

Designing scalable agentic AI infrastructure for government

Implement agentic AI while preserving sovereignty, security, auditability, and operational continuity across environments with Intel® technology.

Contents

Executive summary	1
The AI infrastructure conversation must change	2
Agentic AI decision-making for government technology leaders	4
Architecture requirements for distributed agentic AI	8
CPU coordination checklist.....	9
Design pattern for AI workload placement	11
Reference architecture for local-first agentic AI	12
Workflow sequence	14
Intel capabilities mapped to architecture requirements	17
Conclusion	22
Recommended resources	23
Appendix: The agentic AI environment assessment framework	24

Executive summary

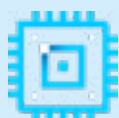
Government agencies are under pressure to use AI to improve mission delivery, accelerate decision cycles, strengthen cybersecurity operations, and modernize citizen services without compromising public trust, sensitive data, or compliance obligations. Technology architects can use agentic AI to build intelligent platforms that help fulfill these complex requirements and move agencies away from isolated prompts in the cloud toward systems that sense, reason, retrieve, plan, and act across workflows. But agentic AI entails infrastructure considerations that need to be addressed.

For government leaders, the key question is not only whether an AI model can perform a task, but whether the full agentic workflow can operate within approved boundaries for data residency, identity, Zero Trust, auditability, acquisition requirements, and mission continuity while efficiently using scarce GPU resources.

A hybrid, local-first approach addresses this by helping agencies place sensitive and repetitive agent tasks closer to data, including client, edge, and data center platforms, while reserving cloud models and GPU acceleration for when they add the most value.

The biggest hidden inefficiency in scaling AI is idle GPUs. Based on internal testing, Intel researchers found that the optimal CPU-to-GPU ratio for certain use cases is around 0.8:1 to 1.4:1, while the right balance remains highly dependent on the workload.¹

For government agencies, the architectural implication is not simply minimizing token volume or adding more GPUs—it is creating a more defensible path to scale where cost, latency, utilization, data exposure, and compliance risks are reduced together rather than traded off against each other.



Intel researchers found that the optimal CPU-to-GPU ratio is around 0.8:1 to 1.4:1.¹

The AI infrastructure conversation must change

Agentic AI introduces challenges that extend beyond model deployment. As AI begins to influence workflows, citizen services, cybersecurity operations, and decision-making, AI architecture has to support more than inference on a single system. AI becomes embedded in execution paths that require persistent monitoring, stateful reasoning, closed-loop actions, and continuous coordination across heterogeneous environments.

The value is not only in running a model but in enabling systems that can interact with data and applications, respond to changing information, and operate across multiple environments with persistence and control. For government leaders, that shift creates a new set of infrastructure requirements that need to be addressed before agentic AI can move from a pilot activity to a trusted, autonomous tool used in production.

Workloads have long had to be coordinated across data centers, clouds, networks, and at the edge, but agentic AI changes the nature of that coordination by shifting predefined flows to dynamic runtime flows. In this model, network conditions, data locality, service latency, memory residency, and I/O availability become inputs into agentic AI decision-making and not merely background optimization variables.

Misalignment across environments, such as slow edge-to-cloud round trips, insufficient memory for context and state, or I/O contention between storage, networking, and accelerator traffic does not simply degrade performance; it can change agent decisions, trigger retries and timeouts, and affect the correctness of outcomes. As a result, coordination across data center, cloud, and edge becomes a system bottleneck rather than a background concern, and any inefficiencies can multiply across thousands of agent loops within a workflow. For technical architects, this means bandwidth, latency, topology, memory behavior, and I/O characteristics can drastically affect throughput, total cost of ownership (TCO), and operations.

Agentic AI requires thinking differently and understanding token usage

These architecture considerations make infrastructure modernization an organizational continuity challenge as much as a technology challenge. Data movement, memory, and system state have to remain efficient across heterogeneous resources. Security, governance, and observability need to be built into architecture rather than added afterward. Public sector agencies have to evolve their operational thinking because few of them can scale AI economically by creating separate, parallel infrastructures.



A hidden cost in scaling AI is the cost of GPU capacity that sits idle because the surrounding CPU, memory, I/O, retrieval, orchestration, and data pipelines cannot keep it continuously fed.¹



For government agencies, this paradigm shift must also align with policy and oversight expectations. For example, federal AI guidance emphasizes governance, use-case inventories, risk management, responsible acquisition, and public trust. These requirements do not argue against agentic AI, but they do help define architecture requirements and planning.

This is why token efficiency has to become part of the conversation about infrastructure. Agentic systems are not one-and-done prompt engines. They operate in loops of planning, retrieving context, calling tools, generating intermediate outputs, validating results, and sometimes repeating the process. Each loop consumes tokens, and even as overall token prices decline, the total operating cost of agentic AI can increase dramatically if workflows consume more tokens to complete tasks. For architects, the relevant unit of analysis becomes cost per successful task, not cost per individual inference.

A hybrid agentic architecture gives architects a practical lever for this new cost structure. Repetitive tasks like file access, document parsing, context preparation, summarization, tool routing, policy checks, and initial content generation can often run locally or within controlled organizational environments before invoking a cloud model.

This is not a blanket rejection of cloud inference; it is intentional placement of AI tools where they belong according to architectural considerations. Cloud resources remain available for advanced reasoning, large-model interaction, or external information retrieval. Locally run AI handles high-frequency steps that would otherwise create avoidable token spend, latency, and unnecessary movement of data across environmental boundaries.

The same logic applies to accelerator planning. The hidden cost in scaling AI is not only the price of GPU capacity, but the cost of GPU capacity that sits idle because the surrounding CPU, memory, I/O, retrieval, orchestration, and data pipelines cannot keep it continuously fed.¹ GPU utilization can be extremely low in poorly balanced environments, and CPU-to-GPU ratios should be tuned by workload. Public sector agencies should baseline real mission workflows, measure how much work runs on CPUs versus GPUs, and design for balanced utilization rather than reflexively purchasing additional, potentially unnecessary GPUs.

Agentic AI decision-making for government technology leaders

Determining whether agentic AI infrastructure can deliver mission value at production scale starts by defining the outcomes that government leaders want, identifying relevant architecture questions, and then supporting the solution with proof points. This high-level process enables public sector agency leaders to preserve operational continuity, coordinate work across heterogeneous resources, govern every step of agentic AI workflows, and generate the evidence needed to justify deployment and sustain public trust.

The most important decisions to make about AI involve:

- **Mission value, controlling costs, and ROI.** Priority should be given to scaling AI in a way that improves mission outcomes while minimizing cloud-token spend, idle CPUs, power consumption, and infrastructure expansion.

Important questions to address are whether the platform can move from pilot to production within real budgets and power, procurement, and operating constraints.

