AnyLog

The AnyLog Edge Data Fabric

Executive Overview

The convergence of edge computing and Artificial Intelligence is transforming the way industrial systems are designed and utilized. Organizations are increasingly expected to analyze operational data where it is generated, support real-time decision making, and enable autonomous operations across geographically distributed environments.

The **AnyLog Edge Data Fabric (EDF)** addresses this challenge by providing a distributed data platform that leaves operational data at the edge while presenting applications, administrators, and AI with a single logical environment. Instead of centralizing operational data, the platform virtualizes distributed resources through a **Virtual Data Lake**, one or more **Unified Namespaces**, and a **Single System Image**, all coordinated by a **Distributed Metadata Layer**.

This architecture enables organizations to:

- Keep operational data under local ownership.
- Query distributed data as a single logical resource.
- Integrate AI through a unified semantic interface.
- Manage distributed infrastructure as a cloud-like platform.
- Scale horizontally by adding autonomous nodes rather than centralized infrastructure.

This guide explains the architectural principles, core components, and distributed services that together form the AnyLog Edge Data Fabric. Rather than describing product features, it presents the architectural decisions that enable distributed data, applications, and AI to operate as a unified platform while preserving the performance, resilience, and autonomy of edge computing.

AnyLog

Table of contents

Chapter 1: Introduction

Building a Distributed Data Platform for Edge AI and Autonomous Operations

Industrial computing is undergoing one of its most significant architectural transitions since the introduction of enterprise databases and cloud computing. Manufacturing plants, utilities, transportation systems, healthcare facilities, telecommunications networks, and smart infrastructure are generating unprecedented volumes of operational data at the edge of the network. At the same time, Artificial Intelligence is moving closer to where this data is produced, enabling real-time analytics, predictive maintenance, autonomous operations, and intelligent decision making directly within operational environments.

For more than three decades, the dominant approach to industrial data management has been to collect operational information into centralized repositories before it can be queried, analyzed, or consumed by applications. Historians, enterprise databases, data warehouses, cloud data lakes, and more recently cloud-native analytics platforms all follow the same architectural principle: **move the data first, then process it**.

This model has served the industry well, particularly when communication networks were reliable, cloud infrastructure was inexpensive, and most analytical workloads operated at the enterprise level. Today, however, industrial environments are becoming increasingly distributed. A single organization may operate hundreds or thousands of factories, manufacturing cells, substations, retail stores, wind farms, offshore platforms, hospitals, or autonomous vehicles. Continuously transferring raw operational data from every location to centralized infrastructure introduces significant costs, consumes valuable network bandwidth, increases latency, and often conflicts with regulatory, privacy, and operational requirements.

Artificial Intelligence further amplifies these challenges. Modern AI systems require access not only to historical data but also to real-time operational information, asset relationships, metadata, and contextual knowledge about the environment in which they operate. Building custom integrations between every AI application and every industrial system quickly becomes impractical. As organizations deploy multiple AI assistants, copilots, and autonomous agents, the complexity of maintaining these integrations grows exponentially.

The industry therefore faces a new architectural challenge. Rather than building increasingly complex infrastructure to centralize operational data, organizations are looking for a way to make distributed data, assets, and services behave as a single logical platform while preserving the performance, resilience, and autonomy of edge computing.

As applications and Artificial Intelligence increasingly need to analyze data and make decisions where operational events occur, continuously moving data to centralized infrastructure becomes

a constraint rather than an advantage. The challenge is no longer how to centralize more data, but how to virtualize distributed resources into a unified operational platform.

This is the problem that the AnyLog Edge Data Fabric (EDF) was designed to solve.

AnyLog Edge Data Fabric (EDF) is a distributed data platform that enables organizations to manage operational data where it is created while presenting applications, administrators, and AI systems with a single logical environment. Instead of treating edge computing as a collection of isolated systems, AnyLog transforms compute nodes into an autonomous yet coordinated platform that virtualizes distributed resources through a single logical view without centralizing the data, as illustrated in Figure 1.

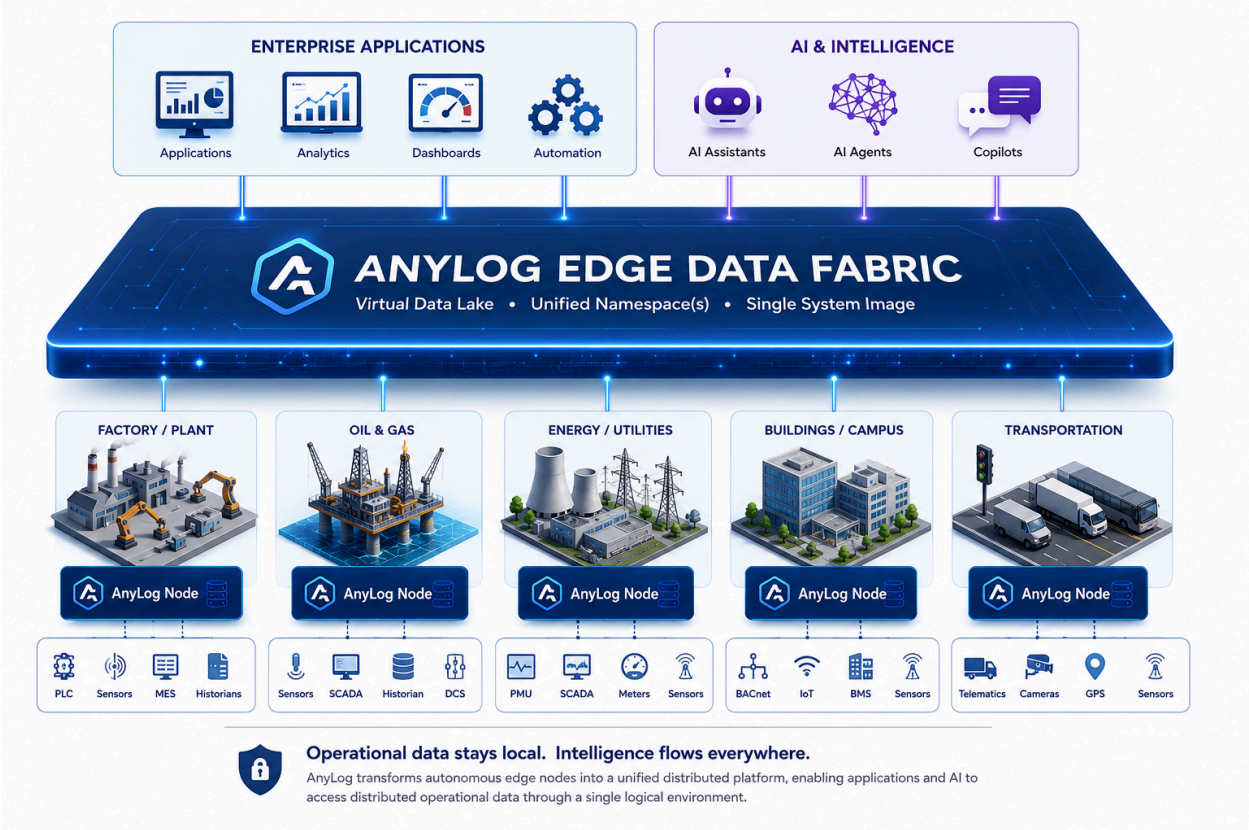

Figure 1. *The AnyLog Edge Data Fabric transforms autonomous edge nodes into a unified distributed platform. Operational data remains within local databases at each site, while applications and Artificial Intelligence interact with a single logical platform that virtualizes distributed data, industrial assets, and infrastructure. This architecture provides a cloud-like operational experience while preserving the performance, resilience, autonomy, and ownership of edge computing.*

The architecture is built upon three simple but powerful principles:

- **Keep operational data where it is generated.**
- **Move processing to the data.**
- **Share metadata instead of operational data.**

Rather than centralizing operational data, the AnyLog Edge Data Fabric distributes the knowledge and processing required to locate, understand, and operate on data wherever it resides. Each node stores and manages its own operational data, while the Distributed Metadata Layer enables participating nodes to discover resources, determine where data resides, coordinate distributed queries, enforce security policies, and cooperate as a single logical platform.

