# 230407.Arm.Hypervisor

Created: 2023-05-01 22:17:24

# Questions

- TCB
  - What are components in TCB? `(TrustZone.0329.md)`
  - What are components in ARM TCB? `(TrustZone.0329.md)`
  - What are Security Building Blocks? `(2023/4/12)` `(Papers.OP-TEE.0406.md)`
  - What compare Security Building Block and TCB and also summarize them in a table. `(2023/4/12)` `(Papers.OP-TEE.0406.md)`
- Virtualization
  - What are types of virtualization techs? `(Papers.OP-TEE.0406.md)`
- Hypervisor
  - What is hypervisor? `(OP-TEE.0327.md)`
  - Is hypvervisor a virtual machine manager? `(Papers.OP-TEE.0406.md)`
  - Is hypervisor a software or a hardware? `(TrustZone.0329.md)`
- Hypervisor & Virtualization
  - Please compare hypervisor and virtualization and also summarize them in a table. `(Papers.OP-TEE.0406.md)`
  - Is TrustZone a hypervisor or virtualization? `(Papers.OP-TEE.0406.md)`
  - Is OP-TEE a hypervisor or virtualization? `(Papers.OP-TEE.0406.md)`
  - Does virtualization depend on hypervisor? `(Papers.OP-TEE.0406.md)`
  - Are there alnatives of hypervisor? `(Papers.OP-TEE.0406.md)`
- VM
  - Does VM require hypervisor? `(Papers.OP-TEE.0406.md)`
  - Can a VM contain VMs? `(Papers.OP-TEE.0406.md)`
- Hypervisor & TrustZone
  - Please compare TrustZone and Hypervisor and also summarize in a table. `(TrustZone.0329.md)`
  - Can hypervisor run in secure world? `(OP-TEE.0327.md)`
  - What is difference of runnin hypervisor in secure world and normal world. Please also summarize it in a table. `(OP-TEE.0327.md)`
  - Does TrustZone require hypervisor? `(OP-TEE.0327.md)`

---

# Q & A

## TCB

Question: What are components in TCB? `(TrustZone.0329.md)`

Answer:

The Trusted Computing Base (TCB) is the set of all hardware, firmware, and software components critical to the security of a system. These components are responsible for enforcing the security policy and maintaining the system’s integrity, confidentiality, and availability. The exact components of a TCB may vary depending on the specific system architecture and implementation. However, some common components in a TCB include:

1. Hardware: - Trusted Platform Module (TPM) - Secure boot hardware support - Hardware security features (e.g., ARM TrustZone, Intel SGX) - Hardware-based encryption and key storage

2. Firmware: - Bootloader (e.g., UEFI Secure Boot, Coreboot) - Trusted firmware (e.g., ARM Trusted Firmware) - Hardware initialization code - Firmware for security peripherals (e.g., TPM firmware)

3. Operating System (OS) components: - Security-focused kernel (e.g., seL4, Xen) - OS security features and policies - Hypervisor (in virtualized environments) - Mandatory access control mechanisms (e.g., SELinux, AppArmor)
4. Security applications and services: - Trusted execution environments (e.g., OP-TEE) - Security-focused libraries and tools (e.g., OpenSSL, GnuPG) - Cryptographic services - Authentication and access control services

These components work together to ensure the security of a system and provide a trusted foundation for applications and services. Note that the components listed above are just examples, and the actual TCB components may differ based on the specific system and its security requirements.

---

Question: What are components in ARM TCB? `(TrustZone.0329.md)`

Answer:

The Trusted Computing Base (TCB) of an ARM system consists of the essential components responsible for maintaining security properties and enforcing security policies. In the context of ARM TrustZone and related technologies, the TCB can include the following components:

