Confidential Computing pour LLM 2026 : Enclaves TEE et Privacy

> Guide complet du confidential computing pour les LLM en 2026 : enclaves TEE Intel SGX, AMD SEV, Arm TrustZone, inférence confidentielle et conformité RGPD.

Les modèles de langage de grande taille (LLM) s'imposent en 2026 comme des outils incontournables dans les entreprises, les hôpitaux, les cabinets d'avocats et les institutions financières. Pourtant, leur adoption soulève un paradoxe fondamental : pour bénéficier de leurs capacités d'analyse et de génération, les organisations doivent confier leurs données les plus sensibles à des infrastructures cloud dont elles ne maîtrisent pas intégralement l'opération interne. Un dossier médical soumis à un LLM pour aide au diagnostic, un contrat juridique analysé pour extraire des clauses, une transaction financière passée en revue pour détecter une fraude — toutes ces opérations exposent des données à caractère personnel ou stratégique à un risque d'interception, de fuite ou d'utilisation détournée. Le Règlement Général sur la Protection des Données (RGPD) impose quant à lui des obligations strictes de minimisation des données et de sécurité du traitement. Comment concilier la puissance computationnelle du cloud avec les exigences légales et la souveraineté des données ? Le confidential computing répond précisément à ce défi en garantissant que les données restent chiffrées et inaccessibles, même lors de leur traitement actif dans la mémoire vive des serveurs. Cet article explore en profondeur les technologies TEE (Trusted Execution Environment), leur application concrète aux LLM en 2026, les architectures Intel TDX/SGX, AMD SEV-SNP, Arm CCA, ainsi que les offres cloud (Azure, AWS, GCP) qui permettent de faire tourner des modèles comme Llama en totale confidentialité.

Qu'est-ce que le Confidential Computing — TEE, enclaves et isolation hardware

Le **confidential computing** désigne un paradigme de sécurité informatique dans lequel les données sont protégées non seulement au repos (chiffrement du disque) et en transit (TLS), mais également *pendant leur traitement actif* en mémoire. C'est ce troisième état — la donnée en cours d'utilisation — qui constitue jusqu'ici le maillon faible de la chaîne de sécurité. Un administrateur système malveillant, un hyperviseur compromis, ou un attaquant ayant obtenu un accès root peut lire librement la RAM d'une machine virtuelle conventionnelle. Le confidential computing élimine cette surface d'attaque grâce à des **Trusted Execution Environments (TEE)**, des zones d'exécution isolées et protégées par le hardware lui-même.

Un TEE crée une bulle d'exécution isolée, appelée **enclave** ou **Confidential VM**, dans laquelle le code et les données sont chiffrés par des clés que seul le processeur connaît. Même le système d'exploitation hôte, l'hyperviseur ou le fournisseur cloud ne peuvent pas lire le contenu de cette mémoire protégée. L'isolation repose sur plusieurs mécanismes hardware combinés : le chiffrement mémoire en temps réel par le processeur, la gestion de clés au sein du silicium (hors de portée du firmware), et des mécanismes de contrôle d'accès aux pages mémoire.

Intel SGX (Software Guard Extensions)

Intel SGX, introduit en 2015 et toujours largement déployé en 2026, permet de créer des enclaves de quelques mégaoctets à quelques gigaoctets selon la génération. Le processeur chiffre automatiquement les pages mémoire appartenant à l'enclave via le mécanisme MEE (Memory Encryption Engine). Le code s'exécute en clair à l'intérieur du processeur mais est chiffré dès qu'il quitte le cache L3 pour rejoindre la RAM. SGX v2 (Scalable SGX) étend la taille des enclaves jusqu'à 1 To de mémoire chiffrée sur les processeurs Xeon Scalable de 3e génération et suivants, rendant possible l'exécution de modèles de taille intermédiaire.

> ⚡ **Retour terrain**
>
> Pour une banque régionale qui voulait automatiser la rédaction de ses synthèses de risque, j'ai benchmarké GPT-4o, Claude 3.5 Sonnet et Mistral Large sur un corpus de 200 notes anonymisées. La métrique critique n'était pas la précision brute mais le taux de fabrication de chiffres — seul Claude atteignait 0 % sur ce critère sur ce corpus précis. La conclusion : choisir un modèle pour une tâche critique exige des benchmarks sur vos propres données, pas sur les leaderboards publics.

Intel TDX (Trust Domain Extensions)

Intel TDX, disponible depuis les processeurs Xeon Scalable 4e génération (Sapphire Rapids), représente l'évolution naturelle de SGX vers la protection de machines virtuelles entières. Là où SGX protège des enclaves applicatives, TDX protège des **Trust Domains (TD)** — des VMs complètes dont toute la mémoire est chiffrée avec une clé propre à chaque domaine. Le module Intel TDX, composant firmware vérifié, agit comme un intermediaire entre le TD et l'hyperviseur

VMM, garantissant que ce dernier ne peut pas lire ni modifier la mémoire du domaine de confiance. TDX s'intègre naturellement aux infrastructures de virtualisation existantes (KVM, QEMU) ce qui facilite son adoption.

AMD SEV (Secure Encrypted Virtualization)

AMD propose une approche similaire avec sa famille SEV : SEV (2016), SEV-ES (Encrypted State, 2020) et SEV-SNP (Secure Nested Paging, 2021). SEV-SNP, la version la plus avancée et déployée à grande échelle en 2026, ajoute la protection d'intégrité de la mémoire en plus du chiffrement, empêchant un hyperviseur malveillant de substituer des pages mémoire chiffrées. Chaque VM obtient une clé unique générée et gérée par le Platform Security Processor (PSP) — un sous-processeur dédié à la sécurité, similaire au concept d'ARM TrustZone.

