TDengine

TDengine 2026 Product White Paper

Bring AI to Every Industrial Data Stream

Contents

Abstract

The industrial world has never lacked data. What it lacks is the ability to extract insight from data and drive decisions. For decades, industrial enterprises have invested heavily in data infrastructure. The cost of data collection, storage, and classification has steadily declined, yet the efficiency of converting data into business value has never met expectations. Three structural problems, each reinforcing the others, have kept industrial data platforms from delivering on their promise: an over-reliance on domain experts, an over-reliance on IT staff, and the absence of built-in deep analytics.

In 2026, AI agents have reached an inflection point. With capabilities in memory, reasoning, planning, and autonomous execution, they offer a new way to break this deadlock. But bringing AI to the plant floor means confronting a set of hard challenges: fragmented data, explosive scale, real-time processing demands, process analytics complexity, deep industry barriers, and the non-negotiable requirement for security and trust.

TDengine's answer is to design the industrial data platform around AI from the ground up. Through a loosely coupled, modular architecture, TDengine covers the full lifecycle of industrial data: data ingestion, storage and query, real-time computing, semantic modeling, controlled and trusted AI agent execution, full-spectrum process analytics, zero-code applications, open data services for multiple use cases, and flexible deployment across public cloud, private cloud, and edge-cloud environments. Its mission is to help every industrial data stream generate insight and drive action with AI. Bring AI to Every Industrial Data Stream.

This white paper lays out TDengine's design philosophy, technical architecture, and industry practices. Written for industrial decision-makers, technical managers, and data practitioners, it provides a complete reference blueprint for building an AI-era industrial data platform.

CHAPTER 01

Three Structural Problems in Industrial Data Platforms

For decades, industrial data platforms have promised to turn data into value, yet the results fall short. Three structural problems hold the entire category back: over-reliance on domain experts, over-reliance on IT staff, and the absence of built-in deep analytics.

01

TDengine

CHAPTER 01

Three Structural Problems in Industrial Data Platforms

Industrial data historians and industrial IoT platforms have existed for decades. Hundreds of vendors compete globally, offering tools for data collection, storage, visualization, and analytics.

The industrial data historian was born on the plant floor. It connects directly to automation systems such as PLCs, DCS, and SCADA, collecting time-series data (temperature, pressure, vibration, current) from equipment at millisecond frequencies. After efficient compression, this data is stored persistently and displayed in real time through process diagrams and trend charts. The core value is letting operators see what is happening on their equipment, whether a compressor is running normally or a production line's key parameters are drifting from target. After decades of refinement, the data historian has become the digital infrastructure of industry, supporting everything from daily monitoring to incident forensics.

The industrial IoT platform emerged as internet technologies penetrated the industrial domain. It layers cloud architecture, IoT connectivity, application development platforms, and data analytics tools on top of the data historian, aiming to bridge the gap from equipment to management, from edge to cloud. Its vision is broader: not just monitoring equipment status, but enabling cross-plant and cross-enterprise data sharing, business collaboration, and intelligent decision-making as a unified platform for industrial digital transformation.

Both categories share the same clear goal: turning industrial data into business value. As long as industrial enterprises keep operating, they need real-time equipment monitoring and analysis of quality, energy consumption, and capacity utilization. This market demand will persist regardless of how production processes and technologies evolve.

Yet in practice, many enterprises have invested heavily in industrial data historians and IoT platforms without seeing commensurate business value. The root cause is not a missing feature in any particular product. It lies in three structural problems that affect the entire category.

1.1 Over-reliance on domain experts: knowledge cannot be retained

Every industry carries its own body of domain knowledge: process parameter optimization in chemicals, load dispatching in power, compliance validation in pharmaceuticals. Building this expertise takes years of practice, and that is not going to change.

The problem is that most platforms treat domain knowledge as a prerequisite for using data, not as an asset that can be learned, amplified, shared, and handed down. In a typical plant, fewer than 2% of the workforce can pull meaningful insight out of time-series data on their own. A veteran engineer with 20 years of experience may hear a machine running differently and know something is wrong. But the reasoning behind that judgment is rarely captured by the system: which vibration pattern sounded abnormal, which temperature trend deserved attention, and why those signals mattered. When that engineer leaves, the knowledge leaves with them.

Traditional platforms give you tools to store and display vast amounts of data, but they do nothing to turn domain knowledge and expert experience into reusable assets. Until AI can amplify what ordinary workers are capable of, the value of industrial data will always be bottlenecked by how few experts you have.

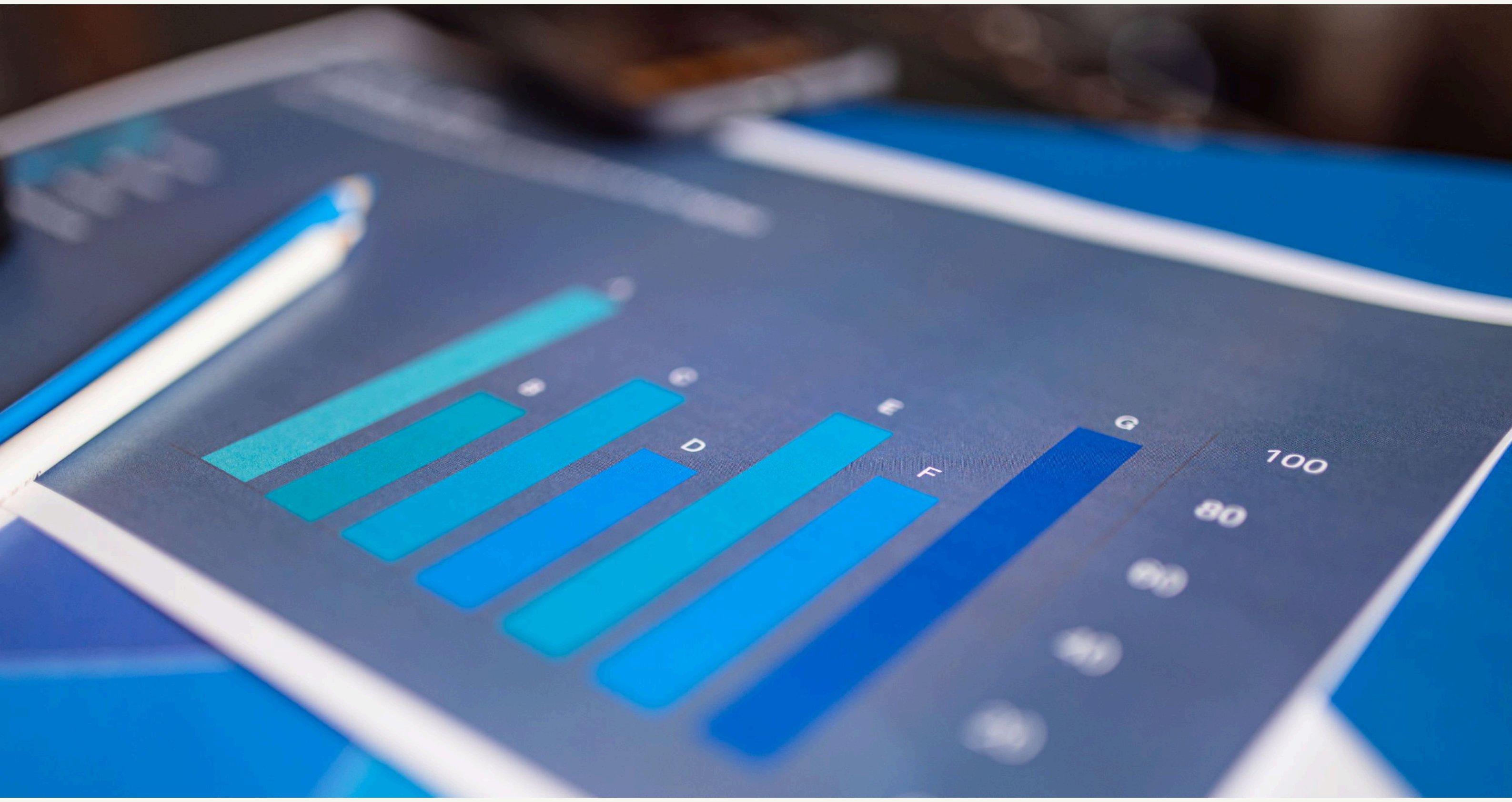
1.2 Over-reliance on IT staff: the path from data to value is too long

Many industrial data platforms are built for IT people, not for business people.

A process engineer asks a routine question: What happened before Line 3 shut down abnormally last week? To answer it, they first need IT to write SQL and extract the data, then need IT again to build the charts. If the question goes deeper, say comparing this shutdown against every shutdown from the past three months, a development team steps in to write scripts, train models, test, and deploy. And if the engineer wants to tweak a process parameter based on the result, that means a change control process and another round with IT to make the adjustment on the line.

From question to answer, this process often takes days, sometimes weeks. The actual insight, the few minutes when the engineer studies the data and makes a judgment, is a small sliver of that wait. Ideas fade while waiting. The rest of the time disappears into handoffs between teams and departments.

Every step runs through IT: ingesting data means writing code, querying means SQL, analysis means Python scripts, visualization means hand-built dashboards, and automation means a developer building the workflow. Every change goes through a development-test-deployment cycle. Business response cycles run in hours; IT delivery cycles in weeks. Enterprises spend heavily, and data still turns into value painfully slowly.



1.3 Lack of built-in deep analytics: insight cannot become action

Most industrial data platforms do one thing well: visualization. They turn equipment status and production metrics into charts. But a chart only answers one question: what happened.

Ask why it happened, or what will happen next, and the platform goes quiet. The usual advice is to export the data as CSV, write a Python or MATLAB script, run the analysis by hand, and push the result back into production as an alert rule or a process change.

That is the gap. Data lives inside the platform, analysis happens in scripts outside it, and the resulting action lands in yet another system. Every handoff adds latency, erodes meaning, and invites error. A single plant can end up maintaining dozens of these export-analyze-writeback pipelines, each one a fragile thread. The deeper cost is who gets left out: analytics stays locked in the hands of the few people who can write code, while the frontline engineers and shift supervisors who actually need the insight wait in line.

1.4 The compounding effect of three problems

The three problems feed one another. Only a few experts can spot a problem. Every business question has to travel through a long IT chain before it gets an answer. And insight, once found, still needs tools outside the platform before it can become action.

The result is a long, costly conversion chain between generating data and releasing value. Enterprises pay heavily to collect, store, and manage data, yet the value they get back falls far short of the investment. This is not the failing of any one product. It is how the entire category is built.

TDengine's mission is to accelerate the path from industrial data to business value by rebuilding the industrial data platform with AI. Bring AI to Every Industrial Data Stream. When the barrier to using data becomes very low and the user experience becomes truly easy, the category can expand significantly, and the value of industrial data can reach every enterprise and every person.

Expert Scarcity

IT Dependency

Analytics Gap

CHAPTER 02

The Challenges of Rebuilding Industrial Data Platforms with AI

AI can now write code, run research, and control robots. But a factory is not a chatbot. Making AI work on the plant floor means confronting six unavoidable challenges, from fragmented data and millisecond response windows to the hard boundary of safe production.

02

CHAPTER 02

The Challenges of Rebuilding Industrial Data Platforms with AI

2026 is a turning point for AI. Reasoning models made a generational leap early in the year. Foundation models such as DeepSeek and GPT kept improving, and AI agents went mainstream. Today's AI can write code, run research, read financial reports, and control robots. More than that, it has memory, reasoning, planning, and the ability to act on its own. AI is no longer a chatbot in a chat window. It is a digital worker that can break a task into steps, coordinate across systems, and run the full loop of analyze, decide, and execute without a human watching over it.

For industry, this means the three problems from Chapter 1, expert scarcity, IT dependency, and the analytics gap, can finally be addressed at the source. AI can help capture decades of experience from veteran engineers and make that knowledge available to a wider group of engineers. It can shrink the query-analyze-model chain from weeks to minutes. It can watch every line and every machine around the clock and raise an alert the moment an anomaly starts to show. This is not a small upgrade to a traditional industrial data platform. It is a rebuild from the ground up.

Bringing AI into a factory is a different matter from having it write essays or draw pictures. The plant floor runs by its own rules. Data sits in legacy systems that are decades old. Response windows are measured in milliseconds. A mistake means lost production or, worse, a safety incident. Every industry also has its own language and its own know-how. If AI does not know what compressor surge means, or has never seen furnace condition stability, it cannot offer useful advice.

Making AI work on the plant floor means confronting six unavoidable challenges.

Six unavoidable challenges

Fragmented Data and Inconsistent Quality

Over decades, industrial enterprises have deployed DCS, SCADA, PLC, MES, ERP, LIMS, and more. Data formats differ, names are chaotic, units are not standardized, and systems are walled off from each other. The same point may be recorded under three names and three units across three systems, and the full operating record of one piece of equipment may have to be stitched together from several databases. Add sensor drift, gaps from interrupted communication, inconsistent sampling rates, and manual entry errors, and AI is not looking at a clean dataset. It is looking at isolated, non-standardized data silos.

Extreme Real-Time Requirements

Industrial AI has to work inside live, high-speed production. Equipment generates data at millisecond frequencies. An anomaly can turn into an incident within seconds, and alerts must fire in under a second. The traditional ingest-then-query-then-compute chain cannot keep up. Every extra system that data passes through adds latency and another point of failure. Data must be aggregated, detected, and alerted on while massive writes are still in flight, and sending all real-time data to AI is not feasible. This calls for stream processing built into the platform itself.

Explosive Data Scale

A single mid-size plant generates terabytes of time-series data a day. A large energy group can manage tens of millions of measurement points. Feeding all that raw data to AI is neither practical nor economical. AI needs refined, semantically enriched information, something like: this equipment's vibration has trended upward over the past 24 hours, deviating 2.3 standard deviations from its baseline. Not a hundred thousand raw vibration readings.

Process Analytics Requires a Comprehensive Approach

Real-time monitoring tells you what is happening now. Answering why it is happening, and what will happen next, needs a full spectrum of methods: metric aggregation, statistical process control, correlation analysis, machine learning, deep learning, and time-series foundation models. Different tasks need different algorithms, but the data should share one semantic model, and the results should be managed and invoked on one platform.

Deep Industry Barriers

AI's knowledge comes from public information on the internet. The most critical industrial knowledge, process manuals, operating procedures, historical fault cases, and decades of judgment from veteran engineers, is not on the internet. It is private, offline, locked in filing cabinets and in the minds of experienced engineers. AI has never seen this knowledge, so it cannot offer meaningful recommendations in industrial settings. To truly speak the language of industry, AI needs this tacit knowledge to be captured systematically. Process manuals need to become searchable knowledge bases, operating procedures need to become skills that agents can call, and historical fault cases need to become training samples.

