CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Fast IDentity Online with Anonymous Credentials (FIDO-AC)

CISPA Helmholtz Center for Information Security

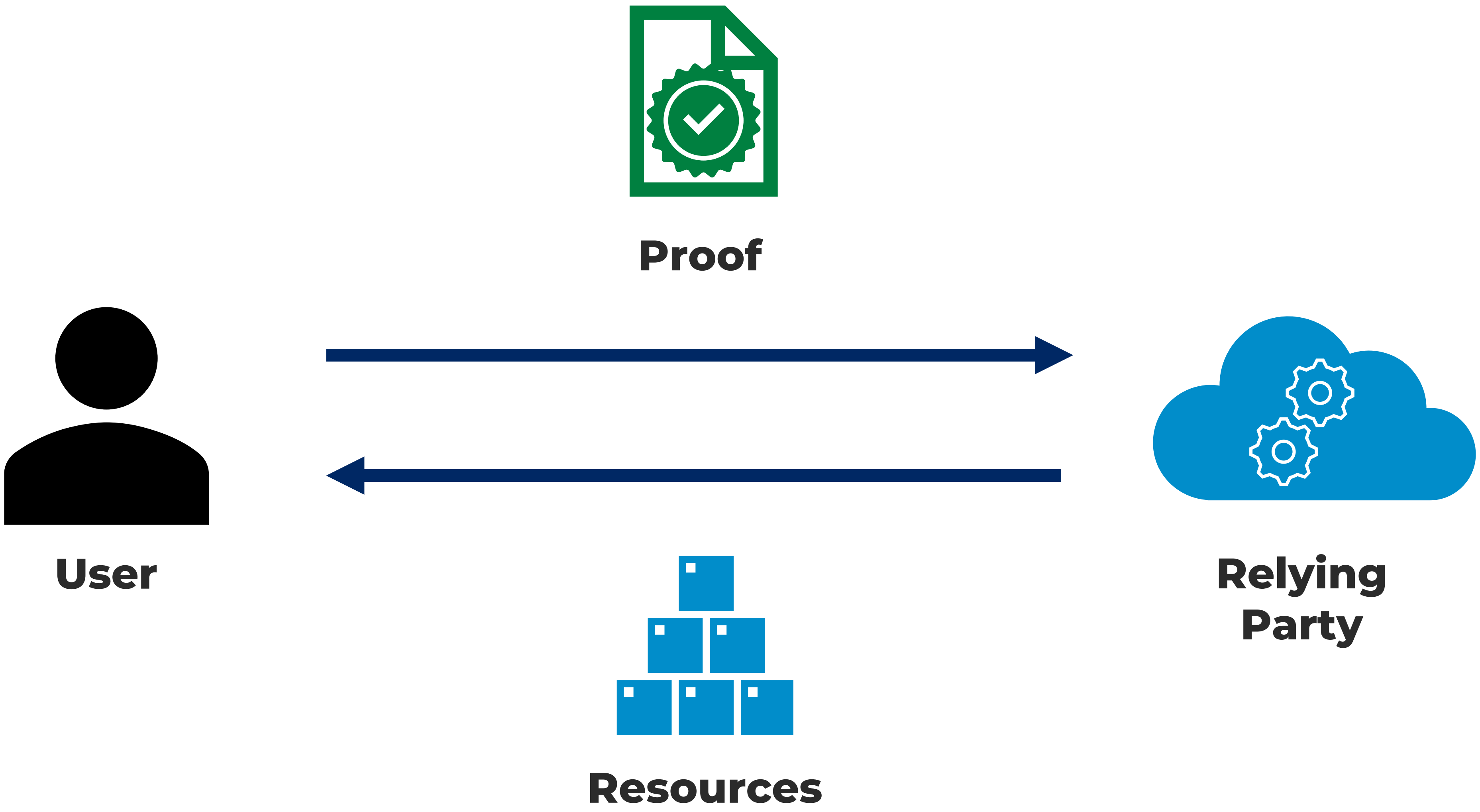
Wei-Zhu Yeoh*, *Gunnar Heide, Lucjan Hanzlik*

