Why Enterprise Holographic AI Belongs at the Edge

What we learned from building and validating an edge-first holographic AI solution

Based on the Cisco Validated Design for Cisco Unified Edge for AI-Driven Holographic Solutions

September 2026

Purpose

This white paper explains why an edge-first, cloud-enabled architecture is a strong approach for enterprise holographic AI. It compares cloud-only and edge-first deployment models and examines their tradeoffs across responsiveness, resilience, data governance, operational control, and scalability. It also shows how Cisco Unified Edge can support local AI inference and enterprise knowledge services, with Cisco Intersight® for fleet management and optional NVIDIA NIM and Splunk® capabilities.

Audience

This white paper is intended for business and technical decision-makers evaluating holographic AI and other interactive AI workloads at the edge. The audience includes CIOs, CTOs, digital-transformation and customer-experience leaders, enterprise and solution architects, AI/ML platform teams, technology partners, and industry analysts.

Contents

Executive summary 3
Why holographic AI changes the infrastructure design 3
Keeps the live interaction close to the user 4
Compare the three placement models 4
Uses an edge-first, cloud-enabled design 4
How we organized the solution 5
Optional GPU serving with NVIDIA NIM 6
Observe the complete service with Splunk 6
How one request moves through the system 7
Measure the whole experience, not just the model 7
Plan for failure behavior 8
Protect the complete content and model path 8
Start with a use case that has measurable value 9
Define success before scaling 9
Move from demo to pilot in five steps 10
What we learned from the demo 10
Learn more 11
Related implementation guidance 11
Source and scope 11

Executive summary

A holographic assistant is different from a chatbot on a larger screen. The user expects a natural conversation, synchronized audio and video, and a response without a noticeable pause. Even a short delay can interrupt turn-taking and make the experience feel artificial.

The first design decision is where to run the live path. A cloud service can provide elastic capacity, but every turn then depends on the WAN. A hybrid design keeps some services local but adds more operating boundaries. For the validated solution, we kept the latency-sensitive application, retrieval, and inference services at the site and used central tools for policy and lifecycle management.

The June 2026 Cisco Validated Design (CVD) shows this approach using Cisco Unified Edge, Intel® Xeon® 6 processors, Proto Spatial Computing Platform (Proto) holographic endpoints, Arcee Foundation Models, Canonical Ubuntu, Cisco® networking, and Cisco Intersight. We separated the endpoint, application, knowledge, and model services so that each layer can be secured, scaled, monitored, and recovered independently.

- **Our design choice:** keep the conversational hot path local, connect it to approved enterprise knowledge, and manage every site as part of one fleet.

This paper explains the design decisions behind the demo, the tradeoffs we considered, and the work needed to move from a demonstration to a production pilot.

Why holographic AI changes the infrastructure design

A holographic interaction combines user input, application logic, knowledge retrieval, model inference, response streaming, audio, and volumetric presentation. The user does not see separate services. They experience one conversation, so a delay or failure in any stage affects the whole experience.

We focused on three requirements:

- **Fast enough for a conversation:** The system must respond while the user is still engaged. This means measuring the complete path, not only model inference.
- **Grounded in enterprise information:** Answers about products, policies, procedures, or services should use approved sources and provide enough context for review.
- **Able to continue at the site:** The core experience should not stop because a WAN link is slow or unavailable.

These requirements made service placement part of the user-experience design, not only an infrastructure choice.

Keeps the live interaction close to the user

The CVD uses the term “edge gap” for the distance between a cloud data center and the site where an interactive AI service is used. That gap includes WAN latency, jitter, bandwidth limits, outages, and policy boundaries.

This does not mean disconnecting the solution from the cloud. It means keeping the functions that affect the live conversation near the user and centralizing the functions that can tolerate WAN delay.

