

Augmenting CDC functionality using REST APIs and Event Streaming

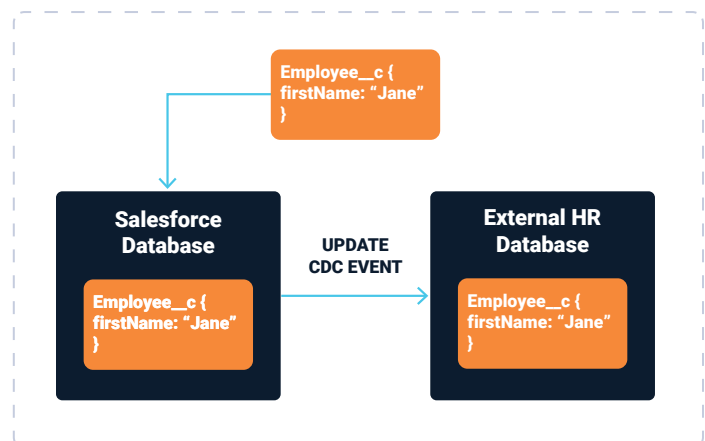
Overview

Change Data Capture (CDC) catches changes made in a database at the row level, and replicates them to other databases. This is done in one of two ways: either each change is itself tracked, or changes are kept in a separate data store (which is kept up to date with the primary one but does not track intermediate states). CDC pipelines can be useful for data replication, such as to a Data Fabric warehouse, or for ETL (Extract, Transform, Load) jobs. CDC functionality can be augmented by the addition of REST APIs and Event Streaming.

How are CDC events produced?

CDC events are generated by the database. The two most common ways are:

- **Read from the write-ahead log:** Captures changes asynchronously while providing low-latency.
- **Use database triggers:** Writes changes synchronously to a specific storage table in the database, providing low-latency but sometimes affecting performance.



CDC events notify subscribers when there are changes in a database. It then picks up the delta of the changes without the need for additional logic.

What CDC Misses

- **Context:** CDC does not track why a change occurred
- **Updates to the system of record (SoR):** When new changes come in, the system can't update both the SoR and data fabric simultaneously, resulting in delays
- **Efficiency:** Because CDC updates an entire row with each change, minor changes are costly, and millions of updates in a short timeframe means slowing to a crawl
- **Speed:** Every change gets updated, but changes take time before they propagate to the database—your data may not be fresh

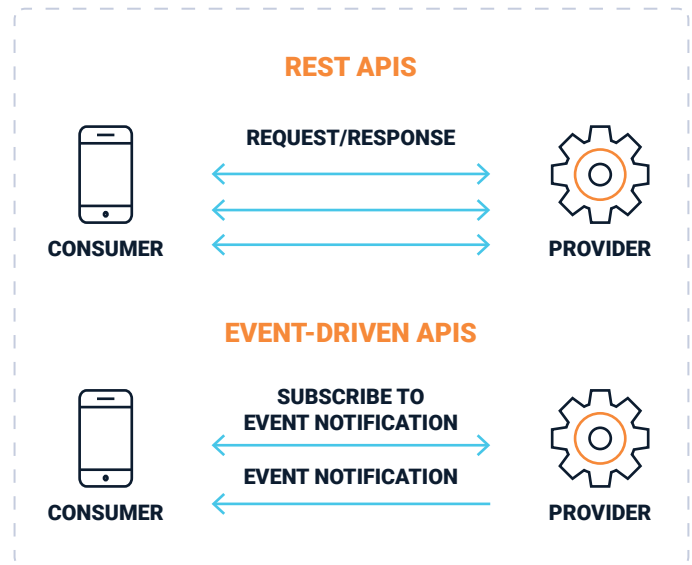
How REST and Event-Driven APIs can help

REST APIs provide a clean interface to both the data and logic of a system. REST APIs don't track or notify when changes occur. Instead, the consumer application makes a request, which the API passes to the provider, triggering the provider to run logic. The logic might write data to a database, retrieve data, or answer a question, and the API returns the result to the consumer. Typically, REST APIs are built inside microservices.

Event-driven APIs, for event streaming, are designed around a publish-subscribe system. The consumer subscribes for notifications. The Provider sends the notification as an event, either periodically or triggered by a need or action.

