Fintech協会「第5回　API・セキュリティ分科会」資料

金融サービス業における OAuthとOpenID Connect

2017年2月23日

崎村 夏彦(@_nat)
米国OpenID Foundation理事長
野村総合研究所　上席研究員

崎村夏彦(Nat Sakimura)

(courtesy of Hired @ APIDays 2016)

- 野村総合研究所上席研究員
  OpenID Foundation **理事長**
- Financial API WG議長
- ISO/IEC JTC 1/SC 27/WG5国内小委員会主査
- OECD/SPDEリエゾン（WG5より）

著作:

- OpenID Connect Core 1.0
- JSON Web Token [RFC7519]
- JSON Web Signature [7515]
- OAuth PKCE [RFC7636]
- OAuth JAR [IETF Last Call]
- Etc.

Editor of:

- ISO/IEC 29184 Guidelines for online notice and consent
- ISO/IEC 29100 AMD: Privacy Framework
- ISO/IEC 27551 Requirements for attribute based unlinkable entity authentication
- Etc.

- https://nat.Sakimura.org/
- @_nat_en (English)
- @_nat (**日本語**)
- Linked.in/natsakimura
- https://www.linkedin.com/in/natsakimura
- https://ja.wikipedia.org/wiki/**崎村夏彦**

モバイルファーストの時代には,
API保護にOAuth 2.0 を使うのは当然.

問題解決?!

モバイルファーストの時代には,
API保護にOAuth 2.0 を使うのは当然だが、
OAuthを使えというだけでは問題解決になっていない。

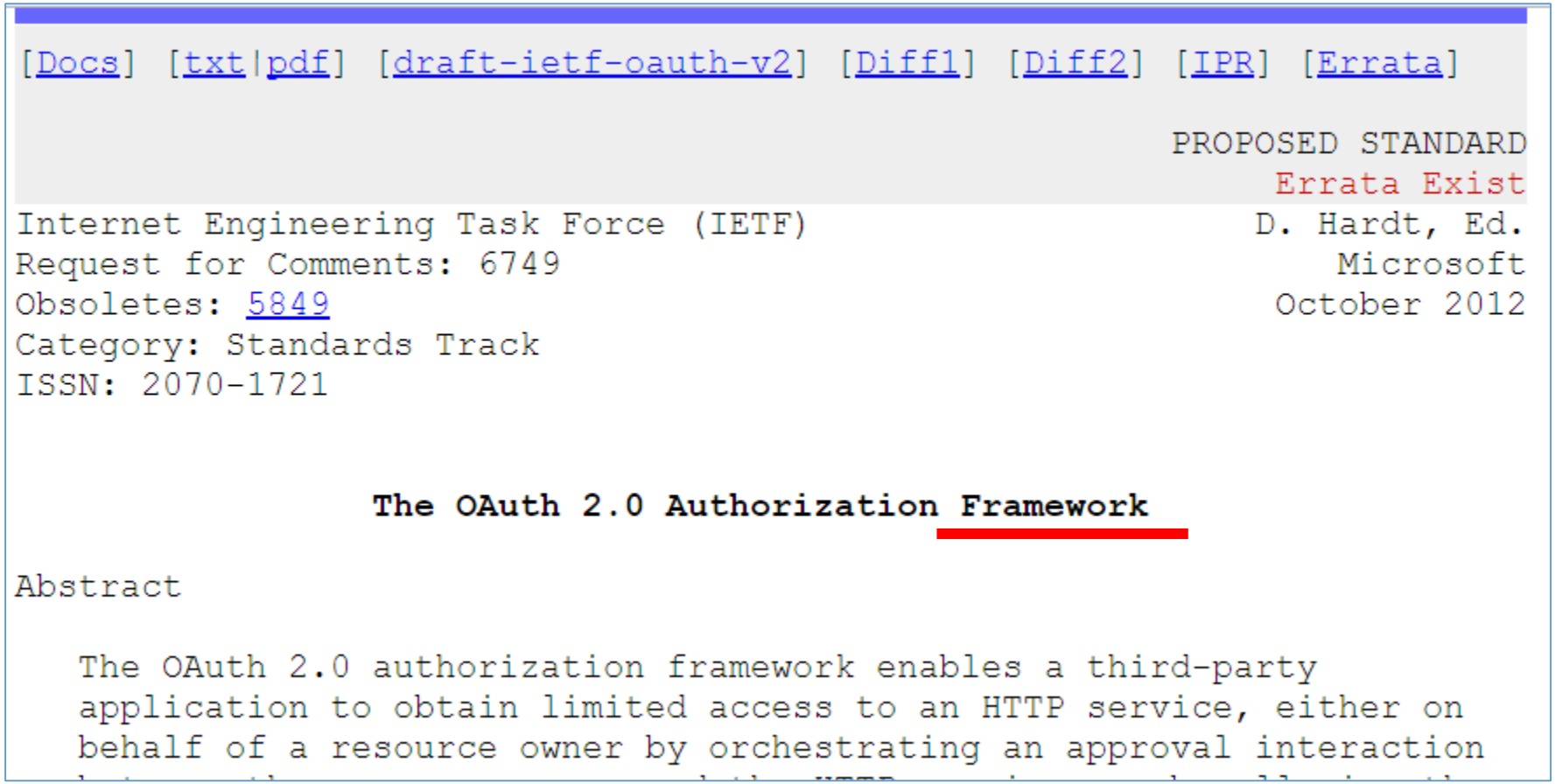
(SOURCE) Photo taken by Nat Sakimura @APIDays on 13th Dec. 2016

“部品を正しく組み合わせることが重要だ。#oauth を使えば良いと言うだけでは解決策になっていない。”

