありえることに注意。また、メッセージの発信元が認証された送信者と独立していたり、背後に隠れているかもしれないように、メッセージを最初に作り出したのが誰かを特定する方法はアプリケーション依存（そして、範囲外である）であることに注意。

Service / サービス　{ Liberty }
特定のサービスまたは情報を提供するために指定されたエントリポイントの集まり。

Service Instance / サービスインスタンス　{ Liberty }
特定タイプのアイデンティティサービスの物理的なインスタンス作成を指す。サービスインスタンスとは、別個のプロトコルエンドポイントにおける実行中 Web サービスのことである。

Service Provider (SP) / サービスプロバイダ (SP)　{ Liberty }
主体者にサービスや商品を提供するエンティティ。

Session / セッション　{ SAML }
(しばしば利用者を伴う) システムエンティティの間において永続する相互作用。ある相互作用の状態を、その相互作用が続くかぎり持続することに象徴される。

Signed Security Token / 署名付セキュリティトークン　{ WSF }
特定の認証機関によって主張され、暗号的に署名されたセキュリティトークン (例えば、X.509 証明書やケルベロスチケット)

Signature Validation / 署名検証　{ WSF }
受信したメッセージが送信されたものと同一であることを検証するプロセス。

Single Sign-On (SSO) / シングルサインオン (SSO)　{ Liberty, WSF }
アイデンティティプロバイダ A との間に存在する認証セッションの証明を使用して、アイデンティティプロバイダ B との新たな認証セッションを作成できること。

Site / サイト　{ SAML }
地理的もしくは DNS 名の意味における、管理ドメインのための非公式な用語。これは、特定の地理的もしくは幾何学的な管理ドメインの部分を参照するかもしれず、あるいは ASP サイトの場合のように、複数の管理ドメインを包括するかもしれない。

SmartCards / スマートカード　{ Liberty }
1 つまたは複数の IC チップを内蔵し、コンピュータの CPU、メモリ、および入出力インターフェースの機能を果たす、耐タンパ性を持つクレジットカード大のデバイス

SOAP (Simple Object Access Protocol)　{ Liberty }
Web で情報やリクエストを伝えるために用いられる XML エンベロープおよびデータエンコーディング技術。一般的には、Web サービスで使用されるプロトコルと考えられる。実際には、HTTP や FTP などのより低レベルの Web プロトコルに使用されるエンベロープのカプセル化形式のことであ

る。(SOAP)

SSL (Secure Sockets Layer Protocol)　{ Liberty }

コネクション型のエンドツーエンドの暗号を使用してクライアント（通常、Web ブラウザ）とサーバ間のトラフィックにデータ守秘性サービスとデータ完全性サービスを提供し、オプションでクライアントとサーバ間のピアエンティティ認証を行うことも可能なプロトコル。「Transport Layer Security」を参照。(RFC2828)

SSO Assertion, Single Sign-On Assertion / SSO アサーション,シングルサインオンアサーション　{ SAML }

自身の寿命を明示的に定義したものが埋め込まれた条件（コンディション）をもつアサーション。それはさらに、主体者の認証に関連する 1 つ以上のステートメントを含んでいる。属性のような主体者についての付加情報も、アサーションの中に含まれているかもしれない。

Subject / 主体者　{ SAML }

セキュリティドメインの環境におけるプリンシパル。SAML アサーションは主体者についての宣言 (declarations) を生成する。

System Entity / システムエンティティ　{ SAML }

コンピュータ/ネットワークシステムにおける能動的な要素。例えば、自動プロセス、プロセス･サブシステム･人の集合、機能的な個別の集合を組み込むような人のグループがある。

T

Time-Out / タイムアウト　{ SAML }

あるイベントが発生していない場合に、ある条件が真になる期間のこと。たとえば、指定された期間のあいだその状態が無活動であったために終了したセッションのことを「タイムアウト」という。

TLS (Transport Layer Security Protocol)　{ Liberty }

SSL プロトコルを発展させたもの。TLS プロトコルによって、インターネット経由の通信のプライバシーを確保することができる。このプロトコルを使用すると、盗聴、改ざん、またはメッセージ偽造を防止しながら、クライアントアプリケーションとサーバアプリケーション間で通信することができる。(RFC2246)