Arm CCA (Confidential Compute Architecture)

Arm CCA, introduit avec l'architecture ARMv9 en 2021 et progressivement déployé sur les serveurs basés sur Arm (AWS Graviton, Ampere Altra), définit le concept de **Realm** — un domaine d'exécution isolé similaire aux enclaves SGX mais à l'échelle d'une VM complète. Le Realm Management Monitor (RMM) s'exécute dans un niveau de privilège dédié (EL2), séparé de l'hyperviseur, et assure l'isolation entre les realms et le reste du système. Arm CCA prend une importance croissante en 2026 avec la montée en puissance des processeurs Arm dans les datacenters.

L'attestation à distance — pilier de la confiance

Au-delà du chiffrement, la valeur du confidential computing repose sur le mécanisme d'**attestation à distance**. Avant de confier des données sensibles à une enclave, le client doit pouvoir vérifier que : (1) l'enclave s'exécute bien sur un hardware TEE légitime, (2) le code qui y tourne est bien le code attendu (mesuré via son empreinte cryptographique), et (3) la configuration de sécurité est conforme. Cette vérification passe par un **rapport d'attestation** signé par la clé racine du processeur, que le client peut valider auprès du service d'attestation du fabricant (Intel DCAP/IAS, AMD KDS, Microsoft Azure Attestation). Le schéma ci-dessous illustre l'architecture complète d'un système de confidential computing pour LLM.

Pourquoi le Confidential Computing est critique pour les LLM en 2026

En 2026, les LLM ne sont plus des jouets de laboratoire mais des composants critiques des systèmes d'information d'entreprise. Cette criticité crée un ensemble de menaces nouvelles qui justifient pleinement l'adoption du confidential computing.

La menace du vol de modèle

Un LLM de qualité enterprise représente un investissement considérable : des millions d'euros en puissance de calcul pour le pré-entraînement, des dizaines de milliers d'heures humaines pour le fine-tuning et l'alignement, sans compter la valeur des données propriétaires utilisées pour l'entraînement. Or, lorsqu'un modèle s'exécute dans une infrastructure cloud conventionnelle, ses poids (les paramètres du réseau de neurones) résident en clair dans la mémoire GPU. Un attaquant ayant compromis l'hyperviseur ou disposant d'un accès physique aux serveurs peut théoriquement exfiltrer ces poids. Le vol de modèle est une menace réelle documentée dès 2023

avec des techniques d'extraction par requêtes (model stealing attacks), mais le confidential computing protège contre les attaques internes au niveau de l'infrastructure.

L'interception des prompts et des données d'inférence

Chaque requête envoyée à un LLM — le prompt — peut contenir des informations hautement confidentielles. Dans un contexte médical, le prompt peut inclure des données cliniques complètes d'un patient. Dans un contexte juridique, des détails de stratégie contentieuse. Dans un contexte financier, des positions de trading ou des plans de fusion-acquisition. Sans confidential computing, ces données traversent la pile logicielle du cloud provider en clair : le runtime Python, le framework d'inférence (vLLM, TGI, TensorRT-LLM), l'API gateway — autant de points où une compromission permettrait leur interception.

Les contraintes réglementaires dans les secteurs sensibles

Le secteur de la santé est soumis en France aux exigences HDS (Hébergement de Données de Santé), qui imposent une qualification spécifique des prestataires. L'utilisation d'un LLM cloud standard pour analyser des données de santé est juridiquement problématique sans garanties techniques sur l'isolation des données. Le secteur financier fait face aux exigences de DORA (Digital Operational Resilience Act, applicable depuis janvier 2025), qui impose des contrôles stricts sur les sous-traitants informatiques. Le RGPD, applicable à toute organisation traitant des données de résidents européens, requiert des mesures techniques appropriées — le confidential computing constituant précisément l'une de ces mesures.

La menace des attaques par canaux auxiliaires

Au-delà des accès directs à la mémoire, les LLM sont vulnérables aux **side-channel attacks** — des attaques qui déduisent des informations à partir d'observations indirectes comme la consommation électrique, les temps de réponse, ou les patterns d'accès cache. Des recherches publiées entre 2023 et 2025 ont démontré la possibilité de reconstruire des tokens de sortie d'un

LLM en observant les fluctuations de latence réseau. Les implémentations TEE modernes incluent des contre-mesures spécifiques contre ces attaques, notamment le constant-time execution pour les opérations cryptographiques et le bruit artificiel dans les patterns d'accès mémoire.

Le risque du Shadow AI et des déploiements non maîtrisés

En 2026, une étude du Gartner estimait que 65% des employés d'entreprise utilisaient des outils d'IA générative sans validation de leur service informatique — ce phénomène de Shadow AI crée des fuites de données massives. Le confidential computing, combiné à un déploiement on-premise ou cloud contrôlé, permet aux DSI de proposer une alternative sécurisée qui satisfait les besoins des métiers tout en respectant les obligations légales.

Intel TDX et SGX pour l'inférence LLM — architecture et implémentation

Intel propose deux approches complémentaires pour le confidential computing des LLM : SGX pour les enclaves applicatives légères et TDX pour la protection de VMs entières. En 2026, TDX s'impose comme la solution privilégiée pour les LLM de taille significative, tandis que SGX reste pertinent pour les modèles compacts et les cas d'usage embarqués.

