ETSI TS 104 033 V1.1.1 (2026-05)

TECHNICAL SPECIFICATION

Securing Artificial Intelligence (SAI);
Security requirements for
an Artificial Intelligence Computing Platform

Reference
DTS/SAI-006

Keywords
artificial intelligence, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Securing Artificial Intelligence (SAI).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the requirements for a security framework of an AI computing platform. It further defines the security functions to be provided by the platform, the components to be implemented in the platform and their interfaces.

The present document is intended for use by designers of AI computing platforms.

The present document extends from the conclusions presented in ETSI GR SAI 009 [i.1].

NOTE: The present document applies to AI computing platforms that are deployed in data centres or edge computing environments. Other forms of computing platform, e.g. mobile phones or embedded devices capable of executing AI functionality, may also refer to this security framework adapted to specific conditions, resource constraints and security requirements.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 104 224: "Securing Artificial Intelligence (SAI); Explicability and transparency of AI processing".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] ETSI GR SAI 009: "Securing Artificial Intelligence (SAI); Artificial Intelligence Computing Platform Security Framework".

[i.2] OWASP: "CycloneDX v1.7".

[i.3] ETSI TR 104 048: "Securing Artificial Intelligence (SAI); Data Supply Chain Security".

[i.4] ETSI EN 303 645: "CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements".

[i.5] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[i.6] ISO/IEC 24970: "Artificial intelligence — AI system logging".

[i.7] ETSI EN 304 223: "Securing Artificial Intelligence (SAI); Baseline Cyber Security Requirements for AI Models and Systems".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

AI application: software program that use AI techniques to perform specific tasks

AI computing platform: computing platform intended to host AI applications

AI model: computer program that has been trained on a set of data to recognize certain patterns or make certain decisions without further human intervention

execution environment: context in which computer code can execute (run)

3.2 Symbols

For the purposes of the present document, the following symbols apply:

{*Security component*} delimits a "security component" described in Annex B that is involved in the interactive procedures

EXAMPLE: {Security management module}, {Host HMEE security function}.

<*Service-based interface*> delimits a "Service-based interface" described in Annex B that is used to deliver relevant information or data between the AI computing platform and platform users

EXAMPLE: <N4>, <S1>.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AES	Advanced Encryption Standard
AI	Artificial Intelligence
BMC	Baseboard Management Controller
BoM	Bill of Materials
CPU	Central Processing Unit
DoS	Denial of Service
GCM	Galios Counter Mode
GPU	Graphics Processing Unit
HBRT	Hardware Based Root of Trust
HMEE	Hardware-Mediated Execution Enclave
HSM	Hardware Security Module
HUK	Hardware Unique Key
JTAG	Joint Test Action Group
LLM	Large Language Model
NIC	Network Interface Card
NPU	Network Processing Unit
OS	Operating System
RPO	Recovery Point Objective
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SDK	Software Development Kit
SE	Secure Enclave
SoC	System on Chip

TCG	Trusted Computing Group
TLS	Transport Layer Security
TPM	Trusted Platform Module
UI	User Interface
VM	Virtual Machine

4 Overview

The AI computing platform is a platform that is optimized to provide an execution environment and related resources to an AI system as shown in Figure 1 with a detailed view of the AI computing platform given in Figure 2.

The baseline requirements from ETSI EN 303 645 [i.4] apply (and shown in Table 1 below) in addition to specific requirements identified in the present document. Refer to Annex A for details.

Table 1: Baseline provision from ETSI EN 303 645 [i.4] mapped to the AI Computing platform requirement

Requirement in ETSI EN 303 645 [i.4]	Equivalent or extended provision in the present document
5.1 No universal default passwords	n/a
5.2 Implement a means to manage reports of vulnerabilities	
5.3 Keep software updated	
5.4 Securely store sensitive security parameters	Addressed in clause 5.1.4.2
5.5 Communicate securely	Addressed in clause 5.1.4.1
5.6 Minimize exposed attack surfaces	
5.7 Ensure software integrity	
5.8 Ensure that personal data is secure	
5.9 Make systems resilient to outages	
5.10 Examine system telemetry data	
5.11 Make it easy for users to delete user data	
5.12 Make installation and maintenance of devices easy	
5.13 Validate input data	
6 Data protection provisions for consumer IoT	

The AI Computing platform shall make provision to support the requirements for transparency and explicability of AI processing defined in ETSI TS 104 224 [1], and should support the baseline security requirements for AI models and systems in ETSI EN 304 223 [i.7]. Refer to Annex F for details. The platform also should respect the recommendations for data supply chain security outlined in ETSI TR 104 048 [i.3].

The AI computing platform is decomposed into three (3) distinct layers:

- Hardware layer, composed of elements to give assurance of hardware enabled secure storage, networking hardware and specialist computing hardware in support of AI functions (e.g. AI Accelerator elements).
- Basic software layer, is an interface to provide the hardware layer capabilities to AI application providers and users and is composed of operating system, chip-level SDK, virtualization component for VM or containers, etc.
- (AI) Application enabling layer, is the AI application facing element of the platform and is composed of different types of deep learning framework, application-level SDK, management module, etc.

NOTE: As the AI Computing platform defined in the present document is intended for deployment primarily in data centres or edge computing environments the elements of a computing platform that would be present for a desktop or UI-centric environment (e.g. graphics processing) are not considered.

The further decomposition of each of the three (3) distinct layers are given in Annex B.

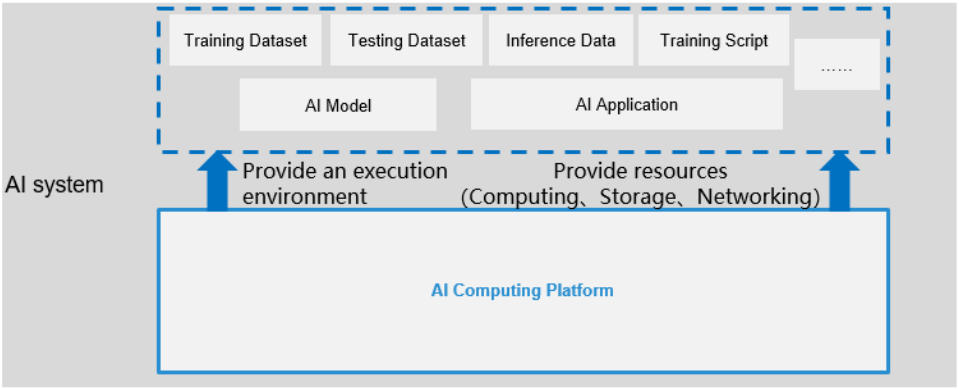

Figure 1: AI computing platform overview

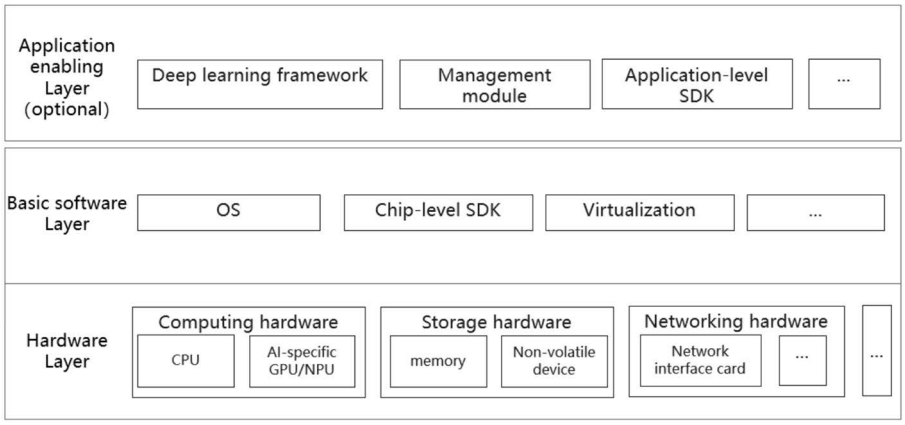

Figure 2: De-composition of a typical AI computing platform

The platform provides resources required for the AI model and associated AI application including (see also ETSI GR SAI 009 [i.1]):

- Computing execution environment (in particular, one or more AI accelerated processors are included), the execution environment includes the fundamental software framework, various kinds of libraries and different hardware drivers.
- Storage to give secure and reliable support of assets including training dataset, testing dataset, inference data and training script, etc.
- Networking.

In ETSI GR SAI 009 [i.1], Table 1 and Table 2, threats to the AI computing platform are summarized and analysed. The threat analysis from ETSI GR SAI 009 [i.1] is adopted without change for the present document. From the security threat analysis provided in ETSI GR SAI 009 [i.1] the following security services are identified and specified in detail in the present document:

- Protection of AI assets in transmission and storage.
- Protection of AI assets during processing.
- AI accelerator resource isolation service.
- Model training recovery service.
- Inference attack detection service.
- AI related log protection.
- Model BoM service.

In addition, the AI Computing platform shall support the following more general services in support of the AI Computing platform specific services listed above:

- Identity management and access control (see clause 5.1.2).
- Integrity protection (see clause 5.1.3):
  - Protection of integrity of system startup data.
  - Protection of integrity of AI model data.
- Data protection (see clause 5.1.4):
  - Data in transit.
  - Data at rest.
- Secure audit (see clause 5.1.5).
- Secure response (see clause 5.1.6).
- Resilience (see clause 5.1.7).

5 Security requirements and security functions of the framework

5.1 Security requirements for the AI computing platform

5.1.1 Overview

The security requirements for the AI computing platform are defined in order to mitigate threats against itself.

The baseline security requirements are grouped as shown in Figure 3 in order to protect the AI Computing platform.

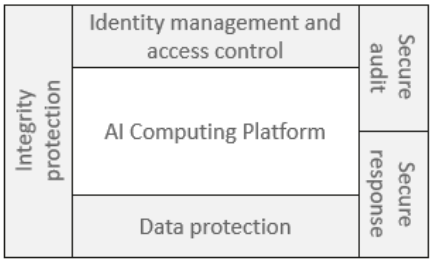

Figure 3: Overview of baseline security requirements of the AI computing platform

NOTE 1: Unless otherwise specified, all security requirements are applicable to the AI computing platform in data centre and edge computing scenarios.

NOTE 2: The implementation reference for security mechanism of the AI computing platform based on clause 5 is provided in Annex D.

5.1.2 Identity management and access control

The AI computing platform shall implement the principle of least privilege where the following requirements on identity management and access control apply:

- External physical interfaces of the AI computing platform, e.g. JTAG ports, shall be disabled after manufacture.

NOTE 1: This is consistent with the principles outlined in clause 5.6 of ETSI EN 303 645 [i.4], particularly provisions 5.6-3 and 5.6-4, to minimize the attack surface.

- Access control shall be implemented to allow only authorized access to the platform via the physical interface.
- Remote access of an account with root privilege shall be prohibited.

NOTE 2: Root privilege violates the principle of least privilege and is consistent with provision 5.6-7 of ETSI EN 303 645 [i.4].

- Only identified and authenticated parties, where authorization to access has been verified, shall be able to access resources on an AI computing platform.

5.1.3 Integrity protection

5.1.3.1 Protection of integrity of system startup data

The following requirements on integrity protection of system startup data apply:

- An AI computing platform shall support and implement a secure boot mechanism.

EXAMPLE: An AI computing platform can meet this requirement by support of solutions including those from the Trusted Computing Group (TCG) which have the capability to cooperate with hardware security modules such as a Trusted Platform Module (TPM) hardware root of trust, a Hardware Security Module (HSM) or Secure Enclave (SE) to facilitate the mechanisms including proactive integrity measurement and remote attestation.

