



OpenID[®] Foundation Financial API WG

<http://openid.net/wg/fapi/>

June 2017

Nat Sakimura

Chairman of the Board, OpenID Foundation
Research Fellow, Nomura Research Institute

Anoop Saxena

FAPI WG co-chair, OpenID Foundation
Architect, Intuit

- OpenID[®] is a registered trademark of OpenID Foundation.
- *Unless otherwise noted, all the photos and vector images are licensed by GraphicStocks.



Do you use Personal Finance Software?

What are the current problems?

When NRI started screen scraping in 2001, we thought it will be a temporally solution.

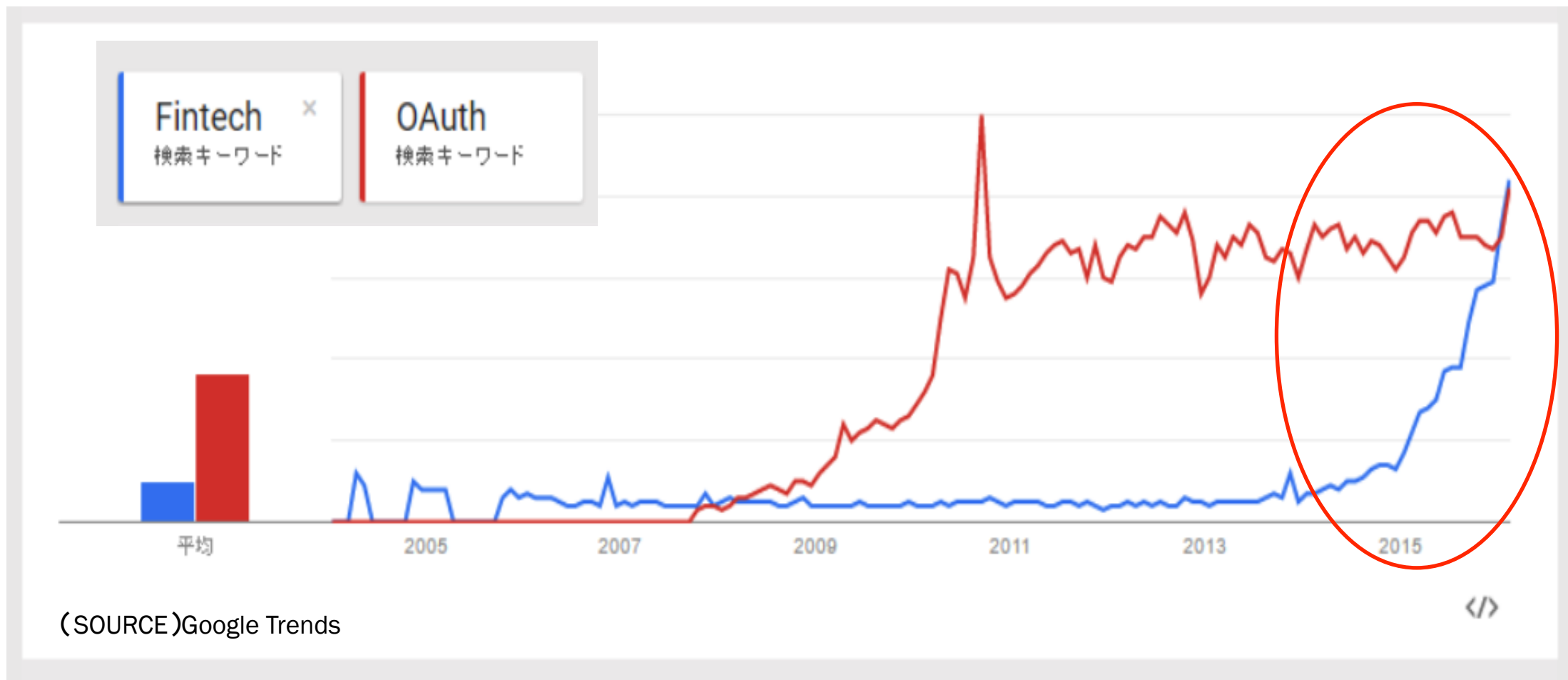
“There was OFX, and SAML was coming. SOAP was gaining momentum.
We should be able to get out of scraping business in a few years time!”

WRONG!

After 15 years, we are still screen scraping.

The situation is changing though.

Fintech is gaining a lot of interest lately

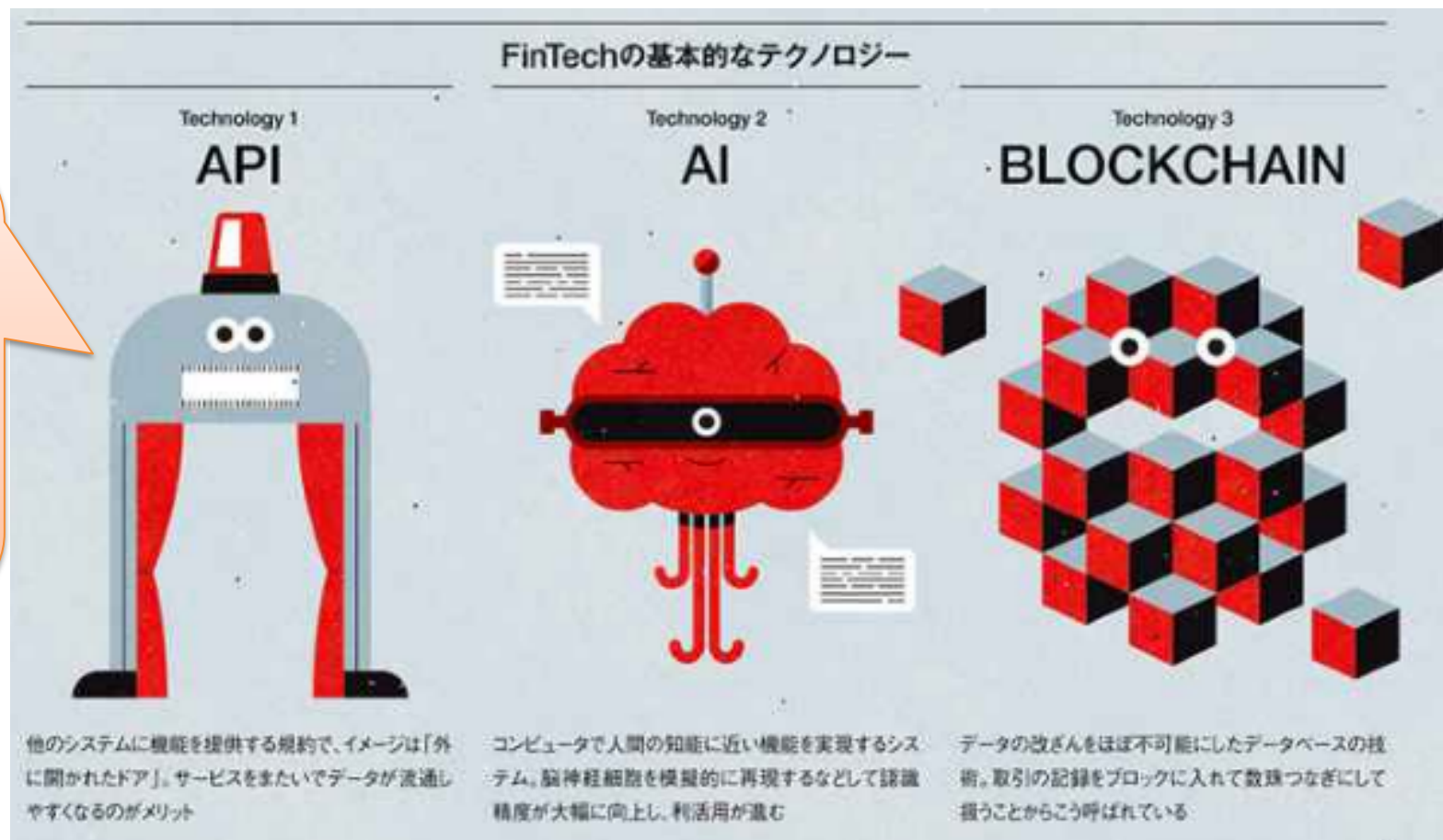


API is known to be one of the three main component of FinTech

Use cases for Identity Federation
API in Financial sector

- 1.Account Opening (incl. KYC)
- 2.Personal Asset Managment
- 3.Payment, Sending Money
- 4.Loan Application
- 5.AI assisted portfolio management