Improving cost per successful task requires a balanced architecture that avoids idle GPUs by using them where they are truly needed and keeping them efficiently fed by CPU, memory, I/O, retrieval, and orchestration pipelines. According to research from the Georgia Institute of Technology and Intel, 50% to 90% of all latency in agentic AI workloads is related to tool processing on CPUs,^{1,2} so keeping retrieval, reasoning, and orchestration on the same platform, such as Intel® Xeon®, makes sense for government agencies. Intel® Xeon® 6 with P-cores also helps reduce costs with up to 10-to-1 server consolidation, delivering up to 68% lower TCO and 80% less power consumption.³

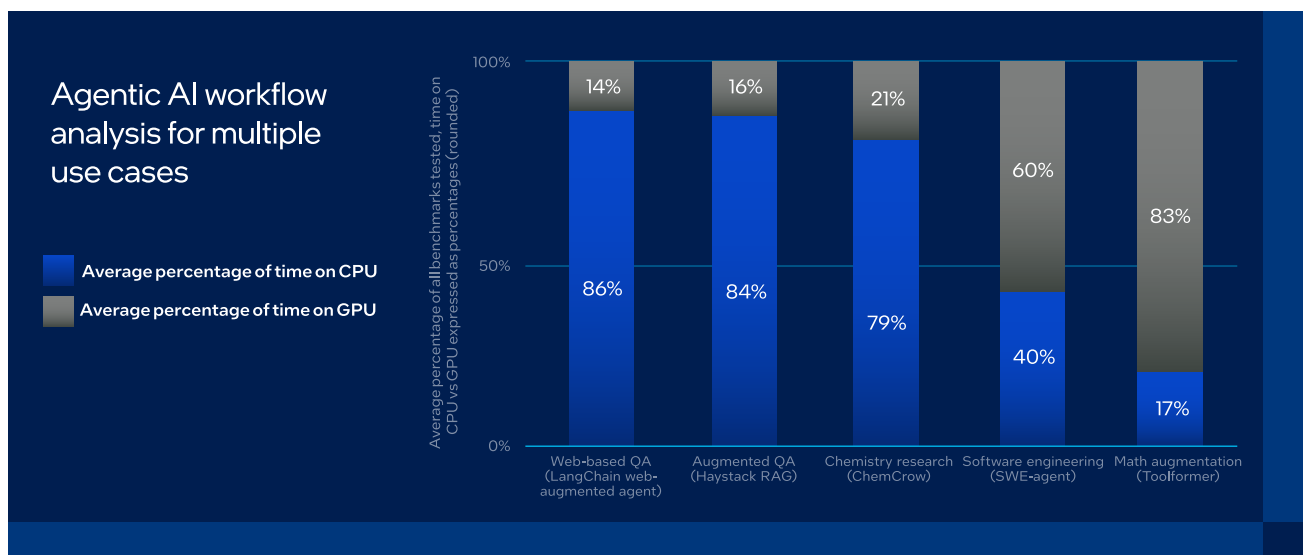


Figure 1. The average percentage of processing time spent on a CPU and a GPU for five different workloads.^{1,2}

- **Reliable mission scale.**

Production AI systems have to remain stable under real-world demand. An important question to address is whether the architecture can support mixed workloads across data center, cloud, network, edge, and client environments with consistent behavior, service continuity, and recoverability.

Intel® platforms are relevant where platform-level reliability, availability, and serviceability (RAS) features, common architecture, and edge-to-data-center deployment consistency help reduce downtime and sustain service continuity. Intel Xeon 6 extends RAS capabilities for data center reliability, while Intel® Xeon® 6 SoC for network and edge increases RAN capacity up to 2.4x⁴ and up to 70% performance per watt RAN improvement versus previous generation for selected workloads.⁵

- **Faster modernization with less disruption.**

The fastest path to production value is often extending AI into existing application estates, data sources, middleware, operational processes, and infrastructure dependencies without creating a separate operating model. An important question to address is how much integration effort, redesign, operational retraining, and adaptation an existing architecture requires.

Intel offers architectural continuity because government agencies can modernize on already broadly deployed x86 foundations and extend agentic AI across familiar environments. Using a consistent x86 platform like Intel Xeon 6 can help government agencies avoid replatforming core applications, revalidating large portions of the software stack, and requiring operations teams to retrain every time underlying technology choices shift.

- **Governed trust and data protection.**

Sensitive agentic AI workflows require protection for data, policy logic, credentials, and mission context as agents retrieve, reason, and act across systems. Important questions to address are whether the architecture can enforce isolation, protect sensitive data in use, verify a trusted execution state, and produce the operational evidence required for policy and compliance. Intel's leadership in confidential computing is relevant because Intel provides both enclave-based and virtual machine (VM)-based trusted execution environments through Intel® Software Guard Extensions (Intel® SGX) and Intel® Trust Domain Extensions (Intel® TDX), along with attestation services that help establish trust across cloud, on-premises, and edge deployments. In an agentic AI architecture, those capabilities can help bind policy enforcement, data protection, and workload verification across the entire execution path.

- **Efficient data movement and infrastructure use.**

Responsive AI workflows depend on efficient movement of data across compute, storage, memory, network, and accelerator domains. Important questions to address are whether the architecture can avoid bottlenecks, excessive data movement, memory pressure, and inefficient placement of AI relative to where data resides. This matters because agentic AI performance depends on the full pipeline, not only model execution speed. Intel® technologies are relevant where strong I/O, integrated acceleration, and networking capabilities can help reduce avoidable data movement, improve infrastructure utilization, and support more efficient execution across rack, power, thermal, and edge constraints.

- **Productive accelerator utilization.**

Scarce GPU resources should benefit mission workflows rather than sit idle because the surrounding system cannot keep them busy. Important questions to address are whether the environment has enough CPU coordination, memory bandwidth, storage throughput, network capacity, and data-preparation capability to keep GPU resources productive. Intel's analysis indicates that for certain use cases the optimal CPU-to-GPU ratio can fall between about 0.8:1 and 1.4:1, but the correct ratio depends on the workload.¹ Government teams should treat this as a planning range to test against real mission workflows, not as a universal procurement formula.

Common constraints that shape those decisions are:

- **Legacy integration pressure.**

New AI capabilities still have to work with installed systems, operational processes, and existing data flows. Lack of operational continuity is a deal-breaker for many industries—especially for government agencies, where service disruptions for constituents can represent more than a mere loss of convenience.

- **Multi-environment complexity.**

Heterogenous environments with multiple data silos often have to work together. Managing edge, network, data center, and cloud environments can be highly complex because each requires specialized skills and optimization. For example, a typical agentic AI workflow might involve generating data at the edge, sharing it with data center applications, and then leveraging models in the cloud. These environments all have to be coordinated, reliable, and available, and if you have different architecture platforms, you must engineer and maintain them separately. Using a consistent foundation eliminates much of this complexity and provides predictability.

- **Budget and timeline pressure.**

Architectural choices are judged not only on technical elegance but on whether they can be implemented within real-world funding constraints and inflexible delivery windows. Additionally, teams that will deploy, govern, and operate AI-ready technology architectures might require training, which entails further costs and time.



Intel® Xeon® 6 with P-cores also helps reduce costs with up to 10-to-1 server consolidation, delivering up to 68% lower TCO and 80% less power consumption.³



- **Regulatory and compliance obligations.**
Infrastructure decisions increasingly need to withstand scrutiny around data handling, sovereignty, auditability, and policy enforcement. For example, National Institute of Standards and Technology AI risk management guidance highlights the need to govern, map, measure, and manage AI risks across systems' life cycles. Federal Risk and Authorization Management Program (FedRAMP) authorization and related cloud security expectations shape how U.S. agencies use external services.
- **Authorization to operate (ATO) and procurement readiness.**
Architecture choices require documentation, telemetry, control evidence, and operational boundaries needed to support security reviews, risk acceptance, acquisition evaluation, and ongoing compliance. These recurring activities must be planned for and begin on day one.
- **Sovereignty, residency, and classified or sensitive boundaries.**
Government AI architectures must define where data, prompts, embeddings, retrieval indexes, intermediate outputs, and model interactions are allowed to reside and run, especially when workflows span on-premises systems, approved clouds, edge environments, and disconnected or mission-constrained locations. To execute Zero Trust strategies, public sector agencies must continuously verify users, devices, workloads, services, and increasingly non-human agent identities.

