Confidential Computing Consortium response to UK Government's "Secure AI infrastructure: call for information"

https://www.gov.uk/government/publications/secure-ai-infrastructure-call-for-information/secure-ai-infrastructure-call-for-information

To: Department for Science, Innovation and Technology (DSIT)

From: Confidential Computing Consortium

Date: 27 February 2026

Subject: Response to Call for Information: Secure AI Infrastructure

Executive Summary

The Confidential Computing Consortium (CCC) welcomes the opportunity to contribute to the UK Government's call for information on securing AI infrastructure. As AI systems become foundational to national security, public services, and economic growth, the threat landscape is expanding beyond traditional cyber risks to include model theft, data poisoning, supply chain compromise, and autonomous agent manipulation.

A resilient AI security framework must move beyond perimeter-based controls and toward architectures rooted in hardware-enforced trust. **Confidential Computing** provides a foundational layer that protects data and models while in use, enables cryptographic attestation of runtime integrity, and supports policy-driven control over sensitive workloads. This response outlines key risk areas—including model extraction, agent compromise, and infrastructure tampering—and proposes a cohesive set of capabilities, from Trusted Execution Environments (TEEs) to Post-Quantum Cryptography (PQC), that collectively establish verifiable trust in AI systems. By embedding hardware-based assurance into AI infrastructure, the UK can strengthen sovereignty, protect innovation, and position itself as a global leader in secure and trustworthy AI deployment.

The Security Challenge and Current Approaches

Assessment of Risks

We assess the primary risks to AI infrastructure not merely as traditional data theft, but as

sophisticated attacks targeting the unique intellectual property and operational integrity of AI systems.Current mitigations include encryption at rest and in transit, IAM and segmentation controls, monitoring and anomaly detection, secure MLOps pipelines, and increasingly Confidential Computing with remote attestation and key release controls. However, these approaches have limitations: traditional controls do not protect data in use; privileged infrastructure access remains a structural risk; API-based extraction cannot be fully prevented; and runtime isolation does not address logical poisoning or adversarial manipulation of agent memory.

- **Model Extraction (Theft of IP):** The risk of unauthorised exfiltration of proprietary model weights is critical. Attackers can extract model weights through a number of channels. Some channels such as abusing API calls are simple but slow, while others like directly dumping memory provide perfect copies of model weights.
- **The Trust Gap:** A critical vulnerability is the "trust gap" between the model owner and the infrastructure provider. Standard cloud security protects against external attackers but often leaves model weights accessible to the cloud provider's hypervisor or privileged administrators. For truly secure AI, the threat model must assume the infrastructure itself is untrusted.
- ***Memory Poisoning*:** Attackers may inject malicious instructions into an agent's long-term memory, which are later triggered to perform unauthorised or harmful actions. The temporal separation between injection and execution makes detection through conventional monitoring extremely difficult**.**
- ***Data Exfiltration*:** Agents often process sensitive information, including credentials, cryptographic material, and identity data. Compromise at the infrastructure layer may allow silent extraction of this information and as accelerator paths may not be fully protected, secrets may leak through logs, debugging, telemetry**.** Current mitigations are contractual controls, Cloud Provider certifications etc. These are governance assurances not cryptographic guarantees that can provide verifiable proof of runtime guarantees.
- **Autonomous Agent Risks:** As AI evolves into autonomous agents, new threats emerge.
    - **Memory Poisoning:** attackers may inject malicious instructions into an agent's long-term memory, which are later triggered to perform unauthorised or harmful actions. The temporal separation between injection and execution makes detection through conventional monitoring extremely difficult.
    - **Data Exfiltration:** agents often process sensitive information, including credentials, cryptographic material, and identity data. Compromise at the infrastructure layer may allow silent extraction of this information.
- **Privacy-Specific Threats related to Context & Memory Scraping.** Unlike traditional software where data is transient, AI agents maintain long-running "context windows" or memories containing sensitive user data (PII) and retrieval results. A key vulnerability is the potential for **memory scraping** or **cold boot attacks** by a malicious host or hypervisor to extract this unencrypted context from RAM or system hardware.

Current Mitigations and Limitations

Current mitigations include encryption at rest and encryption in transit, IAM and segmentation controls, monitoring and anomaly detection, secure MLOps pipelines, and increasingly Confidential Computing with remote attestation and key release controls. However, these approaches have limitations: traditional controls do not protect data in use; privileged infrastructure access remains a structural risk; API-based extraction cannot be fully prevented; and runtime isolation does not address logical poisoning or adversarial manipulation of agent memory.

Capabilities That Could Strengthen Protection

To address the "data in use" vulnerability and the trust gap, we recommend a layered defense strategy centered on **Confidential Computing** and **Verifiable Assurance**.

