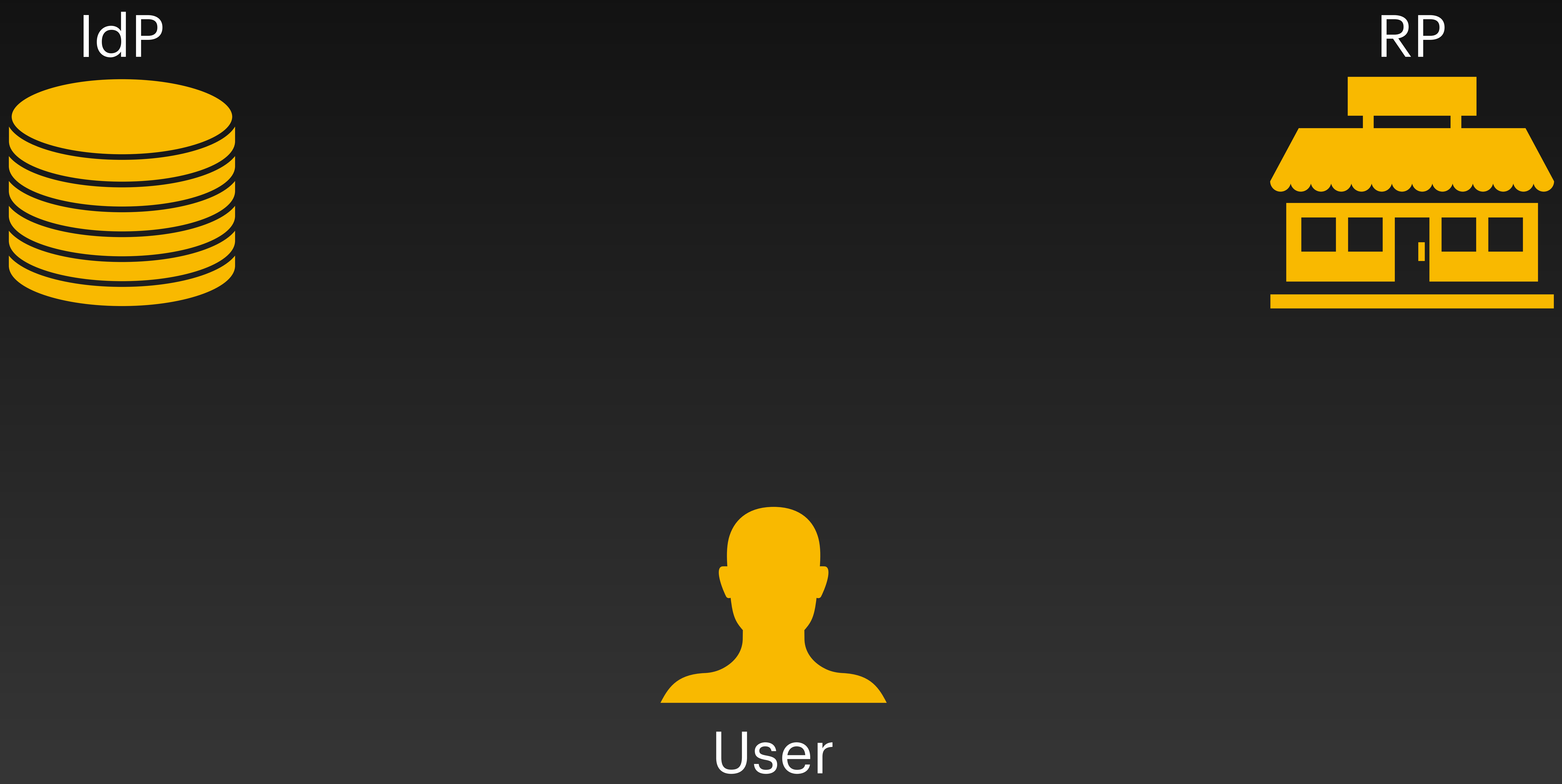
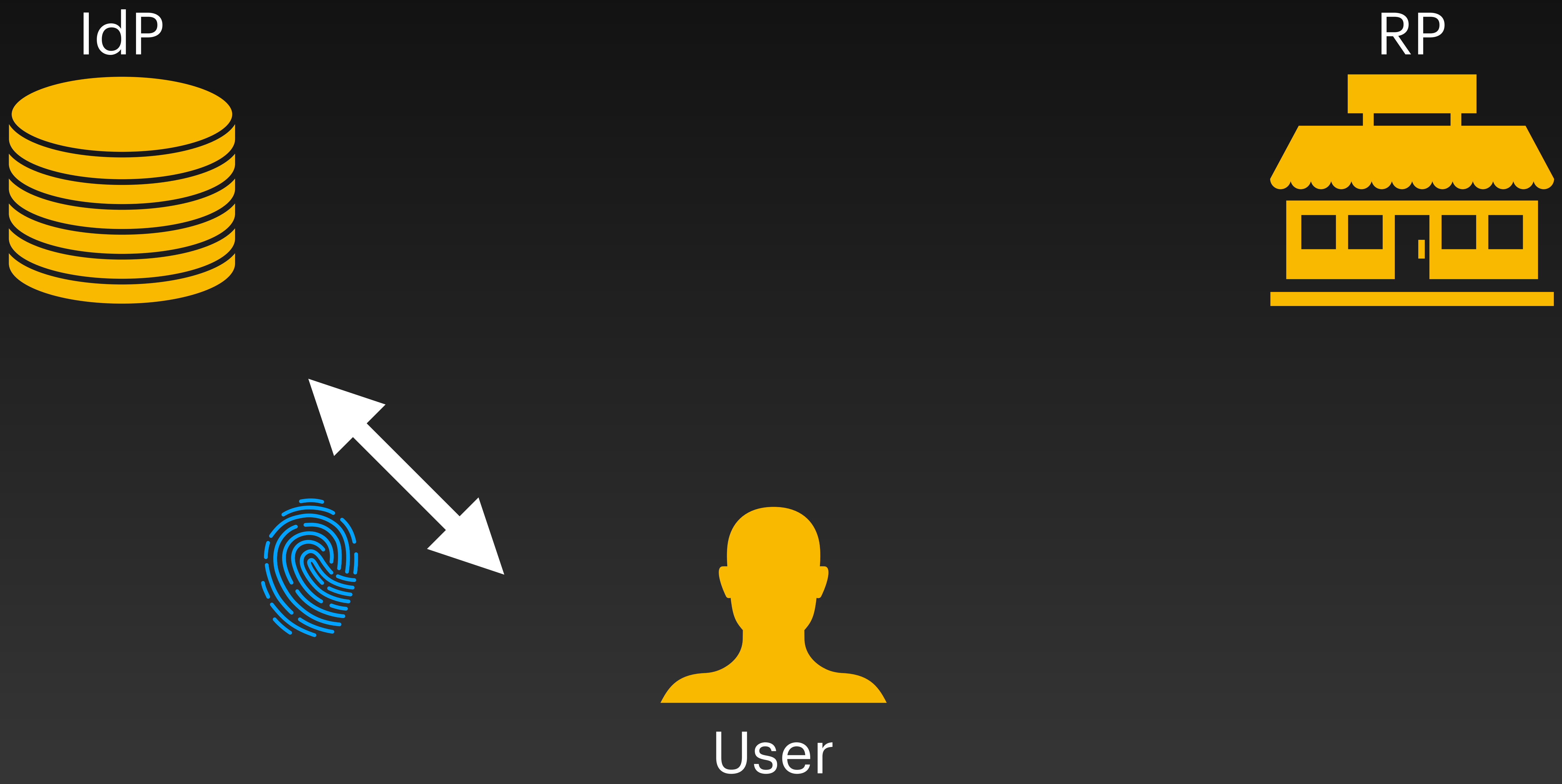
Digital Identity

