

ビジネスエコシステムを実現する BankVision® の API 公開

Release of BankVision API Platform that Realizes Business Ecosystem

大野 上 総, 田島 太 朗

要 約 2015年頃からわが国では、銀行の機能（残高照会など）を WebAPI として公開し、他社から利用させることが検討されてきた。2017年5月には改正銀行法が成立し、銀行と Fintech 企業との API 連携について法制度が整備され、銀行 API 普及への道筋が立った。

日本ユニシスでは、銀行勘定系パッケージ「BankVision®」の機能を API として公開する「BankVision WebAPI 公開サービス」の提供を 2016年度から開始した。さらに 2017年度にはそれらの API をインターネット上にも公開可能とする「Resonatex™」（レゾナテックス）を提供する予定である。Resonatex は、BankVision だけでなく、他のシステムの API 公開にも利用できる。今後銀行 API 活用が進むことにより、業界を越えた企業間の連携による新しいサービスが次々と生み出されていくだろう。

Abstract In year around 2015, it has been studied in Japan to disclose the functions of banks as WebAPI (eg. balance inquiry etc.) and to enable the use of those functions by other companies. The revised Banking Act was enacted in May 2017. Regarding the cooperation between banks and Fintech companies using the API, the legal system was improved and a path to the popularization of the Banking API was established.

Since FY 2016, Nihon Unisys started to offer the “BankVision® WebAPI Public Service”, which provides the functions of the banking account package “BankVision” as an API. In addition, Nihon Unisys plans to start the offering of “Resonatex” which enables those APIs open on the Internet in FY 2017. Resonatex can be used for API release of other systems as well as BankVision. In the future, as the utilization of Banking APIs become active, new services will be generated one after another through the cooperation among the companies across the industries.

1. はじめに

API という技術が、わが国の銀行サービスを大きく変えようとしている。これまでの銀行サービスは、サービスの利用者の声を背景にしつつも、それに対して「サービス供給者である銀行としてどのようなサービスを提供できるのか」という目線で開発・提供されてきた。しかしインターネット・スマートフォン・SNS 等の普及により、社会全体としてサービスの供給者から利用者へのパワーシフトが進みつつある。こうした中で利用者から今求められているのは「利用者が圧倒的な利便性を享受できる」サービスである。この利便性には、直感的に操作できる優れた UI/UX、ストレスのない動作スピード等が含まれる。こうしたサービスを実現するには、サービス供給者の目線から一旦離れ、利用者の目線からサービスを開発する必要がある。ここ数年脚光を浴びている Fintech サービスの多くは、このような考え方で開発されたものである。さらに付加すると、利用者からはこうしたサービスを、安価に（できれば無料で）、かつニーズに応じてスピーディに提供することが求められることは言うまでもない。

一方で、銀行が社会に提供している機能は、今日では重要な社会インフラとなっている。銀行としては利用者の利便性だけを考えるわけにいかず、利用者の安全・安心、従来から提供している機能との整合性・堅牢性なども確保する社会的責任がある。

では銀行が従来からの社会的責任を果たしつつも、今求められているサービスを実現するには、どうすれば良いのだろうか。回答は、サービス供給者である銀行だけですべてを実現するのではなく、サービス利用者側の目線で開発された外部のサービスと連携することである。こうした連携を実現するための手法として脚光を浴びているのが、API (Application Programming Interface) である。APIとは、あるソフトウェアコンポーネントから、別のソフトウェアコンポーネントを呼び出して利用する際の仕様のことである。API自体は古くからある概念だが、本稿で取り上げるAPIは、いわゆるWebAPIのことであり、あるシステム/サービスから、Web技術を用いて、他のシステム/サービスが提供する機能呼び出して利用する機能を指す。特に、自社内だけでなく、他社も含めて利用可能となるものを、オープンAPI*¹と呼ぶ。APIの技術によって、銀行自身が提供するサービスを部品化し、Fintechサービス等の外部サービスと連携させることが可能となる。それによって「利用者が圧倒的な利便性を享受でき」「安全・安心な」サービスを「スピーディかつ安価に」実現しやすくなる。

わが国においては2015年頃から、銀行が自身の持つ機能をAPIとして公開し、外部の企業体から利用させることについて、技術的および法令面等を含めた整備が進んできた。ここでいう銀行の機能とは、例えば、残高や入金明細の照会、振込、融資シミュレーション・申込等が考えられる。2017年5月26日に成立した改正銀行法では、APIを用いてFintech企業等と銀行が協働・連携することの法的枠組みが整備された。それによって前述のような利用者が望むサービスを実現することが、まさにこの改正銀行法の立法趣旨と言えらる。

