Techniques for Implementing Derived Credentials

Francisco Corella (fcorella@pomcor.com)
Karen Lewison (kplewison@pomcor.com)
Pomcor (http://pomcor.com/)

Derived Credential

- Electronic Authentication Guideline: "*A credential issued based on a proof of possession of a PIV credential* "
- Motivation
  - Store credential in a mobile device
  - Use it instead of PIV card for *logical access* (authentication to information systems)
- http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2012-02/feb1_der_cred_ferraiolo_h_fips_201-2.pdf

Challenges

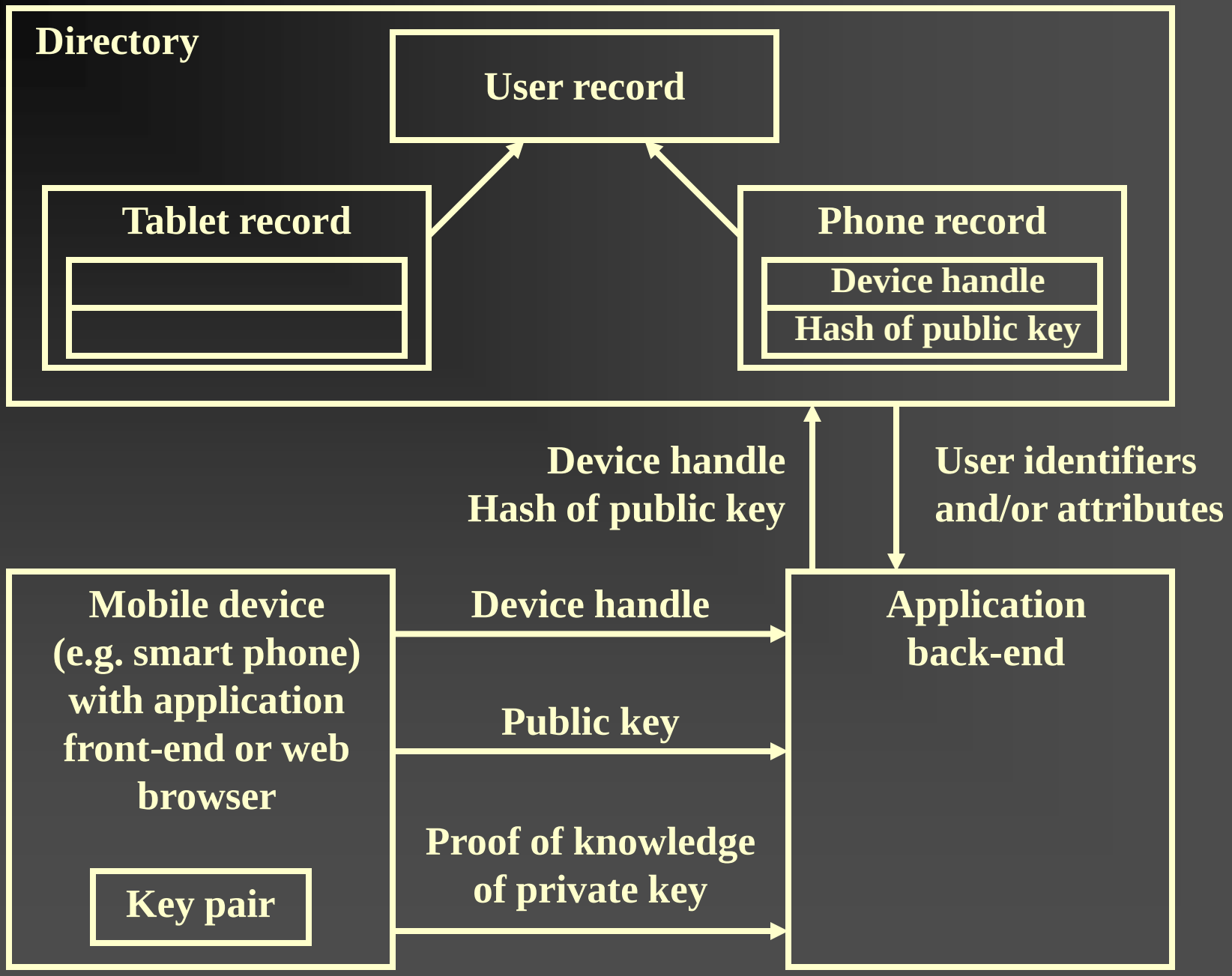
- Complexity of cryptographic and biometric processing for app developers
- No FIPS 140-2 Level 3 tamper resistant storage in mobile devices

Techniques for Addressing Challenges

1. Public key cryptography without certificates
2. Key pair regeneration as an alternative to tamper resistance
3. Encapsulation of cryptographic and biometric processing in black boxes

1. Public Key Cryptography without Certificates

- Mobile device → application (back-end):
  - Database handle of a device record that contains the hash of public key and refers to user record
  - Public key
  - Proof of knowledge of private key
- Application → directory
  - Database handle of device record
  - Hash of public key
- Directory → application
  - User identifier(s) and/or attribute(s)


Directory
User record
Tablet record
Phone record
Device handle
Hash of public key
Device handle
Hash of public key
User identifiers
and/or attributes
Mobile device
(e.g. smart phone)
with application
front-end or web
browser
Key pair
Device handle
Public key
Proof of knowledge
of private key
Application
back-end

2. Key Pair Regeneration as an Alternative to Tamper Resistance

- PIV card stores credentials in tamper-resistant storage
- But mobile devices do not have tamper-resistant storage
- ➔ Encrypt private key under key derived from PIN?
  - That would allow offline attack against PIN
- Instead we propose to regenerate the key pair from the PIN (or from a biometric key)
- ➔ All PINs produce well-formed key pairs, so PINs cannot be tested and offline attack is not possible

RSA Key Pair Regeneration from a PIN

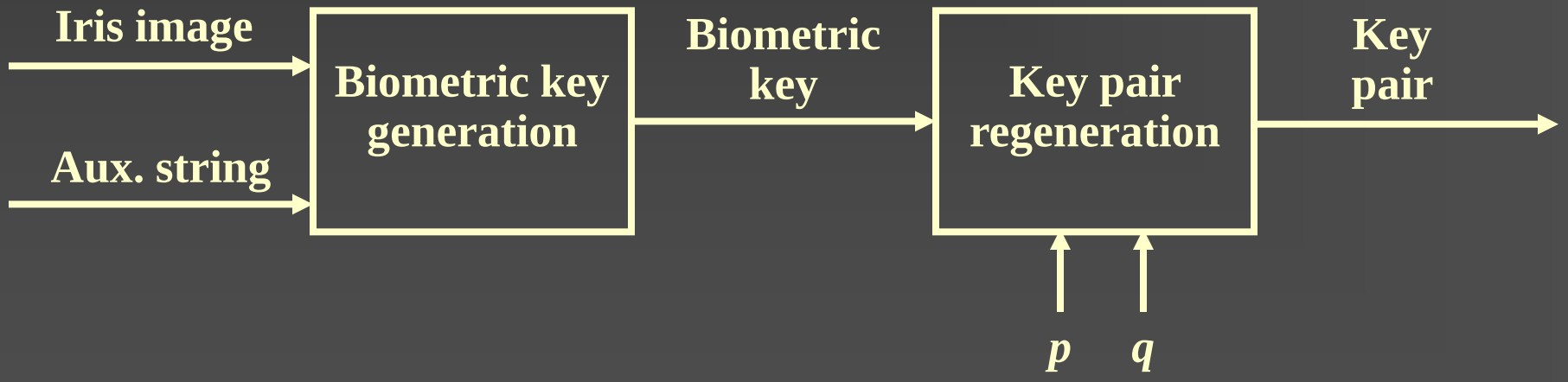
- Idea
  - Store *p*, *q* in device, but not *e* or *d*
  - Generate *d* as a randomized hash of the PIN, of same length as the modulus
  - Compute *e* such that $1 < e < \varphi$ and $ed \equiv 1 \pmod{\varphi}$
- Problem: what if $\gcd(d,\varphi) \neq 1$?
- Solution:
  - Remove from *d* all prime factors $r < 100$ shared with φ.
  - During initial key generation, if *d* has prime factors $r' > 100$ shared with φ, we start over with different *p* and *q* (probability: 0.2%)

RSA Key Pair Regeneration from a PIN (Continued)

- Non-problem:
  - Retaining *p* and *q* does not reduce security (they could be computed from the key pair)
- Non-problem:
  - *d* not vulnerable to small-decryption-exponent attacks

Regeneration from Biometric Key

- Biometric key generated from an iris image (to be taken by device camera) and an auxiliary string
  - *F. Hao, R. Anderson, and J. Daugman. Combining Cryptography with Biometric Effectively. IEEE Trans. Comput., 55(9):1081-1088, 2006.*
  - Biometric template not at risk because not used