Compare the three placement models

Table 1. Placement choices for the live holographic AI path

Factor	Cloud-only	Regional/hybrid	Edge-first
Live response path	Every turn crosses the WAN.	Selected services remain remote.	Conversational hot path stays local.
Primary strength	Elastic capacity and centralized operations	Shared resources with some local control	Predictable timing, locality, and autonomy
Primary tradeoff	WAN delay, jitter, dependency, and data movement	More service boundaries and failure dependencies	Local capacity planning and site operations
Best fit	Asynchronous or non-sensitive interactions	Controlled WAN and common services across sites	Live, site-critical, or governed experiences

For a live holographic interaction, the edge-first option gives the most predictable experience. A cloud-only design can still make sense for burst capacity, training, or services that are not sensitive to delay.

Uses an edge-first, cloud-enabled design

For this type of assistant, we recommend the following placement:

- **Run locally:** endpoint integration, application orchestration, retrieval, model serving, and response streaming
- **Manage centrally:** inventory, policy, lifecycle, blueprints, software distribution, and fleet consistency
- **Use the cloud selectively:** training, large batch jobs, optional overflow, external data sources, and analytics that do not sit in the live conversation path

This removes avoidable WAN round trips from the conversation while preserving centralized policy, updates, and fleet operations.

How we organized the solution

We divided the validated design into four functional layers. The user sees one assistant, but the operations team can manage each service boundary separately.

