PLAID

Financial data access methods:

Creating a balanced approach

OCTOBER / 2016

Introduction

The financial technology landscape continues to advance rapidly. Recent years have seen a proliferation of new firms, applications, and uses — ranging from peer-to-peer payments to automated investing to alternative lending. Consumers and small businesses are reaping the benefits of these new technologies. Yet broadly speaking, mass adoption of fintech is still to come, and there may be a unique window to influence the design of critical technologies underpinning the digital financial services ecosystem.

Many financial services products that improve people's financial lives rely on a user's ability to grant access to, or "permission," personal financial data. For example, micro-saving platform Digit relies on transactional information to automate savings — making it easy to grow a nest egg. Budgeting app Level Money uses financial data to let people know how much they can afford to spend on discretionary expenses while meeting their saving goals.

Data access is the backbone of financial innovation

In these and countless other cases, data access is the backbone of financial innovation; without it, progress in this space might not be possible. But enabling this access is not simple. As the digital financial services ecosystem continues to take shape, data access has raised questions around technology, interoperability, security, compliance, regulations, and user experience.

Engagement on this issue is critical: Secure data access can improve the wellbeing of any consumer or business. Conversely, any reduction in access might restrain consumers from a promising financial frontier. Progress requires partnership across the industry.

This whitepaper examines several approaches that attempt to address the data access challenge in the United States. These include OAuth, legacy screen scraping, modern scraping — more accurately described as Screenless Data Collection (SDC), Open Financial Exchange (OFX), and Durable Data API (DDA). Each of these methods attempts to tackle somewhat different objectives; however, they all merit careful examination as the financial services ecosystem mulls its future.

Interoperability and flexibility are two recurring themes in this examination. Interoperability for developers is essential to scale any connected solution; for example, a developer may struggle to build inclusive products if certain financial institutions restrict data availability and format — or if forced to directly integrate with the existing infrastructure of thousands of different financial institutions. Flexibility empowers financial institutions to make this data available in a way that minimizes technical impact and cost — without putting customers at risk. But interoperability for developers and flexibility for institutions can conflict. Today, trusted intermediaries often play a balancing role — providing technologies that enable interoperability for developers while accommodating the preferences of, and minimizing the demands on, financial institutions.

Distinguishing between

Authentication and data transmission

The exchange of financial data comprises two major components: 1) Authentication (i.e., how data access is permissioned), and 2) Data transmission (i.e., what data is made available, and how it is formatted and transferred). These are separate and distinct concepts, and most financial data specifications primarily address only one of these two.

Authentication refers to any permissioning flow wherein a user allows an app to access their data that resides elsewhere. It is not specific to financial services; for example, Facebook enables authentication for third parties (SEE FIGURE 1). Twitter uses a similar mechanism (i.e., "Authorize Login with Twitter to use your account? This app will be able to read Tweets from your timeline.").

FIGURE 1
EXAMPLE AUTHENTICATION FLOW

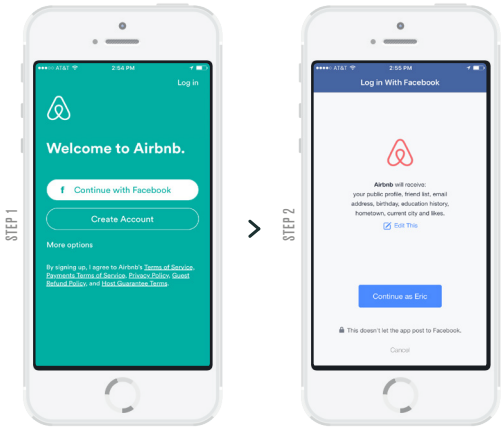

FIGURE 1) *Although commonly associated with bank accounts, authentication refers to any type of permissioning flow; it is not specific to financial services. In this example, Airbnb performs authentication via Facebook.*

To bolster security, authentication flows in financial services and other industries have evolved such that they often include two-factor authentication (2FA). 2FA requires a user to validate her account with two separate types of information. In practice, this might entail verifying a username and password and confirming receipt of an email or text message.