5.1.3.2 Protection of integrity of AI model data

- The following requirements on integrity of AI model data apply: An AI computing platform should support for verifying integrity of AI model before performing inference task.

5.1.4 Data protection

5.1.4.1 Data in transit

NOTE: For the present clause data refers to AI specific data.

The following requirements on protection of data in transit apply:

- The AI computing platform shall protect data transmitted to and from the platform from unauthorized exposure.
- The AI computing platform shall protect data transmitted to and from the platform by only allowing transmission to or from identified, authenticated and authorized entities.

EXAMPLE: Known protocols such as TLSv1.3 [i.5] with appropriate selection and application of cryptographic primitives can be used to satisfy these requirements, e.g. TLS_AES_128_GCM_SHA256 (identifying TLS using AES-128 in Galois Counter Mode for encryption with SHA256 for hashing operations).

5.1.4.2 Data at rest

The following requirements on protection of data at rest apply:

- The AI computing platform shall have the ability to backup, and recover, system configuration parameters and other data that may be required for system recovery.

NOTE: This is consistent with provisions given in clauses 5.7 and 5.11 of ETSI EN 303 645 [i.4].

5.1.5 Secure audit

The following requirements on secure audit apply:

- An AI computing platform shall provide event information about AI assets in the logs.
- Log data should be protected such that it cannot be modified after being recorded.
- Log data shall be protected from unauthorized access (the general access control conditions shall apply).

5.1.6 Secure response

The following requirements for secure response apply:

- An AI computing platform should implement a network traffic detection mechanism on network interfaces and further implement a control mechanism to regulate the traffic and manage abnormal traffic.
- An AI computing platform should implement an intrusion detection mechanism in order to identify network attacks in a timely manner (i.e. at the initial stage of the attack) and support secure responses against these attacks.
- An AI computing platform should implement an anomaly detection mechanism on operating system and provide in-time responses. The scope of the detection should include, but not be limited to, the following:
  - Key file anomaly.
  - Process anomaly.
  - Container anomaly.
  - User/privileged/root account anomaly.

5.1.7 Resilience

In addition to the requirements above, the AI computing platform should be designed to maximize reliability and resilience.

The following requirements on resilience apply:

- An AI computing platform should implement adjustment or protection mechanism for AI-specific accelerators:
  - The mechanism should monitor real-time parameters of GPU or NPU like temperature, power, etc. to estimate the health status of these accelerators.
  - If abnormal status is detected, a coping mechanism should be automatically executed to change certain configurations like degrading the performance or reallocating workflows on these accelerators to avoid further system faults.
- An AI computing platform should have a safe mode which executes a set of minimally required tasks when it has been attacked or disrupted. The required tasks should include at least platform recovery procedure which brings the platform back to a predefined normal state.

5.2 Security services of the AI computing platform

5.2.1 Overview

Security services provided by an AI computing platform are described in clause 5.2. Upon utilizing these services via predefined interfaces, platform users like model providers and AI application providers can effectively reduce the risks on their AI assets like models and datasets. This clause only introduces the effect of each security service while the detailed mechanisms and interfaces of these services are described in Annex B and Annex C.

5.2.2 AI assets protection in transmission and storage

The AI computing platform shall assure the confidentiality of models and datasets when those assets are in transit or at rest (see also clause 5.1.4 above).

The service shall be provided through interfaces by platform. The service should satisfy the following requirements:

- The service should support multi-tenant scenarios where different users utilize different keys to implement encryption and decryption:
  - Each tenant in a multi-tenant scenario should manage their own keys.
  - It shall not be possible for any tenant to gain access to any other tenant's content.
  - It should be difficult for any tenant to infer the existence of any other tenant in a multi-tenant scenario.

NOTE 1: The above requirements are particularly important in a data centre environment where the AI computing platform is shared.

- The service should support hardware-bound decryption so that only the platform with designated AI-specific accelerator has the permission to decrypt the AI assets.

NOTE 2: It is very useful in edge computing scenarios where model providers only allow their well-trained models to be decrypted and deployed on designated edge devices for inference so that intellectual property of models can be protected.

- The service should provide integrity verification capability for AI assets utilizing cryptographic methods before deploying or using the certain assets to check that these assets are not tampered or forged.

NOTE 3: Leveraging the advanced pattern recognition capabilities of Large Language Models (LLM) enhance early detection and response to anomalies, fortifying the security framework of AI asset management.

EXAMPLE: To deploy this service, the secure communication module (refer to clauses B.2.12 and B.2.13) in hardware layer, the integrity protection module in software layer (refer to clause B.3.1), and the security management module (refer to clause B.4.1) and encryption/decryption module (refer to clause B.4.2) in application enabling layer can cooperate to protect AI assets in transmission and storage.

5.2.3 AI assets protection in processing

AI computing platforms need stringent security measures for the protection of AI assets during processing. It includes ensuring the integrity, confidentiality, and reliability of AI system operations and outputs.

The service shall satisfy the following requirements:

- The platform shall establish secure and isolated environments for the processing of AI assets to prevent unauthorized access and ensure data integrity.

NOTE 1: Isolation is crucial in multi-tenant platforms to prevent cross-tenant data breaches and ensure that LLM processing is not influenced by external factors.

- The service should provide mechanisms to protect the connection between CPU resource and AI accelerator resource. The mechanism should be based on cryptographic connection, connection isolation, or combination of both.

- The platform should ensure that model processing is resilient against Denial of Service (DoS) attacks, maintaining availability and functionality under various conditions.

NOTE 2: LLM, due to their resource-intensive nature, can be targets for DoS attacks, impacting their availability and effectiveness.

- The platform should utilize advanced anomaly detection algorithms tailored to model activities, identifying unusual patterns or behaviours that may indicate security incidents.

NOTE 3: Anomaly detection specific to LLM processing can provide early warnings of potential security threats, enabling prompt response measures.

EXAMPLE: To deploy this service, the hardware abnormality detection module (refer to clause B.2.10) and the AI accelerator resource isolation module (refer to clause B.2.11) in hardware layer, and the system abnormality detection module in software layer (refer to clause B.3.3) can cooperate to protect AI assets in processing.

5.2.4 AI accelerator resource isolation

The AI computing platform shall provide an AI accelerator resource isolation service.

NOTE 1: One role of the AI accelerator resource isolation service is to isolate AI assets used by different tenants from each other.

EXAMPLE 1: Utilizing this service, an AI accelerator device such as a GPU or NPU can simultaneously allocate several isolated computing and storage resource for different tenants sharing the device. The service shall satisfy the following requirements:

- The service shall guarantee that the resource allocated to one tenant is only used by this tenant. Other tenants shall not have access to or use the resource.
- The service shall ensure that the resource allocated to one tenant does not exceed its pre-configured quota.
- The service should support hardware-based isolation instead of software-based isolation.

EXAMPLE 2: That is, certain computing resource and memory resource are exclusively allocated to a certain stakeholder rather than time division multiplexing method:

- The service shall provide controlled access only to a restricted group of stakeholders, including administrators.
- Where accelerators are accessed remotely, the service shall provide mechanisms to support traffic segregation (e.g. by means of network isolation).
- The service should support isolation capabilities for prompt, data and reasoning to prevent injection attacks by malicious users during the reasoning process.

NOTE 2: The isolation mechanism can be implemented either through software-based approaches (e.g. using separate processes/modules) or hardware-based approaches (e.g. using isolated hardware channels for prompt, data, etc.).

EXAMPLE 3: To deploy this service, the HBRT module (refer to clauses B.2.1 and B.2.2), the HUK module (refer to clauses B.2.6 and B.2.7), the HMEE module (refer to clauses B.2.8 and B.2.9) and the AI accelerator resource isolation module (refer to clause B.2.11) in hardware layer, and the security management module (refer to clause B.4.1) in application enable layer can cooperate to isolate AI accelerator resource.

5.2.5 Training procedure recovery

An AI computing platform shall provide recovery service for model training procedure when executing model training tasks. This service shall be activated when an AI computing platform meets some system error accidently or suffers from some cyberattack by malicious adversaries.

NOTE 1: This service should be a requirement for data centre and training platform.

This service shall recover the training procedure, the corresponding intermediate data, training results, system parameters, etc. to the time point of system abnormality. It effectively mitigates the risks of loss of important data in the training procedure when the AI computing platform malfunctions.

NOTE 2: This is particularly important for the training of LLM.

The service shall satisfy the following requirements:

- The service shall be able to detect the system abnormality via different aspects of the AI computing platform including the chip level, the server node level and the network level. In this way, an AI computing platform can react in a timely manner to the system error or cyberattack and shorten the Recovery Point Objective (RPO).
- The service shall be able to resume the training procedure as soon as the AI computing platform has been recovered from the abnormal state.

EXAMPLE: To deploy this service, the trusted boot module (refer to clauses B.2.3 and B.2.4) and the minimal system module (refer to clause B.2.5) in hardware layer, the trust measurement module (refer to clause B.3.2) and the system abnormality detection module (refer to clause B.3.3) in basic software layer, and the training procedure recovery module (refer to clause B.4.4) in application enable layer can cooperate to recover training procedure.

5.2.6 Inference attack detection service

The AI computing platform should provide an inference attack detection service for models deployed on it. The service should satisfy the following requirements:

- Detection for inference results: the service should perform secure detection on results of AI inference, preventing the sensitive data from being inferred by exported results.
- Model Agnostic: the service should be designed to be model-agnostic, allowing seamless integration with various types of AI models deployed on the platform.

NOTE 1: By being model agnostic the detection capabilities are not contingent on having knowledge of the internal working of the model.

NOTE 2: The reason for model-agnostic method is that the platform needs to support various kinds of models and it should not be the responsibility of the platform to access the model internal to build some detection capabilities.

EXAMPLE 1: Utilization of AI to Protect AI: the service should deploy one or more detection models alongside the target inference model, leveraging AI capabilities to detect potential attacks in real-time.

EXAMPLE 2: Dynamic Model Switching: the service should support the dynamic switching of detection models in response to different threats and impacts on the target model.

- Minimization of Runtime or Deployment Overhead: the implementation of the service should impose minimal runtime or deployment overhead, ensuring that the performance of the deployed models is not compromised.

NOTE 3: This property emphasizes the platform's commitment to maintaining efficient inference processes without unnecessary computational burdens.

- Data Modality: the service accommodates different data modalities, including multi-modal inputs, enhancing its applicability across a wide range of AI applications. This adaptability allows the platform to support diverse use cases without compromising detection accuracy.
- Defending against multiple attacks: the service should extend its coverage to include the detection of energy and latency attacks, adversarial sample attacks, model stealing attacks and so on.

EXAMPLE 3: To deploy this service, the inference attack detection engine (refer to clause B.4.2) in application enable layer can be performed to detect inference attack.

5.2.7 AI related log storage and transfer requirements

An AI computing platform shall be able to provide stakeholders with a protection service for storing and transferring AI-related logs. The service should be able to prevent AI related logs from being tampered with or deleted, or provide a mechanism to verify that the logs have not been changed.

NOTE 1: This service should be a requirement for an AI computing platform deployed in a data centre.

NOTE 2: The definition of AI related log and the use of AI related log can refer to subclause 5.1 in ISO/IEC 24970 [i.6].

The service shall satisfy the following requirements:

- The service shall be able to allow stakeholders to send AI related logs to the platform for storage.
- The service shall be able to restrict the access and operation permission of different stakeholders, including administrators.
- The service should be able to utilize a cryptography mechanism to verify the integrity of the logs stored in the platform. When integrity is breached, the platform should send alerts. Other mechanisms may also be used to handle the breach.
- The service should be able to securely connect to a trusted third-party storage entity to transfer AI logs when needed.

