アイデンティティ関連要素技術の変遷とプライバシー

板倉　景子

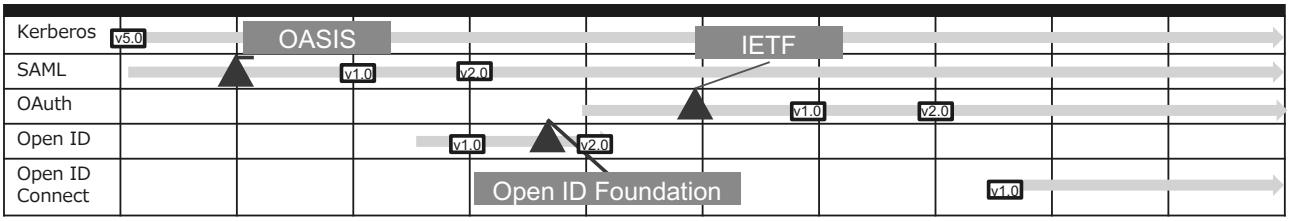
1. はじめに

あらゆるデバイスがインターネットに接続しインターネット上を流通するデータ量も増大しているなか、アイデンティティ管理の重要性も日増しに増大している。

アイデンティティ関連要素技術は多岐に及ぶが今回は前半部分で認証、認可の技術仕様を振り返りながら、後半部分ではもう少し抽象概念である「アイデンティティ」とそれにまつわるプライバシーについて記載する。

1.1 認証 / 認可プロトコル、標準仕様

まず簡単に認証 / 認可のプロトコル、標準仕様について代表的なものを整理する。

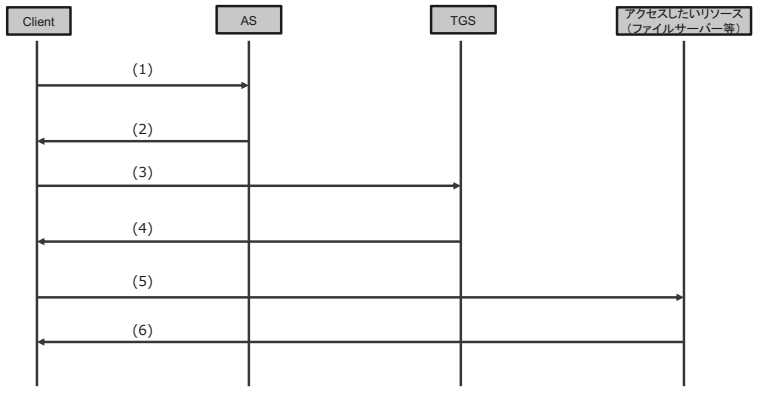


プロトコル/標準仕様年表

1.2 Kerberos

MITにより開発された認証プロトコルであり、1989年から利用されている。Windows Server Active Directory環境におけるユーザー認証として利用されることでも有名である。

ユーザーIDやパスワードといった認証情報はKDC(Key Distribution Center) と呼ばれる認証サーバで一元管理され、KDCが発行するチケットを用いて認証を行う。

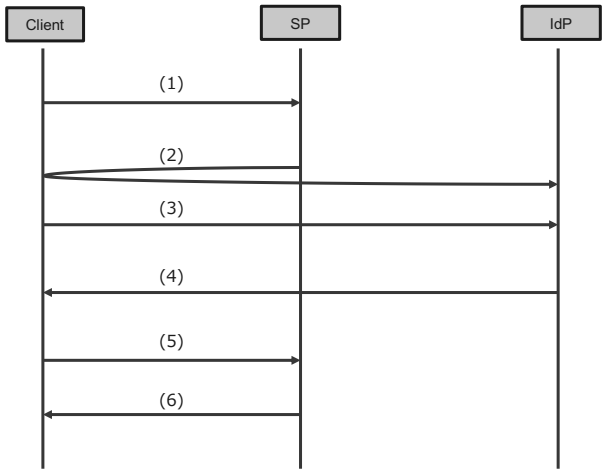


Kerberosシーケンス

1. クライアントは資格情報を入力しASにTGTを要求（Windowsドメイン環境の場合、ログイン時に自動的に実行）
2. 資格情報が正しければASはクライアントへTGTを発行
3. リソースへアクセスするためTGSにSTを要求
4. TGTを確認しTGSはクライアントへSTを発行
5. リソースへSTを提示
6. STを確認しアクセス許可

