Executive Summary

When it comes to developing embedded edge systems for automotive, industrial, autonomous, and safety-critical applications, the present way of working, which has developers dealing directly with many of the underlying hardware and software components, slows productivity and diverts attention away from innovation and differentiation. To help give developers a better way to build for the edge, Arm has established Project Cassini and is leveraging the programs under that initiative to develop a cloud-native software architecture that supports software-defined operations at the edge.

The Increasing Importance of Software

A growing number of today’s most sophisticated embedded edge systems rely on millions of lines of code for operation. Spanning a broad range of use cases in automotive, industrial, autonomous, and safety-critical applications, these advanced edge systems involve a wide variety of frequently-changing elements, diverse hardware devices, and rely on software that is increasingly intricate and expensive to maintain. This high degree of complexity makes development that much more difficult and expensive. This situation becomes even more complicated when mixed criticality workloads are involved where the functional safety requirements need to be handled alongside the traditional workloads and the current software infrastructure and models does not support the workload portability.

Delivering new products can become more a question of integration than innovation, because instead of focusing on new, differentiating features, developers are forced to spend time dealing with variations, changes, and updates in underlying components.

Incompatibilities, due to differences in hardware architectures, low-level firmware, and boot sequences, can complicate development, making it harder to verify that the various parts of the system work together and meet safety requirements. These incompatibilities also complicate things for commercial software vendors, because high-level functions and workloads don’t port easily to different operating systems. Consequently, developers can find themselves spending time altering a third-party solution before it runs smoothly in their target systems.
The lack of standardization also creates issues when trying to re-use software in new systems. Upgrading to the latest version of silicon or adding a new chipset can involve so many software changes that, despite the effort that went into creating the original code, it can be easier to simply discard what’s there and start again.

These issues are exposing limitations in the traditional ways of working, which were established before software became so dominant. The process has to evolve and adopt a more modern, software-driven approach. Developers of complex embedded systems need a new way to build things, so they can re-use more of their code and simplify the process of developing, testing, and deploying application software.

Software re-use lowers TCO and makes developers more effective.

DevOps to the Rescue

Fortunately, there is a ready-made model that embedded systems can use as a guide. The DevOps approach, closely associated with cloud native development practices, has been shown to speed the delivery of high-quality software. It’s time to start putting DevOps to work in the embedded-edge domain.

At Arm, our ongoing work in data centers and other cloud-based operations has shown us just how valuable the DevOps approach can be, especially when it comes to optimizing stacks and lowering the total cost of ownership for applications. We see the long-term need for DevOps in other areas of computing, especially at the edge of the network, and are working on a number of fronts to apply DevOps approaches to embedded edge systems.

Enabling DevOps with Project Cassini

One example of how we’re bringing DevOps to the embedded edge is Project Cassini, a comprehensive industry initiative that is creating a foundation supporting DevOps. Project Cassini will make deploying software onto edge devices easier because hardware and firmware will use standards that enable secure, consistent operation and the ability to run workloads without modification. Project Cassini will also save time, because developers can use portable, commodity software that masks underlying complexity.
Project Cassini uses standards, security, and ecosystem to create a cloud-native experience for edge.

Project Cassini is built on three foundational components: robust standards, security APIs and certification programs, and reference solutions that make DevOps techniques a part of edge development.

**Standards: Arm SystemReady**
The Arm SystemReady program is a foundational certification program that defines a minimum set of hardware and firmware requirements so off-the-shelf operating systems, community software distributions (“distros”), and hypervisors, as well as other workloads, run seamlessly on diverse Arm platforms. For hardware, SystemReady defines a common Base System Architecture (BSA) and a set of market-specific supplements. For firmware, the Base Boot Requirements (BBR) describes standards-based boot recipes and implementations.

**Security: PSA Certified and PARSEC**
To address the need for protection against unauthorized access and data manipulation, two initiatives ensure a baseline of security for all connected devices. The PSA Certified initiative offers a security framework and certification scheme that gives an objective assessment of the quality of the implementation for the device root of trust, thereby establishing a foundation of trust on which to build. The open-source Platform Abstraction for Security (PARSEC) initiative provides secure root-of-trust abstraction and common runtime security services for applications in a platform and architecture-agnostic manner, for application-wide security.

**Ecosystem: Cloud-native stacks**
Given the consistency in hardware, firmware, and security provided by the other foundational components of Project Cassini, it becomes possible to apply modern software development and deployment approaches associated with DevOps to the edge. For example, Project Cassini makes it easier for developers to use cloud-native techniques, such as service-oriented architectures, container orchestration, and Continuous Integration and Continuous Delivery (CI/CD) pipelines, to streamline development and accelerate deployment. Project Cassini also makes it easier for software vendors to offer portable solutions that are easy to integrate into the larger system and run without modification on every device.
Service-Oriented Architecture (SOA) for Embedded Edge

With Project Cassini in place, it becomes possible to apply various DevOps approaches to embedded edge development. In particular, developers can begin using the DevOps design methodology known as a Service-Oriented Architecture (SOA).

SOA is a particularly good fit for the kinds of complex embedded edge systems being developed today. That’s because SOA turns workloads into functional building blocks that can be developed, deployed, and maintained without forcing developers to keep track of low level details. The SOA does this by using abstraction and virtualization. Abstraction hides background details, making it easier to use underlying hardware resources. Virtualization divides resources, so many tasks can run simultaneously on a single physical machine. Virtualization also makes it possible to use containers, which isolate and protect individual workloads. Container orchestration, which is used to manage container runtimes, ensures applications run anywhere consistently, on any infrastructure.

With SOA, developers can focus on differentiation

Standardization is another key element in the SOA methodology, because it provides the consistency when moving through the product stages of development, deployment, and field maintenance.

- Development
  SOA standards enable uniform development practices, such as consistent tooling, industry best practices, and a commitment to Continuous Improvement and Continuous Delivery (CI/CD), helps to improve quality, makes it easier to manage the work of many developers, and gives development teams and companies more ways to collaborate and share code.

- Deployment
  SOA standards make the process of deploying, updating, and validating workloads more repeatable, with greater consistency in the way workloads are deployed to targets, the methods used to check workload integrity, and how system resources are accessed.

- Field Maintenance
  SOA standards enable a consistent way to launch workloads, establish runtime requirements, and manage system resources, so it’s easier to manage and monitor large complex systems comprised of SOA workloads for as long as they’re in use.
The Arm SOA Platform Builds on Project Cassini

Starting with the hardware, firmware, and security standards defined by Project Cassini, we’re creating an SOA development platform that adds support for real-time and safety-critical operations and is tailored for use with embedded edge systems. Our SOA platform is designed to let developers leverage abstraction, virtualization, and standards to move more quickly, evolving and improving products faster than before.

The block diagrams gives a high-level view of how this new SOA platform will operate.

### SOA Platform

- **Project Cassini layer**
  Running on top of Arm partner silicon, Project Cassini establishes a standards-based foundation for secure operation in a diverse edge ecosystem. SystemReady compliance ensures consistent deployment in different products, PSA Certified establishes the root of trust at the hardware level, and PARSEC provides secure root-of-trust abstraction to applications across platforms. The SOA development platform supports use of a Trusted Execution Environment (TEE) in the security domain, to isolate and protect sensitive operations.

- **Hypervisor layer**
  On top of the Project Cassini firmware, the SOA architecture has a hypervisor that creates and runs virtual machines, so multiple operating systems can use the system’s hardware concurrently. This makes it easier to configure low-level functions, such as Memory Partitioning and Monitoring (MPAM) and Generic Interrupt Control (GIC) and further simplifies the configuration of the architectural features of the SoC for mixed criticality workload deployment. The hypervisor also supports Partially Virtualized (PV) interfaces, which simplify access to accelerators and enable binary-compatible workloads. The PV interfaces let applications that were created on earlier releases or technology levels run unchanged and without recompilation.
Board Support Package (BSP)
The items traditionally associated with the BSP for a given System on Chip (SoC) – that is, the hardware specific drivers and operating systems – are associated with VirtIO, an abstraction layer above the hypervisor. Using a virtualized version of the BSP, delivered by the silicon provider, lets developers work with specific hardware architectures without having to keep track of changes to the SoC or its kernel.

Containers and virtual machines
The SOA uses orchestration, container runtimes, and VirtIO enabled virtual machines (VMs) to let individual workloads run consistently on any infrastructure. The workloads take the form of portable containers, which can be developed in-house or by a third party. This gives developers added flexibility, but also makes it easier for commercial software providers to offer useful workloads that can be deployed in any number of systems, regardless of their hardware configuration. The developer may still need to fine-tune a virtualized workload for optimum performance in their target system, but starting with a baseline function that’s essentially ready to deploy saves a considerable amount of up-front time and effort.

Mixed criticality and safety islands
With each workload running in its own container under the hypervisor control, it’s possible to create a mixed-criticality system that supports different risk classifications as defined by Automotive Safety Integrity Levels (ASILs). In a passenger car, for example, a relatively low-risk workload with an ASIL-B rating, such as backup lights, can independently run alongside a high-risk workload with an ASIL-D rating, such as antilock brakes. The two workloads can be created, modified, and evolved separately, yet run concurrently in the same system. Also, that same ASIL-B workload can be treated as a standalone item and quickly deployed to other vehicle types, such as an unmanned delivery vehicle due to its implementation being abstracted from the underlying system.

The SOA platform also supports the use of a safety island, which uses dedicated compute, memory and I/O resources to manage critical functions and ensure the system continues to operate safely under all possible circumstances. The communication and monitoring functions of the safety island will be abstracted to the rest of the system through a standard interface so that workloads that need to access to the safety island can do so in a consistent way.
Industry-Wide Collaboration

To develop the SOA platform, Arm is working across industries to address the needs of all stakeholders and ensure interoperability. For example, we’re collaborating with Arm partners to define solutions for specific segment, so product development and deployment can move faster. We’re also working with upstream open-source projects to enable compliance with solution standards, thereby minimizing out-of-tree patches and other inconsistencies and simplifying product prototyping. At the same time, we’re working with commercial software vendors to enable integration of their go-to-market components, so developers have a faster way to transition from development to production.

Tell Us What You Think

We’re proud to count some for the most influential leaders in technology among our supporters. Feedback from our partners in critical. We welcome responses and seek commentary on the information we’re providing. We look forward to receiving thoughts and ideas on the edge and how it can best be secured and leveraged. To share your feedback with us, we invite you to email project-cassini@arm.com.

Arm is dedicated to collaboration. We welcome your insights on Project Cassini.

Take The Next Step

To learn more about Project Cassini and how it relates to SOA for embedded edge, visit www.arm.com/project-cassini.