(Source)Nikkei BP: FinTech Yearbook



(Source) Nikkei BP: Fintech Revolution P.4

INDUSTRY PUSH > US: FS-ISAC Durable Data API

■ JSON , XML + OAuth 2.0

1. Introduction	13. Entities	13.38. Statement Entity	14.28. MessageFormat
2. Ability Criteria	13.1. Account Entity	13.39. Statements Entity	14.29. MutualFundType
2.1. Performance	13.2. AccountDescriptor Entity	13.40. StockSecurity Entity	14.30. Number
2.2. Scalability	13.3. AccountDescriptorList Entity	13.41. SwapSecurity Entity	14.31. OptionType
2.3. Interoperability	13.4. Accounts Entity	13.42. TaxLot Entity	14.32. OrderDuration
2.4. Extensibility	13.5. AccountsDetailsRequest Entity	13.43. TelephoneNumber Entity	14.33. OrderType
2.5. Security	13.6. Availability Entity	13.44. Transaction Entity	14.34. PaymentFrequency
2.6. Reliability	13.7. Capability Entity	13.45. Transactions Entity	14.35. PositionType
3. Design Principles	13.8. Contribution Entity	13.46. Transfer Entity	14.36. Secured
4. Deployment	13.9. Currency Entity	13.47. TransferStatus Entity	14.37. SecurityIdType
5. Message Transport	13.10. Customer Entity	13.48. Vesting Entity	14.38. SecurityType
6. Service Delivery Expectations	13.11. CustomerName Entity	14. Simple Types	14.39. StockType
7. Message Syntax	13.12. DebtSecurity Entity	14.1. AccountStatus	14.40. String10
8. Security	13.13. DeliveryAddress Entity	14.2. AccountType	14.41. String2
8.1. Model	13.14. DepositAccount Entity	14.3. AssetClass	14.42. String255
8.2. Client Authentication	13.15. DepositTransaction Entity	14.4. BalanceType	14.43. String3
8.3. Token Scope	13.16. Error Entity	14.5. Boolean	14.44. String64
9. Logical Data Model	13.17. FIAttribute Entity	14.6. CallType	14.45. String9
9.1. Entity Identity	13.18. FIPortion Entity	14.7. CompoundingPeriod	14.46. SubAccountType
9.2. Surrogate Identity	13.19. Holding Entity	14.8. CouponMaturityFrequency	14.47. TelephoneNumberType
10. Residual Data	13.20. InvestmentAccount Entity	14.9. DebitCreditMemo	14.48. Timestamp
11. Protocol	13.21. InvestmentBalance Entity	14.10. DebtClass	14.49. TransactionReason
11.1. Headers	13.22. InvestmentLoan Entity	14.11. DebtType	14.50. TransactionStatus
11.2. Errors	13.23. InvestmentTransaction Entity	14.12. DeliveryAddressType	14.51. TransferStatusStatus
12. Resources	13.24. LinelItem Entity	14.13. DepositTransactionType	14.52. UnitType
12.1. POST /account	13.25. LoanAccount Entity	14.14. HeldInAccount	
12.2. POST /account/statement	13.26. LoanTransaction Entity	14.15. HoldingSubType	
12.3. POST /account/statements	13.27. LocAccount Entity	14.16. HoldingType	
12.4. POST /account/transaction/image	13.28. LocTransaction Entity	14.17. Identifier	
12.5. POST /account/transactions	13.29. MutualFundSecurity Entity	14.18. IncomeType	
12.6. GET /account/list	13.30. OpenOrder Entity	14.19. InterestRateType	
12.7. GET /accounts/details	13.31. OptionSecurity Entity	14.20. Inv401kSourceType	
12.8. POST /accounts/details	13.32. OtherSecurity Entity	14.21. InvestmentBalanceType	
12.9. GET /availability	13.33. PaymentDetails Entity	14.22. InvestmentTransactionType	
12.10. GET /capability	13.34. PlannedAvailability Entity	14.23. Iso3166CountryCode	
12.11. GET /customer	13.35. Portion Entity	14.24. Iso4217Code	
12.12. POST /transfer	13.36. SingleAccountDetailsRequest Entity	14.25. LoanPaymentFrequency	
12.13. POST /transfer/status	13.37. SingleAccountDetailsRequestList Entity	14.26. LoanTransactionType	
		14.27. LocTransactionType	

(Source) FS-ISAC FSDDA WG

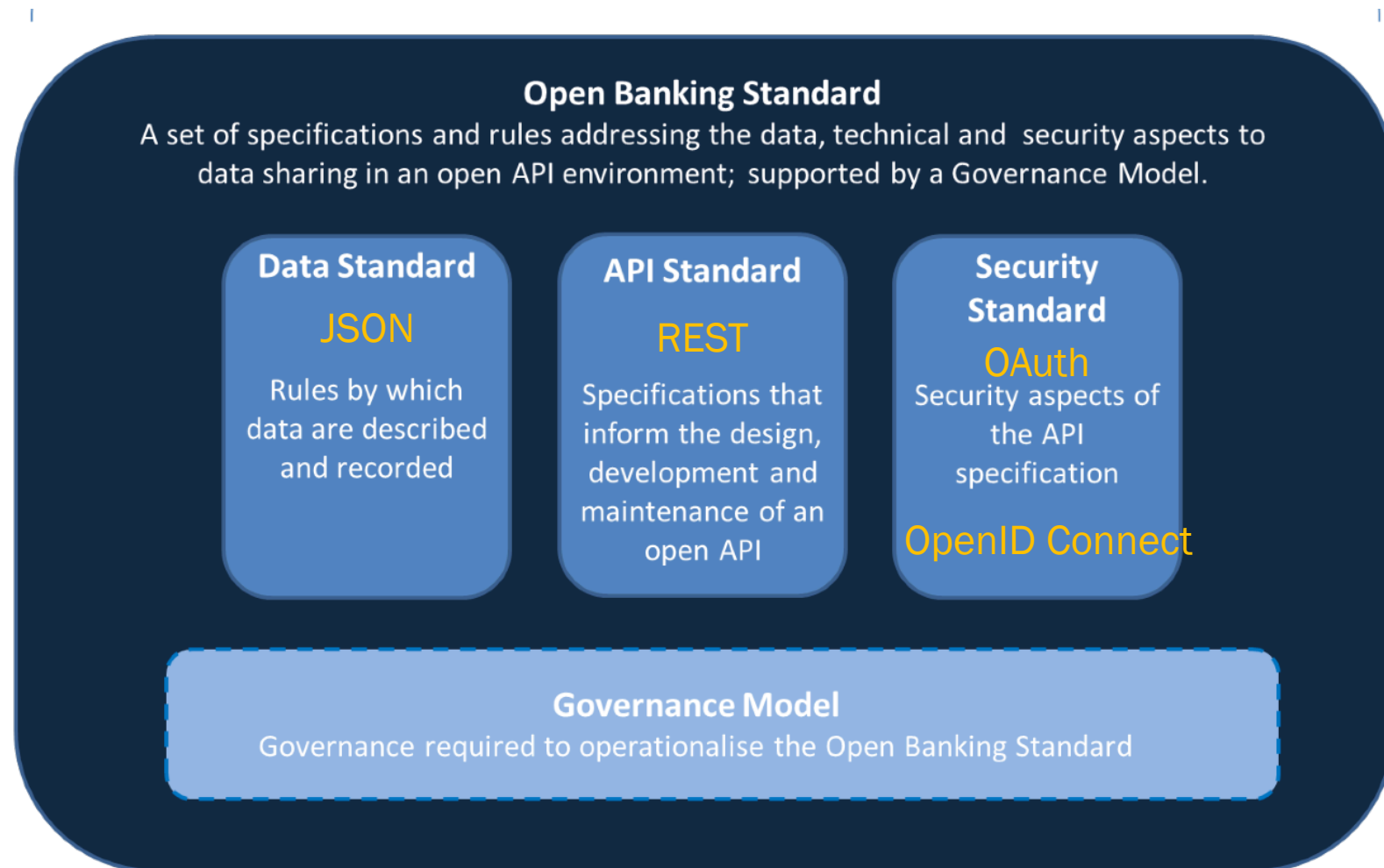
Durable Data API

Financial Services Durable Data API Working Group

Version 1.0
May 2015

REGULATORY PUSH>

EU Payment Service Directive 2 mandates API availability by the end of 2017.