AI Safety and Control

When AI moves from a conversational assistant to an agent that can act on its own, safety changes fundamentally. A chatbot giving a wrong answer does little harm. Industrial AI adjusting a process parameter incorrectly, or triggering an unplanned shutdown, can cause millions in losses or a safety incident. Industrial AI safety is not a product feature. It is part of compliance and corporate liability.

Bringing AI safely onto the plant floor means answering four fundamental questions:

Controllable

Where are AI's behavioral boundaries? Who can authorize its actions? Which decisions can it make on its own, and which need human confirmation?

Trustworthy

On what basis should AI's judgment be trusted? Is it reasoning from a real understanding of the plant floor, or free-associating from internet text? Can its conclusions be verified, challenged, and overturned?

Auditable

When AI makes a consequential decision, can every step of its reasoning, data sources, and trigger conditions be traced back? Without full traceability, AI is a black box, and no enterprise can bear that compliance risk.

Robust

AI can make mistakes, go down, and be attacked. When that happens, production lines must not stop, safety interlocks must not fail, and core processes must not be interrupted. AI must be a capability amplifier, not a new single point of failure.

CHAPTER 03

TDengine:
An AI-Native Industrial Data Platform

TDengine rebuilds the industrial data platform from the ground up, with AI as the foundation. Loosely coupled modules cover the full journey from data collection to intelligent decision-making, each deployable on its own, across public cloud, private cloud, and the edge.

03

CHAPTER 03

TDengine: An AI-Native Industrial Data Platform

TDengine is an AI-native industrial data platform built from multiple core functional modules that cover the full chain from data collection to intelligent decision-making. Each module can be deployed on its own or combined. The smallest deployment starts from the data storage engine. The architecture supports public cloud, private deployment, and edge-cloud hybrid models, and the complete solution covers the full lifecycle of industrial data.

What TDengine sets out to do is rebuild the industrial data platform with AI. For the end user, the ultimate goal is the same as that of industrial data historians and industrial IoT platforms: turn industrial data into business value. The difference is in the approach. TDengine makes AI the foundation and rebuilds every link of the industrial data platform from the ground up, solving the structural problems that traditional platforms cannot overcome.

Bring AI to Every Industrial Data Stream. That is TDengine's product design goal.

3.1 Capability requirements for the new-generation platform

The six challenges of industrial AI are not isolated. They weave together and interact. Solving them calls for a new-generation industrial data platform with six core capabilities.

Unified data ingestion and governance. The platform must connect to industrial protocols such as MQTT, OPC-UA, and OPC-DA, integrate with traditional data historians and relational databases, and support coordinated edge-to-cloud transmission. Data is cleaned, transformed, and standardized at the first point of entry, not patched up afterward.

High-performance time-series storage and computing. The platform must handle ingestion, compression, query, and stream computing for massive volumes of equipment data at millisecond latency. Storage and compute are separated for elastic scaling. Hot and cold data are tiered to minimize storage cost. Stream computing is built into the data storage engine, not bolted on as an external component.

Industrial ontology data modeling. The platform must build a unified semantic layer above the data storage layer, with a data catalog and standards that map raw data from database.table.field to plant.workshop.equipment.measurement_point, giving data full business context. This is the foundation that lets AI understand and reason about the plant floor.

AI agent runtime. The platform must provide AI agents with a safe, controlled, and auditable execution environment. Agent behavioral boundaries are defined by permissions and rules. Reasoning paths are traceable. Decision processes keep complete evidence chains. Core operations continue uninterrupted when AI is unavailable.

Full-spectrum process analytics. The platform must support process analytics across the full spectrum, from real-time monitoring, statistical process control (SPC), and correlation analysis to machine learning, deep learning, and time-series foundation models. Different analyses share the same semantic model, and results are managed and invoked on the same platform without exporting to external tools.

Open channels for three types of consumers. People perform flexible data analysis through zero-code interfaces. AI agents discover and invoke data through semantic paths. External systems integrate data through standard APIs and message queues. Three types of consumers, three channels, one semantic model.

3.2 Fully open, loosely coupled modular architecture

To meet these six requirements, TDengine chose a loosely coupled, independently deployable modular architecture instead of stuffing every function into a tightly coupled mega-platform.

Traditional industrial data platforms do the opposite. Collection, storage, analytics, and visualization are all bundled together. Upgrade the analytics module, and the whole platform goes down. One component underperforming? Scale the entire system. If a customer is satisfied with the collection layer but not with the visualization interface, there is still no practical way to replace only that component. Every plant is different. Some already have a mature collection system, some have built an internal analytics team, and some have special compliance needs for AI model services. A tightly coupled platform forces the enterprise to reshape itself around the vendor's prescribed architecture. Once a vendor is chosen, the enterprise is locked in.

TDengine packages each core capability as an independent module. Components communicate through standardized interfaces, and each is deployed, scaled, and upgraded on its own. They run consistently across public cloud, private cloud, and hybrid environments. The design's key advantages are flexibility and openness.

Flexibility means each component has a clear role and boundary. How data flows through collection, cleaning, storage, modeling, analytics, AI enhancement, and external services, and which nodes it passes through, is set by real user needs, not by a platform-prescribed process.

Openness means every component can be integrated with or replaced by alternatives. Customers can keep their existing collection tools instead of TDengine's ingestion module, use their preferred BI tools instead of the default visualization, and plug their own model services into TDengine's AI governance framework. Existing IT investments do not need to be torn down and rebuilt. They can be embedded directly into TDengine's data processing ecosystem, so customers are never locked into a vendor.

3.3 TDengine architecture overview

The components below form TDengine's product system. They work together in a loosely coupled way, each with its own role and clear responsibilities. Together they form a complete ecosystem for industrial data management and application.

Data aggregation and edge-cloud collaboration. Industrial data is scattered across dozens of systems and formats: PLC, DCS, SCADA, traditional data historians, relational databases, and message queues. taosX is built to unify this data, completing cleaning, transformation, and standardization at the first point of entry. With support for a wide range of industrial protocols and data sources, built-in ETL, and an edge-to-cloud collaborative architecture, it addresses the problems of fragmented data and inconsistent quality. Data ingestion work that once took weeks of ETL development can now be done in hours through graphical configuration.

High-performance time-series data storage and computing engine. Industrial data has high write frequency, large volume, and distinct hot and cold patterns. General-purpose databases cannot handle this load. The TSDB Engine handles ingestion, storage, query, and real-time computing for massive time-series data within a single engine, rather than relying on a patchwork of external components. It addresses the challenges of extreme real-time requirements and explosive data scale, and provides a solid foundation for downstream analytics. The engine is built from five types of independently deployable nodes. Dnode is the runtime carrier. Vnode is the basic storage unit for data shards. Mnode handles cluster metadata management and load balancing. Qnode supports storage-compute separation and cross-database parallel query processing. Snode is the built-in stream processing node, where real-time computing tasks are defined in SQL.

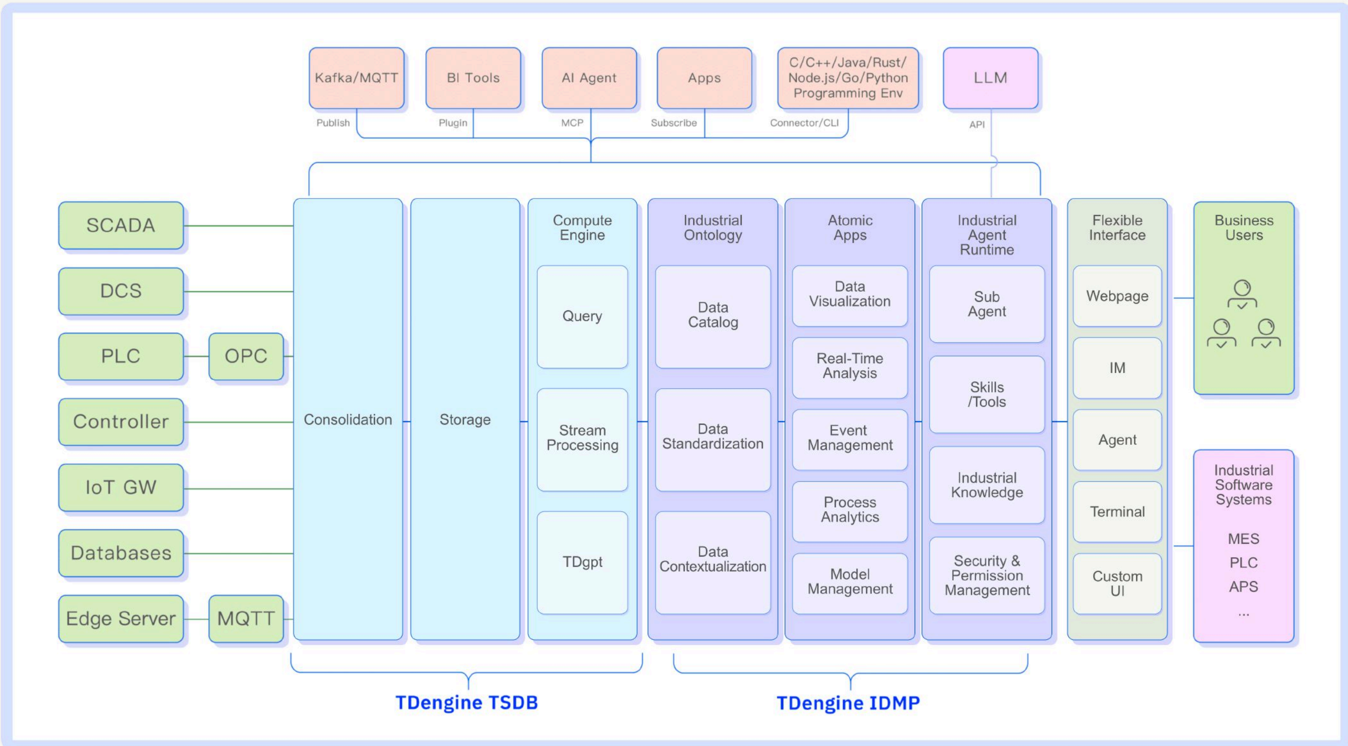
Industrial Ontology. The Industrial Ontology builds a unified business information layer between the data storage layer and the data application layer: an industrial mirror and information network rooted in the real world, built for operational analysis and decision-making, and operable by both AI and people. It plays three roles at once. First, a unified data portal that serves people, AI, and external systems through one semantic model. Second, a comprehensive data catalog that maps raw data scattered across systems into assets with business meaning. Third, a relationship network that carries business semantics, organizing asset hierarchies with a tree structure and lateral relationships with a graph structure, so AI reads the plant floor along meaningful business paths instead of meaningless database and table names. These three roles converge on one purpose: giving AI reasoning a trustworthy foundation, giving security governance a firm anchor, and giving the tacit knowledge enterprises accumulate a platform where it can be captured and passed on.

Industrial Agent Runtime (IAR). IAR is TDengine's enterprise-grade runtime for AI agents, and it solves two problems. First, industry knowledge capture: enterprises can upload private documents such as process manuals, operating procedures, and fault cases, publish skills, and let AI agents work within an industry context. Second, AI safety and control: agent sandbox isolation, identity-inheriting permissions, dual-layer guardrails for prompts and tool calls, task safety classification, and complete audit trails keep every step of AI behavior controllable, traceable, and auditable. An agent is not a privileged role. It inherits the user's permissions and operates within rule constraints.

Industrial data analytics engine. TDgpt is the analytics engine built into the TSDB, deployed close to the data storage layer. It handles the in-database execution of analytics algorithms, so a single SQL statement calls forecasting, anomaly detection, data imputation, and more. Data never leaves the database, and results return in milliseconds. TDmodel, the model management module in the frontend application, handles the model lifecycle at the application layer: data selection, algorithm configuration, model training and evaluation, and one-click deployment to TDgpt, all within one platform. Together, TDgpt and TDmodel form TDengine's industrial data analytics capability, covering the full spectrum from statistical models and machine learning to deep learning and time-series foundation models, so process analytics no longer depends on external tools.

Industrial applications for business users. TDengine ships with default built-in frontend applications. Business users get a completely zero-code interface, where data processing, visual analysis, real-time analysis and monitoring, event batch comparison, panel interpretation, and root cause analysis all happen in a browser. Industrial applications are fully decoupled from the backend data layer, communicating with backend services through standard APIs and MCP, and they can be deployed and scaled horizontally on their own. Users can build personalized, flexible interfaces on top of TDengine's atomic applications and open capabilities, or embed specific TDengine pages into other industrial systems.

TDengine Platform Architecture



These are TDengine's core functional components. Beyond them, two foundational capabilities run through the entire platform, permeating every component, and together they form TDengine's enterprise-grade quality.

Platform-wide security. Platform-wide security covers three layers. System security: RBAC access control, element-level fine-grained access, SSO, encryption in transit and at rest (SM4), backup and disaster recovery, IP whitelisting, and complete audit trails establish the platform's security baseline. AI security: agent sandbox isolation, identity-inheriting permissions, dual-layer guardrails, ontology-enforced constraints, and output cross-validation keep every step of AI behavior under control. Enterprise security policy: human-in-the-loop workflows, business safety interlocks, system-wide Git version management, and e-signature approval for sensitive operations extend security from the technical layer to the governance layer.

Open data services. Open data services define how data assets are consumed and invoked. TDengine sees three types of future industrial data consumers: people, AI agents, and external systems. Each needs a different channel but shares the same semantic model. For people, TDengine provides a zero-code web interface. For AI, native MCP interfaces expose elements, attributes, events, analysis results, and panels directly to external AI agents, so agents need no local server deployment and can read context and run controlled operations through a remote connection. The CLI gives terminals, scripts, and agents an auditable command entry point for automated troubleshooting and batch tasks. For external industrial systems, REST APIs, multi-language SDKs, JDBC/ODBC, and Kafka/MQTT subscriptions form a complete data integration channel. Multiple channels, one semantic model, and one consistent definition of data.

All TDengine components are fully loosely coupled and independently deployable and scalable. Customers can use third-party collection tools for ingestion, their familiar BI tools for analytics, and their own model services for AI. Every component can be integrated or replaced, so no lock-in is required.

