AI Lens: ArcGIS Industry and Technology Pattern

Last generated: July 08, 2026

Table of Contents

Introduction

Organizations across industries are adopting AI to automate workflows, accelerate analysis, and expand access to insights. As AI capabilities mature, architects and IT leaders face new design decisions such as how to integrate AI into existing systems, how to govern AI workloads responsibly, and how geospatial data, capabilities, and reasoning fits.

ArcGIS is a geospatial platform that is enhanced with AI. As a geospatial platform, ArcGIS delivers geospatial data, capabilities, and infrastructure that enable organizations to create, manage, analyze, and map geospatial data. It is enhanced with AI to:

- **Advance geographic science** by automating and streamlining analysis and data processing workflows at scale
- **Improve the experience** by making GIS users more productive, as well as GIS workflows more accessible and efficient
- **Expand the reach and impact** of geospatial data and insights by automating enterprise workflows and enabling agentic applications and systems

AI has become one of the first topics raised when organizations plan new ArcGIS systems. The scope of this lens is to help with that planning. It is written for the architects and technical leaders who have to turn “we should use AI” into concrete design choices, focusing on the patterns, decisions, integration points, and governance that come with putting AI into geospatial systems.

This lens complements and connects the ArcGIS Trust Center, which covers Esri’s AI principles and commitments, and the product documentation covering the features themselves.

Note:

New to ArcGIS? Start with the Introduction to ArcGIS for platform context before working through this lens.

Audience

The lens is intended primarily for technical leaders, IT architects, and IT/GIS professionals who design AI workloads on or with ArcGIS. This includes readers who come from a GIS background and are adding AI to systems they already run, as well as those that know AI well and are working out how geospatial data and capabilities fit into a wider AI strategy. The lens is written with both in mind.

The lens assumes you arrive with a working knowledge of ArcGIS architecture and a general grasp of AI concepts. The next chapter offers a short primer on both, mostly to settle terminology — it does not try to teach AI from the ground up.

How to use this lens

You can read the lens front to back, which is the path we recommend the first time through, or treat it as a reference and pull the chapter you need. A few chapters stand on their own: AI concepts and vocabulary and integration and interoperability.

In particular, the architecture pillars page is worth noting. The ArcGIS Well-Architected Framework is organized around six pillars — reliability, performance and scalability, security, integration, automation, and observability. The pillars chapter in the AI lens does not restate them, it shows where AI workloads change the calculus for each.

This is a living document, refreshed often. Because AI capabilities in ArcGIS move quickly, it refers to them by name and by maturity — announced, private beta, beta, or generally available — rather than by release date. Specific timing shifts often; consult the release notes and the product roadmap for those details.

Additional resources

- The ArcGIS Trust Center is where Esri publishes its AI principles, its commitments, and the AI transparency cards for individual features.
- ArcGIS Living Atlas of the World provides the authoritative content and pretrained models that are continuously referenced in the chapters ahead.
- The Geospatial AI resources page provides capability-specific stories as well as blogs, customer examples, and additional resources.
- The product documentation provides feature-level detail on any products and technologies referenced in this lens.

AI concepts and vocabulary

This chapter establishes shared vocabulary that the rest of the lens depends on. As a primer, it includes concepts and terms centered around AI in general as well as those that are ArcGIS specific. It focuses on those that are most relevant to the target audience for this lens. If the terms here are already familiar, skim for the ArcGIS-specific conventions and move on to the Geospatial AI topic.

AI fundamentals

AI is a broad family of techniques and technologies. The subset most relevant to ArcGIS architecture breaks down along a few key branches.

- **Machine learning (ML)** refers to systems that learn patterns from data rather than following hard-coded rules. A model trained on labeled imagery can learn to identify building footprints; one trained on tabular data can learn to predict failure probabilities for assets. The key architectural implication: ML models need training data, compute for training, and infrastructure for inference — and their outputs are probabilistic, not deterministic.
- **Deep learning (DL)** is a subset of ML built on multi-layer neural networks. It is particularly effective for unstructured data — imagery, text, point clouds — where traditional feature engineering would be impractical. Most of the AI tools in ArcGIS that work with imagery or 3D data use deep learning models under the hood. ML and DL have been part of ArcGIS for over a decade.
- **Computer vision (CV)** is a field of AI focused on extracting structured information from images and video. Object detection, land cover classification, change detection, and feature extraction from satellite or drone imagery all fall under CV. In ArcGIS, computer vision models are delivered as geoprocessing tools, pretrained model packages, and through the arcgis.learn module in the ArcGIS API for Python.
- **Generative AI and large language models (LLMs)** are the newer arrivals. LLMs are trained on massive text corpora and can generate, summarize, translate, and reason about natural language. They are what makes conversational AI assistants possible, and they are the reasoning engine inside AI agents. Generative AI more broadly includes models that produce images, code, and other content from prompts.

A few additional supporting concepts referenced throughout the lens include:

- **Embeddings** are numerical representations of meaning. They convert text, locations, or features into vectors that can be compared for similarity. Embeddings power search (often referred to as semantic search), clustering, and the "knowledge" components of agents.
- **Fine-tuning** adapts a pretrained model to a specific dataset or task. This tends to be a cheaper and faster path than training from scratch when a general, pretrained model gets you 80% of the way there.
- **Grounding** gives an AI system access to reliable, specific context (such as enterprise data, documentation, or schemas) to shape its outputs. An ungrounded LLM generates from its training data alone; a grounded one draws on the information you point it at. The quality of grounding data directly affects the quality of AI outputs.

Agentic AI

An agent is a software entity that can perceive context, reason about goals, and take actions on behalf of a user or organization. Agents are broad in scope, including everything from a simple tool-calling loop to a multi-agent system coordinating across platforms.

What distinguishes agents from other AI patterns is a combination of four properties: they are **goal-oriented** (given objectives rather than step-by-step instructions), **context-aware** (they perceive their environment and the state of their task), **action-capable** (they can invoke tools and APIs, not just generate text), and **autonomous within boundaries** (they decide what to do next, but within constraints set by their designer).

Under the hood, an agent typically combines:

- An **LLM** for reasoning and planning
- **Instructions** (sometimes called a system prompt) that define the agent's operating directive and boundaries
- **Tools** — the actions and external systems the agent can invoke
- **Knowledge** — information sources the agent draws on to ground its reasoning
- **Memory** — state carried across interactions, allowing the agent to build on prior work

Agents are increasingly service-oriented. Two open standards are shaping how they interoperate: **Model Context Protocol (MCP)** defines how an AI model connects to external data sources and tools, and **Agent-to-Agent (A2A)** defines how agents communicate with each other. Both matter for ArcGIS architecture because they determine how geospatial capabilities get exposed to, and consumed by, the broader ecosystem of AI platforms and agents.

Agents often operate in coordination with other agents and tools rather than in isolation. Many enterprise software platforms now offer agentic AI capabilities, creating an ecosystem of agent platforms across cloud providers, business applications, and data platforms. ArcGIS participates in this ecosystem through the standards described above.

ArcGIS-specific terminology

Esri uses a specific set of terms for AI in ArcGIS. The distinctions matter for architecture because they map to different deployment patterns, licensing models, and integration surfaces.

Geospatial AI (shortened to **GeoAI**) is the integration of GIS and AI and represents all AI capabilities Esri builds into ArcGIS. Geospatial AI has three pillars:

- **AI tools and models** — AI-powered geoprocessing, pretrained models, AI services, and end-to-end deep learning workflows. These fuse AI with geospatial data, science, and technology to automate data creation, extraction, and analysis. See the AI tools and models chapter for more information.
- **AI assistants** — in-app natural-language interfaces that help users complete tasks or write code. Built on LLMs and geospatial agents, AI assistants come in two forms: task assistants (which guide users through GIS workflows) and coding assistants (which generate, explain, and troubleshoot Python and Arcade). See the AI assistants chapter for more information.
- **Agentic AI** — AI agents that orchestrate and perform tasks, both with humans as well as autonomously. In ArcGIS, this surfaces through three paths: integration with third-party agentic systems, agentic mapping applications, and geospatial agents. See the Agentic AI chapter for more information.