Together, these decision-making criteria shift your architecture review from a feature checklist to an examination of desired organizational outcomes. To modernize your existing technology stack for AI, put agents where they can run most efficiently and securely, preserve governance as data moves across environments, and generate the evidence that government technology teams need to justify deployment, sustain authorization, and maintain public trust.

Architecture requirements for distributed agentic AI

Distributed agentic AI requires more than model hosting. As AI takes on more responsibilities across public sector workflows, it is doing more than completing individual tasks. The architecture has to support a consistent experience across existing systems, but it must also be able to coordinate work across many types of resources and platforms. At the same time, government systems have to be able to maintain governance and oversight as AI systems evolve.

These requirements can be organized into three practical design priorities for technical architects: continuity across existing environments, orchestration across distributed infrastructure, and governed intelligence at scale.

1. Architectural continuity spanning environments

Continuity is critical because agentic AI is most valuable when intelligence can extend across environments, systems, data, and workflows that public sector organizations already depend on. As AI becomes embedded in execution paths, continuity has to be treated as a system requirement rather than a convenience. Architecture should allow intelligence to operate across existing data center, cloud, network, edge, and client environments without building a separate stack, because few organizations can scale agentic AI economically by creating parallel infrastructures. This also helps reduce disruptions, preserve software compatibility, protect prior investments, and shorten time to value while making it easier to modernize in place.⁶

Intel's x86 platforms provide a familiar foundation across these domains, while AI PC, edge, and data center capabilities create a practical way to run more agent AI workloads close to where data is created and governed. Previously referenced consolidation and performance-per-watt claims for Intel Xeon 6 are relevant because these processors can help architects modernize core infrastructure and free up capacity for AI growth without expanding data center footprints and power consumption or requiring parallel operations.

2. Orchestration across the infrastructure

Many government workloads are already coordinated across data center, cloud, network, edge, and client environments. Agentic AI changes the nature of that coordination by shifting systems from predefined flows to dynamic flows determined at runtime. Orchestration should therefore be treated as a system-level requirement. Architecture should define how work is scheduled, how resources are coordinated, how state is maintained, how data moves across the environment, when GPU use is justified, and when a cloud model is truly required.

In this model, network conditions, data locality, service latency, memory residency, token budget, GPU availability, and I/O throughput become decision inputs to the system itself. Misalignment across environments can affect throughput and costs, but it can also trigger retries, timeouts, unnecessary cloud calls, GPU idle time, and inconsistent outcomes as inefficiencies multiply across repeated agent loops. The CPU remains central to this coordination layer because it manages many of the platform behaviors that keep distributed intelligence coherent, including orchestration, memory access, workload distribution, I/O coordination, security, reliability, GPU input, and operational continuity.

A system-level proof point of this idea is that Intel Xeon 6 is used as the host CPU for NVIDIA DGX Rubin NVL8 systems,⁷ reinforcing the role of the host CPU in GPU-accelerated AI infrastructure. Intel® CPUs are already in many public sector infrastructures, and they are the foundation for making agentic AI work. Upgrading to the latest Intel Xeon 6 processors enables architects to begin running agentic AI today, and as their AI needs grow, scale to GPUs built on the Intel platform.



3. AI governance for trust, efficiency, and scale

Governed intelligence helps enable continuous AI systems to remain trustworthy, observable, and efficient as they scale across data center, cloud, network, edge, and client environments. For public sector agencies, governance should be treated as a mission and compliance requirement rather than a control layer added later. Key architectural design requirements include policy enforcement, protection of sensitive data, verification of trusted execution state, observability, workload placement, energy-aware operations, identity controls for human and non-human actors, and evidence collection for audits, ATO processes, and oversight reviews.

Agentic AI expands the risk surface because agents may access government records, citizen data, operational systems, classified or controlled information, cyber-tooling, and external services while looping actions over time. A local-first, hybrid model helps reduce that exposure by keeping sensitive and repetitive context closer to the systems and users that are allowed to access it. Additionally, Intel® confidential computing and attestation capabilities can help verify trusted execution before sensitive data, credentials, or policies are released. The outcome is a more defensible government operating model: security, cost, mission scale, and public trust are managed together rather than treated as competing requirements.

These coordination requirements set up the practical design pattern that follows: a local-first agentic AI model that determines which agentic steps should run locally, within governed organizational environments, or in the cloud.

CPU coordination checklist

Use this checklist to evaluate whether your technology architecture can support continuous intelligence after AI systems are operating across workflows, services, and distributed environments.



☐ Determine how workload coordination will occur across services, models, accelerators, and environments so AI can operate reliably under mixed production loads.



☐ Assess how memory, state, and I/O behavior will support persistent execution, model context, feeding GPUs, and data pipeline coordination across the infrastructure.



☐ Evaluate how the host CPU will coordinate GPU-accelerated systems, including memory access, feeding GPUs, workload distribution, model security, throughput, and cluster efficiency.



☐ Analyze whether the CPU-to-GPU ratio is tuned to the workflow with enough CPU, memory, storage, networking, and data-preparation capacity to avoid idle GPU time and maintain predictable utilization under production loads.



☐ Determine whether data movement between storage, networking, accelerators, and compute domains can remain efficient as AI evolves and scales across environments.



☐ Evaluate how security boundaries, isolation, policy enforcement, and attestation are maintained as AI operates across the infrastructure coordination layer.



☐ Assess whether telemetry, observability, patching, and compliance evidence collection are sufficient to support governed operations at production scale.

Design pattern for AI workload placement

A practical agentic AI design pattern for government is to separate workflows into three execution zones:

- Local or near-data execution for sensitive and high-frequency steps
- Agency edge or data center execution for governed, shared services and mission context
- Approved cloud or GPU-accelerated execution for advanced reasoning, large-model inference, or external retrieval

This pattern aligns workload placement with mission, trust, utilization, and compliance requirements. Local execution reduces unnecessary cloud calls and token consumption and protects sensitive context. Agency infrastructure provides governed scale, shared services, observability, evidence collection, and CPU coordination. GPUs are reserved for the portions of the workflow that need accelerator performance and can keep those GPUs productively utilized. Approved cloud models are used selectively when their reasoning capabilities, external data access, authorization posture, or model scale justifies the cost, data movement, and compliance review.

The result is a more predictable cost, utilization, and governance model for agentic AI. The operating principle is straightforward: run routine, sensitive, and repetitive agent tasks closer to the user or data and then invoke cloud models or GPU acceleration only when tasks require them.

Where is as important as what

When testing [SuperClaw](#) versus cloud-only agentic AI solutions, SuperClaw demonstrated up to 70% reduction in average cloud compute token consumption running relevant workloads.⁸ Intel's analysis of idle GPUs further supports this: accelerator-heavy deployments can waste budget when CPU, memory, I/O, retrieval, and orchestration are not sized to keep GPUs busy.¹ Together, these proof points support the same architecture argument that workload placement should reduce operating costs, improve responsiveness, increase utilization, and limit unnecessary exposure of sensitive public sector data while preserving access to cloud-scale and GPU-scale intelligence when needed.