NOTE 3: Connecting to a trusted third-party is very useful in scenarios where local storage of the AI computing platform is limited.

EXAMPLE 1: Cloud storage service and storage server are typical third-party storage entity.

- The service should ensure the reliability of the log storage system via log backup and recovery mechanisms.

EXAMPLE 2: To deploy this service, the log protection module (refer to clause B.4.5) in application enable layer may be performed to storge and transfer AI related log securely.

5.2.8 Model BoM service

The AI computing platform should provide a model BoM service for stakeholders to securely record important information in the procedure of AI model training. More information on BoM for AI model is illustrated in Annex E. The platform should provide certain mechanism to protect the integrity and authenticity of the information in model BoM so that stakeholders can utilize the information for forensics, model traceability or other security related aims. The service shall satisfy the following requirements:

- The service shall provide interfaces for stakeholders to send information to the platform for authenticity and integrity proof.
- The key used in authenticity and integrity protection shall be provided and protected by the platform. Hardware-based method should be utilized in the derivation and protection the keys.
- The service shall provide the proof manifest of the model information provided by stakeholders for further use in AI model tracing, forensic, etc.

NOTE: The format of the proof manifest is out of the scope. However, the format needs further research and standardization before model BoM can be widely used.

Annex A (normative): Mapping to baseline requirements of ETSI EN 303 645

The baseline requirements of ETSI EN 303 645 [i.4] whilst defined for IoT are identified as baseline requirements for any computing platform used for AI. Whilst some of the requirements from ETSI EN 303 645 [i.4] cannot be applied directly to the present document this annex identifies those.

Table A.1 copies the PICS proforma from ETSI EN 303 645 [i.4] and identifies applicability to the AI computing platform.

Table A.1: Applicability of provisions from ETSI EN 303 645 [i.4] to the AI Computing platform

Clause number and title			
Reference	**Status**	**Support**	**Detail**
5.0 Reporting implementation			
Provision 5.0-1	M		
5.1 No universal default passwords			
Provision 5.1-1	M F (a)		
Provision 5.1-2	M F (b)		
Provision 5.1-2A	R		
Provision 5.1-3	M F (c)		
Provision 5.1-4	M F (d)		
Provision 5.1-5	M C F (14, e)		
5.2 Implement a means to manage reports of vulnerabilities			
Provision 5.2-1	M		
Provision 5.2-2	R		
Provision 5.2-3	R		
5.3 Keep software updated			
Provision 5.3-1	R F (f)		
Provision 5.3-2	M C (15)		
Provision 5.3-3	M F (g)		
Provision 5.3-4A	R F (g)		
Provision 5.3-4B	R F (h)		
Provision 5.3-5	R F (g)		
Provision 5.3-6A	R F (h)		
Provision 5.3-6B	R F (i)		
Provision 5.3-7	M F (g)		
Provision 5.3-8	M C (12)		
Provision 5.3-9	R F (g)		
Provision 5.3-10	M F (j)		
Provision 5.3-11	R C (12)		
Provision 5.3-12	R C (12)		
Provision 5.3-13	M		
Provision 5.3-14	R C (3)		
Provision 5.3-15A	R C (3)		
Provision 5.3-15B	R C (3)		
Provision 5.3-16	M		
5.4 Securely store sensitive security parameters			
Provision 5.4-1	M F (k)		
Provision 5.4-2	M F (l)		
Provision 5.4-3	M		
Provision 5.4-4	M F (m)		
5.5 Communicate securely			
Provision 5.5-1	M		
Provision 5.5-2	R		
Provision 5.5-3	R		
Provision 5.5-4	R		
Provision 5.5-5	M F (n)		
Provision 5.5-6	R F (o)		
Provision 5.5-7	M F (o)		
Provision 5.5-8	M C (16)		

Clause number and title			
Reference	**Status**	**Support**	**Detail**
5.6 Minimize exposed attack surfaces			
Provision 5.6-1	M F (p)		
Provision 5.6-2	M		
Provision 5.6-3	R		
Provision 5.6-4A	M F (q)		
Provision 5.6-4B	R F (r)		
Provision 5.6-5	R		
Provision 5.6-6	R		
Provision 5.6-7	R		
Provision 5.6-8	R		
Provision 5.6-9	R		
5.7 Ensure software integrity			
Provision 5.7-1	R		
Provision 5.7-2	R F (s)		
5.8 Ensure that personal data is secure			
Provision 5.8-1	R F (t)		
Provision 5.8-2	M F (u)		
Provision 5.8-3	M F (v)		
5.9 Make systems resilient to outages			
Provision 5.9-1	R		
Provision 5.9-2	R		
Provision 5.9-3	R		
5.10 Examine system telemetry data			
Provision 5.10-1	R F (w)		
5.11 Make it easy for users to delete user data			
Provision 5.11-1	M		
Provision 5.11-2	R F (x)		
Provision 5.11-3	R		
Provision 5.11-4	R		
5.12 Make installation and maintenance of devices easy			
Provision 5.12-1	R		
Provision 5.12-2	R		
Provision 5.12-3	R		
5.13 Validate input data			
Provision 5.13-1A	M		
Provision 5.13-1B	M		
6 Data protection provisions for consumer IoT			
Provision 6.1	M		
Provision 6.2	M F (y)		
Provision 6.3A	M F (y)		
Provision 6.3B	M F (y)		
Provision 6.4	R F (w)		
Provision 6.5	M F (w)		
Provision 6.6	M F (z)		
Provision 6.7	R F (aa)		
Provision 6.8	R F (z)		

Clause number and title			
Reference	**Status**	**Support**	**Detail**
Condition: 3) software components are not updateable; 12) an update mechanism is implemented; 14) the consumer IoT device has no resource constraint determined by the use case that prevents the implementation of a mechanism which makes successful brute-force attacks on authentication mechanisms via network interfaces impracticable; 15) the consumer IoT device has no resource constraint determined by the use case that prevents the implementation of an update mechanism; 16) existence of critical security parameters that relate to the consumer IoT device. Feature, capability or mechanism that needs to be present for the corresponding provision to apply: a) passwords can be used to authenticate users against the device or for machine-to-machine authentication; b) pre-installed unique per device passwords can be used to authenticate users against the device or for machine-to-machine authentication; c) cryptographic authentication mechanisms, including password based mechanisms, can be used to authenticate users against the consumer IoT device or for machine-to-machine authentication; d) authentication mechanisms can be used to authenticate users against the consumer IoT device; e) authentication mechanisms can be used for authenticating users or devices via network interfaces; f) software components that are not immutable due to security reasons; g) software components of the device can be updated; h) automatic software updates are supported; i) update notifications are provided when software updates are available; j) software updates can be delivered over a network interface; k) sensitive security parameters exist in persistent storage; l) hard-coded unique per device identities are used in the consumer IoT device for security purposes; m) critical security parameters are used for integrity or authenticity checks of software updates or for protection of communication with associated services; n) the consumer IoT device allows security-relevant changes in configuration via a network interface; o) critical security parameters used by the device can be communicated outside of the device; p) unused network or network accessible logical interfaces exist; q) debug interfaces exist on the device; r) debug interfaces that are physical ports exist on the device; s) secure boot or other mechanism to detect unauthorized changes to IoT device software are supported by the device; t) the consumer IoT device sends personal data to associated services; u) the consumer IoT device sends sensitive personal data to associated services; v) the consumer IoT device includes external sensing capabilities; w) telemetry data can be collected from consumer IoT devices and products; x) personal data can be stored by an associated service; y) the consumer IoT device processes personal data on the basis of consumers' consent; z) the consumer IoT device processes personal data; aa) capabilities to collect data from consumer IoT devices or to processed data on the consumer IoT device, whose purpose is solely to compute an aggregate result.			

Annex B (informative): Security components and composition of the AI computing platform

B.1 Overview of an AI computing platform

In Annex B, each security component in each layer is described in terms of functionality and interaction with other security components in an AI computing platform. The abstract structure of the AI computing platform is shown in Figure B.1.

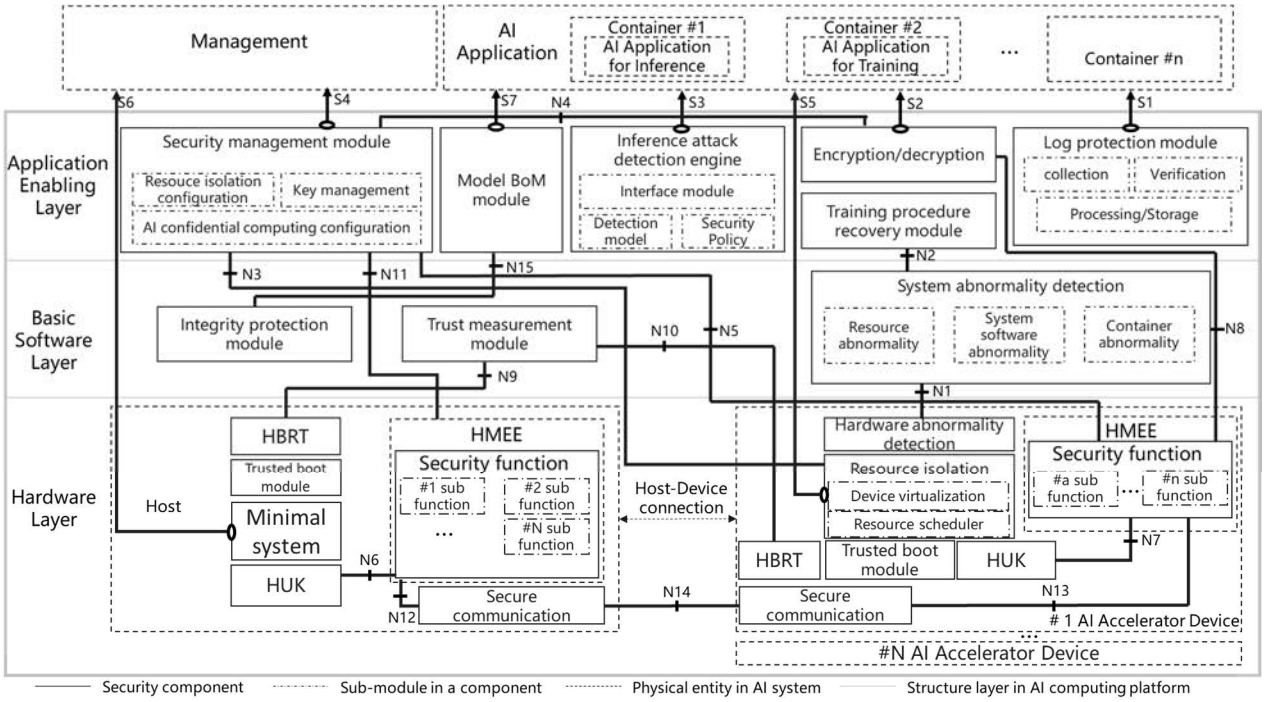

Figure B.1: Overview of the structure

The following reference points and associated service interfaces listed in Table B.1 and Table B.2 are defined with respect to the functional modules shown in Figure B.1.

NOTE 1: Reference points are not directly visible and not testable and are considered as advisory in the context of the present document.

NOTE 2: The components in Figure B.1 can be combined by a vendor, and not all the reference points, interfaces, and the components they connect to have to be discrete.

