ORACLE

Event Management
Cloud Lift Services - Emerging Tech

Pattern Definition Document

October, 2021 | Version 1.0

1 VERSION CONTROL

Version	Author	Date	Comment
1.0	Giovanni Conte	Nov 3rd, 2021	Update after internal review
0.5	Giovanni Conte	Oct 13th, 2021	Review and better design of physical view
0.4	Giovanni Conte	Oct 6th, 2021	Added draft physical view
0.3	Giovanni Conte	Oct 6th, 2021	Added steps and link to workshops
0.2	Giovanni Conte	Sep 22th, 2021	Major update
0.1	Giovanni Conte	Sep 10th, 2021	Initial version

2 ABBREVIATIONS LIST

Abbreviation	Meaning
EDA	Event Driven Architecture
OCI	Oracle Cloud Infrastructure
OIC	Oracle Integration Cloud
API	Applicaton Programming Interface
ATP	Oracle Autonomous Transaction Processing
PaaS	Platform as a service
IaaS	Infrastructure as a service
SLA	Service Level Agreement
FAQs	Frequently Asked Questions

TABLE OF CONTENTS

3 NAME

Event Management for Event Driven Architecture

4 PROBLEM

This pattern addresses 2 main problems (to split under 2 different patterns and / or subpatterns is under evaluation):

1. Complexity and volume of information to be shared across applications and systems is raising together with need to get updated information as soon as it is available (near real-time). Also methods to manage information flow seem too rigid and should achieve greater agility (from Orchestration to Choreography)
2. Systems managing information / transactions are not open enough to publish information updates (events).

Examples of topics could be:

- Generally speaking: need to propagate information from source systems to target ones as soon as they are updated (near real time approach)
- A Portal / Web application which needs asynchronous updates on source information like order activation, shipment status, etc.

In relation to the above use cases some common problem categories are:

- **Getting events from packaged or custom applications**
- **Managing, propagating and publishing events through "Event Hubs"**
- **Processing events with Event Processing engines (Stream Analytics)**

5 CONTEXT

Sharing data is more effective than accumulating: decentralizing, distributing and copying is more powerful than stockpiling. Also connectivity and flow of information is the starting point for innovation and socializing. "Time is now": organizations need to process information as soon as these are created / modified.

This is why Real Time Integration patterns have become more requested by customers and the need is usually to have information available through 2 main ways:

- **APIs** to run Read / Query or Write / Command actions: client asks for information or perform upserts. APIs must be complete and must enable access and manipulation to any Business Object managed by the integrated system
- **Events** which publish updates so that the updated information can be propagated to any target. It usually means replication of source data into a target repository so that information can be processed by a analytics layer or propagation / streaming so that information can flow through systems which get information as per their specific interest

APIs are usually key for **Orchestration**: business or system processes call services to get information, aggregate and use.

Events are usually key for **Choreography**: business or system processes get information they are interested just subscribing to it. There's not a central orchestration. There is a "propagation layer" which makes events available, enriches them if needed: any interested target "subscribes" to it.

This pattern is defining the building blocks for **Event Driven Architecture**, the paradigm related to management events across IT systems.

APIs and events can be grouped in 2 main *categories*:

- **Native**: if exposed by applications (packaged or custom)
- **Platform**: if exposed through a Real Time Mesh framework layer (see next)

Events must be addressed on 2 main *categories*:

1. **Event generation**: make systems able to raise events

2. **Event streaming**: make events travel and reach (be available for) new IT locations to be accessed and consumed by any system

Event generation can happen in different ways. Following are the reference *categories*:

1. **Natively**: systems emit events through different technologies (queues, feeds) This approach leads to higher granularity (so called coarse grained) of information as systems / applications publish complete set of information at Business Object level
2. **Technical**: specific technology components work detecting changes on systems and raise an event for every change This approach leads to lower granularity information (so called fine grained) as it is usually taken from data repositories and Business Objects and must be "rebuilt" composing information (from Raw to Canonical Data). Also unneeded information must be properly filtered and transaction integrity must be preserved.