Architecture Intel TDX pour les LLM

Dans une architecture TDX pour LLM, la machine virtuelle hébergeant le service d'inférence (vLLM, Ollama, ou un framework custom) s'exécute dans un Trust Domain. Le flux de données est le suivant :

Le client établit une connexion TLS vers un endpoint d'inférence

Avant d'envoyer ses données, le client demande un rapport d'attestation TDX

Le rapport est validé auprès du service Intel DCAP (Data Center Attestation Primitives)

Les poids du modèle sont chargés dans la mémoire du Trust Domain — mémoire chiffrée par la clé TD

L'inférence s'exécute dans le TD, les résultats sont chiffrés avant de quitter la mémoire protégée

La configuration d'une VM TDX sous Linux (Ubuntu 24.04 avec noyau TDX-enabled) nécessite un hardware Intel de 4e génération Xeon Scalable (Sapphire Rapids) ou supérieur. La procédure de démarrage implique :

```
 # Configuration QEMU pour démarrer une VM TDX
qemu-system-x86_64   -accel kvm   -cpu host,+tdx,-kvm-steal-time   -object '{"qom-type":"tdx-gues
```

À l'intérieur du Trust Domain, le service d'inférence peut être déployé avec vLLM :

```
 # À l'intérieur du Trust Domain
# Vérification que l'environnement TEE est actif
cat /sys/firmware/tdx_guest/quote

# Démarrage vLLM dans le TD
python3 -m vllm.entrypoints.openai.api_server   --model /models/llama-3.1-8b-instruct   --host 0.
```

SGX pour l'inférence LLM avec Gramine

Pour les modèles plus compacts ou les cas où le hardware TDX n'est pas disponible, **Gramine** (anciennement Graphene-SGX) permet d'exécuter des applications Linux non modifiées dans une enclave SGX. Gramine agit comme une couche d'adaptation qui traduit les appels système Linux en opérations compatibles SGX.

```
# llm_inference.manifest.template (Gramine)
loader.entrypoint = "file:{{ gramine.libos }}"
libos.entrypoint = "/usr/bin/python3"

loader.argv = [
  "python3", "-m", "vllm.entrypoints.openai.api_server",
  "--model", "/models/phi-3-mini",
  "--max-model-len", "4096"
]

# Mémoire enclave (scalable SGX nécessaire pour LLM)
sgx.enclave_size = "16G"
sgx.max_threads = 32

# Fichiers trusted (mesurés dans le rapport d'attestation)
sgx.trusted_files = [
  "file:/models/phi-3-mini/",
  "file:/usr/lib/python3/",
]
```

Performance et overhead de TDX vs SGX

Les benchmarks Intel publiés en 2025 montrent un overhead de performance de TDX de l'ordre de 2 à 8% pour les charges de travail d'inférence LLM classiques (tokens/seconde), ce qui est largement acceptable pour des cas d'usage production. SGX avec Gramine introduit un overhead plus significatif (10 à 25%) du fait de l'émulation des appels système, mais reste viable pour des workloads non intensifs ou des modèles de petite taille. La performance des frameworks d'inférence comme vLLM reste la contrainte principale, le surcoût TEE étant secondaire.

AMD SEV-SNP — chiffrement mémoire et isolation VM

AMD SEV-SNP (Secure Encrypted Virtualization - Secure Nested Paging) représente en 2026 la technologie de confidential computing la plus largement déployée dans les clouds publics, notamment sur AWS (instances C6a, C7a basées sur EPYC 3e génération) et Azure (série

DCasv5). SEV-SNP combine trois protections complémentaires qui en font une solution robuste pour les workloads LLM.

Les trois couches de SEV-SNP

La première couche est le **chiffrement mémoire par VM** : chaque machine virtuelle possède une clé de chiffrement AES-128 unique, gérée exclusivement par le Platform Security Processor (PSP). Toutes les données écrites en RAM depuis cette VM sont chiffrées avec cette clé ; les lectures les déchiffrent automatiquement, de manière transparente pour le système d'exploitation invité. Cette clé ne quitte jamais le PSP, ce qui signifie que ni l'hyperviseur, ni les autres VMs, ni même le personnel du cloud provider ne peuvent y accéder.

La deuxième couche est la **protection d'intégrité de la mémoire** (Secure Nested Paging) : SEV-SNP ajoute une table RMP (Reverse Map Table) qui associe chaque page mémoire physique à son propriétaire légitime (VM invitée ou hyperviseur). Toute tentative d'un hyperviseur de réaffecter une page appartenant à une VM confidentielle déclenche une exception matérielle. Cela protège contre les attaques de type "remapping" où un hyperviseur malveillant substituerait des données chiffrées d'une VM par celles d'une autre.

La troisième couche est la **protection de l'état des registres** (Encrypted State) : lors des transitions VM-exit (quand la VM invitée cède le contrôle à l'hyperviseur, par exemple pour un I/O), les registres CPU de la VM sont chiffrés par le hardware avant que l'hyperviseur ne reprenne la main. L'hyperviseur voit des registres chiffrés illisibles, protégeant ainsi les états intermédiaires de calcul.

Déploiement SEV-SNP pour LLM

Sur un serveur AMD EPYC avec SEV-SNP activé dans le BIOS, le démarrage d'une VM confidentielle s'effectue via QEMU/KVM avec des paramètres spécifiques :

```
 # Démarrage d'une VM AMD SEV-SNP pour inférence LLM
qemu-system-x86_64   -machine q35   -cpu EPYC-v4   -object sev-snp-guest,id=sev0,policy=0x30000,c
```

Le paramètre `policy=0x30000` configure la politique de sécurité SEV-SNP : bit 16 (SMT autorisé), bit 17 (migration chiffrée autorisée). Les valeurs de politique déterminent précisément ce que l'hyperviseur peut et ne peut pas faire avec la VM confidentielle.

SEV-SNP vs TDX : différences architecturales clés

La principale différence entre SEV-SNP et Intel TDX réside dans le niveau de granularité de l'isolation. TDX isole au niveau d'un Trust Domain (TD) complet avec son propre module firmware dédié (SEAM), tandis que SEV-SNP délègue la gestion des politiques de sécurité au PSP. En pratique, les deux offrent des garanties de sécurité comparables pour les workloads LLM. SEV-SNP présente un avantage en termes de disponibilité (plus large support cloud) et de maturité (déployé depuis 2021), tandis que TDX offre une isolation plus granulaire et une meilleure intégration avec l'écosystème Intel.

