



# Building Event Driven Architectures using OpenEdge CDC

**Richard Banville, Fellow, OpenEdge Development**

**Dan Mitchell, Principal Sales Engineer**

October 26, 2018

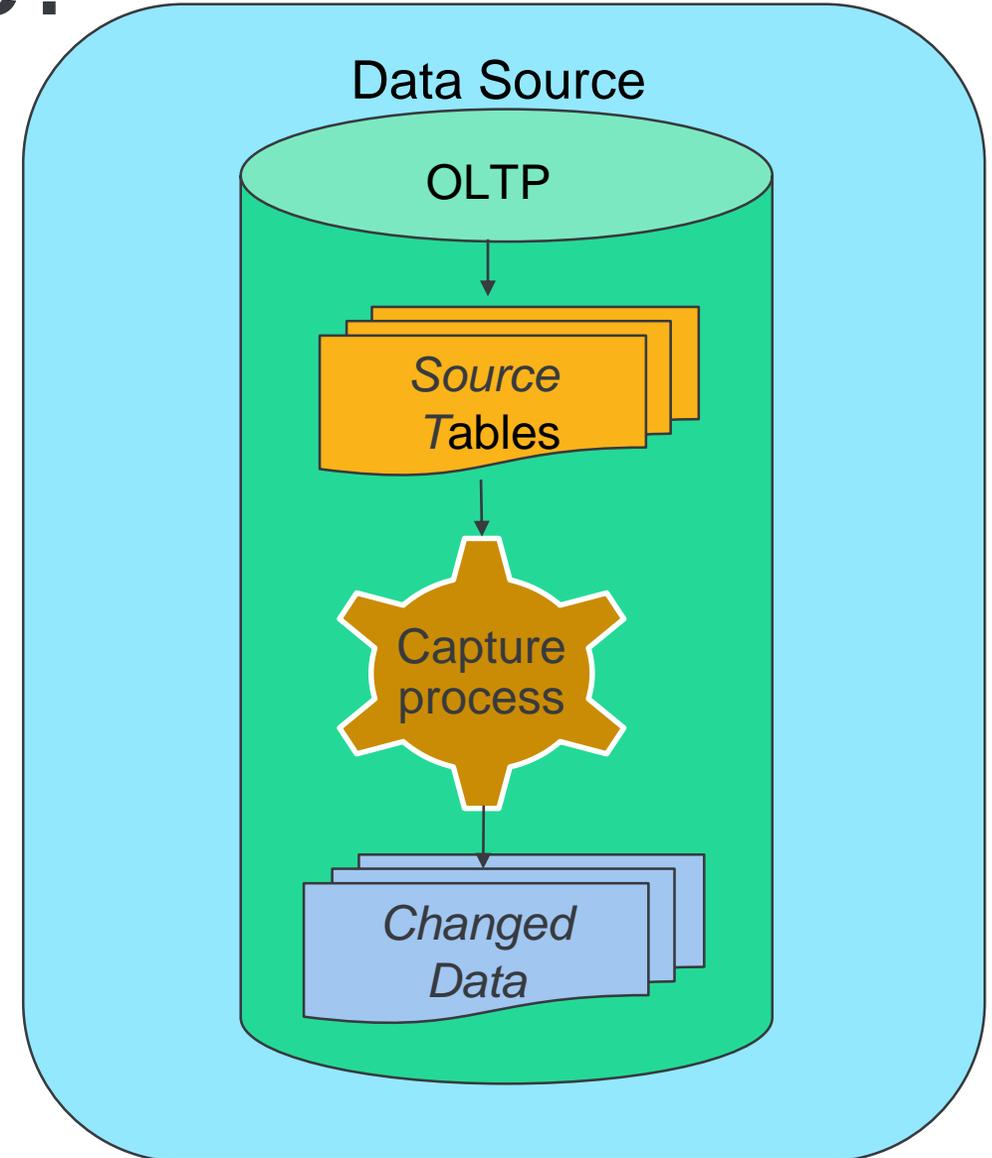


# Agenda

- Change Data Capture (CDC) Overview
- Configuring data capture via policies
  - OpenEdge Explorer/Management
  - ABL APIs
- Data Capture
  - Change Tracking
  - Recording of data value changes
- CDC data as part of an event driven architecture

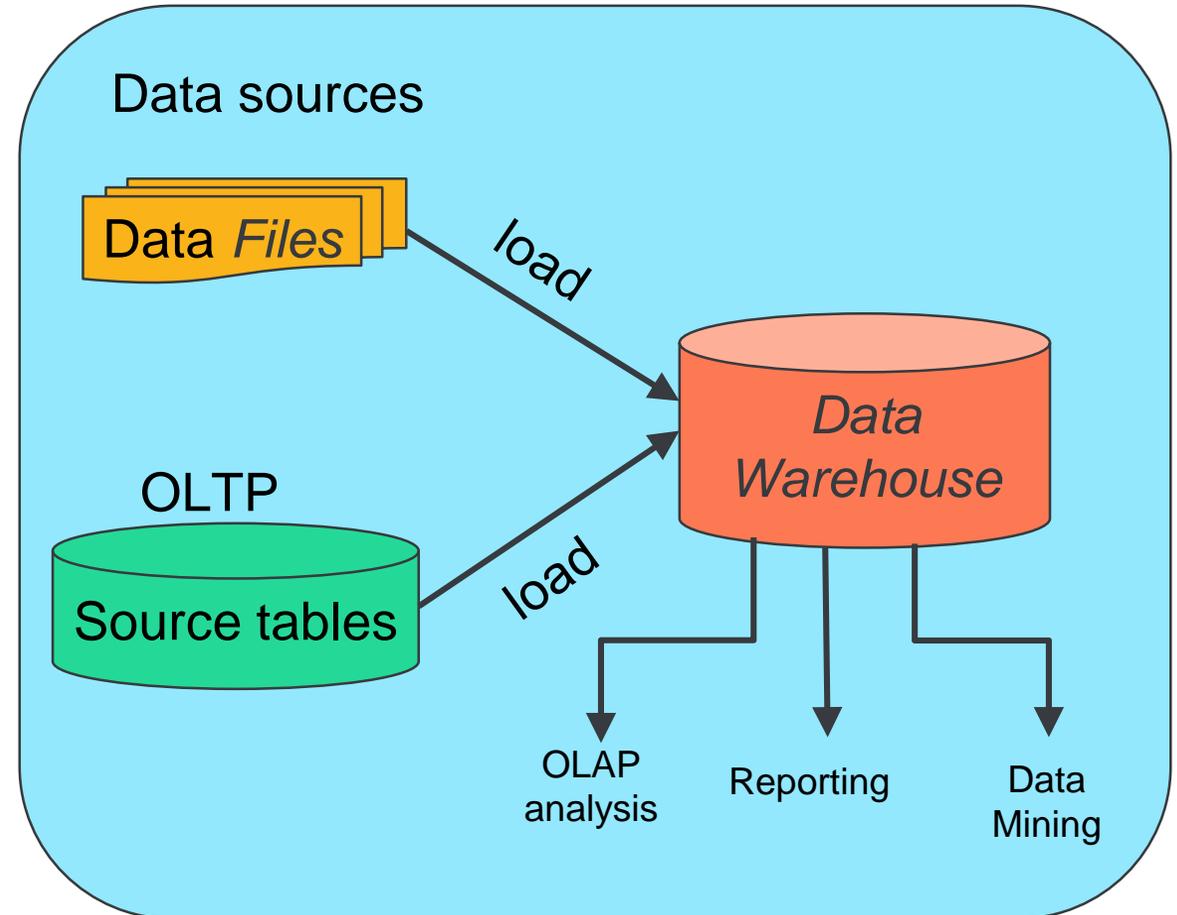
# What is Change Data Capture?

