The Evolving Threat Landscape of OAuth

Securing the Backbone of Modern Authn/Authz

Daniel Fett

About me: Dr. Daniel Fett

- PhD on web protocol security (formal security analysis)
- Contributor to open web standards (IETF OAuth, OpenID Foundation)
  - Best Current Practice for OAuth Security (RFC9700)
  - DPoP (RFC9449)
  - OpenID FAPI
  - OpenID for Verifiable Credentials
  - SD-JWT
- Product owner in the German EUDI Wallet project @ SPRIN-D

In this Talk

What is OAuth 2.0? Quick recap!

Security Challenges for “classic” OAuth & how to address them

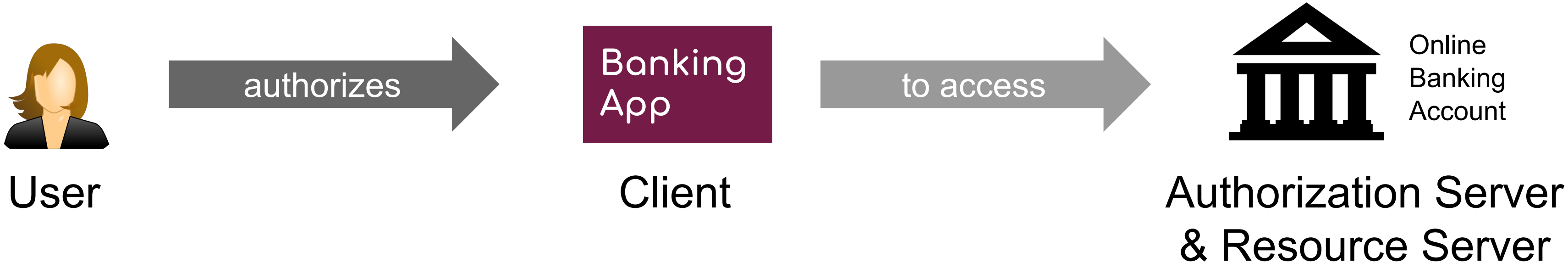
The future of identity ecosystems and new threats

Who is familiar with OAuth?

OAuth 2.0

OAuth is a standard
for federated authorization

Authorization



User

authorizes

Banking App

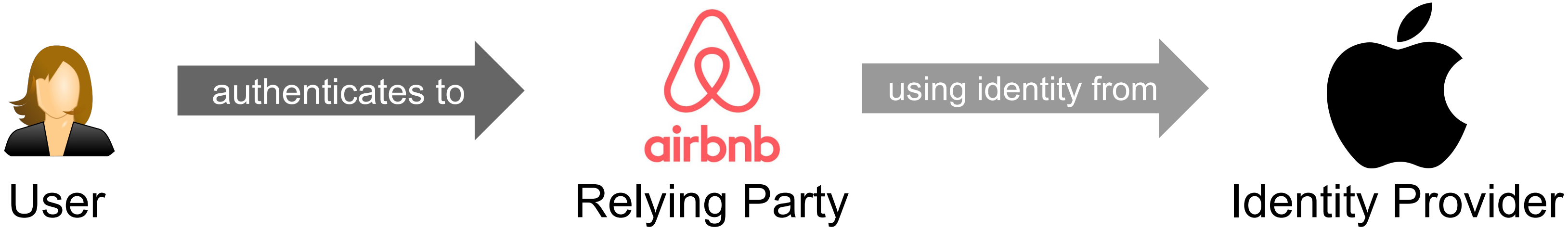
Client

to access

Online Banking Account

Authorization Server & Resource Server

Authentication



User

authenticates to

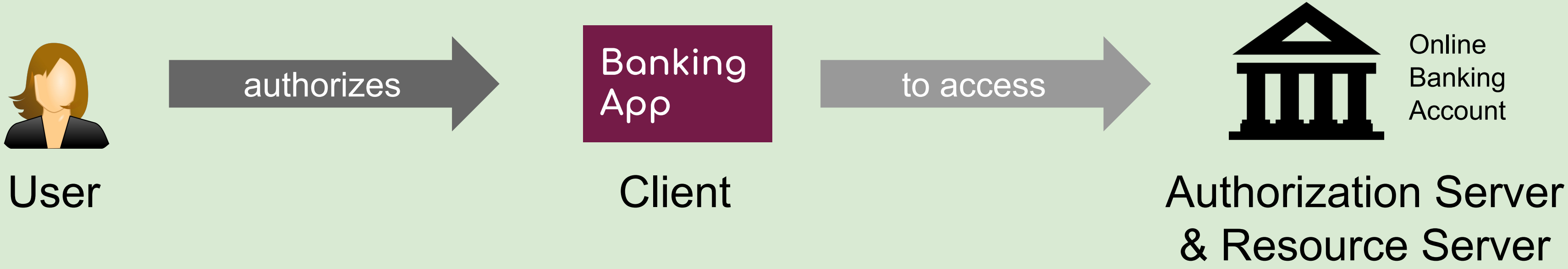
airbnb

Relying Party

using identity from

Identity Provider

Authorization (OAuth)



User

authorizes

Banking App

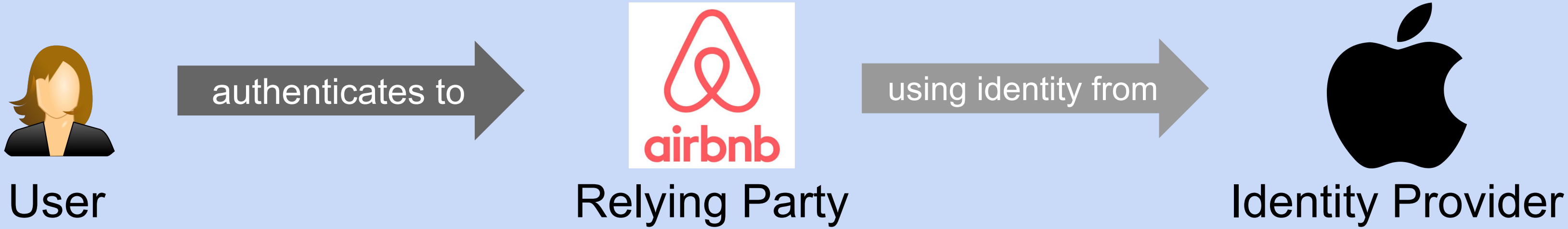
Client

to access

Online Banking Account

Authorization Server & Resource Server

Authentication (OpenID Connect)



User

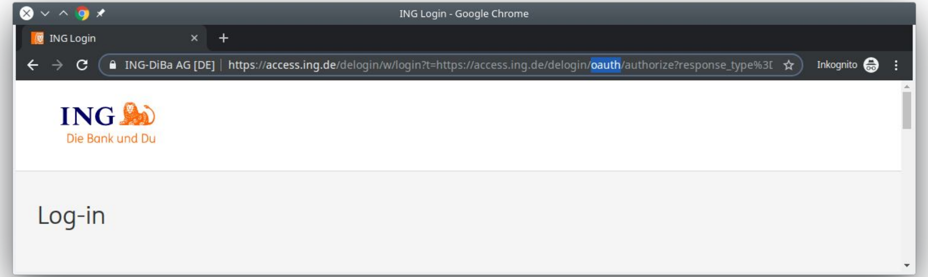
authenticates to

Relying Party

using identity from

Identity Provider

OAuth & friends in the Wild



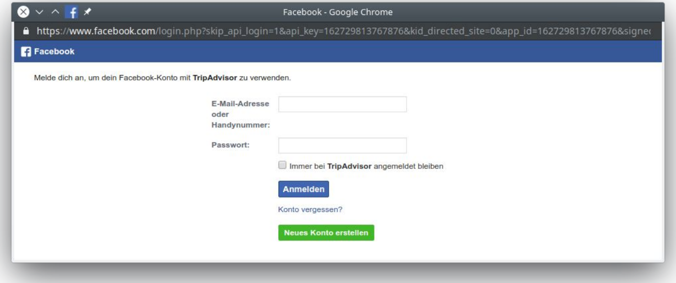
ING Login - Google Chrome

ING Login

ING-DiBa AG [DE] | https://access.ing.de/delogin/w/login?t=https://access.ing.de/delogin/oauth/authorize?response_type%3D

Log-in

Banking



Facebook - Google Chrome

https://www.facebook.com/login.php?skip_api_login=1&api_key=162729813767876&kid_directed_site=0&app_id=162729813767876&signe

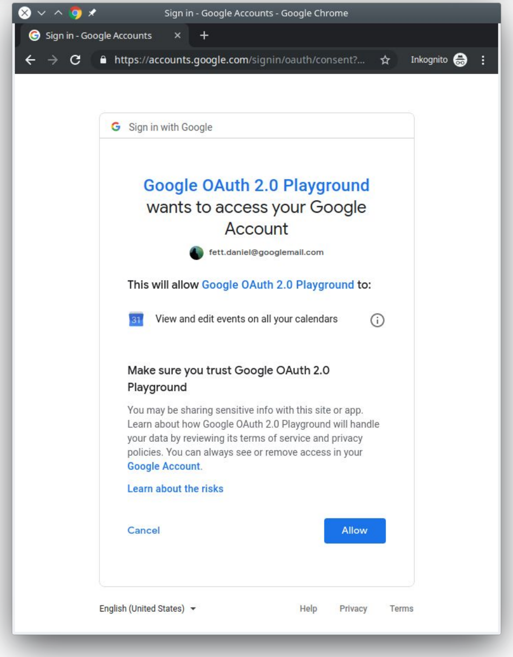
Melde dich an, um dein Facebook-Konto mit **TripAdvisor** zu verwenden.