For technical architects, this changes the design question from "Which single platform runs AI?" to "Which part of the agent loop should run where?" For example, a document-heavy workflow might keep file access, summarization, context preparation, policy checks, and draft generation local or inside a controlled data center and send only the minimum necessary prompts or reasoning requests to a cloud model. This reduces the amount of sensitive context sent beyond local control and lowers the token load associated with repetitious steps. The benefit for government agencies is not merely fewer tokens—it is a more scalable operating model that can support more workflows without compounding increases in cloud costs, latency, organizational burdens, and risk of data exposure.

Reference architecture for local-first agentic AI

The design for any AI-ready architecture is simple:

- Keep sensitive and repetitive work close to the data.
- Escalate only what requires external or large-model reasoning.
- Design every step so cost, control, and quality can be measured per completed workflow.

The following diagram illustrates this architectural principle as the flow of control, policy, and context work through an agentic workflow. The initial layers capture the request, decompose it into tasks, assemble the minimum required context, and apply identity, policy, attestation, and audit controls before routing the work to the appropriate execution environment.

The subsequent layer is the workload placement continuum: private, low-latency actions can remain on the client endpoint or AI PC; governed services and shared retrieval can run at the edge or in the organizational data center; and cloud models can be invoked when advanced reasoning or external retrieval justifies the additional token cost and data movement. The figure's control-plane and data-plane callouts help architects separate governance decisions from the movement of documents, embeddings, prompts, model inputs, intermediate outputs, and final responses.

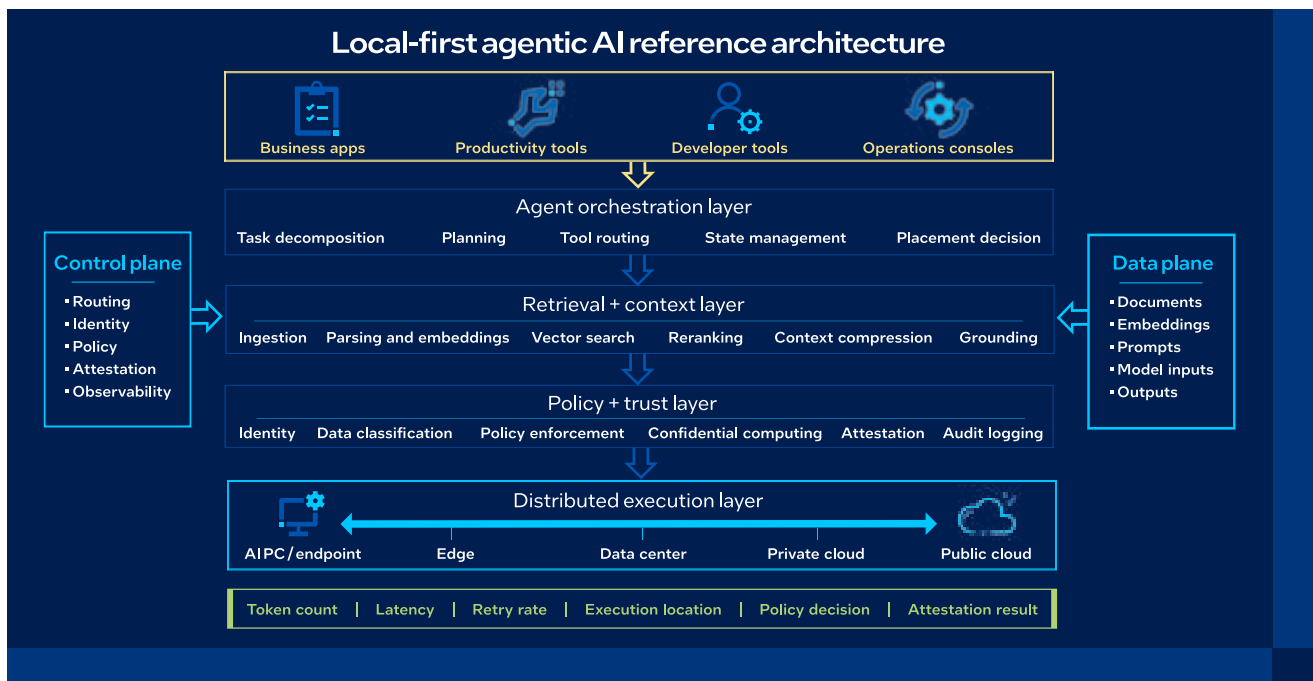


Figure 2. Local-first agentic AI reference architecture.

Local-first agentic AI architecture should be designed as a distributed execution system, not as a single model endpoint. Each of the following five layers of the reference model contributes to the same operating objective of completing more agentic workflows with less unnecessary token consumption, stronger data control, and more predictable performance.

1. Interface and identity context

The interface layer captures requests from mission applications, case-management systems, productivity tools, developer environments, operations consoles, cyber-workflows, and domain-specific agency services. Its role is to normalize the request, identify the user, service, or agent, capture relevant mission context, and pass only the necessary task metadata into orchestration. In government deployments, the interface should preserve audit context such as user or service identity, application source, policy domain, data classification, mission owner, workflow ID, and approval status. Teams can then trace which agent action occurred, why it occurred, which authority permitted it, and which controls applied.

2. Agent orchestration and routing

The orchestration layer is the control point for planning, routing, state management, and accelerator utilization. It decomposes a request into steps, identifies the required tools or models, maintains workflow state, and determines where each step should run. In a local-first architecture, orchestration should not default every action to the cloud or a GPU. Instead, placement decisions should account for task type, data sensitivity, latency requirements, token budget, CPU availability, memory and I/O headroom, GPU utilization, model capability, and policy constraints.

Routine steps such as file parsing, summarization, classification, extraction, tool routing, and draft generation can often run locally or within an agency's edge. More complex reasoning, external retrieval, large-model invocation, or parallelizable inference should be escalated only when those capabilities are required and when the surrounding system can keep GPUs productive.

3. Retrieval, grounding, and context compression

The retrieval and context layer manages the data pipeline that feeds agents. It typically includes document ingestion, parsing, chunking, embedding generation, vector search, reranking, context assembly, and response grounding. In a local-first model, sensitive retrieval steps should run close to where the data resides.

For example, an agent working with internal documents can parse files, generate embeddings, perform vector lookup, and assemble compressed context locally or within a governed organizational environment before invoking a cloud model. This reduces token volume because the cloud model receives only the minimum required context rather than full documents or repeated intermediate prompts. It also reduces data exposure by keeping source content, raw files, and retrieval metadata inside approved boundaries.



4. Policy, trust, and attestation

The policy and trust layer determines whether a given workflow may use specific data, prompts, tools, models, and execution targets. It should include identity and access checks, data classification, policy enforcement, prompt and context filtering, data minimization, secrets handling, attestation, continuous monitoring, and audit logging. For government workloads, this layer should also support Zero Trust principles, ATO evidence collection, FedRAMP-aligned cloud usage where applicable, and documented risk acceptance for high-impact or mission-critical AI use cases.