-- Mark O’Neill, Gartner

@APIDays Paris 2016

OAuth 2.0 はフレームワークである。



[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

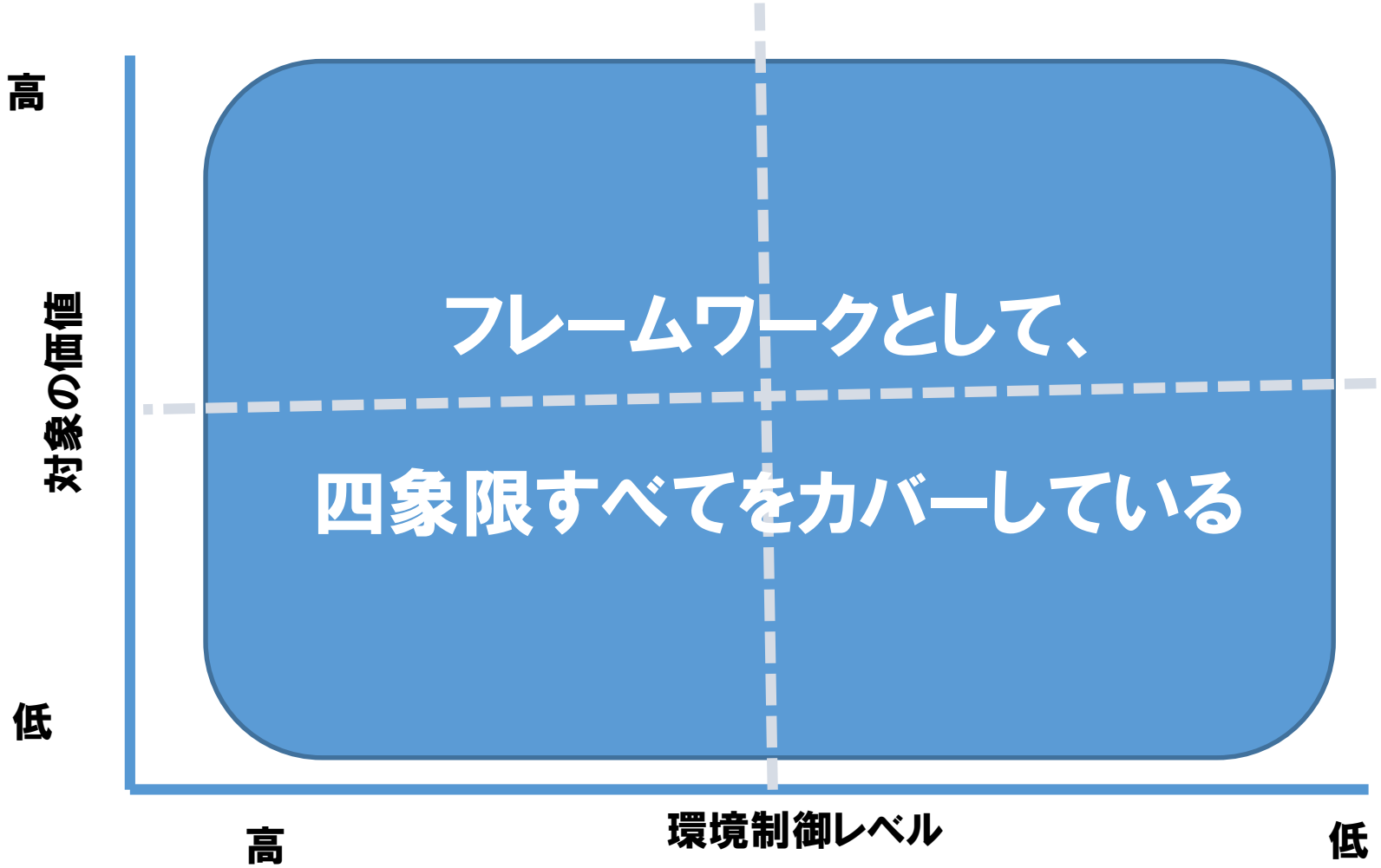
D. Hardt, Ed.
Microsoft
October 2012

The OAuth 2.0 Authorization Framework

Abstract

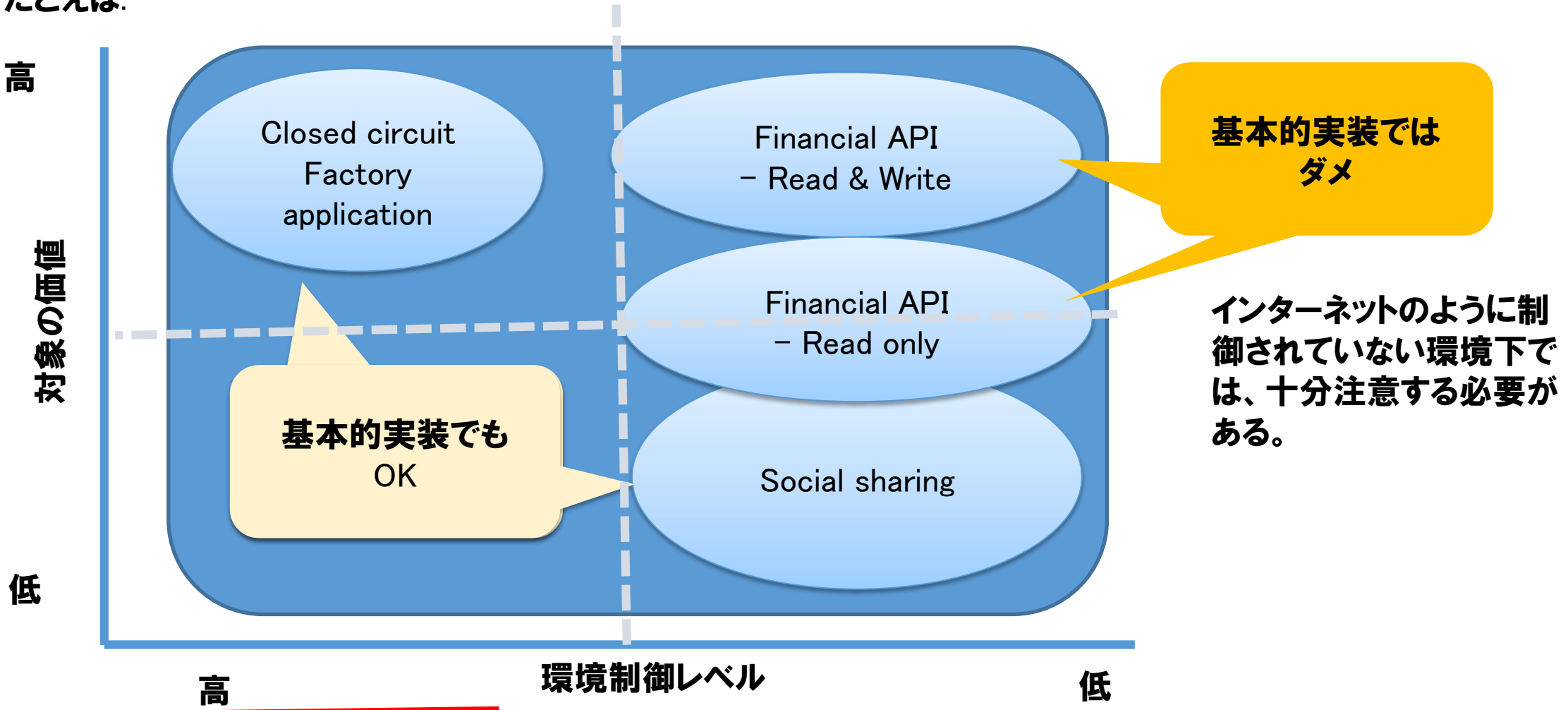
The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction

これにより、様々なサービス環境に対応できるようになっている。



しかしそれは、個別の状況に応じてプロファイルを用意する必要があることを意味している。

たとえば:



OAuth 2.0 関連のオプション機能とセキュリティレベル

認証要求・応答の種類とセキュリティ・レベル

セキュリティ・レベル	機能セット	適用
	JWS Authz Req w/Hybrid Flow	Authz Request protected
