What is Azure Sphere?

Article • 05/09/2022 • 12 minutes to read • 7 contributors

In this article

Azure Sphere is a secured, high-level application platform with built-in communication and security features for internet-connected devices. It comprises a secured, connected, crossover microcontroller unit (MCU), a custom high-level Linux-based operating system (OS), and a cloud-based security service that provides continuous, renewable security.

The Azure Sphere MCU integrates real-time processing capabilities with the ability to run a high-level operating system. An Azure Sphere MCU, along with its operating system and application platform, enables the creation of secured, internet-connected devices that can be updated, controlled, monitored, and maintained remotely. A connected device that includes an Azure Sphere MCU, either alongside or in place of an existing MCU(s), provides enhanced security, productivity, and opportunity. For example:

- A secured application environment, authenticated connections, and opt-in use of peripherals minimizes security risks due to spoofing, rogue software, or denial-of-service attacks, among others.
- Software updates can be automatically deployed from the cloud to any connected device to fix problems, provide new functionality, or counter emerging methods of attack, thus enhancing the productivity of support personnel.
- Product usage data can be reported to the cloud over a secured connection to help in diagnosing problems and designing new products, thus increasing the opportunity for product service, positive customer interactions, and future development.

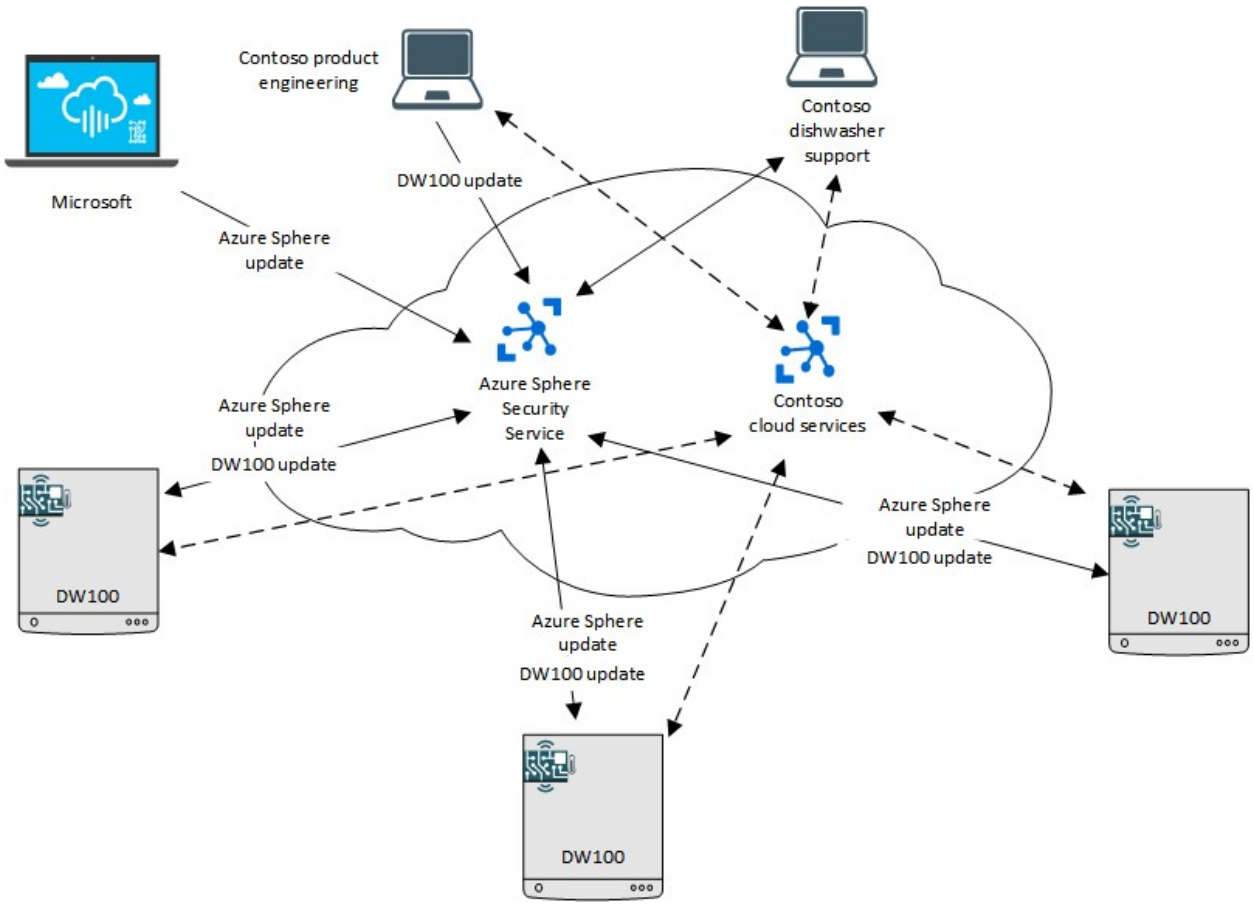
The Azure Sphere Security Service is an integral aspect of Azure Sphere. Using this service, Azure Sphere MCUs safely and securely connect to the cloud and web. The

