CONFLUENT

WHITEPAPER

Unlocking Real-Time Insights From Oracle: A Technical Deep Dive Into Confluent's Oracle XStream CDC Source Connector

Table of Contents

Introduction

Oracle databases hold critical enterprise data, but traditional batch ETL processes create delays, hindering real-time applications like fraud detection. Confluent's Oracle XStream CDC Source Connector bridges this by leveraging Oracle's native XStream technology to stream real-time data changes, transforming Oracle into a dynamic source of real-time intelligence.

This white paper provides a technical deep dive into the connector's architecture, capabilities, and strategic advantages. It will demonstrate how this solution delivers the following.

Unmatched Performance: By using the XStream API, the connector achieves a 2-3x improvement in both throughput and latency compared to traditional change data capture (CDC) approaches such as LogMiner.

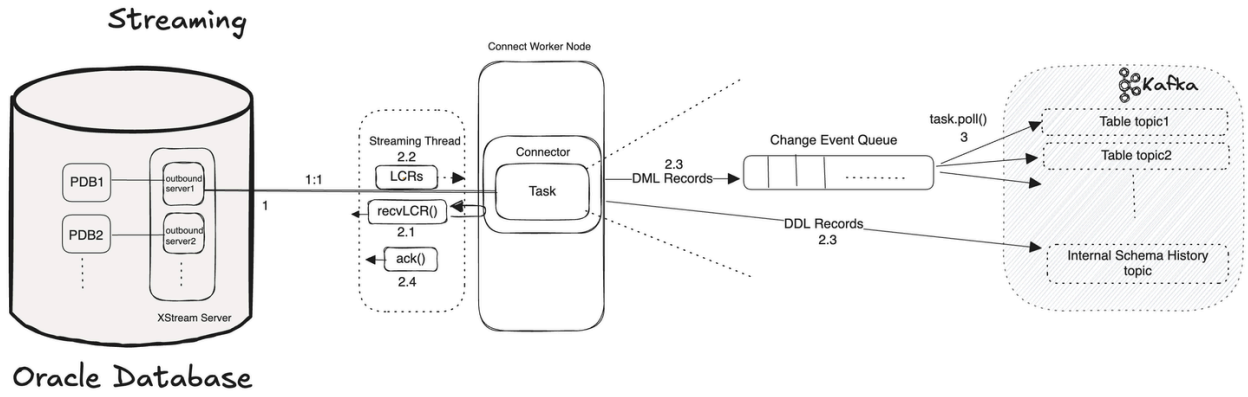
Best-in-Class Architecture: The connector facilitates a shift from inefficient, polling-based data capture to a true, push-based streaming model, minimizing performance impact on the source Oracle database.

End-to-End Governance: The connector works seamlessly with Stream Governance features such as Schema Registry and Stream Lineage, providing a complete, auditable framework for managing data products.

A Modern Architecture: The Power of Oracle XStream

Oracle developed XStream, a feature set purpose-built for high-performance, real-time data integration. The Confluent Oracle XStream CDC Source Connector leverages this native capability, providing a robust foundation for data streaming. The connector uses the XStream Out API, in which a **capture process** within Oracle reads changes from redo logs and converts them into **Logical Change Records (LCRs)** and an **outbound server** pushes them directly to the connector.

This push-based architecture is fundamentally superior, ensuring minimal latency and significantly lower impact on the source database compared to LogMiner's pull-based model. By building on XStream, the Confluent connector transforms the Oracle database into a first-class citizen of a modern data architecture. The stream of change events is published once as a reusable, governable data product on Confluent, which can then be consumed by any number of downstream applications without ever querying the source Oracle system again.