For sensitive workloads, confidential computing can strengthen the trust boundary by protecting data in use, not only data at rest or in transit. Intel SGX can support fine-grained protected execution for selected application components, keys, prompts, or policy logic, while Intel TDX can support confidential VMs for broader workload isolation. Attestation provides a mechanism to verify that the expected trusted environment is in place before sensitive data, credentials, or policies are released to the workload.

5. Distributed execution across client, edge, data center, and cloud

The distributed execution layer runs work in the most appropriate compute location.

- Client endpoints and AI PCs can handle private, low-latency, high-frequency tasks close to the user.
- Edge systems can process operational data where latency, bandwidth, or data residency are primary constraints.
- Data center infrastructure can host shared retrieval services, governed model endpoints, confidential workloads, GPU-accelerated inference, and organizational observability.
- Cloud models can be used selectively for advanced reasoning, external retrieval, or workloads that require model scale beyond the local environment.

This division of labor reduces unnecessary token consumption in the cloud and avoids underutilization of accelerator capacity while preserving access to GPU-scale and cloud-scale intelligence when it is warranted.



Workflow sequence

A typical local-first agentic AI workflow follows a repeatable sequence.

1. The user or application submits a task with identity, context, and policy metadata.
2. The orchestrator classifies the task and determines whether it should run locally, at the edge, in the data center, or in the cloud.
3. The retrieval layer gathers only the context required to complete the task, applying filtering, summarization, and compression before any escalation.
4. The policy layer evaluates whether sensitive data may leave the local or organizational boundary and whether the target execution environment is trusted.
5. The selected model or tool runs the step, and the orchestrator updates state, evaluates output quality, and determines whether another step is required.
6. Telemetry, lineage, token use, data movement, model choice, and control decisions are logged for audit, optimization, and cost analysis.

How this differs from agents running in the cloud

This architecture differs from a cloud-only agentic AI model because cloud inference becomes a selective resource rather than the default for every step. Intel's SuperClaw approach reflects this principle by combining local-first execution, intelligent task routing, context compression, and reusable agent memory to reduce unnecessary cloud interactions. The technical significance of the SuperClaw reduction in average cloud compute token consumption⁸ is that token reduction comes from system design: fewer unnecessary cloud calls, smaller cloud prompts, more local handling of repeated steps, and more disciplined routing of sensitive context.

Routing policy and escalation model

This model makes workload placement explicit. It helps architects define which tasks remain local by default, which can run inside governed organizational environments, when cloud escalation is justified, and which requests must be blocked because they violate policies or trust boundaries.

Task category	Default execution location	Example tasks	Escalation trigger	Governance requirement
Local only	Client endpoint, AI PC, or near-data local environment	Sensitive file access, document parsing, initial summarization, data classification, personal productivity actions, and policy checks	Escalate only when the task requires reasoning depth, model capability, external data, or SLA performance that local execution cannot meet	Keep source content and intermediate outputs within approved local boundaries and log identity, data classification, and policy decision
Agency execution	Organizational edge, private data center, or governed private cloud	Shared retrieval-augmented generation (RAG) services, domain-specific model endpoints, confidential workloads, audit services, and cross-application orchestration	Escalate to the cloud when approved organizational services cannot satisfy model scale, external retrieval, or advanced reasoning requirements	Apply identity, access control, data minimization, attestation where required, and workflow-level audit logging
Cloud eligible	Public cloud or approved external model service	Advanced reasoning, external data retrieval, very large model inference, tasks where cloud model capability justifies token cost and data movement	Use only after policy approval, context minimization, and evaluation of token budget and data sensitivity	Transmit only the minimum required context and record model choice, token use, execution location, and policy approval
Blocked	No execution	Requests that violate data-handling rules, exceed policy boundaries, require untrusted tools, or target unapproved execution environments	None; return policy-based denial or require human review through approved governance workflow	Log denial reason, policy rule, user or service identity, and remediation path

Observability and validation

Observability should be built into the architecture from the beginning. Public sector agentic AI systems are difficult to govern if teams cannot see which steps ran, which models were used, which data was accessed, how many tokens were consumed, where execution occurred, which identities were involved, and which policies were applied.

Core telemetry should include:

- | | |
|------------------------------------|-------------------------------------|
| ▪ Request ID | ▪ Context size |
| ▪ User, service, or agent identity | ▪ Token count |
| ▪ Workflow ID | ▪ Latency |
| ▪ Mission owner | ▪ Retry count |
| ▪ Task classification | ▪ Policy decision |
| ▪ Execution location | ▪ Attestation result |
| ▪ Model selection | ▪ Data classification |
| ▪ Retrieval source | ▪ Approval status and output status |

This telemetry allows architects to optimize cost per successful task, identify workflows that generate excessive tokens or retries, demonstrate that sensitive data remained within approved boundaries, and support audit, compliance, incident response, and authorization reviews.

A strong reference architecture should include both control-plane and data-plane views. Separating these views helps architects evaluate trust boundaries, data movement, token-cost points, failure modes, and performance bottlenecks independently. The reference architecture should also focus on validation of workflow-level behavior rather than model throughput alone. Additionally, architects should track a variety of metrics to connect architecture decisions to desirable mission outcomes, such as lower operating costs, faster response, stronger governance, reduced data exposure, better GPU utilization, and a clearer path from pilot to production.

Architecture implications for technical architects



Cost, token efficiency, and GPU utilization: Treat cloud inference and GPU acceleration as selective resources, not the default path for every agent step.



Governance and data control: Keep sensitive files, retrieval metadata, prompts, and intermediate outputs inside approved local, edge, or data center environments whenever possible.



Latency and responsiveness: Place high-frequency, low-latency tasks close to the user or data source, and reserve cloud escalation for tasks that require large-model reasoning or external data access.



Operational scalability: The control plane should manage routing, policy, trust, observability, orchestration, and accelerator placement. The data plane should define how documents, embeddings, prompts, model inputs, and outputs move across environments.



Validation focus: Validate the architecture with workflow-level metrics, including cost per completed task, cloud-token consumption per workflow, local versus cloud execution ratio, CPU utilization, GPU utilization, CPU-to-GPU balance, latency by step, retry rate, policy decisions, attestation results, and data-boundary violations.

Intel capabilities mapped to architecture requirements

The architecture model discussed in the previous section defines the outcomes that government leaders need from agentic AI infrastructure. These include lower cost per successful task, less unnecessary data movement, stronger governance, better use of existing infrastructure, higher GPU utilization where accelerators are used, and more flexible workload placement.

This section connects those outcomes to Intel platform capabilities that can help government agencies modernize without creating a separate operating model. Proof points are included only where they support a public sector benefit or architecture decision.