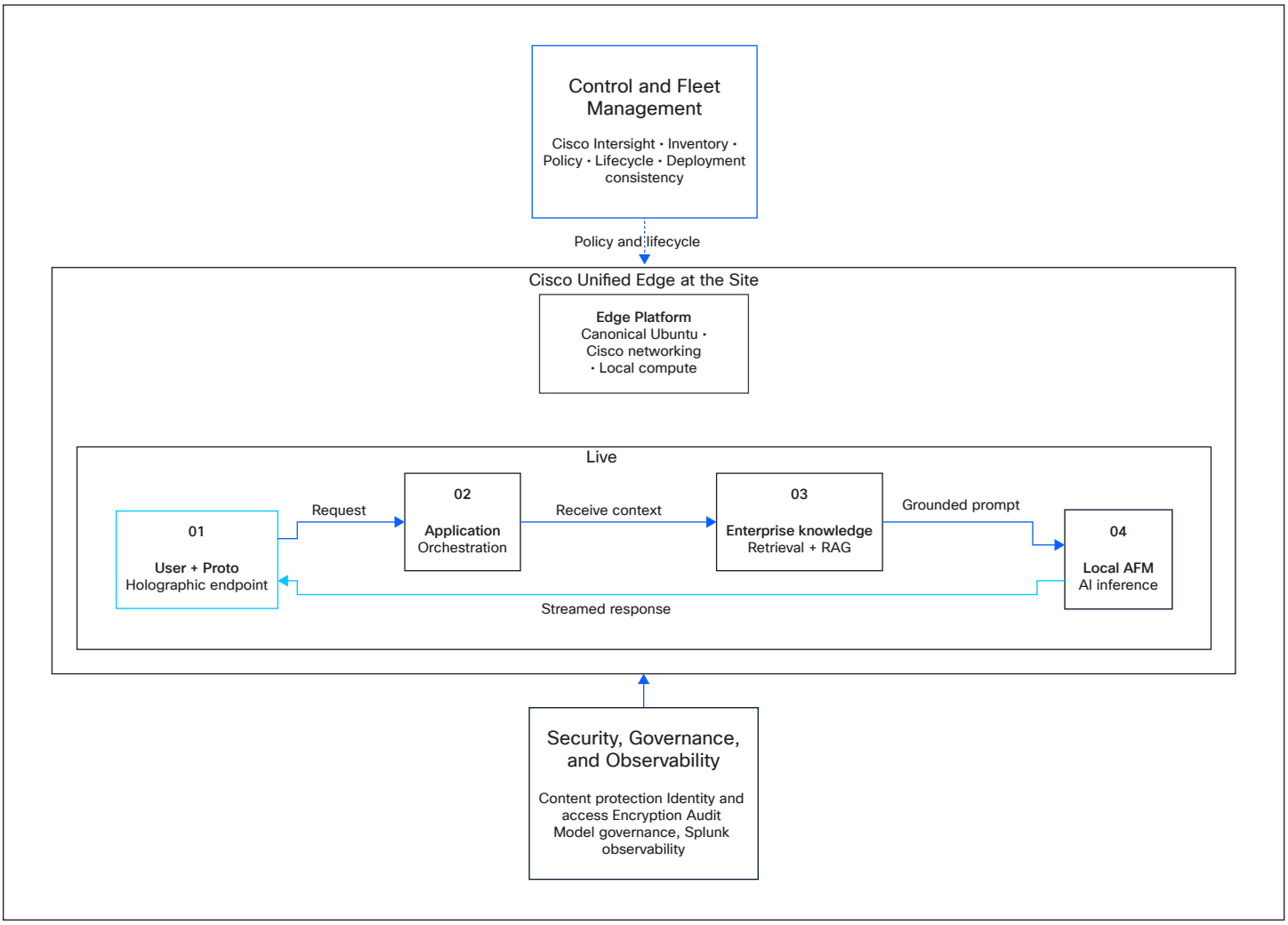

Figure 1. Simplified edge-first interaction architecture derived from the Cisco CVD

1. **Interaction layer**

The Proto holographic endpoint is the user-facing layer. It captures the request and presents the response through the digital human. In the demo, ProtoOS handled the device functions while the AI services ran behind it.

2. **Application layer**

The application service manages the session. It receives the request, applies the workflow, decides whether enterprise context is required, calls the knowledge and model services, and streams the response back to the endpoint.

3. **Knowledge layer**

The knowledge service prepares approved enterprise content for retrieval. It handles ingestion, chunking, embeddings, metadata, indexing, access rules, and source updates.

Retrieval-Augmented Generation (RAG) lets the assistant answer from curated sources instead of relying only on the model's general knowledge. Direct questions can bypass retrieval; policy, product, service, and procedure questions should normally use grounded context.

4. Model layer

The model service generates the answer and streams it to the application. The validated design used an Arcee Foundation Model served locally on Intel Xeon processors. The important design point is the service boundary: the application calls a controlled model endpoint, so the serving implementation can change without changing the endpoint experience.

Cisco Unified Edge provides the local compute and networking platform. Canonical Ubuntu provides the host operating environment, and Cisco Intersight provides inventory, policy, lifecycle, and fleet-level management.

This is what turns one carefully configured demo into a repeatable deployment model. Site services stay local, while approved configurations and operational controls remain consistent across the fleet.

Optional GPU serving with NVIDIA NIM

For a production GPU path, NVIDIA NIM can standardize how a compatible model is deployed and served. It provides OpenAI-compatible APIs, health checks, metrics, structured logs, and trace correlation. The application keeps a consistent model endpoint while the approved NIM image or model profile changes through the release process.

NIM was not part of the demonstrated holographic CVD software baseline, so we treat it as an optional extension. Before using it, validate the exact NIM version, model profile, GPU type, VRAM requirement, precision, concurrency, and latency on the target Cisco Unified Edge configuration. Keep the demonstrated CPU path available until the GPU path meets the same functional, security, and recovery tests.

Observe the complete service with Splunk

Splunk Observability Cloud can add a service-level view across the application, retrieval, inference, Kubernetes, Linux, and GPU layers. At a Kubernetes-based edge site, deploy the Splunk Distribution of the OpenTelemetry Collector as an agent on each node and a gateway for site-level processing and export. For NIM, collect the native metrics, structured logs, and trace identifiers exposed by the serving container.

For this use case, the useful service map starts when the user finishes speaking and ends when the endpoint presents the response. Track end-to-end latency, time to first token, tokens per second, retrieval duration and relevance, application errors, GPU or CPU saturation, queue depth, and site availability. Tag the telemetry with site, cluster, node, model, deployment, endpoint, and request ID so a slow interaction can be traced across services.

Keep telemetry out of the live path. The site experience must continue if the collector, gateway, or WAN link is unavailable; buffer locally and export when connectivity returns. Do not export prompts, retrieved content, or generated answers by default. Capture metadata and correlation IDs unless the security and data-governance teams approve content logging.

