API Best Practices

Leveraging APIs to enable innovation & promote growth

PYGG

This report was written by Ben Lyon of Pygg Enterprises LLC (the "contractor") for The Bill & Melinda Gates Foundation (the "client"). All information herein shall be treated as confidential and is intended solely for the client.

Table of Contents

Executive Summary

It's 2016. I need an API...

We're in the midst of a Cambrian Explosion in software innovation. To quote Marc Andreessen, the co-founder of Netscape, "software is eating the world."

Why? Because the Internet is disrupting every industry, from home appliances to transportation to financial services. And beneath the surface of many of the disruptive applications we use everyday is a growing web of APIs – application program interfaces that enable applications to "talk" to one another. These APIs enable you to do everything from pay for a coffee with your watch to interact with your bank account via WeChat.

According to Apigee, enterprises that offer APIs typically see a 300% increase in traffic. Some enterprises, like Salesforce, realize 50% of total revenue from APIs. So, it's 2016. I need an API...

"The push toward API enablement won't wait. Within the next 24 months, Mindtree expects significant uptake in APIs in financial services."

Mindtree Limited

Table of Contents

An Intro to APIs

What is an API?

An Application Program Interface, or API, enables one application to "talk" to another, whether to consume data, initiate actions, or both.

APIs fall within one of three categories:

1. Internal – APIs for internal use only. For example, a bank might create an internal API in order to launch a mobile application.
2. Partner – APIs intended for a select set of pre-authorized partners (aggregators, service providers, etc.).
3. Open – APIs designed for general use by application providers.

The evolution of APIs

SOA ➡ Web API

The modern "Web API" evolved from Service Oriented Architecture (SOA), which was originally designed for internal, server-to-server applications. For example, a bank might have launched an SOA program to enable easy, cost-effective communication between its own applications.

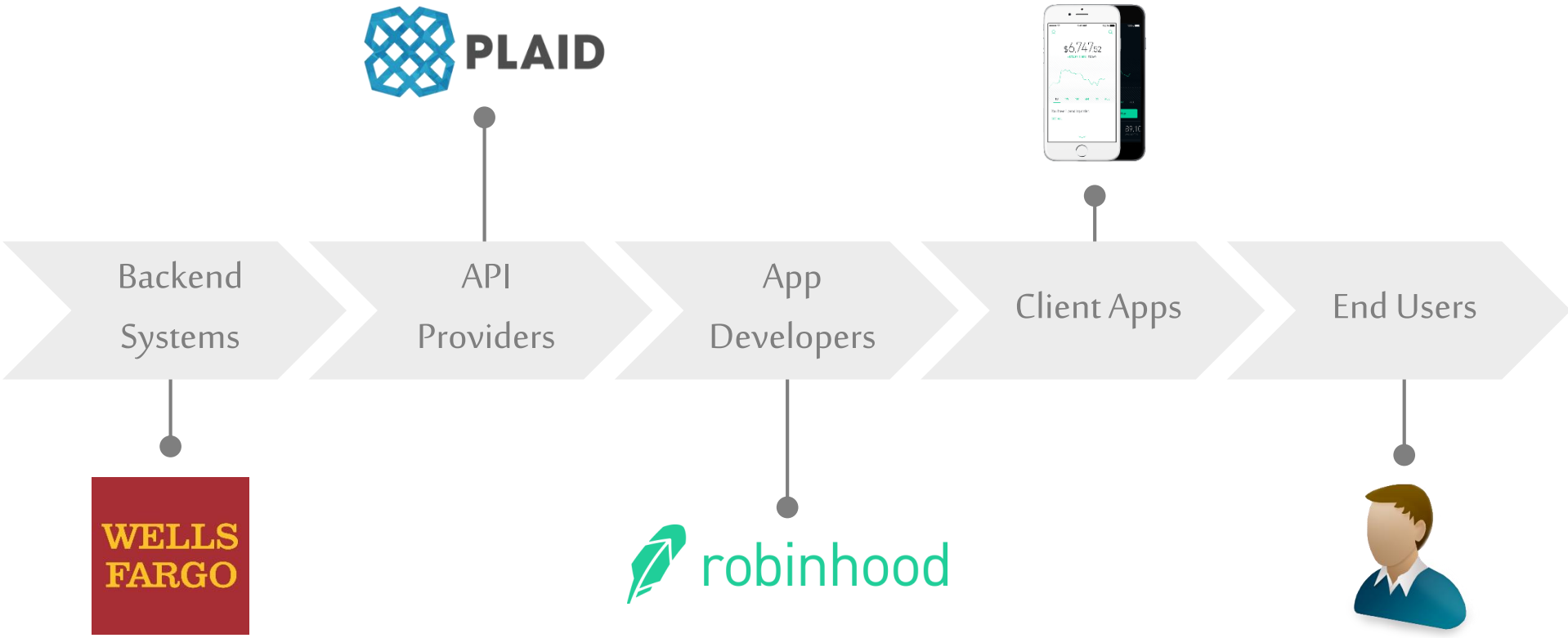
Today, Web APIs have taken the SOA framework and adapted it to a world in which mobile and web applications communicate via the Internet and need the ability to interface with other applications, both internally and externally.