Confidential Computing

- **Addresses Threat:** Model Theft & Infrastructure Trust Gap
- **Description:** Confidential Computing uses attested, hardware-based Trusted Execution Environments (TEEs) to isolate the AI workload. This memory is protected from the host OS, hypervisor, or cloud administrator, and protected by hardware (CPU TEES and increasingly accelerators with Confidential Computing capabilities). (https://www.rand.org/pubs/research_reports/RRA2849-1.html)
- **Assurance:** One approach to providing assurance is by pairing the Confidential Computing environment with a **Key Management system**. The model owner retains the encryption keys in the Hardware Security Modules. The keys are only released to the TEE after a remote attestation - providing cryptographic assurance of the correctness of TCB components and the fact that they match policy - proves the hardware and software stack are unmodified, and if attestation fails (i.e. the environment doesn't match approved policy), keys are not released. Other approaches include generating keys within the TEE and distributing them to other trusted parties (including TEEs).
- Agents making use of or being deployed across accelerators, e.g., GPUs or NPUs, in multi-tenant environments inherit and take on the attack vectors on that shared hardware. In such deployments **Confidential Computing accelerators** should be used. These devices can build secure connections with workloads deployed in Confidential Computing virtual machines.

Confidential Inference & Post-Quantum Cryptography

- **Addresses Threat:** Compromise of Sensitive User Data
- **Description:** We define **Confidential Inference** as an inference service where user

prompts remain encrypted in transit and are only decrypted inside an attested hardware TEE. The client verifies the TEE via remote attestation (including expected hardware and approved software measurements) and establishes a protected session whose keys are bound to that verified environment, preventing cloud operators or intermediaries from accessing prompt contents.
- **Future-Proofing:** To defend against "Store Now, Decrypt Later" attacks, the transport layer must utilise **Post-Quantum Cryptography (PQC)** (e.g., NIST ML-KEM). This ensures that encrypted traffic captured today cannot be decrypted by future quantum computers.

Secure Agent Execution

- **Addresses Threat:** Compromise or Abuse of Autonomous Agents
- **Description:** Agents must be treated as untrusted entities.
    - **Sandboxing and Runtime Isolation**: Agents should be deployed in sandboxing environments that limit their operational scope and abilities. Utilizing open source technologies like Kata Containers (e.g., https://agent-sandbox.sigs.k8s.io/) ensures that agent activities remain isolated from the host system.
    - **Verifiable Identity:** Each agent should have a cryptographic identity bound to its code via remote attestation of the TEE in which it is executing, preventing "rogue" agents from impersonating legitimate ones. This can also allow the operator of the system hosting the agent to prove that they have no ability to tamper or inspect the agent during execution.
    - **Memory Protection:** Defenses must include rigorous scanning of memory inputs to detect and sanitise adversarial content before it is stored, mitigating the risk of memory poisoning or memory decoying technique.

Market Viability

Target Customers

The primary customers for these enhanced solutions are organisations safeguarding high-value IP (model weights) or highly sensitive data (public sector, defense, healthcare) who require technical assurances that exceed contractual promises. Buyers are typically CISOs, Chief Data Officers, and AI platform leaders seeking hardware-backed technical assurances that exceed contractual guarantees. Adoption is increasingly driven by regulatory enforcement (e.g., DORA, EU AI Act, UK GDPR (Data Protection Act 2018)), multi-tenant accelerator risk, and the need to securely commercialise proprietary AI models across jurisdictions.

Response to Initial Research Areas

We map our recommended capabilities to the specific research areas outlined in the call:

- **Area 2: Trusted computing foundations for AI:** We view **Remote Attestation** as the

cornerstone. It allows a system to prove its security state to a remote party before any sensitive data is exchanged. Confidential Computing inherits from and improves upon Trusted Computing by minimizing the Trusted Computing Base including excluding the host operating system and privileged actors on the host.

- **Area 3: Digital rights management (DRM) for models:** We propose using **Confidential Computing** technology as a form of DRM. By binding decryption keys to a specific, attested software image, a model owner can ensure their model *only* runs in an approved, attested environment.
- **Area 4: Verifiable Confidential Computing:** Research should focus on standardising system attestation reports to make them interoperable across different hardware vendors (AMD, Intel, Arm, NVIDIA), ensuring a unified "root of trust" for the UK AI sector. Advanced open source implementations of Confidential Computing include CNCF Confidential Containers (https://confidentialcontainers.org/) which provides a purpose-built confidential virtual machine image that restricts execution to prescribed executables providing stronger assurance than general purpose confidential virtual machines.
- **Area 5: Advanced Cryptography:** Post-Quantum Cryptography is essential to ensure the long-term confidentiality and integrity of data and workloads protected by Confidential Computing. The Confidential Computing Consortium supports industry-wide migration to standardised PQC algorithms and promotes their integration into attestation, key management, and secure communication frameworks.
- **Area 8: Adversarial machine learning defence:** TEEs reduce attack surface by preventing direct memory inspection and host-level exfiltration of model parameters. However, they must be complemented by inference-time monitoring, rate limiting, and adversarial robustness techniques.

Contact Information

Submitted by: Confidential Computing Consortium (https://confidentialcomputing.io)

Point of Contact: Mike Bursell, Executive Director

Email: mbursell@contractor.linuxfoundation.org