High-security & interoperable OAuth 2: What’s the latest?

Joseph Heenan

CTO

Authlete

Daniel Fett

Security and Standardization Expert

Authlete

In this Masterclass

Why is OAuth 2.0 alone insufficient for high-security and interoperable applications?

What is FAPI?

FAPI 2.0 Deep-Dive

Current Status & Adoption

e-health

open
banking

e-signing

open
insurance

OAuth 2.0?

open finance

e-
government

open consumer
data

digital identity
ecosystems

Requirements for high-security & interoperable OAuth 2

Highest Levels of Security

Security/
Hardening

OAuth Security Best Current Practice (IETF draft)

Learnings from practice & research:

- Protect against access token misuse, mix-up attacks, and more
- Avoid insecure options
- Two layers of defense

But: 49 pages, 68 MUSTs and MUST NOTs, > 50 other requirements and recommendations

OAuth 2.1 to the Rescue?

Security/
Hardening

OAuth 2.1 ≈ OAuth 2.0 + Security BCP

But:

- General-purpose profile, does not enforce high-security options
- Not an interoperability profile

Not Interoperable by Default

Interoperability

OAuth 2.x optionality

- grant types
- authentication methods
- security mechanisms
- cryptographic algorithms
- …

Bespoke solutions for common problems

- How to ask for complex consents?
- How to manage existing grants?
- How to achieve non-repudiation?

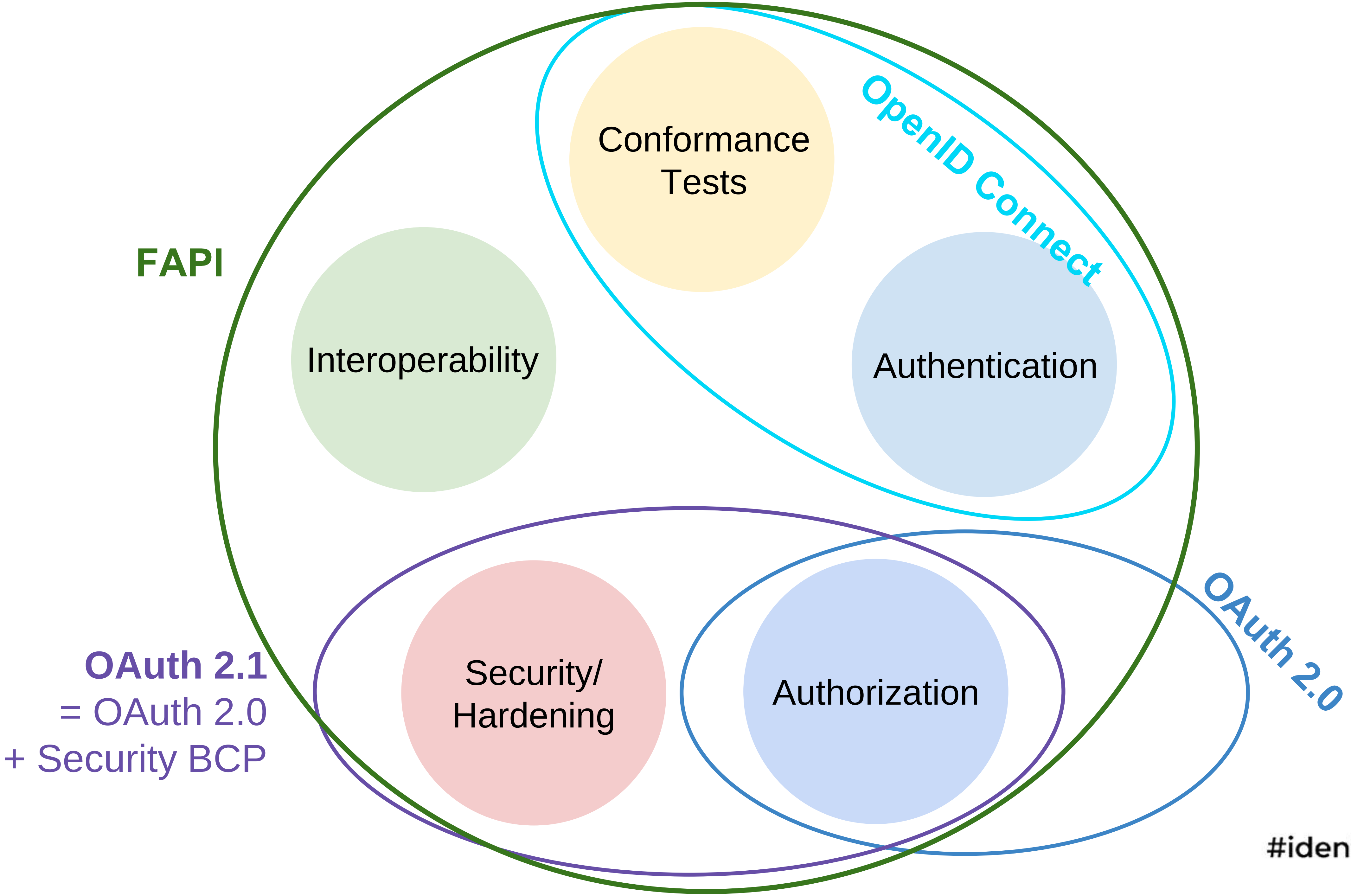
Does Everyone Follow the Rules?

Conformance Tests

(Only) testing ensures that a large-scale ecosystem actually works.

OpenID Connect has conformance tests.

But what about OAuth?



FAPI

OpenID Connect

OAuth 2.0

OAuth 2.1
= OAuth 2.0
+ Security BCP

Conformance Tests

Interoperability

Authentication

Security/ Hardening

Authorization

#identiverse

What is FAPI?

FAPI?

Financial API

FAPI?

~~Financial API~~

Financial API Security Profile

FAPI?

~~Financial API~~

~~Financial API Security Profile~~

Financial-*grade* API Security Profile

FAPI?

~~Financial API~~

~~Financial API Security Profile~~

~~Financial-*grade* API Security Profile~~

FAPI Security Profile

FAPI!

Security, interoperability, and feature profile for OAuth 2.0

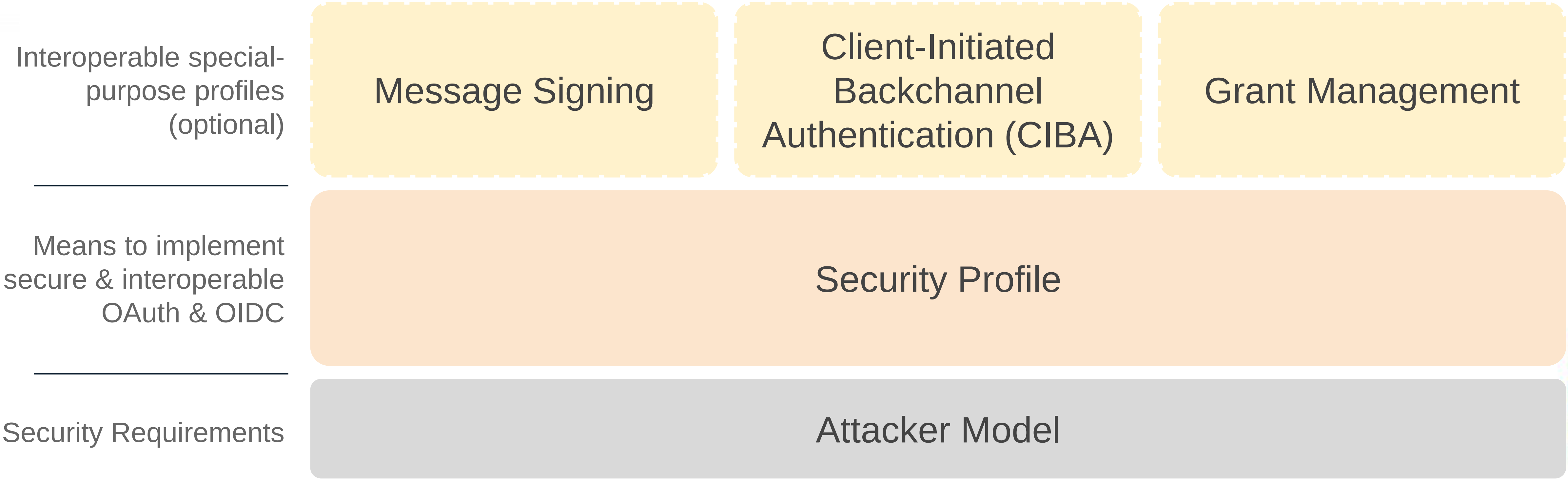
Usable for all high-security APIs:

