# zk-Proofs for EU ID Cards

•••

PSE Hacker House 2024

Raphaël (@0xSileo)
Philipp (@pmuens)

# Research question

*How to prove partial identity data ?*

# Solution

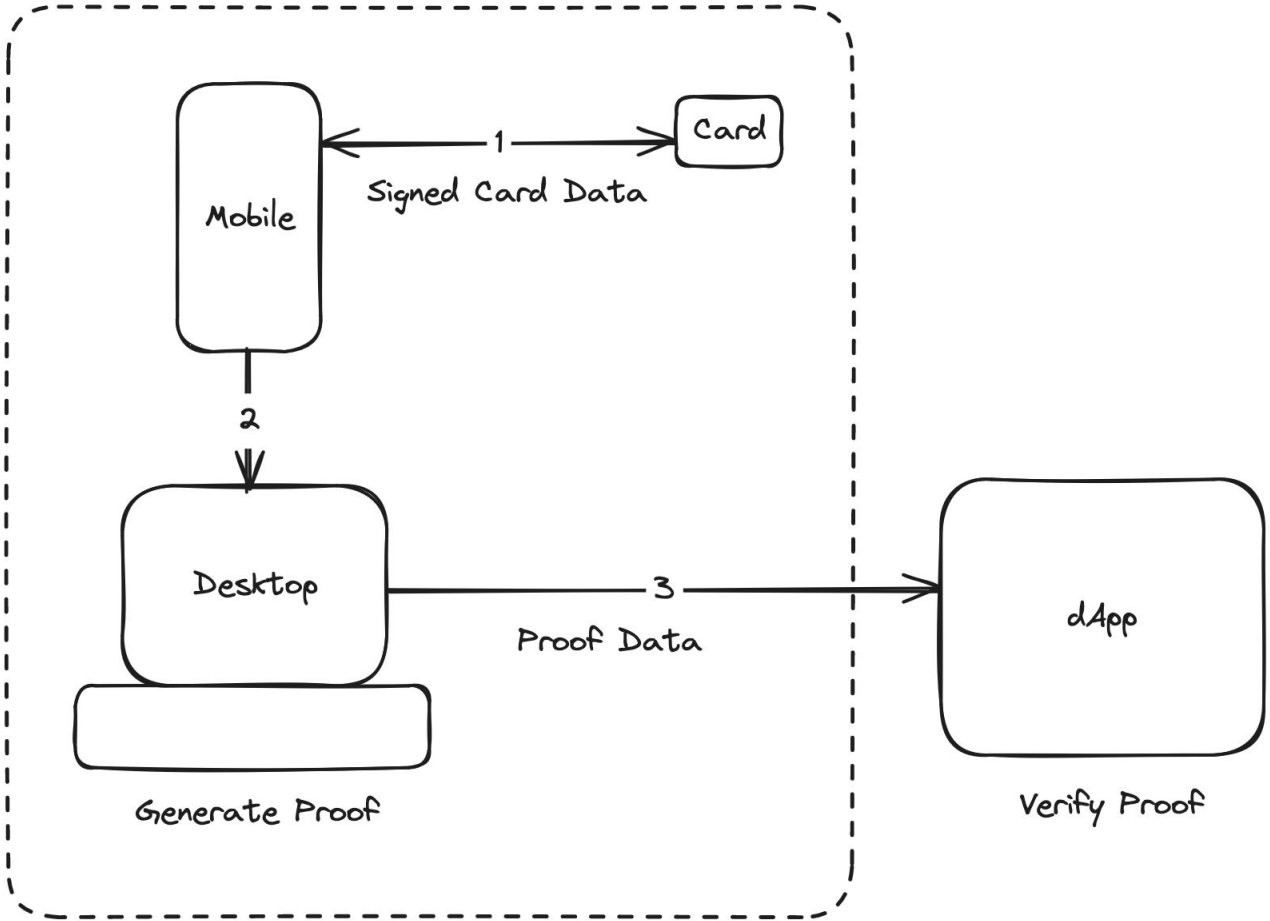
- Use authentication methods that were implemented by governments
- Generate a zk-Proof over this government-attested data
- zk-Proof can be verified by anyone without learning any personal data
- Root of trust is still the government as they signed the data the proof was generated over