3.4 TDengine cloud services

For TDengine, deployment should not be a single choice. Enterprises of different sizes, industries, and compliance requirements have very different needs. Startups want zero-ops rapid launch. Large groups need on-premises deployment with data that never leaves the plant. Multi-plant enterprises want edge autonomy with centralized cloud control. TDengine provides three deployment models: fully managed public cloud, private deployment, and edge-cloud hybrid, all on one unified AI-native architecture with consistent functionality.

Fully managed public cloud. TDengine Cloud is a fully managed time-series data cloud service for all industries, available on four major public cloud platforms: AWS, Microsoft Azure, Google Cloud, and Alibaba Cloud. Users activate it with one click through each platform's Marketplace and get a complete, production-ready industrial data platform instance within minutes.

TDengine Cloud is elastic and serverless: storage and compute are separated, and clusters scale automatically with workload. Users do not manage deployment, scaling, backups, or optimization, and instead focus entirely on turning data into business value. The service offers a 99.9% SLA and professional support, with scheduled automatic backups and disaster recovery capabilities built in. For enterprises with higher data security needs, VPC peering lets TDengine Cloud deploy database instances inside the customer's dedicated virtual private cloud for network isolation and compliance.

In fully managed mode, every TDengine AI-native module is available. Users get a complete industrial data platform, not just a managed database. Built-in data caching, data subscription, and stream processing replace the multiple external components (Redis, Kafka, Flink) a traditional architecture would require, cutting system complexity and total cost of ownership. Through MCP and REST API, cloud instances also serve as the data foundation for AI agents, delivering safe, controlled semantic data services to any AI application on the public cloud.

Private deployment. TDengine Enterprise is the complete private deployment for enterprises with strict compliance requirements, existing data center investments, or a need to run inside physically isolated networks. It scales elastically from a single server to clusters of hundreds of nodes, with full compatibility with Kylin, UOS, and other operating systems as well as various CPU architectures. Private deployments include every enterprise-grade capability: tiered storage, multi-replica high availability, active-active disaster recovery, SM4 encryption, and RBAC access control. Data can remain inside the enterprise's own data center, helping meet strict compliance requirements in sectors such as energy, power, and defense.

Edge-cloud hybrid deployment. Large group enterprises often run multiple geographically dispersed production bases. TDengine supports a multi-tier edge-cloud architecture: edge-side collection and real-time processing nodes give millisecond local response and offline autonomy; plant-side aggregation nodes handle regional data integration, process analytics, and local decisions; group-level cloud deployment of the full platform provides cross-site global analytics and AI model training. Built-in WAL log subscription and data replication push edge-to-cloud synchronization to millions of data points per second, with flexible distribution options like downsampling, conditional filtering, and selective synchronization. During network interruptions, edge nodes can run independently with local data caching and automatic backfill on reconnection, reducing the risk of data loss. This balances the plant floor's extreme real-time and reliability demands with the cloud's strengths in global aggregation and AI analysis: edge autonomy, cloud centralization, and collaborative intelligence.

TDengine's three deployment models share the same core architecture and capabilities. As the business evolves, users can migrate and combine models freely: validate quickly in the cloud, scale out with private deployment, then cover the whole group with edge-cloud collaboration. Capabilities stay consistent and the experience uniform. TDengine Cloud is not a separate product line. It is the same architecture practiced across different infrastructures.

CHAPTER 04

Data Aggregation and Edge-to-Cloud Collaboration

Plant-floor data is scattered across dozens of systems, protocols, and formats, each isolated from the rest. taosX is TDengine's data aggregation and ingestion module, and it ends this fragmentation at the first point of entry, cleaning data before it enters storage.

04

CHAPTER 04

Data Aggregation and Edge-to-Cloud Collaboration

Data on the plant floor is scattered across dozens of systems, protocols, and formats, each isolated from the rest. taosX is TDengine's data aggregation and ingestion module, and its goal is to end this fragmentation at the first point of entry, cleaning data before it enters storage.

taosX is made up of taosX Server and taosX-Agent, the edge agent. The design is simple: users complete industrial data ingestion, cleaning, and transformation through graphical configuration, without writing a line of code. Where a traditional ETL cycle takes two to four weeks, taosX compresses the process to hours.

taosX supports a wide range of industrial data sources: modern industrial protocols such as MQTT, OPC-UA, and OPC-DA, with direct connection to OPC Servers for node subscription and automatic parsing of formats like JSON; traditional data historians such as PI System and AVEVA Historian, with bulk historical migration and real-time synchronization; relational databases such as MySQL, Oracle, SQL Server, and PostgreSQL; message queues such as Apache Kafka and FlashMQ; and batch import of file formats such as CSV and Parquet. taosX also supports data subscription from other TDengine instances for cascaded architectures.

taosX-Agent runs at the edge, near the data source. It collects data from industrial equipment and forwards it over secure channels to the central taosX Server. This edge-cloud architecture addresses common plant-floor network constraints: breakpoint recovery, local caching during network interruptions with automatic backfill after recovery, firewall traversal and DMZ-based one-way upload, and agent-side filtering and preprocessing with store-and-forward to reduce data loss risk and central load.

taosX includes complete built-in ETL. For parsing, JSON Path or regular expressions extract fields from raw messages, and split or regex break one raw field into several. For filtering, messages are written to TDengine only when an expression evaluates to true. For transformation, parsed fields map to TDengine supertable fields, with field mapping, unit conversion, expression transformation, and data type unification. Data is governed at the first point of entry. Users configure everything through the taosExplorer web interface without code, and once configured, the full pipeline from collection to storage runs automatically.

CHAPTER 05

High-Performance Data Storage and Computing Engine

Industrial data has high write frequency, large volume, and distinct hot and cold patterns. General-purpose databases hit I/O bottlenecks under this load. TDengine redesigned its storage and computing engine, the TSDB Engine, from the storage model up, tailored to industrial data.

05

TDengine

CHAPTER 05

High-Performance Data Storage and Computing Engine

Industrial data has high write frequency, large volume, and distinct hot and cold patterns. General-purpose databases facing millions of data points per second quickly hit I/O bottlenecks, and storage costs are just as daunting. TDengine redesigned its storage and computing engine, the TSDB Engine, from the storage model up, tailored to industrial data. In representative industrial workloads, write performance can improve by over 10x, and storage footprint can drop to one-tenth of traditional approaches.

The TSDB Engine is composed of five node types working together: Dnode is the runtime carrier; Vnode handles data storage; Mnode handles cluster management; Qnode handles query computing; and Snode handles stream processing.

Dnode (Data Node) is the runtime instance of taosd on a physical node and the runtime carrier for all logical nodes. A physical machine, virtual machine, or container runs one Dnode. The system scales horizontally by adding Dnodes. A Dnode contains zero or more Vnodes, and at most one Mnode, one Qnode, and one Snode.

Vnode (Virtual Node) is the basic logical unit of data storage. Data collection points are assigned to different Vnodes via consistent hashing. Each Vnode has independent runtime threads, memory space, and persistent storage paths, and is responsible for the complete data of a batch of collection points, including time-series data and tag metadata. Data within a Vnode is partitioned by time period. Each Vnode corresponds on disk to a data file (actual data blocks), a head file (block index information), and an stt file (trailing data), forming a dual mechanism of sharding by collection point plus partitioning by time. For high availability, each Vnode can be configured with three replicas distributed across different Dnodes, with the Raft protocol ensuring data consistency. When a Dnode fails, the replicas of its Vnodes automatically switch over while preserving data consistency. Performance remains stable at the scale of 1 billion time series and 100 data nodes, addressing the high-cardinality problem that has long challenged time-series systems.

Mnode (Management Node) is the management hub of the cluster, responsible for storing and managing all metadata (users, databases, supertable schemas, and Vnode distribution) and continuously monitoring the status of all Dnodes while implementing load balancing. A cluster allows at most three Mnodes, which automatically form a Raft high-availability group. Each Dnode runs at most one Mnode.

Qnode (Query Node) implements storage-compute separation. Qnodes are not bound to specific databases and can process queries across databases in parallel. They support rapid startup, teardown, and elastic scaling in container form on cloud platforms. Combined with the supertable and virtual table design, cross-device aggregation queries can be completed without JOINs. Based on TSBS benchmark testing, some complex queries outperform InfluxDB by more than 400x.

Supporting the high performance of these nodes is TDengine's unique data storage model:

One Table per Data Collection Point	Each sensor gets its own table. Data is stored contiguously in blocks, sorted by time, with precomputation and block indexing. Data from all tables within the same time period is stored in the same data file, organized internally in columnar storage: data of the same column is stored together, and different data types use different compression algorithms to improve compression ratio. The benefits are direct: reads dramatically reduce random access; writes involve only a single writer per table with no locking needed; and append-only writes are extremely fast. Write performance improves by over 10x compared to general-purpose databases.
Supertable	A supertable lets collection points of the same type share a data schema. Metric columns define the collected data; tag columns define static attributes. Aggregate queries simply query the supertable with tag filtering, no JOIN needed. Tags are stored separately from time-series data, dramatically reducing scan volume. Millions of tables can be managed and queried through a single supertable.
Virtual Table	Virtual tables align and merge columns from different tables by timestamp, dynamically generating a logical view. This makes write-first, model-later a reality. Data is written in its most primitive form, and business perspectives are flexibly constructed afterward based on analytical needs.

Beyond these, the TSDB Engine offers two additional major advantages: extreme storage efficiency and full-stack time-series data processing.

Extreme storage efficiency: the TSDB Engine uses columnar storage with two-stage compression. The first stage applies encoding techniques such as Simple8B and Delta-of-Delta. The second stage selects the optimal general-purpose compression algorithm. Since data within the same collection point typically changes minimally, compression ratios are extremely high. In most scenarios, storage cost drops to one-tenth of traditional solutions. Multi-level storage automatically tiers data by hot and cold characteristics, from memory to SSD, HDD, to S3 object storage. The entire process is transparent to applications with no manual archiving required, significantly reducing enterprise data storage costs.

Full-stack time-series data processing: the TSDB Engine also builds caching, stream processing, and data subscription into the database. Each node provides in-memory caching of the latest data for each collection point using a FIFO strategy, ensuring millisecond reads of the latest state without deploying external caching systems like Redis. It also provides data subscription services externally. In traditional architectures, industrial data processing requires chaining together MQTT Broker, Kafka, Redis, Flink, relational databases, and other external components, creating high system complexity, maintenance costs, and failure points. TDengine builds all these capabilities in. One platform replaces the entire technology stack, significantly reducing system complexity and total cost of ownership.

CHAPTER 06

Stream Processing and Intelligent Analytics

Industrial scenarios generate massive real-time computing demands with millisecond response requirements. TDengine builds stream processing and a time-series analytics engine into the platform, sharing the same data storage to complete the full chain from real-time detection to intelligent analysis to result writeback.

06

CHAPTER 06

Stream Processing and Intelligent Analytics

Industrial scenarios generate massive real-time computing demands with millisecond response requirements: data downsampling, scheduled reporting, precomputation acceleration, anomaly detection, low-latency alerting, and more. Traditional solutions require deploying additional stream processing components such as Kafka and Flink, which brings complex architecture and high maintenance costs. The TDengine platform has stream processing and a time-series analytics engine built in. The two share the same data storage and work together to complete the full chain of real-time detection, intelligent analysis, and result writeback.

6.1 TDengine stream processing engine

TDengine's built-in stream processing engine defines stream computing tasks in SQL. Its core characteristic is trigger-compute decoupling: the trigger source and the computation target can be different tables, so computation can run on the trigger table or on other databases and tables. The engine supports multiple trigger types: event windows, state windows, session windows, sliding windows, count windows, periodic windows, and statistical process analysis. It also supports pre-filtering data before an event triggers. Under high-throughput writes, computation results return in milliseconds, with flexible balancing between result timeliness and resource load.

TDengine hides stream processing complexity behind SQL. Anyone who can write SQL can use stream processing, with no new syntax to learn and no Kafka or Flink clusters to deploy and maintain. In many industrial scenarios, it can reduce or eliminate the need for separate Kafka, Flink, or similar stream processing components.

6.2 TDengine time-series analytics engine

TDengine's built-in time-series analytics engine, TDgpt, provides forecasting, anomaly detection, data imputation, and classification. All of it is invoked through a single SQL statement, with no need to export data to Python or deploy a separate model service.

TDgpt consists of four components. The built-in analytics library integrates mature statistical algorithms such as statsmodels, pyculiarity, and pmdarima, covering classic scenarios such as ARIMA and ADFuller testing. The built-in machine learning library includes pre-integrated deep learning frameworks such as PyTorch, Keras, and scikit-learn, supporting more complex time-series modeling. The general-purpose LLM adapter connects to models such as DeepSeek and LLaMA through an OpenAI-compatible interface, letting LLM reasoning act directly on industrial data. The time-series model adapter connects to dedicated time-series models such as Time-MoE and TDtsfm, TDengine's self-developed time-series foundation model, solving complex pattern recognition problems that traditional statistical algorithms struggle with.

TDgpt's open SDK lets developers integrate their own algorithms or models into TDengine. Once integrated, they are invoked with a single SQL statement too. Whether it is a team's proprietary fault prediction algorithm or a deep learning model fine-tuned for specific equipment, everything converges to the single entry point of SQL. TDgpt uses a stateless design and can be deployed on demand with independent scaling and GPU acceleration support.

TDengine's stream processing engine and TDgpt share the same data storage model and computing engine. Real-time alerts from stream processing can immediately trigger complex analysis in TDgpt, and TDgpt's forecast results can be written back to storage through stream processing for later analysis. This keeps the path from data to insight within a millisecond-level response loop.

TDengine

CHAPTER 07

Industrial Ontology

The foundation of industrial intelligence is not algorithms. It is the data model. The Industrial Ontology builds a unified semantic layer between data storage and data application, a network that AI, applications, and people can all understand, access, and use.

07

CHAPTER 07

Industrial Ontology

Industrial enterprises are not short on data. Decades of DCS, SCADA, MES, LIMS, EAM, and ERP deployments have created vast amounts of it. The real gap is a systematic way to manage the knowledge behind that data: how equipment is connected, what each measurement means, which operating context matters, and which past events are relevant.

TDengine's Industrial Ontology addresses this gap by building a unified semantic layer between data storage and data applications. It gives each industrial asset a clear identity, makes relationships explicit, and attaches business meaning to operational data, so AI agents, applications, and people can work from the same understanding.

7.1 What industrial data modeling is

The foundation of industrial intelligence is not algorithms. It is the data model. In the real world of factories, enterprise, plant, production line, equipment, and sensor form a nested organizational structure, each layer carrying specific business meaning. The Industrial Ontology builds a unified semantic layer between the data storage layer and the data application layer. This layer does not store raw data itself. On top of raw data, it builds a semantic network that AI, applications, and people can all understand, access, and use.

