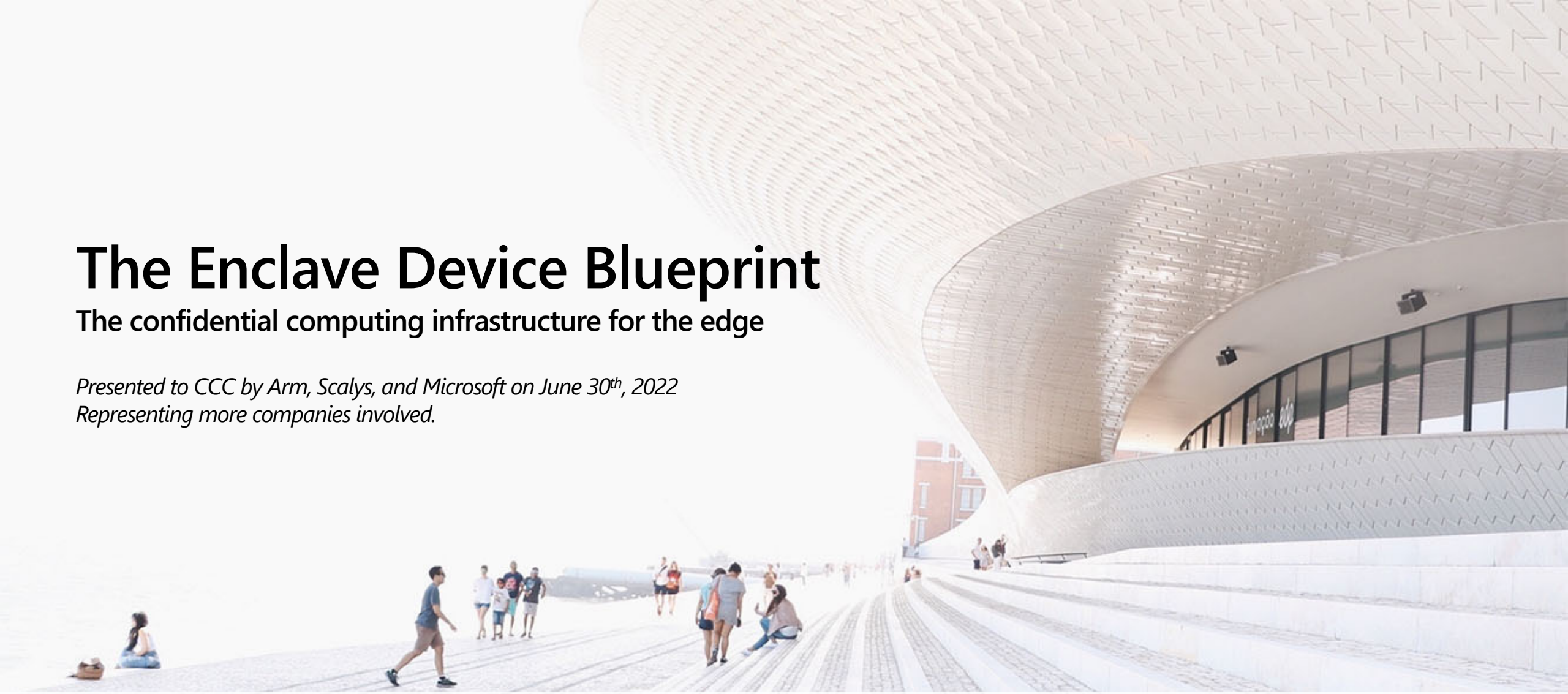
The Enclave Device Blueprint

The confidential computing infrastructure for the edge

Presented to CCC by Arm, Scalys, and Microsoft on June 30th, 2022
Representing more companies involved.



https://aka.ms/edb-whitepaper

Agenda

Brief Intro – Eustace Asanghanwa, Microsoft

Technical deep dive – Paul Howard, Arm

Feedback & Q/A - All

Today’s Goals

Awareness to cc challenges at the edge

Share the enclave device blueprint details to CCC

Seek feedback from CCC

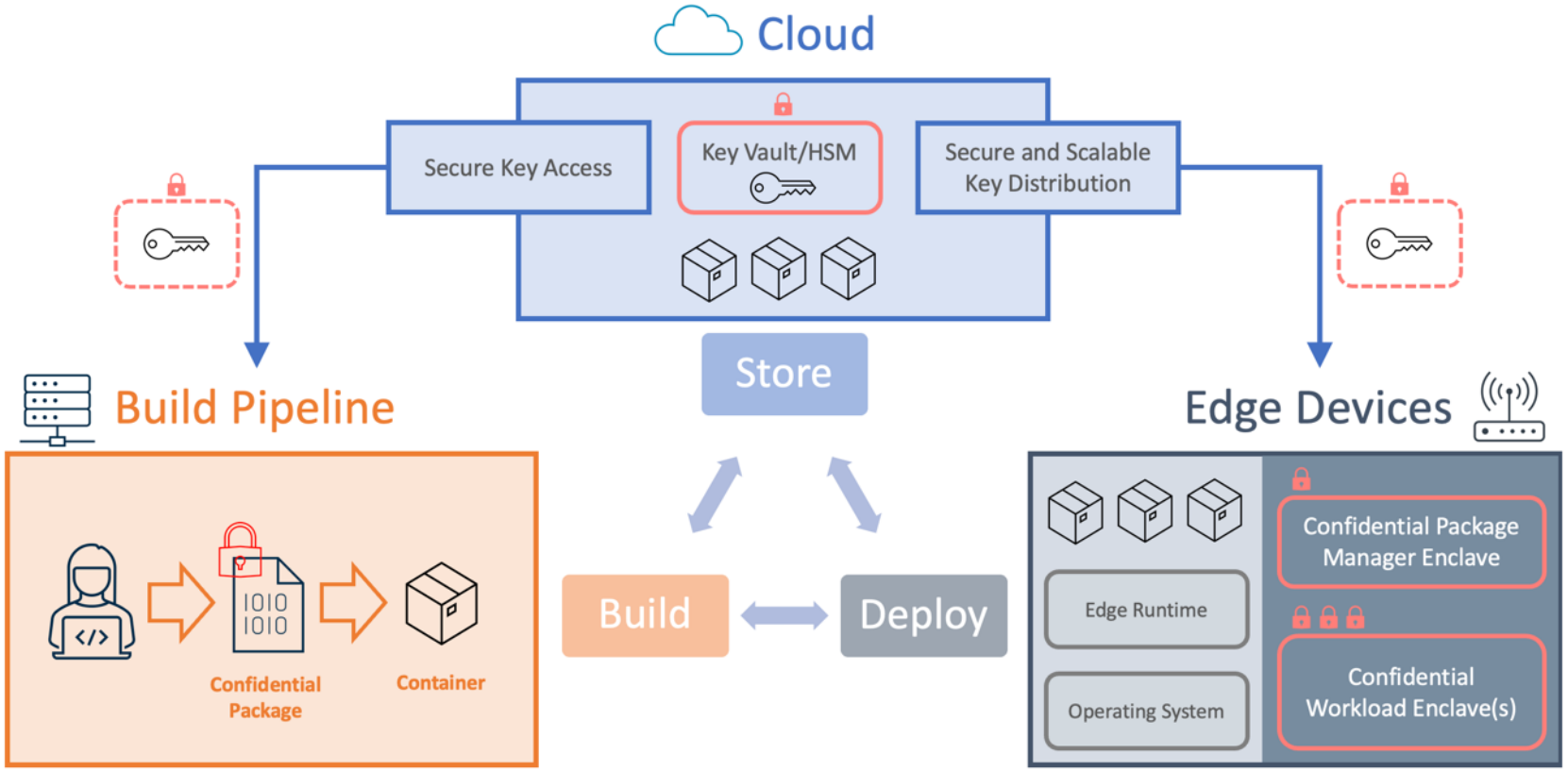
Share that we’re researching a neutral project permanent home (repo and management for long-term support)

Confidential Computing at the Edge

- Top drivers
  - IP protection (ML models, algorithms)
  - Trusted remote command & control
  - Assurances for data autonomy
  - Privacy aware computing
- Major roadblocks
  - CC tamper-resistant device engineering
  - CC workloads deployment at scale
  - Attestation infrastructure
  - Long-term device maintenance
  - Developer tooling

Enclave Device Blueprint

The infrastructure to enable confidential computing at the edge



Any Hardware . Any Operating System . Any Cloud . OE SDK

Enclave Device Blueprint devices by Scalys

Built on Arm TrustZone® TEE from NxP LayerScape® and Renesas RZ product families

https://aka.ms/edb-tbe201

Who Am I?

Paul Howard
Principal System Solutions Architect at **Arm**
paul.howard@arm.com
https://www.linkedin.com/in/paulhoward4/
https://confidentialcomputing.slack.com

@paulhowardarm

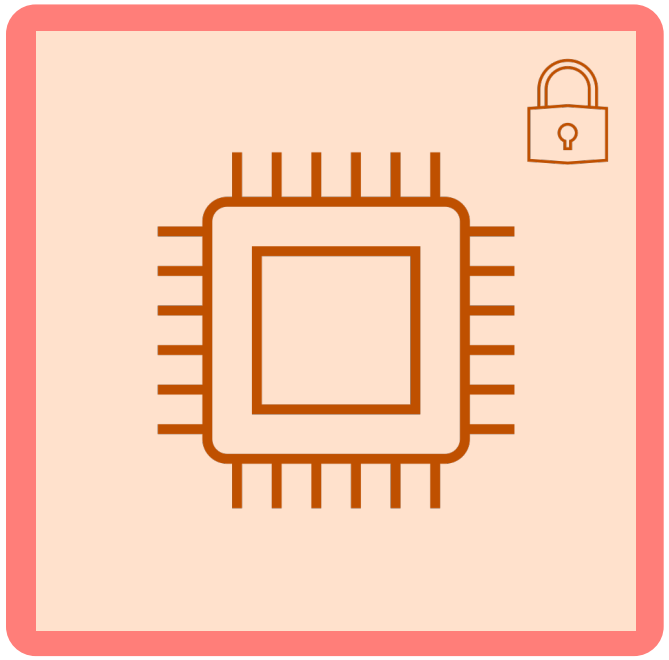
Segment Agenda

- What we set out to achieve
- Architecture
- Reusable components
- Status and open questions (leading to discussion)

Holistic Confidentiality Models

DATA AT REST

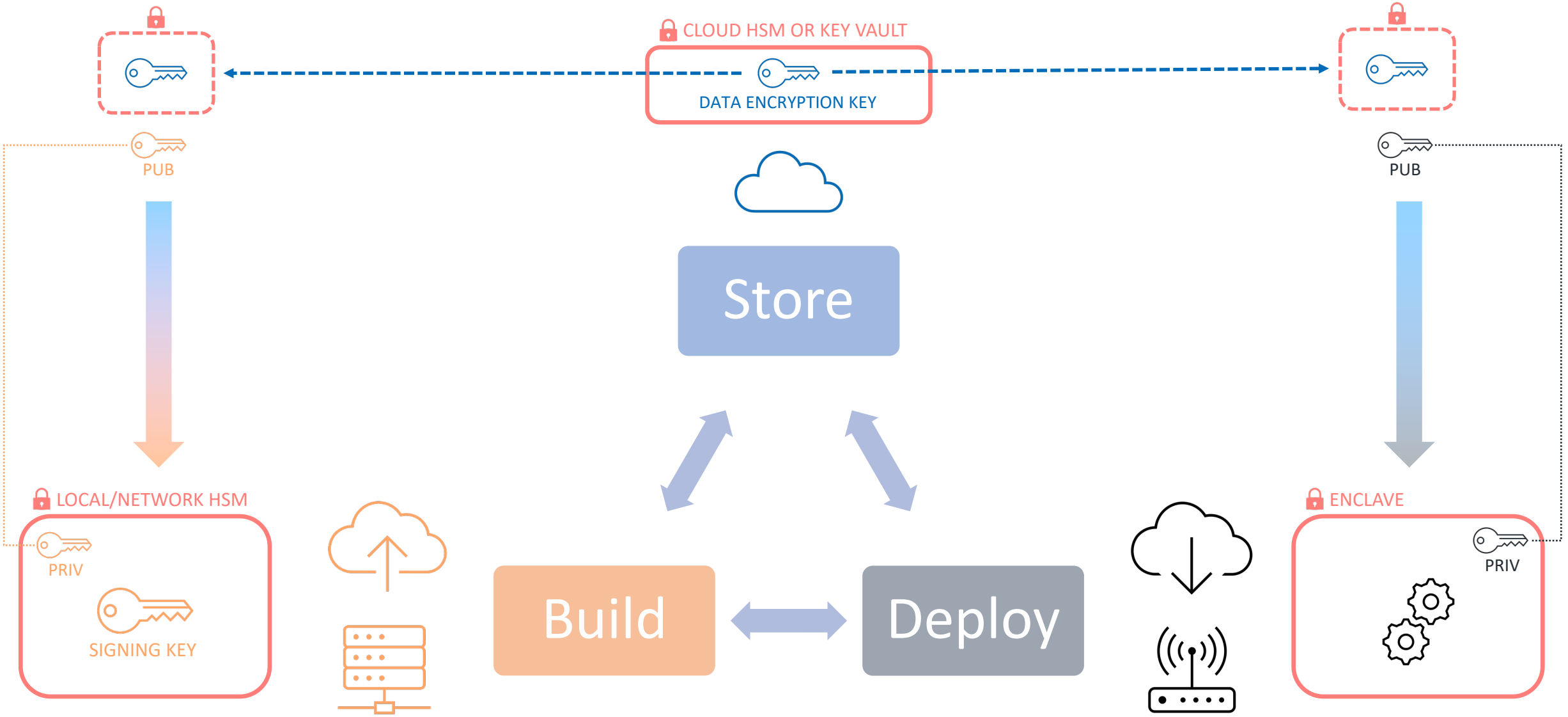
DATA IN TRANSIT



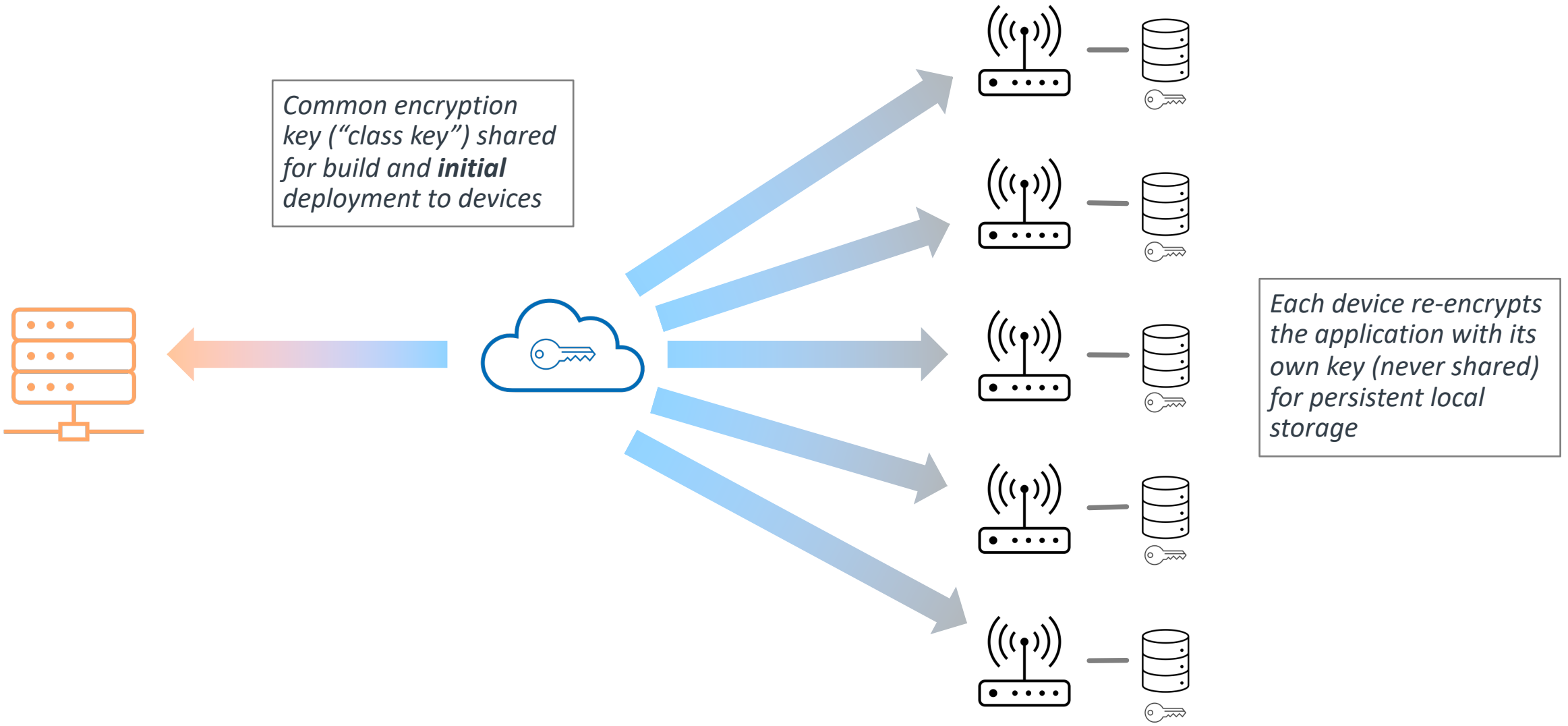
DATA IN USE

END-TO-END PROTECTION

Confidential Applications End-to-End



Scaling to Many Devices



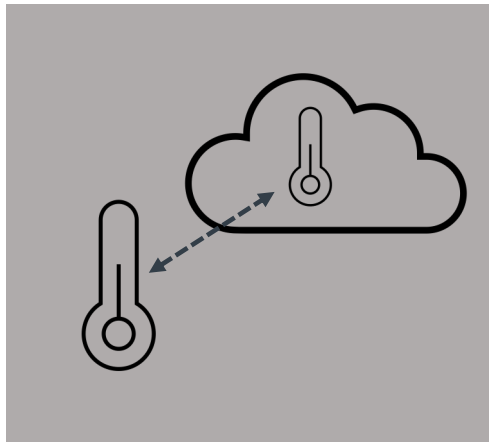
Cloud Technologies for Scale, Reliability, Portability

Cloud HSM

Cloud providers offer managed HSM and key vault solutions, allowing encryption keys to be provisioned and managed within a service-defined secure boundary and access policy

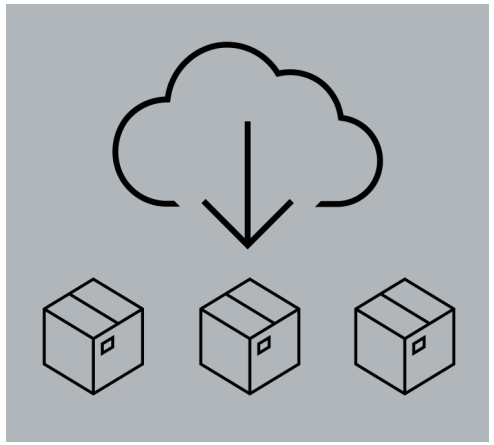
Serverless Functions

Code can be executed in response to HTTP/REST API requests, or on an event-driven or timer-driven basis



Digital Twins

A robust and scalable mechanism for synchronizing data between edge/endpoint devices and the cloud



Edge Modules

Scalable delivery of containerized applications to securely registered and authenticated edge computing nodes

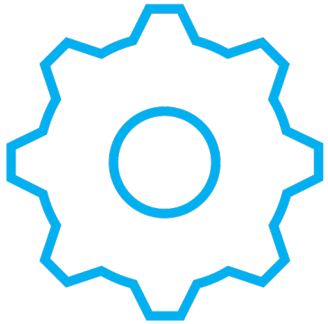
Forming a Blueprint

Design Patterns

+

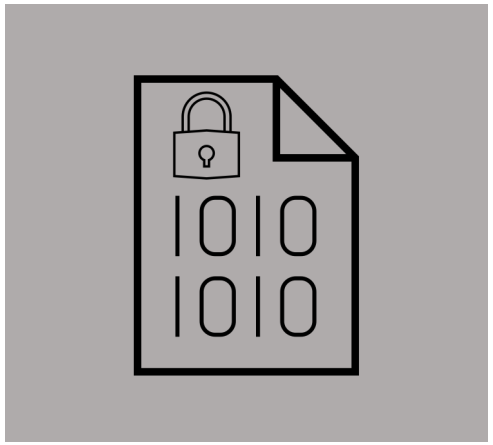
Common Technologies

+



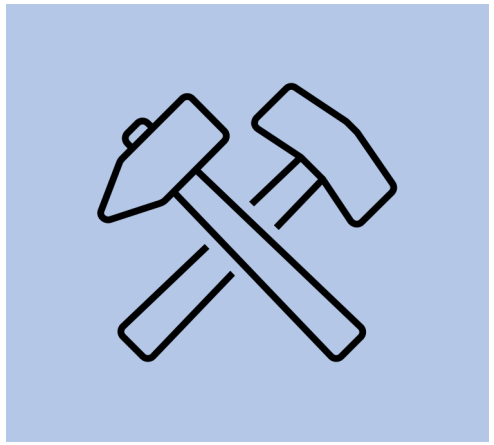
Reusable Components

Reusable Components



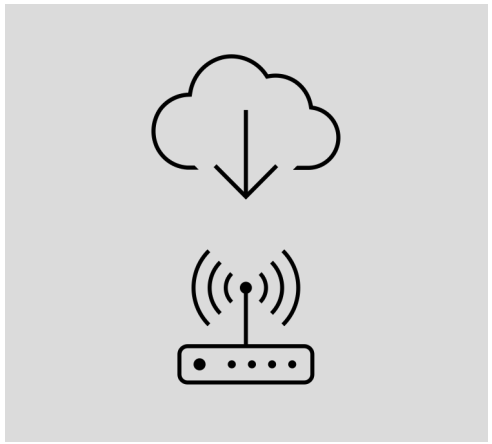
Packaging

The blueprint introduces a new portable file format specification: the **Confdential Package** (.**cpk**) file, which embeds encrypted applications for secure storage and transport.



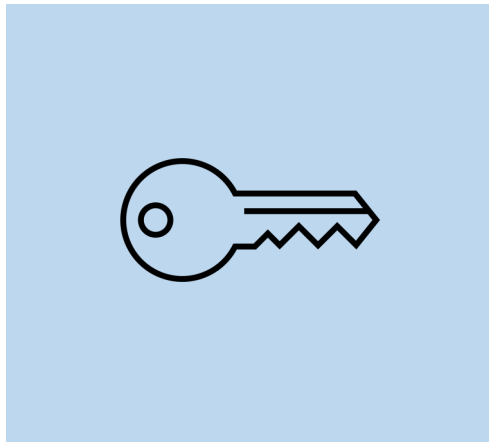
Tooling

The new **cpk-tool** shell command is used to assemble confidential packages in your CI pipeline, and to install them on a target device. Sign within the tool or using offline signatures. Certs can be embedded or referenced.



Installation

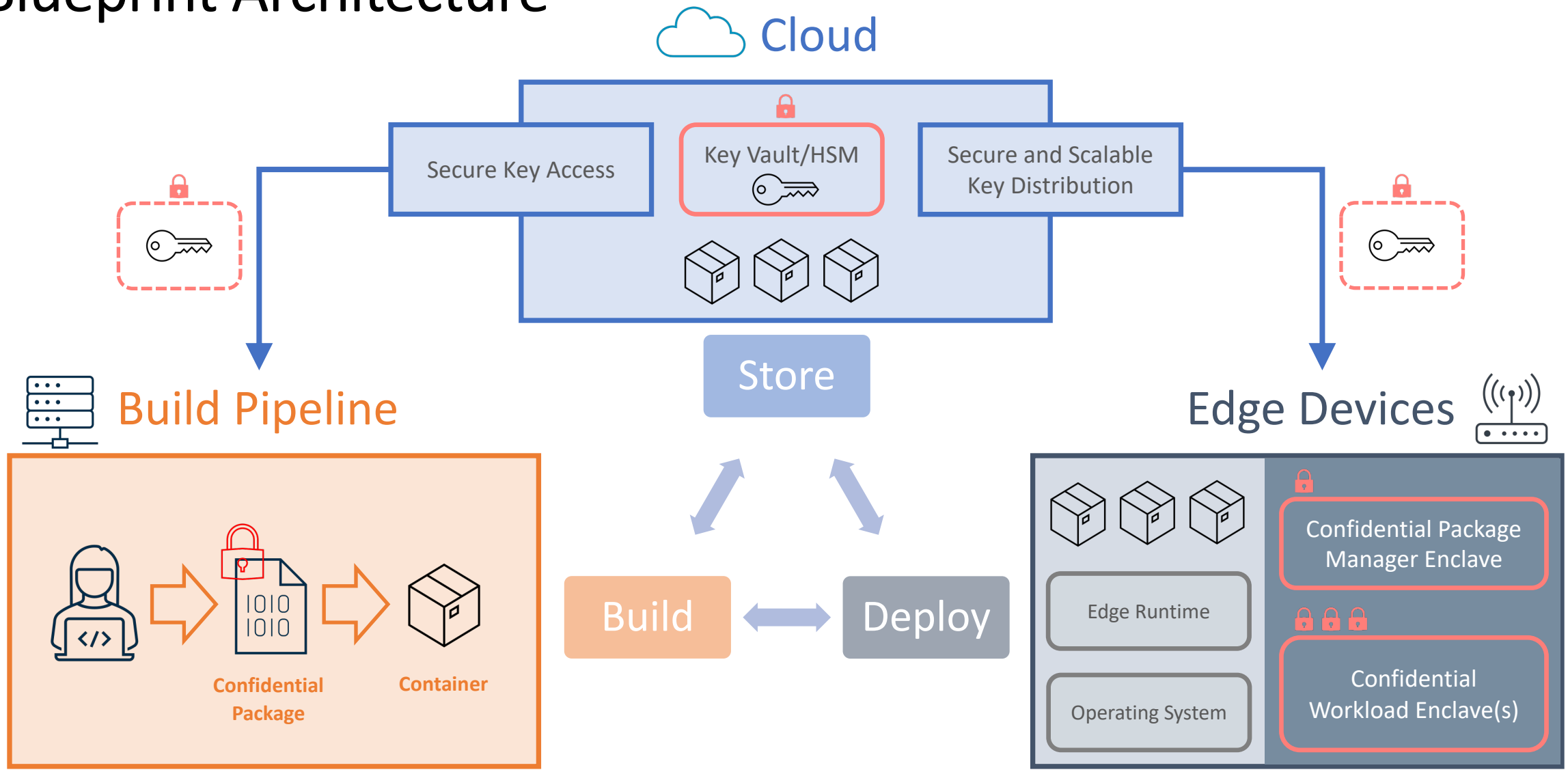
The **Confidential Package Manager** runs in a secure enclave on the target device. The **cpk-tool** interacts with this component to enable the secure delivery of workloads into new enclaves on the device.



Key Management

The blueprint includes **flexible and scalable contracts** for end-to-end key management across the build pipeline, the cloud, and the target device. Supports BYOK mechanisms. Distribution via proven scalable channels.

Blueprint Architecture

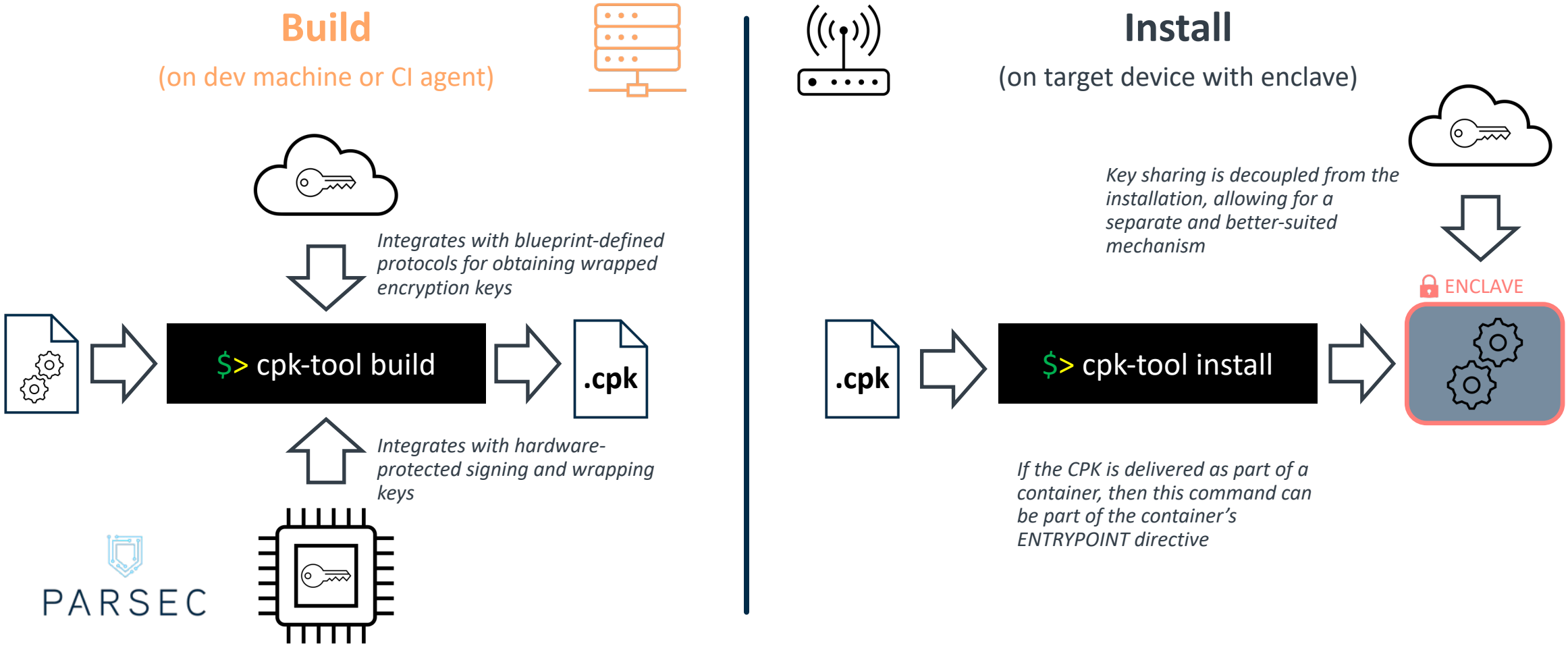


Confidential Packages

- A **vendor-neutral**, **OS-neutral** and **architecture-neutral** binary file format
- A **flexible** and **convenient** way to store and distribute a confidential application
- Bundles the **confidential payload** (typically a compiled application) along with **encryption information** and **signing information**
- Conceptually like some existing formats (such as the OP-TEE **.ta** file), but **independent** of any specific TEE or enclave technology
- Designed for **simplicity** and **extensibility**
- **Agile** with respect to cryptographic algorithms
- Supportive of **flexible signing models and trust chains**, allowing for independence of software vendor relative to target device

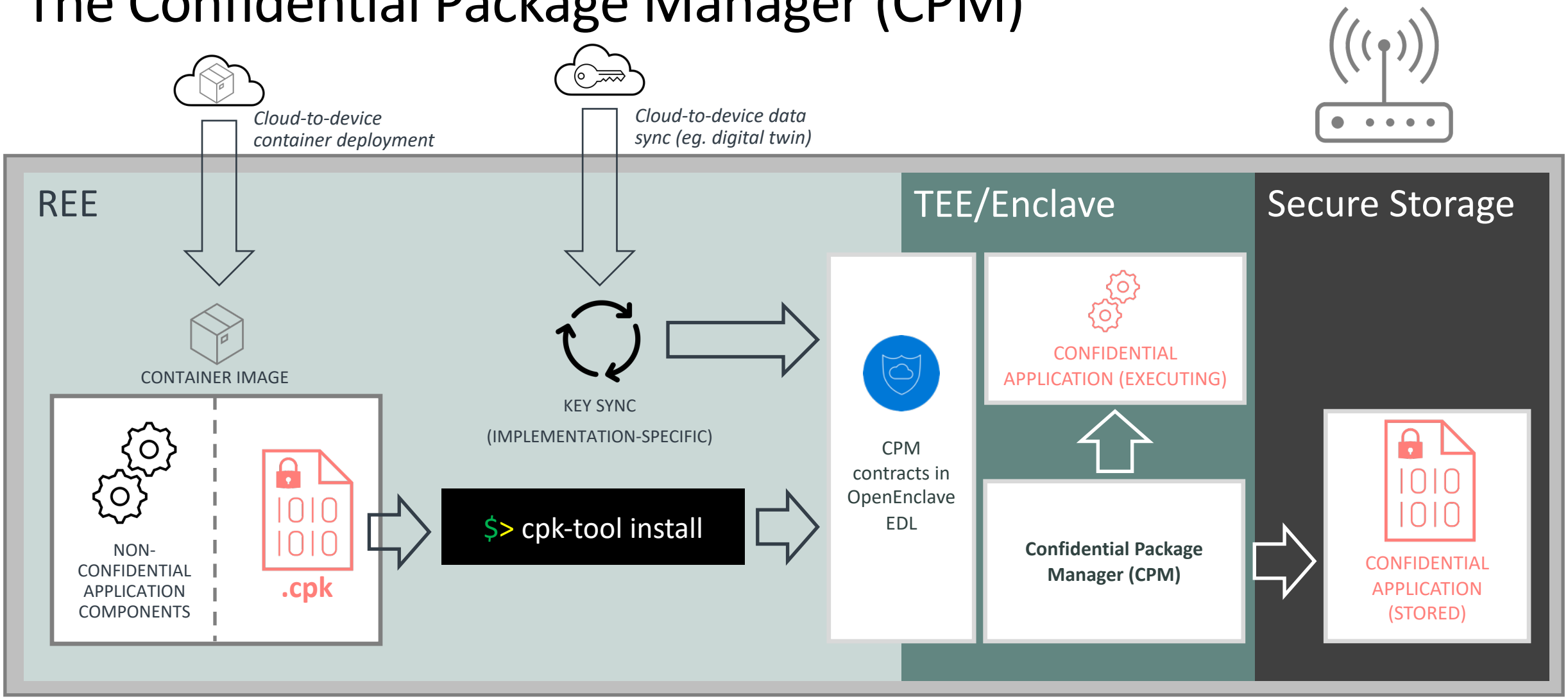
https://github.com/Scalys/ConfidentialPackageSpecification/blob/main/file_format.md

The Confidential Package Tool (**cpk-tool**)



https://github.com/Scalys/ConfidentialPackageTools

The Confidential Package Manager (CPM)



https://github.com/Scalys/ConfidentialPackageManager

Key Sharing Protocols

Implementations Specific to PaaS Cloud Provider
(eg. Cloud HSMs, key vaults, digital twins)

Standard Contracts

ProvisionKey
(HTTP Contract)

```
{
    “key_name”: “my_confidential_app”,
    “algorithm”: ”aes-gcm”,
    “strength”: “256”
}
```

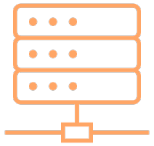
WrapKey
(HTTP Contract)

```
{
    “key_name”: “my_confidential_app”,
    “client_public_key”: “...”,
    “client_cert”: “...”
}
```

DistributeKey
(Desired State Model)

```
{
    “reported_properties”: {
        “device_public_key”: “...”,
        “device_cert”: “...”
    }
}
```

Admin



Build

Device

Status

- Blueprint demonstrated using **Azure IoT Edge** and the **Scalys TrustBox 201**
- Demo available as **Azure sample**: https://github.com/Azure-Samples/Project_Confidential_Apps_for_IoT_with_Enclaves
- Key sharing protocols currently sketchwork – no documented spec
- Attestation and key release policies need to be developed
- Tools and specs are in public GitHub repos within Scalys org – need to decide long-term home for these
- Tools currently coded to PoC/demo level only – they are not complete
- Need to investigate overlap with TEEP and maybe seek better alignment
- No integration with OpenEnclave or VSCode dev extensions yet