

# ImBrain White Paper

## ImBrain™ — White Paper

*An Open-Architecture Industrial Historian for the Modern Plant*

**Author:** Branimir Georgiev, Imbra.Soft **Version:** 0.1 — Draft **Date:** March 2026

---

### Abstract

Industrial historians fall into two categories: enterprise platforms priced at six figures with multi-year implementation projects, and siloed SCADA databases that cannot be queried, aggregated, or reasoned over. Small and medium plants are left with spreadsheets, manual shift reports, and no cross-site visibility. Enterprise plants pay for capabilities they use at ten percent capacity, locked into vendor ecosystems with no migration path.

ImBrain is an open-architecture industrial historian that stores time-series data, alarm events, lab results, and virtual tags — connects plants in a mesh — and lets operators query everything in plain language. The core and agents are open source under AGPL. Analytics plugins are commercial. It runs air-gapped on a \$500 industrial mini PC and scales without architectural changes to enterprise mesh deployments spanning hundreds of sites.

---

## 1. The Historian Market Gap

A process historian is the most fundamental data infrastructure a plant can have. It records what happened, when, and why — the foundation for shift reporting, OEE analysis, quality investigations, energy audits, regulatory compliance, and predictive maintenance. Without a historian, a plant is flying blind. With a poor one, it is flying with a map from ten years ago.

The market for industrial historians is effectively a duopoly at the enterprise end and a void at the SME end.

### Enterprise platforms

The dominant commercial historians — PI System (AVEVA), Wonderware (AVEVA), eDNA — are mature, capable, and expensive. A mid-size plant deployment typically costs €80,000 to €300,000 in licence fees, plus implementation, plus annual maintenance at 20% of the licence value. For a large enterprise with hundreds of plants, the total cost of ownership runs into millions annually.

The cost is not the only problem. These platforms are closed ecosystems. Connectors to non-standard devices require vendor-specific add-ons, each with its own licence. Analytics capabilities are gated behind additional modules. Data export formats are proprietary. Moving to a different platform means a multi-year migration project — which is exactly why customers stay, not because the product has earned their loyalty.

### SCADA databases

At the other end of the market, most PLCs and SCADA systems include some form of local data logging — a flat file, an embedded database, or a proprietary historian module. These databases are siloed by design. They record data for the local system and expose it through the SCADA interface — not through a queryable API, not in a standard format, not

accessible from other systems.

An operations engineer who wants to compare last week's production across three lines must export three CSV files, open them in Excel, and build the comparison manually. An engineer investigating a quality issue must correlate process data from the historian with lab results from the LIMS and alarm events from the SCADA — three separate systems, three separate exports, manual reconciliation.

### **The SME gap**

Small and medium industrial companies — plants with 50 to 500 employees, one to ten production lines, annual revenues of €5M to €100M — are the largest segment of the industrial economy and the worst served by the historian market.

Enterprise platforms are priced for enterprise budgets. A €150,000 historian licence is not a consideration for a plant with a €2M annual IT budget. The result: these plants run on spreadsheets, paper shift reports, and operator memory. They have no systematic way to answer “why did line 2 underperform last Tuesday?” or “what is our OEE trend across the last quarter?”

This is not a data problem. The data exists — in PLCs, sensors, and SCADA systems. It is a tooling problem. The right tool does not exist at the right price.

---

## **2. Why Existing Solutions Fail**

The failure of existing solutions is not accidental. It is structural — the result of business models and architectural decisions that prioritise vendor control over customer value.

### **Vendor lock-in by design**

Commercial historians are designed to be sticky. Proprietary data formats mean that migrating to another platform requires re-engineering the entire data pipeline. Proprietary connector ecosystems mean that adding a new data source requires a vendor-supplied adapter, at vendor pricing. Proprietary analytics modules mean that every new capability is a new licence negotiation.

This is a rational business strategy for the vendor. It is a poor outcome for the customer. A plant that has invested three years and €200,000 in a historian deployment cannot switch platforms without a comparable investment. The vendor knows this. Pricing and feature roadmaps reflect it.

### **Closed analytics**

The most valuable capabilities of a historian — OEE calculation, golden batch comparison, anomaly detection, energy accounting — are gated behind analytics modules that cost as much as the historian itself. A plant that wants OEE reporting does not just need a historian. It needs a historian plus an OEE module plus an integration to pull the data into a dashboard. Each component has its own vendor, its own licence, and its own implementation project.

### **No natural language interface**

Industrial historians were designed for engineers who know the tag names. To query a historian, you need to know that tank pressure is stored as `TK-101_PIC_PV`, not “tank pressure”. You need to know the time format, the engineering units, the downsampling resolution. You need to write a query in a proprietary query language.

Operators — the people who need the data most — cannot use the historian directly. They ask engineers, who run queries, who produce reports, which are out of date by the time they are read. The gap between the data and the decision-maker is measured in days.

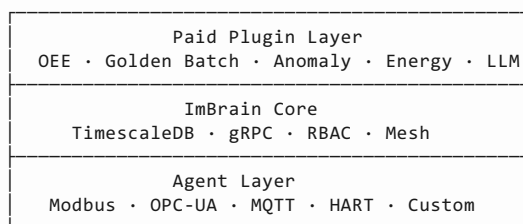
## No mesh, no cross-site visibility

Enterprise plants with multiple sites have no standard way to aggregate data across their historian network. Each site has its own historian, its own schema, its own tag naming convention. Cross-site reporting requires a data warehouse project — ETL pipelines, schema mapping, a central database — a six to twelve month implementation that produces a snapshot, not a live view.

---

## 3. The Open Hourglass Architecture

ImBrain is built on an open hourglass architecture. The metaphor captures the design intent: a thin, open core with a rich ecosystem above and below it.



### Below the core: agents

Go-based agents collect data from any source — PLCs, sensors, lab systems, WinCC alarms, legacy OPC DA servers — using the Imbra Connect SDK. Each agent is a single self-contained binary. A Modbus agent reads registers from a PLC at a configurable interval and forwards DataRecord messages to the Core over gRPC. An OPC-UA agent subscribes to a server and forwards value change events. A LabFileAgent watches a directory for CSV or XML exports from LIMS systems and parses them into structured lab records.

Agents are open source. Official agents are maintained by Imbra.Soft. Community agents are published to the Imbra agent registry under MIT licence. The registry grows with the community — every engineer who writes an agent for their specific device adds to the coverage available to every other ImBrain user.

### The core

The Core is deliberately thin. It receives DataRecord messages from agents over gRPC, stores them in TimescaleDB, and exposes a query interface. It handles RBAC, the mesh protocol, virtual tag computation, and the plugin API. It does not implement analytics, cloud forwarding, or natural language — those are plugin concerns.

This separation is intentional. The Core is AGPL — open, auditable, forkable. Plugins are commercial. The boundary between them is the plugin API — a stable, versioned contract that allows Imbra.Soft and third parties to build analytics on top of the Core without modifying it.

### Above the core: plugins

Paid plugins deliver the capabilities that generate value beyond raw storage:

Plugin	Capability	Business value
OEEPlugin	Overall Equipment Effectiveness — availability, performance, quality	10 / 10
GoldenBatchPlugin	Compare active batch vs. best historical run in real time	9 / 10
StreamGatewayPlugin	Hot-path forwarding to Azure Event Hubs / AWS Kinesis	8 / 10
AnomalyDetectionPlugin	Statistical and ML-based anomaly detection on tag streams	8 / 10
ColdArchivePlugin	Nightly Parquet export to S3 / Azure Blob / GCS	7 / 10
EnergyPlugin	Energy consumption accounting and benchmarking	7 / 10
LLMPlugin	Natural language interface — local or cloud LLM	7 / 10
ForecastPlugin	Time-series forecasting on production and quality metrics	6 / 10
SPCPlugin	Statistical process control — control charts, Cpk, out-of-control detection	6 / 10

## 4. The Data Model

ImBrain stores four record types. This covers the full range of data generated by an industrial plant.

### Numeric tags

The standard historian record — a floating-point value, a tag identifier, a timestamp, and a quality flag. Stored in TimescaleDB with configurable downsampling via continuous aggregates:

```

100ms → retained 7 days (per-tag, opt-in)
1s → retained 30 days
10s → retained 1 year
1min → retained 5 years
1h → retained 10 years
1day → retained 50 years (configurable)

```

Downsampling uses time-weighted averages — physically correct for process data where the value between samples is not zero but the last known value. Compression ratios of 10:1 are typical on raw data.

### Alarm events

Full alarm lifecycle — raised, acknowledged, cleared — with source, class, severity, message, area, tag, value at raise, and operator identity. Stored with JSONB metadata for source-specific fields that do not fit the standard schema. Alarm records are retained in full — no downsampling. They are the audit trail. Default retention is 10 years, configurable at deployment for industries with specific requirements (nuclear: 40 years, food and beverage: 5 years).

## LIMS lab results

Lab samples from quality systems — one record per sample, with per-parameter results, pass/fail status, analyst identity, and sample metadata. Structured to support quality trend analysis, golden batch comparison, and regulatory reporting. JSONB metadata accommodates the variation in lab system output formats without schema migration. Default retention is 15 years, configurable for pharmaceutical (15–30 years) and other regulated industries.

## Text tags

String values with timestamps — recipe names, operator comments, mode strings, product codes. Queryable alongside numeric tags and alarm events for context-enriched analysis.

## Virtual tags

Virtual tags are computed values derived from incoming DataRecords — no separate polling, no lag. The VirtualTagEngine evaluates expressions on each incoming sample:

Type	Example
Expression	$(\text{tag\_A} + \text{tag\_B}) / 2$
Totalizer	Accumulated volume from a flow measurement, with meter rollover and batch reset
Rolling statistic	15-minute rolling average of temperature
State	Derived machine state from multiple discrete inputs
Cross-tag aggregate	Average OEE across all lines in an area

Totalizers handle the operational realities of plant instrumentation: meter rollover at maximum count, dead time during maintenance (hold last value or zero), and batch reset via a trigger tag.

## 5. The Plugin Ecosystem

### OEEPlugin — BV10

OEE (Overall Equipment Effectiveness) is the standard KPI for manufacturing productivity. It is the product of three factors: availability (was the machine running when it should have been?), performance (was it running at its rated speed?), and quality (was the output within specification?).

OEEPlugin calculates all three from ImBrain data — downtime events from alarms, speed from production count tags, quality from LIMS results — and stores OEE scores as virtual tags. Results are available in real time, trended over any time horizon, and comparable across lines, shifts, and sites.

### GoldenBatchPlugin — BV9

The golden batch is the best historical run for a given product and recipe — the batch that produced the highest quality output with the least waste in the shortest time. GoldenBatchPlugin identifies the golden batch from LIMS and process data, and overlays it against the active batch in real time using dynamic time warping for phase alignment.

Operators see, at any point in the active batch, how it compares to the best known run. Deviations are flagged before they become quality failures. The batch scorecard — a 0–100 score updated continuously — gives a single number that summarises batch health.

### Natural Language Interface — LLMPlugin — BV7

The LLMPlugin connects ImBrain to a local (Ollama) or cloud (Claude, GPT-4) LLM and

exposes a natural language query interface. Operators ask questions in plain language:

- “Show me the OEE for line 3 last week, broken down by shift”
- “Find all alarms on the reactor circuit last month where pressure exceeded 8 bar”
- “Compare energy consumption across all sites this quarter”
- “Why did line 2 underperform last Tuesday?”

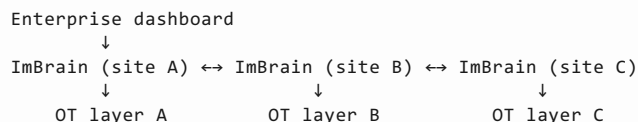
The LLM translates intent into TimescaleDB queries using tag name resolution via vector search. Operators do not need to know tag names, query syntax, or database schema. The historian becomes accessible to everyone in the plant, not just the engineers who configured it.

Vector search (pgvector) resolves natural language descriptions to tag names — “tank pressure” maps to TK-101\_PIC\_PV — enabling the interface to work across any tag naming convention without manual configuration.

---

## 6. The Mesh Network

Multiple ImBrain instances communicate peer-to-peer over MQTT/Sparkplug B, forming a distributed historian network:



### Cross-site queries without data movement

Each site historian stores its own data. The mesh enables queries that span multiple sites without moving raw data to a central server. An enterprise dashboard queries the mesh and receives aggregated results. The raw process data never leaves the site.

This is the correct architecture for industrial data. Raw tag data at 1-second resolution for a large plant is terabytes per year. Moving it to a central server is expensive, creates a single point of failure, and is often prohibited by plant security policy. The mesh aggregates at the query layer, not the storage layer.

### Hierarchical aggregation

The mesh supports hierarchical deployment — unit historians aggregate to area historians, area historians aggregate to plant historians, plant historians aggregate to enterprise historians. Each level stores only what it needs. Aggregation is computed at query time from the level below.

### How instances discover each other

Each ImBrain instance publishes a Sparkplug B birth certificate when it comes online — a retained MQTT message on a well-known topic that announces its identity, site name, and available tag namespaces. Other instances subscribed to that topic register the new peer automatically. When an instance goes offline, a death certificate is published and peers remove it from their active list.

The mesh requires an MQTT broker that all instances can reach. For internet-connected deployments, a managed cloud broker (HiveMQ, EMQTT Cloud) is simplest. For air-gapped plants, each site runs its own local broker and brokers bridge to each other — no site depends on internet connectivity for local historian operation.

### Sparkplug B

Sparkplug B is an MQTT-based specification for industrial data payloads. It provides a standard topic namespace, compressed binary payloads (approximately 20 bytes per tag

value with aliases), and birth/death certificates for device state management. At 10,000 tags per plant at 1-second resolution, Sparkplug B generates approximately 1.6 Mbit/s per plant — well within the capacity of a standard industrial network connection.

---

## 7. Deployment

### Self-contained installer

ImBrain ships as a self-contained installer that bundles everything needed to run — TimescaleDB, the Core, the agent runtime, the plugin framework, and the mesh broker. No cloud dependency. No external licence server. No internet connection required after installation.

The installer uses Podman for container management — rootless, daemonless, compatible with enterprise security policies that prohibit Docker. Each component runs in its own container. The installer handles orchestration, health checks, and restart on failure.

### Air-gapped operation

The installer is designed for air-gapped environments. All dependencies are bundled. Updates are distributed as versioned installer packages, not as online downloads. A plant with no internet connection can run ImBrain indefinitely and receive updates via USB or internal network share.

### Hardware requirements

Minimum for a single-plant deployment:

Component	Minimum	Recommended
CPU	4-core x86-64	8-core
RAM	8 GB	16 GB
Storage	256 GB NVMe SSD	512 GB NVMe SSD
OS	Windows 10 / Ubuntu 22.04 LTS	Ubuntu 22.04 LTS
Network	100 Mbit/s	1 Gbit/s

A \$500 industrial mini PC meets the minimum specification for a plant with up to 5,000 tags at 1-second resolution. Larger plants or higher tag counts require more storage; the CPU and RAM requirements are modest.

### Enterprise fleet management

Enterprise deployments spanning multiple sites require automated update management. ImBrain integrates with Ansible and Salt via a single command:

```
imbrain-ctl upgrade --version 2.3.0 --backup-before
```

The upgrade sequence — pre-flight checks, schema snapshot, graceful shutdown, package swap, migration, health check, automatic rollback on failure — is owned by ImBrain. The automation platform triggers the command and checks the exit code. Imbra.Soft publishes an official Ansible collection (`imbra.imbrain`) on Ansible Galaxy. Enterprise customers install it and supply their inventory and version pin.

Update freezes for maintenance windows or critical production periods:

```
imbrain-ctl upgrade --lock # block automated updates
imbrain-ctl upgrade --unlock # re-enable
```

---

## 8. Licensing

## AGPL for the core

ImBrain Core is licensed under the GNU Affero General Public License (AGPL). AGPL is a strong copyleft licence with a network use clause: any organisation that modifies ImBrain Core and offers it as a service to others must publish their modifications under AGPL.

This protects the open source investment. A cloud vendor cannot take ImBrain Core, build a hosted historian service, and keep their modifications proprietary. They must contribute back. Self-hosters and paying customers are unaffected — AGPL applies only to those who distribute or offer the software as a service.

## Commercial plugins

Analytics plugins — OEEPlugin, GoldenBatchPlugin, AnomalyDetectionPlugin, EnergyPlugin, SPCPlugin, StreamGatewayPlugin, ColdArchivePlugin, LLMPlugin, ForecastPlugin — are licensed commercially. Pricing is per site, per year. The plugin API is stable and versioned — plugins work across Core versions within a major release.

## The open core model

Component	Licence	Who pays
ImBrain Core	AGPL	Nobody — free to use and self-host
Official agents	AGPL	Nobody — free to use
Community agents	MIT	Nobody — free to use
Analytics plugins	Commercial	Paying customers
Support contracts	Commercial	Enterprise customers
Managed cloud	Commercial	Customers who prefer SaaS

This is the same model used by Grafana, Metabase, and GitLab. The core is open — it drives adoption, community, and trust. The plugins are commercial — they generate revenue. The boundary between them is the plugin API.

## 9. Use Cases

### Small and medium plant — first historian

A food and beverage plant with 800 tags, two production lines, and no existing historian. The plant manager has been running on Excel shift reports and cannot answer basic questions about OEE or batch quality trends.

ImBrain is installed on a \$500 mini PC in the control room. A Modbus agent connects to the two PLCs. A LabFileAgent picks up hourly CSV exports from the quality lab. Within a day, the plant has a live historian with 800 tags, full alarm history, and lab results. The OEEPlugin is activated. Within a week, the plant manager has OEE trends, downtime analysis, and the first golden batch comparison.

Total cost: hardware plus one year of OEEPlugin licence. No implementation partner. No multi-year project.

### Multi-site enterprise — mesh deployment

A chemicals group with 12 plants across three countries. Each plant has a different SCADA system, different PLC vendor, and different tag naming convention. The group wants cross-site OEE comparison, a unified alarm dashboard, and a natural language interface for the operations director.

ImBrain is deployed at each plant with site-specific agents. The mesh connects all 12 instances. The group-level ImBrain aggregates cross-site queries. The LLMPlugin is activated at the group level — the operations director asks “compare OEE across all European plants last quarter” and receives a structured report in seconds.

Fleet management via Ansible rolls out updates across all 12 plants in a controlled sequence — canary at one plant, then staged rollout across the rest.

## Energy and utilities — demand response

An aluminium smelter with 15,000 tags and significant electricity costs. The plant wants to participate in EU demand response programmes — reducing production when spot electricity prices spike, in exchange for compensation.

ImBrain stores electricity consumption tags at 1-second resolution alongside spot price data from an external feed. The EnergyPlugin calculates consumption benchmarks and identifies reduction opportunities. Imbra Pact (the companion coordination platform) evaluates contract conditions and coordinates production adjustments with the grid operator — with an on-chain audit trail for regulatory compliance.

---

## 10. Conclusion

The industrial historian market has not served its customers well. Enterprise platforms are priced for enterprise budgets and designed for vendor retention, not customer value. SCADA databases are siloed by design. The SME segment — the majority of the industrial economy — has been left with spreadsheets.

ImBrain is built on a different set of assumptions. Open core means the fundamental infrastructure is free and auditable. The plugin ecosystem means capabilities are modular and priced for what they deliver. The mesh architecture means cross-site visibility does not require a data warehouse project. The natural language interface means the historian is accessible to operators, not just engineers.

The result is a historian that a plant with a €2M IT budget can deploy in a day, and that an enterprise with 200 plants can run as a coordinated fleet. The same architecture, the same codebase, the same plugin ecosystem — from a \$500 mini PC to a global mesh.

Industrial data infrastructure should not cost six figures. It should not require a multi-year implementation. It should not lock customers into vendor ecosystems with no migration path. ImBrain is the alternative.

---

## References

- TimescaleDB — <https://www.timescale.com>
  - Sparkplug B specification — [https://www.eclipse.org/tahu/spec/sparkplug\\_spec.pdf](https://www.eclipse.org/tahu/spec/sparkplug_spec.pdf)
  - MQTT v5.0 specification — <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
  - OPC UA specification — <https://opcfoundation.org/developer-tools/specifications-unified-architecture>
  - pgvector — <https://github.com/pgvector/pgvector>
  - Podman — <https://podman.io>
  - AGPL v3 — <https://www.gnu.org/licenses/agpl-3.0.html>
  - Ansible Galaxy — <https://galaxy.ansible.com>
- 

© 2026 Imbra.Soft. All rights reserved. ImBrain and Imbra Pact are trademarks of Imbra Ltd. Imbra Connect is open source software licensed under the MIT Licence.