セキュアなシステム間連携を支えるOSS認証認可技術

システム間連携に用いられるAPIの認証の標準としてはOAuth 2.0が一般的である。しかしOAuth 2.0は枠組みを定めるにすぎず，実際には既存の認証情報との連携のような作り込みや最新仕様への対応が求められ，適切なソフトウェア選定が重要となる。そのような中，認証分野で注目を集めるのがOSSである。
日立はOSSを活用したAPI管理ソリューションを展開し，既存の認証情報との連携パターン化による作り込みの容易化と，OSSコミュニティでの開発に参画し，最新仕様へのいち早い対応を実現している。

榎本 和史 | Enomoto Kazufumi
中村 雄一 | Nakamura Yuichi

1. はじめに

旧来の情報システムでは必要となるすべての機能を一つに集約したモノリシックな構成が主流であったが，デジタルトランスフォーメーション（Digital Transformation：DX）が推進されている昨今では，独立した複数のシステムをネットワーク経由で連携させて新たなサービスを提供するSystem of Systemsの構成が注目されている。このようなシステム間連携は一つの企業や組織の中で閉じるのではなく，それらを横断することも一般的になってきている。システム間連携のインタフェースとしては，連携のしやすさからREST API（Representational State Transfer Application Programming Interface：以下，本稿では単に「API」と記す。）という形式が主流であり，これらのAPIがインターネットを介して公開され，システム間連携が促進されている。一方で，APIがシステムへの攻撃の入り口となってしまうセキュリティ問題も生じている。実際に国内外でAPIの不正利用によるセキュリティ事故が相次いでいることから，APIを必要なシステムにのみ公開し，攻撃から保護するAPIセキュリティの機構が不可欠である。

本稿では，APIセキュリティにおけるOSS（Open Source Software）認証認可技術と日立の取り組みについて紹介する。

2. APIセキュリティ確保における課題

2.1 標準規格OAuth 2.0

API公開におけるセキュリティとして最も基本的なものは，APIを利用したアプリケーションやそのアプリ

ケーションを利用するユーザーを制限するための認証認可である。APIにおける認証認可の仕組みとしてはIETF（Internet Engineering Task Force）のRFC6749として定められた標準規格であるOAuth 2.0（以下，「OAuth」と記す。）が一般的である。

OAuth以前の外部アプリ連携とOAuthでのAPI連携の概要を**図1**に示す。OAuth以前は，外部アプリとサービス提供者が連携する場合，外部アプリにサービスのID・パスワードを渡し，サービスにログインさせることで連携することが一般的であった。一方でOAuthでは，ID・パスワードではなく，「アクセストークン」と呼ばれる認可情報を外部アプリに渡すことでセキュリティを確保する。流れとしては，最初に外部アプリは，「認可サーバ」にアクセストークンの発行を要求する。この際にユーザー認証が行われ，認証が成功すると認可サーバは外部アプリにアクセストークンを発行する。その後，外部アプリはアクセストークンを付加してAPIアクセスを行うという流れとなる。ここで，ID・パスワードのような認証情報は外部アプリに渡らず，より安全に連携できることから，外部アプリとの連携はOAuthによるAPI連携がデファクトスタンダードとなっている。

2.2 実システムへのOAuth適用における課題

OAuthはAPIの認証認可におけるフレームワークであり，その枠組みに沿った実装の仕方の大部分は実装者に委ねられている。このため，OAuthの仕様に沿っていたとしても，適切に実装されていない場合にはセキュリティ事故を招く危険性がある。

APIを利用した一般的なサービスの構成例と考慮すべきセキュリティ課題を**図2**に示す。

【課題1】OAuthの不適切な実装

OAuthのフローにおいてパラメータの使い方を誤って実装すると，攻撃者にアクセストークンが渡ってしまうような脆（ぜい）弱性が作り込まれることが知られている。また，脆弱性の隙をなくすために日々新たな仕様が策定されており追従も必要となってくる。例えば近年では金融機関の業務など，より高いセキュリティが求められるシステムでの利用を想定したFAPI（Financial-grade API）という上位互換の仕様が策定されており注目されている。