- e-Signing
- e-Government
- Health
- …

FAPI 2.0 — Evolution of FAPI 1.0 based on industry experience:

- Improved security
- Improved interoperability
- Simplified development

FAPI 2.0 Specifications



Interoperable special-purpose profiles (optional)

Message Signing

Client-Initiated Backchannel Authentication (CIBA)

Grant Management

Means to implement secure & interoperable OAuth & OIDC

Security Profile

Security Requirements

Attacker Model

FAPI 2.0 Specifications

Defines the **security properties** that must be ensured and the **attacker capabilities** to protect against.

E.g., **Network Attacker** — has full control over the network.
Plus other strong attackers, e.g., with read access to the authorization request.

Not a threat model — threats can be derived from the attacker capabilities.

Interoper
pu

t Management

Means
secure &
O

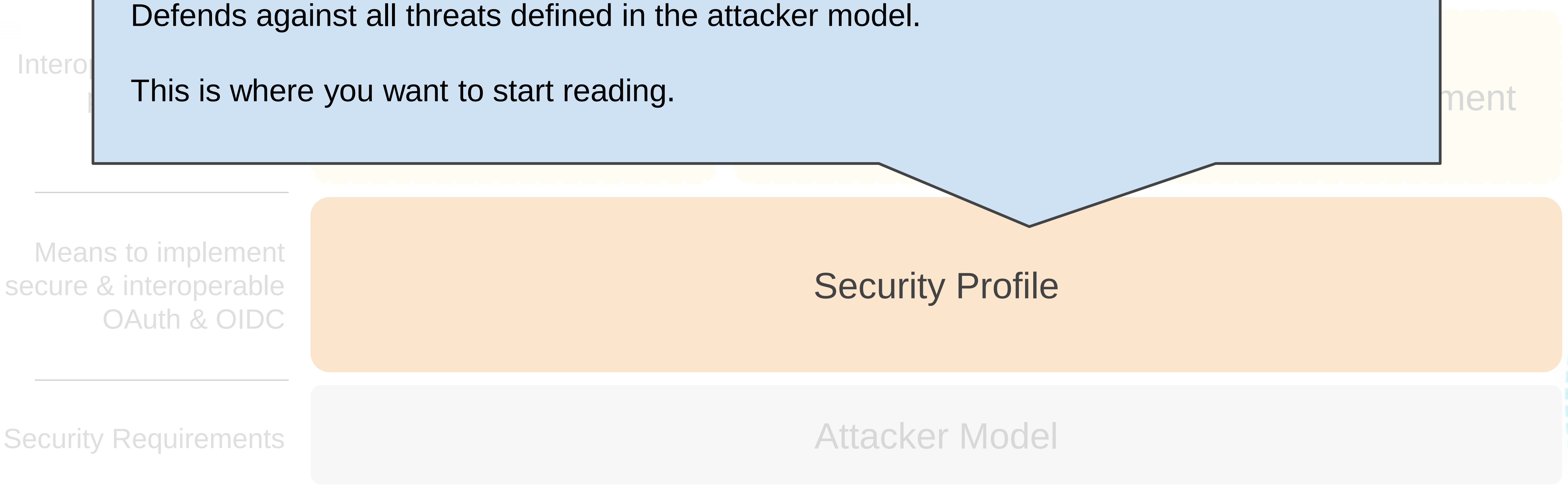
Security Requirements

Attacker Model

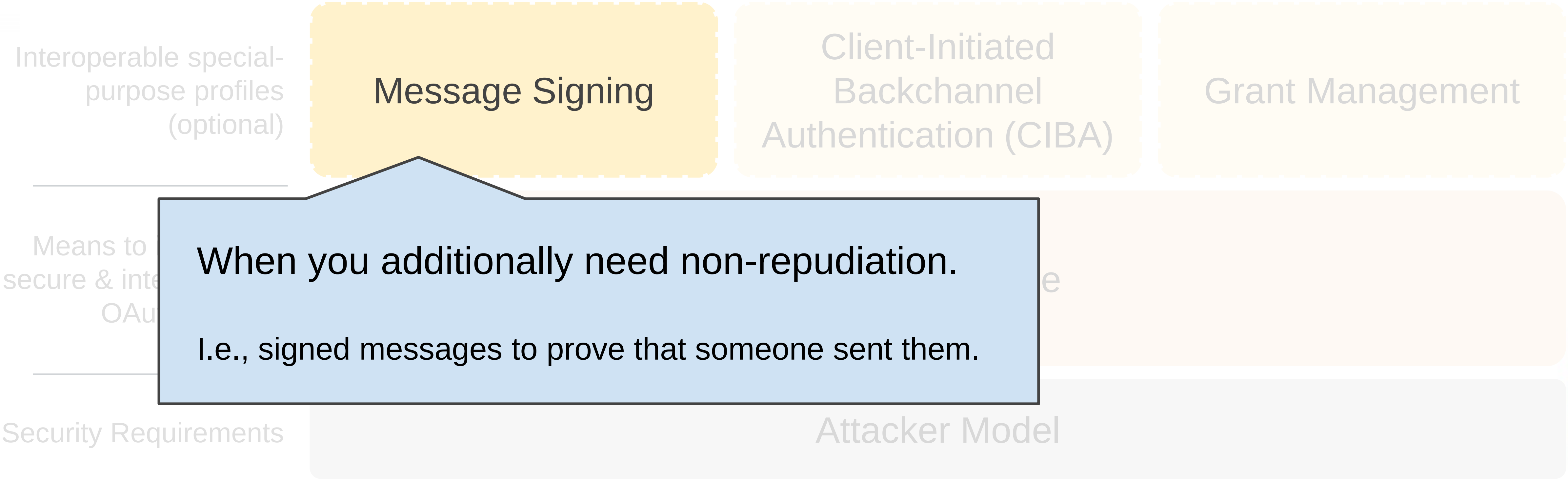
The Dos and Don’ts for secure, interoperable OAuth and OpenID Connect.

Defends against all threats defined in the attacker model.

This is where you want to start reading.