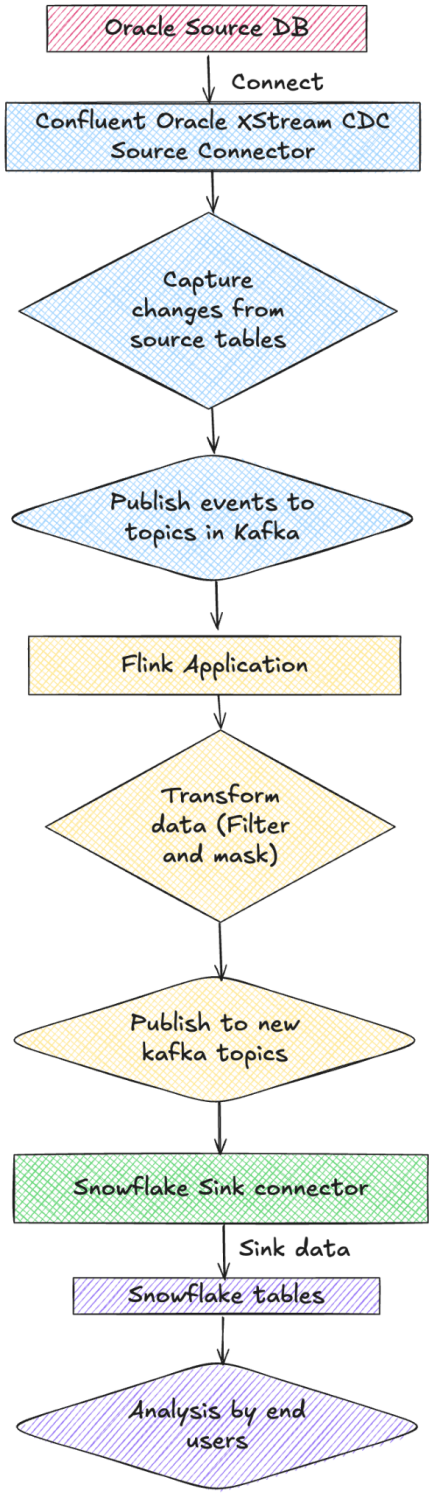
Solution in Practice: Real-Time Disease Trend Analysis

Consider a healthcare provider aiming to identify disease trends in near-real-time from its records system, which is based on Oracle Database (DB). Its goal is to move beyond reactive, weekly public health reports and proactively identify emerging disease trends in near-real-time to better allocate resources and inform the public. The solution involving the new Oracle XStream CDC Source Connector and other products, such as Apache Flink®, would enable the customer to build out this use case.

The end-to-end data flow is as follows:

1. **Source (Oracle DB):** Connect to the Oracle source database through the Confluent connector.
2. **Capture (Confluent's Oracle XStream CDC Source Connector):** This connector captures every change from the PATIENT_ENCOUNTERS table in real time.
3. **Stream (Confluent Cloud):** The raw change events are published to a durable, scalable Apache Kafka® topic (oracle-cdc-patient-encounters), creating an immutable source of truth.
4. **Transform and Govern (Confluent Cloud for Apache Flink®):** A serverless Flink application on Confluent Cloud subscribes to the raw topic to perform streaming ETL. It filters for relevant communicable diseases and masks all personally identifiable information (PII) to comply with HIPAA regulations.
5. **Load (Confluent Snowflake Sink connector):** The cleaned, anonymized data is published to a new topic (analytics-ready-disease-trends) and loaded continuously into Snowflake by the managed sink connector.
6. **Analyze (Snowflake):** Epidemiologists can now build dashboards and run queries on data that is mere seconds old, enabling a rapid, data-driven response to public health events.

This end-to-end flow is visualized below:



Key Connector Capabilities for Enterprise Workloads

The Confluent Oracle XStream CDC Source Connector is designed with enterprise-grade features to handle the complexities of production environments.

Handling Long-Running and Large Transactions

Large batch jobs and long-running transactions challenge other CDC methods, risking out-of-memory errors and high latency. The XStream architecture supports such transactions, intelligently spilling large transactions to disk-based buffer queues within Oracle. This allows it to process massive transactions upon commit without consuming excessive connector memory or blocking the real-time stream, a critical differentiator for enterprise workloads.

Seamless Schema Evolution With DDL Support

Database schemas evolve. When a database administrator adds a new column, less robust CDC solutions can break. Confluent handles this seamlessly. The Oracle XStream Source Connector captures data definition language (DDL) changes (e.g., ALTER TABLE) and, through integration with Confluent Schema Registry, automatically creates and registers a new schema version. Downstream consumers detect the new version and adapt without code changes, ensuring that the data pipeline is resilient to change.

Data Governance for Compliance

Confluent, with Flink, offers a secure-by-design approach in which PII is masked *in-flight*, meaning raw sensitive data is never persisted in the analytics platform. For certain industries—for example, healthcare—proving compliance (HIPAA) is non-negotiable and the governance capabilities would be very useful. The connector also supports client-side field level encryption (CSFLE) so that data is encrypted at the client side itself, thus ensuring added safety of customer data.

Change Event Structure

When dealing with update operations, the connector captures not just the final state of a record after an update but also its state before the update occurred. This comprehensive capture of both the "before" and "after" images of a change provides invaluable context for a wide range of use cases, including audits, compliance, and others.

