OCP L.O.C.K.

Layered Open-source Cryptographic Key-management

Amber Huffman, Principal Engineer, Google

Lee Prewitt, Director Cloud Hardware Storage, Microsoft

FMS the Future of Memory and Storage

Who we are

Google

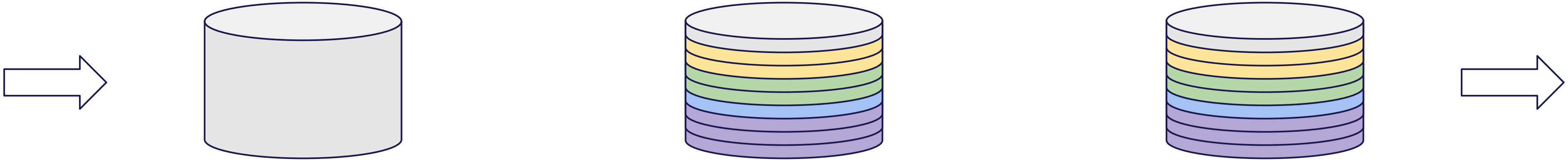
Microsoft

SAMSUNG

KIOXIA

SOLIDIGM™

Life of a data center storage device



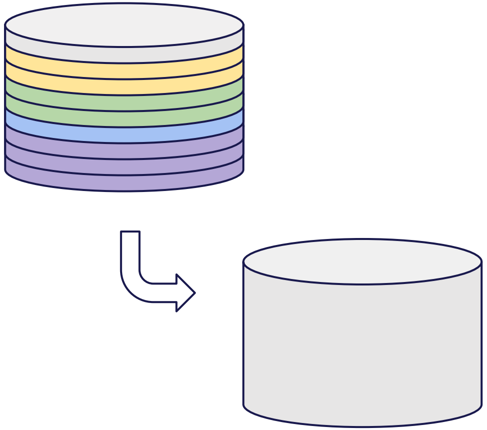
Drive arrives from the supplier

Drive is given user data to store

Drive is decommissioned

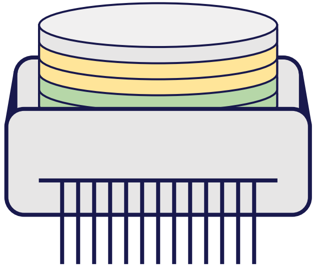
Decommissioning drives

- The physical drive is leaving the data center
- **User data cannot be permitted to escape**



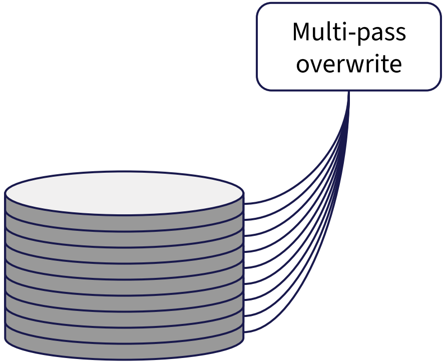
Default policy: destroy the drive

- Safest way to ensure bits on the drive don't escape
- Produces significant e-waste
- Impacts bottomline of drive owner
    - Inhibits second-hand markets



One technique: overwrite

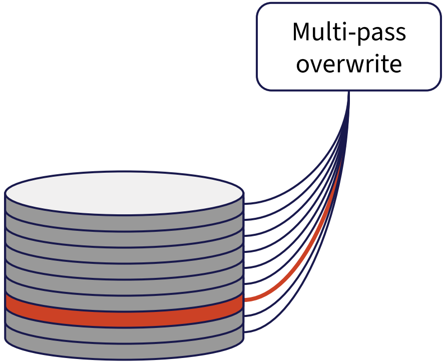
- Write over every piece of data held within the drive
- E*very* portion of the drive must be overwritten, before the drive is allowed to leave in one piece



Multi-pass overwrite

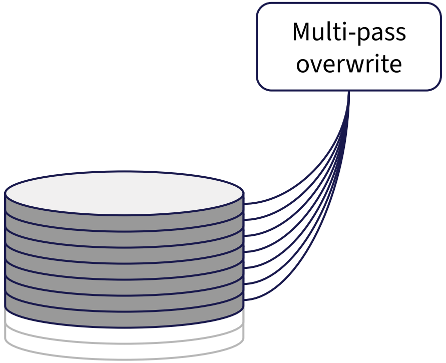
Problem: drive failure

- If *any* portion of the drive cannot be overwritten, erasure fails and the drive must be destroyed
- Ergo, we still destroy a lot more disks than we'd like



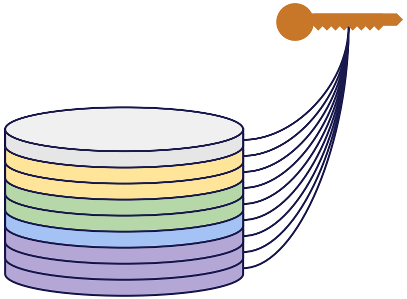
Problem: NVMe page management

- On NVMe drives, bad pages are hidden from the host
- The host cannot even address such pages
- Hidden pages may have user data



Solution: drive encryption

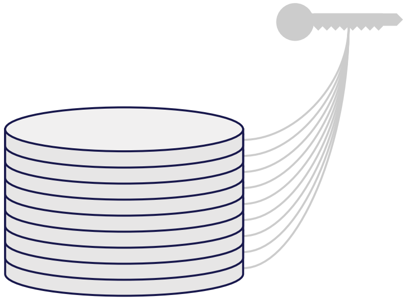
- Ensure all data on the drive is encrypted to a key



Solution: drive encryption

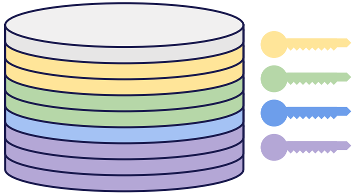
- Ensure all data on the drive is encrypted to a key

- Forget the key



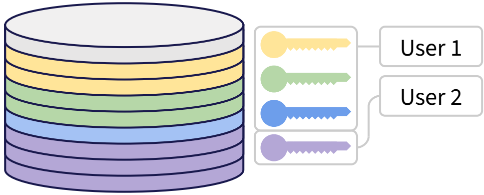
NVMe self-encrypting drives

- The drive manages encryption keys
- Allows granular mapping of keys to address ranges



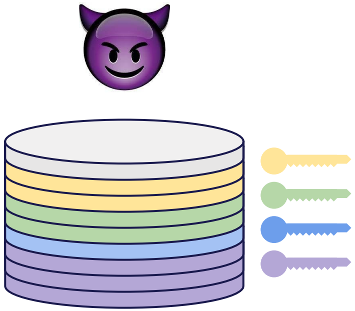
NVMe self-encrypting drives

- The drive manages encryption keys
- Allows granular mapping of keys to address ranges
- Allows granular mapping of keys to users



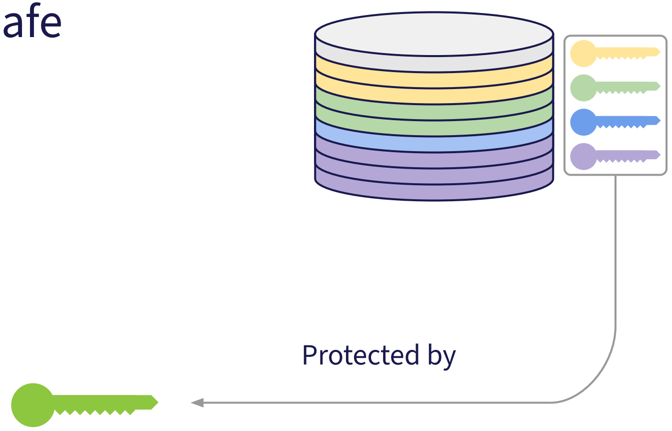
Risk: drive theft

- Keys must be erased before the drive leaves the DC
- If the drive is stolen, the keys survive
- A determined adversary may obtain user data



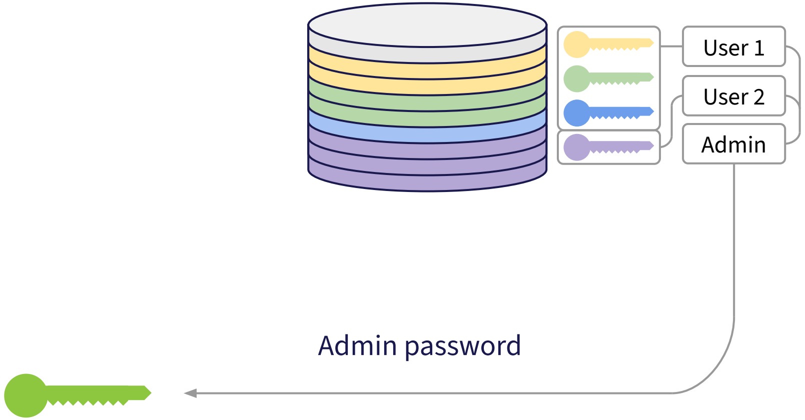
Mitigation: key material held outside the drive

- All media keys protected with a secret the drive does not have
- If the drive is stolen, the 'root secret' remains safe
- By extension, all media keys remain safe

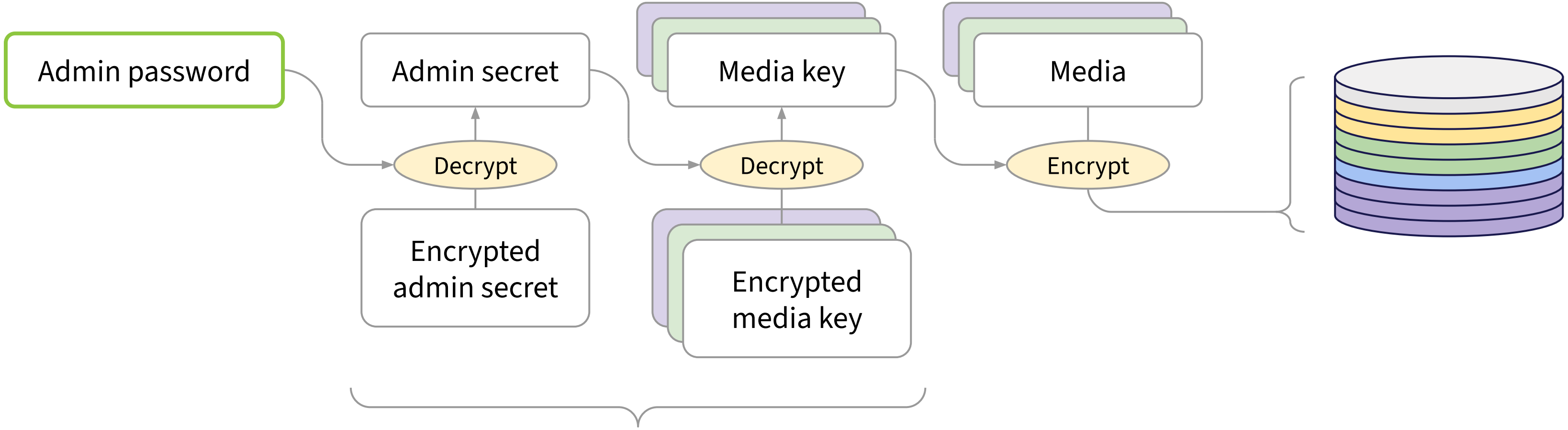


How in practice: admin credentials

- Set up a strong admin password
- Hold the password off-drive, such as in a TPM
- Rely on the drive to transitively protect all media keys with the admin password



A working implementation



Admin password

Admin secret

Decrypt

Encrypted admin secret

Media key

Decrypt

Encrypted media key

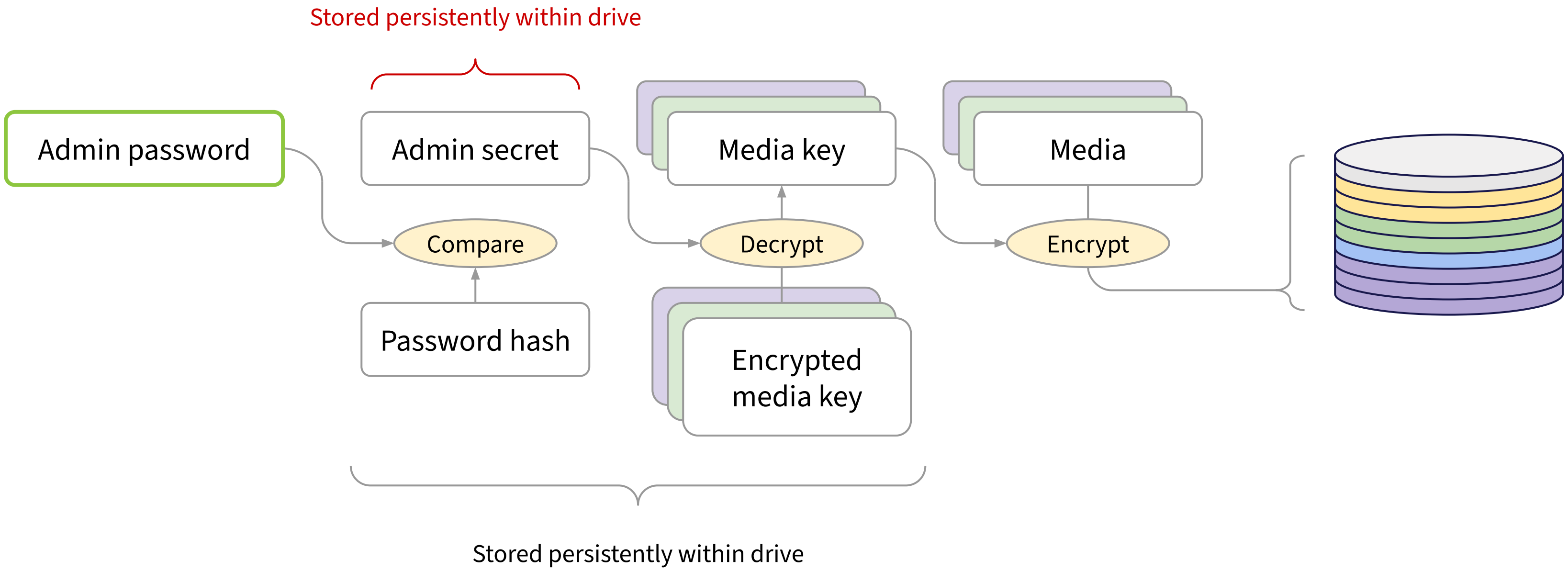
Media

Encrypt

Stored persistently within drive

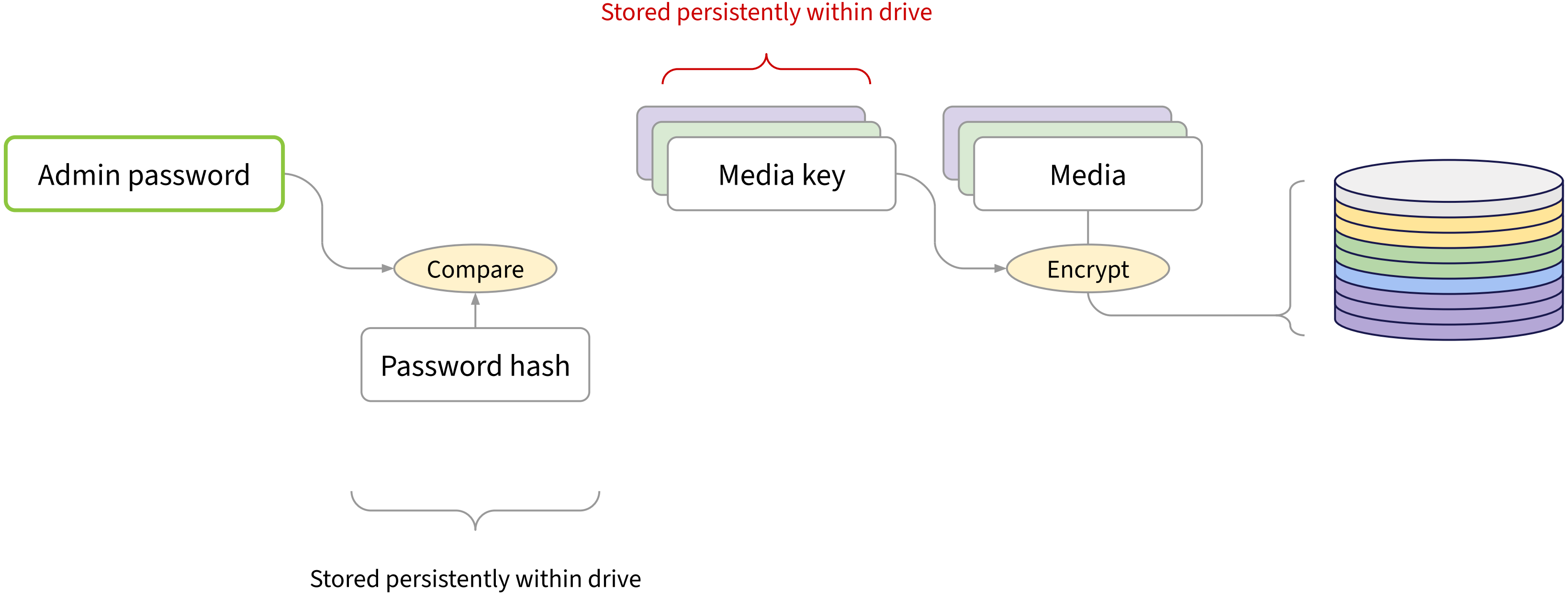
Over-simplified diagram

Broken implementation #1



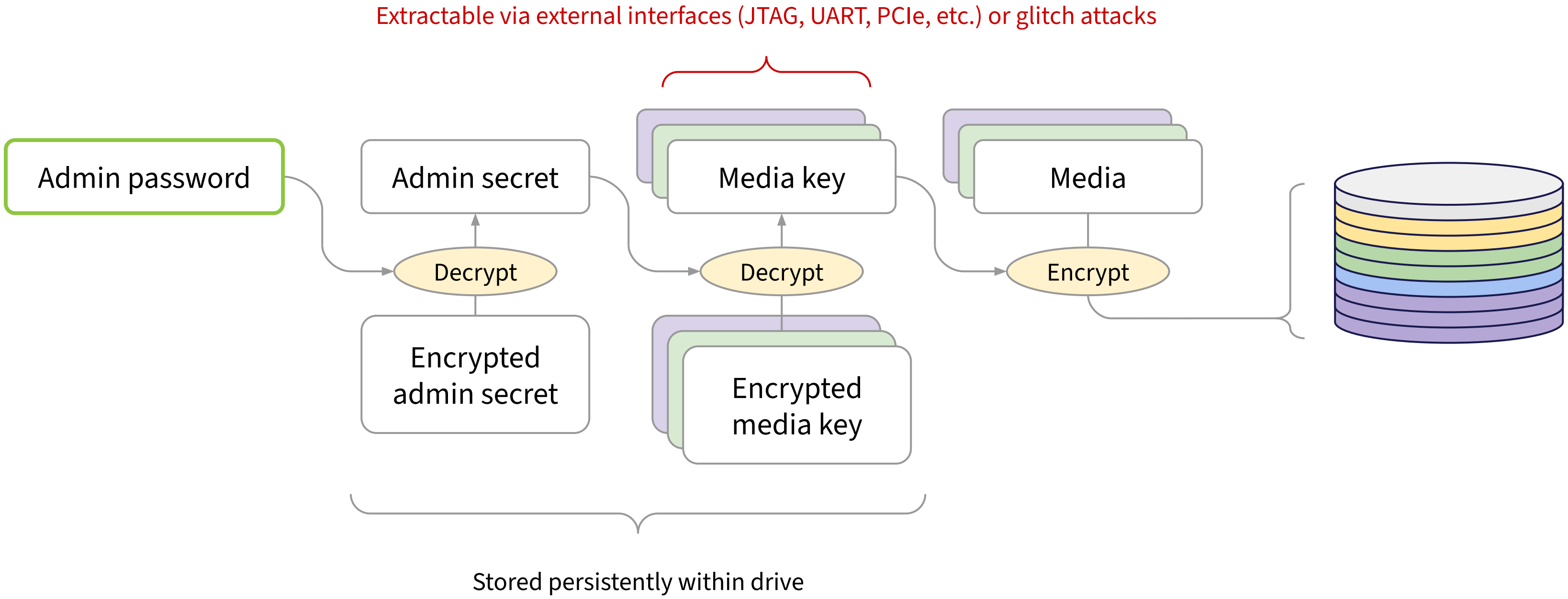
Over-simplified diagram

Broken implementation #2


Stored persistently within drive
Admin password
Media key
Media
Compare
Encrypt
Password hash
Stored persistently within drive

Over-simplified diagram

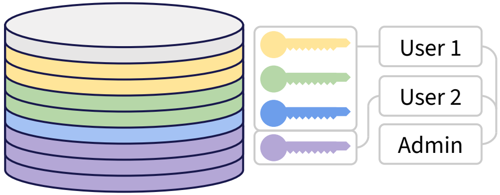
Broken implementation #3


Extractable via external interfaces (JTAG, UART, PCIe, etc.) or glitch attacks
Admin password
Admin secret
Decrypt
Encrypted admin secret
Media key
Decrypt
Encrypted media key
Media
Encrypt
Stored persistently within drive

Over-simplified diagram

Overall problem

- Storage key management is critical to get right
- Threat model is significant
  - Drive theft, supplier infiltration, hardware attacks
- Implementations vary in quality
- Auditing implementations is a chore
  - Post-deployment fixes are herculean

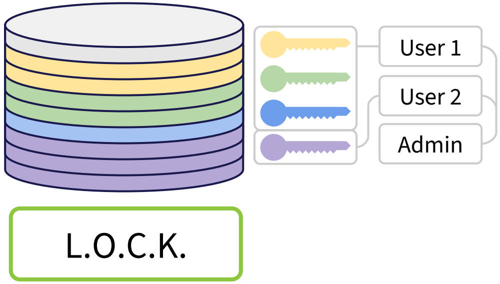


Recall: Caliptra

- Silicon roots of trust are critical components in data center hardware
- Caliptra is an OCP specification for an **internal root of trust IP block for SoCs**
- An open source implementation has been delivered at CHIPS Alliance
    - Ensures consistency, transparency, openness and reusability
- At this level, security should be boring

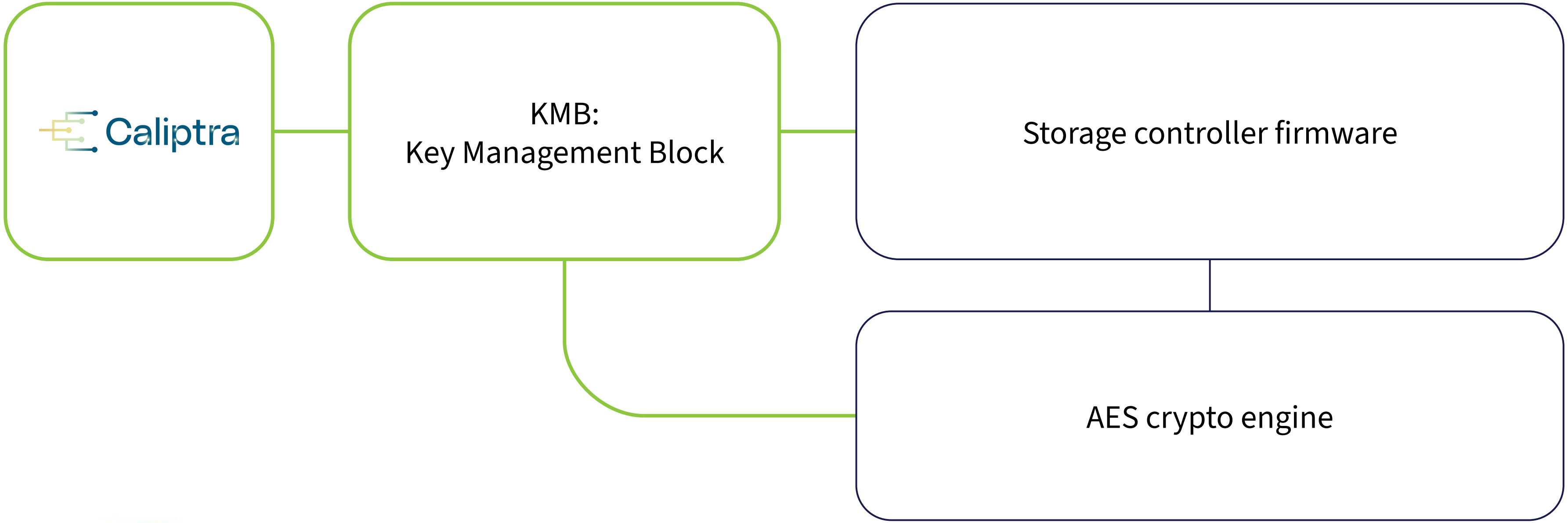
Introducing: OCP L.O.C.K.

- A project to deliver an open implementation at CHIPS Alliance, leveraging and following Caliptra
- Scoped specifically to storage devices
- Provides key management services to the drive and host, utilizing services from Caliptra



Layered
Open-source
Cryptographic
Key-management

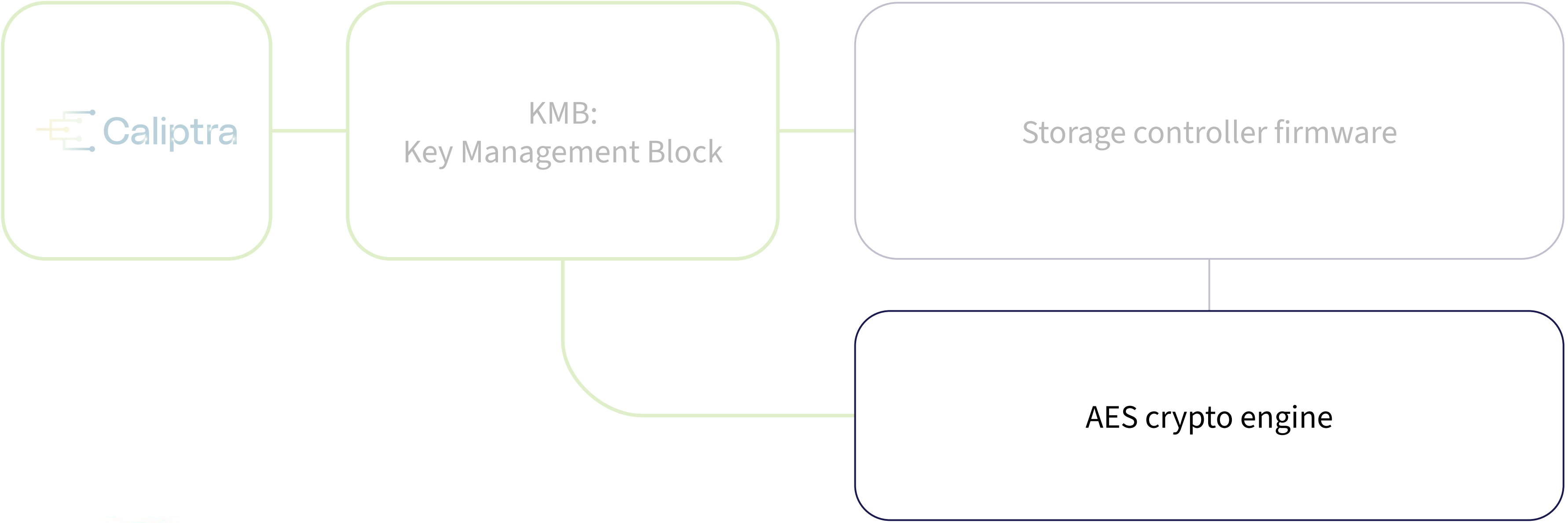
Components


Caliptra
KMB:
Key Management Block
Storage controller firmware
AES crypto engine

Components

AES crypto engine (existing)

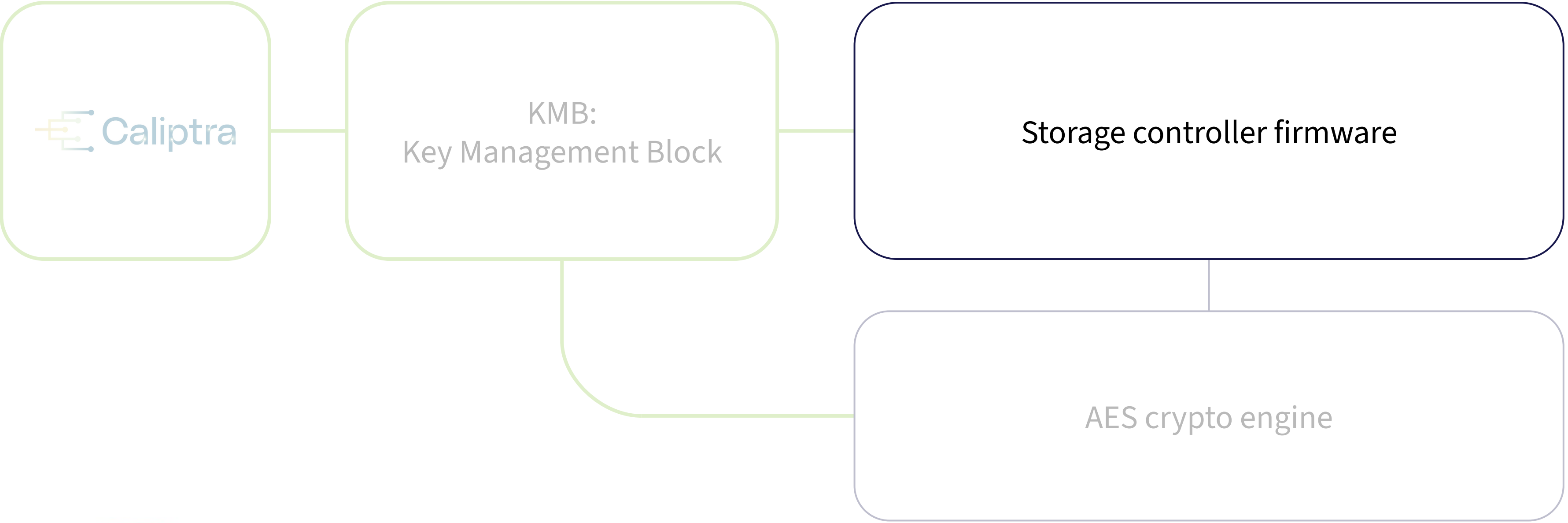
Performs line-rate encryption of data as it enters and exits the storage device



Components

Controller firmware (existing)

Manages users and wrapped keys

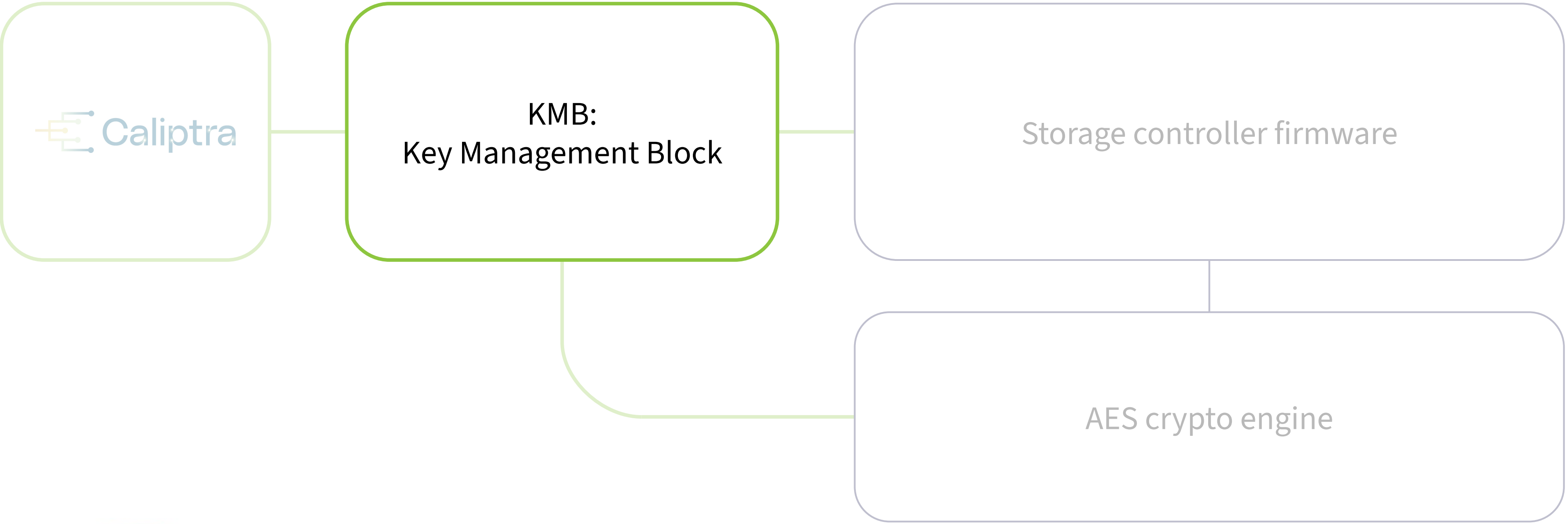


Components

KMB (new)

Generates keys and protects them at rest

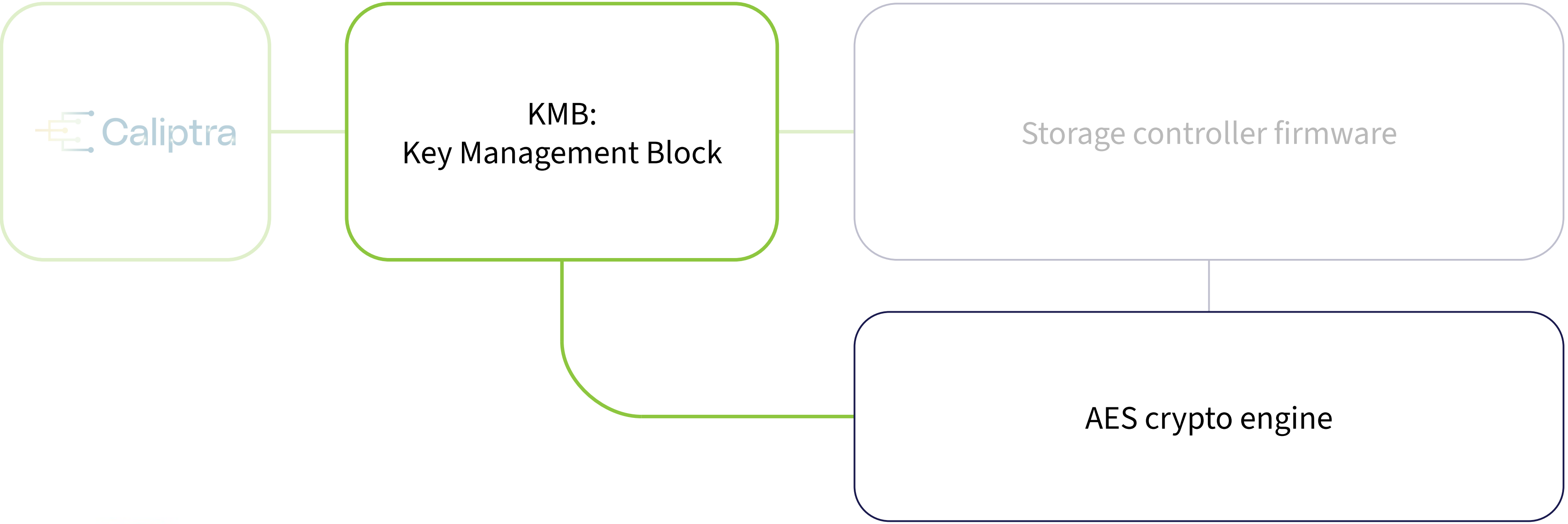
Binds keys to externally-injected seeds



Components

KMB (new)

Securely communicates media keys to the crypto engine

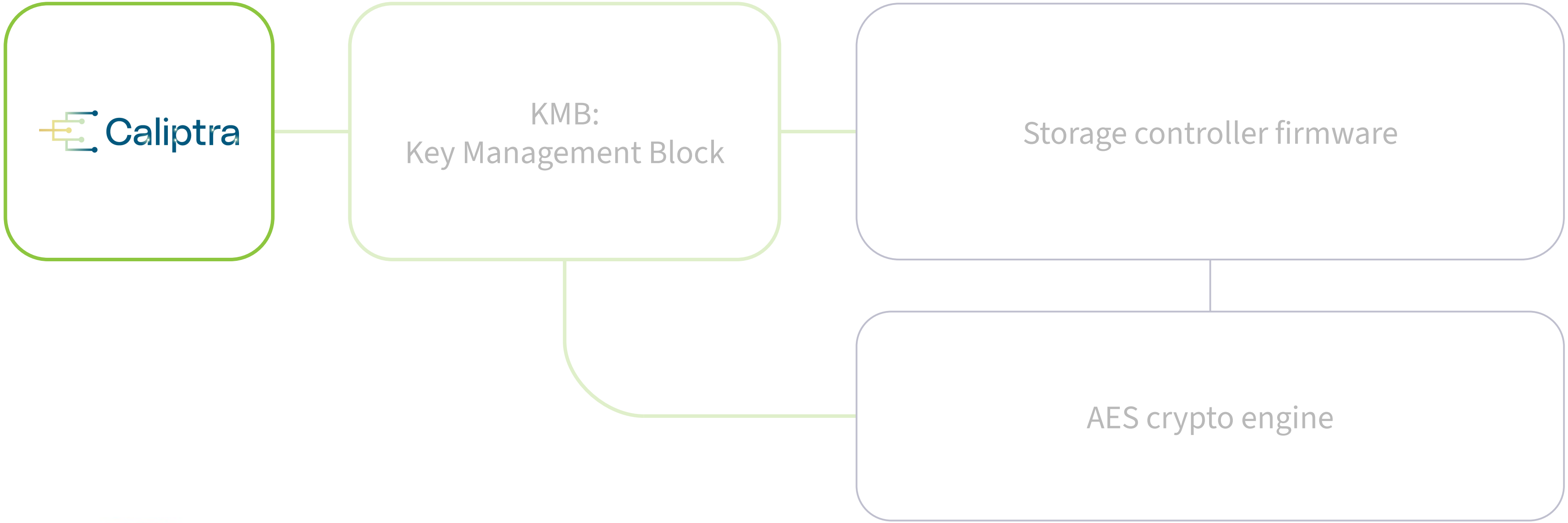


Components

Caliptra

Provides attestation services for KMB

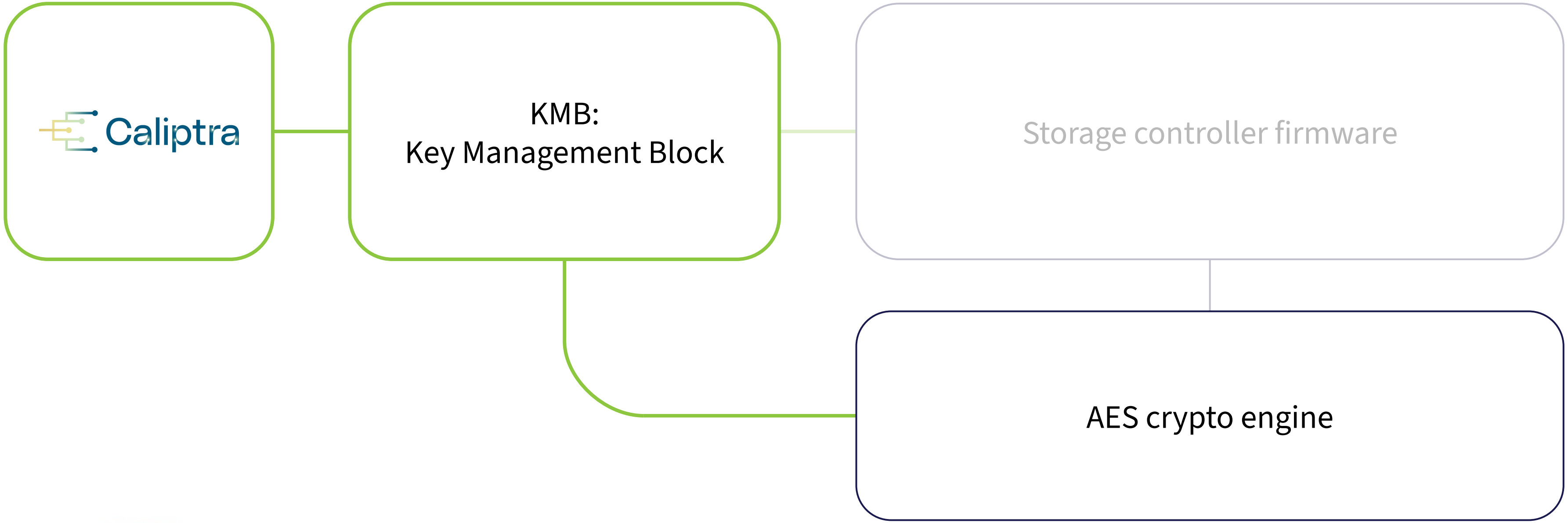
Provides root secrets for media key encryption



Components

Trust boundary

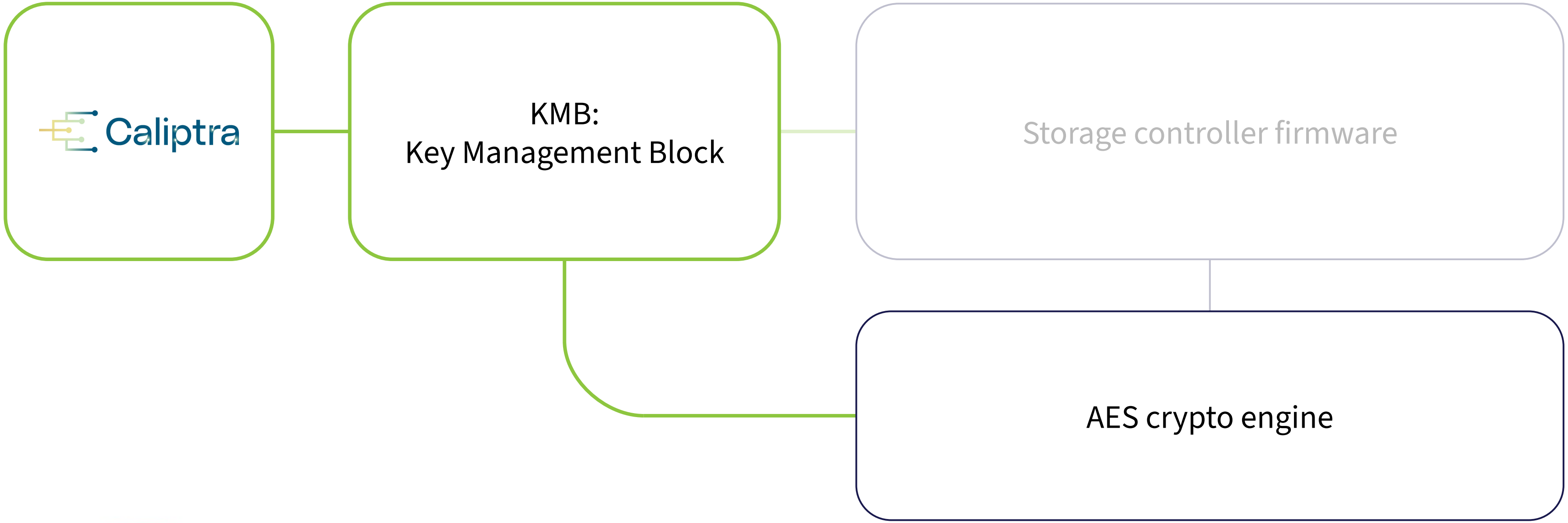
Caliptra + KMB removes system management and control interfaces from the data-at-rest TCB



Components

Trust boundary

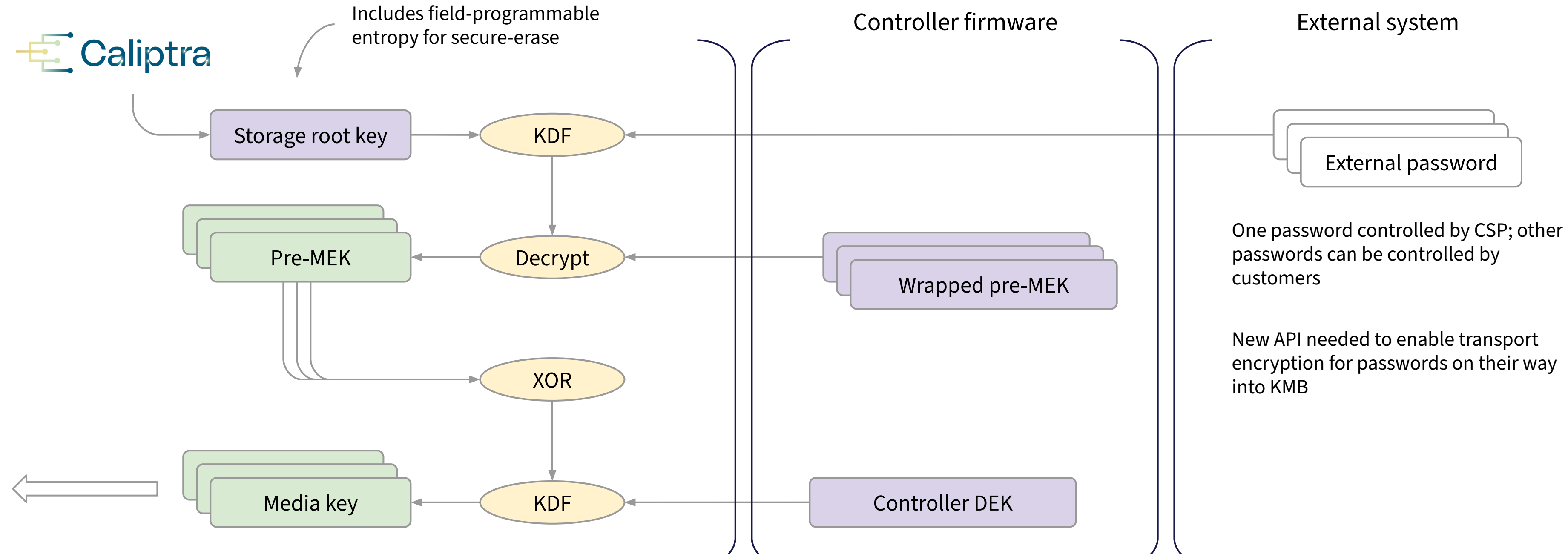
L.O.C.K. enables I/O path innovation, while maintaining a common, minimal TCB



KMB key hierarchy

🟩 Generated secret, visible only to KMB

🟪 Persistently-stored data



Includes field-programmable entropy for secure-erase

One password controlled by CSP; other passwords can be controlled by customers

New API needed to enable transport encryption for passwords on their way into KMB

Securely programmed into the AES crypto engine

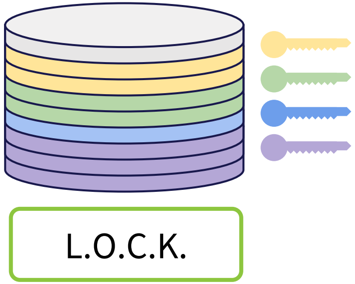
KDF: key derivation function

Summary and Call to Action

- L.O.C.K. will deliver a common IP block for storage devices
- L.O.C.K. ensures secure management of media keys

Call to Action:

- Look for the 0.5 spec later this summer
- Join CHIPS Alliance if interested in collaborating on the implementation



Thank you!