Modernize the infrastructure core

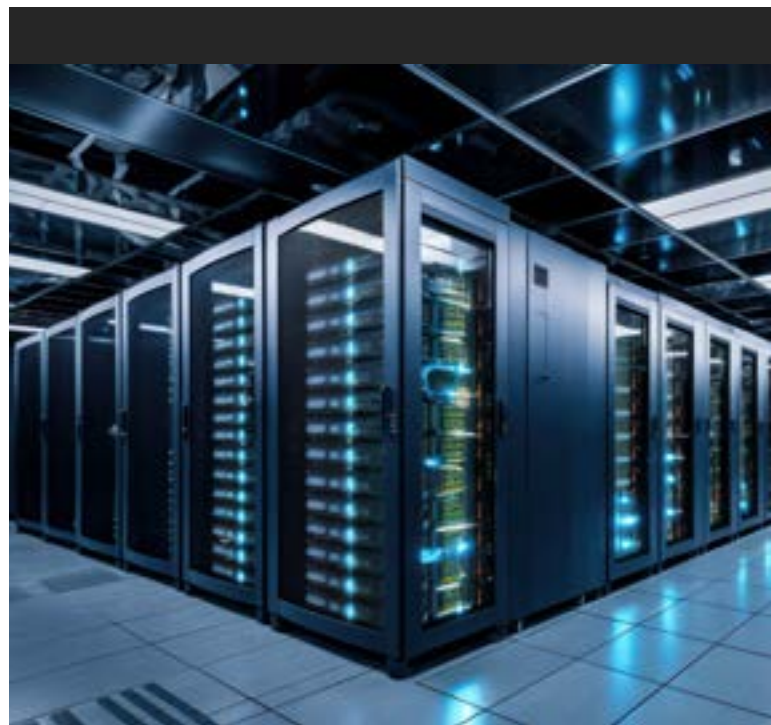
Government agencies can create headroom for agentic AI by modernizing the server foundation they already operate instead of standing up a separate AI stack. The organizational benefit is a more practical path to scale:

- More work per rack
- Better performance per watt
- Lower operational complexity
- Added capacity for AI services without expanding footprint at the same rate

Intel Xeon 6 with Performance-cores (P-cores) can deliver up to 10-to-1 server consolidation in selected use cases and materially lower TCO in selected comparisons, depending on workload, configuration, and baseline environment.³ The architecture implication is that consolidation and efficiency can free capacity for agentic AI growth while preserving operational continuity.

Agencies can improve responsiveness and reduce unnecessary data movement by running more high-frequency AI steps closer to users, devices, and operational processes. The organizational benefit is faster interaction, stronger local control, and better use of cloud-token budgets for tasks that truly require larger models.

Intel® Core™ and Intel® Core™ Ultra platforms support local AI execution and distributed inferencing across client and edge environments, making it practical to keep tasks like summarization, content preparation, classification, and personal productivity actions closer to the user instead of escalating every step to the cloud. With these platforms, agencies can extend AI across more endpoints without making distributed operations harder to manage.



Run more AI work close to the data

Government agencies can improve AI infrastructure efficiency by treating the CPU as the coordination layer that keeps heterogeneous systems productive, including GPU-accelerated environments. The organizational benefit is higher utilization, fewer bottlenecks, and more disciplined accelerator use across mission workflows.

Intel Xeon 6 with P-cores includes built-in Intel® Advanced Matrix Extensions (Intel® AMX) acceleration for AI inference and related deep learning operations, while NVIDIA's DGX Rubin NVL8 announcement identifies Intel Xeon 6 as the host CPU in those GPU-accelerated systems.⁷ Intel's analysis of idle GPUs further reinforces the point that the CPU is a utilization lever because it helps coordinate orchestration, memory access, workload distribution, I/O, model security, and feeding of accelerators.¹ The architecture implication is flexibility, because many inference and RAG-adjacent workloads can remain on CPU-based platforms for cost, simplicity, and operational continuity, while GPUs are reserved and kept efficiently utilized for workloads that require them.

Government agencies can reduce pipeline overhead by moving high-volume data more efficiently within the CPU platform. The organizational benefit is more CPU headroom for orchestration, application logic, and AI-related services without adding separate acceleration tiers for common transfer and transformation tasks. For example, Intel® Data Streaming Accelerator (Intel® DSA) 2.0 on Intel Xeon 6 processors delivers up to 2x more throughput performance than the previous version,⁹ up to 2.85x higher IOPS, and up to 65% latency reduction for large-packet sequential read in one benchmark test.¹⁰ The architecture implication is that routine data movement and transformation can scale more efficiently on the platform that is already coordinating the workflow.

Public sector agencies can scale governed AI workflows without sacrificing security, compression, and performance. They can more efficiently handle cryptographic and compression-heavy operations while preserving CPU resources for orchestration, inference services, and mission applications. [Intel® QuickAssist Technology \(Intel® QAT\)](#) offloads cryptographic and compression/decompression operations from CPU cores on supported Intel Xeon platforms, including Intel Xeon 6. The architecture implication is that security and data-reduction work can grow with the agentic pipeline without consuming the same general-purpose resources needed to coordinate workflows.



Agencies can also improve responsiveness in data-heavy workflows by accelerating filtering, scanning, compression, and decompression closer to where data is processed. The organizational benefit is faster retrieval, more efficient context preparation, and lower overhead for RAG and analytics-driven agentic AI. Intel® In-Memory Analytics Accelerator (Intel® IAA) 2.0 on Intel Xeon 6 processors delivers 2x the data bandwidth of the previous version, 60 GB/s versus 30 GB/s,⁹ and supports scan, filter, compression, and decompression operations for database and analytics workloads. For agentic AI, this supports the retrieval and context layer because faster data preparation can improve workflow responsiveness before model execution begins.



Reduce data movement and pipeline overhead

Government agencies can put AI services closer to operational data when network and edge infrastructure can support predictable movement of data, model state, and service traffic. The organizational benefit is lower latency, better virtualization efficiency, and more consistent behavior for priority workloads without building a separate network architecture.

Ethernet capabilities such as single root I/O virtualization (SR-IOV), remote direct memory access (RDMA), and Intel's Application Device Queues (ADQ) can help reduce software overhead and improve priority traffic handling, while Intel® Xeon® 6 SoC for networking and edge delivers 2.4x more RAN capacity⁴ and up to 70% better performance per watt versus the prior generation for selected network and edge workloads.⁵ The architecture implication is that distributed AI services can scale closer to mission data without exceeding power, latency, or footprint constraints.

Public sector agencies can improve isolation and preserve compute headroom by separating infrastructure services from mission application workloads. The organizational benefit is cleaner trust boundaries, more predictable performance, and more scalable multitenant operations as orchestration, retrieval, model-serving, storage, network, security, and telemetry demands grow.

Intel® Image Processing Unit (Intel® IPU) offloads infrastructure services such as networking, storage, security processing, and telemetry from the host CPU by separating infrastructure functions from tenant or application workloads. The architecture implication is that data center and edge environments can scale shared AI services with less contention on the host platform.

Protect sensitive workflows

Government agencies can strengthen trust in agentic AI by protecting sensitive data, model assets, prompts, retrieval results, policy logic, and orchestration context while they are actively in use. The organizational benefit is greater confidence that sensitive workflows can operate across cloud, data center, and edge environments without unnecessarily exposing critical context.

Intel SGX provides enclave-based isolation for selected application components and sensitive code paths, while Intel TDX extends confidential computing to VMs for broader workload isolation. Intel confidential AI reference materials show how Intel TDX-based confidential VMs and remote attestation can be used to help protect large language model (LLM) inference workflows, including model and user data, throughout the service life cycle.

The architecture implication is stronger control over sensitive agentic workflows without requiring every workload to remain locked in a single environment.

Attestation is the connective tissue between confidential execution and governed AI. It helps verify that a workload is running in the intended trusted state before sensitive data, models, credentials, and policies are released. Intel's confidential AI reference architecture describes remote attestation as part of validating the security status of an LLM inference service environment. This matters for agentic AI because trusted execution cannot be assumed when agents begin acting across distributed systems. Attestation gives architects a mechanism to verify the execution environment before sensitive context, policies, and model assets are exposed.