Trust / 信頼,トラスト　{ WSF }

ひとつのエンティティが、アクションの集合を実行するため、かつ（または）、主体および（または）範囲の集合に関するアサーションを作るために、別のエンティティに進んで依存する特性。

Trusted Authority / 信頼できるオーソリティ　{ Liberty }
Liberty においては、アサーションの発行および保証を担う、信頼できる第三者機関（TTP）を指す。

Trust Domain,Realm / トラストドメイン,信頼領域　{ WSF }
要求元(source)からのクレデンシャルの特定の集合が要求先(target)の適切なセキュリティポリシーを満たしているかどうかを要求元(source)と要求先(target)が決定/同意できるセキュリティ領域。要求先(target)は信頼の決定を、信頼領域内の信頼できる第三者を含めて、ある第三者（これが同意の一部として確立されているならば）に従ってもよい。

TTP　{ Liberty }
信頼できる第三者機関（Trustedthirdparty の略）。

U

Ultimate SOAP Receiver / 最終 SOAP 受信者　{ SAML }
SOAP メッセージの最終的なあて先である SOAP 受信者。それは、SOAP ボディ、およびそれをターゲットにした他の SOAP ヘッダブロックの内容を生成するための処理の責任がある。いくつかの状況では、SOAP メッセージは最終 SOAP 受信者に到達しないかもしれない。例えば、SOAP 仲介者における問題のためである。最終 SOAP 受信者は、同じ SOAP メッセージにおいて SOAP 仲介者にもなりえない。

Uniform Resource Identifier (URI) / URI　{ SAML }
抽象的もしくは物理的なリソースを識別するための簡潔な文字列。URI は Web におけるリソースのための普遍的な指定機構である。URL (Uniform Resource Locator) は、リソースの主要なアクセス機構（例えば、それらのネットワークの「位置」）に結び付けられたアドレススキーマを使用する URI のサブセットである。(SAML より) 抽象的または物理的リソースを識別するための短い文字列。RFC2396 には、絶対および相対形式を含む URI の一般構文と、それらの使用についてのガイドラインが定義されている。(Liberty より)

URI Reference / URI 参照　{ Liberty }
付加的な番号記号 (#) および断片識別子を持つことができる URI。断片識別子は、識別されたリソース内の特定の場所や領域を指定する。

URL (Uniform Resource Locator)　{ Liberty }
URI のサブセット。URL は、名前やその他の属性によってリソースを識別するのではなく、直接的なアクセスメカニズムの表現（たとえばネットワーク上の位置）を通じてリソースを識別する。(RFC2396)

URN (Uniform Resource Names)　{ Liberty }
場所に依存しない永続的なリソース識別子として用いられ、(URN のプロパティを共有する) 他の名前空間を容易に URN 空間に対応させるための名前。(RFC2141)

User / 利用者　{ SAML }
任意の目的のためにシステムとそのリソースを利用する自然人。

User Agent / ユーザーエージェント　{ Liberty }
ユーザーのために Web コンテンツを読み込んで表示するソフトウェア。

User Interface / ユーザインタフェース　{ Liberty }
ユーザーエージェントによって提供されるコントロール (メニュー、ボタン、プロンプトなど) やメカニズム (選択やフォーカスなど)。

V

Validation Service / 検証サービス　{ WSF }
WS-Trust のメカニズムを使用し、提供されたトークンの検証と信頼レベルの評価を行う Web サービス。

VPN (Virtual Private Network)　{ Liberty }
ネットワークを利用する各ユーザーのプライバシーと認証を確保しながら、インターネット経由で運用できるネットワーク。

W

WAP (Wireless Application Protocol)　{ Liberty }
無線デバイスを使用するモバイルユーザーが簡単に情報やサービスにアクセスし、これらを利用できるようにするオープンな国際仕様。

Web Service / Web サービス　{ Liberty }
インターネットプロトコルを使用し、プログラムに使用されるように設計されたサービスを提供するサービス。

Web Service Consumer (WSC) / Web サービスコンシューマ(WSC)　{ Liberty }
Web サービスを利用してデータにアクセスするエンティティ。

Web Service Provider (WSP) / Web サービスプロバイダ (WSP)　{ Liberty }
Web サービスを通じてデータを提供するエンティティ。

WML (Wireless Markup Language)　{ Liberty }
XML に基づくマークアップ言語であり、携帯電話やポケットベルなどの狭帯域デバイスでコンテンツやユーザーインターフェースを指定するのに用いられる。