with anonymous credentials

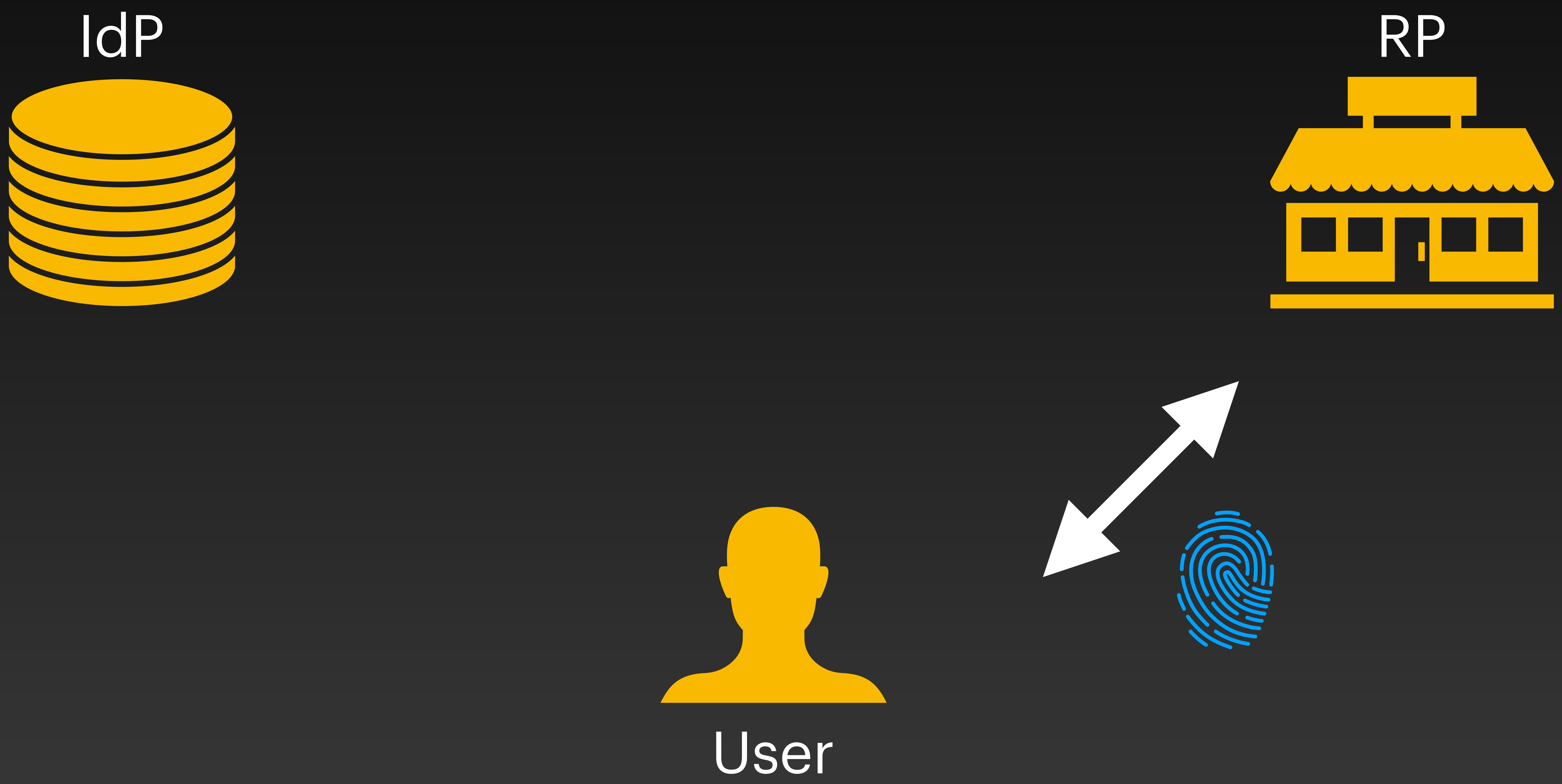
Setting



Setting



Setting

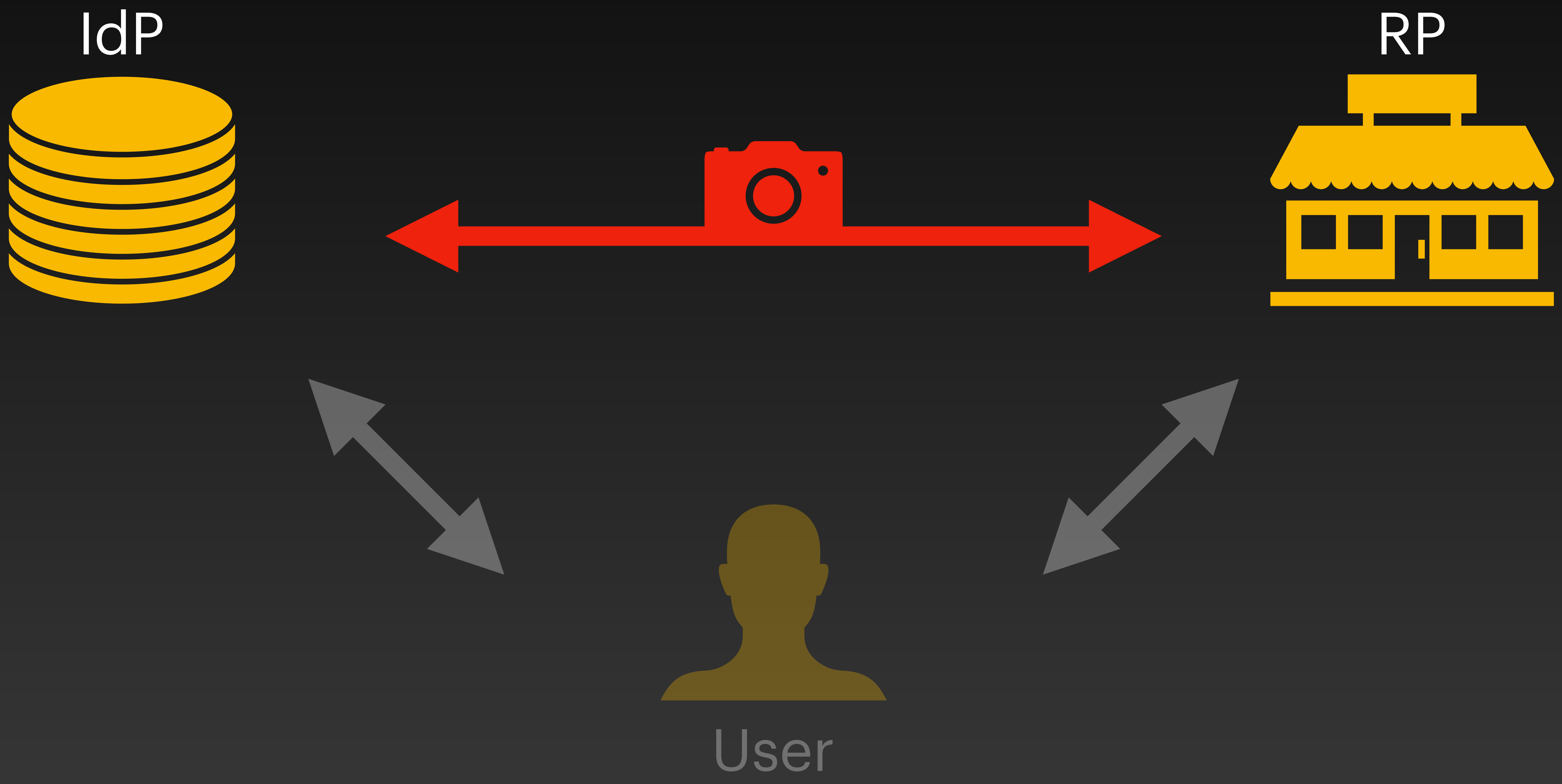


Standard SSO

Several limitations

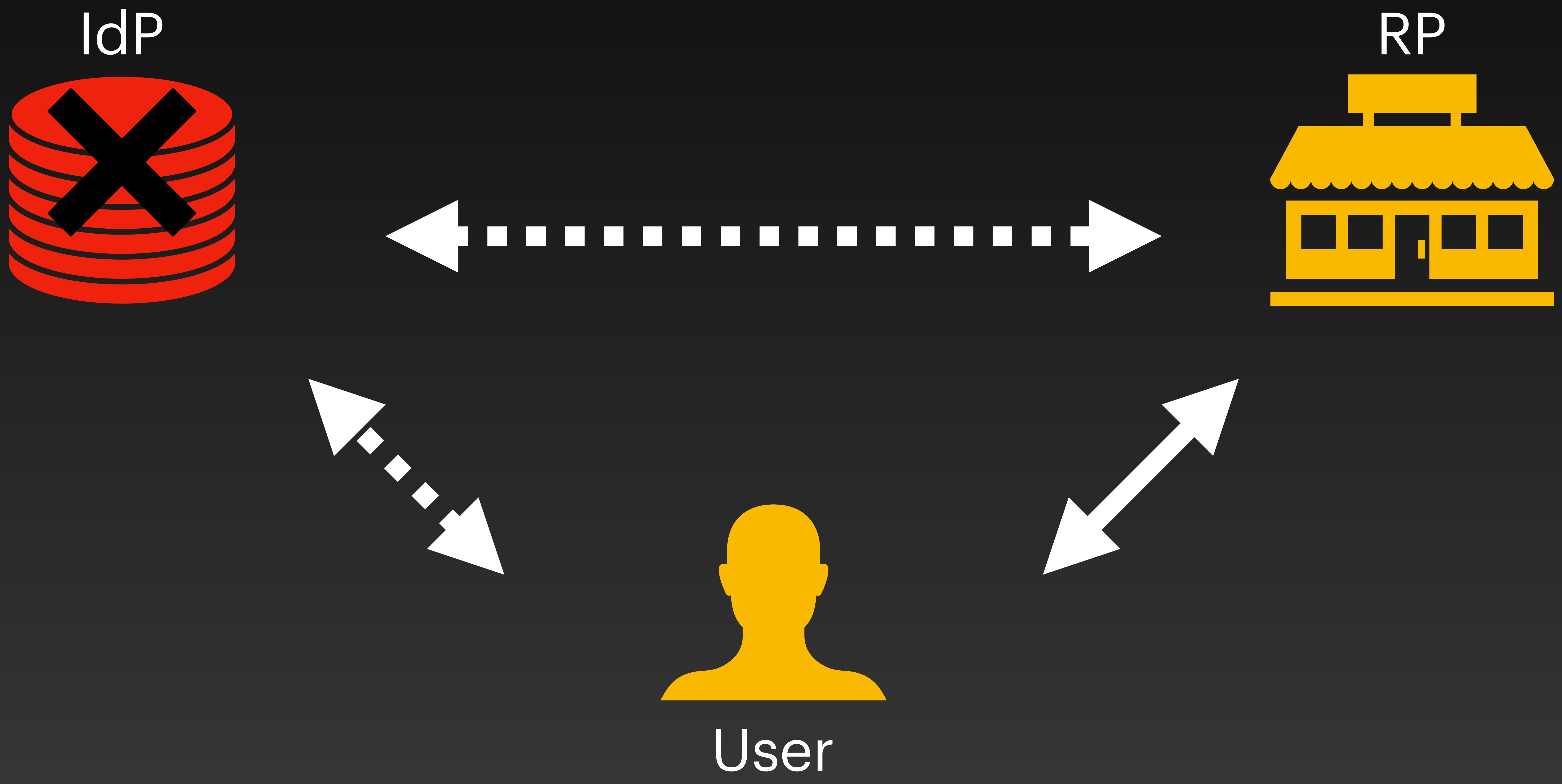
Standard SSO

Limitation I - Poor user and RP privacy



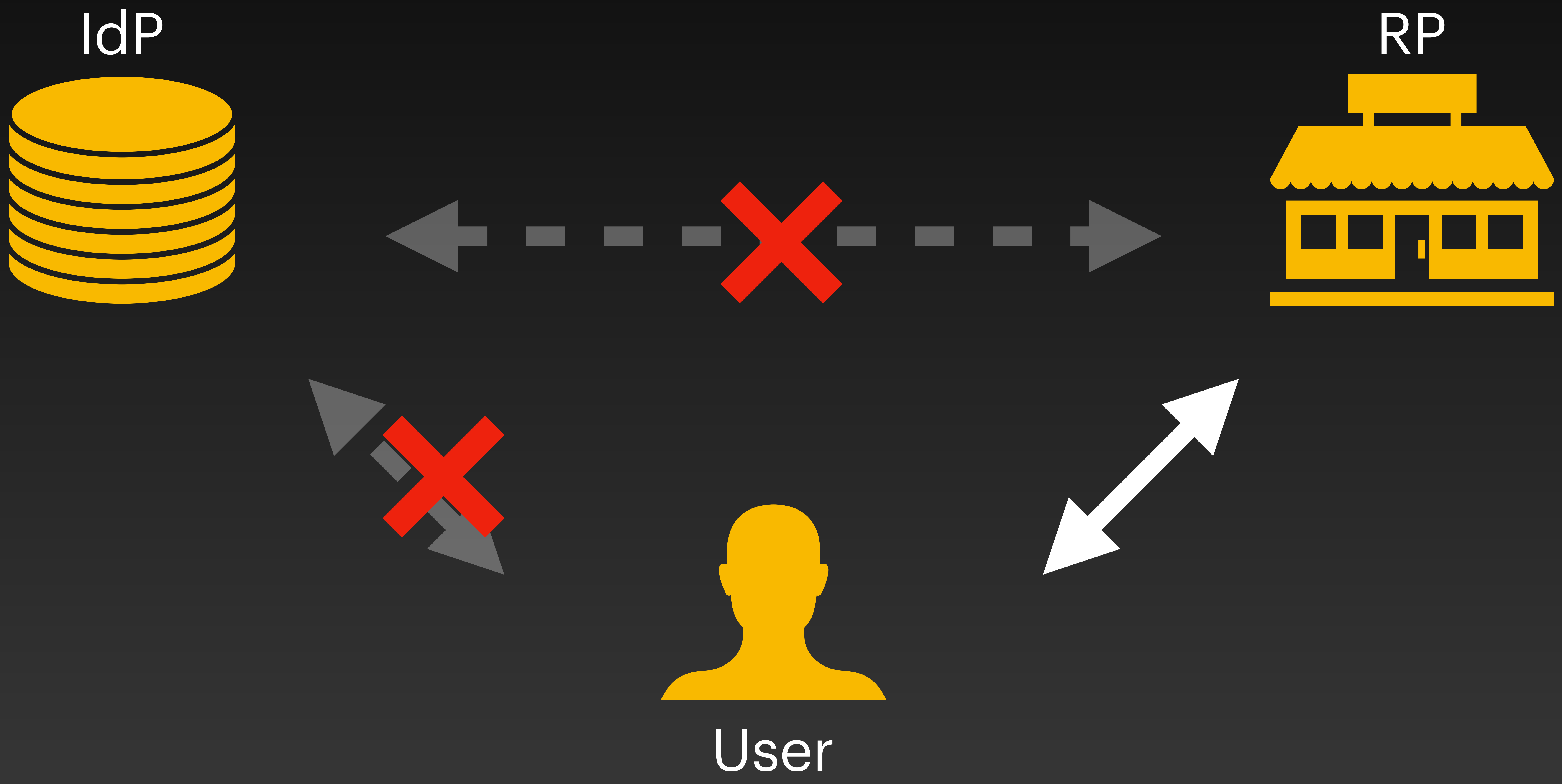
Standard SSO

Limitation II - Requires IdP availability



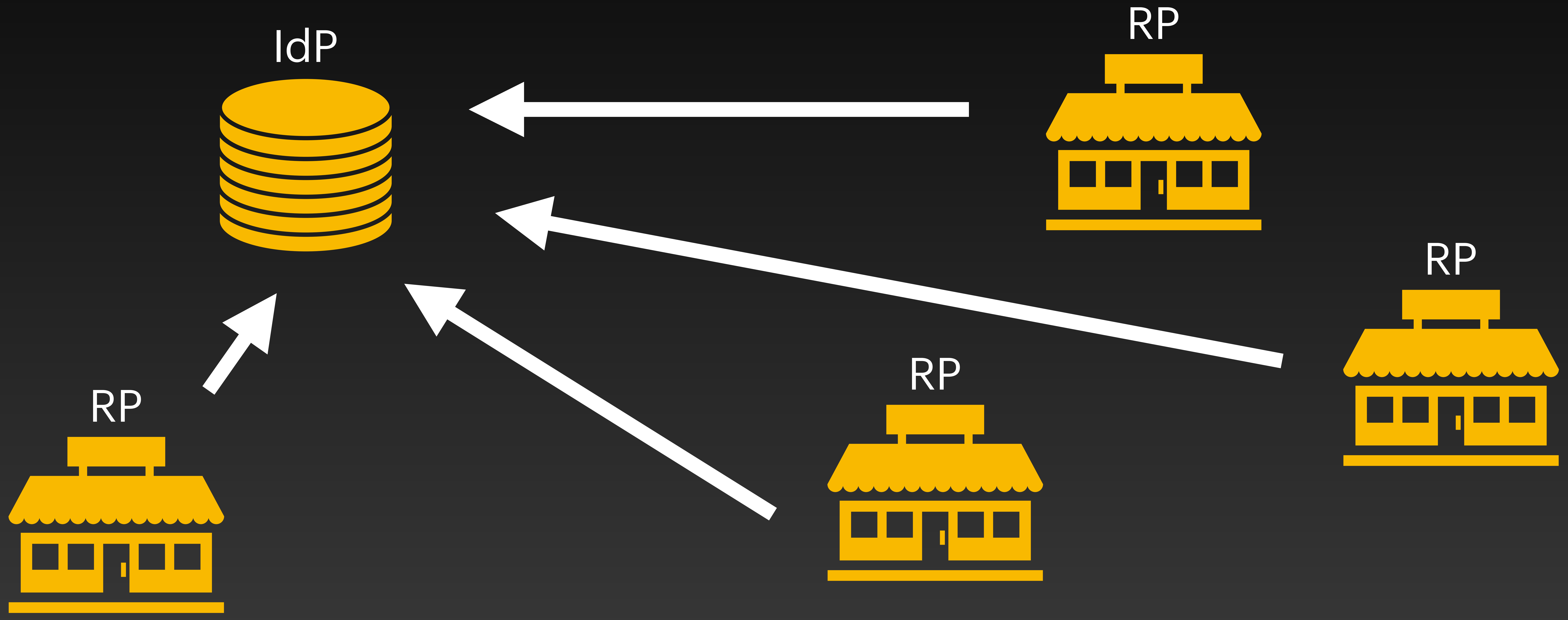
Standard SSO

Limitation III - Does not work offline