(SOURCE) ODI OBWG: The Open Banking Standard (2016)

“LEGO Model” provided by APIs creates a new customer segment “B-to-D”



Laurens Hamerlinck

Innovation Manager
ABN AMRO Bank

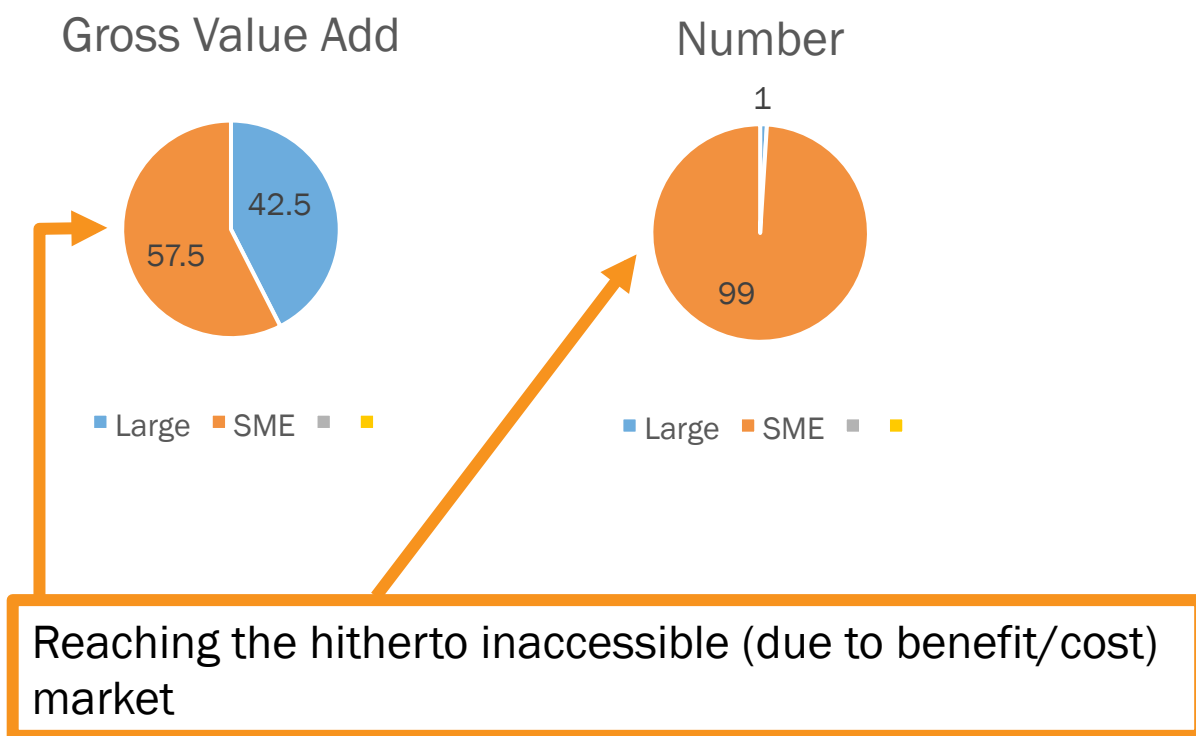
@lhamerlinck

- Open Banking APIs are drawing fintech companies to UK.
- API creates Lego model. You do not need to build everything yourself.
- OEaS:=Our Expertise as a Service
- Financial sector becomes more Open. Not only in EU. Also in US and elsewhere.
 - iOS app platforms did not have any developers in the beginning but see what happened by opening up the ecosystem.
 - What happened to a company who did not open it?
- B-to-D: API = New Customer Segment.

『Bank as a Platform: Exploring a new Role in the Age of Technology』(European Identity Summit 2017 講演より)

Automation through API makes it possible for financial institutions to provide services to hitherto unreachable segment.

Enterprise category	Persons employed	Turnover or	Balance sheet total
Medium	< 250	≤ € 50 m	≤ € 43 m
Small	< 50	≤ € 10 m	≤ € 10 m
Micro	< 10	≤ € 2 m	≤ € 2 m



(source) Eurostat:

[Number of enterprises, persons employed and gross value added \(GVA\) and the share of SMEs, 2012.](#)

- Operational loan provided to small business.
 - Banks providing operational loans to a small business through automated credit clearance based on the ledger data using Artificial Intelligence.
- Transaction insurance offered to SME.
 - Transaction insurance has only been offered to large enterprises due to the low insurance rate compared to the cost of the evaluation of the deals.
 - With APIs, the cost is significantly driven down and now it can be offered to SMEs.

In the era of “Mobile First”, OAuth is an obvious choice for API protection but ...



Saying “use [#oauth](#)” does not solve the problem.

-- Mark O'Neill, Gartner

@APIDays Paris 2016

(SOURCE) Photo taken by Nat Sakimura @APIDays on 13th Dec. 2016

Because OAuth 2.0 is a framework as the name indicates

[\[Docs\]](#) [\[txt|pdf\]](#) [\[draft-ietf-oauth-v2\]](#) [\[Diff1\]](#) [\[Diff2\]](#) [\[IPR\]](#) [\[Errata\]](#)

PROPOSED STANDARD

Errata Exist

Internet Engineering Task Force (IETF)

Request for Comments: 6749

Obsoletes: [5849](#)

Category: Standards Track

ISSN: 2070-1721

D. Hardt, Ed.

Microsoft

October 2012

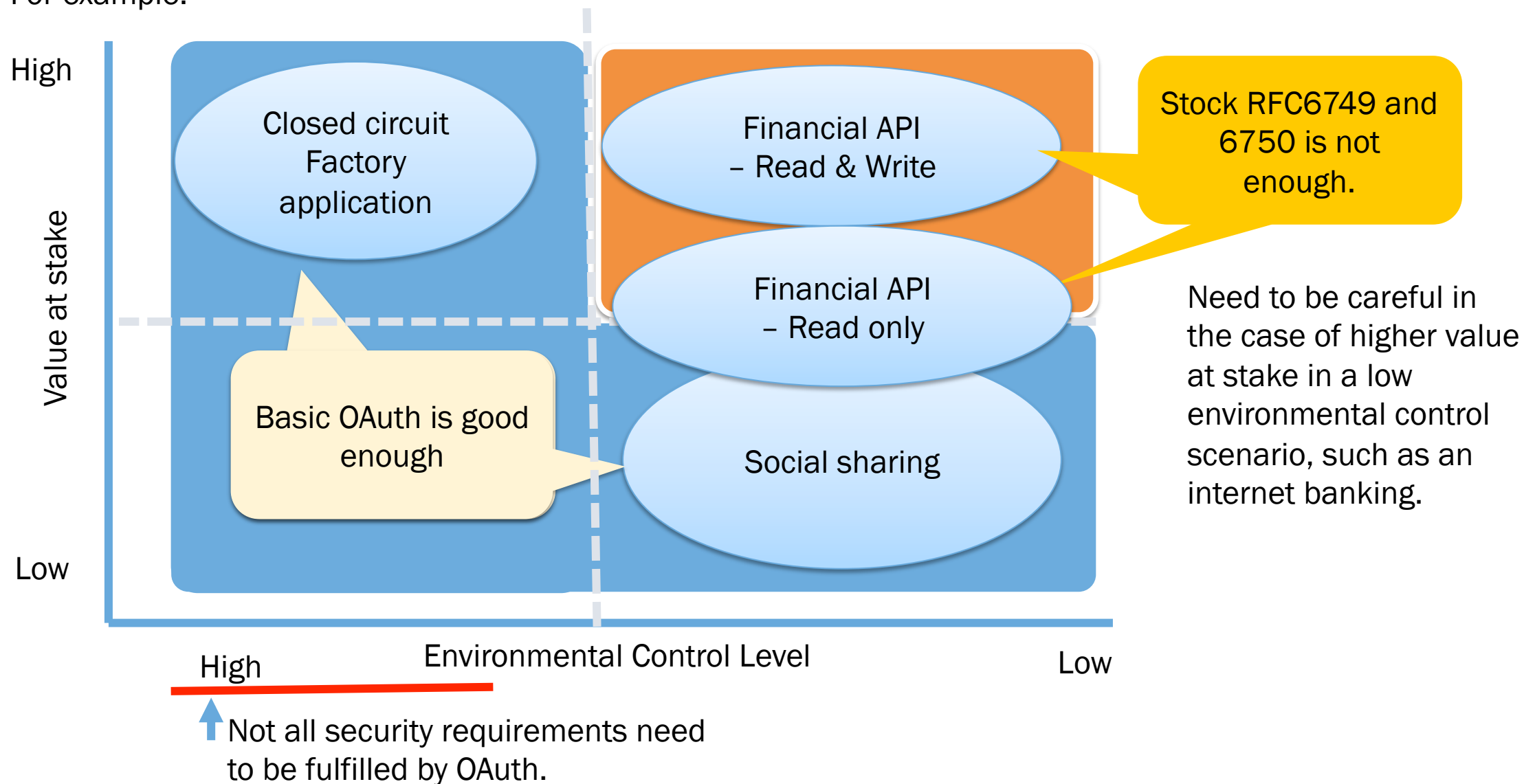
The OAuth 2.0 Authorization Framework

Abstract

The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction

and needs to be **profiled** to suit the circumstances and use case.

