White Paper

The Importance of Security for the Infrastructure Edge

2019

The Importance of Security for the Infrastructure Edge

White Paper

Abstract

This paper offers important information about security considerations for the infrastructure edge. While the “IoT Edge” (also known as endpoint edge, or device edge) typically refers to sensors that collect data from the internet of things (IoT) and feed it upstream, the “infrastructure edge” is most easily thought of as the first aggregation point where any kind of collation or processing of data takes place away from the cloud — and that point is changing.

To satisfy the increasing appetite for bandwidth, providers have been forced to extend their compute capabilities from the cloud to network bridges, protocol translators, and gateways, turning them into a compute platform for data coming in from sensors and devices. By serving as compute resources themselves, these edge devices can enable new use cases that rely on data volumes that would overwhelm available bandwidth to the cloud. In this paper we outline why the infrastructure edge is important and what problems need to be solved, plus offer an introduction to the Platform Security Architecture (PSA) and how it maps to the infrastructure edge.

Contents

The Infrastructure Edge and the Market Requirements

A Trillion Connected Devices

Arm experts have predicted more than one trillion connected devices by 2035, and that translates into a lot of opportunities. Not just for companies that can benefit from the data collected by these devices, but opportunities for providers as well. However, with new technologies come new challenges. Providers must be able to solve many hard problems around the deployment and management of these trillion devices.

Key areas for the industry to address are:

- Providing semiconductor solutions that move, store, process, and secure data with speed and reliability.
- Responding to new demands for bandwidth, power efficiency, and end-to-end security.
- Analyzing data and generating meaningful insights.
- Automating the roll out of new services.
- Minimizing the cost of downtime and maintenance.

Opportunities abound and they'll only be magnified by the continuing rollout of 5G. However, there's one very important factor that must be taken into consideration first: all these new technologies must be built upon common platform security services from the endpoint to the cloud. While the industry attempts to figure out how this will happen, Arm is taking the lead. We're analyzing use cases, evaluating requirements, and leveraging the expertise of our extensive ecosystem of partners to define, design, and secure the "infrastructure edge."

A Foundation of Security

In 2017, Arm announced the Platform Security Architecture (PSA), a framework that provides a fundamental shift in the economics of IoT security, enabling ecosystems to build on a common set of ground rules to reduce the cost, time, and risk associated with IoT security. This architecture offers a route to improved security, helping partners to understand the requirements of designing, developing, and securing IoT devices at the endpoint, no matter their role. Today, our goal is to help the industry to use the PSA in the infrastructure edge market by introducing some additional elements to the program. In this white paper, we cover:

- Arm's vision for the infrastructure edge.
- Why network bridges, routers, and gateways will become a first-order compute platform.
- What foundational specifications are needed for hardware and firmware.

- What standard security services are common with IoT endpoints.
- How isolation allows multiple vendors to run platform security services in the same device.
- How the TrustedFirmware.org reference implementation open source software (OSS) project enables developers.
- How to build a secure infrastructure edge platform.

Increasing Volume of Data

IoT data volume is growing much faster than bandwidth to the cloud. Consider just one use case: in the future, we anticipate 500 million high-definition (HD) image sensors will produce 300 exabytes (10^{18} bytes) of data per month. That will exceed available bandwidth to the edge network, let alone the cloud. The same trend is occurring across the board, and backhauling is not an option. The only way to deal with this deluge of data is to process it at or near the edge. Sufficient compute horsepower at the edge allows organizations to process sensor data and send only what's critical upstream.

