

# ACTIONABLE TIME SERIES ANALYTICS AND AI

## OVERVIEW

In most fin-tech, IoT, IT ops and security applications, data arrives as a time-based series of values and feeds real-time dashboards or triggers alerts. This type of data is called time series and it's been enabling enterprises to monitor changes in data over time for years. Up until now, most enterprises have settled for a reactive approach using traditional time series databases to visualize current trends and run batch analysis after the fact. However, modern businesses need to be proactive with sophisticated predictions and real-time actions which maximize the value of data. This requires new platforms which correlate time series data with multiple variables and large data volumes in real-time, run advanced AI algorithms, generate interactive dashboards and automate actions.

Building AI-driven, predictive and real-time solutions involves processing high volumes of incoming time series data in

different formats or protocols, enriching it in real-time with other variables taken from historical context, operational databases, social or environmental sources, feeding it into AI algorithms and serving the results back to users, dashboards or control systems. This process includes long and tedious integrations of various AI frameworks, data pipelines, legacy time series databases and/or expensive in-memory database solutions.

The Iguazio Continuous Data Platform (CDP) is the first integrated solution for simple delivery of real-time AI applications across cloud, on-premises and edge. With Iguazio's CDP users ingest large data volumes, add relevant historical or operational context, run AI tasks and serve the results in real-time without the complexity and high-costs of traditional solutions.

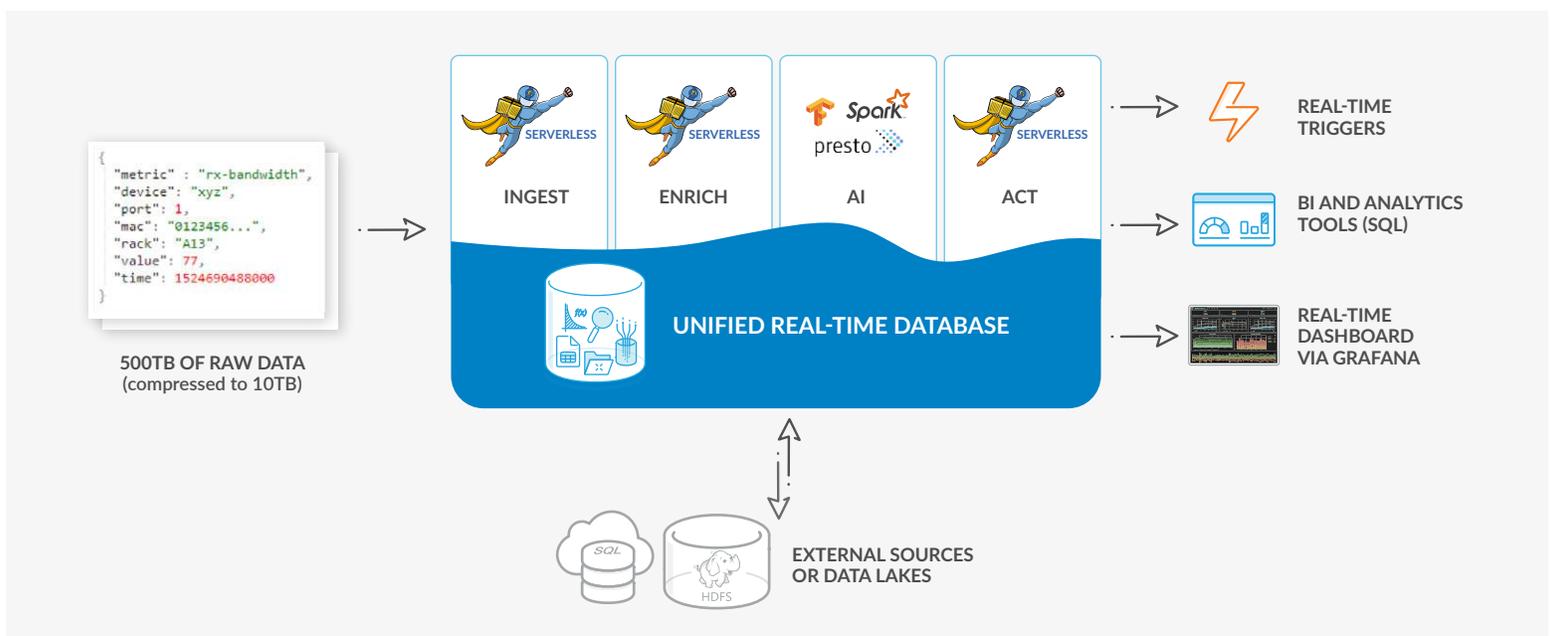


Figure 1 A variety of ingestion and query options

The Iguazio CDP integrates the following features:

- 1 Various standard and open data APIs: SQL, NoSQL (DynamoDB), time series (Prometheus), streaming (Kinesis), object (S3) and file. The data is accessed by multiple APIs simultaneously.
- 2 An innovative real-time database engine designed to reach the performance of in-memory solutions while using the lower cost and high-density Flash storage.
- 3 Open source AI/machine learning services (Spark, Python, TensorFlow) and real-time serverless functions (Nuclio) with low-latency data access.
- 4 End to end security, management and self-service operations in the cloud or edge.

## IGUAZIO TIME SERIES DATABASE (TSDB)

Iguazio's database engine is accessed using low-level APIs called V3IO and provides extreme performance, rich API semantics and multiple indexing services. It allows consistent data access from various workloads and processing engines simultaneously.

Iguazio's TSDB leverages advanced V3IO semantics (such as row and column layouts, array vectors, random and sequential indexes and complex expressions) to provide an extremely fast and scalable TSDB service. It is integrated with the following open-source frameworks to deliver a seamless development experience throughout different stages in the processing pipeline:

- [Prometheus](#) TSDB processor (for metrics scraping and queries)
- [Nuclio](#) serverless functions (for real-time ingestion, stream processing and queries)
- Restful APIs (DynamoDB, Grafana and custom)
- AI: Spark, Python, TensorFlow, Presto (SQL) and more
- Built-in CLI (tsdbctl) for database creation, ingestion and queries

Iguazio's TSDB is natively integrated with Prometheus (as a pluggable TSDB engine). Prometheus comes with a metric scraping layer and provides access to an endless number of plug-ins and clients for data acquisition ([see list](#)). It has an advanced time series query engine ([see link](#)) and alert manager.

Iguazio's TSDB adds the missing features to Prometheus making it the best TSDB solution in the market:

- 1 Horizontal scaling (Prometheus runs with local storage)
- 2 High speed push (ingestion) and streaming in addition to pull-based scraping
- 3 Support for Spark analysis and AI on the same data without copies (read as a Spark native dataframe with Spark SQL query semantics)
- 4 Accelerated queries through pre-aggregation and automatic rollups
- 5 High-availability, consistency, security and management

While traditional TSDBs are limited to a single data type, the Iguazio CDP supports multiple data models (time series, SQL/NoSQL table, document, stream, object, file). This allows real-time correlation of time series data with static and operational data tables for AI inferencing, as well as simplified deployment, security and maintenance.

Iguazio's TSDB works with The Nuclio real-time open source serverless engine. Nuclio supports ingestion from a variety of sources through HTTP or a large variety of streaming/triggering protocols (Kafka, Kinesis, Azure Event-Hub, RabbitMQ, NATS, Iguazio streams, MQTT and Cron tasks) and provides limitless auto-scaling and automatic deployment across cloud, edge and on-premises. Nuclio functions can be customized for pre-processing incoming data (examine metric data, alert and convert formats) to run real-time AI inferencing or handle post-processing (send notifications/triggers, write to external systems and provide custom query APIs).

## TSDB COMPARISON TABLE

	OpenTSDB	Cassandra	InfluxDB	ElasticSearch	Prometheus	Prometheus + Iguazio TSDB
<b>Consistency</b>	Yes	Eventual	No	Yes	No	Yes
<b>HA and scalability</b>	Yes	Yes	No	Yes	No	Yes
<b>Security</b>	No	Partial	Partial	Partial	No	Auth + RBAC
<b>Ingestion</b>	Push	Push	Push	Push	Poll	Push, Poll, Stream
<b>Query functionality</b>	✓	✓	✓✓	✓	✓✓	✓✓✓*
<b>Query perf</b>	Low	Low	High	Low	High	High
<b>Write Perf</b>	20K/s	20K/s	350K/s	30/s	800K/s	800K/s**
<b>Data models</b>	TS	Table (NoSQL)	TS	TS	TS	TS, table (SQL, NoSQL), obj, stream, data frame
<b>Bytes/sample</b>	12	100+	~2***	22	~2***	~2***
<b>Rollups</b>	No	No	Yes	No	No	Yes
<b>Functions (AI and triggers)</b>	No	No	No	No	No	Yes (Nuclio)

\* Supports the advanced PromQL, Spark Dataframe (for AI and SQL), Grafana, CLI, custom functions via Nuclio and Python Pandas (for AI)

\*\* Performance with micro-batch loading

\*\*\* Approx. 2 bytes/sample, using Gorilla XOR compression algorithm

### Sources:

[https://cdn2.hubspot.net/hub/528953/hubfs/Screen\\_Shot\\_2016-08-27\\_at\\_00.32.42.png](https://cdn2.hubspot.net/hub/528953/hubfs/Screen_Shot_2016-08-27_at_00.32.42.png)

<https://blog.netsil.com/a-comparison-of-time-series-databases-and-netsils-use-of-druid-db805d471206>

# TSDB ARCHITECTURE

The solution stores raw data in highly compressed column chunks (using Gorilla/XOR compression variation), with one chunk for every n hours (1hr default). Queries only retrieve and decompress specific columns based on the requested time range.

Users define pre-aggregates (count, avg, sum, min, max, stddev, stdvar, rate) which use V3IO update expressions and store data consistently in arrays per user defined intervals and/or dimensions (labels).

High-resolution queries detect pre-aggregates automatically and selectively access array ranges (skip chunk retrieval, decompression and aggregation), significantly accelerating searches and providing real-time response.

The data can be partitioned to multiple tables (e.g. one per week) or use a cyclic table (goes back to the first chunk after reaching the end) and multiple tables are stored in a hierarchy under a specified path.

Metric names and labels are stored in search optimized keys and string attributes. The Iguazio database engine can run a full dimension scan (searches) at the rate of millions of metrics per second or use selective range-based queries to access a specific metric family.

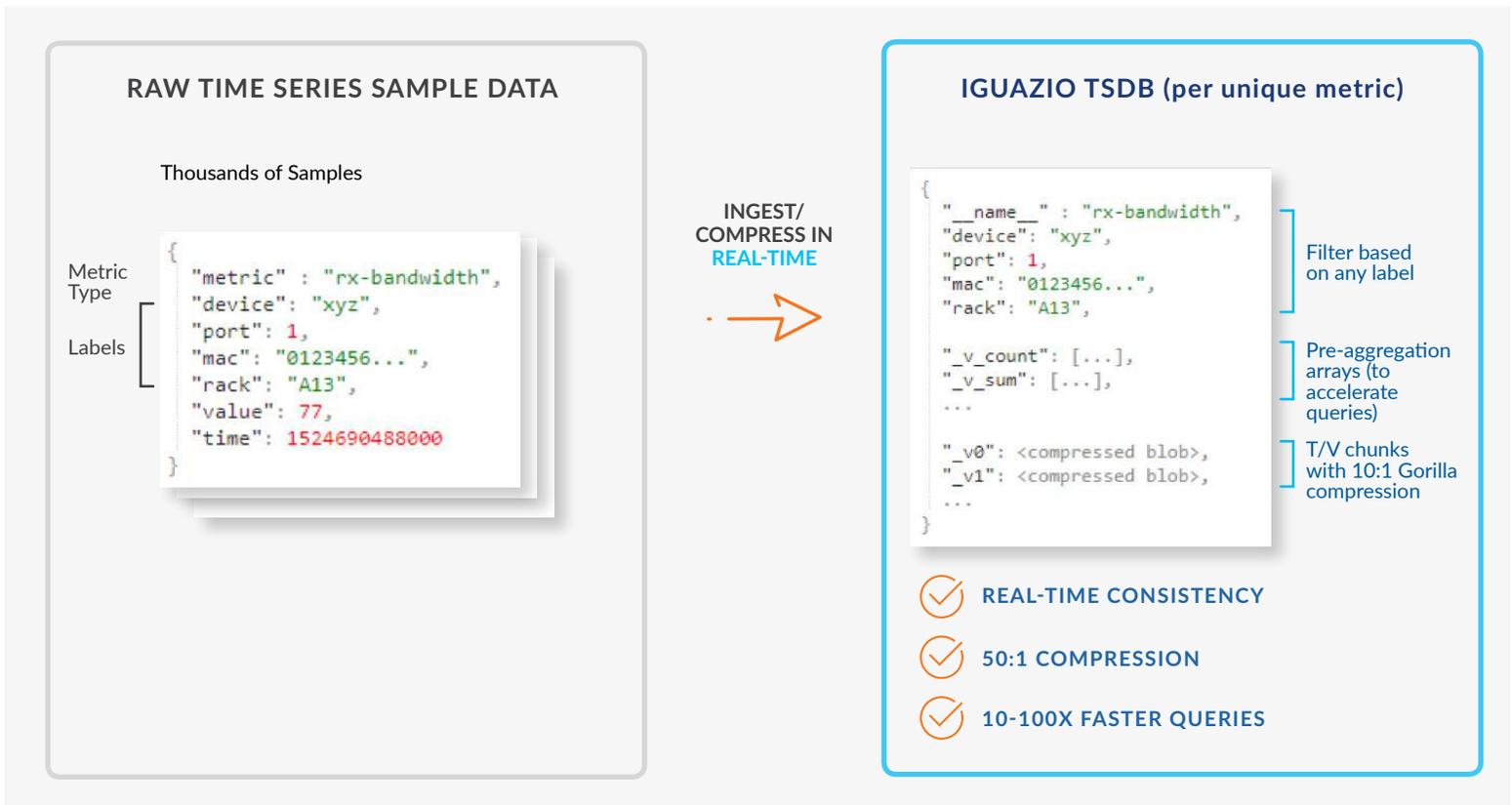


Figure 2 TSDB Internal Data Organization

The use of V3IO's random access keys (Hash based index) allows real-time sample data ingestion/retrieval and stream processing.

To maintain high-performance over low-speed connections, Iguazio implements auto IO throttling. If a link is slow, multiple samples are pushed in a single operation and users

configure the maximum allowed batch (trade efficiency with consistency). IO is performed using parallel connections/workers, enabling maximum throughput regardless of link latency.