Standard SSO

Limitation III - Requires RP registration

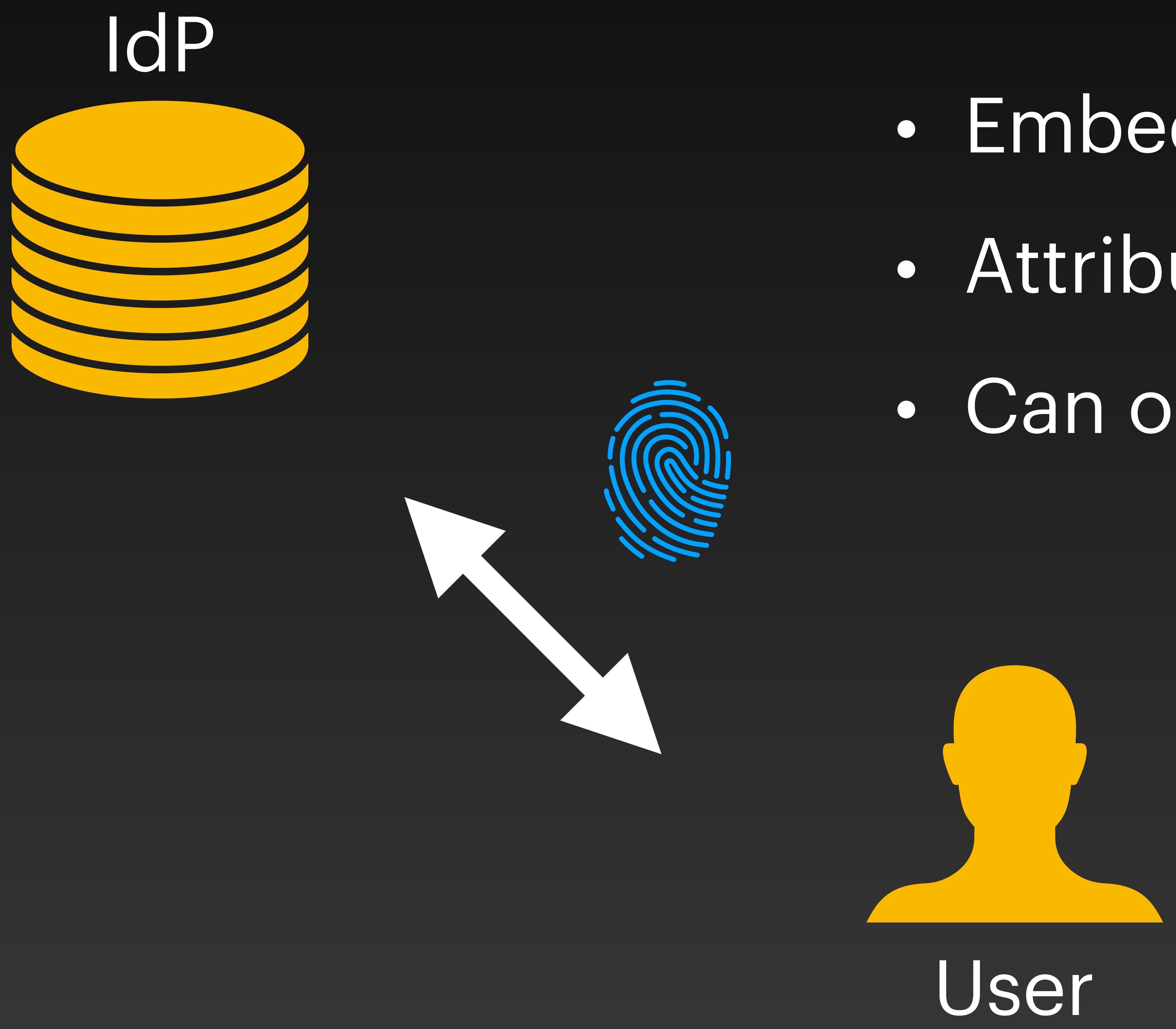


Anonymous Credentials

This is not a new idea

Anonymous Credentials

Setup phase

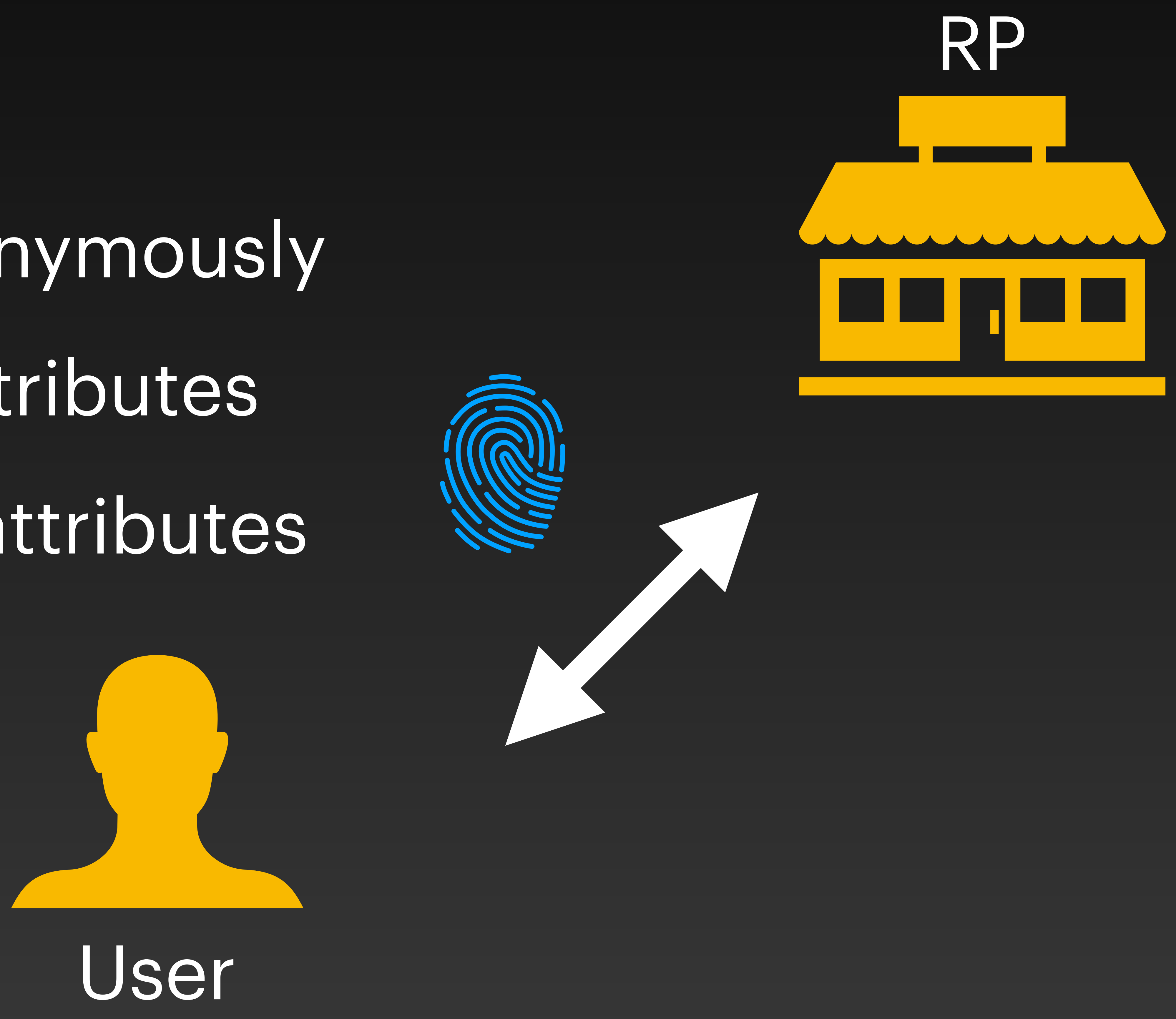


- Embed many user attributes (eg. email)
- Attributes are 'attested' by the IdP
- Can only be issued by the IdP

Anonymous Credentials

Sign-on phase

- No interaction with the IdP
- Can re-use the credential anonymously
- Can selectively show some attributes
- Can prove statements about attributes



What we get

Anon. Credentials

- ✅ Privacy
- ✅ Availability
- ✅ RP and user can be offline
- ✅ RP do not register with IdP

What's the catch?

Anon. Credentials

- ✅ Privacy
- ✅ Availability
- ✅ RP and user can be offline
- ✅ RP do not register with IdP

Standard SSO

- ✅ User usability
- ✅ Performance

EL PASSO

Privacy-preserving, Asynchronous Single Sign-On

What is it?

It is a system contribution (no new crypto)

Anon. Credentials

What is it?

It is a system contribution (no new crypto)

Anon. Credentials

with:

- User usability
- Performance

} **Standard SSO**

What is it?

It is a system contribution (no new crypto)

Anon. Credentials

with:

- User usability
- Performance

} **Standard SSO**

- (Optional) Accountability

Features

User Usability

- Implemented in C++ using MCL crypto library
- User-side client ported to javascript using WebAssembly (Wasm)