The result is a cloud-like operating model. Applications access a Virtual Data Lake, AI interacts through one or more Unified Namespaces, and administrators manage a Single System Image. Together, these virtualization layers provide a unified view of distributed data, assets, and infrastructure while operational data remains in-place.

This guide explains the architectural concepts that make this possible. Rather than describing product features, it presents the design principles behind the platform and the reasoning that led to its architecture. It explores how distributed metadata replaces centralized catalogs, how SQL executes across distributed nodes, how AI uses a simple prompt to discover, query, and reason across distributed data through AnyLog's Model Context Protocol (MCP) server, and how management, security, automation, and high availability become native capabilities of a distributed platform.

Although the examples throughout this guide focus on industrial and operational environments, the architectural principles are broadly applicable to any system in which data is generated across many independent locations and must be accessed, analyzed, and acted upon without sacrificing performance, ownership, or resilience.

As edge computing, Artificial Intelligence, and autonomous systems continue to be adopted, distributed data platforms will become a foundational component of modern digital infrastructure. This guide explains how AnyLog addresses that challenge and provides the architectural framework for understanding how a distributed platform can deliver the operational simplicity of the cloud while preserving the autonomy and performance of the edge.

Key Architectural Insight

The defining architectural decision in AnyLog is not the use of edge computing, distributed SQL, or Artificial Intelligence. It is the decision to **leave operational data where it is generated and virtualize everything around it**. Every capability described in this guide—from distributed queries and Unified Namespaces to AI integration and cloud-like management—builds upon that single architectural principle.

Chapter 2: Architectural Principles

Designing Edge Infrastructure for a Distributed World

Every software platform is shaped by a set of architectural assumptions. Traditional enterprise systems were designed for environments in which applications, databases, and users operated within relatively centralized infrastructure. As cloud computing emerged, these assumptions evolved to emphasize elastic compute, centralized storage, and globally accessible services. Industrial computing, however, presents a fundamentally different set of requirements.

In industrial computing, operational data is generated continuously across geographically distributed facilities, production lines, substations, vehicles, and intelligent devices. Much of this information is required immediately for monitoring, automation, and local decision making. Connectivity between locations may be limited, intermittent, or expensive, and organizations are often unwilling—or unable—to continuously transfer operational data outside their own environments due to regulatory, security, or business considerations.

These realities require a different architectural model.

The AnyLog platform was designed around a small number of principles that collectively define how a distributed data platform should operate. The principles are not independent features; each builds upon the previous one, creating an architecture in which distributed systems cooperate as a single logical platform while preserving the autonomy of every participating node.

Keep Operational Data Local

Operational data has the greatest value where it is generated. Industrial equipment, production lines, substations, vehicles, and other edge systems depend on immediate access to operational information for monitoring, automation, and real-time decision making. Continuously transferring data to centralized infrastructure before it can be analyzed introduces latency between when an event occurs and when the organization can respond. During that time, equipment conditions may change, production may be affected, safety risks may increase, and localized issues may propagate across interconnected systems. The closer processing is to the point where data is generated, the faster and more effectively these operational events can be detected and acted upon.

The increasing adoption of Artificial Intelligence and autonomous operations further reinforces this architectural requirement. Organizations are moving toward **lights-out factories**, autonomous production systems, intelligent utilities, and self-managing industrial infrastructure, where AI continuously monitors operations, predicts failures, optimizes processes, and increasingly makes decisions without human intervention. These workloads require immediate access to operational data and cannot depend on continuously transferring information to

centralized infrastructure before analysis or action can occur. Intelligence cannot move to the edge while the data that drives the decisions is routed elsewhere for processing.

For these reasons, the AnyLog architecture is built on a simple principle: **operational data should remain where it is generated.**

Instead of continuously replicating operational information into centralized databases or cloud platforms, AnyLog stores data within local databases, typically close to the equipment that produced it. Every AnyLog node becomes responsible for managing its own operational information while remaining fully capable of participating in the distributed platform.

Keeping operational data local provides several architectural advantages. It minimizes unnecessary network traffic, preserves ownership of operational information, enables deterministic low-latency processing, and allows facilities to continue operating independently during communication outages. It also creates the foundation for distributed AI, allowing intelligent applications and autonomous systems to analyze operational data, invoke services, and make decisions locally while collaborating with other nodes across the platform when required. Rather than treating local storage as a temporary staging area before data is sent elsewhere, AnyLog treats it as the permanent home of operational information.

This architectural decision shapes every layer of the platform. Because operational data remains distributed, the AnyLog Edge Data Fabric brings processing to the data, shares metadata rather than operational data, coordinates autonomous nodes, and virtualizes access to distributed resources. The following sections explain how these architectural principles work together to create a unified distributed platform for applications, administrators, and Artificial Intelligence.

Bring Processing to the Data

If operational data remains distributed, processing must also become distributed.

Instead of transporting large datasets to centralized applications, AnyLog distributes computation to the nodes where the required information already resides. SQL queries, aggregations, analytics, automation, and AI inference execute locally, using the processing resources available at each node.

Only the information required to satisfy a request traverses the network. Query requests, metadata, and result sets are exchanged between nodes, while the underlying operational data remains in place.

This approach produces substantial benefits. Network utilization decreases because raw operational data is not continuously replicated. Applications receive access to the most current information rather than waiting for synchronization with centralized infrastructure. As additional

nodes are deployed, processing capacity grows naturally because each node contributes its compute resources to the Edge Data Fabric.

The architecture therefore scales by adding resources rather than expanding centralized infrastructure.

Share Metadata Instead of Operational Data

Distributed systems require a common understanding of the resources available throughout the platform. Applications and AI must discover datasets, determine which nodes host the required data, locate available services, understand schemas, identify industrial assets, and enforce security policies without requiring administrators to manually configure every participating node.

AnyLog solves this challenge by separating **metadata** from **operational data**.

Operational data remains within the local databases managed by each AnyLog node. Instead of sharing the raw data, participating nodes continuously publish and consume metadata that describes the distributed environment. This metadata forms the platform's shared knowledge base, allowing every node to understand the resources available across the Edge Data Fabric.

One of the most important functions of the Distributed Metadata Layer is **data location**. As datasets are created, updated, or removed, nodes publish metadata identifying where those datasets reside. When a distributed query is executed, the query coordinator consults the metadata layer to determine exactly which nodes host the required tables before distributing the workload. Applications therefore never need to know where operational data is physically stored—the platform resolves that information dynamically.

Beyond data location, the metadata layer maintains the information required to operate the platform. This includes schema definitions, node registrations, service descriptions, security policies, Unified Namespace definitions, AI tools, aggregation and replication policies, retention policies, MCP services, and other information required for distributed coordination.

Metadata is dynamic not static. Nodes continuously update their own metadata as resources become available or operational conditions change, while simultaneously querying metadata published by other nodes to discover new datasets, services, and capabilities. AnyLog provides a simple native API for publishing, querying, and managing metadata, allowing both platform components and external applications to leverage the Distributed Metadata Layer as a common source of operational knowledge.

To ensure consistency across autonomous nodes, the metadata layer is implemented using either a **blockchain** or with **metadata managers** , depending on deployment requirements. This provides a distributed, tamper-resistant mechanism, while allowing organizations to select the deployment model that best fits their operational requirements.

The Distributed Metadata Layer serves as the **control plane** of the AnyLog Edge Data Fabric, while operational data forms the **data plane**. Because metadata is many orders of magnitude smaller than operational data, it can be synchronized efficiently across the platform, allowing every node to maintain a consistent understanding of where data resides, how it is structured, which services are available, and what policies govern access. This enables autonomous nodes to cooperate as a single logical platform without requiring operational data to be centralized.

Autonomous Nodes, Cooperative Platform

An AnyLog deployment is not built from specialized server types. Every participating node executes the same software platform and is capable of performing data ingestion, distributed query processing, metadata management, automation, AI integration, security enforcement, and management functions.

Each node is autonomous. It continues operating independently regardless of the availability of other nodes or cloud infrastructure. Local applications continue functioning, rules continue executing, and operational data remains accessible even during network disruptions.

At the same time, nodes cooperate through the Distributed Metadata Layer to present a unified platform to applications and administrators. New nodes automatically become discoverable, participate in distributed queries, contribute metadata, and share services according to configured policies.

This balance between local autonomy and distributed cooperation is one of the defining characteristics of the AnyLog architecture.