One additional term introduced here and developed later: a **geospatial AI agent** (or geospatial agent) is an agent specifically configured to handle geospatial tasks. It uses the same components as any agent — LLM, instructions, tools, knowledge, memory — but its tools are GIS tools, its knowledge is geospatial content, and its memory holds spatial context. The Agentic AI chapter covers how these are built and deployed.

Geospatial AI

Geospatial AI is the integration of GIS and AI and represents all AI capabilities Esri builds into ArcGIS.

Capabilities

Geospatial AI in ArcGIS is organized around three capability areas. Each is covered in its own chapter later in the lens; this chapter describes their value proposition and alignment to the architecture of ArcGIS.

- **AI tools and models** automate data creation, extraction, and analysis. They are powered by machine learning, deep learning, and advanced spatial statistics, and they let users extract information from imagery, 3D data, and text, detect patterns, and make predictions from complex datasets. These are the workhorses of applied geospatial science, delivering capabilities that help GIS analysts and data scientists accelerate analytical work at greater scale. AI tools and models are the most mature of the three capabilities, with over a decade of investment behind them.
- **AI assistants** are in-app natural-language interfaces that make GIS more accessible and its users more productive. Task assistants guide users through workflows in apps like ArcGIS Pro, Business Analyst, and Survey123. Coding assistants help users generate and troubleshoot Python and Arcade. Together they lower the barrier for new users while helping experienced practitioners move faster. They are backed by geospatial agents and grounded on the organization's own ArcGIS content.
- **Agentic AI** introduces orchestration, automation, and integration through agents and agentic workflows. Agents can perceive context, reason about goals, and act autonomously or with a human in the loop. In ArcGIS, agentic AI shows up in three places: through integration with third-party agent platforms (where ArcGIS acts as a tool provider via MCP or other mechanisms), through agentic mapping applications (where natural-language interaction drives spatial insights), and through geospatial agents built within ArcGIS itself. This is the newest and fastest-moving of the three capabilities.

The three are not isolated. A single solution might use AI tools and models to extract features from imagery, surface those results through an AI assistant for review, and hand off validated outputs to an agent for downstream distribution.

Value propositions

Geospatial AI delivers value to organizations beyond what GIS or AI can offer on their own.

The geospatial gap in enterprise AI. Most enterprise AI systems are not trained to work with geographic science or extract location-based context. They can reason about text, tabular data, images, and videos; however, they typically cannot answer questions that depend on where things are, how they relate spatially, or how spatial patterns influence outcomes. Geospatial AI bridges this gap by connecting trusted GIS technology with the reasoning power of modern AI.

Automate geospatial workflows. Routine tasks that previously required significant time from specialized staff, such as feature extraction, classification, image inspection, and data QA, can be automated or substantially accelerated. This frees teams to focus on interpretation, validation, and higher-value analysis rather than repetitive production work.

Accelerate analysis at scale. Geospatial AI makes it practical to analyze volumes of imagery, text, and sensor data that would be infeasible by traditional means. Regional-scale land cover classification, national infrastructure monitoring, and pattern detection across millions of records become tractable problems rather than multi-year projects.

Expand access to spatial insights. Assistants and agents make it possible for people without deep GIS expertise to ask spatial questions, find the right data, choose appropriate workflows, and act on results. They also help users learn GIS workflows faster, which broadens the reach and impact of an organization's geospatial investment beyond the GIS team.

Architecture

Geospatial AI does not replace the existing ArcGIS architecture. It extends it.

The base ArcGIS architecture is organized in three tiers: applications, services, and data. Each tier includes both Esri-built and organizational (user/enterprise) built components, for example, Esri delivers broad data and content via Living Atlas whereas an organization brings their own data and content to ArcGIS to use in combination with Esri's data. AI extends this model by adding a new horizontal layer for agents (including Esri-built agents, and customer-built agents) and by introducing AI capabilities into the existing tiers.

Where the three capabilities show up in the architecture:

- **AI tools and models** span all three tiers. Pre-trained and foundational models align with data tier, with Esri-built components provided through Living Atlas. Tools and capabilities align with services tier, including geoprocessing services, AI-powered services, and inference endpoints. Experiences for working with AI tools and models are provided through the application tier, including ArcGIS Pro, ArcGIS Notebooks, and ArcGIS Deep Learning Studio.

- **AI assistants** live in the application tier, surfacing throughout the user experience of ArcGIS applications as in-app interfaces. AI assistants depend on services, models, and data to function.

- **Agentic AI** introduces the new agents layer and opens ArcGIS to external agentic systems through standards-based interoperability (MCP, A2A). Agents orchestrate tools in the services tier, consume data from the services and data tiers, and surface results through applications, including Esri provided applications as well as user-built applications.

ArcGIS plays two roles in the AI ecosystem. As an **AI consumer**, it uses AI internally: LLMs power assistants and agents, ML and DL models power tools. As an **AI producer**, it exposes AI capabilities — tools, services, geospatial content, and agents — for use by external AI systems. An organization's broader AI strategy may rely on ArcGIS in either role or both, and the architectural decisions differ depending on which side of that boundary the design sits.

Finally, geospatial AI introduces several architectural implications that shape every design decision in the chapters ahead:

- **Non-determinism.** AI workloads, particularly those involving LLMs and agents, can produce different outputs from the same input. Systems designed around them need to handle variability rather than assuming repeatable results.

- **New integration surfaces.** MCP and A2A create new boundaries where ArcGIS connects to the broader AI ecosystem. These surfaces need careful review and implementation of authentication, authorization, observability, and governance.

- **Data quality expectations.** AI amplifies the consequences of poor data. Grounding, training, and inference all depend on data that is accurate, well-documented, and governed. The bar for data quality in AI-enhanced systems is higher than it has been for many traditional GIS workflows — a topic explored in depth in the Geospatial data for AI chapter.

Geospatial data for AI

Data is the foundation of every AI system. The accuracy, structure, and accessibility of data directly shape what AI can perceive, reason about, and produce. For geospatial AI in ArcGIS, that foundation is geospatial data, either produced and managed within an organization, built by authoritative providers, or sourced from the community of geospatial organizations and researchers around the world. This chapter covers the data modalities, information models, and quality expectations that AI workloads introduce, and the spatial-specific considerations that come with them.

Modalities

Geospatial AI in ArcGIS works across several data modalities, each with its own characteristics for training and inference. Each is used differently depending on the AI workload: to train models, to ground inference, and to be discovered, queried, and reasoned against through tools. The latter centers around agentic AI workflows and places particular weight on metadata, schemas, and semantic clarity. Note that these modalities are useful categorizations but not definitive boundaries, many data types and workflows fuse data from across these different modalities or use hybrid approaches.

Vector data — feature layers with points, lines, and polygons are the backbone of most GIS systems. For AI tools and models, vector data serves as both training input (labeled features for supervised learning) and inference output (new features generated by a model). For assistants and agents, vector data is queried, filtered, and reasoned against through tools. A natural-language question becomes a spatial query against feature layers, returning the features that answer it. The attribute schemas, topology, and spatial relationships encoded in vector data give all three modes structured context that unstructured formats cannot. In many cases, tabular data without a location column also falls into this category and is integrated into data models and storage options that also apply to vector data.

Imagery — satellite, aerial, drone, and street-level imagery are the foundation for computer vision workflows in ArcGIS, including object detection, classification, segmentation, and change detection. Training requires annotated imagery with known labels; inference runs on new imagery at scale. The spatial resolution, spectral bands, and temporal coverage of the imagery all affect what a model can learn and how well it generalizes. Assistants and agents can also invoke imagery analysis tools and retrieve imagery for context. Learn more about imagery and remote sensing in ArcGIS.

3D data — point clouds, 3D meshes, building models, and scene layers carry spatial structure that two-dimensional representations lose. AI workflows on 3D data include feature extraction from LiDAR, building footprint generation, classification of point clouds, volumetric analysis, and computer vision patterns applied to snapshots of 3D scenes or content. Models trained on 3D modalities need different architectures than those trained on flat imagery, and the data volumes involved make compute planning more significant. Learn more about 3D GIS.

Text and documents — ArcGIS carries text in many forms: attribute text in features and tables (inspection notes, work orders, maintenance logs), document attachments and items (PDFs, reports), survey responses, and metadata. For AI tools and models, ArcGIS supports a broad range of natural language processing (NLP) including classification, entity extraction, summarization, translation, question answering, and sentiment analysis. These are provided through text analysis tools, the `arcgis.learn.text` module of the ArcGIS API for Python, AI-powered services in ArcGIS Online, and pre-trained NLP models in Living Atlas. For assistants and agents, text serves as grounding material and as the medium of interaction itself.

Knowledge graphs — graph-structured representations of entities and the relationships between them, modeling real-world systems that bring together spatial and non-spatial data. ArcGIS Knowledge stores knowledge graphs in a dedicated graph store and supports editing, querying (including the openCypher query language), and link analysis. For AI tools and models, knowledge graphs support pattern detection and investigations that can power fraud, intelligence, and complex relationship discovery workflows. For assistants and agents, knowledge graphs provide exceptionally strong grounding. The semantic structure (entities, relationships, properties, ontology) gives agents a reliable scaffolding to reason against that tends to be more accurate than unstructured retrieval. The ArcGIS Pro Assistant supports the AI-assisted conversion of natural language questions into openCypher queries for use with knowledge graphs. Learn more about ArcGIS Knowledge.

Real-time and time series — sensor feeds, IoT telemetry, real-time streams, and temporal feature classes capture change over time. For AI tools and models, forecasting, anomaly detection, and trend analysis operate on time-stamped spatial data. For assistants and agents, real-time and historical telemetry is queried for current state, recent changes, and patterns — an agent monitoring infrastructure or responding to events draws on these sources through tools. Real-time streaming and historical time series differ architecturally in latency, throughput, and windowing, but both are increasingly central to AI workloads particularly for monitoring, prediction, and event-driven workflows. Learn more about real-time data in ArcGIS.

These modalities are not always separate in practice. Multi-modal AI, or models that combine imagery with text, or vector data with temporal signals, is an active area of development. Esri's geospatial vision language model research, for example, connects satellite imagery with LLMs to allow natural-language queries against image content. As these capabilities mature, the boundaries between modalities will matter less in design than the quality and accessibility of the underlying data.

Geospatial information models

ArcGIS structures geospatial data through information models that carry semantics beyond raw geometry and pixels. These provide additional grounding to AI.

One information model of note is the geodatabase. Geodatabases encode rules and relationships that enable rich geospatial capabilities, such as topologies and networks. Indoor spaces, buildings, and imagery mosaics are other examples of general purpose information models used widely in geospatial workflows with ArcGIS.

Some information models are industry specific, such as the utility network, parcel fabric, pipeline referencing, and roads and highways models. The domain-specific rules and relationships inherent in these models make them particularly valuable to AI assistants and agents. An agent grounded on a utility network model sees not only the geospatial features, it understands connectivity, flow direction, containment, non-spatial assets, and the rules that govern valid network states.

Geospatial information models tend to outperform unstructured document retrieval for geospatial AI workflows. When an assistant or agent can draw on defined schemas, typed relationships, and governed content, its outputs are more reliable than when it must infer structure from flat text. These models are complementary to, not a replacement for, unstructured and traditional, relational data. From an architectural perspective, the incorporation of geospatial information models should be considered as a first-class design decision.

Authoritative geospatial content

AI workflows need reliable, trusted data. In ArcGIS, authoritative geospatial content comes from three sources.

Esri-provided content through ArcGIS Living Atlas of the World includes foundational basemaps, boundary datasets, demographic layers, environmental data, imagery, elevation data, and more. Living Atlas also hosts pre-trained models, including task-specific models (e.g., building footprint extraction, land cover classification with Sentinel-2 imagery) that can be fine-tuned to an organization's data, as well as generalizable models like the Segment Anything Model (SAM) that adapt to a wide range of tasks without specific training. Browse Living Atlas.

Third-party content from trusted contributors and partners brings domain-specific data that Esri does not create directly, including data from national, state and local agencies, weather feeds, commercial imagery, industry-specific datasets, and real-time sensor networks. These sources can serve as training data, as grounding context, or as inference inputs depending on the workflow. Explore data partners.

Organizational content — an organization's own geospatial data, including systems of record that are validated and maintained by their teams. These are often the most valuable input for AI. An organization's asset registry, inspection history, network model, or field survey data carry specificity, relevance, and authority that general-purpose content cannot match. These datasets are uniquely available to AI workloads that are run by this organization, and can bring deep meaning and insights when added to other authoritative sources. This content is also where readiness gaps are most common. Learn more about data management in ArcGIS.

Data quality and AI readiness

Effective AI depends on data that is accurate, well-documented, and governed. This is not a new principle, but AI raises the stakes. A model trained on mislabeled imagery will confidently produce wrong answers. An agent grounded on stale or poorly described content will surface irrelevant results. The failure modes are not always obvious because AI outputs can look plausible even when the underlying data is flawed.

What "AI-ready" means in practice:

- **Accuracy and currency** — data reflects the current state of the world it represents, maintained on a schedule appropriate to its use in AI workflows, and clearly attributed to reflect this currency.
- **Documentation and metadata** — item descriptions, summaries, field descriptions, tags, and additional detailed metadata that are complete enough for both humans and AI systems to understand what the data contains, how it was collected, and what it is appropriate for.
- **Discoverability** — content is findable through a common search interface or pattern, categorized consistently, and organized so that assistants and agents can locate it without human intervention.

- **Governance** — ownership, access controls, sharing rules, and retention policies are defined and enforced.
- **Lineage and provenance** — when AI outputs are derived from specific data sources, the lineage should be traceable. This matters for audit, for debugging, and for trust in the results downstream.

Most organizations have data that is of high value to AI workflows, but that is not yet ready to support them in practice. This gap is usually related to data quality and readiness, and closing that gap is often the highest-leverage preparatory work an organization can do before deploying AI capabilities.

Data access and authorization

AI agents and models need not only access to data, but the right kind of access delivered through governance patterns that ensure security, traceability, and appropriate use. Data in ArcGIS is exposed through multiple patterns, each with different implications for agentic systems.

Web services and data storage — Geospatial data can be accessed through web services that apply business logic and access controls at query time, or as raw data stored in data lakes, databases, or file systems. Web services provide a controlled interface where governance rules are enforced consistently; agents query features through feature services or invoke analysis services that apply predefined workflows. Direct access to stored data offers flexibility but requires agents to be deeply integrated into infrastructure and governance. Most mature AI deployments use a hybrid approach, with agents querying through services for trusted datasets and accessing raw data directly only where necessary.

Authorization and least-privilege access — An agent's data access should be scoped to what it needs for its assigned task. Just as a human user should not have access to all organizational data, an agent should have access only to the data layers it requires to fulfill its function. This principle, combined with service-based access patterns, prevents accidental or malicious exposure of sensitive information. As agents become embedded in organizational processes, authorization patterns become as critical as they are for human users.

Governance and auditability — Every query an agent executes, every dataset it accesses, and every inference it makes should be traceable. Audit logs that record which agent accessed which data, when, and for what purpose are essential for debugging, for compliance, and for building organizational trust in AI-assisted decisions. Governance frameworks should define what data an agent can access, log all interactions, and provide investigators the trail to understand how a particular recommendation or output was derived.

Data creation and update — As agentic systems mature, agents may graduate from read-only access to creating or editing data — submitting new observations, flagging updates, or triggering data validation workflows. This introduces new access considerations: agents that can create or modify data must be subject to even stricter authorization and auditing. The lineage of agent-created data, the verification workflows that validate it, and the rollback mechanisms if an agent introduces errors all become operational requirements. Data governance must evolve to encompass not just discovery and read access, but creation and change management as well.

Spatial data considerations

Location adds dimensions to AI workloads that generic AI guidance does not address. The following are some considerations unique to spatial data.

Location as first-class context. Most AI systems have no inherent sense of geography. They cannot distinguish between two similar addresses in different cities, reason about proximity, or account for the fact that spatial relationships shape outcomes. Geospatial data brings forms of context, including containment, adjacency, distance, connectivity, temporal-spatial patterns, that AI systems can use but cannot derive without purpose-built tools and data. This is a fundamental value proposition of combining GIS with AI.

Temporal dimensions. Many geospatial datasets carry a time component: real-time sensor feeds, time-series imagery, historical feature snapshots, temporal feature classes. AI workloads that operate on this data need to handle temporality explicitly, including time-aware training, temporal windows for inference, and awareness of how patterns evolve. Temporal alignment across time zones, while straightforward in concept, also tends to be a common source of error.

Location embeddings. Location embeddings are spatial-specific, numerical representations of geography that distill complex location data into forms AI models can incorporate efficiently. Embeddings can be applied as input to downstream AI workflows, used to find similarity across locations, or combined with other features to enhance models. Location embeddings represent a new class of geospatial data product worth treating as architecturally first-class alongside raw data. Location embeddings are discussed further in AI tools and models in the context of how they are produced and applied.

Coordinate reference systems and scale. Data collected at different scales, in different projections, or at different resolutions requires careful handling before it can be combined for AI training or inference. A model trained on 30cm imagery will not generalize well to 3m imagery without adaptation. These are familiar GIS concerns, but they carry new weight when data feeds automated pipelines that do not have a human checking each input.

Spatial bias and representativeness. Training data that over-represents urban areas, particular geographies, or certain sensor types will produce models that underperform elsewhere. Architects should evaluate geographic coverage and representativeness as part of model evaluation, not just overall accuracy metrics.

Spatial privacy. Location data can reveal sensitive information about individuals, assets, or operations even when other identifiers are removed. AI workflows that ingest, process, or output spatial data should account for privacy implications — particularly when data crosses organizational boundaries or feeds agents with broad access.

AI tools and models

AI tools and models fuse AI with geospatial data, science, and technology. They work with a variety of data modalities to automate data creation and extraction, as well as perform complex analysis workflows including classification, pattern detection, and prediction/forecasting.

This chapter covers how AI tools and models are delivered, how models move from pretrained to production, and the architectural decisions that come with managing them over time.

How AI tools and models are delivered

ArcGIS provides AI tools and models through several delivery mechanisms, each suited to different user personas and deployment contexts.

AI-powered geoprocessing tools sit alongside other geoprocessing tools in ArcGIS Pro, ArcGIS Online, and ArcGIS Enterprise. They can be found in toolboxes and chained into workflows just like traditional geoprocessing tools, though under the hood they run ML or DL inferencing. The AI is embedded in the tool providing a simplified experience for GIS analysts. Examples include feature extraction from imagery, point cloud classification, and predictive analysis tools. To learn more, see an Introduction to geoprocessing and the GeoAI toolbox.

AI-powered services in ArcGIS Online provide text analysis and image analysis capabilities. These are accessed through ArcGIS applications like Survey123 and Instant Apps, as well as through the ArcGIS REST API and the ArcGIS API for Python (arcgis.ai module). They do not require customers to manage model infrastructure, as Esri operates the services.

Pretrained models in Living Atlas are published as Deep Learning Packages (DLPKs). Each package includes documentation of its intended use, model architecture, training data, performance metrics, input/output specifications, and limitations. Task-specific models, such as building footprint extraction and land cover classification, can be used directly or fine-tuned. Generalizable models like the Segment Anything Model (SAM) adapt to a wide range of tasks without task-specific training. Browse AI models in Living Atlas.

The ArcGIS API for Python exposes two modules central to AI workflows. The arcgis.learn module is the underlying technology Esri uses to build its own AI tools and pretrained models. It is exposed through the ArcGIS API for Python for organizations looking to train deep learning models on their own data using standard frameworks (PyTorch). Additionally, the arcgis.ai module provides access to text and image analysis services from ArcGIS Online.

End-to-end deep learning workflows in ArcGIS Pro, ArcGIS Notebooks, and Deep Learning Studio support the full model lifecycle from data preparation, labeling, training, evaluation, to deployment. These are aimed at data scientists and GIS analysts looking to train custom models rather than use pretrained ones. Deep Learning Studio provides a visual interface for bringing externally trained models into ArcGIS. Learn more in an Introduction to Deep Learning Studio.

Model use and training

ArcGIS supports a spectrum of approaches from off-the-shelf usage to custom models trained from scratch.

Use pretrained models directly. This is the lowest-effort path, and covers many common extraction and classification tasks. If a task-specific model in Living Atlas matches the use case organizations can deploy it without training. The model runs inference on the organization's data using weights trained by Esri on representative datasets.

Fine-tune pretrained models. This applies when a pretrained model gets close to the desired outcome, but not close enough. For example, a pretrained model trained on urban imagery may not produce accurate enough results in rural areas, or a model detects building footprints but not the specific roof types needed for a specific use case. Fine-tuning adapts the model to the organization's data. This requires a much smaller sample of labeled training data than would be required to train a new model from scratch and typically converges faster because the model already has general knowledge.

Train custom models. When no suitable pretrained model exists, or the organization's data and task are unique enough to warrant it, the arcgis.learn module and ArcGIS Pro support end-to-end deep learning workflows. This path requires more data, more compute (typically GPU), and more expertise — but it provides full control over the model architecture, training process, and resulting behavior.

Geospatial foundation models are a newer direction, currently in research. These are models trained on massive geospatial datasets with billions of parameters, designed to be adapted to many downstream tasks. Esri is developing two types of foundation models. The first is a geodemographic foundation model that combines demographics with location data for similarity search, clustering, and regression. The second is a geospatial vision language model that connects satellite imagery with LLMs for natural-language imagery queries. As these models mature, they may shift the default starting point for many workflows from task-specific pretrained models to foundation model adaptation.

Location embeddings

Location embeddings are a newer class of AI output in ArcGIS. They are machine-understandable numerical representations of geography that distill complex location data into vectors that AI models can work with directly. Where traditional geospatial data represents places as coordinates and attributes, embeddings represent places as positions in a high-dimensional space where proximity means similarity.

Location embeddings enable analytical patterns that are difficult or impractical with traditional spatial methods. These include rapid similarity search across large geographies (find locations that resemble a target profile), pattern recognition and change detection (identify where things have shifted over time), clustering without predefined boundaries, and spatial regression where location context is a first-class feature.

Esri’s geodemographic location embeddings combine demographic data with other location characteristics into dense vector representations. The USA geodemographic location embeddings dataset is available through Living Atlas and can be used in similarity search tools provided through ArcGIS Pro. These are early capabilities, and the tooling and available datasets will expand; however, the architectural pattern is important to consider early. Embeddings sit between raw data and model inference as a reusable intermediate representation. Organizations that produce and maintain embeddings for their geographies gain a building block that can feed many downstream workflows without re-processing the source data each time.

Lifecycle and architectural decisions

The lifecycle for AI tools and models has architectural decisions at every stage.

Selection. It is critical to choose the right starting point: use pretrained models, fine-tune existing models, or train new custom models. The decision depends on how well existing models match the task, how much labeled data the organization has, and how much compute and expertise it can invest. For many common geospatial tasks, such as building footprint extraction, road detection, and land cover classification, pretrained models in ArcGIS Living Atlas are the right starting point.

Training and evaluation. When training is required, architects need to plan for GPU compute (cloud or on-premises), storage for training datasets and checkpoints, and an evaluation framework that measures model performance against the business problem, not just abstract accuracy metrics.

Geographic representativeness of training data matters, as a model evaluated only on the areas it trained on may underperform elsewhere.

Deployment. Trained models can be deployed as geoprocessing tools for interactive use, as services for application integration, or as batch processes for large-area inference. The deployment pattern shapes compute requirements, latency expectations, and how results flow into downstream systems. Managed services through ArcGIS Online are the lowest-infrastructure path, whereas self-hosted deployment in ArcGIS Enterprise offers more control over compute and data residency.

Monitoring and maintenance. Models degrade over time as the world they represent changes. For example, land cover shifts, new building types appear, and sensor characteristics drift. Architects should plan for periodic evaluation against ground-truth data, retraining when performance drops below acceptable thresholds, and versioning so that prior model states can be recovered or compared. This is the domain of MLOps, operational practices for managing models in production.

Note:

AI tools and models work across all ArcGIS deployment patterns — SaaS, cloud, on-premises, and disconnected. They do not depend on the LLM gateway that AI assistants and agentic AI features require. For organizations in regulated, sovereign, or air-gapped environments, this is the AI capability most readily available today. See Integration and interoperability for more on deployment considerations.

AI assistants

AI assistants are natural-language interfaces inside ArcGIS applications that help users complete tasks or write code. They are built on LLMs and geospatial agents, grounded on ArcGIS content, and scoped to the user's permissions. This chapter covers the two types of assistants (task and coding), how they work with the user's data, and related architectural considerations.

Task assistants

Task assistants help users complete GIS tasks more easily and quickly. They operate within ArcGIS applications, guiding users through actions, answering questions, and triggering operations on the user's behalf.

- The **ArcGIS Pro assistant** is the most feature-rich task assistant. It can explore ArcGIS Pro documentation, generate Query Layer SQL expressions, generate openCypher queries for knowledge graphs, and trigger common ArcGIS Pro actions through ArcPy and Arcade code generation. It is broad in scope, covering data management, analysis, mapping, and administration tasks within Pro. It is also extensible through the ArcGIS Pro SDK (.NET), allowing organizations to register custom assistant extensions that add domain-specific capabilities.
- Other task assistants are integrated across ArcGIS applications. The **Business Analyst assistant** guides users through site selection, market analysis, and demographic workflows. The **Survey123 assistant** helps survey authors design and configure forms. The **Item Details Assistant** helps content authors describe and document items, improving both discoverability and AI-readiness for downstream workflows.
- Additional task assistants are available or in development for ArcGIS StoryMaps, Hub, Instant Apps, Field Maps, Drone2Map, Solutions, ArcGIS for Microsoft Teams, and ArcGIS documentation. The inventory continues to expand as Esri integrates assistant capabilities across the ArcGIS platform.

Coding assistants

Coding assistants help users generate, understand, and troubleshoot code. They lower the barrier to scripting and automation for users who are less comfortable with code, and they help experienced developers move faster on repetitive tasks.

- The **ArcGIS Notebooks assistant** generates, fixes, and explains Python code within the ArcGIS Notebooks environment. It operates in two tiers. Standard mode handles administrative tasks and light data engineering, while Advanced mode supports more complex analytics and large-scale data workflows. Both modes are grounded on ArcGIS Python APIs, documentation, and coding patterns.
- The **Arcade assistant** helps users write Arcade expressions for pop-ups, visualizations, labeling, attribute rules, and other contexts where Arcade is used. Since Arcade is embedded across many ArcGIS applications, this assistant has broad applicability for configuration workflows that do not require full Python scripting.

Working with your data

AI assistants in ArcGIS are grounded on the content and context relevant to their scope. This grounding is what differentiates them from a generic chatbot; they respond using information from ArcGIS, not just their training data.

Each assistant is grounded differently depending on its context. The ArcGIS Pro Assistant draws on documentation, the application state, and the user's project data. App-specific task assistants draw on the content from within their application scope. Coding assistants draw on API documentation, coding patterns, and the current notebook or code context.

Perhaps most importantly, assistants work within the data and tools the user has access to. Existing identity and access controls in ArcGIS extend naturally to assistant interactions. An assistant cannot see content the user cannot see and cannot invoke operations the user is not authorized to perform. This means the existing security model governs what assistants can do without requiring a separate permission layer.

Data used with AI assistants remains private to the organization. Prompts and data are not used to train AI models without explicit authorization, and customers retain ownership of their prompts and data. When third-party AI services are involved (as they are today through the LLM gateway), Esri uses enterprise-class AI instances to segment and protect customer data from other customers. These principles are documented in detail in the ArcGIS Trust Center.

The effectiveness of assistants depends directly on the quality of the data they can access. Well-described items with complete metadata, clear field descriptions, and meaningful tags produce better assistant responses than poorly documented content. The AI-readiness practices described in geospatial data for AI apply directly here, and investing in data quality and documentation pays dividends when working with assistants.

Architectural considerations

AI assistants introduce several architectural considerations beyond those of traditional ArcGIS applications.

- **Progressive rollout through role privilege.** A dedicated role privilege for AI assistants lets administrators control which users have access. This enables organizations to take a staged approach: enable assistants for specific roles or groups, observe usage and behavior, then expand access as confidence grows. Assistants are not enabled by default; administrators or users must opt in.

- **User type licensing.** AI assistant capabilities vary by user type. Higher user type levels include more assistant capabilities. Architects designing solutions should account for which assistants and features are available at each licensing level, particularly when planning organization-wide rollouts.

- **LLM hosting and connectivity.** AI assistants today depend on LLMs hosted through ArcGIS Online, which serves as the LLM gateway. This means assistants require connectivity to ArcGIS Online to function, and are not currently available in fully disconnected or air-gapped environments. Self-hosted LLM support is on the roadmap for ArcGIS Enterprise, which will provide a path for customers who need to operate assistants in sovereign or disconnected environments. The scope of self-hosted LLM support is still being defined. See integration and interoperability for more detail on the LLM gateway model and deployment considerations.

- **Cost and consumption.** Some assistant interactions may consume credits in ArcGIS Online. Monitoring assistant usage is part of the broader observability story covered in architecture pillars. Learn more about credits in ArcGIS Online.

- **Extensibility.** The ArcGIS Pro SDK allows developers to write and register custom assistant extensions in .NET, adding domain-specific capabilities to the ArcGIS Pro assistant. This means organizations can extend the assistant’s scope beyond what Esri provides out of the box, for example adding custom tool invocations, domain-specific knowledge, or organization-specific workflows. Custom extensions inherit the same security model as the core assistant.

Agentic AI

Agentic AI introduces orchestration, automation, and integration through agents and agentic workflows. This is the newest of the three geospatial AI capabilities and the fastest moving. It is also the most architecturally significant, because it changes how ArcGIS connects to the broader enterprise AI ecosystem.

Geo-centric vs geo-enabled workflows

Agentic AI in ArcGIS follows two complementary architectural patterns, and most organizations will eventually use both.

- **Geo-centric** workflows bring agentic AI into GIS. This includes agentic mapping applications and geospatial agents built within ArcGIS. The GIS environment is the center of gravity, and the agents serve its workflows. A utility company deploying a geospatial agent to automate network tracing and validation is working in a geo-centric pattern.
- **Geo-enabled** workflows bring GIS into broader agentic ecosystems. Here an external agent platform orchestrates the work, and ArcGIS participates as a data, tool, and/or agent provider. The center of gravity is outside of the GIS, often a cloud or enterprise-level agentic platform, and ArcGIS contributes geospatial capabilities that the external system lacks natively. A logistics company whose supply-chain agent calls ArcGIS for routing and geocoding through MCP is working in a geo-enabled pattern.

These two patterns are not in competition. They both serve to make geospatial intelligence available wherever decisions are being made. The architectural axis is how agents are orchestrated: in a geo-centric pattern, ArcGIS concepts and users lead the workflow definition, whereas in a geo-enabled system, the users and capabilities of that other system drive the decisions. Both patterns need identity propagation, access control, observability, and governance, but the specifics differ depending on which side of the boundary the design sits. The integration and interoperability chapter covers the architectural details.

Integrating with agentic AI systems

Most AI systems and agent platforms have no native geospatial capabilities. They can call APIs, read databases, and generate text, but they cannot geocode an address, calculate a service area, understand topological relationships, or produce a map without external help. Additionally, many organizations have rich, authoritative geospatial data in their GIS they want to expose to agents. ArcGIS fills this gap by providing geospatial intelligence to external agents.

One way that ArcGIS exposes its capabilities to external agents is through MCP (Model Context Protocol). MCP is an open interoperability standard that defines how AI models and agents connect to external data sources and tools. By supporting MCP natively, ArcGIS makes it possible for any MCP-compliant agent to call geospatial operations, such as geocoding, routing, spatial queries, geoprocessing, content search, map export, geoenrichment, elevation, places, and more, without the agent platform needing to understand GIS internally. Beyond MCP services, the REST APIs and methods exposed by ArcGIS can also be used by other agentic systems, either as a direct user prompt or as part of a well-defined skill.

MCP support is rolling out across ArcGIS Enterprise, ArcGIS Online, and ArcGIS Location Platform. Each deployment target has its own rollout cadence and initially exposes a different set of capabilities. The set of supported operations will expand over time. For organizations that need additional or domain-specific agentic integration for their specific needs, Esri Professional Services and partners can deliver integration through APIs and custom solutions.

The architectural depth of MCP integration, including what is exposed, how authentication works, how to govern agent traffic, is covered in integration and interoperability.

Agentic mapping applications

Agentic mapping applications use AI agents to expand the experience of mapping through natural-language interaction. Rather than requiring users to know which tools to run or how to configure a query, an agentic mapping application lets them describe what they need in plain language and the agent figures out how to deliver it.

Esri provides two paths for building these applications.

- **Configurable templates** offer a low-code path. The Data Explorer template for ArcGIS Instant Apps lets organizations configure a web map with an agentic chat interface. Users can ask questions about the data, filter and query features, and explore spatial patterns through conversation rather than through traditional GIS tool interfaces. The Data Explorer template is in beta.
- **Developer components** offer a full-code path. AI components in the ArcGIS Maps SDK for JavaScript give developers the building blocks for custom agentic mapping applications: chat interfaces, out-of-the-box agents for navigation and data exploration, and the ability to build

custom agents with domain-specific logic. These components are in beta. To learn more, see an Introduction to building agentic mapping applications with ArcGIS Maps SDK for JavaScript.

Both paths follow the same architectural pattern, they are web applications that embed an agent and ground it on the organization's spatial content. They inherit the same security, identity, and data access model as other ArcGIS web applications.

One note on building practices. AI-assisted code generation can accelerate prototyping of agentic mapping applications significantly. And many of the ArcGIS Maps SDKs can be used for AI-assisted software development. Keep in mind that production applications require the same engineering discipline as any production system, including code review, testing, security audit, performance validation, and observability. AI code generation can be a development accelerator, but is not a substitute for production architecture and engineering.

Geospatial agents

Geospatial agents are agents designed specifically for geospatial tasks. They use the same components as any agent (LLM, instructions, tools, knowledge, memory), but their tools are GIS tools, their knowledge is geospatial content, and their memory holds spatial context. They incorporate geospatial reasoning and skills to accomplish multi-step work and advanced workflows that previously required manual orchestration.

Esri is building geospatial agents through two paths.

- **Prebuilt agents** are provided by Esri and are ready-to-use for common geospatial workflows, such as navigation assistance, data exploration, and spatial Q&A. Prebuilt agents are currently in beta and exposed through ArcGIS Maps SDK for JavaScript. They give organizations a starting point for agentic experiences without requiring custom agent development.

- **Custom agents through Agent Builder.** ArcGIS Agent Builder is a forthcoming visual tool for designing geospatial agents without coding. It will allow organizations to configure custom agents by specifying instructions, connecting tools, pointing to knowledge sources, and defining boundaries — producing agents that can be deployed in ArcGIS applications or exposed to external systems. Agent Builder is currently in an alpha release stage and broader access is planned for the near future.

Both paths are early in their definition and best practice identification. Organizations planning for geospatial agents should understand the direction and begin thinking about use cases, data readiness, and governance. It is recommended to plan deployments around the capabilities intended for beta or GA, not those that are still in alpha.

- **Use beyond ArcGIS.** Geospatial agents built in ArcGIS are designed to work inside ArcGIS applications. But Esri is also planning to expose them to third-party agentic systems through the Agent2Agent (A2A) protocol, which would allow a geospatial agent to participate in multi-agent workflows orchestrated by external agentic AI platforms. This positions geospatial agents as both a geo-centric capability (used inside ArcGIS) and potentially a geo-enabled one (consumable from outside). A2A support and scope will be confirmed as the capability matures.

Integration and interoperability

Geospatial AI in ArcGIS operates within a larger AI ecosystem that includes LLMs, AI models, external agentic platforms, and the architectural constraints under which organizations deploy them.

This chapter covers how ArcGIS integrates with that ecosystem in exposing geospatial capabilities to external AI systems through open standards, supporting interoperability of AI tools and models, integrating with the LLMs that power assistants and agents, and operating across the deployment topologies organizations require.

ArcGIS as MCP server

MCP (Model Context Protocol) is an open interoperability standard for Agentic AI that defines how AI models and agents connect to external data sources and tools. By natively supporting MCP as a server, ArcGIS makes its geospatial capabilities available to any MCP-compliant AI client without that client needing native geospatial expertise.

For example, an agent running on any MCP-compliant platform can call ArcGIS to geocode an address, calculate a route, query feature layers, run a geoprocessing tool, search for content, export a map, geoenrich a location, retrieve elevation data, or find nearby places. The agent does not need to know how GIS works internally, it calls a "tool" in the ArcGIS MCP server and gets a result.

MCP support is rolling out across the ArcGIS foundational products:

- **ArcGIS Online** - Esri's Software-as-a-Service (SaaS) deployment option
- **ArcGIS Enterprise** - Esri's server software, self-hosted deployment option. Runs on the cloud or on-premises, and supports Windows, Linux, and Kubernetes
- **ArcGIS Location Platform** - Esri's Platform-as-a-Service (PaaS) deployment option

Each of these exposes a different set of capabilities initially, with the set of available tools, resources, and prompts expanding over time. Organizations should evaluate which deployment target aligns with their architecture and monitor release notes for capability additions.