The Industrial Ontology is a unified data portal for AI, applications, and people. On one side it connects to data sources; on the other, it connects to three types of data consumers: people view through panels and ask questions through AI; systems connect through REST API, JDBC/ODBC, and Kafka/MQTT; and AI agents invoke directly through MCP. Three types of consumers share a single semantic model. Different interfaces, the same data consistency. This approach aligns closely with the Data Fabric concept discussed in the industry for years: a metadata-driven unified semantic layer that frees data from physical location constraints, organizing, governing, and consuming data as data assets and data services.

The Industrial Ontology is a global data map and catalog. Industrial data sources are extremely diverse: time-series databases, relational databases, MES, WMS, ERP, and file systems in various formats. One goal of industrial data modeling is to map raw data scattered everywhere onto one or more asset catalog trees composed of elements and attributes. To upper-layer applications, this layer hides all the complexity of the underlying data. There is no need to know which system, which database, or which table the data resides in.

The Industrial Ontology is a data association network for industrial data. The real difficulty of industrial data modeling has never been getting the data. It is managing the relationships between data. Which production line did a piece of equipment come from? What does it feed downstream? Who is responsible for it? Which business metrics does it affect? What events has it experienced? These connections are scattered in the minds of business staff, in the wiring diagrams of SCADA and MES, and in process manuals and Excel sheets. As long as they stay scattered, no matter how much underlying data is collected, it remains a collection of isolated islands. People cannot find the full context. Applications cannot align their understanding. AI has no reliable path for reasoning. The task of the Industrial Ontology is to turn these connections from tacit knowledge into explicit objects, continuously maintained in a unified, governable, queryable, and AI-traversable form.

The Industrial Ontology built by TDengine contains multiple layers of information architecture.

The bottom of the Industrial Ontology is the data source and storage layer. The Industrial Ontology itself does not store raw data. In industrial scenarios, the raw monitoring signals continuously generated by SCADA, DCS, OPC, and other industrial systems are uniformly collected and stored by TDengine's high-performance TSDB time-series engine.

Above this is the core of the Industrial Ontology: the data model layer. TDengine fully defines the information structure of digital objects in the industrial world across four dimensions.

Industrial objects: the objects that need to be managed and accessed in industrial scenarios, including elements, attributes, real-time analyses, events, metrics, panels, analysis models, and business rules. In TDengine's Industrial Ontology, every physical or logical asset in the industrial environment (a plant, a production line, a piece of equipment, or a sensor) is represented as an element. Elements are the fundamental building blocks of the Industrial Ontology. Elements contain attributes. Attribute changes trigger real-time analyses, which in turn generate events. Elements have statistical metrics and visualization panels, and are associated with business rules and analysis models.

Object relationships: the reference relationships between industrial objects. Elements have multiple types of reference relationships: strong references, weak references, and composite references. Attributes attach to elements or events. Attributes have reference calculations and limit-value relationships between each other. Analyses and events connect to elements and attributes through references, and link the monitoring-to-alert chain through triggers and generations. Templates and instances connect through derivation, enabling define once, apply everywhere. Beyond reference relationships, elements also have production process relationships, control redundancy relationships, and impact propagation relationships, such as upstream/downstream relationships in production flows and interlock relationships.

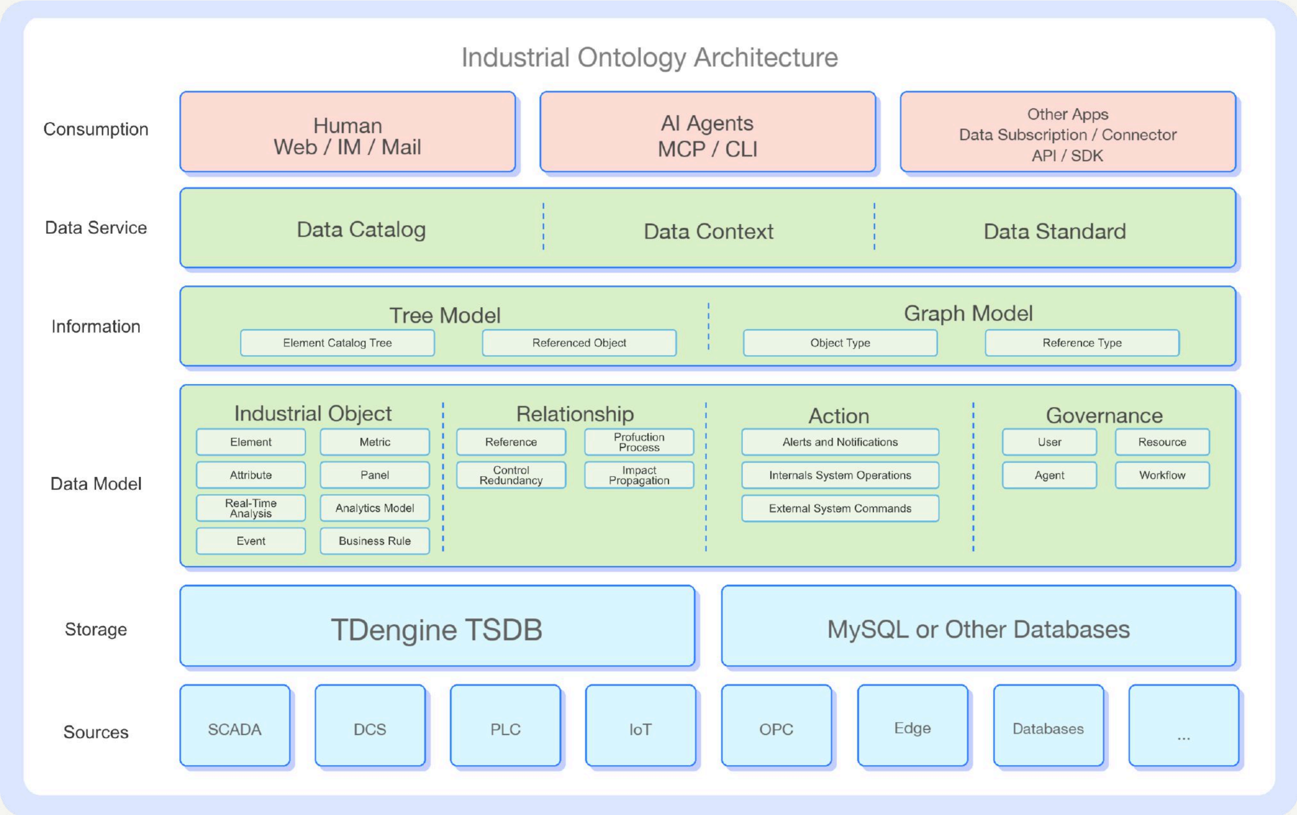
Decision actions: the actions and tasks that elements and events can trigger, such as alerts and notifications, internal operations, and external commands. TDengine converges all execution capabilities that can change reality state into action templates uniformly managed by the system.

Access control: the permission management of resources within the Industrial Ontology. Different users and agents have different security-level access and management permissions for different resources. Clear process loop definitions exist for specific sensitive operations, such as human-in-the-loop.

Above the data model layer is the information organization layer. TDengine uses a tree model to build the asset hierarchy skeleton and a graph model to complete cross-level lateral relationships. The tree model and graph model together form the complete Industrial Ontology.

The data service layer transforms the Industrial Ontology's information into three types of externally available data services. The data catalog hides the complexity of underlying data, making data discoverable, searchable, and governable. Data contextualization gives each value business semantics, such as unit, range, ownership, and operating condition, making data understandable. Data standardization automatically transforms multi-source data, ensuring global consistency of data standards and statistical measures.

These data services are provided to external consumers through the information consumption layer: people access data through zero-code web interfaces and natural language queries; AI agents traverse the Industrial Ontology and invoke platform capabilities through MCP; and external systems and applications integrate data through REST API, SDK, and message queues. Three types of consumers, multiple channels, sharing a single semantic model.



7.2 Industrial objects and information structure

All entities on the plant floor are abstracted by the Industrial Ontology into four core object types:

Object Type	Meaning	Real-World Counterpart
Element	Basic unit of the asset model	Group, plant, workshop, production line, equipment, sensor, common point
Attribute	Measurement dimension or descriptive characteristic of an element	Temperature, pressure, flow, current, operating status, KPI
RT Analysis	Real-time computation logic referencing elements and attributes	SPC monitoring, energy consumption comparison, root cause analysis
Event	Operational record with start/end time and severity level	Compressor surge event, batch start/stop, shutdown event, equipment fault

Elements and events are independent objects. Attributes and real-time analyses are dependent objects: they must attach to an element or event to exist. Consumption objects such as visualization panels, dashboards, annotations, and associated documents present information about these core objects from different perspectives. For each object type, TDengine provides templates for batch definition of reusable information structures: define once, apply everywhere.

Around these four object types, the Industrial Ontology organizes their relationships using a two-layer tree-and-network information structure:

Tree structure carries the most intuitive hierarchical relationships. Using elements as the basic unit, it organizes them from group, plant, workshop, production line, process section, equipment, measurement point into a clear asset catalog tree. TDengine supports flexibly building multiple tree structures from different business perspectives, either by organizational structure or by equipment type classification. Each layer can carry attributes, analyses, panels, and events.

Network structure fills in non-hierarchical relationships that the tree cannot cover above the tree layer. Through the Reference system, it uniformly expresses directed connections between objects: upstream/downstream material dependencies, shared utility points (air, water, electricity, gas), cross-line borrowing, references between analyses and events, and derivations and dynamic limit-value associations between attributes. An enumeration sets mechanism provides preset reference types for different object types, so users can select them directly in the interface or create and edit new relationship types as their business scenarios require.

The tree tells AI what the object is and where it belongs. The network tells AI what it relates to, what it affects, and what affects it. Together, they form the complete Industrial Ontology: a digital mirror of the real industrial world in digital space.

7.3 Data contextualization and standardization

A column named current in a database table is just a number. Only by knowing which meter produced it, where it is installed, what the unit is, and what the normal range is does it become meaningful. The Industrial Ontology transforms raw measurement values into AI-ready industrial data assets through two complementary mechanisms.

Data contextualization gives data business meaning. At the element level: name, position in the tree, template category, description, GPS coordinates, manufacturer and model, and other static metadata, along with associated engineering documents and P&ID drawings, tell AI what this asset is. At the attribute level: physical unit, upper and lower limits, target value, display precision, and data references, tell AI what this measurement means. At the event level: event type, severity level, start and end time, trigger conditions, and associated snapshots, tell AI what happened and how urgent it is. The AI engine indexes this contextual information to provide more accurate answers in Q&A and analysis.

Data standardization unifies data across systems. Across different systems, the same kind of data often has inconsistent naming and units. Some record temperature as WD, others as temp. Some collect in Fahrenheit, others in Celsius. Templates predefine standardized field structures and apply them in batch to instances. The physical unit conversion engine automatically performs transformations. Data references map data from different sources and different table structures to unified business attributes. Without moving data or writing code, unified modeling of multi-source heterogeneous data is achieved, ensuring comparability and consistency of data across systems and time periods.

7.4 Model once, benefit everywhere

The Industrial Ontology is not a one-time project deliverable. It is a continuously appreciating data asset infrastructure.

For visualization, it means panels are no longer blank canvases. When a user navigates to a piece of equipment, they see a complete, context-rich view of that equipment rather than starting configuration from scratch. For event monitoring, it means alerts are no longer isolated notifications. Each event knows which equipment and production line it belongs to, so operators can immediately locate the scope of impact. For AI, it is the essential reasoning foundation. AI does not face database.table.field. It understands data along the path of plant, workshop, equipment, measurement point. Only contextualized and standardized data can become truly AI-ready data assets. For team collaboration, it makes data assets discoverable, shareable, and governable. New employees no longer need veteran staff to verbally explain what this measurement point represents. The asset catalog is the living documentation.

The Industrial Ontology provides a solution path for the structural problems of traditional industrial data platforms. By mapping heterogeneous data sources to a consistent semantic view through a unified asset catalog, it addresses the problem of fragmented data that is difficult to aggregate. By abstracting raw signals into information assets with business context, it allows AI to understand and consume data. By providing a tree-and-network information structure, it gives AI reasoning paths that can be followed, traversed, and verified, lowering industry barriers and giving AI the background knowledge to understand the plant floor. The Industrial Ontology does not solve any single specific problem. But it provides a common semantic foundation for other components: stream processing, AI agents, process analytics, and model training. Without this foundation, no matter how powerful AI becomes, it cannot be applied effectively on the plant floor.

CHAPTER 08

Industrial Agent Runtime

Bringing AI into industrial scenarios is not as simple as connecting an API. A chatbot giving a wrong answer is harmless. Industrial AI getting a process parameter wrong can mean a production line stoppage or a safety incident. TDengine builds an Industrial Agent Runtime where AI can be deployed safely and operate in a trusted, governed way.

08

CHAPTER 08

Industrial Agent Runtime

Adding a chat window to industrial software is not the same as adopting AI. A model can answer questions, but by itself it cannot understand industrial data, call system tools, or operate within clear safety boundaries. Without those capabilities, AI remains a conversation layer and never reaches the production workflow.

TDengine's Industrial Agent Runtime (IAR) is built to close that gap. It sits above the LLM, connects to mainstream models, and gives industrial AI the runtime infrastructure it needs to work in production: a Knowledge Base that carries enterprise process context, Skills that turn expert procedures into reusable workflows, and governance controls that keep every action controlled, traceable, and auditable.

8.1 Why industrial AI needs a runtime

Bringing AI into industrial scenarios is not as simple as connecting an API. A chatbot giving a wrong answer is harmless. Industrial AI getting a process parameter wrong or misjudging equipment status can mean a production line stoppage or a safety incident.

TDengine does not build large language models, nor does it provide a built-in LLM service. What TDengine builds is a built-in Industrial Agent Runtime (IAR). The core purpose of this module is to give industrial users an enterprise-grade environment where AI can be deployed safely and operate in a trusted, governed way. It uniformly connects to and manages mainstream LLMs, providing AI agents with the complete infrastructure needed for operation: LLM connectivity and scheduling, system-enforced context, skills orchestration and execution, output validation, review, and auditing.

The TDengine IAR architecture is built from the bottom up: LLM integration, agent framework, agent capability extension, AI applications, and security governance.

At the bottom layer, IAR provides open LLM configuration management. TDengine supports connecting to mainstream LLMs, including cloud-hosted models and locally deployed private models. Different models can be configured for different tasks to achieve optimal model-task matching. Prompt management is also provided.

TDengine IAR is a fully functional AI agent runtime, including agent execution, cognitive, and capability extension frameworks.

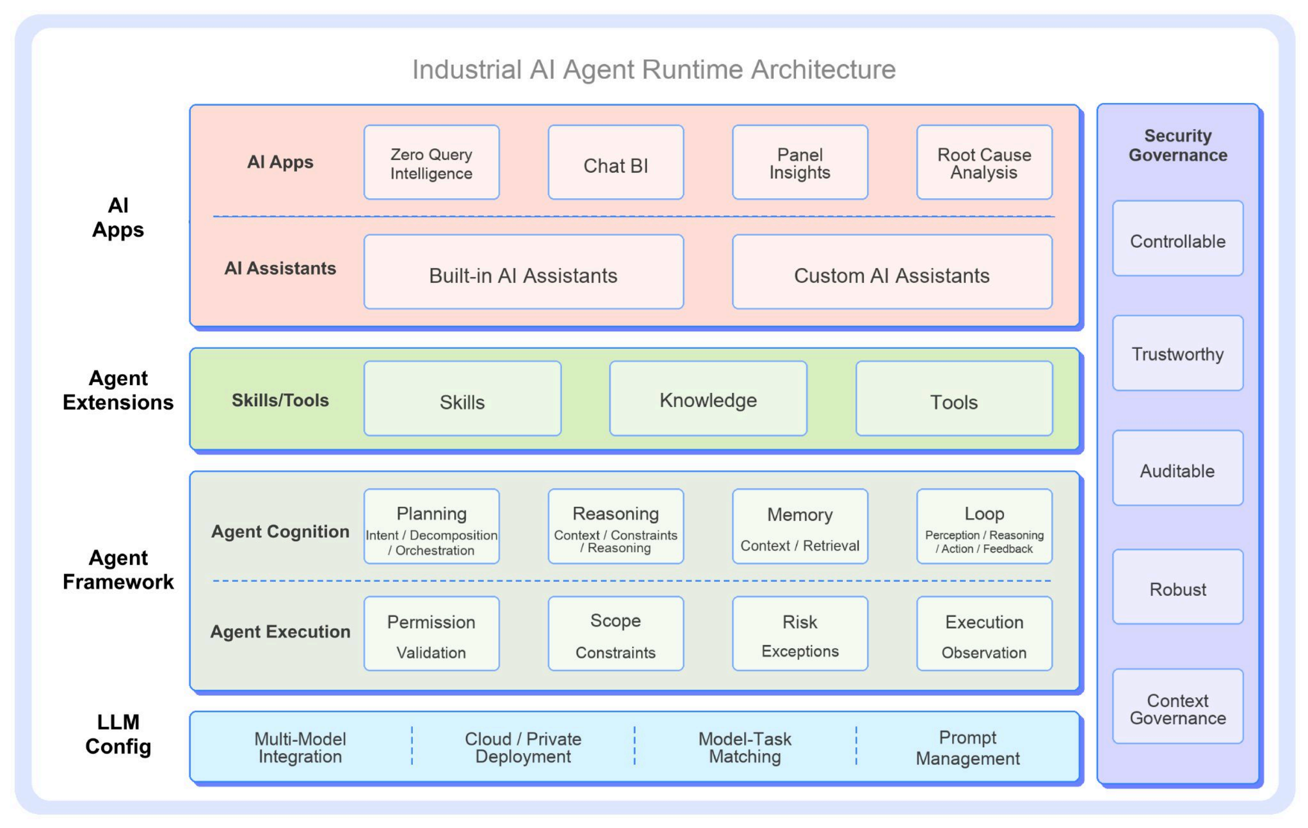
Agent execution framework: IAR includes a built-in agent execution and observation framework, supporting complete caller permission management where agent permissions are fully equivalent to the initiating user's system permissions. It supports scope-of-work control with explicit limits on agent-executable tasks and content. It also supports comprehensive risk management strategies, including circuit breaker, rate limiting, and retry mechanisms. TDengine IAR supports the standard ACP protocol, meaning users can not only use TDengine's built-in agents but also introduce external agents at any time, building an open agent ecosystem.

Agent cognitive framework: built on top of the execution and observation framework, IAR establishes an agent cognitive framework adapted to industrial scenarios, including Planning (intent recognition, task planning and orchestration, task routing), Reasoning (context awareness, inference, constraints), Memory (short-term, medium/long-term memory, summary compression and retrieval), and Agent Loop (the agent's perceive, reason, act, feedback cycle), enabling agents to execute complex tasks more intelligently with LLM support.

Agent extension framework: TDengine IAR includes Skills, Knowledge Management, and Tools as agent capability extension modules, significantly enhancing agents' ability to execute complex tasks and making agents more knowledgeable about industrial scenarios.

On top of the AI agent runtime, TDengine pre-installs dozens of system-level skills, builds 13 out-of-the-box AI assistants, and supports users in customizing personalized agent assistants based on the agent capability extension framework. At the application layer, TDengine has also built typical AI scenario applications around the most common industrial data analysis use cases: Zero Query Intelligence, Chat BI, panel interpretation, and root cause analysis.

Running through all the above components, TDengine IAR has built a complete enterprise AI security governance system covering four dimensions: controllable, trustworthy, auditable, and robust, plus context governance. In industrial scenarios, AI safety is not an optional feature. It directly determines whether production lines can run, whether incidents can be avoided, and whether compliance audits can be passed.


Industrial AI Agent Runtime Architecture
AI Apps
AI Apps
Zero Query Intelligence
Chat BI
Panel Insights
Root Cause Analysis
AI Assistants
Built-in AI Assistants
Custom AI Assistants
Agent Extensions
Skills/Tools
Skills
Knowledge
Tools
Agent Framework
Agent Cognition
Planning
Intent / Decomposition / Orchestration
Reasoning
Context / Constraints / Reasoning
Memory
Context / Retrieval
Loop
Perception / Reasoning / Action / Feedback
Agent Execution
Permission
Validation
Scope
Constraints
Risk
Exceptions
Execution
Observation
LLM Config
Multi-Model Integration
Cloud / Private Deployment
Model-Task Matching
Prompt Management
Security Governance
Controllable
Trustworthy
Auditable
Robust
Context Governance

8.2 AI Functions: thirteen out-of-the-box AI assistants

TDengine IAR defines workflows through Skills to execute different tasks. AI capabilities built on Skills are called AI Functions. Each AI Function corresponds to one or more Skills.

TDengine IAR provides 13 out-of-the-box AI assistants (AI Functions) through pre-installed System Skills. Each assistant focuses on a specific industrial data task domain with complete context for that domain, dedicated tools, and validation logic. They cover the complete work chain from data query, visualization, stream processing, and alert management, to anomaly detection, trend forecasting, root cause analysis, and health assessment, to event retrieval, comparative analysis, report generation, and document Q&A. Users describe their needs in natural language, and AI helps complete the backend technical steps, from finding the right equipment and measurement points to constructing queries, configuring analysis rules, and generating visualizations. Anyone can talk to data directly. No waiting for IT. No writing code.

AI Assistant	Capability Description
Data Query	Ask industrial data questions in natural language. No need to know table names or field names. Get answers within seconds.
Panel Creation	Describe requirements in natural language. AI automatically generates trend charts, bar charts, pie charts, and other visualization monitoring panels.
Analysis Tasks	Configure stream processing tasks in natural language. No need to understand trigger types and window parameters. The system completes the technical implementation automatically.
Alert Configuration	Describe risk scenarios in natural language. AI automatically generates alert rules, letting the people who know the business best define alert logic directly.
Metric Calculation	Define composite business metrics (such as OEE, energy efficiency ratio) in natural language. Metrics update continuously with real-time data and become platform-wide referenceable data assets.
Anomaly Detection	Identify deviations from normal patterns before values exceed thresholds, catching potential problems earlier than threshold-based alerting.
Trend Forecasting	Predict future changes based on historical trends, sending preventive alerts before problems occur. Shifts operations from reactive response to proactive prevention.

AI Assistant	Capability Description
Root Cause Analysis	Automatically trace causes after events occur, working backward layer by layer along the equipment relationship network. Turns half-day expert investigations into minutes of AI-powered reasoning.
Health Assessment	Continuously track the health status of every piece of equipment, generating health scores and degradation trends so maintenance resources are precisely directed where needed most.
Event Query	Search historical events in seconds across dimensions such as time, severity level, equipment, and keywords. No more manual page-by-page filtering.
Comparative Analysis	Complete multi-device, multi-period, multi-year horizontal comparisons with a single sentence. Turns half a day of data wrangling into seconds of intuitive charts.
Report Generation	Automatically collect panel charts, analysis results, and event records. Generate structured operational reports in one sentence. Returns people's time to decision-making.
Document Q&A	After uploading equipment manuals and maintenance guides, AI provides precise answers based on enterprise private knowledge. Turns flipping through documents into asking documents.

These AI Functions can be triggered in multiple ways: from a page, for example when a user clicks the panel interpretation button on a panel, the panel interpretation assistant starts; event-driven, when an alert is generated, the root cause analysis assistant automatically begins retrieving related data and generating an investigation report; or user-initiated, by describing needs in natural language in the AI Chat interface, with the corresponding AI Function responding in real time.

8.3 Skills extension: turning industry knowledge into reusable capabilities

Industrial domains have deep industry barriers. Every enterprise has its own unique process standards, operating procedures, and fault handling experience. General-purpose AI knows little about any of this. The Skills mechanism is designed precisely to address this challenge. It turns industry knowledge scattered in human minds and documents into standardized capability units that AI agents can call, compose, and iterate on.

TDengine IAR includes an extensible Skills management system. Each Skill is defined by four types of properties: name, description, type (system pre-installed or user-defined), and ownership, which can belong to a specific AI Function or serve as a common Skill. System pre-installed Skills are viewable but cannot be edited or deleted. Users can also upload their own customized Skills: work process description documents written in natural language containing business scenarios and industry knowledge, which can then configure or create personalized AI applications. Customized Skills support full CRUD operations and version management. Users can iterate, roll back, or publish at any time.

Beyond Skills bound to specific AI Functions, IAR also supports a class of Common Skills. These do not belong to any single AI Function but are shared and called by all AI Functions, for example, cross-system data queries, universal unit conversions, and standard alert routing. Common Skills allow AI capabilities to be reused and composed like a component library.

The significance of Skills extends far beyond the technical layer. Through Skills, a company's years of accumulated process judgment, fault investigation procedures, and emergency response plans can become explicit, structured, and version-controlled. Tacit knowledge that once lived only in the minds of veteran experts becomes a digital asset the enterprise can pass on.

8.4 Knowledge base management: giving AI industry context and business awareness

The reason general-purpose AI struggles to offer valuable recommendations in industrial scenarios is that its knowledge base comes from public internet information, while the core industrial knowledge (process manuals, operating procedures, historical fault cases) is not on the internet. It is private, offline, locked inside enterprise filing cabinets and expert minds.

The task of TDengine's Knowledge Base management is to transform this scattered, tacit industry knowledge into structured information that AI can retrieve and reference. Enterprises can upload documents such as process manuals, operating procedures, equipment maintenance guides, and historical fault cases to the TDengine platform. The AI engine automatically indexes this content. After this, in AI Chat Q&A, root cause analysis, data interpretation, and other scenarios, AI can provide answers grounded in the enterprise's actual processes and operational experience, rather than relying on generic knowledge for speculation.

The Knowledge Base and Skills are complementary: Skills define how to do it, turning industry experience into executable AI workflows. The Knowledge Base provides what the basis is, giving AI reasoning an enterprise-grade, traceable source of knowledge. Together, they transform AI from a generic conversational tool into an industrial agent with genuine industry cognitive capability.

8.5 AI safety: controllable, trustworthy, auditable, robust

The safety risk of industrial AI is not at the level of a chatbot giving a wrong answer. AI getting a process parameter wrong or misjudging equipment status can cause millions in material loss or a safety incident. Enterprise AI trust does not come from AI being naturally reliable and absolutely correct. It comes from AI being embedded in an enterprise operating environment with business semantics, permission boundaries, human-in-the-loop collaboration, execution constraints, complete audit trails, and continuous evaluation.

Controllable

AI's behavioral boundaries are enforced through five layers of mechanisms. (1) Agent Sandbox: each AI agent runs in an isolated environment and can only access explicitly authorized elements and data. (2) Identity-inheriting permissions: TDengine provides RBAC based on user roles. Agents do not have independent superuser permissions; they only inherit the permissions of the initiating user. (3) Dual-layer guardrails: the prompt layer forcibly injects ontology context and permission boundaries; the tool call layer performs secondary verification before every operation. (4) Behavioral rules: three preset behavioral boundaries (autonomous, requires human confirmation, absolutely non-delegable). (5) Task safety levels: low-risk tasks execute autonomously; medium-risk tasks require user confirmation before execution; high-risk tasks require human-in-the-loop final decision-making.

Trustworthy

Reasoning along Industrial Ontology paths is the fundamental guarantee of trust. AI's reasoning paths are constrained by the Industrial Ontology information structure, and reasoning that deviates from the ontology can be blocked or flagged according to governance rules. Every conclusion must clearly indicate the data sources, analysis paths, and ontology nodes on which its reasoning is based. It must be traceable, verifiable, and challengeable. AI output results are automatically cross-validated against the ontology before presentation: do the referenced elements and attributes exist? Are values within upper and lower limits? Are association relationships consistent with the ontology?

Auditable

TDengine IAR records every key node of an AI agent's workflow, from task triggering, Skill invocation, LLM interaction, and tool execution to user feedback, forming a complete evidence chain. Logs are tamper-proof and support multi-dimensional queries by time, user, task, and element. When anomalies occur, root causes can be located quickly.

Robust

AI is an amplifier, not a dependency. Panels or analyses generated by TDengine AI can, once created, continue running without AI, consuming no further tokens. Even if AI services are temporarily unavailable or produce incorrect results, TDengine's core workflows (data ingestion, storage, query, stream processing, alert notification) continue to operate independently and stably, unaffected by AI status. This helps ensure that AI does not introduce new vulnerabilities into production continuity or safety.

TDengine

CHAPTER 09

Model Training and Management

Industrial AI is only as good as the models behind it. TDengine closes the loop from data selection to one-click model deployment, so analytics models are trained, evaluated, deployed, and monitored on one platform instead of being scattered across external tools and personal machines.

