ISLET

Enabling On-Device Confidential Computing
Accelerating the adoption of CC on end user devices

Safety by Construction

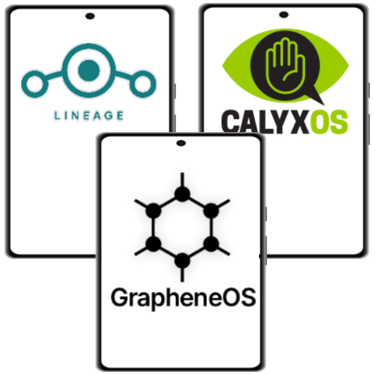
Rust-based

Formally Verifiable

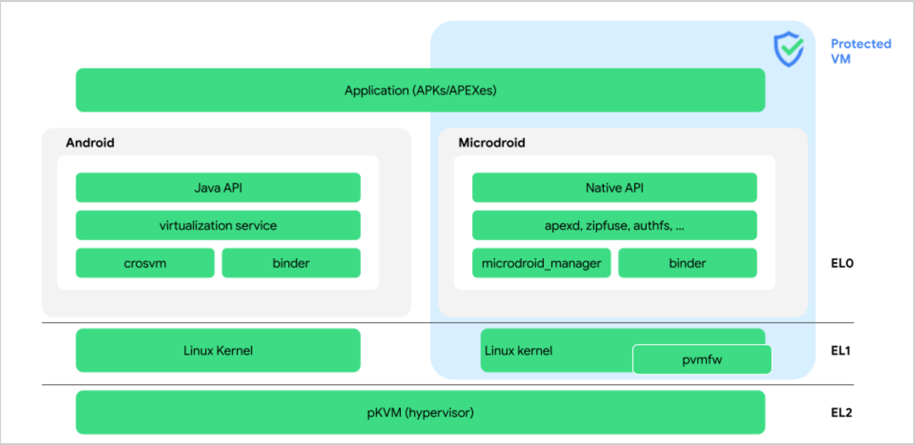
Why On-device CC?

- **Main Goal : On-device CC**
  - Protect user privacy on end user devices by applying CC technology like ARM CCA.

- **Motivation**
  - Growing demands of TEE for privacy apps. For any 3rd party apps. Even private from OS or device vendors.
  - Trends in mobile: More isolation against host OS (e.g., Android Virtualization Framework)
  - User device is the first place where user information is collected.



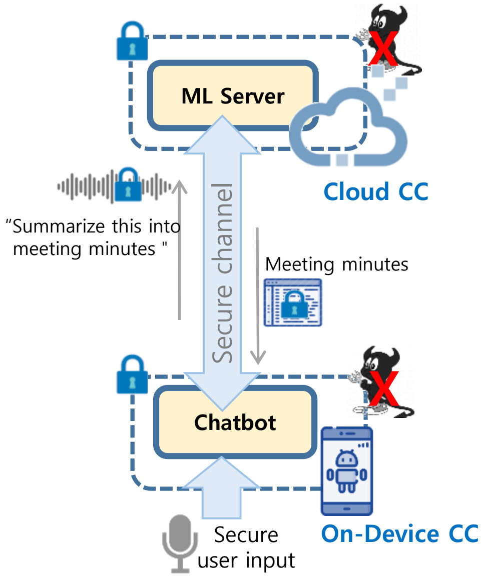
The private and secure mobile
Operating Systems
with Android app compatibility.



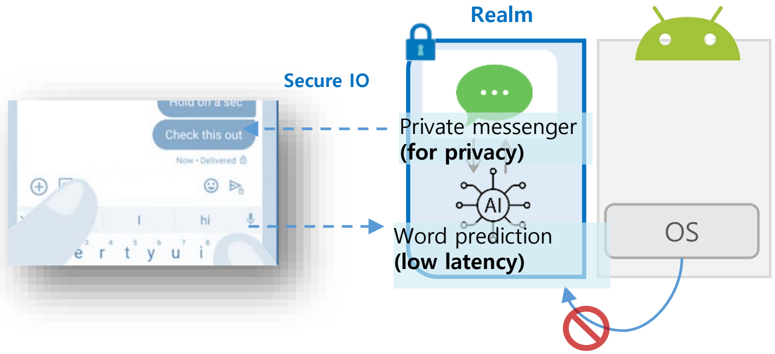
Android Virtualization Framework:
VM’ s memory is protected from the host.

Why On-device CC?

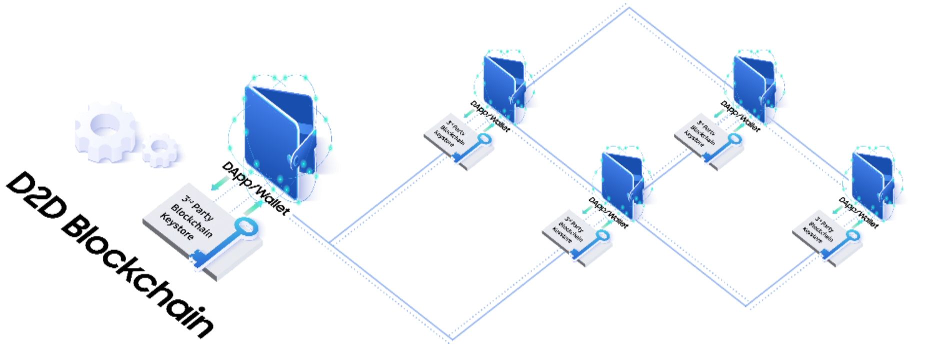
- **Opportunities with on-device CC**



[End-to-End CC]



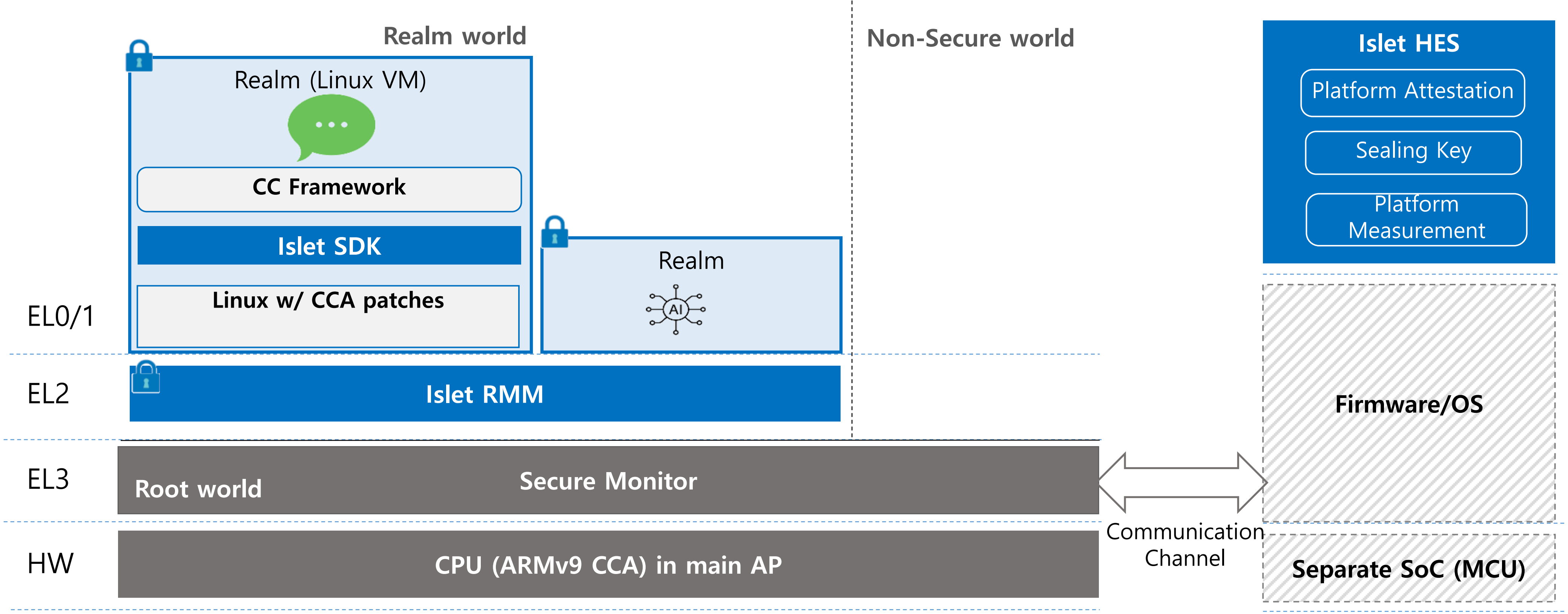
[More end device computing]
Computation offloading related to user privacy specific computing which used to be in server side



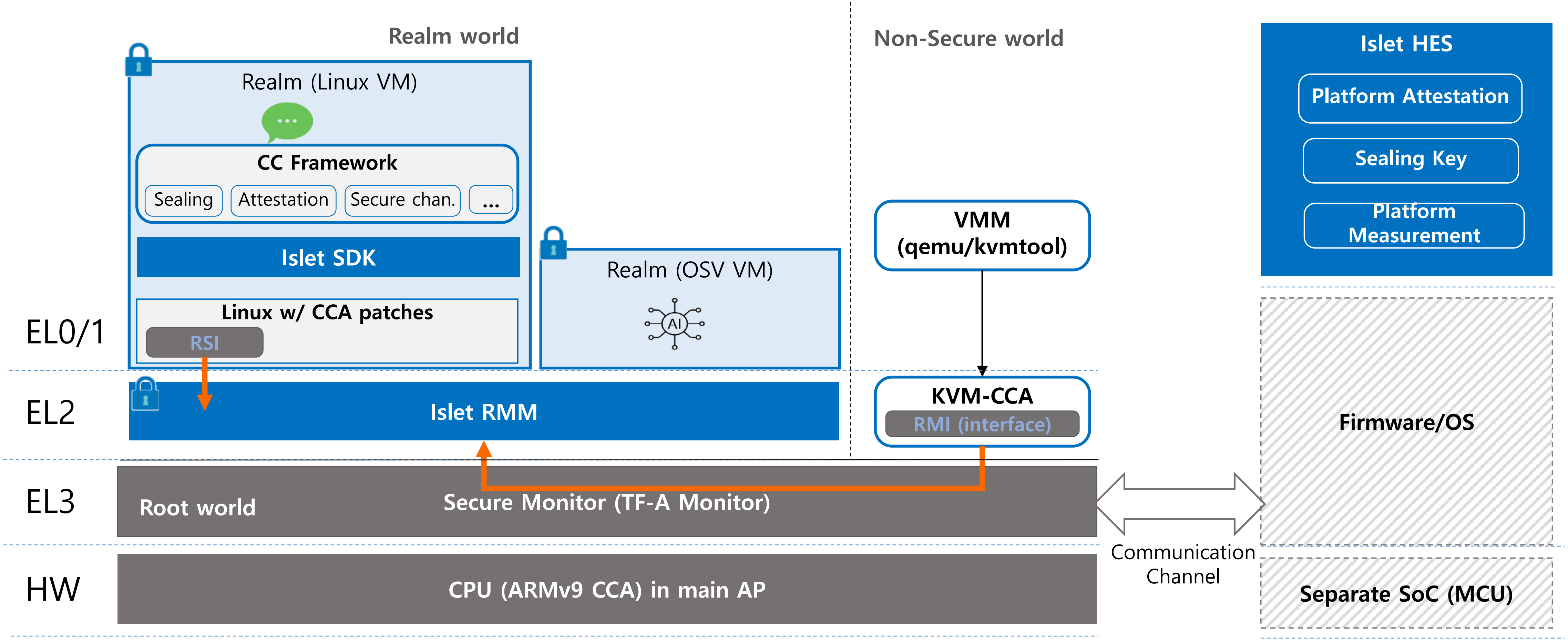
[Enabler of future computing model: M2M]

Islet Architecture

- **A project to enable confidential computing on Arm devices**
- **Focus on developing Rust-based CC platform SW on Arm CCA**
  - **CC platform SW :**
    - RMM : runs confidential VMs, aka realms, in a separate world on Arm CCA
    - HES : provides platform attestation and TCB integrity measurement
  - **SDK**



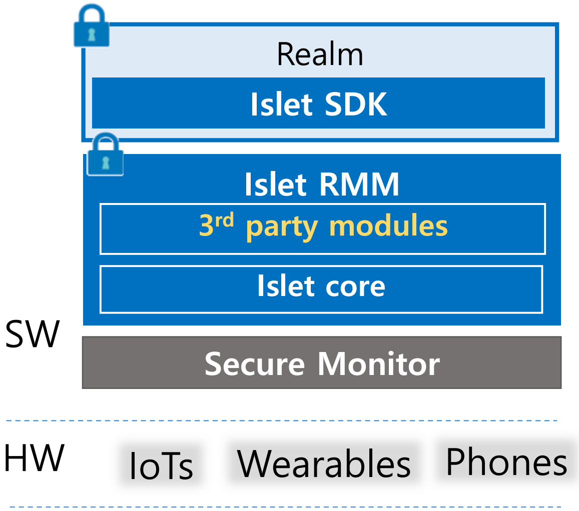
Islet Architecture



High-level architectural diagram

Islet: Safety by Construction

- Design the system with current known security challenges from the beginning
- Exploit the safeness of Rust language
  - Memory, and concurrency control safety
  - Safe Isolation of modules



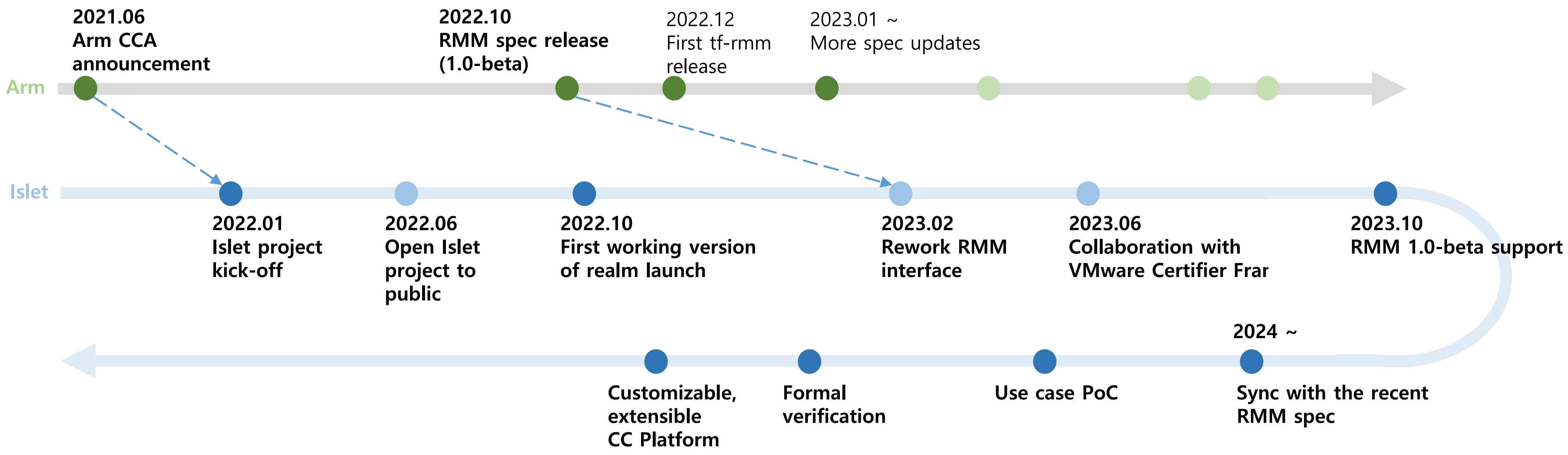
Customizable and extensible CC platform w/ minimal change

- Customizable for each HW
- Isolation of the 3rd party modules
- Islet core protected from the 3rd party modules

- Formal verification

Islet: History, Status, and Plan

- **History and current status**



Arm

2021.06
Arm CCA announcement

2022.10
RMM spec release (1.0-beta)

2022.12
First tf-rmm release

2023.01 ~
More spec updates

Islet

2022.01
Islet project kick-off

2022.06
Open Islet project to public

2022.10
First working version of realm launch

2023.02
Rework RMM interface

2023.06
Collaboration with VMware Certifier Frar

2023.10
RMM 1.0-beta support

2024 ~

Sync with the recent RMM spec

Use case PoC

Formal verification

Customizable, extensible CC Platform

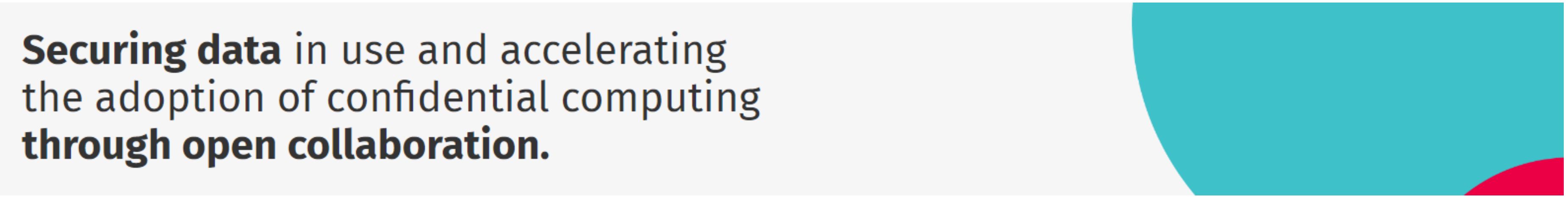
- **Long term plan**
  - Use case PoCs
  - Formal verification
  - Extensible CC platform
  - More open collaboration

Alignment with CCC’s Mission

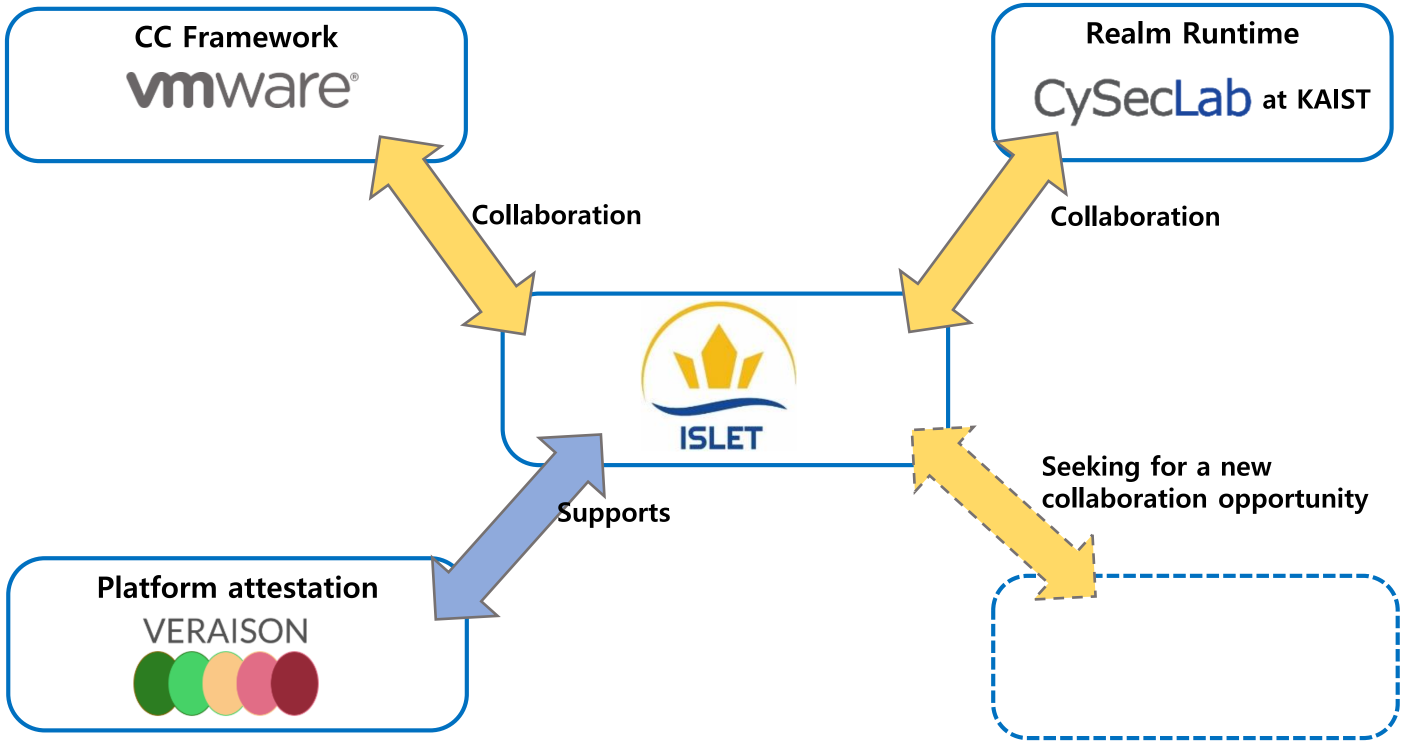
- **Why valuable to CCC community?**
    1. Diversifying the CC landscape
        - by providing an additional open-source project for ARM-based CC platforms
    2. Accelerating the CC adoption
        - With Use case-focused approach by demonstrating how CC can be used in a visible manner on these devices.
    3. Providing CC platform as a building block for CC with minimal change
        - Customizable and extensible CC platform

Alignment with CCC's Mission

- Open collaboration



Securing data in use and accelerating the adoption of confidential computing **through open collaboration.**



CC Framework

vmware

Collaboration

Realm Runtime

CySecLab at KAIST

Collaboration

ISLET

Supports

Seeking for a new collaboration opportunity

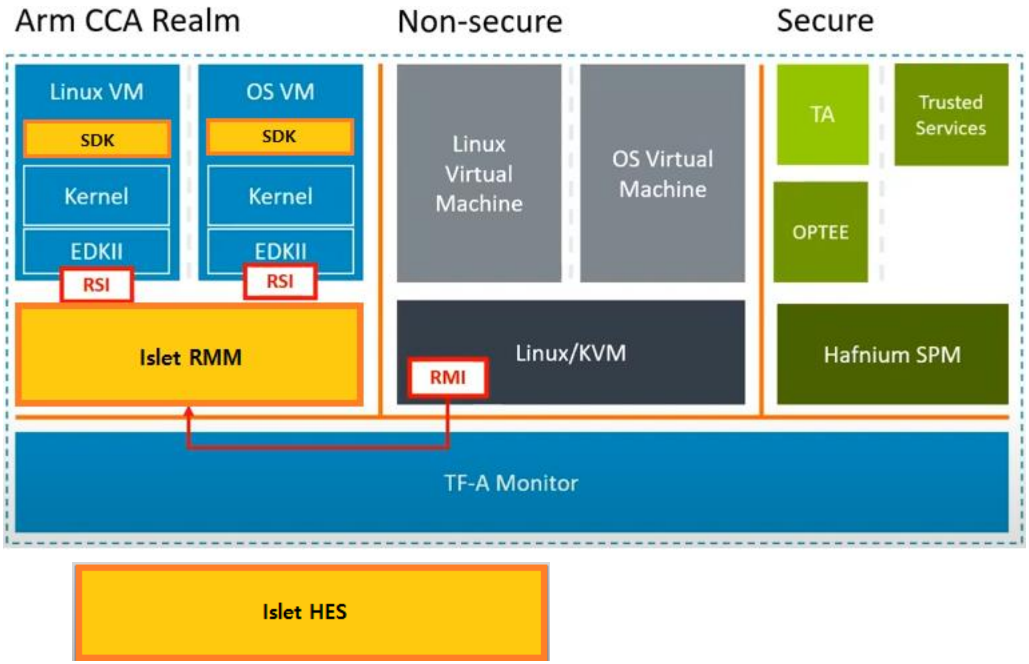
Platform attestation

VERAISON

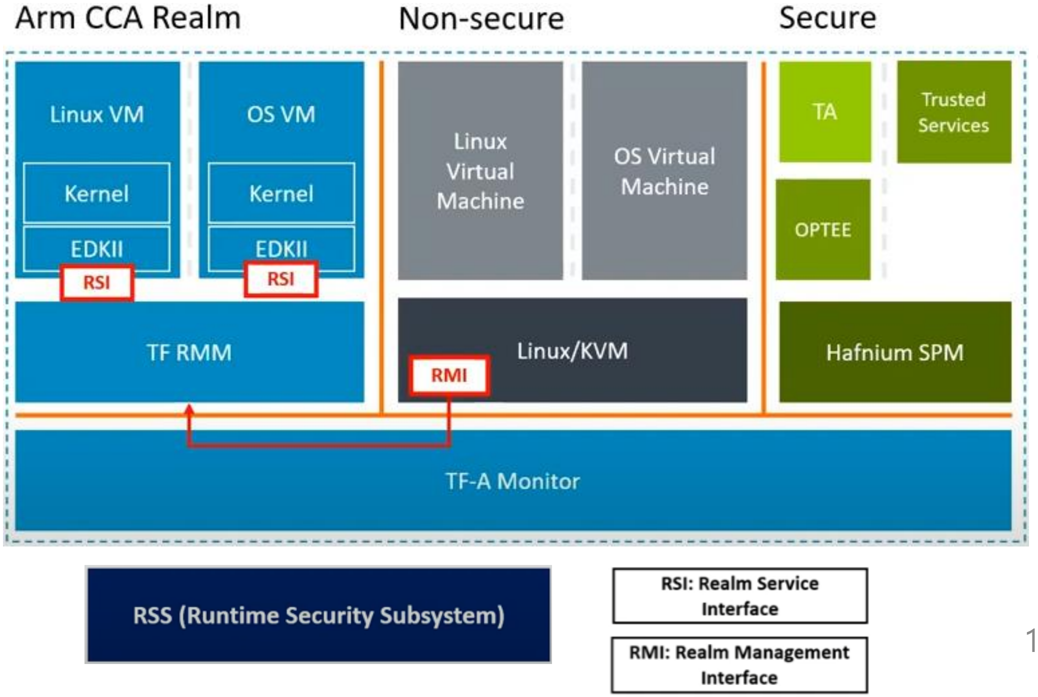
Alignment with CCC’s Mission

- **Overlapping existing projects**
  - tf-rmm and tf-a Runtime Security Service (RSS) projects by Arm
  - C vs. Rust
  - General features and for server vs. for end user devices
  - Diversity providing a broader selection of CC platforms
  - Will make effort to coordinate with the trusted firmware projects and Islet.

ISLET



Root



Root

TrustedFirmware .org