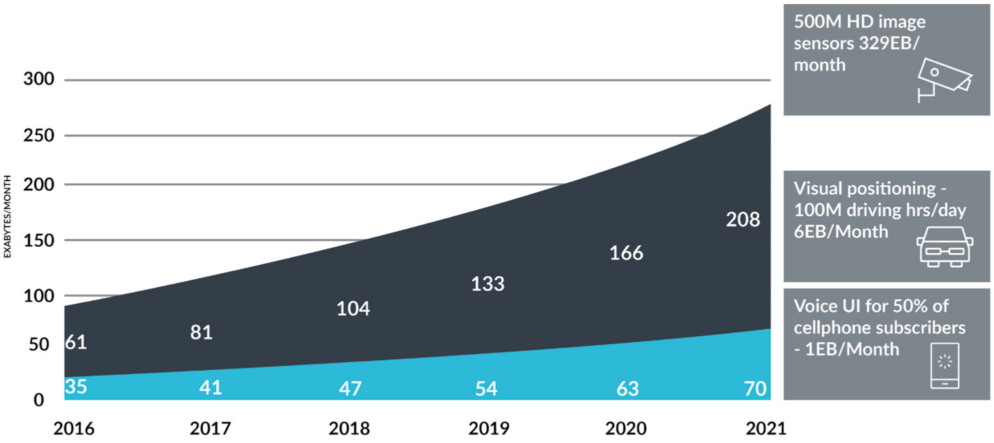


Fig 1: Actual and projected internet capacity according to the Cisco Visual Networking Index. The higher figure is the total capacity, the lower subtracts content delivery networks and carrier services, to give a proxy for cloud bandwidth. Emerging use cases will, in combination, far exceed available capacity.

Beyond bandwidth, processing at the infrastructure edge needs to emerge for a variety of reasons:

- **Data Privacy:** Whether due to the sensitive nature of the data, company policies, or in response to regulatory requirements, many organizations prefer to keep their data on premises or locally. The infrastructure edge allows for the processing and storage of that data within the constraints of such policies or regulation.

- **Autonomy:** By leveraging compute power at the edge, vendors can continue to provide service when the internet backhaul, or other points, are temporarily unavailable and data can't reach the cloud. By strengthening the resilience of the larger architecture, the impact of outages is mitigated. This is especially true and important for critical services.

- **Real-time Response:** Increases in both volume of data, and compute per datum results in additional latency. Infrastructure edge devices with sufficient processing power and storage enable a compute-ready internet edge with local analytics and autonomous decision-making. Without the need to transfer data outside of the infrastructure edge device, a real-time response is ensured.

- **Pervasive AI/ML:** Analytics are increasingly reliant on deep learning inference. Machine learning (ML), which requires higher compute loads, is replacing classical algorithms. This means more data transfer, requiring more bandwidth and resulting in more latency. AI or ML processing engines at the infrastructure edge allow data-driven decisions to be pushed back to the infrastructure edge device—in some cases with little to no cloud dependency. This improves efficiencies in bandwidth and energy and reduces latency in decision-making.

Infrastructure Edge: Key Requirements

As the world leader in supplying IP and technologies for embedded systems, Arm is well positioned to identify and delineate the key requirements of the infrastructure edge. As such, Arm is now working with its partners to develop relevant materials to assist organizations in understanding and achieving the requirements outlined below, particularly in the following areas: cloud native application deployment, software development models, and enterprise networking management technologies.

Here are six main requirements for the development of the infrastructure edge:

1. Extending IoT Threat Models

To adequately address threats, the infrastructure edge security model must start with the IoT endpoint. Examples of IoT threats and some counter-measures include:

- **Physical attacks**
  For invasive and non-invasive physical attacks, counter-measures include tamper-resistance and side-channel attacks in all forms, such as simple power analysis (SPA) and differential power analysis (DPA).

- **Communication attacks**
  Communication attacks can exploit the network or insecure communication protocols to intercept, spoof, or disrupt messages between the device and the cloud. Example counter-measures include encryption of data over the network using, for example, TLS.

- **Attacks on secure assets**
  One of the most common attacks is where an attacker accesses restricted resources, so it's important that a device is built with this threat in mind.
  To counter these threats, platforms should implement a secure Root of Trust (RoT) and RoT services (including crypto, attestation, secure storage). Restricted resources and assets (secrets) should be maintained in a hardware-isolated domain and key operations performed via secure services.

- **Lifecycle attacks**
  As a device lives for a long time after it leaves the production line, it's important to consider what could happen 10 years to 15 years later and make appropriate provisions. Lifecycle attacks could target a device left in a state with debug interfaces exposed, which is often known as 'insecure debug'.

- **Attacks on the boot process**
  Attacks on the boot process include illegitimate firmware upgrades and disrupting verified and measured boot.

