



Data Monolith to Data Mesh

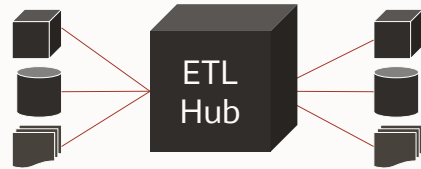
Future of Data Integration Tools and a
focus on Oracle GoldenGate and Stream Processing

Oracle Development

September 2020



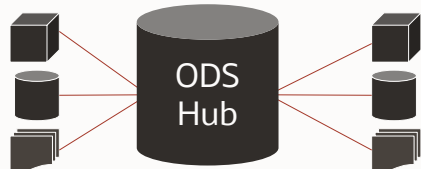
For more than 30yrs, Data Integration Architecture:



ETL Tools...



Vendor DI Tools...



Kimball EDWs...



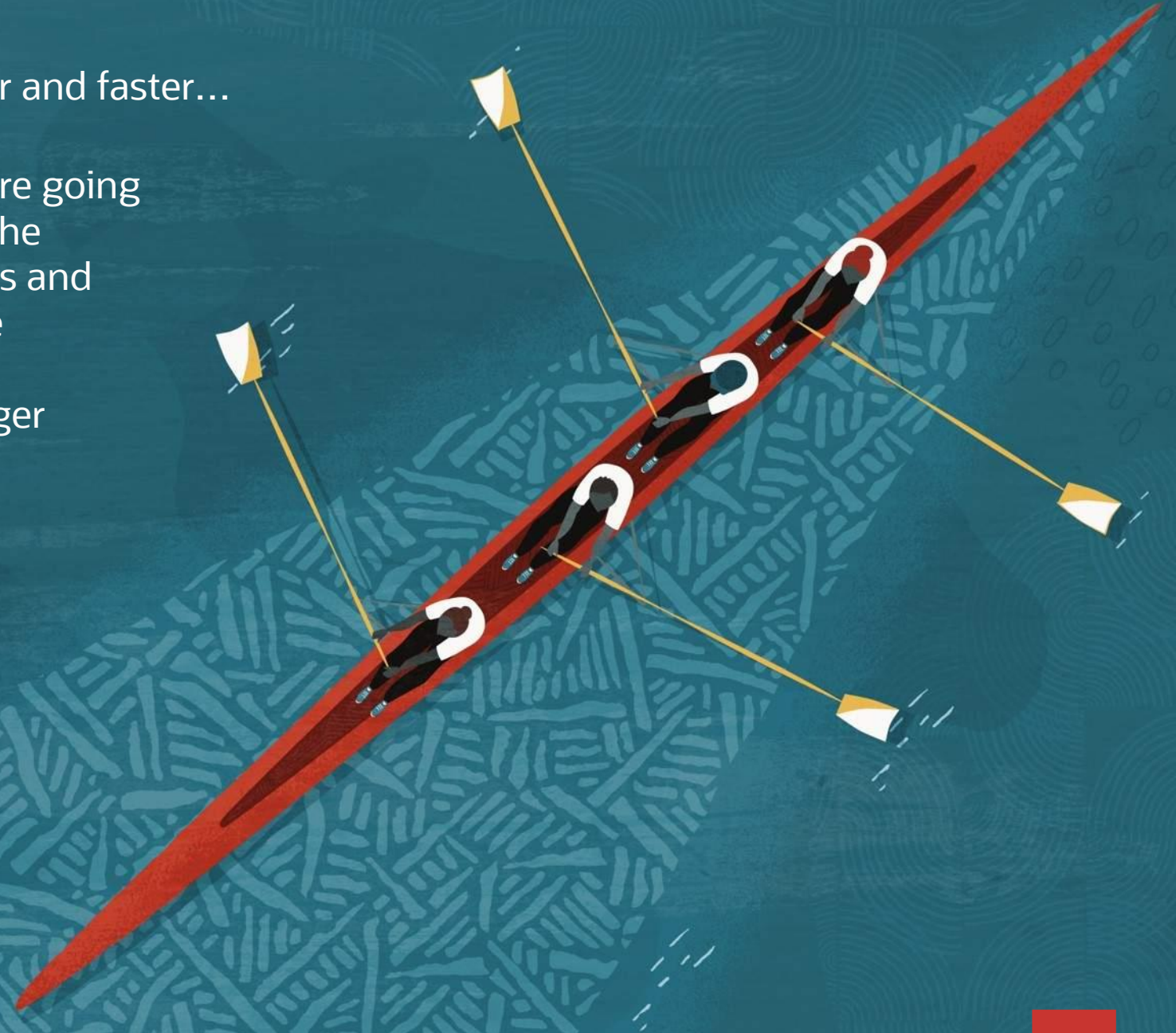
Data Lakes...

Is “Hub and Spoke” our destiny forever...or could we be on a journey to somewhere else?

The world around us will keep moving faster and faster...

...**IT systems** and the **data** that fuel them are going to need to become faster and more agile...the **people processes** that we follow for DevOps and DataOps must also be faster and more agile

Our old ways of Data Integration are no longer sufficient to meet the future.



Data Fabric | Stream Processing | Data Mesh

FOR ENTERPRISE

Now Te

Forrester's

April 16, 2020

Why Read T

You can use en

organization an

select from a di

Enterprise arch

expect from a d

Tags: Big Data

FORRESTER®

Insights

Who We Serve

Our Expertise

What We Offer

Who We Are

LOG IN

BECOME A CLIENT

Gartner

Search

Advanced Search

BOOKMARK

SHARE

DOWNLOAD

LIST

PRINT

CHART

MAY 2020

ASSESS ?

Data mesh

we must cer

data team to

federated an

support of a

products); it

interoperab

on the exp

If your orga

a diverse se

Implementa

an organizat

Gartner

Technical Professional

This resear

Stream Process

Paradigm

Published: 9 April 2019 ID: GO

Analyst(s): Sumit Pal

Digital businesses mu

continuous intelligenc

professionals impleme

integration, event pro

forbes.com/sites/janakirammsv/2020/09/20/service-meshthe-new-battleground-for-the-platform-wars

Forbes


2,805 views | Sep 20, 2020, 08:34am EDT

Service Mesh - The New Battleground For The Platform Wars

Janakiram MSV Senior Contributor @Cloud

I cover Cloud Computing, Machine Learning, and Internet of Things

The rise of containers has led to the microservices paradigm, where software is developed and deployed as fine-grained services. Multiple services work in tandem to deliver the expected functionality from the application.



Service Mesh - The New Battleground PIXABAY



The future of Data Integration...is Mesh

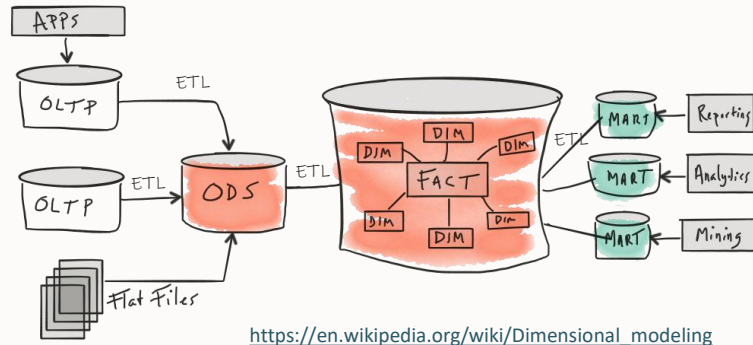
...a **new generation** of Data Mesh capabilities will leave behind the Monolithic Tools of the past to **interconnect** modern, **multi-cloud**, **data-driven** Applications, Analytics, Data Services and Data Products of all types



Evolution towards Real-Time Data Mesh

Monoliths

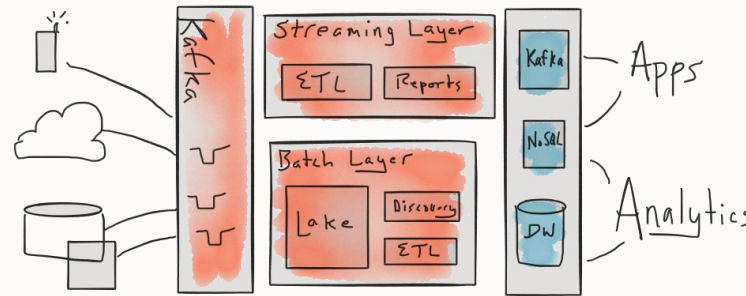
Industry 3.0: Hub and Spoke



This data pattern, popularized by Ralph Kimball and Bill Inmon, has been the foundation for enterprise data management since 1993.

It is transaction consistent, can scale up nicely for most use cases, and is based on SQL, lingua-franca for most tools.

Transitional: Kappa Hub



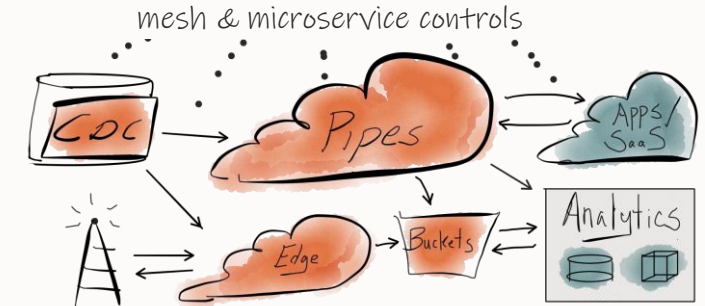
Lambda: <http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html>
Kappa: <https://www.oreilly.com/radar/questioning-the-lambda-architecture/>

By 2010, the Lambda (big data) pattern was common. In 2014, Jay Kreps (of LinkedIn) questioned the Lambda Architecture and spawned Kappa.

Kappa architecture principles consider batch processing as a special case of stream processing. Use a historized event log to process both real-time as well as batch processing.

Mesh

Mature: Distributed Kappa



By 2020, IT infrastructure has dramatically changed – networking, containers, cloud, compute, Kubernetes, IoT etc have all pushed data to the edge and redefined DevOps.

The emerging model is a distributed mesh of data, logs, stream data pipelines, change events, and time series data that flows seamlessly across clouds.

Data Mesh Series



<https://www.youtube.com/playlist?list=PLbqmhpwYrIZJ-583p3KQGDAd6038i1ywe>

Part 1: CDC and Distributed Commit Logs	Best Practices for Maintaining Transaction Consistency with Replication and Kafka	Managing Table to Topic Mappings, Accounting for Schema Evolution etc.	How to Handle the Change Stream: Partial Supplemental CDC, Caching, Lookups etc.	Deployment Topologies: Mid-tier, End-Point, Topic Partitions etc.
Part 2: Microservices Data Architecture w/CDC	Microservices Design Patterns for the Data Tier	Understanding the GoldenGate Microservices Architecture	Event Patterns for CDC: <ul style="list-style-type: none">• Transaction Outbox• CQRS with CDC• Event Sourcing with CDC• Saga with CDC	Event Driven Processing, with CEP, ESP Time Series, & GoldenGate Stream Analytics
Part 3: Demo of Application & Data Integration Mesh	What's in a Name? Data Fabric, Data Hub, Data Mesh, Service Mesh	Purpose of a Data Architecture: Operational vs. Analytic Use Cases	Demonstration Video: Retail / Inventory Analysis <ul style="list-style-type: none">• Sources: Weather.com, Oracle DB, Retail Cloud, AWS S3, Salesforce• Targets: Data Lake (Object Storage and Autonomous Data Warehouse), Data Services (Mobile APIs), Stream Analytics	
Part 4: Monolith to Mesh, the Future of DI Tools	Brief History of (Monolithic) Integration Tools	Future of Data Integration is Mesh	Business Value of a Data Mesh (vs. the Monoliths)	Creating Successful Data Products

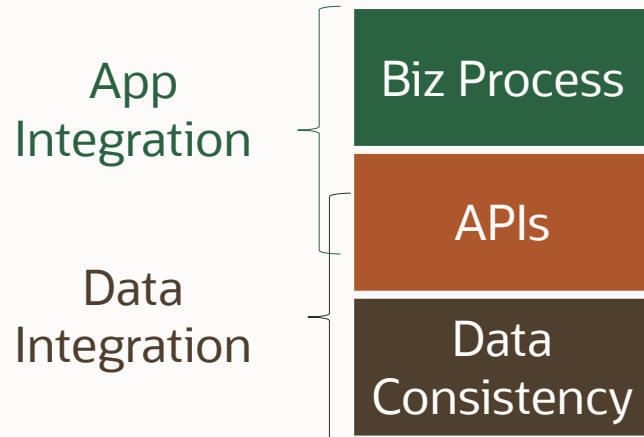


Agenda

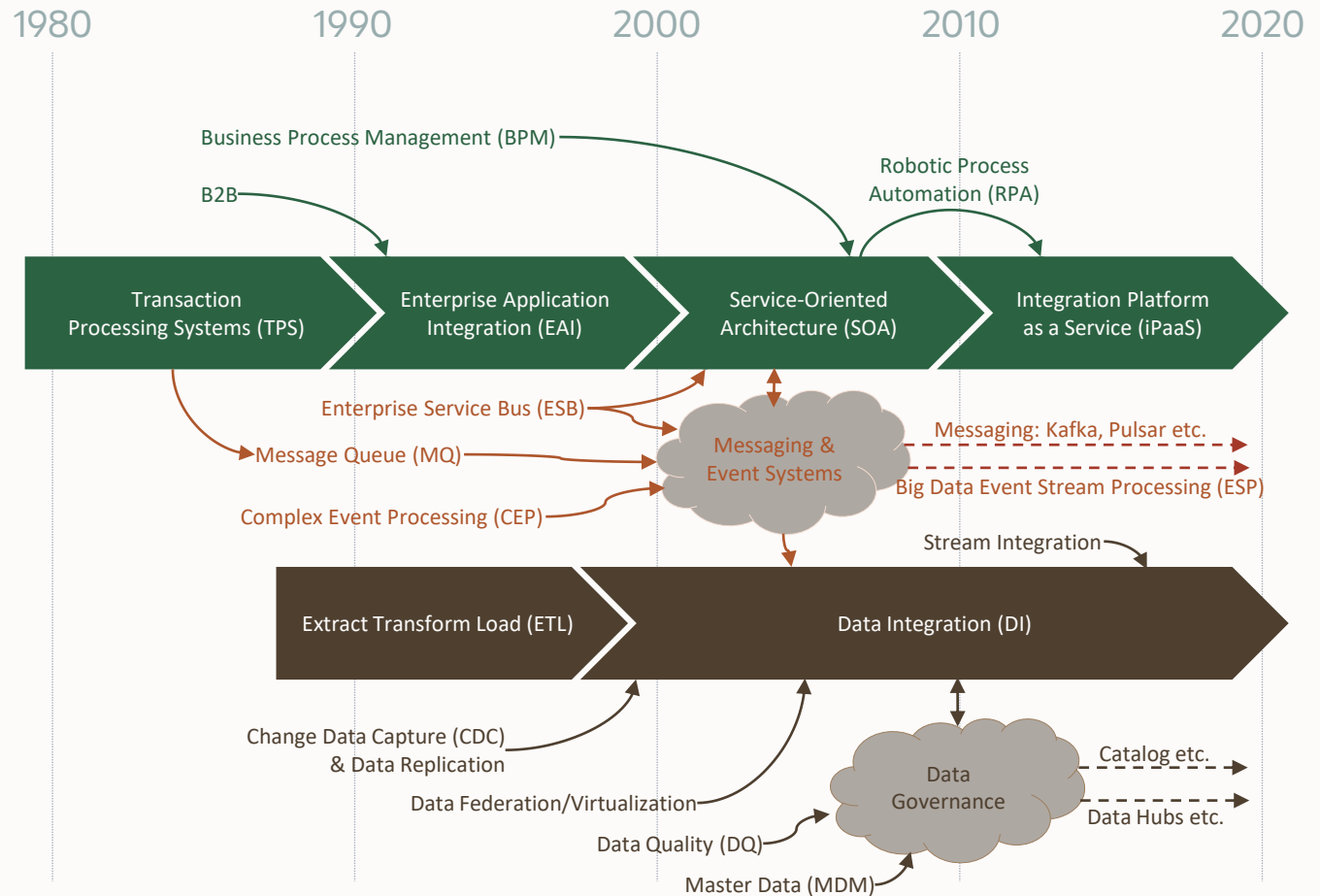
- 1 Brief History of Integration Tools
- 2 Data Mesh as a Next Step
- 3 GoldenGate Strategy for Data Mesh
- 4 GoldenGate Event Stream Processing
- 5 Call to Action

Brief History of Enterprise-class Integration Tools

Historically, integration tools have focused on specific tiers of the software architecture:

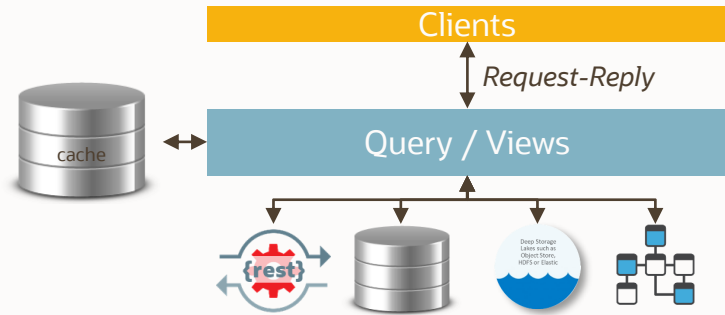


Focus on committed, reliable and ACID-grade data, typically for both Operational (OLTP) and Analytic (OLAP) workloads...