Configuration Rather Than Development

Distributed systems are traditionally associated with lengthy integration projects. New deployments often require custom connectors, data transformation software, schema mapping, application development, and infrastructure engineering before operational value can be realized.

AnyLog minimizes this effort through a configuration-driven architecture.

The software platform is modular, and capabilities such as protocol connectors, distributed SQL, the Unified Namespace, automation, AI integration, security, and high availability are enabled through configuration rather than application development. Nodes can be deployed on gateways, industrial computers, servers, virtual machines, or cloud infrastructure using Docker, Kubernetes, IBM Open Horizon, Dell NativeEdge, or conventional operating system installations.

As operational requirements evolve, organizations enable or modify platform services by updating configuration and policies rather than rewriting software.

This approach significantly reduces deployment time while allowing the same platform to support a broad range of industrial environments.

Horizontal Growth

Traditional enterprise platforms scale vertically by continuously expanding centralized infrastructure. Every new operational site contributes additional data and workload to the same shared resources, increasing costs and creating larger centralized bottlenecks. Industrial operations, however, grow horizontally by adding new facilities, production lines, substations, vehicles, and intelligent devices.

The AnyLog Edge Data Fabric is designed around this operational reality. Instead of scaling centralized infrastructure, it scales by adding autonomous nodes. Every new node contributes not only additional operational data, but also processors, memory, storage, metadata, platform services, and local operational intelligence. At the same time, operational data is distributed across a larger number of nodes, reducing the amount of data managed by each individual node while increasing the aggregate computing resources available to the platform.

Distributed queries, analytics, automation, and AI workloads execute concurrently across participating nodes, allowing processing capacity to grow naturally with the deployment. Rather than concentrating ever-larger workloads on centralized infrastructure, computation is distributed throughout the Edge Data Fabric and executed where the data resides.

The result is a fundamentally different scaling model. In centralized architectures, growth continually increases demand on a fixed pool of shared resources. In the AnyLog architecture, every new location also contributes computing resources to the platform. As deployments expand, both data and processing capacity grow together, allowing the Edge Data Fabric to scale from a handful of nodes to thousands of distributed locations without creating larger centralized bottlenecks.

Built on Open Standards

A distributed platform should not require organizations to replace existing investments.

AnyLog is built on widely adopted industry standards for data acquisition, communication, storage, and application integration. It collects operational data through protocols and southbound interfaces, such as OPC UA, MQTT, REST, Modbus, gRPC, and standard database connectors. Applications, analytics platforms, and AI systems interact with AnyLog through northbound interfaces, such as SQL, REST APIs, gRPC, and the Model Context Protocol (MCP).

This standards-based architecture allows AnyLog to integrate with existing industrial infrastructure and operate across different hardware, operating systems, and deployment

environments. Organizations can integrate AnyLog and other technologies incrementally without replacing the systems and applications already in place.

Architectural Perspective

Keeping operational data local makes it practical to preserve ownership and reduce infrastructure, egress, and data services costs. Bringing processing to the data enables distributed SQL, local AI inference, and scalable analytics. Sharing metadata instead of operational data allows autonomous nodes to cooperate as a unified platform. Configuration-driven deployment simplifies implementation, while horizontal scalability ensures that the platform becomes more capable as new nodes are added. Finally, the use of open standards allows the platform to integrate naturally with existing operational environments.

These principles are not isolated design choices. Together, they define an architecture in which distributed resources operate with the simplicity of a single system while retaining the resilience, performance, and autonomy required for edge computing.

Key Architectural Insight

The AnyLog architecture is built on a single foundational idea: **operational data should remain where it creates value**. Every other architectural principle—distributed processing, shared metadata, autonomous nodes, horizontal scalability, and unified management—exists to make that decision practical. Rather than optimizing the movement of data to centralized systems, AnyLog optimizes the movement of knowledge and computation across a distributed platform.

Chapter 3: The AnyLog Architecture

From Autonomous Nodes to a Unified Platform

Distributed systems are often perceived as collections of independent computers connected by a network. Each system stores its own data, runs its own applications, and is managed independently. While this model reflects the physical reality of edge computing, it places a significant burden on application developers and system administrators. Applications must track where data resides and route requests accordingly, administrators must manage hundreds or thousands of independent systems, and AI must integrate separately with each operational technology environment.

The objective of the AnyLog architecture is to remove this complexity.

Rather than exposing individual systems, AnyLog transforms autonomous edge nodes into a single logical platform. Each node continues operating independently and retains ownership of its operational data, yet applications, administrators, and AI systems interact with the deployment as though it were a unified environment.

This is achieved through virtualization.

Operational data remains distributed, but the platform virtualizes access to data, industrial assets, and infrastructure. As a result, physical distribution becomes largely transparent to software while organizations continue benefiting from the performance, resilience, and autonomy of edge computing.

The AnyLog Node

The fundamental building block of the platform is the **AnyLog Node**.

An AnyLog node is a complete software platform that can be deployed on industrial gateways, embedded computers, edge servers, virtual machines, cloud instances, or enterprise servers. Each node executes the same software regardless of where it is deployed. There are no specialized database servers, management servers, or AI servers. Instead, functionality is enabled through configuration, allowing each deployment to activate only the services required for its operational role.

Each node manages one or more local databases selected by the organization and provides a collection of distributed services that may include data ingestion, SQL processing, metadata management, automation, monitoring, scheduling, AI integration, aggregation, replication, and security enforcement.

Because every node is autonomous, it continues operating even when communication with other nodes is interrupted. Local applications continue accessing operational data, automation continues executing, and AI can continue performing local inference using the information available within the node.

This autonomous design is fundamental to supporting industrial environments where uninterrupted operation is essential.

Building a Distributed Platform

The AnyLog Edge Data Fabric is built from autonomous nodes that cooperate to form a single distributed platform. Every node runs the same software and can perform one or more roles based on its configuration, location, and available resources. Depending on the deployment, a node may ingest operational data, manage local databases, execute distributed queries, provide gateway services, perform AI inference, host replicated data, or monitor the health of the platform.

Each node continuously publishes its capabilities, supported protocols, available services, hosted datasets, and other metadata to the **Distributed Metadata Layer**. This shared knowledge allows the Edge Data Fabric to automatically discover resources, identify where data

resides, select the appropriate nodes to execute requests, and coordinate distributed processing.

Applications, administrators, and AI do not connect directly to individual nodes—they connect to the Edge Data Fabric. The platform automatically discovers the required resources, selects the participating nodes, coordinates distributed execution, and returns a unified response. As new nodes, datasets, and services are added, they become immediately available through the fabric without requiring applications to be modified or redeployed.

This approach fundamentally changes how distributed systems are integrated. Instead of building and maintaining point-to-point integrations with individual systems, applications integrate once with the Edge Data Fabric. The platform handles service discovery, data location, protocol selection, workload distribution, and result aggregation transparently.

Every node remains fully autonomous. It continues collecting, storing, processing, and serving its local operational data even when communication with other nodes is interrupted. When connectivity is available, nodes exchange metadata, coordinate distributed workloads, and contribute their resources to the platform. The result is a distributed system that behaves as a single logical platform while preserving the resilience, autonomy, and local ownership of every participating node.

The Distributed Metadata Layer

Cooperation between autonomous nodes depends on a common understanding of the distributed environment.

AnyLog achieves this through the **Distributed Metadata Layer**, which serves as the platform's shared knowledge base.

Rather than storing operational data, the metadata layer stores information that describes the platform topology. This includes schema definitions, node registrations, service capabilities, security policies, Unified Namespace definitions, aggregation policies, replication policies, AI tools, user permissions, and other information required for distributed coordination.

Because metadata is lightweight compared with operational data, it can be synchronized efficiently across the platform. Every node therefore develops a consistent understanding of the distributed environment without requiring centralized administration or centralized ownership of operational data.

The Distributed Metadata Layer is discussed in detail in Chapter 7, but it is introduced here because it provides the architectural foundation that allows independent nodes to behave as a single platform.

Virtualizing the Distributed Environment

The purpose of the AnyLog architecture is not to hide distribution completely. Instead, it virtualizes the aspects of the platform that would otherwise complicate application development and system operation.

Three complementary virtualization layers achieve this objective, as illustrated in Figure 2.