- Ability to intercept data as it changes
- Change data storage
  - Relational tables
  - Flat change logs



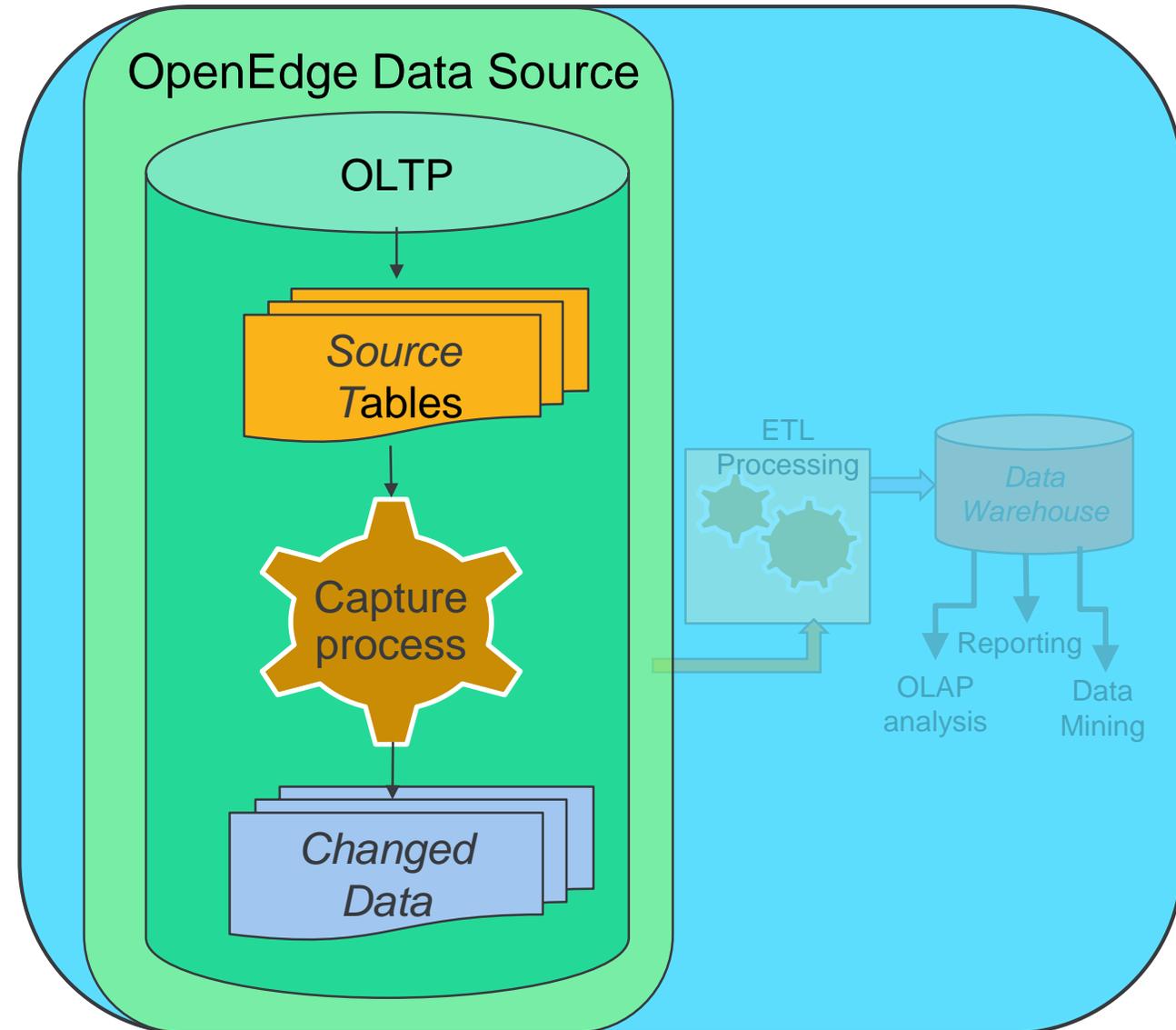
# Why Change Data Capture?

- Data Warehousing
  - Repository of data from various sources
  - Input to strategic business decisions
- Avoid warehouse bulk reload
  - Time consuming
  - Some data is irrelevant to business needs
  - Requires coordination of data sources
- Business needs
  - Capture changes to data of interest only
  - Extract, Transform, Load (ETL)
    - Tools / mechanisms process captured data



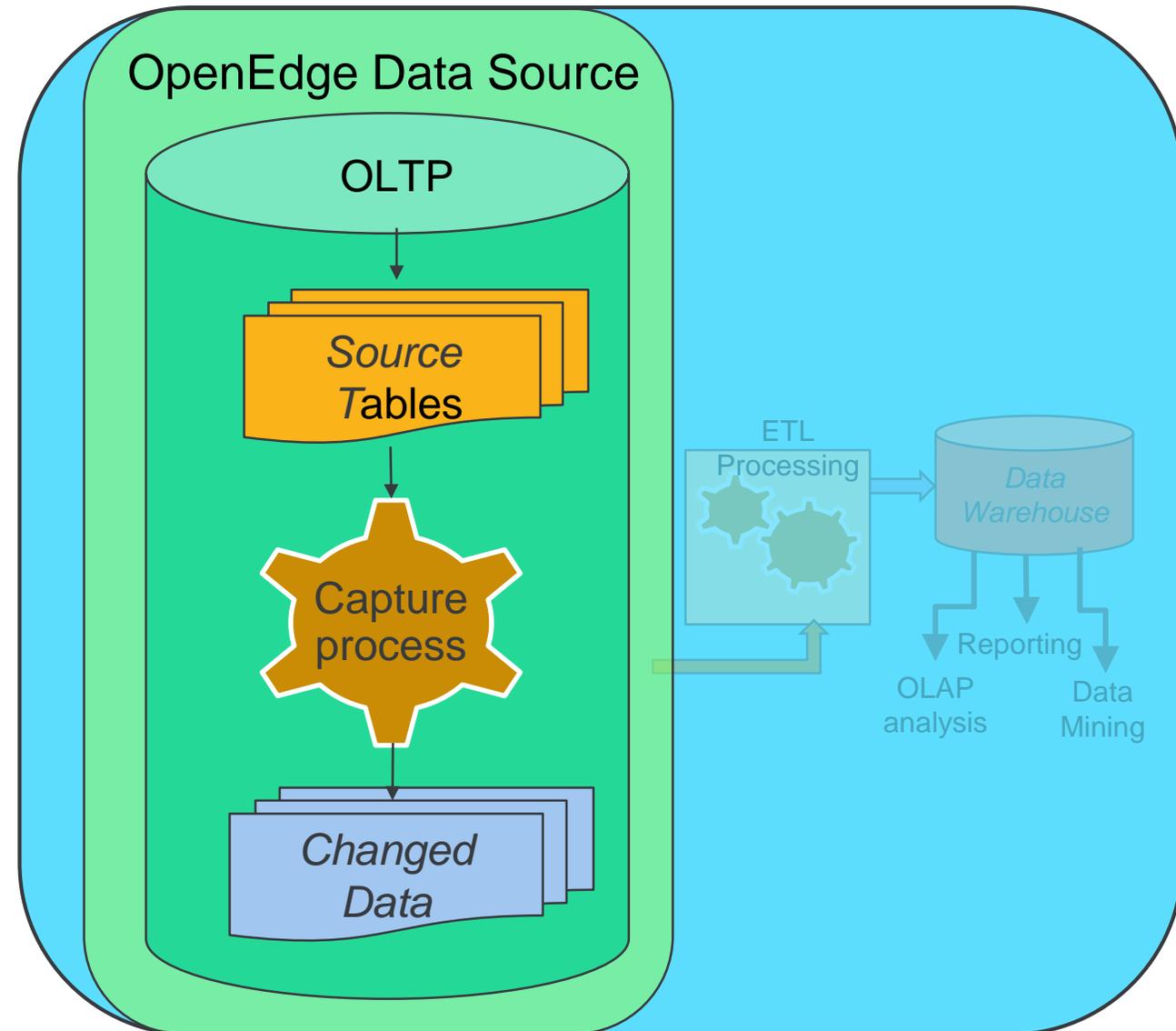
# OpenEdge Change Data Capture

- Policy driven configuration
- Database triggers
  - Capture data within same DB
- Change Tracking
  - Event occurrence
  - Transaction and time stamp
  - Unique identification
- Change Data
  - Record changes of field data in “native” format

