Dataddo

TECHNICAL WHITEPAPER

Real-time CDC and database replication. Every change, at the speed of agents.

How Dataddo captures every committed change from Oracle, SQL Server and SAP S/4HANA, bulk-loads large tables in parallel from any supported database, and delivers both to warehouses, lakes, event streams and AI agents - in Dataddo Cloud or inside your own infrastructure with Dataddo Hybrid.

Published
October 2026
Dataddo

Dataddo

AT A GLANCE

Read the log, not the database. Load in parallel, not row by row.

SQL SERVER CDC

35,000

change events per second, sustained, at 351 ms median latency from source commit to visible to agent

ORACLE CDC

63,000

changes per second captured and drained (817,772 changes in 13 s), with no GoldenGate licence

BULK LOAD, STANDARD TABLE

22x

faster than bcp on SQL Server: 1.8 s per million rows on a 10-column table with 3 billion+ records

BULK LOAD, ULTRA-WIDE TABLE

5x

faster than bcp on SQL Server: 28.9 s per million rows on a 250-column table with 1 billion records

- **Change capture** for Oracle, SQL Server, PostgreSQL, MySQL, MongoDB, SAP HANA, SAP S/4HANA and more, with deletes and commit order preserved.
- **Mesh Ingestion™** parallel bulk loading for every supported database engine.
- **Two deployment options:** Dataddo Cloud, or Dataddo Hybrid with the data plane in your own cloud or data center.
- **Any destination:** warehouses, lakehouses, databases, object storage, event hubs, BI tools and AI agents over MCP.

Bring your own workload: the most convincing benchmark is the one run on your database.

Executive summary

AI agents, fraud checks and operational dashboards are only as good as the freshness of the data behind them. Batch replication leaves them a schedule behind and cannot see deleted rows. Self-built streaming stacks of Kafka, Debezium and connectors solve freshness, but they turn into a platform that someone has to run indefinitely.

Dataddo reads every committed insert, update and delete straight from each database's transaction log, keeps commit order, and delivers the changes to the destination of your choice as a managed service. The same platform bulk-loads large tables in parallel for initial loads and re-syncs, so one tool covers both ends: streaming change capture and high-volume full loads.

Workload	Best measured result
SQL Server change data capture	**35,000 events/s**, sustained 351 ms median, 568 ms p90 latency; internal benchmark
Oracle change data capture	**63,000 changes/s** 817,772 changes captured and drained in 13 s; customer proof of concept
Bulk load, standard table	**1.8 s per million rows**, 22x faster than bcp SQL Server to BigQuery; 10 columns, 3 billion+ records
Bulk load, ultra-wide table	**28.9 s per million rows**, 5x faster than bcp SQL Server to BigQuery; 250 columns, 1 billion records

Table 1. Best measured results per workload. Conditions are described in Section 5 and the appendix.

- **Architecture.** A control plane in Dataddo's cloud orchestrates data planes that do the extraction and loading. Rows never pass through the control plane (Section 1).
- **Deployment.** Dataddo Cloud runs everything as a managed service. Dataddo Hybrid runs the data plane inside your own cloud or data center, so records and credentials stay in your perimeter (Section 2).
- **Coverage.** Native change capture per engine, including SAP S/4HANA (Section 3), per-table replication methods with parallel bulk loading for every database engine (Section 4), and delivery to warehouses, lakehouses, databases, object storage, event hubs, BI tools and AI agents (Section 6).
- **Agent-ready.** The same changes can feed a governed semantic layer that AI agents query over the Model Context Protocol (MCP), with a median of 351 ms from source commit to visible to agent (Section 7).

1 Architecture: how it works

The platform separates deciding from doing. A control plane in Dataddo's cloud handles configuration, orchestration, scheduling and monitoring. Data planes do the extraction, transformation and loading, and run where the data lives. Only instructions go down to a data plane and only status, metrics and metadata come back; rows never pass through the control plane.