The **Virtual Data Lake** presents distributed operational data as though it were contained within a single logical database. Applications query information using standard SQL without needing to know where data is physically stored.

The **Unified Namespace** provides a logical representation of industrial assets independent of the databases that contain their operational information. Applications and AI navigate logical assets and their relationships rather than physical systems and relational tables, and multiple namespaces can coexist to support different operational perspectives over the same distributed data.

The **Single System Image** virtualizes the infrastructure itself. Administrators monitor and manage geographically distributed nodes through a unified operational view that resembles the experience of managing a centralized cloud platform.

Together, these three virtualization layers transform a distributed collection of autonomous systems into a cloud-like operational environment where operational data remains local, processing executes at the edge, and every node retains its autonomy.

Architectural Perspective

The AnyLog architecture is based on a simple observation: **distributed infrastructure does not require distributed complexity**.

By separating operational data from metadata, virtualizing access to distributed resources, and allowing autonomous nodes to cooperate through shared knowledge, the platform provides a unified environment for applications, administrators, and Artificial Intelligence.

Rather than replacing existing industrial systems, AnyLog builds a distributed layer above them. Existing databases, protocols, applications, and operational technologies continue performing their specialized functions while the platform provides the coordination, virtualization, and intelligence required to transform them into a coherent distributed data platform.

Key Architectural Insight

The AnyLog architecture does not centralize operational resources—it **virtualizes** them. Data remains in local databases, assets remain independent of physical storage, and infrastructure

remains geographically distributed. What becomes centralized is the **experience** of using the platform. Applications see a Virtual Data Lake, AI sees a Unified Namespace, and administrators see a Single System Image, while the underlying systems continue operating autonomously.

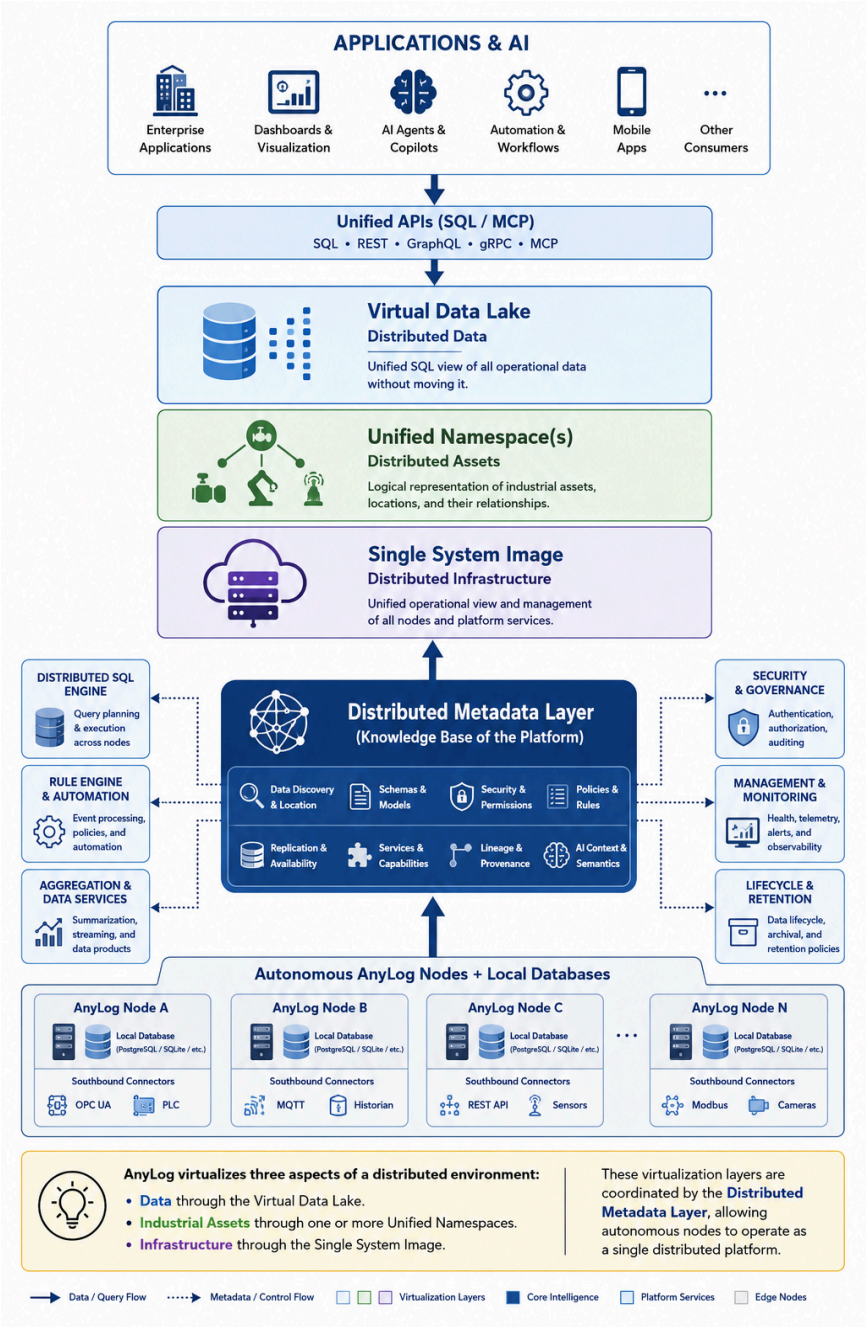

Figure 2. AnyLog Architecture Overview.

The AnyLog architecture transforms autonomous edge nodes into a unified distributed data platform. Operational data remains in local databases managed by each node, while the

*Distributed Metadata Layer provides the shared knowledge required to discover resources, coordinate distributed processing, enforce policies, and manage the platform. Applications and AI interact through unified APIs with a **Virtual Data Lake**, one or more **Unified Namespaces**, and a **Single System Image**, allowing distributed data, assets, and infrastructure to be accessed and managed as a single logical system.*

Chapter 4: Data Integration and Unified APIs

Connecting Operational Systems

The value of an industrial data platform depends on its ability to integrate with existing operational systems. Manufacturing equipment, sensors, programmable logic controllers (PLCs), historians, enterprise databases, cloud platforms, and business applications all contribute to an organization's operational picture. These systems have evolved over many years and typically use different communication protocols, data models, and storage technologies.

Replacing this infrastructure is rarely practical or desirable. Instead, the **AnyLog Edge Data Fabric** operates as an independent distributed data layer above existing systems. It interfaces directly with historians, industrial control systems, databases, cloud platforms, and enterprise applications while presenting applications and AI with a single logical interface to operational data distributed across edge locations, historians, and local databases.

Integration therefore becomes more than connecting devices and applications. It transforms independent operational systems into participants of a unified distributed platform, allowing applications and AI to integrate once with the **Edge Data Fabric** rather than separately with every historian, database, and industrial system.

Southbound Integration

Operational data enters the AnyLog platform through **Southbound Connector Services**. These services establish communication with industrial devices, applications, and databases using industry-standard protocols and interfaces.

The connector framework is intentionally modular. Each AnyLog node enables only the connectors required for its deployment, allowing the same software platform to operate efficiently on embedded gateways, industrial servers, enterprise systems, or cloud infrastructure. Connectors are activated through configuration rather than software development, enabling organizations to adapt deployments without modifying application code.

The platform includes native support for widely adopted industrial and enterprise technologies such as OPC UA, MQTT, Modbus, REST, gRPC, relational databases, object storage, and

file-based interfaces. Because the framework is open, organizations can also integrate proprietary systems or leverage third-party connectors when required.

Regardless of the protocol, every connector performs the same architectural function. It acquires operational information and streams it to the local AnyLog node. From that point forward, the ingestion process is fully automated and governed by policies. Incoming data is normalized, existing schemas are reused or new schemas are generated automatically, metadata is published to the Distributed Metadata Layer, and the data becomes immediately discoverable across the Edge Data Fabric. Policies determine how the data is retained, archived, replicated, aggregated, monitored, and secured, while making it available through the platform's unified APIs. Once ingested, the original communication protocol becomes largely irrelevant—the data is now part of the distributed platform and can be consumed consistently by applications, administrators, and AI regardless of how it was acquired.

Automatic Schema Management

One of the most persistent challenges in distributed industrial environments is maintaining consistent schemas across independently managed systems. Even identical equipment installed at different facilities frequently produces different table structures, naming conventions, or data types. Traditional integration projects often require extensive data mapping and transformation before information from multiple locations can be analyzed together.