本稿では、銀行APIが登場した背景、銀行APIにより実現が期待されるサービスや社会的なメリットなどについて概観するとともに、日本ユニシス株式会社（以降、日本ユニシス）が提供する「BankVision[®]」のAPI公開を例に「銀行API」の具体的な内容について説明する。2章で銀行APIの成り立ちと現状、3章でBankVisionが提供するAPI、4章で銀行APIの今後とビジネスエコシステムについて述べる。

2. 銀行APIが登場した背景

1995年以降、インターネットの爆発的な普及に伴い、インターネットを介したシステム/サービス/ビジネス間の連携が検討され進められるようになった。

インターネットにはあらゆる業種の企業がフラットに接続されている。そのためどのような相手とどのような取引/連携を行うのかを予め明確にしておくことは難しく、またインターネットに接続された企業同士の連携は基本的にPeer to Peerで行われる。こうした特性から、Webサービス等のインターネットを介したシステム間連携においては、種々の連携で共通的な部分（ここでは共通仕様と呼ぶ）のみを規定し、個々の業務に係る部分は個別の業務に合わせて設計する階層構造が採られた。特に2000年頃からは、ネットワークで接続されたシステム同士を、Webベースの共通仕様（当時はhttp・SOAP・XML等）で連携させる「Webサービス」という概念が一般化した。Webサービスは現在の（Web）APIの原型とも言える。

Webサービスを利用したシステム間連携の試みはすぐに成功したとはいえないが、その後試行錯誤の中で概念的・技術的に進化し、次第にAPIという概念に変わっていった。APIによ

でシステム/サービス間の連携が加速することで、ビジネスの高度化が促され、従来は考えられなかったような新しい便利なサービスも登場してきた。本章では、APIの特徴と、銀行APIの現状について述べる。

2.1 ビジネス観点から見た API による連携の価値

ビジネスの観点から見た API による連携の特徴（価値）は、次の2点である。

1) API を利用する（呼び出す）側の視点

API を利用することで、API で提供される機能を自身（利用する側）のサービスに取り込むことができる。通常は何らかの機能を自社で準備するには投資と時間が必要だが、それらなしでも、外部のアセットを活用してスピーディにサービスを提供できる。

2) API を提供する（呼び出される）側の視点

自社のサービスを、API を利用する側のサービスの中に組み込んで提供し、そこから収益を得ることができる。自社が元々保有している機能を API 化して他社に提供することによって、それが新たな収益源となる可能性が生じる。

著名な例として、ここでは Uber を挙げる。Uber は自動車配車をビジネスとしている企業であり、タクシードライバーだけでなく一般人の自動車も配車する等のビジネスモデルのユニークさで注目されてきたが、API の活用においても先駆的である。

まず Uber は自社ビジネスに必要なシステム機能の多くを、自社で構築せず、他社が提供する API を呼び出すことによって実現している。具体的には、配車料金の決済に利用する「決済」の機能、顧客からの連絡を受ける「通話/SMS」の機能など、様々な分野で API を利用している。API の利用によって、こうした機能を自社で準備することなく、コアビジネスに集中できる体制を作り出している。

また Uber 自身も、配車機能を API として提供している。この配車 API を呼び出す企業は、自社のサービスの中に Uber の配車機能を取り込んで、自社のサービスの一環として提供することができる。一般的に配車機能を提供するには設備投資・人件費はじめ多額の投資が必要だが、API があればそれを呼び出すだけで、設備投資等は不要で Uber の機能が手に入る。

また Uber は特に宣伝をしなくとも、API を提供するだけで、その API を利用する他社が宣伝費用を払って集客をしてくれることになり、そこから収益を上げることも可能となる。

このように API を用いて、サービスとサービス、ビジネスとビジネスを連携することによって、新たなサービス・価値を創出し、共存共栄のビジネスモデルを描くことができる。こうして生まれる新たな経済圏が API エコノミーである。

2.2 銀行機能を活用する Fintech サービスの登場と問題点

わが国では、いわゆる Fintech 企業が 2012 年頃から相次いで設立され、新しいサービスを提供して急成長してきた。新しいサービスの代表例として、PFM (Personal Financial Management) サービスや、クラウド会計サービス等が挙げられる。こうしたサービスを提供する Fintech 企業は、銀行が保有するデータを入手し、それを自社のデータと組み合わせ、自社のサービスとしてユーザに提供するビジネスを行っている。

PFM サービスは、顧客（エンドユーザ）が、自身が様々な金融機関等に保有する金融資産の情報を一元的に管理できるようにするサービスである。PFM サービスの例を図1に示す。