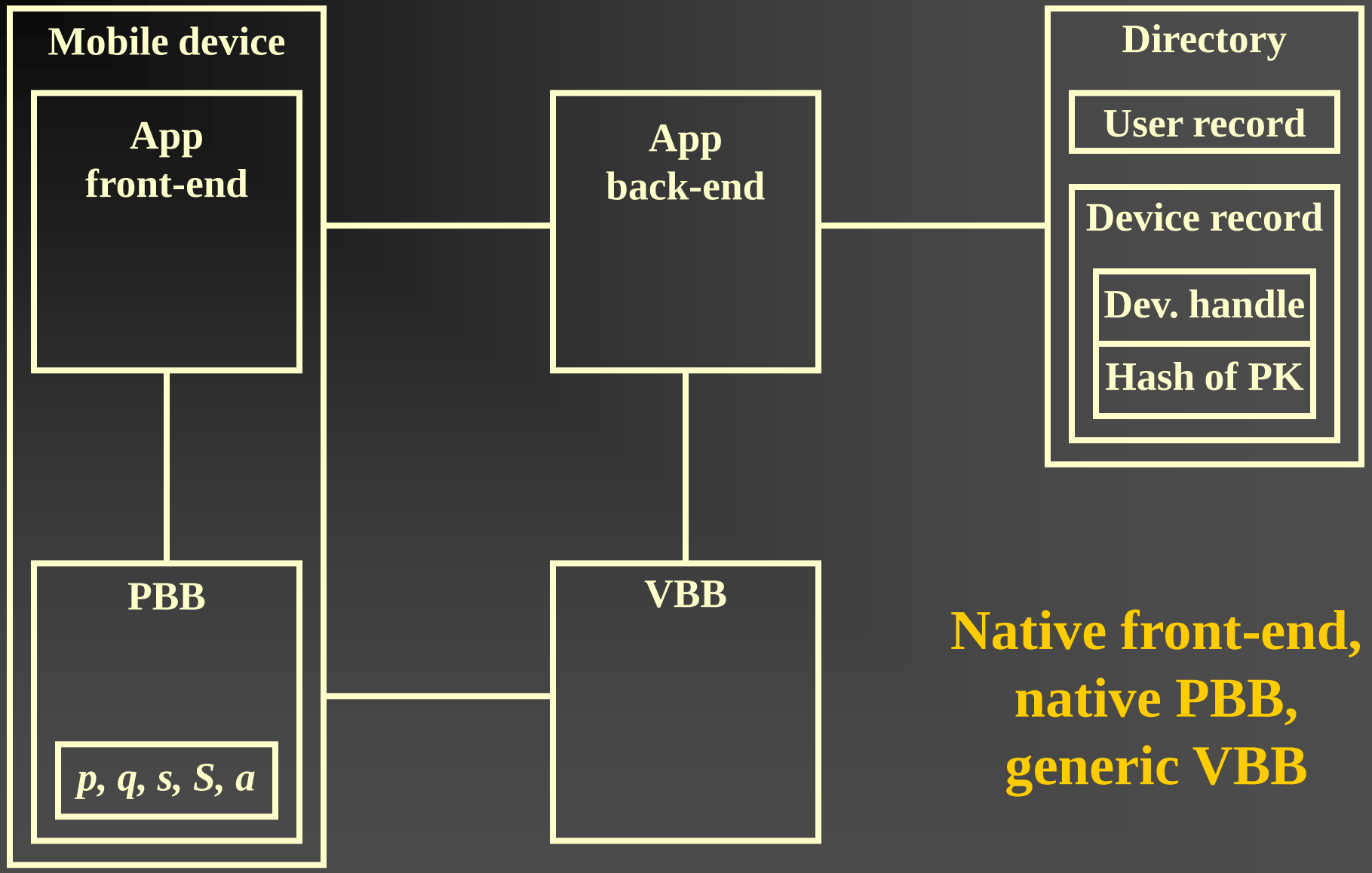


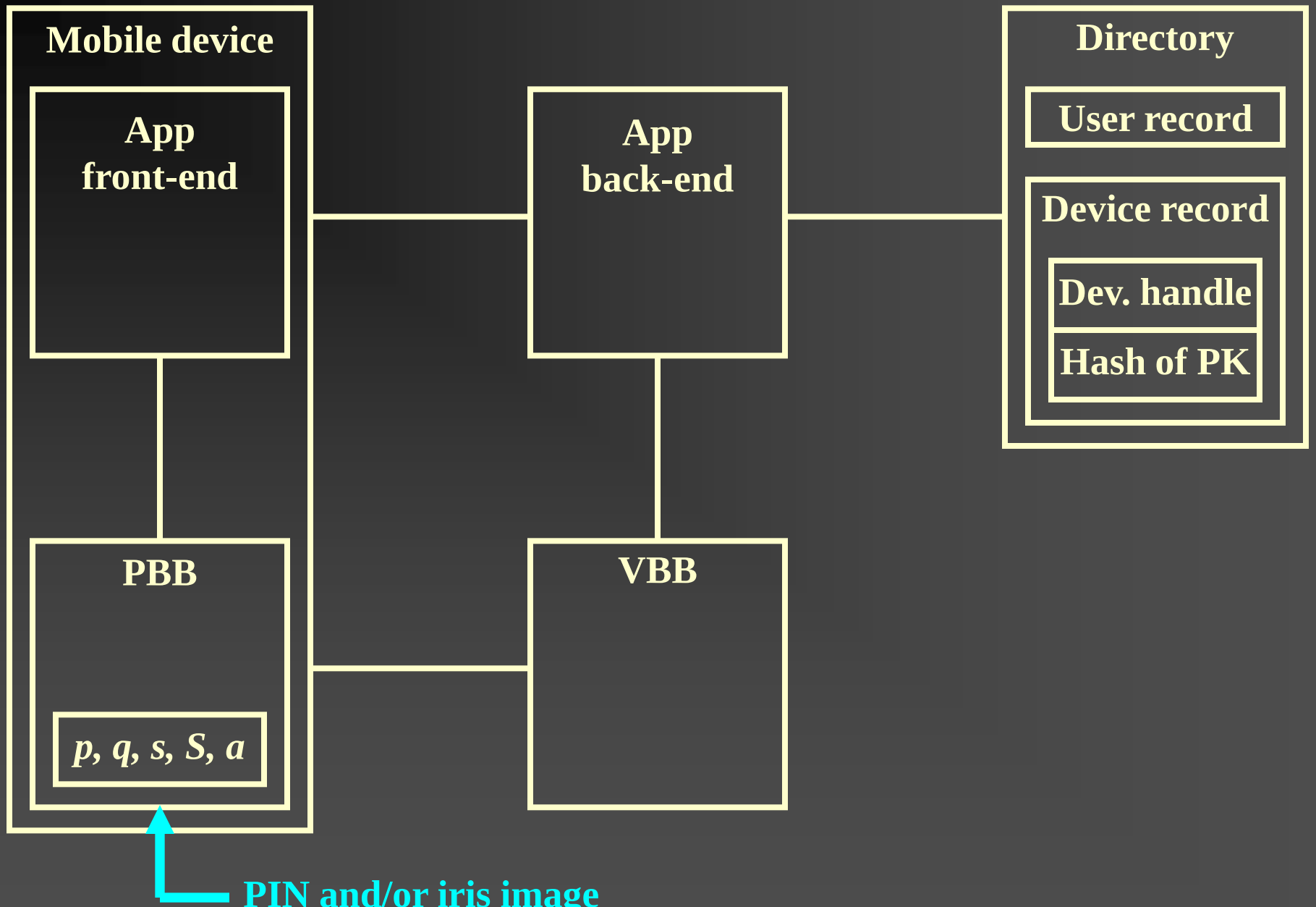
Three-Factor Authentication

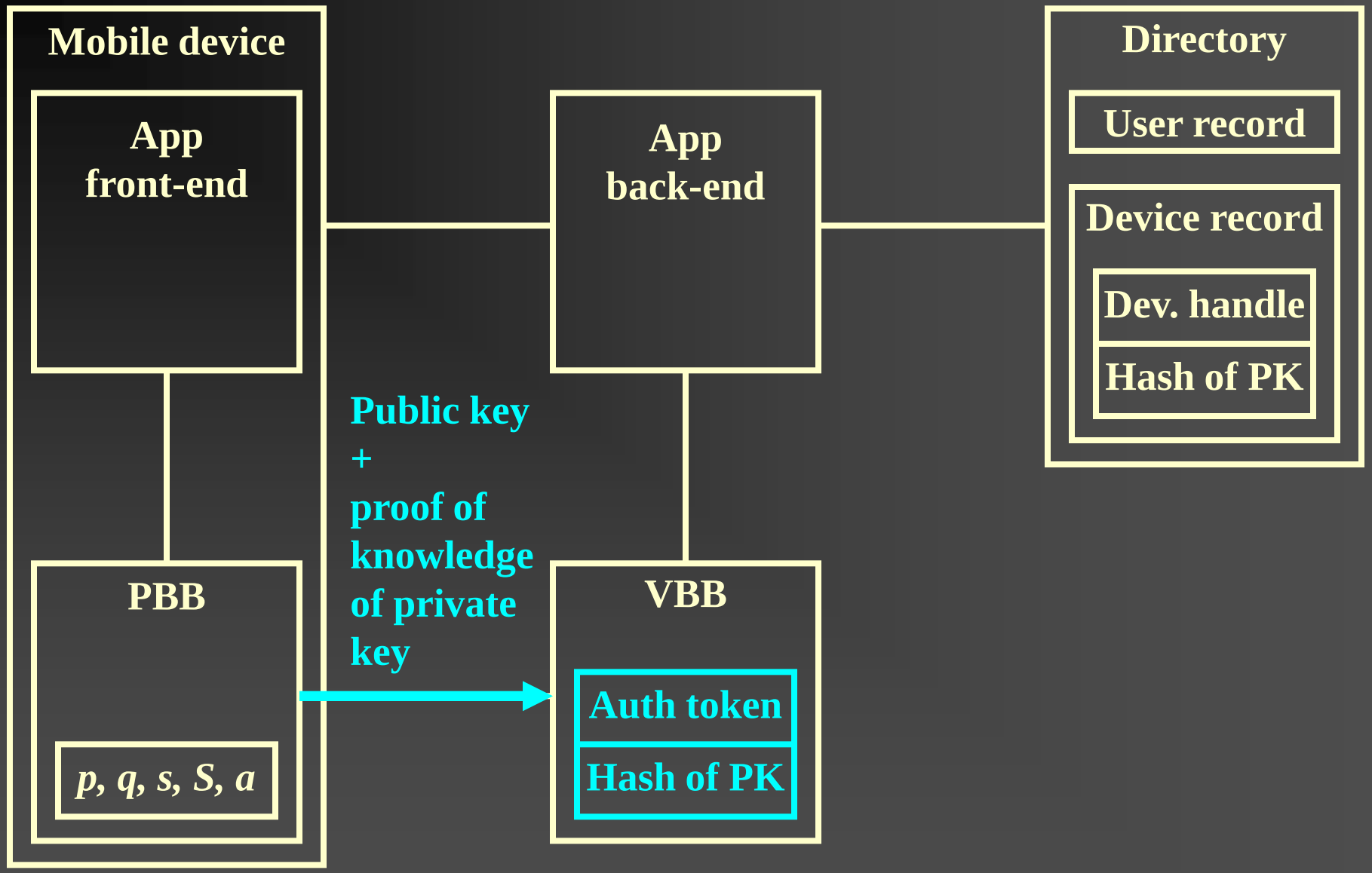
- Key pair + PIN + iris image
- Biometric key used to regenerate key pair
- PIN used to
  - Encrypt auxiliary string, or
  - Scramble the biometric key generation algorithm (suggested by Hao et al.)