AnyLog addresses this problem through automatic schema management.

Whenever new data is ingested, the receiving node first consults the Distributed Metadata Layer to determine whether an existing schema already describes the incoming information. If an appropriate schema exists, the node adopts that definition and stores the data using the established structure.

If no matching schema is found, the node automatically generates a new schema from the incoming data and publishes it to the Distributed Metadata Layer. Once published, this schema becomes immediately available throughout the platform. Any future node receiving similar data will reuse the existing definition rather than creating a new one.

This approach enables independently deployed nodes to converge towards a common logical schema without centralized administration or manual synchronization. Applications, distributed SQL, and AI can therefore interact with consistent data structures regardless of where the operational data originated.

This capability is particularly valuable in large deployments where thousands of edge nodes are installed incrementally over many years.

Unified Northbound APIs

Just as AnyLog standardizes the way data enters the platform, it also standardizes the way applications query it.

Applications, dashboards, enterprise systems, analytics platforms, and AI models all interact with the same distributed environment through a common set of **Northbound APIs**. These interfaces expose both operational data and metadata while shielding consumers from the physical organization of the underlying infrastructure.

Traditional applications typically access the platform using standard SQL or REST APIs. Additional interfaces such as gRPC and Kafka are available for environments that require alternative communication models. Artificial Intelligence interacts through the **Model Context Protocol (MCP)**, which provides semantic access to operational data, metadata, services, and the Unified Namespace.

Although these interfaces serve different types of consumers, they all operate on the same distributed architecture. Regardless of whether the request originates from a dashboard, an enterprise application, or an AI agent, the platform resolves the request using the Distributed Metadata Layer and the distributed query engine before returning a unified response.

Applications therefore integrate once—with the platform—not with every industrial system.

A Single API for Data and Metadata

One of the distinguishing characteristics of the AnyLog architecture is that operational data and metadata are exposed through the same logical platform.

Traditional applications often rely on separate mechanisms to retrieve operational data, discover schemas, locate databases, inspect system configuration, and understand asset relationships. These capabilities are typically implemented by different teams, coordinated across multiple project managers, and connected through independently maintained APIs and configuration files. The result is typically a brittle web of dependencies in which a single team inevitably changes an API, schema, or endpoint without notice and unintentionally disrupts applications, integrations, and workflows owned by other teams.

AnyLog unifies these capabilities through a consistent programming model.

Applications no longer need separate integrations for each database, asset system, API, or infrastructure team.

Instead, metadata is treated as a first-class resource, which enables applications to dynamically discover new datasets, nodes, schemas, and services as they are introduced. This makes applications significantly more adaptive by reducing their dependence on static configuration, manually maintained mappings, and undocumented interfaces.

This unified approach simplifies software development while allowing the distributed platform to evolve without disrupting consuming applications.

Configuration Rather Than Integration Projects

Industrial integration projects have traditionally required months of custom development before applications could begin using operational data. Connectors, schema mapping, data transformation, and application-specific interfaces were typically developed as independent software projects, making deployments costly to build and difficult to maintain.

AnyLog replaces much of this effort with configuration.

Connector services, protocol support, schema management, metadata publication, APIs, and platform services are all enabled through configuration and policies rather than custom code. The same software platform can therefore be deployed across heterogenous operational environments while requiring only deployment-specific configuration.

As organizations expand their infrastructure, they configure additional nodes rather than developing new integration software. This significantly reduces deployment time and allows engineering effort to focus on business applications instead of infrastructure.

Integrating with Historians and Cloud Platforms

The AnyLog Edge Data Fabric is designed to complement existing historians, enterprise data platforms, and cloud infrastructure rather than replace them. Most organizations have already invested in historians for process monitoring, cloud platforms for enterprise analytics, and data lakes for long-term hosting. These systems continue to provide value and remain important components of the overall information architecture.

The difference lies in **what information is transferred**.

Traditional architectures continuously stream every operational event from edge systems into centralized repositories, regardless of whether the information will ever be used. As deployments grow, this approach consumes significant network bandwidth, increases storage costs, and requires centralized infrastructure to process and retain massive volumes of raw operational data.

The AnyLog architecture follows a different model. Operational data remains within the local AnyLog nodes, where it is immediately available for monitoring, automation, AI inference, and distributed SQL queries. Instead of continuously forwarding every event to enterprise systems, the platform uses policy-driven aggregations to determine **what information should be shared** and **when**.

Aggregations are calculated close to where the data is generated and can represent production summaries, equipment utilization, energy consumption, quality metrics, alarms, statistical measurements, or any other business-relevant information. Only these aggregated results are transferred to historians, enterprise databases, or cloud platforms, dramatically reducing the volume of information that traverses the network.

When detailed operational analysis is required, applications continue to access the original data through the AnyLog Edge Data Fabric using distributed SQL or the platform's unified APIs. There is no need to replicate every event to centralized infrastructure simply to preserve access to historical operational information.

This architecture allows historians and cloud platforms to focus on enterprise visibility, long-term analytics, and business intelligence, while the AnyLog Edge Data Fabric manages operational data at the edge and delivers only the information that creates enterprise value.

Key Architectural Insight

Data integration is not merely about collecting information. It is the process by which isolated operational systems become part of a distributed data platform. Once data has been ingested, normalized, and described through shared metadata, every application, administrator, and AI model accesses it through the same logical architecture, independent of its original source, protocol, or physical location.

Chapter 5: Distributed Query Processing

Querying Data Without Centralizing It

For decades, enterprise data platforms have followed a common architectural pattern. Before data can be queried, analyzed, or consumed by applications, it is first collected into a centralized repository. Whether the destination is a relational database, a historian, a cloud data lake, or a data warehouse, the underlying assumption remains the same: operational data must be moved before meaningful analysis can begin.

This approach simplifies query execution because all information resides in one place. However, as industrial environments become increasingly distributed, it also introduces significant challenges. Large volumes of operational data must be continuously transferred across the network, centralized infrastructure must be expanded as deployments grow, and analytical workloads often operate on replicated data rather than on the most current operational information.

AnyLog was designed around a different assumption.

Instead of moving operational data to a centralized query engine, AnyLog moves the query to the operational data.

Operational information remains within the local databases managed by each AnyLog node. When an application requests information, the platform automatically discovers where the required data resides, executes the query locally at those locations, and returns only the results required to satisfy the request.

From the perspective of the application, the entire distributed environment behaves as though it were a single database, even though the underlying information remains distributed across autonomous systems.

A Distributed Query Engine

Every AnyLog node is capable of receiving and coordinating distributed queries.

Applications submit standard SQL through SQL, REST APIs, or the Model Context Protocol (MCP) without specifying where the data is stored. The receiving node temporarily assumes the role of **Query Coordinator**, becoming responsible for planning and managing the distributed execution of the request.

Rather than maintaining a static catalog of participating databases, the Query Coordinator consults the Distributed Metadata Layer to determine which nodes host the requested datasets. Using this information, it constructs an execution plan that distributes the workload only to the nodes required to satisfy the query.

Each participating node executes the SQL statement locally against its own operational databases using its own processing resources. The resulting records are returned to the Query Coordinator, which aggregates the partial result sets into a single unified response before returning it to the requesting application.

Throughout this process, operational data never leaves its local databases except for the records explicitly requested by the query.

Processing Data In-Place

The architectural significance of distributed query processing extends far beyond reducing network traffic. By executing queries where operational data resides, the AnyLog Edge Data Fabric eliminates the need to continuously move large datasets to centralized infrastructure for analysis. The network carries query requests and result sets rather than raw operational data, significantly reducing bandwidth consumption, storage requirements, and cloud infrastructure costs.

Processing data in-place also enables applications and AI to operate on the most current operational information. Queries execute directly against local databases, eliminating the delay associated with replicating or synchronizing data to centralized repositories before it can be analyzed.

Equally important, organizations retain complete ownership and control of their operational data. The Edge Data Fabric uses the Distributed Metadata Layer to identify the nodes that host the requested data, coordinates query execution across authorized participants, and applies security policies locally at each node. Parallel execution minimizes response time while ensuring that only authorized nodes participate in satisfying the request.