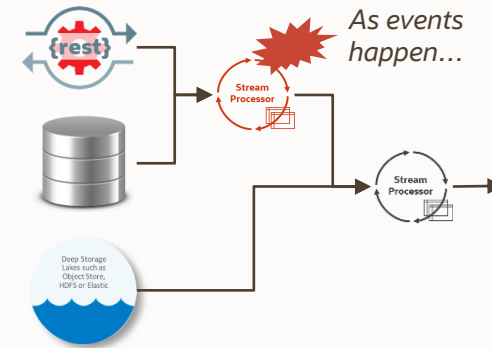


Traditional Data Integration Core Capabilities

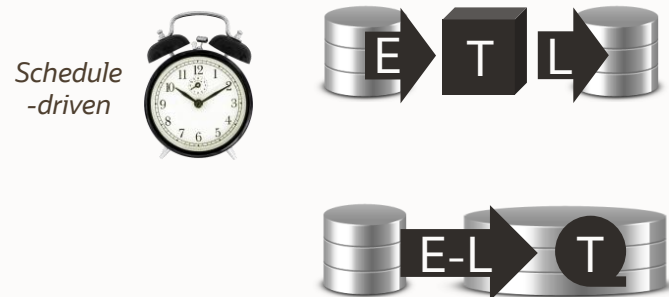
Data Federation



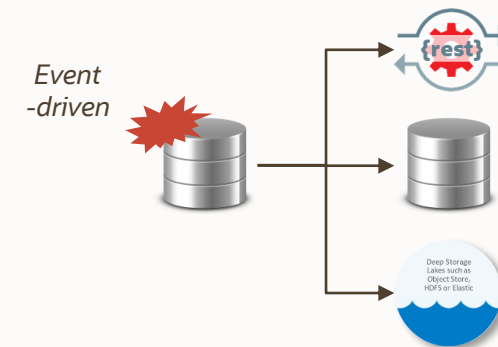
Stream Integration



ETL & ELT



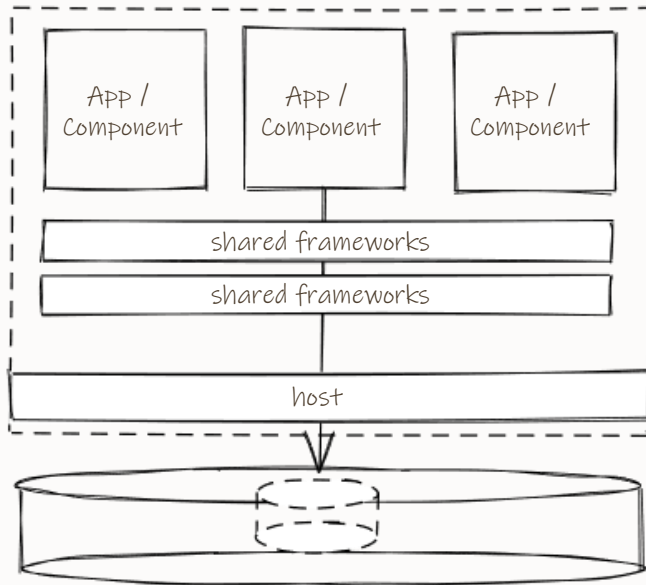
CDC & Replication



Software Design: Monoliths to Microservices

Monoliths

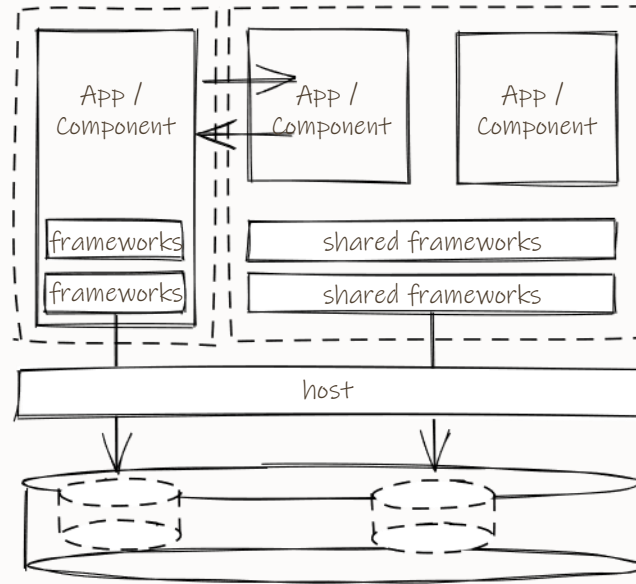
Note: a monolith with REST APIs is still a monolith, and putting a monolith in a container/K8S doesn't make it a microservice!



Classic Monolith

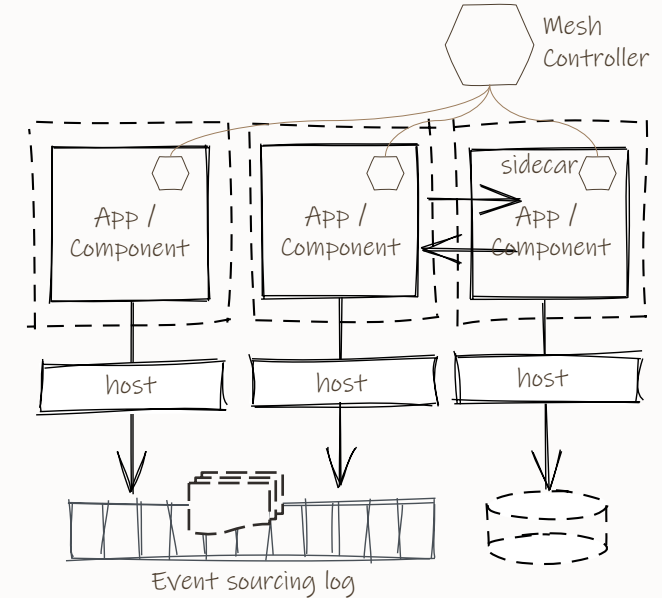
Very coarse grained, many components within single App boundary, layers of shared frameworks, dependencies on one/more schema, single host is often mandatory

Mesh



“Minilith” / Client Server

Coarse grained, some components may be independently upgraded, but cross-component dependencies still generally tightly-coupled. Single host is preferred. Dependencies between Apps and shared schema still exist.

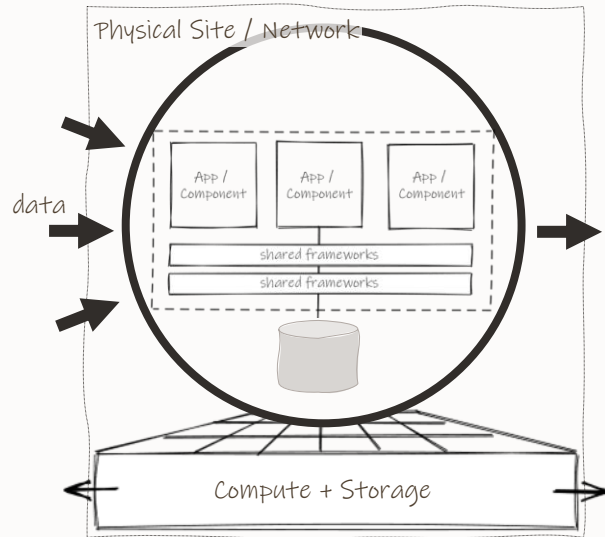


Microservice & Mesh

Component isolation. Encapsulated App schema, or use of Event Sourcing instead. All comms via public APIs. Shared schema is an anti-pattern. Many components may be stateless, serverless/ephemeral

Data Integration Tools Must Also Transition...

Monoliths



Data Hub (Classic Monolith)

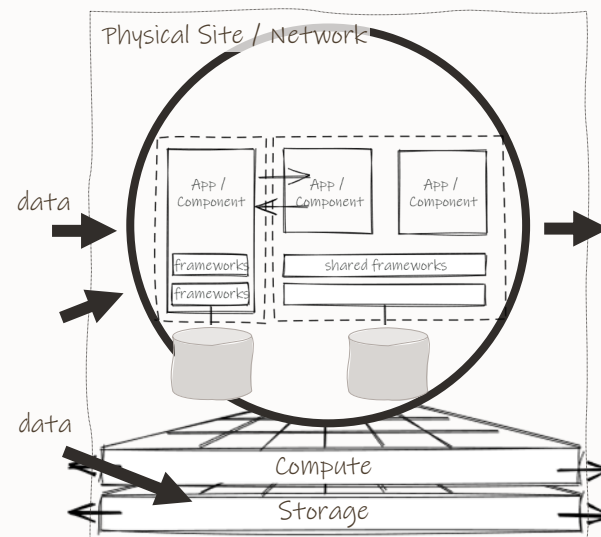
Batch ETL:

- Ab Initio
- DataStage Grid, PowerCenter (pre-10.x)
- Hadoop (CDH, HDP etc)

Streaming / Realtime:

- IBM Streams, Software AG Streams
- Lambda Big Data Architecture (eg: Apache Hadoop + Apache Storm)

Mesh



Data Hub (Minilith)

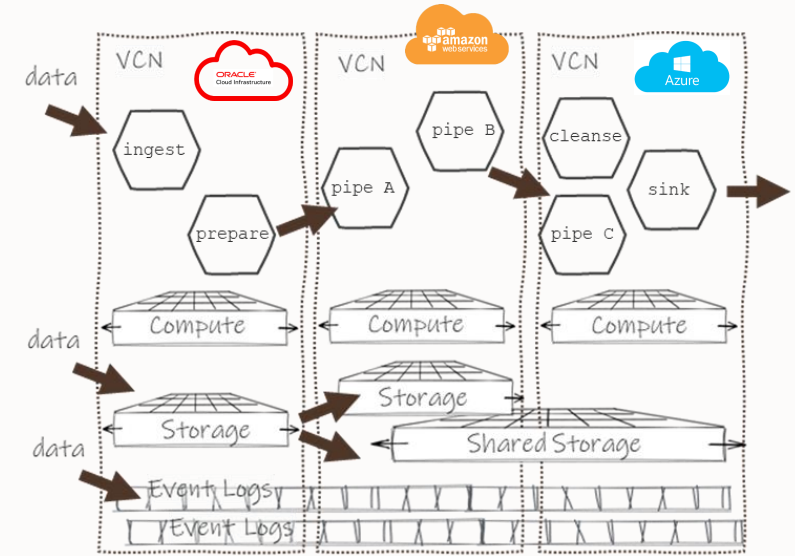
Batch ETL / ELT / Cloud Native:

- PowerCenter (10.x and higher)
- Talend/Stitch, ODI, SAP, SAS etc

Streaming / Realtime:

- Qlik/Attunity, IBM IIDR, etc.
- Kappa Big Data Architecture (including: Confluent KSQL and Flink)

Mesh



Data Mesh (Microservices)

Streaming / Realtime:

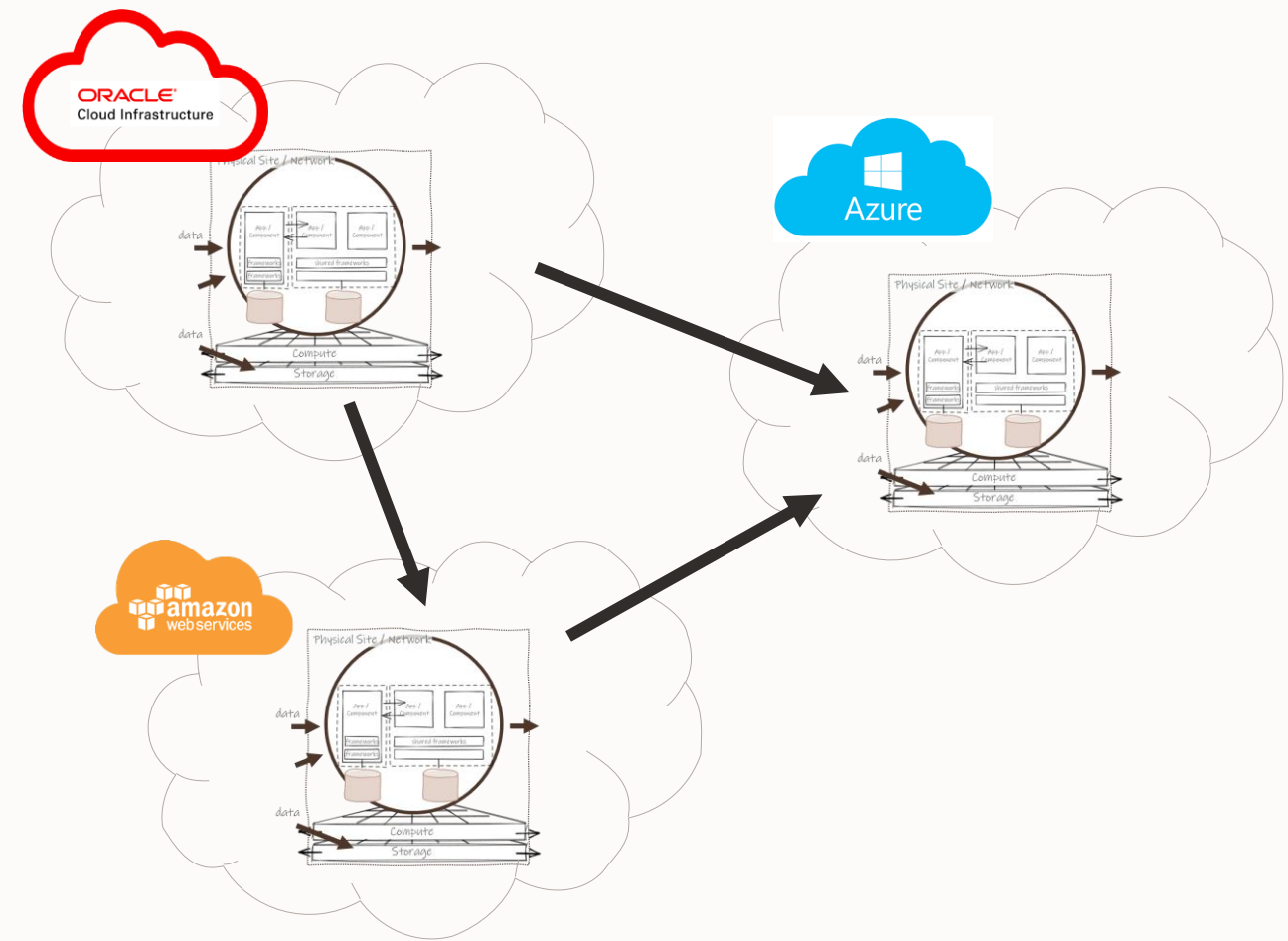
- GoldenGate + GoldenGate Stream Analytics
- AWS Kinesis + Lambda + Glue Streaming
- StreamSets DataOps
- WSO2 Stream Integrator
- Event Sourcing Pattern (bespoke microservices)



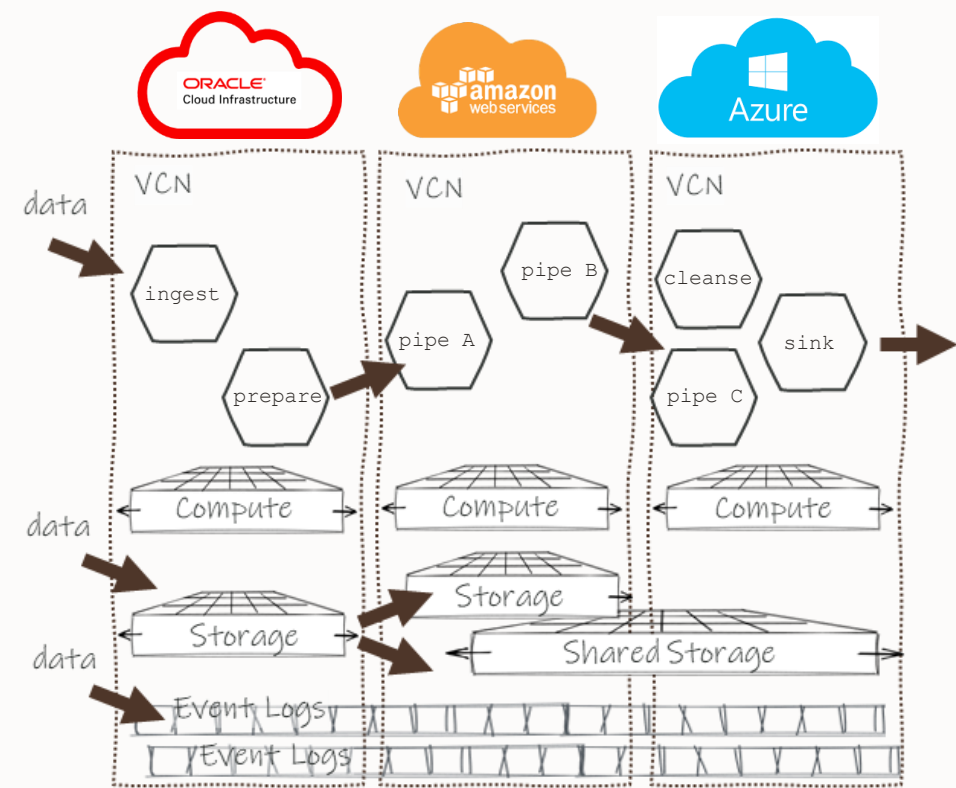
Graph of Hubs != Data Mesh

Monoliths

Note: a monolith with REST APIs is still a monolith, and putting a monolith in a container/K8S doesn't make it a microservice!



Distributed



Benefits of a “real” Data Mesh (vs. monolithic DI Hubs)

Monolithic Data Integration Issues:

- **High Cost of Operations**
 - DevOps - Continuous Integration Continuous Delivery (CI/CD) not possible
 - Integrations usually require Waterfall approach with many dependencies
- **Error-prone Lifecycle Management**
 - Data Hub / Hub and Spoke architecture require tight-coupling, all-or-nothing upgrades, migrations, bug fixes
- **Slow Pace of Innovation**
 - Batch processing is a “lock in”, a chain of dependent batches are only as fast as the slowest link

Data Mesh Changes Things by:

- **Microservices-based Components**
 - CI/CD is possible if the DI software is itself a distributed Mesh of small components
 - Agile methodologies work better when the software is de-coupled
- **De-coupling the Infrastructure**
 - Patterns for Data Mesh are explicitly event-driven thereby taking advantage of serverless & K8S frameworks
- **Shift to a “Event Time” Data Architecture**
 - Timing of data pipelines can be from millisecond to hours, as a business decision not a technical limitation

Agenda

- 1 Brief History of Integration Tools
- 2 Data Mesh as a Next Step
- 3 GoldenGate Strategy for Data Mesh
- 4 GoldenGate Event Stream Processing
- 5 Call to Action



What is a Data Mesh?

Data Mesh is a data-tier architecture to integrate and govern enterprise data assets across distributed multi-cloud environments – three defining characteristics are:

Data Product Oriented:

- *Low code management of data services, data models, governance and easy integrations to analytics / data science*

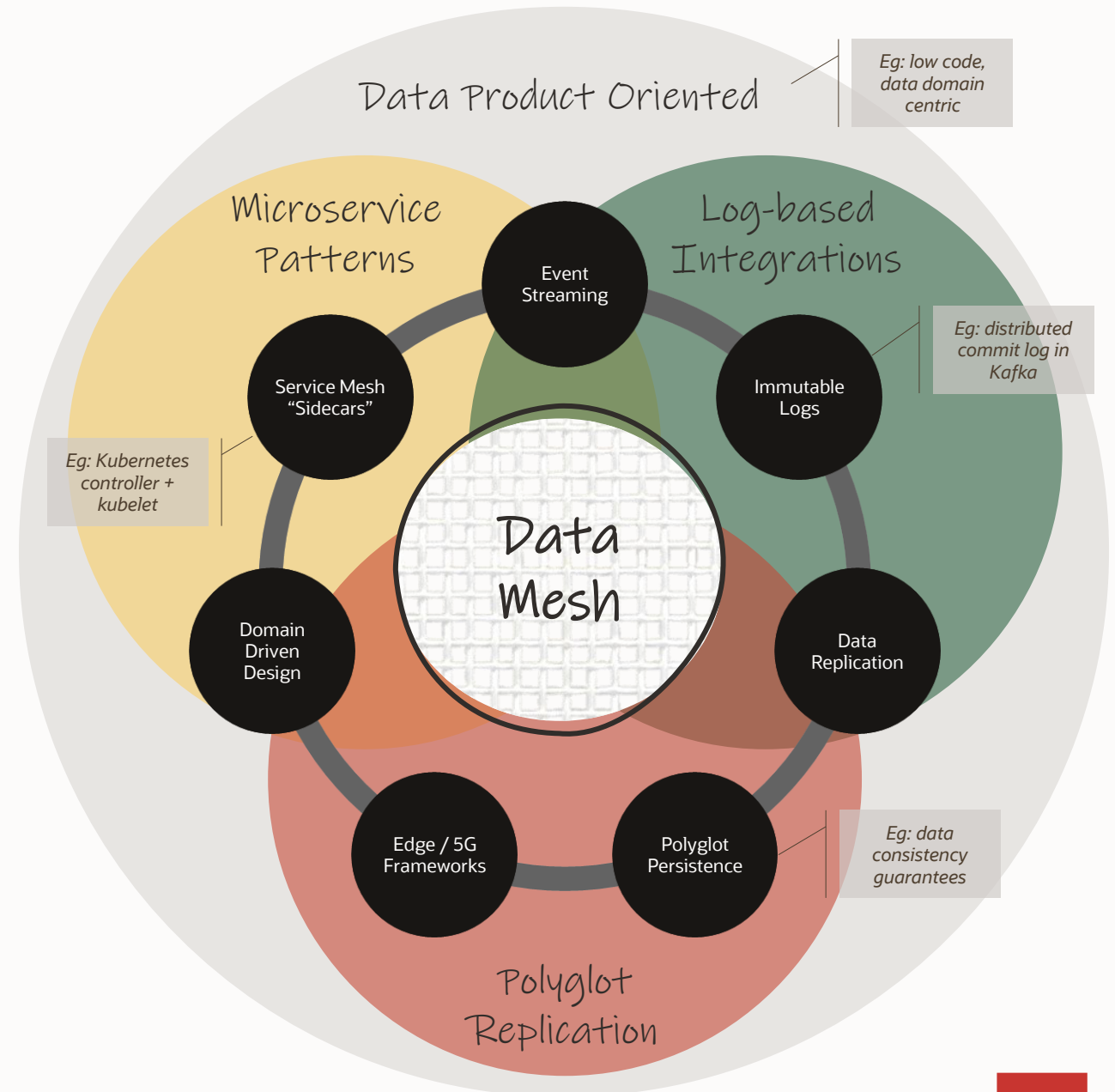
De-Centralized Processing:

- *De-centralized data processing; no ETL/Hubs/Lake monoliths*
- *Microservices / Service Mesh deployments, utilization of “sidecar proxy” patterns, encapsulation, etc.*
- *Simplified continuous integration continuous delivery (CICD) and lifecycle management (LCM) across public/private clouds*

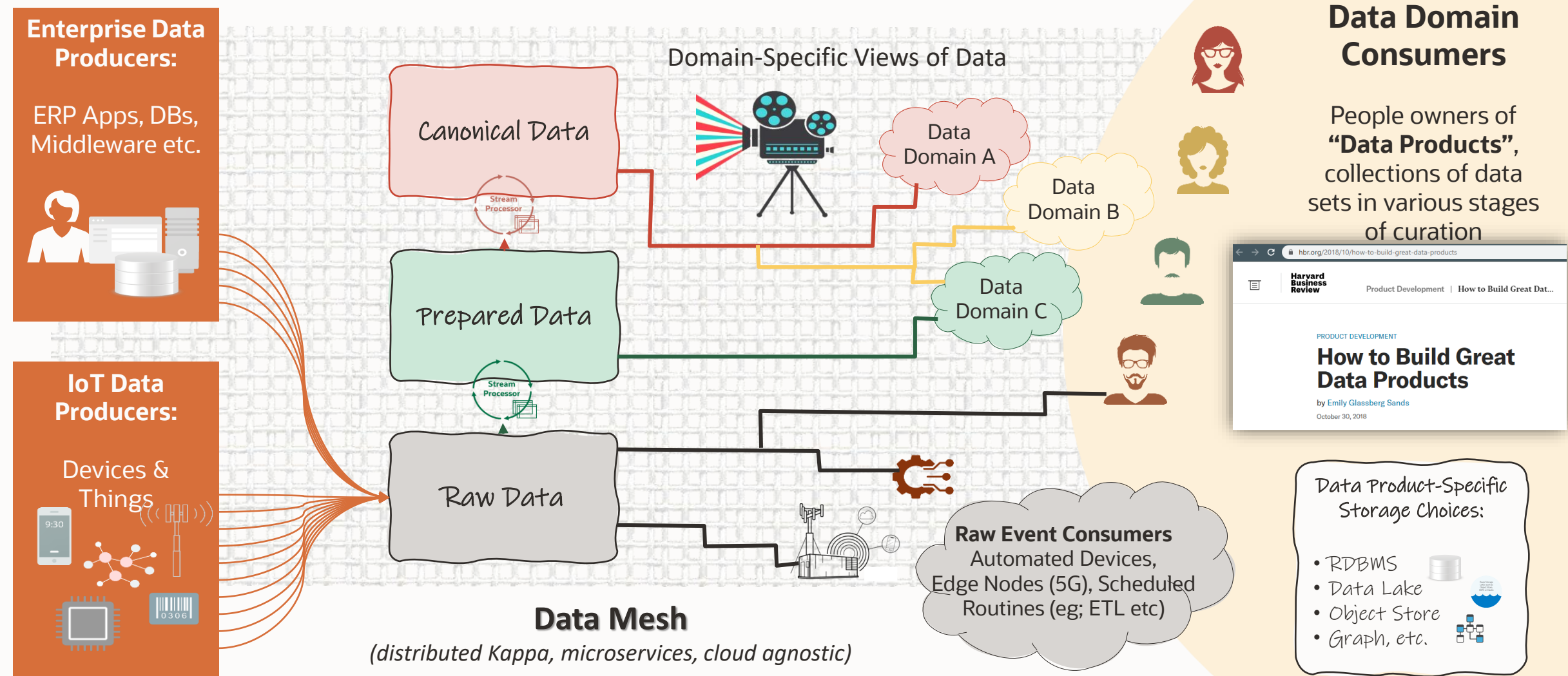
Event-Driven, Stream Centric:

- *Real-time by default, batch patterns only when necessary*
- *Immutable event logs for messaging and data store events*
- *Semantics for consistent (ACID) and inconsistent data sets*

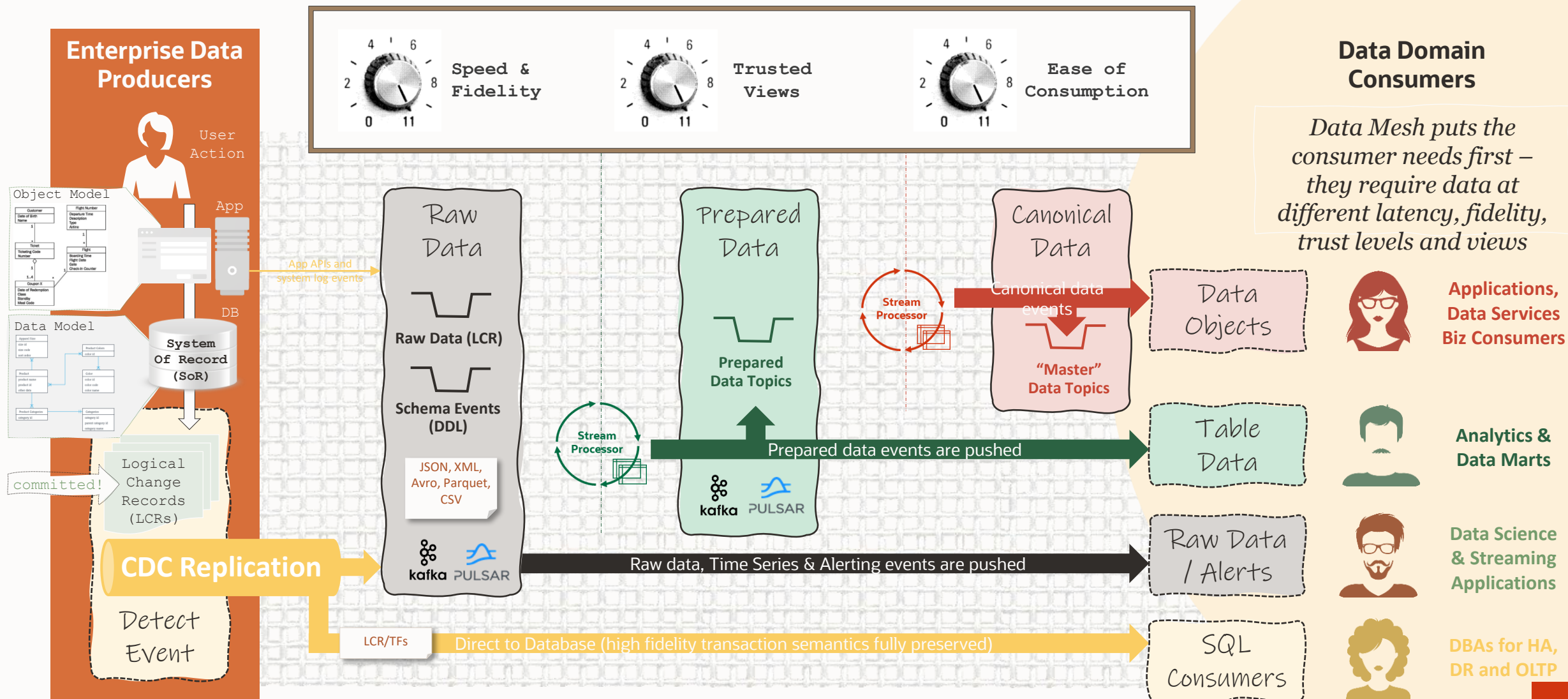
<https://en.wikipedia.org/wiki/ACID>



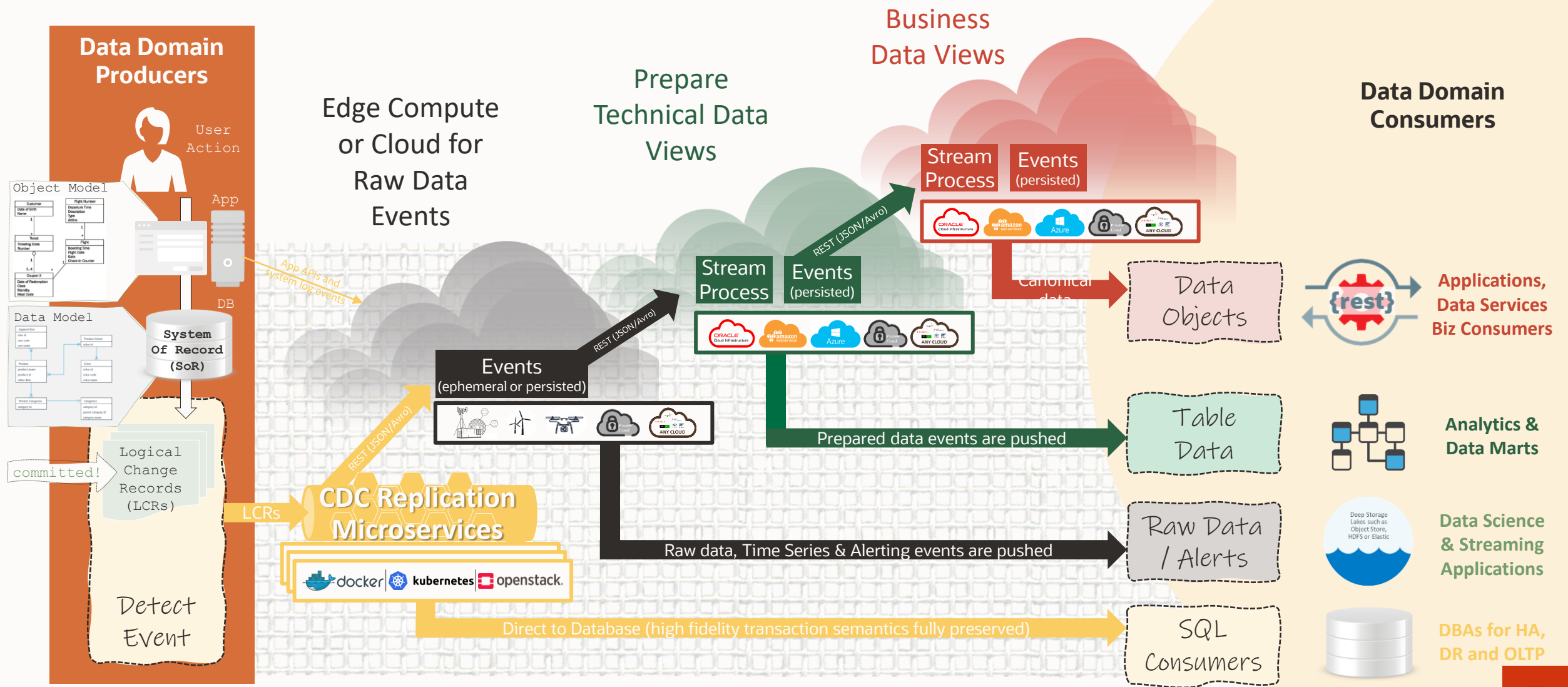
Data Mesh Conceptual View – Data Domains



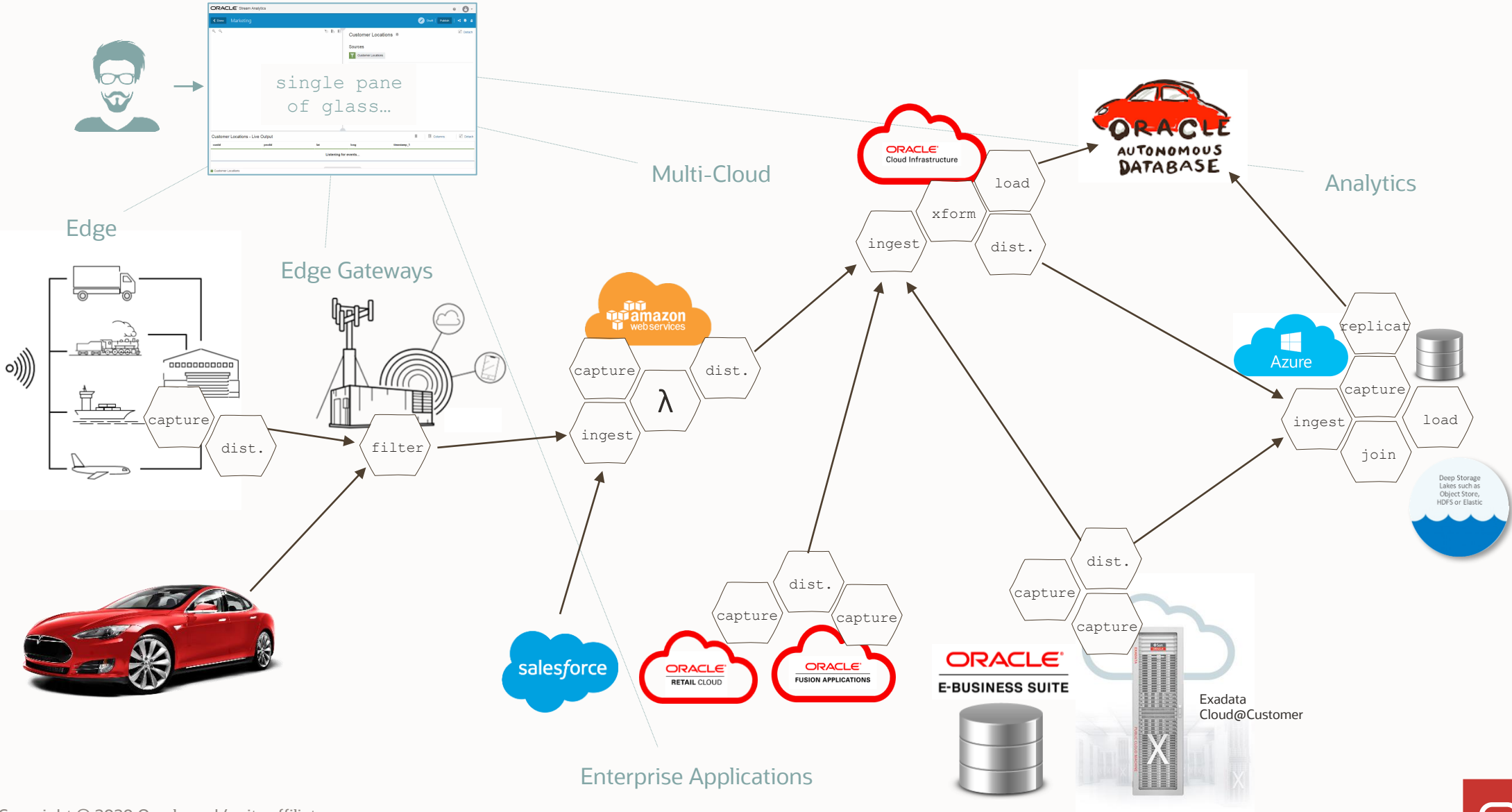
Consumer-Driven, Event-Centric Data Mesh



Distributed by Design, Microservices Based

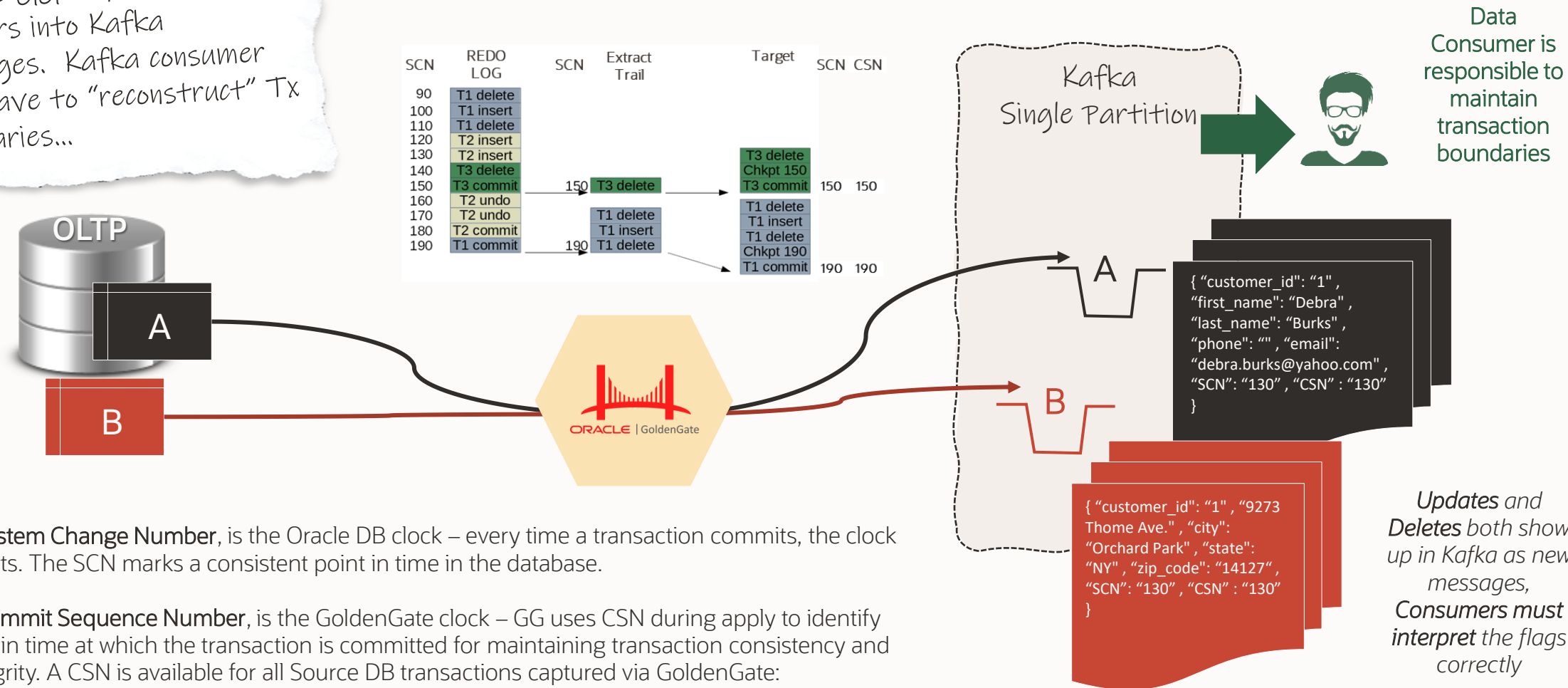


Example 1: Mesh of Data Integration Microservices



Example 2: Maintain Data Consistency in Pipelines

Include GG sequence numbers into Kafka messages. Kafka consumer may have to "reconstruct" Tx boundaries...

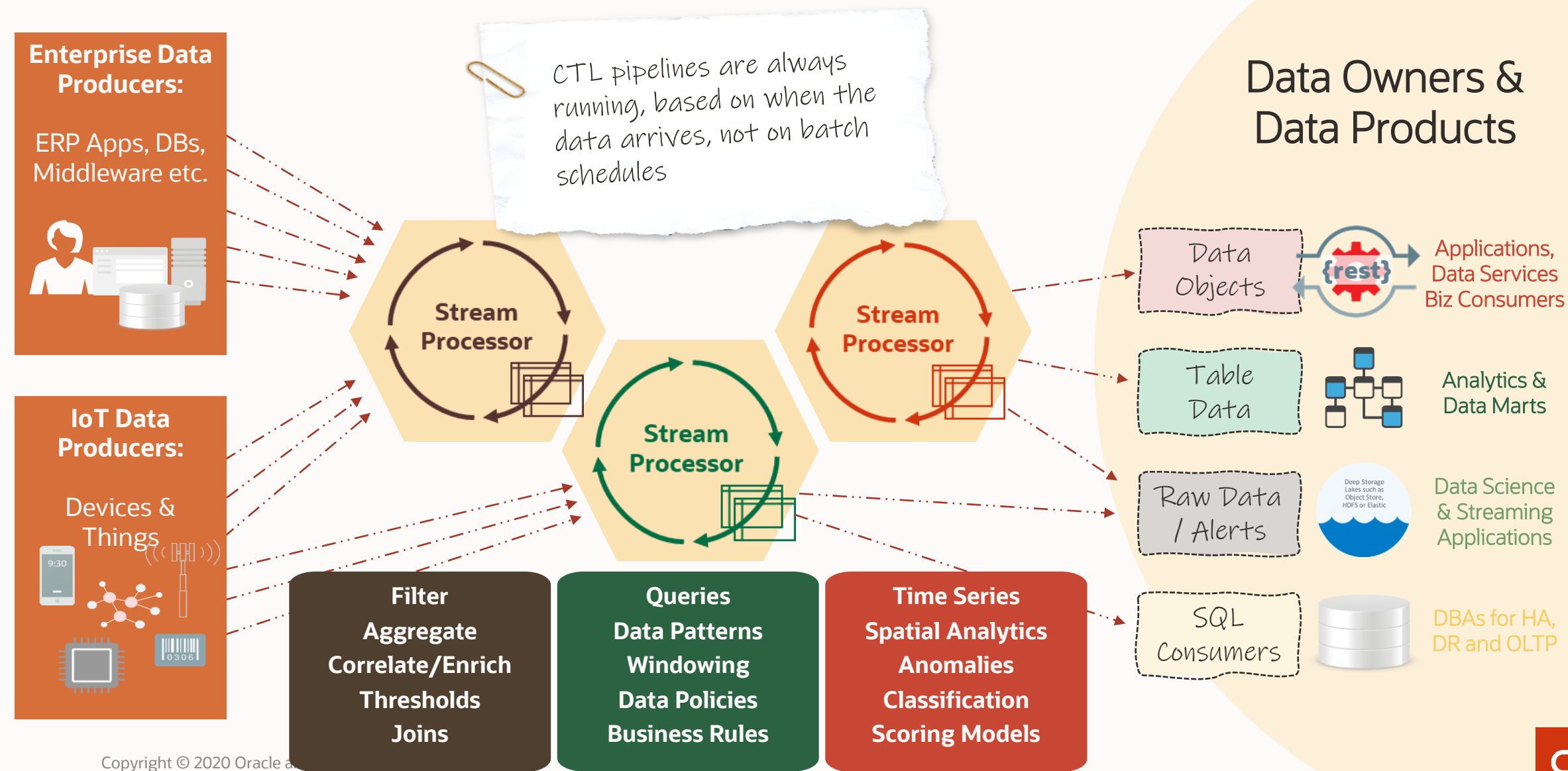


SCN – System Change Number, is the Oracle DB clock – every time a transaction commits, the clock increments. The SCN marks a consistent point in time in the database.

CSN – Commit Sequence Number, is the GoldenGate clock – GG uses CSN during apply to identify the point in time at which the transaction is committed for maintaining transaction consistency and data integrity. A CSN is available for all Source DB transactions captured via GoldenGate:
<https://docs.oracle.com/en/middleware/goldengate/core/19.1/admin/commit-sequence-number.html>

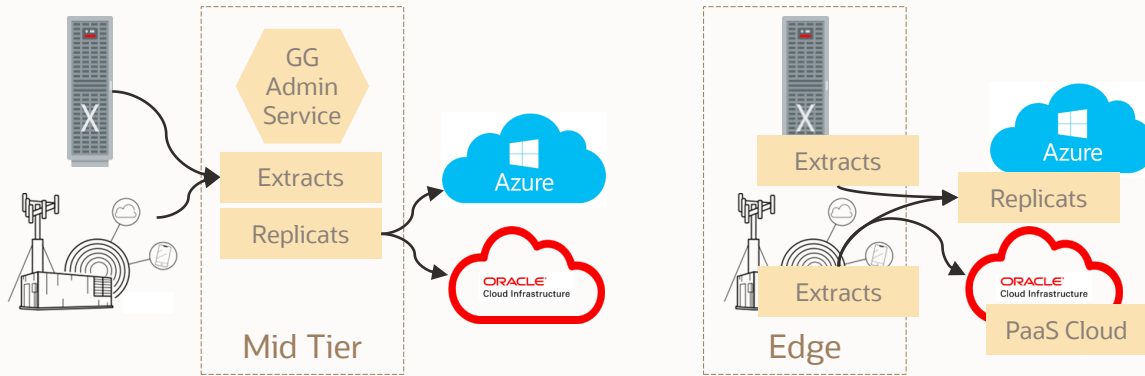


Example 3: Continuous Transformation and Loading (CTL)

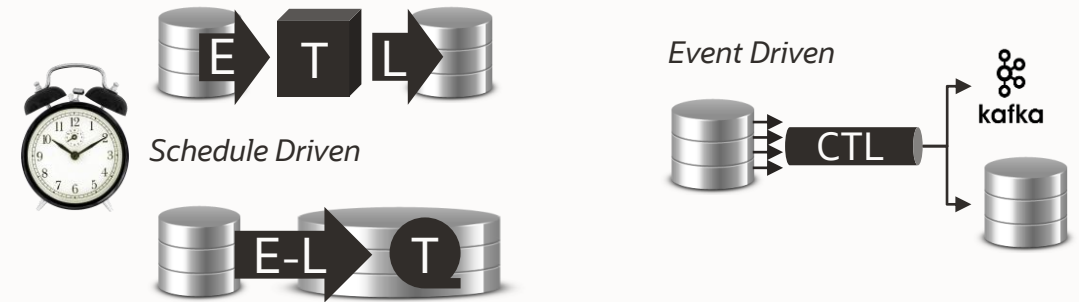


Top “Tactical IT” Data Mesh Opportunities

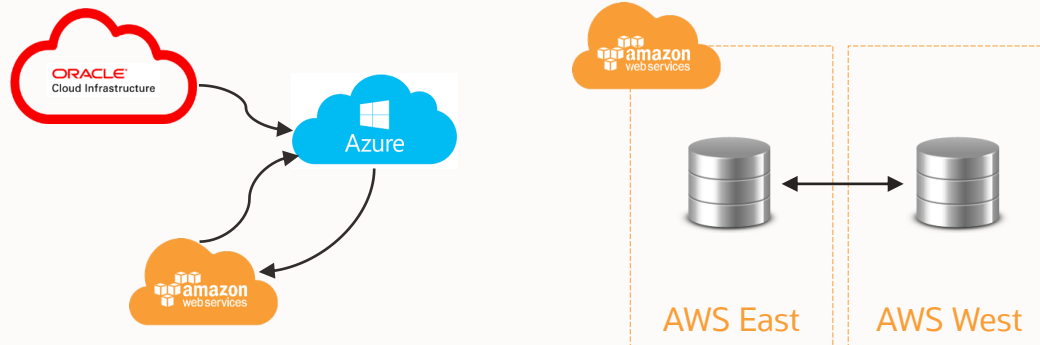
Distributed Data Lake Ingestion



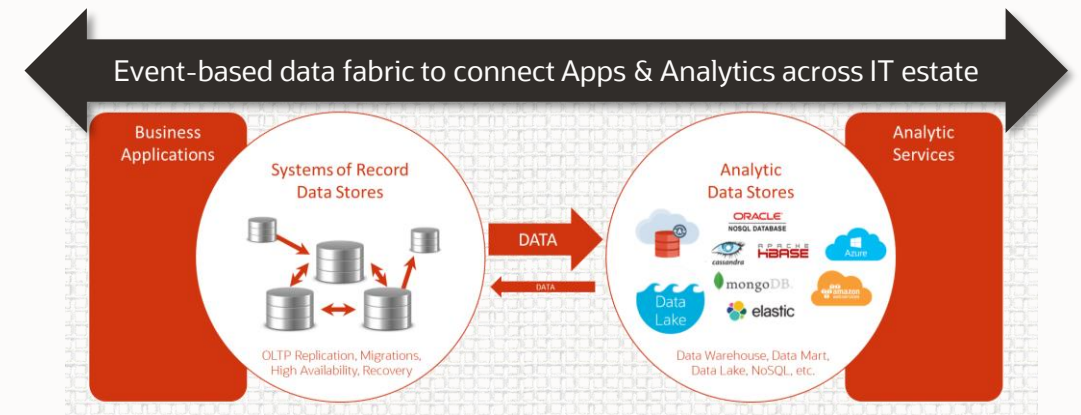
Move From Batch ETL to Streaming CTL (Continuous Transformation and Loading)



Multi-Cloud (VCN) Operational Replication

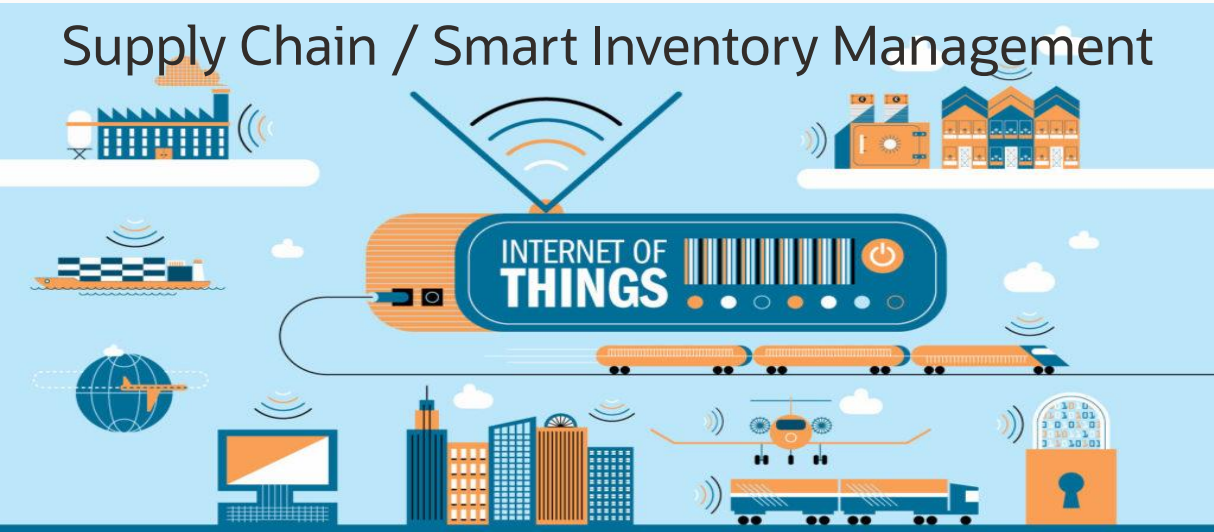
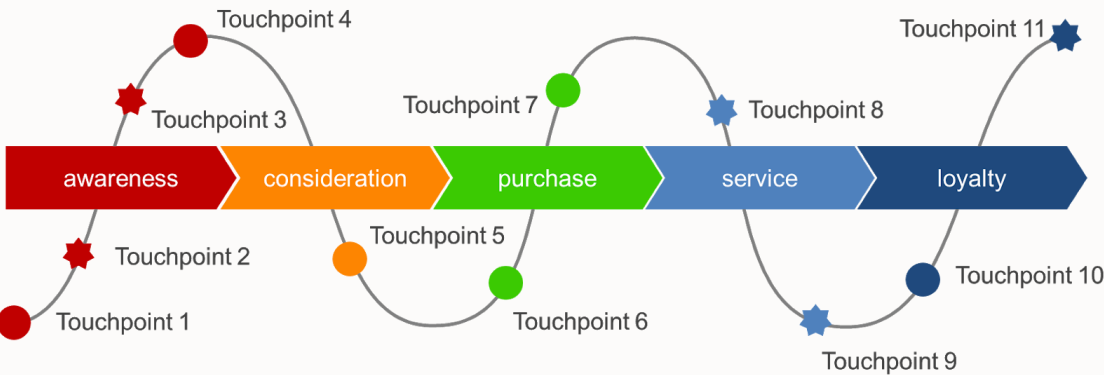


DataOps for Microservices Apps & Analytics



Top Data-Driven Business Transformation Solutions

Next Best Action / Realtime Offers



Logistics / Location Intelligence



Agenda

- 1 Brief History of Integration Tools
- 2 Data Mesh as a Next Step
- 3 GoldenGate Strategy for Data Mesh
- 4 GoldenGate Event Stream Processing
- 5 Call to Action



GoldenGate Microservices

GoldenGate is itself a set of microservices that human users or other services may interact with

Embedded User Interface:

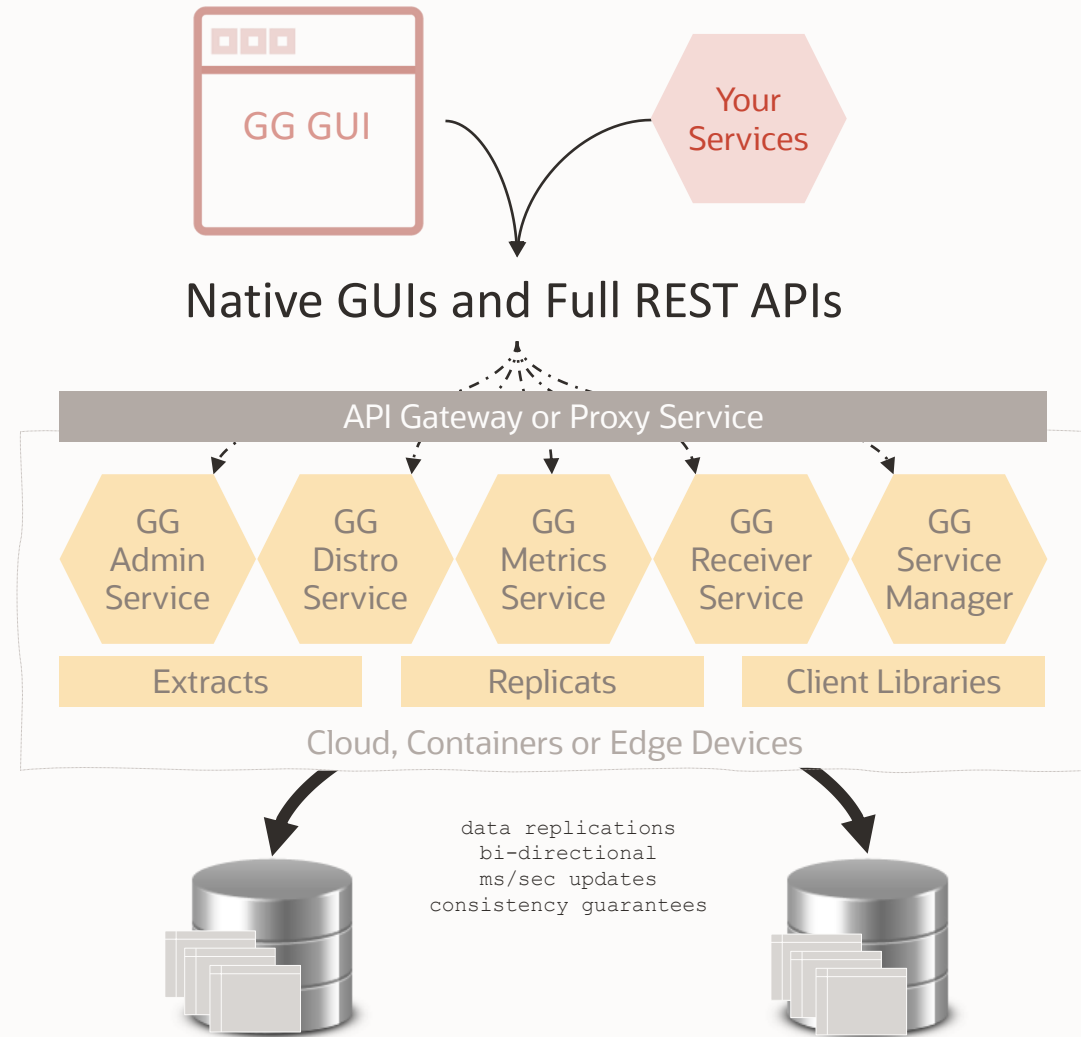
- *C-based microservices with embedded HTTP client for native JavaScript based GUI*
- *Oracle Jet frameworks for intuitive interaction model*

REST native APIs:

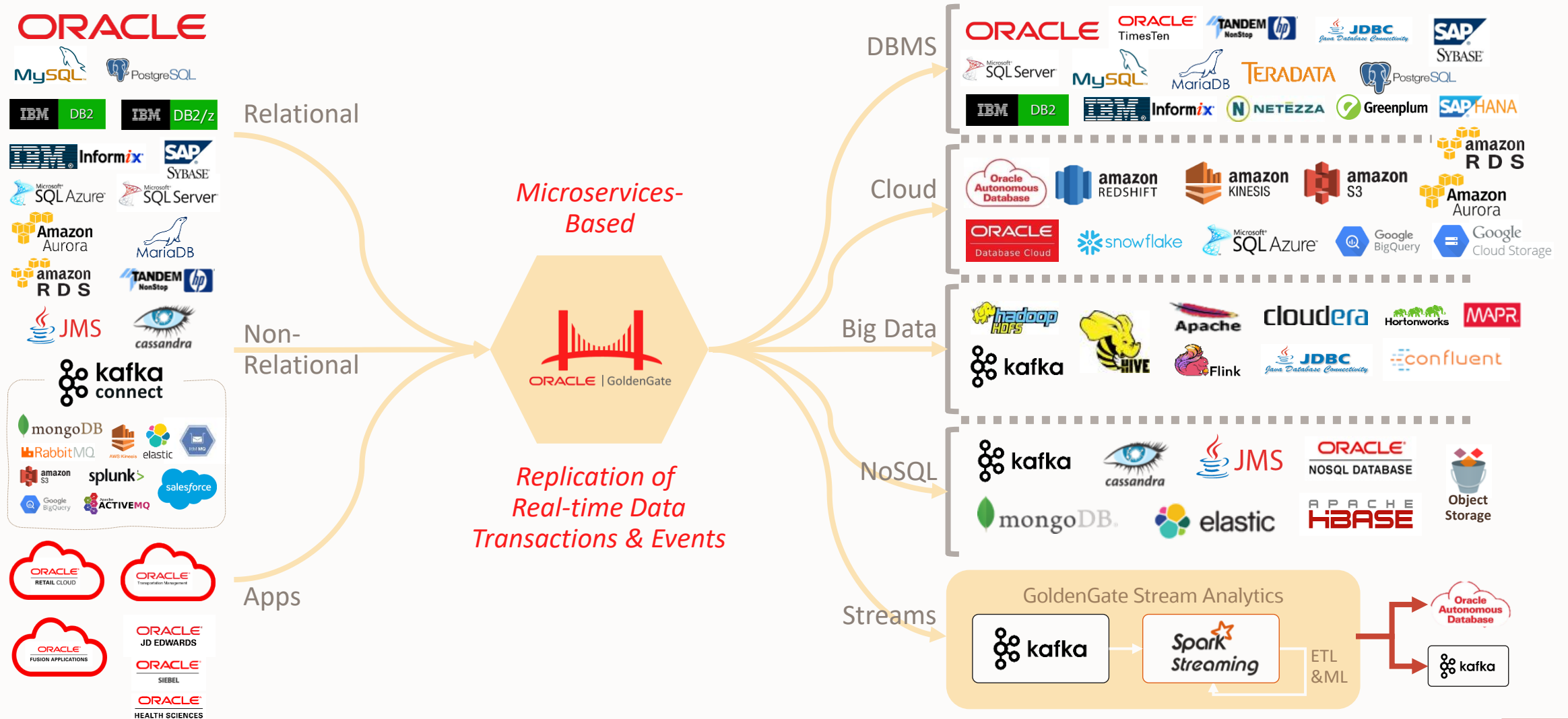
- *Fully REST native*
- *Also available, a Command Line Interface (CLI) produces REST calls to the native services*

Full GoldenGate Replication Capabilities:

- *100% coverage for all traditional GG replication patterns; fully capable of HA/DR use cases*



Transaction Outbox for the Whole Enterprise

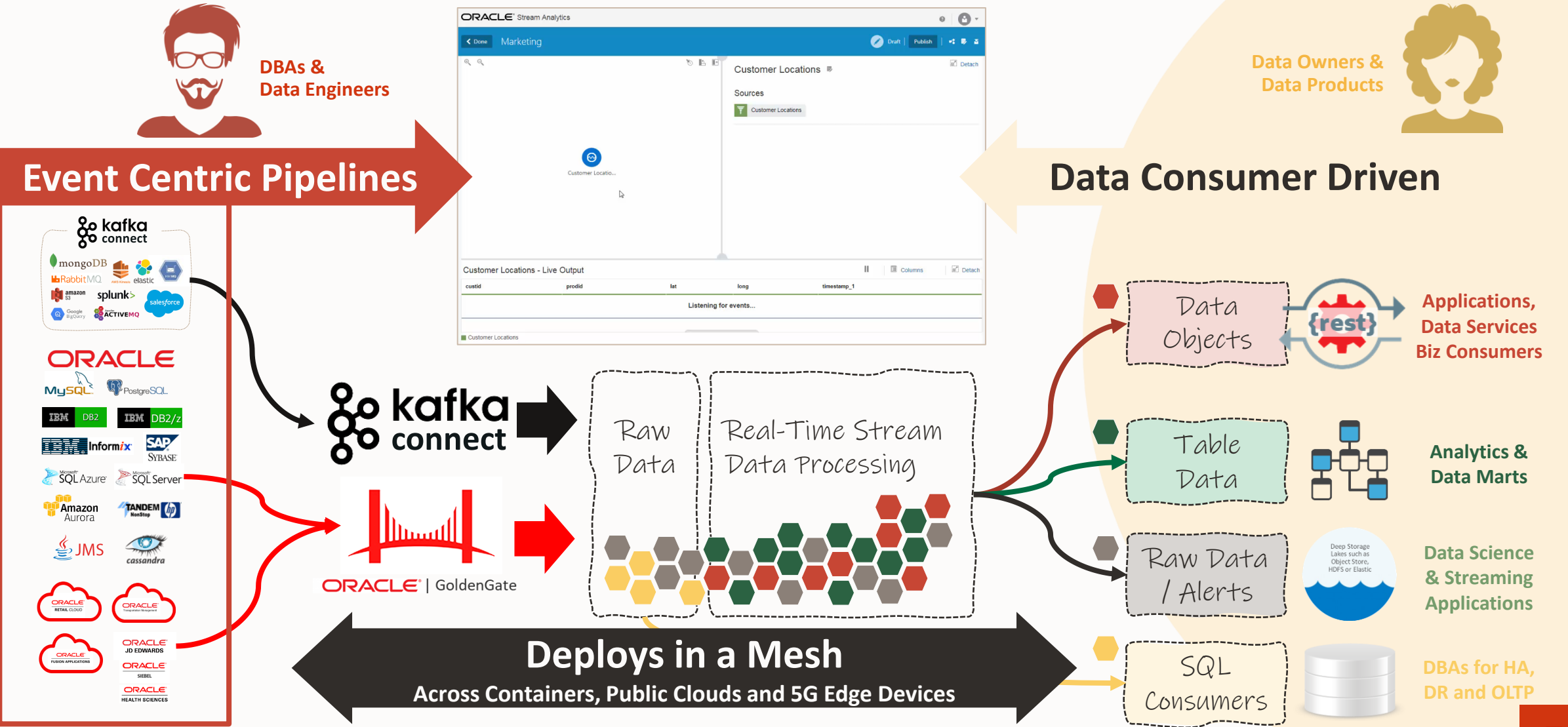


<https://www.oracle.com/middleware/technologies/goldengate.html>

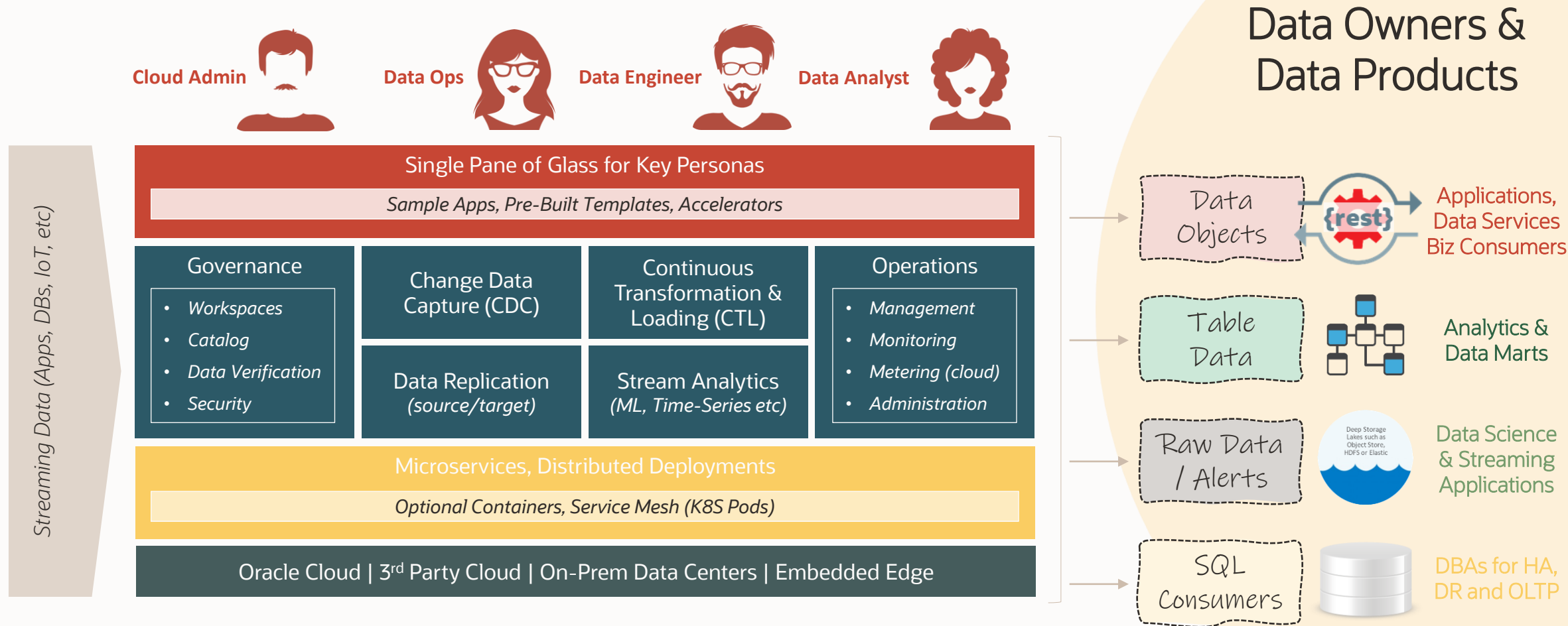
Copyright © 2019 Oracle and/or its affiliates.



Single Pane of Glass for Real-Time Data Mesh

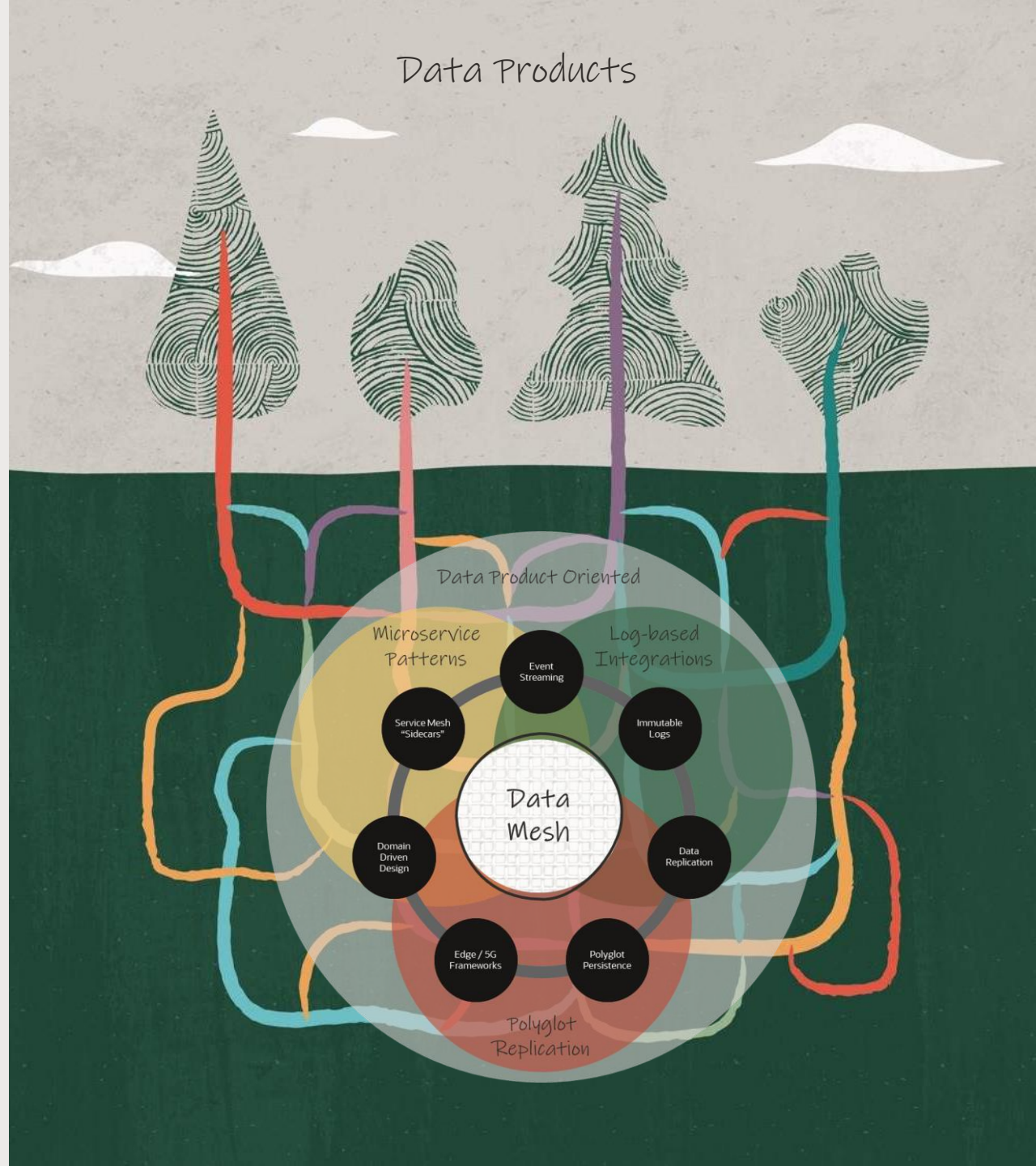


GoldenGate Direction: Mesh Platform for Fast Data



Oracle Focus for Data Mesh: Consistent Enterprise Data

- ERP Applications:
 - Ledger, HR, Supply Chain
- Financial & Ecommerce Apps
- Inventory and Logistics Data
- Operational Data Availability
- Consistent Data Stores
 - Trusted Data Lakes
 - Dependable Analytics

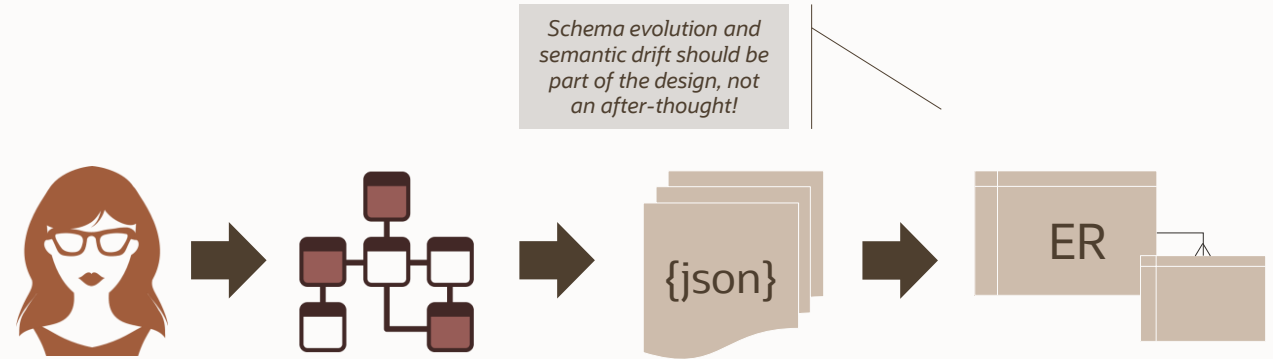


A Note about Data Fidelity and Impedance Mismatch

Impedance Mismatch:

As a **User** interacts with the **Application Object Model**, data may be serialized as **Message Payloads** or persisted in **Database Tables**.

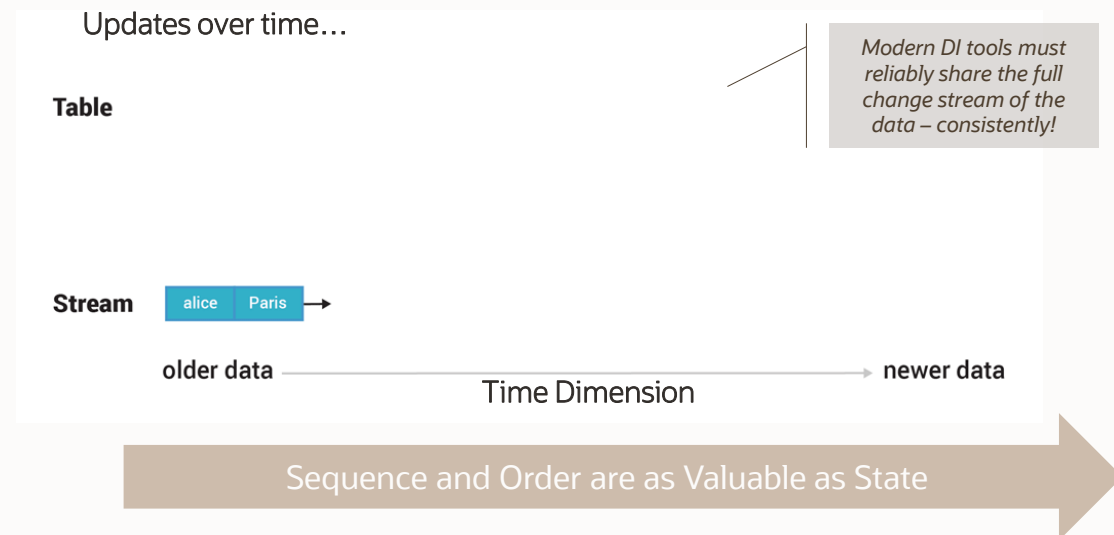
Impedance mismatch can lead to lossy semantics → each time data formats change, there is often **some loss of meaning at the data level**.



Data Fidelity:

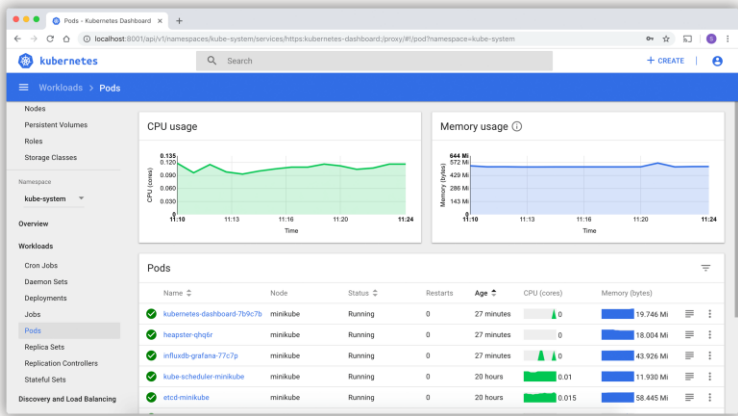
A **SQL view** on a **Table**, or a **batch ETL** job, only tell a small part of the “data story”... you are only seeing the **state of the data at a point-in-time**.

It is the Commit Log (eg; the change stream) that tells the whole story, **a full narrative of the data**.

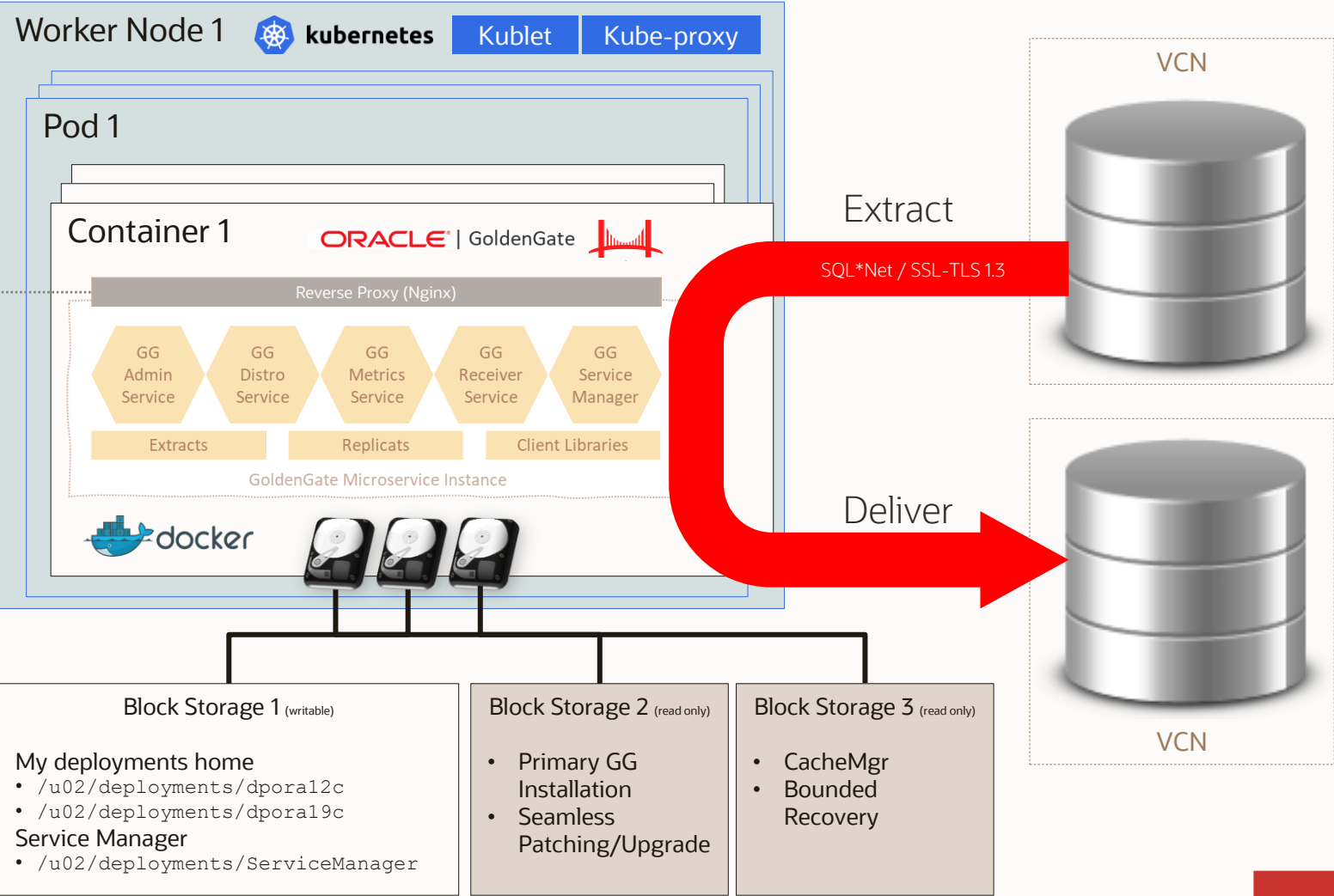
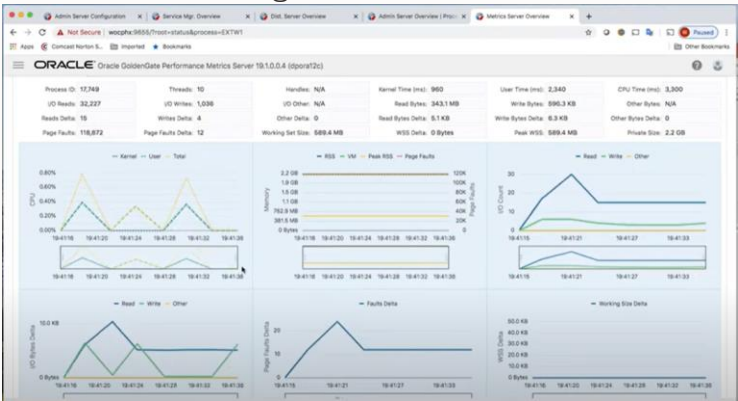


Running GoldenGate in a Service Mesh

Pod Management



GoldenGate Management

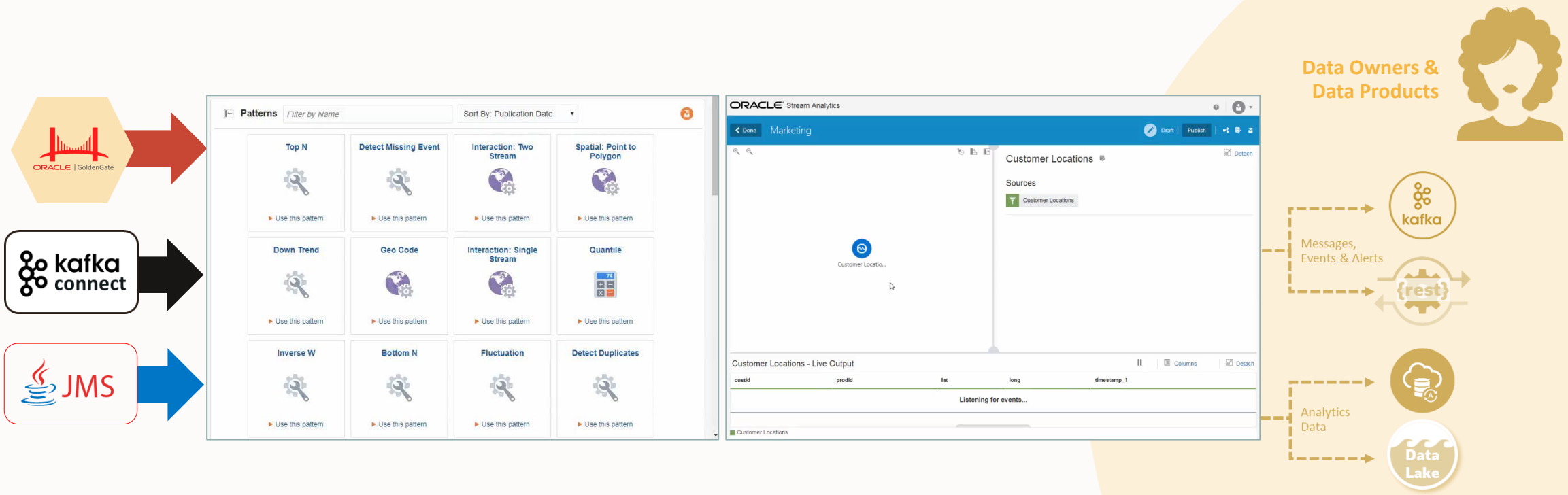


Agenda

- 1 Brief History of Integration Tools
- 2 Data Mesh as a Next Step
- 3 GoldenGate Strategy for Data Mesh
- 4 GoldenGate Event Stream Processing
- 5 Call to Action



Oracle GoldenGate Stream Analytics

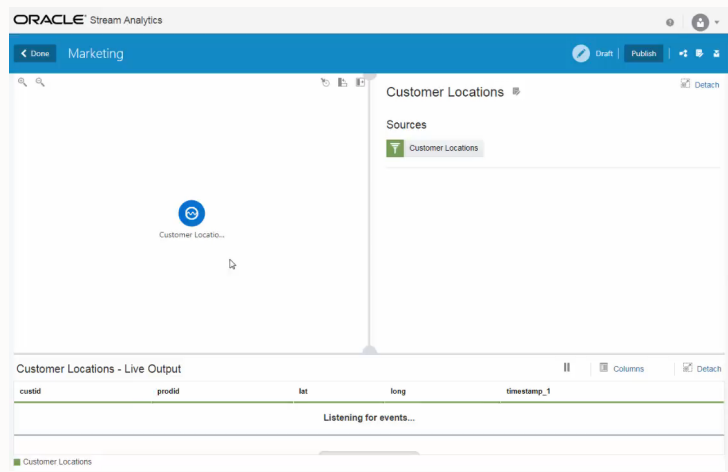


Ingest Events	Select Processing Patterns	Build Event Pipelines	Serve Data Downstream
<i>GoldenGate events from any DB or NoSQL; also supports events from Kafka, JMS, AQ and MQTT</i>	<i>Rich set of pre-built patterns will dramatically improve developer efficiency and time-to-value</i>	<i>Easily leverage geo-fencing, machine-learning, and other reference data within the stream</i>	<i>Data can be delivered out to Kafka, databases, or easily staged for external ETL jobs</i>

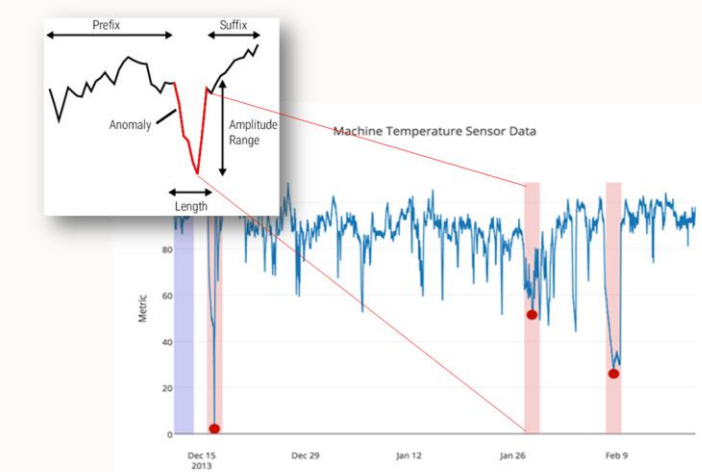


GoldenGate Stream Analytics

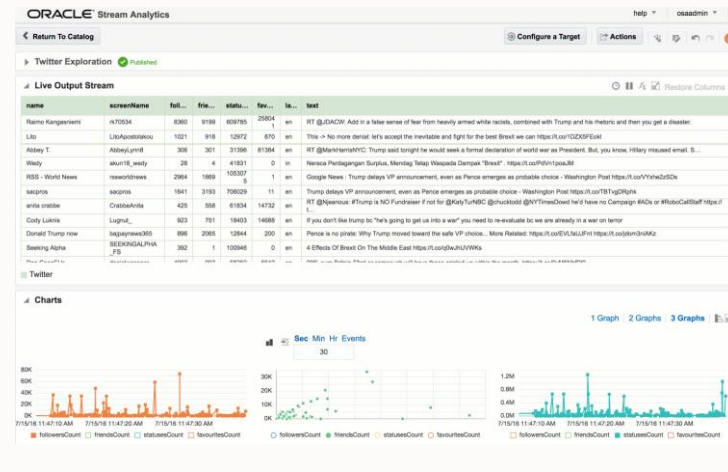
Interactive, Low-Code Designer



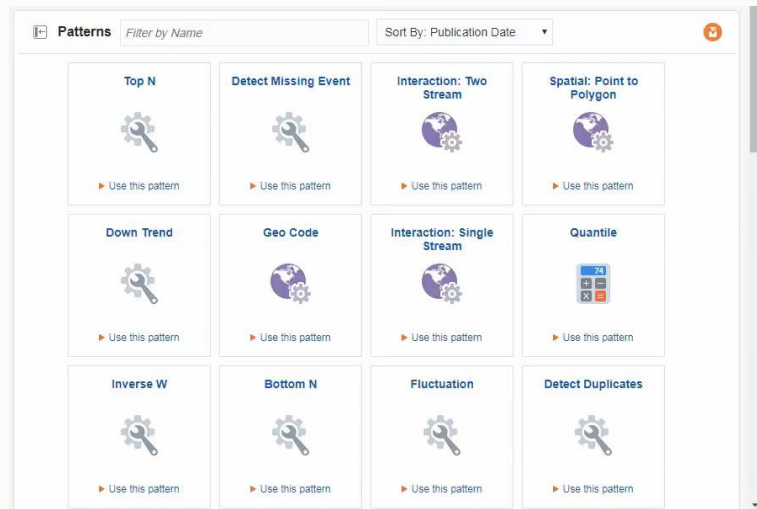
Time Series Analytics



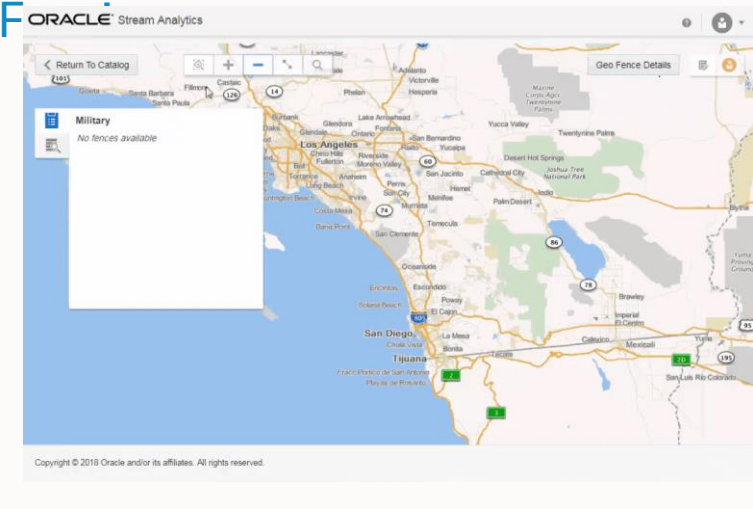
Event Driven Dashboards



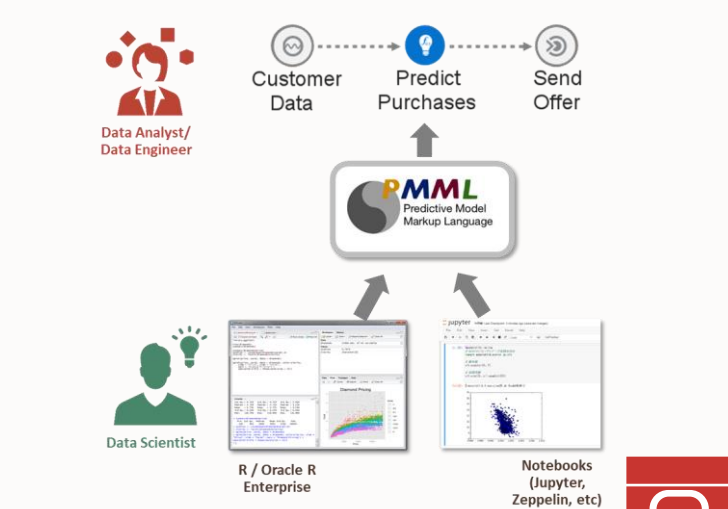
Patterns / Accelerators



Geo-Spatial Analysis & Geo-



Predictive Analytics / ML



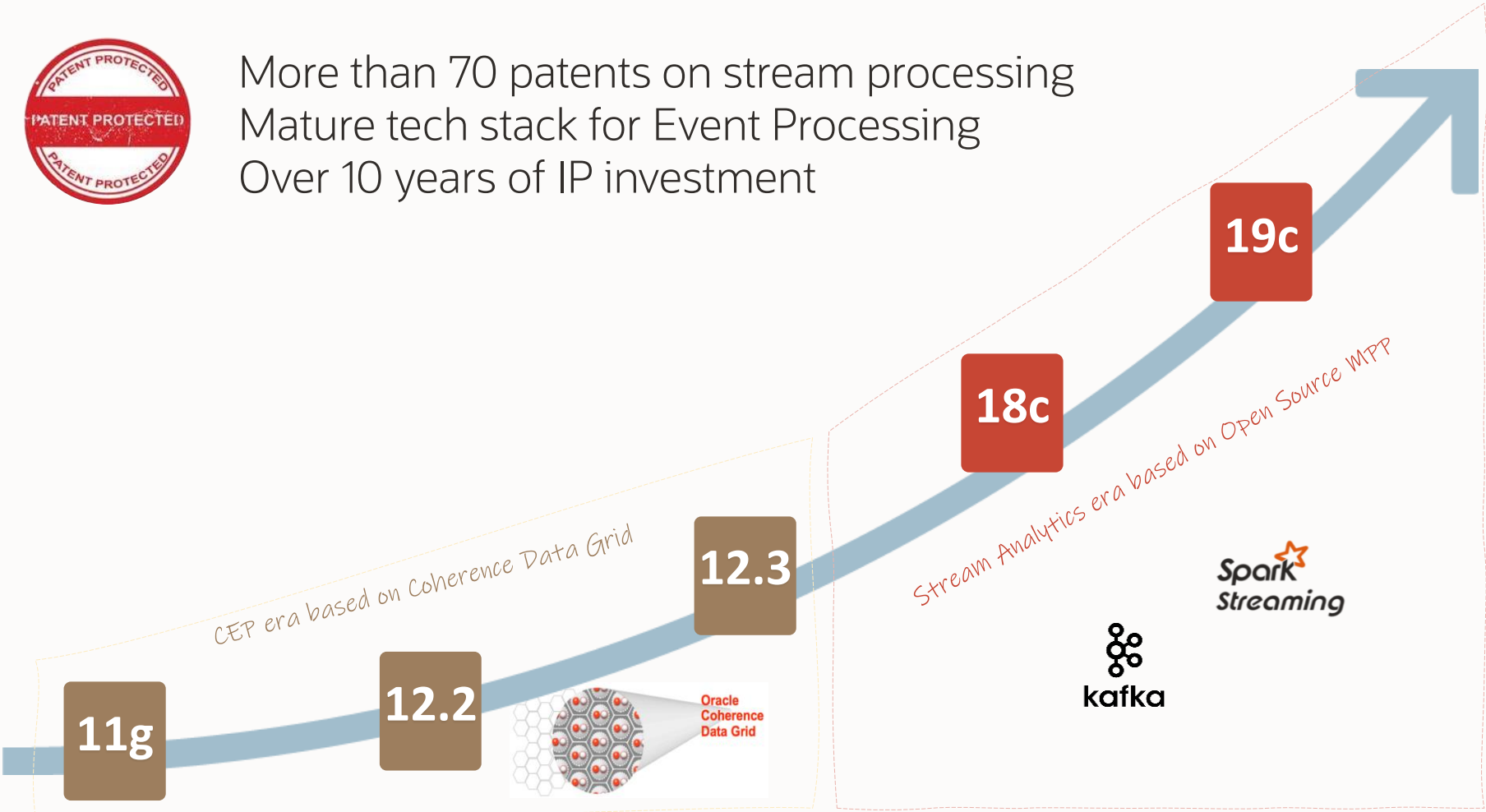
Stream Processing Leadership



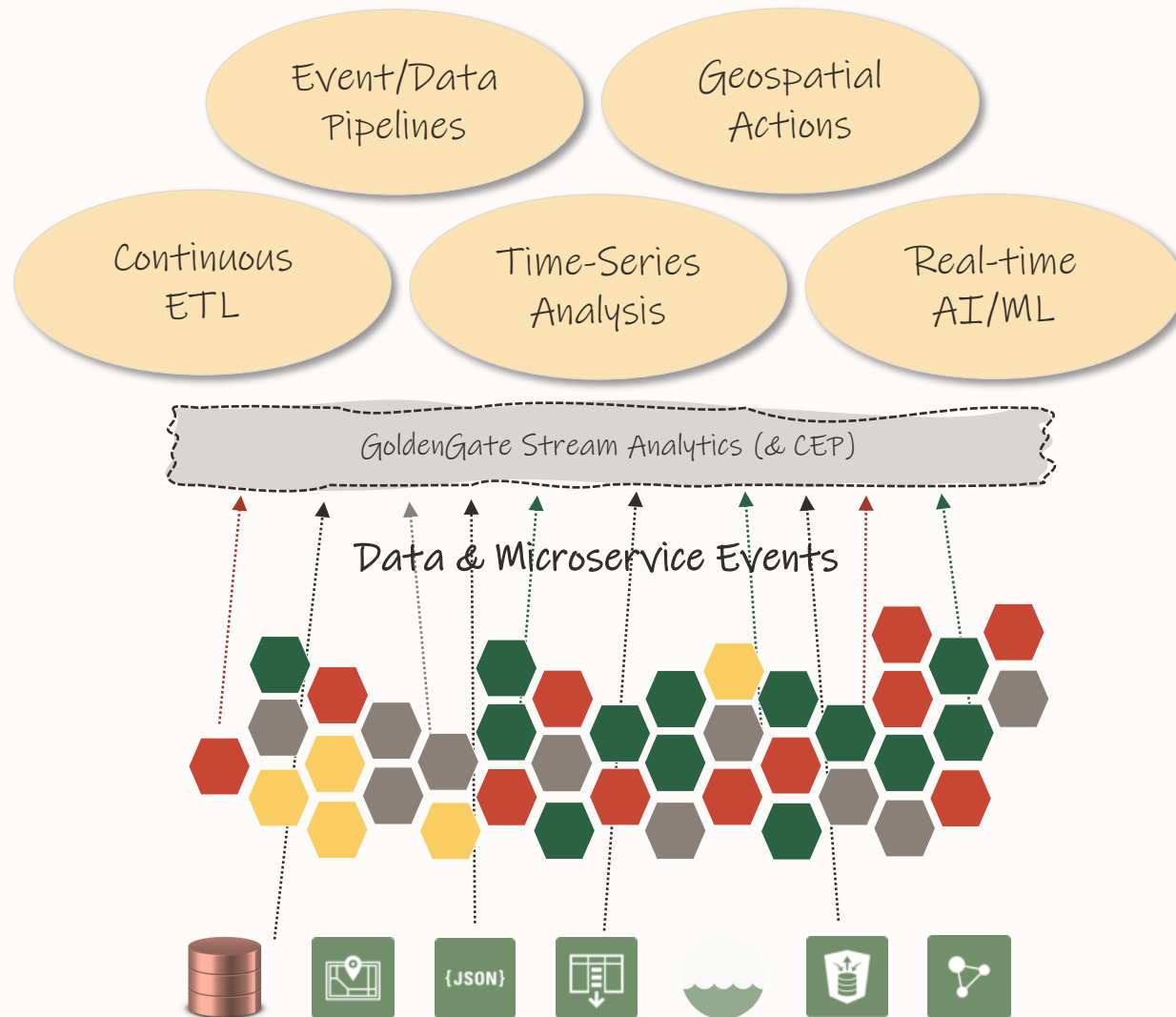
Leader for global initiative on streaming standards



More than 70 patents on stream processing
Mature tech stack for Event Processing
Over 10 years of IP investment



Use Cases for Event Driven, Stream Data Processing



There has been a widespread awakening to the benefits of Event Drive Architecture (EDA) for increasing the scalability and agility of business systems. [...] Stream analytics is based on the mathematics of complex-event processing (CEP). CEP is a computing technique in which incoming data about what is happening (event data) is processed as it arrives (data in motion or recently in motion) to generate higher level, more useful, summary information (complex events).

W. Roy Schulte (of Gartner), March 2020:
[EDA is Suddenly Popular Will Stream Analytics be Next?](#)



Agenda

- 1 Brief History of Integration Tools
- 2 Data Mesh as a Next Step
- 3 GoldenGate Strategy for Data Mesh
- 4 GoldenGate Event Stream Processing
- 5 Call to Action



What Next?

Ask your Oracle Rep for a demo!

Oracle #1 in Data Fabric Strategy

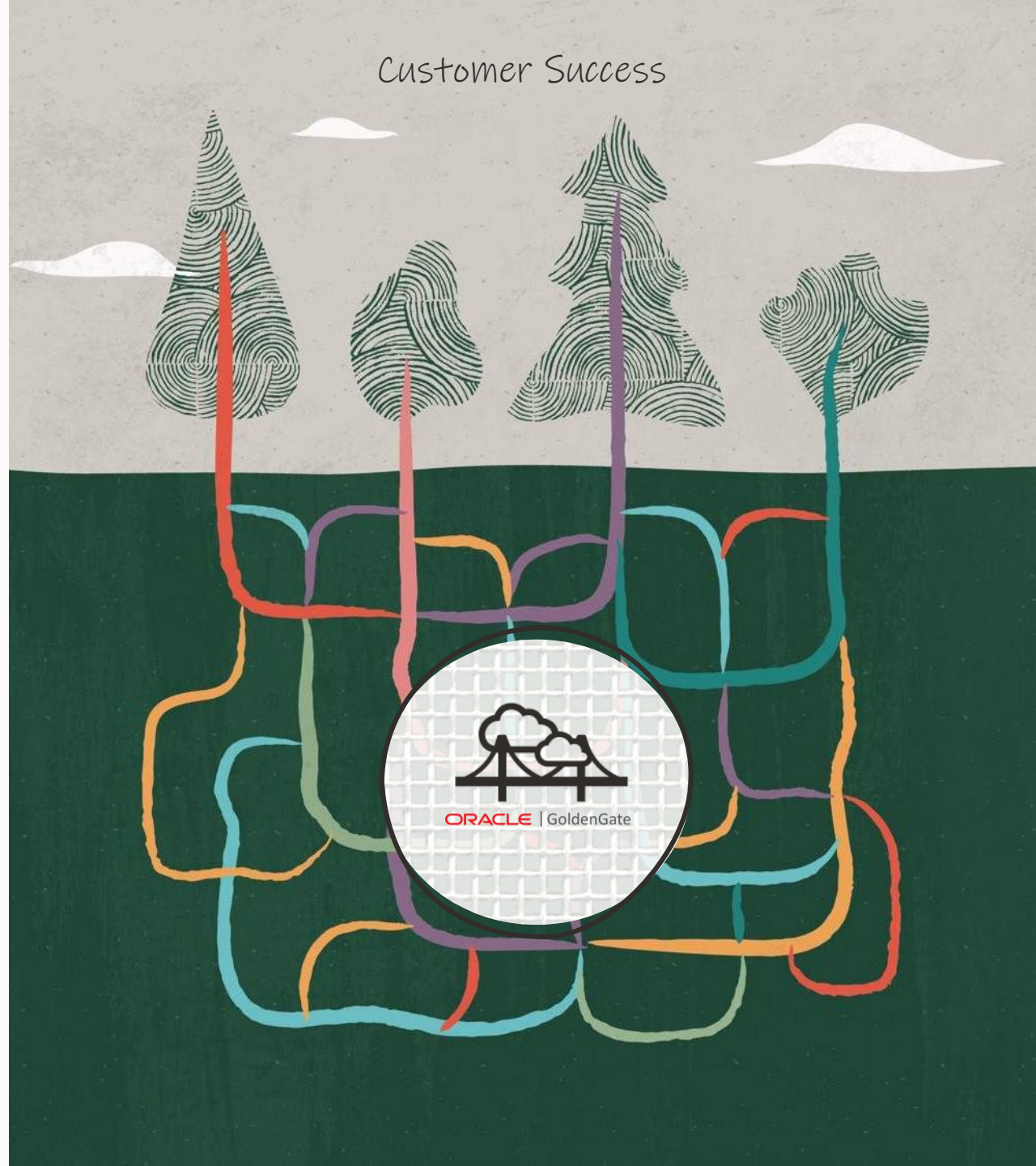
<https://blogs.oracle.com/dataintegration/oracle-forresterwave-datafabric-2020?xd=co&f=66bcf41f-e285-4ccc-a5b5-1c790cab0db0>

GoldenGate YouTube | Data Mesh:

<https://www.youtube.com/playlist?list=PLbqmhpwYrIZJ-583p3KQGDAd6038i1ywe>

Free Trial of GoldenGate Streaming:

https://cloudmarketplace.oracle.com/marketplace/en_US/listing/70961838



ORACLE



Our mission is to help people see
data in new ways, discover insights,
unlock endless possibilities.