Macquarie University

Michal Kepkowski, Dali Kaafar,*

*Equal contribution

Authentication



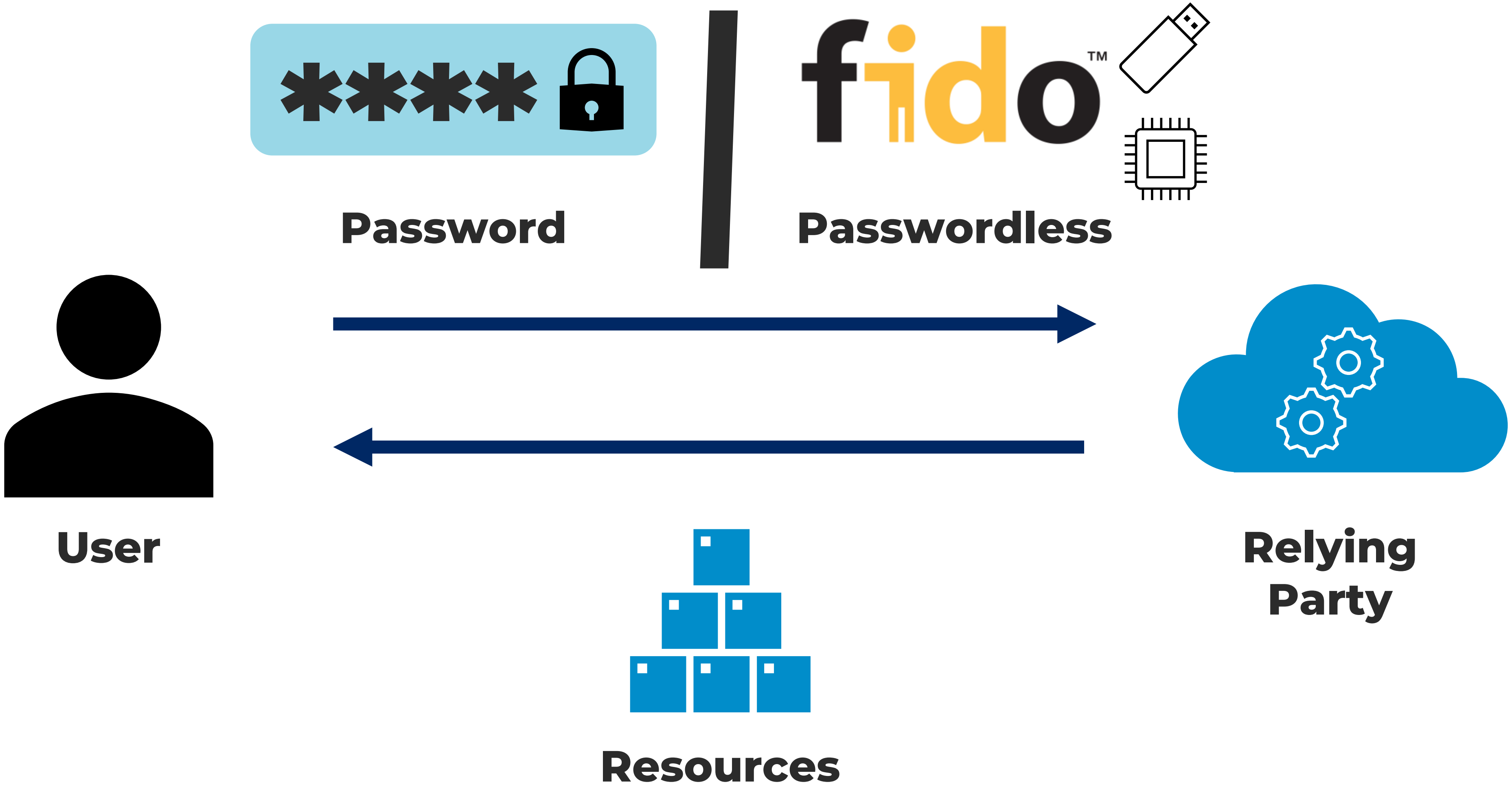
Proof

User

Relying Party

Resources

Authentication



Password

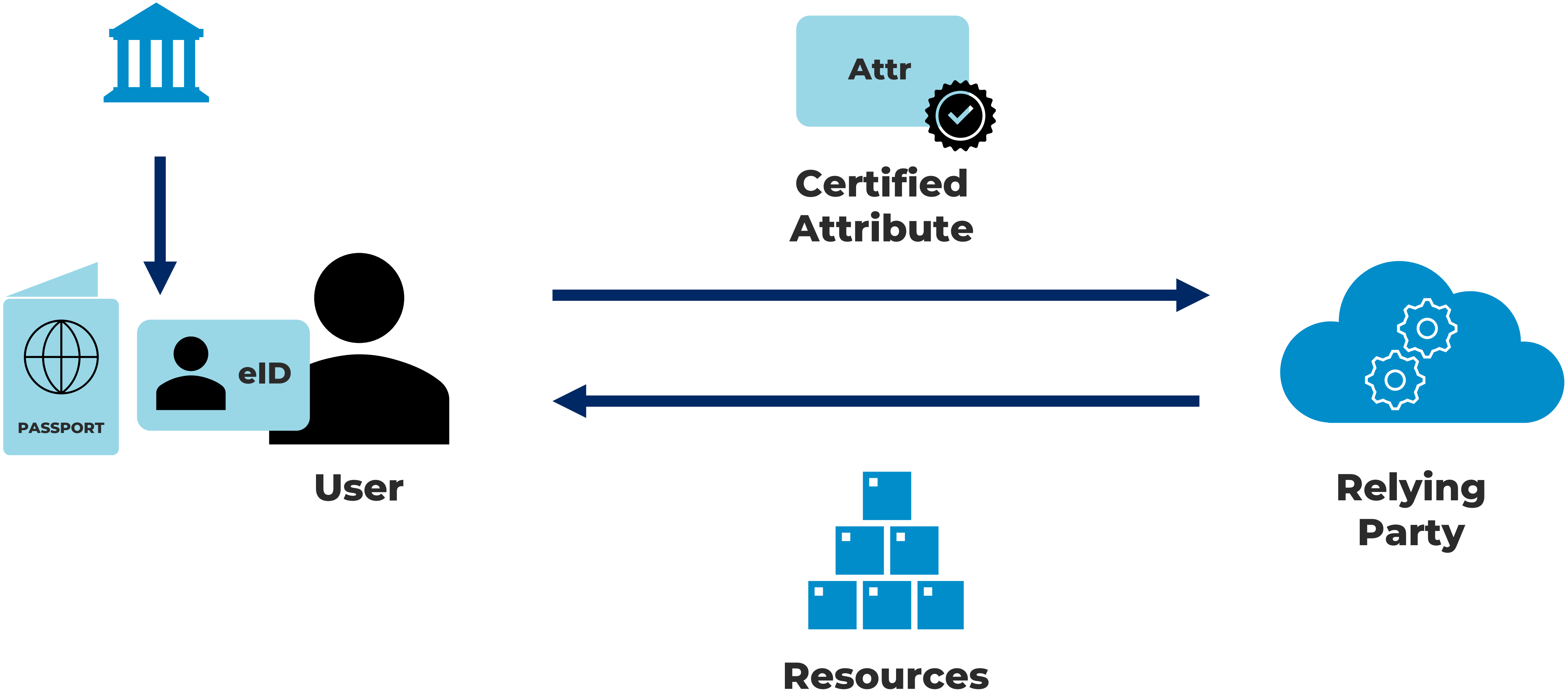
Passwordless

User

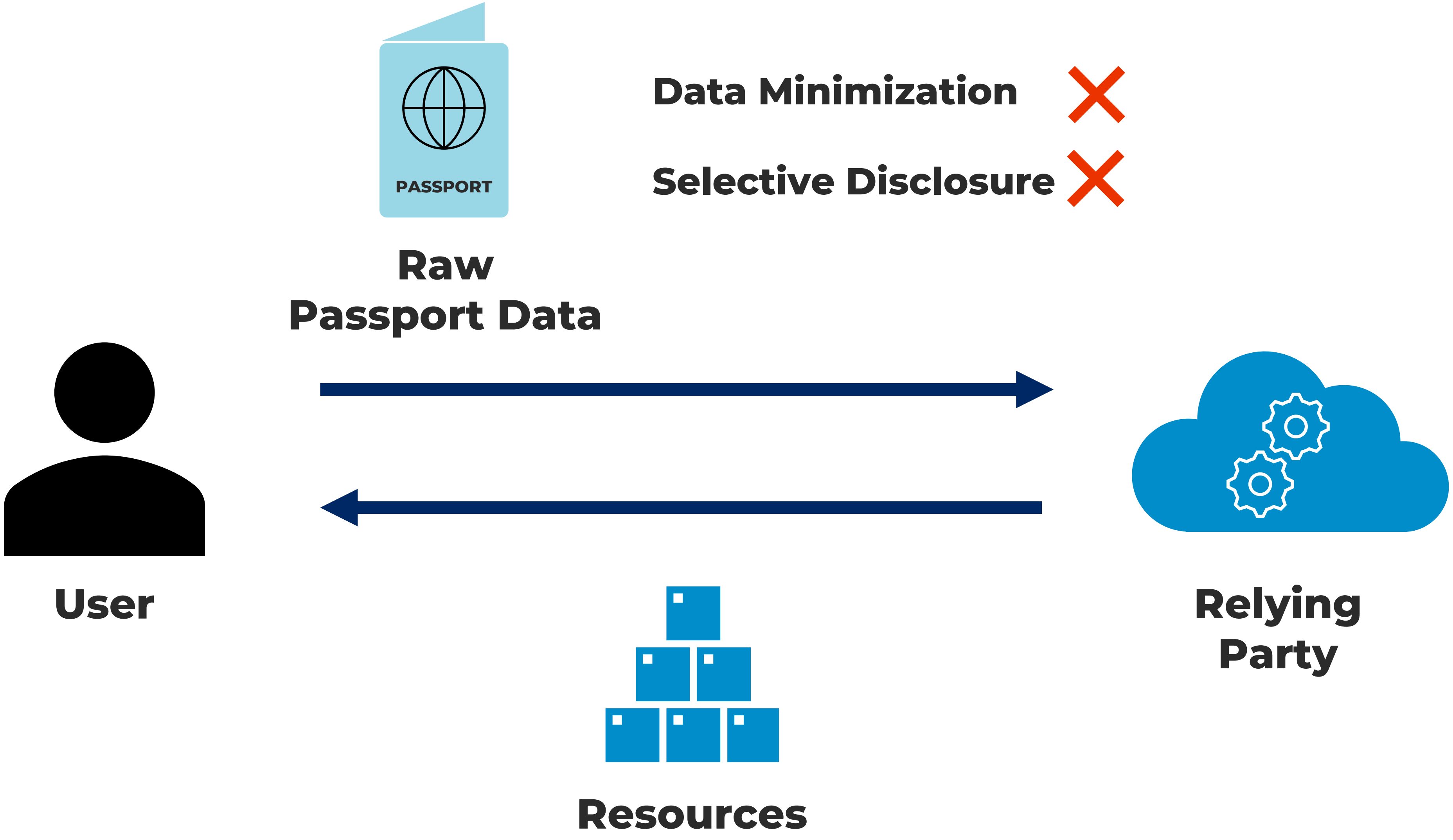
Relying Party

Resources