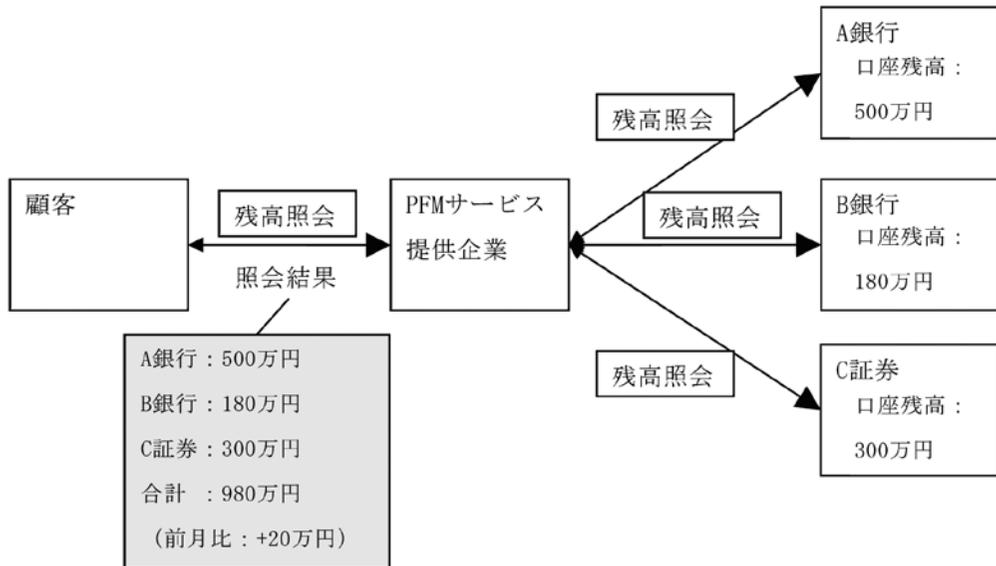


図1 一般的な PFM サービスの提供スキーム

図1の例では、顧客はA銀行・B銀行・C証券の3金融機関に口座を保有している。

PFM サービスが顧客に代わって、各金融機関（A銀行・B銀行・C証券）が保持している顧客口座の情報を入手し、その合計や前月などからの変動を計算してくれる。顧客からすると、これまでは各行に保有する資産は、個別にしか照会できなかった。そのため、複数の金融機関にわたり存在する資産の合計や変動を正確に把握することには手間がかかったが、PFM サービスを利用することで、きわめて簡単に正確かつ有効な資産管理が可能となる。資産を複数の金融機関に保有する顧客にとって、非常に利用価値の高いサービスであると言える。

このようなサービスを実現するには、前提として、PFM サービスが顧客に代わって金融機関から顧客の口座情報を入手する必要がある。しかし従来は、顧客自身ではない第三者（ここでは PFM サービス）が、銀行から口座情報を入手するための機能/インターフェース（つまり API）は存在しなかった。そのため Fintech 企業は、銀行が提供する機能/データを、いわゆるスクレイピングと呼ばれる手法によって取得・利用してきた。Fintech 企業が行うスクレイピングでは、銀行のインターネットバンキング（IB）サービスに、Fintech 企業の提供するサービス（システム）が人間（顧客）になり代わってログオン/アクセスし、得られた Web ページを解析して残高等のデータを入手する。

しかしスクレイピングという手法は、API が提供されないがゆえの苦肉の策であり、種々の問題を抱えている。まず顧客の立場からすると、Fintech サービスを銀行 IB サービスにログオンさせるために、自身のクレデンシャル情報（ユーザ ID/パスワード等）を Fintech サービスに預ける必要がある。顧客としては本来、PFM サービス事業者に対しては、自身の口座への残高照会程度の範囲に限定してアクセスを許可したいところだが、そうした制御はできないため、それ以外の処理（たとえば出金など）を行われるのではないかという懸念が生まれる。

Fintech 企業にとってもこうした懸念はビジネス上マイナスに働く。

銀行としても、Fintech サービスが自行 IB サービスにアクセスしてくることで、IB サービスの提供ベンダに支払う利用料金が高額となったり、相手が Fintech サービスと分からないため、契約関係が成り立たず、ビジネスにならないといった問題があった。だが、仮にこの連携を API で実現し、契約等も含めて整理することができれば、銀行はこの API を新たな収益源として活用することも考えられるようになる。

2.3 わが国における銀行 API の活用に向けた環境整備の状況

こうした状況の中、2015 年頃からはわが国においても、銀行 API の活用について、官民ともに議論が本格化してきた。

2016 年度には全国銀行協会（全銀協）において「オープン API のあり方に関する検討会」が設置され、わが国における銀行 API のあり方について議論が行われた。この検討会からは 2017 年 7 月に検討結果報告が公表され、合わせて個人/法人口座の「残高照会」「入金明細照会」についての電文仕様標準も公表されている。