Features

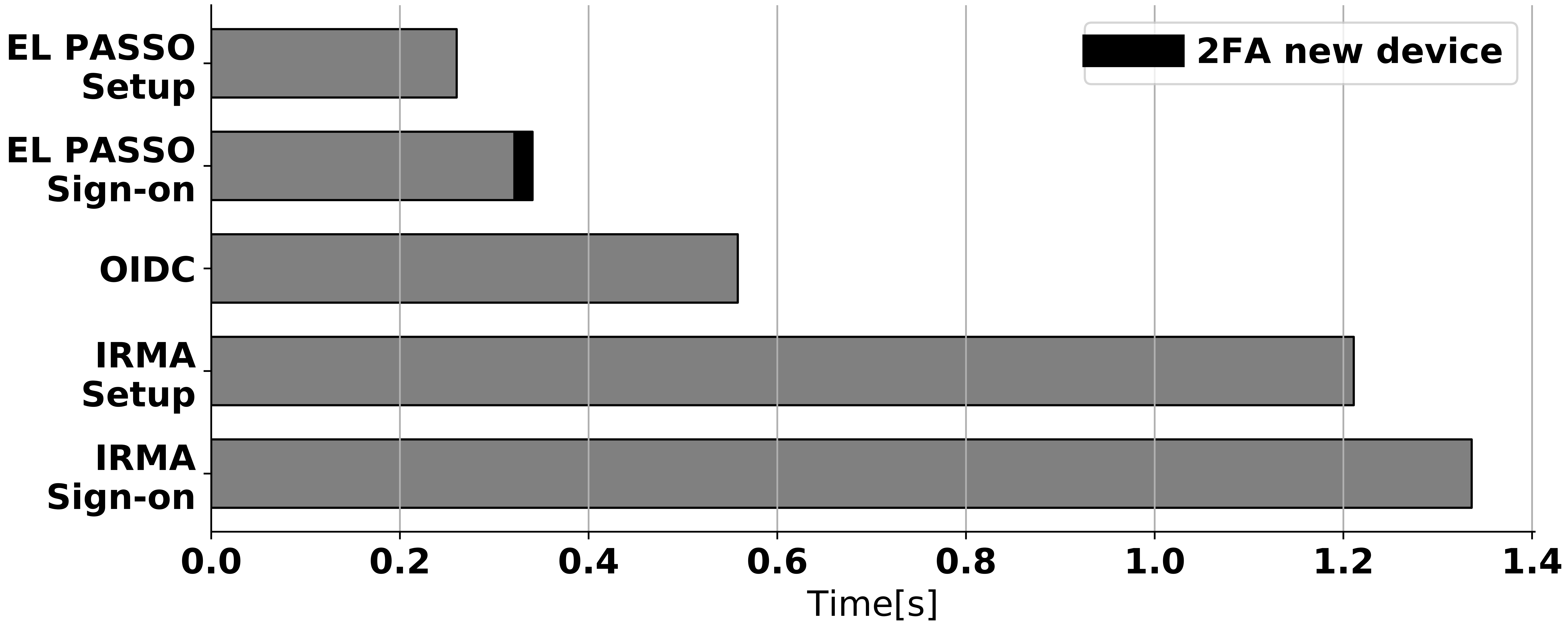
User Usability

- Implemented in C++ using MCL crypto library
- User-side client ported to javascript using WebAssembly (Wasm)

- Executable footprint: 178KB (including Wasm bin, js glue code)
- All user-side operations are handled by Wasm in the browser
- Wasm module cached, marked immutable, sandboxed
- User secrets stored in the browser's password manager
- User state: 600 bytes (3 attributes)