Table B.1: Reference points and associated services

Reference point	Definition and purpose
N1:	The N1 reference point between {Hardware abnormality detection module} and {System abnormality detection module} should be used to transmit hardware real-time monitoring data and abnormality alerts.
N2:	The N2 reference point between {System abnormality detection module} and {Training procedure recovery module} should be used to transmit system real-time monitoring data of an AI computing platform and abnormality alerts both from the system software and hardware.
N3:	The N3 reference point between {Security management module} and {Resource isolation module} should be used to transmit AI-specific resource isolation management and configuration data of a certain AI-specific accelerator.
N4:	The N4 reference point between {Security management module} and {Encryption/decryption module} should be used to transmit cryptographic keys used in the model and data encryption and decryption.
N5:	The N5 reference point between {Security management module} and {AI accelerator security function} should be used in key management scenarios to transmit cryptographic keys used in the model, data encryption and decryption related to device binding application scenarios, and AI confidential computing management/configuration data of certain confidential computing environments related to certain AI accelerators.
N6:	The N6 reference point between {Host HUK} and {Host HMEE security function} should be used to transmit a host HUK to a security function module in a host HMEE.
N7:	The N7 reference point between {AI accelerator HUK} and {AI accelerator HMEE security function} should be used to transmit an AI accelerator HUK to a security function module in a host HMEE.
N8:	The N8 reference point between {Encryption/decryption module} and {AI accelerator HMEE security function} should be used in AI assets encryption/decryption scenarios to transmit cryptographic keys used in the model and data encryption and decryption related to device binding application scenarios.
N9:	The N9 reference point between {Host HBRT} and {Trust measurement module} should be used to transmit integrity related data of firmware and software on the host side of as AI computing platform in integrity verification scenarios.
N10:	The N10 reference point between {AI accelerator HBRT} and {Trust measurement module} should be used to transmit integrity related data of firmware and software on the AI accelerator side of as AI computing platform in integrity verification scenarios.
N11:	The N11 reference point between {Security management module} and {Host HMEE security function} should be used to transmit AI confidential computing management and configuration data of certain confidential computing environments related to host side of the platform.
N12:	The N12 reference point between {Host secure communication module} and {Host HMEE security function} should be used to transmit data to be transferred between host and certain AI accelerators, and security policies that used in AI confidential computing mechanism. The data may be encrypted depending on security policies.
N13:	The N13 reference point between [AI accelerator secure communication module] and [AI accelerator HMEE security function] should be used to transmit data to be transferred between the certain AI accelerator and its host, and security policies used in AI confidential computing mechanism. The data may be encrypted depending on security policies.
N14:	The N14 reference point between [Host secure communication module] and [AI accelerator secure communication module] should be used to securely transmit data between host and specific AI accelerators. The data may be protected via cryptographic methods or/and isolation methods depending on security policies.
N15:	The N15 reference point between {Model BoM module} and {Integrity protection module} should be used to transmit AI Model related information that needs to be proved in terms of integrity and authenticity and the proof made based on the information.

AI applications access the functions and features of the AI computing platform through the following interfaces.

Table B.2: Service interfaces between the AI application and the AI computing platform

Service	Interface	Description
Log protection and verification	S1	to invoke the AI log protection service.
Encrypt/decrypt	S2	to invoke the encryption/decryption service.
Inference attack detection	S3	to invoke the inference attack detection service.
Security management	S4	to implement security management for the relevant security services and components.
Isolated resource allocation	S5	to request isolated AI-specific computing resource to execute AI model training or inference tasks.
Recovery	S6	to recover the platform to a predefined operation state when the platform gets attacked or malfunctions.
Model BoM inventory service	S7	to acquire integrity and authenticity proof of model relevant information that may be used in Model BoM inventory.

The definition of each of the security components is given in clauses B.2, B.3 and B.4 below.

B.2 Security components in hardware layer

B.2.1 Host Hardware Based Root of Trust (HBRT)

Host HBRT should act as RTS and RTR for integrity measurement on the host side. The functionality and the cooperation with other security components are described as follows:

- Functionality: A host HBRT should record the integrity measurement from the host side of an AI computing platform into security storage and report these measurement values through interfaces to upper layer components according to needs. These values in security storage should be protected from deletion and tampering. It is usually implemented on the motherboard of an AI computing platform.
- Coordination: A host HBRT should communicate with a trust measurement module through <N9>.

B.2.2 AI accelerator HBRT

AI accelerator HBRT should act as RTS and RTR for integrity measurement exclusively for an AI accelerator. The functionality and the cooperation with other security components are described as follows:

- Functionality: An AI accelerator HBRT should record the integrity measurement from the AI accelerator in an AI computing platform and report these measurement values through interfaces to upper layer components as needed. These values in security storage should be protected from deletion and tampering. It is usually embedded on AI accelerator SoC or AI accelerator board.
- Coordination: An AI accelerator HBRT should communicate with a trust measurement module through <N10>.

B.2.3 Host trusted boot module

Trusted boot module should be inherently trusted and secure by design. It should act as a tamper-resistant trust anchor in the host side of an AI computing platform. The functionality and the cooperation with other security components are described as follows:

- Functionality: A host trusted boot module should contain initial boot code for AI computing platform upon which other components in the platform will be successively verified and started. Initial boot code in host trusted boot module should not be tampered. It is usually implemented in CPU SoC.
- Coordination: None.

B.2.4 AI accelerator trusted boot module

AI accelerator trusted boot module should act as a tamper-resistant trust anchor in each AI accelerator of an AI computing platform. The functionality and the cooperation with other security components are described as follows:

- Functionality: An AI accelerator trusted boot module should contain initial boot code for the starting of the accelerator. Initial boot code in AI accelerator trusted boot module should not be tampered. It is usually implemented in GPU/NPU SoC.
- Coordination: None.

B.2.5 Minimal system

Minimal system should act as a secure box for an AI computing platform to execute a set of minimally required tasks when it has been attacked or disrupted. The functionality and the cooperation with other security components are described as follows:

- Functionality: A minimal system should be responsible to implement a secure mode for an AI computing platform when the platform has been attacked or is out of work. It should contain the minimally required codes for system recovery in any conditions except for physical destruction.
- Coordination: None.

B.2.6 Host Hardware Unique Key (HUK)

Host HUK should act as the unique identity credential of an AI computing platform. The functionality and the cooperation with other security components are described as follows:

- Functionality: A HUK should contain a random value which is used to generate some other keys or a predefined key that is utilized to protect other keys. HUK should not be manipulated after first setting at factory and it should only be accessed by the authorized component.
- Coordination: A HUK module should communicate with host HMEE security function module through <N6>.

B.2.7 AI accelerator HUK

AI accelerator HUK should act as the unique identity credential of a certain AI accelerator device. The functionality and the cooperation with other security components are described as follows:

- Functionality: An AI accelerator HUK should contain a random value which is used to generate some other keys or a predefined key that is utilized to protect other keys. The key should not be manipulated after first setting at factory and it should only be accessed by the authorized components.
- Coordination: An AI accelerator HUK module should communicate with AI accelerator HMEE security function module through <N7>.

B.2.8 Host Hardware Mediated Execution Environment (HMEE)

Host HMEE security function module refers to a security component that should be deployed and operate in host side HMEE. The functionality and the cooperation with other security components are described as follows:

- Functionality: Host HMEE security function module should provide a subset of crucial and sensitive security functions from AI application or system management. It should contain some parts of application codes from AI application and system software to execute specific secure and sensitive operations. The module should be deployed in and protected by host HMEE. Only authorized components within an AI system should be allowed to communicate to the module. Furthermore, communications between authorized components and this module should be restricted, i.e. output data from HMEE to rich execution environments strictly controlled.

- Coordination: A HMEE should communicate with security management module through <N11>, with HUK through <N6> and with secure communication module through <N12>.

B.2.9 AI accelerator HMEE

AI accelerator HMEE security function module refers to a security component that should be deployed and operate in the HMEE of the AI accelerator side, the functionality and the cooperation with other security components are described as follows:

- Functionality: AI accelerator HMEE security function module should provide a subset of crucial and sensitive security functions from AI application or system management. GPU/NPU-based unique information should be protected and securely processed in HMEE of AI accelerator side. The function provided by this security module should support some security services/functions in AI computing platform to bind designated device to fulfil certain security objective. Only the authorized module should be allowed to communicate with this module. For remote modules on other hosts, this module should utilize secure communication protocol like TLS to communicate.

- Coordination: An AI accelerator HMEE should communicate with security management module through <N5>, with AI accelerator HUK through <N7> and with encryption/decryption module through <N8>.

B.2.10 Hardware abnormality detection module

Hardware abnormality detection module should monitor on hardware states and operation condition to detect possible errors on hardware. The functionality and the cooperation with other security components are described as follows:

- Functionality: A hardware abnormality detection module should act as a monitor to estimate whether underlying hardware like CPU, memory space, GPU/NPU and NIC works as expected. When malfunction or errors from hardware happens, this module should timely detect these abnormalities and alert relevant security components via a certain interface for quick response. It should also provide interfaces for upper layer OS and other applications to retrieve the real-time state and operation statistics.

- Coordination: A hardware abnormality detection module should communicate with system abnormality detection module through <N1>.

B.2.11 AI accelerator resource isolation module

AI accelerator resource isolation module should provide isolation of resource of an AI accelerator like computing unit and memory space for different users. The functionality and the cooperation with other security components are described as follows:

- Functionality: An AI accelerator resource isolation module should act as a manager to isolate resource of an AI accelerator according to configuration. Different users should only utilize their authorized computing unit and memory space. If an error happens in one isolated space or a malicious user, it should not affect other users' resource. Also, this component should manage that isolated resource utilized by each user should not exceed the pre-configured quota. In addition, an interface should be provided by this component for administrators to configure relevant parameters like resource quota for a certain user.

- Coordination: An AI accelerator resource isolation module should communicate with security management module through <N3>.

- Coordination: None.

B.2.12 Host secure communication module

Host secure communication module should act as the executor on host side of secure communication between host and AI accelerators. The functionality and the cooperation with other security components are described as follows:

- Functionality: Host secure communication module should cooperate with AI accelerator secure communication module to protect the communication between host and specific AI accelerators utilizing cryptographic method or isolation method. The module should communicate with Host HMEE security function module to execute possible cryptographic operation (e.g. key configuration) and policy enforcement (e.g. access control policy) for the secure communication between host side and AI accelerators, and to bridge the host HMEE and certain AI accelerator HMEE based on the cooperation with AI accelerator secure communication module on that AI accelerator.

- Coordination: A host secure communication module should communicate with host HMEE security function module through <N12> and AI accelerator secure communication module through <N14>.

B.2.13 AI accelerator secure communication module

Similar as host secure communication module, AI accelerator secure communication module should act as the executor on each AI accelerator side of secure communication between host and AI accelerators. The functionality and the cooperation with other security components are described as follows:

- Functionality: An AI accelerator secure communication module should execute the secure communication with host. It should cooperate with host secure communication module to protect the communication between the accelerator it locates and the host. The module should communicate with AI accelerator HMEE security function module to execute possible cryptographic operation and policy enforcement for the secure communication between host side and AI accelerators. Also, the module cooperates with host secure communication module to bridge the host HMEE and the AI accelerator HMEE.

- Coordination: An AI accelerator secure communication module should communicate with AI accelerator HMEE security function module through <N13> and host secure communication module through <N14>.

B.3 Security components in basic software layer

B.3.1 Integrity protection module

Integrity protection module should utilize cryptographic methods and underlying hardware to implement integrity protection for other components and important AI assets. The functionality and the cooperation with other security components are described as follows:

- Functionality: An integrity protection module should provide integrity protection for AI applications and other security components in basic software layer and upper AI application enabling layer. This module should utilize cryptographic algorithms like hash and digital signature to guarantee the integrity and authority of program codes and AI assets. It should also work in coordination with security components in hardware layer as needed to utilize hardware-based integrity protection.