	Hybrid Flow*1 (confidential client)	Authz Response protected
	Code Flow (confidential client)	Client authentication
	Implicit Flow	No client authentication
	Plain OAuth	*Anonymous*

*1) `state`インジェクションの回避のために、‘s_hash’を含む。

トークンの種類とセキュリティ・レベル

セキュリティ・レベル	トークンの種類	適用
	Token Bind Token	Authz Request protected
	持参人トークン (Bearer Token)	Authz Response protected

金融APIのためのプロファイルを作る上では、複数の要因を考慮する必要がある。

これらの多くはしばしば無視され、結果として非常に危ないOAuth 2.0実装を産んでいる。

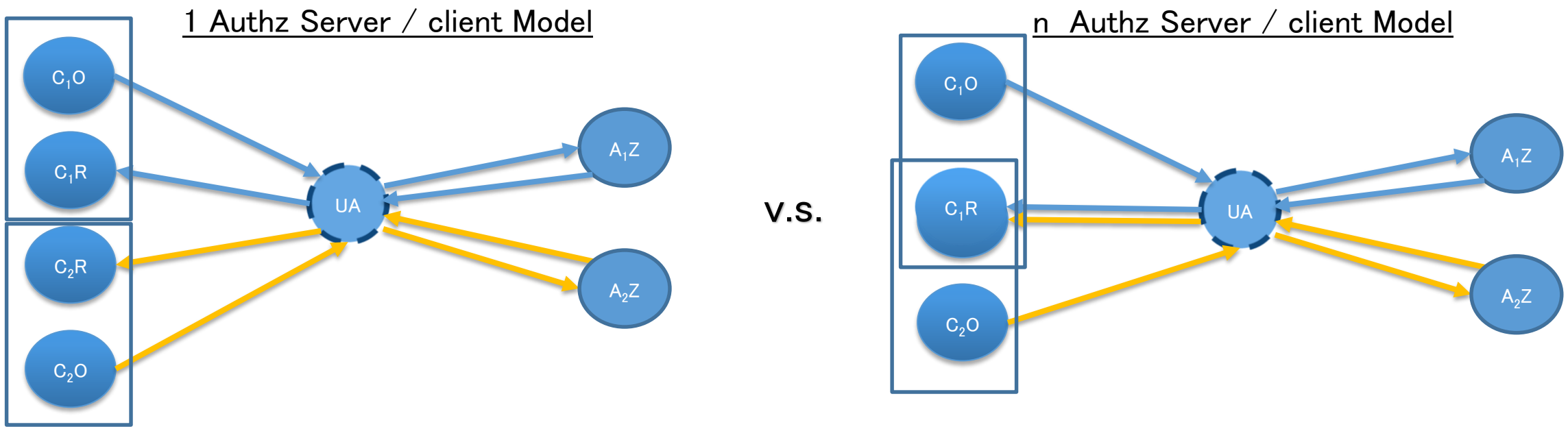
要因の例：

- ■1クライアント1サーバの前提
- ■メッセージ認証（要求・応答）
- ■送信者認証
- ■受信者認証
- ■利用者認証
- ■メッセージ秘匿性
- ■トークンフィッシング／リプレイ