FIGURE 2
COMMON DATA ACCESS METHODS

	AUTHENTICATION	DATA TRANSMISSION
OAuth	✓	
Screen Scraping		✓
OFX	*Suggests OAuth*	✓
DDA	*Suggests OAuth*	✓

FIGURE 2) *Well known U.S. data access methods have different objectives. An access method does not need to address both authentication and data transmission; these are separate and distinct components of financial data exchange.*

Data transmission is distinct from authentication. Data transmission itself can encompass multiple elements; among them are availability, format, and transfer. Data availability refers to the specific fields (such as "payment amount" or "zip code") that are offered through a data specification. Format describes how data is structured — JSON or XML, for instance. Finally, data transfer protocol refers to how data is moved. Screen scraping, for example, is strictly a means of data collection; it does not implicate a mechanism for authentication (SEE FIGURE 2).

A strong financial data exchange needs both secure authentication and efficient data transmission. But treating these components as separate and distinct can be helpful in supporting flexibility for financial institutions.

A strong financial data exchange needs both authentication and data transmission. But treating these components as separate and distinct can be helpful

Authentication

LEGACY METHODS

Early methods of permissioning financial data access, which started in the 1990s, may have been advanced at the time but would not be considered so today. Users typically provided bank login credentials directly to an app, which then retained credentials for future access. This approach was palatable in the cases of large platforms with significant security resources. But consumers then began adopting an increasing array of apps — including offerings from smaller, less well-vetted developers, who also held credentials. This led to a proliferation of user credentials stored by developers, which created additional security threat vectors.

CONTEMPORARY APPROACHES

Today, trusted third-party intermediaries have developed authentication methods that significantly improve upon legacy practices.

In one example, a trusted intermediary makes it possible for a user to permission access to personal financial data from within an app. The user gives the intermediary permission to provide the app a unique token that delivers data access; use of the token means that the developer does not see or handle user credentials.

Moreover, the intermediary can identify itself to the financial institution so that the institution knows which third party is contacting its server on behalf of the user. In this way, the intermediary supports interoperability so that developers can focus on innovating — while also protecting the institution's (and user's) security interests.

The approach described above has several advantages over legacy methods. From a security perspective, the app does not see or handle credentials, mitigating security risk. From a user's perspective, the consistent, in-app experience reduces conversion barriers, making it preferable to alternatives such as OAuth. From an institution's perspective, this approach grants visibility into the identity of the third party accessing data.

TOKENIZATION

Tokenization is the substitution of a non-sensitive data element for a sensitive element. In the realm of financial data access, a bank or trusted intermediary might swap user credentials for a unique "token" — typically a random character string — that is provided to a fintech app (e.g., Digit or Level Money). The app then uses the token to access that specific customer's data. Use of this token eliminates the need for apps to store and access sensitive information like credentials. Only the bank or trusted intermediary can access the "token vault" mapping the token to the sensitive data element. The token is therefore useless if compromised.

OAUTH

Overview

OAuth is a popular standard for tokenized authentication. The roots of OAuth grew out of Twitter, and a first version of the standard was published by an industry group in 2007. OAuth 2.0 was released in 2012 to improve OAuth flows, particularly for mobile apps. Beyond Twitter, other technology companies — including PayPal and

▸

- Facebook — were early adopters. More recently, traditional financial institutions have shown interest, although no major U.S. bank has fully implemented OAuth at the time of this writing.

Considerations

Older, legacy authentication approaches shared user credentials with both intermediaries and apps. Current authentication best practices share credentials with trusted intermediaries, but not with apps (see above section on contemporary approaches to authentication). OAuth takes this practice one step further, sharing credentials with neither trusted intermediaries nor apps. Limiting credentials to financial institutions provides a security benefit. But relatively speaking, this improvement is modest, because trusted intermediaries already restrict credentials from the thousands of small apps that would present a greater risk for compromise. Because existing authentication approaches already eliminate most systemic security risk, the incremental security benefit of OAuth for financial services is less than sometimes portrayed.

The incremental security benefit of OAuth for financial services is less than sometimes portrayed

Additionally, implementing and maintaining OAuth can be expensive for institutions, requiring substantial management, product, technical, and design resources. By contrast, currently available authentication methods may compel an institution to vet the trusted intermediary upfront, but require minimal ongoing cost or effort by the institution thereafter. Many institutions may find it more practical to mitigate risk through robust security audits of trusted intermediaries than to build an OAuth solution.

As a final consideration, the standard OAuth implementation flow would redirect the user from a third-party app to the institution's website or app — then return the user back to the third-party app. Redirects like this may deteriorate the user experience, creating new barriers for consumers to hurdle to benefit from innovative financial services. A particularly cumbersome

INCENTIVES FOR CREATING A WORLD-CLASS CUSTOMER ONBOARDING EXPERIENCE

On the surface, banks might appear to have little incentive to create a seamless onboarding flow for customers to link their accounts to third-party apps. Easing access to alternative tools could logically cause consumers to spend less time directly with their bank. On the basis of this thinking, some have argued that the security benefits of OAuth are a red herring put forth by banks in an attempt to wrest greater control of apps' onboarding experience.