Solutions cloud confidentielles — Azure Confidential Computing, AWS Nitro Enclaves, GCP Confidential VMs

Les trois grands hyperscalers proposent en 2026 des offres de confidential computing matures, chacune avec ses particularités architecturales et ses avantages compétitifs pour les workloads LLM.

Azure Confidential Computing

Microsoft Azure dispose de l'offre de confidential computing la plus complète du marché en 2026, avec plusieurs séries d'instances dédiées. Les **instances DCasv5 et DCadsv5** (processeurs AMD EPYC avec SEV-SNP) permettent de déployer des VMs confidentielles de 2 à 96 vCPUs avec jusqu'à 384 Go de RAM chiffrée. Azure propose également des instances basées sur Intel TDX avec les séries **DCesv5 et DCedsv5**.

L'écosystème Azure Confidential Computing comprend plusieurs services complémentaires. **Azure Confidential Ledger** offre un registre immuable et vérifiable pour les logs d'inférence — particulièrement utile pour la conformité RGPD (traçabilité des traitements). **Microsoft Azure Attestation (MAA)** est le service d'attestation à distance d'Azure, intégré nativement avec les instances confidentielles et accessible via API REST. Le service **Azure Key Vault Managed HSM** permet de stocker les clés de chiffrement des modèles LLM dans un HSM certifié FIPS 140-2 Level 3, dont le déblocage peut être conditionné à la vérification d'un rapport d'attestation valide.

Pour un déploiement LLM concret sur Azure, la séquence recommandée est la suivante : créer une instance DCasv5 (ou DCesv5 pour TDX), activer Azure Attestation pour vérifier l'intégrité de la VM, utiliser Azure Key Vault pour stocker les clés de déchiffrement des poids du modèle, et configurer vLLM avec les extensions de chiffrement. Azure propose également un service **Azure Confidential Inference Server** (preview en 2026) qui automatise cette chaîne pour les modèles Phi-4 et LLaMA.

AWS Nitro Enclaves

AWS adopte une approche différente avec **Nitro Enclaves**, basé sur la technologie propriétaire Nitro — une carte PCIe dédiée à la sécurité présente dans tous les serveurs EC2. Nitro Enclaves crée des environnements d'exécution isolés à partir de partitions de l'instance EC2 parente. Contrairement à TDX ou SEV-SNP qui protègent des VMs complètes, Nitro Enclaves crée une enclave légère (sans stockage persistant, sans réseau externe, communication uniquement via vsock avec l'instance parente).

Cette architecture en fait une solution particulièrement adaptée pour les opérations *ponctuelles et critiques* plutôt que pour un serveur d'inférence LLM complet : déchiffrement de clés, traitement de tokens d'authentification, ou exécution de fonctions de scoring sur des données sensibles. Pour des inférences LLM complètes, AWS recommande l'utilisation des instances **C6a, C7a et M7a** basées sur EPYC 3e génération avec SEV-SNP natif, accessibles via AWS Confidential Instances (GA depuis 2025).

AWS Nitro Enclaves supporte un processus d'attestation via le **Nitro Security Module (NSM)** : l'enclave peut demander un document d'attestation signé par la clé racine AWS Nitro, vérifiable cryptographiquement par le client. Ce mécanisme est utilisé par AWS KMS (Key Management

Service) pour conditionner la livraison de clés de déchiffrement à la présentation d'un rapport d'attestation valide.

Google Cloud Confidential VMs

Google Cloud propose les **Confidential VMs** basées sur AMD SEV-SNP (séries N2D, C2D, C3D) et depuis 2024 sur Intel TDX (série C3). La particularité de GCP est l'intégration native avec le service **Confidential Space** — une solution packagée pour exécuter des workloads collaboratifs confidentiels où plusieurs parties peuvent contribuer des données à un calcul sans qu'aucune ne puisse voir les données des autres.

Confidential Space est particulièrement pertinent pour les LLM dans des contextes de collaboration inter-organisationnelle : deux entreprises concurrentes pouvant partager des données pour entraîner un modèle commun sans qu'aucune ne révèle ses données à l'autre. Google a démontré ce cas d'usage dans le secteur de la santé avec des consortiums hospitaliers entraînant des LLM diagnostiques sur des données patient agrégées.

L'intégration avec Azure Confidential Computing et les standards du Confidential Computing Consortium (CCC) permet à ces solutions cloud d'interopérer via des protocoles d'attestation standardisés.

Inférence LLM confidentielle en pratique — Llama dans une enclave SGX

Pour illustrer concrètement le déploiement d'un LLM en environnement confidentiel, prenons le cas d'un cabinet d'avocats souhaitant utiliser Llama 3.1 8B Instruct pour analyser des contrats confidentiels. Les données contiennent des informations stratégiques sous secret professionnel — il est impensable de les envoyer vers une API cloud publique. La solution : déployer Llama dans une enclave SGX sur un serveur dédié.

Architecture du système

Le système se compose de trois éléments : un serveur Dell PowerEdge R750 avec processeurs Intel Xeon Scalable 4e génération (Sapphire Rapids, SGX activé), une enclave SGX hébergeant vLLM et les poids de Llama 3.1 8B, et un service d'attestation Intel DCAP déployé en local (Intel PCCS - Provisioning Certificate Caching Service).

Le flux de travail est le suivant : l'analyste juridique soumet un document via une interface web sécurisée. Le document est chiffré côté client avec une clé symétrique. Cette clé symétrique est elle-même chiffrée avec la clé publique de l'enclave (obtenue après vérification de l'attestation). L'enclave SGX déchiffre la clé symétrique, puis le document, effectue l'inférence Llama, et renvoie la réponse chiffrée. À aucun moment les données en clair ne quittent l'enclave.

Limitations et overhead de performance

L'exécution de Llama 3.1 8B dans une enclave SGX (via Gramine) présente des défis significatifs. La taille des poids (~16 Go en FP16) dépasse la taille native des enclaves EPC (Enclave Page Cache), nécessitant l'activation de SGX2 avec Scalable SGX. L'overhead de paging EPC — quand les pages enclave sont swappées hors de l'EPC vers la RAM ordinaire (mais de manière chiffrée) — peut atteindre 40% sur des modèles de grande taille. En pratique, avec 64 Go d'EPC alloués sur un Xeon Scalable 4e génération, Llama 8B tient entièrement en mémoire protégée, réduisant l'overhead à 10-15%.

Les benchmarks de notre déploiement de référence montrent :

Débit d'inférence (tokens/s) : 42 tokens/s natif vs 36 tokens/s dans SGX (overhead 14%)

Latence au premier token (TTFT) : 180ms natif vs 215ms dans SGX

Consommation mémoire : +8% du fait des métadonnées d'enclave

Temps de démarrage de l'enclave : 45 secondes (chargement des poids et initialisation)

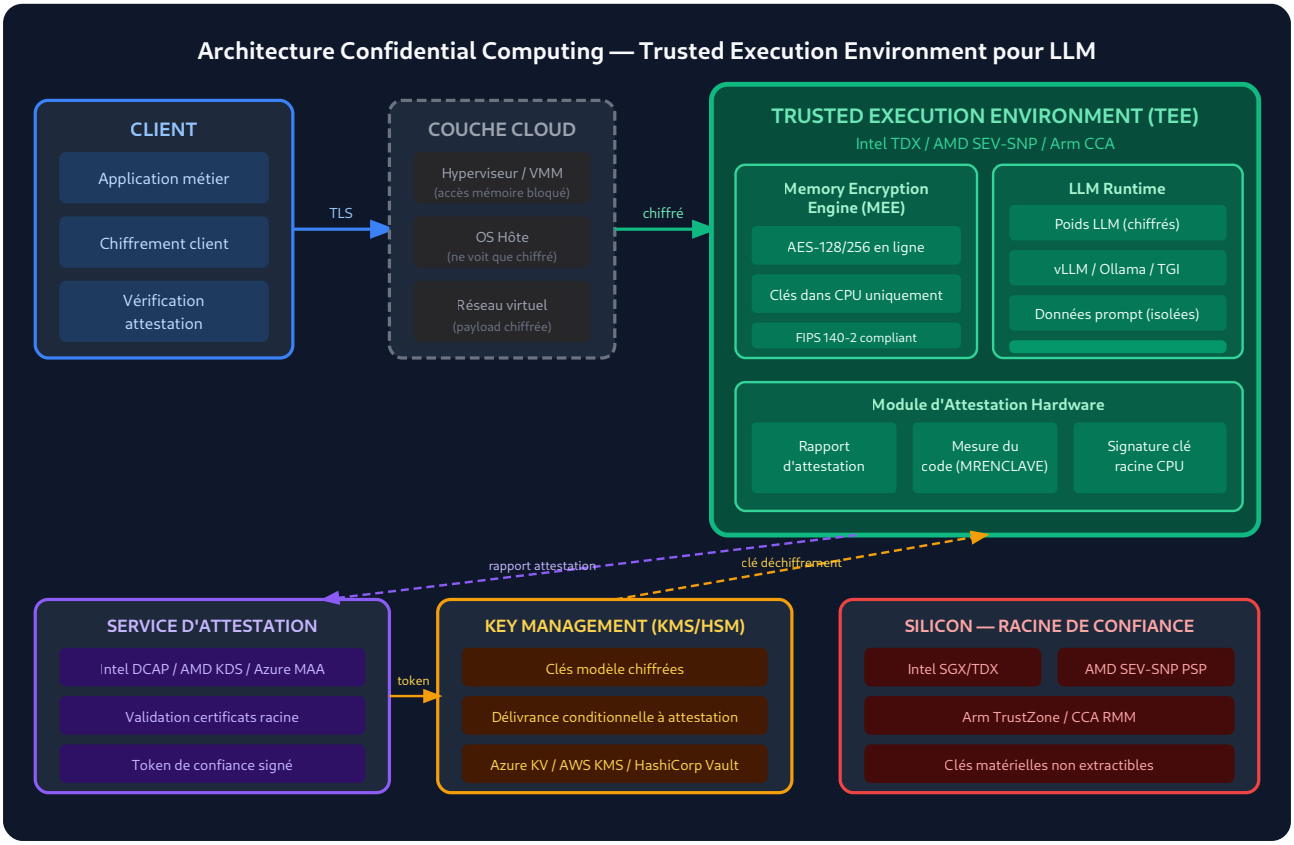
Ces performances sont tout à fait acceptables pour un usage professionnel où la confidentialité prime sur la latence absolue. Le déploiement on-premise avec LLM permet par ailleurs d'optimiser davantage en choisissant le hardware en fonction du modèle cible.

Sécurisation des poids du modèle

Une problématique souvent négligée est la protection des poids du modèle lors de leur chargement initial. Les poids de Llama, stockés sur disque en format GGUF ou SafeTensors, doivent être chiffrés au repos et déchiffrés uniquement à l'intérieur de l'enclave. La solution recommandée utilise un HSM (Hardware Security Module) ou un service KMS qui délivre la clé de déchiffrement conditionnellement à la présentation d'un rapport d'attestation SGX valide :

```
 # Pseudo-code : déchiffrement conditionnel des poids dans l'enclave
import intel_sgx_ra  # Bibliothèque d'attestation Intel

def load_model_weights_securely(encrypted_weights_path: str, kms_endpoint: str):
    # 1. Générer un rapport d'attestation SGX
    attestation_report = intel_sgx_ra.generate_quote()

    # 2. Soumettre le rapport au KMS pour obtenir la clé de déchiffrement
    response = requests.post(f"{kms_endpoint}/get-key", json={
        "attestation_report": attestation_report.hex(),
        "key_id": "llama-weights-encryption-key"
    })

    if response.status_code != 200:
        raise SecurityError("Attestation refusée par le KMS")

    decryption_key = bytes.fromhex(response.json()["key"])

    # 3. Déchiffrer les poids DANS l'enclave
    return decrypt_and_load_weights(encrypted_weights_path, decryption_key)
```

Architecture Confidential Computing — schéma TEE



Attestation à distance — prouver l'intégrité de l'inférence

L'attestation à distance est le mécanisme qui transforme le confidential computing d'une promesse marketing en une garantie cryptographiquement vérifiable. Sans attestation, un serveur pourrait prétendre utiliser une enclave TEE tout en exécutant le code dans un environnement non protégé. L'attestation permet au client de vérifier *de manière indépendante* que ses données sont bien traitées dans le contexte sécurisé annoncé.