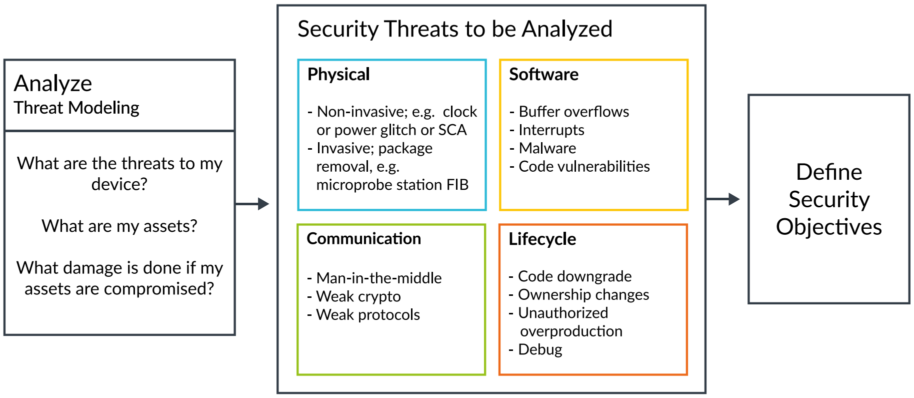
  - Verified boot: the process of loading and verifying code for its authenticity (i.e., it is from a known trusted source), and that it has not been corrupted or modified in any way.

  - Measured boot: is the process of cryptographically measuring the code and critical data, for example using a TPM, so that the security state can be attested to later.

Common IoT Threats

Security cannot be an afterthought

Fig 2: Key security threats for IoT devices, which should be reconsidered in the security model for infrastructure edge use cases



What are the threats to my device?

What are my assets?

What damage is done if my assets are compromised?

For more information on threat models for IoT devices, see the Platform Security Architecture Overview white paper.

2. Mirror the Cloud Developer Experience

Developers cannot be expected to learn new design patterns and development paradigms for a trillion devices. Therefore, programming should be as simple and familiar as possible.

Informal surveys have indicated that developers working on infrastructure edge applications prefer the cloud native DevOps model, a microservice architecture and containerized deployment of applications. Current cloud models are flexible enough to extend to the infrastructure edge.

A runtime on the device supports deployment of containerized software components that implement dynamically deployed services, rather than a monolithic software update of the entire device. Just like the cloud, the heterogeneity of edge devices and wide range of compute capabilities, functionality and constraints mean the same runtime environments must be supported at the edge, including operating system (OS) containers, virtual machines (VM) and separation kernels.

3. Support Secure Multitenancy

With multitenancy, mutually untrusted tenants or services are running on the same platform. This allows for workload consolidation and reduces deployment costs by rolling out new services as software updates rather than deploying new boxes.

An example of a multitenancy use case: original equipment manufacturers (OEMs) are currently looking at business models offering infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) that extend to the gateway. These models enable new third-party services to gain access to IoT data not available in the cloud.

To achieve multitenancy security and ensure integrity, data must be isolated and kept confidential and inaccessible by other tenants and performance must be guaranteed to ensure availability. This data includes the service itself (download, loading) in addition to data subsequently generated and maintained by the service.

These documents offer more information on security requirements for multitenancy:

- A document covering the OpenFog reference architecture is available here, Arm has been a significant contributor.
- This whitepaper covers OpenFog security requirements and approaches.

4. Manage and Orchestrate Just Like Enterprise IT

Device management must be simplified via a single console. Rather than employing multiple consoles, switch configurations, and activities to perform everyday tasks, only a single command or action should be required to deploy an entire platform. Unified end-to-end orchestration is a priority for enterprises, and this requires merging the worlds of information technology (IT) and operational technology (OT). This merge is challenging and requires significant innovation in software and services. This software will be built on fundamental RoT services for identity, remote attestation, and provisioning secrets.

5. Multi-Cloud, Public/private Architecture Deployment Orchestration

Many of our customers say it's unlikely they'll choose a single vendor for gateways, devices, cloud, and associated areas, but would prefer to implement best-of-breed technologies from multiple vendors. Therefore, platform standards are essential— particularly around security—these will enable vendors to build on top of the platform to solve the problems around management and orchestration, as well as to offer value-added services.

6. Real-time response

To achieve real-time response, the requirements found in current embedded systems must be met. Depending on the specific requirements, many approaches are possible, including a dedicated real-time CPU, hypervisor, or real-time operating system (RTOS).