How one request moves through the system

1. A grounded request follows this sequence:
2. The user starts a request at the Proto endpoint.
3. The endpoint sends the request to the application service.
4. The application decides whether the request needs enterprise context.
5. When context is required, the retrieval service returns relevant approved content.
6. The application combines the request and context and sends the prompt to the model service.
7. The model returns a streamed response through the application.
8. The endpoint presents the response through the avatar, audio, or application interface.

A direct question skips retrieval. A policy, product, service, or procedure question normally uses the grounded path. The application remains the decision point, so the endpoint does not need a different workflow for each intent.

- The endpoint presents the experience. The application and platform services carry the enterprise responsibilities behind it.

Measure the whole experience, not just the model

Model speed is only one part of a natural holographic conversation. The complete path includes capture, application processing, optional retrieval, inference, streaming, network transport, and rendering. We therefore measure the full interaction.

Table 2. CVD latency targets for a fluid interaction

Metric	CVD target SLO	Why it matters
End-to-end interaction latency	< 1 second	From the user utterance to initiation of the avatar response
Local inference latency	< 64 ms first chunk	Helps the application begin streaming a response promptly
Volumetric jitter	< 20 ms	Reduces variance in packet arrival for holographic streaming
Ingress-to-egress path	< 20 ms	Limits network transit time inside the Cisco Unified Edge fabric

These values are CVD design targets, not universal guarantees. In a pilot, test the selected model, quantization, prompt and output length, concurrency, retrieval load, compute allocation, endpoint, and site network under normal and peak conditions.

The CVD also provides an expected serving baseline for AFM 4.5B on an AMX-enabled Intel Xeon platform: about 64 ms to the first token, about 40 ms between tokens, and about 24 tokens per second at concurrency 1. Use this result to check the model-serving layer; it is not the end-to-end interaction target.

Plan for failure behavior

We defined the expected behavior for each major failure:

- **WAN degradation or loss:** Local application, retrieval, and inference services preserve the core interaction. Cloud-based management, analytics, or alternate services may be delayed.
- **Application-service failure:** Request handling stops. Treat this as a high-priority recovery event and show the user a clear message.
- **Retrieval or vector-database failure:** The assistant may fall back to a general mode, but it must not present an ungrounded answer as enterprise-approved.
- **Model-service failure:** The endpoint shows a fallback message or uses an approved alternate model path that has already been tested.
- **Ingestion failure:** The existing index can remain in service while freshness degrades. Alert the content owner and block partial or unapproved updates.
- **Endpoint failure:** The user loses the experience even when backend services are healthy. Monitor the device and maintain a replacement procedure.

Each runbook should identify the owner, alert, user-visible behavior, recovery action, and validation test.

Protect the complete content and model path

An immersive answer can feel authoritative, which makes source and access controls important. Security has to cover document ingestion, storage, retrieval, inference, response delivery, and management.

The design covers six control areas:

- **Content protection:** keep enterprise documents, embeddings, and retrieval data in controlled stores with role-based access
- **Secure communications:** encrypt API and service traffic. For sensitive deployments, use mutual TLS between application and inference services.
- **Consistent access control:** authenticate and authorize ingestion, retrieval, model, administrative, and management actions
- **Data governance:** align service placement, retention, and residency with organizational and regulatory requirements
- **Monitoring and auditability:** log content changes, service access, configuration changes, and AI workflow activity. Retrieval logs should connect a grounded answer to the approved source content used.
- **Model-path trust:** review how prompts, retrieved context, and generated outputs are handled at every model endpoint

In practical terms, the assistant should retrieve only authorized information, and every grounded answer should be traceable to approved content.

Start with a use case that has measurable value

The holographic format attracts attention, but a production use case needs more than a strong demo. We look for a repeated information or expertise gap, approved content, a measurable outcome, and a clear service owner.