Authentication



Authentication



Data Minimization ✗

Selective Disclosure ✗

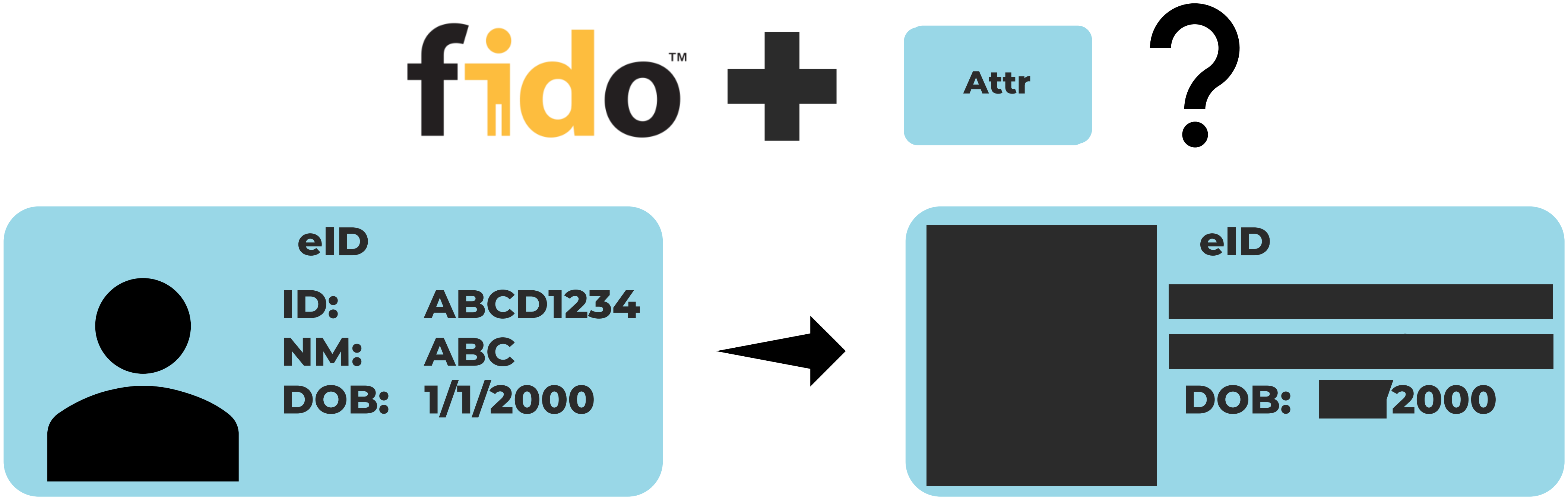
Raw
Passport Data

User

Relying
Party

Resources

Challenge

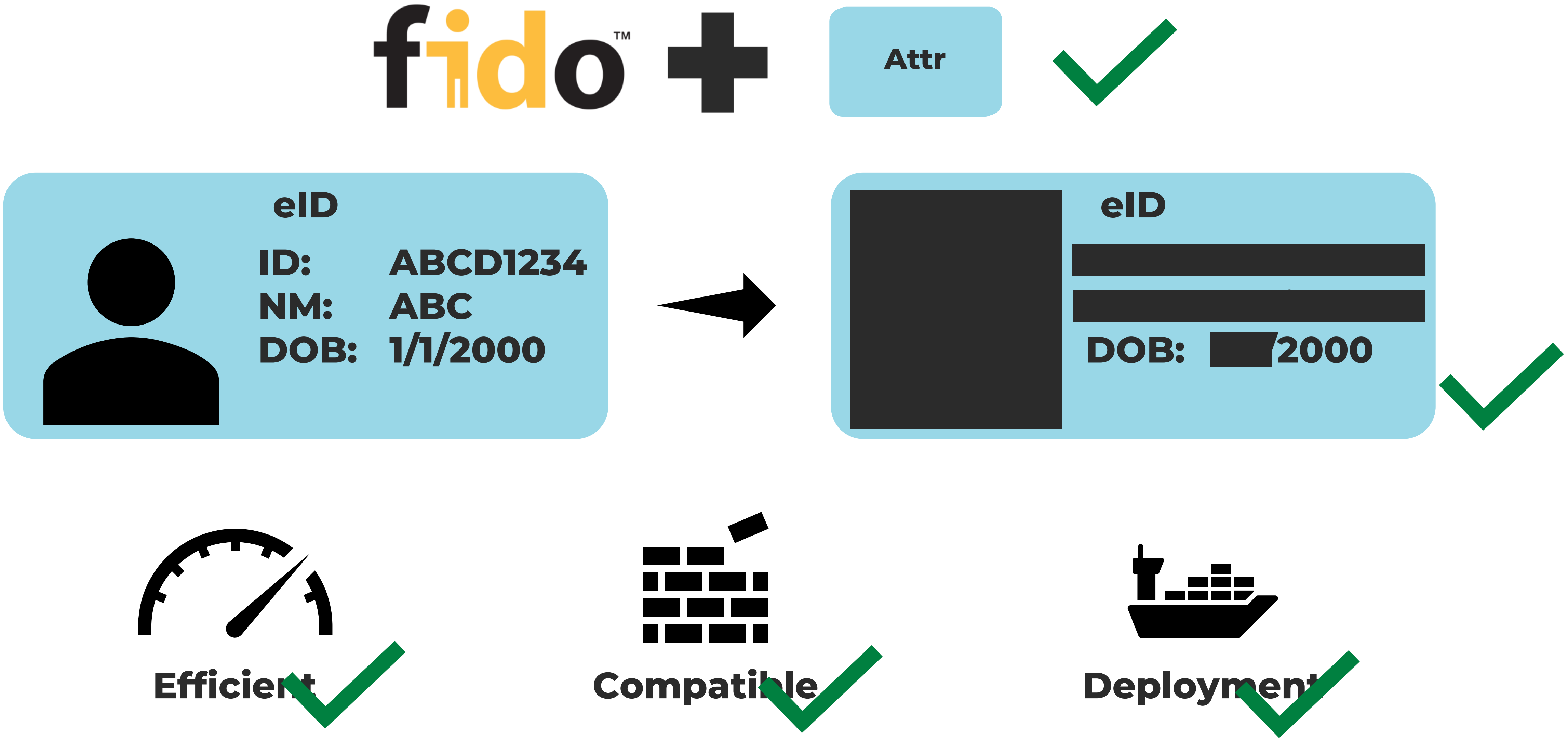


Efficient

Compatible

Deployment

Contribution



eID
ID: ABCD1234
NM: ABC
DOB: 1/1/2000

→