金融機関向けのプロファイルは
これら全てを解決する必要がある。

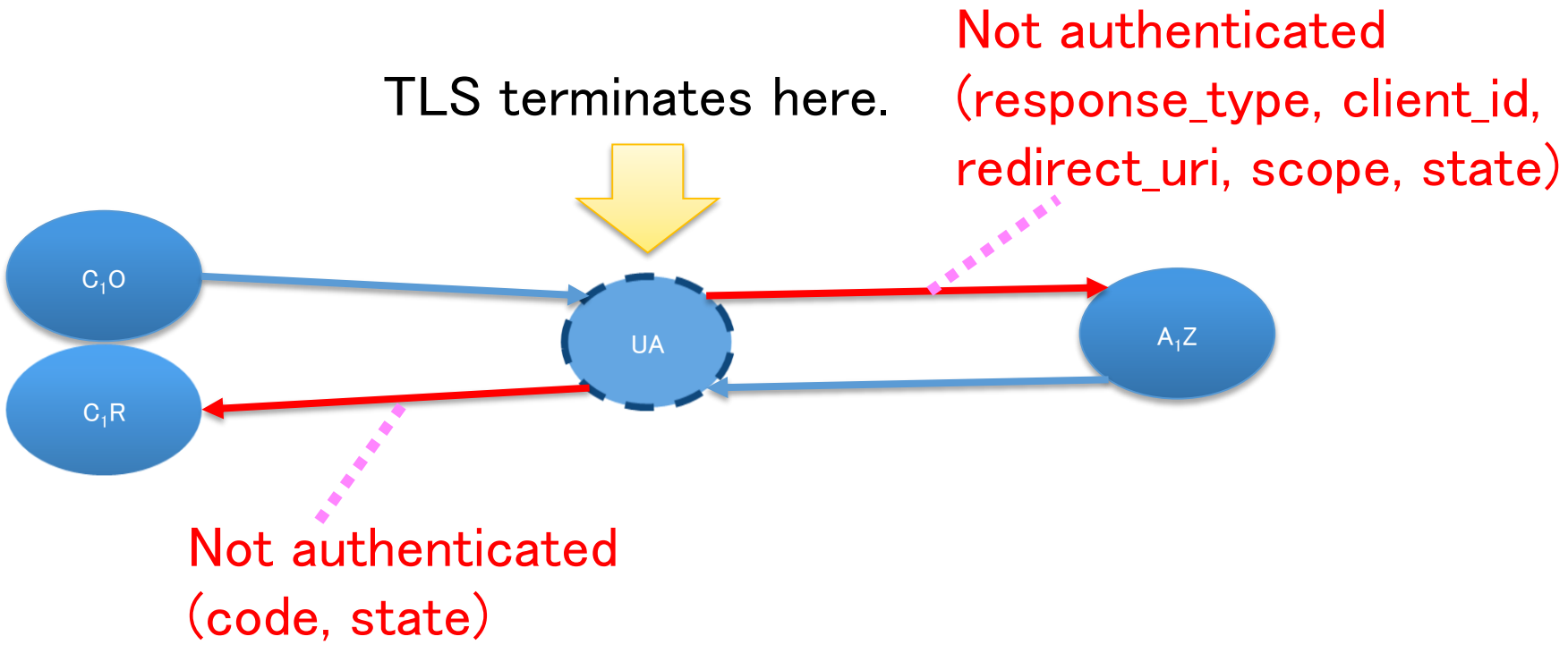
OAuth's primary security assumption is that there is only 1 Authz Server per client:

- 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.



