

UTOUG · WEDNESDAY, SEPTEMBER 30, 2026 · 12:00 PM MDT

Oracle GoldenGate Microservices Architecture

From Fundamentals to OCI

Rohit Mishri

oradbf.com

ABOUT THE SPEAKER

Rohit Mishri

Senior database and cloud architect who designs, migrates, modernizes and operates enterprise database and data platforms.

Focus areas: Oracle Database, GoldenGate Microservices, GoldenGate for Big Data, Exadata, Autonomous Database, RAC and Data Guard, OCI / AWS / GCP, Kafka streaming, and automation with Terraform, Python and REST APIs.

19+

years in enterprise databases
and cloud

2018

presenter at Oracle
OpenWorld

**Oracle · Amazon ·
Nordstrom · Walmart**

migrations, HA architectures,
real-time replication

oradbf.com

articles on Oracle,
GoldenGate, cloud and
troubleshooting

AGENDA

The next sixty minutes

-
- | | |
|--|--|
| 01 What GoldenGate is and the problems it solves | 05 Self-managed GoldenGate vs OCI GoldenGate |
| 02 Microservices Architecture fundamentals | 06 Tooling: REST API, Admin Console, Admin Client |
| 03 Extract, trail files, Distribution Paths, Replicat | 07 Security, connectivity, monitoring, operations |
| 04 End-to-end data movement, source to target | 08 Benefits, challenges, and Q&A |
-

Real-time replication for the problems downtime can't solve

Oracle GoldenGate captures committed changes from a database's transaction log and delivers them to one or more targets, continuously and in transaction order.

Near-zero-downtime migration

Move to new hardware, versions or cloud while the source stays live

High availability & DR

Active-passive or active-active copies across sites and regions

Real-time integration

Keep operational systems in sync without batch ETL windows

Reporting offload

Feed a reporting or analytics copy so OLTP stays fast

Streaming & big data

Publish change events to Kafka, object storage and data lakes

Heterogeneous replication

Oracle, MySQL, PostgreSQL, SQL Server and more, in either direction

GoldenGate reads committed changes from the **transaction log**, not the tables, and replays them on the target.

Low source impact

Log-based capture; no triggers or table scans on the application's path

Transactional integrity

Only committed work moves, in commit order, with checkpoints for restart

Flexible delivery

Filter, map and transform in flight; one-to-many, many-to-one, bidirectional

From Classic to Microservices Architecture

Area	Classic	Microservices
Control	Manager process, GGSCI	Service Manager and per-deployment services
Interfaces	Command line only	Admin Console, Admin Client, REST API
Moving trails	Data Pump Extract + Collector	Distribution Service → Receiver Service
Security	Added on and configured by hand	TLS, users and roles, credential store built in
Automation	Scripts wrapped around GGSCI	Every operation available over REST
Direction	Deprecated in 21c, desupported in 23ai	The architecture going forward

One Service Manager, many services

CLIENTS

Admin Console

Browser UI

Admin Client

Command line

REST API

Scripts, Terraform, CI

HTTPS



Service Manager

Starts, stops and watches every deployment on the host

Deployment: one GoldenGate instance, e.g. "Source"

Administration Service

Creates and runs Extract and Replicat; hosts the Console

Distribution Service

Sends trail data to other deployments over paths

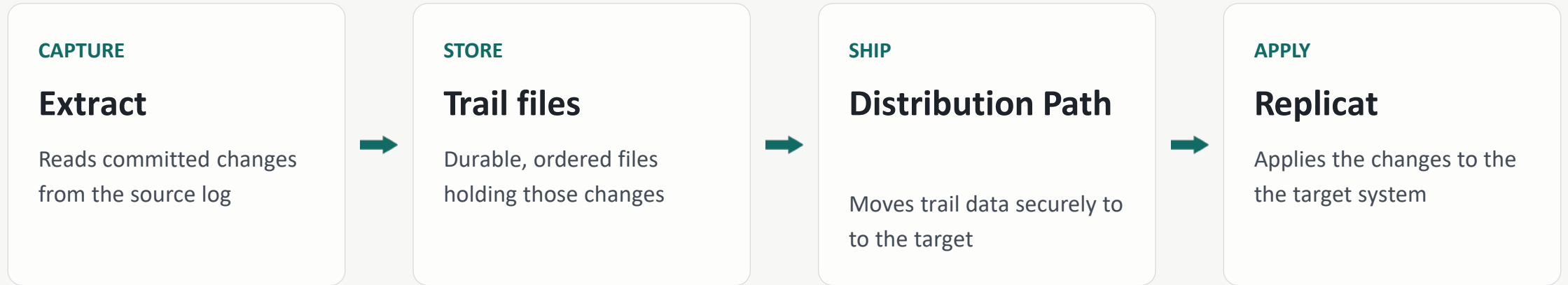
Receiver Service

Accepts incoming trail data and writes it locally

Performance Metrics

Collects process metrics and status for monitoring

Four building blocks carry every change



Each stage checkpoints its position, so any one can stop and restart without losing or duplicating a transaction.

Extract and trail files

Extract

- Integrated Extract reads redo through the database's logmining server
- Registered with the database; captures committed transactions only
- Parameter file names the tables, filters and mappings
- Checkpoints record its read and write positions for restart

Trail files

- A sequence of files in GoldenGate's platform-independent format
- Named by a two-character prefix plus sequence number, e.g. ea000000001
- Local trail on the source, remote trail on the target
- Buffers changes while the network or target is unavailable; purge once consumed

Distribution Paths and Replicat

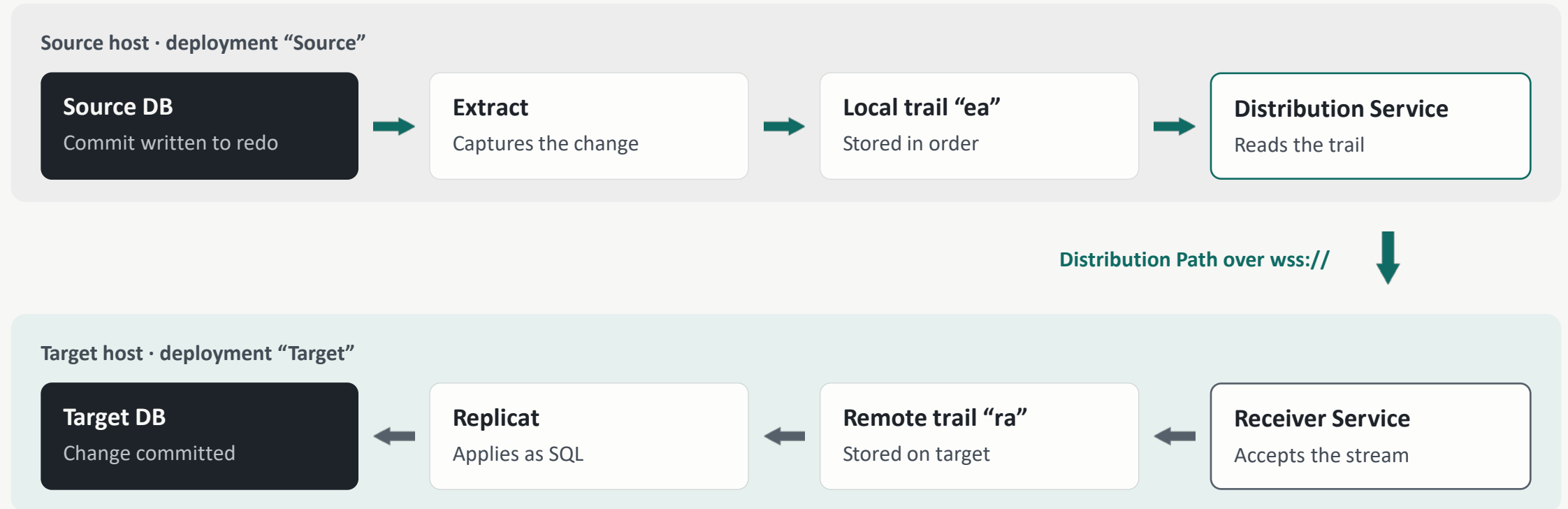
Distribution Path

- Defined in the Distribution Service: source trail → target Receiver Service
- Protocols include secure WebSocket (wss), ws, UDT, and ogg for Classic targets
- Target-initiated paths let the target pull through a DMZ or firewall
- Can filter which tables it sends

Replicat

- Reads the remote trail and applies changes to the target
- Parallel Replicat: applies independent transactions concurrently, respecting dependencies
- Integrated, Coordinated and Classic modes for specific needs
- A checkpoint table on the target prevents double-applying after restart