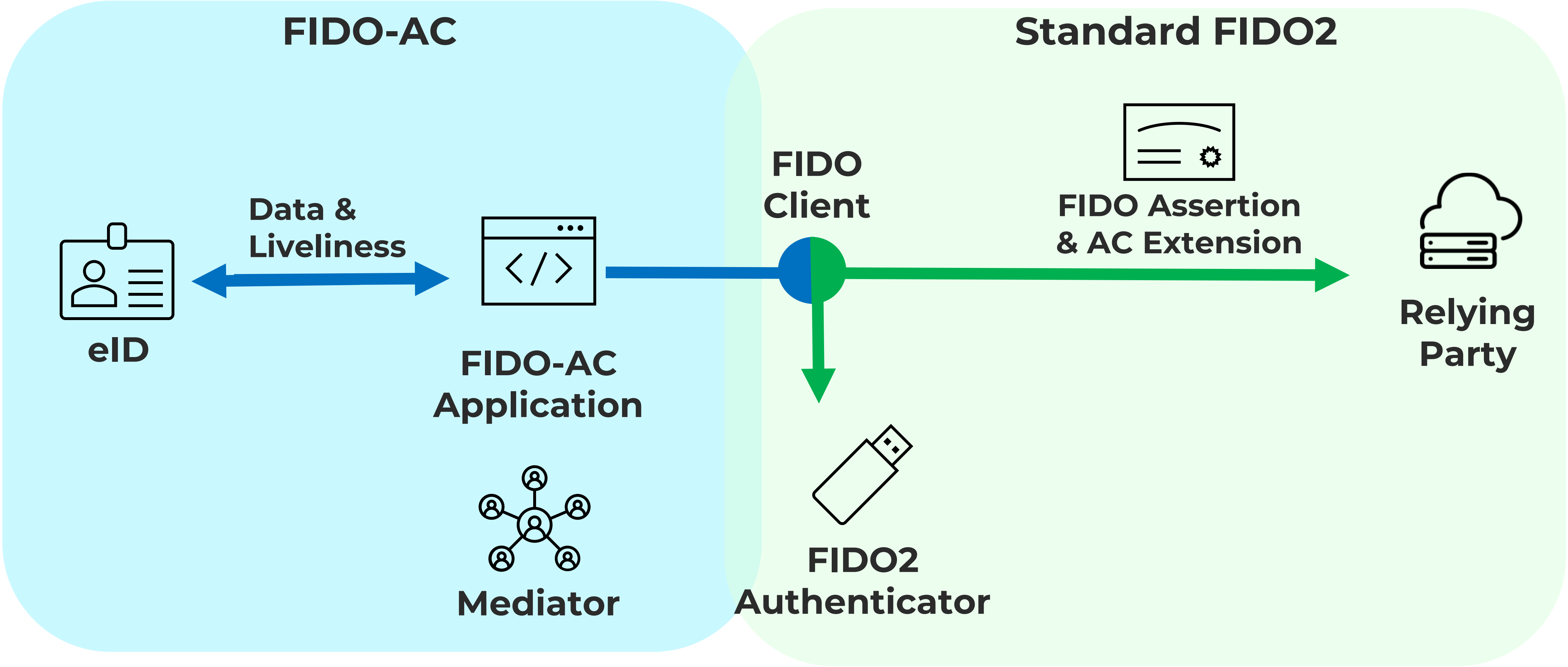
[redacted]
eID
[redacted]
DOB: [redacted]2000 ✓

Efficient ✓

Compatible ✓

Deployment ✓

FIDO-AC Overview



FIDO-AC Design Elements

Attr

1. Certified Attribute

In our instantiation,
ICAO MRTD (e.g., eID, ePassport)

2. Mediator

For generically interfacing with different eID frameworks.

