IMES DISCUSSION PAPER SERIES

オープンAPIの認可処理におけるセキュリティ：FAPIと海外のセキュリティ要件集の比較

宇根正志（うねまさし）

Discussion Paper No. 2026-J-4

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

IMES Discussion Paper Series 2026-J-4
2026 年 4 月

オープンAPIの認可処理におけるセキュリティ：FAPIと海外のセキュリティ要件集の比較

宇根正志（うね まさし）*

要　　旨

金融分野では、フィンテック事業者が金融機関のオープン API を用いて顧客の口座の取引情報を取得して顧客に提供する参照系サービスや、決済の実行を金融機関に指示する更新系サービスを提供している。こうしたサービスを安全に実現するうえで、金融機関がフィンテック事業者に顧客の情報へのアクセスを許可するための認可処理のセキュリティ要件を適切に設定することが必要である。こうした要件を整理したセキュリティ要件集の標準として、2025 年 2 月に FAPI 2.0 が公表された。今後、金融機関やフィンテック事業者が自社のサービスのセキュリティ要件を検討する際には、FAPI 2.0 を参照することが有用である。また、FAPI 2.0 の標準化完了より前に公表されている海外のセキュリティ要件集をみると、FAPI 2.0 と整合的なセキュリティ要件が既に含まれているだけでなく、FAPI 2.0 よりもリスク低減に資する要件も含まれている。海外のオープン API のサービスと遜色ないセキュリティを確保する観点から、金融機関やフィンテック事業者は、FAPI 2.0 をベースラインとして参照しつつ、海外のセキュリティ要件集に含まれている、リスク低減に資する要件の採用についても検討することが重要であろう。

キーワード：オープン API、更新系サービス、参照系サービス、セキュリティ要件集、認可処理、FAPI 2.0
JEL classification: G21、O33

* 日本銀行金融研究所参事役（E-mail: masashi.une@boj.or.jp）

本稿は 2026 年 2 月 4 日時点の情報に基づいて作成された。本稿の作成に当たっては、乗松隆志氏（株式会社日立製作所）から有益なコメントを頂戴した。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りはすべて筆者個人に属する。

目　　次

１．はじめに

オープン API（application programming interface）は、API の提供者が外部の組織に使用させるために公開した API を指す。金融サービスでは、金融機関が API の提供者となり、電子決済等代行業者などのフィンテック事業者が API を用いて顧客の口座の取引情報を取得して顧客に提供する参照系サービスや、決済の実行を金融機関に指示する更新系サービスがよく知られている。

こうしたサービスでは、フィンテック事業者が顧客に代わって複雑な処理を自動的に実行し、顧客の利便性向上に加え、金融機関から得たデータを分析して顧客に有益な情報（余裕資金の有効活用の手段など）を提供することができる。セキュリティの観点では、フィンテック事業者が顧客のパスワードを用いて各金融機関にアクセスする（ウェブ・スクレイピング）などの対応が不要となり、顧客の個人情報の保護やフィンテック事業者による不正行為の防止などのメリットを期待することができる。

こうしたサービスのセキュリティ向上を実現するためには、金融機関がフィンテック事業者に顧客の情報へのアクセスを許可するための処理（認可処理）において、第三者による攻撃（フィンテック事業者へのなりすまし、通信データの改変など）を想定してリスク評価を行ったうえで、リスク低減のためのセキュリティ要件を設定し、それを実現するように処理フローを実装する必要がある。ただし、各金融機関がそれぞれ異なるセキュリティ要件を設定した場合、フィンテック事業者の側からみると、アクセスする先の金融機関によって（同一内容のサービスであったとしても）セキュリティのレベルが異なり、不適切なセキュリティ要件を設定していた金融機関との処理において許容できないリスクが生じてしまうおそれがある。したがって、各金融機関が同一のセキュリティ要件を適切に設定することが望ましい。

オープン API を活用したサービスにおいて一定のセキュリティを確保する方法として、セキュリティ要件を定めたドキュメント（セキュリティ要件集）を策定し、それに準拠することを金融機関やフィンテック事業者に求めるというアプローチがある（Competition and Market Authority 2021）。イギリスのオープン・バンキング（Open Banking）では、認可処理に関するプロトコルの標準仕様である OAuth 2.0[1]のセキュリティ要件集で、OpenID Foundation[2]によって 2021 年 3 月

---

[1] OAuth 2.0 のフレームワークは RFC 6749 として標準化されている（Hardt 2012）。RFC（Request for Comments）は主にインターネット上での通信向けの標準仕様であり、個々の標準仕様は RFC に続く番号によって識別されている。RFC の標準化は、インターネット技術の調査・研究、技術仕様の策定などを行っている非営利団体 IETF（Internet Engineer Task Force）によって実施されている。OAuth 2.0 におけるエンティティや認可処理のフローは補論 1 を参照されたい。

[2] OpenID Foundation は、各個人が自らの識別情報を適切に使用・管理する環境の実現を目指し

に公表された FAPI 1.0（Financial-grade API Security Profile 1.0、Sakimura, Bradley, and Jay 2021a, b）に準拠することが必須とされている。また、イギリスのオープン・バンキングと並んで比較的早期にオープン API の活用に着手したブラジルのオープン・ファイナンス（Open Finance）やオーストラリアのコンシューマー・データ・ライト（Consumer Data Right）では、FAPI 1.0 をベースとしつつ検討が行われ、独自のセキュリティ要件集が策定されている。

FAPI 1.0 はセキュリティの向上などを企図して改訂され、2025 年 2 月、FAPI 2.0（FAPI 2.0 Security Profile、Fett, Tonge, and Heenan 2025）が標準化された。例えば、①認可の内容に関する情報の保護を強化する要件、②使用する暗号アルゴリズムの強度を向上させる要件、③金融機関やフィンテック事業者による顧客への情報提供を強化する要件などが追加・修正されている。FAPI 2.0 は最新の脅威への対応を考慮して策定されており、今後、セキュリティ要件集のベースラインとして活用されると見込まれる。

日本では、オープン API を活用した金融サービスにおける認可処理として OAuth 2.0 に基づく認可処理の採用が推奨されている（一般社団法人全国銀行協会［2017］）。セキュリティ要件集については、本稿執筆時点において筆者が知る限り、金融分野の業界団体などによって公表されているものはなく、金融機関やフィンテック事業者が個別にセキュリティ要件を検討・設定しているのが実情とみられる[3]。最近では、FAPI 2.0 に準拠したセキュリティ要件に基づくシステムの開発を検討している金融機関もあり（株式会社みんなの銀行［2024］）、今後、日本においても FAPI 2.0 への関心が一段と高まるとみられる。

海外のオープン API のサービスと遜色ないセキュリティを実現するという観点[4]からは、本邦金融機関やフィンテック事業者は、FAPI 2.0 をベースラインとして参照することに加えて、実際に運用されている海外のオープン API のセキュリティ要件集も視野に入れておくことも重要である。ブラジルのオープン・ファイナンスとオーストラリアのコンシューマー・データ・ライトにおけるセキュリティ要件集をみると、FAPI 2.0 と整合的な要件が既に含まれているだけでなく、FAPI 2.0 よりリスク低減に資する要件も含まれている。サイバーセキュリティの世界では、セキュリティのレベルが低い先が主な攻撃対象となる傾向がある。こ

て、認可・認証などの関連技術の標準仕様の策定などを行っている非営利団体である。

3 金融機関がフィンテック事業者と接続する際のセキュリティ上の留意点をまとめたチェックリストが金融情報システムセンターによって発表されている（公益財団法人金融情報システムセンター［2026］）。ただし、同チェックリストでは、認可処理などにおいて達成すべきセキュリティ目標が記載されているものの、技術仕様に関するセキュリティ要件は記載されていない。

4 仮に、自社のオープン API におけるセキュリティが海外のものより劣っていた場合、自社のサービスが攻撃者の標的になる可能性が高まると考えられる。したがって、海外と少なくとも同等のセキュリティを実現することがセキュリティ対策上望ましい。

の点を踏まえると、金融機関やフィンテック事業者は、自社のオープン API のセキュリティ要件を検討する際に、海外のセキュリティ要件集についても考慮することが有用であろう。

2 節ではオープン API を用いた金融サービスの処理例を紹介し、3 節では FAPI 2.0 の主なセキュリティ要件を説明する。4 節では、海外の主なオープン API の活用事例として、イギリス、ブラジル、オーストラリア、欧州連合における技術仕様やセキュリティ要件集を説明するとともに、FAPI 2.0 と比較する。

２．オープン API を用いた金融サービスの処理例

本節では、オープン API を用いた金融サービスの処理の例として、イギリスのオープン・バンキングにおける口座情報の取得や決済の実行に関する技術仕様 Read-Write API Profile（Open Banking Limited 2024a、2024 年 6 月公開）を参照して説明する。

（１）エンティティ

主なエンティティは以下のとおりである。

- 決済サービスの顧客（PSU：payment service user）
- 決済用の口座を管理しつつ決済サービスを提供する金融機関（ASPSP：account servicing payment service provider）
- 決済サービスの顧客と金融機関を仲介しつつ、決済に関連するサービスを提供するフィンテック事業者（TPP：third party provider）である。

フィンテック事業者の種類として、決済サービスの顧客に口座の取引履歴などの情報（口座情報）を提供する場合（AISP : account information service provider）や、決済の処理を金融機関に依頼する場合（PISP : payment initiation service provider）が想定されている。AISP と PISP は、それぞれ参照系サービス、更新系サービスを提供するフィンテック事業者に対応する。