For example:

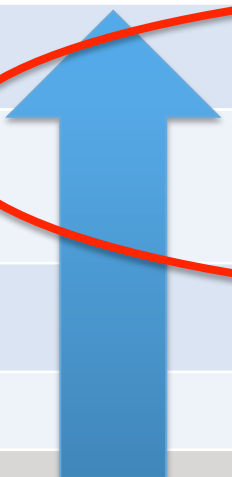


RFC6749 state of the source, destination, and message authentication

	Sender Authentication	Receiver Authentication	Message Authentication
Authorization Request	Indirect	None	None
Authorization Response	None	None	None
Token Request	Weak	Good	Good
Token Response	Good	Good	Good

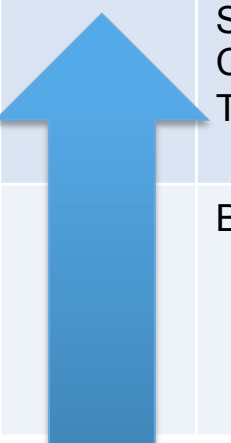
OAuth 2.0 related options and the security levels

Authorization Request/Response types and the security levels

Security Level	Authorization types	Description
	JWS Authz Req w/Hybrid Flow	Authz Request protected
	Hybrid Flow* ¹ (confidential client)	Authz Response protected (ID Token acts as the detached signature for the response.)
	Code Flow (confidential client)	Client authentication
	Implicit Flow	No client authentication
	<i>Plain OAuth</i>	<i>Anonymous</i>

*1) state injection taken care of by including 's_hash'

Token Types and the security levels

Security Level	Token Type	Description
	Sender Constrained Token	Only the named Party with a correct Key can use the token
	Bearer Token	Anyone can use The token



e.g., tighten up the source, destination, and message authentication


	Sender Authentication	Receiver Authentication	Message Authentication
Authorization Request	Request Object	Request Object	Request object
Authorization Response	Hybrid Flow	Hybrid Flow	Hybrid Flow
Token Request	Good	Good	Good
Token Response	Good	Good	Good

To create an appropriate OAuth profile for financial API use, we need to consider multiple factors:

These are not taken into account too often resulting in too many unsafe OAuth 2.0 implementations.

Example of factors:

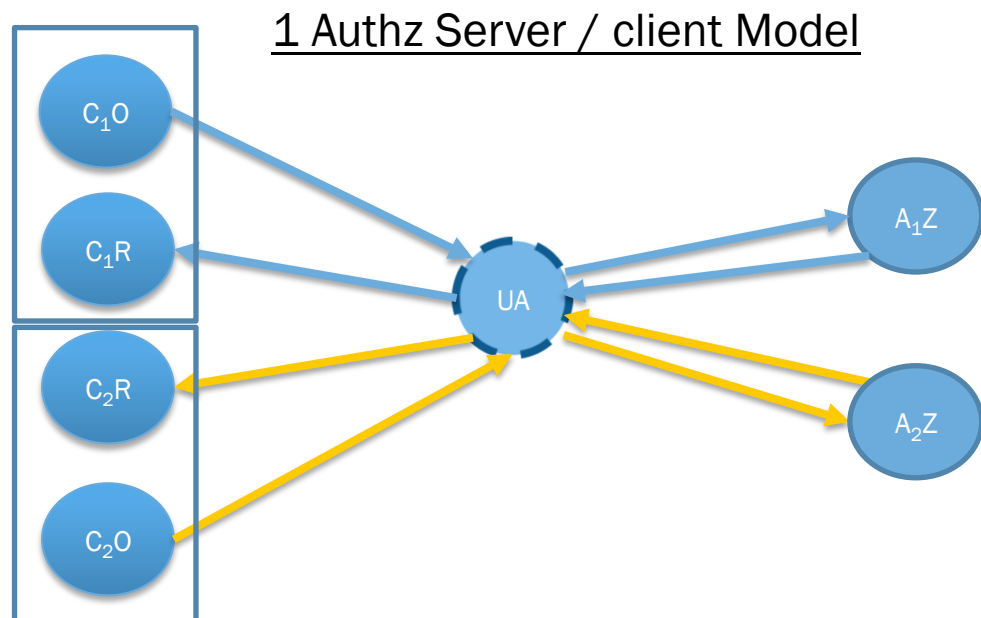
- 1 Server/client assumption
- Message authentication
- Source authentication
- Destination authentication
- User authentication
- Message confidentiality
- Token Phishing/Replay



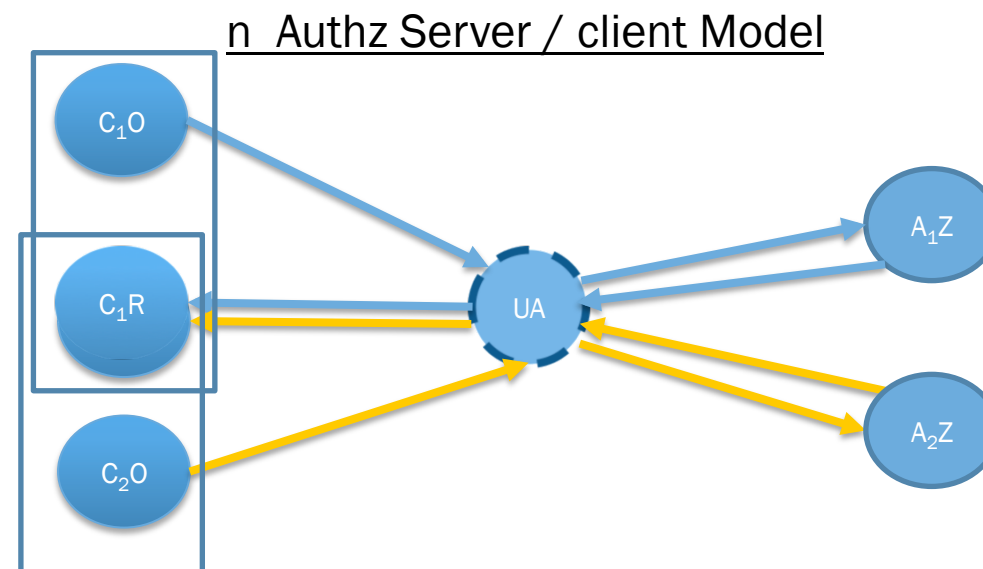
Financial API Profile needs to Solve them all.

OAuth's primary security assumption is that there is only 1 Authz Server per client:

- In case of a Personal Finance Management Software/Client, it will necessarily have multiple Authz Servers.
 - Make sure to have virtual separation, i.e., having different redirect endpoints for each server to avoid Authz server mix-up attack etc.