1. Secure Boot: This is the mechanism that ensures the integrity and authenticity of the boot process, making sure that only trusted software components are executed during the system startup.
2. ARM TrustZone: A hardware-based security feature in ARM processors, which provides a secure execution environment (Secure World) and isolates it from the less secure environment (Normal World).
3. Secure Monitor: This is the lowest-level software component running in the Secure World. It manages transitions between the Secure and Normal Worlds and provides an interface for Secure World services.
4. Trusted Firmware-A (TF-A) or other trusted firmware: This firmware runs in the Secure World and provides various services, such as secure boot, power management, and platform-specific configurations.
5. OP-TEE or other Trusted Execution Environment (TEE) implementations: TEEs, like OP-TEE, provide a secure environment for running trusted applications (TAs) within the Secure World, ensuring that sensitive code and data are protected from unauthorized access or tampering.
6. Secure Partition Manager (SPM): This component manages Secure Partitions (SPs) within the TrustZone Secure World, including creating, configuring, and managing SPs, as well as inter-partition communication and resource management.
7. Trusted Applications (TAs): These are secure software components that run within the Secure World and provide security-sensitive services or perform security-critical operations.
8. Security-sensitive hardware components: These components, such as cryptographic accelerators, secure storage, and hardware random number generators, provide additional security features and services to the system.
9. Secure peripherals: These are peripherals that can be accessed only from the Secure World, providing additional isolation and protection for sensitive data and operations.

Please note that the specific components in an ARM TCB may vary depending on the system's architecture, SoC, and security requirements. The list above provides a general overview of the typical components in an ARM TrustZone-based TCB.

---

Question: What are Security Building Blocks? `(2023/4/12)` `(Papers.OP-TEE.0406.md)`

Answer:

Security Building Blocks refer to the fundamental components or technologies used to design and implement a secure system. These building blocks form the foundation for creating secure software, hardware, and network architectures. They are essential for safeguarding sensitive information and ensuring that systems are resistant to unauthorized access, manipulation, or other forms of attack. Some common security building blocks include:

1. Encryption: The process of encoding data to protect it from unauthorized access. Encryption algorithms and protocols, such as AES, RSA, or TLS, are widely used to secure data during transmission or storage.
2. Authentication: The process of verifying the identity of a user, device, or system to ensure that only authorized entities

can access the system or its resources. Common authentication methods include passwords, biometric data, or digital certificates.

3. Authorization: The process of granting or denying access to specific resources or actions based on an authenticated entity's privileges. Role-based access control (RBAC) and attribute-based access control (ABAC) are examples of authorization models.
4. Integrity: Ensuring that data and systems remain unaltered and free from tampering or corruption. Cryptographic hash functions, digital signatures, and checksums are examples of techniques used to maintain data integrity.
5. Confidentiality: The protection of sensitive information from unauthorized disclosure. Encryption, access control, and secure storage solutions are used to maintain data confidentiality.
6. Auditing and Logging: The process of recording system events, such as user actions or security incidents, to detect potential threats and ensure compliance with security policies or regulations.
7. Secure Communication: Ensuring that data transmitted between systems or devices remains confidential, authentic, and unaltered. Secure communication protocols like TLS/SSL and VPNs are used to protect data during transmission.
8. Secure Boot: The process of verifying the integrity of a system's firmware and software during startup, ensuring that only authorized code is executed.
9. Hardware Security Modules (HSMs): Physical devices that provide secure storage and processing for cryptographic keys and other sensitive information.

These building blocks, when combined and implemented correctly, create a robust and secure foundation for any system, ensuring the protection of data and the prevention of unauthorized access or attacks.

---

Question: What compare Security Building Block and TCB and also summarize them in a table. `(2023/4/12)` `(Papers.OP-TEE.0406.md)`

Answer:

Security Building Blocks and Trusted Computing Base (TCB) are two related concepts in the context of secure system design. Here is a comparison between the two:

| Aspect | Security Building Blocks | Trusted Computing Base (TCB) |
|---|---|---|
| Definition | Fundamental components or technologies used to design and implement a secure system. | The smallest set of software and hardware components that are critical to the security of a system. |
| Purpose | To create secure software, hardware, and network architectures by providing a foundation for safeguarding sensitive information and ensuring system resistance to unauthorized access or manipulation. | To ensure that the system operates securely and enforces security policies by isolating critical components from the rest of the system. |
| Components/Elements | Encryption, Authentication, Authorization, Integrity, Confidentiality, Auditing and Logging, Secure Communication, Secure Boot, and Hardware Security Modules (HSMs). | TCB includes the operating system's kernel, security mechanisms, hardware components (e.g., CPU, memory), and other essential components that enforce security policies. |
| Trust | Security Building Blocks are used to create a trusted system by combining various security mechanisms and techniques. | The TCB is the core of a trusted system, and it is assumed to be secure and trustworthy. Its correct implementation and operation are crucial to the overall security of the system. |
| Scope | Security Building Blocks cover a wide range of security aspects in a system, such as data protection, access control, and secure communication. | TCB is a more focused concept, mainly concerned with the critical components that enforce security policies and protect the system's security properties. |

In summary, Security Building Blocks are the fundamental components and technologies used to create secure systems, while the Trusted Computing Base (TCB) represents the minimal set of critical components essential for ensuring the security of a system. Security Building Blocks can be seen as the elements that are combined to create a secure system, whereas the TCB is the core of the system, ensuring its overall security.