Parallel Execution Across the Platform

Distributed query processing naturally enables parallel execution.

Each participating AnyLog node contributes its own processors, memory, storage, and database engine to the execution of distributed workloads. Each node evaluates only the portion of the query associated with its local datasets before returning the results to the Query Coordinator.

Unlike centralized platforms, where growing datasets place increasing demands on a single database server, the AnyLog architecture distributes both operational data and processing capacity across the platform.

As additional nodes are deployed, the amount of operational data increases, but so does the total computing power available to process it. The platform therefore scales horizontally by adding autonomous nodes rather than vertically by expanding centralized infrastructure.

This architecture allows the AnyLog Edge Data Fabric to scale from a single facility to globally distributed industrial environments while preserving the same programming and query model, regardless of whether the data resides on one node or thousands.

Metadata-Driven Execution

The Distributed Metadata Layer plays a central role in query execution.

Before processing begins, the Query Coordinator consults metadata to determine where datasets reside, which replicas are available, and which security policies apply.

Because this knowledge is maintained independently of the operational data itself, applications remain completely unaware of changes within the physical infrastructure. Databases may be relocated, replicated, expanded, or replaced without affecting application logic or SQL statements, or AI workflows.

This approach separates the logical view presented to applications from the physical organization of the underlying platform. Applications interact with the Edge Data Fabric, while

the platform dynamically discovers resources, coordinates execution, and routes requests to the appropriate participating nodes.

This separation enables the AnyLog Edge Data Fabric to evolve continuously while preserving a stable programming model, allowing distributed infrastructure to change without requiring applications to be redesigned or redeployed. .

AI Uses the Same Architecture

Distributed query processing is not limited to traditional applications.

Artificial Intelligence uses exactly the same execution model.

When an AI application submits a request through the Model Context Protocol, the platform performs the same sequence of operations used for SQL queries. Metadata identifies the relevant datasets, participating nodes execute processing locally, and the platform assembles the results before presenting them to the AI model.

This common execution engine ensures that applications, dashboards, analytics platforms, and AI systems all operate on the same distributed view of operational information.

There is no separate AI database, no independent data synchronization process, and no specialized infrastructure required for AI workloads. Artificial Intelligence becomes another consumer of the distributed platform.

Performance Characteristics

The AnyLog Edge Data Fabric achieves high performance by distributing both data ingestion and query execution across autonomous nodes. As the deployment grows, operational data, storage, and processing capacity expand together, allowing performance to scale horizontally rather than concentrating workload on centralized infrastructure.

Data ingestion is inherently parallel. Each node independently collects, stores, and processes data from its assigned assets, making new information immediately available for local applications, distributed queries, and AI. Because data is ingested directly at the edge, it does not need to traverse centralized brokers, databases, or cloud pipelines before it can be used.

Query execution is equally distributed. Applications submit requests to the Edge Data Fabric rather than individual nodes. The platform uses the Distributed Metadata Layer to identify where the required data resides, selects the appropriate participating nodes, executes queries in parallel, and returns a unified result. As additional nodes are deployed, both the available data and the aggregate processing capacity increase, allowing the platform to support larger workloads without creating centralized bottlenecks.

Architectural Perspective

Distributed query processing is the capability that transforms the architectural principles introduced in earlier chapters into a practical operating model.

Keeping operational data local would provide limited value if applications still needed to know where every dataset was stored. Likewise, a distributed metadata layer would be of little use if applications had to manually coordinate execution across multiple databases.

The distributed query engine removes these responsibilities from the application.

Applications describe *what* information they require. The platform determines *where* the information resides, *how* it should be retrieved, *which* nodes should participate, and *how* the results should be combined.

This separation between logical intent and physical execution allows organizations to manage geographically distributed operational data with the simplicity traditionally associated with a single database while preserving the autonomy, performance, and resilience of distributed edge computing.

Key Architectural Insight

The defining characteristic of the AnyLog query engine is not that it executes SQL across multiple databases. It is that **applications no longer need to understand the physical organization of operational data**. By moving queries to the data rather than moving data to the query, the platform preserves local ownership, minimizes network traffic, enables horizontal scalability, and provides a single logical view of a distributed operational environment.

Chapter 6: Distributed Platform Services

From a Data Platform to an Operational Platform

The previous chapters described how operational data is integrated into the AnyLog platform, how it becomes discoverable through distributed metadata, and how applications query distributed datasets as though they were contained within a single database.

Those capabilities establish the foundation of the platform, but they are only part of the architecture.

A modern industrial data platform must do more than store and retrieve information. It must service information to where it creates value, manage the lifecycle of operational data, automate repetitive tasks, remain available despite failures, and scale naturally as organizations expand their operations.

Rather than implementing these capabilities as separate products or external services, AnyLog integrates them into the platform itself. Each service operates on the same distributed architecture, shares the same metadata model, and is managed through the same policy framework. As a result, organizations configure platform behavior rather than assembling independent technologies.

This chapter describes the distributed services that transform AnyLog from a distributed query platform into a complete operational platform.

Moving Information Instead of Data

Keeping operational data close to where it is generated is one of the fundamental architectural principles of AnyLog. Nevertheless, there are many situations in which information should be shared beyond the local environment.

Enterprise dashboards require production summaries from multiple facilities. Executives monitor key performance indicators across an organization. Cloud applications consume long-term trends, while regulatory systems require periodic reporting. None of these use cases require continuous replication of every operational measurement.

AnyLog therefore distinguishes between **operational data** and **business information**.

Rather than transferring complete datasets across the network, the platform computes summaries close to where the data is generated and distributes only the resulting information. Aggregations such as production totals, energy consumption, equipment utilization, quality metrics, or environmental statistics are calculated locally before being published to other nodes or enterprise systems.

Applications that require summary information retrieve aggregates. Applications that require operational detail continue querying the original data directly from its source.

Managing the Data Lifecycle

Operational data changes in value over time.

Recent information is typically accessed for real-time monitoring, process control, automation, and AI inference. Historical information becomes increasingly important for trend analysis, regulatory compliance, predictive maintenance, and long-term business intelligence. As data ages further, it is often retained primarily for governance and auditing purposes.

The AnyLog platform manages this lifecycle through configurable retention policies.

Organizations define how long operational data remains in high-performance local databases, when historical information should be archived, where archived data should be stored, and how

long it should be retained before deletion. These policies are distributed through the metadata layer and enforced automatically by participating nodes.

Archival storage may consist of object storage, cloud repositories, file systems, or other long-term storage technologies selected by the organization. Regardless of where historical information resides, it remains part of the logical platform and can continue to be accessed through the same SQL interface.

Applications therefore interact with a consistent data model while the platform optimizes storage and retention behind the scenes.

Automation Through Policies

Large distributed environments cannot depend on administrators or centralized applications to perform every operational task manually.

Each AnyLog node therefore includes a policy-driven automation framework that continuously evaluates operational conditions and performs predefined actions when those conditions are satisfied.

Unlike traditional automation systems that rely on centralized orchestration, AnyLog evaluates policies locally where operational data resides. The **Rule Engine** monitors incoming data, executes scheduled tasks, triggers aggregations, manages retention, invokes AI analysis, generates alerts, or coordinates administrative activities across the distributed platform.

Because automation is defined through policies rather than application code, organizations modify platform behavior by changing configuration rather than developing software. As operational requirements evolve, new automation strategies can be deployed quickly while maintaining consistent behavior across the platform.

The **Rule Engine** therefore becomes an operational service that coordinates platform activities rather than a standalone automation product.

Scaling by Adding Nodes

Industrial environments expand by adding new facilities, production lines, substations, retail locations, renewable energy sites, or intelligent devices. The AnyLog architecture reflects this reality.

Every new AnyLog node contributes more than operational data. It also contributes additional processors, memory, storage, metadata, services, and operational intelligence.

As deployments grow, processing capacity grows together with the operational workload. Distributed SQL executes concurrently across participating nodes, automation is evaluated

locally, AI inference can execute closer to the data, and aggregation workloads are naturally distributed throughout the platform.

Unlike centralized architectures, where additional data often creates larger processing bottlenecks, AnyLog grows by distributing both storage and computation across autonomous nodes.

The platform therefore becomes more capable as it expands, with every new node contributing additional data, processing capacity, storage, and services.