But growing evidence indicates that banks may view engagement with apps as an opportunity, not a threat. Results from a Plaid survey of U.S. fintech users suggest that better enabling third parties could in fact strengthen banks' position as the hub of most consumers' financial lives. For instance, 80 percent of consumers expressed a preference for banks to help them manage their financial apps. More broadly, consumers are increasingly asking their banks to make it easier to connect to apps.

Data transmission

▷ flow may even compromise principles of interoperability, challenging developers' ability to engage would-be users with apps.

An emerging alternative — called Screenless Exchange — embraces much of the OAuth standard, while addressing many of the shortcomings with OAuth redirect flows. Screenless Exchange is an authentication flow that contains a bank's permissioning interface (i.e., "Enter your user name and password") entirely within an app — supported in the background by a trusted intermediary, such as Plaid. User credentials are seen by neither the app nor the intermediary, replicating a key security advantage of redirect-based OAuth. Additionally, the consistent, in-app user experience may improve conversion.

Summary

OAuth is already a popular authentication standard among leading consumer apps — and has parallels with enterprise identity management tools such as Okta and OneLogin. As banks increasingly contemplate OAuth, they can choose the tokenization solution — including OAuth derivatives such as Screenless Exchange — that best works for them. Trusted intermediaries offer one possible mechanism to balance institutions' need for customized security against developers' need for a turnkey integration and streamlined user experience.

Scraping is interoperable with any number of security protocols

SCREEN SCRAPING

Overview

Perhaps no other topic under the data access umbrella has created as much buzz as screen scraping. Scraping refers to the practice of automatically collecting permissioned data from a website. Although sometimes maligned in the press and elsewhere for supposed security risks, these claims are misleading: Scraping is itself distinct from authentication. Instead, it is merely a means to move permissioned consumer data from one place to another using existing infrastructure. Scraping is interoperable with any number of security protocols. Moreover, it is the only currently available tool to enable a fully inclusive fintech ecosystem.

Scraping is not a new technology, and leading practitioners have made substantial technical improvements over time. Related approaches, such as web crawling — which usually indexes an entire website, rather than transferring details of a specific page — have existed for nearly as long as the consumer-facing internet. None other than Google is arguably the world's largest web crawler, utilizing the technology to build its search engine in service of its two billion users. Over the past decade, scraping has powered the growth of major sites in real estate, travel, social media, and retail.

In financial services, early software providers like Quicken allowed users to import scraped data for narrow needs such as bill payment and accounting. But as the number of scrapers, users, and data elements at each institution increased, banks began to report heavy and sometimes unpredictable traffic patterns.

▸

- This, in turn, catalyzed meaningful advances in scraping practices; the modern approach is aptly described as Screenless Data Collection (SDC). Leading practitioners now minimize the amount of data they pull, and a traditional webpage is not rendered. Rather, the SDC practitioner queries only raw text without complementary formatting information, images, advertisements, or other data that has little value and high latency. In fact, the load imposed on bank servers by efficient SDC can be less than that of an API-based data access specification, such as DDA.

Moreover, greater ongoing partnership between leading third parties and financial institutions has helped improve the practice of SDC. For instance, trusted intermediaries will typically use a header or whitelisted IP address to identify their traffic to a bank's server. This allows the bank to verify that the traffic is legitimate. Additionally, trusted intermediaries can collaborate with banks to appropriately manage traffic to servers.

Considerations

A fair assessment of screen scraping should distinguish between legacy and advanced practices; this section considers the newer practice of SDC.

Most importantly, SDC works: Timely information is accurately gathered and exported, and consumers directly benefit from accessing data in powerful new platforms. Banks can enable consumer data access while leveraging existing technology at minimal additional cost. Crucially, SDC promotes an inclusive financial infrastructure: Smaller banks and credit unions that cannot afford to provide a custom data interface or API can still enable their customers — who increasingly want to use digital financial products — to permission access.

The technology's perceived drawbacks are well documented. As with other methods, accuracy and uptime are imperfect, and traffic can affect banks' server loads. Third parties that perform SDC often must manage against unpredictable changes in bank infrastructure. And, despite improvement, institutions may prefer greater visibility into this process.