E-Mail-Adresse oder Handynummer:

Passwort:

□ Immer bei **TripAdvisor** angemeldet bleiben

Facebook



Sign in - Google Accounts - Google Chrome

Sign in - Google Accounts

https://accounts.google.com/signin/oauth/consent?...

Google OAuth 2.0 Playground wants to access your Google Account

fett.daniel@googlemail.com

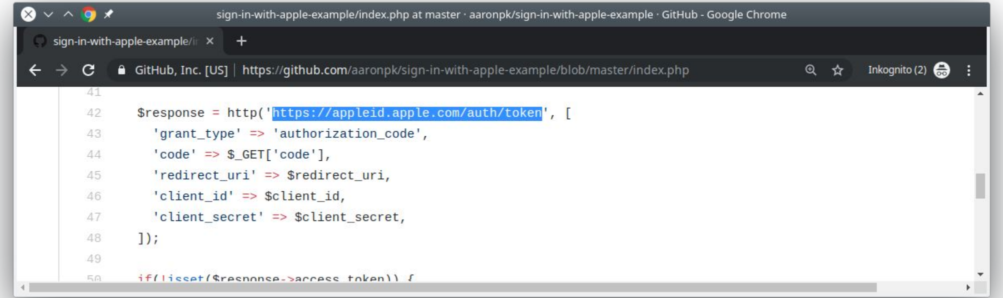
This will allow Google OAuth 2.0 Playground to:

View and edit events on all your calendars

Make sure you trust Google OAuth 2.0 Playground

You may be sharing sensitive info with this site or app. Learn about how Google OAuth 2.0 Playground will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Google



sign-in-with-apple-example/index.php at master · aaronpk/sign-in-with-apple-example · GitHub - Google Chrome

GitHub, Inc. [US] | https://github.com/aaronpk/sign-in-with-apple-example/blob/master/index.php

```
$response = http('https://appleid.apple.com/auth/token', [
  'grant_type' => 'authorization_code',
  'code' => $_GET['code'],
  'redirect_uri' => $redirect_uri,
  'client_id' => $client_id,
  'client_secret' => $client_secret,
]);

if(!isset($response->access_token)) {
```

Apple

e-health

open banking

e-signing

open insurance

OAuth 2.0!

OAuth & friends in the Wild

ING

Log-in

Banking

Facebook

```
$response = http('https://appleid.apple.com/auth/token', [
  'grant_type' => 'authorization_code',
  'code' => $_GET['code'],
  'redirect_uri' => $redirect_uri,
  'client_id' => $client_id,
  'client_secret' => $client_secret,
]);
```

Apple

Google OAuth 2.0 Playground wants to access your Google Account

This will allow Google OAuth 2.0 Playground to:

View and edit events on all your calendars

Make sure you trust Google OAuth 2.0 Playground

Cancel Allow

Google

open finance

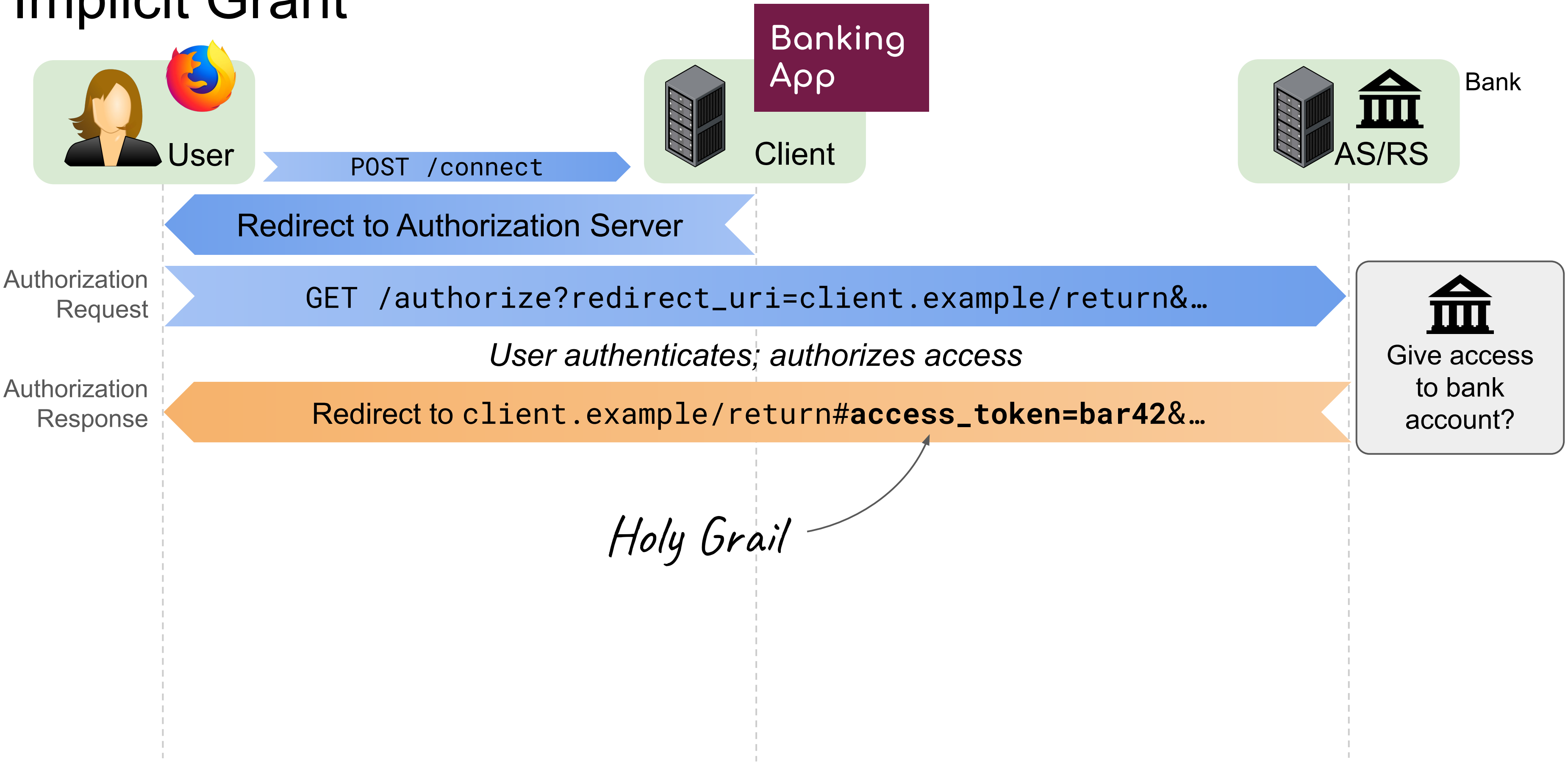
e-government

open consumer data

digital identity ecosystems

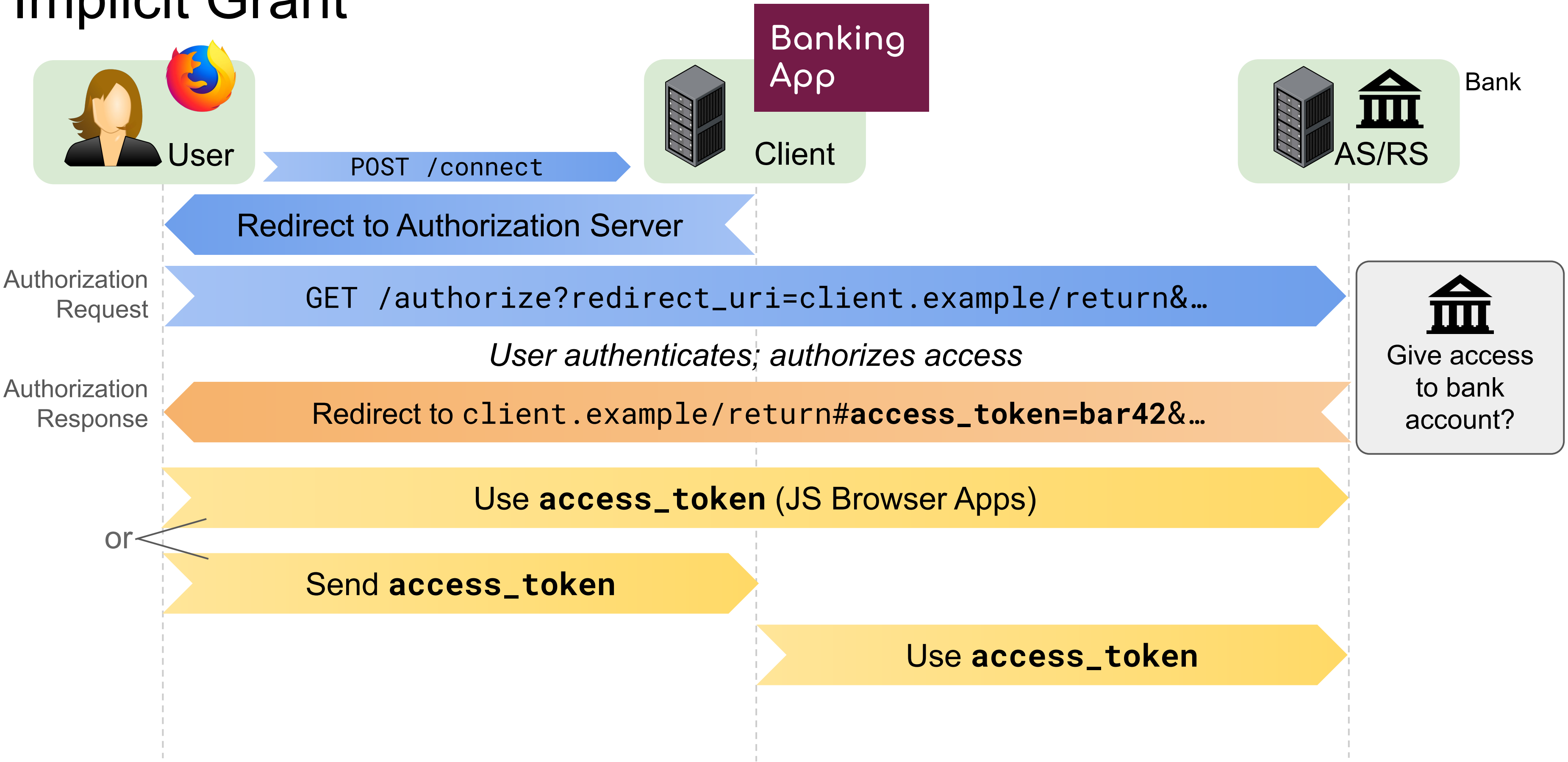
OAuth from 10.000 feet

Implicit Grant



Banking App
User
Client
Bank
AS/RS
POST /connect
Redirect to Authorization Server
Authorization Request
GET /authorize?redirect_uri=client.example/return&…
Give access to bank account?
User authenticates; authorizes access
Authorization Response
Redirect to client.example/return#**access_token=bar42**&…
Holy Grail

Implicit Grant



User

Banking App

Client

Bank

AS/RS

POST /connect

Redirect to Authorization Server

Authorization Request

GET /authorize?redirect_uri=client.example/return&…

User authenticates; authorizes access

Give access to bank account?

Authorization Response

Redirect to client.example/return#**access_token=bar42**&…

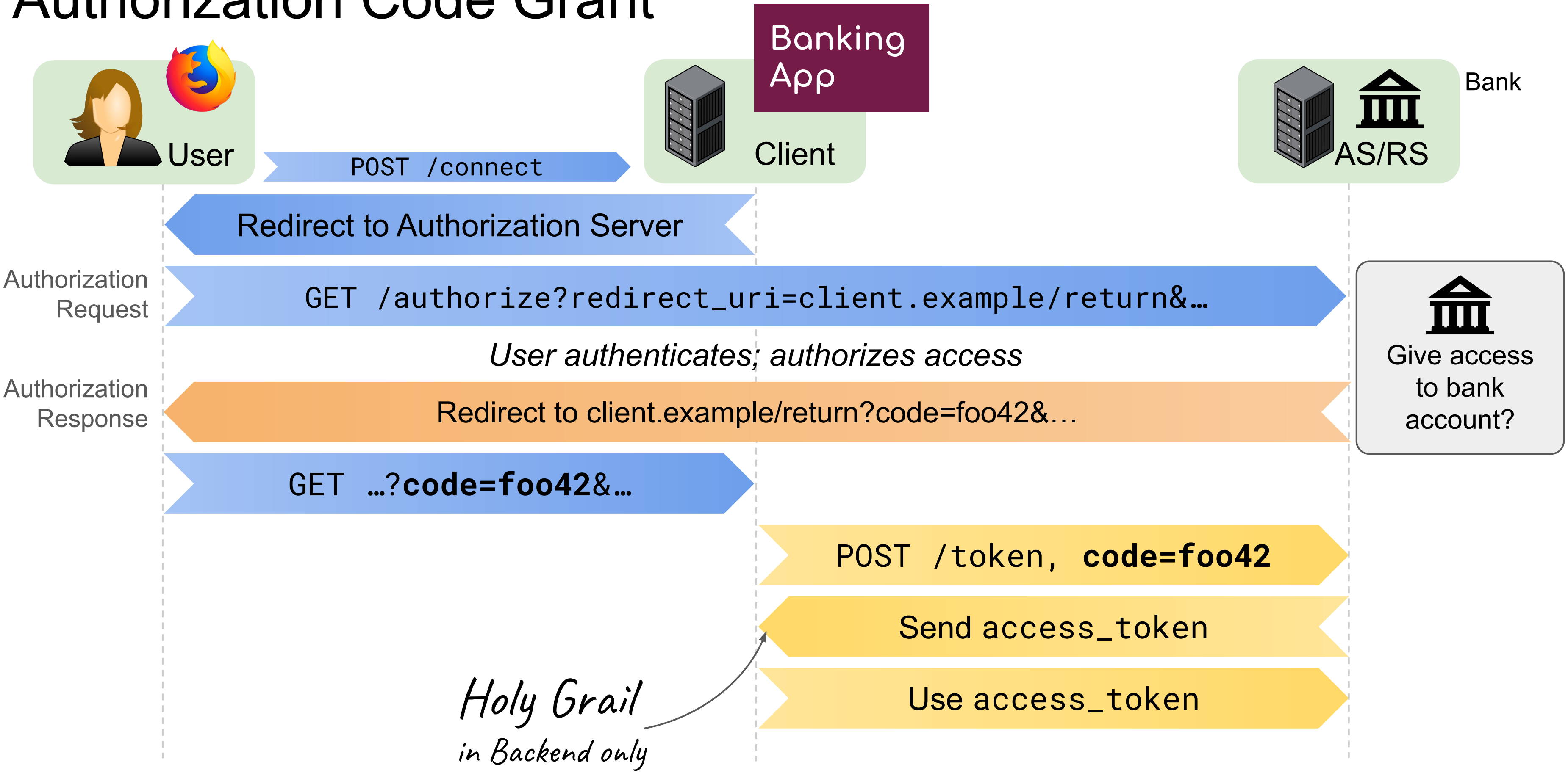
Use **access_token** (JS Browser Apps)

or

Send **access_token**

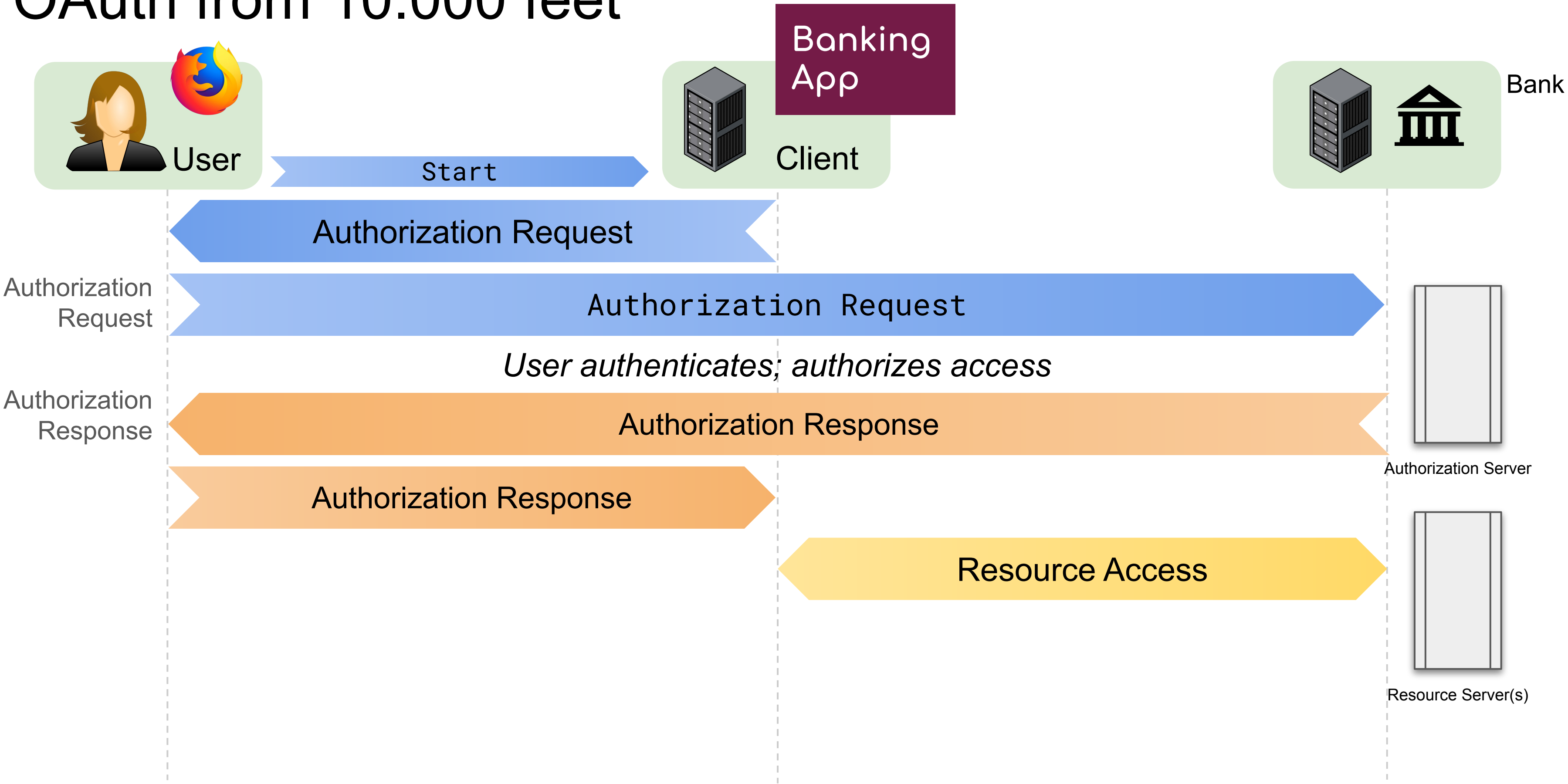
Use **access_token**

Authorization Code Grant



Banking App
User
Client
Bank
AS/RS
POST /connect
Redirect to Authorization Server
Authorization Request
GET /authorize?redirect_uri=client.example/return&…
User authenticates; authorizes access
Give access to bank account?
Authorization Response
Redirect to client.example/return?code=foo42&…
GET …?code=foo42&…
POST /token, code=foo42
Send access_token
Use access_token
Holy Grail
in Backend only