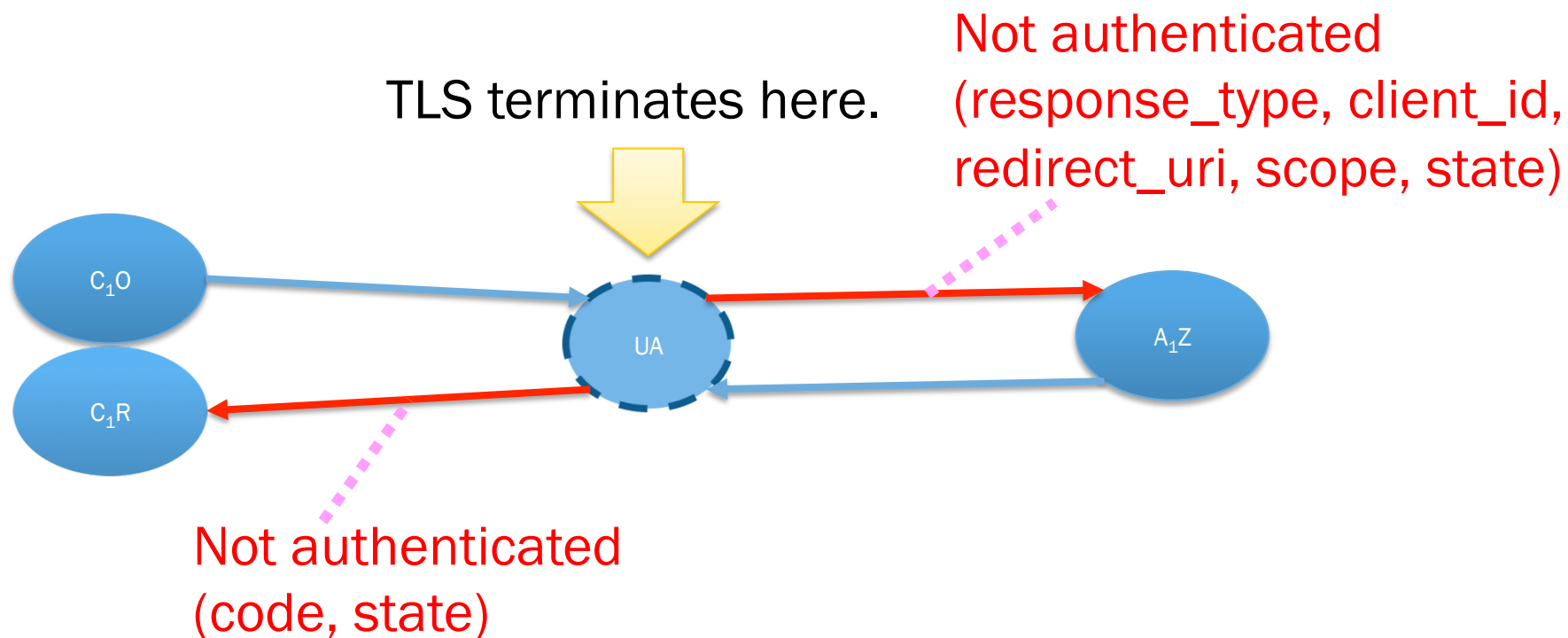


V.S.



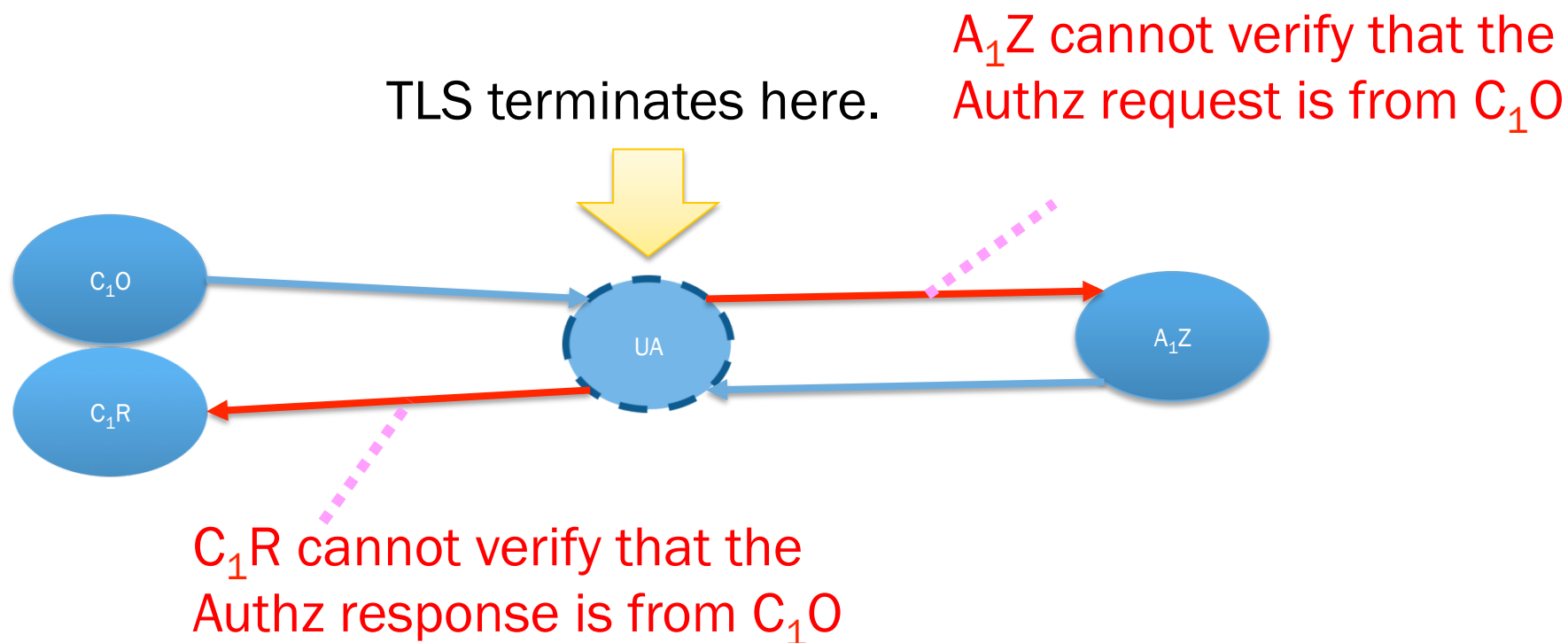
Message Authentication Problems

- Communication through UA are not authenticated and thus can be tainted, but often used without taint check.
- Neither 'code' nor 'state' can be taken at its face value, but we do...



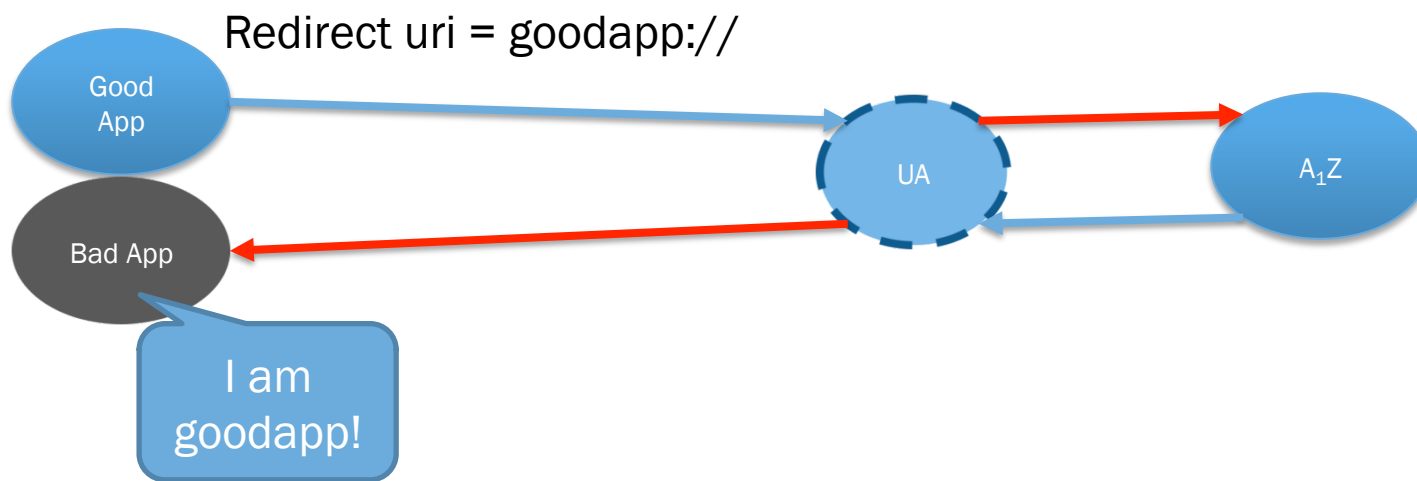
Message Source Authentication Problems

- Since the authorization request and response goes through the browser, the receiving ends cannot be sure of who is the real sender.



Message Destination Authentication Problems

- We are in a mobile app world, right?
- “Code phishing” on public clients a.k.a. mobile apps
- Custom scheme etc. can be hijacked by malware on the device.
 - It has been exploited against popular apps.
 - RFC7636 OAuth PKCE exists for the mitigation of this problem.



Identity and authentication problems

- OAuth has no notion of user identity.
- User authentication is “out of scope”.

**Say OAuth is an Authentication
standard again.**

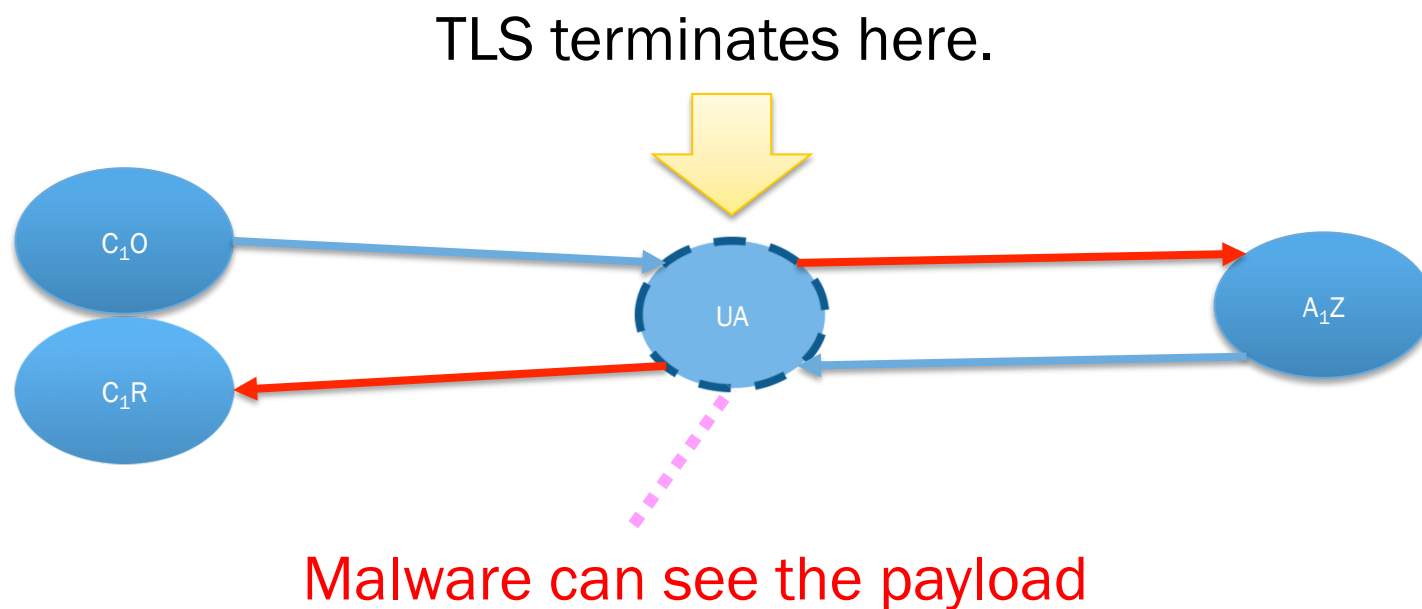


I dare you. I double dare you.

Created by [@nishantk](#)

Message confidentiality problems

- Authorization request is not encrypted in the application layer thus can be seen by the Man-in-the-browser etc.
- And we know that malware abounds.
 - The most popular Online Banking attack in Japan since 2014 is man-in-the-browser.

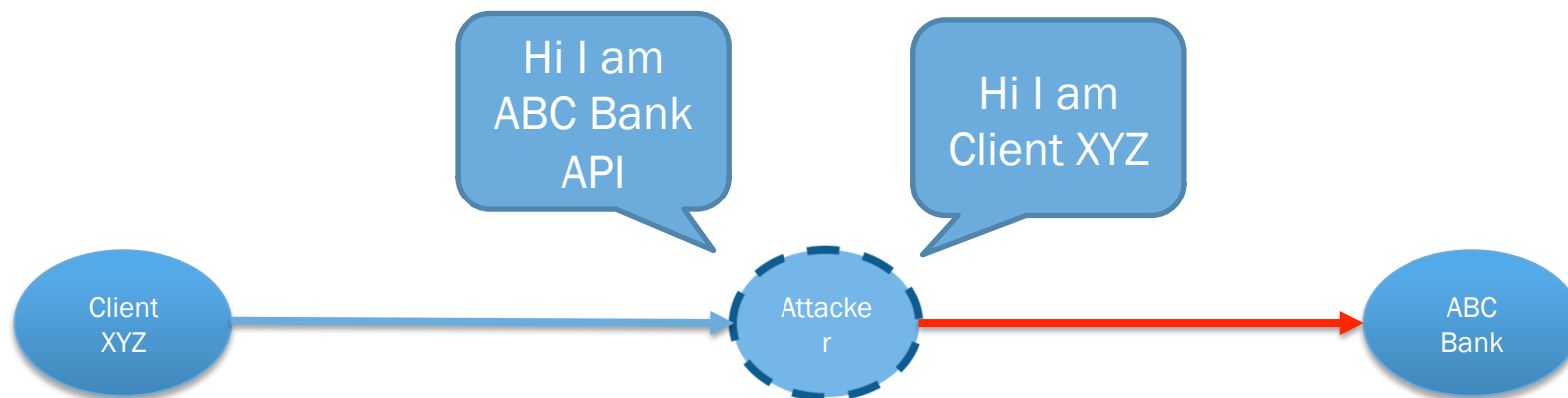


Token Phishing / Token Replay

■ Clients sends token requests and resource requests to forged/compromised servers. Then, these servers can act as a hostile client to replay the request.

● E.g.,

- Sending a fake email to developer that the endpoints has been changed. (We know that about 1 in 20 trained engineer gets phished.)
- Combination of TLS certs mis-issuances and DNS spoofing, etc. ← there seems to be real examples for the attacks against banks.



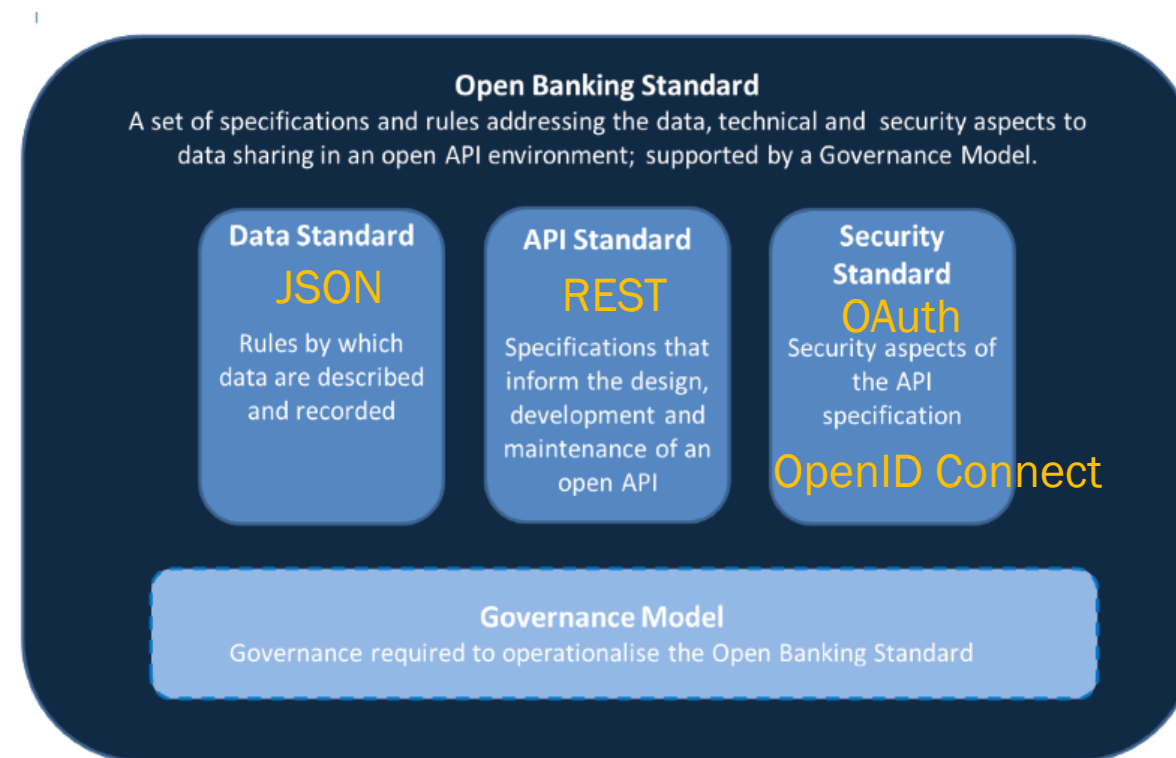
To solve these problems, OpenID Foundation Financial API (FAPI) WG was formed.

■ Scope

- The goal of FAPI is to provide JSON data schemas, REST APIs, and security & privacy recommendations and protocols to:
 - applications to utilize the data stored in the financial account,
 - applications to interact with the financial account, and
 - users to control the security and privacy settings.

Both commercial and investment banking account as well as insurance, and credit card accounts are to be considered.