【課題2】不適切なユーザー認証

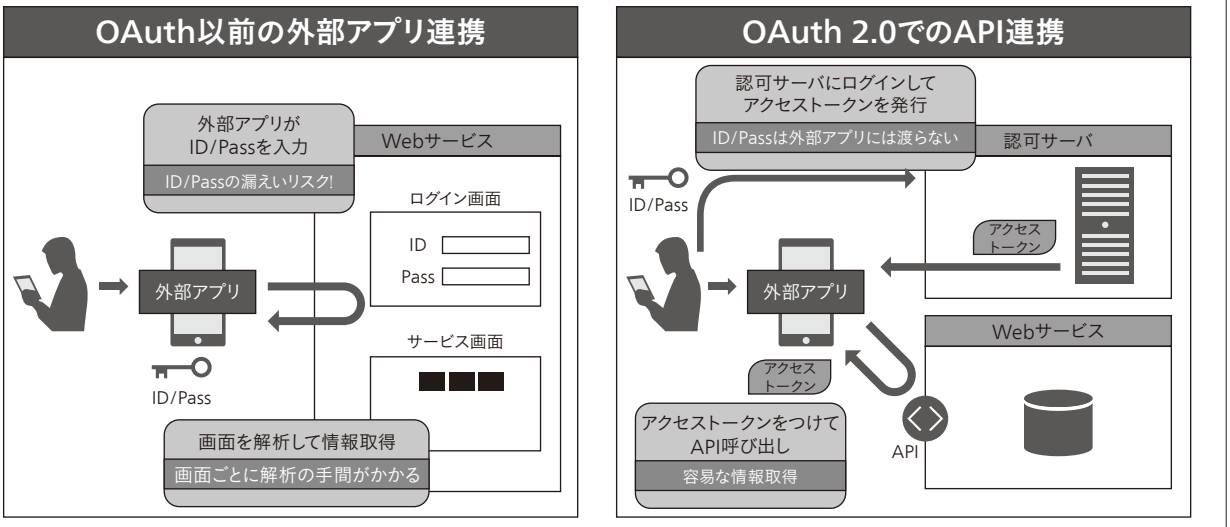
OAuthにおいて「認証」をどのように行うかは任意となっており，極端なケースでは簡単に推測可能な文字列のみで認証を済ませていることもある。このような場合，攻撃者に容易にアクセストークンを取得され，不正にAPI呼び出しが可能となってしまう。

【課題3】不適切なトークンの取り扱い

OAuthでは，認可サーバが発行したアクセストークンをどのように取り扱うかは定められておらず，問題になることがある。例えば，アクセストークンの失効管理が行われていないと，API連携の解除が永久にできないこ