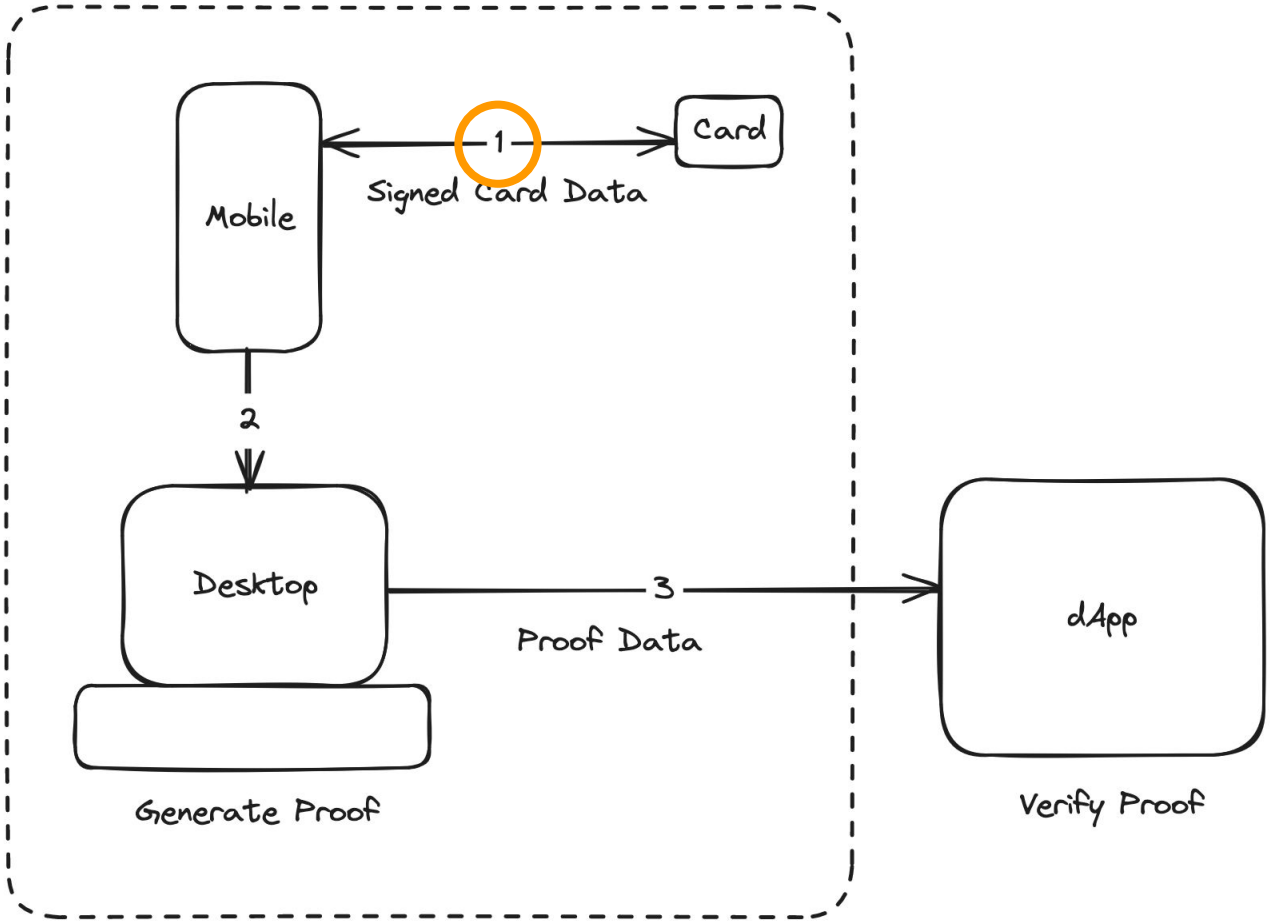
# How it works

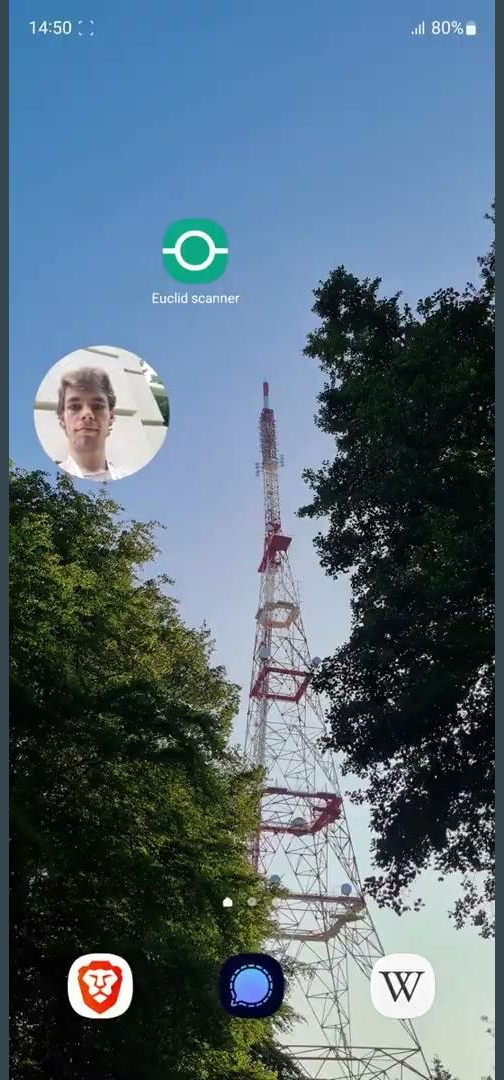
1. User visits dApp where proof of personhood is required (e.g. to do KYC)
2. User opens app on mobile device
3. User uses app to scan EU Government ID Card
4. NFC reader reads the signed data from the ID Card
5. Mobile phone generates proof over data from ID Card locally on the device
6. Proof can be sent to dApp which can easily verify it
7. If valid, user is granted access

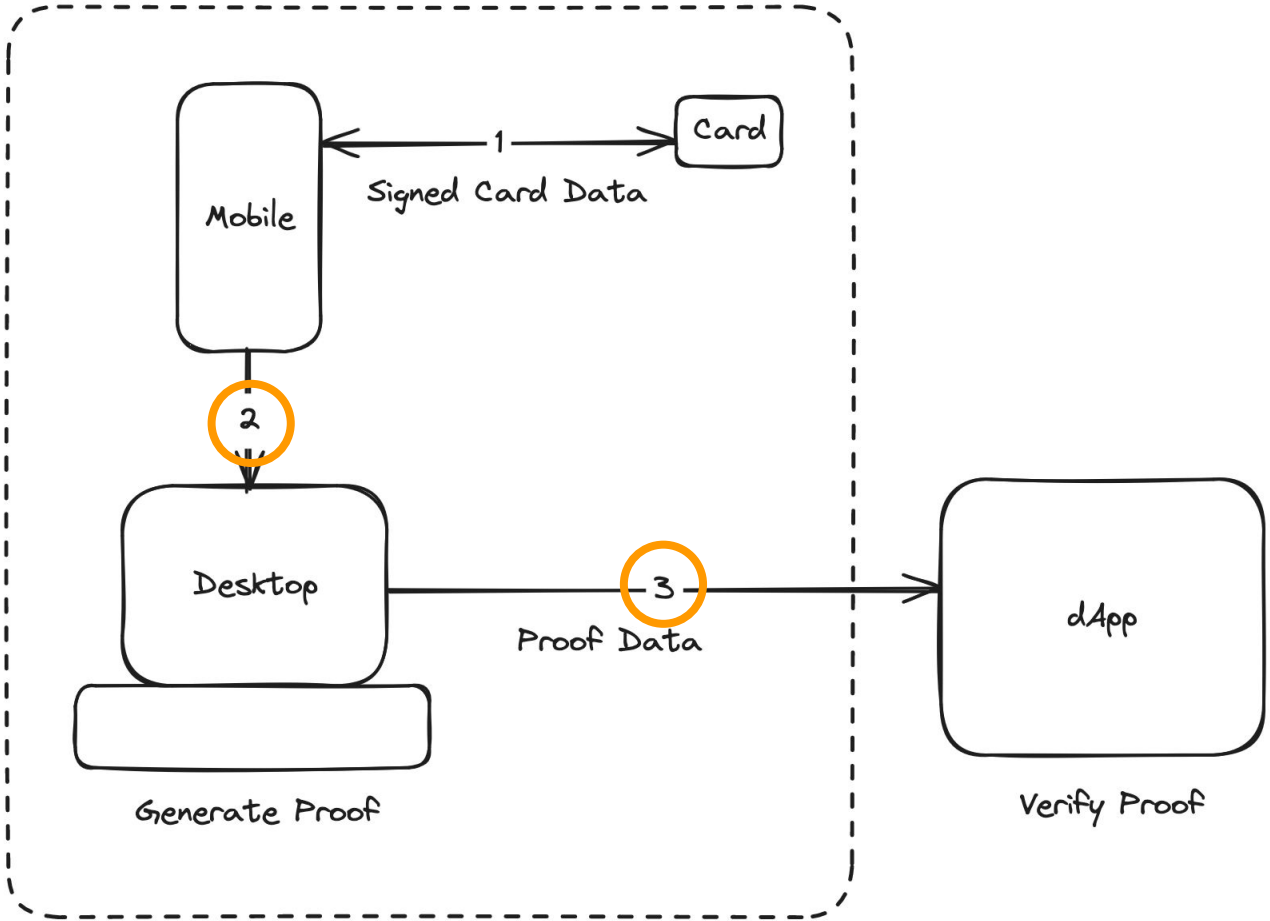
→ dApp doesn’t learn anything about the user ←