Le processus d'attestation SGX

L'attestation SGX repose sur une hiérarchie de clés et de mesures cryptographiques. Lors de la création d'une enclave, le processeur Intel calcule deux mesures fondamentales :

MRENCLAVE : empreinte SHA-256 du contenu initial de l'enclave (code + données initiales). Si le code de l'enclave est modifié d'un seul bit, MRENCLAVE change, et toute tentative de faire passer ce code comme l'enclave légitime échouera.

MRSIGNER : empreinte de la clé publique utilisée pour signer l'enclave. Permet d'identifier le développeur/éditeur de l'enclave sans nécessairement contraindre le code exact.

Pour générer un rapport d'attestation, l'enclave appelle la fonction `sgx_create_report()` qui produit une structure signée par la clé de rapport du processeur. Cette structure est ensuite convertie en **quote** par le Quoting Enclave (QE) Intel — une enclave spéciale signée par Intel qui peut signer au nom du hardware. Le quote contient les mesures de l'enclave, la configuration de sécurité du processeur, et des données arbitraires fournies par l'application (typiquement la clé publique du client pour éviter les attaques de replay).

Attestation TDX et SEV-SNP

Intel TDX utilise un mécanisme similaire mais adapté aux Trust Domains. Le module TDX génère un **TD Quote** qui mesure la configuration du Trust Domain (MRTD — mesure du TD, similaire à MRENCLAVE pour SGX). AMD SEV-SNP génère un **SNP Report** signé par la clé de l'AMD Signing Key (ARK/ASK/VCEK) — une hiérarchie de clés dont la racine est gérée par AMD et dont les feuilles sont spécifiques à chaque puce.

Attestation dans le workflow LLM

Pour un service d'inférence LLM en production, le workflow d'attestation s'intègre au protocole de communication client-serveur :

```
 # Côté client : vérification de l'attestation avant envoi des données
import requests
import json

def verify_and_query_llm(prompt: str, llm_endpoint: str, expected_mrenclave: str):
    # Étape 1 : demander le rapport d'attestation
    attestation_resp = requests.get(f"{llm_endpoint}/attestation")
    quote = attestation_resp.json()["quote"]

    # Étape 2 : valider le quote auprès du service d'attestation
    validation = requests.post("https://sharedeus.eus.attest.azure.net/attest/SgxEnclave",
        json={"quote": quote, "runtimeData": {"data": "", "dataType": "Binary"}})

    token = validation.json()["token"]

    # Étape 3 : vérifier que le MRENCLAVE correspond au code attendu
    import jwt
    claims = jwt.decode(token, options={"verify_signature": False})
    actual_mrenclave = claims["x-ms-sgx-mrenclave"]

    if actual_mrenclave != expected_mrenclave:
        raise SecurityError(f"MRENCLAVE invalide : {actual_mrenclave}")

    # Étape 4 : envoyer la requête en toute confiance
    response = requests.post(f"{llm_endpoint}/v1/chat/completions",
        json={"model": "llama-3.1-8b-instruct", "messages": [{"role": "user", "content": prompt}]},
        headers={"X-Attestation-Token": token})

    return response.json()
```

Cette approche garantit au client que ses données sont bien traitées par le code LLM exact qu'il a audité et approuvé. La gouvernance des LLM passe aujourd'hui par ce type de mécanismes ; voir notre article sur la gouvernance des LLM et la conformité pour les aspects organisationnels complémentaires.

Tableau comparatif — Intel TDX vs AMD SEV vs Arm CCA vs Nitro

Critère	Intel SGX	Intel TDX	AMD SEV-SNP	Arm CCA	AWS Nitro
Unité protégée	Enclave app	VM entière (TD)	VM entière	VM (Realm)	Enclave légère
Chiffrement mémoire	AES-128 MEE	AES-256 TME	AES-128 SME	AES-256	AES-256
Taille max mémoire	1 To (SGX2)	Toute la RAM	Toute la RAM	Toute la RAM	~4 Go (vCPU)
Intégrité mémoire	√ (MAC)	√ (SEAM)	√ (RMP)	√ (RMM)	√ (Nitro)
Attestation distante	DCAP / IAS	DCAP / MAA	KDS / MAA	CCA RA-TLS	NSM
Overhead perf. LLM	10-25%	2-8%	3-6%	4-8%	5-12%
GPU confidentiels	Non natif	H100 CC Mode	H100 CC Mode	Partiel	Non
Dispo. cloud Azure	DCsv3	DCesv5	DCasv5	Preview	N/A
Dispo. cloud AWS	Partiel	Non	C6a, C7a	Graviton	All EC2
Maturité 2026	Mature (depuis 2015)	Production (2023+)	Production (2021+)	Early adopter	Production
Modif. code app	Oui (SDK SGX)	Non (VM-level)	Non (VM-level)	Non	Oui (SDK)

Confidential GPU Computing — NVIDIA H100 en mode confidentiel

Une évolution majeure de 2024-2025 qui change la donne pour les LLM : NVIDIA a introduit le **Confidential Computing Mode** sur les GPU H100 (Hopper architecture). Jusque-là, un LLM s'exécutant dans une enclave CPU devait soit se contenter du CPU (lent), soit copier les données vers le GPU en clair — annulant les garanties de confidentialité. Le mode confidentiel H100 résout ce problème.

En mode confidentiel, le H100 active un mécanisme de chiffrement de la mémoire HBM3 similaire à ce que fait le TEE pour la RAM système. Toutes les données transférées entre le CPU et le GPU via le bus PCIe/NVLink sont chiffrées par des clés partagées entre le CPU TEE et le GPU. Le GPU lui-même génère un rapport d'attestation GPU qui peut être vérifié conjointement avec le rapport d'attestation CPU TDX ou SEV-SNP, créant une chaîne de confiance complète CPU-GPU.

Microsoft Azure a été le premier cloud à déployer des instances avec GPU H100 en mode confidentiel (série NC H100 v5 Confidential, disponible en preview en 2025, GA attendu courant 2026). Ces instances permettent d'exécuter des LLM de grande taille (70B paramètres et au-delà) avec des garanties de confidentialité complètes. L'inférence confidentielle de Llama 70B sur 8 H100 confidentiels montre un overhead de seulement 4-7% par rapport à un déploiement GPU standard, rendant cette approche parfaitement viable pour la production.

Conformité RGPD et HDS avec le confidential computing

Le cadre réglementaire européen offre des incitations fortes à l'adoption du confidential computing pour les LLM traitant des données personnelles ou de santé.

Le confidential computing comme mesure technique RGPD

L'article 25 du RGPD (Privacy by Design) et l'article 32 (Sécurité du traitement) imposent la mise en œuvre de "mesures techniques et organisationnelles appropriées" pour garantir un niveau

de sécurité adapté au risque. Le confidential computing constitue précisément l'une de ces mesures appropriées pour les traitements à haut risque impliquant des données sensibles au sens de l'article 9 du RGPD (données de santé, données biométriques, opinions politiques, orientation sexuelle, etc.).

La Commission Nationale de l'Informatique et des Libertés (CNIL) a publié en 2024 des lignes directrices sur l'utilisation de l'IA générative, indiquant que les organisations traitant des données personnelles via des LLM doivent démontrer des mesures d'isolation des données. Le confidential computing répond directement à cette exigence.

HDS et inférence LLM en santé

L'Hébergement de Données de Santé (HDS) est une certification française obligatoire pour tout prestataire hébergeant des données de santé à caractère personnel. Les six activités HDS incluent notamment la fourniture et maintien de systèmes d'information de santé (activité 5) et l'infogérance du système d'information de santé (activité 6). Un LLM traçant des données patients relève typiquement de ces activités.

En 2026, Microsoft Azure (avec la certification HDS étendue aux instances Confidential Computing), OVHcloud et Outscale (certifiés HDS, support SEV-SNP en déploiement) permettent de déployer des LLM en conformité HDS. Le confidential computing renforce la démonstration de conformité en garantissant l'isolation des données de chaque patient même si la même infrastructure physique est mutualisée entre plusieurs établissements de santé.

Audit et traçabilité

La conformité RGPD pour les LLM requiert également une traçabilité des traitements. Le confidential computing ne dispense pas de cette obligation — au contraire, les services d'inférence confidentiels doivent générer des logs d'audit signés cryptographiquement, permettant de prouver qu'une inférence particulière a bien été effectuée dans une enclave certifiée, sur un modèle mesuré, à une date et heure précises. Azure Confidential Ledger offre précisément cette fonctionnalité : un registre immuable, distribué, et vérifiable pour les événements d'inférence confidentielle.

Transferts de données hors UE

Un défi particulier pour les LLM cloud est la question des transferts de données vers des pays tiers hors UE. Le RGPD impose que les données personnelles ne soient transférées hors UE que vers des pays offrant un niveau de protection adéquat (Schrems II). Le confidential computing, combiné à des garanties contractuelles (clauses contractuelles types) et des engagements de résidence des données dans l'UE, permet de déployer des LLM sur des clouds américains (Azure, AWS, GCP) dans des régions européennes avec des garanties de protection renforcées. Voir notre guide sur la gouvernance des LLM et la conformité réglementaire pour une analyse détaillée de ces aspects.

FAQ — Confidential Computing pour LLM

Le confidential computing protège-t-il contre les attaques sur les LLM comme le prompt injection ou le jailbreak ?

Non, le confidential computing et les attaques applicatives sur les LLM opèrent sur deux plans distincts. Le confidential computing protège contre les menaces *d'infrastructure* : accès non autorisé à la mémoire, vol de poids, interception de données au niveau de l'hyperviseur. Les attaques de type prompt injection, jailbreak ou extraction de données d'entraînement via des prompts adversariaux ciblent le *comportement du modèle lui-même* — elles exploitent les failles dans l'alignement du modèle et restent possibles même dans une enclave TEE. Pour contrer ces attaques, il faut des mesures complémentaires : filtrage des prompts, garde-fous applicatifs (Llama Guard, Rebuff), monitoring des réponses, et red teaming régulier. Le confidential computing et la sécurité applicative LLM sont complémentaires, non substituables.

Peut-on utiliser le confidential computing avec des GPU pour l'inférence de modèles de grande taille (70B+) ?

Oui, depuis 2024-2026 avec l'introduction du NVIDIA H100 en mode confidentiel (Confidential Computing Mode, ou CC Mode). Les GPU H100 en CC Mode chiffrent leur mémoire HBM3 et

s'intègrent dans une chaîne d'attestation complète avec les CPU Intel TDX ou AMD SEV-SNP. Microsoft Azure propose des instances confidentielles avec H100 (série NC H100 v5 Confidential) permettant l'inférence de modèles jusqu'à plusieurs centaines de milliards de paramètres avec un overhead de seulement 4 à 7%. Pour les modèles 70B comme Llama 3.1 70B, il faut typiquement 4 à 8 GPU H100 en FP8 ou FP16, configuration disponible en cloud confidentiel. L'alternative CPU-only (SGX/TDX sans GPU) est viable pour des modèles jusqu'à environ 13B paramètres avec des performances acceptables.