また、銀行 API の活用に関する内容を含む改正銀行法が、2017 年 5 月 26 日に可決・成立し、銀行 API の利活用についての基礎的な法整備が整った。この改正銀行法における、銀行 API 関連のトピックを、図 2 とそれ以降の本節に記載する。

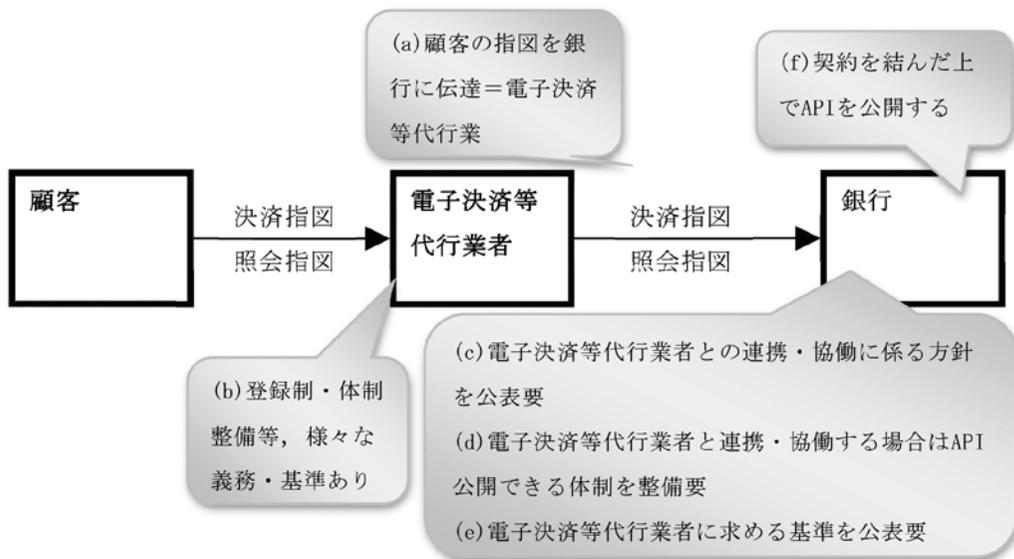


図 2 2017 年改正銀行法における銀行 API 関連のポイント

(a) 次の二つの業務を「電子決済等代行業」と呼ぶ。

- ・銀行顧客（以下、顧客）から自身の口座に関する資金移動取引の指図を受け、それを電子的な方法で銀行に伝達する。
- ・顧客からの指図を受けて、銀行から顧客自身の口座に関する情報を電子的な方法で取得して、その情報を顧客に提供する。

- (b) 電子決済等代行業を営むには内閣総理大臣の登録が必要であり、登録を受けて電子決済等代行業を営む者を、「電子決済等代行業者」と呼ぶ。電子決済等代行業者には、一定の基準が設けられ、顧客情報の保護体制整備等につき、様々な義務が課される。
- (c) 銀行は、電子決済等代行業者との連携・協働に関する方針を、改正銀行法の公布後9か月以内に公表する必要がある。
- (d) 電子決済等代行業者と契約して連携・協働する銀行は、改正銀行法の施行後2年以内に、電子決済等代行業者が顧客から銀行サービス用のユーザID/パスワード等を提供されなくても^{*2} APIを利用してサービス提供できるような体制の整備に努める必要がある。
- (e) 銀行は、契約するにあたって電子決済等代行業者に求める事項の基準を作成し、公表しなければならない。不当に差別的な取り扱いをすることは許されない。
- (f) 銀行は、電子決済等代行業者と契約を結んだうえで、APIを当該業者に対して公開し利用可能とする。

ここで言う「電子決済等代行業者」は、主にFintech企業を念頭においたものである。これまでは電子決済等代行業者のような存在については想定されておらず、法的な位置づけが曖昧であったことが、銀行業界におけるAPI連携が進まない要因の一つであった。今回の改正銀行法によって、電子決済等代行業者自身の法的な位置づけや満たすべき基準、さらには銀行側での対応や電子決済等代行業者との関係性についても、整理され明確化された。これによりわが国においても、銀行APIの活用が大きく進むことが期待される。

2.4 銀行APIの活用に向けた課題

銀行APIの公開を実現するうえで、特に留意しなければならない点として、セキュリティ対策が挙げられる。公開した銀行APIが、銀行や顧客の意図に反して呼ばれてしまった場合、残高等の情報の第三者への漏洩や、APIの種類によっては不正送金などの被害が発生する可能性がある。またAPIは主にインターネットに対して公開することが想定されるため、外部からのインターネットを介した攻撃等の可能性もある。こうした脅威に対処するため、銀行APIの公開においては、次のような点に考慮が必要である。