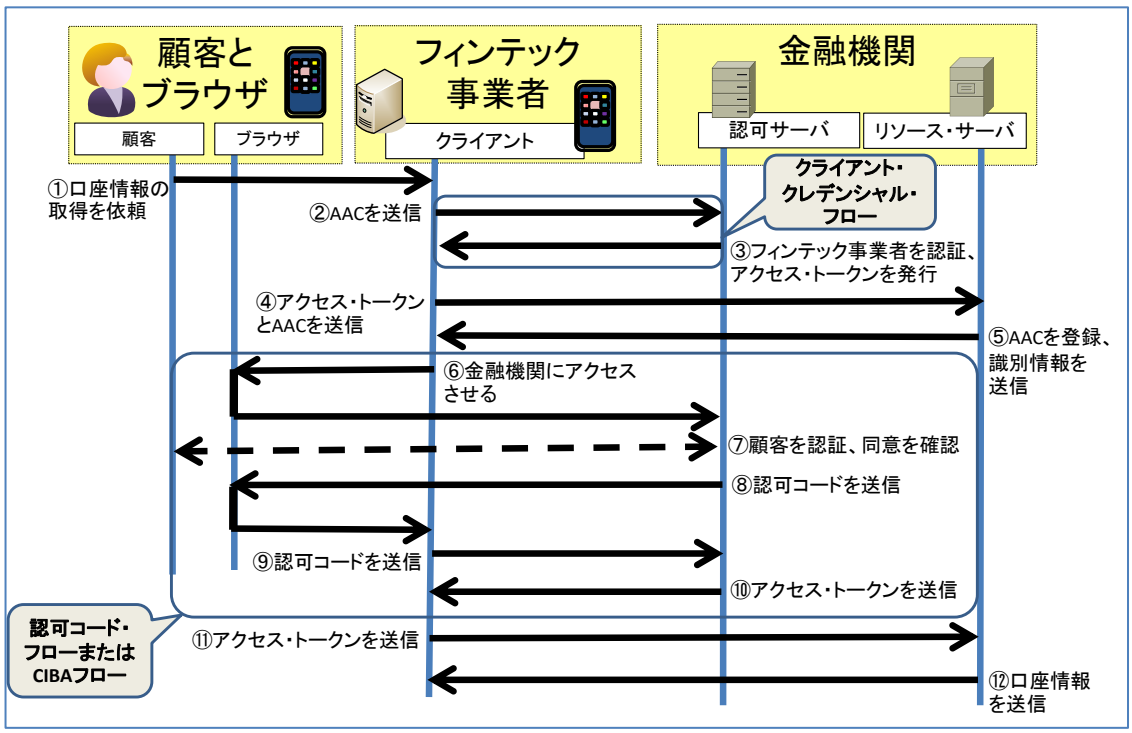
（２）口座情報の取得の処理フロー

フィンテック事業者が金融機関から口座情報を取得する際の処理フローは、フィンテック事業者による処理内容の登録、金融機関による顧客の同意の確認、口座情報の提供という流れとなる（Open Banking Limited 2024b、図 1 参照）。

【フィンテック事業者による処理内容の登録】

① 顧客（PSU）はフィンテック事業者に口座情報の取得を依頼する。

図 1　オープン API によって口座情報を取得する処理例



② フィンテック事業者は処理の内容を特定するデータ AAC（account-access-consent resource）[5]を金融機関（認可サーバ）に送信する。
③ 金融機関（認可サーバ）はフィンテック事業者を認証しアクセス・トークンを発行する。
④ フィンテック事業者はアクセス・トークンと AAC を金融機関（リソース・サーバ）に送信する。
⑤ 金融機関は AAC を登録し、その識別情報をフィンテック事業者に返信する。

【金融機関による顧客の同意の確認】

⑥ フィンテック事業者は顧客のブラウザを金融機関（認可サーバ）にアクセスさせる（認可リクエスト）。
⑦ 金融機関は、顧客を認証し、AAC の内容を同意していることを顧客に確認する（同意認可〈consent authorization〉）。
⑧ 金融機関は、顧客のブラウザを経由してフィンテック事業者に認可コードを送信する（認可リスポンス）。
⑨ フィンテック事業者は認可コードを金融機関（認可サーバ）に送信し、アク

[5] AAC は、処理の内容、AAC の有効期限、取得が許可される情報の範囲（当該事象の発生日時で特定）に関する情報などから構成される。

セス・トークンを要求する（トークン・リクエスト）。
⑩ 金融機関はフィンテック事業者にアクセス・トークンを送信する（トークン・レスポンス）。

【口座情報の提供】
⑪ フィンテック事業者は金融機関（リソース・サーバ）にアクセス・トークンを送信する（リソース・リクエスト）。
⑫ 金融機関は AAC で指定された口座情報をフィンテック事業者に送信する。

上記のうち、②と③の処理は OAuth 2.0 で規定されているクライアント・クレデンシャル・フロー[6]に基づいて実行される。⑥から⑩の処理は、認可コード・フローまたは CIBA フローに基づいて実行される[7]。

（３）決済実行時の処理フロー

顧客がフィンテック事業者を介して金融機関に決済の実行を依頼する際には、フィンテック事業者による処理内容の登録、金融機関による顧客の同意の確認、決済の実行という流れとなる（Open Banking Limited 2024c、図 2 参照）。

【フィンテック事業者による処理内容の登録】
① 顧客はフィンテック事業者に決済の実行を依頼する。
② フィンテック事業者は決済の処理の内容を示すデータ POC（payment-order consents resource）を金融機関（認可サーバ）に送信する。
③ 金融機関はフィンテック事業者を認証しアクセス・トークンを送信する。
④ フィンテック事業者は金融機関（リソース・サーバ）にアクセス・トークンと POC を送信する。
⑤ 金融機関は POC を登録し、その識別情報をフィンテック事業者に送信する。

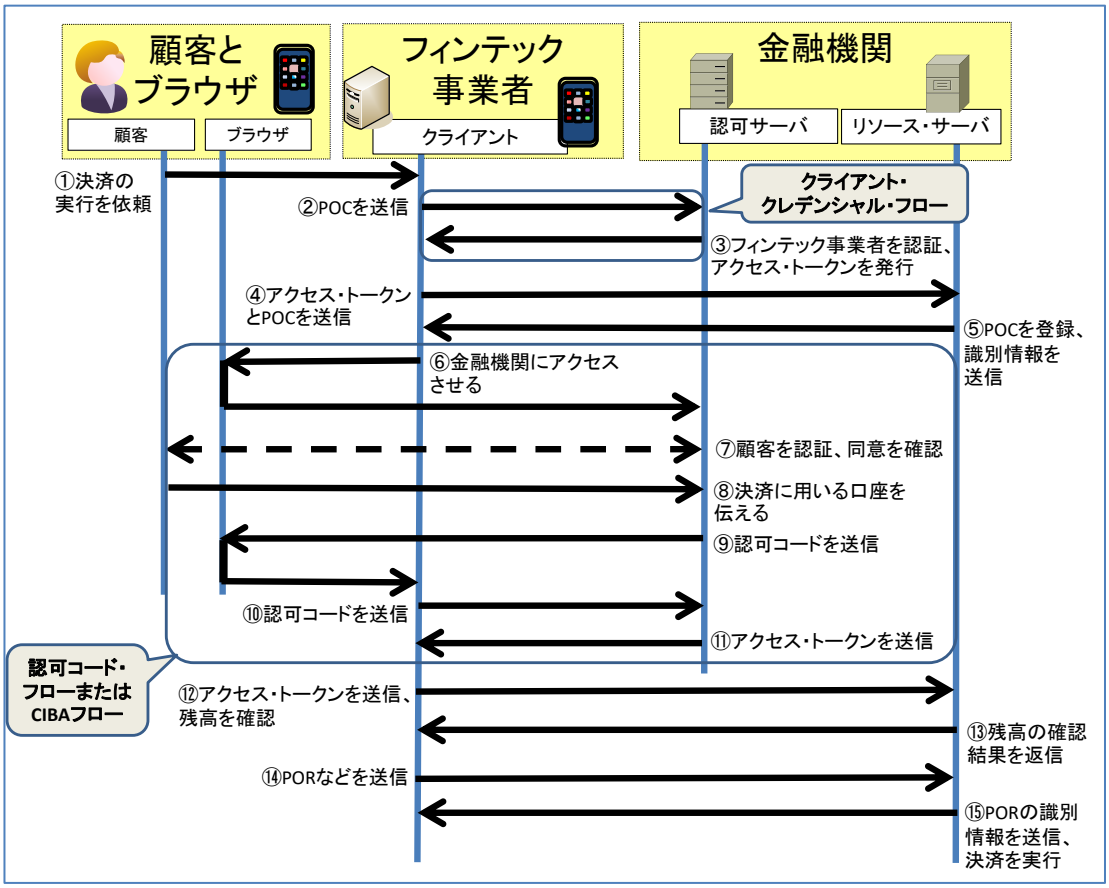
【金融機関による顧客の同意の確認】
⑥ フィンテック事業者は顧客のブラウザを金融機関（認可サーバ）にアクセスさせる（認可リクエスト）。
⑦ 金融機関は顧客を認証し、POC の内容に同意していることを顧客に確認す

