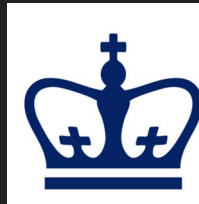


TZ - Password Manager

THARUN KUMAR JAYAPRAKASH(TJ2557)
HUGO MATEOUSK(HM2953)

Estimated Total Hours: 12



Plans (Past Week)

- Setup OP-TEE Environment on Raspberry Pi
- Architecture - detailed slides, threat model
- Secure API definition
- Basic “frontend” (unsecured application)
- Basic “backend” (secured application)



What Actually Happened

Work completed:

- Encryption/decryption and key management architecture
- Modularized host application, advanced TEE internal API functions used for TA

Major challenges+roadblocks:

- Faced Issues running OP-TEE on Raspberry Pi 4b
- Header file management and file structure when compiling as a part of the TEE image (and QEMU)

Attribution:

- Official OP-TEE support resources exist only for Raspberry Pi 3

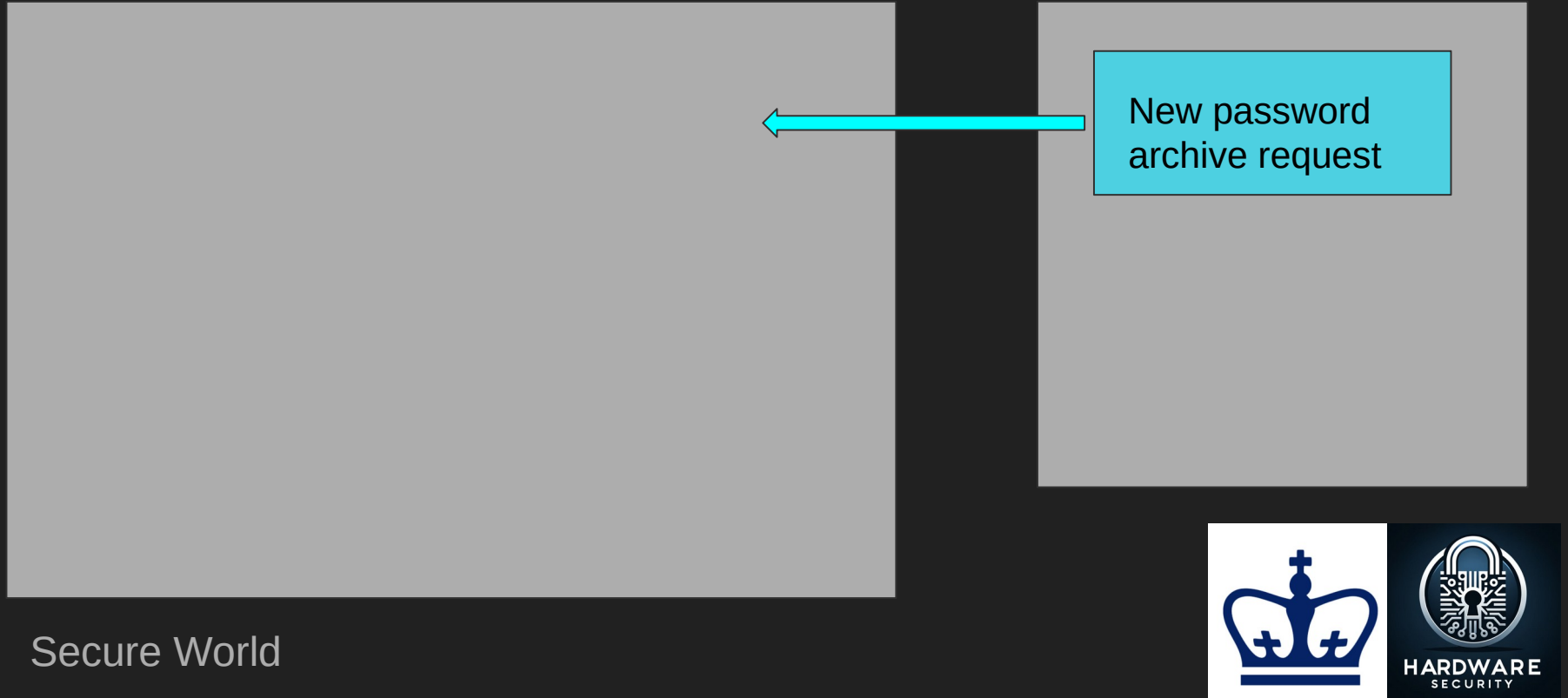


Plans (Next Week)

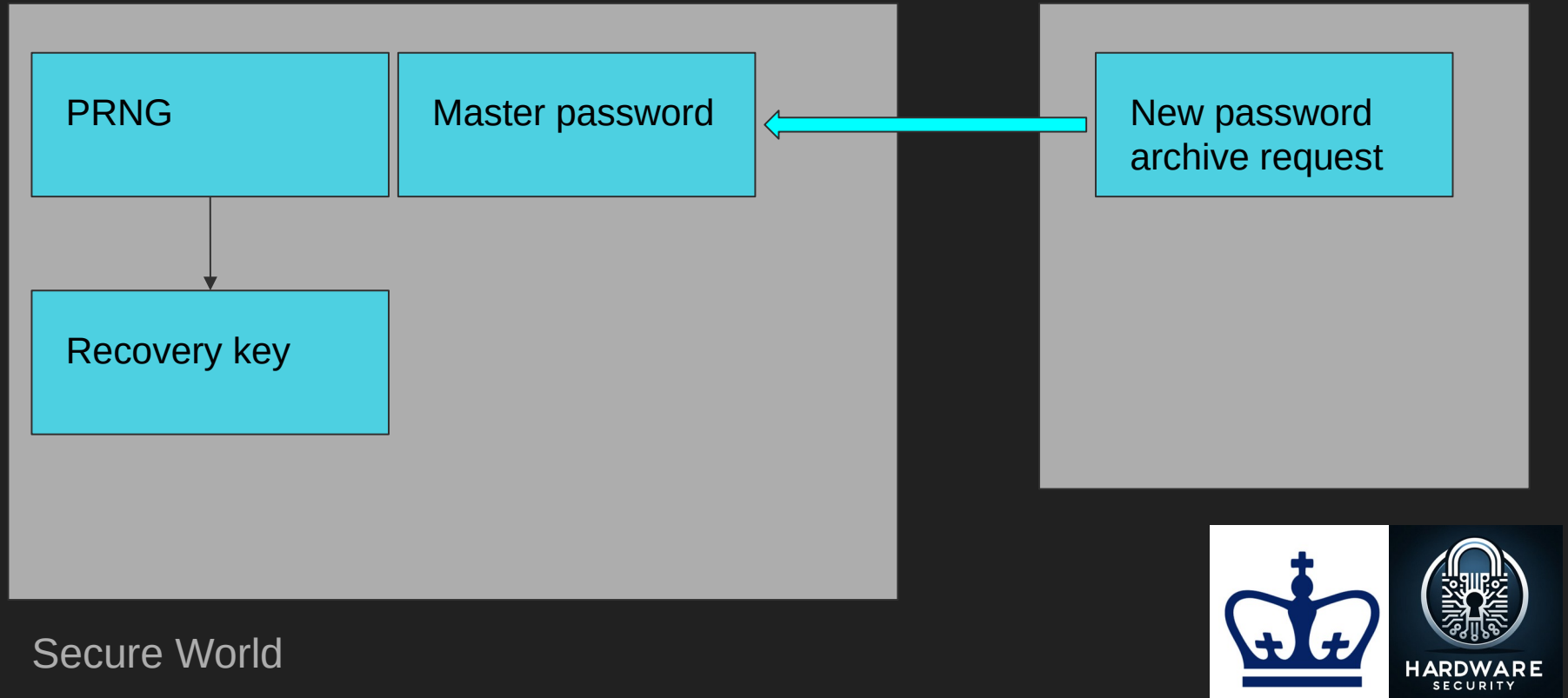
- Continuing with attempts to make it work on RPi 4
- Key derivation and secure storage tests for TA
- Host application basic “UI” and application logic



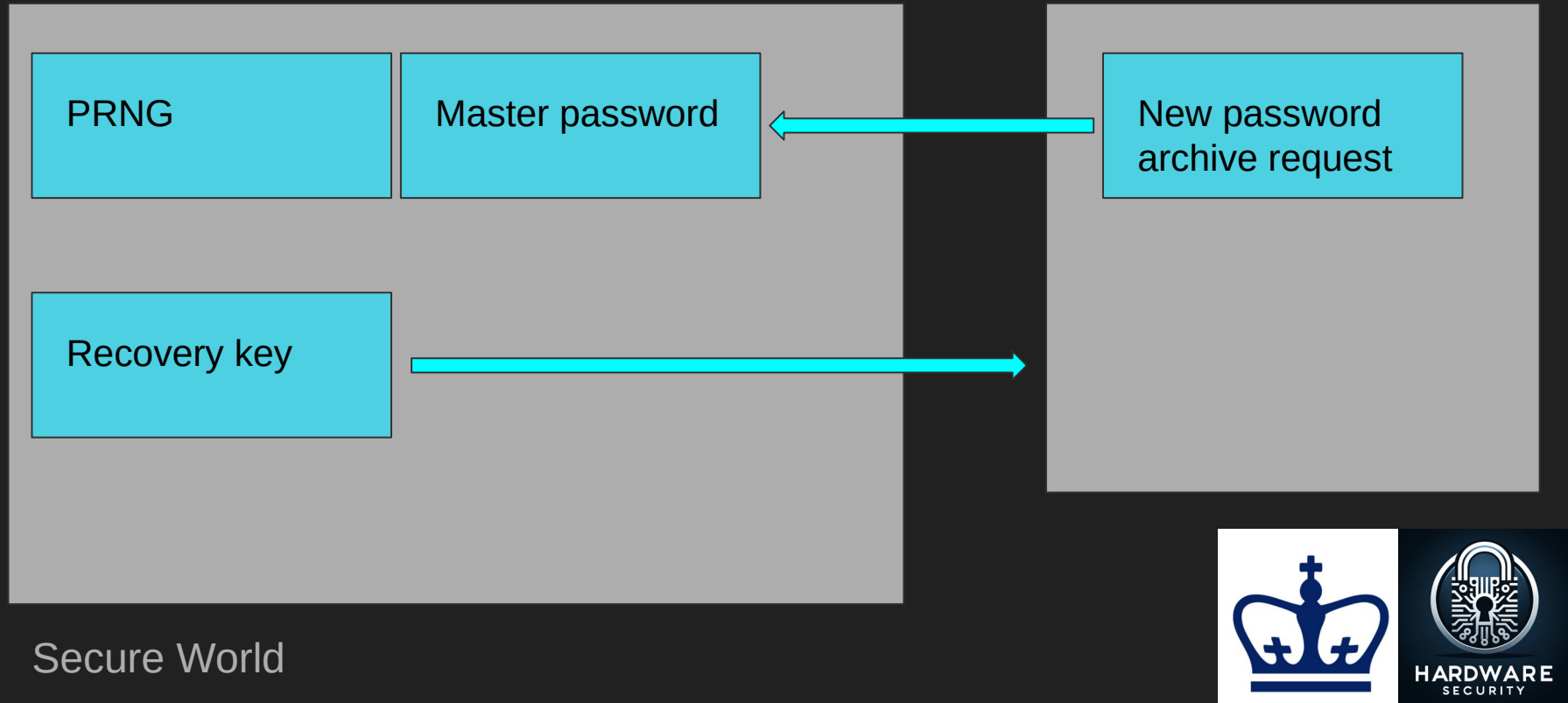
Architecture



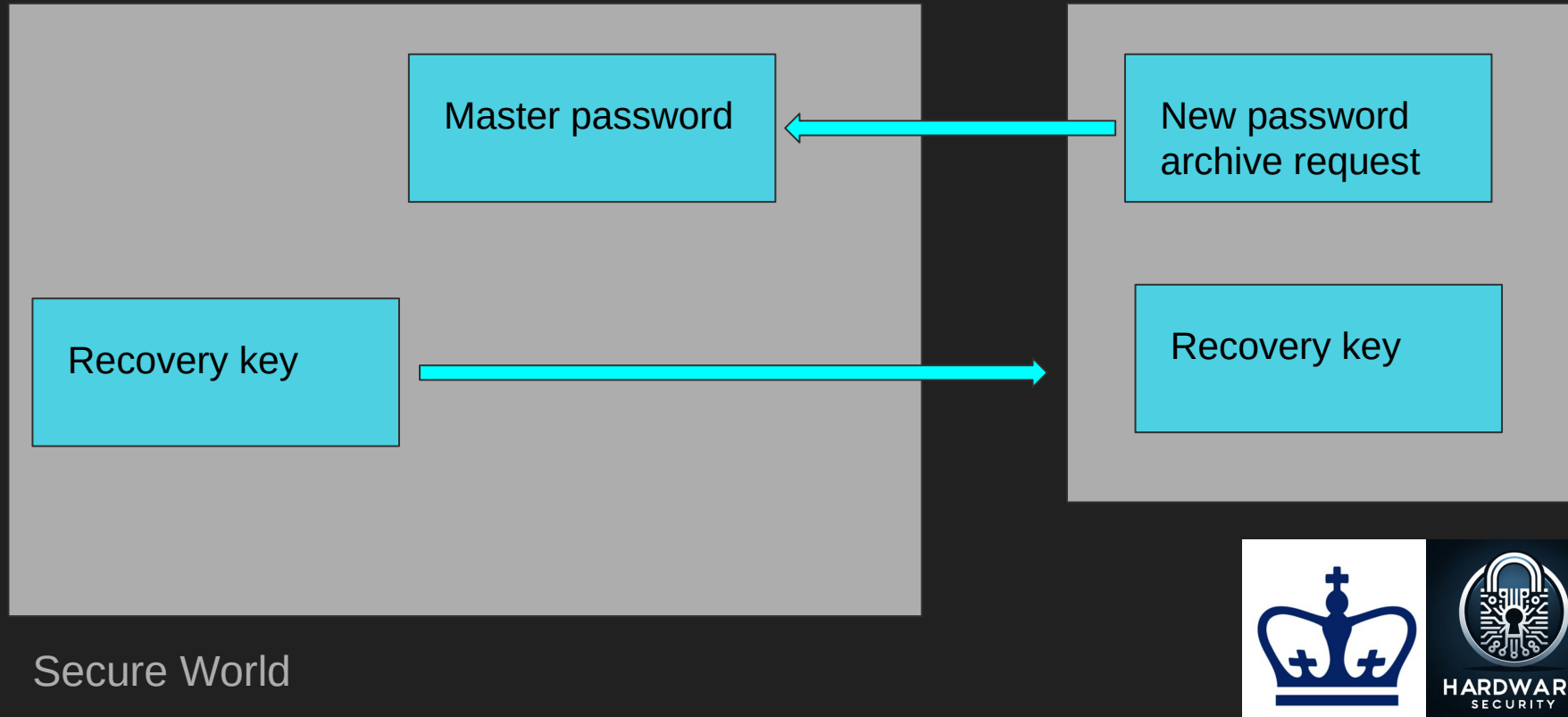
Architecture



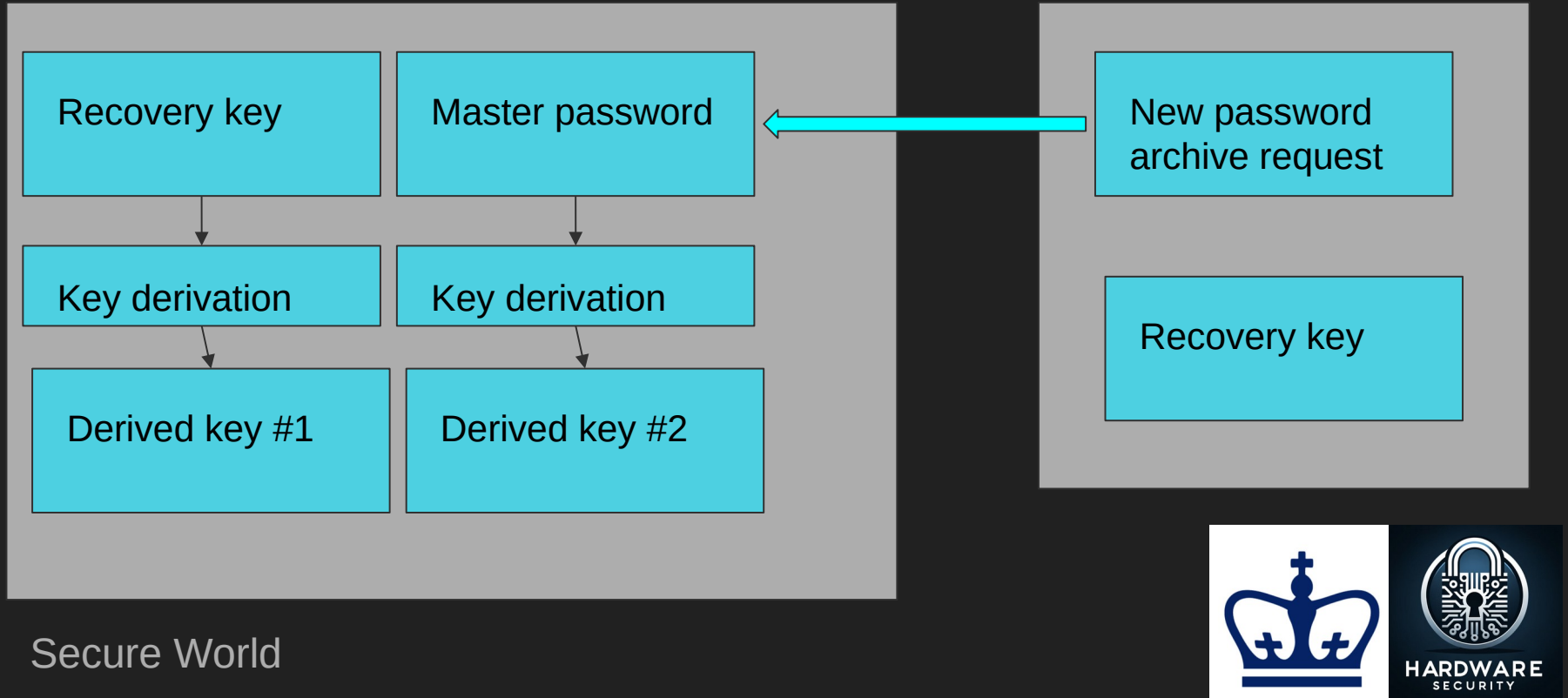
Architecture



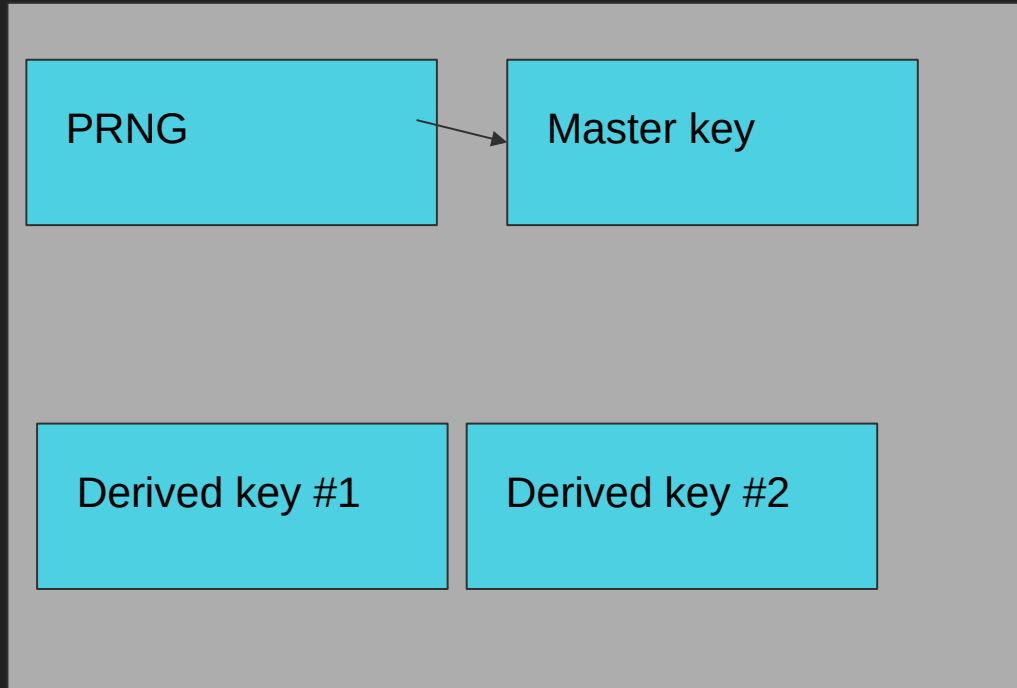
Architecture



Architecture

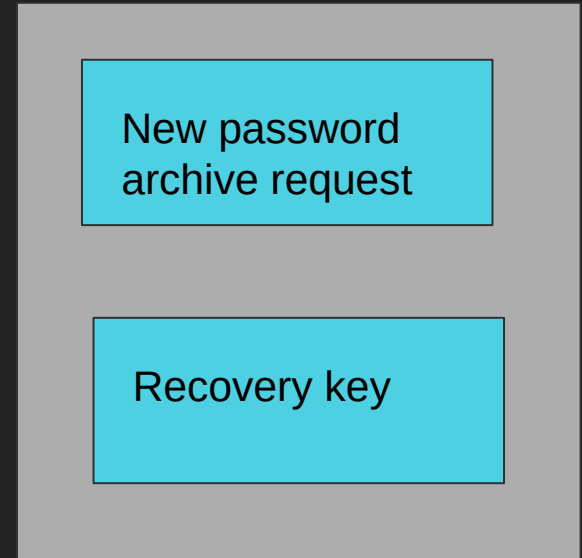


Architecture

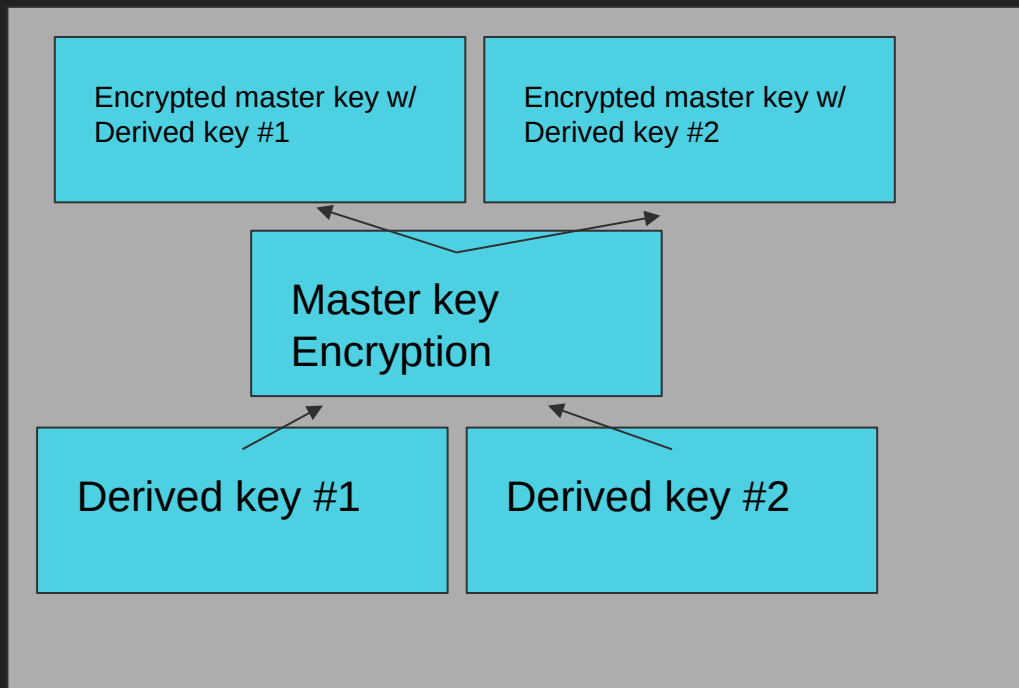


Secure World

Normal World

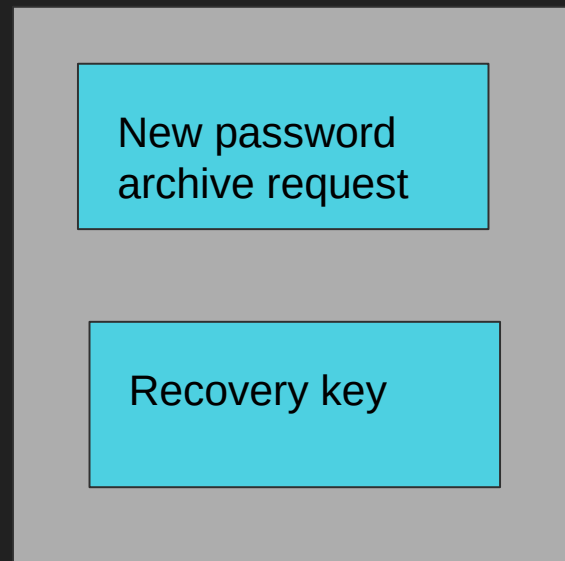


Architecture

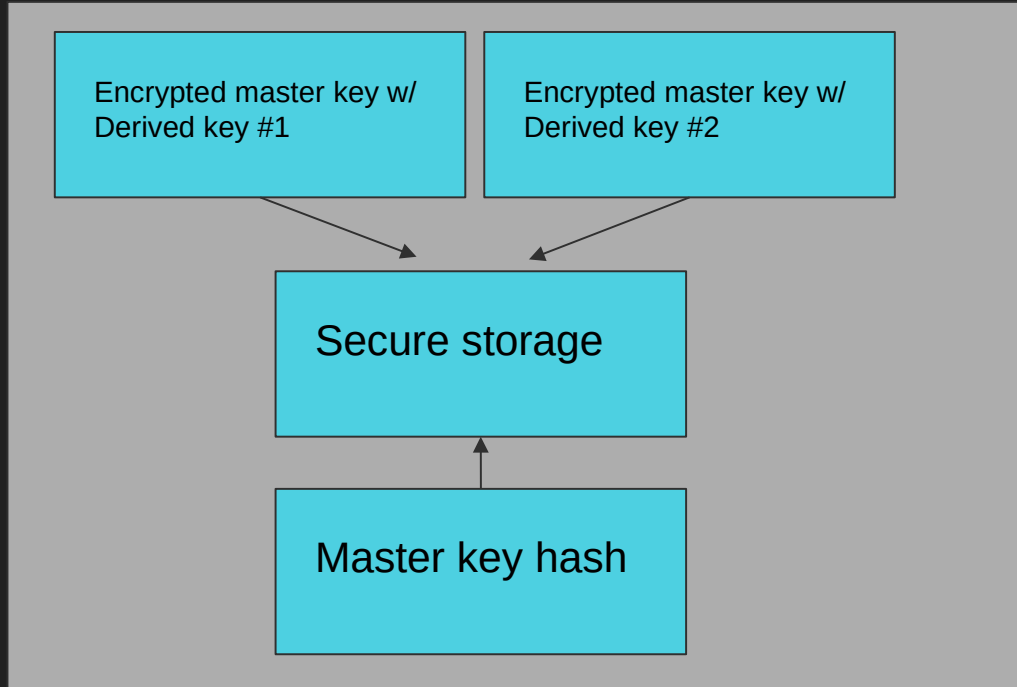


Secure World

Normal World

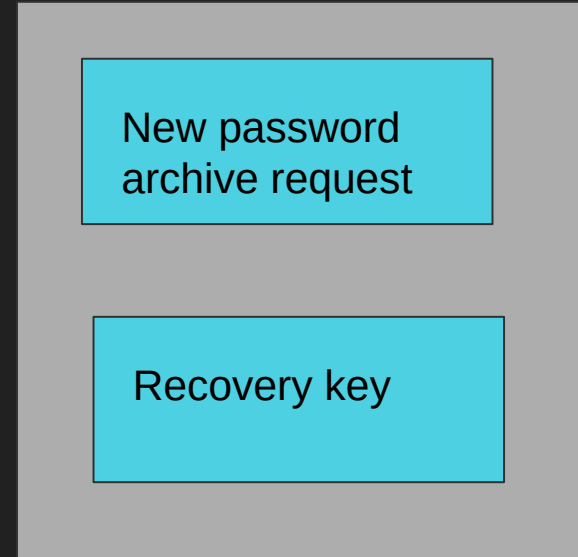


Architecture

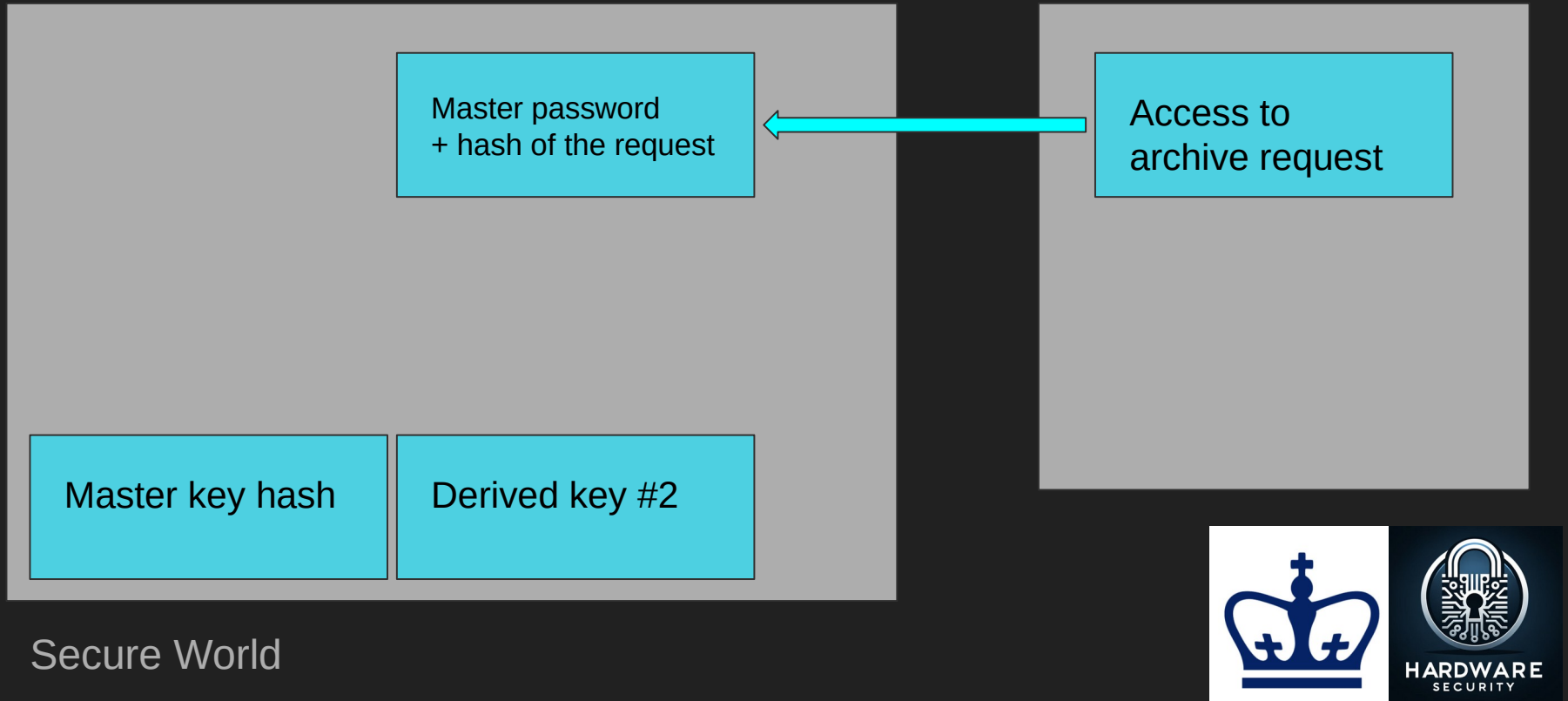


Secure World

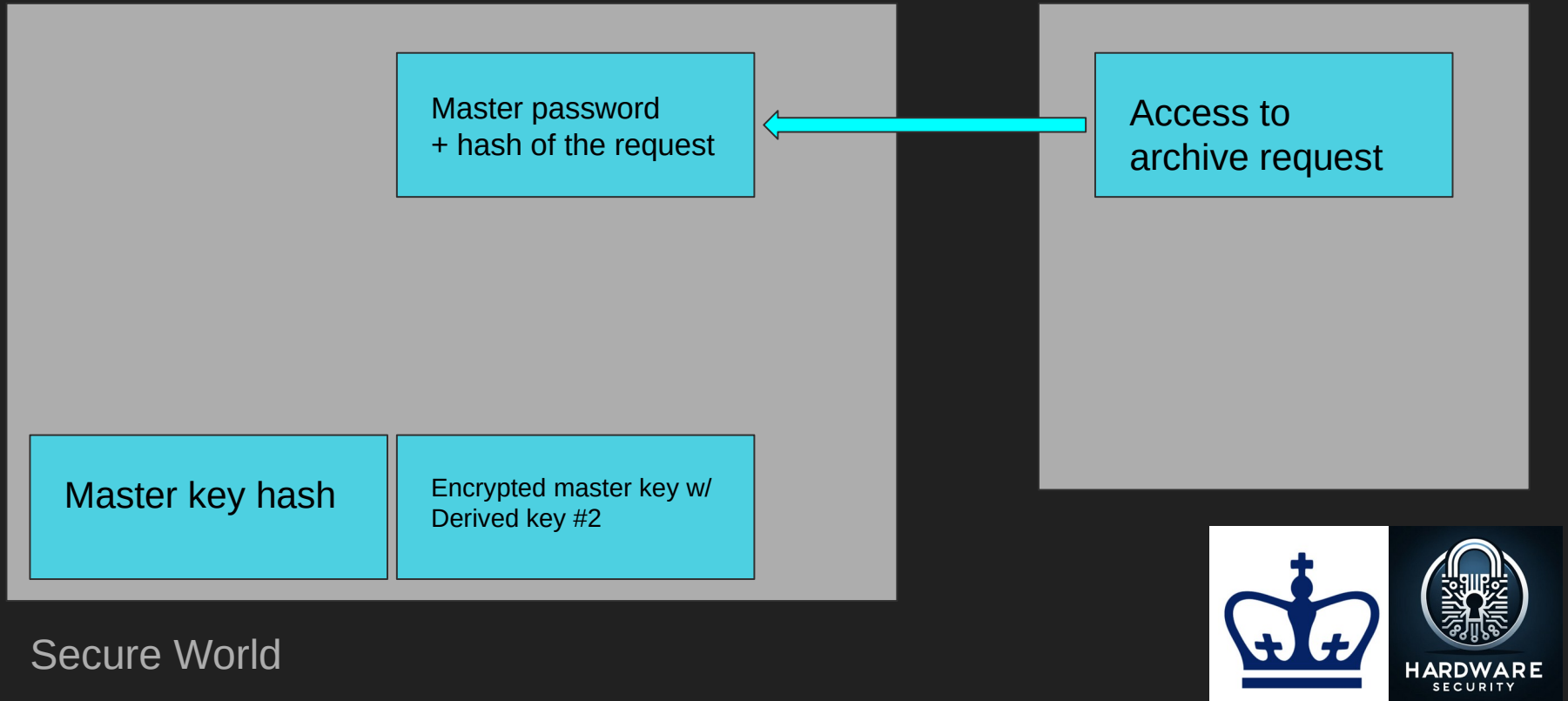
Normal World



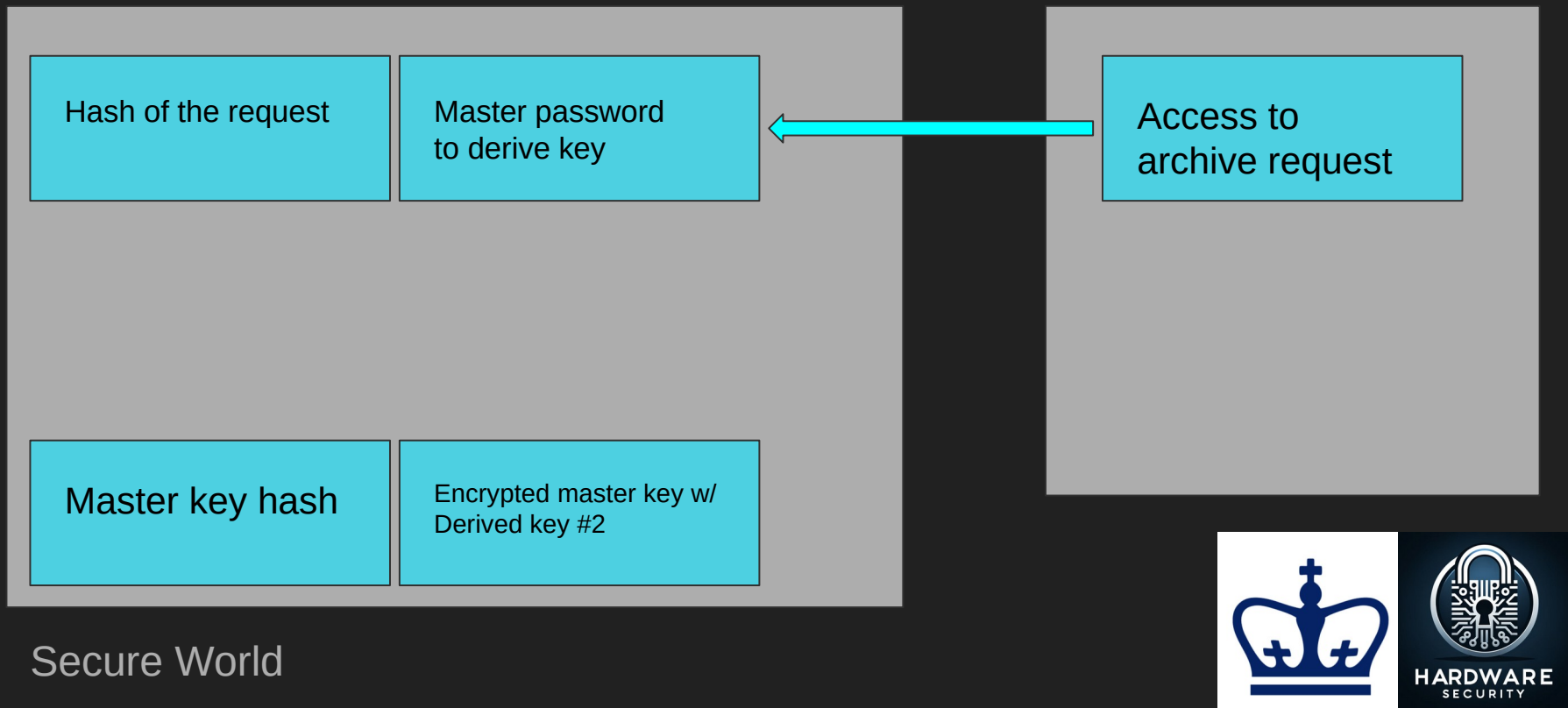
Architecture



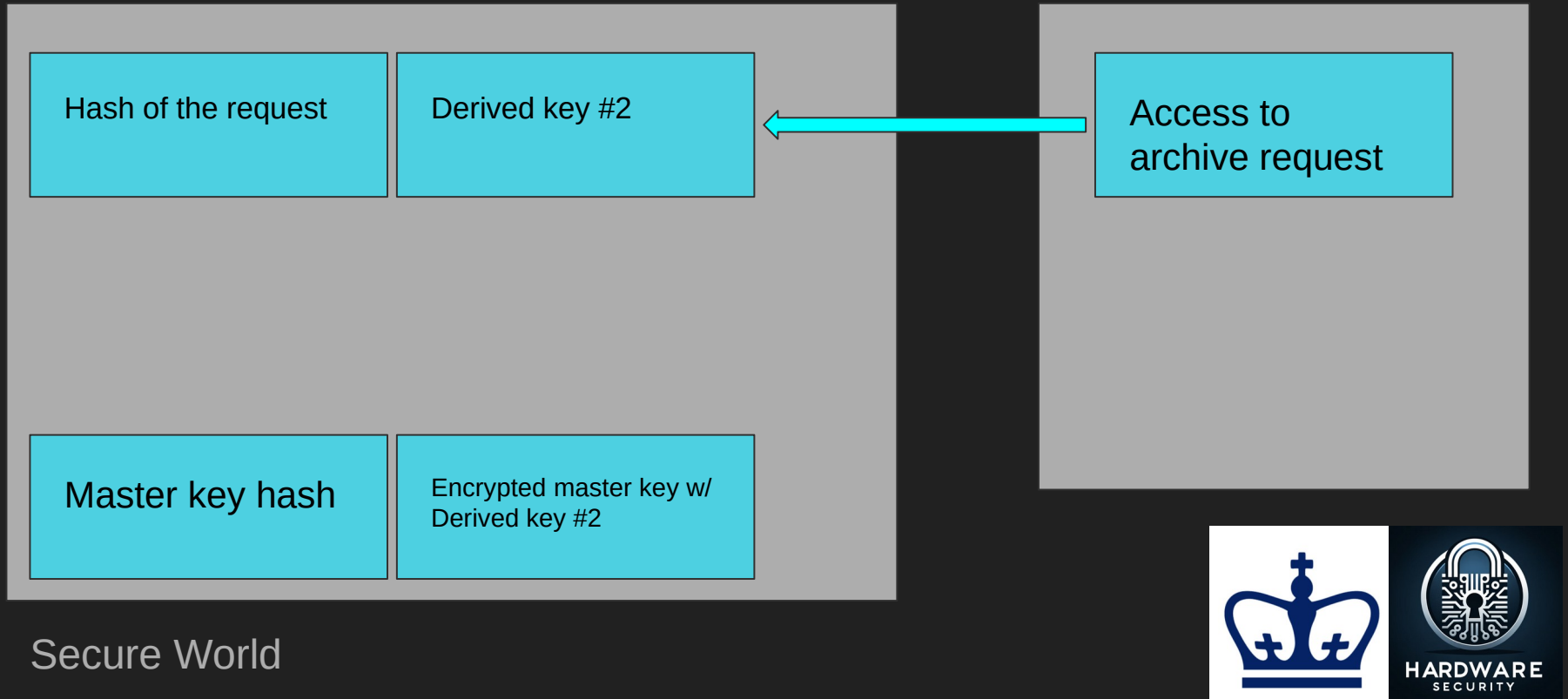
Architecture



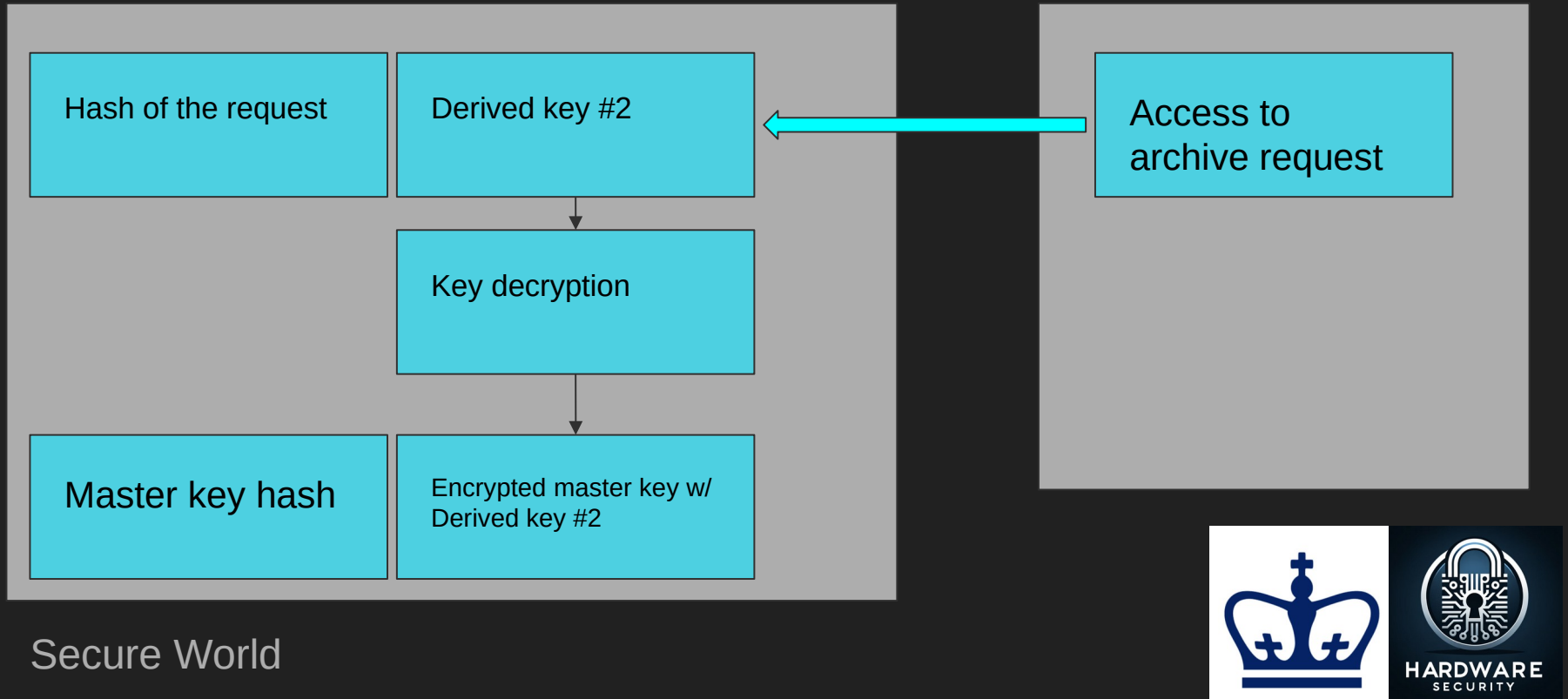
Architecture



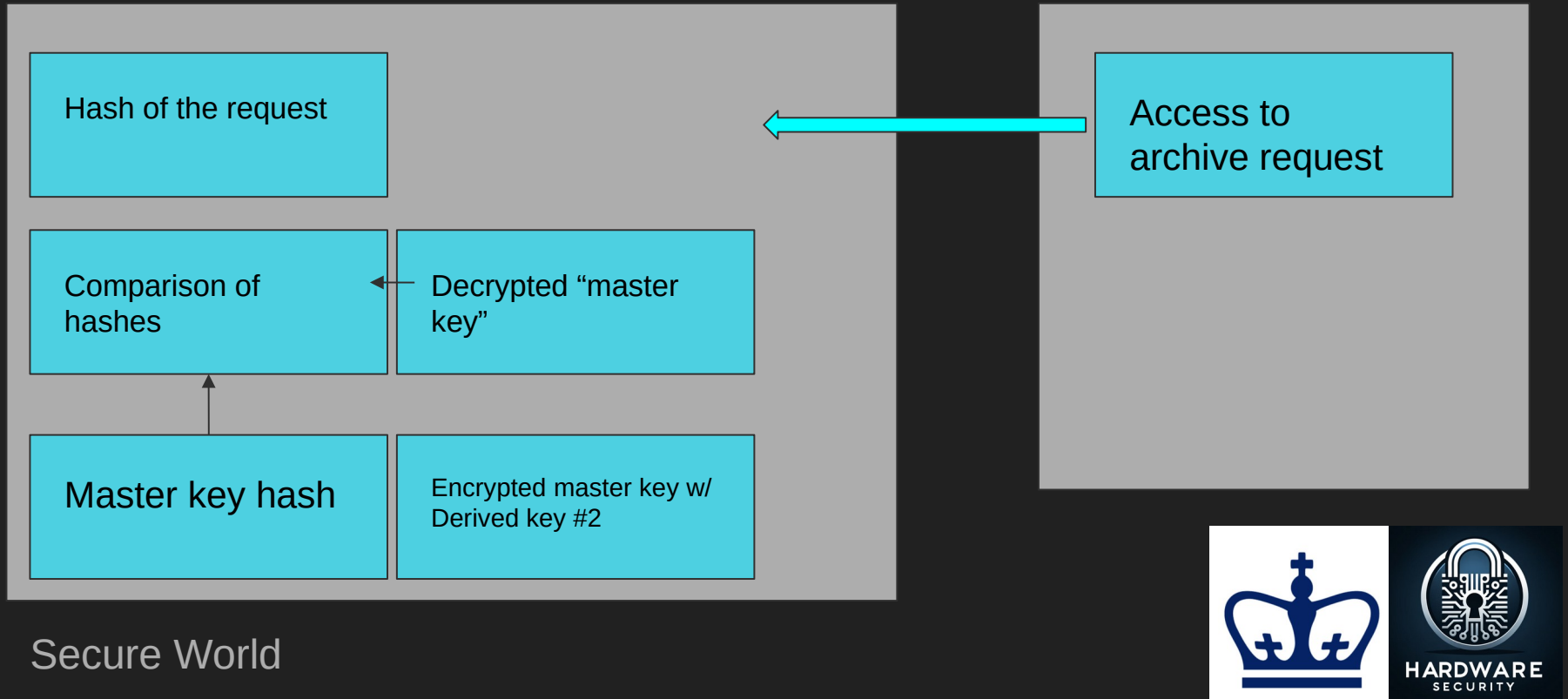
Architecture



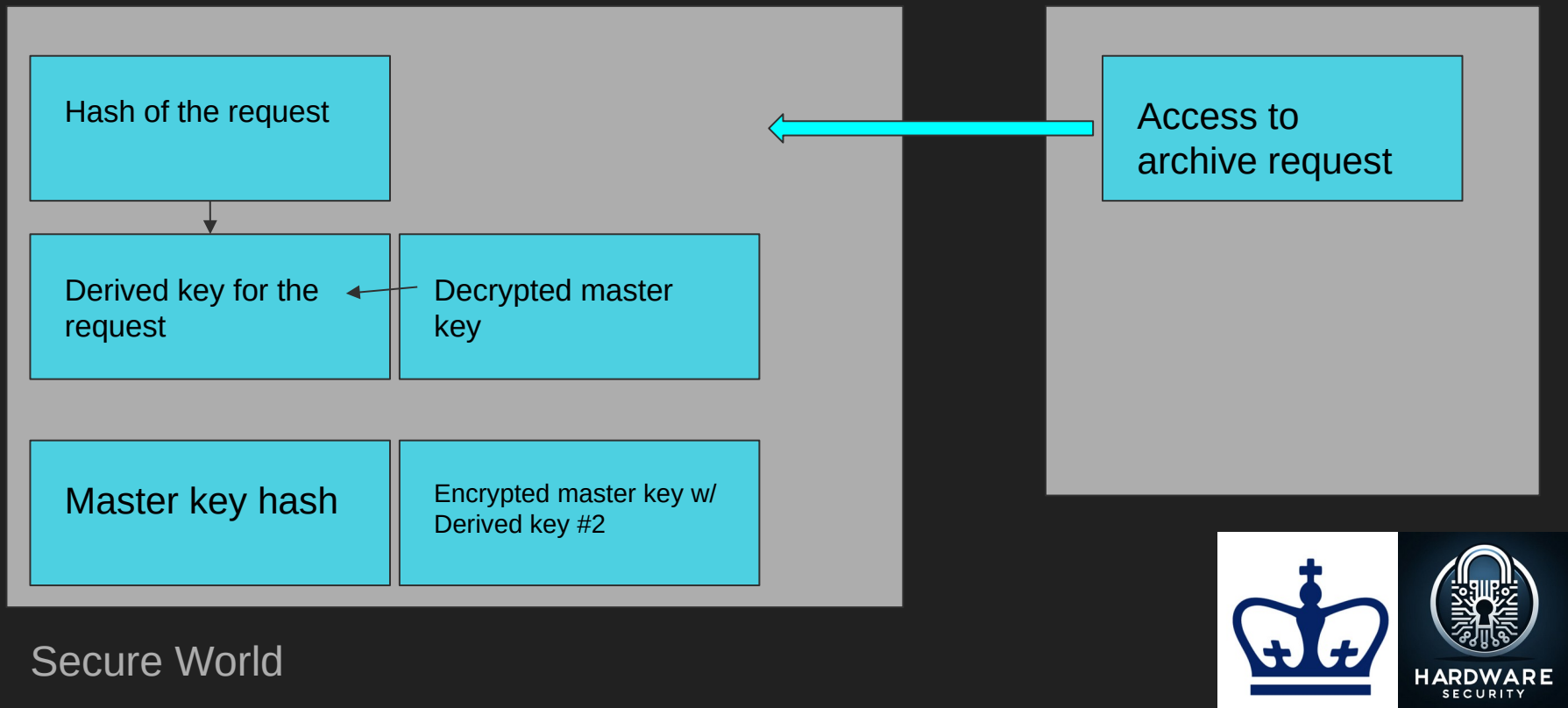
Architecture



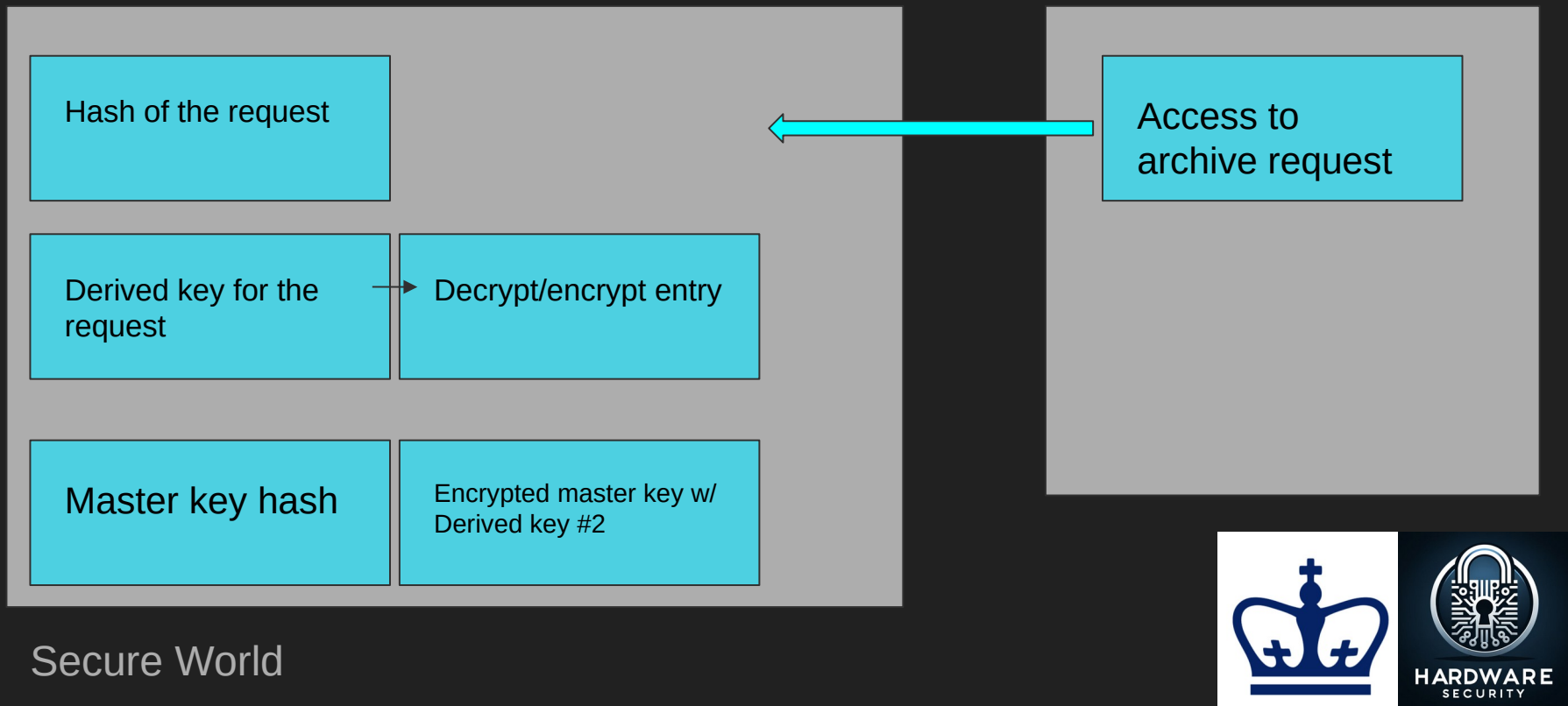
Architecture



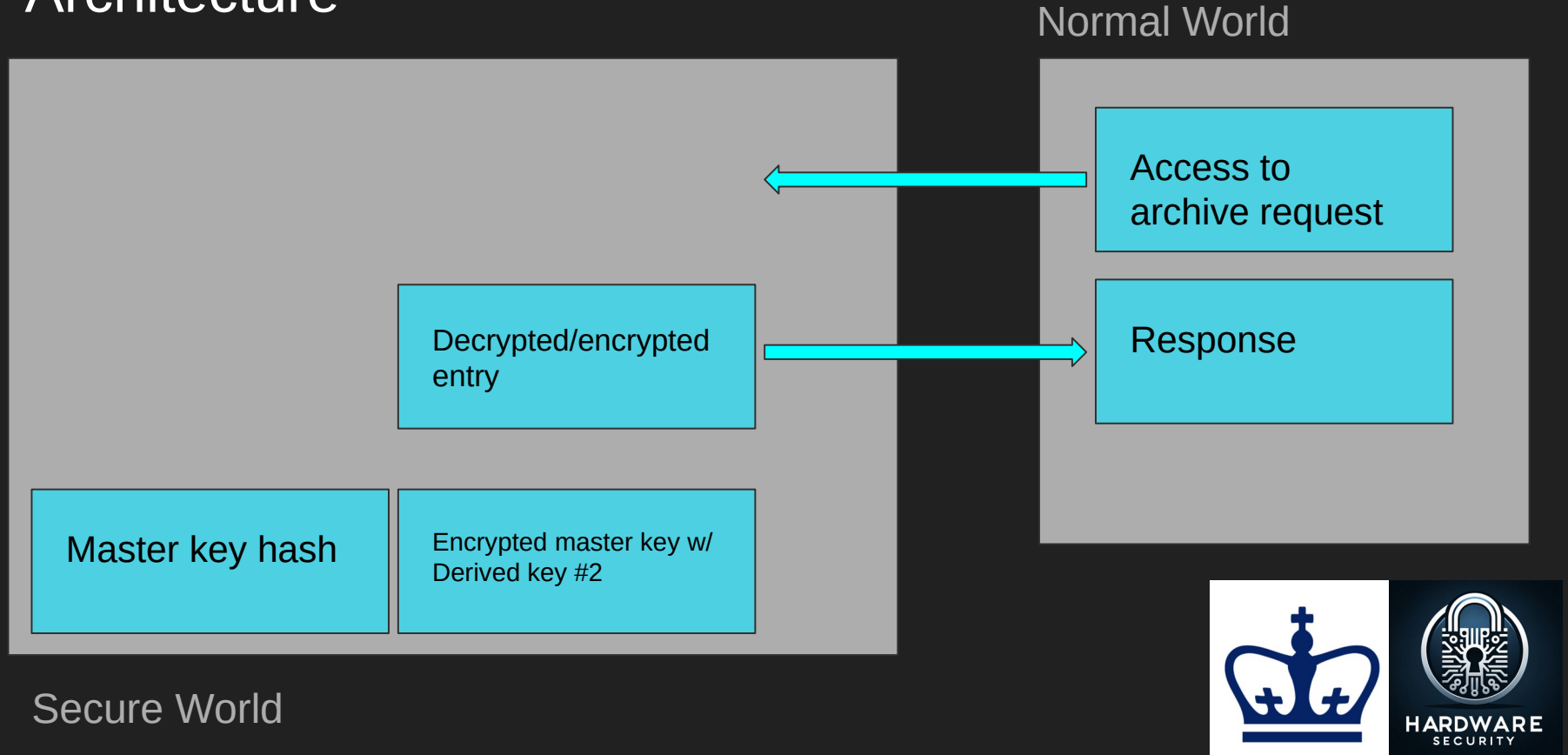
Architecture



Architecture



Architecture



Summary/Overall Progress

- Encryption/decryption and key management architecture prepared
- Modularized host application
- Advanced internal TEE API functions tested