Summary

Scraping is a crucial technology underpinning the digitization of financial services and the internet more broadly. Eliminating it is neither practical nor likely to happen any time soon. Most U.S. financial institutions lack resources to develop alternatives, such as open APIs for third parties. Fintech developers do not have the resources to integrate directly with the byzantine landscape of institutions. Thus, SDC — when performed by a trusted, sophisticated intermediary — evens the playing field for small developers and small financial institutions alike, allowing them to fully participate in the ecosystem. Efforts to eliminate screen scraping in financial services would disproportionately harm small app developers and consumers — especially those who choose to bank at smaller financial institutions.

Screenless Data Collection evens the playing field for small developers and small financial institutions alike

OFX (OPEN FINANCIAL EXCHANGE)

Overview

At its core, OFX is a data transmission specification. OFX was intended to ease transfer of financial data among institutions and large software companies. It now includes provisions related to authentication, recommending OAuth. But the core specification can in fact be combined with any number of authentication variants.

OFX is somewhat unique in that it relies on dedicated servers at financial institutions to transport data; permissioned third parties can directly connect to OFX servers to query data. The specification has been adopted by more than 7,000 financial institutions.

History

OFX 1.0 was developed in 1997 by Intuit, Microsoft, and Checkfree (now part of Fiserv). It gained traction slowly, but by 2005 OFX adoption exceeded 3,000 institutions.

The specification evolved incrementally over its first 10 years, but major change came in 2006, when the financial industry introduced 2FA and the specification moved to support it. Version 2.1.1 marked a major attempt to expand beyond a data specification by including an authentication layer.

From there, development went dormant until 2015, when an OFX consortium relaunched—this time with a broader set of industry representatives, including newer fintech firms. This awakening was largely a response to the introduction of the specification for DDA. The most recent version of OFX, released in 2016, preserved most of the existing OFX data specification, but added new data fields and introduced the concept of OAuth tokenization.

Now, the OFX consortium is confronting existential questions about its future, such as whether to develop a new in-house version of the specification or align more closely with another data access effort.

Considerations

True to its roots, transferring data remains OFX's key strength.

But as an end-to-end specification, the latest OFX version leaves room for improvement. First, the specification remains highly prescriptive—some might say clunky—at nearly 700 pages. Despite this specificity, OFX does not support the exchange of several new types of financial data, which developers need to innovate. For example, OFX is not equipped to accommodate extensive identity information (for fraud prevention) or certain complex account types. OFX also relies on an older architecture and data format, making it cumbersome to evolve the specification (such as to widen data availability). Nimbleness is necessary for any modern financial data specification, as innovation often demands quick changes to meet new consumer and developer needs.

Second, OFX has become increasingly specific about authentication, conflating this with data transmission. For instance, OFX accommodates OAuth, recommending a redirect flow to authenticate users. This specificity may not work for all institutions, some of which may choose to prioritize a native user experience.

Summary

Despite its shortcomings, OFX 2.2 has so far seen traction with a handful of institutions. That said, some institutions have taken a piecemeal approach to implementation, such as by limiting the available data fields. Customized

implementations mean that OFX in practice is not a truly consistent, interoperable data transmission standard.

Absent a significant expansion of data availability and a more flexible approach to authentication, it seems unlikely that OFX 2.2 as written can begin to approach industry-wide adoption

DDA (DURABLE DATA API)

Overview

DDA is a recently developed data specification that is lightweight and uses modern formats. DDA also calls for authentication through OAuth, though this recommendation appears to be less of a focus.

Customized implementations mean that OFX in practice is not a truly consistent, interoperable data transmission standard

History

An industry working group from the Financial Services Sharing and Information Sharing and Analysis Center (FS-ISAC) released DDA in May 2015. This working group comprised several financial institutions as well as a small number of financial data third parties. DDA was intended to improve data exchange relative to OFX. And in its short life, DDA has garnered substantial interest from stakeholders, including at least one major bank that is actively working to adopt it.

Considerations

With regard to its objective, DDA has several advantages over OFX. In general, it is a modern, concise, and more flexible specification that places greater emphasis on developers' needs. For example, DDA allows for standard, modern data formats; by contrast, OFX uses a custom, legacy format.

Where it is feasible to implement, DDA also has marginal advantages over Screenless Data Collection. DDA delivers more control to banks regarding who accesses what data on their systems. Additionally, DDA helps ensure that the data delivered is accurate.