High Availability Through Distribution

Industrial operations require continuous access to operational data even when hardware, networks, or individual systems experience failures.

AnyLog approaches availability as an architectural property rather than a feature added to centralized infrastructure.

Because every node operates autonomously, local applications continue functioning even when communication with the rest of the platform is interrupted. Data ingestion, SQL processing, automation, AI inference, and management continue operating using the information available locally.

Where additional protection is required, configurable replication policies maintain copies of selected datasets on multiple nodes. The distributed query engine automatically considers these replicas when constructing execution plans, allowing applications to continue accessing information according to the organization's availability policies.

The combination of autonomous nodes and policy-driven replication minimizes operational risk, preserves availability across diverse deployment conditions, and eliminates dependence on any single participant.

Architectural Perspective

Distributed platforms should not simply provide access to distributed data. They should also provide the operational services required to manage that data throughout its lifecycle.

AnyLog accomplishes this by integrating aggregation, lifecycle management, automation, scalability, and availability directly into the platform.

As a result, the platform evolves beyond a distributed query engine into a distributed operating environment for industrial data.

Key Architectural Insight

Distributed platform services are not independent applications layered on top of the AnyLog architecture—they are **native capabilities of the platform**. Because they share the same metadata, policy framework, and distributed execution model, organizations manage aggregation, automation, lifecycle, scalability, and availability as parts of a single distributed operating environment rather than as separate technologies.

Chapter 7: Operating the Distributed Platform

From Distributed Systems to One Operational Environment

Designing a distributed platform is only part of the challenge. Once hundreds or thousands of edge nodes have been deployed, organizations must operate them as a coherent system. Administrators need to understand where operational data resides, monitor the health of the platform, diagnose failures, manage policies, enforce security, and coordinate activities across geographically distributed locations.

Traditional edge deployments often require administrators to manage each system independently. Every gateway, server, or industrial computer becomes another machine that must be monitored, configured, and maintained. As deployments expand, operational complexity grows much faster than the infrastructure itself.

AnyLog was designed to solve this problem by making a distributed edge deployment operate like a cloud platform.

Although operational data, computing resources, and services remain physically distributed, the platform presents a unified operational environment through which administrators can monitor, manage, secure, and coordinate the entire deployment. Applications experience a single data platform, while administrators experience a single operational platform.

This capability is achieved through the combination of the **Single System Image**, the **Distributed Metadata Layer**, and a common policy framework that governs every service within the platform.

A Single System Image

One of the defining characteristics of modern cloud platforms is that administrators rarely manage individual servers. Instead, they interact with a unified management environment that represents the entire infrastructure.

AnyLog extends this operational model to distributed edge computing.

The **Single System Image (SSI)** provides a unified operational view of each participating AnyLog node regardless of its physical location. From a single interface, administrators can observe the operational status of the platform, discover participating nodes, inspect services, monitor databases, review policies, and manage distributed resources.

This does not mean that infrastructure has been centralized. Each node continues to execute independently and maintains complete ownership of its operational data. The Single System Image simply virtualizes the operational experience, allowing geographically distributed systems to be managed as though they were part of a single computing environment.

As new nodes are deployed, they automatically become visible within the platform. Likewise, when nodes become unavailable, their operational status is immediately reflected without requiring manual configuration.

Unified Management

The Single System Image provides much more than system monitoring.

Administrative operations can be directed toward an individual node, a selected group of nodes, or the entire distributed platform. Configuration changes, policy updates, metadata inspection, service management, and operational diagnostics are all performed through the same management model regardless of deployment size.

Because every node executes the same software platform, administrators manage capabilities rather than individual server types. New services can be enabled through configuration, policies can be updated centrally and distributed automatically, and operational behavior remains consistent across every participating location.

This greatly simplifies the administration of large-scale edge deployments while preserving the flexibility to configure individual nodes according to local requirements.

Observability and Diagnostics

Operating a distributed platform requires visibility into more than the health of individual systems. Administrators must understand how distributed services behave across the entire environment.

The AnyLog platform therefore offers observability on every distributed service.

Administrators can determine which nodes host particular datasets, inspect distributed metadata, monitor replication activity, review aggregation processes, evaluate automation policies, and trace the execution of distributed SQL queries.

When a distributed query is executed, the platform can identify the participating nodes, display execution times for each node, show how results were combined, and reveal where delays

occurred. This level of transparency allows distributed execution to be analyzed using concepts similar to database execution plans while providing insight into the behavior of the entire distributed platform.

Diagnostic tools also simplify troubleshooting by allowing administrators to verify connectivity, inspect node health, analyze metadata synchronization, review policy execution, and examine platform activity without manually accessing every participating system.

Security as a Distributed Service

Managing a distributed platform requires more than protecting network boundaries. Every participating node must be able to establish trust, authenticate other nodes, verify the integrity of exchanged information, and enforce consistent security policies without relying on a centralized security server.

Unlike traditional architectures that protect a centralized repository of operational data, the AnyLog Edge Data Fabric secures autonomous nodes that cooperate through authenticated communication, cryptographic identities, and distributed security policies.

Every AnyLog node is assigned a unique cryptographic identity based on public and private key pairs. Nodes and applications authenticate one another using digital certificates, while cryptographic signatures verify the origin and integrity of exchanged messages. Communication between nodes can be encrypted, ensuring that metadata, queries, and results remain protected while traversing untrusted networks. These cryptographic mechanisms establish trust between autonomous nodes without requiring a centralized authority to participate in every transaction.

Security policies are maintained within the Distributed Metadata Layer and distributed across the Edge Data Fabric. Every node independently enforces the security policies associated with its own operational data. Applications, administrators, AI systems, and peer nodes are authenticated before accessing platform services, and every request is evaluated according to locally enforced authorization policies. Because policy enforcement occurs where the data resides, organizations retain complete control over who may access their information and under what conditions.

The Distributed Metadata Layer provides a distributed and tamper-resistant repository for security policies, node identities, service registrations, and other platform metadata. This ensures that participating nodes maintain a consistent view of the platform while eliminating the need for a centralized metadata server.

Because operational data remains local, participation in the distributed platform never requires organizations to surrender ownership of their information. Data is shared only according to locally enforced policies, while trust between nodes is established through cryptographic verification rather than implicit network trust.

This architecture naturally aligns with **Zero Trust** principles. Trust is established through authenticated identities, cryptographic verification, and explicit authorization for every operation, rather than through network location or centralized ownership. As a result, organizations can securely collaborate across distributed industrial environments while maintaining complete control over their operational data.

Collaboration Without Centralization

Most AnyLog deployments operate within a single organization, providing a unified platform across factories, production lines, substations, retail stores, or other distributed operational environments. However, many industrial ecosystems extend beyond organizational boundaries. Manufacturers collaborate with suppliers, equipment vendors, logistics providers, utilities, contractors, and customers, creating opportunities to securely share operational information across multiple organizations.

The AnyLog Edge Data Fabric supports this model **when collaboration is required**, but it does not depend on it. Organizations can operate completely independently or selectively participate in a shared distributed platform.

When enabled, each organization continues operating its own AnyLog infrastructure while selectively publishing metadata that describes the resources, datasets, or services it chooses to expose. Applications discover these resources through the Distributed Metadata Layer and access operational information according to the security policies defined and enforced by the organization that owns the data.

Because operational data remains under local ownership, organizations determine **what information is shared, with whom, and under what conditions**. Participation in a collaborative environment does not require centralizing databases or relinquishing administrative control.

This architecture allows multiple independent organizations to cooperate securely while maintaining complete operational and administrative independence. Collaboration therefore becomes a configurable business decision, implemented through policies, rather than an architectural requirement.

Operating an AI-Native Platform

Artificial Intelligence is becoming an operational component of industrial systems rather than a standalone analytical tool. As AI becomes integrated into day-to-day operations, it must be managed alongside applications, databases, and infrastructure.

Within the AnyLog architecture, AI is treated as another consumer of the distributed platform.

AI models interact through the Model Context Protocol using the same metadata, security framework, and distributed execution engine that support traditional applications. Administrators can monitor AI activity, manage AI services, control access through policies, and trace AI interactions with operational resources.

Because AI shares the same architectural foundation as every other platform service, organizations avoid the operational complexity of maintaining a separate AI infrastructure.