---

## Virtualization

Question: What are types of virtualization techs? `(Papers.OP-TEE.0406.md)`

Answer:

There are various types of virtualization technologies, each targeting a specific layer of computing or specific use cases. Here are some of the major types:

1. Hardware or Platform Virtualization: This type of virtualization involves the creation of virtual machines (VMs) that share the physical hardware resources of a host system. Examples of hypervisors that enable this type of virtualization include VMware ESXi, Microsoft Hyper-V, and KVM.

2. Operating System Virtualization: This technology involves the creation of isolated environments, or containers, within a single host operating system. Each container shares the host OS's kernel but can have its own file system, libraries, and applications. Examples of containerization platforms include Docker, LXC, and Kubernetes.

3. Application Virtualization: This technique enables running applications in isolated environments on a host system without needing to install them. Application virtualization ensures that the application's dependencies do not interfere with the host system or other applications. Examples include Microsoft App-V and VMware ThinApp.

4. Network Virtualization: This type of virtualization abstracts the physical network infrastructure, allowing multiple virtual networks to share the same physical hardware. Network virtualization enables better utilization of resources, improved security, and easier network management. Examples include VMware NSX and Cisco ACI.

5. Storage Virtualization: Storage virtualization abstracts the physical storage devices, allowing them to be pooled and managed centrally. This enables better resource utilization, improved performance, and easier management. Examples of storage virtualization technologies include software-defined storage (SDS) solutions, such as Ceph and VMware vSAN.

6. Desktop Virtualization: This technology enables the delivery of a virtual desktop environment to users over a network. It separates the user's desktop environment from the physical hardware, allowing for centralized management, better resource utilization, and improved security. Examples of desktop virtualization solutions include VMware Horizon and Citrix Virtual Apps and Desktops.

These are just a few examples of virtualization technologies, and many more specific solutions and implementations exist in the industry.

---

## Hypervisor

Question: What is hypervisor? `(OP-TEE.0327.md)`

Answer:

A hypervisor, also known as a virtual machine monitor (VMM), is a type of software that enables the creation and management of virtual machines (VMs) on a physical host machine. The hypervisor is responsible for abstracting the underlying hardware resources of the host and allocating them to the virtual machines, allowing multiple VMs to run simultaneously on a single host.

There are two main types of hypervisors:

1. Type 1 hypervisor (Bare-metal): This type of hypervisor runs directly on the host's hardware without the need for an underlying operating system. It provides strong isolation between VMs and efficiently manages hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Microsoft Hyper-V, and Xen.

2. Type 2 hypervisor (Hosted): This type of hypervisor runs on top of an existing operating system, which acts as the host.

The VMs are created and managed within the host OS. Type 2 hypervisors typically have higher overhead and lower performance compared to Type 1 hypervisors, but they can be easier to set up and manage. Examples of Type 2 hypervisors include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

Hypervisors enable efficient use of computing resources, making it possible to run multiple operating systems and applications on the same physical hardware, improving resource utilization, and enabling greater flexibility in managing workloads. They are widely used in data centers, cloud computing environments, and for virtualization in general.

---

Question: Is hypvervisor a virtual machine manager? `(Papers.OP-TEE.0406.md)`

Answer:

Yes, a hypervisor is often referred to as a virtual machine manager (VMM). A hypervisor is a software or firmware that manages the creation, execution, and management of virtual machines (VMs) on a host system. It allows multiple VMs to run concurrently on the same physical hardware, while providing isolation and resource management between them.

Hypervisors are responsible for abstracting the physical hardware resources and allocating them to the VMs, handling the VMs' requests for resources, and ensuring that the VMs do not interfere with each other or the host system. They play a crucial role in the virtualization process, making it possible to create and manage VMs efficiently and securely.

---

Question: Is hypervisor a software or a hardware? `(TrustZone.0329.md)`

Answer:

A hypervisor is a software component that enables virtualization, allowing multiple operating systems or environments to run on a single physical host. Hypervisors can be classified into two types:

1. Type-1 hypervisors (bare-metal hypervisors): These hypervisors run directly on the hardware, providing better performance and more efficient resource management. Examples include VMware ESXi, Microsoft Hyper-V, and Xen.

2. Type-2 hypervisors (hosted hypervisors): These hypervisors run on top of a host operating system, sharing the system's resources with the guest operating systems. Examples include VMware Workstation, Oracle VirtualBox, and Parallels Desktop.

In summary, hypervisors are software components that enable virtualization and are not hardware themselves. However, some modern processors have hardware-assisted virtualization features that can improve the performance and security of hypervisors.

---

## Hypervisor & Virtualization

Question: Please compare hypervisor and virtualization and also summarize them in a table. `(Papers.OP-TEE.0406.md)`

Answer:

Hypervisor and virtualization are related concepts but serve different purposes. Here's a comparison and summary of both in a table:

| Aspect | Hypervisor | Virtualization |
|---|---|---|
| Definition | A software layer or firmware that manages and controls virtual machines (VMs). | The process of creating multiple isolated virtual environments on a single physical hardware. |
| Purpose | Allows multiple VMs to run concurrently on a single hardware platform. | To efficiently utilize hardware resources, improve flexibility, and increase security by isolating workloads. |
| Types | 1. Type 1 (Bare-metal): Runs directly on hardware. Examples: VMware ESXi, Microsoft Hyper-V, Xen, and KVM.<br>2. Type 2 (Hosted): Runs as a software layer on top of an existing operating system (OS). Examples: VMware Workstation, VirtualBox, and Parallels Desktop. | 1. Full Virtualization: VMs have no direct access to the hardware and are managed by a hypervisor.<br>2. Paravirtualization: VMs are aware of the hypervisor and can make calls to the hypervisor for better performance.<br>3. OS-level Virtualization: Creates isolated instances (containers) within a single OS kernel, sharing the same OS but having their own file systems and process spaces. Examples: Docker and LXC. |
| Functionality | Allocates resources such as CPU, memory, storage, and network to VMs, and manages their life cycle. | Separates applications, OSs, and hardware resources, enabling multiple workloads to run simultaneously on a single physical host. |
| Performance Impact | Hypervisors may introduce some overhead due to resource management and virtual-to-physical resource mapping. | Virtualization may cause a performance impact, depending on the type and extent of virtualization used. However, paravirtualization and OS-level virtualization generally have lower overhead compared to full virtualization. |

In summary, a hypervisor is a software layer or firmware that enables and manages virtualization. Virtualization, on the other hand, is the process of creating isolated virtual environments on a single physical hardware platform, allowing for more efficient resource utilization, flexibility, and increased security.

---

Question: Is TrustZone a hypervisor or virtualization? `(Papers.OP-TEE.0406.md)`

Answer:

TrustZone is neither a hypervisor nor virtualization. TrustZone is a hardware-based security technology developed by ARM. It provides an isolated and secure environment called the Trusted Execution Environment (TEE) for running sensitive and security-critical applications, alongside the Rich Execution Environment (REE) where the main operating system and applications run.

TrustZone creates two separate worlds - the secure world and the normal world - by partitioning the hardware resources such as processor, memory, and peripherals. The secure world is designed to protect sensitive data and operations, while the normal world runs the typical operating system and applications. The two worlds are isolated at the hardware level, ensuring that security vulnerabilities in the normal world do not compromise the secure world.

In summary, TrustZone is a security feature designed to provide hardware-level isolation for sensitive operations and data, rather than a hypervisor or virtualization technology.

---

Question: Is OP-TEE a hypervisor or virtualization? `(Papers.OP-TEE.0406.md)`

Answer:

OP-TEE (Open Portable Trusted Execution Environment) is neither a hypervisor nor virtualization. It is a Trusted Execution Environment (TEE) designed to work with ARM TrustZone technology. OP-TEE is an open-source software solution that provides a secure environment for running security-critical applications.

When used in conjunction with TrustZone, OP-TEE provides the secure operating system and services that run within the secure world created by TrustZone. It manages access to secure resources and protects sensitive data and operations from unauthorized access in the normal world.

In summary, OP-TEE is a TEE solution that works with TrustZone to provide a secure environment for security-critical applications, rather than a hypervisor or virtualization technology.