service ensures that the device boots only with an authorized version of genuine, approved software. In addition, it provides a secured channel through which Microsoft can automatically download and install OS updates to deployed devices in the field to mitigate security problems. Neither manufacturer nor end-user intervention is required, thus closing a common security hole.

Azure Sphere scenario

To understand how Azure Sphere works in a real-world setting, consider this scenario.

Contoso, Ltd., is a white-goods product manufacturer who embeds an Azure Sphere MCU into its dishwashers. The DW100 dishwasher couples the MCU with several sensors and an onboard high-level application that runs on the Azure Sphere MCU. The application communicates with the Azure Sphere Security Service and with Contoso's cloud services. The following diagram illustrates this scenario:



Contoso network-connected dishwashers

Starting from the top left and moving clockwise:

- Microsoft releases updates for the Azure Sphere OS through the Azure Sphere Security Service.
- Contoso product engineering releases updates to its DW100 application through the Azure Sphere Security Service.
- The Azure Sphere Security Service securely deploys the updated OS and the Contoso DW100 application software to the dishwashers at end-user locations.
- Contoso dishwasher support communicates with the Azure Sphere Security Service to determine which version of the Azure Sphere software and the DW100 application software should be running on each end-user device and to glean any error-reporting data that has been reported to the service. Contoso dishwasher support also communicates with the Contoso cloud service for additional information.
- Contoso cloud services support applications for troubleshooting, data analysis, and customer interaction. Contoso's cloud services may be hosted by Microsoft Azure, by another vendor's cloud service, or by Contoso's own cloud.
- Contoso DW100 models at end-user locations download updated OS and application software over their connection to the Azure Sphere Security Service. They can also communicate with Contoso's cloud service application to report additional data.

For example, sensors on the dishwasher might monitor water temperature, drying temperature, and rinse agent level and upload this data to Contoso's cloud services, where a cloud service application analyzes it for potential problems. If the drying temperature seems unusually hot or cool—which might indicate a failing part—Contoso runs diagnostics remotely and notifies the customer that repairs are needed. If the dishwasher is under warranty, the cloud service application might also ensure that the customer's local repair shop has the replacement part, thus reducing maintenance visits and inventory requirements. Similarly, if the rinse agent is low, the dishwasher might signal the customer to purchase more rinse agent directly from the manufacturer.

All communications take place over secured, authenticated connections. Contoso support and engineering personnel can visualize data by using the Azure Sphere Security Service, Microsoft Azure features, or a Contoso-specific cloud service application. Contoso might also provide customer-facing web and mobile applications, with which dishwasher owners can request service, monitor dishwasher resource usage,

or otherwise interact with the company.

Using Azure Sphere deployment tools, Contoso targets each application software update to the appropriate dishwasher model, and the Azure Sphere Security Service distributes the software updates to the correct devices. Only signed and verified software updates can be installed on the dishwashers.

Azure Sphere and the seven properties of highly secured devices

A primary goal of the Azure Sphere platform is to provide high-value security at a low cost, so that price-sensitive, microcontroller-powered devices can safely and reliably connect to the internet. As network-connected toys, appliances, and other consumer devices become commonplace, security is of utmost importance. Not only must the device hardware itself be secured, its software and its cloud connections must also be secured. A security lapse anywhere in the operating environment threatens the entire product and, potentially, anything or anyone nearby.