WSDL (Web Services Description Language)　{ Liberty }
Web サービスのインターフェースを記述する一般的な技術。http://www.w3.org/TR/wsdl/を参照。

X

XML (eXtensible Markup Language)　{ Liberty }
情報や文書を Web で交換できるようにエンコードするための W3C 技術。[XML],[XMLCanon],[XMLDsig],[xmlenc-core],[Schema1],[Schema2]を参照。

XML addressing / XML アドレッシング　{ Liberty }
(XML コーディングを使用して、)別のサービスにあるデータを検索して参照する方法。

XML Attribute / XML 属性　{ SAML }
XML 要素の開始タグに埋め込まれている XML データ構造であり、名前と値をもつ。例えば、下記の斜体箇所は XML 属性の実例である。<Address AddressID="A12345">…</Address>属性も参照のこと。

XML Element / XML 要素　{ SAML }
XML 文書中のそのような構造の間にて階層的に整理され、開始タグおよび終了タグ、あるいは空白タグのいずれかで示される XML データ構造。例えば以下のようになる。

XML Namespace / XML 名前空間　{ SAML }
URI 参照によって識別される名前のコレクション。XML 文書の中で要素型および属性名として使用される。XML 名前空間はしばしば XML スキーマと関連付けられる。例えば、SAML は二つのスキーマを定義しており、それぞれが個別の XML 名前空間をもつ。

XML Schema / XML スキーマ　{ SAML }
XML 文書の集合において用いられるべきマークアップ言語の規則について記述するために、W3C によって開発されたフォーマット。小文字では、「scheme」や「XML schema」がこのフォーマットにおける個々のインスタンスである。例えば、SAML は 2 つのスキーマを定義しており、1 つはセキュリティアサーションを符号化する XML 文書における規則を含み、1 つは要求/応答のプロトコルメッセージを符号化する XML 文書における規則を含む。スキーマは XML 要素および XML 属性だけでなく、これらの構築に適用されるデータ型 (datatypes) も定義する。

Z

ZIC (Zero Install Client)　{ Liberty }
Liberty 専用の拡張機能を有しない、一般的に使用される HTTP ベースのユーザーエージェント。たとえば、標準的な Web ブラウザは ZIC である。

用語の表記について

本用語集は SAML、Liberty、WS-Federation で用いられる SSO に関係する用語をまとめたものである。用語の表記法は以下の通りである。

用語の英語表記 / 日本語表記 {SAML, Liberty, WSF のいずれの用語集から来たか}
用語の説明

参考文献

[SAMLCore]
Assertions and Protocol for the OASIS Security Assertion Markup Language(SAML) V1.1 Committee Specification, OASIS, 2 Sep 2003

[SAMLBind]
Bindings and Profiles for the OASIS Security Assertion Markup Language(SAML) V1.1 Committee Specification, OASIS, 2 Sep 2003

[SAMLXsd]
Assertion Schema,OASIS, 2 Sep 2003, http://www.oasis-open.org/committees/download.php/3408/oasis-sstc-saml-schema-assertion-1.1.xsd

[SAMLPXsd]
Protocol Schema,OASIS, 2 Sep 2003, http://www.oasis-open.org/committees/download.php/3407/oasis-sstc-saml-schema-protocol-1.1.xsd

[SAMLSec]
Security and Privacy Considerations for the OASIS Security Assertion Markup Language(SAML) V1.1,OASIS, 2 Sep 2003

[SAMLConform]
Conformance Program Specification for the OASIS Security Assertion Markup Language(SAML) V1.1,OASIS, 2 Sep 2003

[SAMLGloss]
Glossary for the OASIS Security Assertion Markup Language(SAML) V1.1,OASIS, 2

Sep 2003

[LibArch11]

Liberty アーキテクチャ概要 バージョン 1.1, Liberty Alliance Project, Jan 15, 2003, http://projectliberty.org/specs/archive/v1_1/draft-liberty-architecture-overview-v1.1-JP-final.pdf

[LibArch12]

ID-FF アーキテクチャ概要, Liberty Alliance Project, http://www.projectliberty.org /jp/resources/liberty-architecture-overview-v1.2-JP.pdf

[LibWSF]

Liberty アイデンティティ・Web サービスフレームワーク概要 Draft 1.0-07, Liberty Alliance, 12 Sep 2003, http://www.projectliberty.org/jp/resources/liberty-idwsf-overview-v1.0-07-JP-03.pdf