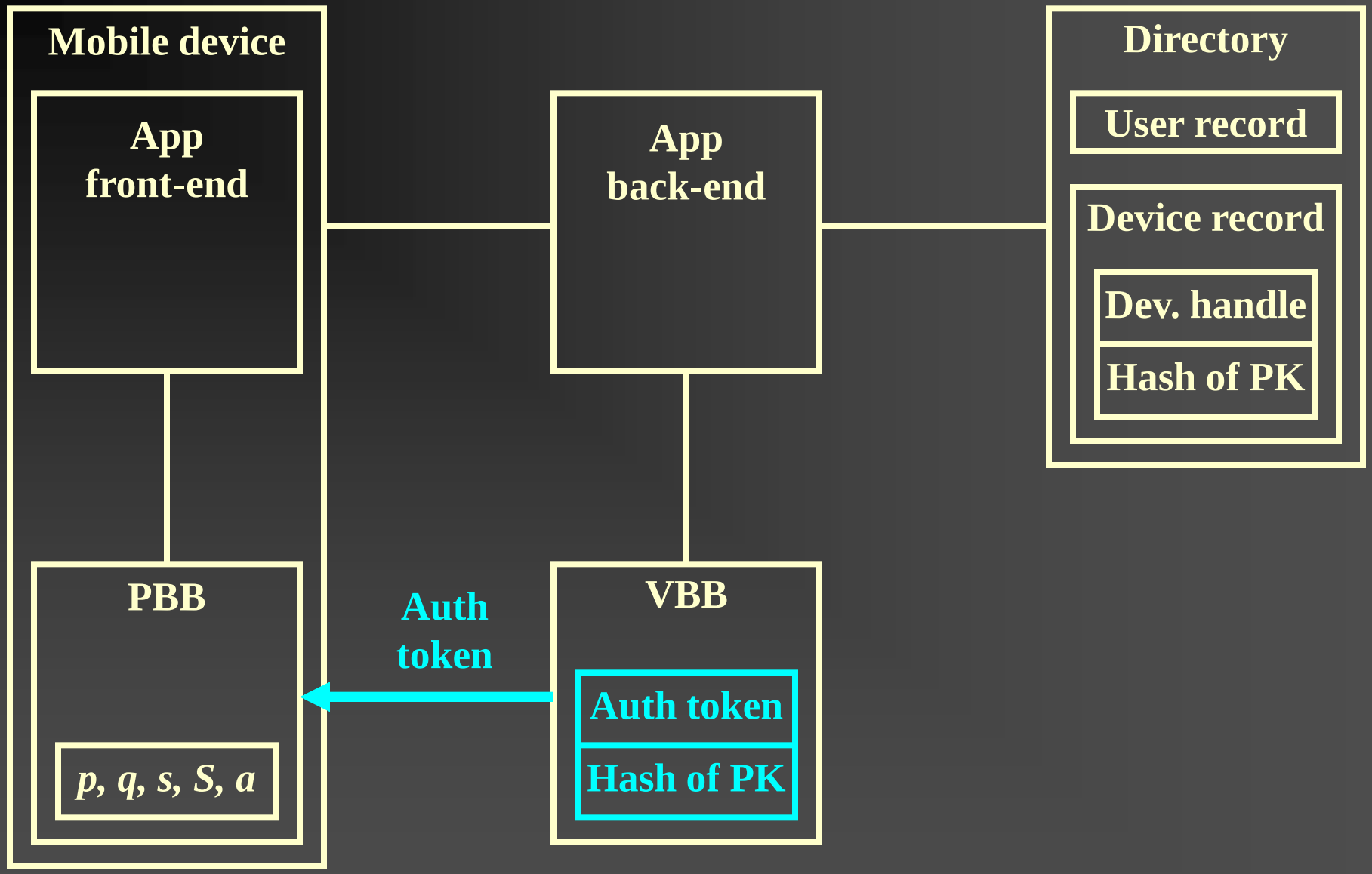
3. Encapsulation of Cryptographic and Biometric Processing

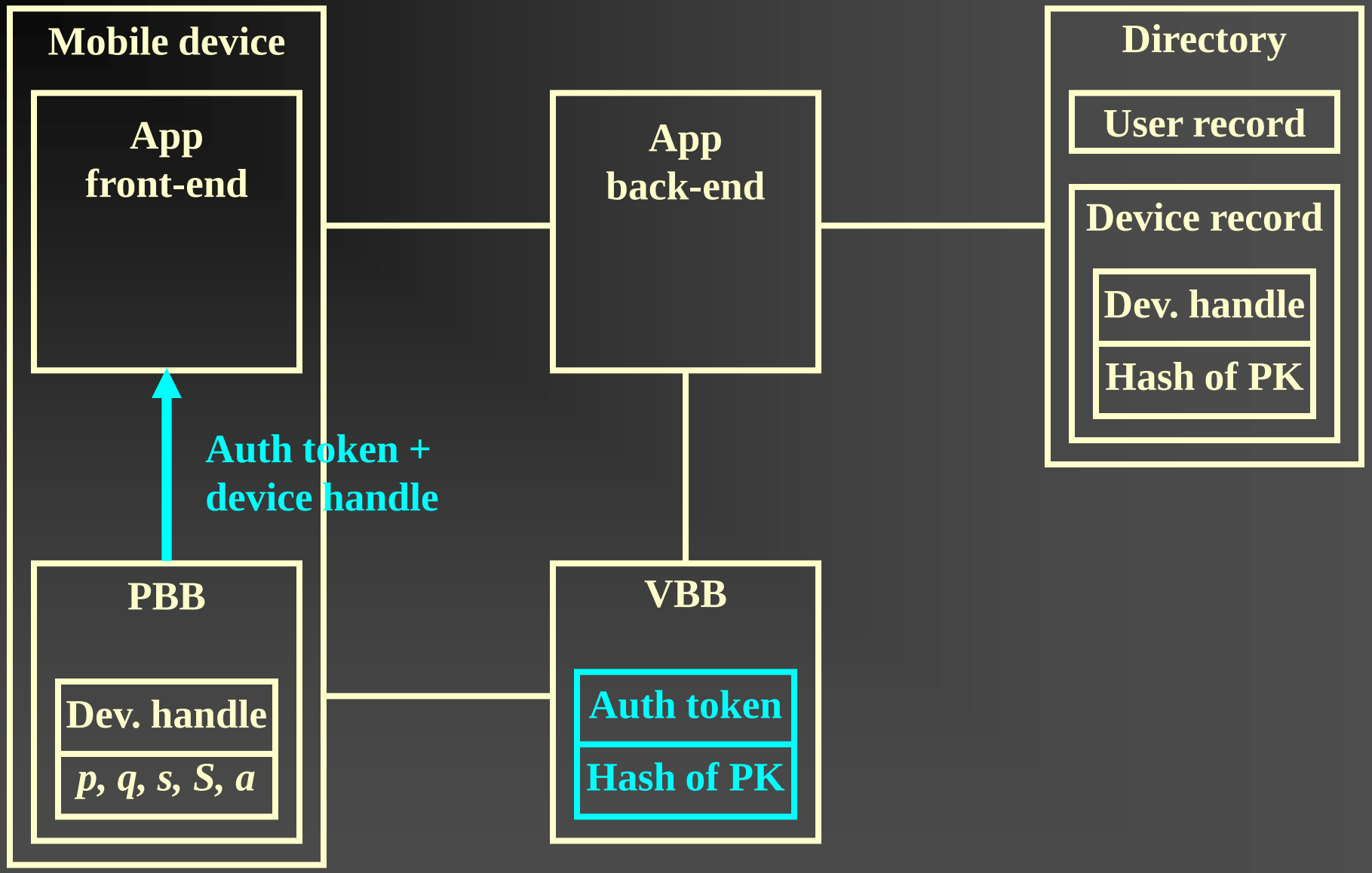
- Application outsources cryptographic and biometric complexities to a Prover Black Box (PBB) and a Verifier Black Box (VBB)
- PBB is in mobile device
- VBB online, trusted by application
  - Could be implemented as a generic server appliance
- Many possible configurations
  - In some configurations, outsourcing protocol uses “native URLs” (available in iOS and Android) for interapp communications within the device