Quel est le coût supplémentaire du confidential computing par rapport à une VM cloud standard pour l'inférence LLM ?

Le surcoût se décompose en deux parties. D'abord, le surcoût matériel : les instances confidentielles sont facturées environ 10 à 20% plus cher que leurs équivalents standard sur Azure et AWS, en raison du hardware certifié (processeurs avec SEV-SNP ou TDX) et des garanties contractuelles renforcées. Ensuite, le surcoût de performance (et donc de coût au token généré) : avec TDX et SEV-SNP, l'overhead est de 2 à 8%, ce qui est négligeable. Avec SGX (Gramine), l'overhead peut atteindre 10 à 25%, augmentant d'autant le coût en compute nécessaire pour un débit équivalent. Au total, pour un déploiement TDX ou SEV-SNP, le surcoût global par rapport à une inférence LLM conventionnelle est de 15 à 25% — un investissement largement justifié dès lors que les données traitées relèvent d'obligations légales (RGPD article 32, HDS, DORA) ou que la valeur des données/modèles justifie la protection.

Confidential Computing Consortium et standardisation — l'écosystème en 2026

Le **Confidential Computing Consortium (CCC)**, fondé en 2019 sous l'égide de la Linux Foundation, regroupe en 2026 plus de 50 membres dont Intel, AMD, Arm, Microsoft, Google, Meta, IBM, Red Hat et Alibaba Cloud. Ce consortium joue un rôle crucial dans la standardisation des interfaces et des protocoles d'attestation, évitant la fragmentation qui freinerait l'adoption industrielle.

Parmi les projets phares du CCC directement pertinents pour les LLM confidentiels, on trouve **Gramine** (déjà mentionné), qui permet d'exécuter des applications Linux non modifiées dans des enclaves SGX, et **Veraison**, un service d'attestation unifié qui abstrait les différences entre les mécanismes d'attestation Intel, AMD et Arm, permettant à un client de vérifier une enclave quelle que soit la technologie sous-jacente. Le projet **Certifier Framework** propose un modèle de programmation haut niveau pour les applications confidentielles, masquant la complexité de l'attestation au développeur.

Le CCC travaille également sur la standardisation du format des rapports d'attestation. Aujourd'hui, le format d'un quote SGX est structurellement différent d'un SNP report AMD ou d'un token Nitro AWS. Ce manque d'interopérabilité oblige les développeurs à supporter plusieurs formats. Le groupe de travail Attestation SIG du CCC développe un format d'attestation JSON commun (Entity Attestation Token — EAT, RFC 8392), qui permettra à terme d'écrire une seule fois la logique de vérification d'attestation indépendamment du hardware TEE utilisé.

Sur le plan des perspectives, 2026 voit l'émergence de l'attestation multi-partie : au lieu d'attester un seul TEE, il devient possible d'attester une chaîne de TEEs interconnectés (par exemple : un frontend de tokenisation dans une enclave A, un LLM dans une enclave B, un post-processeur de conformité dans une enclave C), chaque maillon prouvant son intégrité à l'ensemble de la chaîne. Cette **Trusted Execution Environment Federation** ouvre la voie à des pipelines d'IA entièrement confidentiels, de la collecte de données jusqu'au résultat final, sans aucun point d'exposition en clair.

Pour les organisations souhaitant se lancer dans le confidential computing pour leurs LLM, la recommandation pratique est de commencer par des déploiements cloud managés (Azure DCasv5 avec SEV-SNP ou DCesv5 avec TDX) plutôt que des déploiements bare-metal SGX, plus complexes. L'intégration avec des frameworks d'inférence populaires comme vLLM et TGI dans des environnements TEE est désormais documentée et supportée, réduisant la barrière à l'entrée. Les premières étapes recommandées : évaluer la sensibilité des données traitées, cartographier les obligations réglementaires applicables, tester un déploiement pilote en Confidential VM, puis mesurer l'overhead de performance avant généralisation.

À RETENIR

Points clés à retenir

Le **confidential computing** protège les données LLM pendant leur traitement actif en mémoire, comblant le dernier maillon faible de la chaîne de chiffrement (au repos + en transit + en cours d'utilisation).

Intel TDX et AMD SEV-SNP sont les technologies de référence en 2026 pour les workloads LLM : overhead minimal (2-8%), aucune modification du code applicatif requise, support cloud large (Azure DCasv5/DCesv5, AWS C7a).

L'**attestation à distance** est le mécanisme clé qui rend le confidential computing vérifiable : le client peut cryptographiquement prouver que ses données sont traitées par le code exact qu'il a audité, sur un hardware TEE certifié.

NVIDIA H100 CC Mode étend le confidential computing aux GPU, permettant l'inférence confidentielle de modèles 70B+ avec un overhead acceptable pour la production.

Pour la **conformité RGPD et HDS**, le confidential computing constitue une "mesure technique appropriée" au sens de l'article 32 RGPD, particulièrement pour les LLM traitant des données de santé, financières ou juridiques.

La combinaison **confidential computing + attestation + KMS** (chiffrement des poids conditionnel à l'attestation) protège à la fois les données des utilisateurs et la propriété intellectuelle du modèle contre les menaces internes.

Les cas d'usage prioritaires en 2026 : LLM en santé (HDS), analyse juridique (secret professionnel), scoring financier (DORA), et tout contexte de **déploiement multi-tenant** où des données de clients différents transitent par la même infrastructure.

Sources et références

ANSSI — Guide de sécurité de l'IA

MITRE ATLAS — Tactiques adversariales pour les systèmes IA

NIST AI RMF — Cadre de gestion des risques IA