---

[6] クライアント・クレデンシャル・フローは、クライアントが認可コードの代わりに自身のクレデンシャルを認可サーバに送信してアクセス・トークンを得る方式である。

[7] CIBA フロー（Client Initiated Back-Channel Authorization Frow）は、顧客が金融機関からの認証要求に（フィンテック事業者とやり取りするデバイスとは）別のデバイスによる通信路（back channel）で対応する認可フローの標準仕様である（Tonge *et al.* 2019）。

図 2 オープン API における決済の実行の処理例



る（同意認可）。

⑧ 顧客は決済に用いる口座を金融機関に伝える。

⑨ 金融機関は、顧客のブラウザを経由してフィンテック事業者に認可コードを送信する（認可レスポンス）。

⑩ フィンテック事業者は認可コードを金融機関（認可サーバ）に送信し、アクセス・トークンを要求する（トークン・リクエスト）。

⑪ 金融機関はアクセス・トークンをフィンテック事業者に送信する（トークン・レスポンス）。

【決済の実行】

⑫ フィンテック事業者はアクセス・トークンを金融機関（リソース・サーバ）に送信し、口座に残高が十分に存在することを確認する（funds confirmation）。

⑬ 金融機関は確認結果をフィンテック事業者に返信する。

⑭ フィンテック事業者は、決済の処理内容を示すデータ POR（payment-order resource）をその識別情報とともに金融機関（リソース・サーバ）に送信する。

⑮ 金融機関は POR の識別情報をフィンテック事業者に送信し、決済の処理を実行する。
⑯ フィンテック事業者は POR の識別情報などを金融機関に送信し、決済の処理結果を問い合わせることができる。

上記のうち、②と③の処理はクライアント・クレデンシャル・フローに基づいて実行される。⑥から⑪の処理は、認可コード・フローまたは CIBA フローに基づいて実行される。

３．セキュリティ要件集の標準 FAPI 2.0

FAPI という用語は、FAPI 1.0 では「Financial-grade API」の略称であったが、FAPI 2.0 では（略称ではなく）FAPI そのものが正式名称とされた。これは、FAPI 2.0 が金融分野だけでなく公共や医療などの他分野でも活用されうるとの認識に基づいている[8]。本節では、FAPI 2.0 のセキュリティ目標と主なセキュリティ要件を説明する。FAPI 2.0 における想定環境、攻撃者の類型、主なセキュリティ要件を適用した認可コード・フォローについては補論 2 を参照されたい。

（１）セキュリティ目標

FAPI 2.0 のセキュリティ目標は OpenID Foundation の攻撃者モデル（attacker model）において以下のとおり示されている（Fett 2025）。ここでは、攻撃者が特定のフィンテック事業者や金融機関の顧客としてアカウントを保持している状況を前提としており、（自分以外の）他の顧客に対して攻撃を試みる状況を想定している。

- **【不正アクセスなどの防止】**攻撃者が、他の顧客の情報にアクセスしたり、（他の顧客が権限を有する）決済の処理を実行したりすることができない。
- **【不正ログインの防止】**攻撃者が、特定のフィンテック事業者のサービスにおいて他の顧客のアカウントへログインすることができない。
- **【他の顧客の誘導の防止】**攻撃者が、特定の金融機関において顧客として保持しているアカウントへ他の顧客を誘導したり[9]、特定のフィンテック

---

[8] 例えば、ノルウェーにおける医療サービスの公的ネットワーク NHN（Norwegian Health Network）では、医療機関がサービスを使用する際の認証基盤（HelseID）のセキュリティ要件集として FAPI 2.0 を採用している（OpenID Foundation 2025）。

[9] 例えば、攻撃者のアカウントへ顧客をアクセスさせ、そのアカウントに紐づく口座において送金などの取引を実行させる。攻撃者は後から自分のアカウントにアクセスして取引履歴を確認することによって、顧客の取引相手に関する情報（氏名、口座番号など）を盗取するという攻撃

事業者のサービスにおいて顧客として保持しているアカウントへ他の顧客を誘導したりする[10]ことができない。

（２）セキュリティ要件

FAPI 2.0 の主なセキュリティ要件は以下のとおりである[11]。

- **【コンフィデンシャル・クライアントの指定】**フィンテック事業者は、クライアント認証に用いる署名生成鍵などの秘密情報を保持する。こうしたクライアントはコンフィデンシャル・クライアントと呼ばれる。
- **【メタデータの提供方法の指定】**金融機関は、フィンテック事業者に自分のメタデータを、RFC 8414[12]や OpenID Connect Discovery[13]によって適切に提供する。
- **【認可フローの指定】**金融機関は、インプリシット・フローやリソース・オーナー・パスワード・クレデンシャル・フローによる認可リクエストを拒否すること（顧客のパスワードの漏洩の防止などが目的）[14]。
  —— この結果、認可コード・フローとクライアント・クレデンシャル・フローのみが使用可能である。
- **【クライアント認証方法の指定】**金融機関によるフィンテック事業者の認証の方法は、MTLS に基づくクライアント認証（以下、OAuth MTLS）[15]または

が想定される。

[10] 例えば、攻撃者が、顧客を自分のアカウントにアクセスさせたうえで、取引先の銀行の口座を登録するように誘導する攻撃が想定される。攻撃者は、後から自分のアカウントにアクセスし、顧客が登録した口座の情報を盗取する。

[11] FAPI 2.0 の主なセキュリティ要件がどのような攻撃への対応策となりうるかについては宇根［2025］において整理されている。

[12] RFC 8414（OAuth 2.0 Authorization Server Metadata）は、金融機関のメタデータ（サーバの識別子、認可リクエストやトークン・リクエストの送信先アドレス、認可プロトコルの処理フローの識別子など）の種類、メタデータのリクエストやレスポンスのメッセージ形式などを定める標準仕様である（Jones, Sakimura, and Bradley 2018）。

[13] OpenID Connect Discovery は、フィンテック事業者が OpenID Connect に基づく処理を実行する際に金融機関のメタデータを取得する方法を定める標準仕様である（Sakimura *et al.* 2023）。OpenID Connect は、フィンテック事業者が金融機関の認証結果に基づいて顧客の属性を検証し、顧客の属性情報を取得するプロトコルである（Sakimura *et al.* 2014）。

[14] これらの認可フローおよび認可コード・フローについては補論 1（2）を参照されたい。

[15] MTLS（Mutual Transport Layer Security）は、一般にクライアントとサーバが TLS においてそれぞれサーバ証明書とクライアント証明書を用いて相互に認証する手法である。OAuth 2.0 では、MTLS によってクライアント認証などを実行する方法が用いられる（RFC 8705、Campbell *et al.* 2020）。金融機関はクライアント証明書を用いてフィンテック事業者を認証し、その証明書と紐づけてアクセス・トークンを生成する（証明書のハッシュ値をアクセス・トークンに埋め込むな

private_key_jwt[16]とすること。

- **【認可リクエストの保護（PAR の必須化）】**金融機関は PAR[17]をサポートするとともに、PAR を使用しない認可リクエストを拒否すること。金融機関は認可リクエストの登録時にフィンテック事業者を認証すること。
- **【PAR の有効期間の設定】**金融機関は、PAR による認可リクエストの登録の有効期間（登録完了時点から失効までの時間）を 600 秒未満に設定すること。
- **【金融機関（認可サーバ）の検証】**フィンテック事業者は、認可レスポンスを受信した際に、認可コードの発行者の識別情報（iss）を用いて金融機関を特定・検証すること。
- **【認可コードの再使用の禁止】**金融機関は、一度使用された認可コードによるトークン・リクエストを拒否すること。
- **【認可コードの有効期間の設定】**金融機関は認可コードの有効期間を 60 秒未満とすること。
- **【署名アルゴリズムの指定】**金融機関とフィンテック事業者は、通信メッセージ（JWT）に適用する署名アルゴリズムに関して、RSA-PSS、ECDSA、EdDSA のいずれかを用いること。RSA の場合は最小鍵長を 2048 ビットとするほか、ECDSA または EdDSA の場合は最小鍵長を 224 ビットとすること。
- **【認可コードの不正使用防止（PKCE の必須化）】**金融機関は PKCE[18]の使用をフィンテック事業者に要求すること。

ど）。金融機関は、アクセス・トークン受信時に送信元のクライアント証明書とアクセス・トークンの対応関係を検証する。

16 private_key_jwt は、フィンテック事業者が自分の署名を金融機関へ送信し、金融機関が署名を検証してフィンテック事業者を認証する方法である（Sakimura *et al.* 2014）。署名は、JavaScript の記法を基にしたテキスト・ベースのデータ交換フォーマット JSON（JavaScript Object Notation）によるオブジェクト JWT（JSON Web Token）で表現される。

17 PAR（Pushed Authorization Request）は、フィンテック事業者が認可の内容を暗号化・署名付与したうえで事前に金融機関へ送信・登録する方法である（RFC9126、Lodderstedt *et al.* 2021）。金融機関は、認可の内容を受信すると、その内容や署名を検証し、当該データの格納場所の情報（request_uri）を返信する。フィンテック事業者は、request_uri と自分の ID のみを顧客のブラウザ経由で金融機関へ送信し、金融機関は事前に受信していた認可の内容に基づいて処理を開始する。これにより、認可の内容が顧客のブラウザから漏洩するリスクなどを低減可能となる。