- Coordination: An integrity protection module should communicate with a model BoM module through <N15>.

B.3.2 Trust measurement module

Trust measurement module should enable the capabilities of host HBRT and AI accelerator HBRT. The functionality and the cooperation with other security components are described as follows:

- Functionality: A trust measurement module should utilize the security capability of underlying HBRT. In specific, this module should provide an interface for upper applications and OS in an AI computing platform to send measurement values to a HBRT for security storage and fulfils remote attestation procedure with HBRT.

- Coordination: A trust measurement module should communicate with a host HBRT through <N9> and an AI accelerator HBRT through <N10>.

B.3.3 System abnormality detection module

System abnormality detection module is usually deployed upon the host OS of an AI computing platform and should monitor the system state and operation condition of an AI computing platform. The functionality and the cooperation with other security components are described as follows:

- Functionality: A system abnormality detection module should act as a monitor to estimate whether an AI computing platform works as expected and to identify whether an AI computing platform is being or has been attacked. This module should collect hardware monitoring information from hardware abnormality detection module and should monitor OS and other software in basic software layer. Based on monitoring information, this module should identify whether the system has been or is being attacked. When malfunction or errors in the AI computing platform happens, this module should timely detect these abnormalities and alert relevant security components via certain interfaces for a quick response. To cooperate with other system components, it should provide interface for AI applications or relevant security components to retrieve the real-time state and operation statistics of an AI computing platform.

- Coordination: A system abnormality detection module should communicate with a hardware abnormality detection module through <N1> and a training procedure recovery module through <N2>.

B.4 Security components in application enabling layer

B.4.1 Security management module

Security management module should provide management and configuration of related security services in an AI computing platform. It should provide interfaces for administrators to manage security functions and services in terms of security parameter configuration and sensitive information management like cryptographic keys. The functionality and the cooperation with other security components are described as follows:

- Functionality: A security management module should act as an interface for platform users to manage the security services of an AI computing platform. This module should provide interfaces to create, delete and configure the isolation instance of an AI accelerator, to create, delete and configure AI confidential computing environment of the platform, and to execute key management for the encryption/decryption service including the interfaces for different tenants to manage their own keys containing the operations for creation, deletion or authorization to other users. It should communicate with security functions in an HMEE for confidential computing environment management, key protection and utilization hardware security capabilities. This module should also communicate with the encryption/decryption module and an AI accelerator HMEE security function module on a remote AI computing platform utilizing secure communication protocol like TLS.

  EXAMPLE: The communication between security function in HMEE and security management module may be implemented with the help of a specific HMEE agent to bridge the communication between security functions in HMEE and security management module, as some HMEE techniques do not permit the direct communication with the components outside of HMEE.

- Coordination: A security management module should communicate with a hardware isolation module through <N3>, an encryption/decryption module through <N4>, an AI accelerator HMEE security function through <N5>.

B.4.2 Inference attack detection engine

An inference attack detection engine should provide a model-agnostic detection service for AI application. The functionality and the cooperation with other security components are described as follows:

- Functionality: An inference attack detection engine should analyse inference queries and related sample sent to inference models and judge whether queries are malicious or samples are adversarial. This engine should include a detection model for queries analysis and inference sample analysis. The engine should be able to change detection models for different detection tasks according to different threats in application scenarios. It should provide an interface for AI application to invoke inference attack detection service to protect models deployed on AI computing platform in inference stage.
- Coordination: None.

B.4.3 Encryption/decryption module

Encryption/decryption module should implement a cryptographic process for AI assets. It should protect the confidentiality of AI assets. The functionality and the cooperation with other security components are described as follows:

- Functionality: An encryption/decryption module should utilize predefined keys and certain cryptographic algorithms to encrypt AI assets and, on the contrary, decrypt AI assets with the appropriate key when authorized users need to use them. In an AI computing platform, this module should communicate with a security management module to acquire available cryptographic keys for encryption and decryption. For platform users, this module should provide interfaces for users to send AI assets into this module for encryption and retrieve AI assets from the module. It should be designed to automatically implement cryptographic procedures upon the cooperation with a security management module according to the user identity.

NOTE: In the scenario of implementing automatic encryption/decryption, the platform users do not need to consider the relevant keys during the cryptographic computation and only need to store assets and load assets as common practice, which facilitates the integration of an encryption/decryption mechanism into an existing AI application logic.

- Coordination: An encryption/decryption module should communicate with a security management module through <N4>.

B.4.4 Training procedure recovery module

Training procedure recovery module should be responsible for quick recovery of model training procedure. This module should resume the training at the very time point when the system is attacked or goes wrong accidently. The functionality and the cooperation with other security components are described as follows:

- Functionality: A training procedure recovery module should first identify training procedure abnormality, and timely back up execution context and important training data. The module should identify an abnormality through two ways: firstly, it monitors the training process and identify whether the training software functions as expected; Secondly, this module receives the abnormality alerts from underlying abnormality detection modules via interfaces. Last but not the least, this module should restart the training procedure with backup data automatically if the abnormality resolved or sends alerts to platform users for human intervention.

NOTE: The automatic recovery can be implemented in different ways. Redundant GPU/NPU device scheduling and reallocation can be a practical countermeasure.

- Coordination: A trust measurement module should communicate with a system abnormality detection module through <N2>.

B.4.5 Log protection module

Log protection module should provide a log protection service for AI applications running on an AI computing platform. It should collect AI-relevant logs from AI applications and protect the integrity, authority and original sequence of these logs. The functionality and the cooperation with other security components are described as follows:

- Functionality: A log protection module should utilize cryptographic methods and underlying hardware to protect collected logs generated from AI applications and AI computing platform users. This module should utilize cryptographic methods like hash algorithm and Merkle tree structure to record the evidence of correct logs. It should provide interfaces for AI applications and relevant platform users to send logs into this module. This module should provide another interface for the verification of logs in terms of correct integrity, authority and sequence. When a bunch of logs needs to be verified in scenarios like audit or forensic, this module should verify the correctness of these logs and provide the conclusion to users via an interface.
- Coordination: None.

B.4.6 Model BoM module

Model BoM module should provide model BoM service for AI assets processed or stored in an AI computing platform. The functionality and the cooperation with other security components are described as follows:

- Functionality: A model BoM module should utilize cryptographic methods and possibly underlying hardware to produce some certain proof for model BoM information so that integrity and authenticity of the information can be protected and verified. It should provide an interface for receiving information about model from stakeholders or AI applications during model training that needs to be protected in terms of integrity and authenticity. And the module should provide interfaces for platform users to acquire proof manifest for certain information about model with the integrity and authenticity guaranteed. The key used in the method should be protect by hardware-based methods utilizing the components in the hardware layer of the platform.
- Coordination: A model BoM module should communicate with an integrity protection module through <N15>.

B.5 Composition of a typical AI computing platform

Composition of a typical AI computing platform is shown in Figure B.2:

a) **Computing resources:** Physical computing resources and virtual computing resources are included. Physical computing resources can be a single AI server or a cluster consisting of multiple AI servers. Virtual computing resources are logical computing resources that are abstracted from physical computing resources and make it agnostic to underlying hardware architecture difference. including VMs and containers.

b) **Scheduling components:** The resource scheduling components and task scheduling components are included.

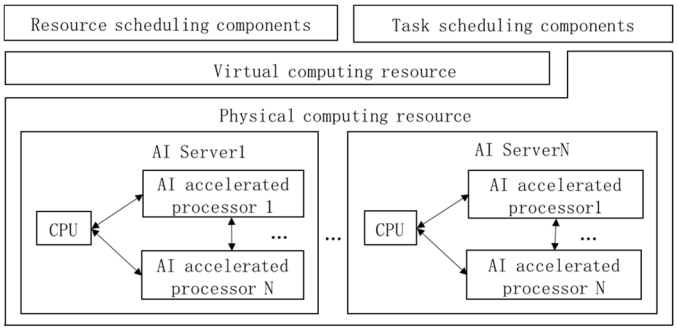

Figure B.2: Composition of a typical AI computing platform

The present document focuses on AI computing platforms that are deployed in data centres or edge computing environments. AI computing platforms in these environments have more resources (continuous computing power and energy supplement) for computing tasks, and for constructing security capabilities as well. In addition, these environments can involve more complex scenarios like ones supporting multi-tenant settings and remote access usage which introduce specific threats and may need corresponding mitigations.

Other types of platforms like mobile phones or embedded devices capable of executing AI functionality may also refer to this security framework and make some tailoring according to their specific conditions, resource constraint and security requirements.

Annex C (informative): Security mechanisms in the framework

C.1 Overview

This annex provides detailed descriptions of the security mechanisms that fulfil the security capabilities and/or security services described in clauses 5.1 and 5.2. The description involves security components and corresponding coordination, interaction information, sequence diagrams, interactive interface between users and AI computing platform, etc.

C.2 AI assets encryption/decryption

See ETSI GR SAI 009 [i.1], clause 9.2 for more information.

C.3 AI confidential computing

C.3.1 Mechanism description

AI confidential computing mechanism should be executed to fulfil run-time protection service. An interface <S4> should be provided for administrators to manage the AI confidential computing environment.

C.3.2 Involved security components

The mechanism should need the following components:

- {Host HMEE security function module} described in ETSI GR SAI 009 [i.1], clause B.2.8.
- {AI accelerator HMEE security function module} described in ETSI GR SAI 009 [i.1], clause B.2.9.
- {Host secure communication module} described in ETSI GR SAI 009 [i.1], clause B.2.12.
- {AI accelerator secure communication module} described in ETSI GR SAI 009 [i.1], clause B.2.13.
- {security management module} described in ETSI GR SAI 009 [i.1], clause B.4.1.

C.3.3 Reference point and service-based interface

The mechanism should provide the following reference points and service-based interfaces:

- Reference point: <N11>
- Reference point: <N12>
- Reference point: <N13>
- Reference point: <N14>
- Service-based interface: <S4>

C.3.4 Mechanism procedure

This mechanism should include two procedures, i.e. AI confidential computing environment establishment and AI confidential computing communication.