Change capture reads each engine's own transaction log and records its position in that log. Every committed insert, update and delete arrives in commit order, and after any interruption the pipeline resumes from the last recorded position without losing a committed change.

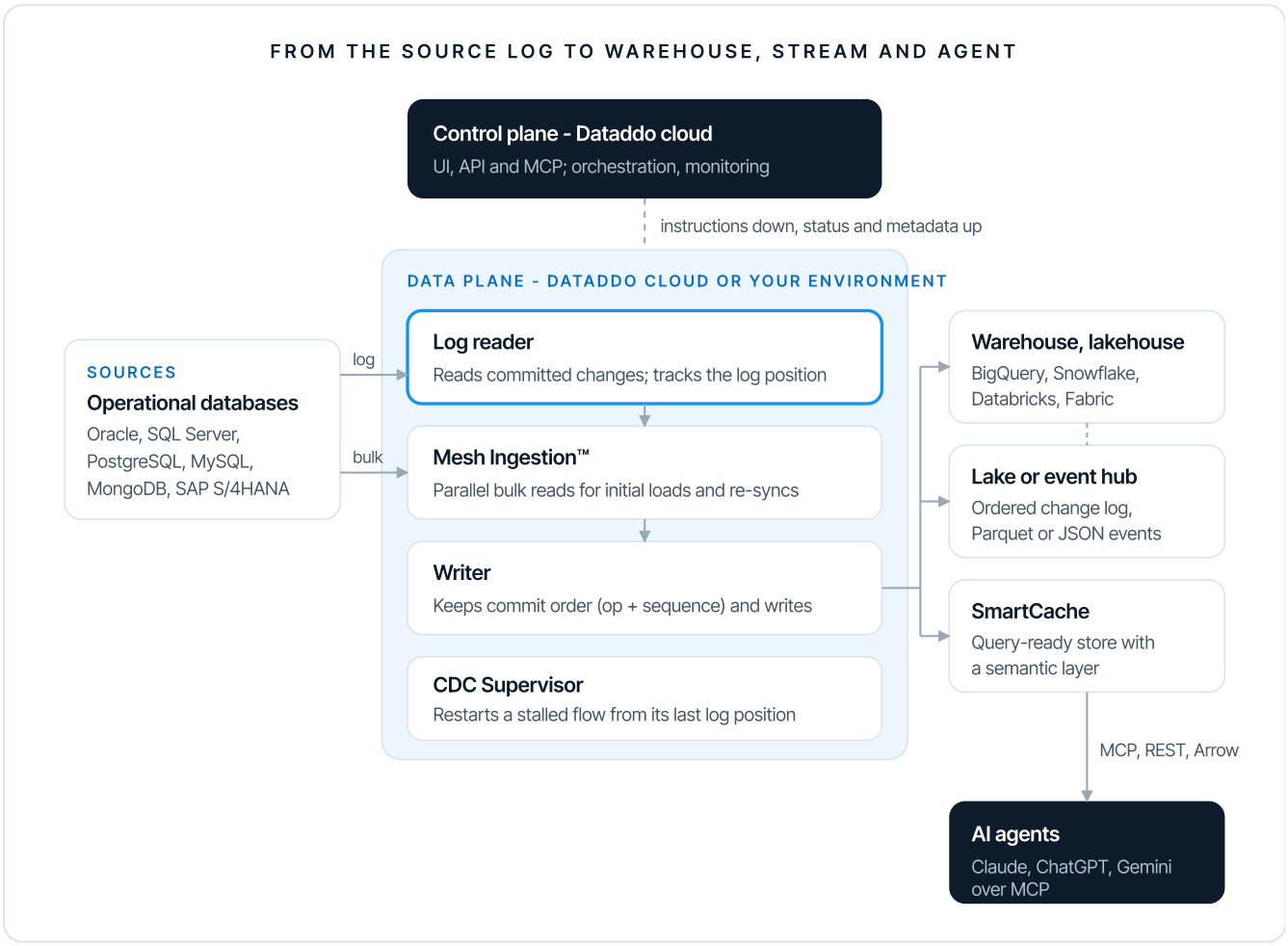


Figure 1. Dataddo CDC and replication architecture. In Dataddo Hybrid, everything inside the data plane runs in your environment.