Additional MCP support is also planned for adjacent use cases. This includes MCP for ArcGIS Monitor to support system observability and operations, as well as MCP for ArcGIS developers to support application development through coding agents.

For organizations that need additional or domain-specific agentic integration, Esri Professional Services and partners can deliver integration through extensions of the MCP services in the foundational products, or through direct integration with ArcGIS REST APIs and custom solutions.

LLM gateway and BYO support

AI assistants and agentic features in ArcGIS depend on large language models (LLMs) for reasoning. Today, those LLMs are accessed through a gateway in ArcGIS Online rather than called directly by applications.

The **gateway pattern** means that AI clients (assistants, agents, applications) call an intermediary service that handles routing to LLM providers, authentication, rate limiting, policy enforcement, observability, and cost attribution. Organizations do not configure or operate this layer directly, instead Esri manages it. The benefits include operational simplicity and consistent behavior across all AI features. The tradeoffs include requiring connectivity to ArcGIS Online.

Self-hosted LLM support is on the roadmap for ArcGIS Enterprise. When available, this will allow organizations to operate their own LLMs for AI workloads. This may be motivated by data residency, sovereignty, latency, cost, or other considerations. The scope of self-hosted LLM support is still being defined; however, it is expected to cover a defined set of models and capabilities, not arbitrary LLM use. Architects with self-hosted requirements should plan to evaluate the BYO model when its scope is published, and account for the operational burden of hosting, securing, and maintaining LLM infrastructure.

Note:

For self-hosted LLM support, Esri is planning to support a small, defined list of LLMs. Additionally, certain ArcGIS AI Assistants may only support the LLM gateway pattern.

The cross-cutting concerns that LLM gateways address, including cost attribution, policy enforcement, and observability, apply regardless of whether the gateway is Esri-managed or self-hosted. Learn more in the architecture pillars chapter.

AI tools and models interoperability

Organizations may require that their AI tools and models work across environments, including models from ArcGIS used outside of ArcGIS, and models from outside of ArcGIS used in ArcGIS.

Models from ArcGIS. Pretrained models in Living Atlas are published as Deep Learning Packages (DLPKs). Each includes documented inputs, outputs, architecture, training data, and limitations. DLPKs are designed for ArcGIS but are built on standard frameworks (PyTorch), supporting use in other AI environments where appropriate. Models trained using the arcgis.learn module produce standard PyTorch artifacts. Models can be used for inference by calling ArcGIS tools such as an ArcGIS Notebook or a Raster Analytics workflow, which can work from DLPKs, scale with virtual GPU capacity, and publish results back to the organization's ArcGIS Enterprise or ArcGIS Online system.

Models into ArcGIS. Deep Learning Studio provides a visual interface for bringing externally-trained models into ArcGIS. Organizations that train models in other environments (e.g., cloud ML platforms, research frameworks) can bring them into ArcGIS for inference on geospatial data, provided the model architecture is compatible.

Location embeddings and foundation models are positioned as building blocks that can feed workflows both inside and outside ArcGIS. Location embeddings are standard vectors usable in any ML pipeline. Geospatial foundation models will be adaptable to downstream tasks in multiple environments. Specific interoperability patterns are still emerging given the early maturity of these capabilities.

Hybrid, disconnected, and sovereign systems

Many organizations operate in environments where AI workloads must function without persistent cloud connectivity. Geospatial AI operates in these environments as follows:

AI tools and models work across all deployment patterns. Pretrained models, organization-trained models, geoprocessing tools with embedded AI, and the arcgis.learn module all operate in cloud, hybrid, on-premises, and disconnected environments. They do not depend on LLMs and the LLM gateway. For organizations in constrained environments, AI tools and models are the geospatial AI capability most readily available today.

AI assistants and agentic features require connectivity today. These capabilities depend on LLMs accessed through the ArcGIS Online gateway; therefore AI assistants and Esri-hosted agentic capabilities are not currently available in fully disconnected, air-gapped, or sovereign deployments. Organizations needing agentic or assistant capabilities in these environments today can work with Esri Professional Services and partners on custom solutions.

Self-hosted LLM support is the path forward. The BYO LLM direction for ArcGIS Enterprise is the primary architectural path to supporting AI assistants and agentic features in environments that cannot connect to ArcGIS Online. Once available, customers will be able to deploy these capabilities within their own infrastructure boundaries. Timing and scope are still being defined.

Architects designing for constrained environments should build their near-term AI strategy around the capabilities that are available today (AI tools and models), plan for the capabilities that are coming (self-hosted LLM for assistants and agents), and engage Esri Professional Services for custom integration where the gap between current availability and organizational need is urgent.

Architecture pillars

The ArcGIS Well-Architected Framework is organized around six pillars: reliability, performance and scalability, security, integration, automation, and observability. Each has established guidance for traditional ArcGIS systems. AI workloads build on existing guidance by introducing new concerns, new failure modes, and new design decisions that architects need to account for.

Learn more about the architecture pillars for ArcGIS systems.

Security and trust

AI workloads expand the security surface in ways that traditional ArcGIS systems do not.

The ArcGIS Trust Center defines six principles for trusted AI in ArcGIS: security, privacy, transparency, fairness, reliability, and accountability. AI transparency cards document the functionality, validation, and safeguards of individual AI features. Architects designing AI workloads on ArcGIS should treat these principles as design constraints, not aspirational statements.

- **Identity for AI workloads.** AI assistants generally inherit the user's identity and permissions, and work with their current interface and experience of using ArcGIS. Agent identities are more complex — an agent may act as the user (inheriting their scope), act as itself (with its own identity and bounded permissions), or act on behalf of a user with delegated scope. Agents also often work beyond a single interface or application, accessing remote systems, multiple data sources, and even interacting with external agents. Architects should plan for appropriate and carefully scoped identity propagation through MCP-connected agents and across A2A interactions, particularly when agents cross organizational boundaries.

- **AI-specific attack surface.** AI workloads introduce vulnerability classes that traditional systems do not face, including prompt injection (malicious input manipulating AI behavior), data exfiltration through model outputs, and tool misuse (agents invoking operations beyond their intended scope). Guardrails include input validation, output filtering, tool-call constraints, use of a governed LLM gateway, and human-in-the-loop checks for high-stakes actions. These are active concerns in any system that exposes LLM-powered interfaces to users or external inputs and should be strongly considered when designing AI workloads with ArcGIS.

- **Data handling across trust boundaries.** When AI workloads use third-party LLMs through the gateway, organizational data crosses trust boundaries. Esri uses enterprise-class AI services to segment each organization's data, and prompts are not used for model training without explicit authorization. Architects designing solutions with sensitive data should understand where data flows, what protections are in place, and whether additional segmentation is warranted for their specific data classification. See the Trust Center for details on Esri's data handling practices.

Performance and scalability

AI workloads have cost and compute profiles that differ from traditional GIS.

- **Consumption-based cost.** AI capabilities may carry per-token (LLMs), per-inference (models), or per-call (services) pricing through consumption-based models that differ from the per-user or per-server cost patterns.

- **GPU and compute planning.** Training deep learning models and running high-volume inference both benefit from GPU compute. Architects should plan capacity, including cloud GPU instances, on-premises GPU hardware, or hybrid approaches, based on training cadence, inference volume, and latency requirements. For organizations using managed AI services through ArcGIS Online, Esri handles the compute. Learn more about ArcGIS Notebooks, which provide a common environment for training models with efficient use of GPU resources.

- **Scaling patterns.** Common patterns for managing AI cost and latency include batching (grouping inference requests), caching (reusing results for repeated queries or common embeddings), and tiered inference (using a smaller, cheaper model for routine requests and escalating to larger models only when needed). These patterns apply to both model inference and LLM interactions.