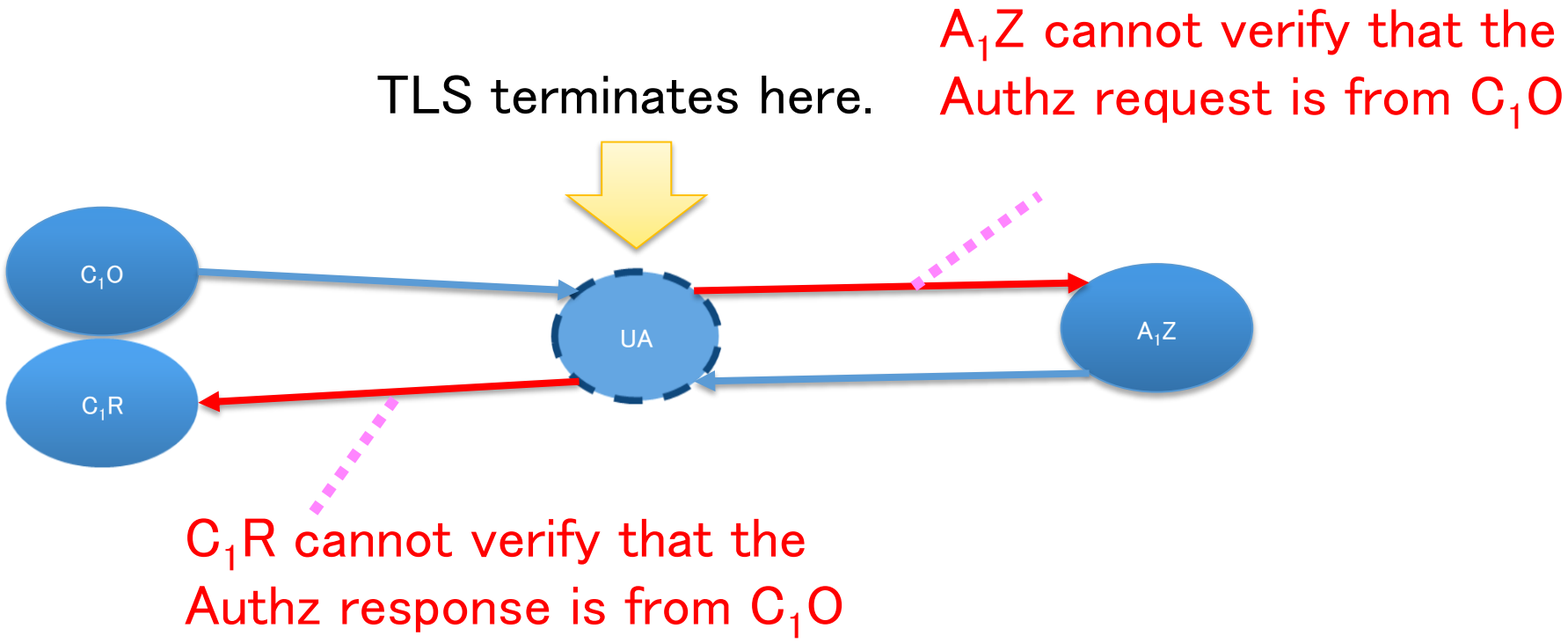
Message Authentication Problems

- 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...



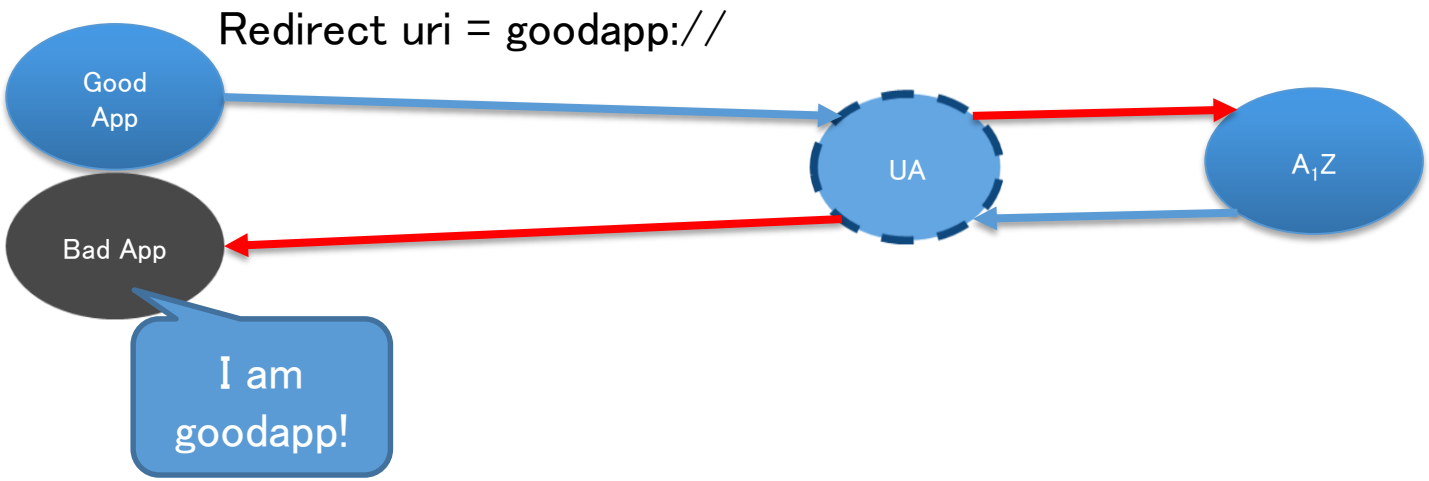
Message Source Authentication Problems

- 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”.

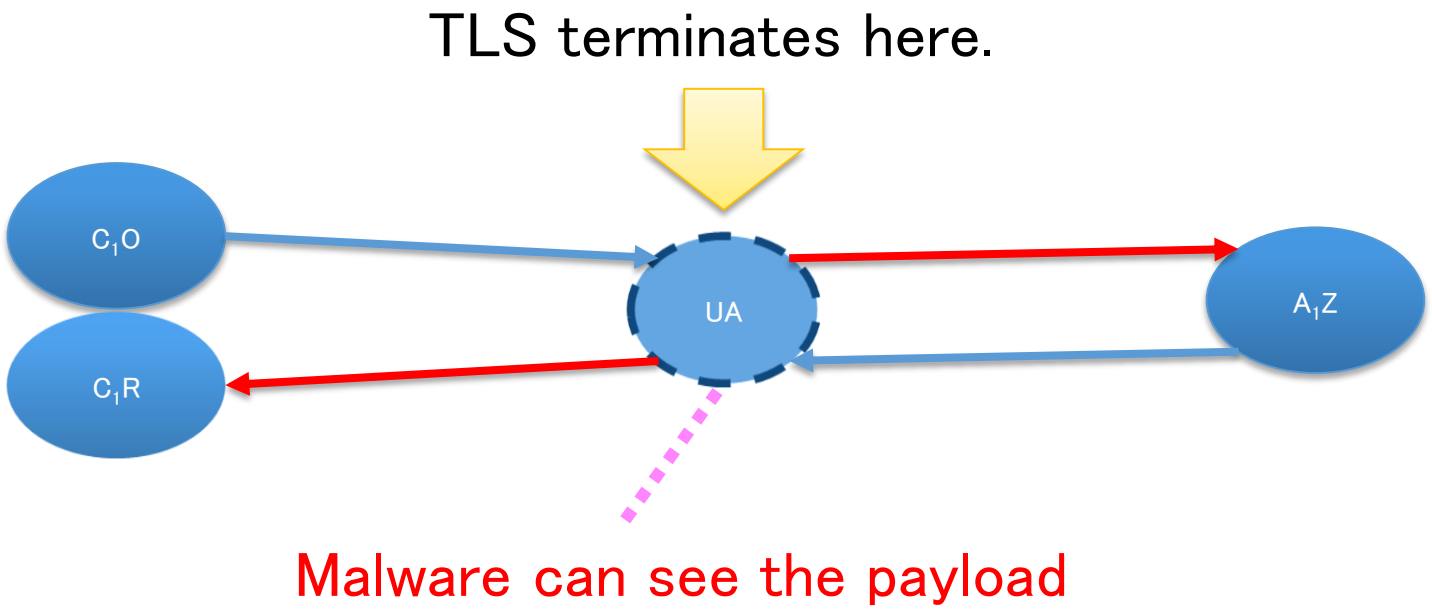
Say OAuth is an Authentication standard again.