At the same time, DDA could refine two areas of the specification in its next iteration. First, DDA is largely designed for a specific need: personal financial management (PFM). Indeed, several of DDA's leading proponents are PFM providers that rely primarily on transactional data. Focus on a single use threatens to exclude thousands of developers creating apps for other core consumer and small business financial services, such as direct deposit and bill pay. Rigid data availability would complicate future efforts to evolve DDA to support innovation.

Second, DDA currently embraces a narrow definition of the OAuth flow rather than giving institutions flexibility to choose the authentication method that works for them. More broadly, DDA could soften its focus on authentication, and instead commit to excelling as a data standard.

Summary

In all, DDA is a welcome contemporary specification for data access. It can be improved, particularly by broadening potential uses beyond PFM and softening its focus on authentication. But on the whole, DDA enables

- relatively wide data availability, is lighter than OFX, and uses modern architecture and formats.

OTHER APPROACHES

The United States has seen other efforts to create a data access standard. These include TxPUSH, spearheaded by a single third-party in 2015, as well as the Interactive Financial eXchange Forum, which has broader representation. Neither of these efforts has gained significant traction, nor are they suited to address the priorities of flexibility and interoperability raised in this whitepaper.

Conclusion

There are many approaches to data access in the United States today. Each has slightly different objectives; none seems likely to be a silver bullet.

Yet careful consideration of these approaches, which underpin the entire digital financial services ecosystem, is essential to continued innovation in financial services. Without robust, secure access to personal financial data, progress in this space is sure to stall — to the detriment of consumers everywhere.

It is critical to strike a balance between flexibility and control for institutions, on the one hand, and interoperability for developers on the other. This balance is likely only with collaboration — potentially including participation by trusted intermediaries. And collaboration requires partnership.

Together, financial institutions, trusted intermediaries, and developers can build an ecosystem that promotes secure data access and empowers both consumers and the organizations that serve them.

FOR MORE
INFORMATION, VISIT
– **plaid.com**

QUESTIONS
– **info@plaid.com**

Glossary of terms

— **2FA** Two-factor authentication. A security protocol that requires a user to validate her account by providing two of three distinct types of information (something the user knows, something the user possesses, something the user is)

— **AUTHENTICATION**[1] The act of verifying a user's ability to access an account. Along with data transmission, it is one of two major components of any secure financial data exchange

— **DATA TRANSMISSION** Elements of a financial data exchange unrelated to authentication; includes data availability, format, and transfer protocol

— **DDA** Durable Data Application Programming Interface (API); a modern data access specification first released by FS-ISAC in 2015

— **FS-ISAC** Financial Services Information Sharing and Analysis Center; financial services industry forum originally established in 1999 for threat analysis and sharing

— **HEADER** An identifier inserted in a browser's HTTP request that allows a financial institution to know the identity of a third party contacting its server

— **JSON** JavaScript Object Notation; a modern open standard format to transfer data

— **NATIVE USER EXPERIENCE** A process of linking an app to another party, like a financial institution, that occurs entirely within the app (i.e., natively)

— **OAUTH** An open standard to enable authorization popular among technology companies; has gained recent attention for potential financial services applications

— **OFX** Open Financial Exchange; a data access specification originally created by Intuit, Microsoft, and Checkfree in 1997

— **REDIRECT** A step within the process of linking an app to a financial institution that shifts the user from the third-party app to another interface, such as the financial institution's website or app

— **SCREEN SCRAPING (LEGACY METHODS)** Traditionally, the automated copying of all data, including visual, from a rendered consumer-facing webpage

— **SCREENLESS DATA COLLECTION** A set of advanced data collection methods designed for speed, accuracy, and nominal server impact; only minimal text data is queried, and a screen is never rendered

— **SCREENLESS EXCHANGE** A tokenized authentication flow derived from the OAuth standard that allows a user to remain within a native app experience while linking an account, rather than relying on a redirect

— **TOKENIZATION** Substitution of a non-sensitive data element (e.g., random character string) for a sensitive data element (e.g., bank account routing number)

— **WHITELISTING** The practice of a third party identifying to a financial institution the specific IP addresses it may use to contact that institution's servers, which the institution in turn commits to permit

— **XML** Extensible Markup Language; an open standard format to transfer data first published in 1996

[1] Technically, authentication is the act of verifying one's ability to access an account, whereas authorization involves granting permission to a third party to access information (such as transaction history) contained in that account. Because these steps are often, though not always, sequential, for simplicity we refer to them together as "authentication."