FAPI 2.0 Specifications



Interoperable special-purpose profiles (optional)

Message Signing

Client-Initiated Backchannel Authentication (CIBA)

Grant Management

Means to secure & inte OAu

e

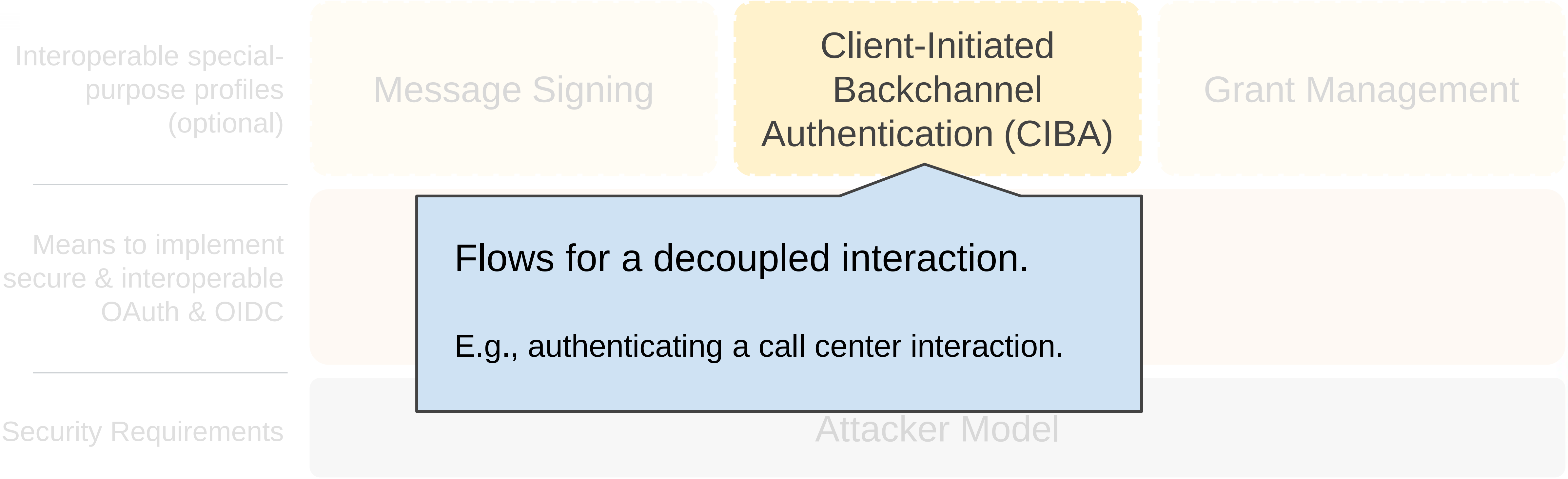
Security Requirements

Attacker Model

When you additionally need non-repudiation.

I.e., signed messages to prove that someone sent them.

FAPI 2.0 Specifications



Interoperable special-purpose profiles (optional)

Message Signing

Client-Initiated Backchannel Authentication (CIBA)

Grant Management

Means to implement secure & interoperable OAuth & OIDC

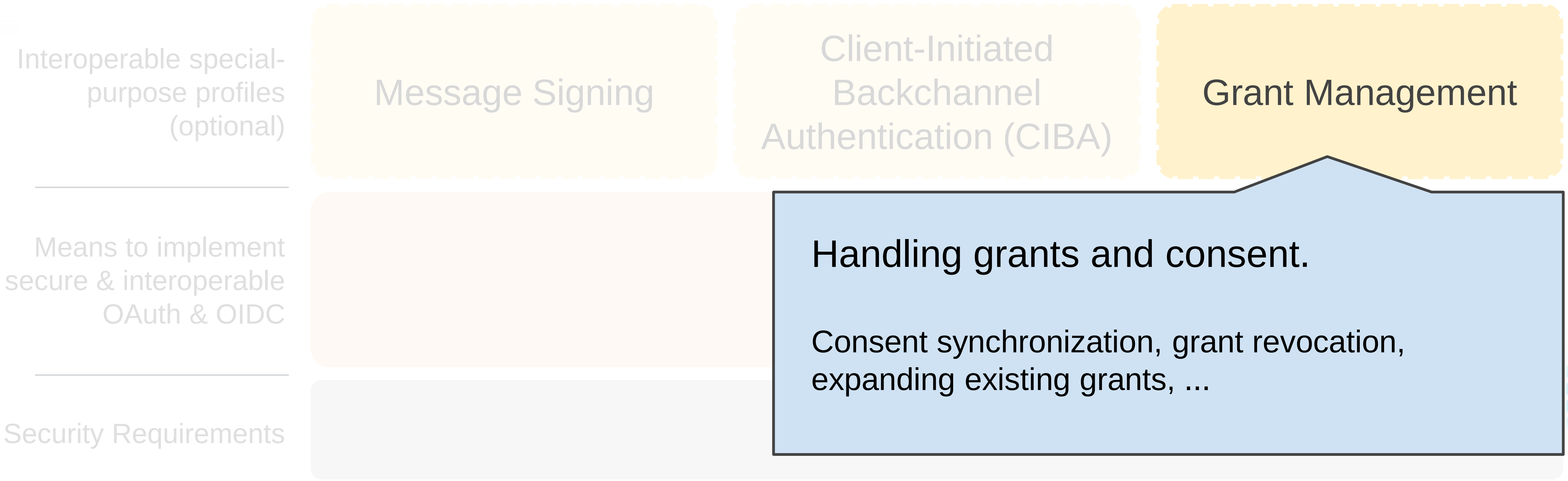
Flows for a decoupled interaction.

E.g., authenticating a call center interaction.

Security Requirements

Attacker Model

FAPI 2.0 Specifications



Interoperable special-purpose profiles (optional)

Means to implement secure & interoperable OAuth & OIDC

Security Requirements

Message Signing

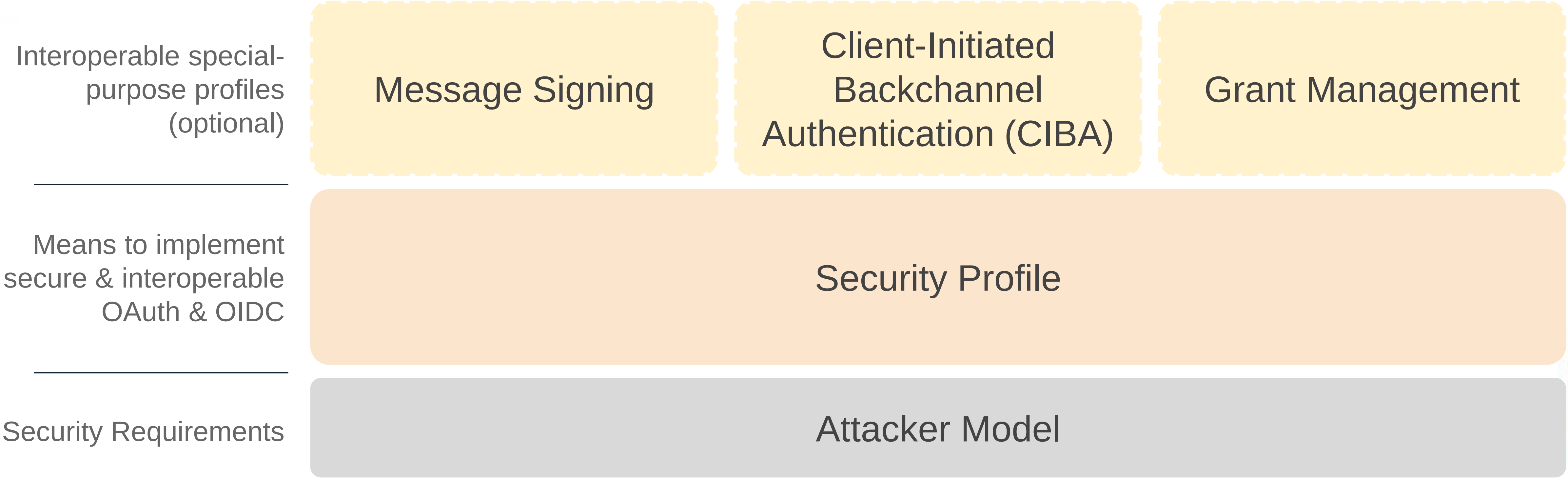
Client-Initiated Backchannel Authentication (CIBA)

Grant Management

Handling grants and consent.