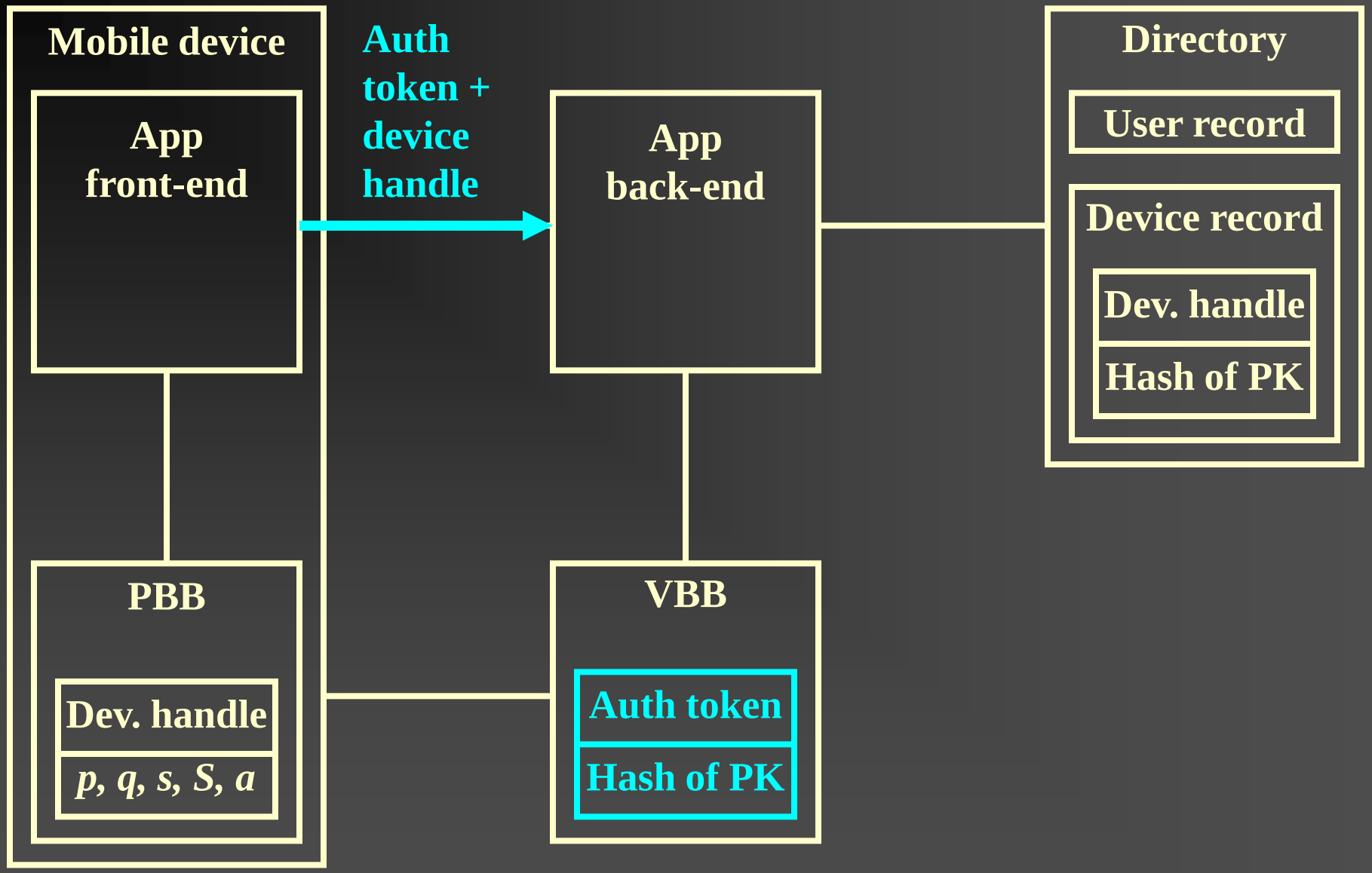

Mobile device
App front-end
PBB
p, q, s, S, a
App back-end
VBB
Directory
User record
Device record
Dev. handle
Hash of PK
Native front-end, native PBB, generic VBB

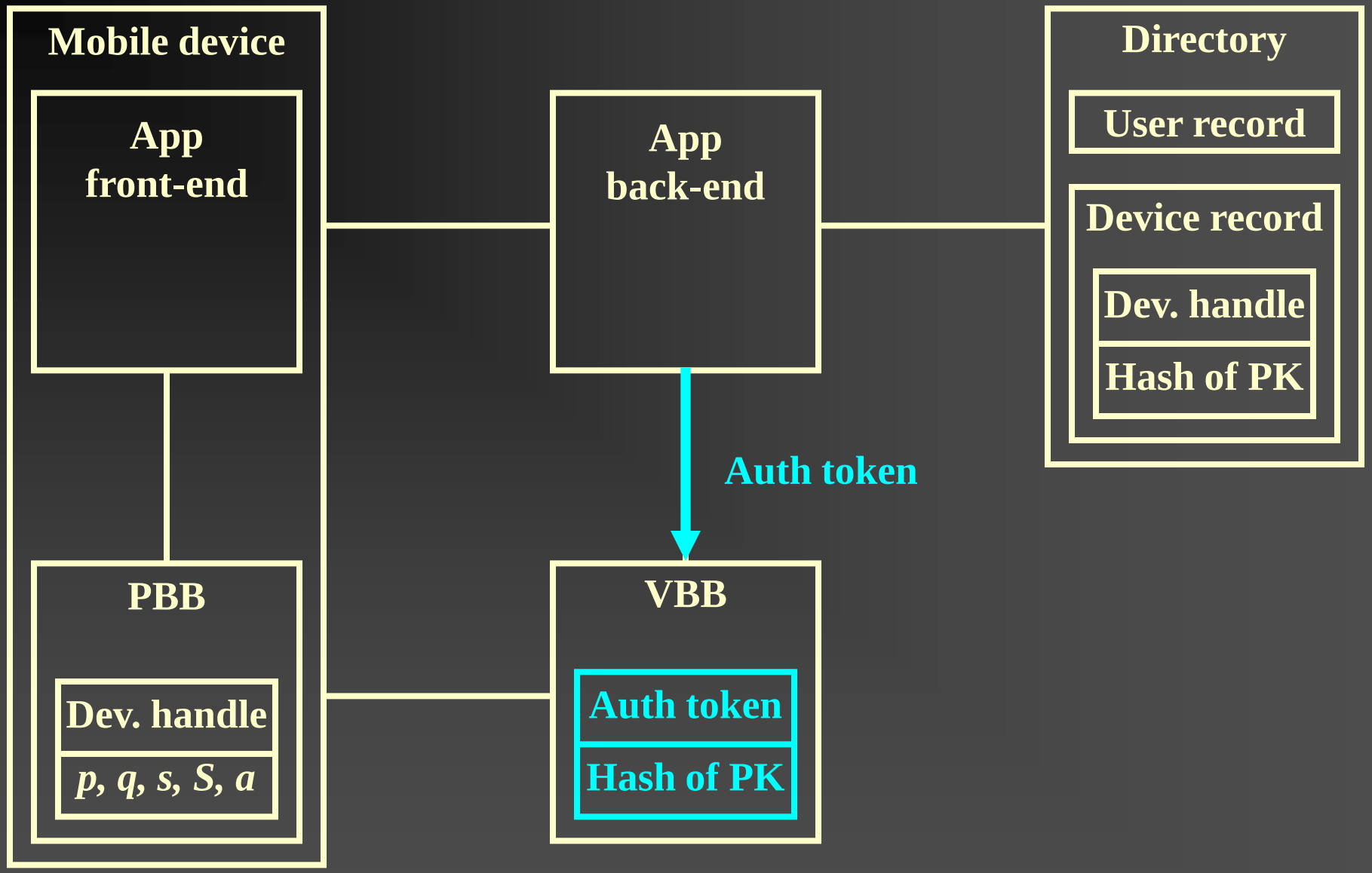

Mobile device
App front-end
PBB
p, q, s, S, a
App back-end
VBB
Directory
User record
Device record
Dev. handle
Hash of PK
PIN and/or iris image

