
proxy

- Who we are
- What we are building
- Bridging URL and IRL
- Current and future challenges

# Who we are

- Hardware × Software
- Principles~
  - Privacy first
  - User always in control
- Six years of building reader hw, wearable hw, and phone “wallets” for enterprise access control (employee badges)
- Brought mobile access control to physical security industry~
  - `Access for Apple Wallet` ~ launch partner
  - `CSA ACWG` ~ technical vice-chair (https://csa-iot.org/all-solutions/#access-control)

# What we are building

- Non-custodial wallet for digital identity and digital assets
- Wearable hardware wallet (NFC, BLE, SE)
- Mobile software wallet
- Complementary, a balance of convenience, robustness, security

# Unique challenges

- Limited power and physical space
- Limited i/o
- High degree of integration
    - Fewer component options, less design flexibility
- Table stakes for functionality~
    - Must be multi-purpose, little appetite for single function wearable devices
    - Must support existing use cases, work on existing infrastructure

# URL and IRL

## Experience × Integration × Security

Wallets Abound - The Daily Gwei #488

- People have 80+ existing identifiers
- People have existing things they do with them daily~
  - payments
  - transport, ticketing
  - physical access control (office, home, car)
  - logical access control (otp, passwordless)
  - digital ids (mDL, MRTD, vaccine passport)
  - cryptoassets (coins, tokens, LNURL)
- There exists both legacy and new terminal infrastructure

# Challenge: features

- SE limited in functionality
  - JavaCard for most SE applications
  - “New” curves
    - Some (*e.g.* `secp256k1`) can be done with current hw, but questions about side channel resistance
  - New algorithms (signatures, ciphers)
  - New protocols
    - *e.g.* `ISO18013-5` (mDL) finalised 2021-09
    - Structured request/response and signing schemes for selective disclosure, CBOR, COSE, AES-GCM
- Evolving much faster than traditional “secure” industries, hw vendors are chasing a rapidly moving target…

# Challenge: certs and costs

- SE constrained by cert requirements
  - certification is long and costly (EMVCo, GlobalPlatform, individual payment networks)
  - frozen hardware, OS, API, and applet code
- Vendors unwilling to make changes
- Vendors take forever to implement new things
  - `JCAPI 3.1` (2019) – who has implemented it?
  - `JCAPI 3.0.5` (2015) – far from universal
- NXP SE050E added AES-GCM/CCM, Curve448 in 2022 (!)~
  - likely take another 12mo before this is usable in a new product design by mere plebs

# Challenge: expertise

- Often don’t have all necessary expertise in-house
  - either a software house using off-the-shelf silicon,
  - or a silicon manufacturer or integrator outsourcing all software dev and security
- JavaCard OS, runtime, applet dev often by 3rd parties
- Lack of openness in docs and specs at integrator level often not by design, but a side-effect of too many cooks (“too hard to cut through all the contracts”)

# What should we do?

- Fastest way to support new things
  - general purpose compute with SE-level guarantees is the most future-proof and the most flexible, but...
- Co-exist with existing JavaCard and certified applets
- Integrated solutions should be flexible, modular
- Isolation levels
  - let some parts move faster than others
  - let everyone write code
    - open source, developers, dedicated small volume
- Can we help iterate faster or cheaper on hw implementations?


@proxy ~

@simonratner ~