[LibBind]

Liberty Bindings and Profiles Specification Version 1.1, 15 Jan 2003, Liberty Alliance Project, http://www.projectliberty.org/specs/liberty-architecture-bindings-profiles-v1.1.pdf

[LibAuthn]

Liberty Authentication Context Specification Version 1.1, 15 Jan 2003, Liberty Alliance Project, http://www.projectliberty.org/specs/liberty-architecture-authentication-context-v1.1.pdf

[LibSchema]

Liberty Protocols and Schema Specification Version 1.1, Liberty Alliance Project, http://www.projectliberty.org/specs/liberty-architecture-protocols-schema-v1.1.pdf

[LibWsfComp]
Liberty Alliance & WS-Federation: A Comparative Overview, Liberty Alliance Project, Oct 2003, http://www.projectliberty.org/resources/whitepapers/wsfed-liberty-overview-10-13-03.pdf

[WST]
Web Services Trust Language (WS-Trust) Version 1.0, VeriSign, Dec 18 2002, http://www.verisign.co.jp/webservice/ws-trust.pdf

[WSF-WP]
Federation of Identities in a Web Services World, A Joint Whitepaper from IBM Corporation and Microsoft Corporation, Version 1.0, July 8, 2003, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-federation-strategy.asp

[eAuth]
e-Authentication, US-Gov, http://www.cio.gov/eauthentication/

[NISTeAuth]
DRAFT Recommendation for Electronic Authentication, NIST, Nov 2003,http://www.cio.gov/eauthentication/documents/NISTsp800-63.pdf

[医療 PKIGL]
平成 14 年度保健医療福祉情報セキュリティ推進事業 医療用公開鍵基盤ガイドライン(暫定版), 財団法人 医療情報システム開発センター

メンバーリスト

事務局

松山　博美　電子商取引推進協議会　主席研究員

主査

鈴木　優一　　エントラストジャパン株式会社

副主査

漆島　賢二　　セコム株式会社

坂上　勉　　三菱電機株式会社

島　成佳　　日本電気株式会社

下江　達二　　富士通株式会社

技術顧問

木村　吉博　　経済産業省

有識者

牟田　学　　行政書士

TF７メンバー

氏名	会社名
伊藤　康宏	エントラストジャパン株式会社
千葉　直子	日本電信電話株式会社
柿崎　竜人	NTT コムウェア株式会社
田口　慶二	日本ベリサイン株式会社
松岡　浩平	NTT コムウェア株式会社
佐藤　雅史	セコム株式会社
遠藤　由紀子	日本電気株式会社
冨高　政治	富士通株式会社
松永　和男	株式会社日立製作所
鬼沢　祥司	日本ユニシス株式会社
竹田　義聡	三菱電機株式会社
有定　耕平	大日本印刷株式会社
北折　昌司	株式会社東芝
菅野　健司	株式会社帝国データバンク

増井　久之＊	香川大学
荻原　利彦	NTT コミュニケーションズ株式会社
加賀谷　誠	NTT コミュニケーションズ株式会社
中泉　隆	ニフティ株式会社
小林　智恵子	株式会社東芝
糸岡　崇	富士電機株式会社
武井　光太郎	大日本印刷株式会社
鍛冶　俊彦＊	株式会社日本電子貿易サービス

（注）＊はオブザーバー

SWG1 メンバー

氏名	会社名
河田　悦夫	株式会社ＮＴＴドコモ
関野　公彦	株式会社ＮＴＴドコモ
本間　史夫	日本認証サービス株式会社
川那部　恭	ニフティ株式会社
中村　真二＊	株式会社メディアフュージョン
田辺　望＊	株式会社メディアフュージョン
道坂　修	株式会社ＮＴＴデータ

（注）＊はオブザーバー

以上順不同

（奥付）

平成１５年度　経済産業省　受託業務
ＥＣ技術基盤の相互運用性に関する調査研究事業
（取引相手先の属性認証技術等の調査）
SAML利用検討報告書
平成　１６年　３月　発行

発行所　　財団法人　日本情報処理開発協会
電子商取引推進センター
東京都港区芝公園３丁目５番８号
機械振興会館　３階
TEL：０３（３４３６）７５００
印刷所　　（所　名）
（住　所）
TEL：

この資料は再生紙を使用しています。