Component	Runs in	What it does
Control plane (UI, API, MCP)	Dataddo cloud	Configuration, orchestration, scheduling, monitoring and governance; receives status, metrics and metadata only
Log reader	Data plane	Reads binlog, WAL, SQL Server CDC tables, Oracle MLOG or LogMiner, or the MongoDB oplog, and tracks the log position
Mesh Ingestion™	Data plane	Splits a table into primary-key chunks for 4 to 64 parallel workers; used for initial loads, backfills and re-syncs
Writer	Data plane	Keeps commit order and writes as a materialized table (CDC write mode), as an ordered change log, or as upsert plus delete
CDC Supervisor	Data plane	Checks every running process and restarts a stalled or failed one from its persisted log position
Data Quality Firewall	Data plane	Harmonizes data types and blocks bad records on the way in
SmartCache and semantic layer	Data plane	Optional deduplicated store for recovery, reprocessing and AI delivery, with defined entities and metrics for agents

Table 2. Platform components. The platform is built as Golang microservices on Kubernetes and is managed through the UI, a REST API or Terraform.

2 Deployment: Dataddo Cloud and Dataddo Hybrid

Dataddo is deployed in one of two ways. Both use the same control plane in Dataddo's cloud; the difference is where the data plane runs. One control plane can orchestrate several data planes at once, so sensitive databases can be processed in your own environment while cloud sources run in Dataddo Cloud, all under one interface.

Option	Control plane	Data plane	Best for
Dataddo Cloud	Dataddo cloud	Managed by Dataddo on AWS, Azure or Google Cloud, or on EU providers Hetzner, OVHcloud and STACKIT, across 16 residency locations	Cloud databases and warehouses; fastest start
Dataddo Hybrid	Dataddo cloud	Your own cloud account or on-premises data center, on Kubernetes, Red Hat OpenShift, VMware Tanzu or bare metal	Databases behind a firewall; regulated data that must stay inside your perimeter

Table 3. The two deployment options.

What Dataddo Hybrid keeps inside your perimeter

- **Records and credentials stay with you.** Instructions and schedules go down to the data plane; status, metrics and metadata go up. Rows never leave your environment except to the destinations you configure, and credentials stay in your vault.
- **No agents on the database.** Connectivity uses reverse SSH or VPC tunneling. Nothing is installed on the source database, and no SDK or polling infrastructure is needed.
- **Simple prerequisites.** A Kubernetes cluster, plus outbound connections to Dataddo's repositories (for automated updates) and to the control plane. A proof of concept can run on a single virtual machine.
- **SmartCache inside your environment.** When SmartCache is used in Hybrid, it runs in your environment as well, and pipelines can be configured to retain no data at all.

Security and reliability

Area	What is included
Certifications	SOC 2 Type II and ISO/IEC 27001
Encryption	AES-256 at rest and TLS in transit, with KMS integration for AWS, Azure and Google Cloud
Access control	SSO (Google, Microsoft Entra ID, Okta), role-based access control, MFA, permission templates and separate dev, staging and production environments
Observability	Flow-level monitoring with Grafana, Prometheus and OpenSearch; alerts to Slack, PagerDuty, Jira, ServiceNow or webhooks; external orchestration with Airflow
Recovery	Disaster recovery with an RPO under 15 minutes and an RTO under 1 hour

Table 4. Security, observability and recovery.

3 Change capture per engine

Each engine is read through its own native change log. Enabling it is a one-time database setting; everything after that is managed. Every replicated table needs a primary key or a unique column.

Engine	Capture mechanism	Position tracked	One-time setup
Oracle	Materialized view log (MLOG) with managed purge; LogMiner as an alternative for 11g to 19c	SCN	MLOG on tracked tables, or LogMiner access
SQL Server	CDC tables; or transaction-log processing, from log backups (1 to 15 min) or active log polling (1 to 5 s)	LSN	CDC enabled on the database and tables
PostgreSQL	WAL logical replication	Replication slot	wal_level = logical and a free slot
MySQL	Binary log (binlog)	Binlog position	Binlog enabled
MongoDB	Oplog	Resume token	Read access to the change stream
SAP S/4HANA	SAP Operational Data Provisioning (ODP): changes are queued in the Operational Delta Queue and read through the official OData API; no direct HANA connection	Delta token	S/4HANA 1909 FPS01 or later (Cloud 1905 or later); ODP extraction services exposed on SAP Gateway

Table 5. Native capture by engine. Log-based CDC is also available for SAP HANA, Amazon Aurora and RDS, Google Cloud SQL, AlloyDB and CockroachDB.

On SAP S/4HANA, the first sync performs a full initial load and arms change tracking in the delta queue; from then on only inserts, updates and deletes are transferred. Activating ODP adds database triggers on the base tables, so each write carries a small overhead on the SAP side.

Every change event carries its operation type (insert, update or delete) and a sequence identifier from the engine's log, such as the SCN on Oracle or the LSN on SQL Server. Consumers can therefore order changes deterministically and materialize the current state of each row, even with at-least-once delivery. Log-based capture detects deleted rows while asking practically nothing of the production database: no triggers, no extra polling queries and no shadow tables.

Oracle: CDC without GoldenGate

Oracle desupported continuous LogMiner mining in 19c, and its streaming alternative, XStream, requires a GoldenGate licence. Dataddo supports both of the remaining native paths and manages their full lifecycle, so teams can choose by workload rather than by licence.

	MLOG (materialized view log)	LogMiner
Reads from	A change log on each tracked table	Redo logs
Oracle support	Native and supported in current releases	Continuous mining desupported from 19c; polled at a configurable interval
Selectivity	Only the selected tables, rows and columns	Whole redo stream, filtered by Dataddo
Source impact	Adds work to each tracked transaction; sized to the change rate, with the log kept near zero by continuous draining	Lower; reads logs after commit
Watch for	DROP PARTITION produces no MLOG rows	Large CLOB columns; quoted identifiers

Table 6. Oracle capture options. Many teams start with LogMiner where it is available and keep MLOG as the future-proof path.

4 Replication methods and bulk loading

Not every table needs streaming. Each table can use the sync method its change pattern needs, and methods combine within one pipeline, for example query-based upserts with log-based deletes. Flows for every table in a database can be generated automatically.

Method	New rows	Updates	Deletes	Source impact	Best for
Log-based CDC	Yes	Yes	Yes	Very low	High-volume, real-time sync
Timestamp	Yes	Yes	No	Low to medium	Frequently edited tables
Row sequence	Yes	No	No	Low	Append-only logs and transactions
Custom SQL	Depends	Depends	Depends	Variable	Complex filters and joins

Table 7. Replication methods, selectable per table.

- **Mesh Ingestion™ for bulk loads, on every database engine.** A table is split into primary-key chunks and read by parallel workers, 4 by default and up to 64. Throughput scales with workers until the source or the network saturates; about one vCPU per worker is a good starting point.
- **Write modes.** Insert, upsert, delete and truncate-insert, with automatic schema creation in the destination.
- **Backfill.** CDC captures changes from its starting position; historical rows come from a one-time bulk load, after which the change stream takes over without a gap.
- **Source coverage.** Pointer-based incremental replication also covers Snowflake, Amazon Redshift, Databricks, ClickHouse, CockroachDB, Azure SQL, IBM Informix, Firebird and Vertica.

5 Performance: measured throughput

Agents that act on data need changes within a second, not after the next batch. On SQL Server, Dataddo's CDC engine sustained 35,000 change events per second with a median of 351 ms from source commit to visible to agent.

THE KEY DIFFERENCE - STREAMING SPEED (SQL SERVER CDC, INTERNAL BENCHMARK)

351 ms
median latency (p50), from source commit to visible to agent

568 ms
90th percentile latency (p90)

35,000
change events per second, sustained

For bulk loads, row width matters as much as row count. Mesh Ingestion™ was measured on two SQL Server tables loaded into Google BigQuery: a standard 10-column table with more than 3 billion records, and an ultra-wide 250-column table with 1 billion records. The baseline is the same tables exported with Microsoft's bcp utility.

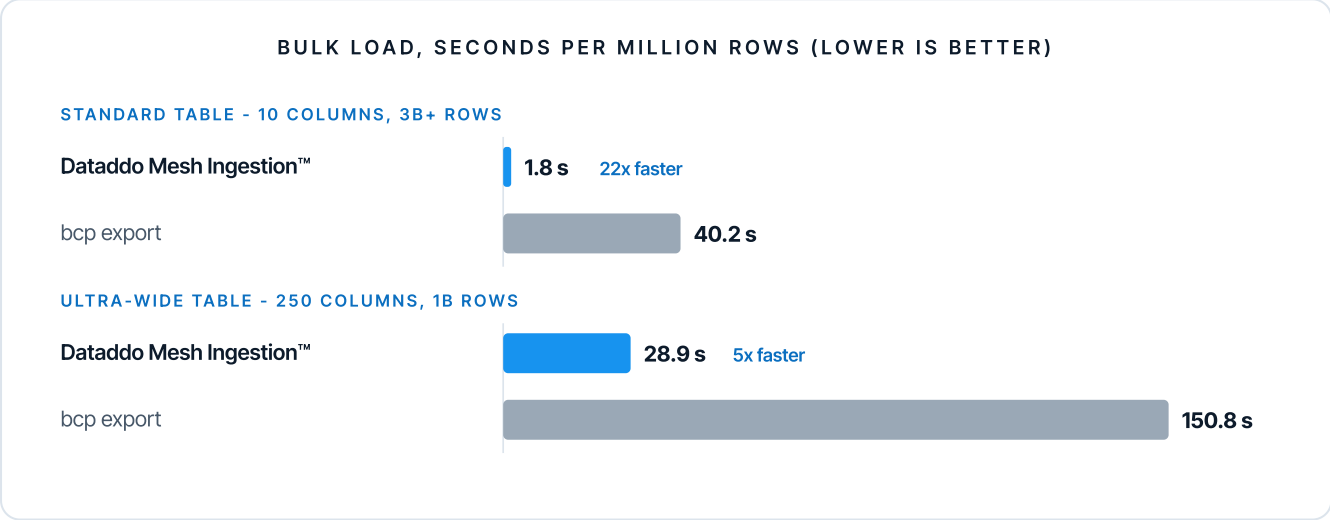


Figure 2. Average time to transfer one million records, SQL Server to Google BigQuery. Mesh Ingestion™ is available for every supported database engine.

Change data capture

Test	Throughput	Latency
SQL Server CDC Internal benchmark	35,000 events/s, sustained	351 ms p50, 568 ms p90
Oracle MLOG, capture and drain Customer proof of concept	817,772 changes in 13 s (about 63,000/s)	Not measured

Table 8. Change data capture results. The Oracle test used a synthetic load generator writing inserts, updates and deletes on a production-class Oracle system.

Bulk loads with Mesh Ingestion™

Table	Shape	Dataddo	bcp export
Standard table SQL Server to Google BigQuery	10 columns, 3 billion+ records	**1.8 s** per million rows	40.2 s per million rows (22x slower)
Ultra-wide table SQL Server to Google BigQuery	250 columns, 1 billion records	**28.9 s** per million rows	150.8 s per million rows (5x slower)

Table 9. Bulk load results with Mesh Ingestion™.

ROW WIDTH

A 250-column row carries about 25 times the columns of a 10-column row, yet took about 16 times as long per million rows. Wide tables gain less from parallel reads than narrow ones, but still loaded five times faster than the bcp baseline.

6 Supported destinations

Any supported source can feed any supported destination. For change data capture, warehouses and databases receive a materialized table that mirrors the source, including deletes; lakes and event hubs receive the ordered change log.

