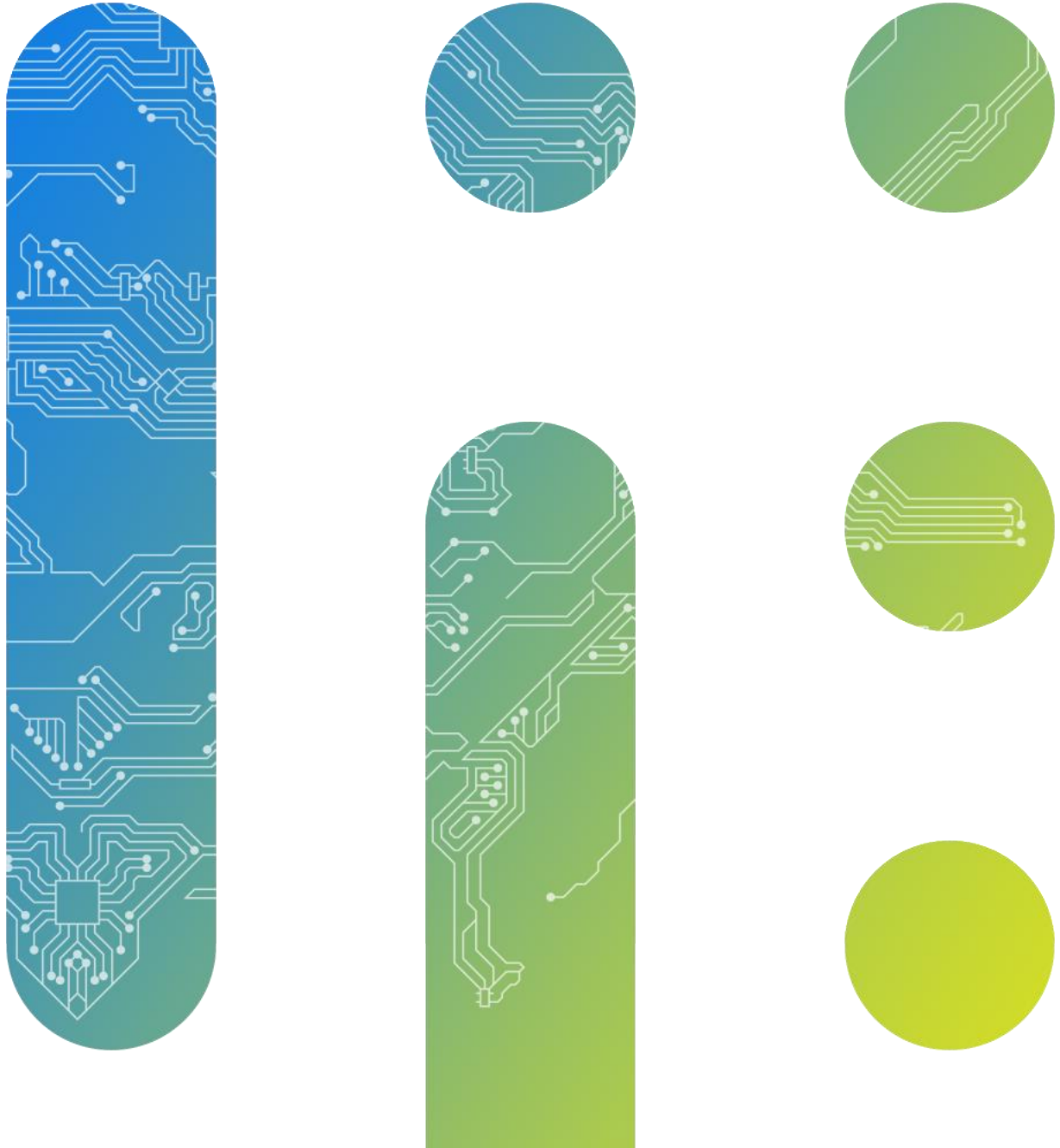


Secure Device Lifecycle

Stuart Kincaid

Security Architect, Silicon IP

skincaid@insidesecure.com

D&R IP-SOC Days

Grenoble - December 2018

inside secure

Product Lifecycle Overview

Before the product…

- Start at the product definition and design – not once you have the product!
- Adding a Unique Device identity or keys
- Adding Assets
- Debugging the application
- What happens at the end of the product life?

Product Lifecycle Overview

Once we have the product

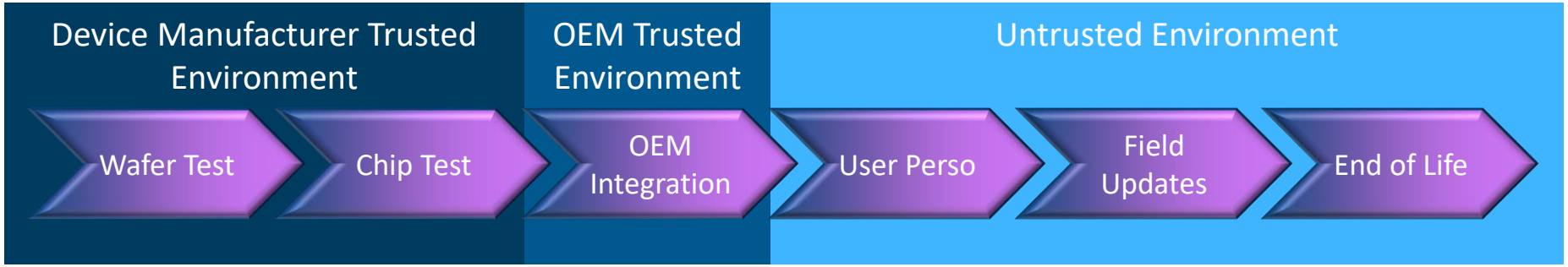
- Manufacturing and testing
- Software development and debug
- OEM customisation
- In the field
- End of life

Manufacturing Test

- What is being programmed?
  - Root Key Material – Device Identity or Hardware Unique Key (HUK)
  - Debug Authorisation / Authentication Key
  - Firmware Authentication and Confidentiality Keys
- Test access enable mechanism to open JTAG (or proprietary) interface
  - Having a fully open test interface enabling OTP to be programmed is not recommended
    Unprogrammed devices / wafers could be intercepted, re-purposed, cloned etc
  - Use a ‘transport key’ in hardware – could be in ROM, RTL or a combination of both
  - Once initial provisioning is completed, transport key unlock mechanism is locked out
  - Subsequent access requires knowledge of provisioned key material
  - Using hard fuses in OTP permanently locks out test access – can be a good option but can also limit the flexibility required to support multi-stage provisioning

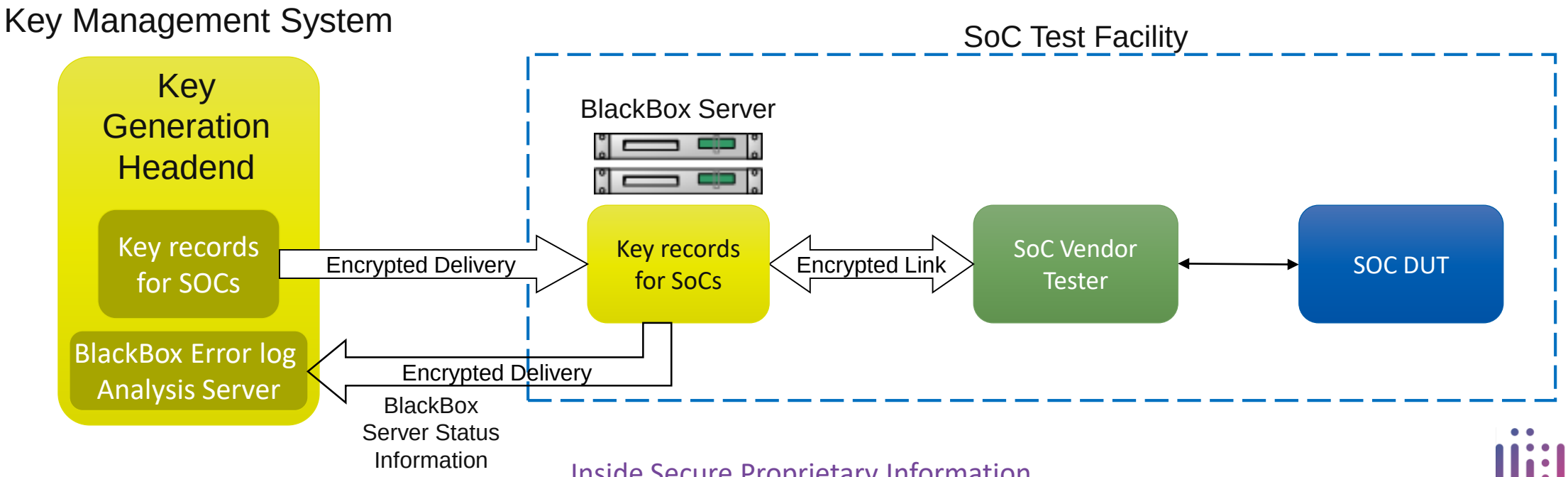
What is trusted and when?

- The manufacturing environment can vary from site to site
- In some cases, the chip manufacturer may not fully *trust* the contract test house that is performing wafer test and initial provisioning and may wish to take additional steps to protect the provisioning data. This is a complex problem to solve and requires a fully integrated, secure provisioning setup.
- However, the wafer test environment is usually secure and is considered *trusted* by the chip manufacturer.
- It may also be the case that additional key material, such as required for host secure boot is only known at chip or even board test and these stages may also be *trusted*.
- In almost all cases, it is considered *untrusted* once the product is at the user or in the field



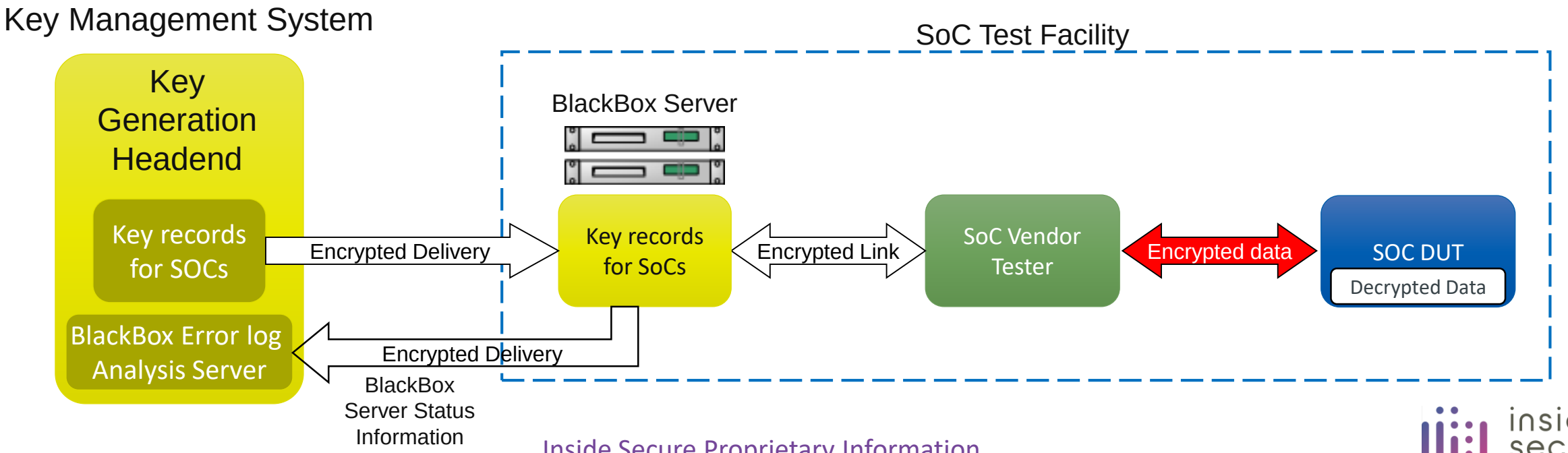
Manufacturing Test

- How do we get the identity, keys and other assets into the product?
- We need a provisioning solution – trusted environment case
  - Secure identity and key generation
  - Traceability
  - Customisable to fit with each vendor's specific requirements



Manufacturing Test

- What about the untrusted test environment?
  - Device data may be encrypted after generation using a shared provisioning key
  - The data is then fully encrypted all the way from the Key generation to the DUT, only being decrypted before storage in OTP
  - Data cannot be ‘spied’ upon in the tester or between the tester and the DUT.
  - Authenticate the tester to the SOC DUT (& may be vice versa!)

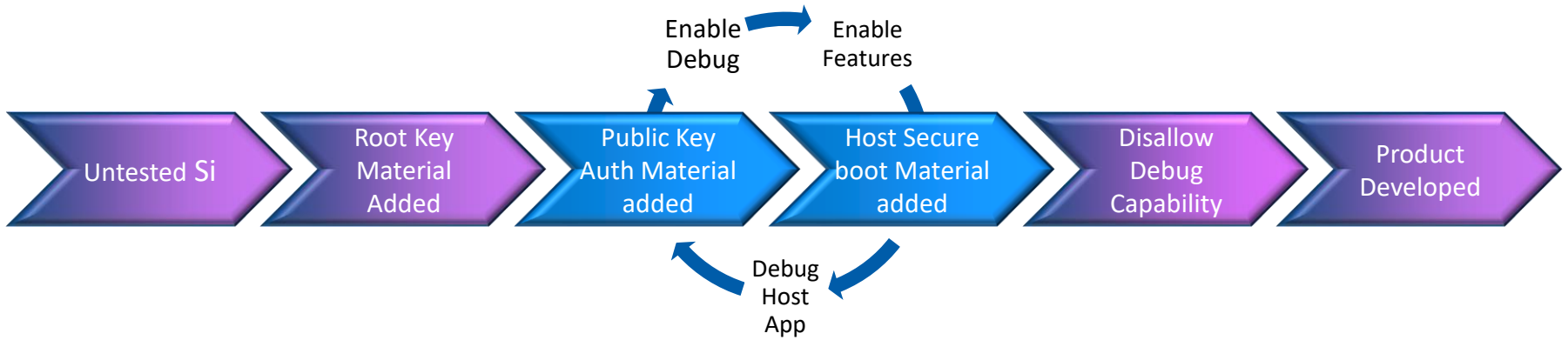


Manufacturing Test

- What if I don’t have or don’t want to use a provisioning system?
- HUK can be generated on-chip and programmed into the OTP
  - Requires TRNG to create unique keys
  - Device must be ‘functional’ ie all FW needs to be available
  - Will push the provisioning operation further downstream
- Pre-generated encrypted key blob can be imported directly into the device
  - Requires a ‘provisioning’ key to be present in hardware
  - Once new material is decrypted & authenticated, they can be programmed into the OTP
  - Pushes the provisioning operation further downstream

Debug / Development

- Debug needs to be protected – devices must never go to the field with debug enabled
- The key material added during provisioning enables a cryptographic unlock mechanism to be employed to control debug access
- Debug enablement can be limited depending on the user / lifecycle stage



OEM Customisation

- The OEM may wish to customise the product and enable or disable certain features depending on the end customer or use case
- Additional key material can be added
  - If the OEM has their own keys, these may be added functionally to the OTP
- The OEM FW can be signed using their own keys
  - The FW may be bound to the device using keys derived from the assets programmed previously
- OEM FW can be debugged if correct auth key has been provisioned

Secure Boot

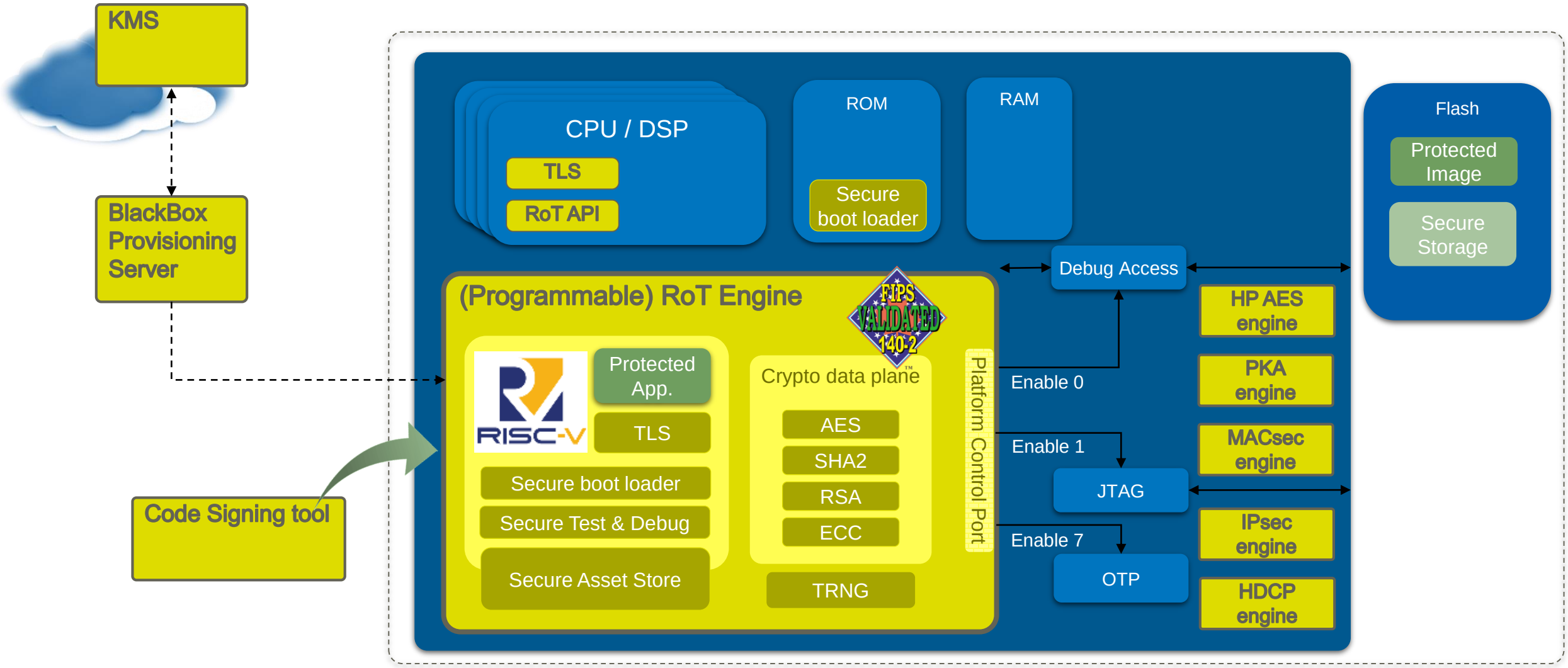
- As we now have key material available in the product, we have the ability to securely boot the host system
- FW Images created and signed with private key
  - Public key (or hash of it) stored in OTP is used to authenticate the image
- Secret symmetric root key used to decrypt the image
- Anti-rollback protection
  - Manages monotonic counters in OTP to store the image version
- Secure Boot toolkit provides
  - The image signing & encryption tool
  - A secure boot loader library to execute on the host CPU
  - Multi-stage boot & signature delegation (certificates) support
  - Customizable schemes to cope with the platform requirements

End of Life

- The device may reach it's end of life and be 'terminated' by the system
- Or, the device may detect multiple threats or attacks and decide that life is not worth living and terminate itself
- Or a functional fault may occur – detected by Power-On-Selftest say, and that is cause for stopping
- Doesn't matter why it happens, you still need to take care of the assets that are present by 'zeroising' (or vice versa depending on your logic) them
- But don't remove the device manufacturer auth key – we may want to perform some failure analysis

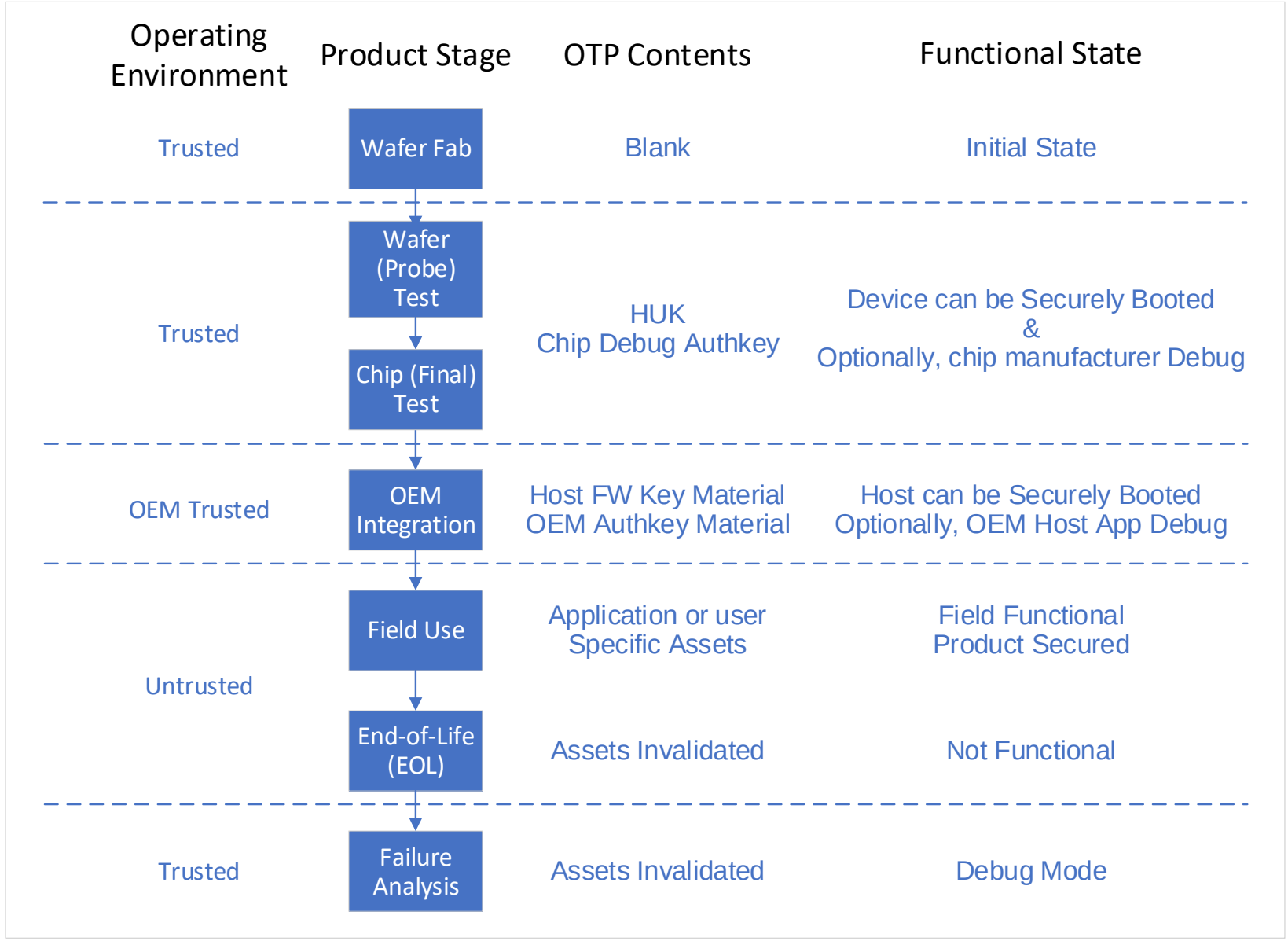
Enjoy the benefits of IP re-use

Inside Secure Root-of-Trust solution



inside secure

Summary of Lifecycle Stages

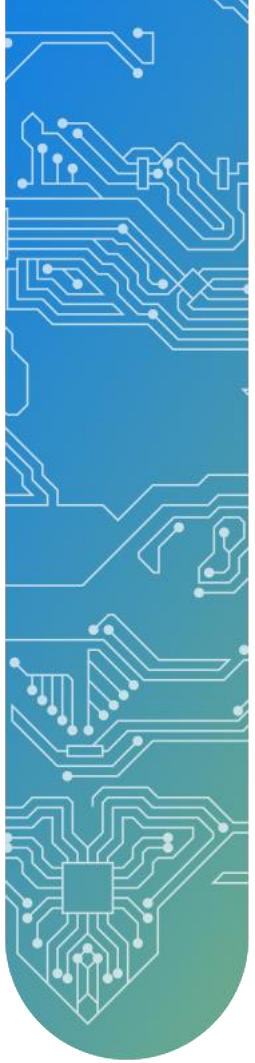


Summary – Best practices "How to Secure Your Product"

- Consider the lifecycle security of your product at a early stage in the design process
  - ✓ Match security grade to potential **impact of attack**
  - ✓ The longer the product lifespan, the **higher security** it will require
  - ✓ One size does not fit all
- Security is **unlike** other technologies
  - ✓ Functional testing does not assure security
  - ✓ Penetration testing are long, expensive and has no coverage metrics
  - ✓ Therefore **Get market-proven, mature solution**
- Security issues will happen!
  - ✓ Automatic software upgrade is essential





Thank you

skincaid@insidesecure.com

inside secure