For AI confidential computing environment establishment, as illustrated in Figure C.1, the detailed procedure should be executed as follows:

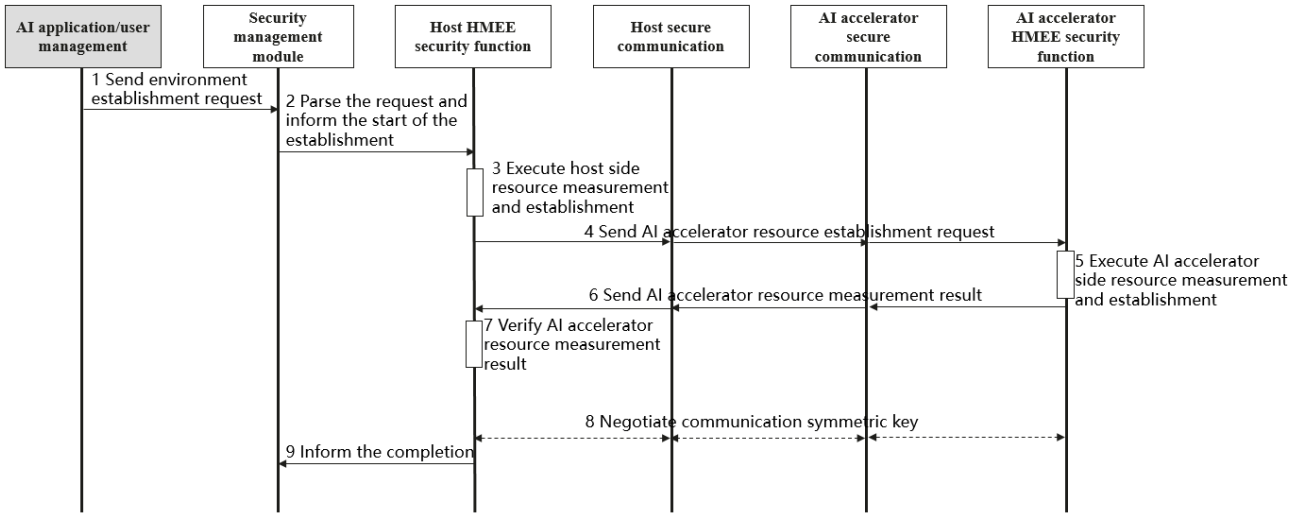

Figure C.1: Mechanism sequence diagram for AI confidential computing environment establishment

- Step 1: Administrator sends a request for establishment of new AI confidential computing environment.
- Step 2: Upon receiving the request, {Security management module} starts the establishment of the environment via <N11>. Usually, several prerequisites should be firstly checked by {Security management module} before the establishment.

EXAMPLE 1: Prerequisites include whether residual resource can satisfy the expected quota in request, and whether the administrator has the permission to execute the operation.

- Step 3: {Host HMEE security function module} measures the resource and environment on host side and establishes the environment on host side. The measurement may utilize existing security methods to verify whether the environment is trusted.
- Step 4: After the establishment of the environment on host side, {Host HMEE security function module} sends AI accelerator resource establishment request to {AI accelerator HMEE security function module} with the help of secure communication module on each side. <N12>, <N13> and <N14> should be utilized to implement the communication.
- Step 5: Upon receiving the request, {AI accelerator HMEE security function module} measures the resource and environment on AI accelerator side and establishes the environment.
- Step 6: {AI accelerator HMEE security function} sends the measurement result to {Host HMEE security function module} with the help of secure communication module on each side. <N12>, <N13> and <N14> should be utilized to implement the communication.
- Step 7: {Host HMEE security function module} receives the result and verify the measurement. The verification can utilize existing attestation methods like local attestation and remote attestation.

- Step 8 (optional): If security policies in the request of Step 1 indicate that the communication channel between host and AI accelerator needs cryptographic protection, {Host HMEE security function module} and {AI accelerator HMEE security function} should negotiate communication symmetric key used for communication encryption/decryption with the help of secure communication module on each side. <N12>, <N13> and <N14> should be utilized to implement the communication. Several key exchange protocol can be used in this procedure like Diffie-Hellman key exchange protocol. The symmetric key should be stored and managed by {Host secure communication module} and {AI accelerator secure communication module} on each side for following communication.

- Step 9: {Host HMEE security function module} informs {Security management module} the completion of the establishment.

For AI confidential computing communication, as illustrated in Figure C.2 as an example where host side environment acts as sender while AI accelerator side environment as receiver, the detailed procedure should be executed as follows:

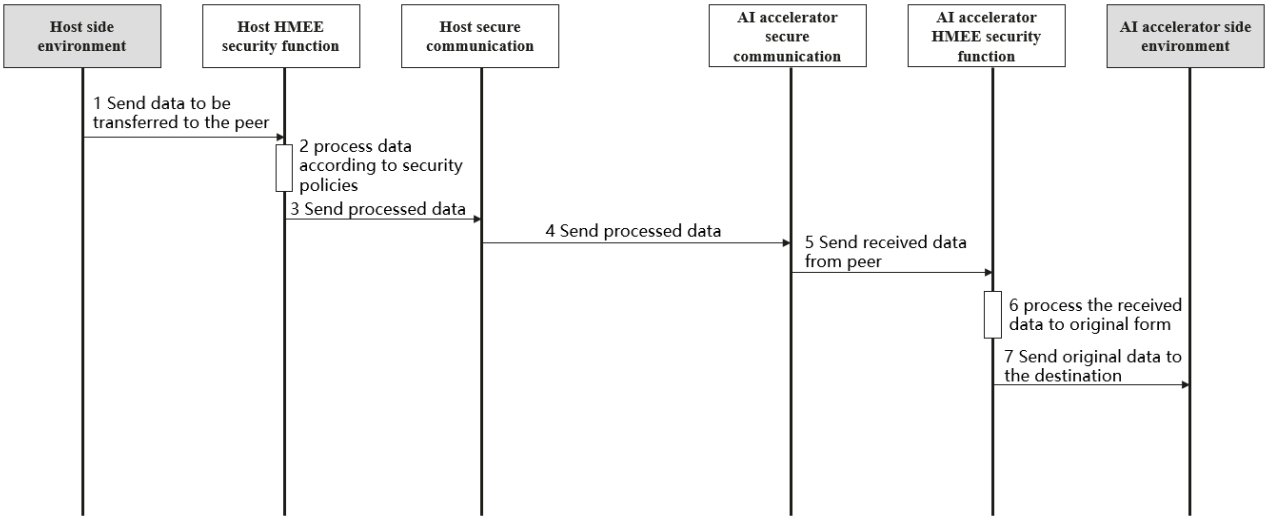

Figure C.2: Mechanism sequence diagram for AI confidential computing communication with example from host side to AI accelerator side

- Step 1: The application in host side environment of AI confidential computing environment sends data for AI related computation acceleration.

- Step 2: {Host HMEE security function module} receives the data and process the data according to security policies of the environment.

EXAMPLE 2: Encryption of the data or setting certain flags about the data for transmission isolation are two typical methods for processing.

- Step 3: After the processing, {Host HMEE security function module} sends the processed data to {Host secure communication module} for transmission.

- Step 4: {Host secure communication module} receives the processed data and sends the processed data to the peer {AI accelerator secure communication module}.

- Step 5: {AI accelerator secure communication module} receives the data and processes the data to the original form, then sends the original data to {AI accelerator HMEE security function module}.

- Step 7: {AI accelerator HMEE security function module} sends the original data to AI accelerator side environment for AI related computation.

NOTE: The scenario where AI accelerator side environment acts as sender while host side environment as receiver also fits this diagram in the same manner with the opposite direction.

C.4 AI accelerator resource isolation

See ETSI GR SAI 009 [i.1], clause 9.3 for more information.

C.5 Training procedure recovery

See ETSI GR SAI 009 [i.1], clause 9.5 for more information.

C.6 Inference attack detection

See ETSI GR SAI 009 [i.1], clause 9.6 for more information.

C.7 AI related log protection

See ETSI GR SAI 009 [i.1], clause 9.4 for more information.

C.8 Model BoM proof mechanism

C.8.1 Mechanism overview

Model BoM proof mechanism should be executed to fulfil model BoM service. An interface <S7> should be provided for AI application to invoke the service.

C.8.2 Involved security components

The mechanism should need the following security components:

- {Integrity protection module} described in clause B.3.1.
- {Model BoM module} described in clause B.4.6.

C.8.3 Reference point and service-based interface

The mechanism should provide the following reference point and service-based interface:

- Reference point: <N15>
- Service-based interface: <S7>

C.8.4 Mechanism procedure

Illustrated in Figure C.3, the detailed procedure for Model BoM proof mechanism should be executed as follows:

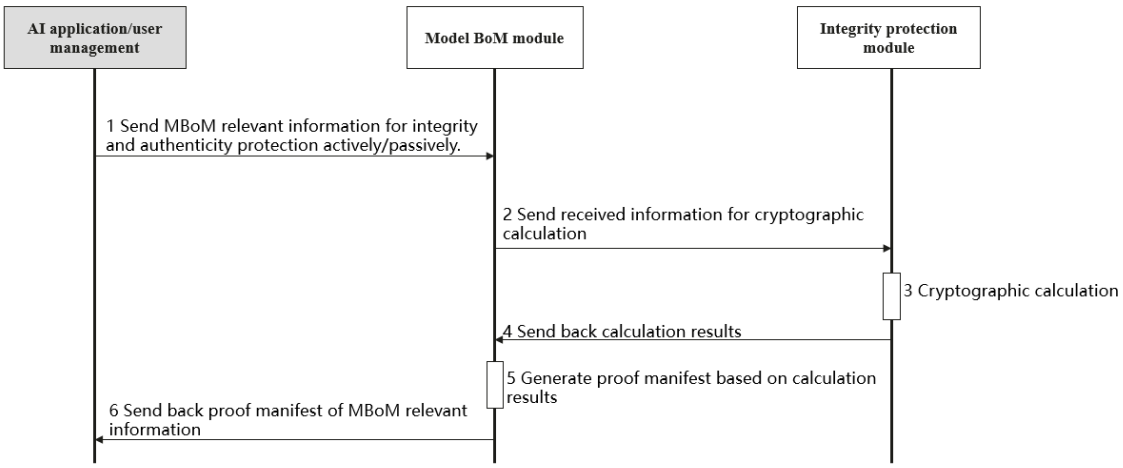

Figure C.3: Mechanism sequence diagram for Model BoM proof mechanism

- Step 1: An AI application and related components send Model BoM relevant information to {Model BoM module} actively or passively via <S7> for integrity and authenticity protection and proof manifest of the information.

NOTE: AI application and related components can send Model BoM information on their demand actively or passively based on the request from {Model BoM module}.

- Step 2: {Model BoM module} receives the information and sends the information to {Integrity protection module} via <N15> for cryptographic calculation on the information to obtain hash value or signatures.
- Step 3: {Integrity protection module} receives the information and execute the cryptographic calculation according to certain algorithm.
- Step 4: {Integrity protection module} sends back the calculation results to {Model BoM module} via <N15>.
- Step 5: {Model BoM module} receives the results and generates proof manifest based on these calculation results.
- Step 6: {Model BoM module} sends back the proof manifest to the calling AI application and related components.

C.9 Measured boot

See ETSI GR SAI 009 [i.1], clause 9.7 for more information.

C.10 Recovery from minimal system

See ETSI GR SAI 009 [i.1], clause 9.8 for more information.

Annex D (informative): Implementation reference for security mechanism of AI computing platform

D.1 Overview

In the present annex reference deployment implementations for security mechanism defined in the main body of the present document are described in detail. Each implementation reference introduces how relevant security components in a certain mechanism are realized in a practical AI computing platform and how interfaces are utilized by platform users, AI application or other components outside the scope of the security framework.

D.2 AI assets encryption/decryption mechanism

See ETSI GR SAI 009 [i.1], clause 9.2.5 for more information.

D.3 AI confidential computing mechanism

The reference deployment of the AI confidential computing mechanism is illustrated in Figure D.1 as an implementation example to better understand the mechanism and its utilization. In an AI system, users can integrate AI confidential computing management function in their own management software or portal by invoking <S4> when developing code logic of their management software. When platform user utilizes AI confidential computing mechanism to protect AI assets in run-time, {Host secure communication module} and {AI accelerator secure communication module} need to be deployed together on host side and AI accelerator respectively. Two communication modules are used to connect a combined confidential computing environment. This combined confidential computing environment consists of host isolated environment, AI accelerator isolated environment and the protected communication channel between two isolated environments. An AI application will mainly be deployed in host side environment and run its logic and send data to AI accelerator environment for computing acceleration with the computing resource in the environment. In this case, only the owner of the combined environment can access and use it while other platform users including root administrator of platform OS and hypervisor administrator cannot access the environment due to the hardware-enabled isolation on resources and connection between host and AI accelerators.