1.3 SAML

SAMLはSecurity Assertion Markup Languageの略で、OASIS3によって策定された、認証情報や属性情報をHTTPやSOAPなどで連携するためのXMLベースの標準仕様である。



SAMLシーケンス

1. クライアントはSPへアクセスする。
2. Idpへリダイレクトし認証要求を送付
3. クライアントは資格情報をIdPへ送付
4. IdPの認証サービスで認証しAssertionを生成し署名
5. クライアントはAssertionをSPへPOST
6. SPにてAssertion署名を検証し検証結果がOKであればサービスへのアクセス許可

1.4 OAuth

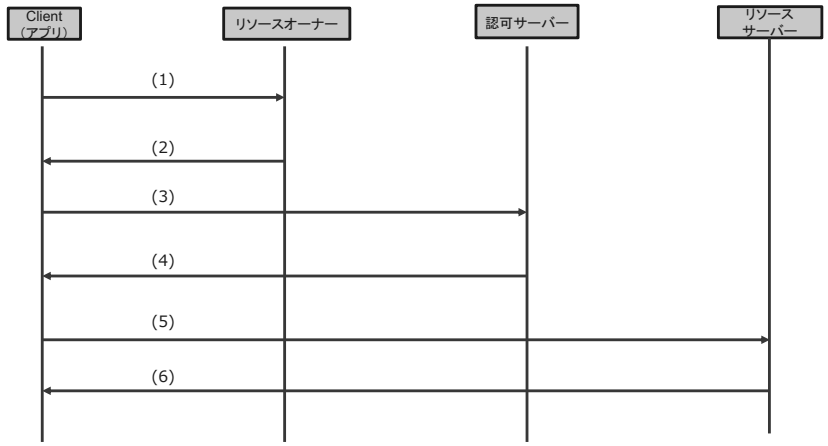
これまでの認証/認可のアーキテクチャは、複数のサービスが連携する場合はクレデンシャル情報をそのまま渡していたり、各社各様のAPIアクセス認可の仕組みを提供していたりした。(Flickr Auth, 2005 Google AuthSub, Yahoo! BBAuth, …)

しかし、マイクロサービスのような複数サービスが協調して動作するようなアーキテクチャではサービスごとに認証/認可を行う必要があるため、認証情報やアクセス制御ポリシーの管理が煩雑になる。

OAuthとは対象のリソースへのアクセスを許可させるためのフレームワークであり、認証結果や認可情報をAPIサーバ間で共有することで認証情報やアクセス制御ポリシーを一元化し管理が容易になるという利点がある。

OAuth2.0について解説されている「OAuth2 in Action」では、このトークンをValet Keyと喩えている。OAuth1.0は2007年12月にOAuth Core 1.0として公開され、2009年6月にセキュリティの改善がされOAuth Core 1.0 Revision Aとしてリリースされた。そして、2010年4月にはThe OAuth 1.0 Protocol (RFC 5849) として定義され、2012年にOAuth2.0のV2-31が出ている。

OAuthはSAML,Kerberos,WS-* と違って、クライアントアプリケーションがトークンの中身を分析しないことも特徴の一つといえる。ただし、「誰が」というのを表しているわけではないので、別の人がそのトークンを利用してしまえばアクセスができてしまう。