18 PKCE（Proof of Key for Code Exchange）は、金融機関がフィンテック事業者と認可コードを紐づけることによって認可コードの送信者を確認する方法である（RFC 7636、Sakimura, Bradley, and Agarwal 2015）。フィンテック事業者は乱数とそのハッシュ値を生成し、ハッシュ値を金融機関に送信する。金融機関はハッシュ値を埋め込んだ認可コードを生成する、または、認可コードと紐づけてハッシュ値を記録するとともに、認可コードをフィンテック事業者に送信する。フィンテック事業者は認可コードと乱数を金融機関に送信し、金融機関は、乱数からハッシュ値を生成した後、認可コードに含まれている（または認可コードとともに記録されている）ハッシュ値と比較する。一致する場合、金融機関はフィンテック事業者を正当な送信者と判断する。

- **【認可レスポンスの暗号化による保護】**金融機関は、認可レスポンス（認可コードを含む）を、暗号化されていないネットワーク上で送信しないこと。
- **【アクセス・トークンの不正使用防止】**金融機関は記名式トークン（sender-constrained access token）のみを発行し、その送信元の確認方法として OAuth MTLS または DPoP[19]を用いること。
- **【リフレッシュ・トークンのローテーションの禁止】**金融機関は、原則としてリフレッシュ・トークンのローテーション[20]を実施しないこと。
- **【金融機関による顧客への情報提供】**金融機関は、顧客が認可の内容を正しく認識したうえで認可の処理を開始できるように、必要な情報（例えば、クライアントの識別子、認可の範囲）を顧客に提供することが望ましい（不正なサイトへの誘導や認可の内容の誤解の防止）。
- **【フィンテック事業者による顧客への情報提供】**フィンテック事業者は、顧客の同意によってのみ認可処理を開始すること、および、顧客が認可処理の開始を適切に認識するように情報を提供することを通じて認可処理の開始を保護すること（不正なサイトへの誘導や認可の内容の誤解の防止）。

４．海外の主なセキュリティ要件集と FAPI 2.0 との比較

本節では、海外における金融サービス向けのオープン API の主な活用事例として、検討が比較的早期に着手されたイギリスのオープン・バンキング（Open Banking）、ブラジルのオープン・ファイナンス（Open Finance）、オーストラリアのコンシューマー・データ・ライト（Consumer Data Right）、欧州連合のオープンファイナンス API フレームワーク（openFinance API フレームワーク）を取り上げる。また、これらのうち、独自のセキュリティ要件しているブラジル、オーストラリア、欧州連合のケースに関して、FAPI 2.0 と比較する。

---

[19] DPoP（Demonstrating Proof of Possession）は金融機関がフィンテック事業者の署名によってアクセス・トークンの送信者を確認する方法である（RFC 9449、Fett *et al.* 2023）。フィンテック事業者は、アクセス・トークン取得時に金融機関へ署名検証鍵と署名を送信し、金融機関は署名検証後に署名検証鍵のハッシュ値をアクセス・トークンに埋め込む（アサーション型トークンの場合）、または、アクセス・トークンと紐づけて記録する（ハンドル型トークンの場合）。フィンテック事業者は、リソース・リクエストの際にアクセス・トークンなどに対する署名と（署名生成に用いた署名生成鍵に対応する）署名検証鍵を送信する。金融機関は受信した署名検証鍵を用いて署名を検証し、その署名検証鍵のハッシュ値を、アクセス・トークン内の署名検証鍵のハッシュ値、または、認可サーバから取得した署名検証鍵のハッシュ値と比較して送信元を確認する。

[20] リフレッシュ・トークンは、アクセス・トークンが無効となった際に新しいアクセス・トークンを（認可処理を再度実行することなく）金融機関から得るために用いるトークンである。リフレッシュ・トークンのローテーションは、金融機関がリフレッシュ・トークンに基づいて新しいアクセス・トークンを発行する際にリフレッシュ・トークンも新しく発行する処理である。

（１）イギリスのオープン・バンキング

イギリスでは、2017 年、競争・市場庁（Competition and Markets Authority）が金融サービスの向上や競争促進などを目的としてオープン・バンキングを開始した（Competition and Markets Authority 2021）。オープン・バンキングでは、金融機関が保持している顧客情報の提供や決済の実行に加えて、住宅ローンをはじめとする融資、保険、身元確認などのサービスがオープン API によって実現されている（Open Banking Limited 2026a）。オープン API の技術仕様やセキュリティ要件の策定、オープン・バンキングの利用に関する各種統計情報の公表は、Open Banking Limited（OBL）によって行われている（Open Banking Limited 2026b）。

API の技術仕様 Read-Write API Profile では、金融機関やフィンテック事業者に対して、セキュリティ要件集として FAPI 1.0 のセキュリティ要件を満たすように実装することが必須とされている（Open Banking Limited 2024d）[21]。独自のセキュリティ要件集は策定されていないようである。

（２）ブラジルのオープン・ファイナンス

イ．概要

ブラジルでは、2021 年、通貨政策や信用制度に関する最高意思決定機関である国家通貨審議会（National Monetary Council）とブラジル中央銀行が、金融サービスにおけるオープン API 活用の取組みとしてオープン・バンキングを開始した（Open Finance Brasil 2022）。当初、オープン API の活用の対象は、金融機関による各種サービスに関する情報や顧客データの提供、決済の処理に限定されていたが、2022 年からは、住宅ローンや保険などにサービスの対象が拡大され、オープン・ファイナンスとの呼称が採用された。オープン・ファイナンスの活動は、オープン API の技術仕様やセキュリティ要件集の策定・管理、API の使用状況などを示す統計情報の公表などを担当する Open Finance Brasil（OFB）によって推進されている（Open Finance Brasil 2024）。

セキュリティ要件集である Open Finance Brasil Financial-Grade API Security Profile については、最新版（バージョン 1.0）の実装者向け文書（Implementers Draft 3）が 2024 年 5 月に発表されている（Open Finance Brasil Initial Structure 2024）。このセキュリティ要件集は、FAPI 1.0 をベースとしつつ、より高いレベルのセキュリティを達成することを目的として独自に策定されている。

---

[21] オープン・バンキングでは、OpenID Foundation が提供するコンフォーマンス・テスト・ツール（FAPI 1.0 に準拠したソフトウェアであることを検証）を用いて FAPI 1.0 の各セキュリティ要件を満たしていることを確認済みのソフトウェア製品を使用することを必須としている。

ロ．主なセキュリティ要件

セキュリティ要件集（実務者向け文書バージョン 3）に記載されている主なセキュリティ要件は以下のとおりである。

- **【フィンテック事業者の認証方法の指定】**金融機関（認可サーバ）はフィンテック事業者（クライアント）の認証に private_key_jwt を使用すること。
- **【メタデータの提供方法の指定】**金融機関は、メタデータを RFC 8414 または OpenID Connect Discovery による方法で提供すること。
- **【PAR の必須化】**金融機関はフィンテック事業者に対して PAR の実行を要求すること。
- **【PAR の有効期間の設定】**金融機関は、PAR による認可リクエストの登録の有効期間（登録完了時点から失効するまでの時間）を 60 秒以上としなければならない。
- **【PKCE の必須化】**金融機関は、フィンテック事業者に対して PKCE の使用を要求しなければならない。
- **【アクセス・トークンの有効期間の設定】**金融機関は、アクセス・トークンの有効期間を 300 秒から 900 秒の間に設定すること。
- **【リフレッシュ・トークンのローテーションの禁止】**金融機関は、原則としてリフレッシュ・トークンのローテーションを実施しないこと。
- **【暗号アルゴリズムの指定】**金融機関とフィンテック事業者は、通信メッセージへ付与する署名用のアルゴリズムとして RSA-PSS 署名（ハッシュ関数として SHA-256 を使用）を用いること。また、通信メッセージの暗号化に用いるアルゴリズムとして、公開鍵暗号には RSA-OAEP、共通鍵暗号には AES（256 ビット鍵長）を用いること。
- **【TLS に用いる暗号スイートの指定】**金融機関は、TLS による通信において用いる暗号アルゴリズムの組合せ（暗号スイート）として以下をサポートすること。
  - 鍵共有：ECDHE、署名：RSA、共通鍵暗号：AES（128 ビット鍵長）、ハッシュ関数：SHA-256
  - 鍵共有：ECDHE、署名：RSA、共通鍵暗号：AES（256 ビット鍵長）、ハッシュ関数：SHA-384

ハ．FAPI 2.0 との比較

（イ）FAPI 2.0 と整合的な要件

ブラジルのセキュリティ要件集に含まれる要件のうち、FAPI 2.0 に含まれるものと整合的な要件は以下のとおりである。

・　フィンテック事業者の認証方法
・　PAR の必須化
・　PKCE の必須化
・　リフレッシュ・トークンのローテーションの禁止
・　暗号アルゴリズムの指定
・　コンフィデンシャル・クライアントの指定

コンフィデンシャル・クライアントの指定に関する要件は明記されていないが、フィンテック事業者の認証方法として private_key_jwt を指定しており、それを実現するためにはフィンテック事業者は署名生成鍵を保持する必要がある。これは、フィンテック事業者がコンフィデンシャル・クライアントであることを前提としたものであり、FAPI 2.0 と整合的である。

（ロ）FAPI 2.0 よりもリスク低減に資する要件

アクセス・トークンの有効期間の設定に関する要件は FAPI 2.0 には含まれていない。アクセス・トークンの有効期間の設定は、アクセス・トークンが不正に使用されるリスクを低減するという観点で望ましく、リスク低減を意識した要件といえる