- 1) 2.3節(a)に記載したような、API実行の指図（認可）を顧客が行わない限り、APIを実行できないよう制御する。
- 2) 2.3節(f)に記載したとおり、銀行と契約を結んだ電子決済等代行業者以外からは、APIを実行できないよう制御する。
- 3) 上記1)の確認において、2.3節(d)に記載したとおり、銀行サービス用のユーザID/パスワード等が電子決済等代行業者には渡らないよう配慮する。
- 4) その他、インターネットへの公開に係る不正取引・侵入等のリスクへの対策を行う。

3. BankVisionのAPI公開

日本ユニシスは、2007年からオープン勘定系システム「BankVision[®]」を提供している。BankVisionには様々な優れた特長があるが、オープン基盤およびオブジェクト指向技術の採用によって、新技術・新機能との親和性が高いこともその一つであり、それによってBankVisionは提供開始後10年の間に、様々な進化を遂げてきた。

こうした特長を活かし、日本ユニシスでは2015年からBankVisionの機能(取引)をAPIとして公開し活用する取り組みを進めてきた。2016年には「BankVision WebAPI 公開サービス」をリリースし、BankVisionの取引をAPI化して他システムとの連携を迅速・容易に実現可能とした。2017年にはこの取り組みをさらに進化させ、いわゆるオープンAPIとしてパブリッククラウド上にBankVisionのAPIを公開し、インターネットを介して電子決済等代行業者から呼び出せる仕組みづくりを行っている。

3.1 BankVision APIの概要

BankVisionのAPI公開(以下、BankVision APIと略す)の概要を、図3に示す。

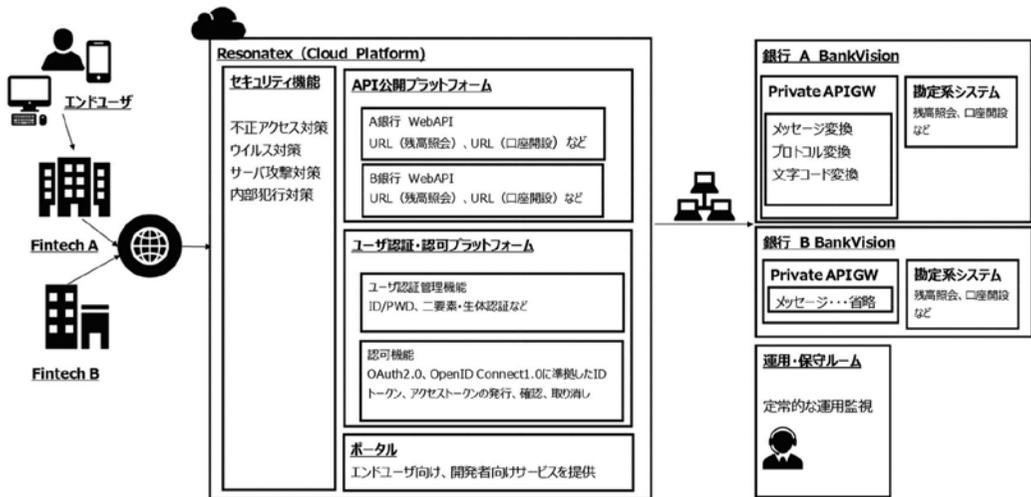


図3 BankVision APIの実現イメージ

BankVision APIは、「BankVision Private API GW」(以降、単に「Private API GW」と記載)と「Resonatex^{TM*3}」(レゾナテックス)の二層で構成されている。

まず Private API GW は、BankVisionの一部として位置づけられ、BankVision 勘定系システムの機能(取引)をAPI化し公開する役割を持っている。2016年に「BankVision WebAPI 公開サービス」として提供開始したのはこの部分である。Private API GWの最も重要な機能は、APIで利用されるプロトコル(http)と、BankVision 勘定系が内部的に利用しているプロトコル(MIDMOST電文)の相互変換である。BankVisionを利用する銀行はこうした変換処理を含むAPIをPrivate API GWの機能によってプログラミングレスで容易に開発し、Private API GW上に自由に公開して他のシステムから呼び出して利用することができる。一方で、いわゆるオープンAPIを実現するためには、2.4節に記載したようなセキュリティ上の考慮が必要であるが、Private API GWにはそうした機能は備わっていない。そのため Private API GWは、原則として閉域ネットワーク内で利用することを想定している。

Resonatexは、Private API GWによって閉域ネットワーク内に公開されたAPIを、インターネットに安全に公開する役割を持つ。Resonatexは閉域ネットワークでPrivate API GWと接続され、インターネットとBankVisionとのセキュリティ境界として機能する。