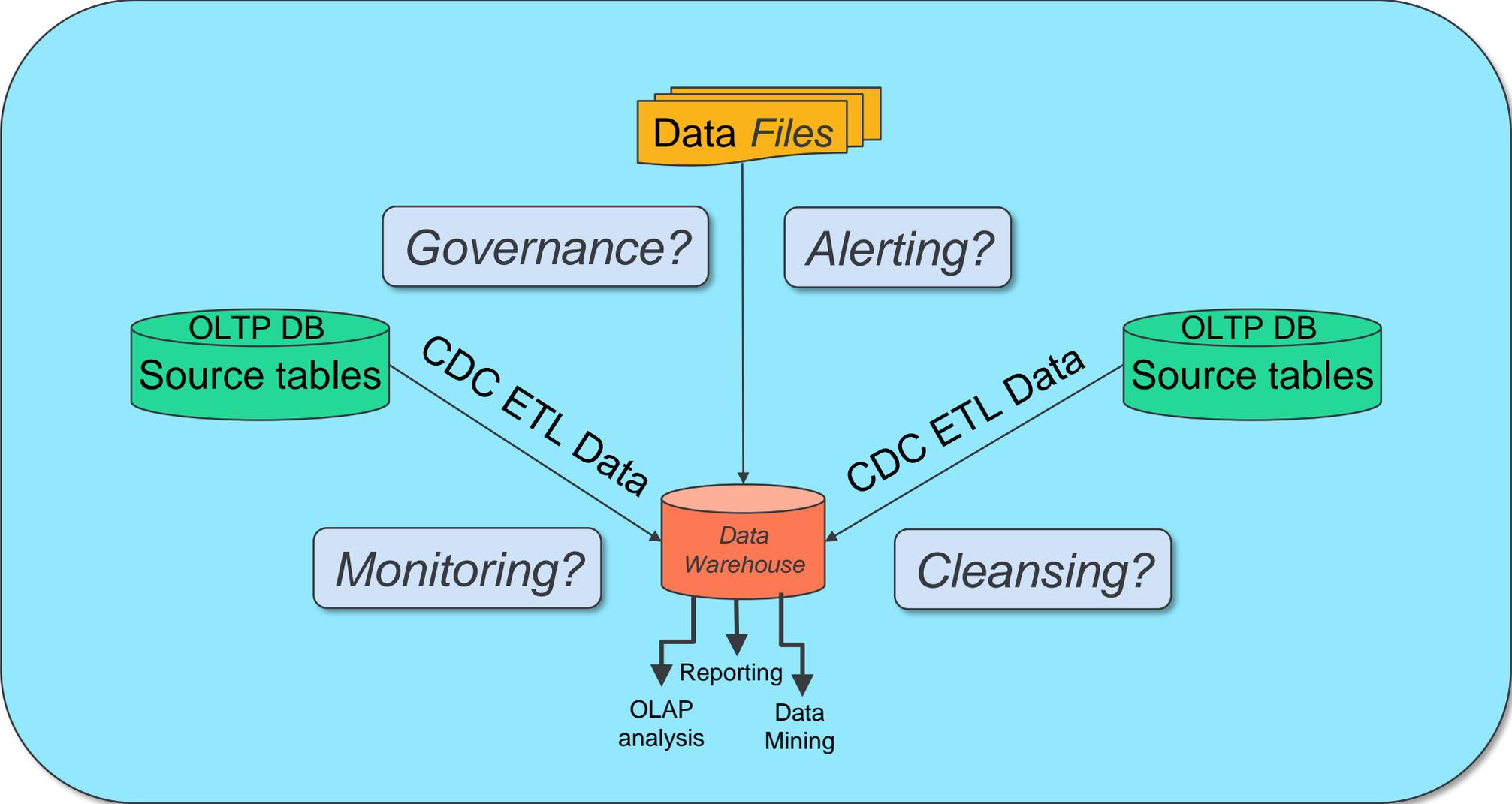


# Why OpenEdge Change Data Capture?

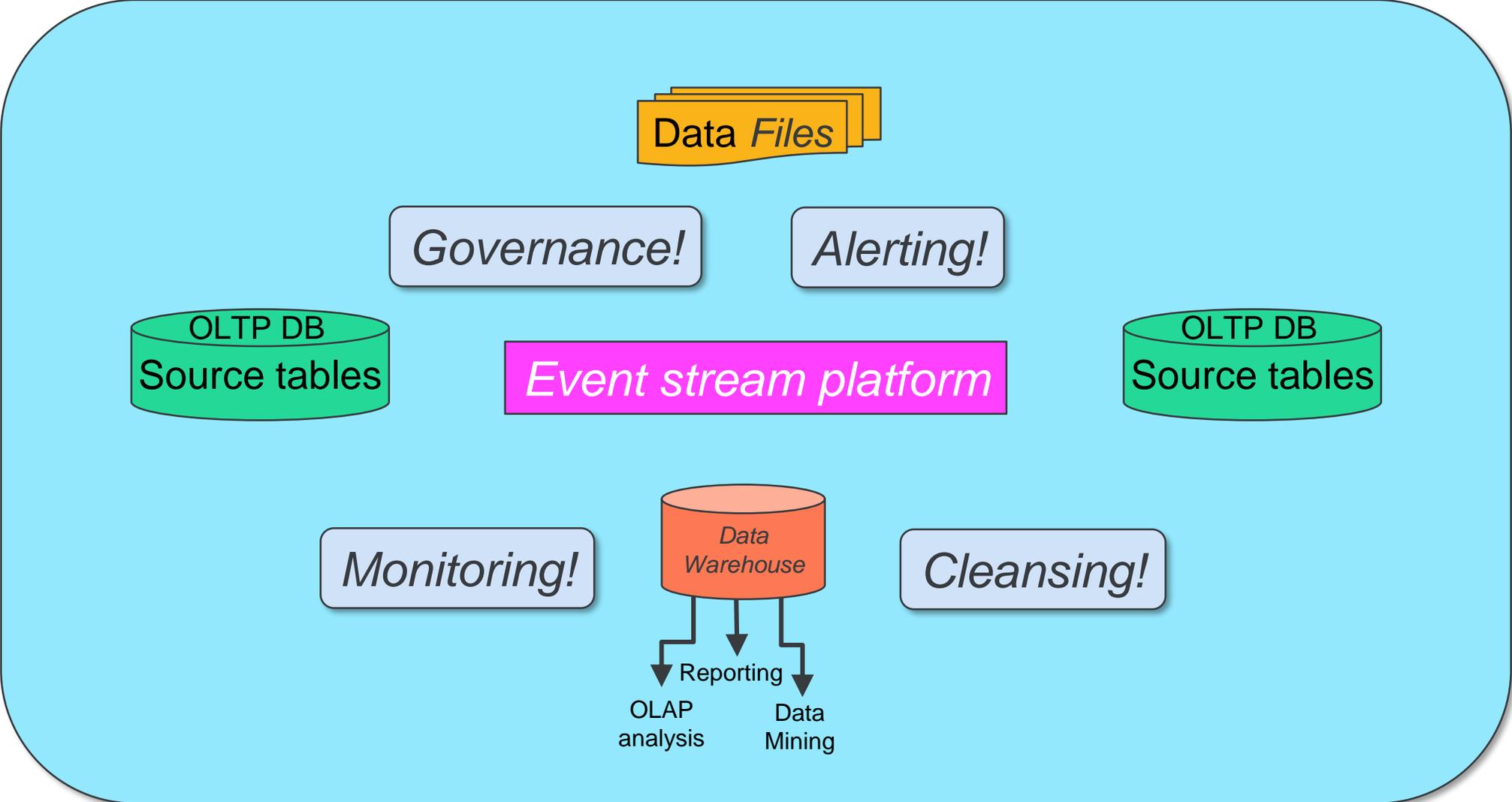
- Alternative to trigger-based replication
- Eases deployment costs
  - Configuration
  - Maintenance
  - Handling of schema changes
- And its fast
  - Using CDC, Pro2 experienced 2.5x performance improvement
- ETL languages
  - ABL – write your own
  - SQL – yours or off the shelf