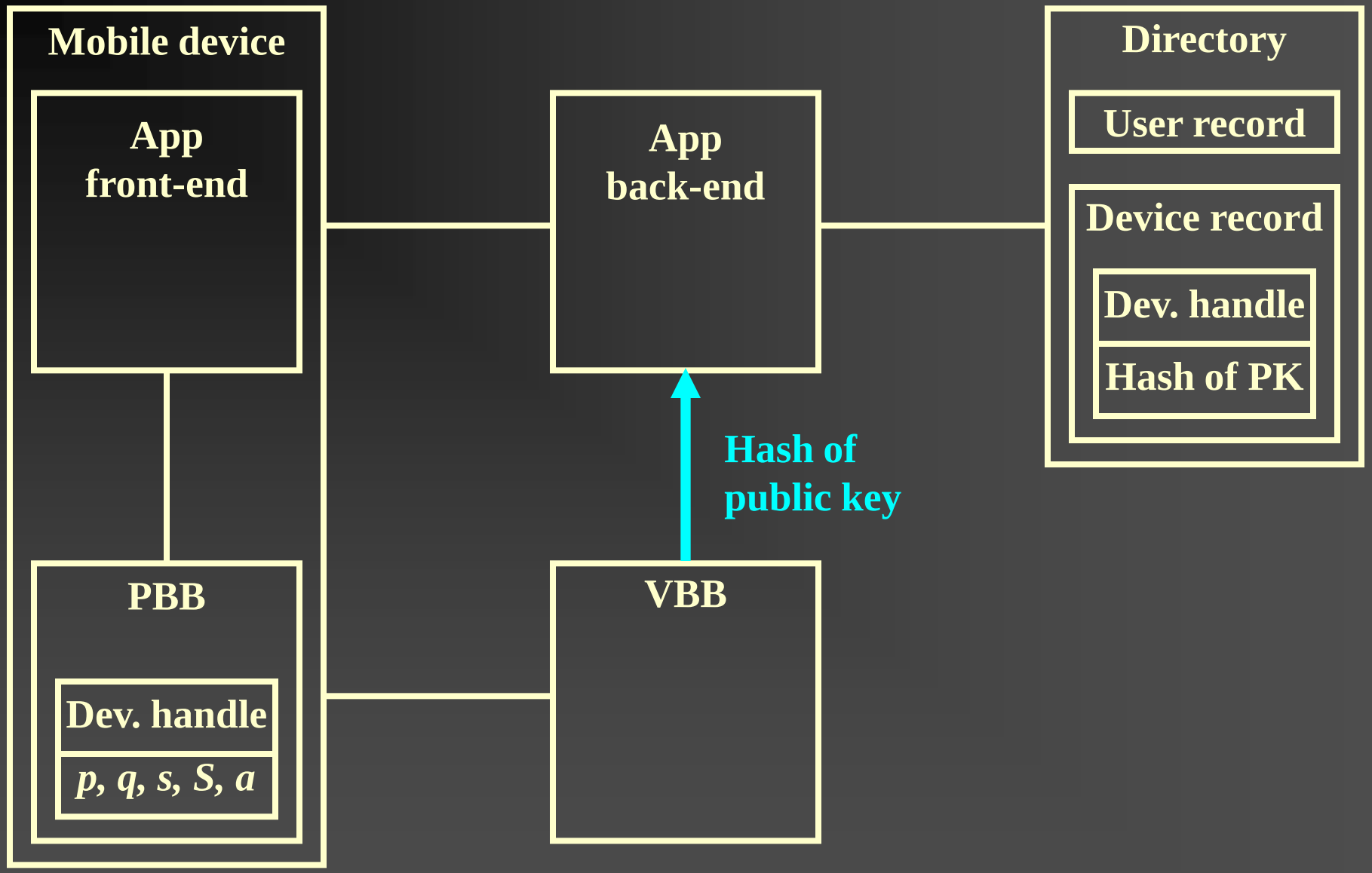

Mobile device
App front-end
PBB
p, q, s, S, a
Public key + proof of knowledge of private key
App back-end
VBB
Auth token
Hash of PK
Directory
User record
Device record
Dev. handle
Hash of PK

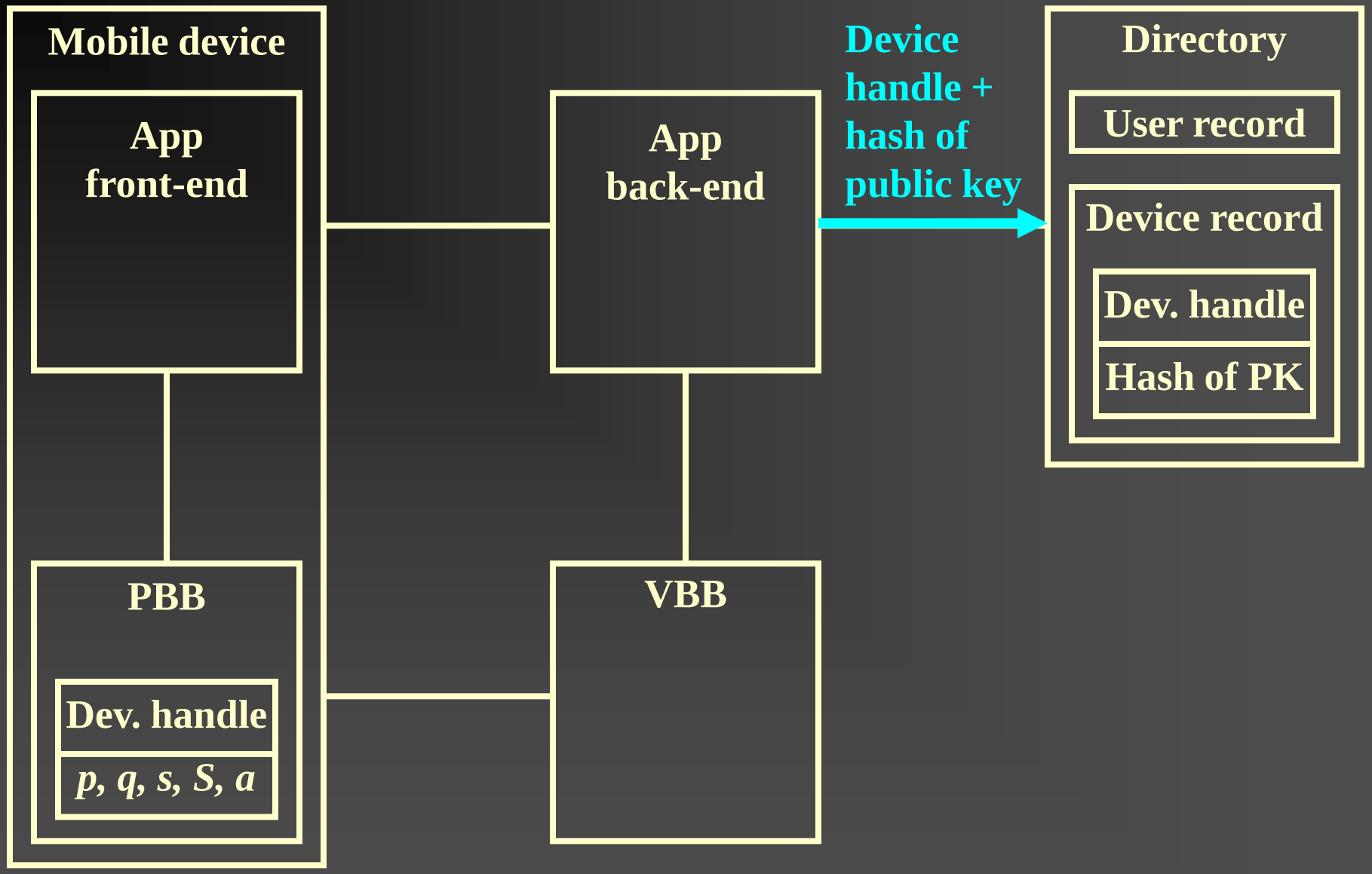

Mobile device
App front-end
PBB
p, q, s, S, a
Auth token
App back-end
VBB
Auth token
Hash of PK
Directory
User record
Device record
Dev. handle
Hash of PK