3.2 BankVision API へのアクセス概要と Resonatex

図4は、エンドユーザ（顧客）から BankVision API に至るアクセスの経路を示したものである。

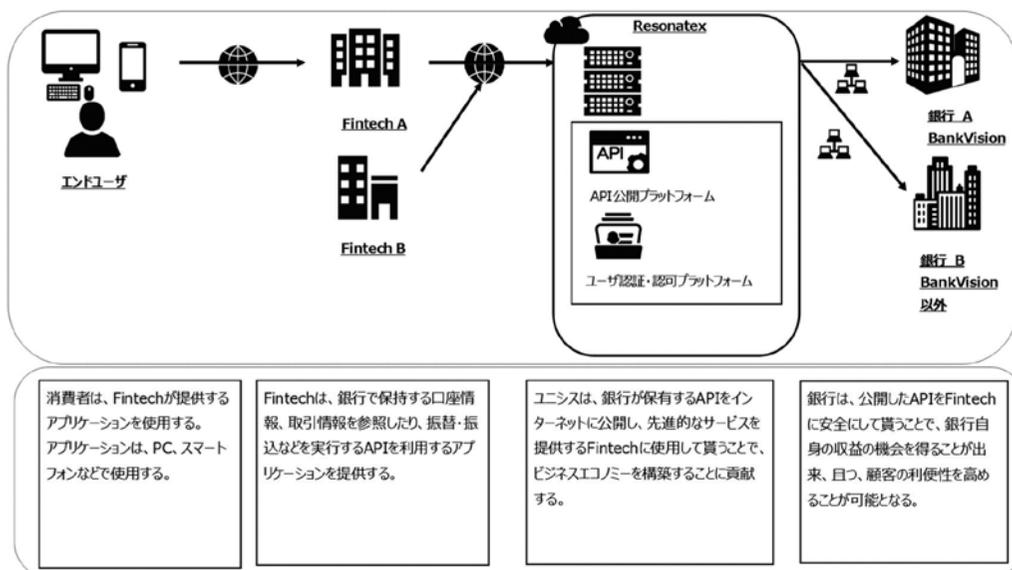


図4 BankVision API へのアクセス概要

Resonatex は、API 公開プラットフォーム、およびユーザ認証・認可プラットフォームからなる。エンドユーザは、Fintech（電子決済等代行業者）アプリケーションを経由して、Resonatex 上に公開された銀行の API にアクセスし、自身の口座情報の参照、振込などを行う。API へのアクセスにおけるセキュリティ上の脅威は、Resonatex において取り除かれたうえで、BankVision（内の Private API GW）に転送される。この際、2.4 節 1) に記載したとおり、API 実行が顧客（エンドユーザ）の指図であることの確認が必要である。そのためにユーザ認証・認可プラットフォームで、エンドユーザの認証およびアクセス認可の処理を行う。合わせて、API 公開プラットフォームにて、2.4 節 2) に記載した、API の呼び出し元が正当な（銀行と契約を結んだ）Fintech 企業であることの確認を行う。

Resonatex の特徴の一つは、Resonatex が複数の API 接続先（Fintech 企業）と複数の API 提供企業（銀行）を繋ぐハブとなっており、API へアクセスするためのエンドユーザのクレデンシャル情報（ID/PWD など）をそこで一元的に管理している点である。

一人のエンドユーザが複数の銀行に口座を保有している場合を考える。この場合一般的には、それぞれの銀行ごとに個別のクレデンシャルが必要となる。ところが Resonatex においては、単一のクレデンシャルのみで複数の銀行口座へのアクセスを認可することも可能であり、エンドユーザの利便性は飛躍的に高まる。

なお、Resonatex は BankVision に特化したものではなく、その他の金融機関やその他業種（小売など）の API 公開においても使用可能である。Resonatex と Private API GW の二層構造とした理由はここにあり、業種の壁を越えたプラットフォームとして利用可能とすることを目指している。

3.3 API 公開における要素技術

BankVision API の API 公開プラットフォームは、WebAPI のアクセス方式としてデファクトスタンダードである REST/JSON に準拠するよう実装している。またユーザ認証・認可プラットフォームは、同様に認可フローとして OAuth2.0 に準拠するよう実装している。本節ではこれらの要素技術について簡単に述べる。

3.3.1 REST/JSON

REST とは、URI として表現されたネットワーク上のリソースに対して、HTTP のメソッド (GET/POST/etc) でアクセスし、必要な情報を取得する形式である。データ表現形式を JSON とする場合、一般的に REST/JSON と呼ばれる。REST/JSON に準拠することにより、API を使用するアプリケーションの開発者が、API の URI を参照するだけで、どのリソース (口座、取引明細など) にアクセスする API かを、容易に理解可能となる。

API を実行するのに必要となる URI および HTTP のメソッドは、API 提供者 (つまり銀行) が自由に定義可能としている。例えば口座情報の照会は、REST の原則に従うと、HTTP/get で以下のような URI に対してアクセスすることが想定される。