Based on Microsoft's decades of experience with internet security, the Azure Sphere team has identified seven properties of highly secured devices . The Azure Sphere platform is designed around these seven properties:

Hardware-based root of trust. A hardware-based root of trust ensures that the device and its identity cannot be separated, thus preventing device forgery or spoofing. Every Azure Sphere MCU is identified by an unforgeable cryptographic key that is generated and protected by the Microsoft-designed Pluton security subsystem hardware. This ensures a tamper-resistant, secured hardware root of trust from factory to end user.

Defense in depth. Defense in depth provides for multiple layers of security and thus multiple mitigations against each threat. Each layer of software in the Azure Sphere platform verifies that the layer above it is secured.

Small trusted computing base. Most of the device's software remains outside the trusted computing base, thus reducing the surface area for attacks. Only the secured Security Monitor, Pluton runtime, and Pluton subsystem—all of which Microsoft provides—run on the trusted computing base.

Dynamic compartments. Dynamic compartments limit the reach of any single error.

Azure Sphere MCUs contain silicon counter-measures, including hardware firewalls, to prevent a security breach in one component from propagating to other components. A constrained, "sandboxed" runtime environment prevents applications from corrupting secured code or data.

Password-less authentication. The use of signed certificates, validated by an unforgeable cryptographic key, provides much stronger authentication than passwords. The Azure Sphere platform requires every software element to be signed. Device-to-cloud and cloud-to-device communications require further authentication, which is achieved with certificates.

Error reporting. Errors in device software or hardware are typical in emerging security attacks; errors that result in device failure constitute a denial-of-service attack. Device-to-cloud communication provides early warning of potential errors. Azure Sphere devices can automatically report operational data and errors to a cloud-based analysis system, and updates and servicing can be performed remotely.

Renewable security. The device software is automatically updated to correct known vulnerabilities or security breaches, requiring no intervention from the product manufacturer or the end user. The Azure Sphere Security Service updates the Azure Sphere OS and your applications automatically.

Azure Sphere architecture

Working together, the Azure Sphere hardware, software, and Security Service enable unique, integrated approaches to device maintenance, control, and security.