図1｜OAuth以前の外部アプリ連携とOAuth 2.0でのAPI連携の概要

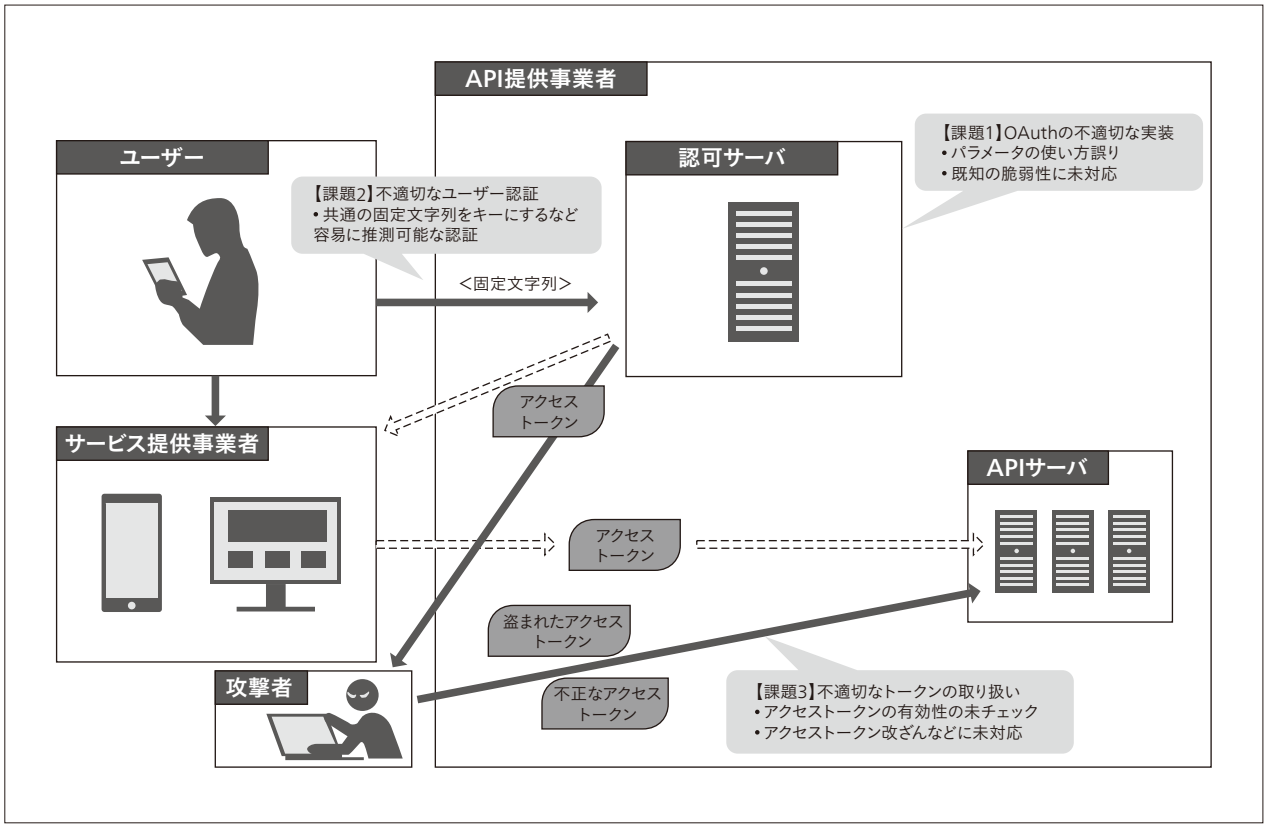
OAuth以前では連携する外部アプリが認証情報を保持するためセキュリティリスクが高かった。OAuth 2.0ではアクセストークンを用いることで，外部アプリに認証情報を渡す必要がなく，より安全に連携できる。



注：略語説明
API（Application Programming Interface）

図2｜OAuth 2.0における一般的なサービスの構成例と考慮すべきセキュリティ課題

OAuth 2.0はAPI認可の枠組みを定義するものであり，実装の多くは実装者に委ねられている。このため実際のシステム適用にはさまざまなセキュリティ上の課題があり，適切な実装が必要である。



とにつながる。また，APIサーバがアクセストークンを受け取った際，それが偽造されていたり，他者のアクセストークンだったりする可能性がある。アクセストークンの改ざん確認や誰に対して発行されたものかを確認する仕組みが別途必要になるが，これらのチェックを怠ると，攻撃者により本来権限のないAPIを呼び出されることにつながる。

実際にAPIを利用したサービスでは，上記のような考慮点が抜け落ちた実装によるセキュリティ事故が発生し続けており，セキュリティに対する目が厳しくなっているのが現状である。