Table 3. Examples that connect immersive interaction to measurable value

Experience	Grounded interaction	Useful pilot measures
Healthcare guide	Wayfinding, preparation steps, and service workflows from approved local content	Response time, retrieval accuracy, wayfinding completion, escalations
Retail assistant	Product location, promotions, inventory context, and store services	Product-search time, interaction completion, conversion support, satisfaction
Remote expert	Procedures, diagnostics, and 3D guidance for a technician at the worksite	Time to expertise, recovery time, first-time fix support, avoided travel

The CVD also describes secure virtual advisors, logistics command and control, immersive executive collaboration, and a 24/7 public-service concierge. The architecture stays the same; the content, integrations, risk profile, and success measures change.

Define success before scaling

Before the pilot, define a small scorecard:

- **Interaction responsiveness:** time from the completed user request to the start of the avatar response at normal and peak load
- **Contextual relevance:** percentage of tested questions answered from the correct approved source, plus review of unsupported or misleading answers
- **Reliability and continuity:** successful interaction rate, service availability, and expected behavior during WAN and dependency failures
- **Operational viability:** time to detect and recover, content-update turnaround, alert quality, and site support effort
- **Business outcome:** the selected use-case measure, such as faster wayfinding, lower product-search time, fewer repetitive support questions, or shorter expert-response time
- **User acceptance:** completion rate, satisfaction, and willingness to use the experience again

Use the CVD values as reference points, not as business targets. Establish the current baseline, set thresholds for the selected use case, and test whether users notice the improvement.

Move from demo to pilot in five steps

1. **Choose one useful interaction**

Start with a narrow interaction where speed, presence, or access to expertise matters. Define the user, the task, and what a successful conversation looks like. A focused patient-guidance or product-location flow is easier to validate than an assistant expected to answer every question.

2. **Design the local live path**

Place the endpoint, application, retrieval, and inference services to meet the experience target. Size compute for the model, prompt length, concurrency, and retrieval load. Reserve network and GPU or CPU capacity so other site workloads do not create unpredictable pauses.

3. **Prepare and govern the knowledge**

Identify content owners, approval workflow, metadata, retention policy, and update frequency. Test chunking and retrieval with real questions. Define what the assistant must decline, escalate, or hand to a person.

4. **Instrument the experience and platform**

Measure the complete user interaction as well as model throughput and resource use. Use Cisco Intersight for platform and fleet operations, and Splunk Observability Cloud for cross-service metrics, traces, logs, and events. Tie alerts and runbooks to user impact.

5. **Test failures and repeatability**

Impair the WAN, stop retrieval, restart the model, introduce a bad document, and validate the endpoint fallback. Then rebuild the environment from the approved blueprint. The pilot is ready to scale when the team can reproduce and recover the service.

Production-readiness check: Can another team deploy the same pattern at a second site, connect it to approved local content, meet the experience target, and recover it using documented procedures?

What we learned from the demo

Our main takeaway is practical: a holographic AI solution must be designed as one end-to-end service. The endpoint, application, enterprise knowledge, model, network, and operations tools all affect the user experience. Keeping the live path at the edge reduces avoidable WAN delay, while centralized management keeps the deployment controlled and repeatable. Start with one bounded use case, measure the complete interaction, govern the knowledge, and test failure behavior before scaling.

Learn more

Read the Cisco Validated Design, Cisco Unified Edge for AI-Driven Holographic Solutions Deployment Guide, published June 2026: Open the full Cisco Validated Design

Related implementation guidance

NVIDIA NIM architecture – inference APIs, health checks, observability, and security behavior

NVIDIA NIM model profiles – hardware, VRAM, precision, and model-profile validation

Splunk Kubernetes data collection – OpenTelemetry-based collection of metrics, traces, logs, and events

Cisco Secure AI Factory with NVIDIA FAQ – Cisco Unified Edge and Splunk observability context for enterprise AI

Cisco Unified Edge with Red Hat Solutions – example Splunk monitoring flow for Cisco Unified Edge, vLLM, and NVIDIA GPU telemetry

Source and scope

This paper summarizes the design decisions and lessons from the Cisco CVD. Confirm current product versions, hardware options, configuration steps, and interoperability requirements in the deployment guide before implementation. Performance values shown here are design targets or expected serving baselines, not guarantees.

C11-x6742751-00 09/26