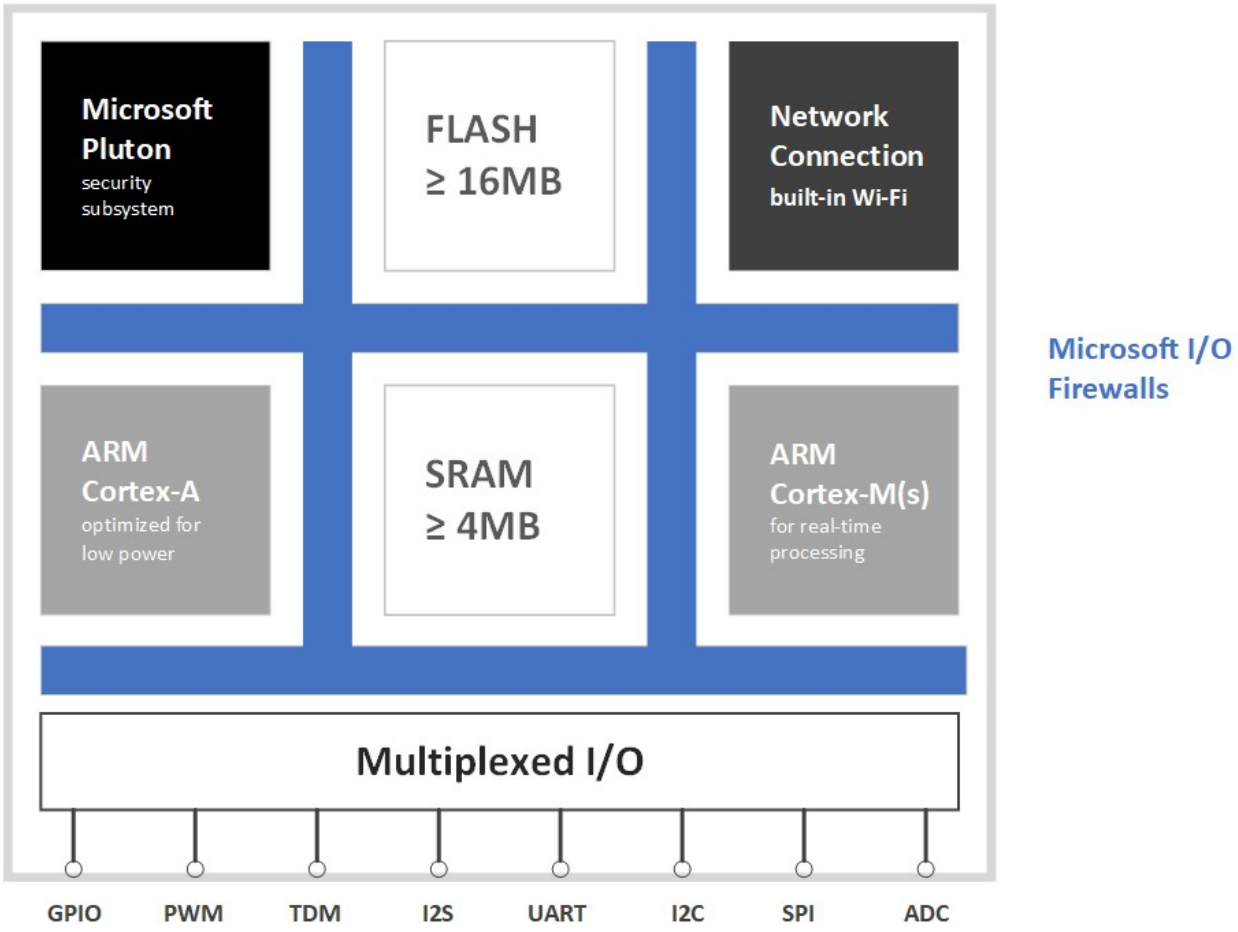
The hardware architecture provides a fundamentally secured computing base for connected devices, allowing you to focus on your product.

The software architecture, with a secured custom OS kernel running atop the Microsoft-written Security Monitor, similarly enables you to concentrate your software efforts on value-added IoT and device-specific features.

The Azure Sphere Security Service supports authentication, software updates, and error reporting over secured cloud-to-device and device-to-cloud channels. The result is a secured communications infrastructure that ensures that your products are running the most up-to-date Azure Sphere OS. For architecture diagrams and examples of cloud architectures, see Browse Azure Architectures.

Hardware architecture

An Azure Sphere crossover MCU consists of multiple cores on a single die, as the following figure shows.



Azure Sphere MCU hardware architecture

Each core, and its associated subsystem, is in a different trust domain. The root of trust resides in the Pluton security subsystem. Each layer of the architecture assumes that the layer above it may be compromised. Within each layer, resource isolation and dynamic compartments provide added security.

Microsoft Pluton security subsystem

The Pluton security subsystem is the hardware-based (in silicon) secured root of trust for Azure Sphere. It includes a security processor core, cryptographic engines, a hardware random number generator, public/private key generation, asymmetric and symmetric encryption, support for elliptic curve digital signature algorithm (ECDSA) verification for secured boot, and measured boot in silicon to support remote attestation with a cloud

service, as well as various tampering counter-measures including an entropy detection unit.

As part of the secured boot process, the Pluton subsystem boots various software components. It also provides runtime services, processes requests from other components of the device, and manages critical components for other parts of the device.

High-level application core

The high-level application core features an ARM Cortex-A subsystem that has a full memory management unit (MMU). It enables hardware-based compartmentalization of processes by using trust zone functionality and is responsible for running the operating system, high-level applications, and services. It supports two operating environments: Normal World (NW), which runs code in both user mode and supervisor mode, and Secure World (SW), which runs only the Microsoft-supplied Security Monitor. Your high-level applications run in NW user mode.

Real-time core(s)

The real-time core(s) feature an ARM Cortex-M I/O subsystem that can run real-time capable applications as either bare-metal code or a real-time operating system (RTOS). Such applications can map peripherals and communicate with high-level applications but cannot access the internet directly.