Event streaming can be made in different ways. Following are the reference *categories*:

1. **Direct**: there is a specific direct path between source and target. Usually this is done pushing raw data directly into target repositories. This is standard approach for Data Replication and is very common.
2. **Event propagation & management**: collaboration of different components enable propagation of events. Real time events can be combined through pipelines of stream data processing, time series analysis, etc., and propagated through a distributed topology to make information available to interested destinations.

6 SOLUTION

The way to address the described problem with the analyzed context relies on the combination of multiple components. In defining the solution and the needed components it is important to have an "open & hybrid" mindset which covers Oracle technologies and third party (mostly open source) frameworks. Being a complex solution it is better to address it driven by the *categories* defined in the "Context" section of this document. The solution will focus more on Events, using APIs just for any enrichment needs.

6.1 Event generation

As previously said can happen **natively** or can be obtained through an additional **technical** approach.

6.1.1 Native event generation

Native event generation is usually based on the ability of Business Applications to publish events. The way to get these events from Oracle SaaS application relies on capabilities of Oracle Integration Application Adapters. Availability of adapters for the specific application can enable the event generation.

Also, currently Oracle SaaS applications are making available an internal capability to publish / get changes of their database through a embedded installation of Golden Gate which acts as the "extract server" or availability of a stand by database to be accessed by an external Golden Gate. The status of this initiative is tracked in GoldenGate for SaaS Applications and Packaged Apps confluent page. This initiative originates for the high demand from customer to access the applications' database, which is the source of any real time master data access: any event can be get from there through Change Data Capture approach.

6.1.2 Technical event generation

Native event generation usually refers to applications and systems which are not capable to publish events: in this case technical ways to get information updates are required. The most common way is to get access to the database (either primary database or a stand-by copy) and get events through Change Data Capture approach. **Oracle OCI Golden Gate** is the key component to enable this capability.

6.2 Event streaming

As previously said can be **direct** or can be obtained through collaboration of different components to build a **event streaming** approach.

6.2.1 Direct propagation

Usually happens when real time data replication happens and there is a clear, distinct link between a origin datasource and a target one. It is a typical approach for data replication and is made using Change Data Capture capabilities. Oracle Golden Gate is the key component to enable this capability.

6.2.2 Event propagation

This case is the core for this Event Based Architecture pattern: it is based on many components. These is usually need for a mesh of events to be analyzed, managed, propagated and consumed.

Event propagation is the framework responsible to guarantee flow, persistence and access to events. It is based on so called **Event Hub**: most common Event Hub is **Kafka**. Oracle based Event Hub is **OCI Streaming Service**

6.2.3 Event Management

It is the core of the solution, where real time logic resides.

It is based on so called **Event processing** which the framework responsible to analyze, filter, enrich, propagate and republish events. Most common Event processing is **Spark**. Oracle based Event Hub is **Oracle Golden Gate Stream Analytics** which uses Spark internally.

6.3 Overall solution picture

Figure 1 shows a logical representation of the pattern **Event Management**, with all the Oracle Cloud Infrastructure components which are part of the Integration pattern.