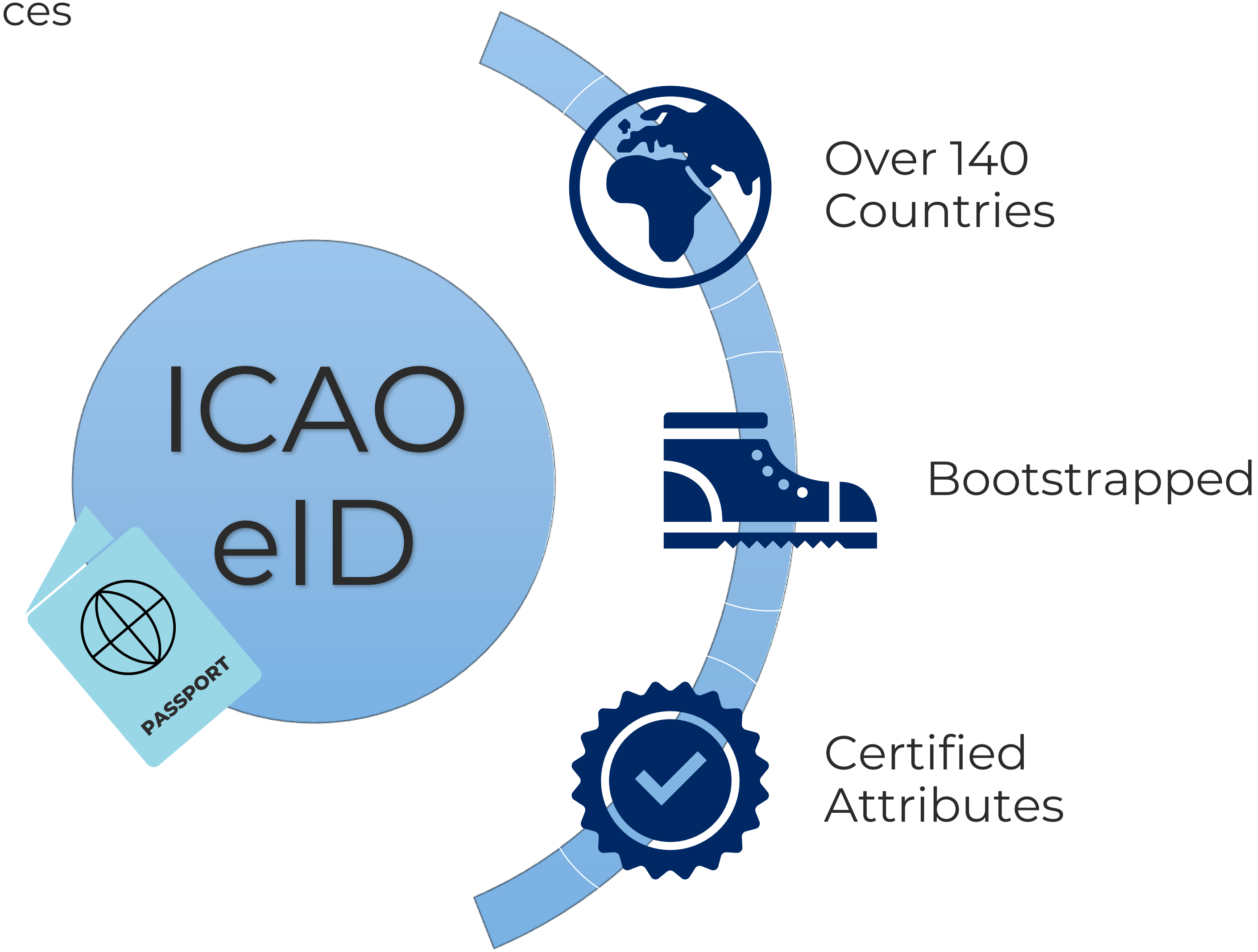
EXT.

3. FIDO2 Extension

For binding with FIDO.

Credentials

Certified Attributes Sources



ICAO eID

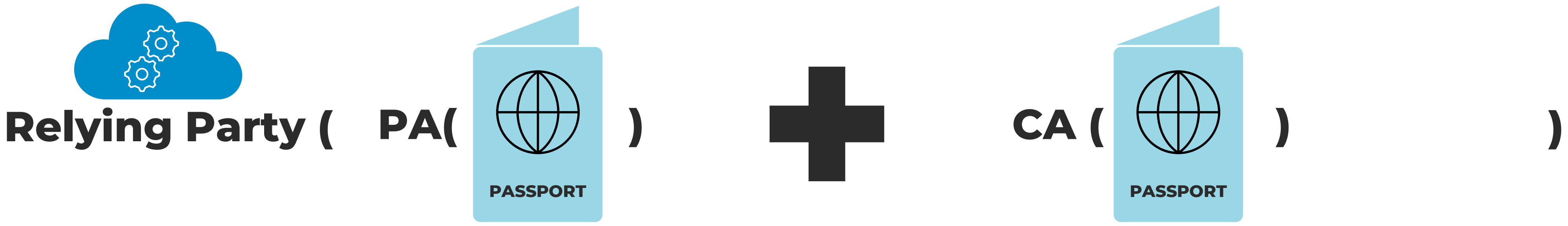
Over 140 Countries

Bootstrapped

Certified Attributes

Anonymous Credential

Approach 1: Pure Server-side Verification



Relying Party (PA( ) + CA ( ))

- Verify certificates
- Verify liveliness

Cross-domain User Linkability Violates Attributes Privacy

Anonymous Credential

Zero-Knowledge Proof

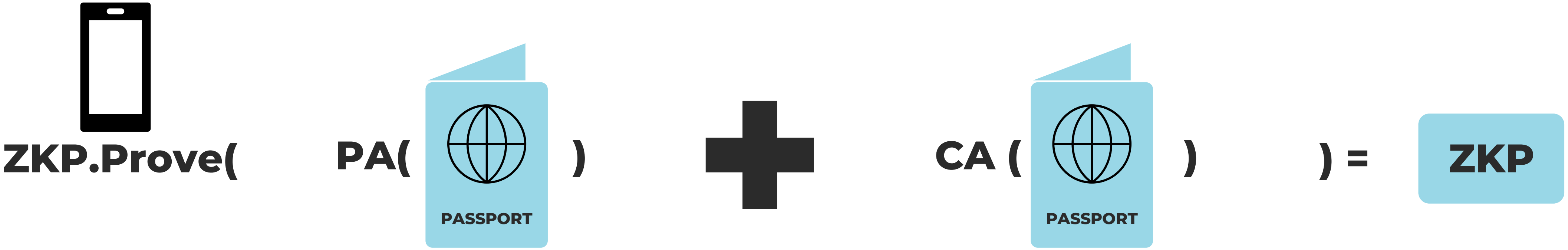
ZKP.Prove(CRS, statement, public-inputs, private-inputs) = Proof

ZKP.Verify(CRS, statement, proof, public-inputs) = True/False

Completeness: Verifier is convinced by correct proof.
Soundness: Prover cannot prove false statement.
Zero-Knowledge: Verifier learns nothing more than the statement.

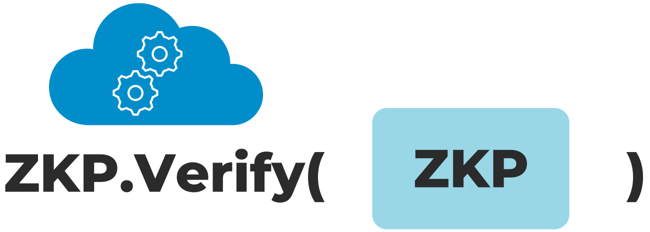
Anonymous Credential

Approach 2: Local Proof-of-Interaction and Proof-of-Attributes



ZKP.Prove(PA( ) + CA ( )) = ZKP

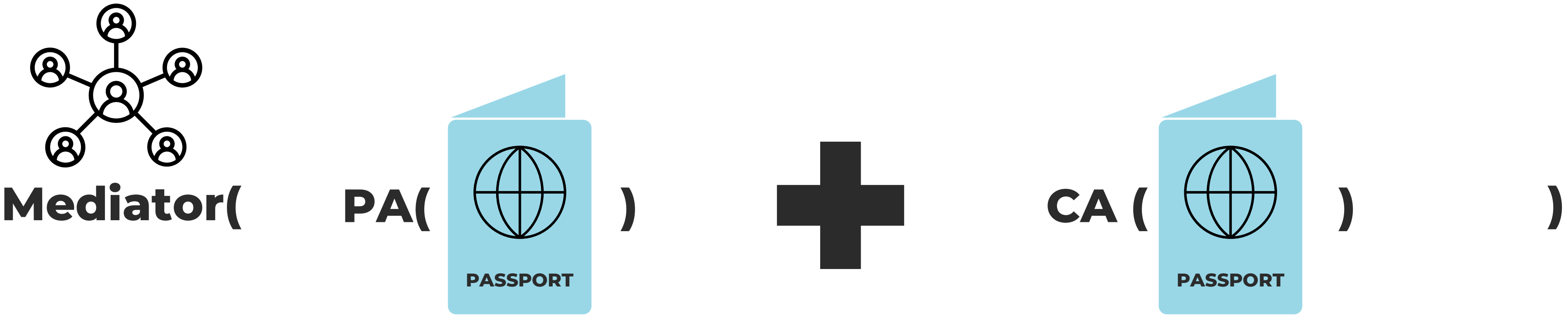
Not Efficient + Deniable Transcript



ZKP.Verify(ZKP)

Anonymous Credential

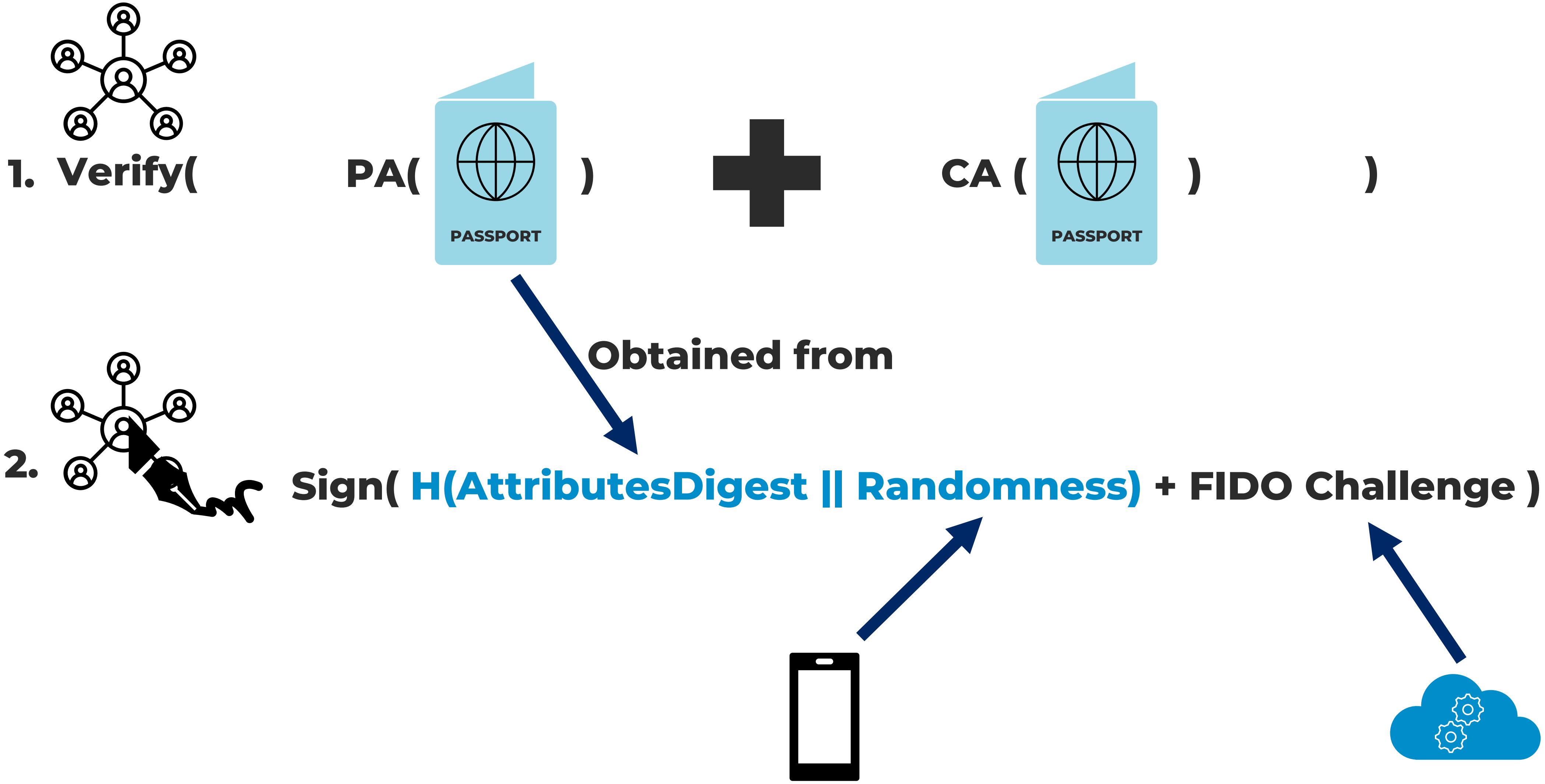
Introducing Mediator



Mediator(PA( ) ✚ CA ( ))

- Don’t learn about server policy.
- Don’t learn about eID attributes.
- Generic Solution: eID agnostic.

Mediator

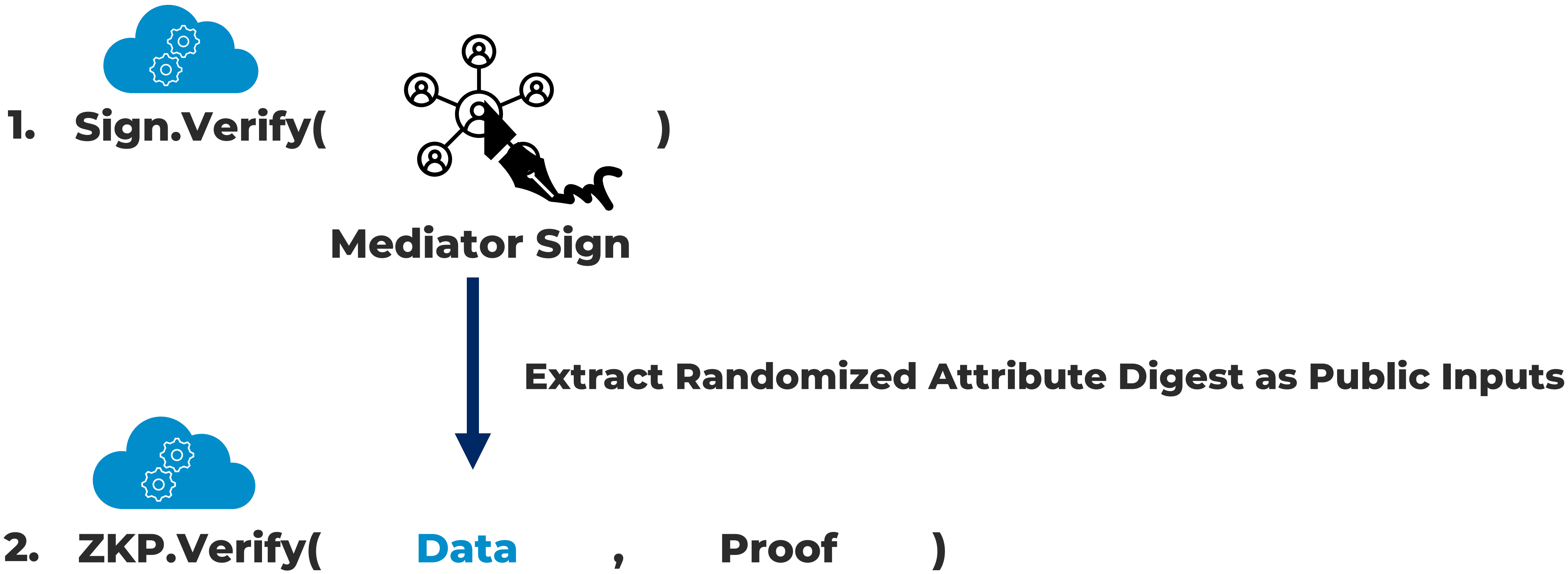


1. Verify(PA( ) + CA ( ))

Obtained from

2. Sign(H(AttributesDigest || Randomness) + FIDO Challenge)

Relying Party



1. Sign.Verify(Mediator Sign)

Extract Randomized Attribute Digest as Public Inputs

2. ZKP.Verify(Data , Proof)

Zero-Knowledge Proof

Data = Public Inputs

AttributesDigest || Randomness = Private Inputs

Prove

Data = H(AttributesDigest || Randomness)

and

AttributesDigest = H($bit_0, \ldots, bit_n$)

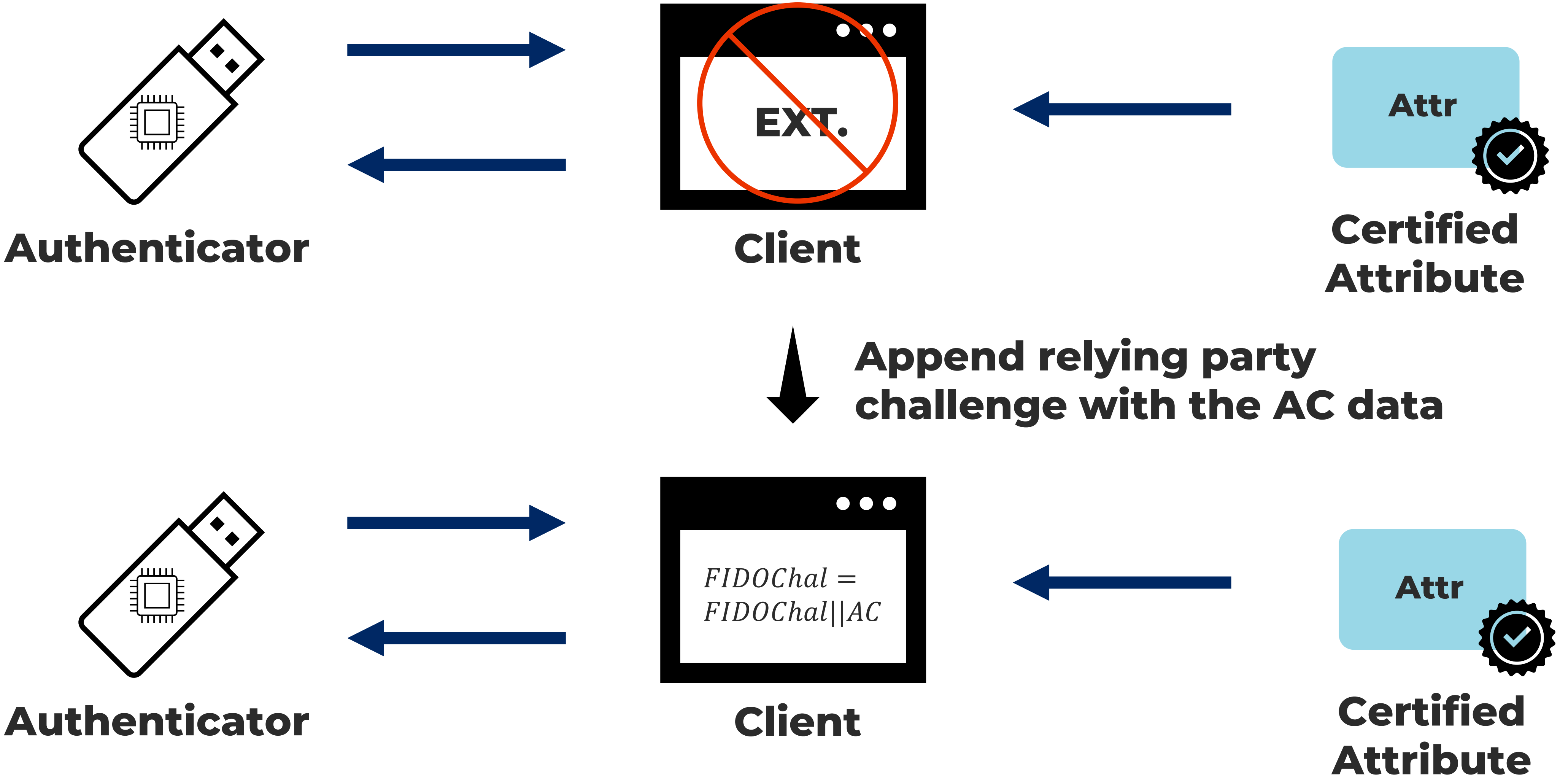
and

Attribute = Parse$(bit_i, \ldots, bit_j)$

and

True == Policy-Sat(Attribute)

FIDO2 Extension



Authenticator

EXT.