# Demo Time


Mobile
1
Card
Signed Card Data
2
Desktop
3
Proof Data
dApp
Generate Proof
Verify Proof


Mobile
1
Card
Signed Card Data
2
Desktop
3
Proof Data
dApp
Generate Proof
Verify Proof


14:50
80%
Euclid scanner


Mobile
1
Card
Signed Card Data
2
Desktop
3
Proof Data
dApp
Generate Proof
Verify Proof

# Challenges

1. Lack of proper test data when starting the project
2. EU Government IDs vary when it comes to the cryptography that's used (e.g. RSA with 2048 bits vs. RSA with 4096 bits)
3. Reading NFC data can be very error prone as different cards support different features which might or might not be implemented by libraries
4. We've used a library for on-device proof generation that's still in its early stages
5. Code to translate the SOD data into the proper binary format was written in JavaScript but we had to use Kotlin for our Android app
6. Long-running Browser processes are killed by mobile OS arbitrarily
7. Compile times / time to do the Trusted Setup per circuit
8. Integrating many different forks of libraries we've built upon
9. Need for a paid Apple Developer account to be able to build NFC apps
10. Working with sensitive data means the computation can't be outsourced

# Future Improvements

- Go full mobile
- Implement selective disclosure
- Add support for more EU Government-Issued ID Cards
- Request signature from ID Card / use as wallet
- Implement use cases for generated proofs

# Use cases

- Alcohol vending machines
- Children-friendly social network
- Semaphore integration (proof of lawfulness)
- E-Voting
- Deathtech

# Shoutouts

- The whole **PSE** Team with its mentors who helped tremendously to debug issues and troubleshoot in general throughout the week
- The **Mopro** Team for their library we're using for our native apps
- The **zkPassport** Team for their NFC scanning code we studied and use
- The **zk E-Mail** Team for their Circom circuits we're using
- The **Anon Aadhaar** Team for their anon-eu project which our project is based on
- All the **Open Source Maintainers** whose libraries we're using
- **Estelle, Raphaël's girlfriend** for lending us her ID card

Webapp: https://sileo.dev/Euclid
Code: https://github.com/0xSileo/Euclid (diff)

—

Thank You!

Questions?