OAuth from 10.000 feet



Security Challenges for classic OAuth

Challenge 1: Implementation Flaws

- We still see many implementation flaws
- Known anti-patterns are still used
  - Insufficient redirect URI checking (code/token is redirected to attacker)
  - `state` parameter is not used properly to defend against CSRF
  - …
- Clients worse than authorization/resource servers

- [Li et al., 2014]
  60 chinese clients, **more than half** vulnerable to CSRF
- [Yang et al., 2016]
  Out of 405 clients, **55%** do not handle `state` (CSRF protection) correctly
- [Shebab et al., 2015]
  25% of OAuth clients in Alexa Top 10000 vulnerable to CSRF
- [Chen et al., 2014]
  89 of 149 mobile clients vulnerable to one or more attacks
- [Wang et al., 2013]
  Vulnerabilities in Facebook PHP SDK and other OAuth SDKs
- [Sun et al., 2012]
  96 Clients, **almost all** vulnerable to one or more attacks

Challenge 2: High-Stakes Environments

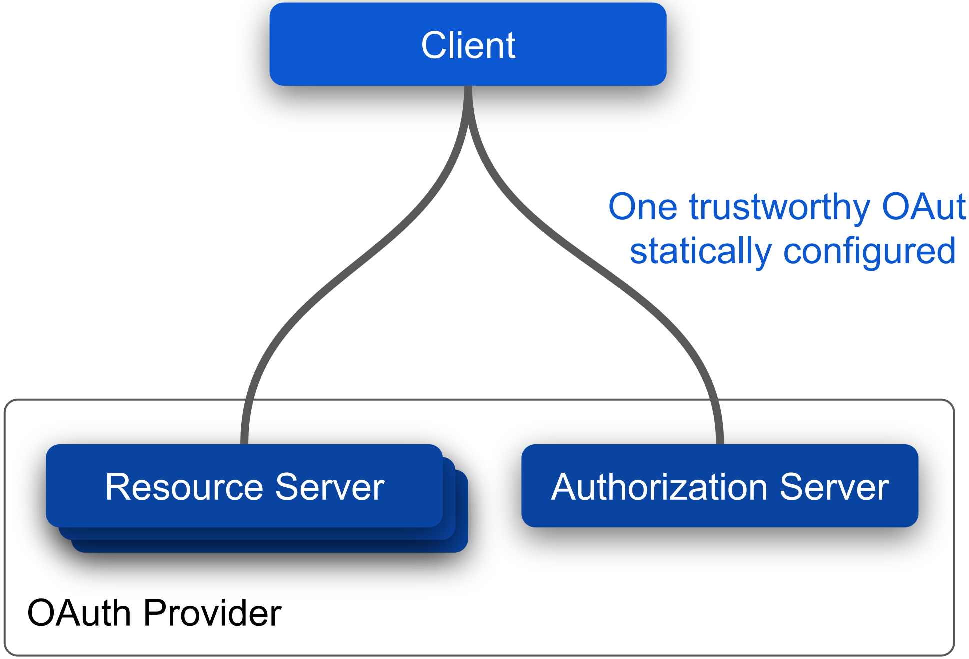
New use cases require a very high level of security

- **Open Banking:** Account access, payments, wire transfers
- **eHealth:** Access to health data
- **eSigning:** Legally binding digital signatures
- **Wallets (EU Digital Identity Wallets, eIDAS 2.0):**
  Identification on *Level of Assurance High*

Far beyond the scope of the original security threat model!

Challenge 3: Large-scale Open Ecosystems

Originally anticipated:



One trustworthy OAuth provider, statically configured per client

Challenge 3: Large-scale Open Ecosystems

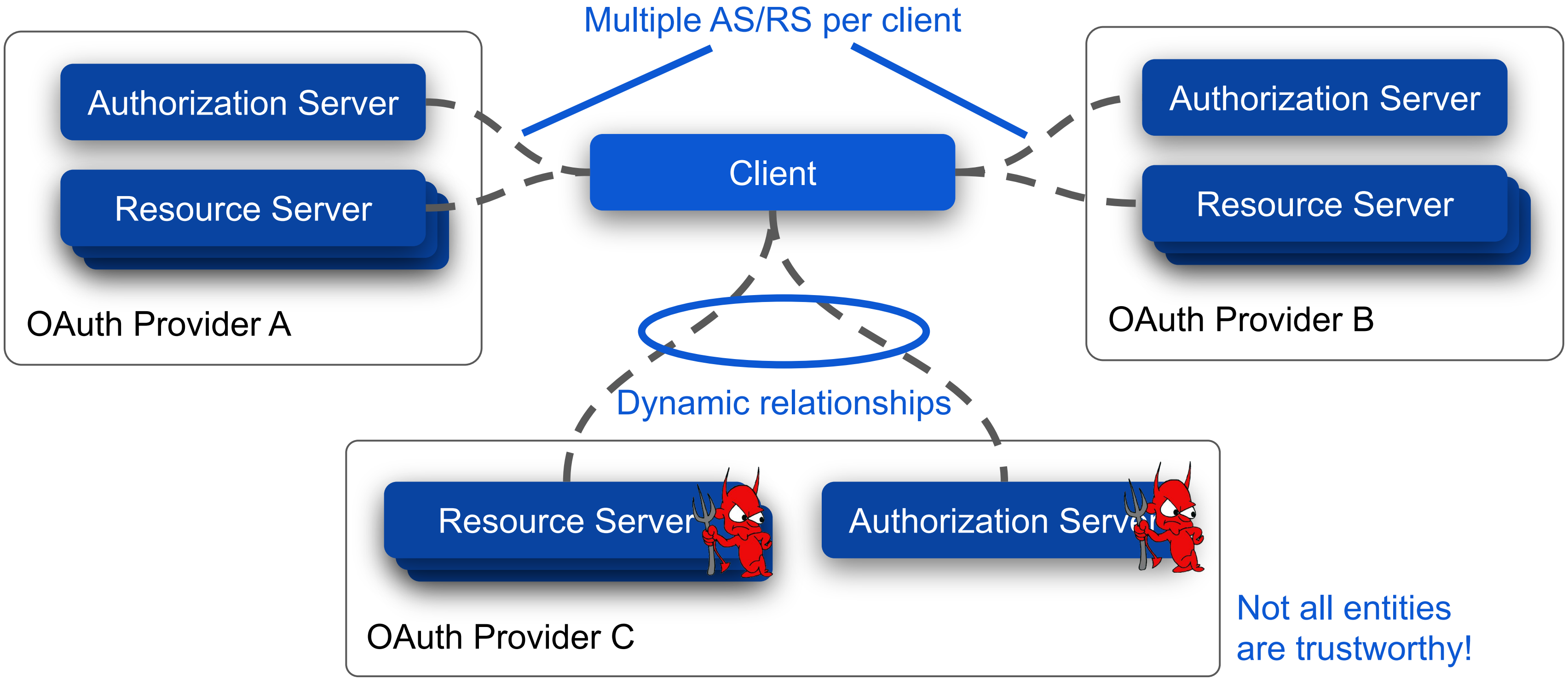
Recent examples:

- Payment Services Directive 2
  - Open banking interface required for european banks
  - > 5000 banks in europe
  - Similar initiatives all over the world
  - One client - thousands of potential OAuth providers

- MCP - Model Context Protocol
  - Open protocol to connect AI models to different data sources and tools
  - Dozens of servers publicly available

Challenge 3: Large-scale Open Ecosystems

Today:



Multiple AS/RS per client

Authorization Server

Resource Server

OAuth Provider A

Client

Authorization Server

Resource Server

OAuth Provider B

Dynamic relationships

Resource Server

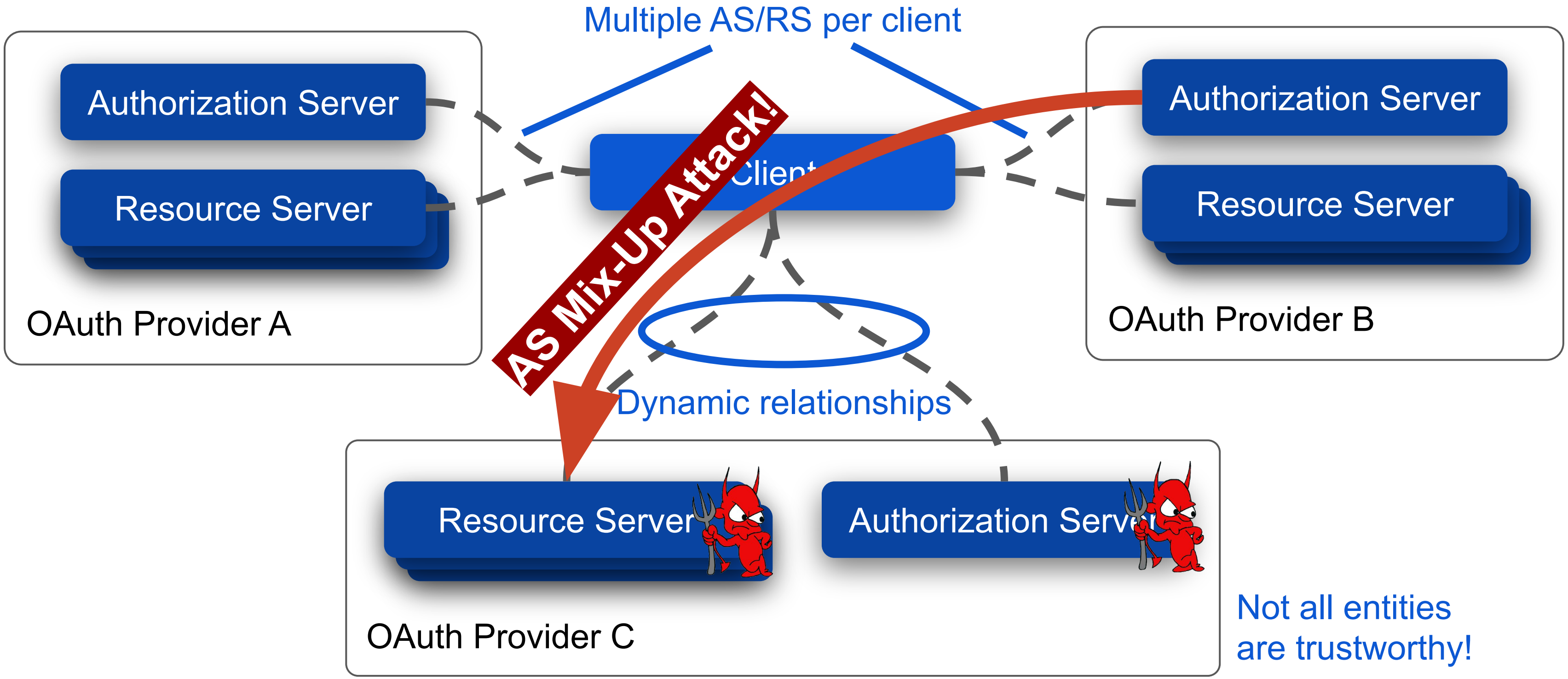
Authorization Server

OAuth Provider C

Not all entities are trustworthy!

Challenge 3: Large-scale Open Ecosystems

Today:



Not all entities are trustworthy!

How to address these challenges?

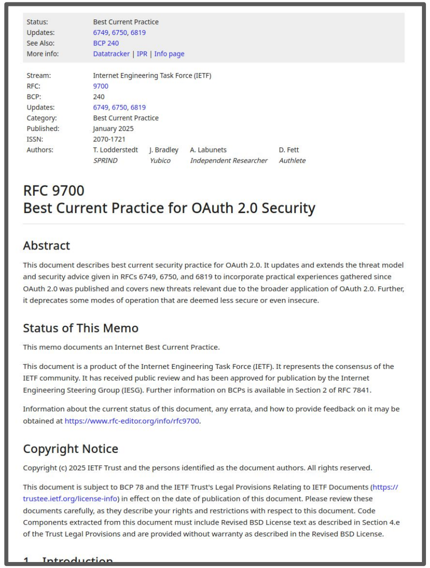
Securing Your Grandfather’s OAuth

- RFC9700: Best Current Practice for OAuth 2.0 Security
- OAuth 2.1
- FAPI 2.0

RFC9700: Best Current Practice for OAuth 2.0 Security

~10 years of collected OAuth security knowledge

- Refined and enhanced security guidance for OAuth 2.0 implementers
- Complements existing security guidance in RFCs 6749, 6750, and 6819



Status:	Best Current Practice
Updates:	6749, 6750, 6819
See Also:	BCP 240
More info:	Datatracker | IPR | Info page

Stream:	Internet Engineering Task Force (IETF)			
RFC:	9700			
BCP:	240			
Updates:	6749, 6750, 6819			
Category:	Best Current Practice			
Published:	January 2025			
ISSN:	2070-1721			
Authors:	T. Lodderstedt *SPRIND*	J. Bradley *Yubico*	A. Labunets *Independent Researcher*	D. Fett *Authlete*

RFC 9700
Best Current Practice for OAuth 2.0 Security

Abstract

This document describes best current security practice for OAuth 2.0. It updates and extends the threat model and security advice given in RFCs 6749, 6750, and 6819 to incorporate practical experiences gathered since OAuth 2.0 was published and covers new threats relevant due to the broader application of OAuth 2.0. Further, it deprecates some modes of operation that are deemed less secure or even insecure.

Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9700.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

1. Introduction

- Updated, more comprehensive Threat Model
- Description of Attacks and Mitigations
- Simple and actionable recommendations

Input from **practice** and **formal analysis**

OAuth 2.1

Updated version of OAuth 2.0

Includes all mitigations required by the Security BCP document

Removes less secure options and flows

OpenID FAPI

Security, interoperability, and feature profile for OAuth 2.0

Implements all the security recommendations from the OAuth Security BCP

Usable for all APIs, including high-security applications.

FAPI 2.0: Latest version

FAPI?

Financial API

FAPI?

~~Financial API~~

Financial API *Security Profile*

FAPI?

~~Financial API~~

~~Financial API *Security Profile*~~

Financial-*grade* API Security Profile

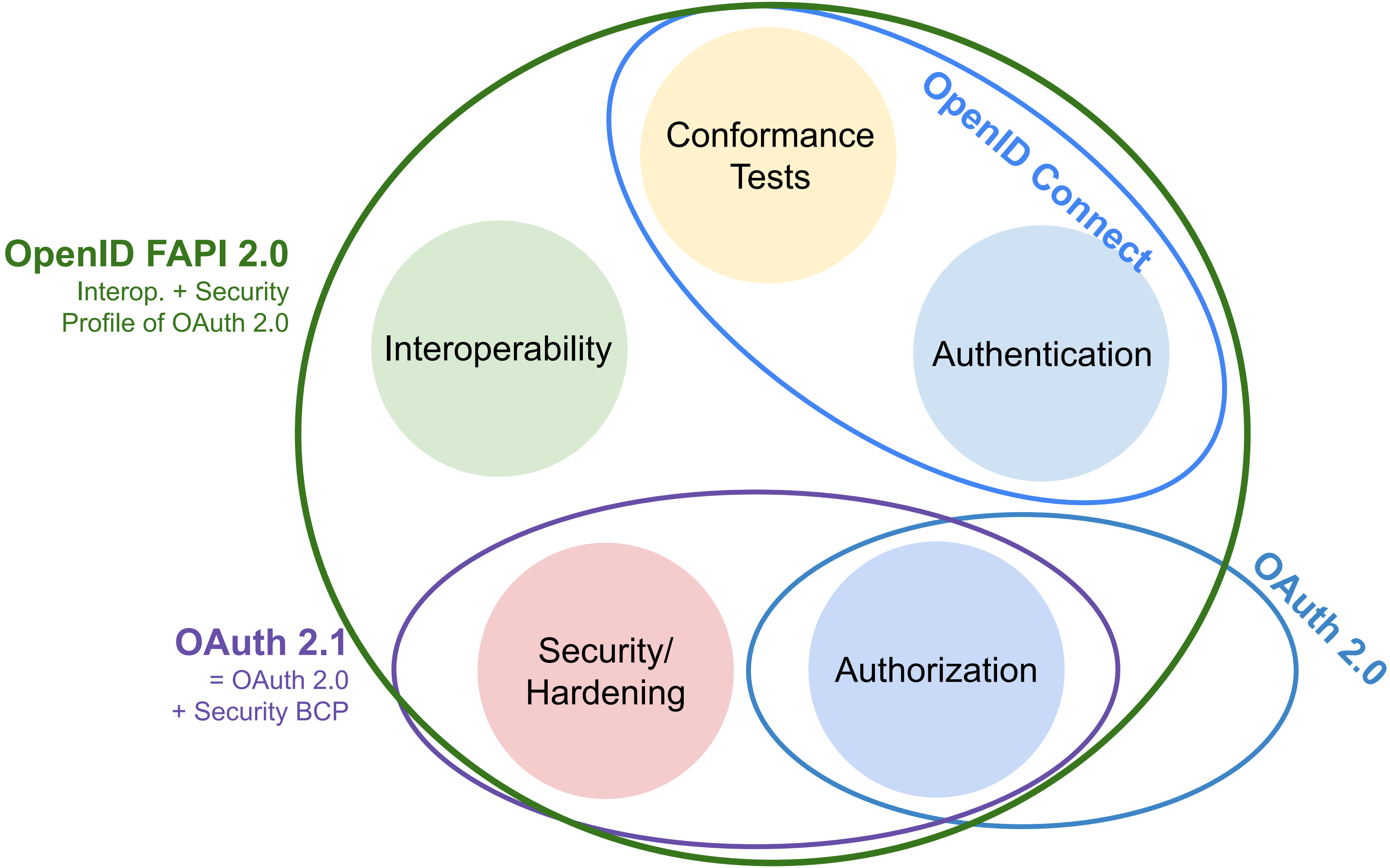
FAPI?