（３）オーストラリアのコンシューマー・データ・ライト

イ．概要

オーストラリアでは、2020 年、公正な企業活動や消費者保護に関する政策の立案・執行を担当する競争消費者委員会（Australian Competition and Consumer Commission）が、オープン API を介して顧客情報を適切に保護しつつ流通させる取組みとしてコンシューマー・データ・ライトを開始した（Commonwealth of Australia 2026）。コンシューマー・データ・ライトは、2020 年に金融分野に適用され、2022 年にエネルギー分野に適用された。今後、通信など他の業界へも展開される予定である。オープン API の技術仕様やセキュリティ要件集は、コンシューマー・データ・ライトを実現するための各種標準の策定を担当する Data Standards Body によって策定・管理されている（Data Standards Body 2026）。

セキュリティ要件集 Information Security Profile は FAPI 1.0 をベースとして独自に策定され、最新版が 2022 年 9 月に公表されている（Data Standard Body 2025）。

ロ．主なセキュリティ要件

オーストラリアの最新版のセキュリティ要件集に含まれている主なセキュリ

ティ要件は以下のとおりである[22]。

- **【フィンテック事業者の認証方法】**金融機関は、フィンテック事業者の認証方法として OAuth MTLS を使用しないこと。代わりに private_key_jwt などを使用すること[23]。
- **【コンフィデンシャル・クライアントの指定】**金融機関は、コンフィデンシャル・クライアントのみをサポートすること。
- **【認可コード・フローの使用】**フィンテック事業者は認可コード・フローのみを使用すること。
- **【認可コードの再使用の禁止】**フィンテック事業者は認可コードを再使用しないこと。
- **【PAR の必須化】**フィンテック事業者は PAR を使用すること。
- **【PAR の有効期間の設定】**金融機関は、PAR による認可リクエストの登録の有効期間を 10 秒から 90 秒の間に設定しなければならない。
- **【PKCE の必須化】**フィンテック事業者は PKCE を使用すること。
- **【各種リクエストの有効期間の設定】**フィンテック事業者は、各種リクエストの有効期間を 60 分以下に設定しなければならない。
- **【アクセス・トークンの有効期間の設定】**金融機関は、アクセス・トークンの有効期間を 120 秒から 600 秒の間に設定すること。
- **【リフレッシュ・トークンのローテーションの禁止】**金融機関はリフレッシュ・トークンのローテーションを実施してはならない。
- **【リフレッシュ・トークンのステータス確認の方法】**金融機関は、フィンテック事業者がリフレッシュ・トークンのステータス（失効の有無など）を確認するための方法（RFC 7662、Richer 2015）をサポートしなければならない。
- **【暗号アルゴリズムの指定】**金融機関とフィンテック事業者は、通信メッセージへ付与する署名用のアルゴリズムとして RSA-PSS 署名または ECDSA

---

[22] コンシューマー・データ・ライトにおける API の仕様では、金融機関を含む事業者が顧客に関する情報を第三者に提供する汎用的なシナリオを前提としている。そのため、エンティティは、①顧客（customer）、②顧客データの保持者（data holder）、③顧客データの取得者（data recipient）、④顧客データの取得者が準備するソフトウェア製品（software product）とそれぞれ呼ばれている。これらを OAuth 2.0 のエンティティに対応させると、顧客データの保持者および取得者がそれぞれ金融機関とフィンテック事業者、顧客データの取得者が準備するソフトウェア製品がブラウザに相当することから、これらの名称を使用して説明する。

[23] OAuth MTLS を用いないこととした理由として、OAuth MTLS を別の認証目的（トランザクション認証〈transaction security〉など）で使用することを想定している旨が記述されている。トランザクション認証は、顧客が送金などの決済の処理を実行する直前に、金融機関がその可否を顧客に改めて確認するための処理の総称である。

（いずれもハッシュ関数として SHA-256 を使用）を用いること。また、通信メッセージの暗号化に用いるアルゴリズムとして、公開鍵暗号には RSA-OAEP、共通鍵暗号には AES（128 ビット鍵長または 256 ビット鍵長）を用いること。

ハ. FAPI 2.0 との比較

（イ）FAPI 2.0 と整合的な要件

以下の要件が FAPI 2.0 と整合的であり、ブラジルとほぼ同じである。

- フィンテック事業者の認証方法
- PAR の必須化
- PKCE の必須化
- 認可コードの再使用の禁止
- リフレッシュ・トークンのローテーションの禁止
- 暗号アルゴリズムの指定
- コンフィデンシャル・クライアントの指定

（ロ）FAPI 2.0 よりもリスク低減に資する要件

以下の 4 つの要件は、いずれも FAPI 2.0 に含まれておらず、リスク低減に資する内容である。

・アクセス・トークンの有効期間の設定

FAPI 2.0 はアクセス・トークンの有効期間に関するセキュリティ要件を定めていないが、オーストラリアの要件集は 120 秒から 600 秒としており、アクセス・トークンが不正に使用されるリスクの低減が期待される。

・PAR の有効期間の設定

FAPI 2.0 では、有効期間を 600 秒以下とすることとしている。オーストラリアの要件集では、これよりも短い有効期間（10 秒から 90 秒の間）を要求しており、PAR の処理が不正に操作されるリスクの低減が期待される。

・各種リクエストの有効期間の設定

FAPI 2.0 は各種リクエストの有効期間に関するセキュリティ要件を定めていないが、オーストラリアの要件集は 60 分以内に制限しており、各種リクエストが不正に使用されるリスクの低減が期待される。

・認可コード・フローの使用

FAPI 2.0 では、OAuth 2.0 の認可フローのうち、認可コード・フローとクライアント・クレデンシャル・フローのみを使用可能としている。クライアント・

クレデンシャル・フローは認可コードを使用する代わりにフィンテック事業者のクレデンシャル（パスワードなどの秘密情報）を用いるフローであり、金融機関が顧客の同意を直接確認したうえで一度限り使用できる認可コードを用いる認可コード・フローの方が顧客同意の確実性の点で望ましい。

（４）欧州連合のオープンファイナンス API フレームワーク

イ．概要

欧州連合では、2015 年、オンライン決済のセキュリティ向上などを目的とする欧州決済指令第 2 版（PSD2: Payments Services Directive 2）が成立し、加盟各国が 2018 年 1 月までに国内法を整備することとなった。PSD2 では、金融機関やフィンテック事業者に対して高いセキュリティを実現する顧客認証（Strong Customer Authentication）を実施するほか、金融機関に対して顧客情報をフィンテック事業者（Third Party Provider）にオープン API を介して開放することが義務付けられた。

PSD2 に準拠するオープン API の仕様は、域内の各種決済サービスの技術仕様の標準化を担当しているベルリン・グループ（Berlin Group）がオープンファイナンス API フレームワークの一部として策定している（Berlin Group 2026）。セキュリティ要件については、オープン API におけるプロトコルの機能とセキュリティ対策（Protocol Functions and Security Measures）に関するガイドラインに記述されている（Berlin Group 2025、最新版はバージョン 2.3〈2025 年 10 月公表〉）。このガイドラインは、認可フローに関して OAuth 2.0 に基づくフローの採用をオプションとしているほか、セキュリティ要件集として FAPI を参照していない。これは、欧州域内のさまざまな金融機関やフィンテック事業者において活用されることを想定し、オープン API の仕様の自由度を高めることを意図したものであると考えられる。

ロ．主なセキュリティ要件

プロトコルの機能とセキュリティ対策に関するガイドラインでは、OAuth 2.0 に基づく認可処理向けのセキュリティ要件がブラジルやオーストラリアの場合と比べると少ない。セキュリティ対策を検討する際には、OAuth 2.0 向けの最新のベスト・プラクティス（Best Current Practice for OAuth 2.0 Security、Lodderstedt *et al.* 2025）を参照することを推奨しており、具体的なセキュリティ要件の設定を各金融機関やフィンテック事業者が委ねる姿勢がうかがわれる。これは、加盟各国における通信環境や規制が異なるなかで、各金融機関やフィンテック事業者がそれぞれ直面している状況に見合ったセキュリティ対策を検討・実施できるようにためと考えられる。

主なセキュリティ要件は以下のとおりである。

- **【フィンテック事業者の認証方法】**フィンテック事業者を認証する方法として OAuth MTLS を使用すること。
  - TLS のバージョンは 1.2 以降とすること。
  - 暗号アルゴリズムを選択する際には、NIST が推奨する暗号強度や各国の IT セキュリティ当局の要求事項を参照すること。
- **【各種リクエストへの署名付与】**フィンテック事業者は各種リクエストのメッセージに対して署名を付与すること。
- **【認可コード・フローの使用】**認可の処理フローとして認可コード・フローを使用すること。
- **【PKCE の必須化】**認可コードの不正な使用への対策として PKCE を使用すること。

ハ. FAPI 2.0 との比較

フィンテック事業者の認証方法および PKCE の必須化の要件は FAPI 2.0 と整合的である。

認可コード・フローの使用に関する要件は、その他の認可フローを許容している FAPI 2.0 と比べて、顧客の同意を金融機関が直接確認できるという点で望ましいといえる。