Native Support for Oracle RAC

The connector, along with the XStream API, is purpose-built for large production workloads, which usually have Real Application Clusters (RAC) deployed.

Support for Large Objects

The connector natively supports large objects like BLOB, CLOB, and NCLOB. These are treated the same as other records and are processed/stored in the same table topic.

The Confluent Advantage

The choice between Confluent and traditional tools represents a fundamental architectural decision. The Confluent connector is based on Oracle XStream, which is among the best-in-class solutions for getting change data from Oracle. Other tools use slightly older technology and don't give similar benefits.

Feature	Confluent Streaming With XStream API	Other Oracle CDC Solutions
Data Latency	Sub-second to seconds; true real-time streaming	Minutes to hours; higher latency syncs
Source DB Impact	Minimal; push-based architecture via native XStream API	Higher; relies on querying Oracle's LogMiner tech or relies on reverse engineering the logs
DDL Handling	Seamless evolution governed by Schema Registry	Fragile; many DDL changes trigger costly full-table resyncs
Transformation	Powerful, stateful in-flight processing with Apache Flink	Post-load transformations within the data warehouse
Data Reusability	High; a single stream serves infinite consumers	Low; requires a new pipeline for each new use case
Important Things to Consider	Requires Oracle-side configuration of XStream	Strict limits on table/column names, unsupported data types, and row counts without encountering lag

Deployment and Monitoring

The connector is available both as a self-managed option and as fully managed across AWS, Azure, and Google Cloud Platform. This provides flexibility in how the connector is deployed. Deployment involves three main steps:

1. **Oracle Database Prerequisites:** Prepare the Oracle database by enabling ARCHIVELOG mode, activate XStream replication, and create the necessary users and privileges.
2. **Connector Installation (Confluent Platform):** Install the connector plugin onto Confluent Connect workers and add the required Oracle JDBC and XStream client libraries.
3. **Connector Configuration:** Deploy the connector with a JSON configuration specifying database connection details, the XStream outbound server name, and tables to include.

Please refer to the official documentation for detailed steps on how to deploy the connector.

Once deployed, the connector's health can be monitored via available metrics, providing visibility into snapshot progress, streaming latency, and schema history status.

Metric Group	Key Attributes	What It Tells You
Snapshot Metrics	SnapshotRunning, RemainingTableCount, RowsScanned	Progress and health of the initial data load
Streaming Metrics	Connected, MilliSecondsSinceLastEvent, MilliSecondsBehindSource	Real-time health, latency, and throughput of the CDC stream
Schema History Metrics	Status, ChangesApplied, LastAppliedChange	Health of DDL processing and schema evolution

Performance Profile: Efficiency and Compliance at Scale

XStream consistently and significantly outperforms LogMiner across various TPROC-C datasets that were used for comparison, demonstrating substantially higher efficiency, lower latency, and greater stability. The workload consisted of short/small transactions as well as long-running transactions to mimic real-world scenarios.

Key Comparative Insights

- **Throughput and Efficiency:**
  - With long-running transactions, XStream has a performance advantage, with its throughput being 209% higher than LogMiner's. In this scenario, LogMiner's own performance degraded by more than 31% from its baseline.
- **Latency and Real-Time Capability:**
  - XStream operated in near-real-time with a negligible lag of under two seconds.
  - LogMiner could not achieve real-time processing, immediately showing a 12-minute lag. This lag **worsened by 83%** when handling long-running transactions, while XStream's latency remained unaffected.
- **Resource Management and Stability:**
  - The Oracle XStream CDC Source Connector requires significantly smaller memory allocation. For our testing, we allocated ~80% less memory to the connector.
  - Under a large transaction workload, the Oracle XStream CDC Source Connector handled the load with ease, using only **62.5%** of its much smaller memory allocation. This means the LogMiner connector required **640% more memory** than XStream for the same task, highlighting a critical difference in resource efficiency and stability.

Conclusion

Unlocking the value of legacy Oracle databases requires a shift from high-latency batch processes to a true data streaming platform. The Confluent Oracle XStream CDC Source Connector is the cornerstone of this modern architecture, providing a solution that is not only 2-3x faster than traditional alternatives but is also architecturally superior, reliable, and cost-effective. By integrating data movement with in-flight processing and end-to-end governance, Confluent empowers organizations to transform their Oracle data from a static repository into the central nervous system of their event-driven enterprises.

confluent.io/connectors