`www.XXXXXYYYYY.co.jp/ExampleBank/futsuyokinn/店番/科目/口座番号`

3.3.2 OAuth2.0

図5は、エンドユーザが Fintech 企業に対して、自分の口座情報に BankVision API を介したアクセス権限を認可する仕組みを示したものである。

2.3 節(d)および 2.4 節(3) で述べたとおり、改正銀行法においては、Fintech 企業を介した銀行 API へのアクセスの際に、顧客 (エンドユーザ) が自身のクレデンシャル情報を Fintech 企業に渡さずにすむようにすることが求められている。これを可能とするのが、OAuth2.0 である。Resonatex においては OAuth2.0 に準拠した実装がなされている。図5に示すフローの中でエンドユーザは自身のクレデンシャル情報を Fintech に渡していない。

なお OAuth2.0 は認可のフレームワークを示したものに過ぎず、金融機関向けの API で利用するにあたっては、そのリスクに応じてセキュリティに十分配慮した実装が必要となる。こうした実装仕様については、現時点 (2017 年 7 月) では OpenID Foundation Financial API WG において検討されている状態であり、確定はしていないものの、Resonatex においても確定を見越して実装する予定である。

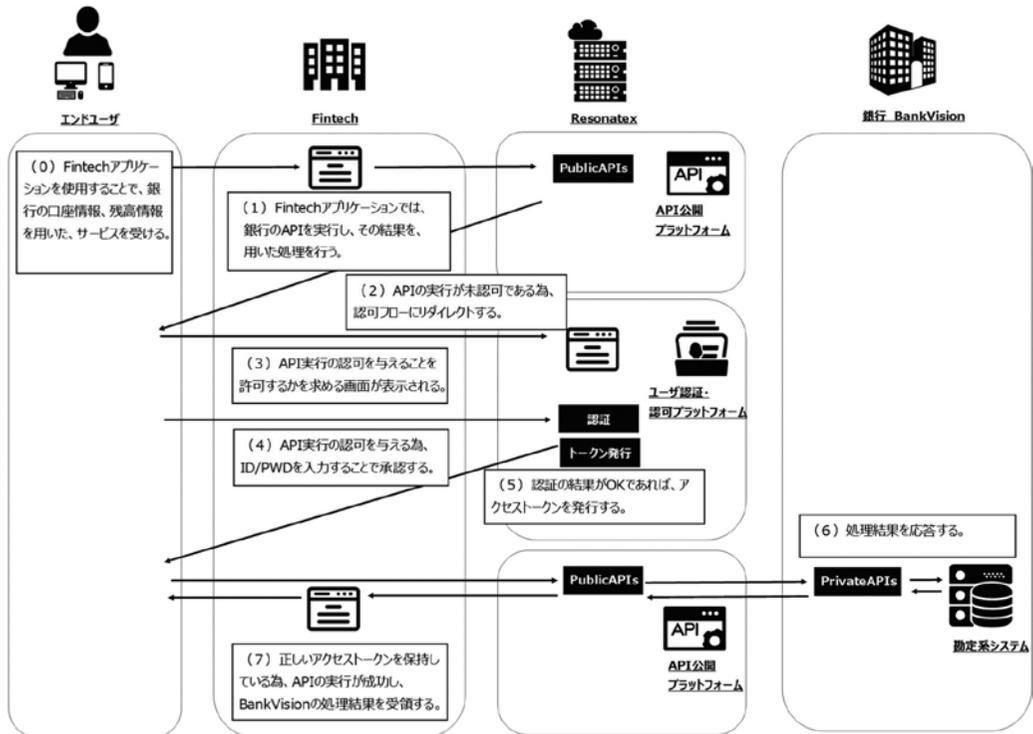


図5 OAuth2.0認可フローの例

4. 銀行 API の今後、ビジネスエコシステムの形成

2017年7月現在、わが国における銀行APIの活用については、ごく限定された範囲に留まっているのが実情である。しかし、これまでに述べた銀行APIに関する機運の高まりや、改正銀行法など法制度面の整備の進展、Bank Vision WebAPI公開サービスなど技術的基盤の整備の進展などにより、今後銀行APIが大きく発展・普及していくことは確実である。本章では銀行APIの発展の方向性について考察する。

4.1 APIを提供する銀行の増加

遅くとも2020年春頃までには、大多数の銀行がAPIを提供する体制を整えようとする。

改正銀行法により銀行は2018年3月までに、電子決済等代行業者との連携・協働の方針を公表する必要がある。ここで大多数の銀行が、電子決済等代行業者と連携・協働していくことを表明するとみられる。連携・協働していくことを表明した銀行は、改正銀行法の施行から2年間の間にAPI提供が可能な体制を整備する必要があり、その期限が2020年春となる。