---

Question: Does virtualization depend on hypervisor? `(Papers.OP-TEE.0406.md)`

Answer:

Yes, virtualization typically depends on a hypervisor. A hypervisor is a critical component of virtualization technology, as it is responsible for creating, managing, and running virtual machines (VMs) on a physical host system. The hypervisor manages the distribution of resources, such as CPU, memory, and storage, among the various VMs and provides isolation between them.

There are two types of hypervisors:

1. Type 1 (bare-metal) hypervisors: These hypervisors run directly on the host's hardware, without the need for a host operating system. Examples include VMware ESXi, Microsoft Hyper-V, and Xen.

2. Type 2 (hosted) hypervisors: These hypervisors run on top of a host operating system and manage virtual machines within the host environment. Examples include Oracle VirtualBox, VMware Workstation, and Microsoft Virtual PC.

In both cases, the hypervisor is the key component that enables virtualization, providing the necessary management and isolation between virtual machines running on a single physical host.

---

Question: Are there alnatives of hypervisor? `(Papers.OP-TEE.0406.md)`

Answer:

Yes, there are alternatives to hypervisors for creating isolated environments. Some of these alternatives include:

1. Containers: Containers are a lightweight form of virtualization that enables the running of multiple isolated applications on the same host without the need for a full-fledged hypervisor. Containers share the host's kernel and can run directly on the host's operating system, leading to less overhead compared to hypervisors. Examples of container technologies include Docker, LXC, and Kubernetes.

2. Paravirtualization: Paravirtualization is an approach to virtualization where the guest operating system is modified to be aware of the virtualized environment and communicates directly with the host's hardware through a thin layer called the paravirtualization interface. This approach can offer better performance than full virtualization, but it requires modifying the guest operating system, which may not always be feasible. Xen is an example of a paravirtualization platform.

3. Application-level virtualization: Application-level virtualization isolates individual applications rather than entire operating systems, allowing them to run in a sandboxed environment. This can provide isolation and security benefits without the overhead of full-system virtualization. Examples of application-level virtualization include Java Virtual Machine (JVM), which runs Java applications in a sandboxed environment, and Microsoft's .NET runtime.

4. OS-level virtualization: OS-level virtualization provides isolated environments by partitioning a single operating system into multiple isolated instances called zones, jails, or namespaces. Each instance has its own file system, network stack, and process space, but they all share the same kernel. Examples of OS-level virtualization technologies include Solaris Zones, FreeBSD Jails, and Linux namespaces.

These alternatives each have their own advantages and trade-offs, and their suitability depends on the specific use case and requirements of the system being virtualized.

---

## VM

Question: Does VM require hypervisor? `(Papers.OP-TEE.0406.md)`

Answer:

Yes, virtual machines (VMs) typically require a hypervisor to function. A hypervisor is the essential component responsible for creating, managing, and running VMs on a physical host system. It allocates the physical resources, such as CPU, memory, and storage, among the VMs and provides isolation between them. VMs are essentially software-based environments running on the hypervisor, which simulates the hardware and allows multiple VMs with different operating systems to share the same physical hardware resources.

There are alternative virtualization technologies, such as containerization (e.g., Docker), which do not rely on a hypervisor but instead use the host operating system's kernel to isolate and manage containers. However, in the context of traditional virtual machines, a hypervisor is a necessary component for their operation.

---

Question: Can a VM contain VMs? `(Papers.OP-TEE.0406.md)`

Answer:

Yes, a virtual machine (VM) can contain other VMs, a concept known as nested virtualization. In nested virtualization, a hypervisor runs inside a virtual machine created by another hypervisor. This creates a hierarchy of virtual machines, where the primary hypervisor manages the first-level virtual machines, and the nested hypervisors manage the second-level virtual machines.

Nested virtualization can be useful in various scenarios, such as:

1. Testing and development: Nested virtualization allows developers to test and experiment with different hypervisors and configurations without needing multiple physical machines.
2. Training: Nested virtualization can be used for educational purposes, providing hands-on experience with different virtualization technologies.
3. Cloud environments: Nested virtualization can help cloud providers support customers who want to use their own hypervisors on top of the cloud provider's infrastructure.