~~Financial API~~

~~Financial API *Security Profile*~~

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

FAPI



OpenID FAPI 2.0

Interop. + Security
Profile of OAuth 2.0

OpenID Connect

OAuth 2.1

= OAuth 2.0
+ Security BCP

OAuth 2.0

Conformance Tests

Interoperability

Authentication

Security/ Hardening

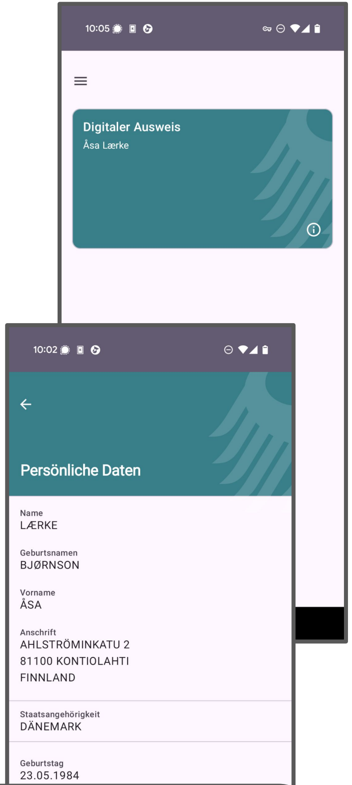
Authorization

And then

The Wallets Came Along

What the heck are Identity Wallets?

- Paradigm shift:
  - From server-based to user-centric identities
  - From identity providers to credential providers
- Not technically new — but now gaining traction world-wide
- EUDI Wallet:
  - To be provided until Christmas 2026
  - By all member-states
  - EU-wide interoperability
  - Official documents and other attestations (membership cards, tick
- US: Mobile Drivers License



10:05

Digitaler Ausweis

Åsa Lærke

10:02

Persönliche Daten

Name
LÆRKE

Geburtsnamen
BJØRNSON

Vorname
ÅSA

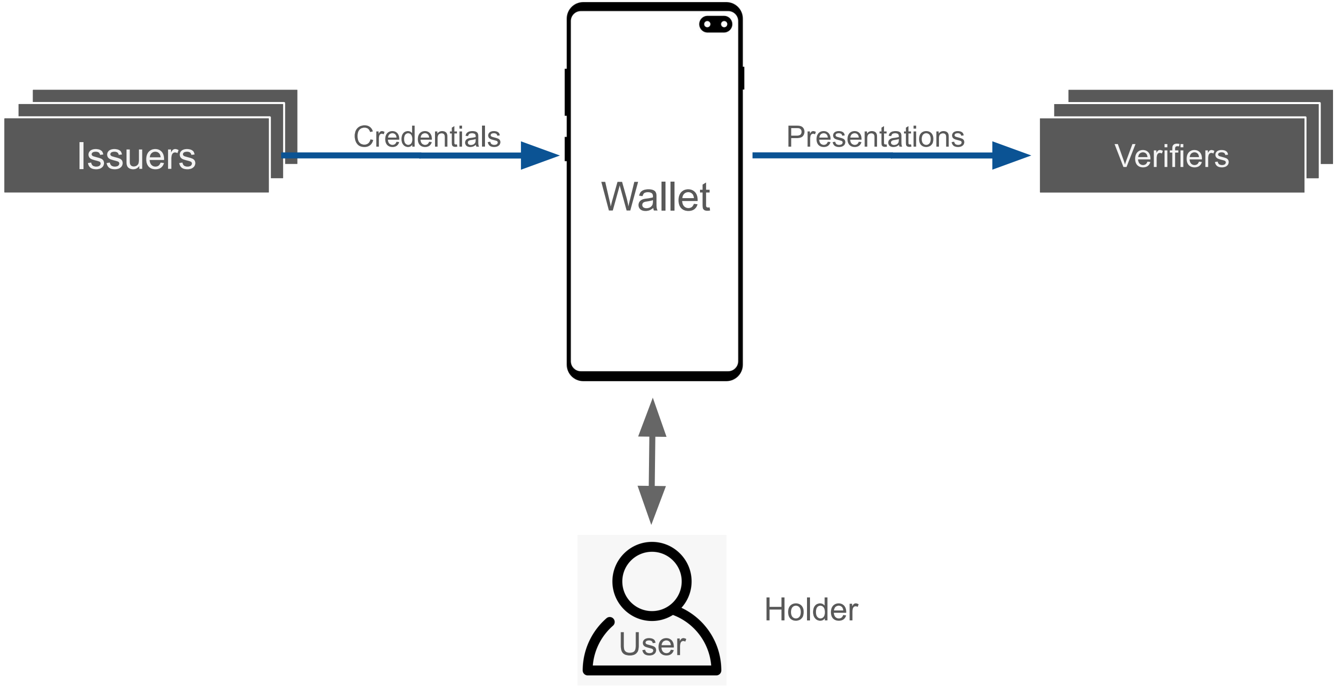
Anschrift
AHLSTRÖMINKATU 2
81100 KONTIOLAHTI
FINNLAND

Staatsangehörigkeit
DÄNEMARK

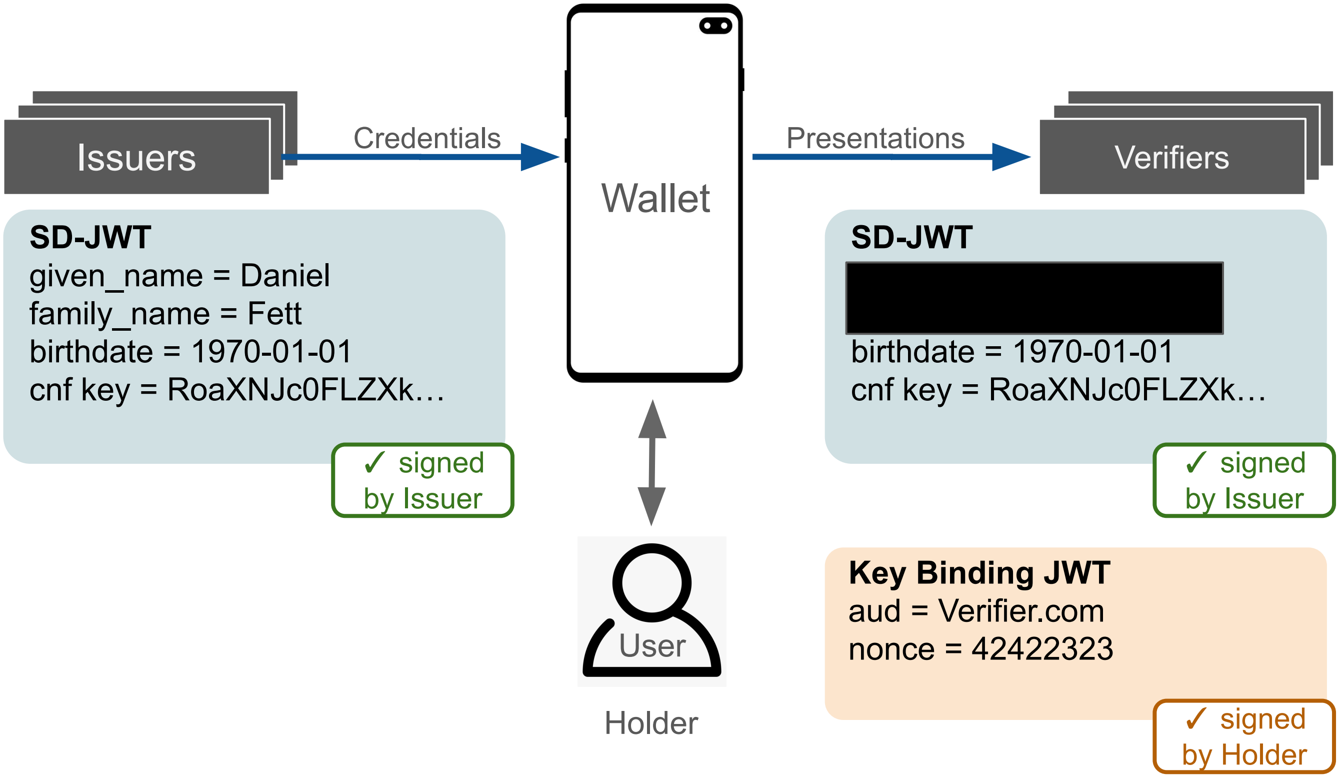
Geburtstag
23.05.1984

Disclaimer:
not related to crypto wallets
not blockchain-based

Identity Wallets



Identity Wallets



Issuers

Credentials

Wallet

Presentations

Verifiers

SD-JWT
given_name = Daniel
family_name = Fett
birthdate = 1970-01-01
cnf key = RoaXNJc0FLZXk…

✓ signed by Issuer

User

Holder

SD-JWT
[redacted]
birthdate = 1970-01-01
cnf key = RoaXNJc0FLZXk…

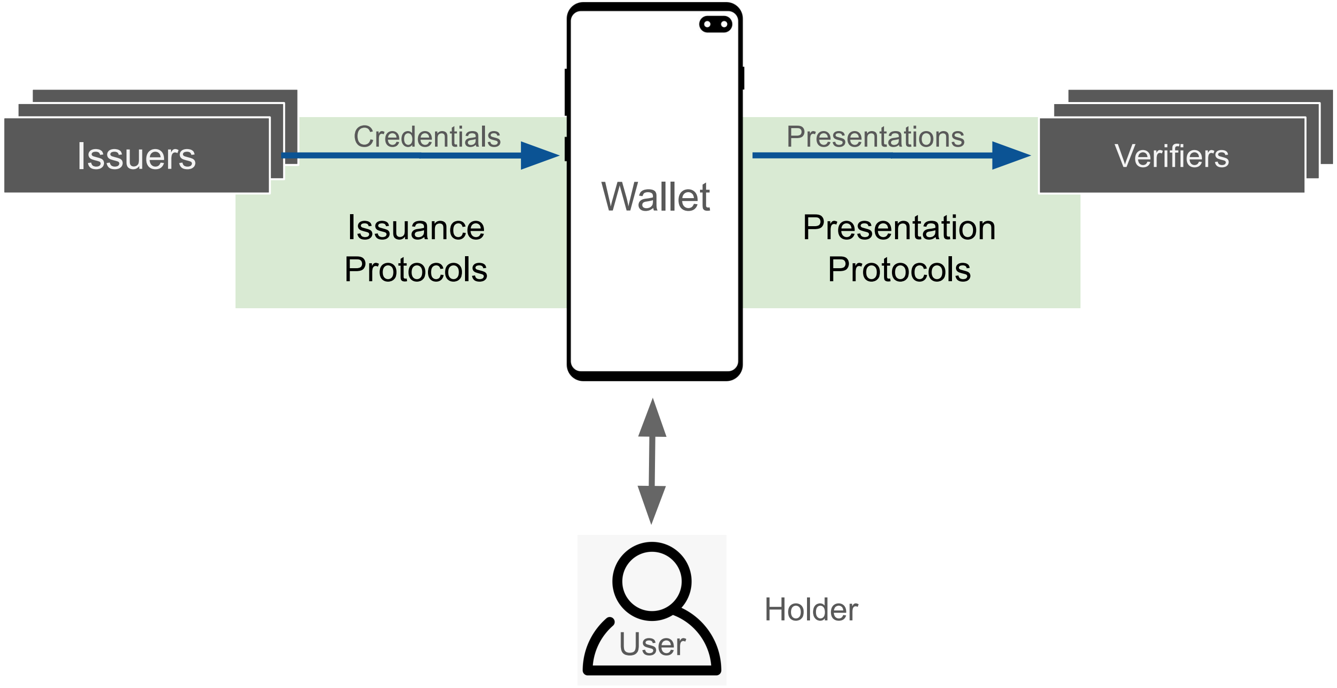
✓ signed by Issuer

Key Binding JWT
aud = Verifier.com
nonce = 42422323

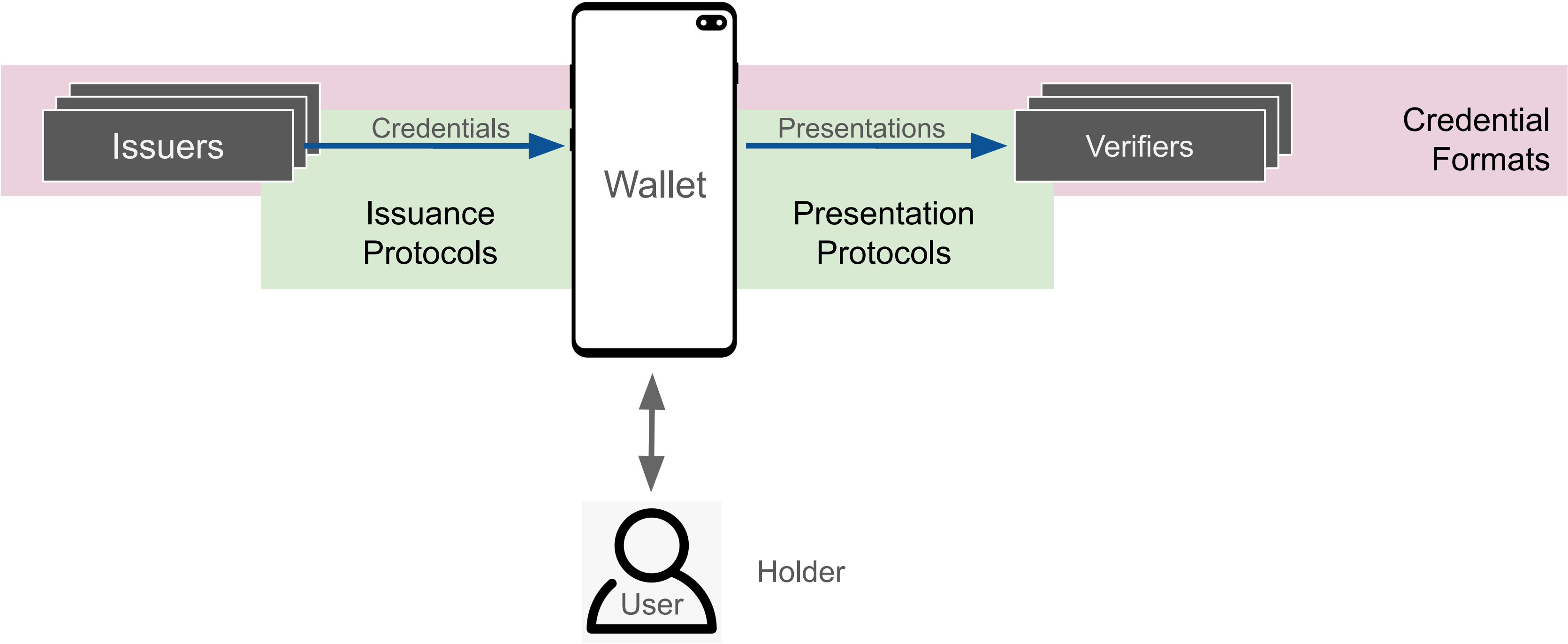
✓ signed by Holder

Under the Hood

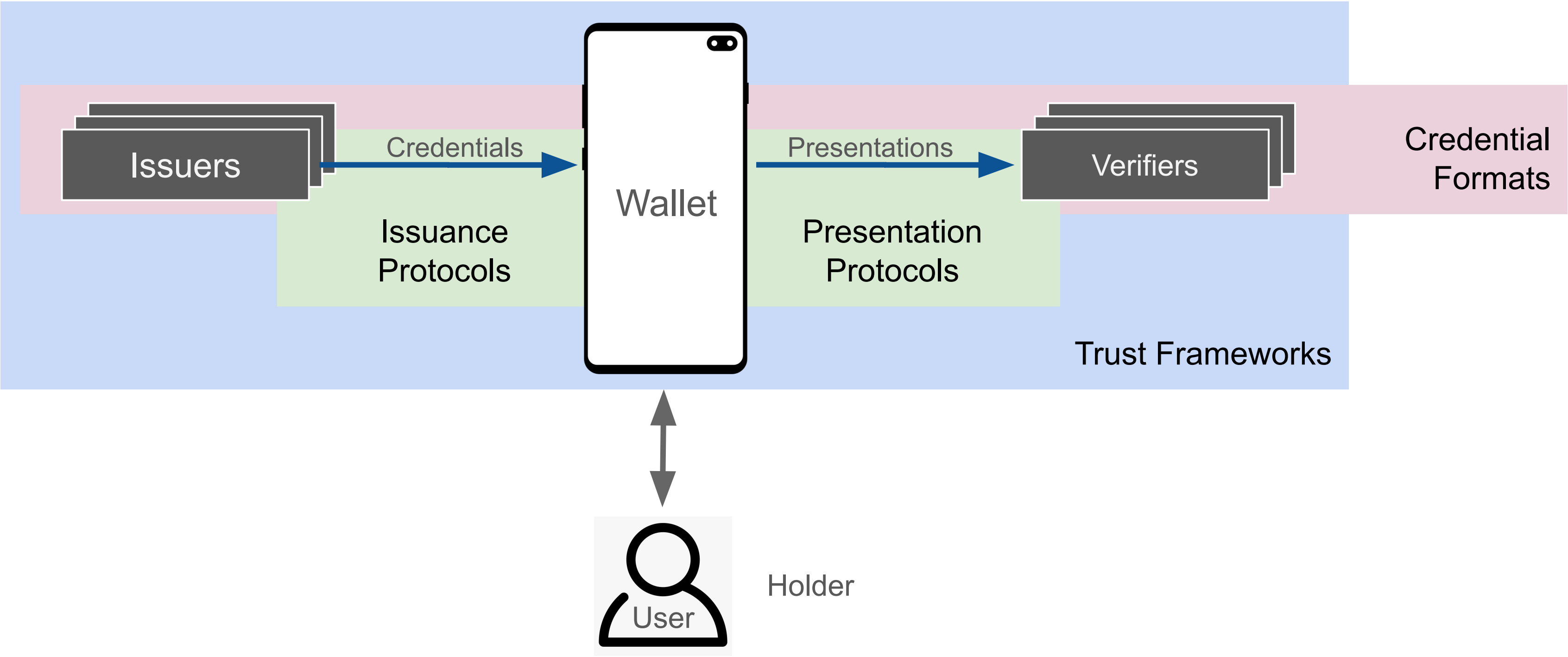
Wallet Ecosystems



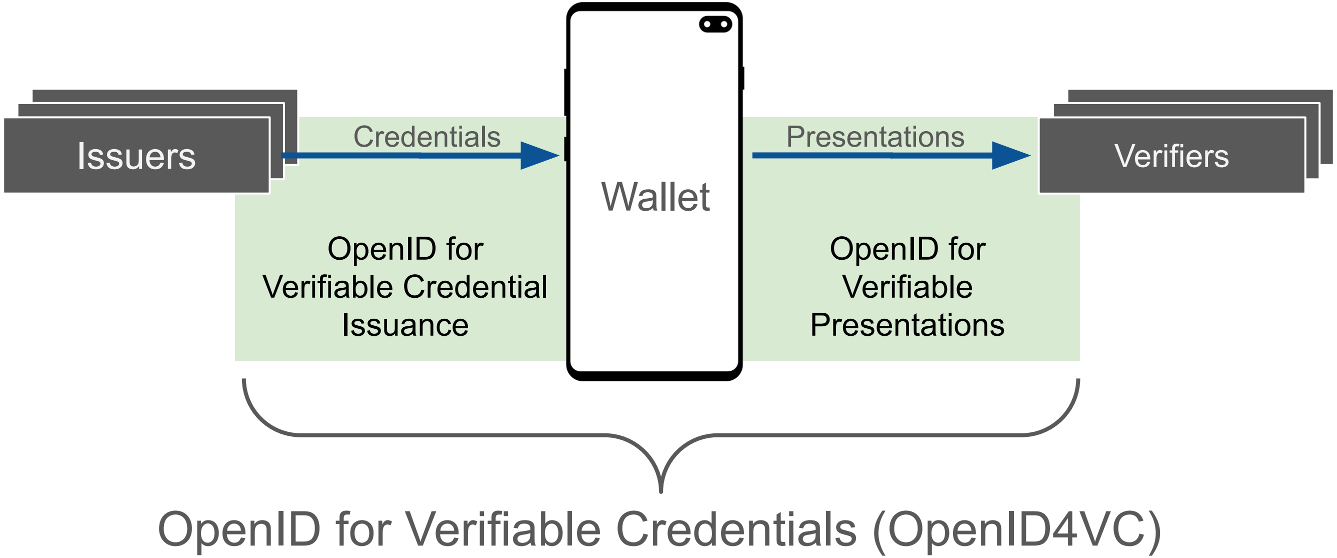
Wallet Ecosystems



Wallet Ecosystems

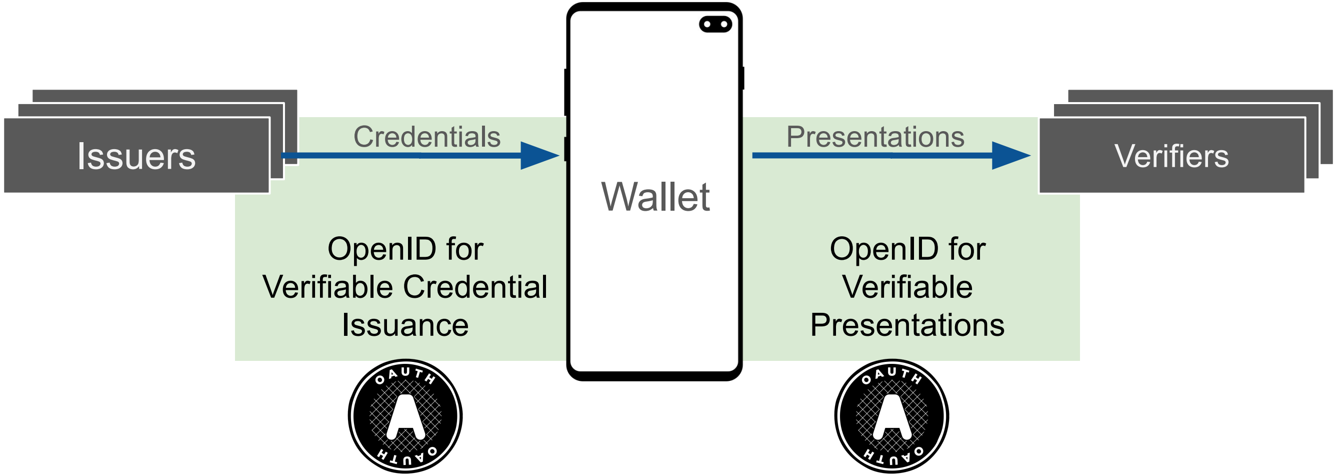


Protocols for Wallet Ecosystems


Issuers
Credentials
Wallet
Presentations
Verifiers
OpenID for Verifiable Credential Issuance
OpenID for Verifiable Presentations
OpenID for Verifiable Credentials (OpenID4VC)

OPENID4VC
OAUTH
A
OAUTH
imgflip.com

Protocols for Wallet Ecosystems


Issuers
Credentials
Wallet
Presentations
Verifiers
OpenID for Verifiable Credential Issuance
OpenID for Verifiable Presentations
OAUTH
OAUTH

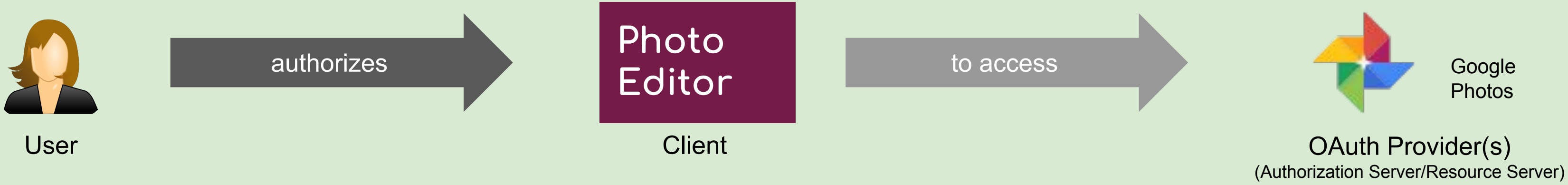
OpenID for Verifiable Credential Issuance

- Wallet acts as OAuth Client
- Issuer acts as Authorization Server
- Similar to OpenID Connect

OpenID for Verifiable Presentations

- Verifier acts as OAuth Client (Relying Party)
- Wallet acts as Authorization Server
- Mostly classic OAuth

OAuth



User

authorizes

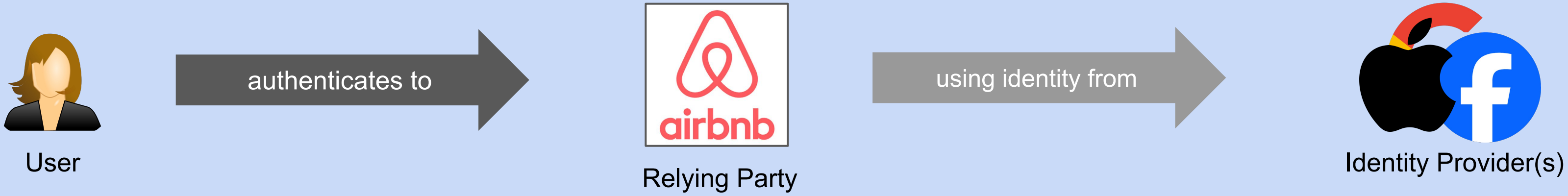
Photo Editor

Client

to access

OAuth Provider(s)
(Authorization Server/Resource Server)

OpenID Connect



User

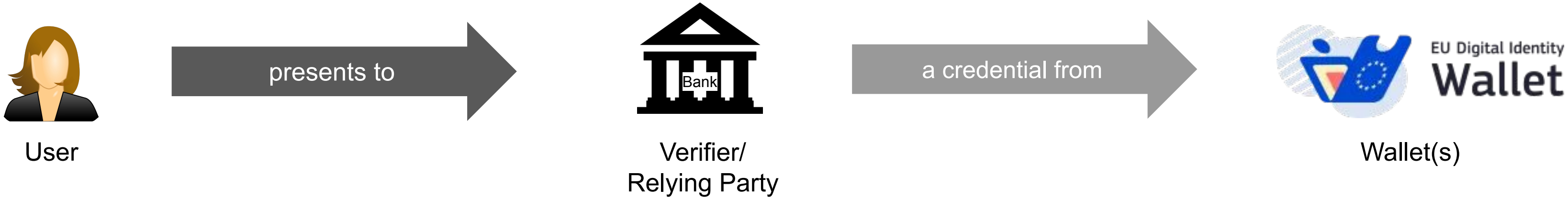
authenticates to

Relying Party