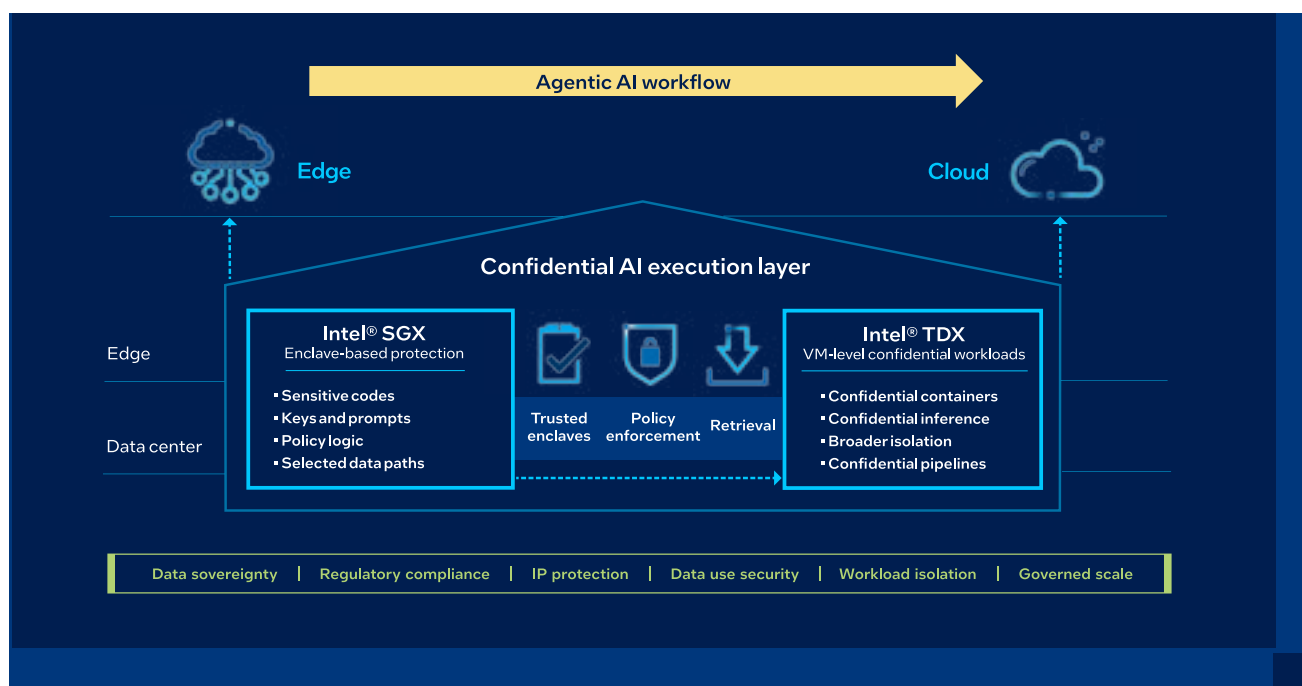


Figure 3. Confidential AI and agentic AI infrastructure with Intel® confidential computing.



Preserve software portability

Public sector agencies can reduce implementation friction by preserving software portability across frameworks, runtimes, and heterogeneous hardware environments. The organizational benefit is a faster path from pilot to production with fewer separate development tracks by environment. For example, open ecosystem software such as oneAPI, Open Platform for Enterprise AI (OPEA), and OpenVINO™ toolkit can help architects improve portability, optimize AI workloads, and support deployment across client, edge, data center, and cloud environments. The architecture implication is a more consistent implementation path that preserves optimization options without fragmenting the operating model.

Government agencies can make software choices more actionable by matching tools, optimization libraries, AI frameworks, and deployment software to specific architecture goals like model optimization, edge deployment, open orchestration, performance tuning, and cloud-to-edge portability. The organizational benefit is a clearer implementation path for agentic AI on existing infrastructure without a complete redesign. Intel provides software resources across oneAPI, OPEA, the OpenVINO toolkit, and related optimization and deployment tools that can support heterogeneous AI development and deployment.



Conclusion

The challenge for government is no longer whether AI can be deployed, but how can organizations implement it in a way that improves mission outcomes without weakening sovereignty, security, compliance, public trust, or fiscal discipline.

Agentic AI increases the potential value of that effort by enabling systems that can sense, reason, retrieve, and act across workflows and distributed operations, but it also raises the architectural bar.

Success depends on infrastructure that:

- Extends intelligence across the installed base
- Coordinates heterogeneous resources under real operating conditions
- Preserves governance as workflows scale
- Avoids both unnecessary cloud-token consumption and underutilized GPU capacity

Intel technologies can support:

- Continuity
- Workload coordination
- Infrastructure efficiency
- Trusted execution
- Software portability
- Evidence-based operations
- Balanced CPU-to-GPU utilization across the architecture

The winning architecture will not be the one that sends the most work to the largest model or buys the most GPUs, but the one that places each step where it can deliver the best mission outcome with the least unnecessary cost, risk, latency, data movement, and operational complexity.



Recommended resources

[Continuity with VMware](#)

Government infrastructure architects can use this webcast to evaluate how to extend AI and modernization strategies across the installed base while preserving operational consistency.

[Smart modernization with Dell and Intel](#)

This solution overview helps architects understand how refresh and modernization approaches can help reduce complexity while preparing existing infrastructure for AI-related demands.

[Modernize government data centers for AI](#)

This white paper helps public sector architects evaluate how data center modernization can support performance, efficiency, and scalability requirements as AI moves into production environments.

[Data center developer technical library](#)

This is a useful resource for architects who need deeper technical material on infrastructure, workload behavior, optimization, and platform guidance related to data center AI deployments.

[Intel® AI Assistant Builder](#)

Architects and developers can use this website to evaluate how local AI assistant and agent capabilities can be built and tested on Intel platforms as part of a local-first or hybrid agentic AI design pattern.

[Confidential Computing Zoo](#)

Architects can use this GitHub collection of solutions to evaluate practical confidential AI deployment examples, including cloud-native AI inference, federated learning, big data analytics, confidential containers, Intel SGX, Intel TDX, and remote attestation.

[Go to the USAI website](#)

[Read the e-book: Reshaping government AI with smarter systems](#)

[Read the brief: The new economics of AI for government](#)

The agentic AI environment assessment framework

This practical framework helps infrastructure architects evaluate whether their current government environments can support agentic AI without unnecessary rebuilding. It is best used after the main architecture model has been reviewed because it turns the white paper's design principles into an assessment tool for potential use cases.

1. Inventory the current workflow and operating model.

Identify the organization's:

- Workflow name
- Agency or mission owner
- Unit of work
- Reporting-period volume
- Target outcome
- Current deployment model
- Data classification
- Approving authority

This establishes which processes are being evaluated and whether they currently run in centralized, hybrid, cloud, edge, disconnected, or mission-constrained environments.

2. Baseline workflow cost, efficiency, and utilization.

Measure:

- Total workflow cost
- Completed units
- Average power draw
- Elapsed processing time
- Total response time
- Total queries
- Actual resource used
- Available capacity
- CPU utilization
- GPU utilization
- Accelerator wait time
- Local vs cloud execution ratio

These inputs provide a repeatable view of cost, energy use, responsiveness, infrastructure efficiency, and accelerator productivity over time.