（５）FAPI 2.0 との比較結果のまとめ

本節（4）で比較した結果をまとめると以下のとおりである。

- ブラジルとオーストラリアのセキュリティ要件集は、①フィンテック事業者のなりすまし防止、②認可リクエストの内容の保護、③認可コードの不正使用防止、④リフレッシュ・トークンの不正使用防止、⑤暗号アルゴリズムによる通信データの保護の観点で、FAPI 2.0 と同等のセキュリティを既に要求しているといえる。
- 欧州連合のガイドラインは、①フィンテック事業者のなりすまし防止、②認可コードの不正使用防止の観点で、FAPI 2.0 と同等のセキュリティを既に要求しているといえる。
- ブラジルのセキュリティ要件集は、アクセス・トークンの有効期間の設定の要件を含んでおり、アクセス・トークンの不正使用によるリスク低減の観点で FAPI 2.0 よりも望ましい。
- オーストラリアのセキュリティ要件集は、①アクセス・トークン、各種リク

エスト、PAR の不正使用によるリスク低減の観点、および、②顧客の同意確認の確実性の観点で、FAPI 2.0 よりも望ましい。

・ 欧州連合のガイドラインは、顧客の同意確認の確実性の観点で、FAPI 2.0 よりも望ましい。

５．おわりに

本稿では、オープン API におけるセキュリティ要件集として最近標準化された FAPI 2.0 を紹介し、FAPI 2.0 公表以前に策定されている海外のセキュリティ要件集やガイドラインの内容と比較した。ブラジルやオーストラリアの要件集は、FAPI 2.0 と整合的な要件を既に複数含んでいたほか、FAPI 2.0 よりもリスク低減に資する要件も含んでいた。欧州連合のガイドラインも、FAPI を参照しているわけではないが、FAPI 2.0 と整合的な要件やリスク低減に資する要件を含んでいた。

今後、本邦金融機関やフィンテック事業者がオープン API の認可処理におけるセキュリティ要件を検討する際には、FAPI 2.0 を参照するだけでなく、海外のセキュリティ要件集を参照することも有用である。例えば、アクセス・トークンや各種リクエストの有効期間に関する要件を設定する場合、FAPI 2.0 にはこれらの要件が含まれておらず、独自に検討する必要があるが、その際に、オーストラリアのセキュリティ要件集に含まれている要件を参照することができる。こうした対応は、海外のオープン API のサービスと同程度のセキュリティを確保するという観点からも重要である。これは、仮に海外のサービスよりもセキュリティが劣ったサービスを提供していた場合、自社がサイバー攻撃の主な標的となってしまう可能性があるためである。

海外のセキュリティ要件集には、一部ではあるが、FAPI 2.0 よりもリスク低減に資する要件が含まれていた。こうした対応の意図や背景は明らかでないが、将来のサイバー攻撃の高度化を見据えて、より高度なセキュリティ対策を早期に実現するために設定した可能性もある。一般に、サイバー攻撃は時間とともに高度化・巧妙化し、既存のセキュリティ対策の効果は徐々に低下する傾向にあるが、オープン API のセキュリティに関しても同様である。FAPI 2.0 という標準に準拠するだけでなく、サイバー攻撃の先行きの動向も考慮しつつ、より高度なセキュリティ対策を必要に応じて採用するという姿勢も有用であろう。

以　上

【参考文献】

一般社団法人全国銀行協会、「銀行分野のオープン API に係る電文仕様標準について 第 2 版」、一般社団法人全国銀行協会、2017 年（https://www.zenginkyo.or.jp/fileadmin/res/abstract/council/openapi/openapi_sp_1.pdf、2026 年 3 月 25 日）

宇根正志、「オープン API のセキュリティ：認可処理における脆弱性と対策の高度化」、『金融研究』第 44 巻第 1 号、日本銀行金融研究所、2025 年、19～48 頁

株式会社みんなの銀行、「世界トップレベルのセキュリティ規格『FAPI』に準拠した BaaS プラットフォームを最新仕様 FAPI 2.0 に対応」、株式会社みんなの銀行、2024 年（https://corporate.minna-no-ginko.com/information/corporate/2024/03/15/505、2026 年 3 月 25 日）

公益財団法人金融情報システムセンター、「API 接続チェックリスト〈2025 年 12 月版〉公表のお知らせ」、公益財団法人金融情報システムセンター、2026 年（https://www.fisc.or.jp/document/fintech/007138.php、2026 年 3 月 25 日）

Berlin Group. 2025. “openFinance API Framework Implementation Guidelines: Protocol Functions and Security Measures, Version 2.3.” Berlin Group. Accessed March 25, 2026. https://www.berlin-group.org/openfinance-downloads.

————. 2026. “PSD2 Access to Bank Accounts.” Berlin Group. Accessed March 25, 2026. https://www.berlin-group.org/psd2-access-to-bank-accounts.

Campbell, Brian, John Bradley, Nat Sakimura, and Torsten Lodderstedt. 2020. “OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens.” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc8705.html.

Commonwealth of Australia. 2026. “Consumer Data Rights: Rollout.” Commonwealth of Australia. Accessed March 25, 2026. https://www.cdr.gov.au/rollout.

Competition and Markets Authority. 2021. “Update on Open Banking.” Competition and Markets Authority. Accessed March 25, 2026. https://www.gov.uk/government/publications/update-governance-of-open-banking/update-on-open-banking.

Data Standard Body. 2025. “CDR Data Standards.” Github. Accessed March 25, 2026. https://consumerdatastandardsaustralia.github.io/standards.

————. 2026. “Data Standards – Consumer Data Right.” Data Standards Body. Accessed March 25, 2026. https://dsb.gov.au/consumer-data-right/data-standards.

Fett, Daniel. 2025. “FAPI 2.0 Attacker Model.” OpenID Foundations. Accessed March 25, 2026. https://openid.net/specs/fapi-attacker-model-2_0-final.html.

————, Brian Campbell, John Bradley, and Torsten Lodderstedt. 2023. “OAuth 2.0 Demonstrating Proof of Possession (DPoP).” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc9449.html.

————, Dave Tonge, and Joseph Heenan. 2025. “FAPI 2.0 Security Profile.” OpenID Foundations. Accessed March 25, 2026. https://openid.net/specs/fapi-security-profile-2_0-final.html.

Hardt, Dick. 2012. “The OAuth 2.0 Authorization Framework.” Accessed March 25, 2026. https://www.rfc-editor.org/rfc/pdfrfc/rfc6749.txt.pdf.

Jones, Michael B., and Dick Hardt. 2012. “The OAuth 2.0 Authorization Framework: Bearer Token Usage.” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc6750.html.

————, Nat Sakimura, and John Bradley. 2018. “OAuth 2.0 Authorization Server Metadata.” IETF. Accessed March 25, 2026. https://rfc-editor.org/rfc/pdfrfc/rfc8414.txt.pdf.

Lodderstedt, Torsten, John Bradley, Andrey Labunets, and Daniel Fett. 2025. “Best Current Practice for OAuth 2.0 Security.” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc9700.html.

————, Brian Campbell, and Nat Sakimura. 2021. “OAuth 2.0 Pushed Authorization Requests.” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc9126.html.

Open Banking Limited. 2024a. “Open Banking Read-Write API Profile – v4.0.” Github. Accessed March 25, 2026. https://openbankinguk.github.io/read-write-api-site3/v4.0/profiles/read-write-data-api-profile.html.

————. 2024b. “Account and Transaction API Profile – v4.0.” Github. Accessed March 25, 2026. https://openbankinguk.github.io/read-write-api-site3/v4.0/profiles/account-and-transaction-api-profile.html.

————. 2024c. “Payment Initiation API Profile – v4.0.” Github. Accessed March 25, 2026. https://openbankinguk.github.io/read-write-api-site3/v4.0/profiles/payment-initiation-api-profile.html.

————. 2024d. “Getting Started – Open Banking API Security Profile.” Open Banking Limited. Accessed March 25, 2026. https://standards.openbanking.org.uk/security-profiles/get-started-obl-api-security-profile.

————. 2026a. “Case Studies: Open Banking in Real Life.” Open Banking Limited.

Accessed March 25, 2026. https://www.openbanking.org.uk/case-studies.

————. 2026b. “API Performance Stats.” Open Banking Limited. Accessed March 25, 2026. https://www.openbanking.org.uk/api-performance.

Open Finance Brasil Initial Structure. 2022. “Qual a Diferença Entre Open Banking e Open Finance?” Open Finance Brasil. Accessed March 25, 2026. https://openfinancebrasil.org.br/2022/11/17/qual-a-diferenca-entre-open-banking-e-open-finance.

————. 2024. “Open Finance Brasil Financial-Grade API Security Profile 1.0 Implementers Draft 3.” Atlassian Net. Accessed March 25, 2026. https://openfinancebrasil.atlassian.net/wiki/spaces/OF/pages/245760001/EN+Open+Finance+Brasil+Financial-grade+API+Security+Profile+1.0+Implementers+Draft+3.

OpenID Foundation. 2025. “Scaling FAPI 2.0 to Transform Healthcare Security in Norway.” OpenID Foundation. Accessed March 25, 2026. https://openid.net/scaling-fapi-2-0-to-transform-healthcare-security-in-norway/.

Richer, Justin. 2015. “OAuth 2.0 Token Inspection.” IETF. Accessed March 25, 2026. https://datatracker.ietf.org/doc/html/rfc7662.

Sakimura, Nat, John Bradley, and Naveen Agarwal. 2015. “Proof Key for Code Exchange by OAuth Public Clients.” IETF. Accessed March 25, 2026. https://www.rfc-editor.org/rfc/rfc7636.