09

CHAPTER 09

Model Training and Management

9.1 Module positioning and scope

Analytics modeling is an essential capability of an industrial AI data platform. Exporting data to an external platform for modeling, then copying the results back into the production environment, is a perfect example of the analytics gap that holds traditional industrial data platforms back. TDengine instead brings data, modeling, and inference together on a single platform, closing the loop from data selection to one-click model deployment.

TDengine's time-series analytics engine, TDgpt, already runs advanced analytics in-database, including time-series forecasting, anomaly detection, data imputation, and classification. But users also need a complete frontend tool interface to visually select data, configure analysis algorithms, compare training results, and manage model versions.

For this, TDengine adds a model development and management module to its frontend application. Building on TDgpt's time-series analytics capabilities and the standardized, contextualized AI-ready data in the Industrial Ontology, it helps users complete the full machine learning workflow: model development, training, evaluation, validation, publishing, and asset management.

The business value is direct. In the past, data analytics teams wrote separate scripts for every production line and equipment type, exported data, trained models on external platforms, then copied model files back to the production environment for deployment. The whole process took weeks, sometimes months, and models sat scattered on personal computers, never becoming organizational assets. TDengine compresses all of this into a single platform. Data is already contextualized and standardized in the Industrial Ontology, ready to be referenced for training. Once trained, a model deploys to TDgpt with one click for in-database inference, returning results in milliseconds. Every training run, every inference, every performance change is recorded in the management repository, traceable, auditable, and reproducible.

TDengine covers the full analytics spectrum of model types: classic ARIMA and linear regression, machine learning methods like random forest and gradient boosting, and deep learning and time-series foundation models. Users can fine-tune TDtsfm, TDengine's self-developed time-series foundation model, or open-source models like Time-MoE on their own industrial data, adapting them to specific equipment and processes. Whatever the type or origin, every model follows one unified development, evaluation, registration, deployment, and monitoring process.

The model development and management module can be deployed independently, and supports dedicated GPU servers for time-series foundation model training and fine-tuning.

9.2 Training repository and management repository

TDengine divides the model lifecycle into two stages, each managed by its own repository. This mirrors how enterprises actually divide labor: data scientists focus on model experimentation and development in the training repository, while operations and governance teams focus on stable operation and compliance after deployment in the management repository. The two repositories operate independently, without interfering with each other.

Training repository. Oriented toward model development and training. Users select data variables through a visual interface to create model datasets, reference elements and attributes in the Industrial Ontology directly, configure algorithm parameters, and start training. The system automatically records the configuration, input data, and performance metrics of every training run. AI can assist with algorithm recommendations, parameter tuning, and interpreting training results. The entire process stays inside the TDengine platform, and data never leaves the domain.

Management repository. Oriented toward model deployment and governance. After evaluation and validation, trained models are registered in the management repository with complete metadata: algorithm type, training data range, performance metrics, and applicable scenarios, forming a searchable, comparable model asset catalog. Administrators can deploy registered models to TDgpt with one click for in-database inference, and inference results are directly callable through SQL. The system continuously monitors the inference performance and data drift of deployed models, and automatically triggers retraining recommendations when performance degrades.

TDengine also provides model dataset management capabilities, helping modelers define, manage, and maintain datasets across various modeling scenarios.

CHAPTER 10

User Interface and Frontend Applications

AI's value has to be delivered through the interface. TDengine removes the barriers between people and data, offering zero-code web applications plus IM conversations, an Excel add-in, the CLI, and MCP, so every user, from business staff to frontline engineers to developers, interacts with the platform in the most natural way.

10

CHAPTER 10

User Interface and Frontend Applications

10.1 Positioning and architecture

The structural problems of traditional industrial data platforms and the core challenges of AI adoption appear to be independent, but they share a common root cause: the distance between platform capabilities and users is too great. Industry knowledge is locked in expert minds. Data querying requires SQL skills. Analysis modeling requires a Python environment. Every barrier filters out the people who actually need the data. And AI's value ultimately has to be delivered through the user interface. If the interface itself has barriers, none of the technical breakthroughs upstream can translate into business results.

TDengine's interface design principle is simple: let every user interact with the platform at the lowest possible barrier and in the most natural way. TDengine ships with built-in zero-code web applications that are fully functional and powerful, but they are not the only entry point. IM conversations, the Excel add-in, the CLI command line, and the MCP protocol together form a multi-channel interaction system that covers business staff, frontline engineers, professional analysts, and IT developers.

10.2 Built-in interface and analysis applications

TDengine's built-in frontend web application helps business users complete every analysis task, from data monitoring and process analytics to AI insights, through a zero-code interface. Users do not need to write SQL. They do not need to write Python. All data work happens in a browser through clicking and dragging.

Architecturally, TDengine's frontend web application is fully decoupled from the data layer, communicating with backend services through TDengine's standard APIs and MCP. It can be deployed independently, either as part of an integrated deployment or on a standalone web server or container, with support for horizontal scaling. The web application is replaceable and extensible. Users can build personalized web applications on top of TDengine's open data services, or embed any TDengine web page into other industrial systems.

TDengine's frontend application is powerful, complete, and 100% zero-code, with an extremely low barrier to entry and excellent user experience. It includes the following capabilities:

Data Visualization	Information is organized around assets. Every element has its own panels and dashboards. When a user navigates to a piece of equipment, they see a complete view of that equipment, not a blank canvas. The system includes built-in trend charts, bar charts, scatter charts, heatmaps, histograms, candlestick charts, and other panel types, with an interactive experience comparable to tools like Grafana but with added business context. Panels automatically sense the engineering units, upper and lower limits, and hierarchy position of the associated element. The system supports panel templates and dashboard templates: define once, apply everywhere. When assets scale to thousands of devices, template-driven consistency is essential.
Zero-Code Real-Time Analysis Creation	Users can create real-time analyses for any element through a graphical interface: select trigger type, define computation expressions, specify output targets. The underlying computation runs on TDengine's stream processing engine inside the TSDB. AI assists with creation: describe "calculate the average power factor for the past 15 minutes" in natural language, and the system auto-fills the configuration form. It also supports anomaly detection powered by TDgpt with one-click deployment of SPC statistical process control rules. No need to manually write threshold rules. After selecting attributes and algorithms, the system automatically identifies anomaly intervals.

Full Event Lifecycle Management

Equipment trips, temperature exceedances, and batch completions are operational records with clear start and end times, which is what TDengine defines as events. Events are automatically generated by real-time analysis rules, with built-in template-defined naming, severity levels, and acknowledgment workflows. In the AI era, the business value of structured events is especially prominent. Events are the bridge between industrial data and AI agents: feeding AI the structured record "Compressor surge, started 10:23:15, lasted 12 seconds, severity critical" is far more effective than feeding it millions of raw measurement point readings.

Process Analytics

A built-in analysis workbench for completing various process analyses through zero-code interactive operation, including: window analysis, scanning historical data by conditions and highlighting qualifying time windows; correlation analysis, calculating correlation coefficients between attributes and evaluating p-values; time-series forecasting, powered by TDgpt with automatic pattern detection; batch analysis and comparison, multi-batch overlay comparison, time alignment, envelope analysis, and quality traceability; profile search, selecting a reference waveform and sliding-scan matching for the most similar historical periods; and one-click deployment of monitoring rules with automatic calculation of process metrics such as Cp and Cpk.

AI Capability Invocation and Configuration Management

TDengine IAR's AI Functions are embedded directly into users' work scenarios through the web application, with complete agent configuration management and observability:

AI Application	Description
Chat BI	Ask questions in natural language within the AI Chat interface. AI automatically generates queries and visualizations.
Panel Interpretation	One-click trend summarization, anomaly interpretation, and business recommendations for the current visualization panel.
Zero Query Intelligence	The system automatically senses the current business scenario and proactively pushes real-time analyses and panel recommendations.
Root Cause Analysis	After an equipment event triggers, automatically retrieves related data, generates hypotheses, cross-validates, and outputs an investigation report.

Automated Tasks

Support automatic triggering and execution of various predefined tasks: scheduled automatic generation of operational data visualization panels pushed to specified email addresses or Feishu groups; when event alerts trigger, automatic initiation of root cause analysis and generation of investigation reports; continuous monitoring of process parameters based on real-time data, automatically creating analysis tasks and notifying responsible persons when deviations from targets occur. These automated workflows are defined through AI Agent Skill orchestration. Users do not need to write code; they complete them through configuration.

Action Push and Execution

TDengine is not only an analysis and insight platform but also an action trigger platform. Built-in action templates allow anomalies discovered by real-time analysis and business recommendations generated by AI to be executed through multiple channels: Webhook callbacks push alerts and handling recommendations to third-party systems (such as work order systems, MES, ERP) in real time; API interfaces let external systems actively pull analysis results and recommended actions; email and instant messaging notifications deliver key insights to responsible persons. For example, when SPC real-time analysis detects a continuous decline in the process capability index Cpk, the system not only generates a quality alert but can also push process parameter adjustment recommendations to the MES system via Webhook. From data anomaly to operational action, the process stays connected end to end.

The purpose of industrial data analysis is not to generate a good-looking chart. It is to drive decisions and actions. TDengine connects AI's analytical insights with the enterprise's operational workflows, achieving a complete closed loop from data to decision execution. Raw data enters the platform, goes through storage, computing, modeling, analysis, and AI enhancement, and ultimately outputs as executable action commands to the plant floor. This is the end-to-end closed loop that TDengine pursues.

10.3 Other flexible interaction interfaces

The web is not TDengine's only interaction interface. Different types of users need different entry points and channels. The platform should not be designed only for IT staff, nor should it assume that all users can always log into a web page. Beyond the web application, TDengine provides multiple interaction channels covering different scenarios.

AI Conversations in Instant Messaging

Frontline industrial staff, inspectors, shift supervisors, and duty engineers, often lack the conditions to write SQL or operate an analytics platform, but they also need quick data answers. TDengine supports direct integration with mainstream instant messaging platforms such as Feishu and WeCom, letting users initiate data queries, execute analysis tasks, and receive alert notifications directly in familiar IM conversation windows using natural language. A user simply sends "What is the OEE trend for Line 3 over the past hour?" to the chatbot in WeCom, and TDengine automatically retrieves data, generates charts, and returns them in real time. When equipment anomalies occur, alert messages are pushed directly to designated message groups. Responsible persons can launch root cause analysis directly within the conversation. No need to log into a web platform. No need to switch tools. Platform capabilities extend to every frontline worker's fingertips.

Analyze Data Directly in Excel

A large portion of industrial enterprises' daily operational management revolves around Excel: quality reports, daily energy consumption reports, shift handover records. For these users, switching to an entirely new analysis platform is itself a cost. TDengine provides a native Excel add-in, letting users directly query time-series data, element attributes, events, and aggregation results within their familiar spreadsheet environment, then use pivot tables, charts, and other familiar tools for secondary analysis. Data no longer goes through the tedious export-import-cleaning process. In Excel, users can directly access the data assets in the Industrial Ontology.

CLI Command Line and Automation Integration

For automation scripts, CI/CD pipelines, AI agent task orchestration, and similar scenarios, graphical interfaces are actually less efficient. TDengine provides a CLI (command line interface) supporting data query, system management, and automated operations, with structured output formats (JSON, CSV, Table) that can serve directly as input for downstream scripts. The CLI lets TDengine's capabilities be embedded into any automated workflow, becoming a programmable component of the enterprise data infrastructure.

MCP: A Semantic Data Channel for AI Agents

Web pages and IM conversations are interaction methods designed for people. MCP (Model Context Protocol) is designed for AI agents. Through MCP, AI agents can directly discover and invoke modeled industrial data within TDengine. Users can also introduce third-party industrial agents. By connecting to TDengine with an API Key, they can complete context data reads and execute authorized operations. TDengine is not trying to monopolize AI. It acts as AI infrastructure, providing safe, controlled semantic integration and execution paths for different types of AI capabilities.

TDengine

CHAPTER 11

Enterprise Platform Governance

For an enterprise-grade platform, security, compliance, and manageability are not add-on features. They are the entry threshold. TDengine unifies system security and AI safety under a single console, so the enterprise governs users, resources, models, and compliance from one place.

11

TDengine

CHAPTER 11

Enterprise Platform Governance

For an enterprise-grade application platform, security, compliance, and manageability are not add-on features. They are the entry threshold.

TDengine's enterprise platform governance works at two layers. System security covers the platform infrastructure itself, from user authentication, access control, and data encryption to backup and disaster recovery. AI security focuses on the agent runtime environment, from behavioral control and reasoning validation to audit traceability. Both layers are unified under a single management console, configured and monitored centrally by the enterprise.

User and Permission Management

The platform supports multiple single sign-on (SSO) methods, including OAuth 2.0 and LDAP. Role-based permissions provide element-level fine-grained control: users in different roles can access different plant, workshop, or equipment subtrees, keeping data access to the principle of least privilege. For programmatic access scenarios such as MCP, CLI, and SDK, it also supports user-level API Key creation, rotation, and lifecycle management.

System Resource Management

TDengine manages all system resources uniformly, including elements and attributes, real-time analyses, events, panels, dashboards, AI Functions, Skills, Knowledge Base documents, and system configurations. Each resource type supports its own five-level security policy: viewable, editable, change reason required on edit, secondary password verification, and MFA multi-factor authentication. Resource security policies are enforced globally, independent of specific users and roles. Even a system administrator must complete secondary authentication to modify a resource protected by MFA.

Version Management and E-Signature

All system elements, attributes, panels, real-time analyses, events, process analyses, AI Function configurations, and Agent Skills are fully managed through Git-based version management. Every change is traceable, comparable, and rollback-able. Users can keep private versions that become visible to all once published. For sensitive operations, such as modifying process alarm thresholds, publishing new AI agents, or changing critical business rules, the system can require MFA and e-signature confirmation. The modifier must verify their identity, state the reason for change, and wait for approval before the change takes effect. All related records are written to the audit log.

Audit Trails

The system automatically records tamper-proof operation logs covering all system operations: user logins, element creation and modification, analysis rule changes, event handling, and AI agent invocations. Logs support multi-dimensional filtering, querying, and export by time, user, and operation type, meeting enterprise compliance audit and security traceability requirements.

System Notification Configuration

The management console supports configuring system basic settings, notification contacts, notification templates, and email servers. Event alerts can be sent via email, Feishu, or any Webhook-capable method. Unacknowledged alerts support escalation notifications.

Data Backup and Recovery

Administrators can configure automatic backup policies and recover data from backups at any time. The system provides full and incremental backups, and can combine file-based backup methods for extra data protection.