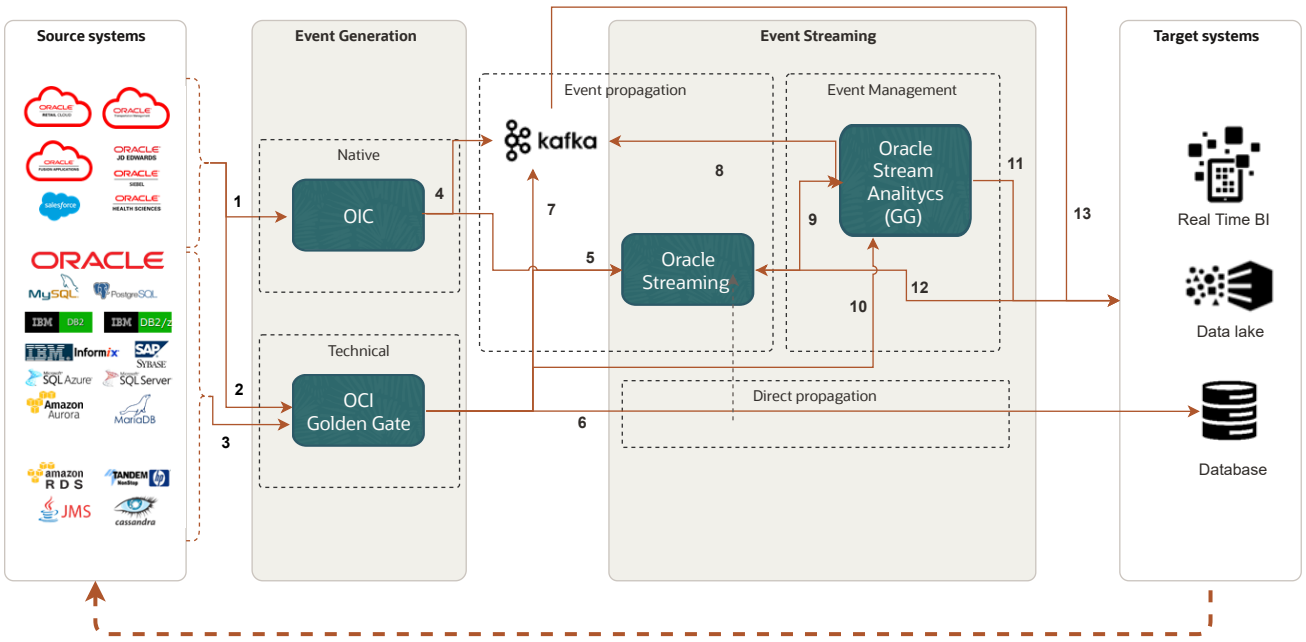

Figure 1: Logical Architecture

The description of the different paths is the following:

1. OIC can get events from SaaS through its Application adapters: we called these *native events* as are natively exposed by applications.
2. OCI Golden Gate can get events from Oracle SaaS applications: we called these *native events* as are natively exposed by applications through an embedded technology layer made available in the applications' scope. These differ from events of item 1 as are usually fine grained.
3. OCI Golden Gate can get events from applications' / systems' datasources: we called these *technical events* as taken directly from technology / data layer used by applications.
4. OIC can put events into Kafka Event component of Propagation layer through its Kafka adapter. This is a very common component customers have already.
5. OIC can put events into Oracle Streaming component of Propagation layer through its Oracle Streaming adapter. This layer is conceptually similar to Kafka one, has compatible APIs and can be used in substitution (or complement) to Kafka. Similarly OCI Golden Gate can feed events into Oracle Streaming (**TO BE VERIFIED**).
6. OCI Golden Gate can propagate event data directly to a target database: this is very common data replication subpattern.
7. OCI Golden Gate can feed events in Kafka.
8. Oracle Stream Analytics can get real time feeds from Kafka: it also can send back "enriched" events to Kafka as a target so that these can be consumed by any interested client through event subscription mechanism.
9. Oracle Stream Analytics can get real time feeds from Oracle Streaming: it also can send back "enriched" events to Oracle Streaming as a target so that these can be consumed by any interested client through event subscription mechanism.
10. Oracle Stream Analytics can get real time feeds from OCI Golden Gate as a stream source.
11. Oracle Stream Analytics, after performing its Event processing pipelines, can update any target directly.
12. Oracle Streaming events can be consumed by any interested target client.
13. Kafka events can be consumed by any interested target client.

Choreography of events is enables through the Event Propagation layer: Kafka and/or Oracle Streaming are the channels through which any interested party can subscribe to events he is interested in. This happen internally through links 8 and 9 but most importantly externally through links 12 and 13.

6.4 Deployment Architecture

Figure 2 shows a physical representation for the OCI infrastructure cam be required for the pattern **Event Management**

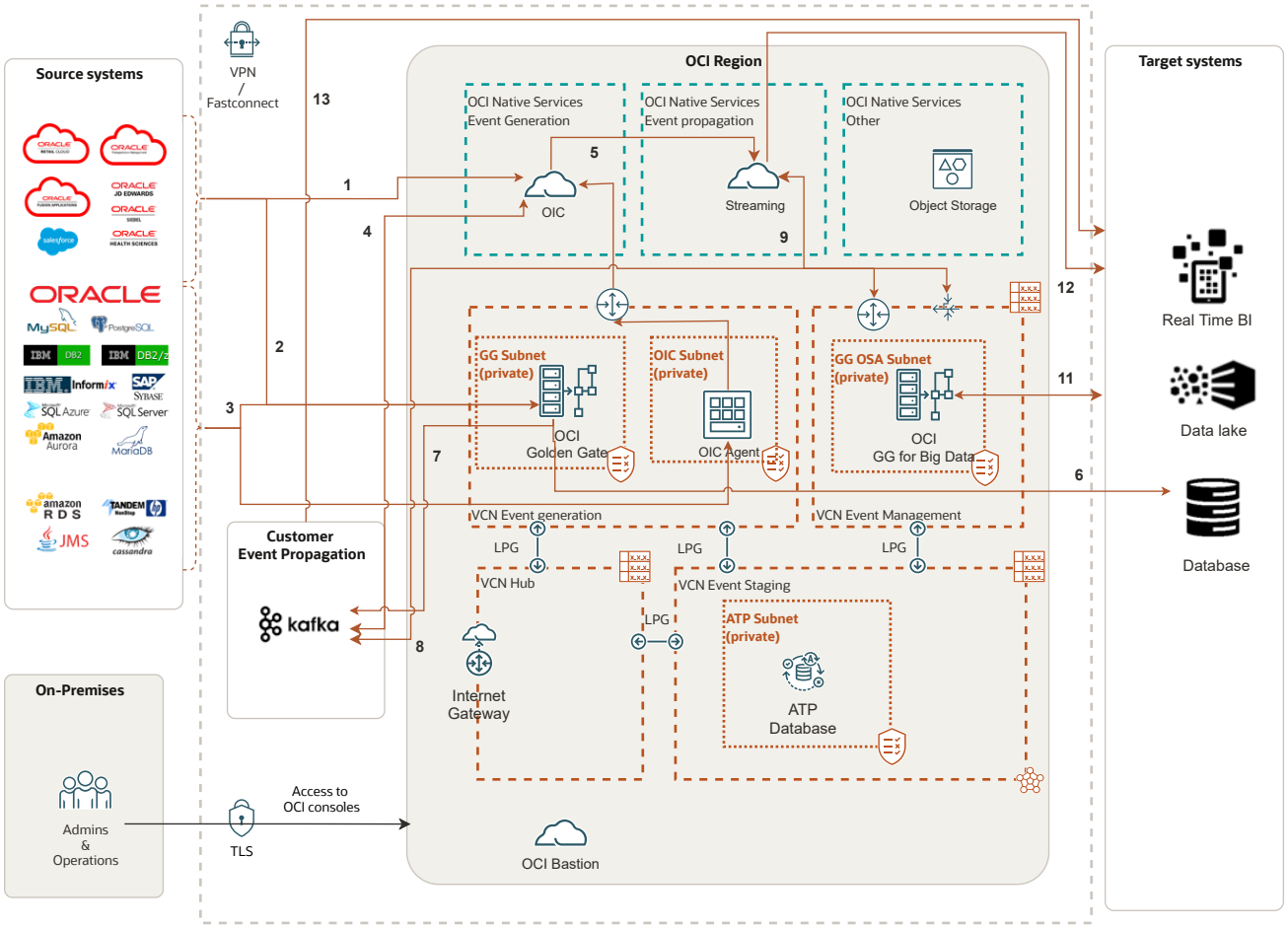

Figure 2: Physical Architecture for Event Management

The deployment architecture details how the OCI Cloud Services should be partitioned in the physical infrastructure and the main relationships between them.

Networking components represented:

- VCN: the OCI virtual, private network.
- Subnet: isolates Cloud Services applying proper security rules. Contain virtual network interface cards (VNICs), which attach to instances.
- Internet Gateway: manages access from Public Internet for user access or API exposure. It is virtual router you can add to your VCN to enable direct connectivity to the internet.
- Service Gateway: enables cloud resources without public IP addresses to privately access Oracle services.
- NAT Gateway: it gives cloud resources without public IP addresses access to the internet without exposing those resources to incoming internet connections.
- Local Peering Gateway: is the process of connecting two VCNs in the same region so that their resources can communicate using private IP addresses without routing the traffic over the internet or through your on-premises network.

Components of the OCI architecture are deployed in regions: for each region, availability domains are used to guarantee business continuity.

Cloud Services represented:

- Oracle Integration Cloud (OIC): fully managed service that allows to integrate your applications, automate processes, gain insight into business processes and create visual applications.

- Autonomous Database (ATP): self-driving, self-securing, self-repairing database service.
- OCI Golden Gate): is a fully managed, native cloud service that moves data in real-time, at scale.
- Oracle Stream Analytics: allows users to process and analyze large scale real-time information by using sophisticated correlation patterns, enrichment, and machine learning.
- Oracle Streaming: provides a fully managed, scalable, and durable solution for ingesting and consuming high-volume data streams in real-time to be processed continually and sequentially in a publish-subscribe messaging model.
- Object Storage: enables safe and secure store or retrieve data directly from the internet or from within the cloud platform.

The logical and physical re-partition of the components on the various networks (VCN and subnet) is as follow:

- **VCN Hub** - provides access to the a cluster's private network from the public internet. It serves as the public entry point for accessing a private network from external networks like the internet. Traffic must flow through the VCN Hub to access the private network, and you can set up security mechanisms on the VCN Hub to handle that traffic. In order to enable this traffic no VMs are needed inside VCN Hub VCN: OCI Bastion Service will be used to enable this access and it will use VCN Hub as the hub to reach the other VCNs.
- **VCN Event Generation** - will be used to position the Cloud Services (an their related networking components and configurations) needed to perform real-time data generation of events getting from Oracle SaaS Applications or source repositorie (either SQL or NoSQL). The component provisioned here is Oracle OCI Golden Gate. Oracle OCI Golden Gate resides on a private subnet.
- **VCN Event Generation** - will be used to position the Cloud Services (an their related networking components and configurations) needed to perform real-time data management (use of correlation patterns, enrichment, machine learning, filtering, etc). The component provisioned here is Oracle Marketplace Stream Analytics. Oracle Marketplace Stream Analytics resides on a private subnet.
- **VCN Event Staging** - will be used to position the ATP database (an its related networking components and configurations) which store all related Event Management temporary data needed when for event propagation and event management. Database resides on a private subnet.

There are then a logical set of networks in which the OCI Cloud native services are positioned:

- **OCI Native Services - Event Generation**: it is related to OIC component which gets native SaaS events.
- **OCI Native Services - Event propagation**: it is related to OCI STremaing component which propagates event as a publish/ subscribe event hub.

7 SAMPLE USE CASES AND TECHNICAL DETAILS

As mentioned above the pattern described so far is complex and could be divided in sub-patterns.

It is useful to mention some workshop labs that can explain how to build some of the steps described.

7.1 Oracle Stream Analytics Event Management - Path 8, 11, 8

Refer to Oracle GoldenGate Stream Analytics Workshop

7.2 Oracle Golden Gate direct data replication - Path 3, 6

Refer to Replicate Data Using OCI GoldenGate Workshop

TO BE CONTINUED - More use cases to be done

8 BILL OF MATERIALS

Following is the list of items to be used for the overall pattern architecture. For VM needed for agents a standard set commonly used is suggested.

Oracle Cloud Service role	Oracle Cloud Service	Billing metric	Part #
Integration	Oracle Integration Cloud Standard Ed.	msg packs (5000 msg/h)	B89639
OCI Golden Gate	"Oracle Cloud Infrastructure - GoldenGate"	OCPU Per Hour	B92992
Oracle Streaming	Oracle Cloud Infrastructure - Streaming - PUT or GET	Gigabytes of Data Transferred	B90938
Oracle Streaming Storage	Oracle Cloud Infrastructure - Streaming - Storage	Gigabyte Per Hour	B90939
Oracle GoldenGate Stream Analytics	Oracle GoldenGate Stream Analytics Marketplace image	BYOL	...
Database	Oracle Autonomous Transaction Processing	OCPU Per Hour	B90453
Database Storage	Oracle Autonomous Transaction Processing - Exadata Storage	Terabyte Storage Capacity Per Month	B90455
VM	OCI - Compute - Standard - AMD E3 - OCPU	OCPU Per Hour	B92306
VM	OCI - Compute - Standard - AMD E3 - Memory	GB / hr	B92307

9 NOTES

1. The pattern is suitable to be used (or to be enhanced with) IOT based patterns