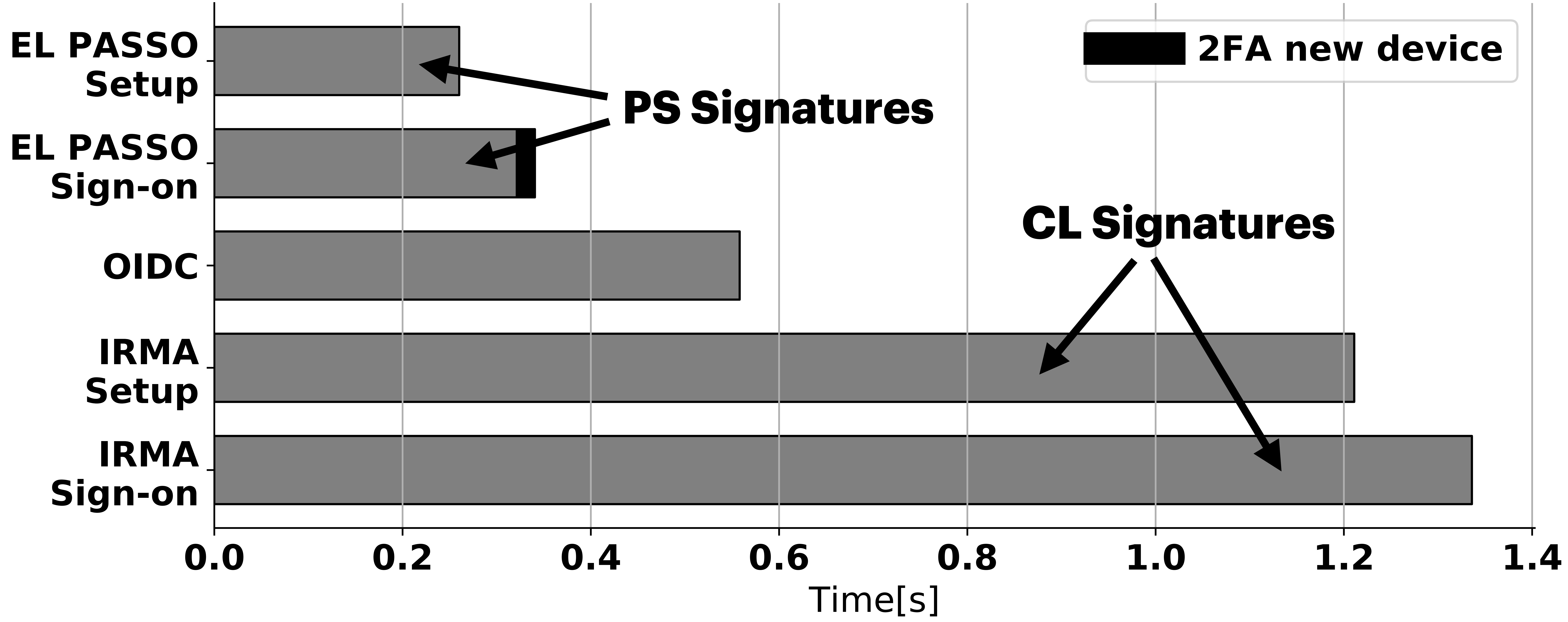
Features

Performance



Features

Performance



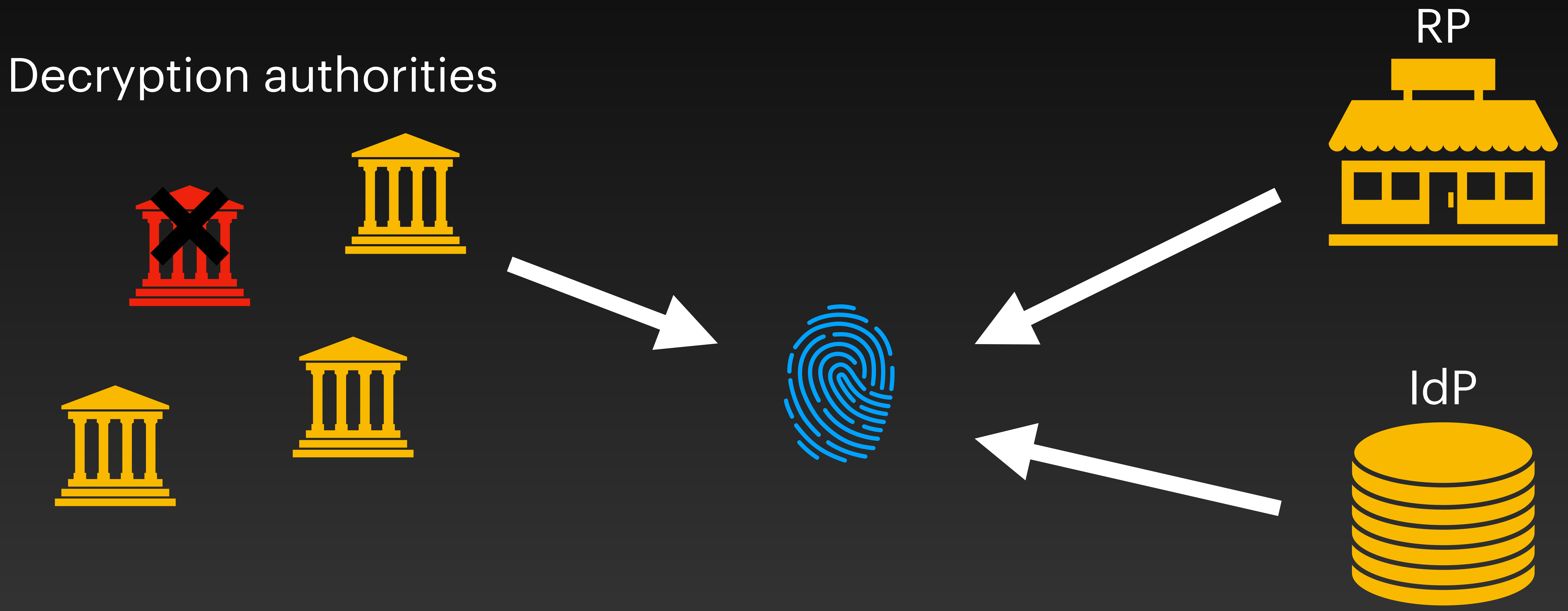
Features
Performance

Low-end devices

Operation	Latency [s]	CPU time @ user [s]
EL PASSO Setup	0.72±0.16 (+190%)	0.11±0.001 (+397%)
EL PASSO Sign-on	0.82±0.18 (+125%)	0.18±0.004 (+262%)
OIDC	0.80±0.02 (+45%)	NA
IRMA Setup	30.295±0.39 (+2420%)	29.68±0.27 (+4390%)
IRMA Sign on	34.182±0.49(+2458%)	33.891±0.43 (+3640%)

Features

(Optional) Accountability



Additional Features

More in the paper

- Multi-device support
- 2FA support
- Device theft recovery
- Login as guest

Conclusion

EL PASSO

- **Paper:** https://arxiv.org/abs/2002.10289
- **Code:** https://github.com/Zhiyi-Zhang/PSSignature

asonnino@fb.com

Alberto Sonnino

EXTRA

Construction

Anonymous credentials

Setup Phase

$\text{PrepareBlindSign}(pk, M_h, \phi) \rightarrow (d, \Lambda, \phi)$

$\text{Sign}(sk, M_p, \Lambda, \phi) \rightarrow \tilde{\sigma}$

$\text{Unblind}(d, \tilde{\sigma}) \rightarrow \sigma$

Sign-on Phase

$\text{Prove}(pk, M_p, M_h, \sigma, \phi') \rightarrow (M_p, \Theta, \phi')$

$\text{Verify}(pk, M_p, \Theta, \phi') \rightarrow b$

Construction

Setup phase

RequestID$(s) \rightarrow \Lambda$

Cred.PrepareBlindSign$(pk, s) \rightarrow (d, \Lambda)$

ProvideID$(sk, \gamma, info, tp, \Lambda) \rightarrow \tilde{\sigma}$

Cred.BlindSign$(sk, (\gamma, tp, info), \Lambda) \rightarrow \tilde{\sigma}$