AI Safety Governance

TDengine's AI safety governance is built around four dimensions: controlled, trusted, auditable, and robust. Which LLM to use, which model services to connect to, and what types of operations models can perform are all configured centrally by system administrators according to enterprise security policies. Individual users cannot connect to LLMs on their own, keeping AI invocation under centralized enterprise control.

Deployment and License Management

The platform supports multiple deployment topologies, including single instance, high availability, private deployment, public cloud, and hybrid cloud, with deployment methods such as Docker, install packages, Ansible, and Helm. The management console provides license item viewing and usage monitoring.

Certifications and Compliance

TDengine has completed SOC 2 Type II examination and maintains ISO 27001 and ISO 27017 compliance/certification materials, with security documentation available through TDengine's legal and security resources. It has also completed adaptation work for mainstream enterprise operating systems and CPU architectures.

CHAPTER 12

Open Data Services

TDengine is designed as an open industrial data platform. Product capabilities are exposed through documented programmatic interfaces, so data can move across open protocols, standard interfaces, and semantic data channels. The goal is to avoid customer lock-in and prevent new data silos.

12

TDengine

CHAPTER 12

Open Data Services

TDengine is designed as an open industrial data platform. Product capabilities are exposed through well-documented programmatic interfaces. In the AI era, openness is not a nice-to-have. It is a necessity. Industrial AI applications require reliable programmatic access to contextualized data. A closed system that locks data behind proprietary protocols becomes a bottleneck for the entire AI architecture. What TDengine outputs is not just SQL query results. It is data services with business semantics and data context.

Traditional Data Interfaces

TDengine supports standard JDBC and ODBC interfaces, connecting directly to BI and visualization tools such as Power BI, Grafana, Tableau, and Seeq. It also provides a native Excel add-in, letting business users query time-series data, element attributes, events, and aggregation results directly in Excel without writing code or SQL. For industrial users accustomed to building quality reports and daily operational reports in Excel, this means consuming data in a tool they already know. TDengine's built-in process analytics capabilities cover many scenarios that have traditionally required specialized analysis software such as Seeq and TrendMiner. Users can complete the full workflow, from data query to deep analysis, within a single platform, without integrating additional analysis tools.

Data Subscription and Publishing

TDengine itself can be treated as a message queue. Users define topics with SQL and subscribe to data over Kafka or MQTT protocols, with precise control over distribution granularity. It can also serve as a data source, publishing processed data in real time to Kafka, MQTT Broker, Flink, Spark, and other external systems. Event alerts trigger third-party systems through standard Webhooks. Visualization panels can be embedded in any web application via iframe.

AI-Oriented: MCP and CLI

TDengine natively supports the MCP (Model Context Protocol). External AI agents can directly discover and consume industrial data through MCP, looking at a semantic entry point of "plant.workshop.equipment.measurement_point" rather than "database.table.field." Access supports Streamable HTTP and SSE, with API Key authentication management. The CLI (Command Line Interface) serves terminals, scripts, and AI agents, supporting data query, system management, and automated operations. It is suitable for terminal operations, programmatic automation, and agent tasks.

External System-Oriented: API and SDK

A complete REST API described by the OpenAPI specification supports automatic SDK generation for any language. Powerful SDKs are provided for Java and Python, with connectors available for Go, Rust, C/C++, and other languages, supporting enterprise-grade features such as connection pooling, automatic reconnection, and batch writing.

Data Flows Where It Is Needed

The prerequisite for bringing AI to every industrial data stream is that data can flow freely to wherever it is needed, whether stored in public cloud, private cloud, or edge nodes. TDengine's open data services are exactly this bridge: open protocols, standard interfaces, and semantic data channels. The goal is to avoid customer lock-in and prevent new data silos.

TDengine

CHAPTER 13

Industrial Deployments and Applications

TDengine serves organizations across dozens of industries as a modern alternative to traditional data historians. It has been installed more than 1 million times worldwide and is trusted by over 1,000 customers.

13

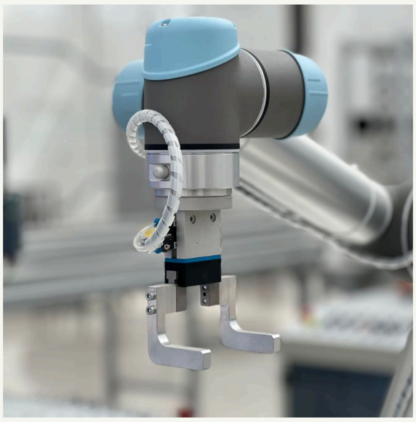
TDengine

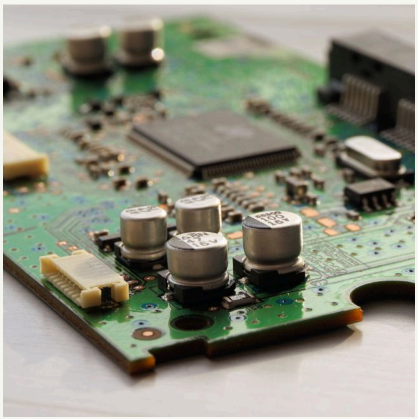
CHAPTER 13

Industrial Deployments and Applications

Industrial operations increasingly depend on time-series data for monitoring, control, analytics, and AI. As a modern alternative to traditional data historians, TDengine is used in production environments to replace legacy data infrastructure, reduce storage and server costs, support high-frequency data ingestion, and make operational data available for real-time analysis.

With over 1 million installations worldwide and more than 1,000 paying customers, TDengine has been deployed in production environments ranging from process industries and power systems to manufacturing, transportation, utilities, research, and digital infrastructure.





TDengine

13.1 Oil, petrochemicals, and energy

Oil, petrochemical, and energy companies often manage large volumes of process data across plants, fields, stations, and control systems. They need reliable ingestion, fast query, and operational access without adding more fragmented systems.

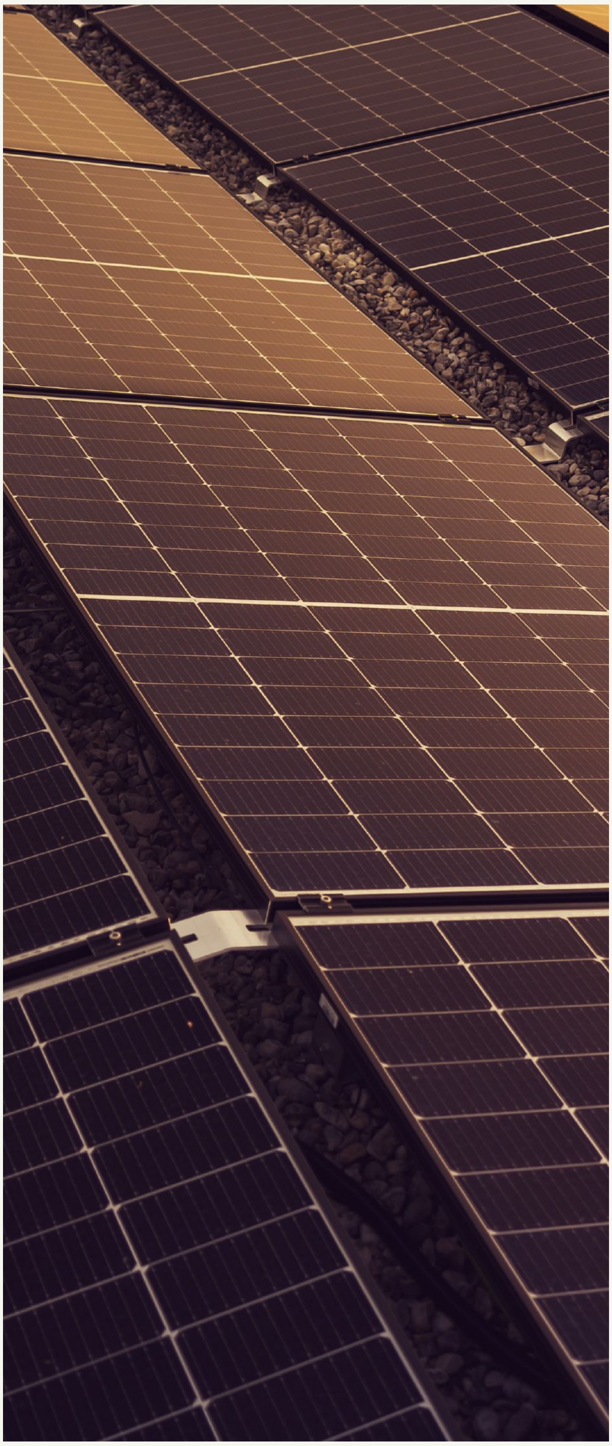
In one petrochemical deployment, TDengine replaced an imported traditional data historian and expanded system capacity by 4x. More than 700 process diagrams could be displayed at high speed, reducing dependence on a closed proprietary system.

In an oilfield deployment, TDengine replaced a traditional relational database used for industrial time-series workloads. Storage cost fell by 85%, while write performance improved. This type of result reflects the benefit of using a database designed for high-cardinality, high-frequency time-series data instead of a general-purpose relational database.

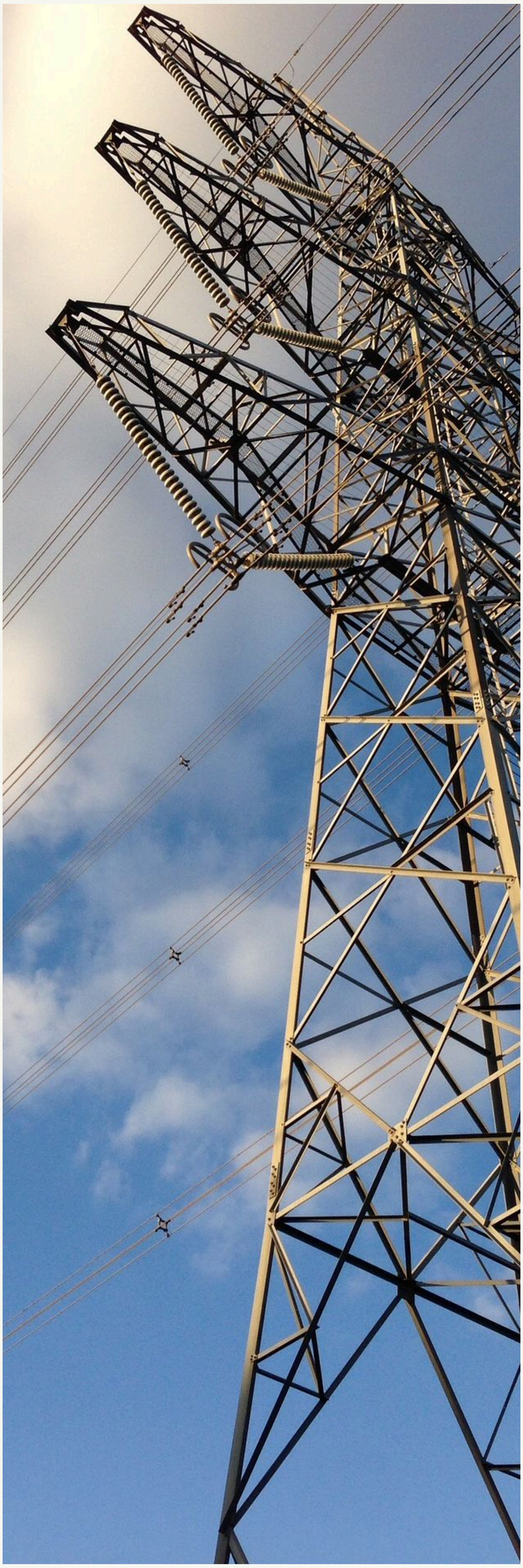
Energy operators have also used TDengine to consolidate and accelerate production data systems. One digital operations platform processes 10 billion data points per day on TDengine while reducing server count by more than 60%. Another software provider consolidated more than 40 Oracle databases into 9 TDengine clusters covering the full exploration and production workflow. In another deployment, TDengine replaced OpenTSDB and improved computation timeliness by up to 100x.

In new energy scenarios, TDengine has been used to unify data across wind, solar, and energy storage sites. One centralized control center used TDengine to manage massive heterogeneous data from 9 wind-solar-storage sites, replacing the previous "one site, one system" model with a shared data foundation. An energy storage provider migrated from an open-source time-series database and improved write performance by 18x while reducing storage footprint to one-tenth of the previous system. A global energy storage company built real-time analytics on TDengine and achieved a data compression ratio close to 1:20. A solar data platform uses TDengine to stably support 4 billion new data points per day.

Together, these deployments show TDengine's role in energy environments that need large-scale ingestion, lower infrastructure cost, and real-time operational analytics.



13.2 Power and utilities



Power generation and utility companies need reliable monitoring, fast alerting, and long-term historical analysis across equipment, plants, and distributed assets. These workloads place heavy pressure on data ingestion, storage, and query performance.

In one hydropower deployment, TDengine supported an AI-assisted intelligent inspection system. The system reduced monitoring workload by 60% and helped increase annual power generation by approximately 300 million kWh. The operating model shifted from manual screen monitoring and experience-based judgment toward automated inspection and intelligent alerts.

In another power automation deployment, TDengine replaced a Hadoop-based data architecture and resolved persistent data loss during writes. For industrial operations teams, this kind of replacement affects the reliability of downstream monitoring, reporting, and analytics.

Utility deployments show similar patterns. A large water utility platform built on TDengine supports million-level device connectivity and reduced data query latency from 10 seconds to 10 milliseconds. In city gas pipeline monitoring, TDengine helped improve overall system efficiency by more than 30%.

These examples show TDengine supporting utility-scale operational data workloads where reliability, latency, and long-term maintainability matter.

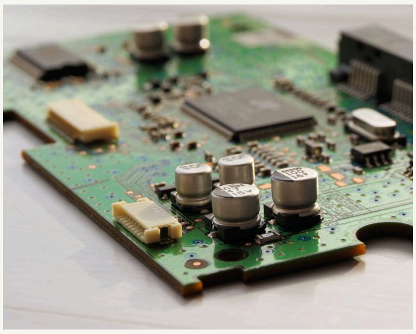
13.3 Smart manufacturing, mining, and industrial equipment

Manufacturing and mining environments generate large volumes of equipment, production, quality, and process data. Traditional databases can struggle when these workloads combine high write frequency, long retention periods, large numbers of measurement points, and frequent historical queries.