(Source) OpenID Foundation Financial API WG draft charter



(SOURCE) ODI OBWG: The Open Banking Standard (2016)

For details, see:

<https://openid.net/wg/fapi/>

Why OpenID Foundation?

Right People

- Authors of OAuth, JWT, JWS, OpenID Connect are all here.

Right IPR

- Royalty Free, Mutual Non-Assert, so that everyone can use it freely.

Right Structure

- Free to join WGs. (Sponsors welcome)
- WTO TBT Compliant Process.

In a IPR safe and Completely Open Environment

■ IPR regime

- Mutually assured patent non-assert
- Trademark (OpenID®) control against false claim of the spec support
- Certification support to reinforce the interoperability

■ Completely Open Environment

- Free of charge to join the WG as long as you file the IPR agreement
- Bitbucket (git) to track the changes
 - File an issue and send a pull request!

■ Made possible by these sponsors!

Sustaining corporate members (board members)



verizon✓

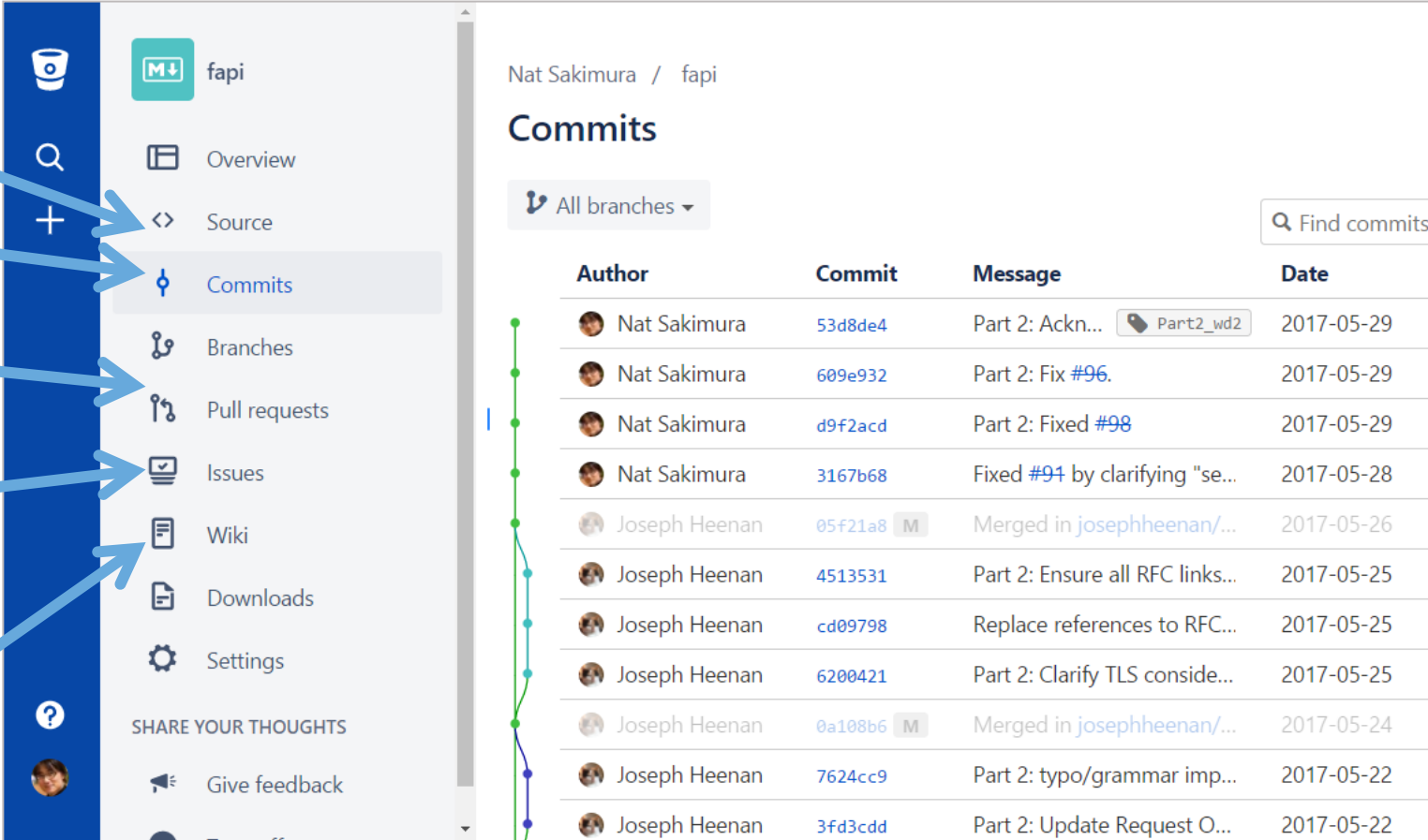
Corporate members



Non-profit members



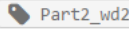


WG works through the weekly conference calls (alternating times for the Atlantic and the Pacific time zones), the mailing list, and the project repository (<https://bitbucket.org/openid/fapi/>)



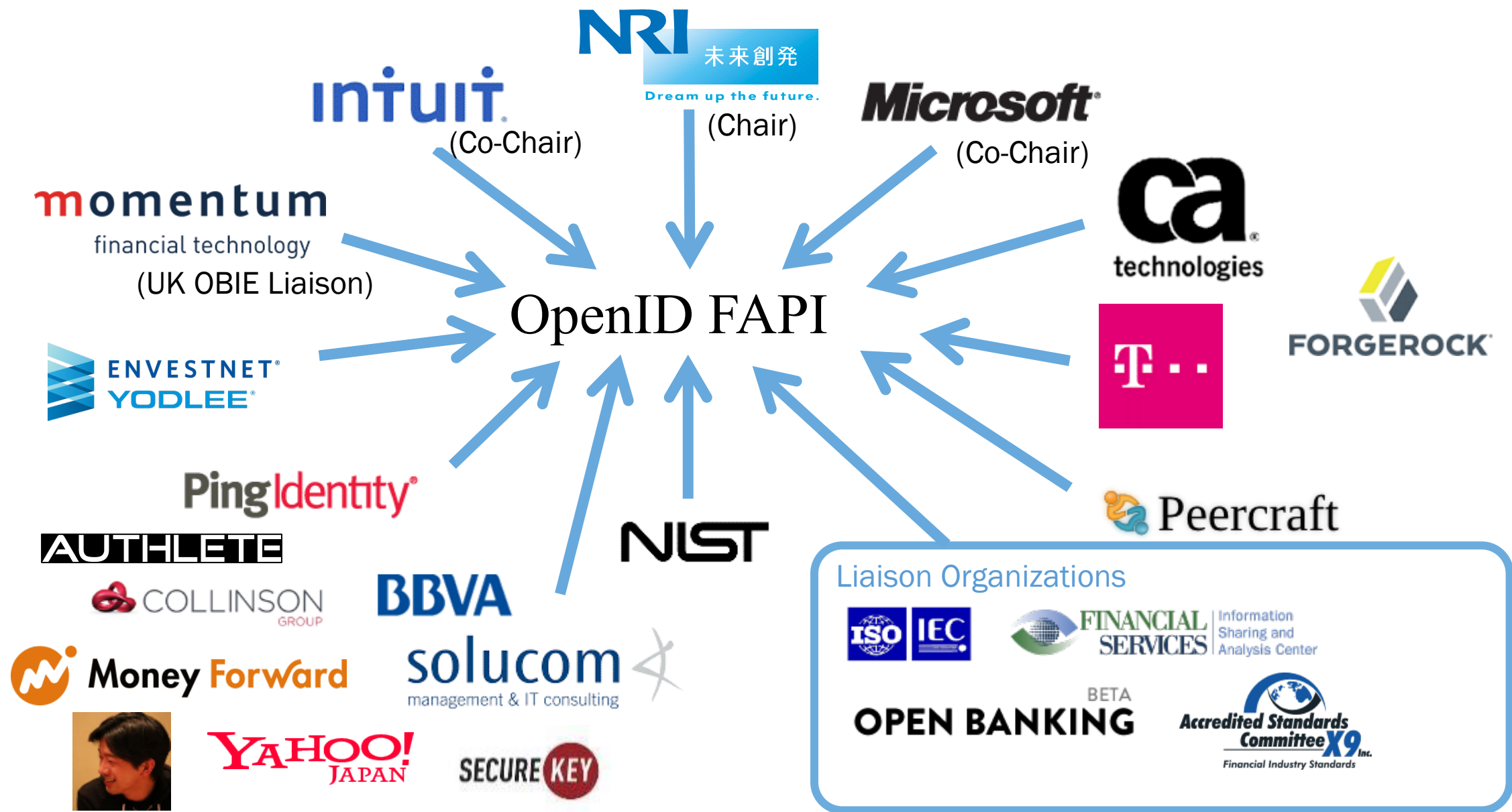
The screenshot shows the Bitbucket web interface for the 'fapi' repository. On the left, a dark blue sidebar contains navigation icons and a menu. Blue arrows point from text labels to specific items in this menu:

- Draft Text** points to the document icon at the top of the sidebar.
- Commit History** points to the 'Commits' item in the sidebar menu.
- Pull Request** points to the 'Pull requests' item in the sidebar menu.
- Issue Tracking** points to the 'Issues' item in the sidebar menu.
- Meeting Notes etc.** points to the 'Wiki' item in the sidebar menu.