using identity from

Identity Provider(s)

OpenID for Verifiable Presentations



User

presents to

Verifier/
Relying Party

a credential from

Wallet(s)

Security Challenges for Wallet Ecosystems

- Key storage on mobile devices
- Cross-device flows
- Lack of secure biometric methods
- Complex EU-scale trust framework
- New protocols and standards

(also various privacy topics — let’s discuss if you’re interested)

What could possibly go wrong?

- Insufficient identification of the Verifier
- Identification process taken out of context
- User data can be forged
- Credentials could be forwarded to third parties
- …

Phishing

Call to Action

Implementers, Security Experts, Pentesters, Red Teamers:

- Expect a new tool for identification — the Wallet
- Make yourself familiar with the specifications and get involved
- Expect old & new vulnerabilities and prepare accordingly
- Use provided tooling (conformance tests) and resources (security considerations)

Dr. Daniel Fett
SPRIN-D
daniel.fett@eudi.sprind.org

Linkedin:

Thank you!

Requested Links

(added after the talk)

EUDI Wallet Project Website (not super interesting yet):
https://bmi.usercontent.opencode.de/eudi-wallet/eidas2/start/

Blueprint for the ecosystem (architecture etc.):
https://bmi.usercontent.opencode.de/eudi-wallet/eidas-2.0-architekturkonzept/

Wallet architecture details:
https://bmi.usercontent.opencode.de/eudi-wallet/wallet-development-documentation-public/

SPRIND job postings: https://sprind.org/wir/jobs