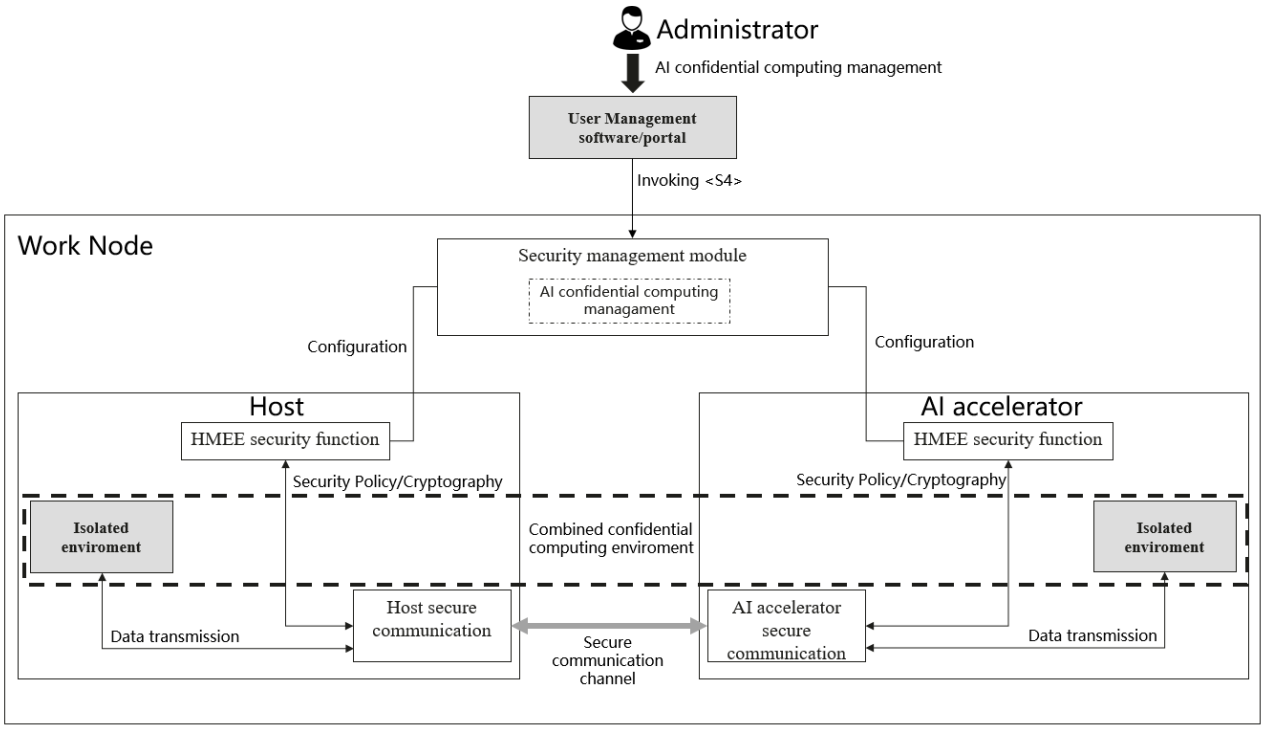

Figure D.1: Diagram of reference deployment model for AI confidential computing mechanism

The mentioned implementation example only reflects one typical mode of AI confidential computing mechanism. There are also some other modes for this mechanism as shown in Figure D.2. {Host secure communication module} and {AI accelerator secure communication module} are each host side and each AI accelerator will handle the connection between their own environments. {Host HMEE security function module} and {AI accelerator security function module} needs to configure the policies for the corresponding secure communication module to realize the connection protection and control via <N12> and <N13>. The connection between corresponding {Host secure communication module} and {AI accelerator secure communication module} in the same combined confidential computing environment should utilize the same isolation policies or same cryptographic keys.

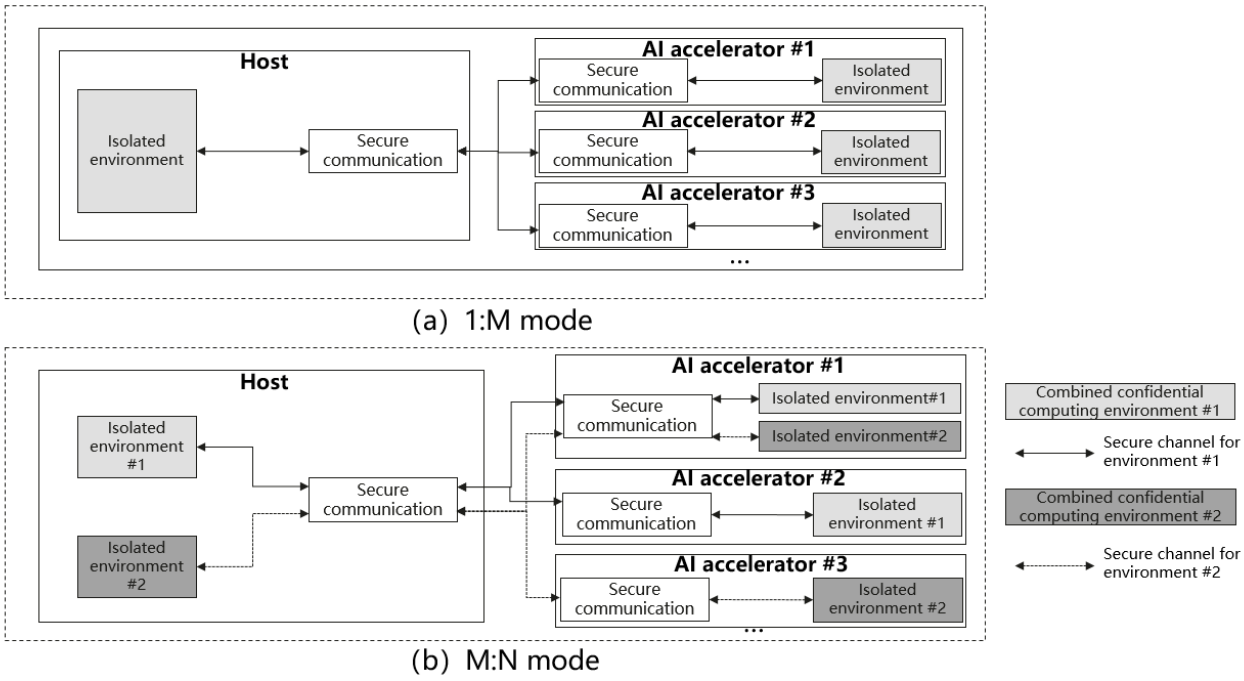

Figure D.2: Other modes of AI confidential computing mechanism

D.4 AI accelerator resource isolation mechanism

See ETSI GR SAI 009 [i.1], clause 9.3.5 for more information.

D.5 AI related log protection mechanism

See ETSI GR SAI 009 [i.1], clause 9.4.5 for more information.

D.6 Inference attack detection mechanism

See ETSI GR SAI 009 [i.1], clause 9.6.5 for more information.

D.7 Model BoM proof mechanism

The reference deployment of Model BoM proof mechanism is illustrated in Figure D.3 as implementation example to better understand the mechanism and its utilization. In this example, AI application for training and testing of a commercial AI model needs many dependencies to fulfil the tasks. In this example, the dependencies for training process include pre-trained model information, training dataset information, training sever information and so on. While the dependencies for test process may include test performance metrics, test dataset information, test organization that executes the test and so on. The application can invoke <S7> provided by {Model BoM module} to require proof manifest for all these dependencies to keep record of these dependencies and guarantee the authenticity and integrity of these records. When supervisors plan to investigate a certain model or AI applications, Model BoM inventory will provide sufficient and detailed information for this model or AI application, the integrity and authenticity of which can be guaranteed by cryptographic methods. In addition, {Integrity protection module} can further utilize hardware security components described in clause B.2 to enhance the security of this mechanism. One possible example is to utilize HMEE components for key management and digital signature calculation.

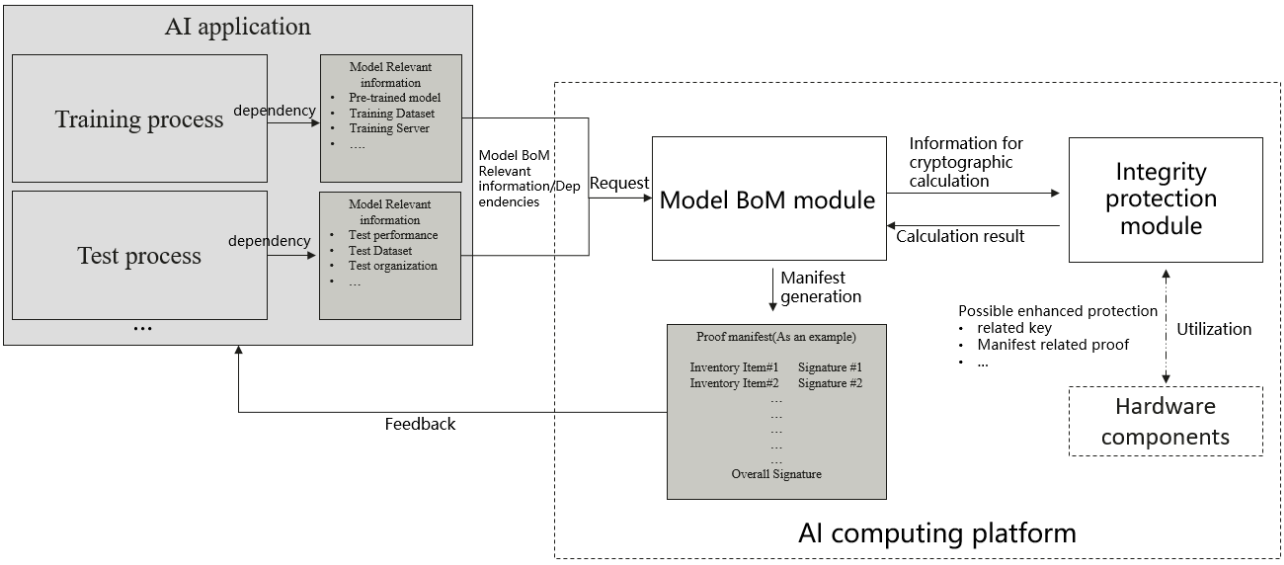

Figure D.3: Diagram of two reference deployment model

D.8 Recovery from minimal system

The reference deployment of recovery mechanism is illustrated in Figure D.4 as implementation example to better understand the mechanism and its utilization. In this example, management software provides the system recovery function by utilizing {minimal system} via <S6> to invoke the recovery capabilities provided by AI computing platform. With the help of this function, platform users can easily and quickly recover the platform to a predefined initial state via a user portal of management software in the situation that the platform operates abnormally or is being attacked. A recommended model for the deployment of recovery is to utilize a Baseboard Management Controller (BMC) chip as the {minimal system} and integrate BMC interface into the device management portal for recovery operation.