A New Framework

	SOA	APIs
Integration Goal	Internal or to partners	External, often to customers
Project Driver	IT costs	Business revenues
Interface Consumer	Enterprise architects	Application developers

Chart adapted from CA Technologies (ca.com)

The API value chain



The API value chain

In the previous slide, we illustrated the API Value Chain that enables an end user to deposit funds to their Robinhood account in order to buy stocks. Here is how each participant in the value chain contributed to making that possible:

Backend Systems	The end user has a current account at Wells Fargo Bank.
API Providers	Plaid enables 3rd parties to access end user bank accounts.
App Developers	Robinhood has access to multiple banks via Plaid Connect.
Client Applications	Robinhood asks the end user to connect their bank account.
End Users	End user gives Robinhood permission to connect to Wells Fargo.

Ideal roles and responsibilities

The overarching goal across the API value chain should be to provide a consistent, convenient, and secure End User experience. To enable this, the ideal roles and responsibilities of each party are as follows:

Backend Systems	Responsible for maintaining redundant, stable platform. Should focus on ensuring platform uptime and notifying "downstream partners" (API Providers) in advance of any changes.
API Providers	Responsible for building and maintaining a stable interface with backend systems and exposing them to Application Developers via a modern, lightweight and well-documented API.
App Developers	Responsible for building and maintaining integration with API Provider(s) to ensure a functional Client Application. May also contribute software development kits (e.g. iOS) to API Providers.
Client Applications	Responsible for providing seamless End User experience. Should be configured to automatically send crash reports to the App Developer in order to facilitate the speedy resolution of issues.
End Users	Although the End User is not "responsible" for maintaining the API value chain, they should be enabled to report bugs, errors and feedback within the Client Application.

An Intro to APIs

Why do APIs matter?

Put simply, a financial service provider that doesn't offer APIs is leaving money on the table. APIs open a backend system to countless application developers at global scale and, when implemented correctly, contribute to significant revenue growth.

Take Salesforce, for example. Salesforce became famous for its "No Software" slogan, which captured its mission to replace traditional software with an open suite of web-based services. Today, the name "Salesforce" is virtually synonymous with the "Software-as-a-Service" movement and APIs contribute 50% of the company's total revenue.

In the context of financial services, APIs should be viewed as a low-cost, scalable channel for acquiring new customers, serving existing customers and enhancing existing services.

"Open APIs play an active, strategic and crucial role in our infrastructure because in addition to our own retail and business banking products, we also offer 'no-stack banking' to non-banks, retailers or challenger banks."

Matthias Kröner, Fidor Bank

An Intro to APIs

What APIs can we build?

There are tens — if not hundreds — of financial service API types in use today. These APIs provide services as simple as confirming an individual's date of birth and as complex as augmenting account information with meta-data.

In the following slides, we will see 30 financial service API types in use around the world. Although the majority of these APIs are only available in the West, they should give us a sense for what is possible in emerging markets.

In India, for example, many of the API types in the following slide are being enabled via "The India Stack," a government-sponsored initiative to offer open APIs as a public good.

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services	Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data	Offers	Pending Payments	Process Chargebacks	Refund Payments	Remittances
Reverse Payments	Rewards	Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services	Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data Augment individual or business data with available meta-data (e.g. Yelp reviews, social media accounts, etc.).		Pending Payments	Process Chargebacks	Refund Payments	Remittances
		Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services	Investment Holdings	Knowledge-based Authentication	Make Payments
 www.plaid.com		Pending Payments	Process Chargebacks	Refund Payments	Remittances
		Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services Location of bank branches and ATMs, daily foreign exchange rates, interest rates by product offering, etc.		Knowledge-based Authentication	Make Payments
Meta-Data	Offers			Refund Payments	Remittances
Reverse Payments	Rewards	Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	 MasterCard www.mastercard.com		Knowledge-based Authentication	Make Payments
Meta-Data	Offers			Refund Payments	Remittances
Reverse Payments	Rewards	Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	**Business Characteristics** Incorporation date, employer ID number, merchant category code, ISO compliance certificates, production processes, etc.		Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification			Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data	Offers	Pending Payments	Process Chargebacks	Refund Payments	Remittances
Reverse Payments	Rewards	Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments			Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification			Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data	Offers	Pending Payments	Process Chargebacks	Refund Payments	Remittances
Reverse Payments	Rewards	Signature Verification	Taxpayer ID	Transfer Funds	Watch List Referencing

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services	Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data	Offers	Pending Payments	Process Chargebacks	**Watch List Referencing** Referencing businesses or individuals against known watch lists, sanctions lists, etc. for AML, CFT and KYC compliance.	
Reverse Payments	Rewards	Signature Verification	Taxpayer ID		

Sample financial service APIs

Accept Payments	Account Verification	Address Verification	Balance Inquiry	Balance Verification	Bill Payments
Bulk Payments	Business Characteristics	Cash Withdrawal	Contact Verification	Date of Birth Verification	Deposit Funds
Identity Verification	Income Verification	Information Services	Investment Holdings	Knowledge-based Authentication	Make Payments
Meta-Data	Offers	Pending Payments	Process Chargebacks	 www.experian.com	
Reverse Payments	Rewards	Signature Verification	Taxpayer ID		

Broader API categories

For ease of reference, we might group financial service APIs into the following broad categories: Compliance, Information, Infrastructure and Value-Added Services.

Category	Description	Example API Types
Compliance	Services that enable 3rd parties to verify business and individual identities, detect anomalies, etc.	Blockscore Business and People Verification, Experian Business IQ and Precise ID, etc.
Information	Information services such as ATM and bank branch location, foreign exchange rates, interest rates, etc.	MasterCard Merchant Identifier, PrivatBank Information API, Xignite, etc.
Infrastructure	The finance "rails." Services that extend access to bank charters, licenses, payments processing, etc.	Braintree, Dwolla, Plaid, solarisBank, Stripe, etc.
Value-Added Services	Services that augment financial service data with additional information and/or increase End User utility.	Plaid Meta-Data, Capital One Offers/Rewards, Alternative credit scoring, etc.

An Intro to APIs

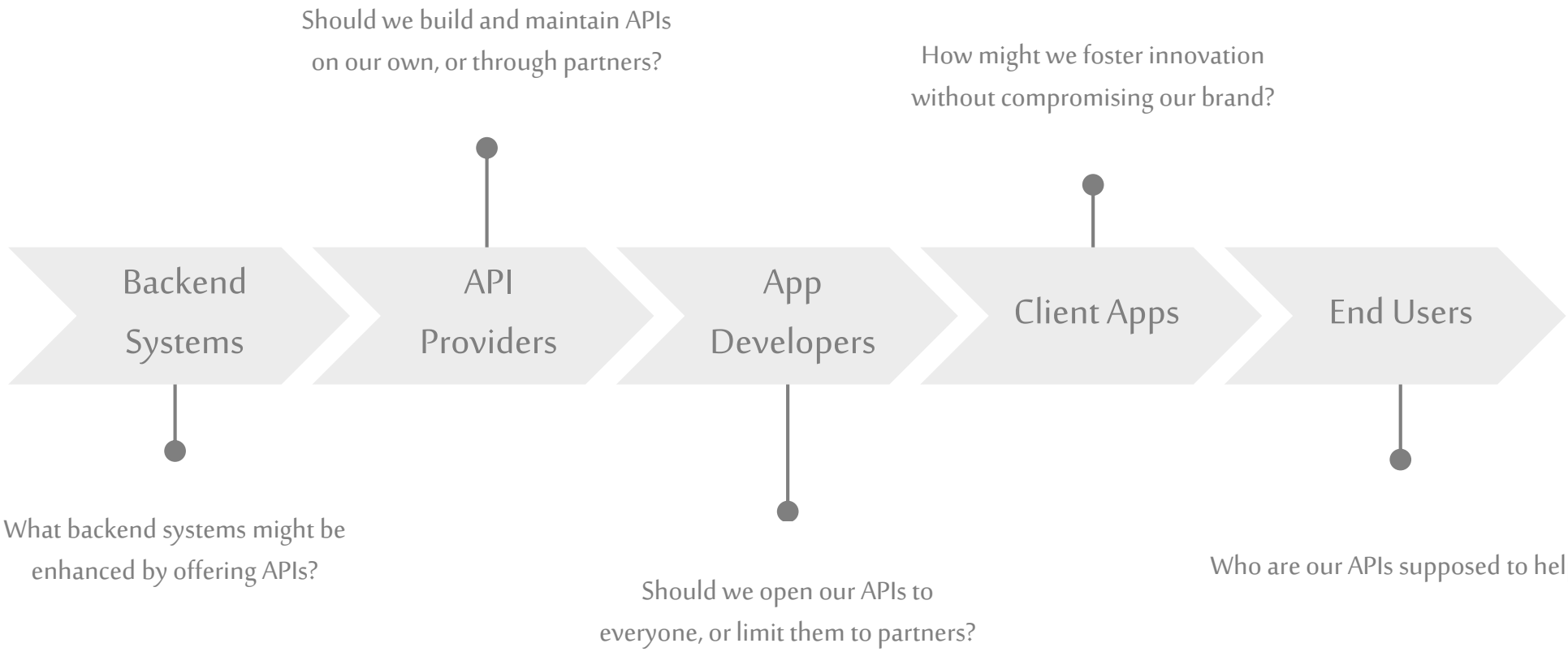
Defining an API strategy

Successful APIs are not products; they are dynamic services driven by dedicated teams. As such, building a successful API requires a robust API strategy.

Assuming we want to build an API, we first have to answer the question: What do we hope to achieve? For example:

- I want to build an Internal API to enable unified services that utilize data from multiple business units.
- I want to build a Partner API to enable trusted partners to launch services and build developer communities on my behalf.
- I want to build an Open API to enable application developers worldwide to create client applications using my backend systems.

Then we have to ask…



Example API strategies

Most financial service providers already offer Internal or Partner APIs. That said, it's worth noting that the Open API framework is becoming commonplace – and in some cases mandated – across the industry.

Strategy Type	Example
Internal API	Banks often engage their core banking system providers in order to build internal APIs to facilitate mobile- and web-based services (e.g. mobile banking, online banking, etc.). These APIs are not shared outside of the organization, but are rather intended to connect disparate applications and databases internally.
Partner API	In Kenya, Safaricom partnered with Commercial Bank of Africa (CBA) to launch the M-Shwari service. The partnership entailed giving CBA limited access to Safaricom user data via an API in order to create an underwriting model for issuing loans.
Open API	In Germany, solarisBank launched its "Banking-as-a-Platform" service to "power an array of fintech services." solarisBank offers Open APIs for banking, payments and add-on services and sees 3rd party application developers as a strategic channel for acquiring new customers, enhancing service offerings for existing customers and launching new services. Their model might be called the "Amazon Web Services" of banking.

If you don't build it, someone will…

As a word of caution, startups have been "hacking" de facto APIs together for years, often without the permission of providers. This begs the question: If I don't launch my own APIs, will I lose control?

Sample "Hacks"	Description	Potential Risks
Method Hooking	Techniques used to alter or augment the behavior of an operating system or application by intercepting function calls, messages, or events. See www.teller.io.	Negative End User experience if parameters change, resulting in broken application. Unsupervised 3rd party access to sensitive data. Susceptible to malicious rootkit exploit.
Optical Character Recognition (OCR)	Using OCR technology to parse information from structured documents, such as invoices, statements, etc. See www.canopy.sg.	Negative End User experience if document structure changes, resulting in broken application. Unsupervised 3rd party access to sensitive data.
Screen Scraping	Using "robots" to parse the HTML of a website or application, giving 3rd parties both read- and write-level access to the application. See www.yodlee.com.	Negative End User experience if HTML changes, resulting in broken application. Unsupervised 3rd party access to sensitive data. Potential to appear as "denial of service" attack.

An Intro to APIs

From strategy to action...

Ultimately, we should be able to identify the API strategy that best positions us to achieve our goal(s). Once identified, then we have to align our business units, resources and key performance indicators to support that strategy.

Any API strategy will require dedicated resources and deliberate effort. Building a developer community takes time and energy; it doesn't happen overnight. That said, we have to set realistic expectations in terms of how quickly we will see a return on investment.

Focus, Patience and Evangelism – these will be essential.

“This isn’t a one-month project where a bank develops a set of APIs and then calls it quits — it is a long-term journey.”

Chae An, IBM Financial Services

Table of Contents

What developers expect…

Vint Cerf, the co-founder of the Internet Protocol, describes the Internet, not as a web of routers and servers that transmit packets of binary information, but as a "design philosophy."

His point is not to diminish the physical infrastructure of the Internet, but rather to highlight that the Internet itself is only possible because of a set of shared principles and standards.

Today, these shared principles and standards are well-enshrined within the developer community, and developers expect applications and APIs to adhere to them. In the following slides, we will review some of the key shared principles and standards that are specific to financial service APIs.

Modern, lightweight and secure

Expectation	Description	Examples
RESTful API	The modern, lightweight alternative to SOAP. REST enables faster results and simple transactions.	Used by BBVA, Capital One, Dwolla, Experian, Fidor Bank, Open Bank Project, Square, etc.
JSON Format	The lightweight alternative to XML. Now considered the "gold standard" format by developers.	Used by BBVA, Capital One, Dwolla, Fidor Bank, Plaid, Simplify Commerce, Stripe, Square, etc.
Webhooks	User-defined HTTP callbacks made by HTTP(S) POST. Facilitates real-time consumption of information.	Used by Blockscore, Capital One, Dwolla, Plaid, Simplify Commerce, Stripe, Square, etc.
HTTP Verbs	A standardized method for interacting with an application using common commands (e.g. GET, POST).	Used by virtually every FinTech startup.
TLS 1.2	Secure alternative to SSL 2.0. TLS 1.2 prevents 3rd parties from eavesdropping/tampering with messages.	Used by virtually every FinTech startup.
OAuth 2.0	OAuth 2.0 is the next evolution of the OAuth protocol. It facilitates secure authentication flows.	Used by virtually every FinTech startup.

Modern, lightweight and secure

Expectation	Description	Examples
HTML <iframe>	An inline frame used to securely embed another HTML document within the current HTML document. Especially important for keeping sensitive data (e.g. card details) off the client application or server.	Dwolla.js, Kontomatik "Signin Widget," Plaid Link, Stripe.js, Yodlee Fastlink, etc.
Swagger Description Language	The most popular API design framework. A simple, powerful representation of RESTful API. Enables interactive documentation and client SDKs generation.	Capital One, Dwolla, Stripe, Square, etc.

stripe

"The new version of Stripe.js meets [PCI DSS] criteria by performing all transmission of sensitive cardholder data within an iframe served off of a stripe.com domain controlled by Stripe."

Supporting Ecosystem

Expectation	Description	Examples
Developer Portal	A single interface through which developers can read API documentation, get API keys, access a sandbox, etc.	dashboard.plaid.com, dashboard.stripe.com, etc.
Sandbox	A testing environment that isolates untested code from the production environment or repository.	uat.dwolla.com, Blockscore Test (sk_test), etc.
Status Page	A page that lists API statuses (i.e. uptime) in real-time and documents known changes and issues.	status.dwolla.com, status.stripe.com, etc.
Software SDKs	"Helper libraries" that help developers create applications in different programming languages.	Java (including Android), iOS, PHP, Python, C#, Ruby, Node.js, Go, .NET, Perl, etc.
How-To Guides	Documents that provide step-by-step instructions to new developers, tailored to specific use cases.	Plaid Quickstart, Stripe Quick Start Guides, etc.
Community	Events, forums and shared repositories that promote collaboration. Driven by "API Evangelist(s)."	Developer Forums, GitHub Repositories, Hackathons, etc.

Global standards

Expectation	Description	Examples
ISO 8601	The ISO standard for date and time formats.	Date: 2016-03-11
ISO 3166	The ISO standard for country codes.	United States (US), Australia (AU), etc.
ISO 4217	The ISO standard for currency codes.	US Dollar (USD), Australian Dollar (AUD), etc.
ISO 20022	The "Universal financial industry message scheme."	Used by SWIFT, Visa, etc.
ISO 8583	The ISO for card-originated interchange messages.	Most ATM transactions utilize ISO 8583.
UTF-8	Character encoding standard. Covers virtually every character, punctuation mark, and symbol in the world.	The dominant character encoding standard for the World Wide Web.

"APIs must be simple to succeed. Banks must put themselves in the mind of the developer who is going to build an app and has just three days to do it."

Stephane Dubois, Xigite

Table of Contents

Business Models

Choosing the right model

The right API Strategy is nothing without the right business model. As discussed earlier, building and maintaining an API requires a dedicated team and deliberate effort, entailing real costs. In turn, we have to identify a business model that justifies our upfront and ongoing investment, but also enables application developers to build sustainable businesses using our APIs.

There is no "silver bullet" in this regard. Instead, we should start by envisioning "the world as it ought to be" (to quote Visa's founder, Dee Hock) and agreeing to a set of basic principles. For example: We should price our APIs in a way that is inviting to application developers, but also rewards us – and them – when the client applications they develop succeed.

As a foundational principle, there should be little-to-no upfront investment and volume-based pricing tiers.

Common monetization strategies

Model	Description	Examples
Transaction Fees	Charge a fee for every transaction. This model is best suited when processing payments.	Authorize.net, Braintree, Stripe, etc.
API Call Fees	Retroactively bill application developers for every API call their application makes to your API.	Plaid offers various pricing models, including a “per usage” fee.
Support Fees	Charge a monthly (sometimes tiered) support fee for providing customer support to application developers.	Dwolla charges users between $250 - $1,500 per month for “Priority Support.”
Licensing Fees	Charge application developers a monthly or yearly licensing fee for accessing your APIs.	Yodlee offers tiered monthly and annual pricing models based on usage.
Revenue Sharing	Charge application developers a percentage of revenue generated via their client applications.	Commonly used by mobile network operators (e.g. MTN, Safaricom, etc.).

"Learn to think differently and don't do APIs the same way as everything before. Banks must begin thinking as technology companies."

Megan Minich, Silicon Valley Bank

Table of Contents

Contact

Ben Lyon

hi@benlyon.com

+1.206.294.9519