I dare you. I double dare you.

Created by @nishantk

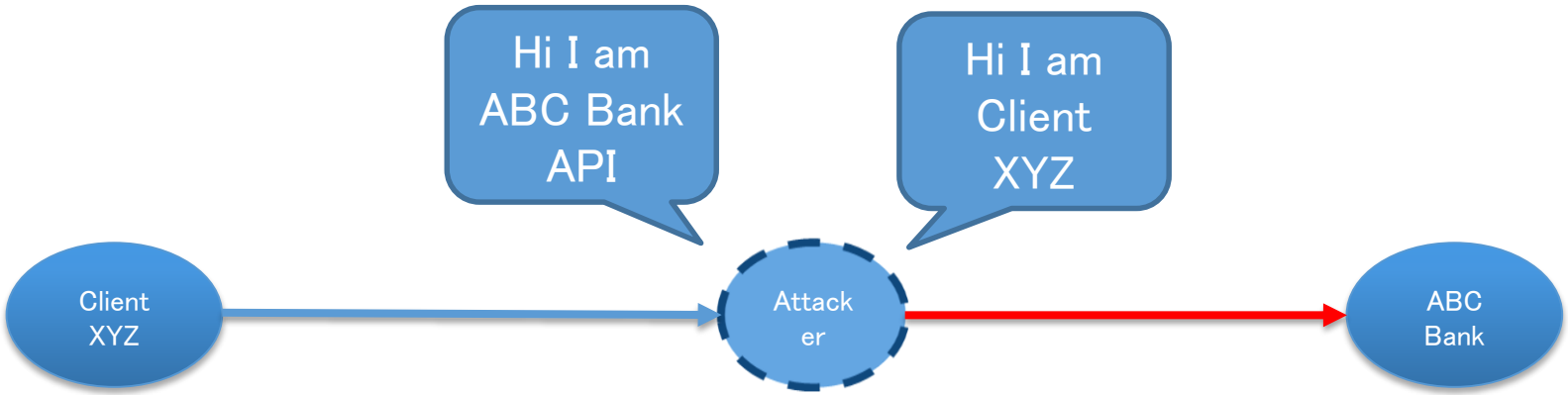
Message confidentiality problems

- 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

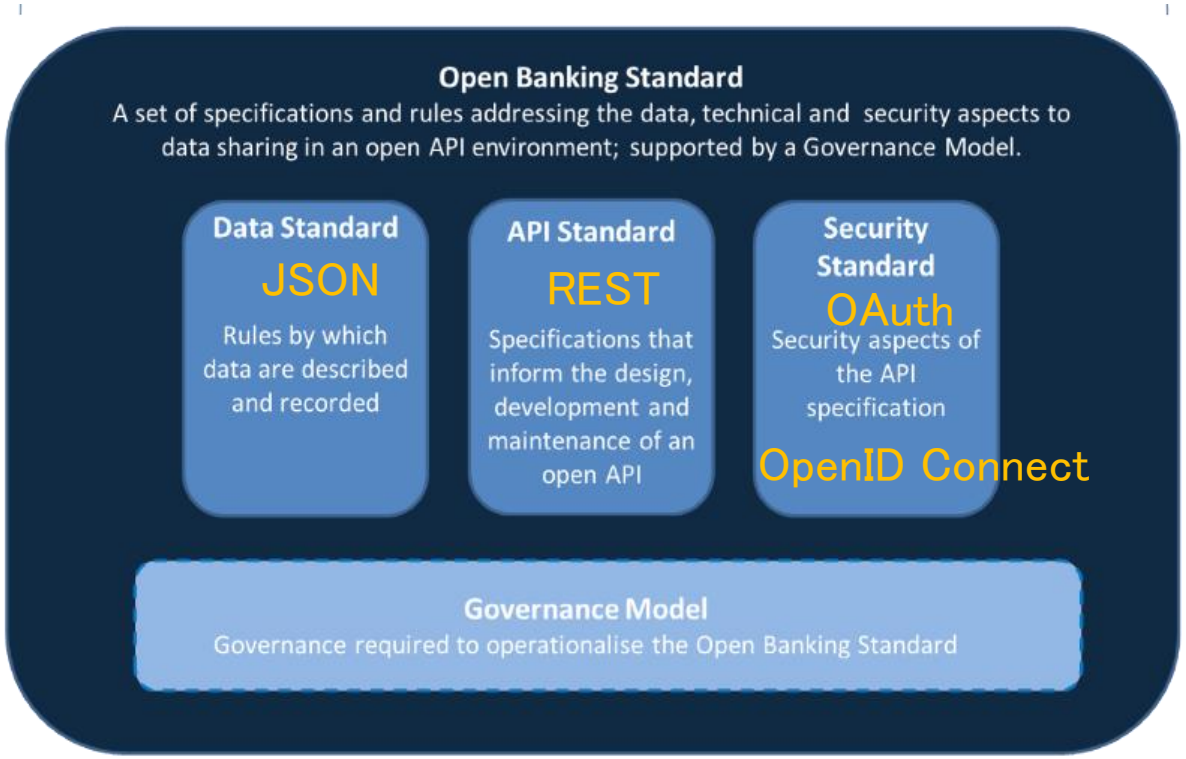
- Clients sends token requests and resource requests to forged/compromised servers. Then, these servers can act as a hostile client to replay the request.
  - E.g.,
    - Sending a fake email to developer that the endpoints has been changed. (We know that about 1 in 20 trained engineer gets phished.)
    - Combination of TLS certs mis-issuances and DNS spoofing, etc. ← there seems to be real examples for the attacks against banks.