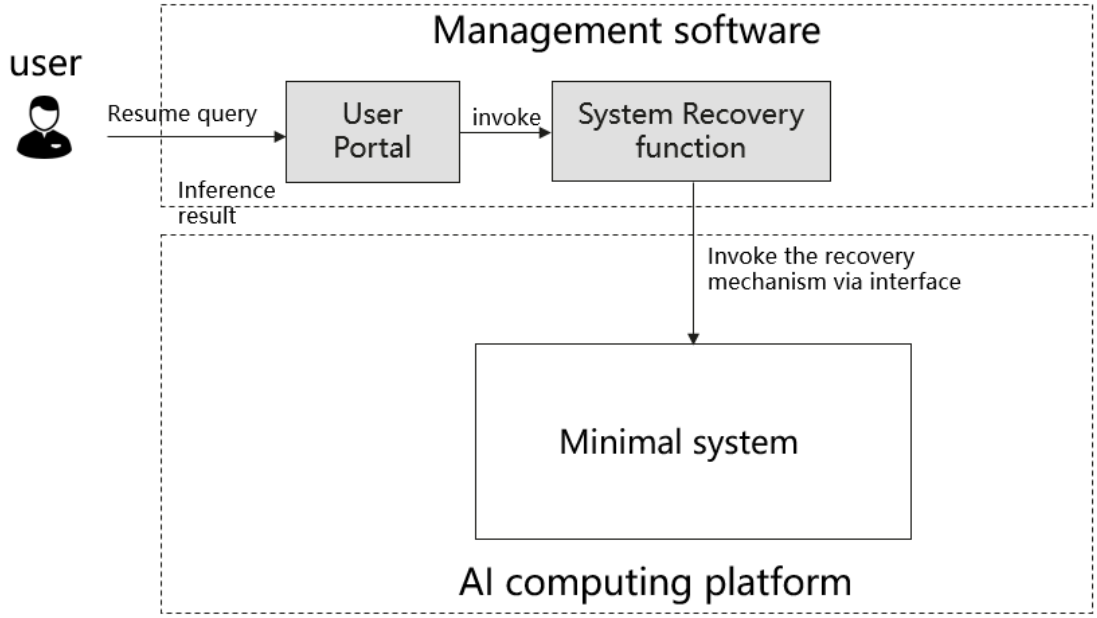

Figure D.4: Reference deployment model of recovery from minimal system

Annex E (informative): Model BoM overview

E.1 Model BoM description

Corresponding to software BoM, AI model, as a critical asset in AI system, also needs to utilize some kinds of methods for the recording and presentation of model dependency list to ensure:

- Model consistency.
- Integrity.
- Traceability.

By providing a list of components, sources, and dependencies for supporting:

- An accurate and detailed list of model dependencies.
- Model compliance management.
- Model vulnerability awareness.
- Model tracing.

To this end, Model Bill of Materials (Model BoM or AI BoM) can be utilized. Model BoM, like the definition of software BoM, is the nested inventory, a list of ingredients and dependencies that are relevant during the model training procedure. The inventory of Model BoM includes the information that are specific to AI scenarios that is not covered by the current format of software BoM inventory. The following items are some of the possible examples that are exclusively be presented in model BoM:

- Training data set and testing data set.
- Pretrained model information.
- Model structure information.
- Model related properties like inference accuracy.

The conceptual architecture of Model BoM is illustrated in Figure E.1. With this inventory, stakeholders can fulfil the following goals:

- Enhancing the model traceability by describe the detailed building process and dependencies of the model in a structured manner in its training.
- Checking consistency and integrity of AI model before deployment.
- Usage control based on Model BoM information and security policy before deployment.
- Auditing and tracing the dependencies and processes of the model for clarifying responsibilities and forensic.

NOTE 1: The conceptual architecture refers to OWASP CycloneDX specification v1.7 [i.2] which has been published as a subset of software BoM. Also, Linux® SPDX 3.0 specification will include Model BoM.

NOTE 2: Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

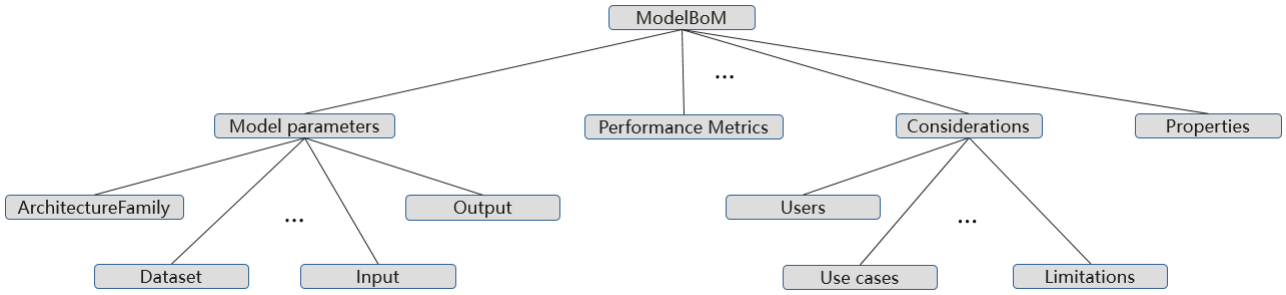

Figure E.1: Conceptual architecture of Model BoM

NOTE 3: The detailed structure of each item in nested inventory of Model BoM needs standardization. However, it is not in the scope of the present document.

E.2 Model BoM Threats and its mitigations

Model BoM will surely improve the transparency and traceability of a model deployed in an AI system. However, what contained in Model BoM inventory can also expose important information about this model that can be utilized by adversaries to implement attack to AI model itself or the whole system. The following are the negative effect after introducing AI BoM:

- Attackers can utilize the information in Model BoM for analysing the attack surface of AI Model as well as AI system.
- Attackers can shorten the time duration and lower the cost of attack with the help of the information exposed by Model BoM. That is, attackers can implement specific attack for a certain model based on the relevant information exposed by Model BoM.
- Attackers can manipulate the content of Model BoM inventory to mislead or commit fraud to anyone who will refer to Model BoM for management, maintenance or other operations.

As a result, Model BoM needs to mitigate the risks introduced by information exposure before it can be implemented in real world AI system. The following principles and best practices can be applied:

- The items in the Model BoM should be designed according to the requirements in application scenarios. The information in the inventory should be carefully selected as a trade-off between transparency and security. Suitability is the best principle rather than comprehensiveness.
- The item in the Model BoM can be categorized into public accessible class and private class. The items in public accessible class can be accessed by anyone publicly as the item will not expose key information that will affect the security of AI model or AI system. The items belonging to private class contain important information that may introduce risks. Private class should be accessed under strict control based on access control mechanism. Besides, information in these private items should be protected by hardware or cryptographic methods to protect the confidentiality.
- The function described in clause B.2.2 should be utilized to protect the integrity of Model BoM.

Annex F (normative): Mapping to baseline requirements of ETSI EN 304 223

ETSI EN 304 223 [i.7] defines baseline security requirements for AI models and systems. Some of the items from ETSI EN 304 223 [i.7] can be mapped into the present document for clarifying the roles of AI computing platform and give a reference on how it shall behave in AI system.

The mapping relationship from ETSI EN 304 223 [i.7] to security requirements for an AI Computing Platform is listed in Table F.1.

Table F.1: Mapping relationship from ETSI EN 304 223 [i.7] to the AI Computing platform

Reference items	Mapping items	Detail
4.0 Stakeholders		
Table 4-1 System Operators	The role of AI computing platform	According to the definitions of Table 4-1, the role of the AI computing platform can be mapped as system operators.
5.1 Secure Design		
Provision 5.1.2-2	5.1.6	To withstand adversarial AI attacks and unexpected inputs, the AI computing platform should comply with the requirements of secure response.
	5.1.7	To withstand AI system failure, the AI computing platform should comply with the requirements of resilience.
	5.2.5	To withstand AI system failure, the AI computing platform should comply with the requirements of training procedure recovery.
	5.2.6	To withstand AI system failure, the AI computing platform should comply with the requirements of inference attack detection service.
Provision 5.1.2-3	5.1.5	To document and create an audit trail in relation to the AI system, the AI computing platform should comply with the requirements of secure audit.
	5.2.7	To create an audit trail in relation to the AI system, the AI computing platform should comply with the requirements of AI related log storage and transfer.
	5.2.8	To ensure that the audit trail including the operation, and lifecycle management of models, datasets and prompts, the AI computing platform should comply with the requirements of Model BoM service.
5.2 Secure Development		
Provision 5.2.1-1	5.2.8	To satisfy the provisions, the AI computing platform should comply with the requirements of Model BoM service.
Provision 5.2.1-2		
Provision 5.2.3-1		
Provision 5.2.1-3	5.2.5	To withstand disaster recovery, the AI computing platform should comply with the requirements of training procedure recovery.
Provision 5.2.1-3.1		
Provision 5.2.1-4	5.2.2 5.2.3	To protect sensitive data, the AI computing platform should comply with the requirements of AI assets protection in transmission, storage and processing.
Provision 5.2.1-4.1		
Provision 5.2.2-1	5.1.2 5.2.4	To satisfy the provisions, the AI computing platform should comply with the requirements of identity management and access control and AI accelerator resource isolation.
Provision 5.2.2-3		
5.3 Secure deployment		
Provision 5.3.1-1	See note	The deployed AI system when using an AI computing platform as specified in the present document reinforces the choices made during the secure development phase and alongside the provisions for secure maintenance give assurance that the AI system operates as designed.
Provision 5.3.1-2		
Provision 5.3.1-2.1		
Provision 5.3.1-2.2		
Provision 5.3.1-3		

Reference items	Mapping items	Detail
5.4 Secure Maintenance		
Provision 5.4.2-1	5.1.5	To log system and user actions, the AI computing platform should comply with the requirements of secure audit.
Provision 5.4.2-4	5.2.6	To detect sudden or gradual changes in behaviour, the AI computing platform should comply with the requirements of inference attack detection service.
5.5 Secure End of Life		
Provision 5.5.1-1	5.2.8	To satisfy the provisions, the AI computing platform should comply with the requirements of Model BoM service for AI model tracing.
NOTE: The present document provides provisions that are enabled in deployment of any AI system and thus all the provisions of the present document apply to the active deployment of the AI system.		

Annex G (informative): Bibliography

- ETSI GS NFV-SEC 012: "Network Functions Virtualisation (NFV); Security; System architecture specification for execution of sensitive NFV components".
- ETSI GS NFV-SEC 009: "Network Functions Virtualisation (NFV); NFV Security; Report on use cases and technical approaches for multi-layer host administration".

Annex H (informative): Change history

Date	Version	Information about changes
December 2023	V0.0.1	Convert from GS-015 v0.0.6 to TS format
March 2024	V0.0.2	Add Table of content and new LLM-specific requirement in clause 6.2
April 2024	V0.0.3	Update Figure 3. Add sub-clause 7.12 and 7.13 in clause 7, sub-clause 9.3 in clause 9 and A.3 in Annex A
August 2024	V0.0.4	Change the title and scope. Add Annex A for AI computing platform description. Several requirement adjustments to focus on AI-specific security requirement
November 2025	V0.0.7	Revise the editorial errors. Adjust the structure of it including rearranging clause 7 at clause 6.5, clause 8 at Annex C and Annex D at Annex E. Add Annex D for Implementation reference for security mechanism of AI computing platform. Replace the figures 4 and 6.
January 2026	V0.0.8	Revise the provisions of subclause 5.2.7. Move clause 6 to Annex B. Add Annex F for mapping from EN 304 223.
March 2026	V0.0.11	Replace the "shall" to "should" in Annex B. Add "end of life" in Annex F for mapping from EN 304 223. Add an item to prevent injection attack at clause 5.2.3.
April 2026	V0.0.12	Fixed grammar and spelling mistakes throughout, and supplemented abbreviations. Add "secure deployment" in Annex F for mapping from EN 304 223.
April 2026	V0.0.14	Make editorial changes, and move 2 references to Annex G Bibliography. Change the reference version of OWASP: "CycloneDX" from 1.5 to 1.7.

History

Version	Date	Status
V1.1.1	May 2026	Publication