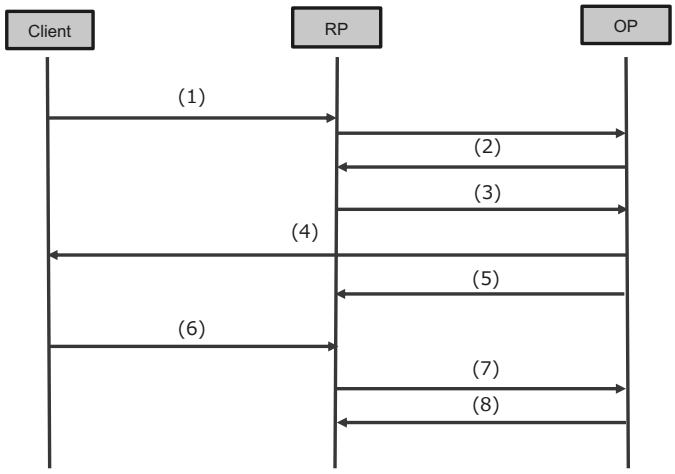


OAuthシーケンス

1. クライアントはResource Owner（ユーザー）へアクセスを許可するかどうか問い合わせ
2. Resource Ownerがアクセスを許可
3. クライアントは認可サーバーへ認可権限をリクエスト
4. 認可サーバーはクライアントへアクセストークンを発行
5. クライアントはアクセストークンをリソースサーバーへ提示
6. リソースサーバーはアクセストークンがデータの利用権限を持っていることを確認し、アクセス許可

1.5 Open ID Connect

Open ID ConnectはOAuth2.0の拡張仕様の一つであり、OAuth2.0を利用して認証やID連携をする際に必要な機能を標準化したものといえる。これを利用することによりWebアプリケーションやネイティブアプリケーション間でアイデンティティ情報を流通させる仕組みをより簡単に安全に実現できるようになった。



Open ID Connectシーケンス

1. RP登録用エンドポイントにPOSTリクエスト
2. 動的RP登録
3. RPはOPに認可リクエスト
4. OPはユーザーへアクセス権限付与の同意をリクエスト
5. ユーザーが同意したらOPは認可応答をRPへ返答
6. ユーザーはアクセストークン、IDトークンをRPへ提示
7. RPはアクセストークンを用いてUserInfoエンドポイントへにリクエスト
8. UserInfoエンドポイントはユーザーに紐づくアイデンティティ情報を返却

1.6 FIDO

なお、パスワードに変わる新たな認証標準としてのFIDOについても触れておきたい。FIDOとはFIDOアライアンスによって策定されている認証標準であり、パスワードの代わりにユーザーの公開鍵と署名を送付することで認証を行う仕組みである。なお、秘密鍵はユーザーのスマートフォンなどに保管されるためサーバー側に送付する必要が

ない。技術仕様が公開されているため、様々なメーカーが自社製品やサービスで利用を進めている。

1.7 社会的背景

かつて情報の活用が企業内に閉じていた時代から、それが企業間、インターネットを通じた第三者へと変化していくにつれその背景にある要素技術も変遷をたどっている。アイデンティティ関連要素技術の変遷は企業の境界を超えた情報の利活用の変遷とも言えるのではないか。

1.7.1 [昨今の動向]情報利活用によるサービス改善・新サービス提供への期待の高まり

総務省「平成28年 通信利用動向調査」によれば、インターネットを利用している個人の割合は83.5%であり、上昇傾向にある。

また、スマートフォンを保有する個人の割合も年々上昇しており、56.8%の個人が保有し、個人がいつでもどこでもインターネットにアクセスし様々なサービスを利用できる環境が整ってきている。

加えて、あらゆるデバイスがインターネットに接続するようになり、インターネット上を流通するデータ量は増大の一途を辿っている。

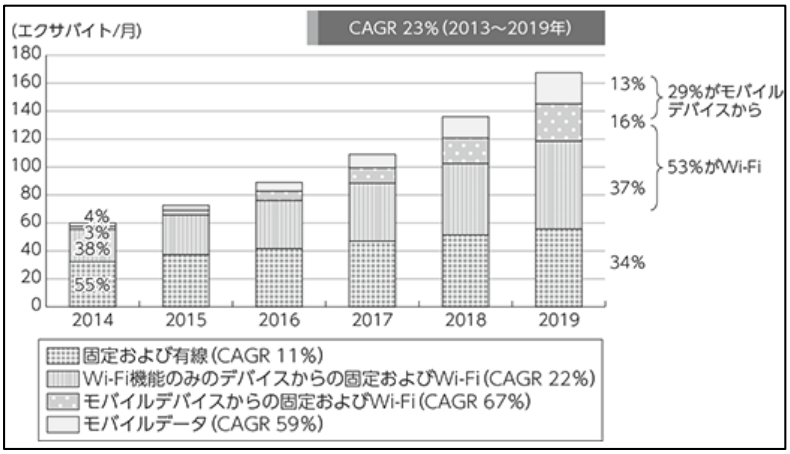


図1　データトラフィックの推移及び予測
出典：総務省「平成28年度　情報通信白書」

このような環境下において、インターネット上に流通する情報を利活用することによる既存のサービスの改善や、新たなサービスの提供へ期待が高まっている。

また、金融機関などの機密情報がwebや端末からのアクセスが年々増える傾向であり。イギリスの金融機関の統計を見ると、2007年から2018年の間で日常的にオンラインバンキングサービスを使用している個人の割合が40%程上昇しているという。このように色んなモノやサービスがインターネット環境で連携していく中でアイデンティティ管理の重要性が増加している。

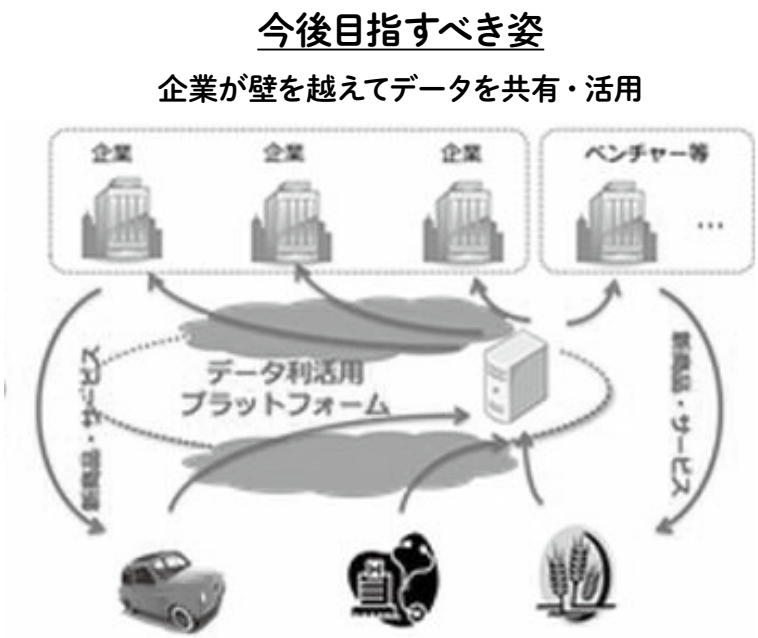


図 2　データを利活用した新たな付加価値の創出
出典：経済産業省「データ駆動型イノベーション」

1.8 本人確認手段の多様化

OAuthやOpen ID Connectといった認証 / 認可の技術はWebサービスにおける本人確認技術としても利用されている。昨今、Webサービスの利用においてはソーシャルログイン機能の普及が進んでおり、メルカリ、Airbnb、Uberといったサービスも Facebookアカウントでの個人情報登録及びログインが可能となっている。

事業者	会員数	提供サービス
Facebook	20 億	認証、ソーシャルグラフ
Twitter	1 億 7500 万	認証、ソーシャルグラフ
Google	1 億 7000 万	認証、アドレス、決済、Google Apps
Yahoo! Japan	2500 万	認証、アドレス、決済、ポイント

出典：Facebook社 Mark Zuckerbergによる投稿及びSocial Media Lab記事より筆者作成
https://www.facebook.com/zuck/posts/10103831654565331
https://gaiax-socialmedialab.jp/post-30833/

1.8.1 eKYC

「eKYC」とは、オンラインで完結する本人確認方法のことである。「電子的にあなたの顧客を知る」という意味の英語「electronic Know Your Customer」の略から来ている。

これまで銀行や保険会社、資金移動業者や仮想通貨交換業者などの業務の一部ではオンラインでの本人確認ができなかった。そのため、これらの業者とユーザーが一定の取引を行う際には、オフラインで本人確認が行われてきた。

これが、2018年11月の犯罪収益移転防止法施行規則の改正によって、オンラインで完結する本人確認方法、つまり「eKYC」の利用が一定の範囲（以下の2つの方法）で可能となった。