UnblindID$(d, \tilde{\sigma}) \rightarrow \sigma$

Cred.Unblind$(d, \tilde{\sigma}) \rightarrow \sigma$

Construction

Sign-on phase (prove Id)

ProveID$(pk, \sigma, \gamma, info, tp, dns) \rightarrow (\zeta, \Theta, M_p, \phi', f)$

$$info \rightarrow info_p, \quad info \rightarrow info_h$$

Construction

Sign-on phase (prove Id)

ProveID$(pk, \sigma, \gamma, info, tp, dns) \to (\zeta, \Theta, M_p, \phi', f)$

$$info \to info_p, \quad info \to info_h$$

$$\zeta = \left(H^*(dns)\right)^s$$

Construction

Sign-on phase (prove Id)

ProveID$(pk, \sigma, \gamma, info, tp, dns) \rightarrow (\zeta, \Theta, M_p, \phi', f)$

$info \rightarrow info_p$, $info \rightarrow info_h$

$$\zeta = \left(H^*(dns)\right)^s$$

$$M_p = (info_p, tp)$$

$$M_h = (s, \gamma, info_h)$$

Construction

Sign-on phase (prove Id)

ProveID$(pk, \sigma, \gamma, info, tp, dns) \rightarrow (\Theta, M_p, \phi'(\zeta, f))$

$info \rightarrow info_p$, $info \rightarrow info_h$

$\zeta = \left(H^*(dns)\right)^s$

$M_p = (info_p, tp)$

$M_h = (s, \gamma, info_h)$

Cred.Prove$(pk, M_p, M_h, \sigma, \phi') \rightarrow (\Theta, M_p, \phi')$

$$\phi' = \{\zeta = \left(H^*(dns)\right)^s \wedge f(info_h) = 1\}$$

Construction

Sign-on phase (verify Id)

VerifyID$(pk, M_p, \Theta, dns, \phi'(\zeta, f)) \rightarrow b$

Cred.Verify$(pk, \Theta, \phi'(\zeta, f)) \rightarrow b'$

$$b = (b' = 1 \ \wedge \ tp > now)$$

ζ is the user id