Connectivity and communications

The first Azure Sphere MCU provides an 802.11 b/g/n Wi-Fi radio that operates at both 2.4GHz and 5GHz. High-level applications can configure, use, and query the wireless communications subsystem, but they cannot program it directly. In addition to or instead of using Wi-Fi, Azure Sphere devices that are properly equipped can communicate on an Ethernet network.

Multiplexed I/O

The Azure Sphere platform supports a variety of I/O capabilities, so that you can configure embedded devices to suit your market and product requirements. I/O

peripherals can be mapped to either the high-level application core or to a real-time core.

Microsoft firewalls

Hardware firewalls are silicon countermeasures that provide "sandbox" protection to ensure that I/O peripherals are accessible only to the core to which they are mapped. The firewalls impose compartmentalization, thus preventing a security threat that is localized in the high-level application core from affecting the real-time cores' access to their peripherals.

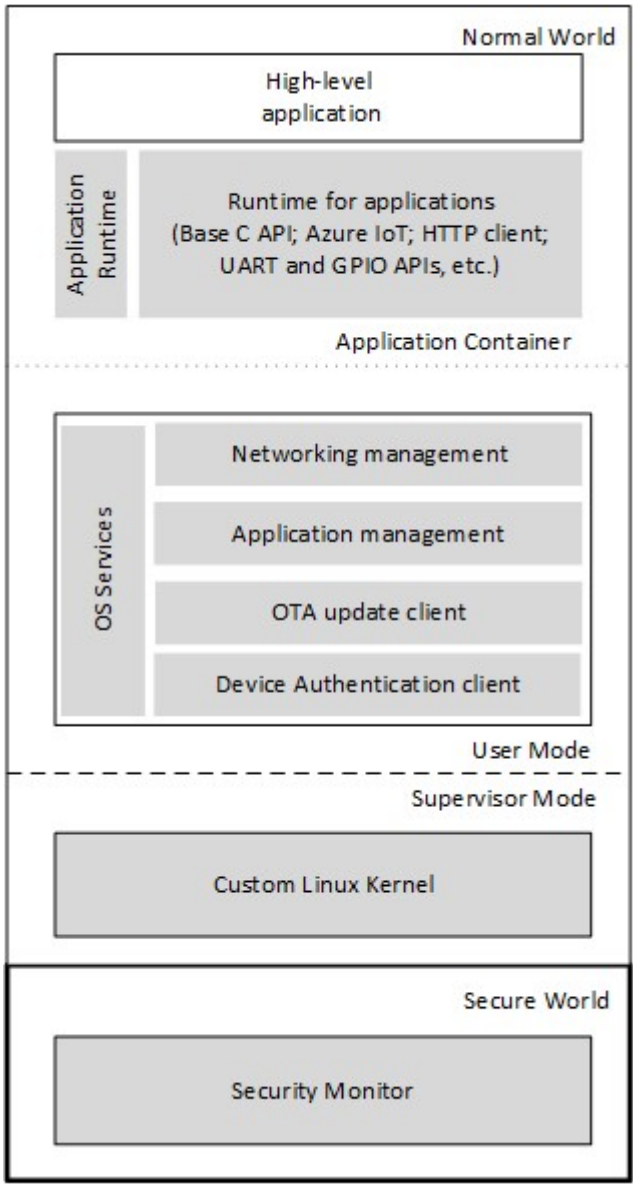
Integrated RAM and flash

Azure Sphere MCUs include a minimum of 4MB of integrated RAM and 16MB of integrated flash memory.

Software architecture and OS

The high-level application platform runs the Azure Sphere OS along with a device-specific high-level application that can communicate both with the internet and with real-time capable applications that run on the real-time cores. The following figure shows the elements of this platform.

Microsoft-supplied elements are shown in gray.



High-level Application Platform

Microsoft provides and maintains all software other than your device-specific applications. All software that runs on the device, including the high-level application, is signed by the Microsoft certificate authority (CA). Application updates are delivered through the trusted Microsoft pipeline, and the compatibility of each update with the Azure Sphere device hardware is verified before installation.

Application runtime

The Microsoft-provided application runtime is based on a subset of the POSIX standard. It consists of libraries and runtime services that run in NW user mode. This environment supports the high-level applications that you create.

Application libraries support networking, storage, and communications features that are

required by high-level applications but do not support direct generic file I/O or shell access, among other constraints. These restrictions ensure that the platform remains secured and that Microsoft can provide security and maintenance updates. In addition, the constrained libraries provide a long-term stable API surface so that system software can be updated to enhance security while retaining binary compatibility for applications.

OS services

OS services host the high-level application container and are responsible for communicating with the Azure Sphere Security Service. They manage network authentication and the network firewall for all outbound traffic. During development, OS services also communicate with a connected PC and the application that is being debugged.

Custom Linux kernel

The custom Linux-based kernel runs in supervisor mode, along with a boot loader. The kernel is carefully tuned for the flash and RAM footprint of the Azure Sphere MCU. It provides a surface for preemptable execution of user-space processes in separate virtual address spaces. The driver model exposes MCU peripherals to OS services and applications. Azure Sphere drivers include Wi-Fi (which includes a TCP/IP networking stack), UART, SPI, I2C, and GPIO, among others.

Security Monitor

The Microsoft-supplied Security Monitor runs in SW. It is responsible for protecting security-sensitive hardware, such as memory, flash, and other shared MCU resources and for safely exposing limited access to these resources. The Security Monitor brokers and gates access to the Pluton Security Subsystem and the hardware root of trust and acts as a watchdog for the NW environment. It starts the boot loader, exposes runtime services to NW, and manages hardware firewalls and other silicon components that are not accessible to NW.

Azure Sphere Security Service

The Azure Sphere Security Service comprises three components: password-less authentication, update, and error reporting.

- **Password-less authentication.** The authentication component provides remote attestation and password-less authentication. The remote attestation service connects via a challenge-response protocol that uses the measured boot feature on the Pluton subsystem. It verifies not merely that the device booted with the correct software, but with the correct version of that software.

  After attestation succeeds, the authentication service takes over. The authentication service communicates over a secured TLS connection and issues a certificate that the device can present to a web service, such as Microsoft Azure or a company's private cloud. The web service validates the certificate chain, thus verifying that the device is genuine, that its software is up to date, and that Microsoft is its source. The device can then connect safely and securely with the online service.

- **Update.** The update service distributes automatic updates for the Azure Sphere OS and for applications. The update service ensures continued operation and enables the remote servicing and update of application software.

- **Error reporting**. The error reporting service provides simple crash reporting for deployed software. To obtain richer data, use the reporting and analysis features that are included with a Microsoft Azure subscription.

All data stored with the Azure Sphere Security Service is encrypted at rest by default. The Security Service stores data in Azure Storage, Cosmos DB, and Azure Key Vault, using the data encryption at rest implementation for each such service.

Recommended content

Azure Internet of Things (IoT) technologies and solutions

Describes the collection of technologies and services you can use to build an Azure IoT solution.

Introduction to the Azure Internet of Things (IoT)

Introduction explaining the fundamentals of Azure IoT and the IoT services, including examples that help illustrate the use of IoT.

What is Azure IoT Central

Azure IoT Central is an IoT application platform that simplifies the creation of IoT solutions. It helps to reduce the burden and cost of IoT management operations, and development. This article provides an overview of the features of Azure IoT Central.

IoT concepts and Azure IoT Hub

This article discusses the basic concepts for new users of Azure IoT Hub

Azure IoT aPaaS and PaaS solution options

Explains why it's a good idea to start your IoT journey with IoT Central

Get started with Azure IoT

Guidance on how to get started on your IoT journey. Why you should start with the application platform as a service (aPaaS) model.

Overview of the Azure Certified Device program

An overview of the Azure Certified Device program for our partners and customers. Use these resources to start the device certification process. Find out how to certify your device, from IoT device requirements to publishing your device.

Choose an Internet of Things (IoT) solution in Azure - Azure Architecture Center

Use Azure IoT Central or individual Azure platform-as-a-service (PaaS) components to build, deploy, and manage internet-of-things (IoT) solutions.

Show more