The main content area on the right is titled 'Commits' and shows a list of recent commits. A search bar 'Find commits' is located at the top right of this section. Below the commit list, a vertical timeline visualization shows the sequence of commits.

Author	Commit	Message	Date
Nat Sakimura	53d8de4	Part 2: Ackn... 	2017-05-29
Nat Sakimura	609e932	Part 2: Fix #96 .	2017-05-29
Nat Sakimura	d9f2acd	Part 2: Fixed #98	2017-05-29
Nat Sakimura	3167b68	Fixed #94 by clarifying "se..."	2017-05-28
Joseph Heenan	05f21a8 	Merged in josephheenan/...	2017-05-26
Joseph Heenan	4513531	Part 2: Ensure all RFC links...	2017-05-25
Joseph Heenan	cd09798	Replace references to RFC...	2017-05-25
Joseph Heenan	6200421	Part 2: Clarify TLS conside...	2017-05-25
Joseph Heenan	0a108b6 	Merged in josephheenan/...	2017-05-24
Joseph Heenan	7624cc9	Part 2: typo/grammar imp...	2017-05-22
Joseph Heenan	3fd3cdd	Part 2: Update Request O...	2017-05-22

Working Together



Current Specs.

■ Financial Services – Financial API –

● Part 1: Read Only API Security Profile

<http://openid.net/specs/openid-financial-api-part-1.html>

- Implementer's Draft (I-D) ~Implementations going on

● Part 2: Read and Write API Security Profile

<http://openid.net/specs/openid-financial-api-part-2.html>

- Under Public Review

● Part 3: Open Data API

- Waiting for the UK OBIE Contribution

● Part 4: Protected Data API and Schema - Read only

- Bank Account – Based on the US FS-ISAC Contribution

● Part 5: Protected Data API and Schema - Read and Write

- Waiting for UK OBIE Contribution
 - Using Claims Request to obtain granular consent

Swagger files are going to be provided



Probably need to be registry entries rather than “Parts”

Financial Services – Financial API --

Part 1: Read Only API Security Profile

- Note: ISO Keywords, “shall”, “should”, “may”, “can” are used.
- Lots of “shall”s. Need to fulfill them all for an adequate security level.

5.2.2. Authorization Server

The Authorization Server

- shall support both public and confidential clients;
- shall provide a client secret that adheres to the requirements in section 16.19 of [OIDC] if a symmetric key is used;
- shall authenticate the confidential client at the Token Endpoint using one of the following methods:
 1. TLS mutual authentication [TLSM];
 2. JWS Client Assertion using the `client_secret` or a private key as specified in section 9 of [OIDC];
- shall require a key of size 2048 bits or larger if RSA algorithms are used for the client authentication;
- shall require a key of size 160 bits or larger if elliptic curve algorithms are used for the client authentication;
- shall support [RFC7636] with S256 as the code challenge method;
- shall require Redirect URIs to be pre-registered;
- shall require the `redirect_uri` parameter in the authorization request;
- shall require the value of `redirect_uri` to exactly match one of the pre-registered Redirect URIs;
- shall require user authentication at LoA 2 as defined in [X.1254] or more;
- shall require explicit consent by the user to authorize the requested scope if it has not been previously authorized;
- shall verify that the Authorization Code has not been previously used if possible;
- shall return the token response as defined in 4.1.4 of [RFC6749];
- shall return the list of allowed scopes with the issued access token;
- shall provide opaque non-guessable access tokens with a minimum of 128 bits as defined in section 5.1.4.2.2 of [RFC6819].
- should clearly identify long-term grants to the user during authorization as in 16.18 of [OIDC]; and
- should provide a mechanism for the end-user to revoke access tokens and refresh tokens granted to a Client as in 16.18 of [OIDC].

Adoption among the industry is great!

Japanese Banker's Association Recommendation (16 March 2017)



一般社団法人
全国銀行協会

[一般の方
教えて！
くらしと銀行](#) [金融犯罪の手口](#) [学校教育や
消費者教育に携わる方](#) [全銀協の活動を
知りたい方](#) 

全銀協の活動を知りたい方

[全銀協の活動を知りたい方](#) [ニュース&トピックス](#) [「オープンAPIのあり方に関する検討会」\(第9回\) 議事要旨の公表について](#)

[ニュース&トピックス](#) [全銀協からの意見書・要望書](#) [会長記者会見](#)

ニュース&トピックス詳細

平成29年3月16日

各位

一般社団法人全国銀行協会

**「オープンAPIのあり方に関する検討会」(第9回) 議事要旨
の公表について**

「オープンAPIのあり方に関する検討会」(事務局：一般社団法人全国銀行協会)の第9回会合を開催(平成29年2月27日)しましたので、議事要旨を別添のとおり公表いたします。また、これまでの検討会における討議およびその後のメンバーからのコメントを踏まえた「オープンAPIのあり方に関する検討会報告書」の中間的な整理(案)は、添付ファイルのとおりです。

なお、本検討会の設置につきましては、平成28年10月21日に当協会のウェブサイトで公表しておりますので、ご参照ください。

- 「オープンAPIのあり方に関する検討会」(第9回) 議事要旨 
- 「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-【中間的な整理(案)】」

[全銀協の活動](#)

[各種統計資料](#)

[組織概要](#)

[採用情報](#) 

[関係団体](#) 

Open Banking Implementation Entity Announcement (17 May 2017)

BETA
OPEN BANKING

ABOUTCUSTOMERSDEVELOPERSAPI PROVIDERSINDUSTRYCONTACT

MAY
17
2017

Open Banking forms collaboration with OpenID Foundation

The Open Banking Implementation Entity (OBIE), the organisation responsible for the open API banking standard, today announces its collaboration with the OpenID Foundation's Financial API Working Group.

[Read More](#)

APR
13
2017

CMA Appoints New Trustee for Open Banking Implementation Entity

The Competition & Markets Authority ('CMA') has, today, announced that Imran Gulamhuseinwala will become the new trustee for the Open Banking Implementation Entity (the 'IE').

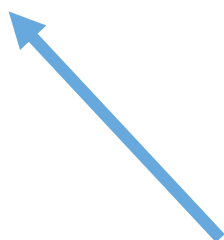
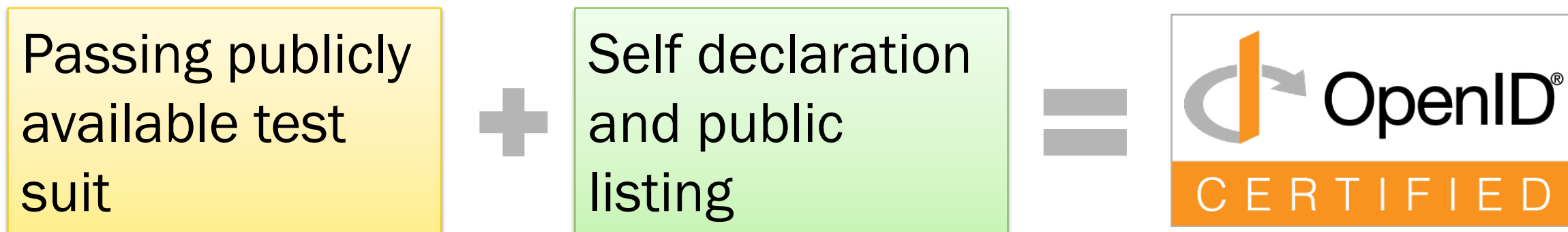
[Read More](#)



How do you know that it has been
implemented correctly?

A certification test suite is being planned to be provided online

For more details, see <http://openid.net/certification/>



We currently only have a generic test for Basic OpenID Connect capabilities.
We need to add tests for FAPI.
Directed funding is being sought now.

Join the group!

`https://openid.net/wg/fapi/`