A Cloud Operating Model for the Edge

Perhaps the most significant operational contribution of the AnyLog architecture is that it brings cloud-style management to distributed edge computing without requiring cloud-centralized infrastructure.

Applications experience a Virtual Data Lake rather than independent databases.

Industrial assets are represented through one or more Unified Namespaces rather than isolated information models.

Administrators manage a Single System Image rather than hundreds of independent systems.

Artificial Intelligence interacts through a common semantic interface rather than proprietary integrations.

Together, these virtualization layers allow the platform to operate with the simplicity traditionally associated with cloud computing while preserving the autonomy, resilience, and low-latency characteristics of edge deployments.

Key Architectural Insight

The objective of the AnyLog architecture is not to centralize the edge—it is to **centralize the operational experience**. By combining the Single System Image, distributed metadata, policy-driven management, and unified security, the platform enables geographically distributed systems to be monitored, managed, and secured as though they were part of a single cloud environment, while every node continues operating autonomously and retaining ownership of its operational data.

Chapter 8: Building an AI-Native Edge Data Platform

The Next Generation of Industrial Computing

Industrial computing is entering a new phase. For decades, digital transformation focused on collecting operational data and making it available for reporting, visualization, and enterprise

analytics. More recently, cloud computing expanded the ability to store and process this information at unprecedented scale.

The emergence of Artificial Intelligence introduces a fundamentally different requirement.

AI systems are not passive consumers of historical information. They continuously interpret operational conditions, correlate data from multiple sources, make recommendations, invoke software services, and increasingly participate in autonomous decision making. To perform these functions effectively, AI requires access to far more than raw sensor values. It needs context, relationships between assets, knowledge of available services, and the tools to access the data.

Traditional industrial architectures were not designed for this style of computing. They often rely on proprietary integrations, centralized repositories, and application-specific interfaces that become increasingly difficult to maintain as organizations deploy multiple AI assistants, copilots, and autonomous agents.

Supporting Artificial Intelligence therefore requires more than adding a Large Language Model (LLM) or Small Language Model (SLM) to existing infrastructure. It requires a platform that presents the distributed operational environment as a coherent and discoverable system.

This is the role of the AnyLog Edge Data Fabric.

A Platform Rather Than a Collection of Technologies

Throughout this guide, we have described individual architectural components: the AnyLog Node, the Distributed Metadata Layer, the Virtual Data Lake, the Unified Namespace, distributed SQL, the Model Context Protocol, automation, and the Single System Image.

Although each component provides significant value independently, their true strength lies in how they synergetically operate.

The Distributed Metadata Layer provides the knowledge required to understand the platform. The Virtual Data Lake allows applications to access distributed operational data through standard SQL without knowing where it resides. The Unified Namespace provides logical representations of industrial assets that are independent of physical databases. The Single System Image presents distributed infrastructure through a unified operational view. The Model Context Protocol exposes these resources to Artificial Intelligence through a common semantic interface.

Together, these components create a distributed platform that behaves as a single logical environment while preserving the autonomy of every participating node.

This collection of technologies are the defining characteristics of the AnyLog Edge Data Fabric.

Virtualizing the Distributed Environment

At the heart of the platform is a simple architectural concept: **virtualization**.

Traditional virtualization abstracts physical servers into virtual machines. Cloud platforms abstract physical infrastructure into elastic services. AnyLog extends this idea to operational data.

Instead of exposing physical databases, communication protocols, and edge systems directly to applications, the platform virtualizes the distributed environment through three complementary layers.

The **Virtual Data Lake** presents distributed operational data as a single logical resource.

The **Unified Namespace** provides one or more logical representations of industrial assets and their relationships.

The **Single System Image** presents geographically distributed infrastructure through a unified management model.

Applications, administrators, and AI interact with these logical views rather than individual systems, while the Edge Data Fabric abstracts the complexity of the underlying distributed infrastructure.

A Common Foundation for Applications and AI

Historically, enterprise applications and Artificial Intelligence have evolved as separate technology stacks. Business applications interact with databases through APIs and SQL, while AI systems often require additional integration layers to obtain operational context and invoke software services.

The AnyLog architecture removes this distinction.

Applications and AI consume the same distributed platform.

Traditional software retrieves operational information through SQL, REST, or gRPC. AI systems use the Model Context Protocol to discover metadata, navigate the Unified Namespace, execute distributed queries, and invoke platform services. Both rely on the same metadata, security policies, and distributed execution engine.

This unified architecture eliminates the need to build separate data infrastructures for analytics, automation, and AI. Instead, all consumers operate on a common foundation that evolves naturally as the platform grows.

Enabling Autonomous Operations

Artificial Intelligence is only one component of autonomous industrial systems. True autonomy requires real-time access to operational data, distributed decision making, automation, policy enforcement, security, and continuous observability.

The AnyLog Edge Data Fabric provides this operational foundation.

Operational data remains close to the systems that generate it, allowing local decisions to be made with minimal latency. Distributed SQL provides enterprise-wide visibility without requiring centralized data repositories. Policy-driven automation coordinates routine operational tasks, while the Single System Image enables administrators to manage geographically distributed infrastructure through a unified operational model.

Artificial Intelligence becomes a participant in the AnyLog Edge Data Fabric rather than a separate technology stack. Authorized AI agents access the same Virtual Data Lake, Unified Namespace, and Distributed Metadata Layer used by applications and administrators, giving them a shared operational view of distributed data, assets, services, and platform capabilities.

The Edge Data Fabric provides a shared operational context for every authorized AI agent, eliminating the need for agents to exchange data locations, schemas, asset relationships, and service information directly with one another. **Instead of creating an expanding web of agent-to-agent dependencies, every agent interacts with the same distributed platform.**

As organizations increasingly automate manufacturing, energy production, transportation, telecommunications, healthcare, and critical infrastructure, this distributed architecture provides the foundation upon which autonomous operations can be built.

Looking Beyond the Cloud

Cloud computing transformed enterprise IT by centralizing compute and storage resources. Edge computing extends this model by recognizing that not all information should be centralized.

The next stage of industrial computing is unlikely to replace either model. Instead, organizations will increasingly operate hybrid environments in which cloud and edge platforms cooperate.

Within this architecture, the cloud becomes a participant rather than the center of the system.

Long-term analytics, enterprise applications, large-scale AI training, and business reporting continue benefiting from cloud infrastructure. Operational processing, local AI inference, automation, and real-time decision making execute where operational data is generated.

AnyLog enables these environments to function as a single platform while allowing organizations to determine which information remains local and which information should be shared.

Rather than competing with the cloud, the platform extends the cloud to distributed operational edge environments.

An Architecture Designed to Evolve

One of the defining characteristics of successful software architectures is their ability to accommodate technologies that did not exist when the architecture was first designed.

Industrial protocols will continue to evolve. New databases will emerge. AI models will improve. Communication technologies will change. Computing hardware will become more powerful, storage capacity will continue to increase, and infrastructure will expand across increasingly distributed environments.

The architectural principles described throughout this guide remain applicable regardless of these changes.

Operational data will continue being generated at the edge.

Applications and AI will continue requiring a unified and real-time view of distributed resources.

Organizations will continue demanding local ownership, security, resilience, and scalability.

By separating physical infrastructure from logical operation through metadata-driven virtualization, AnyLog provides an architecture capable of evolving with future technologies rather than depending on specific products or protocols.

Chapter 9: Conclusion

The AnyLog Edge Data Fabric represents a different way of thinking about managing distributed edge data.

Rather than centralizing operational information before it can be used, the platform coordinates autonomous nodes through shared knowledge. Instead of requiring applications to understand distributed infrastructure, it virtualizes data, assets, and infrastructure into a single logical environment. Rather than building separate architectures for enterprise software, operational technology, and Artificial Intelligence, it provides a common foundation upon which all three can operate.

The result is a distributed data platform that combines the operational simplicity traditionally associated with cloud computing with the performance, resilience, and autonomy required for edge computing.

As organizations continue building intelligent factories, connected infrastructure, autonomous transportation systems, distributed energy networks, and AI-driven industrial operations, architectures based on these principles will become increasingly important.

The future of industrial computing is not defined by where data is stored. It is defined by how effectively distributed data, applications, and Artificial Intelligence operate as a unified platform while preserving the autonomy of every participating node. The AnyLog Edge Data Fabric delivers that architecture.