Category	Destinations	How changes land
Cloud warehouses and lakehouses	Google BigQuery, Snowflake, Databricks, Amazon Redshift, Microsoft Fabric (OneLake), Azure Synapse, Azure SQL Database	Materialized table (CDC write mode) or upsert plus delete
Databases	SQL Server, PostgreSQL, MySQL, MariaDB, Oracle, IBM Db2, CockroachDB, AlloyDB, Amazon RDS (MySQL, PostgreSQL, MariaDB), Google Cloud SQL (MySQL, PostgreSQL)	Materialized table (CDC write mode) or upsert plus delete
Object storage and data lakes	Amazon S3, Azure Blob Storage, Google Cloud Storage, SFTP; Parquet, CSV, JSON and JSONL files; Apache Iceberg tables through a REST catalog, including AWS Glue	Ordered change log
Event streaming	Azure Event Hubs, with full-row JSON events carrying the operation type and sequence number	Ordered change log
AI and agents	Any MCP client, including Claude, ChatGPT, Gemini, Cursor and Le Chat; SmartCache over REST or Apache Arrow	Through SmartCache and the semantic layer
BI tools	Power BI, Tableau, Looker Studio, Databox	Scheduled sync
Business applications	Salesforce, HubSpot, Klaviyo, Exact Online (reverse ETL)	Scheduled sync

Table 10. Destinations by category. New database destinations are typically added within one to two months on request.

7 From change to agent

A fresh replica is only half of what an agent needs; the other half is meaning. Agents consume Dataddo's change stream three ways: by querying the warehouse or lake it keeps current, by subscribing to ordered change events, or by reading SmartCache directly over MCP, REST or Apache Arrow. Over MCP, agents see defined entities, fields and metrics with plain-language descriptions instead of raw tables, and never send SQL to the source.

Path	How the agent gets the change	Best for
Warehouse or lake	Queries the materialized table that CDC keeps current, including deletes	Analytical agents and BI on the same data
Change events	Subscribes to the ordered change log on an event hub or in object storage	Agents that react to each change as it happens
SmartCache over MCP	Asks in business terms through the semantic layer; Dataddo resolves the query against a deduplicated store	Conversational agents in Claude, ChatGPT, Gemini and other MCP clients

Table 11. Three ways agents consume the change stream. On SQL Server, a change is visible to the agent a median of 351 ms after it is committed at the source.

8 Appendix: methodology

Benchmark	Method	Scope and limits
SQL Server CDC latency and throughput	Time from commit at the source to the change being visible to the consumer, under a sustained load of 35,000 change events per second. Percentiles: 351 ms p50, 568 ms p90, 1,117 ms p99.	Internal benchmark on Dataddo's SQL Server CDC engine; results on other engines, hardware and networks may differ.
Oracle CDC throughput	A synthetic load generator wrote inserts, updates and deletes to a production-class Oracle system; MLOG capture and drain of 817,772 changes was timed at 13 seconds.	Customer proof of concept; short burst, throughput only, end-to-end latency not measured. MLOG adds load on the source in proportion to the change rate.
Bulk loads	Average time to transfer one million records from SQL Server to Google BigQuery with Mesh Ingestion™, on a 10-column table (3 billion+ records) and a 250-column table (1 billion records). Baseline: the same tables exported with bcp.	Results depend on row width, worker count, network and source capacity. The bcp baseline covers export only.

Table 12. Benchmarks cited in this paper.

This whitepaper was produced by Dataddo. Product capabilities and certifications are as published on dataddo.com in October 2026 and may change. Performance figures come from Dataddo benchmarks and a customer proof of concept; results depend on hardware, network, schema and change patterns. Third-party product names belong to their owners and are used to identify supported systems.

Test it on your own database

Bring your own workload. We run change data capture and bulk loads against your own database, in Dataddo Cloud or inside your perimeter with Dataddo Hybrid, and show you the numbers.

Book a proof of concept

Read the CDC docs