3. 日立のAPI認証認可に対する取り組み

3.1 認可サーバ Keycloak

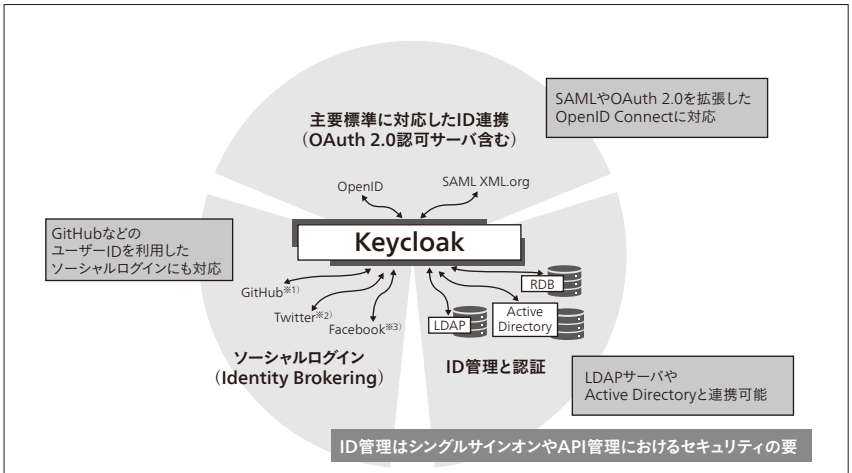
OAuthを利用した認証認可を実現するうえで特に重要となるものが，アクセストークンの発行と管理をつかさどる認可サーバである。近年この分野ではOSSである「Keycloak」が急速に存在感を増している（**図3**参照）。Keycloakは，もともとはSAML（Security Assertion Markup Language）などに対応したシングルサインオンのための認証サーバの実装であるが，OAuthに対応したAPIの認可サーバとしての機能も備えている。認証および認可サーバとしての機能を広くカバーしているだけでなく，多数の拡張ポイントが用意されており，個々のシステム要件に合わせたカスタマイズを行いやすい。

また，KeycloakはOSSであるため，開発コミュニティに誰でも参加できることが大きな特徴である。実際，開発コミュニティは活発であり，日々新たな標準に対応した機能開発が進んでいる。日立はKeycloakの開発コミュニティに参画し，主要機能の開発や不具合修正，利便性向上のための改修など広く貢献している。コミュニティ貢献で得た技術的に深い知見を基にして，Keycloakの商用版であるRed Hat, Inc.のRed Hat※) Single Sign-Onのサポートサービスや設計・構築支援サービスを提供している。さらには，実案件に適用することで生じた課題や機能追加ニーズを踏まえたパッチを開発コミュニティに投稿し，Keycloakに反映させ，Keycloakを日立の顧客ニーズに沿ったものにしていく活動を行っている（**図4**参照）。

※）Red Hatは，米国およびその他の国におけるRed Hat, Inc.の登録商標または商標である。

図3｜Keycloakの特徴

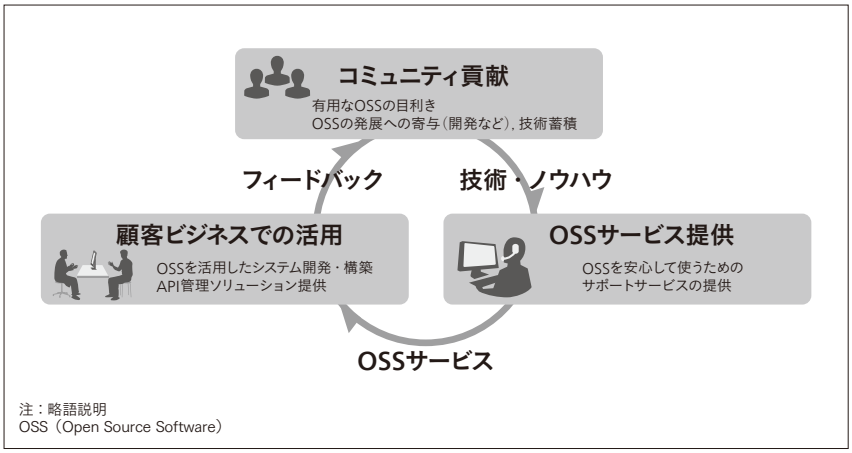
Keycloakはシングルサインオンの認証サーバやAPI連携における認可サーバとしての機能を持つ。OAuth 2.0などの標準仕様への対応やソーシャルログイン，LDAPサーバなどとの連携が特徴である。



注：略語説明など
SAML（Security Assertion Markup Language），XML（Extensible Markup Language），
LDAP（Lightweight Directory Access Protocol），RDB（Relational Database）
※1）GitHubは，GitHub, Inc.の登録商標または商標である。
※2）Twitterは，Twitter, Inc.の登録商標または商標である。
※3）Facebookは，Facebook, Inc.の登録商標または商標である。

図4｜日立のKeycloakコミュニティへの貢献とビジネス

日立はKeycloakのコミュニティへの貢献で得られた知見を基にサポートサービスの提供を行い，顧客へのビジネスでの活用を推進する。さらにコミュニティへフィードバックするサイクルによりKeycloakを発展させ，より価値あるサービスを提供していく。



注：略語説明
OSS（Open Source Software）

3.2 API管理ソリューション

日立はKeycloakを中心としたOSS認証認可技術をコアとして，セキュアなAPI連携を実現するAPI管理ソリューションを提供している。本ソリューションによるシステム構成を**図5**に示す。

(1) OAuthと最新標準への対応

認可サーバとしてKeycloakを採用し，OAuthに対応する。また最新標準のFAPIへの対応についても，日立が中心となってKeycloakコミュニティで開発を行っており，FAPIの基本的な機能要件を満たすことができる[1)]。

(2) 適切なユーザー認証の実現

ユーザー認証の強度を高めるためには，ID・パスワードによる認証に加え，多要素認証が必要であり，Keycloak標準のワンタイムパスワード認証にて多要素認証を実現することができる。また，最新の多要素認証の標準仕様であるWebAuthnについても日立が中心となってコミュニティで開発を行い，対応した[2)]。

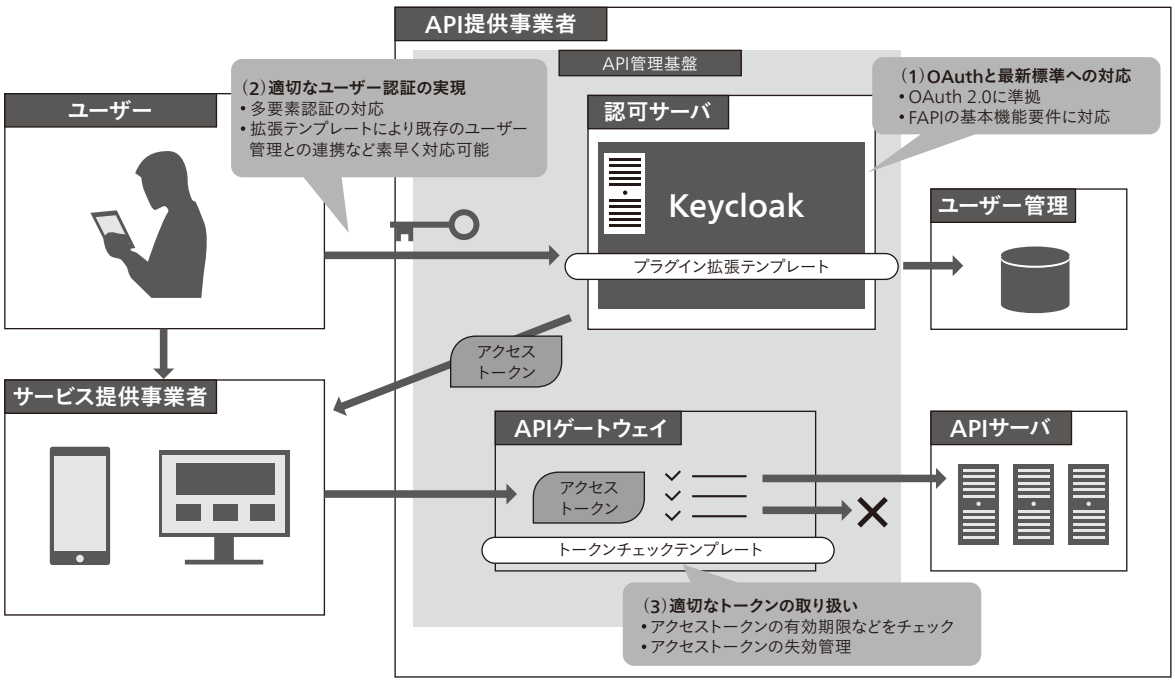
一方で，実案件においては，既存のシステムで管理される認証情報を利用して認証を行うニーズが高い。この場合，システム個別要件が入り込んでくるため標準機能だけでは実現できず，Keycloakの拡張を作り込む必要がある。日立では，典型的な拡張パターンをテンプレート化しており，テンプレートを適用することで迅速に既存の認証情報を利用することができる。

(3) 適切なトークンの取り扱い

2.2節の【課題3】で記したように，トークンの失効管理と，APIコール時のトークンの確認が必要になってくる。トークンの失効管理については，Keycloakの標準機能で対応できる。APIコール時のトークンの確認についてはAPIサーバ前に配備されたAPIゲートウェイにて実施する。APIゲートウェイとは，APIの流量制御やアクセス制御を行う要素であり，さまざまな種類のソフトウェア実装が存在する。このAPIゲートウェイにて，アクセストークンの改ざん検知や有効期限の確認を行うことにより，APIサーバに正しいアクセストークンが渡る

図5｜Keycloakを中心としたセキュアなAPI管理の構成例

Keycloakを認可サーバとしてセキュアなAPIを実現するシステム構成例を示す。セキュリティの課題をKeycloakとAPIゲートウェイにより解決している。



注：略語説明
FAPI（Financial-grade API）

ことを保証できる。ここでのAPIゲートウェイにおけるアクセストークンチェック処理については，APIゲートウェイ側で拡張が必要になることが多い。日立では，この拡張処理についても，主要なAPIゲートウェイ向けにテンプレート化している。

日立ではセキュアなAPI連携を実現するために上記の構成を基本とし，上流コンサルティングから，構築，運用保守までトータルにカバーし，ソリューションとして提供している。これまで，金融分野のキャッシュレスサービスや鉄道交通分野のMaaS（Mobility as a Service）などさまざまなデジタルサービス案件に本ソリューションを適用し，APIセキュリティを確保したAPI管理基盤を構築した実績がある。

4. おわりに

本稿では，APIセキュリティにおけるOSS認証認可技術と日立の取り組みについて紹介した。DXの推進によりAPIによるシステム間の連携が加速し，APIセキュリティの重要性が増している。最先端の規格対応の開発などのOSSコミュニティへの貢献を通じて得られた技術やノウハウを日立の強みとして，それらを生かしたAPIセキュリティを確保したシステムをより容易に素早く構築するためのソリューションの提供・拡大を行っていく所存である。

参考文献など

1) T. Norimatsu: What are protected and secured by security requirements for APIs providing financial services, APIdays Paris 2019 (2019.12), https://www.slideshare.net/ssuserbeb7c0/apidays-paris-2019-financialgrade-api-securityprofile
2) T. Norimatsu: WebAuthn support for keycloak, DevConf.cz 2020 (2020.1), https://static.sched.com/hosted_files/devconfcz2020a/78/webauthn-support-keycloak.pdf

執筆者紹介

榎本 和史
日立製作所 サービス&プラットフォームビジネスユニット
SoftwareCoE OSSソリューションセンタ 所属
現在，OSSを活用したAPI管理ソリューションの開発に従事

中村 雄一
日立製作所 サービス&プラットフォームビジネスユニット
SoftwareCoE OSSソリューションセンタ 所属
現在，OSSを活用したAPI管理ソリューションの開発やコンサルティング業務に従事
博士（工学）
情報処理学会会員