A hybrid model extends your business

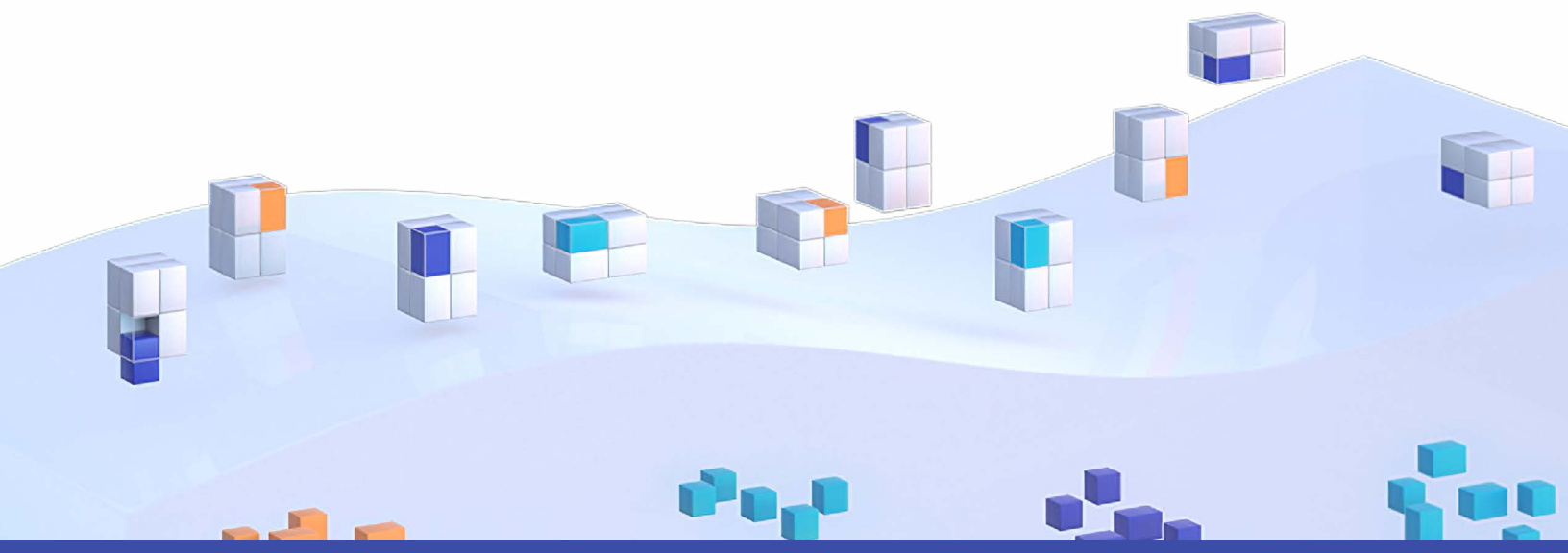
CDC processes, REST APIs, and Event-Driven APIs each provide a different type of access geared towards specific use cases. Full business functionality results from leveraging the type of access best suited for each purpose. OpenLegacy's platform works seamlessly with CDC functionality, and provides you with the ultimate flexibility for both REST API and Event Streaming use cases, augmenting any CDC solution.



What is a Data Fabric?

A data fabric is a design concept for attaining reusable and augmented data integration services, data pipelines and semantics for flexible and integrated data delivery.

Gartner



This table captures why businesses should consider a hybrid model:

Capability	CDC	REST APIs	Event Streaming (e.g. Kafka)
Track data changes	Yes. The main strength of CDC	No. You could build in tracking but it isn't part of the API	No
Sync database	Yes	No	No
Optimize the amount of data in an update	Limited. Updates a whole row of data, which can be fairly large even for a small change	Yes. Build APIs to update only the data you actually need	Yes. Configurable to say exactly what data gets streamed
Leverage SoR logic	No	Yes. Bi-directional support designed to update the SoR	No
Work asynchronously	Yes	Yes	Yes
Build a digital system with access to SoR	No. Not designed to build digital system, just to update a digital database	Yes	No. Updates the digital system but can't contact the SoR
Stream data to other systems quickly	No. Goes to another database, not directly to other systems	Limited. Makes data available but only if a request comes through the API	Yes. Designed so other systems subscribe and receive the data as required
Allow SoR logic access to external information	No. Only designed to allow other systems access to SoR data	Yes. Allows SoR logic to call external REST APIs	No. Just streams data out, isn't designed to return information back to a core system

About OpenLegacy

OpenLegacy's Digital-Driven Integration enables organizations with legacy systems to release new digital services faster and more efficiently than ever before. It connects directly to even the most complex legacy systems, bypassing the need for extra layers of technology. It then automatically generates APIs in minutes, rapidly integrating those assets into exciting new innovations. Finally, it deploys them as standard microservices or serverless functions, giving organizations speed and flexibility while drastically cutting costs and resources. With OpenLegacy, industry-leading companies release new apps, features, and updates in days instead of months, enabling them to truly become digital to the core.