こうした課題を解決するために組成されたのが、OpenID Foundation のFinancial API WGである。

- 目的
  - セキュリティ＋プライバシー~プロファイル、JSON data schema, REST APIs,
    に関する勧告を提供し、
  - アプリケーションが金融口座に保管されているデータを利用すること、
    アプリケーションが金融口座とやりとりすること、
    利用者がセキュリティとプライバシー設定をすること、
    を可能にすることである。
- 銀行・証券口座のみならず、保険およびクレジットカード口座も考慮対象とする。

(Source) OpenID Foundation Financial API WG draft charterを元に、崎村試訳



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

FAPI WGの詳細情報　↓

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

なぜOpenID Foundationで行うのか？

Right People	・OAuth, JWT, JWS, OpenID Connect **の全著者が所属**
Right IPR	・**無償、相互不主張提供とする知財ポリシー→だれでも自由に実装可能**
Right Structure	・**WG加盟は無料（スポンサーは歓迎だが）** ・WTO TBT **協定準拠のプロセス.**

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!

NRI 未来創発 Dream up the future. × OpenID®

Sustaining corporate members (board members)

Google KDDI Microsoft NRI The Office of the National Coordinator for Health Information Technology ORACLE PayPal PingIdentity Symantec verizon

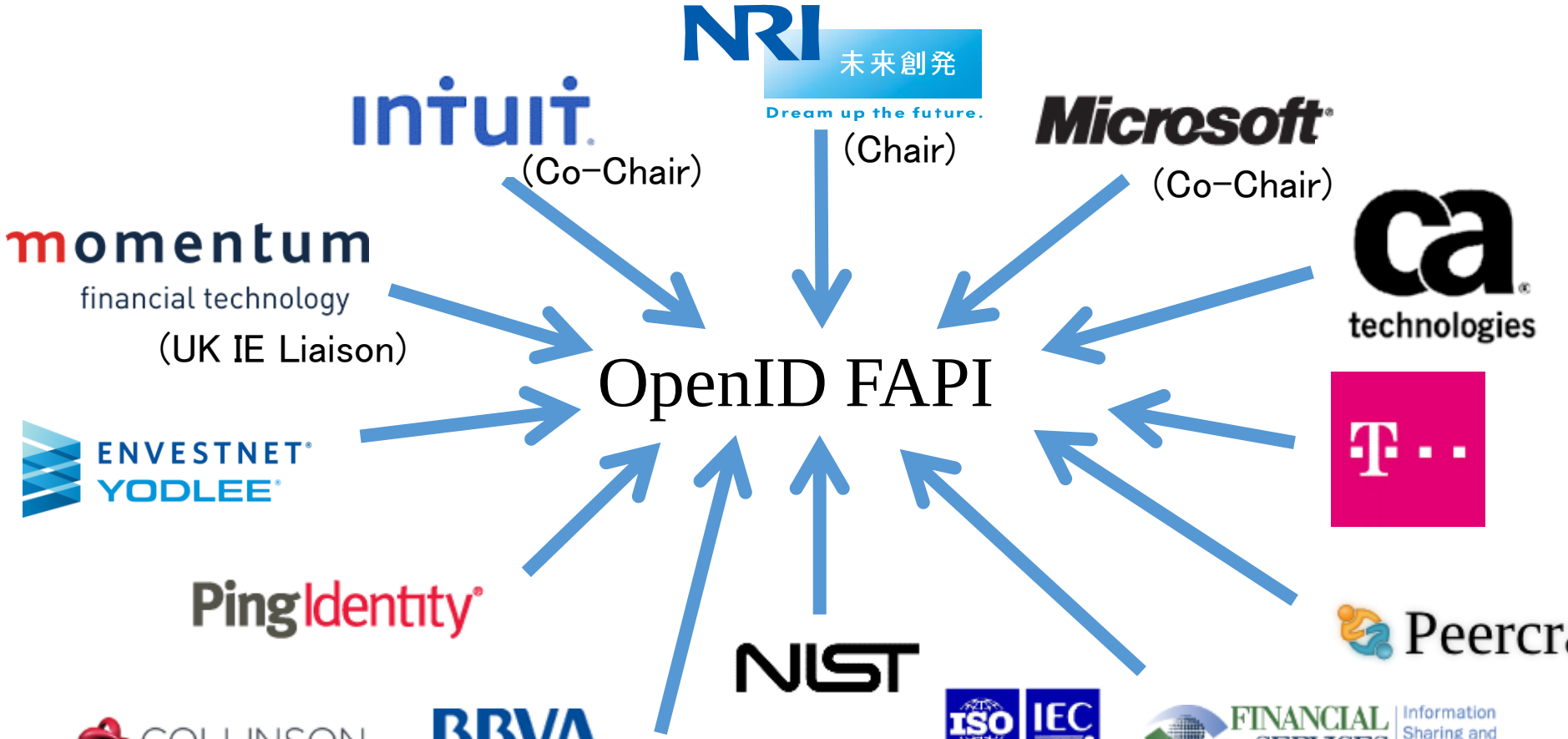
Corporate members

10Duke aplcomp Auth0 AYOY Coherent Cloud Deutsche Telekom T-- FORGEROCK gigya GSMA intuit janrain mydex nextiva okta onelogin PacificEast Peercraft Ribose RSA salesforce.com SANDA SECURE KEY SyncForce TrustX viewds vmware

Non-profit members

Alexandria Institute cz.nic RT pay

Working Together



UK Implementation Entity

現在の仕様構造と今後の見込み

- 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) **～実装開始**
  - Part 2: Read and Write API Security Profile
    - **近々ドラフト完成へ向けて作業中、3月～4月に** Public Review＋I-D**投票**
  - Part 3: Open Data API
    - UK OBS**の動きを見ながら調整**
  - Part 4: Protected Data API and Schema - Read only
    - **銀行口座** - US FS-ISAC DDA / OpenBank Project / Figo **などを参考に作成**
    - **証券口座 – 現在**NRI**で試案作成中**
  - Part 5: Protected Data API and Schema - Read and Write
    - **同上**

Swagger**ファイルを提供。**

レジストリ化？

ISO 20022**？**

Financial Services – Financial API --
Part 1: Read Only API Security Profile

- **注意）**keywords **が、**IETF**的なのと違います。（**ISO**の**Keywords, “shall”, “should”, “may”, “can” **を使っているため）。**
- **いっぱい** “shall” **があります。全部やらないとセキュリティ・レベル、保てません。**

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].

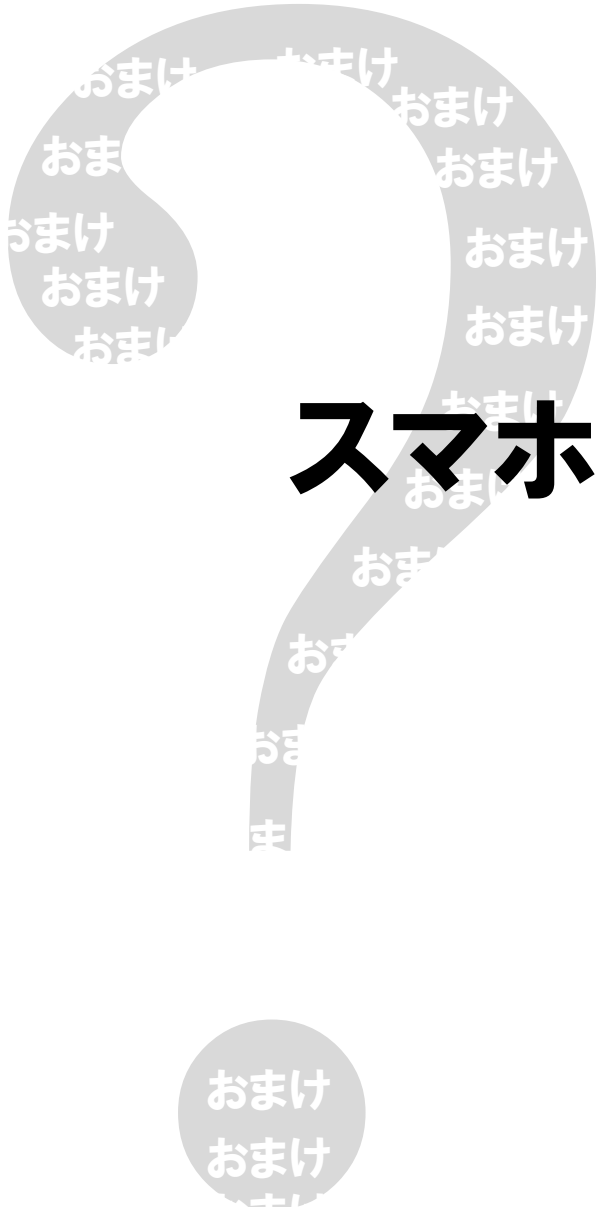
（出所）Financial Services – Financial API –– Part 1: Read Only API Security Profile Implementer’s Draft

これらを正しく実装できているか、
どうやったらわかるか？

Certification test will be available online

仕様が正しく実装されているかどうかは、Certification によって確認できるようにしていく。

- オンライン提供されているテスト・スイートを使って、正しく実装されていることを確認。
- 結果をSelf Certify して登録・公開
  - → FTC法5条の配下に入る。これによって信頼性を担保。
  - また、ログも公開されるので、虚偽の申告は、他者が指摘可能。
- 現在はOP Certificationが正式提供中。
- 来週RP Certificationが発表される予定。
- FAPIにおいても、同様のスキームでテスト可能にする予定。
- Certificationの詳細については、
  http://openid.net/certification/ 参照



スマホ・アプリでOAuthをつかうとき どうしていますか？

モバイルにおける認証時のベスト・プラクティス

■OAuth 2.0 for Native Apps

- ●OpenID Foundationの Native Apps WGの検討結果をIETFに持ち込んだもの。
- ●https://tools.ietf.org/html/draft-ietf-oauth-native-apps-07
- ●Native apps MUST use an external user-agent to perform OAuth authentication requests.
  - ・Embedded User-agent (e.g.WebView) は使ってはいけない.
- ●Authorization servers MUST support the following three redirect URI options.
  - ・App-declared Custom URI Scheme Redirection
  - ・App-claimed HTTPS URI Redirection
  - ・Loopback URI Redirection
- ●Public native app clients MUST protect the authorization request with PKCE [RFC7636].
- ●Authorization servers that still require a shared secret for native app clients MUST treat the client as a public client

Join the group!

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