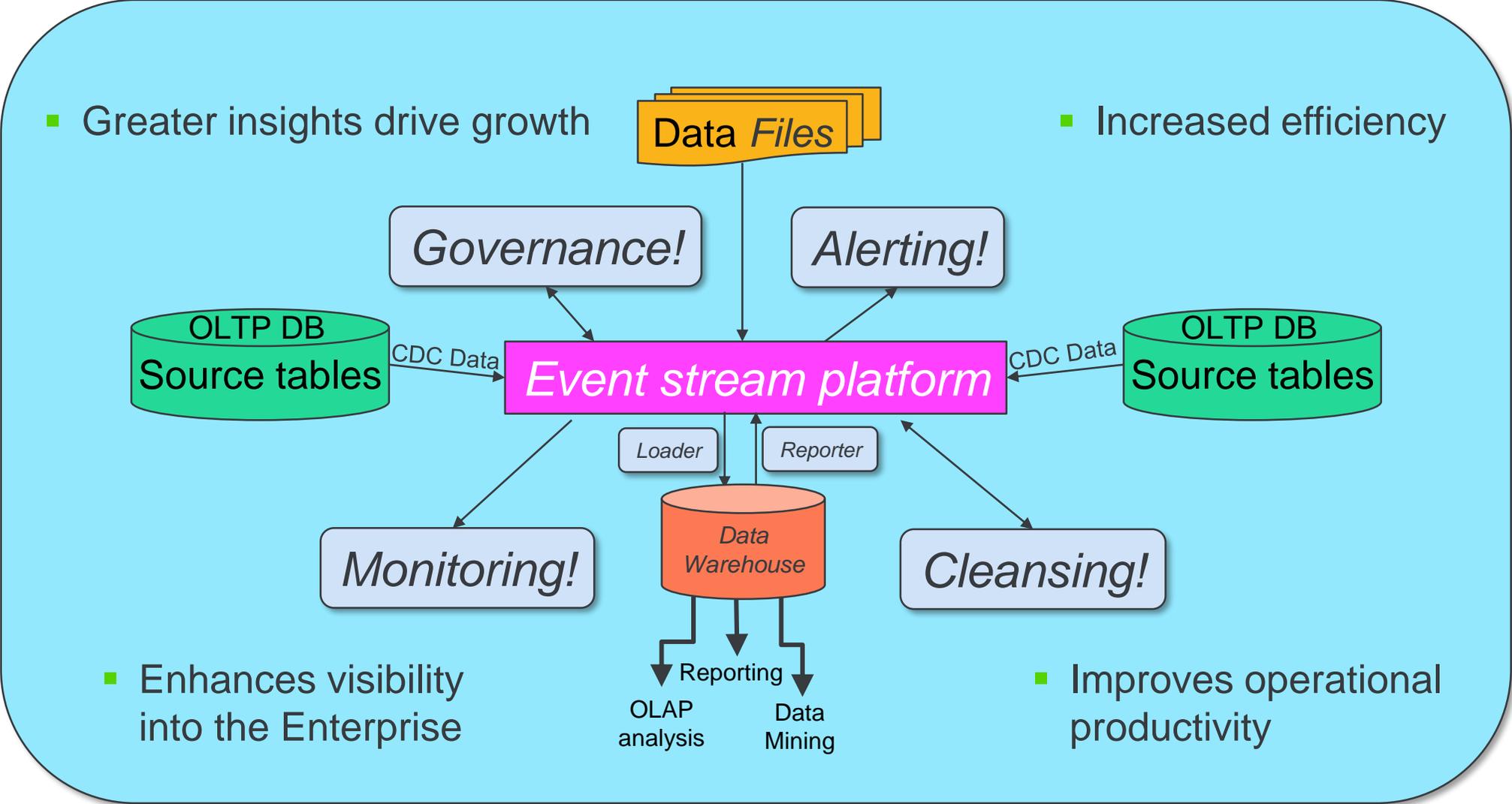
# Change Data Capture in your Enterprise



# Change Data Capture in your Enterprise



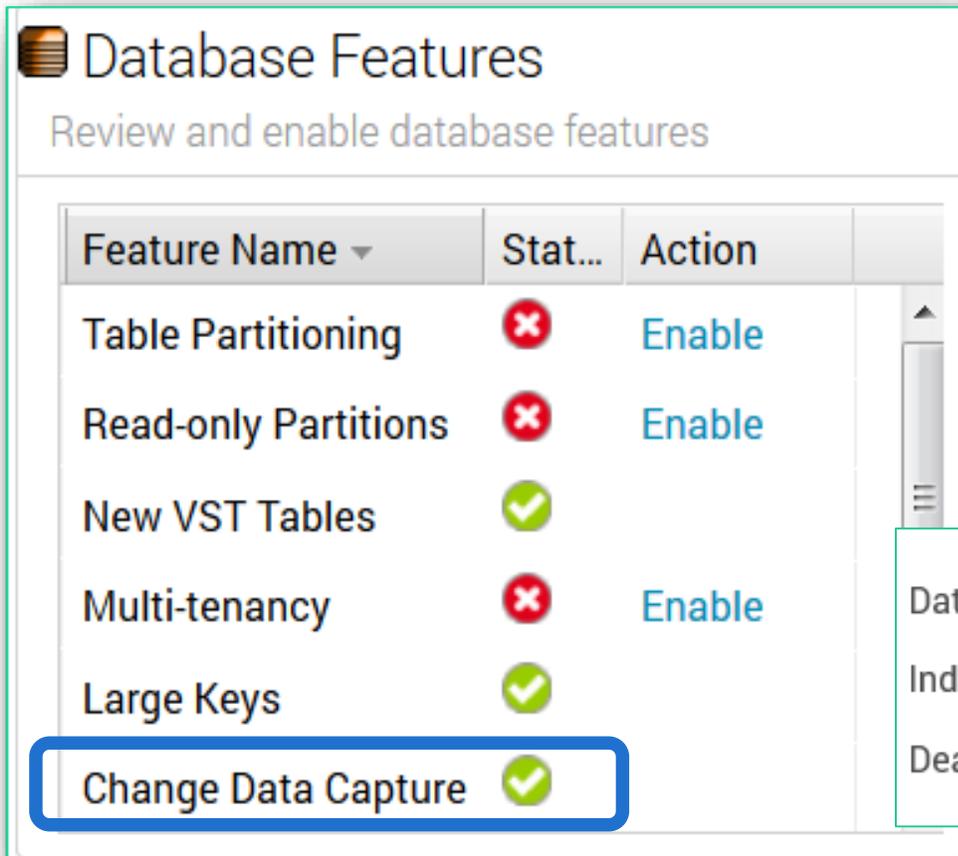
# Change Data Capture in your Enterprise



# **First things first: Policy Configuration**

# Enabling CDC for a Database

- Command line or via Database Admin Console in OEM/OEE
  - Enabled online or offline



Database Features

Review and enable database features

| Feature Name               | Stat...  | Action |
|----------------------------|----------|--------|
| Table Partitioning         | ✘        | Enable |
| Read-only Partitions       | ✘        | Enable |
| New VST Tables             | ✔        |        |
| Multi-tenancy              | ✘        | Enable |
| Large Keys                 | ✔        |        |
| <b>Change Data Capture</b> | <b>✔</b> |        |

```
proutil <db> -C enableCdc  
area CDC_Tracking_Data  
indexarea CDC_Tracking_Index  
deactivateidx
```

|                   |  |
|-------------------|--|
| Data area:        | <input type="text" value="Customer/Order Area"/> |
| Index area:       | <input type="text" value="Order Index Area"/>    |
| Deactivate Index: | <input type="checkbox"/>                         |

# Configuring CDC Policies in OpenEdge

*OpenEdge  
Management &  
Explorer*

*Dump &  
Load Policy*

*Change Data  
Capture ABL API*

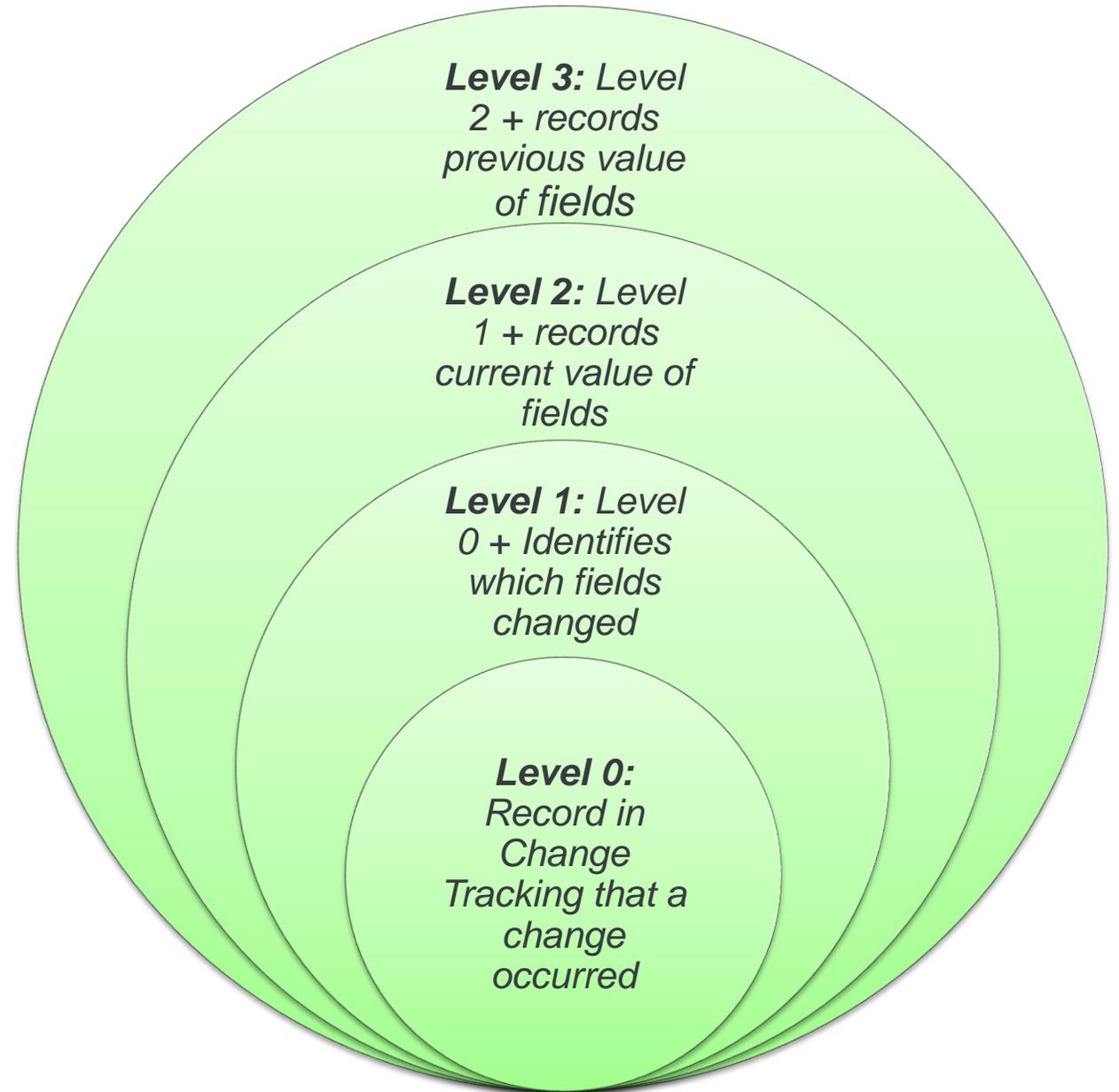
# Change Data Capture – Policies

- Define source table & data capture levels
- Maintained through
  - OpenEdge Explorer/Management - Database Management Console (GUI)
  - ABL APIs (programmatic)
  - “Special” dump and load
- CDC Policies:

| Policy Info             | Policy Table Name | Instance                              |
|-------------------------|-------------------|---------------------------------------|
| CDC source table policy | _Cdc-Table-Policy | One record per source table           |
| CDC field info policy   | _Cdc-Field-Policy | One record per field per source table |

# Policy Levels

- Levels build on each other
- Amount of data collected is flexible
  - Based on your business needs
- Levels 1 thru 3 require fields specification



# Configuring Record Change Policy

Progress® OpenEdge® Management

Database Administration / sports2000 / Create Change Data Capture

CANCEL SUBMIT GENERATE POLICY PROGRAM

Create Change Data Capture policy

Policy name: PolicyOrder01

Description:

Database connection: sports2000

Table: PUB.Order

Level:  Minimal(0)  Minimal with Fieldmap(1)  Medium(2)  Maximum(3)

State:  Active  Inactive

### Table list

| Table name     | Schema name |
|----------------|-------------|
| Feedback       | PUB         |
| InventoryTrans | PUB         |
| Invoice        | PUB         |
| Item           | PUB         |
| LocalDefault   | PUB         |
| Order          | PUB         |
| OrderLine      | PUB         |
| POLine         | PUB         |
| PurchaseOrder  | PUB         |
| RefCall        | PUB         |

Displaying tables 1 - 25 of 25

SELECT CANCEL

# Configuring Field Change Policy

CANCEL SUBMIT GENERATE POLICY PROGRAM

Create Change Data Capture policy

Policy name: PolicyOrder01

Description:

Database connection: sports2000

Table: PUB.Order

Level:  Minimal(0)  Minimal with Fieldmap(1)  Medium(2)  Maximum(3)

State:  Active  Inactive

Encrypt policy:

Identifying fields:

Data area: Customer/Order Area

Index area: Customer/Order Area

Change table: OrderChangeTable

Change table owner: bill

Select Change Data Capture Fields Show All fields Note, Marking the checkbox on the field as checked will add corresponding

| <input type="checkbox"/>            | Field name | Data type | Enable identifying field    | Field order | Description |
|-------------------------------------|------------|-----------|-----------------------------|-------------|-------------|
| <input checked="" type="checkbox"/> | BillToID   | integer   | <input type="checkbox"/> NO |             |             |

Change Table Properties

# CDC Field Policy - Level >0

- A field policy is required
- Unlimited field policy entries allowed
- Change data captured only for selected fields
- Identifying fields 1 to 15
  - Indexing optional

Select Change Data Capture Fields Show All fields *Note, Marking the checkbox on the field as checked will add*  
*for this Change Data Capture policy*

| <input type="checkbox"/>            | Field name   | Data type | Enable identifying field    | Field order          |
|-------------------------------------|--------------|-----------|-----------------------------|----------------------|
| <input checked="" type="checkbox"/> | BillToID     | integer   | <input type="checkbox"/> NO | <input type="text"/> |
| <input checked="" type="checkbox"/> | Carrier      | character | <input type="checkbox"/> NO | <input type="text"/> |
| <input type="checkbox"/>            | Creditcard   | character | <input type="checkbox"/> NO | <input type="text"/> |
| <input checked="" type="checkbox"/> | CustNum      | integer   | <input type="checkbox"/> NO | <input type="text"/> |
| <input type="checkbox"/>            | Instructions | character | <input type="checkbox"/> NO | <input type="text"/> |
| <input checked="" type="checkbox"/> | OrderDate    | date      | <input type="checkbox"/> NO | <input type="text"/> |
| <input type="checkbox"/>            | Ordernum     | integer   | <input type="checkbox"/> NO | <input type="text"/> |

# Change Data Capture Policy Information

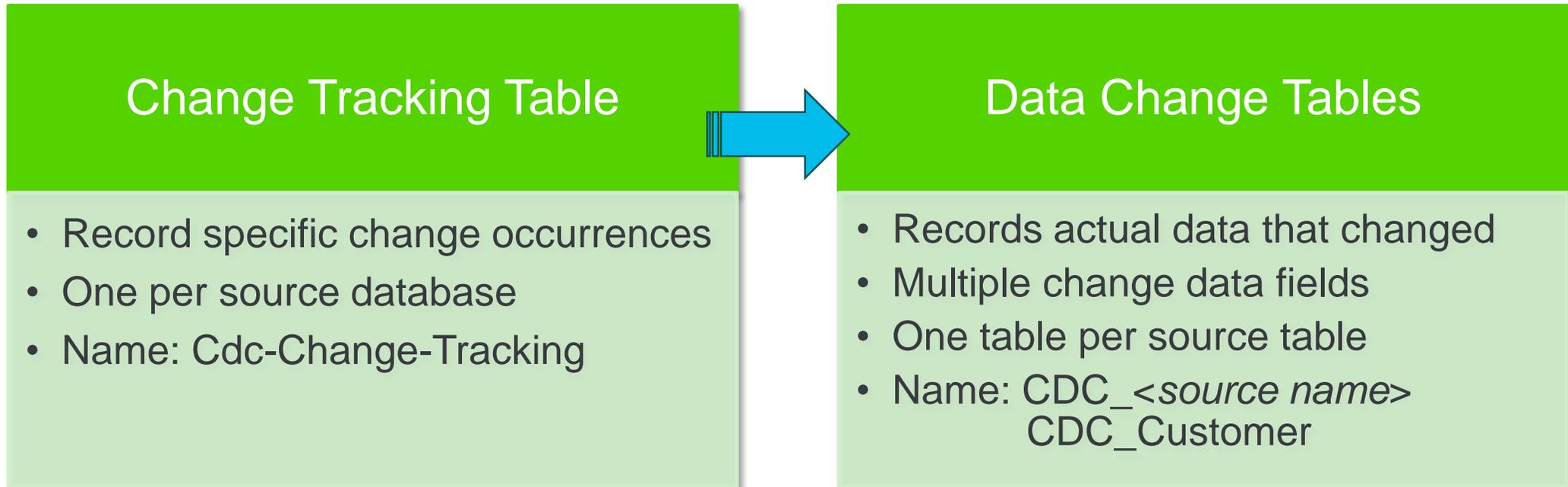
- Schema changes to source tables

| Schema Change      | Affect / rules                     | Action                         |
|--------------------|------------------------------------|--------------------------------|
| Adding a new field | CDC will not be affected           | Needs field policy for capture |
| Renaming a field   | Cascades to Change Table fields    | This is automatic              |
| Deleting a field   | CDC no longer captures data for it | Existing data remains          |
| Renaming the table | No change for CDC                  |                                |
| Deleting the table | No active CDC policy can exist     | Policy must already be deleted |

- Policies can be dumped (to .cd file) and loaded
  - Via Database Administration or Database Admin Console
  - For load, the associated schema must match the original database

# Data Capture

# OpenEdge Change Data Capture – Table Relations



# Change Tracking Table – Record Format

| <b>_cdc-Change-Tracking</b> |                  |              |                  |            |                   |
|-----------------------------|------------------|--------------|------------------|------------|-------------------|
| _Policy-Id                  | _Tran-Id         | _Time-Stamp  | _Change-Sequence | _Operation | _Change-Field-Map |
| _Recid                      | _Source-Table -# | Partition-Id | _Tenant-Id       | _User-Misc | _User-Name        |

- Ordering: \_Source-Table, Time Stamp, \_Change Sequence
  - unique per policy
- Operation: 1: Create, 2: Delete, 3: After Update, 4: Before Update, 5-11: FIELD deleted in some way
- Change Field Map: Identifies which fields changed
- \_User-Misc: User/application updateable
  - Intended use: identify a change has been processed (i.e. loaded into the warehouse)

# Change Tables – Record Format (field level data)

- No change table for CDC policy level 0
- **One** record per Create\*, Delete operation
- **Two** records for update operation if policy level is 3

| Change Table – CDC_customer |                    |                         |                        |             |
|-----------------------------|--------------------|-------------------------|------------------------|-------------|
| _Tran_id                    | <u>_Time-Stamp</u> | <u>_Change-Sequence</u> | _Continuation-Position | _ArrayIndex |
| _Fragment                   | Capture Col1       | Capture Col2            | Capture Col3           | ...         |

- Ordering of changes by Change-Sequence or Time-Stamp
- Child reference of Change Tracking table
- Also contains identifying field information

# Change Tables – Record Format

- No change table for CDC policy level 0
- **One** record per Create\*, Delete operation
- **Two** records for update operation if policy level is 3

| Change Table – CDC_customer |              |                  |                        |             |
|-----------------------------|--------------|------------------|------------------------|-------------|
| _Tran_id                    | _Time-Stamp  | _Change-Sequence | _Continuation-Position | _ArrayIndex |
| _Fragment                   | Capture Col1 | Capture Col2     | Capture Col3           | ...         |

- Field level data requested
  - One column for each field

# Change Tables – Record Format

- No change table for CDC policy level 0
- **One** record per Create\*, Delete operation
- **Two** records for update operation if policy level is 3

## Change Table – CDC\_customer

| _Tran_id  | _Time-Stamp  | _Change-Sequence | _Continuation-Position | _ArrayIndex |
|-----------|--------------|------------------|------------------------|-------------|
| _Fragment | Capture Col1 | Capture Col2     | Capture Col3           | ...         |

- **\_Fragment**: Identifies split record order (> 32,000 bytes)
- **\_Continuation-Position**: Which field was split
- ABL built-in methods help deal with this



# Processing Captured Changes

## Using an Event Driven Architecture

# But first: How to access CDC Records?

**ABL Batch Client** e. g. reading new entries on a scheduled basis.

---

- Find new entries by `_Source-Table-Number` and `_Change-Sequence` or `_Time-Stamp`
- Build the “piece of information” to publish
- Update field `„_cdc-Change-Tracking._User-Misc“` for tagging processing status.

**SQL-Client** e. g. reading a filtered set of records.

---

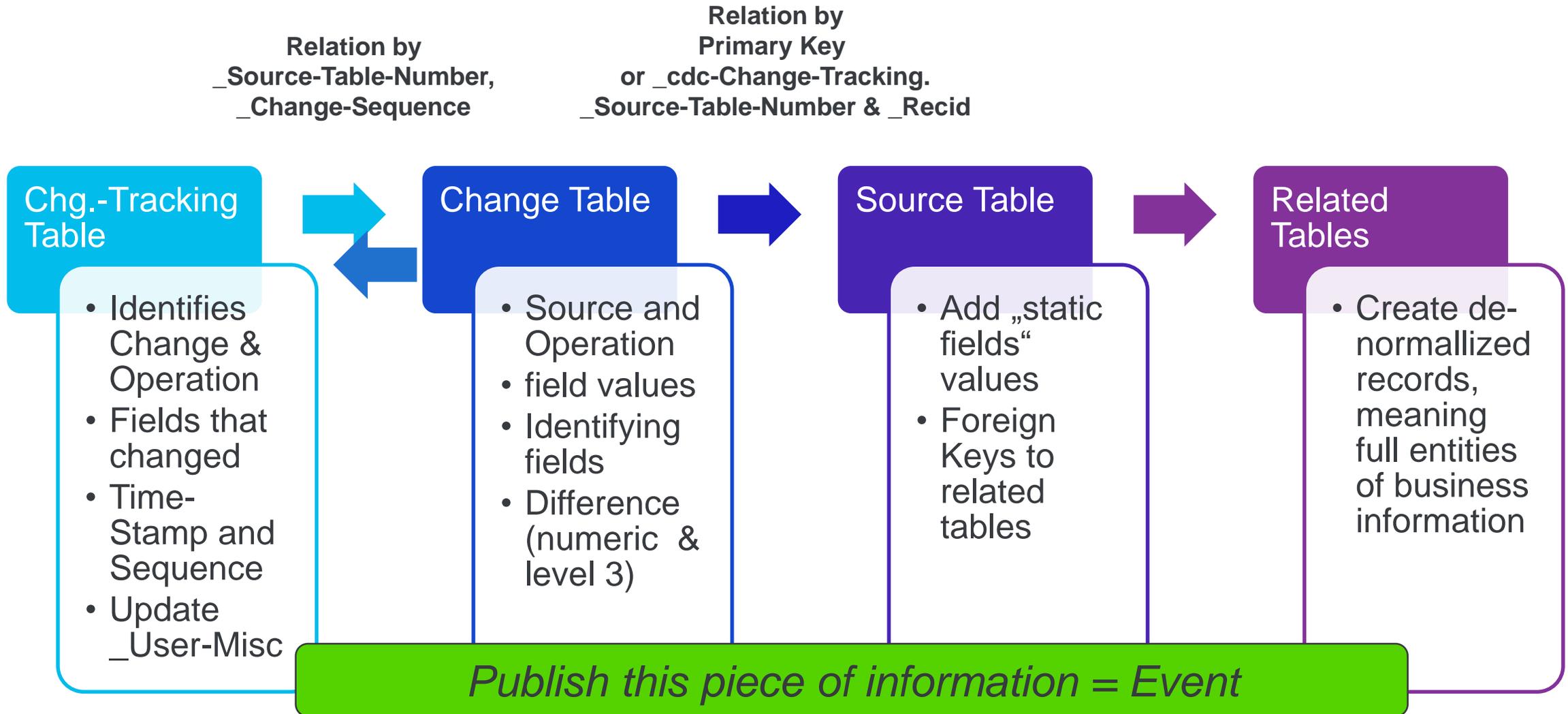
- Add Views to provide de-normalized data, including aggregates
- Access the CDC tables in DB Schema „PUB“

**PASOE** Provide an API for processing CDC logic

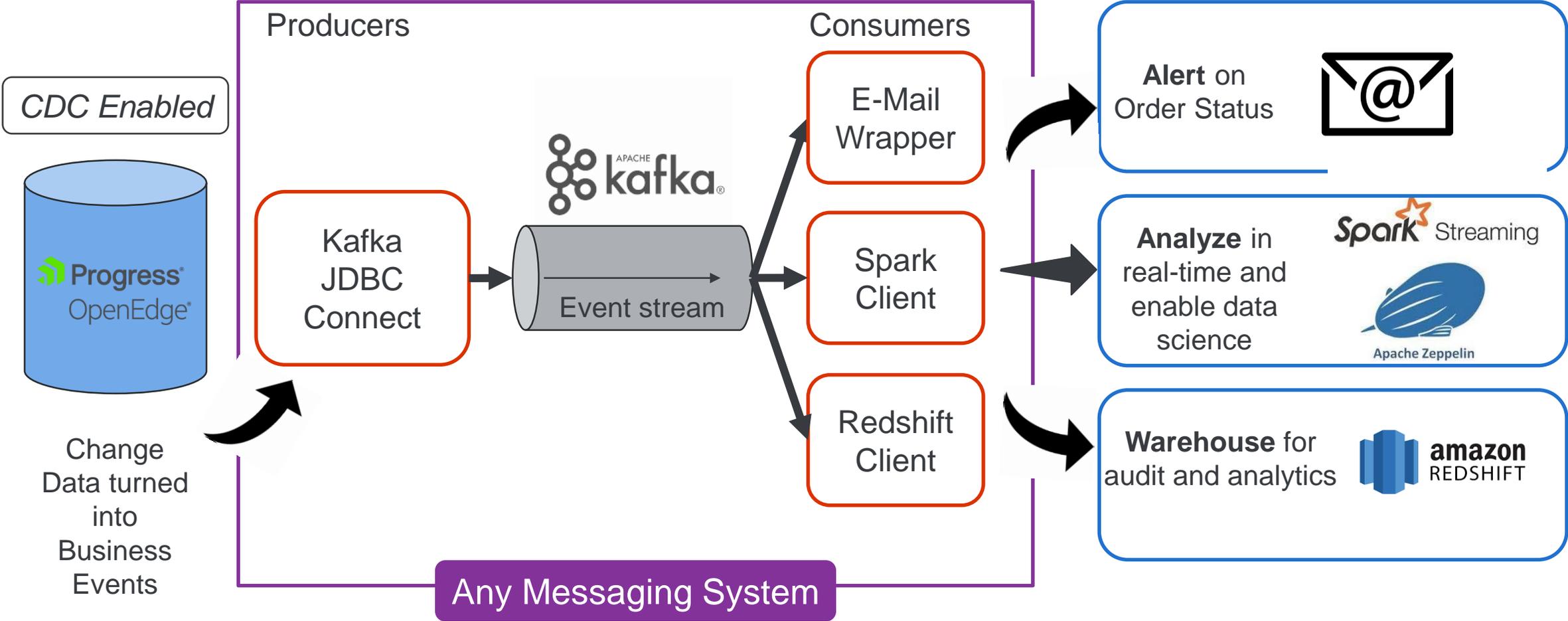
---

- e. g. Dynamic fetches
- Expose Operations via REST

# How to build the Entity that gets published

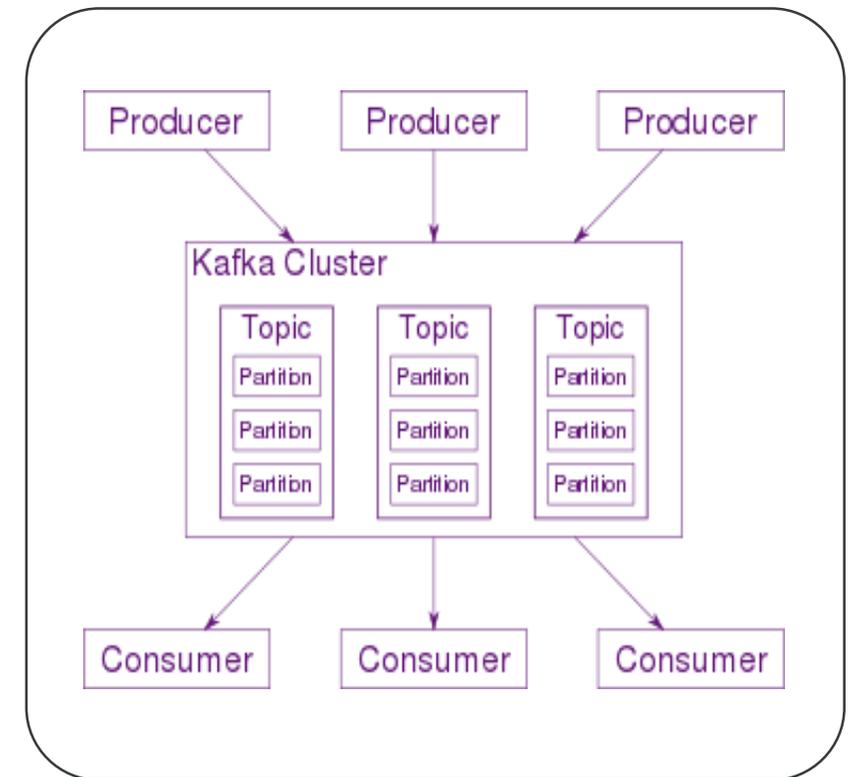


# An Event-Streaming Architecture



# Apache Kafka in a Nutshell

- Kafka is a Messaging System, handling **Key-Value Entries**. Originally developed by LinkedIn.
- Kafka runs on a cluster of one or more servers (called brokers), distributes and replicates partitions for performance and fault-tolerance.
- Arbitrarily many processes called **Producers** feed messages into different "partitions" within different "**topics**".
- Other processes called **Consumers** can read messages from Topics.
- Another Server process called **Zookeeper** coordinates concurrent consumer access to Kafka.



# Demo Producers and Consumers

## Our Producers

- are “Kafka JDBC Connect“ instances
- reading records from Change Tables via SQL & JDBC
- and passing it into a Kafka Topic named by the OE Source Table Name
- executed each 5 seconds implementing Polling

## Our Consumers

- 3 custom programs
- reading new entries from topics on a scheduled bases
- passing these into Target Applications using their APIs

# Our “Target Applications”

## An E-Mail Client

- Receiving Order Information in a human-readable format.