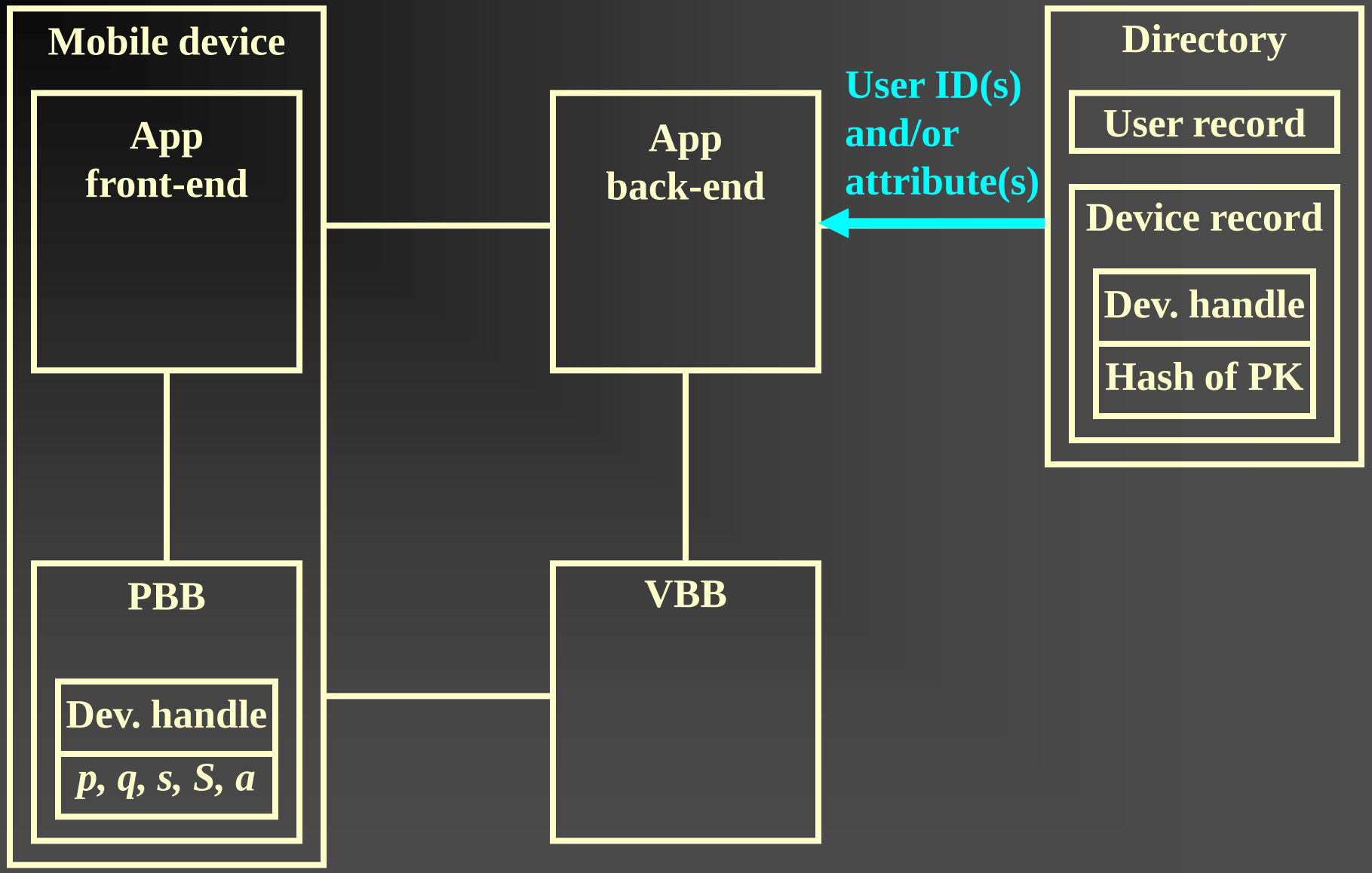

Mobile device
App front-end
Auth token + device handle
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
Auth token
Hash of PK
Directory
User record
Device record
Dev. handle
Hash of PK


Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
Auth token + device handle
App back-end
VBB
Auth token
Hash of PK
Directory
User record
Device record
Dev. handle
Hash of PK


Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
App back-end
Auth token
VBB
Auth token
Hash of PK
Directory
User record
Device record
Dev. handle
Hash of PK

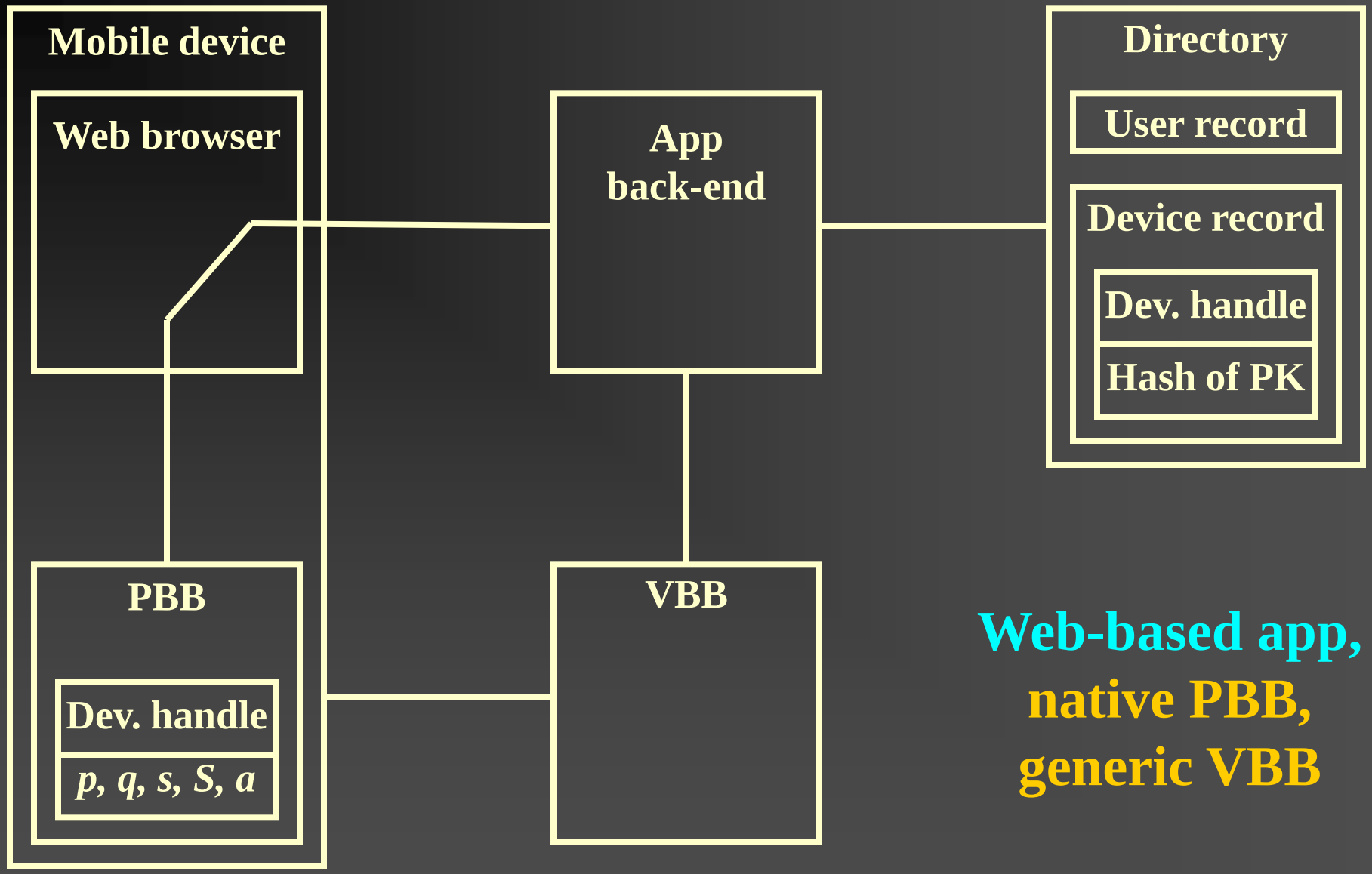

Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
Hash of public key
Directory
User record
Device record
Dev. handle
Hash of PK


Mobile device
App front-end
App back-end
Device handle + hash of public key
Directory
User record
Device record
Dev. handle
Hash of PK
PBB
Dev. handle
p, q, s, S, a
VBB

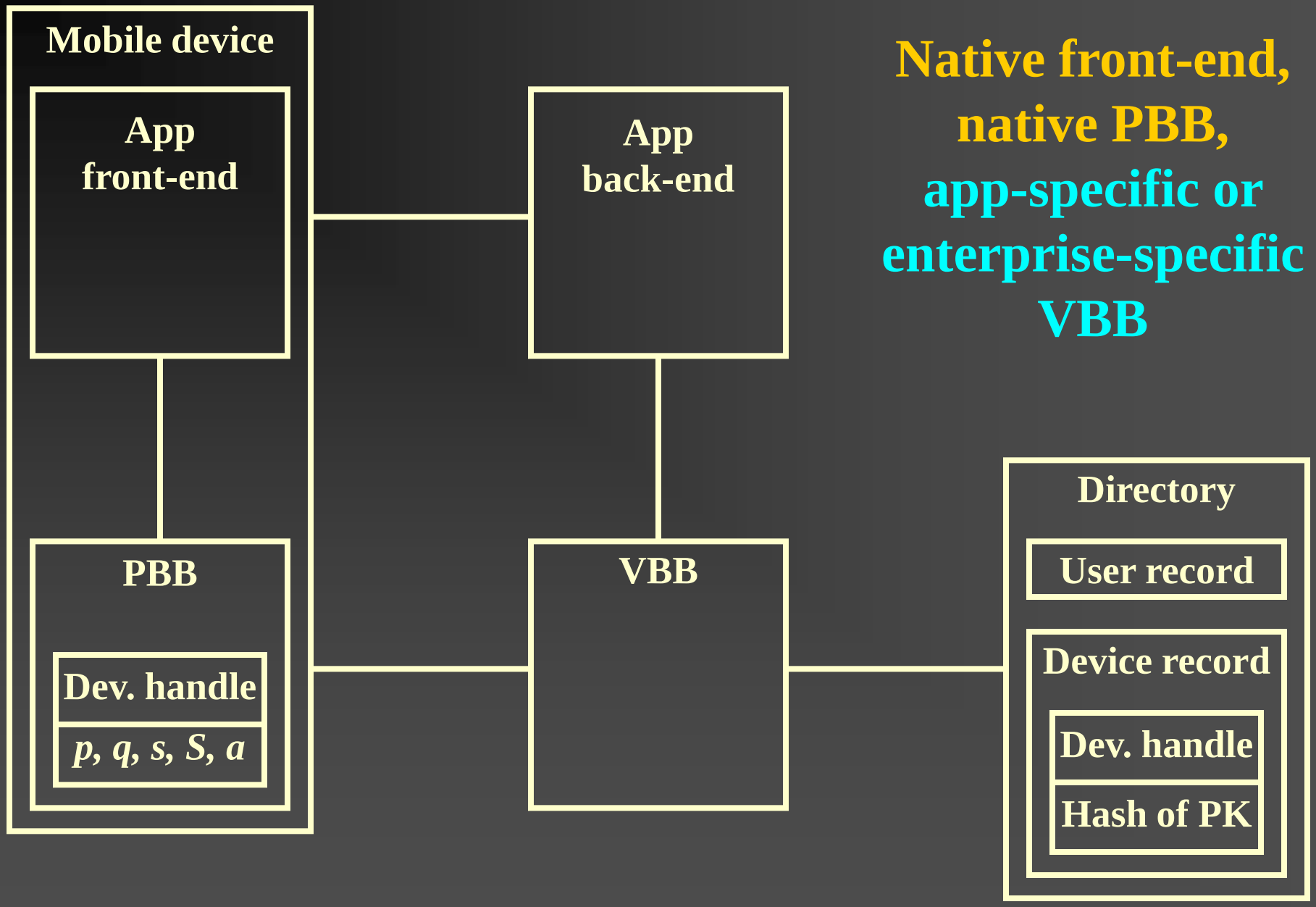

Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
User ID(s) and/or attribute(s)
Directory
User record
Device record
Dev. handle
Hash of PK

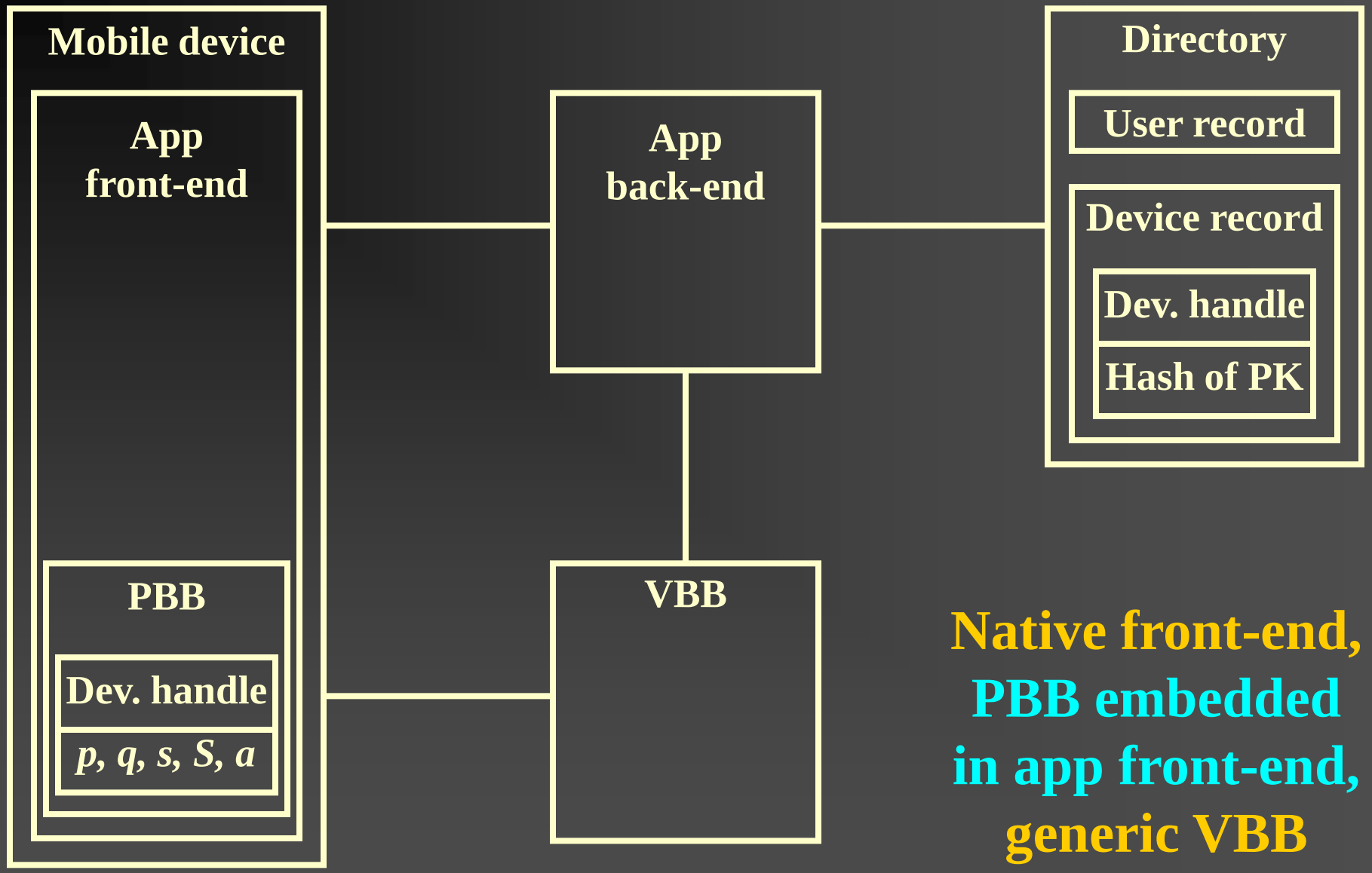
Many Possible Configurations

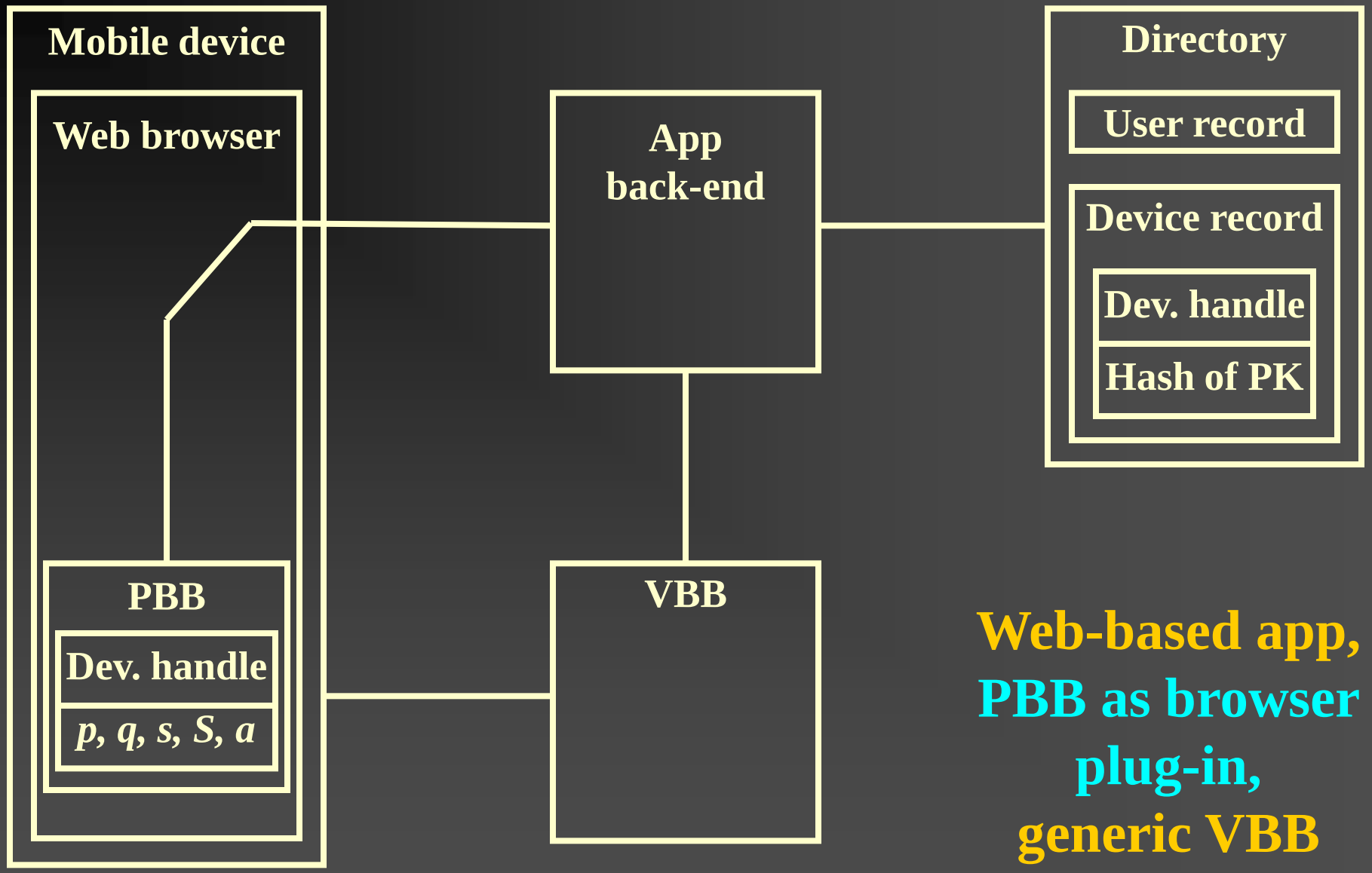
- App
  - May have native front-end (as shown), or
  - May be accessed through a web browser
- PBB
  - One credential for multiple apps
  - Different credentials for different apps
  - May be embedded in application front-end
  - Browser plug-in ➔ works on desktops and laptops
- VBB
  - May be a generic server appliance
  - May be app- or enterprise-specific, and access the directory
- Multiple security domains

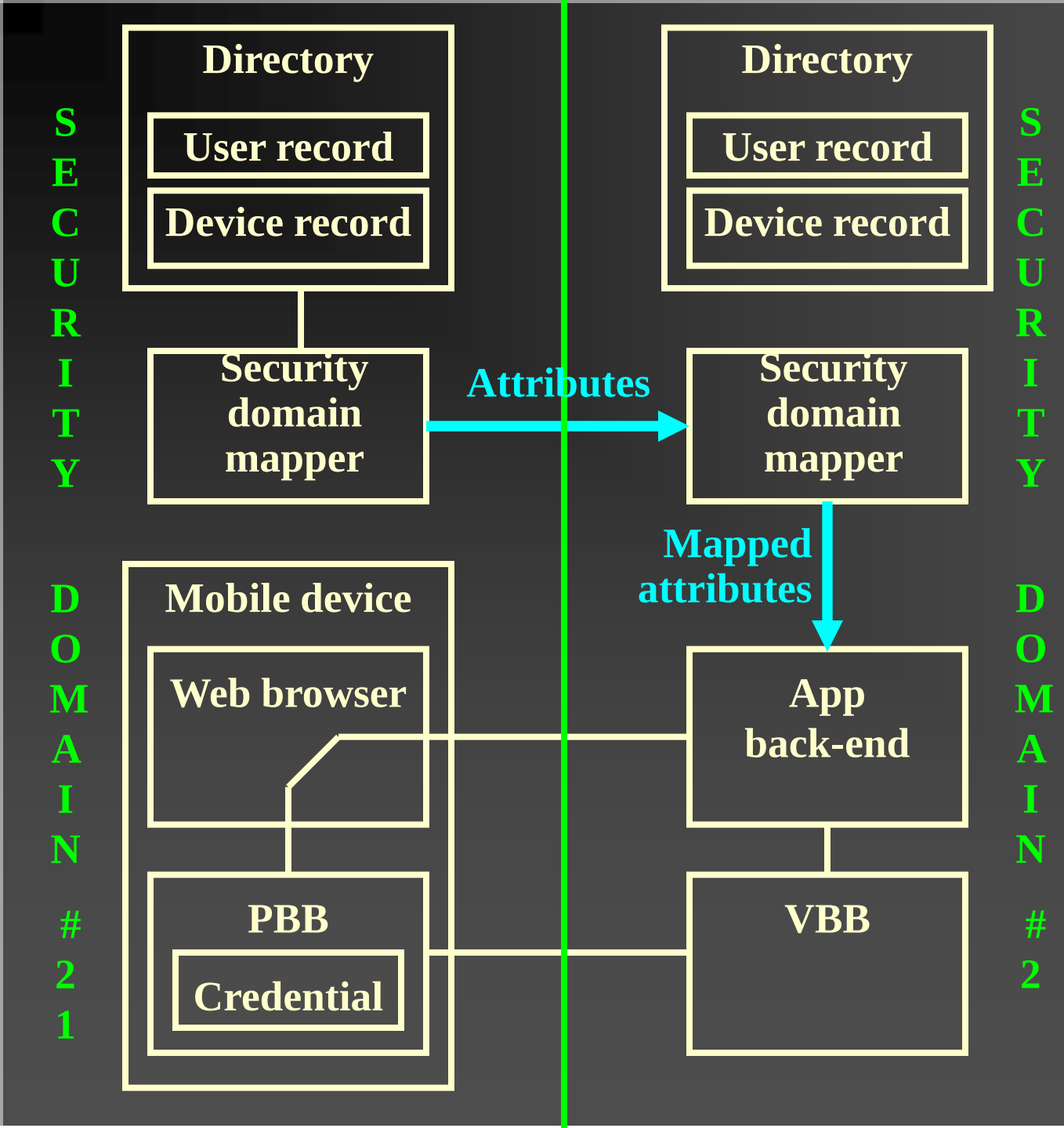


Web-based app, native PBB, generic VBB


Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
Directory
User record
Device record
Dev. handle
Hash of PK
Native front-end, native PBB, app-specific or enterprise-specific VBB


Mobile device
App front-end
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
Directory
User record
Device record
Dev. handle
Hash of PK
Native front-end,
PBB embedded
in app front-end,
generic VBB


Mobile device
Web browser
PBB
Dev. handle
p, q, s, S, a
App back-end
VBB
Directory
User record
Device record
Dev. handle
Hash of PK
Web-based app,
PBB as browser plug-in,
generic VBB



Multiple security domains

Device handler augmented by Security Domain ID

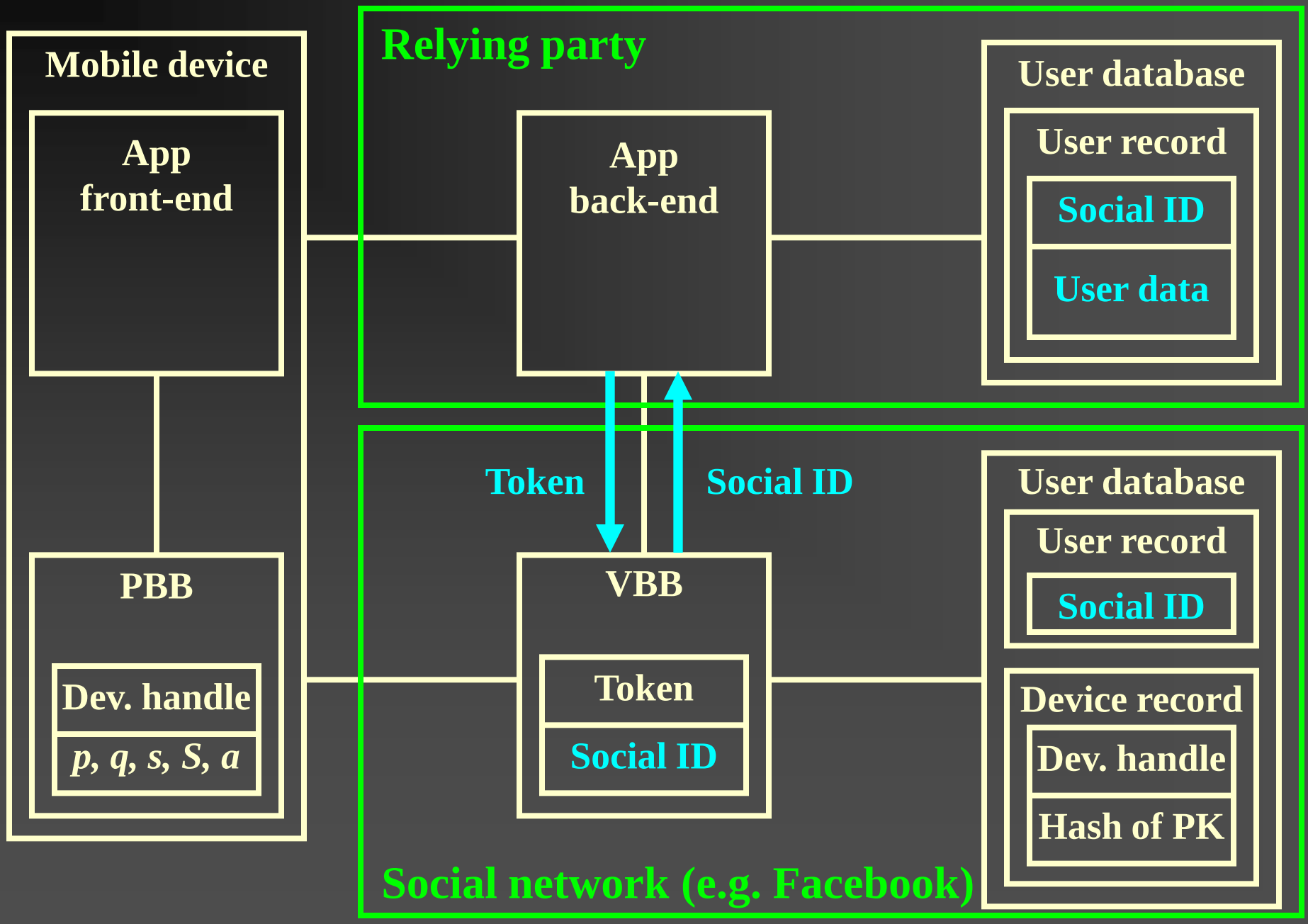
Beyond Derived Credentials

- Password elimination on the Web at large without sacrificing privacy
- Social login without passwords
- Effective data protection for locked phones

Eliminating Passwords without Sacrificing Privacy

- Authentication by userid-password provides anonymity, unlinkability and unobservability
- Alternatives being proposed (OpenID, SAML, etc.) redirect to a third party
  - Third party identifies user ➔ *no anonymity*
  - Authentication to different relying parties *can be linked* via the third party identifier
  - Third party *observes* the transaction
- Our techniques eliminate passwords and preserves privacy (anonymity, unlinkability and unobservability) because they do not involve a third party

Social login without passwords



Data Protection Problem

- Data protection in iPhone locked by a PIN
  - Data encrypted by key hierarchy including a key derived from PIN and a “hardware key” that cannot be extracted from the silicon by a casual user
  - PIN protected against offline attack by hardware key
- But:
  - Vulnerabilities ➔ hardware key used for offline attack using the phone’s own processor; exhaustive attack on 4-digit PIN takes 40 min
  - Hardware key could be extracted by probing the silicon

Data Protection Solution

- Encrypt data under symmetric key
- Store symmetric key in online server, or split it over several servers using Shamir's *k*-of-*n* secret sharing technique
- Retrieve key over secure connection(s), *authenticating with a key pair regenerated from a PIN and/or a biometric*, so that tampering with phone does not help attacker

Risks of Mobile Applications?

- Mobile computing architecture potentially more secure
  - Apps are sandboxed
- But vulnerabilities allow rooting
  - Routinely used for jailbreaking and by forensic tools
  - GMU, NIST, NSA working on hardened Android kernel
    - Hardening should include interapp communications
- Our data protection technique…
  - Protects data against exploitation of vulnerabilities after seizing device
  - But malware running while legitimate user is using the device could capture PIN or biometric data

For more information…

- Whitepapers
  - http://pomcor.com/whitepapers/DerivedCredentials.pdf
  - http://pomcor.com/whitepapers/MobileAuthentication.pdf
- fcorella@pomcor.com
- kplewison@pomcor.com