3. Calculate the baseline operating metrics.

Use a consistent set of formulas to compare current-state performance over time:

- $\text{Cost per case} = \text{total workflow cost} \div \text{by completed cases}$
- $\text{Watts per query} = \text{average power draw} \times \text{elapsed time} \div \text{queries processed}$
- $\text{Average latency} = \text{total response time} \div \text{total queries}$
- $\text{Utilization} = \text{actual resource used} \div \text{available capacity}$
- $\text{GPU utilization} = \text{active GPU time} \div \text{available GPU time}$
- $\text{CPU-to-GPU balance} = \text{CPU capacity actively supporting the workflow} \div \text{GPU capacity actively supporting the workflow}$

4. Measure governance and control maturity.

Evaluate whether the environment is ready for governed AI operation at scale:

- $\text{Lineage completeness} = \text{required lineage fields present} \div \text{required lineage fields}$
- $\text{Control effectiveness} = \text{controls passed} \div \text{controls tested}$

These measures help determine whether traceability and control performance are improving as workflows evolve.

5. Compare modernization versus rebuild scenarios.

After the baseline is established, compare the impact of modernizing existing infrastructure against building a separate AI environment. The comparison should assess whether a refresh path improves cost per case, watts per query, utilization, and control maturity more efficiently than a parallel deployment model with duplicated infrastructure and operations.

6. Evaluate AI-readiness gaps.

Identify where the current environment needs improvement to support agentic AI. Typical assessment areas include:

- Workload coordination
- CPU headroom
- Memory and I/O behavior
- GPU utilization
- CPU-to-GPU balance
- Data movement
- Security and attestation
- Observability
- Compliance evidence collection

7. Confirm authorization and oversight readiness.

Determine whether the workflow can produce the documentation, telemetry, control evidence, risk acceptance artifacts, and operational playbooks needed for security authorization, acquisition review, incident response, and ongoing governance.

A useful way to apply this framework is to start with one high-value workflow, baseline it using a repeatable structure, and then compare the current state to a target-state modernization plan. The same structure can be used to track workflow inventory, cost and energy, lineage and controls, GPU utilization, and modernization readiness as workloads scale.

A simple scoring model can make this assessment easier to apply across multiple workflows. One practical approach is to score each workflow on scale of a 1 to 5 across seven of the following baseline dimensions and then calculate a weighted modernization readiness score to compare candidates for refresh, optimization, or redesign.

1. Cost per case (weight: 20%). Score 1 if cost per case is significantly above the business target; score 5 if it is consistently at or below target.

2. Watts per query (weight: 15%). Score 1 if energy use per query is materially above the baseline objective; score 5 if the workflow operates efficiently within the target range.

3. Average latency (weight: 15%). Score 1 if response time regularly misses service expectations; score 5 if latency is consistently within acceptable business thresholds.

4. Utilization (weight: 15%). Score 1 if infrastructure is materially underused or overly constrained; score 5 if available capacity is being used efficiently without creating bottlenecks.

5. Lineage completeness (weight: 10%). Score 1 if required lineage fields are frequently missing; score 5 if lineage is consistently complete and traceable.

6. Control effectiveness (weight: 10%). Score 1 if key controls fail frequently or remain untested; score 5 if controls consistently pass and exceptions are minimal.

7. GPU utilization and CPU-to-GPU balance (weight: 15%). Score 1 if GPU resources are frequently idle because CPU, memory, I/O, retrieval, or orchestration cannot keep them fed; score 5 if GPU use is consistently tied to workflows that benefit from GPU execution and show strong utilization.

The weighted modernization readiness score can then be calculated as: (cost per case score x 0.20) + (watts per query score x 0.15) + (average latency score x 0.15) + (utilization score x 0.15) + (lineage completeness score x 0.10) + (control effectiveness score x 0.10) + (GPU utilization and CPU-to-GPU balance score x 0.15).

As a practical guide:

- Workflows scoring below 2.5 might require redesign or remediation before scaling.
- Workflows scoring between 2.5 and 3.9 might be good candidates for targeted modernization.
- Workflows scoring 4.0 or higher might be well positioned for broader agentic AI deployment.

¹ Based on “Avoid the GPU Idle Tax: Choosing the Right CPU:GPU Ratios for Agentic AI,” a blog by Intel, May 2026. For more information, visit community.intel.com/t5/Blogs/Tech-Innovation/Artificial-Intelligence-AI/Avoid-the-GPU-Idle-Tax-Choosing-the-Right-CPU-to-GPU-Ratios-for/post/1749346.

² Based on “Towards Understanding, Analyzing, and Optimizing Agentic AI Execution: A CPU-Centric Perspective,” a research paper coauthored by the Georgia Institute of Technology and Intel, April 2026. For more information, visit arxiv.org/pdf/2511.00739.

³ See [7T21] at intel.com/processorclaims: Intel® Xeon® 6. Results may vary.

⁴ See [7ND21] at intel.com/processorclaims: Intel® Xeon® 6. Results may vary.

⁵ See [7ND22] at intel.com/processorclaims: Intel® Xeon® 6. Results may vary.

⁶ Based on “Intel® Xeon® 6 Processors with Performance-cores (P-cores): Reliability, Availability, and Serviceability,” a technical paper by Intel, March 2025. For more information, visit <https://www.intel.com/content/www/us/en/content-details/848948/content.html>.

⁷ Based on “Intel Xeon 6 used as Host CPUs in NVIDIA DGX Rubin NVL8 Systems,” a release by Intel Newsroom, March 2026. For more information, visit newsroom.intel.com/data-center/intel-xeon-6-used-as-host-cpus-in-nvidia-dgx-rubin-nvl8-systems.

⁸ Token consumption benchmark testing based on combination of table indexing and query tools/skills workloads. Cloud LLM based on GLM-5 model available on OpenRouter: [GLM 5 - API Pricing & Benchmarks](https://openrouter.ai/docs/pricing). Local LLM based on quantized Qwen 3.6-35B-A3B model - https://huggingface.co/unsloth/Qwen3.6-35B-A3B-GGUF/blob/main/Qwen3.6-35B-A3B-UD-Q4_K_XL.gguf – served with llama.cpp with “thinking mode” set to “off” on the Intel Core Ultra Series 3 processor. Testing was conducted on Intel Core Ultra X7 358H system with Intel Arc B390 built-in GPU and 64GB of memory running on Microsoft Windows 11 Pro. Results as of May 9, 2026.

⁹ Based on Intel internal assessment. Results may vary.

¹⁰ See [N16] at intel.com/processorclaims: 5th Generation Intel® Xeon® Scalable Processors. Results may vary.

SuperClaw is built based on the OpenCode framework, with additional hybrid AI capabilities, privacy controls, local context management, model routing, governance, and platform optimization developed by Intel.

Performance varies by use, configuration and other factors. Learn more at [www.Intel.com/PerformanceIndex](https://www.intel.com/PerformanceIndex). Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details.

AI features may require software purchase, subscription or enablement by a software or platform provider, or may have specific configuration or compatibility requirements. Data latency, cost, and privacy advantages refer to non-cloud-based AI apps. Learn more at www.intel.com/AIPC.

No product or component can be absolutely secure.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel technologies may require enabled hardware, software, or service activation.

© Intel Corporation. Intel, the Intel logo, Intel Core, Xeon, OpenVINO, the OpenVINO logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.