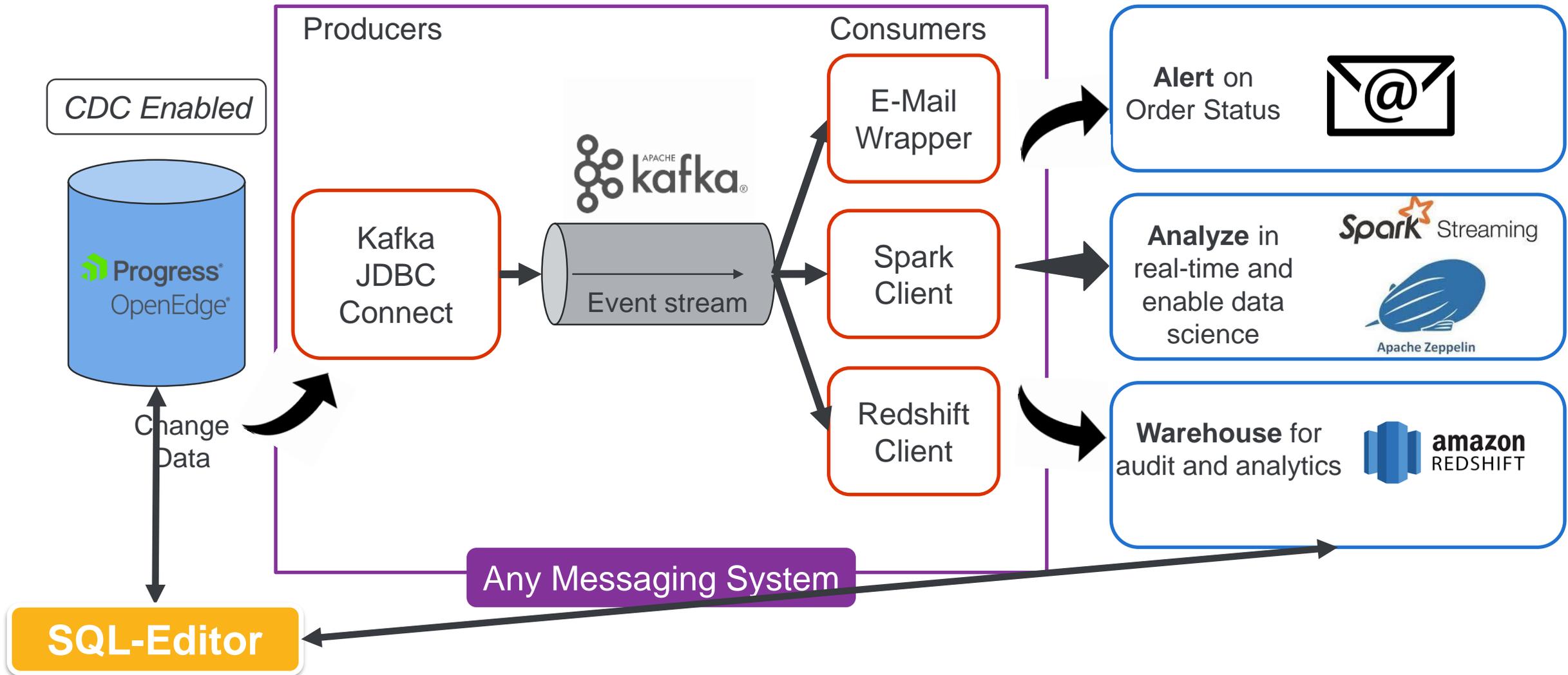
## An Apache Spark Database providing Real-Time Analytics

- Batching the stream of records into 20 sec batches, applying analysis instantly.

## An Amazon Redshift Data Warehouse

- Which gets kept in synch with the Source Database by the CDC track

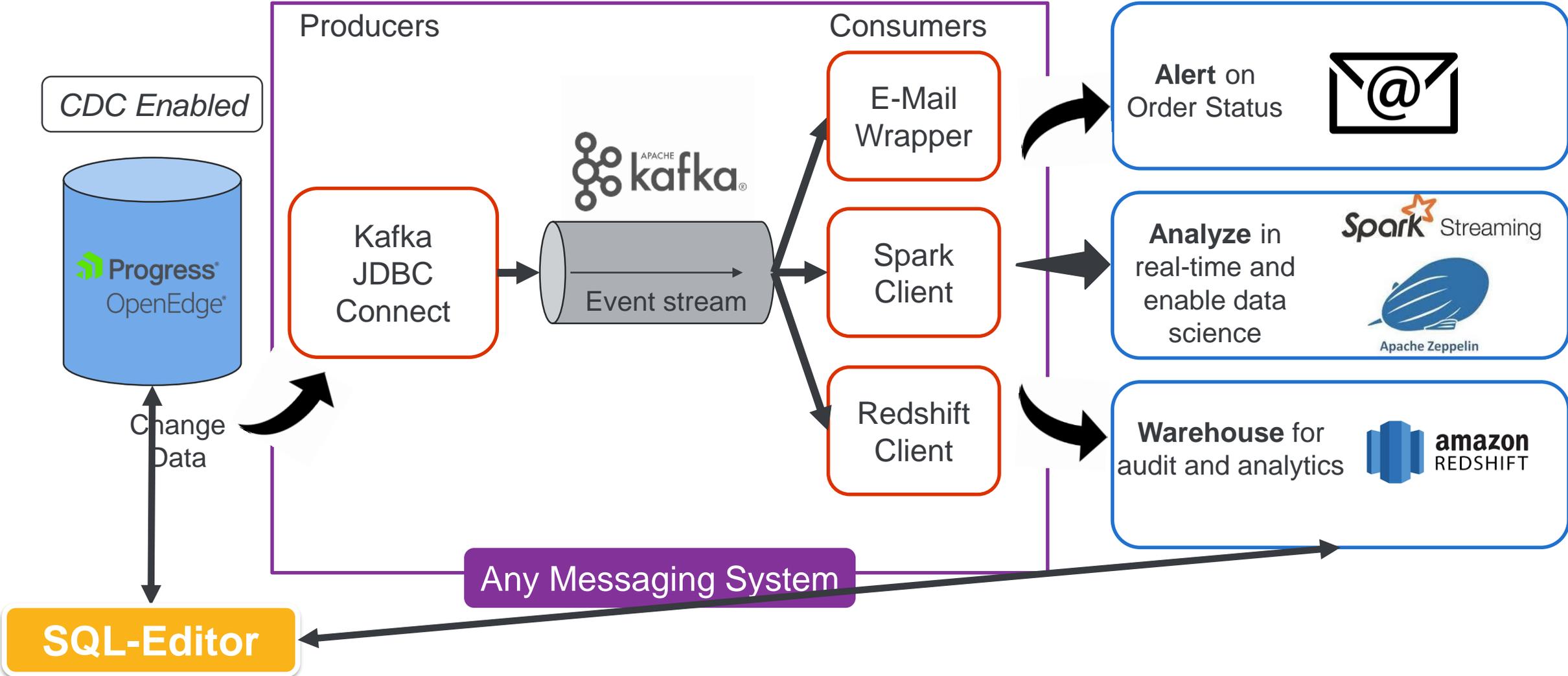
# The full picture



# Demo Time



# The full picture



# For more information

- Blog on Event Architecture with Kafka by Saikrishna Teja Bobba
  - <https://www.progress.com/blogs/build-an-etl-pipeline-with-kafka-connect-via-jdbc-connectors>
- Documentation OpenEdge CDC
  - [https://documentation.progress.com/output/ua/OpenEdge\\_latest/index.html#page/gscdc%2Fchange-data-capture.html%23](https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/gscdc%2Fchange-data-capture.html%23)
- How-To Videos introducing OpenEdge CDC
  - Find four OpenEdge Videos, published Summer 2018 on <https://www.progress.com/video?product=progress-openedge>

# Change Data Capture(CDC) in OpenEdge 11.7



## Easy to use

- *Policy driven with GUI configuration*
- *Multiple configuration levels*
- *Code can be generated for you*



## Performant

- *Recorded by internal database triggers (similar to auditing)*
- *Data stored in multiple "change" tables*



## Flexible

- *Data stored in "native format"*
- *Schema change aware*
- *ETL helper function for ABL and SQL*

*A component for delivering event driven architectures in your enterprise*

# Thank You!

Richard Banville

Dan Mitchell