————, ————, and Edmund Jay. 2021a. “Financial-Grade API Security Profile 1.0 – Part 1: Baseline.” OpenID Foundation. Accessed March 25, 2026. https://openid.net/specs/openid-financial-api-part-1-1_0.html.

————, ————, and ————. 2021b. “Financial-Grade API Security Profile 1.0 – Part 2: Advanced.” OpenID Foundation. Accessed March 25, 2026. https://openid.net/specs/openid-financial-api-part-2-1_0.html.

————, ————, Michael B. Jones, and Edmund Jay. 2023. “OpenID Connect Discovery 1.0 Incorporating Errata Set 2.” OpenID Foundation. Accessed March 25, 2026. https://openid.net/specs/openid-connect-discovery-1_0.html.

————, ————, ————, Breno de Medeiros, and Chuck Mortimore. 2014. “OpenID Connect Core 1.0 Incorporating Errata Set 1.” OpenID Foundation. Accessed March 25, 2026. https://openid.net/specs/openid-connect-core-1_0-errata1.html.

Sheffer, Yaron, Peter Saint-Andre, and Thomas Fossati. 2022. “Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS).” IETF. Accessed March 25, 2026. https://www.rfc-

editor.org/rfc/rfc9325.pdf.

Tonge, Dave, Joseph Heenan, Torsten Lodderstedt, and Brian Campbell. 2019. "Financial-Grade API: Client Initiated Backchannel Authentication Profile." OpenID Foundation. Accessed March 25, 2026. https://openid.net/specs/openid-financial-api-ciba.html.

補論１．OAuth 2.0 に基づく認可処理

（１）エンティティ

OAuth 2.0 に登場する主なエンティティは、リソース・オーナー、ユーザ・エージェント、クライアント、リソース・サーバ、認可サーバである。ここでは、理解しやすさの観点から、フィンテック事業者が金融機関にオープン API を介してアクセスし、顧客の口座の取引履歴を取得する場合を想定して説明する。

- **リソース・オーナー（顧客）**

リソース・オーナーは金融機関の顧客であり、フィンテック事業者の顧客でもある。ここでのリソースとは、フィンテック事業者が取得する顧客の情報（金融機関に顧客が開設している口座の取引履歴など）やフィンテック事業者が顧客に代わって実行する取引（送金など）を意味している。以下では、理解しやすくするために、リソース・オーナーを顧客と呼ぶ。

- **ユーザ・エージェント（ブラウザ）**

ユーザ・エージェントは、顧客がフィンテック事業者や金融機関とやり取りするためのソフトウェアであり、顧客の端末のブラウザが想定されている。以下では、ユーザ・エージェントをブラウザと呼ぶ。

- **クライアント**

クライアントは、フィンテック事業者が顧客にサービスを提供するためのアプリケーション・ソフトウェアである。クライアントには、顧客の端末内で動作するもの（ネイティブ・アプリ） と、フィンテック事業者のサーバで動作するもの（サーバ・アプリ）がある。

OAuth 2.0 では、ネイティブ・アプリに関して、各アプリが固有の秘密情報（暗号鍵など）を安全に使用することができない場合（パブリック・クライアント）を対象としている。サーバ・アプリに関しては、固有の秘密情報を安全に使用することができる場合（コンフィデンシャル・クライアント）を想定している。

- **リソース・サーバ**

リソース・サーバは金融機関のサーバであり、顧客の口座を管理して口座の取引履歴を提供したり送金を実行したりする役割を担っている。

- **認可サーバ**

認可サーバは金融機関のサーバであり、顧客の認可を確認したうえでフィンテック事業者に顧客のリソースへのアクセスを許可する役割を担っている。

（２）認可処理

OAuth 2.0 に基づく認可処理の種類として、①認可コード・フロー、②インプ

リシット・フロー[24]、③クライアント・クレデンシャル・フロー、④リソース・オーナー・パスワード・クレデンシャル・フロー[25]が挙げられる。どのフローを選択するかは、フィンテック事業者がアプリケーションやそれを実装する環境を考慮して決定することができる。このように、サービスの内容や実装環境に応じて認可処理の内容を選択・設計することができる点が OAuth 2.0 の特長の 1 つである。

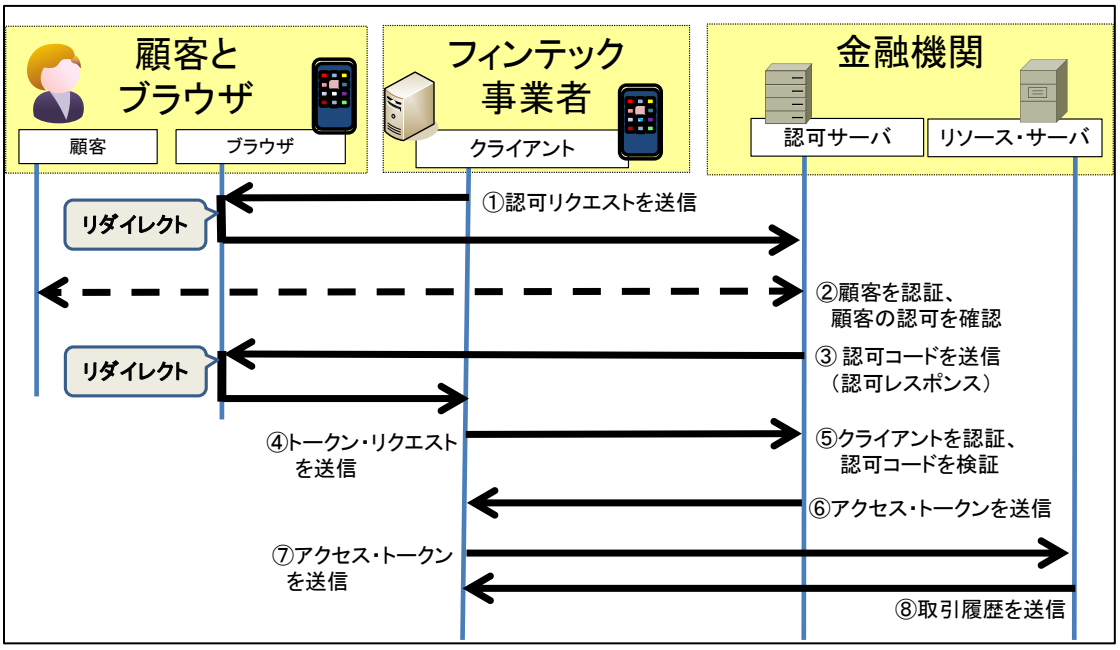
ここでは、代表的な処理フローである認可コード・フローの概要を紹介する（図 A-1 を参照）。

【認可コード・フロー】

① フィンテック事業者は、顧客の口座の取引履歴へのアクセスを求めるメッセージ（認可リクエスト）を金融機関（認可サーバ）に送信する。その際、認可リクエストは顧客のブラウザを経由して送られる。

② 金融機関は、顧客を認証し、顧客がフィンテック事業者によるアクセスを認めているか否か（認可の有無）を確認する。

図 A-1　OAuth 2.0 における認可コード・フロー（取引履歴を取得するケース）



24 インプリシット・フローは、金融機関が認可リクエストを検証した後、（認可コード・フローにおける認可コードを発行せずに）アクセス・トークンを発行する方式である。

25 リソース・オーナー・パスワード・クレデンシャル・フローは、フィンテック事業者が顧客からユーザ名とパスワードを受け取り、それを金融機関に送信することによってアクセス・トークンを得る方式である。

③ 金融機関は、認可の有無の確認結果を示すデータ（認可を確認した場合には認可コード、そうでない場合はエラー・メッセージ）を含むメッセージ（認可レスポンス）をフィンテック事業者に送信する。認可レスポンスは顧客のブラウザを経由して送られる。
④ フィンテック事業者は、金融機関を認証した後、認可コードを送信し、アクセス・トークンの発行を依頼する（トークン・リクエスト）。
⑤ 金融機関は、フィンテック事業者を認証した後、認可コードを検証する。具体的には、認可コードが有効であること、アクセス・トークンの送信先が認可コードの送信先と一致していることなどを確認する。
⑥ 金融機関は、認可コードの検証結果を示すデータ（成功の場合はアクセス・トークン、そうでない場合はエラー・メッセージ）をフィンテック事業者に送信する（トークン・レスポンス）。リフレッシュ・トークンも発行・送信する場合がある。
⑦ フィンテック事業者は、アクセス・トークンを金融機関（リソース・サーバ）に送信する。
⑧ 金融機関は、アクセス・トークンの検証が成功した場合、フィンテック事業者に口座の取引履歴を送信する。

補論２．FAPI 2.0 の概要

（１）想定環境

FAPI 2.0 の攻撃者モデル（Fett 2025）に記載されている主な想定環境は以下のとおりである。

- 各エンティティは TLS（バージョン 1.2、または、それよりも新しいバージョン[26]）による暗号通信を実施するほか、TLS のサーバ証明書を検証する。
- TLS は適切に動作し、通信データの一貫性や機密性が確保される。
- 各エンティティは、暗号データの復号鍵と署名検証鍵を安全に入手する。
- 顧客が使用する（端末などの）機器やブラウザは期待通りに動作する。
- 攻撃者によって悪用されていないエンティティは期待通りに動作する。
- フィンテック事業者や金融機関（認可サーバ）による顧客の登録・本人確認、当人確認、ID・アクセス管理は期待通りに動作する。
- 顧客による各セッションは他の顧客や攻撃者に対して適切に保護される。
- フィンテック事業者は、金融機関（認可サーバ）へのアクセス先の情報を安

[26] FAPI 2.0 では、TLS 1.2 によって接続する場合、十分な安全性を確保できる暗号アルゴリズムを採用するなど、TLS の安全な使用のための推奨事項（Sheffer, Saint-Andre, and Fossati 2022）に従うことが規定されている。

全に入手し、適切に金融機関へアクセスする。
・ 暗号鍵などの秘密情報はランダムで安全に生成され、攻撃者が効率的に推定することは困難である。

また、以下の各事項がスコープ外とされている。

・ 認可コードやアクセス・トークンなどの漏洩を引き起こす実装上の不具合（例えば、データベースの設定ミス、OS やブラウザの設定ミスなど）
・ 金融機関（認可サーバ）においてマルウェアなどを実行させる攻撃（remote code execution）
・ フィッシングによる攻撃
・ 金融機関やフィンテック事業者による顧客の本人確認の手法
・ ファイアウォールなどのセキュリティ機器の設定
・ ソフトウェア開発手法

（２）攻撃者の類型

攻撃者モデルでは、想定する攻撃者の類型として表 A-1 の 5 つが挙げられている。特に、A3a と A5 の攻撃者は、金融機関（認可サーバとリソース・サーバ）にアクセス可能であり、相当高度なスキルや権限を有していることが前提とされていることから、比較的強力な攻撃者といえる。

A3 の攻撃者に関しては、例えば、フィンテック事業者から金融機関（認可サーバ）へ送信された認可リクエストを（金融機関が受信する前に）横取りし、攻撃

表 A-1 攻撃者の類型

類型	攻撃者ができること・できないこと
A1	・ 通常の顧客としてプロトコルに参加し、メッセージを送受信可能。 ・ ブラウザ開発ツールや自作のソフトウェアなどのツールを用いて、既存のメッセージの解析や新メッセージの生成が可能。 ・ 顧客を不正なサイトに誘導するためのリンクを送信可能。 ・ 他のエンティティ間のメッセージの横取りやブロックは不可能。 ・ 金融機関（認可サーバ）として振舞うことは不可能。
A1a	・ A1 の能力に加えて、金融機関（認可サーバ）から取得したメッセージを使用したり、顧客を金融機関へ誘導したりすることができる。
A2	・ 無線アクセス・ポイントを悪用したり、脆弱なネットワーク・ノードを不正に操作したりすることができる。 ・ メッセージの横取り・ブロック・改変が可能。
A3a	・ A1 の能力に加えて、認可リクエストを盗聴可能。 ―― ただし、認可レスポンスは盗聴不可能。
A5	・ A1 の能力に加えて、リソース・リクエストを盗聴可能。 ―― ただし、リソース・レスポンスは盗聴不可能。

者自身のリソースへアクセスする内容のものに差し替える攻撃[27]が想定される。

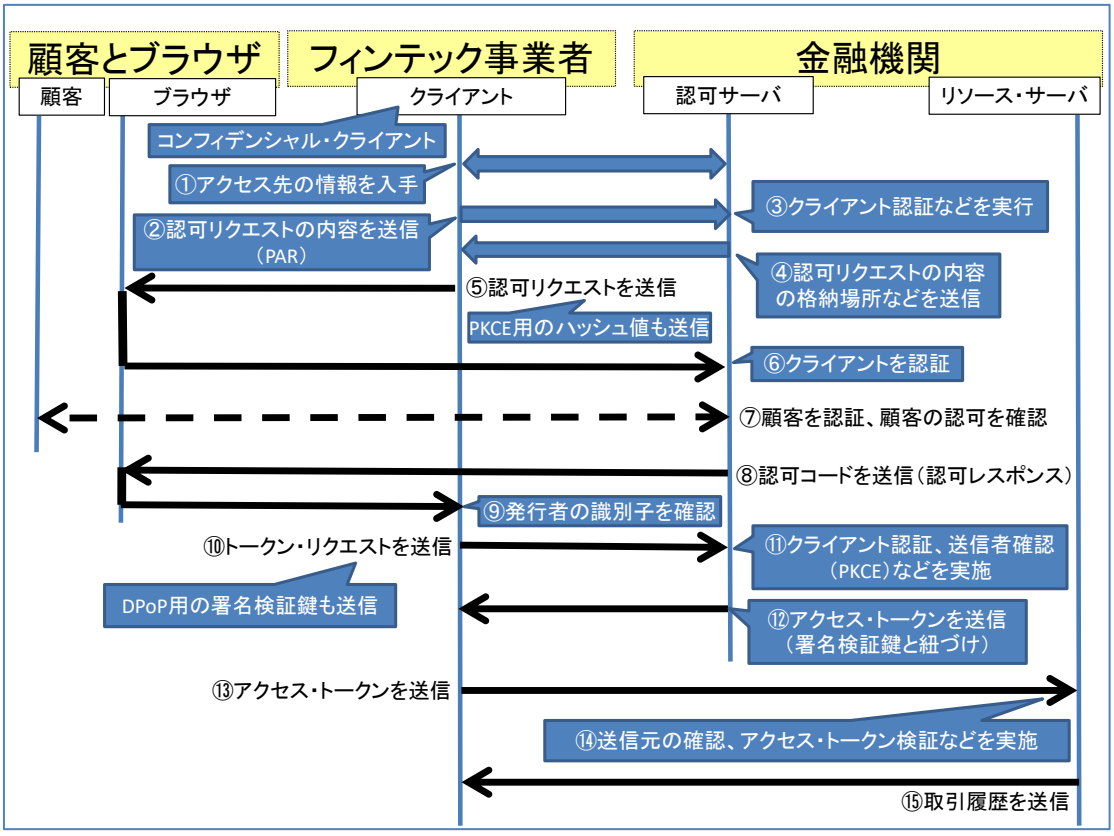
A5 の攻撃者に関しては、例えば、金融機関（リソース・サーバ）が DPoP によってアクセス・トークンの送信元を確認する場合、フィンテック事業者の署名を攻撃者が横取りし、それを後から金融機関に送信してリソースへのアクセスを試みる攻撃が想定される。

（３）セキュリティ要件を適用した認可コード・フロー

2 節（2）で説明した認可コード・フローに FAPI 2.0 の主なセキュリティ要件を適用すると、以下の処理フローとなる（図 A-2 を参照）。

① フィンテック事業者（コンフィデンシャル・クライアント）は、PAR を実行するために、金融機関（認可サーバ）のアクセス先の情報を入手する。

図 A-2 FAPI 2.0 の主なセキュリティ要件を適用した認可コード・フロー



注：図中の吹出しの部分が主に要求事項として追加された処理を表す。

[27] この攻撃は Cross Site Request Forgery と呼ばれ、顧客（被害者）に攻撃者自身のリソースへアクセスさせ、顧客の取引に関する情報を盗取する攻撃である。例えば、顧客に送金などの取引を攻撃者の口座上で実行させ、後から自分の口座の履歴をチェックして、顧客による送金の相手や金額などの情報を入手する場合が想定される。

② フィンテック事業者は、認可リクエストの内容（例えば、顧客の口座の取引履歴へのアクセスを依頼）に関するデータを金融機関へ送信する（PAR を採用）。
③ 金融機関は、フィンテック事業者を認証するとともに、認可リクエストの内容に関するデータなどを確認する。
④ 確認終了後、金融機関は、認可リクエストの内容に関する情報の格納場所の情報などをフィンテック事業者に送信する。
⑤ フィンテック事業者は、認可リクエストとして、自分の ID と、認可リクエストの内容に関する情報の格納場所を示す情報を、顧客のブラウザを経由して金融機関に送信する。このとき、PKCE を実行するために乱数のハッシュ値も送信する。
⑥ 金融機関はフィンテック事業者を認証する。
⑦ 金融機関は、顧客を認証し、フィンテック事業者へのアクセスを認めるか否かを確認する。
⑧ 金融機関は、確認が成功した場合には認可コードを生成し、認可コードや発行者の識別情報（iss）を含むメッセージ（認可レスポンス）をブラウザ経由でフィンテック事業者に送信する。このとき、金融機関は、認可コードと PKCE 用のハッシュ値を紐づけて記録する。
⑨ フィンテック事業者は認可レスポンスに含まれる発行者の識別子を確認する。
⑩ フィンテック事業者は、認可コードに加えて、DPoP を使用する際には署名検証鍵を金融機関に送信し、アクセス・トークンの発行を依頼する（トークン・リクエスト）。
⑪ 金融機関は、クライアント認証（OAuth MTLS または private_key_jwt を使用）を実行した後、PKCE によって認可コードの送信元を確認する。さらに、認可コード自体の検証（認可コードが使用済みか否かなど）も行う。
⑫ 金融機関は、アクセス・トークンを生成してフィンテック事業者に送信する（トークン・レスポンス）。DPoP を使用する場合、金融機関は、アクセス・トークンと DPoP 用の署名検証鍵を紐づけて記録する。
⑬ フィンテック事業者は、アクセス・トークンを金融機関（リソース・サーバ）に送信する。
⑭ 金融機関は、アクセス・トークンの送信者を確認する（DPoP または OAuth MTLS を使用）ほか、アクセス・トークンの有効性などを検証する。
⑮ 上記検証が成功した場合、金融機関は取引履歴をフィンテック事業者に送信する。