Client

Attr

Certified Attribute

Append relying party challenge with the AC data

Authenticator

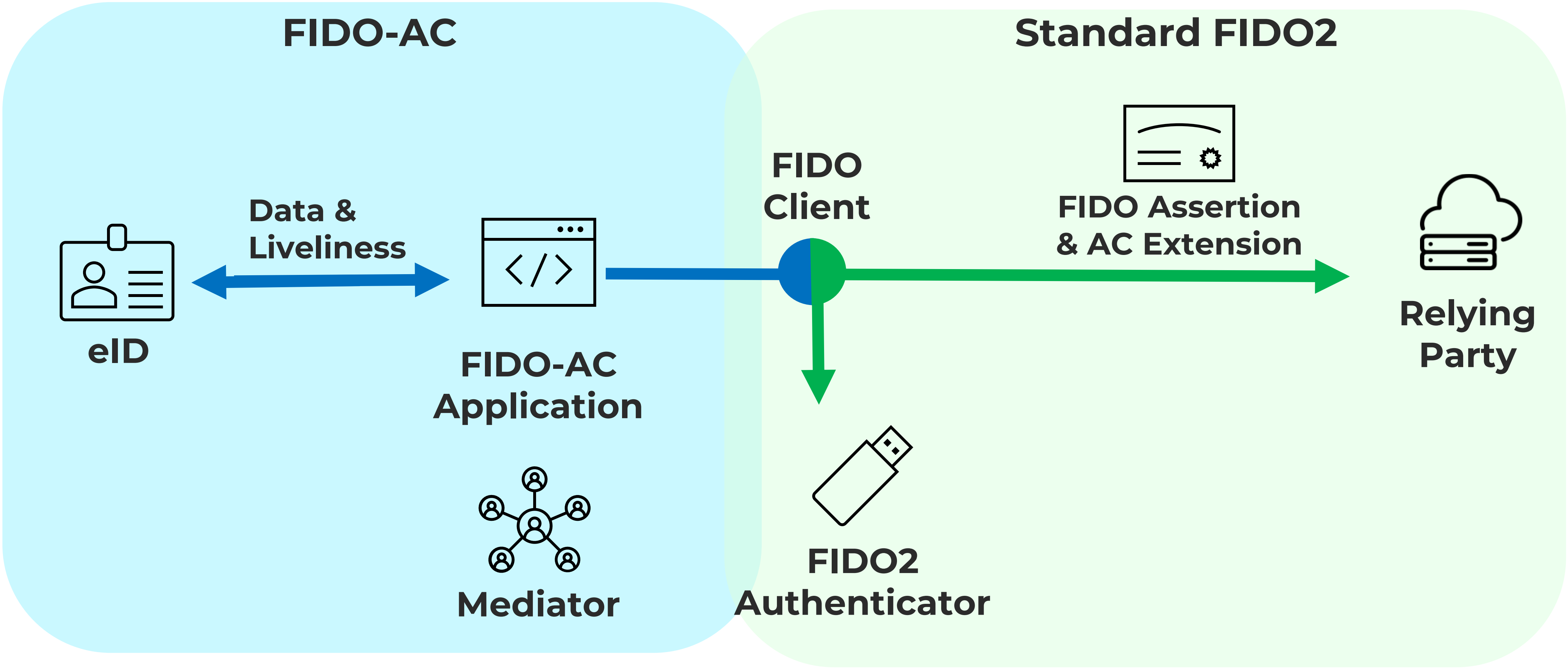
$FIDOChal = FIDOChal||AC$

Client

Attr

Certified Attribute

FIDO-AC Summary



FIDO-AC

Standard FIDO2

eID

Data & Liveliness

FIDO-AC Application

Mediator

FIDO Client

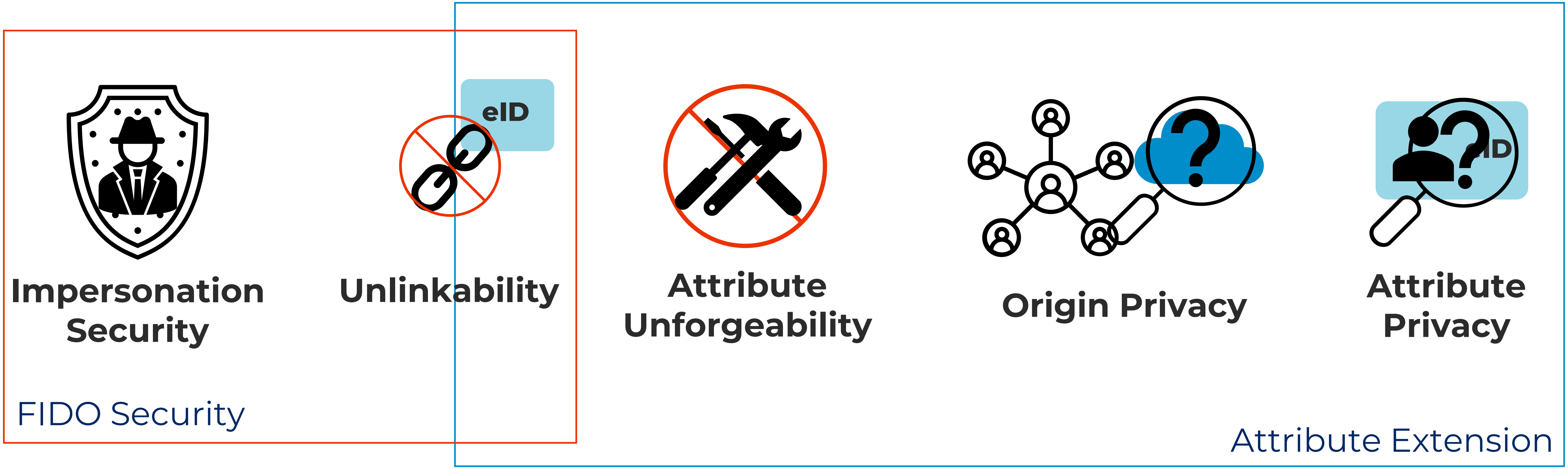
FIDO2 Authenticator

FIDO Assertion & AC Extension

Relying Party

Security Analysis

- Introduced passwordless authentication (PA) model with attribute (PAwA) based on PA model from Hanzlik' SP23.
- FIDO with attribute without mediator
- Due to compatibility and efficiency:
  - Introduced PAwA with mediator (PAwAM)



Impersonation Security

Unlinkability

Attribute Unforgeability

Origin Privacy

Attribute Privacy

FIDO Security

Attribute Extension

Security Analysis

Mediator Thread Model

	Mediator-Verifier		Mediator-Prover	
Unlinkability	None:	✗*	✓	
	TEE:	✗*		
	C-TEE:	✓		
Attribute Unforgeability	✓		None:	✗
			TEE:	✓
			C-TEE:	✓

* - For ICAO eID, other eID might achieve stronger property.

Proof-of-Concept Implementation

Android Mobile App

ePassport Interaction.

Local Mediator backed by Android Key Attestation.

Groth16’ ZKP using rust-arkwork library

FIDO-AC Server

Trusted Setup for Groth16’

JavaScript fidoac.js that bridges communication between FIDO-AC Mobile App and Relying Party logics.

Relying Party Server

Standard FIDO verification with updated challenge state and dockerized FIDO-AC extension verification.

Performance Evaluation

Operation	Platform	Time (ms)	SD (ms)
eID Reading	Mobile	1059.4/0.0^{cached}	37.58
Liveliness Check	Mobile	738.92	47.06
ZK Verify	Cloud PC	8.19	0.29
ZK Prove	Mobile	3375.61*	95.25

Mobile - Pixel 6 Pro
Cloud PC - a Standard D4s v3 Microsoft Azure Cloud Instance
* - Preprocessing possible

FIDO-AC

Fast Identity Online with Anonymous Credentials

Implementation demo

Summary

- FIDO-AC Framework
- Combining FIDO, eID and ZKP to create FIDO-AC
- Practical and privacy-preserving.
- Proof-of-Concept Implementation: