General Information

1.1. Name of Project

Islet

1.2. Project Description (what it does, why it is valuable, origin and history)

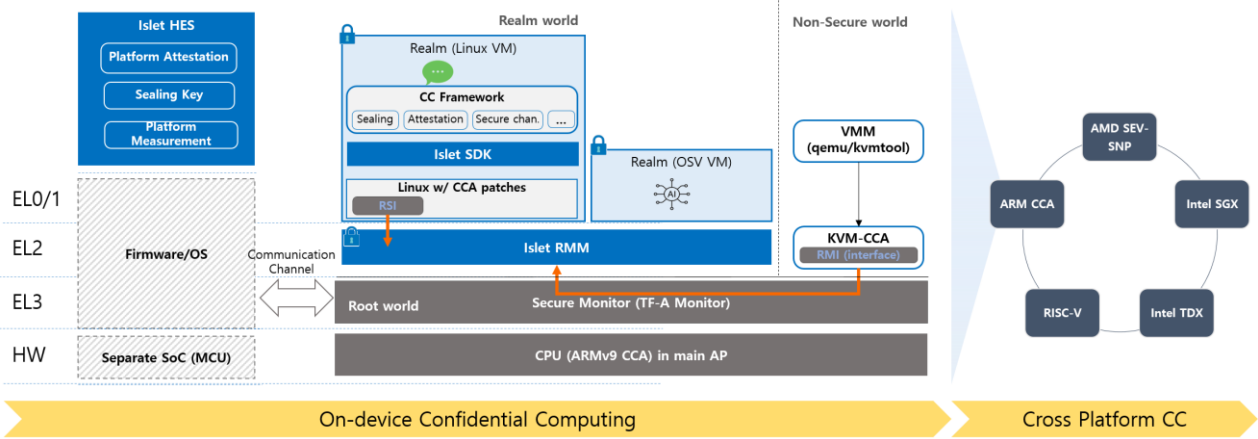
What it does:

Islet is an open-source software project that enables confidential computing on ARM architecture devices using the ARMv9 CCA, primarily aimed at protecting user privacy on end-user devices.

Islet provides a CC platform for running virtual machines (VMs) confidentially, with an SDK for easy integration with other confidential computing frameworks at upper layers.

The platform consists of two key components: Islet Realm Management Monitor (RMM) and Islet Hardware Enforced Security (HES).

- Islet-RMM manages the confidential VMs, known as realms and operates at EL2 in the Realm world on the application processor cores.
- On the other hand, Islet-HES performs device boot measurement, generates platform attestation reports, and manages sealing key functionality within a secure hardware IP apart from the main application processor.



Overall Architecture of Islet (NOTE: Currently Islet-HES runs as a stand-alone PC application that communicates with the AEM FVP emulator via UART)

In designing Islet, we aim to to address the current security challenges in confidential computing technologies right from the very beginning.

- To ensure our software safety, we have chosen to use the Rust programming language, known for its unique security model that ensures memory safety and concurrency safety.

- With the Rust compile-time safety validation, we can securely integrate third-party modules into Islet RMM without compromising security. This allows for customized CC platform in various applications like IoT, wearables, and mobile phones. Moreover, by introducing well-crafted RMM interfaces, the integration process can be accomplished with minimal changes.

- Moving forward, we also plan to incorporate formal verification techniques to further enhance the security of our design and implementation.

Why it is valuable:

- Islet provides an open-source project for ARM-based CC platforms, which are in the early stages with a relatively low number of projects. In addition, Islet addresses the need for user privacy protection on end-user devices. While current confidential computing solutions primarily focus on server-side protection, it is equally important to safeguard user information at the user device since that is where private data collection initially occurs. Furthermore, as more and more users rely on privacy apps such as private messengers, secure emails, password managers, and web browsers with privacy settings, there is a growing need to ensure privacy on user devices.

- We aim to explore and implement on-device use cases on the Islet platform and identify additional features that need to be added to support them. This use case-focused approach will bring unique improvements to the CC platform for user devices and accelerate adoption by demonstrating how CC can be used in a visible manner on these devices.

- The value of Islet extends beyond diversifying the CC landscape; it also facilitates innovative and secure computation. Enabling CC on user devices will not only establish end-to-end CC throughout the entire data processing path, but it will also aid in the development of a secure computation model that allows for the processing of user private data on the user device using the same components that previously were employed at the server side without disclosing business logic. Islet enables users to have more options for self-sovereignty by allowing them to maintain control over their private data without server involvement.

Origin and history:

Islet was first conceptualized and developed by the Security & Privacy team within Samsung Research and made public in 2022, with the goal of drawing in collaborators from the broader community. The team made strides in 2023, incorporating support for the RMM specification by ARM's definition to further align Islet with industry standards.

1.3. How does this project align with the Consortium's Mission Statement

We align with this mission statement in some aspects:

- Our mission is in line with CCC's mission to accelerate the acceptance and adoption of confidential computing by defining a confidential platform for ARM devices and demonstrating its practical use cases in a visible manner. By showcasing the benefits and capabilities of confidential computing on these devices, we contribute to the acceleration of adoption in the market.
- An extensible and customizable CC platform with minimal change while safety is maintained is made possible by the choice of Rust programming language.
- In order for Islet to serve as one of the foundational building blocks of CC, we are pursuing efforts to further improve its interface with other confidential computing projects. Islet is open to collaborating with others on this, and as a beginning, it has started collaborating with the VMWare Certifier Framework for CC project.

1.4. Project website URL

We are hosting a project website using Github pages within Samsung at https://github.com/Samsung/islet.

1.5. Social media accounts

N/A

Legal Information

2.1. Project Logo URL or attachment (Vector Graphic: SVG, EPS)

https://github.com/Samsung/islet/blob/main/doc/res/logo-title.jpg

2.2. Project license. We recommend an OSI-approved license, so if the license is not one on the list, explain why

Apache 2.0

2.3. Existing financial sponsorship

Islet is initiated and developed by members of Samsung Research. This support is going to continue for the foreseeable future.

2.4. Trademark status

Islet is not trademark protected.

Technical Information

3.1. High level assessment of project synergy with existing projects under the CCC, including how the project compliments/overlaps with existing projects, and potential ways to harmonize over time. Responses may be included both inline and/or in accompanying documentation.

In summary, the Islet project complements existing projects within the CCC by introducing a new CC platform for ARM architecture. Its potential synergies lie in collaborating with other projects, contributing standardized APIs, integrating with CC software stacks, and leveraging compatibility with existing software to harmonize and enhance interoperability over time. This will contribute to the overall advancement of confidential computing within the Confidential Computing Consortium community.

- As the first CC platform for the Arm CCA listed under the Confidential Computing Consortium, Islet can actively integrate with other CC software stacks and explore collaboration opportunities. By employing Islet for ARM devices as a new target, the existing CC runtime projects like Enarx can increase its compatibility and interoperability. Additionally, Islet has been collaborating on the development of CC frameworks, like the VMware Certifier Framework for Confidential Computing, to standardize on an easy-to-use, platform-independent API for creating and operating confidential computing applications. Islet is also supporting the Veraison project for platform attestation.

- Islet implements the Realm Management Monitor (RMM) Specification defined by Arm. This alignment enables Islet to run existing software, such as Linux, in both the normal world and the realm world with ARM CCA patches. By leveraging compatibility with existing software stacks, Islet harmonizes with other CCC projects, allowing developers to utilize their existing applications and tools while benefiting from the security and confidentiality features of Islet.

- While not directly under CCC projects, the tf-rmm and tf-m Runtime Security Service (RSS) projects from the trusted-firmware community contribute to the same goal. These projects, mainly contributed by ARM and written in C, focus

on general features. Islet, on the other hand, stands as an independent implementation written in Rust, catering to user device-specific use cases. This diversity in implementation language and project focus allows for a wider range of CC platforms for different purposes. Additionally, Islet and the tf-rmm project may work together for the same goal and share ideas based on the shared objective.

3.2. Describe the Trusted Computing Base (TCB) of the project. If the project supports multiple environments please describe each TCB. Also identify if dependencies of other project (both CCC or non-CCC) TCBs are taken.

- TCB of Islet-RMM
  - HW: ARMv9.2 CCA for application processors
  - Device bootloaders: provided by device OEM or the tf-a project
  - Trusted Monitor code at EL3 : provided by device OEM or the tf-a project
- TCB of Islet-HES
  - Firmware for HES
    - Device bootloaders: provided by device OEM
    - Operating System: provided by device OEM

Note that Islet is currently running on an ARM simulator, FVP, and using code from the trusted-firmware project community for device bootloaders and the monitor at EL3. As to the Islet-HES component, currently it can be compiled as a stand-alone Linux application that communicates with the firmware running on the FVP simulator.

3.3. Project Code of Conduct URL. We recommend a Contributor Covenant v2.0 based Code of Conduct, so if the Code of Conduct is not based on that, explain why.

https://github.com/Samsung/islet/blob/main/CODE_OF_CONDUCT.md

3.4. Source control URL

https://github.com/Samsung/islet

3.5. Issue tracker URL

https://github.com/Samsung/islet/issues

3.6. External dependencies (including licenses, and indicate whether each is a build time or runtime dependency)

https://github.com/Samsung/islet/blob/main/doc/dev-dependencies.md

3.7. Standards implemented by the project, if any. Include links to any such standards.

- Realm Management Monitor (RMM) Specification 1.0-bet1: implemented at Islet RMM and Islet HES. We intend to update Islet to be compatible with the most recent version 1.0-eac5 and maintain Islet's compatibility with ARM's releases.
- For Islet-HES
    - CBOR: the data serialization format
    - COSE: cryptographic envelope of the attestation token
    - SHA-256, SHA-512 specified in FIPS 180-2 and Digital Signature Standard FIPS 186-4
    - CTR mode Key Derivation Function (KDF) specified in NIST SP 800-108 Rev. 1
        - Pseudorandom Function (PRF) is based on SHA-256 (FIPS 180-2) and AES (FIPS 197)
    - a key generation by testing candidates specified in FIPS 186-3, B.4.2
        - Random Bit Generator (RBG) is based on HKDF (RFC 5869) SHA-256 (FIPS 180-2)

3.8. Release methodology and mechanics

Islet is released via GitHub (https://github.com/Samsung/islet/releases).

The major and minor versions of the software are released by the maintainers periodically. Individual patches are submitted by contributors as pull requests and require sufficient reviews and approvals of the leadership team before being merged into the mainstream version.

3.9. Names of initial committers, if different from those submitting proposal

```
Bokdeuk Jeong (bd.jeong@samsung.com)
Changho Choi (ch754.choi@samsung.com)
Heeill Wang (hihi.wang@samsung.com)
Jinbum Park (jinb.park@samsung.com)
Lukasz Pawelczyk (l.pawelczyk@samsung.com)
Michal Szaknis (m.szaknis@samsung.com)
Piotr Sawicki (p.sawicki2@samsung.com)
Sanwan Kwon (sangwan.kwon@samsung.com)
Sunwook Eom (speed.eom@samsung.com)
Zofia Abramowska (z.abramowska@samsung.com)
```

3.10. List of project's official communication channels (slack, irc, mailing lists)

Islet uses the Github channels at https://github.com/Samsung/islet/discussions to conduct project communication in public.

3.11. Project Security Response Policy

Our security policy is available at https://github.com/Samsung/islet/SECURITY.md.

3.12. Preferred maturity level (Incubation, Graduation, or Emeritus)

Incubation

3.13. Any additional information the TAC and Board should take into consideration when reviewing your proposal.

None