なお銀行は「電子決済等代行業者と連携・協働しない」という方針を表明し、API提供しないことも可能である。しかしその場合、銀行のビジネスは従来チャネルに留まったものとならざるを得ず、新しいサービスから取り残される危険性が高い。そのため、「連携・協働する」、すなわちAPI提供する方針とする銀行が大多数を占めることになると思われる。

4.2 銀行が提供するAPIの種類が増加・業務範囲の拡大

現状の銀行APIの多くは口座情報等の照会を中心としたものであるが、今後は振込などの

資金移動を伴う取引や、預金・決済以外の領域、たとえば融資（シミュレーション、実行）・諸届・投資信託などにも広がっていくと想定される。

また、各種の金利情報や ATM・支店所在地の情報などを API で提供することも、今後は広がっていくことが予想される。

4.3 API を利用するサービス・企業・業種の増加・拡大

改正銀行法などにおいて、銀行 API を利用する企業として想定されているのは、主に Fintech 企業である。しかし銀行 API として提供される業務の種類が増えるにつれ、金融以外の業種にも、API の利用が広がっていくことが期待される。例えば不動産業や自動車販売業など、高額消費財を扱う業種では、ローンなどの金融商品を扱うことが多く、銀行 API の利用によりサービスの高度化が期待できる。

また足元に目を転じて、銀行内で自行の API を利用して、システム間のリアルタイム連携を実現したり、新しいチャンネル（自行アプリ、インターネットバンキング等）に利用していくことも考えられる。前章に記載したとおり、BankVision WebAPI 公開サービスでは、こうした利用方法も容易に実現でき、既に一部の銀行では本番業務で活用している。

4.4 API の利用方法の高度化

前節までに述べたような銀行 API 活用の拡大が進むにつれて、その利用形態もさらに高度化されていく。例えば全銀協の「オープン API のあり方に関する検討会」などでも、「連鎖」という考え方が議論に上っている。これは、銀行が提供する API を、Fintech 企業が自社の機能と組み合わせる上で、新しい別の API として、更に別の企業に提供する、というものである。逆に銀行から他の企業が公開する API を呼び出し利用することも、今後は考えられる。

5. おわりに

これまで述べたように、今後はわが国においても銀行 API の普及・活用が確実に進んでいく。銀行という高度なセキュリティ管理を求められる業界において API 活用が進むということは、社会的にセキュアな API 利用に関するモデルケースが整備されるということであり、他の業界も含めてさらに API 活用が進むことにつながっていくと考える。

それによって各企業・団体は、自社の持つ価値あるアセットを API として提供し、業界を超えて他社のアセットと組み合わせることで、新しいサービス・ビジネスが出現し、そこに新たなエコシステムが生まれることになるだろう。

BankVision API は、本格的な API エコノミー時代に向けた始まりに過ぎない。3.2 節で述べたとおり、Resonatex は、BankVision だけでなく様々な銀行・金融機関、さらには業界の壁を超えて他業種でも利用可能なプラットフォームであり、日本ユニシスが目指す新たなエコシステムを生むための触媒となるものである。日本ユニシスでは、Resonatex と BankVision API を、これまでになかった新しいサービス・ビジネスを実現するために、今後も発展させていく予定である。

* 1 オープン API という言葉には、いくつかの意味合いがあるが、本稿ではこのように定義する。

- * 2 2017年7月時点でこれを実現する代表的な方式が、RFC6749で規定されるOAuth2.0である。
- * 3 企業同士のAPI接続を実現させることによって、革新的なサービス・事業の創造を提供する仕組み。

- 参考文献** [1] 金融庁, 国会提出法案(第193回国会)銀行法等の一部を改正する法律(平成29年3月3日提出, 平成29年5月26日成立), 2017年5月
- [2] オープンAPIのあり方に関する検討会, オープンAPIのあり方に関する検討会報告書—オープン・イノベーションの活性化に向けて—, 全国銀行協会, 2017年7月

執筆者紹介 大野上 総 (Kazusa Ohno)

1999年日本ユニシス(株)入社。金融機関向けミドルウェア「MIDMOST」や、世界初のオープン勘定系システム「BankVision[®]」の開発など、金融機関向けのインフラ領域を担当。現在、BankVIision[®] WebAPI公開サービスの企画・開発を担当。



田島 太朗 (Taro Tajima)

2003年日本ユニシス(株)入社。無店舗金融機関向けの勘定系パッケージ「BankStar[®]」の開発・適用・保守を担当。2013年から「BankVIision[®]」関連の開発に携わる。現在、BankVIision[®] WebAPI公開サービスの企画・開発を担当。