1. 「顔写真付きの身分証明書と自分の顔」の写った写真の送付、もしくは「自分の顔とICカード形式の身分証データ」の送付
2. 1と同様に「身分証明書」の画像もしくは「身分証データ」をオンラインで送付した上で、ユーザー名義の銀行口座開設やクレジットカード発行時の本人確認記録を他の事業者を通して確認することで本人確認

直近でも、メルペイは4月23日、LINE Payは4月24日、顔認証を利用したオンライン本人確認（e-KYC）を相次いで導入した。[2]

1.8.2 Self Sovereign Identity（SSI）

このような本人確認手段の変化の中で新たに出てきている概念がSelf Soverign Identity（自己主権型アイデンティティ）という概念である。これはユーザーが個人の情報を保持、コントロールし、サービスプロバイダにどこまで提供するかどうかを個人自身で決定するという考え方である。具体的にはEthereum開発者であるFabian VogelstellerがGitHubでSSIを実現するEthereum規格案を公開したりしている。

1.9 情報銀行

データの主権を個人に持たせるという点では、政府にて進めている施策として「情報銀行」やPDS（Personal Data Store）の取り組みも紹介しておきたい。

PDSとは個人が本人のデータを蓄積・管理し、他者と限定的に共有して活用することを可能にする仕組みであり、「情報銀行」とは集中PDSとして個人の情報を預託し、匿名化等の処理を施した上で事業者に提供するデータブローカである。

データ流通・利活用に関する国民の不安や不信感を払しょくするためにも、これらの取り組みを通じてデータ流通への個人の関与を強化していくことを進めている。

1.10 アイデンティティ関連要素にまつわるインシデントとプライバシー課題

インターネット上で様々なサービスが提供される中、そのサービスのいくつかではセキュリティ事故も発生している。2018年ではデータ侵害による被害の81%は二要素認証を使用しない脆弱なID・パスワードに対して行われており、[3]パスワードよりも強固な生体認証、行動分析、ハード・ソフトセキュリティトークンなどの重要性は増加している。

---

[2] https://jp.merpay.com/news/2019/04/ekyc_postpay/

[3] Verizon Data Breach Investigation Report
https://enterprise.verizon.com/resources/reports/DBIR_2018_Report_execsummary.pdf

企業でセキュリティ事故が発生し個人情報を悪用されることによりプライバシーが侵害されることの他に、企業がデータを利活用した結果、プライバシー侵害が生じる場合もある。
インターネット上に流通する情報量が増大し、かつその情報を処理する技術も進んだ結果として、これまでは特定の個人を識別できなかった情報が複数の情報と組み合わせることにより、特定の個人を識別することができるようになった。
企業はそれらの情報を利活用して、新たなサービスを提供したり、既存のサービスを改善しているが、消費者にとってはそれをプライバシーの侵害と感じ、問題となるケースがある。
法規制において、このプライバシー問題は現在時点でも曖昧性が残っており、事業者のセキュリティ対策が不十分であり、法規に違反しプライバシー侵害となる事例だけでなく、事業者が意図的に法制度の規範が曖昧な領域へ挑戦していることにより発生している事例もある。

本寄稿では深くふれることができなかったが、サービス提供者側の企業または個人がどのようなプライバシー意識を持っているか、どのようにサービス購入者との信頼関係を構築していくべきか議論の余地がある。
そもそも、プライバシーというのは画一的に語ることが非常に難しい概念であり、関係性の中で成立する相対的な概念といえる。現実をいかにリスクベースで評価できるかがプライバシー対策の重要なポイントであるだろう。

2. 終わりに

データを囲い込むのではなく活用するという流れの中で個人が自分自身のデータの主導権を握るべきという考え方が進んでいる。
自分を必要最低限にどうやってIdentifyしてもらうのかということを今後も考えていかなければいけないだろう。

参考文献

[1] サイバーセキュリティ.com「個人情報漏洩事件一覧」
https://cybersecurity-jp.com/leakage-of-personal-information（2017年7月18日参照）

[1] Justin Richer Antonio Sanso (2017) OAuth2 in Action

[2] https://w3c-ccg.github.io/did-spec/