Oracle® Communications Billing and Revenue Management

AI Services Framework Overview

Release 15.2
G49672-01
January 2026

Oracle Communications Billing and Revenue Management AI Services Framework Overview, Release 15.2

G49672-01

Contents

About This Content

Describes how to use AI-powered product, bundle, and offer recommendations in Oracle Communications Billing and Revenue Management (BRM).

Audience

This guide is intended for developers.

1

Overview of AI in Oracle Monetization Suite

Oracle Monetization Suite provides artificial intelligence (AI) and machine learning (ML) services that help you optimize revenue management and billing operations. This document introduces the AI features available in Oracle Monetization Suite.

Topics in this document:

About AI in Oracle Monetization Suite

Oracle Monetization Suite offers AI-based microservices that enable you to build end-to-end machine learning (ML) workflows. You use these workflows to create customized models and implement monetization and charging features tailored to your business needs.

The microservices are highly extensible and configurable. They deliver flexibility and scalability, and are certified for use with both Oracle Cloud Infrastructure (OCI) and select third-party technologies.

> **Note**
>
> Oracle recommends using OCI for a seamless experience, including OCI Data Flow, OCI Data Science, OCI Object Storage, and advanced model management capabilities provided by OCI Data Science.

You can integrate the AI microservices with Oracle Monetization Suite products such as Billing and Revenue Management (BRM), Elastic Charging Engine (ECE), Billing Care, and Oracle Communications Convergent Charging Controller.

With AI functionality in Oracle Monetization Suite, you can:

- **Create data sources to build efficient ML workflows**. Collect and fetch data in bulk or as needed, and use this data to train models.
- **Train models based on configured data sources**. You select which features, such as currency, country, available cash balance, and current deal, to include in model training.
- **Use inference services to make predictions**. Analyze current data and generate predictions for future scenarios. For example, based on a customer's usage pattern, you can predict the offer a customer is most likely to choose.

> **Note**
>
> The AI-based microservices framework and related capabilities are newly introduced and are currently offered as experimental functionality.

2

About System Architecture

Learn about the system architecture for machine learning (ML) workflows and models that you create using the artificial intelligence (AI) microservices provided in Oracle Monetization Suite.

Topics in this document:

System Architecture Overview

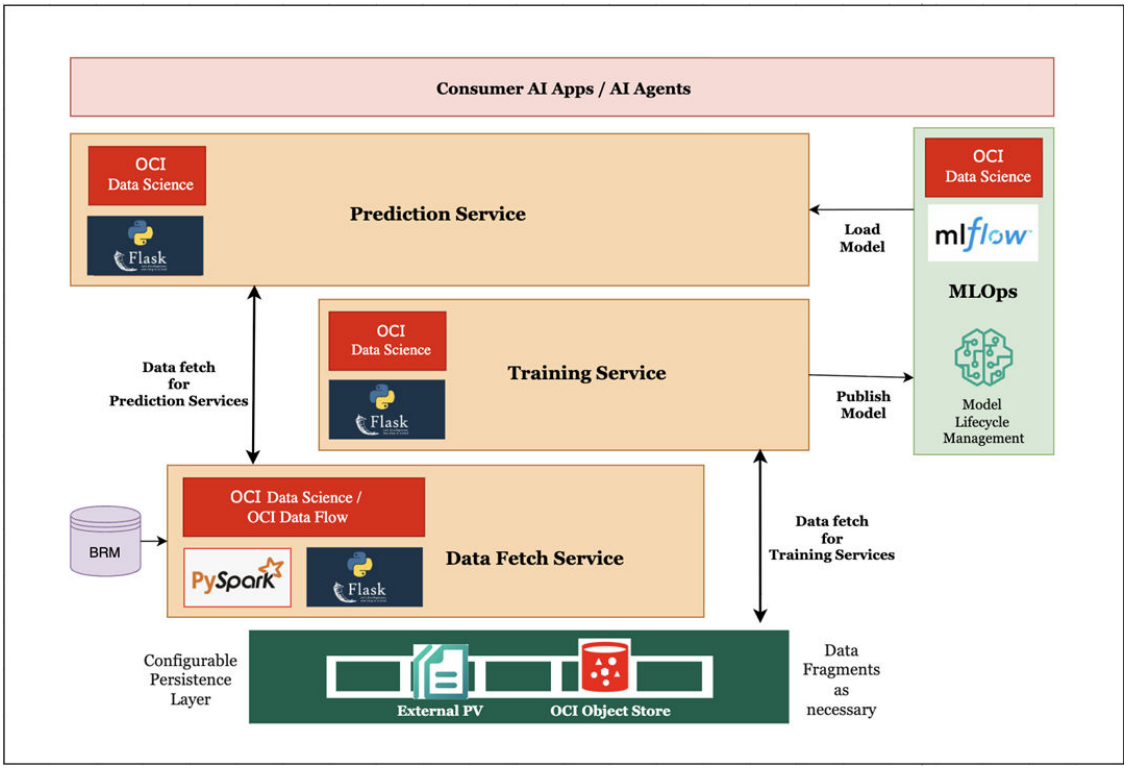
The system architecture uses a modular approach, with distinct stages for data storage, data fetching, model training, model deployment and management, prediction services, and integration with client applications or AI agents. Each stage consists of independently deployable microservices, which operate as pods in a Kubernetes environment.

The key components of the architecture include:

- **Client applications or AI agents**: End-user applications or automated agents that consume AI-driven services.
- **Prediction services**: Inference services that use deployed ML models.
- **ML Ops – model management**: A repository for managing, versioning, and monitoring trained models.
- **Training services**: A framework that supports model training using curated data.
- **Data layer**: The source layer for preparing, fragmenting, and supplying data through a configurable data service mesh.
- **Persistence layer**: The storage repository for both raw and processed data.

Figure 2-1 shows the high-level system architecture and its major components.

Figure 2-1 System Architecture Overview



In Figure 2-1, the data layer ingests raw data from source systems, transforms and enriches it, and caches it as domain-specific data fragments. This data is then stored in the persistence layer, which securely stores both raw and processed datasets for downstream use. When you develop a model, training services access high-volume, engineered data from the persistence layer for model training.

After training, the model is published to the ML Ops (model management) layer for versioning, tracking, and lifecycle governance. The prediction services leverage the published model, fetch relevant, cached input data from the data layer, and use them to respond to real-time prediction requests. These predictions are then consumed by the client applications or AI agents to use for different purposes like getting insights for decision making.

About Client Applications or AI Agents

Client applications or AI agents serve as the interface between end users and AI-ML models. These may include Oracle Monetization Suite applications, such as Billing Care, or AI-driven agents that invoke prediction services for real-time or batch inference.

> **Note**
>
> The client must handle the batch processing as there is no dedicated API for this.

About Prediction Services

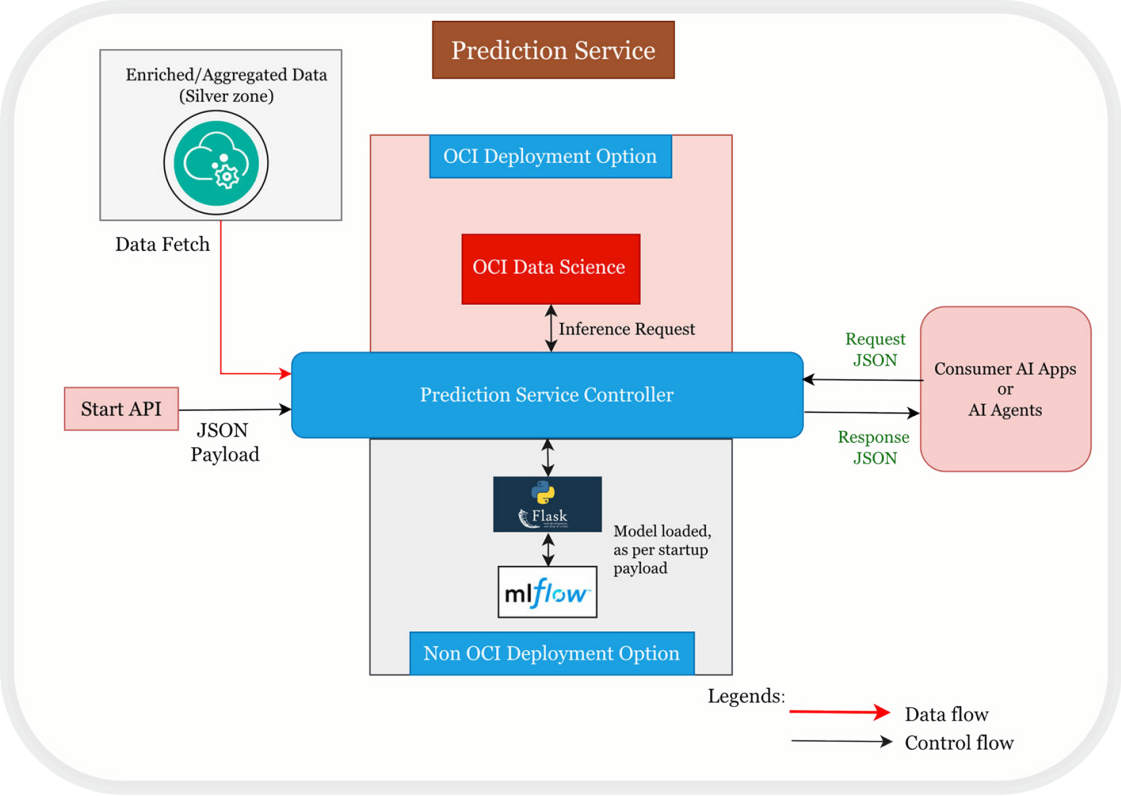
Prediction services are specialized microservices that perform inference operations within the architecture. The microservices retrieve the required input data from the data mesh in the data service layer. The data mesh is a collection of services responsible for owning and caching specific data fragments for immediate use. This design ensures low latency and high throughput, optimizing overall performance.

When a prediction request is received, the prediction services dynamically load the appropriate trained model from the model management layer and use it, with the relevant input data, to generate predictions. Client applications or AI agents then consume and act upon these predictions.

You can use the prediction services by connecting it to a model on OCI Data Science or implement them using Python Flask sample frameworks.

Figure 2-2 shows the high-level architecture of the Prediction Service layer.

Figure 2-2 Prediction Service Layer Architecture



About ML Ops – Model Management

The ML Ops (model management) layer functions as the central repository and governance platform for trained models. Once a model is trained by the training services, it is published here for versioning, tracking, and lifecycle management. The model management system stores metadata for each model, including version history and associated metrics.

This layer is certified with OCI Data Science offering ML Ops capabilities and with the open-source MLFlow technology.

About Training Services

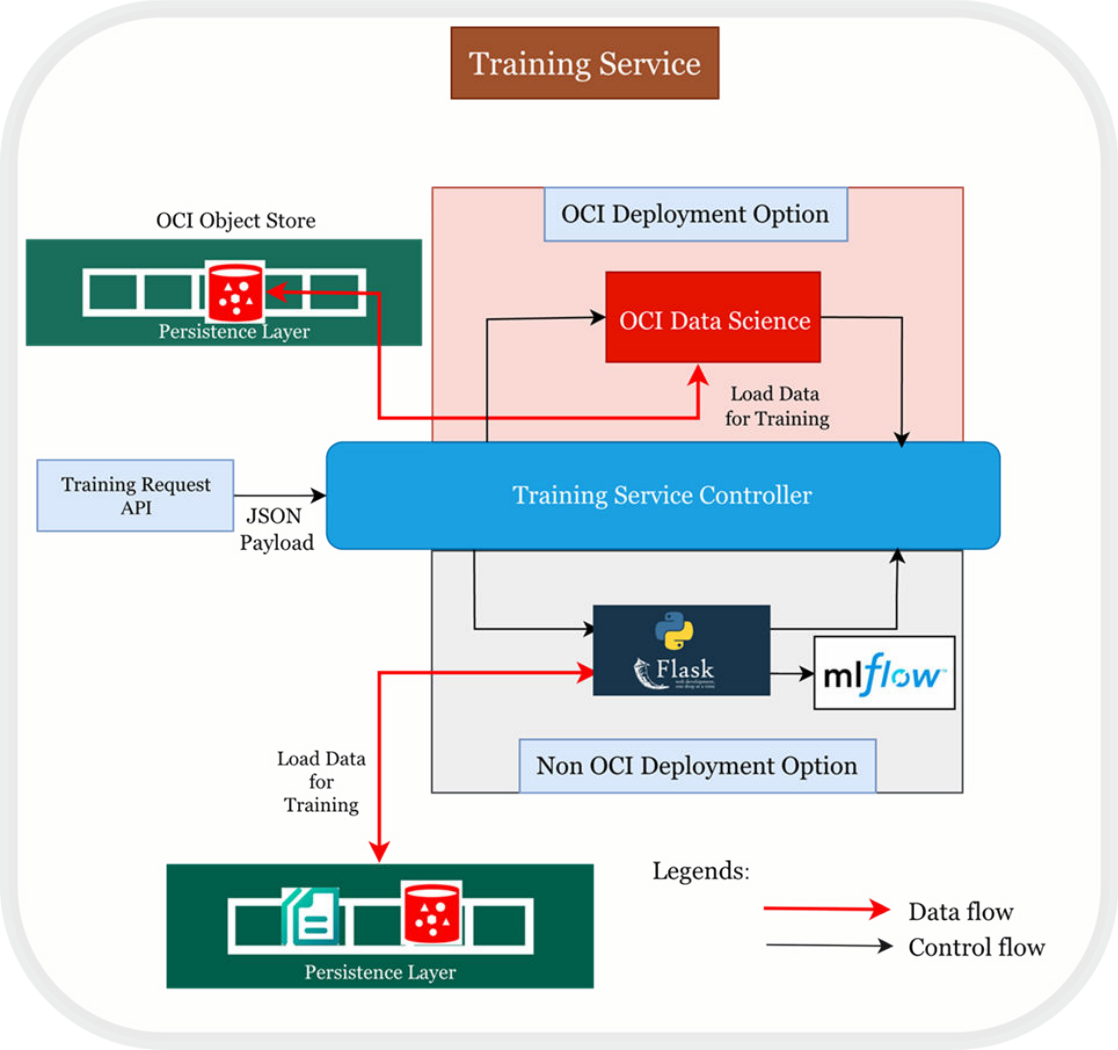
Training Services are primarily responsible for orchestrating machine learning model development. The microservices begin with fetching the input data directly from the persistence layer as the data is required in bulk for high-volume training workload. Using the persistence layer also reduces the time for data fetching in this case.

The microservices then run the training jobs to train the model based on the input data. When the training is complete, the model is pushed to the model management layer along with the relevant metadata or metrics.

You can deploy training services on an OCI environment using OCI Data Science or on a non-OCI environment like Kubernetes.

Figure 2-3 shows the high-level architecture of the Training Service layer.

Figure 2-3 Training Service Layer Architecture



About the Data Layer

The data layer is the primary source that prepares, fragments, and supplies the required data for downstream ML tasks. Similar to the Medallion architecture, this layer includes two data zones:

- **Bronze zone**: Contains raw, unprocessed data ingested from primary sources.
- **Silver zone**: Holds cleaned, enriched, and aggregated data optimized for modeling.

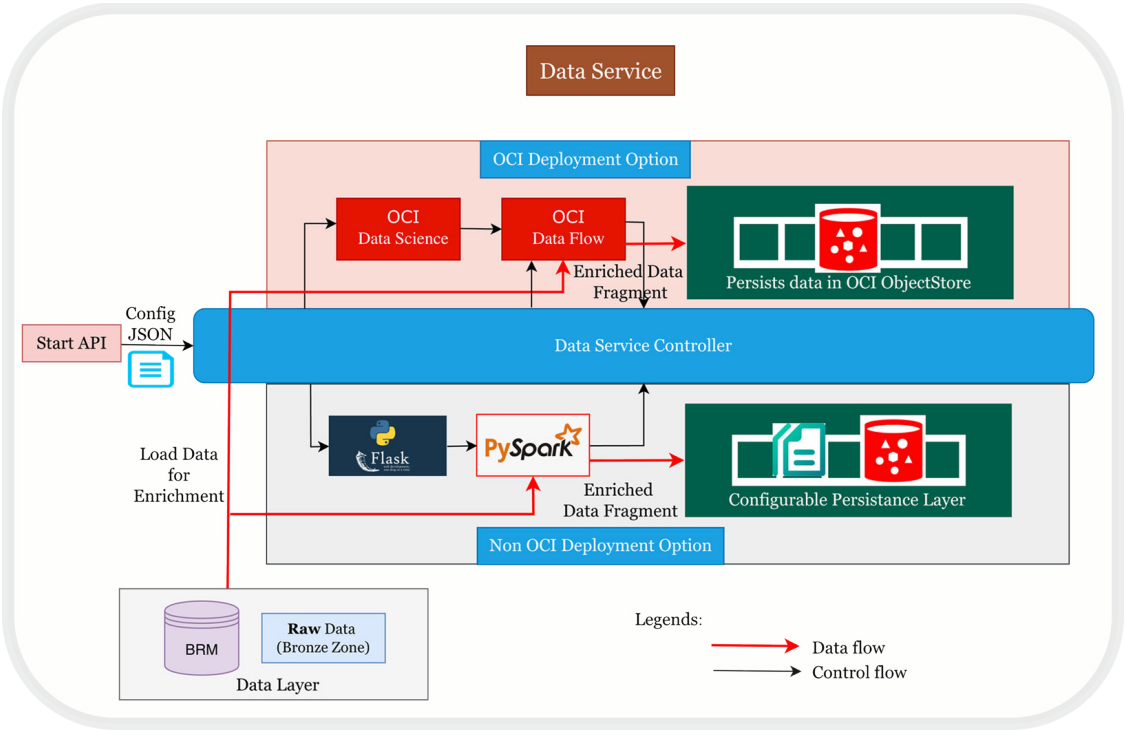
The silver zone implements a data mesh, in which each service owns and caches a specific data fragment (for example, a service for subscriber usage data in Billing and Revenue Management).

Data fragments in the silver zone are cached and validated for quick access. Each service includes metadata describing its data. You configure data services in the silver zone to regularly fetch updates from the bronze zone and update cached fragments and the persistence layer as needed.

Data services can be deployed on OCI or non-OCI environments. For OCI deployments, data engineering either uses OCI Data Flow directly or call OCI Data Flow using OCI Data Science, and data is stored in OCI Object Storage. For non-OCI environments, data engineering is handled by PySpark, and engineered data is persisted to either PVC or OCI Object Storage, based on the configuration.

Figure 2-4 shows the high-level architecture of the Data Service layer.

Figure 2-4 Data Service Layer Architecture



About the Persistence Layer

The persistence layer acts as the centralized, long-term storage repository for all data. After data services process raw data from the bronze zone and cache it for immediate use, they persist this data for durability and downstream analytics, training, and prediction.

Data services maintain and secure the persistent data, ensuring it is current and available. They also use the persistence layer to identify and fetch new or changed data from the raw data source, enabling efficient incremental updates.

The data can be persisted in one of the following:

- Persistent Volume Claims (PVC) for containerized environments
- OCI Object Store for scalable, object-based storage

3

AI Reference Implementation: Offer Recommendation

Learn about the features you can implement using machine learning (ML) workflows and models created with the artificial intelligence (AI) microservices available in Oracle Monetization Suite.

Topics in this document:

About the AI Reference Implementation

Oracle Monetization Suite provides extensible microservices that let you build ML workflows to implement AI-powered features and functionalities. You use inference services to generate valuable insights, make accurate predictions, and deliver personalized suggestions to your customers.

You can use these microservices to create models tailored to your business needs. Models integrate with Oracle Monetization Suite products such as Billing and Revenue Management, Elastic Charging Engine, Billing Care, and Oracle Communications Convergent Charging Controller. You can train models using your own data and adapt them to your specific requirements using the training services.

You can configure, add, or remove the model's features according to your business needs. This may include features like country, state, currency, and usage pattern of the customers.

> **Note**
>
> The AI-based microservices and associated features are still in an experimental phase. Hence, this release offers only one AI-based feature.

Next Best Offer Recommendation

Oracle Monetization Suite uses AI to recommend the best customized offers, allowing you to present top recommendations for products, bundles, or services relevant to each customer.

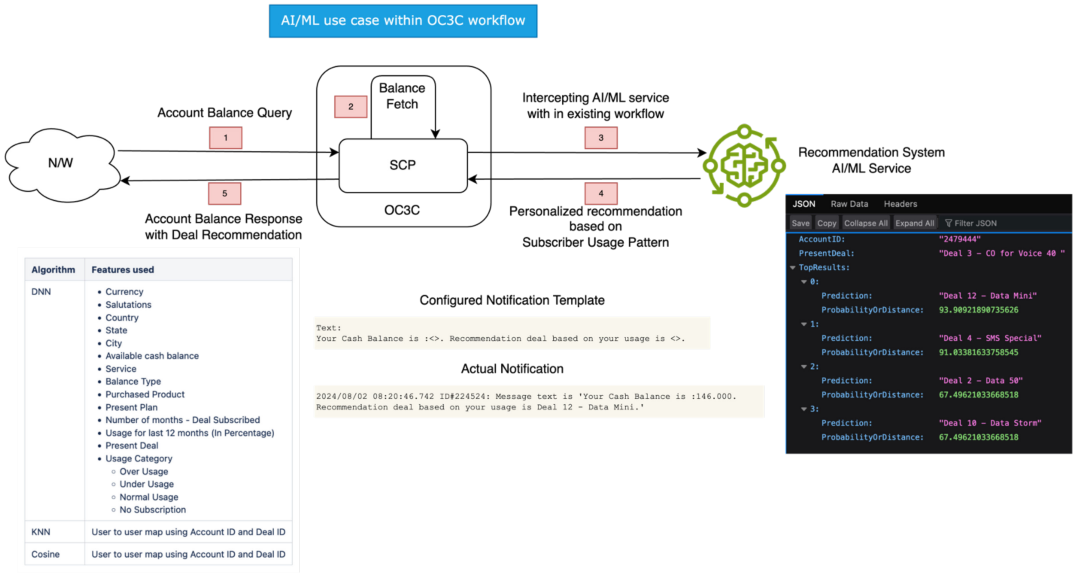
The recommendation engine analyzes a combination of customer profile attributes, engagement history, product catalog information, and past purchase behavior to suggest the most appropriate offers for each customer. You can use these recommendations to target customers with highly relevant offers and support personalized marketing and upselling strategies.

For example, the system can propose offers or products based on a customer's usage trends when a contract is about to expire or if the customer indicates interest in changing plans. You

can also use the feature to identify top-performing offers or products within specific customer segments.

Figure 3-1 shows a sample end-to-end workflow of the AI- and ML-based model within Convergent Charging Controller.

Figure 3-1 Sample Workflow for AI-ML Model Within Convergent Charging Controller



In Figure 3-1, the workflow starts when the system receives a network event or request, such as a customer action or usage update. The Service Control Point (SCP) collects and formats the required customer information and forwards it to the prediction engine.

The prediction engine uses trained models, such as a deep neural network (DNN), k-nearest neighbors (KNN), or cosine similarity, along with configured features like country, usage, and product, to generate personalized offer recommendations. The prediction service returns the top recommended offers, ranked with a probability or score.

The recommendation, along with account and usage information, is formatted into a customer-friendly message and sent through the SCP back to the network, ultimately delivering a personalized offer or alert to the customer.

> **Note**
>
> Figure 3-1 is only an example with Convergent Charging Controller as the product and not restricted to it. You can integrate AI with other products under Oracle Monetization Suite based on your own configurations.