However, nested virtualization can also introduce additional overhead and complexity, which can negatively impact the performance of the nested VMs. Additionally, not all hypervisors support nested virtualization, and hardware virtualization support may be required for acceptable performance.

---

## Hypervisor & TrustZone

Question: Please compare TrustZone and Hypervisor and also summarize in a table. `(TrustZone.0329.md)`

Answer:

TrustZone and hypervisors serve different purposes in providing security and isolation in computer systems, but they can also be complementary. Here's a comparison between the two:

TrustZone: 1. Purpose: Provides hardware-based security and isolation between the secure and non-secure worlds. 2. Scope: Limited to ARM-based systems. 3. Implementation: Built into the ARM processor, enabled by the SoC vendor. 4. Isolation: Creates two separate execution environments (secure world and normal world). 5. Use Cases: Secure boot, key storage, cryptographic operations, and DRM.

Hypervisor: 1. Purpose: Enables the virtualization of multiple operating systems or environments on a single physical host. 2. Scope: Applicable to various processor architectures (x86, ARM, etc.). 3. Implementation: Software-based, can be a Type-1 (bare-metal) or Type-2 (hosted) hypervisor. 4. Isolation: Creates multiple virtual machines (VMs) or containers, each running a separate OS or application. 5. Use Cases: Server consolidation, sandboxing, running multiple OSes on a single device, and cloud computing.

Summary in a table:

| Feature | TrustZone | Hypervisor |
|---|---|---|
| Purpose | Hardware-based security | Virtualization |
| Scope | ARM-based systems | Multiple platforms |
| Implementation | Built into ARM processor | Software-based |
| Isolation | Secure and normal worlds | Virtual machines/containers |
| Use Cases | Secure boot, key storage, crypto operations, DRM | Server consolidation, sandboxing, multiple OSes, cloud computing |

Keep in mind that TrustZone and hypervisors can be used together in some cases to enhance the overall security of a system. For example, a TrustZone-enabled secure world can host a secure hypervisor to further isolate and protect sensitive applications.

---

Question: Can hypervisor run in secure world? `(OP-TEE.0327.md)`

Answer:

Yes, a hypervisor can run in the secure world, but it is less common. Typically, hypervisors run in the normal world, and they manage VMs in the normal world while ensuring their isolation. However, there are secure world hypervisors, like ARM TrustZone-based hypervisors, which are designed to run in the secure world and manage secure partitions or secure VMs.

A secure world hypervisor manages the secure partitions, including OP-TEE, and ensures their isolation from each other and the normal world. In this configuration, OP-TEE acts as one of the secure partitions managed by the secure world hypervisor. This approach allows multiple secure services to coexist in the secure world, each running in its own isolated partition or secure VM.

It's worth mentioning that having a hypervisor in the secure world adds complexity to the system, and its implementation must ensure that it does not introduce vulnerabilities or weaken the security guarantees provided by the secure world.

---

Question: What is difference of runnin hypervisor in secure world and normal world. Please also summarize it in a table.`(OP-TEE.0327.md)`

Answer:

| Aspect | Secure World Hypervisor | Normal World Hypervisor |
|---|---|---|
| 1. Security | Enhanced security and isolation | Less secure environment |
| 2. Trust boundaries | Well-defined trust boundaries for secure services | Trust boundaries managed within the normal world |
| 3. Resource allocation | Resource constraints due to limited secure world size | More resources available for the hypervisor |
| 4. Complexity | Increased development and maintenance complexity | Simpler development and maintenance process |
| 5. Attack surface | Smaller attack surface due to isolation | Potentially larger attack surface |
| 6. Isolation | Strong isolation between secure and normal world | Less strict isolation between guest OSs and hypervisor |

---

Question: Does TrustZone require hypervisor? `(OP-TEE.0327.md)`

Answer:

No, TrustZone does not require a hypervisor. TrustZone is a hardware-based security technology implemented in Arm processors that provides a secure execution environment by creating two separate worlds: the Secure World and the Normal World. TrustZone itself is not dependent on a hypervisor.

A hypervisor is a software layer that manages and coordinates multiple virtual machines (VMs) running on a single hardware platform. Hypervisors can be used in conjunction with TrustZone to provide additional isolation and management capabilities for virtualized systems.

While TrustZone and hypervisors can work together, they serve different purposes and are not dependent on one another. TrustZone focuses on providing a secure environment for trusted applications, while hypervisors manage VMs and their resources.

---