One change, source to target



Eight steps to a running pipeline

1 Prepare the source

ARCHIVELOG, supplemental logging, ENABLE_GOLDENGATE_REPLICATION

2 Store credentials

Credential store aliases, not passwords in files

3 Enable table logging

ADD SCHEMATRANDATA or ADD TRANDATA so keys are logged

4 Create and start Extract

Integrated Extract, registered, writing a local trail trail

5 Create the Distribution Path

Local trail → target Receiver Service over wss://

6 Instantiate the target

Data Pump export as of a known SCN, then import

7 Create and start Replicat

Parallel Replicat, starting after that SCN

8 Validate

Watch lag, compare row counts or use Veridata
Veridata

The same pipeline as commands

```
-- Source deployment
CONNECT https://src-host DEPLOYMENT Source AS oggadmin
ALTER CREDENTIALSTORE ADD USER ggadmin@srcpdb ALIAS srcdb
DBLOGIN USERIDALIAS srcdb
ADD SCHEMATRANDATA hr
ADD EXTRACT exthr, INTEGRATED TRANLOG, BEGIN NOW
REGISTER EXTRACT exthr DATABASE
ADD EXTTRAIL ea, EXTRACT exthr
START EXTRACT exthr
ADD DISTPATH dphr SOURCE trail://src-host:9002/services/v2/sources?trail=ea
TARGET wss://tgt-host:9103/services/v2/targets?trail=ra
START DISTPATH dphr
-- Target deployment
ADD REPLICAT repthr, PARALLEL, EXTTRAIL ra, CHECKPOINTTABLE ggadmin.ogg_ckpt
START REPLICAT repthr AFTERCSN <instantiation SCN>
```

Self-managed GoldenGate vs OCI GoldenGate

Topic	Self-managed (on-prem or IaaS)	OCI GoldenGate
Install & patch	You install, patch and upgrade	Oracle-managed; you pick upgrade timing
Capacity	Your hosts, storage and HA design	Deployments sized in OCPUs, optional auto-scaling
Access	Your network and certificates	Private endpoints, OCI IAM, OCI Vault
Monitoring	Your own tooling	OCI Monitoring metrics, Logging, alarms
Control	Full OS and file-system access	No OS access; supported connection types
Cost model	GoldenGate licenses plus infrastructure	Per OCPU-hour, or bring your own license

The building blocks of the managed service

Deployment

The managed GoldenGate instance, created per technology type

Connections

Reusable definitions for databases, Kafka, object storage and more

Assignments

Attach a connection to the deployments that may use it

Networking

Private endpoints in your VCN subnet, controlled by NSGs

IAM & Vault

Policies for who manages what; secrets and keys in OCI Vault

Observability

Metrics, logs, alarms and events alongside the rest of OCI

Inside the deployment it is the same Admin Console, Admin Client and REST API you use on-premises.

Three ways in, one set of services



Admin Console

Web UI for each service: build Extracts, Replicats and paths, read reports, watch statistics.

Best for learning and triage



Admin Client

Command line that replaces GGSCI; connects over HTTPS and runs OBEY scripts.

Best for repeatable runbooks



REST API

JSON over HTTPS for every operation; the Console and Client are built on it.

Best for automation and CI/CD

Automating with the REST API

```
# List Extracts in a deployment
curl -u oggadmin -X GET \
https://ogg-host/services/v2/extracts

# Details and status of one Extract
curl -u oggadmin -X GET \
https://ogg-host/services/v2/extracts/EXTHR

# Start it
curl -u oggadmin -X PATCH \
-H "Content-Type: application/json" \
-d '{"status":"running"}' \
https://ogg-host/services/v2/extracts/EXTHR
```

Same operations

Everything the Console does

Least privilege

A dedicated automation user with only the role it needs

Fits pipelines

Terraform, Ansible, monitoring and CI jobs

Secure by default, planned by design

Security

- TLS for services and wss for Distribution Paths
- GoldenGate users with Security, Administrator, Operator or User roles
- Credential store aliases keep passwords out of files
- On OCI: IAM policies, Vault secrets, private endpoints

Connectivity

- Plan ports for Service Manager and each service, or front them with a reverse proxy
- Open only the path direction you need; use target-initiated paths across a DMZ
- Size bandwidth and latency for peak redo, not average
- Test DNS, certificates and firewall rules before go-live

What to watch on day two

Lag

Heartbeat table for true end-to-end lag; alert on trend, not a single spike

Process health

Status and statistics from the Performance Metrics Service

Logs & reports

ggserr.log, process report files and discard files

Housekeeping

Trail purge tasks and auto-restart profiles for processes

Alerting

OCI alarms, or your own tooling polling the REST API

Change control

Patching, DDL handling and versioned parameter files

Key benefits and common challenges

Benefits

- Near-zero-downtime migrations and upgrades
- Low-impact, log-based capture
- Heterogeneous and bidirectional replication
- API-driven operations that fit automation
- A managed option on OCI

Challenges

- Source prep: supplemental logging, keys on every table
- Consistent instantiation at a known SCN
- Unsupported data types and DDL edge cases
- Lag under large batch transactions
- Conflict handling in active-active, licensing, skills

THREE THINGS TO TAKE HOME

- 1** Extract → trail → Distribution Path → Replicat is the whole data path.
- 2** Microservices adds secure, API-driven control; the same model runs on-premises and in OCI.
- 3** Success is decided by preparation, instantiation and day-two operations.

Q&A

Questions?

Thank you, UTOUG.

SPEAKER

Rohit Mishri
Mishri

LINKEDIN

linkedin.com/in/rohit-mishri

ARTICLES

oradbf.com