Consent synchronization, grant revocation, expanding existing grants, ...

FAPI 2.0 Specifications



Interoperable special-purpose profiles (optional)

Message Signing

Client-Initiated Backchannel Authentication (CIBA)

Grant Management

Means to implement secure & interoperable OAuth & OIDC

Security Profile

Security Requirements

Attacker Model

FAPI 2.0 Deep-Dive

FAPI 2.0: Security Hardening

Security/
Hardening

OAuth Security Best Current Practice RFC incorporated.	Protect against redirect URIs manipulation, mix-up attacks, etc.
Disallow less secure options (e.g., implicit grant)	Avoid potential security issues
Pushed Authorization Requests to protect authorization request data	Ensure confidentiality and integrity of authorization request.
Sender-constrained access tokens via OAuth Mutual TLS or OAuth DPoP.	Prevent misuse of stolen tokens, provide defense-in-depth.

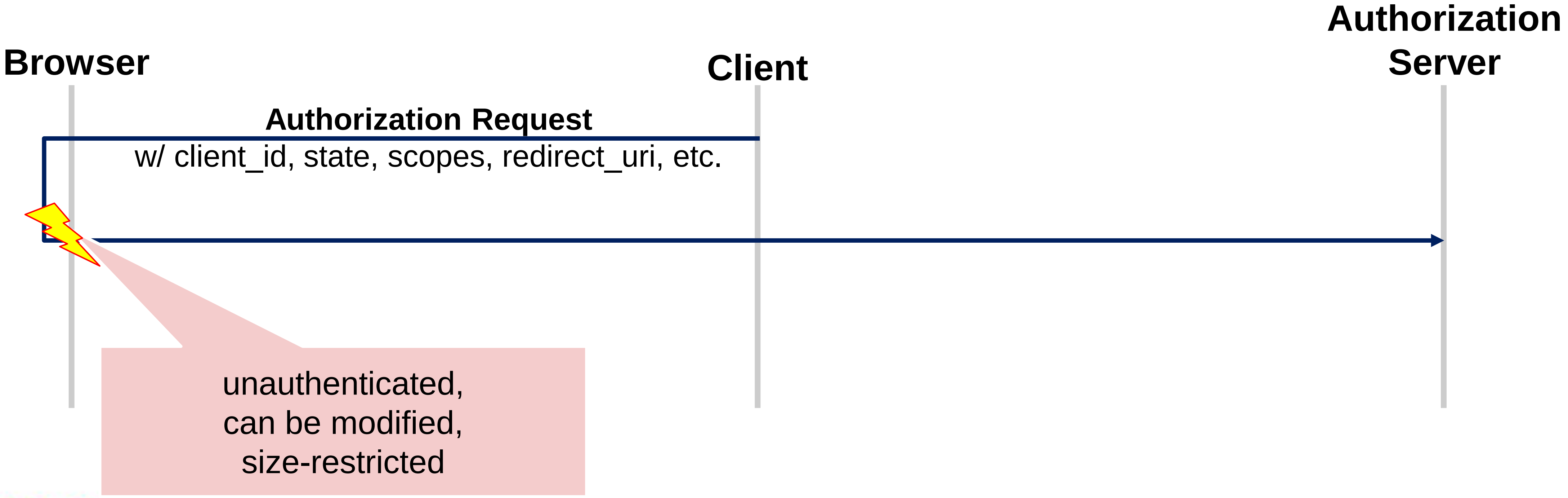
FAPI 2.0: Security Hardening

Security/
Hardening

OAuth Security Best Current Practice RFC incorporated.	Protect against redirect URIs manipulation, mix-up attacks, etc.
Disallow less secure options (e.g., implicit grant)	Avoid potential security issues
Pushed Authorization Requests to protect authorization request data	Ensure confidentiality and integrity of authorization request.
Sender-constrained access tokens via OAuth Mutual TLS or OAuth DPoP.	Prevent misuse of stolen tokens, provide defense-in-depth.

Pushed Authorization Requests (PAR)

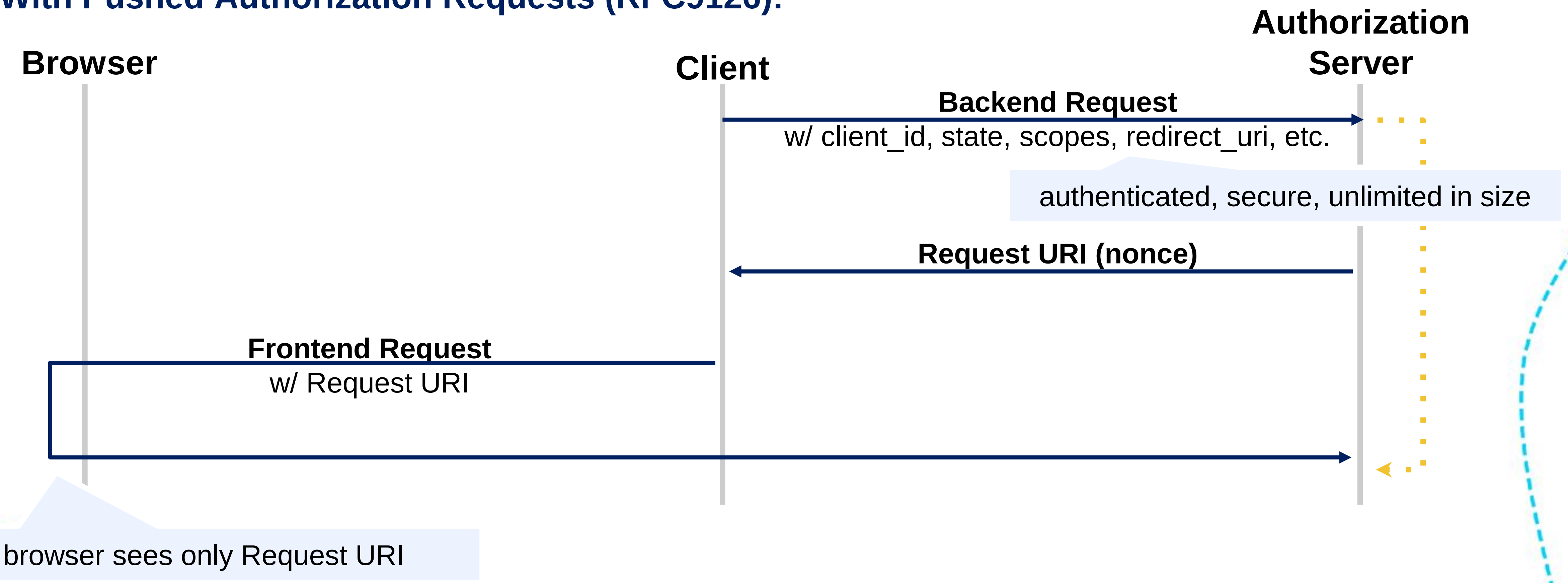
Traditional OAuth 2.x:


Browser
Client
Authorization Server
Authorization Request
w/ client_id, state, scopes, redirect_uri, etc.

unauthenticated,
can be modified,
size-restricted

Pushed Authorization Requests (PAR)

With Pushed Authorization Requests (RFC9126):



Browser

Client

Authorization Server

Backend Request
w/ client_id, state, scopes, redirect_uri, etc.

authenticated, secure, unlimited in size

Request URI (nonce)

Frontend Request
w/ Request URI

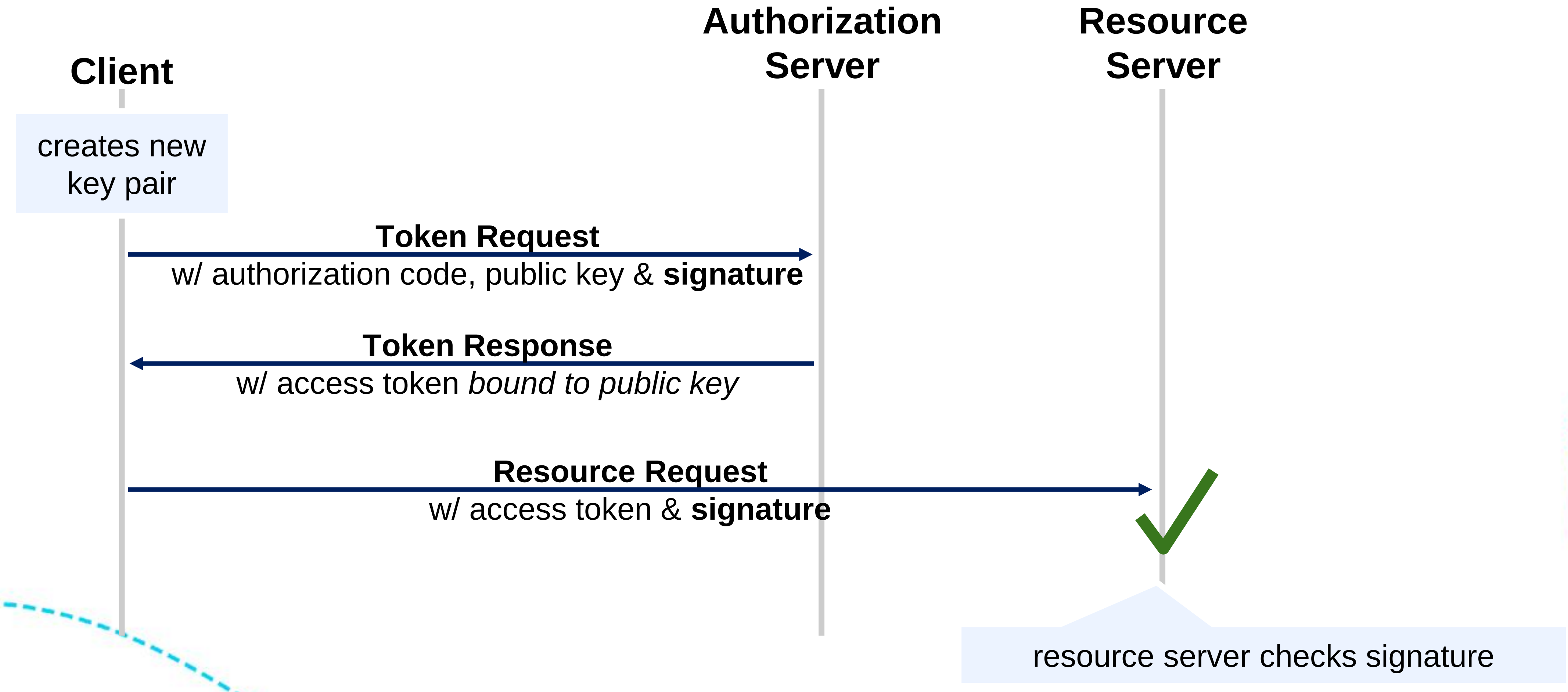
browser sees only Request URI

FAPI 2.0: Security Hardening

Security/
Hardening

OAuth Security Best Current Practice RFC incorporated.	Protect against redirect URIs manipulation, mix-up attacks, etc.
Disallow less secure options (e.g., implicit grant)	Avoid potential security issues
Pushed Authorization Requests to protect authorization request data	Ensure confidentiality and integrity of authorization request.
Sender-constrained access tokens via OAuth Mutual TLS or OAuth DPoP.	Prevent misuse of stolen tokens, provide defense-in-depth.

Sender-Constrained Access Tokens



Client

Authorization Server

Resource Server

creates new key pair

Token Request
w/ authorization code, public key & **signature**

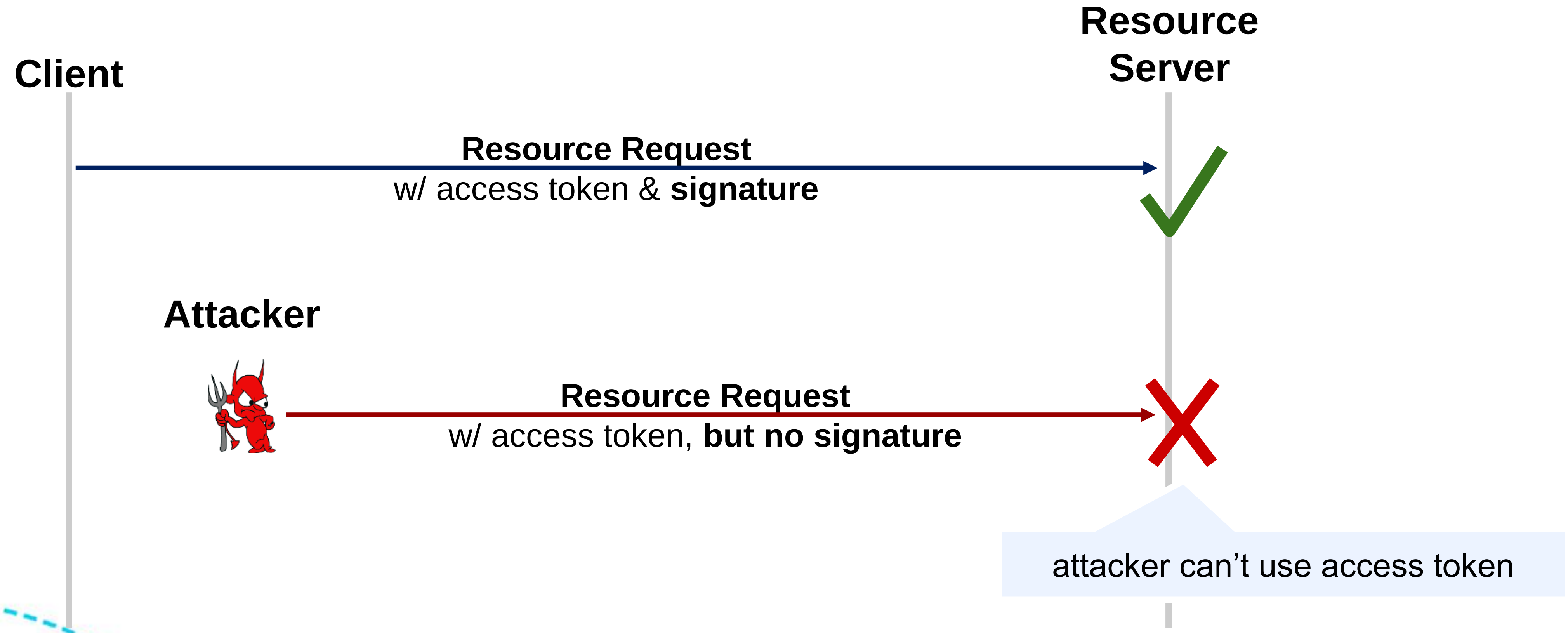
Token Response
w/ access token *bound to public key*

Resource Request
w/ access token & **signature**

resource server checks signature

Sender-Constrained Access Tokens

What if the access token leaks?



Resource Server

Client

Resource Request
w/ access token & **signature**

Attacker

Resource Request
w/ access token, **but no signature**

attacker can’t use access token

Choose Your Flavor

DPoP	Mutual TLS (MTLS)
Application layer	Network layer
Headers w/ signature over request URI	TLS client authentication
JWK key pairs	X.509 certificates (can be self-signed)
Can be used for web app clients	Request fully protected
No integration on network layer needed	Can be used for client authentication

FAPI 2.0: Security Hardening

Security/
Hardening

OAuth Security Best Current Practice RFC incorporated.

Protect against redirect URIs manipulation, mix-up attacks, etc.

Disallow less secure options (e.g., implicit grant)

Avoid potential security issues

Pushed Authorization Requests to protect authorization request data

Ensure confidentiality and integrity of authorization request.

Sender-constrained access tokens via OAuth Mutual TLS or OAuth DPoP.

Prevent misuse of stolen tokens, provide defense-in-depth.

FAPI 2.0: Security Hardening

Security/
Hardening

Asymmetric client authentication instead of client secrets.

Robust client authentication.

High-security cryptographic algorithms, TLS recommendations, …

Secure encryption, signing,
and well-protected network layer.

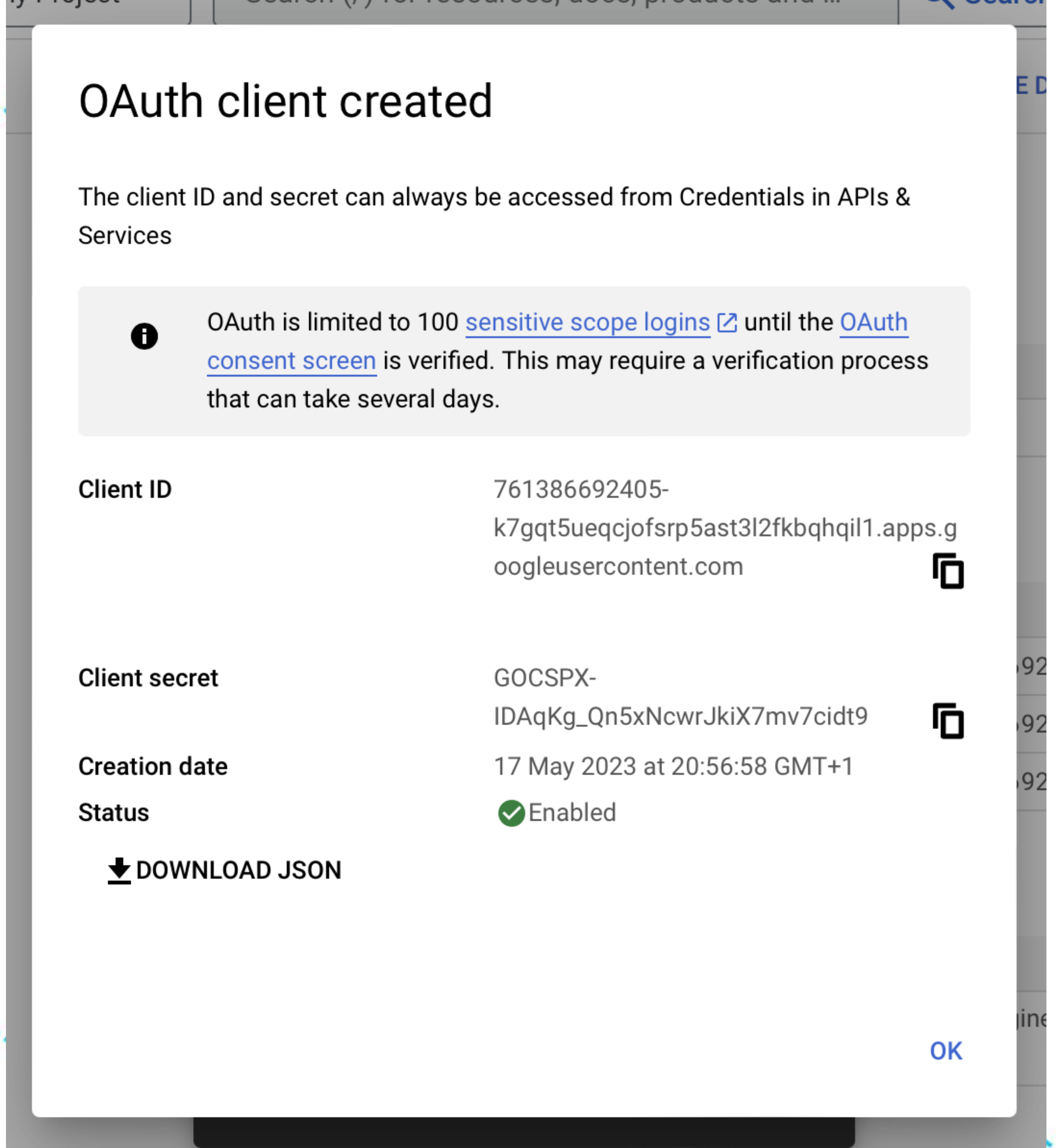
Require use of PKCE

Protect authorization codes
even when stolen.

FAPI 2.0: Security Hardening

Security/ Hardening

Asymmetric client authentication instead of client secrets.	Robust client authentication.
High-security cryptographic algorithms, TLS recommendations, …	Secure encryption, signing, and well-protected network layer.
Require use of PKCE	Protect authorization codes even when stolen.



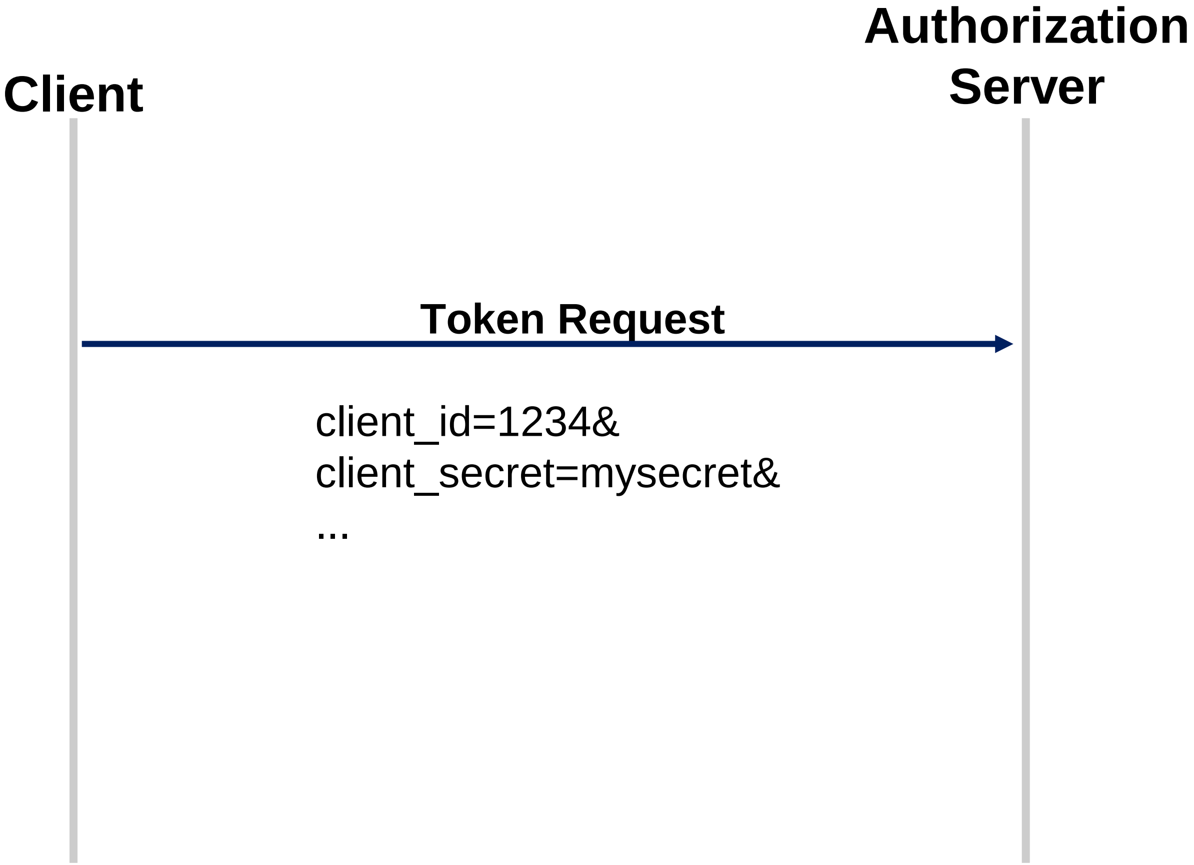
OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth is limited to 100 sensitive scope logins until the OAuth consent screen is verified. This may require a verification process that can take several days.

Client ID	761386692405-k7gqt5ueqcjofsrp5ast3l2fkbqhqil1.apps.googleusercontent.com
Client secret	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
Creation date	17 May 2023 at 20:56:58 GMT+1
Status	✓ Enabled

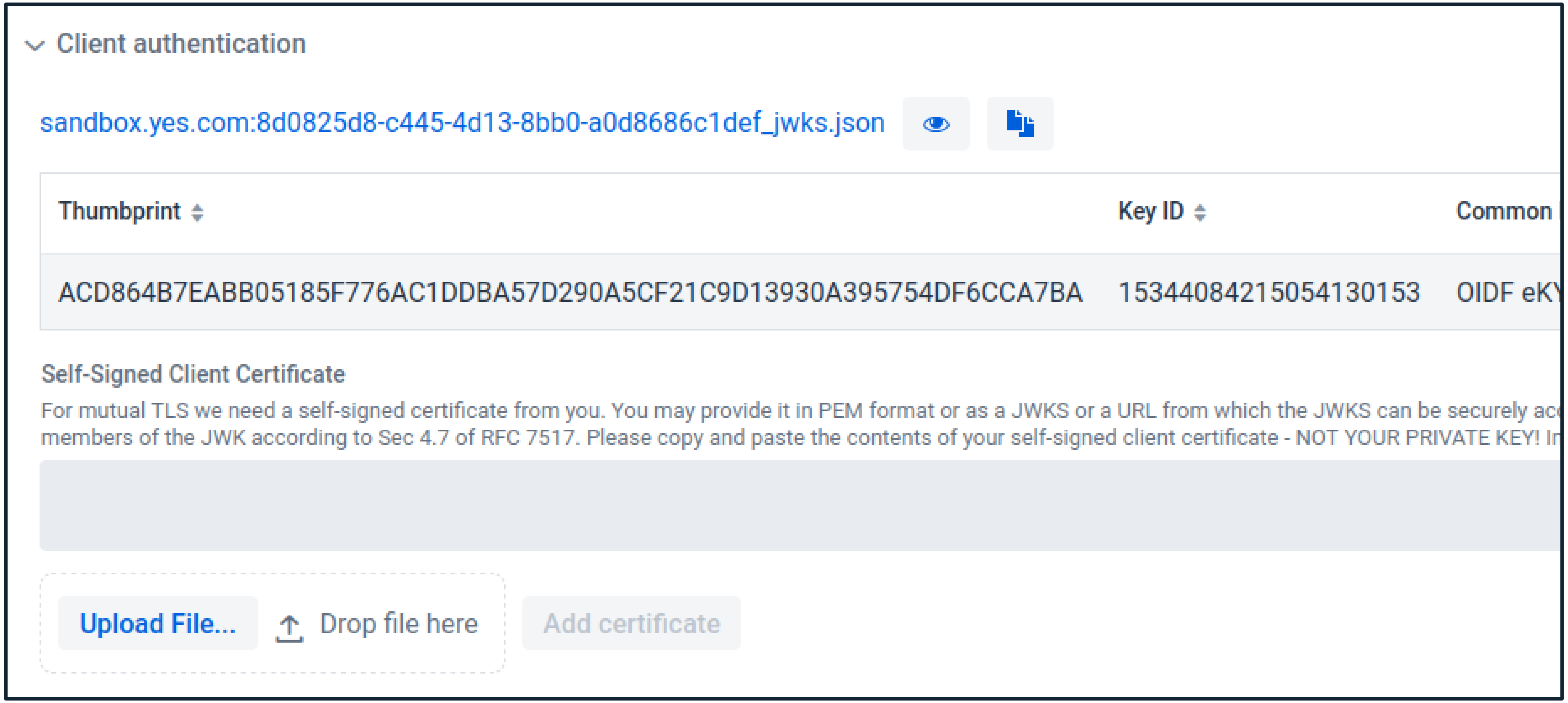
Client Authentication: client_secret_post



client_id=1234&
client_secret=mysecret&
...

Client Authentication: Choose Your Flavor

private_key_jwt	Mutual TLS (MTLS)
Application layer	Network layer
Signed JWT	TLS client authentication
JWK key pairs	X.509 certificates (can be self-signed)
Can be used for web app clients	Request fully protected
No integration on network layer needed	Can be used for client authentication



Client authentication

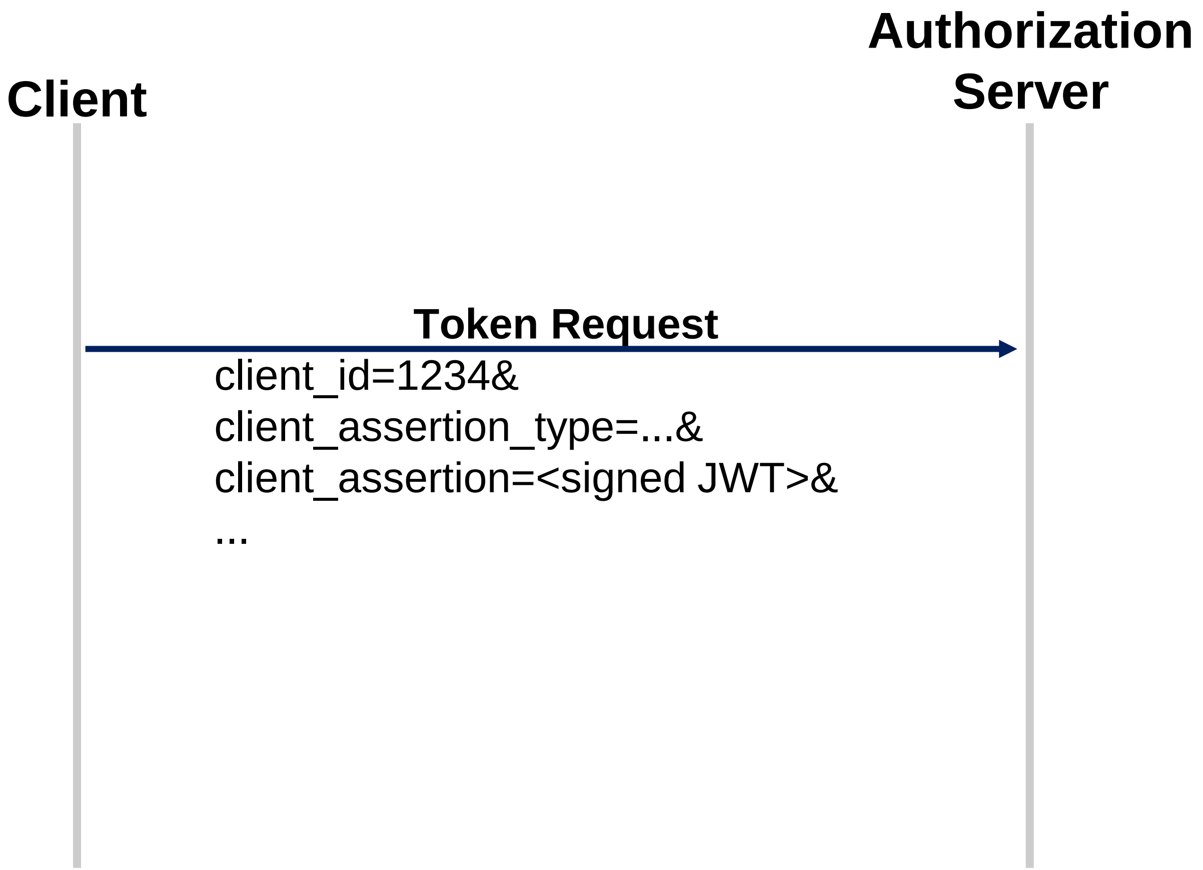
sandbox.yes.com:8d0825d8-c445-4d13-8bb0-a0d8686c1def_jwks.json

Thumbprint	Key ID	Common
ACD864B7EABB05185F776AC1DDBA57D290A5CF21C9D13930A395754DF6CCA7BA	1534408421505413O153	OIDF eKY

Self-Signed Client Certificate

For mutual TLS we need a self-signed certificate from you. You may provide it in PEM format or as a JWKS or a URL from which the JWKS can be securely ac
members of the JWK according to Sec 4.7 of RFC 7517. Please copy and paste the contents of your self-signed client certificate - NOT YOUR PRIVATE KEY! In

Client Authentication: private_key_jwt



Client

Authorization Server

Token Request

client_id=1234&
client_assertion_type=...&
client_assertion=<signed JWT>&
...

FAPI 2.0: Security Hardening

Security/ Hardening

Asymmetric client authentication instead of client secrets.	Robust client authentication.
High-security cryptographic algorithms, TLS recommendations, …	Secure encryption, signing, and well-protected network layer.
Require use of PKCE	Protect authorization codes even when stolen.

Security: We Didn't Wing It!

Security/
Hardening

Formal protocol security analysis
by University of Stuttgart, Germany researchers to protect against flaws in the protocol.

→ Well-understood security properties based on **attacker model**.

FAPI 2.0: Interoperability

Interoperability

Reduced protocol options

Ensure on-the-wire interoperability.

Pushed Authorization Requests (PAR)

Replace bespoke solutions like *authorization resources*,
ensure interoperability and security,
minimize data in front-channel.

FAPI 2.0: Conformance Tests

Conformance Tests

In-depth conformance testing	Facilitate interoperability in large-scale ecosystems
Official OpenID Foundation certification program	Ensure compatibility of software & solutions

Current Status & Adoption

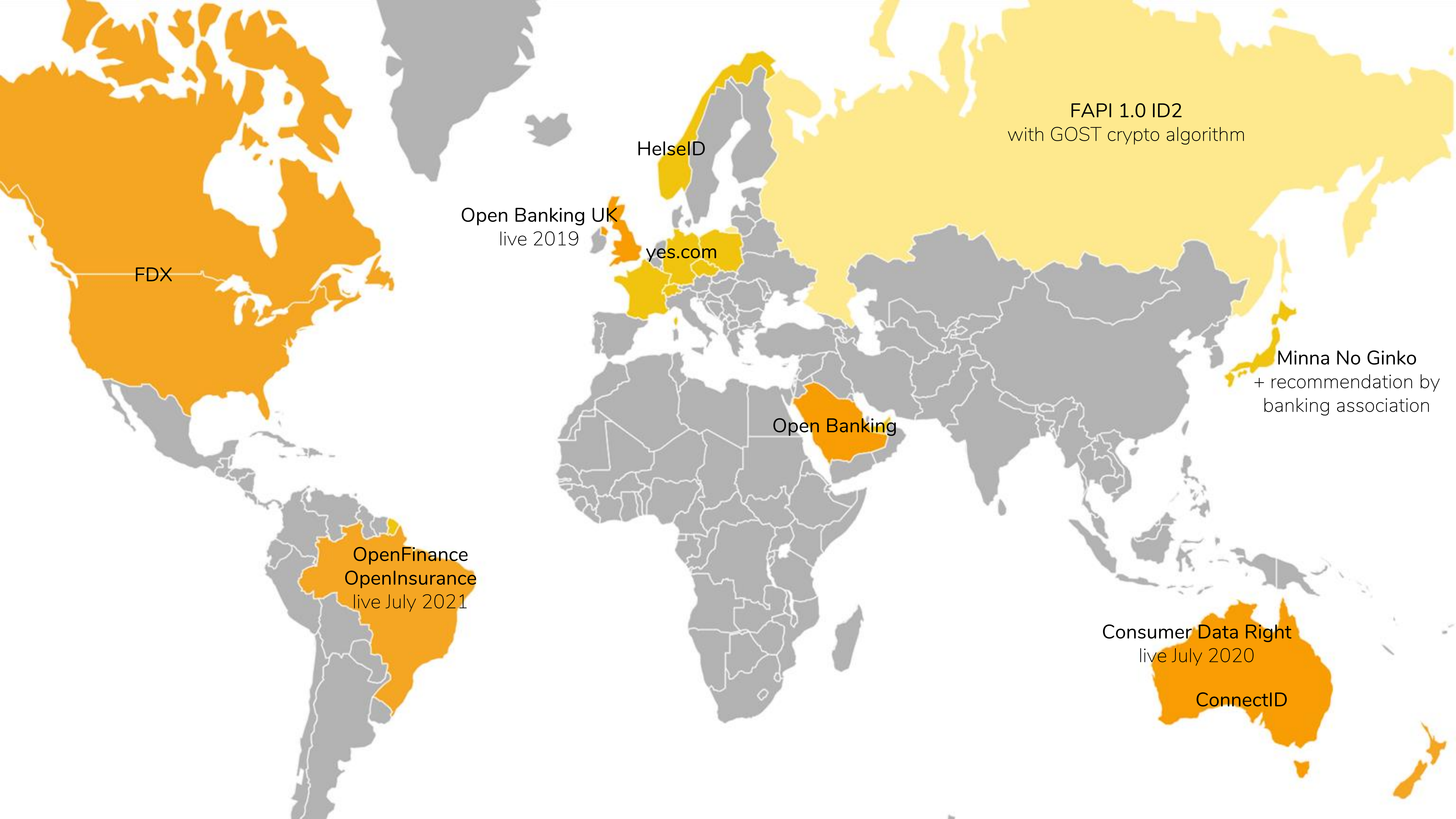
Is FAPI 2.0 ready to use?

Yes!

All specifications have reached “implementer’s draft”

- Stable numbered version of the specification
- Implementer’s drafts are never changed
- IPR protection

FAPI 2.0 Security Profile “Final” due around end of 2023


FAPI 1.0 ID2
with GOST crypto algorithm
HelseID
Open Banking UK
live 2019
yes.com
FDX
Minna No Ginko
+ recommendation by
banking association
Open Banking
OpenFinance
OpenInsurance
live July 2021
Consumer Data Right
live July 2020
ConnectID

FAPI 2.0 Everywhere?

FAPI 2.0 is a shortcut towards state-of-the-art security & interoperability for all kinds of APIs.

Only for short-lived tokens in lower-security applications, FAPI 2.0 might be too much.

High-security & interoperable OAuth 2?

FAPI 2.0 is the 'batteries included' spec for high-security ecosystems:

- Latest security recommendations
- On-the-wire interoperability
- Comprehensive conformance testing
- Feature-rich extensions
- Growing world-wide adoption

THANK YOU!
Questions?