Infrastructure Edge Enables Third-Party Opportunity

The architecture Arm proposes is aimed at introducing and enabling standardization around the fundamental requirements for a secure infrastructure edge. There is a significant opportunity for third-party providers all along the value chain, from silicon provider to system integrator, to build upon this secure platform. This is especially true when threat models, company policies or government regulations dictate higher levels of threat mitigation. PSA APIs from endpoint through edge to the cloud will act as the foundation for building value-added software and services.

Using the Platform Security Architecture in the Infrastructure Edge

In this second section of the white paper, we take a look at the existing Platform Security Architecture (PSA) program, which was launched for constrained IoT devices and how this can successfully map for use in the infrastructure edge.

What is the PSA?

The PSA provides a security framework that allows security to be consistently designed in, at both a hardware and firmware level. The PSA is a four-stage process, with a set of holistic deliverables to guide companies through each stage. These deliverables include a set of sample threat models and security analyses, hardware and firmware architecture specifications, and an open source firmware reference implementation.

The fourth and final stage is PSA Certified, which currently offers certification for constrained IoT devices via an independent body. This allows the ecosystem to recognize that a device is built to security standards, without worrying about the different design patterns or implementation specifics.

The PSA framework has been embraced by the IoT ecosystem as a rallying point for designed-in security, from the ground up, with top players adopting the framework, and certifying their products today. This is the essential stepping-stone to provide the backbone of secure devices for secure services. PSA is a small step to take today, for a giant leap for IoT security.

Platform Security Architecture (PSA)

The open device security framework, with independent testing

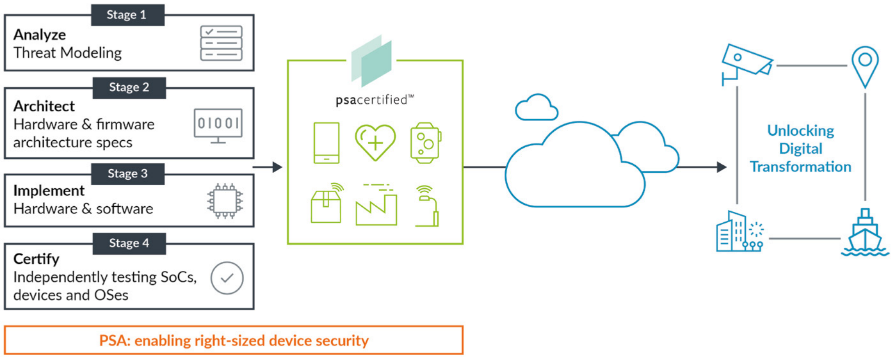

Stage 1
Analyze
Threat Modeling
Stage 2
Architect
Hardware & firmware architecture specs
01001
Stage 3
Implement
Hardware & software
Stage 4
Certify
Independently testing SoCs, devices and OSes
psacertified™
Unlocking Digital Transformation
PSA: enabling right-sized device security

Fig 3: An overview of the PSA for constrained IoT framework, the relationship to PSA Certified and how it enables digital transformation

The PSA provides a set of security goals with key security functionality that should be deployed in a system, along with setting expectations for the quality of implementation. It details how to create a PSA-RoT with a set of secure services (APIs) that the rest of the system can use. Many of the requirements detailed in the previous section boil down to security and this approach raises the bar for the entire industry, making essential security components ubiquitous.

The PSA describes a scalable implementation for building the PSA-RoT for devices. This market is still wide and diverse, so the PSA allows for different market needs and anticipates different valid design patterns including: Armv7-M or Armv8-M CPUs, independent of employing Arm TrustZone, or other security IP, including secure elements.

In the constrained IoT endpoint market, there are many security weaknesses and the PSA offers guidance on how to navigate the security risks. However, the infrastructure market is more mature and thus the PSA fits within existing practices and existing industry standards. The infrastructure edge is somewhat a merge of the two, bringing some elements of infrastructure and IoT, which means the PSA can help bridge the gap by providing a common architecture model and common set of secure services.

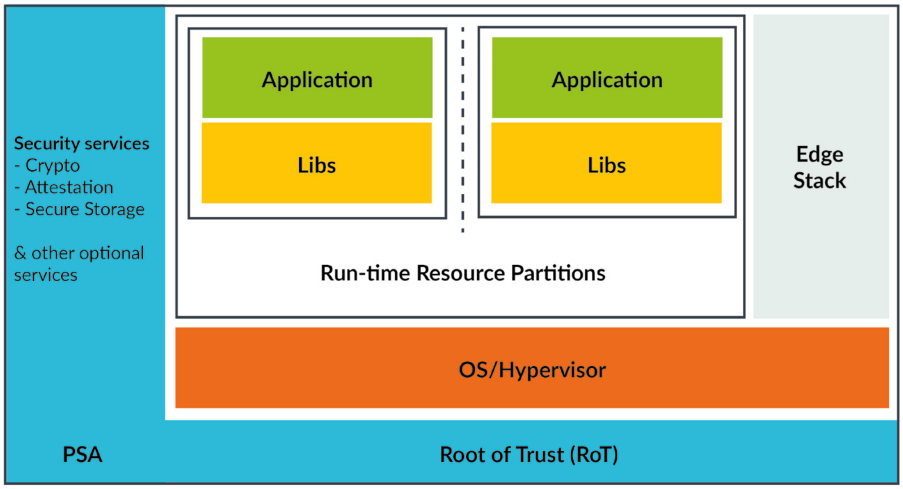
Mapping the PSA to the Infrastructure Edge

As described above, the current PSA framework targets constrained IoT devices, but is now evolving to include rich IoT devices, which is in turn extensible to the infrastructure edge. The PSA for the IoT includes a set of foundational specifications that describe how to build a RoT in the Arm architecture and an open source reference implementation. We expect that PSA for the infrastructure edge will similarly comprise foundational specifications and an open source reference implementation of the secure world platform firmware.

PSA for Constrained IoT Stages and Deliverables	Planned Mapping to PSA for the Infrastructure Edge	Arm Architecture Specific vs. Generic
Analyze Threat models.	We believe that the infrastructure edge market is not mature and so new threat models and security analysis (TMSA) documents are needed. We plan to deliver an example TMSA-Edge document to help the ecosystem and the broader industry make reasonable judgements about requirements.	Generic.
Architect Architecture specifications offer explanations on how to create hardware and software which comply with the PSA security model goals.	**Trusted Boot and Firmware Update (TBFU):** this specification is generic and is applicable to infrastructure, as well as constrained IoT devices. The existing **Trusted Base System Architecture (TBSA) specification** will be modified to provide specific guidance for the infrastructure edge. This document will be called TBSA-Edge. **Firmware Framework-A (FF-A)**, previously named Secure Partition Client Interface (SPCI), will be available in the near future. This document defines common secure-world isolation across all A-profile markets. More information is also available in our white paper: Isolation Using Virtualization in the Secure World. **Server Base Security Guide (SBSG):** The SBSG describes requirements and gives implementation guidance specific to infrastructure, referencing the relevant industry standards for server. The SBSG maps market specific requirements to the other PSA specifications.	The PSA APIs are generic.

	The SBSG is available now and covers: + Protection of firmware and critical data including secure update, detecting corruption. + TPM 2.0 integration, including measured boot. + UEFI security features, which are one of the boot loaders of choice for the infrastructure edge. PSA APIs: are the same set we use in PSA for constrained IoT, as there is some ongoing work to make sure they are generic. There will also be other APIs in use specifically for the infrastructure market.	
Implement Open source trusted firmware.	Trusted Firmware-A (TF-A) is already mature and supports the A-profile CPUs. It will implement support for TPM 2.0 and measured boot. We are collaboratively developing additional projects such as a security microservice, named 'PARSEC' that will provide additional scope outside of TF-A.	TF-A is for the Arm A-profile architecture. PARSEC is generic and supports RoT implementations not based on TF-A.
Certify Independent assurance scheme.	We are still exploring if a certification scheme is needed in the infrastructure edge market. We think that an objective assessment of a device RoT could be beneficial to providers of management services, who need to load their credentials onto a device.	The existing PSA certification scheme is independent and therefore tests against generic requirements.

Fig 4: An example software stack for the infrastructure edge, comprising an OS, edge stack and container runtime hosting multiple applications. The software stack builds upon a root of trust and security services.



More Details on Core PSA Specifications

PSA contains components—such as threat models, APIs and a certification scheme—that are generic and independent of the Arm architecture. As we extend the generic components to comprehend the requirements of the infrastructure edge, we are creating new specifications that describe ways to meet these requirements using tools in the Arm architecture.

These Arm specific specifications include the Server Base Security Guide (SBSG) and core PSA specifications for the Arm A-Profile architecture. The core PSA specifications describe a way to build the RoT and services based on the Arm architecture. We expect these core specifications (listed below) to be common across all markets that use the A-Profile architecture and to be developed from both the current PSA for M-Profile and Client (handset) architectures.

One benefit to leveraging this investment across the entire ecosystem is improving the balance between cost and risk level. Since security is a trade-off between these two factors, the more the cost is amortized via standardization and common reference software, the better the standard of security against a given investment.

1. The Firmware Framework (FF-A)

The Firmware Framework (FF-A) describes how to build isolation on the platform, to isolate security services from the rich OS, and to isolate one security service from another. This allows multiple different security services to run in the system in a secure environment. For example, a silicon vendor might have their own services for attestation and secure boot, while the OEM might provide their own services to generate session keys for a TLS connection.

For more information, download our white paper: Isolation Using Virtualization in the Secure World.

The benefit of hosting runtime security services in TrustZone is that they are isolated from the rich software stack, so if the rich software stack is compromised then key assets are protected. The benefit of using a secure partition is that OEM or third-party software provider platform services can be added to provide security services beyond what is defined in PSA. When TrustZone virtualization (Secure EL2) is available in future silicon, secure partitions will provide a migration path.

2. The Trusted Base System Architecture (TBSA)

The Trusted Base System Architecture (TBSA) provides requirements for the underlying hardware to support the secure world software stack described in FF-A.

3. The Trusted Boot Firmware Update (TBFU)

The Trusted Boot Firmware Update (TBFU) provides requirements and outlines techniques for verified boot and update using certificates.

4. The PSA APIs

The PSA APIs provide standard security services for cryptography, attestation and secure storage. Our goal is to support the exact same APIs across all markets, with a wide range of possible implementation types. We expect implementations to support other standard APIs such as TPM 2.0 and PKCS #11. Providing the same PSA APIs/services on infrastructure platforms and IoT endpoints provides familiarity for developers who are also developing software for endpoints.

Open Source Reference Implementation

TrustedFirmware.org will provide a reference implementation of the entire secure world software architecture, as described by the PSA specifications above: SBSG, FF-A, TBFU. The currently available Trusted Firmware for A-Profile (TF-A) reference software supports verified boot up to the normal world boot loader. The boot loader then continues building the chain of trust, for example U-Boot verified boot or UEFI secure boot. Our plan is that TF-A will provide a complete reference implementation of the SBSG capabilities, including measured boot based on Trusted Platform Module 2.0 (TPM 2.0).

The reference implementation will include a Trusted OS (OP-TEE) and related Trusted Applications (TA) implementing the PSA security services, and other reference services.

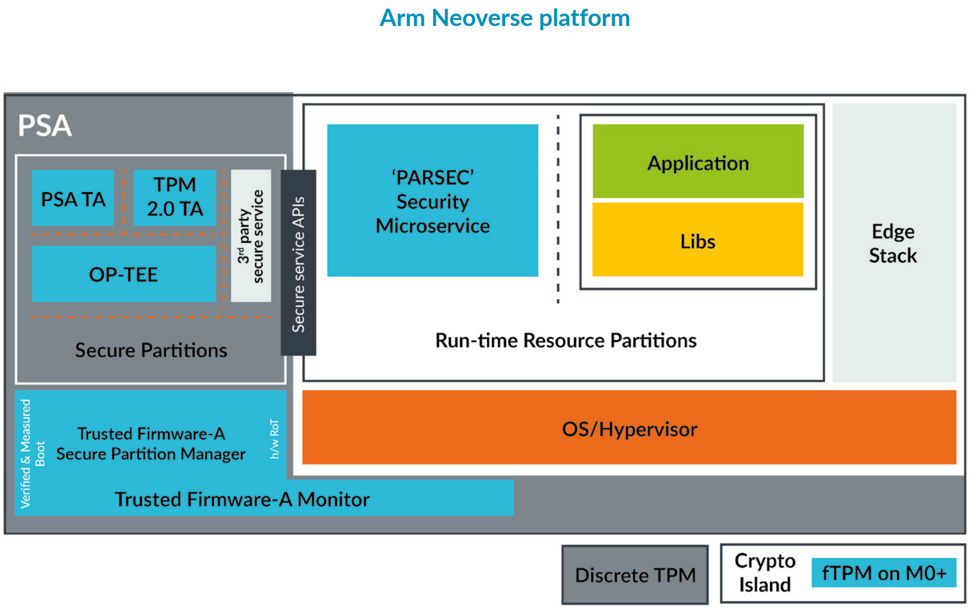


Fig 5: Planned PSA Infrastructure reference software, including a complete secure world software stack, reference implementation of security services, and a microservice that surfaces them to applications and runtimes.

The figure above shows the reference software stack that we are planning for infrastructure PSA, targeting infrastructure edge OS distributions.

A secure partition manager in the secure world enables multiple reference security services:

- A PSA Trusted Application that implements the PSA APIs for UEFI, TF-A or other runtime services to use.
- A trusted application that implements the TPM 2.0 APIs and services. The trusted application will support:
  - A discrete TPM (dTPM) device.
  - A firmware TPM (fTPM) implemented entirely in a TA in Arm TrustZone.
  - A TPM implemented in a secure element (SE), with a reference implementation of a firmware TPM in Arm CryptoIsland.

PARSEC

PARSEC is a reference security microservice running within the OS distribution, which surfaces the PSA APIs using an IPC mechanism. The purpose of this microservice is to provide both abstraction and arbitration for applications and runtime software.

Abstraction in the microservice means that the same PSA APIs are always available across diverse RoT service implementations. Abstraction is important because target devices have different RoT implementations to meet cost and security requirements. For example, a secure element, a trusted execution environment such as Arm TrustZone, or a combined approach. Abstraction also supports a typical development/deployment lifecycle where different implementations of secure services are used for development, testing, and end deployment.

Arbitration by the microservice enables multiple applications/runtimes to use the security capabilities of the device, while ensuring that the secrets for each application are isolated.

The benefit of this reference software for silicon partners is to shorten the time-to-market and to reduce the cost of building a system with security compliant to industry standards referenced in the SBSG.

The benefit for developers is to have a common baseline of security services that span constrained IoT endpoints and infrastructure, with multiple implementation types that suit a range of device costs and security profiles.

All of this reference software is being developed collaboratively and in the open, using permissive open source licenses (BSD and Apache v2) and we welcome contribution.

For more information on PARSEC, visit the github.

In Conclusion

As detailed at the start of this paper, there is a significant opportunity with the impending mass deployment of IoT and the infrastructure edge. Therefore, it is important that providers actively plan now for the everything-connected world, with its rapidly increasing attack surface and global deployment. Of course, there are security concerns and challenges, as is the case with most technologies today.

Arm and our partner ecosystem are closely monitoring this space to provide relevant and informed guidance. We will provide reference implementations and guidance to shorten the development cycle for partners, allow them to innovate where needed, and provide compatibility going forward

To be clear, it is not our intent to dictate approaches, methodologies, or solutions. Arm is simply recommending guidelines on how to navigate this evolving, nascent industry. Our goal is to remain aware of new developments, stay flexible in our perspective, and be responsive to changes as they occur.

Feedback from our partners is critical. We welcome responses and seek feedback on the information we're providing. We look forward to receiving thoughts and ideas on the infrastructure edge and how it can best be secured and leveraged.

Please email project-cassini@arm.com to share your feedback with us.

Useful Resources and Links

Trusted Computing Group specifications

PARSEC GitHub

OpenFog reference architecture

OpenFog security requirements and approaches whitepaper

The Platform Security Architecture Overview white paper

All brand names or product names are the property of their respective holders. Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given in good faith. All warranties implied or expressed, including but not limited to implied warranties of satisfactory quality or fitness for purpose are excluded. This document is intended only to provide information to the reader about the product. To the extent permitted by local laws Arm shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information.