Reliability

AI workloads, particularly those involving LLMs and agents, introduce non-determinism that changes how reliability is defined.

- **Non-determinism as the baseline.** Traditional systems are expected to produce the same output for the same input. AI systems, especially generative ones, may not. Reliability for AI workloads means consistent behavior within acceptable bounds rather than deterministic, identical outputs. This shifts how architects think about testing (outcome-based rather than output-based), error handling (graceful degradation rather than binary success/failure), and user expectations (probabilistic rather than deterministic).

- **LLM availability and fallback.** AI assistants and agentic workflows depend on LLM availability through the gateway. Architects should design for graceful degradation when LLMs are unreachable, such as deterministic fallback behaviors, clear user messaging, and bounded retry logic.
- **Model degradation over time.** AI models can lose accuracy as the world they represent changes, such as land cover shifts, new built infrastructure, and sensor characteristics drift. Architects should plan for periodic evaluation against ground truth data, retraining when performance drops below thresholds, appropriate messaging to users to explain the limitations of a specific model, and intentional and semantic versioning to enable rollback or access to previous capabilities as new models are released. This should be considered an operational discipline as opposed to a one-time design decision.

Automation

AI introduces new categories of automation alongside traditional deterministic workflows.

- **Agentic automation.** Agents introduce non-deterministic automation, reasoning about goals, choosing actions within boundaries, and producing outcomes that may vary based on context. This is architecturally distinct from deterministic automation (scripts, scheduled tasks, workflow rules) where outcomes are fixed and repeatable. Architects should consciously decide which category a workflow belongs to, as not everything necessarily benefits from agentic orchestration.
- **Lifecycle automation (MLOps and GenAIOps).** Models, agents, prompts, and AI-powered services have their own lifecycles. Automation patterns include model deployment pipelines, evaluation automation, drift monitoring, retraining triggers, and prompt versioning. These practices, sometimes called MLOps for model workloads or GenAIOps for generative and agentic workloads, are the AI-specific complement to the traditional automation pillar.
- **Evaluation as infrastructure.** Evaluation of AI outputs, including model accuracy, agent task completion rates, and response quality, is best treated as automated infrastructure that runs continuously as opposed to ad-hoc testing performed occasionally. Continuous evaluation supports both reliability and trust, and it creates the feedback loop that enables informed decisions about retraining, prompt changes, or capability expansion.

Observability

AI workloads require visibility into signals that traditional telemetry does not capture.

- **New observability targets.** Beyond system metrics (CPU, memory, latency, error rates), AI workloads generate signals that architects should plan to capture. These include prompt and response content, agent reasoning traces, model inference outputs, tool-call sequences, and cost telemetry (token usage, credit consumption, GPU hours).
- **Privacy in AI logging.** Prompts and responses may contain sensitive content, such as user queries with personal information, attribute data from protected datasets, or organizational information. Logging strategies in agentic mapping applications and geospatial agents should balance visibility with privacy. See the Trust Center for details on Esri's observability in AI practices.
- **Decision logging for audit.** Agentic workflows typically make autonomous decisions, and often also involve human-in-the-loop steps. Both the agent and human actions should be logged for auditability, including what was decided, when, by whom (human or agent), what reasoning was available, and what action followed. This serves as the foundation for compliance, post-incident analysis, and building organizational trust in AI systems over time.

Integration

Integration is the connective tissue for AI workloads in ArcGIS. The integration and interoperability chapter covers the architectural mechanics, including MCP, the LLM gateway, model interoperability, and deployment topology. At the pillar level, three concerns are worth surfacing:

- **API management for AI traffic.** Agent-to-ArcGIS calls through MCP, LLM gateway traffic, and model inference requests all create API traffic patterns that architects should plan for. Rate limiting, authentication, quota management, and traffic observability apply to AI integration surfaces the same way they apply to traditional service integrations, though the traffic patterns may differ (bursty, token-heavy, or long-running).
- **Identity flow across AI boundaries.** When an external agent calls ArcGIS through MCP, whose identity does the call carry? When a geospatial agent calls an external service, what credentials does it present? Identity propagation across AI integration boundaries is an active design decision that architects must resolve based on their security model, not leave to defaults. See the security and trust pillar for more information.

- **Data flow boundaries.** AI integrations create data flows that may cross organizational, cloud-provider, or jurisdictional boundaries. Architects should map where data travels in their AI architecture, particularly for prompts containing organizational data, inference results derived from protected sources, and agent interactions that span multiple systems.

Next steps and resources

Organizations approach geospatial AI from different starting points, with different goals and different levels of readiness. The path forward depends on an organization's goals and what it's prepared for today. The following are some suggestions for how to get started.

Getting started

Match your starting point to your goal. The three geospatial AI capabilities align with different organizational objectives:

- To improve **user productivity** and make GIS more accessible across the organization, start with AI assistants. Enable them for targeted user groups through role privileges, observe adoption, and expand.
- To advance **analytical work**, including automated extraction, classification, prediction, or pattern detection at scale, start with AI tools and models. Use pretrained models from Living Atlas for common tasks, then fine-tune or train custom models where needed.
- To **automate multi-step workflows** or integrate ArcGIS with broader enterprise AI systems, start with Agentic AI. Expose ArcGIS capabilities through MCP for existing agent platforms, or explore agentic mapping applications for natural-language access to spatial data.

These are not mutually exclusive. Many organizations will pursue all three, but starting with a focused goal produces faster results and clearer architectural patterns than trying to adopt everything at once.

Match your starting point to your readiness. AI readiness has several dimensions, including data and organizational readiness.

On data readiness, AI workloads demand accurate, well-documented, governed data. If item details are sparse, metadata is inconsistent, or systems of record are poorly maintained, the highest-leverage first investment is making organizational data more AI-ready. The practices in Geospatial data for AI apply here.

On organizational readiness, AI adoption raises governance questions that go beyond technology. Who is authorized to deploy agents? What data can be used for training? How are AI outputs validated before they inform decisions? Establishing principles and governance may be more valuable than deploying capabilities if organizational clarity on these questions and other governance or compliance-related topics is lacking. See the ArcGIS Trust Center for more guidance.

Start with some quick wins. The following are commonly pursued, and typically low effort, quick wins to consider getting started with.

- Turn on AI assistants for a pilot group using the role privilege for progressive rollout
- Apply pretrained models from Living Atlas to a common analytical task (building footprint extraction, land cover classification)
- Expose ArcGIS capabilities through MCP to an existing agentic AI platform for teams already invested in agent-based workflows
- Use the Item Details assistant to improve metadata quality across a high-value content collection

Design for production from the start. Even pilot deployments benefit from the discipline described in architecture pillars. These include, but are not limited to, human-in-the-loop checkpoints for high-stakes decisions, evaluation frameworks for AI outputs, observability for AI-specific signals, and alignment with the ArcGIS Trust Center principles.

Where to get help. For complex integrations, specialized expertise, or accelerating time-to-value, Esri Professional Services and partners deliver tailored solutions covering implementation, integration, and architectural consulting. Connect with Esri Professional Services.

Key resources

To learn more about AI in ArcGIS and the broader architectural context this lens sits within, consider the following resources.

- ArcGIS Trust Center — Trusted AI — Esri's AI principles, commitments, safeguards, and AI transparency cards
- ArcGIS Architecture Center — The six-pillar framework, system patterns, and Architecture Library
- ArcGIS Living Atlas of the World — Authoritative geospatial content and pretrained AI models

- Esri Geospatial AI resources — Public blogs, use cases, and technical content on AI capabilities in ArcGIS
- ArcGIS product documentation — Feature-level guidance on products and technologies referenced in this lens
- Esri Professional Services — Implementation, integration, and architectural consulting