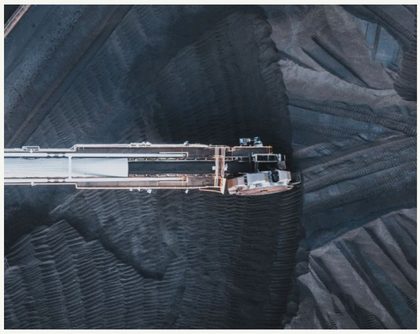
In a CNC machine tool deployment, TDengine supported a predictive maintenance system that improved model accuracy by 25% and reduced unplanned downtime. TDengine provided the time-series data layer needed for model training, equipment monitoring, and maintenance decision-making.

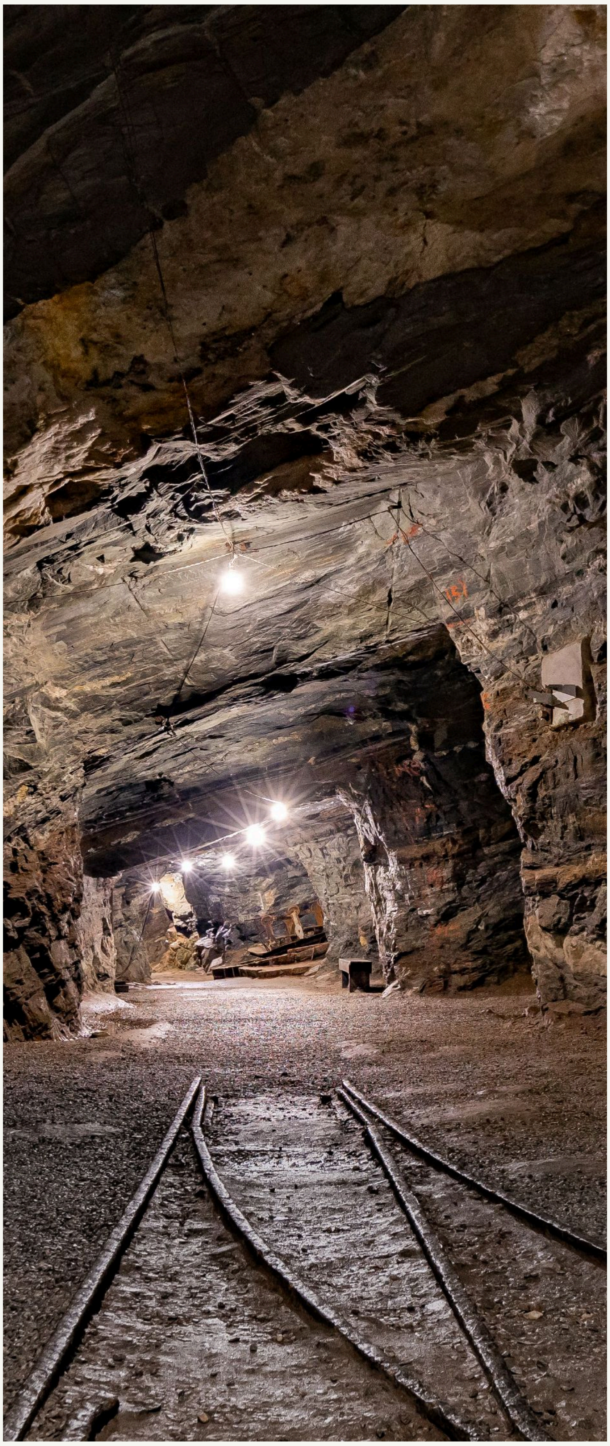
A steel industry IoT platform ran on TDengine for four years with zero faults while reducing storage cost to one-tenth of the previous level.

An electronics manufacturer improved query performance by 15x and optimized storage space by 40x.



In mining, one enterprise compressed 1.1 billion equipment data records to 12GB, making large-scale historical data retention more practical.





TDengine has also been used in heavy industrial equipment scenarios. A coal mining equipment manufacturer accelerated TB-scale data queries from minutes to seconds. In tobacco manufacturing, a major factory built a plant-wide data layer with 100,000 measurement points under high-frequency collection, validating TDengine's stability in ultra-large-scale industrial environments.

These deployments show TDengine's fit for manufacturing and equipment-intensive industries where storage efficiency, query speed, and operational reliability all affect the value of industrial data.

13.4 Automotive, connected vehicles, logistics, and IT infrastructure

Automotive, logistics, and IT infrastructure workloads often involve large distributed fleets, high-frequency status data, and frequent operational queries. TDengine is used in these scenarios to support real-time monitoring, dispatching, database monitoring, and historical analysis.

In a nationwide battery-swap network, a new energy vehicle company used TDengine to support real-time monitoring and intelligent dispatching. Status data from thousands of swap stations and tens of thousands of battery bays is ingested at millisecond-level latency, supporting battery-swap decisions.

In another new energy vehicle data platform, TDengine improved compression performance by 10x to 20x. A construction machinery enterprise shortened its data export process from 1 hour to 10 seconds after replacing a traditional MySQL-based solution.

In logistics and IT infrastructure, a major logistics platform replaced MySQL with TDengine for database monitoring. Storage footprint was reduced to one-tenth of the previous level, while the platform retained the ability to analyze raw historical data from the past 45 days on demand.

These examples show TDengine supporting high-volume operational data outside traditional plant environments, especially when the workload depends on fast ingestion, compact storage, and responsive historical query.

13.5 Pharmaceuticals, food production, and scientific research

Research, pharmaceutical, and food production environments need reliable data management across experiments, production processes, quality workflows, and long-term analysis. In these scenarios, the value of time-series data depends on whether it can be searched, compared, reproduced, and connected to process context.

In a chemical research institute, TDengine IDMP was adopted to connect the data management chain from laboratory R&D to pilot scale-up. Experimental data that had previously been scattered across Excel files and personal computers became structured assets that could be searched, compared, and reproduced.

In food manufacturing, a large enterprise migrated from an open-source solution to TDengine and improved query response from seconds to milliseconds. This kind of improvement directly affects production reporting, quality analysis, and operational decision-making.

In scientific research, the National Astronomical Data Center built a time-domain astronomical data management platform on TDengine, providing efficient storage and query capabilities for petabyte-scale astronomical observation data.

These deployments show that TDengine is not limited to factory-floor monitoring. It can also serve as a time-series data layer for research, experimentation, and scientific workloads where scale, query speed, and data reproducibility are essential.

13.6 From deployment to data value

Across these deployments, TDengine is used when existing systems can no longer keep up with the scale, cost, or query requirements of industrial time-series data. In many cases, the first step is infrastructure replacement: moving from traditional data historians, relational databases, Hadoop-based systems, or other time-series databases to a platform built for high-frequency industrial workloads.

The value does not stop at storage and performance. Once operational data is ingested, compressed, queried, and organized in a consistent way, it becomes usable for real-time monitoring, historical analysis, AI inspection, predictive maintenance, root cause analysis, and business reporting.

For industrial enterprises, this is the practical path from raw time-series data to operational value: first make the data reliable and accessible, then make it available to the applications, engineers, and AI systems that depend on it.

TDengine

CHAPTER 14

Summary and Outlook

Founded in 2017, TDengine began as a purpose-built time-series database for industrial workloads. After its core engine was open-sourced in 2020, it grew into one of the world's most widely adopted time-series databases, with over 1 million installations worldwide. With the launch of TDengine Cloud in 2022, the platform expanded from self-managed deployments to fully managed cloud environments, and has since evolved into an AI-native industrial data platform for the full industrial data lifecycle.

14

TDengine

CHAPTER 14

TDengine Global Community

A global developer and customer footprint across industrial data infrastructure.

25,000+
Stars

5,000+
Forks

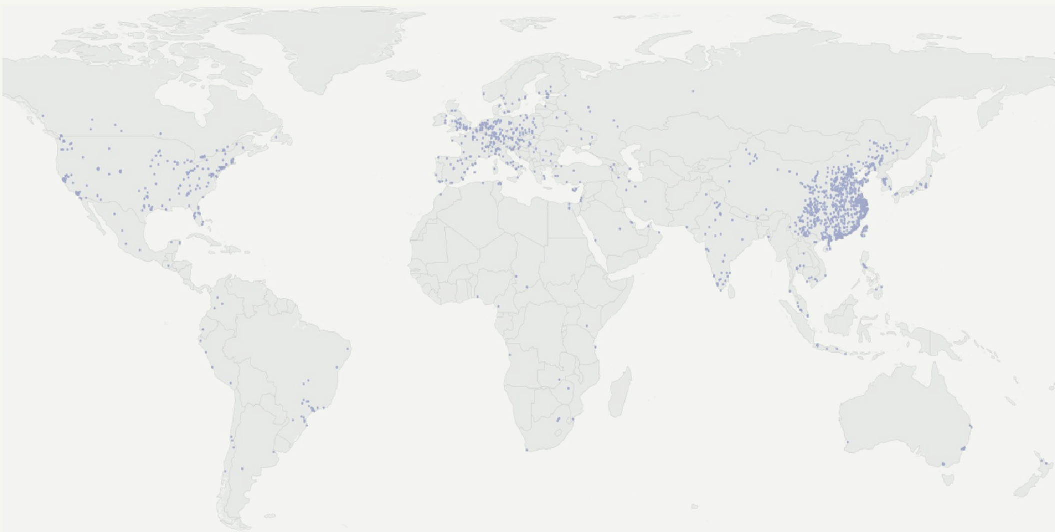
28,760+
Pull Requests

1,081K+
Instances

700+
Daily Installations

1,000+
Paid Customers

Data as of July 2026



Why TDengine

Traditional data historians (such as PI System) provide mature, site-level time-series collection with strong industrial protocol support. Their limitations appear when organizations need to centralize data across sites, add semantic context beyond proprietary asset frameworks, run advanced analytics without exporting data, or enable AI-driven workflows. TDengine can coexist with existing data historians while providing these capabilities.

General-purpose time-series databases offer solid storage and query performance but lack industrial connectors, asset modeling, stream processing, and AI capabilities. Building a complete industrial data platform on a general-purpose TSDB requires integrating and maintaining multiple additional components.

Cloud data lakes are designed for general analytics at scale, not for the millisecond ingestion and real-time processing that industrial environments demand. They require significant customization for industrial protocols and real-time workloads.

DIY open-source stacks (Kafka + Flink + TSDB + Redis + Grafana) provide flexibility but shift the integration and maintenance burden to the operator. In many organizations, total cost of ownership can exceed commercial platforms once engineering time, ongoing maintenance, and reliability risk are included.

TDengine's differentiated position is that it combines purpose-built industrial time-series storage with built-in stream processing, asset contextualization, governed AI, and multi-channel access, in a single platform, with a free entry point for production use.

Capability	Traditional data historians	General-purpose TSDBs	Cloud data lakes	DIY open-source stacks	TDengine
Industrial time-series storage	Purpose-built, high performance	Purpose-built, high performance	General-purpose, often requires tuning	Depends on chosen database	**Purpose-built, 10x+ write performance**
Industrial connectors	Often extensive (OPC, Modbus, etc.)	Often limited, community-driven	Typically requires custom ingestion	Typically manual per-source integration	**20 zero-code connectors (OPC, MQTT, Kafka, PI System, etc.)**
Asset context and ontology	Often proprietary Asset Framework	Typically not included	Typically requires external modeling	Must be built from scratch	**Built-in Industrial Ontology with tree + graph model**
Real-time analytics	Basic calculations, streaming varies by vendor	Typically requires external tooling	Usually requires separate stream processor	Typically requires Kafka + Flink + custom code	**Built-in stream processing with SQL, millisecond response**
AI readiness	Typically requires data export to external tools	Typically requires data export to external tools	Typically requires data export to external tools	Typically requires data export to external tools	**In-database AI analytics (TDgpt), governed AI Agent Runtime**
Deployment flexibility	Primarily on-premises	Varies by vendor	Typically cloud-first	Self-managed only	**Cloud, on-premises, edge, hybrid**
Free production entry point	Typically requires paid license	Often open-source core, enterprise features gated	Often includes free tier with limits	Free but requires integration labor	**Free forever for deployments of up to 5,000 tags, with full platform functionality**

A Practical Path to AI-Native Industrial Data

The problem TDengine set out to solve is not new. For two decades, industrial data historians and IoT platforms have promised to turn data into value. The bottleneck was never the storage layer. It was the conversion chain: expert-dependent analysis, IT-gated workflows, and analytics that happened outside the platform.

TDengine's approach is to compress that chain. Data ingestion goes from weeks of coding to hours of configuration. Queries go from SQL to natural language. Analysis stays inside the platform instead of being exported to Python scripts. AI operates under governance instead of running as an unaccountable black box. And the platform itself is free to start, so organizations can prove value on one production line before committing to enterprise scale.

Bring AI to Every Industrial Data Stream. This is the commitment TDengine was built to deliver.

Next steps

- **Download TDengine All-in-One** and start with a deployment of up to 5,000 tags, free forever
- **Explore TDengine Cloud** at cloud.tdengine.com for a fully managed instance in under 60 seconds
- **Talk to an engineer** at tdengine.com/contact to discuss your specific industrial data challenges

TDengine

TDengine® is an industrial AI infrastructure company. Its AI-native industrial data platform is purpose-built to make industrial data a true factor of production in the age of AI, giving industrial enterprises everywhere an open, independent, and intelligent next-generation data foundation.

The platform has two integrated components. TDengine TSDB is a high-performance, distributed time-series database with built-in caching, stream processing, and data subscription. TDengine IDMP is an AI-native industrial data management platform that combines an industrial ontology, industrial atomic applications, and an industrial agent runtime. Together, they bring AI capabilities to industrial operations, delivering a full-stack solution that spans the entire industrial data flow, from collection, storage, and real-time analytics to visualization, event management, root-cause analysis, and AI-powered insights. For organizations running traditional data historians or legacy industrial IoT platforms, TDengine is a modern alternative, built for the AI era.

TDengine's core technology is developed entirely in-house, backed by multiple patents. The codebase is 100% self-developed and does not depend on any open-source database or third-party core software. The TDengine TSDB kernel is open-sourced under the AGPL license and has surpassed one million installations globally, making it one of the most widely adopted time-series databases in the world.

We believe that foundational software must be open to build a truly thriving ecosystem, and that industrial AI must be accessible to realize the full value of industrial data. That is why TDengine is free to use, forever, for deployments of up to 5,000 tags, with full platform functionality and no trial deadline. Every industrial company, regardless of size, deserves equal access to advanced data infrastructure and AI capabilities. Together, we can bring global industry into a new era of AI.

3031 TISCH WAY
SAN JOSE, CA 95128
SUPPORT@TDENGINE.COM

TDengine is free forever
for up to 5,000 tags
Download TDengine Free Tier

Follow us:

Join our Discord community:

Watch Tutorial Videos: