Oracle® Communications Billing and Revenue Management

AI Services Framework Overview



Release 16.26.0
G54438-01
October 2026



Oracle Communications Billing and Revenue Management AI Services Framework Overview, Release 16.26.0

G54438-01

Copyright © 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

About This Content

1 Overview of AI in Cloud Scale Monetization

About AI in Cloud Scale Monetization 1

2 About System Architecture

System Architecture Overview 1
About the Data Layer 2
About the Persistence Layer 3
About Training Services 4
About ML Ops – Model Management 5
About Prediction Services 5
About Client Applications and AI Agents 6

3 AI Reference Implementations

About the AI Reference Implementation 1
Next Best Offer Recommendation 1
Integrating Next Best Offer Recommendation with BRM and Siebel 3
Time-Series Anomaly Detection 3
Billing Care AI Agent for Bill Explanation 3

Copyright © 2026, Oracle and/or its affiliates.

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.

Copyright © 2026, Oracle and/or its affiliates.

1

Overview of AI in Cloud Scale Monetization

Oracle Communications Cloud Scale Monetization 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 Cloud Scale Monetization.

Topics in this document:

- About AI in Cloud Scale Monetization

About AI in Cloud Scale Monetization

Cloud Scale Monetization 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 other private cloud environment.

> **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 Cloud Scale Monetization products such as Oracle Communications Billing and Revenue Management (BRM), Oracle Communications Elastic Charging Engine (ECE), Oracle Communications Billing Care, and Oracle Communications Convergent Charging Controller.

With AI functionality in Cloud Scale Monetization, 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**. Select which features, such as user data, subscription data, usage data, metrics data, 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.
- **Detect anomalies in time-series operational metrics**. Use trained models to analyze current data and identify unusual patterns or behavior.
- **Use AI agents to answer supported product questions in natural language**. For example, you can use the Billing Care AI Agent to retrieve and summarize supported billing information.

Copyright © 2026, Oracle and/or its affiliates.

You can also use AI services with external data sources and AI agents. For example, you can collect data from BRM and Siebel, train models that return multiple related prediction outputs, and use an AI agent to request predictions through a Model Context Protocol (MCP) interface.

This capability supports a range of use cases, including, but not limited to, next best offer recommendations, in which the AI services use customer, usage, product, and offer data to recommend suitable charge and discount offers.

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

Copyright © 2026, Oracle and/or its affiliates.

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 Communications Cloud Scale Monetization.

Topics in this document:

- System Architecture Overview
- About the Data Layer
- About the Persistence Layer
- About Training Services
- About ML Ops – Model Management
- About Prediction Services
- About Client Applications and AI Agents

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:

- **Data layer**: The source layer for preparing, fragmenting, and supplying data through a configurable data service mesh.

  For more information, see "About the Data Layer".

- **Persistence layer**: The storage repository for both raw and processed data.

  For more information, see "About the Persistence Layer".

- **Training services**: A framework that supports model training using curated data.

  For more information, see "About Training Services".

- **ML Ops – model management**: A repository for managing, versioning, and monitoring trained models.

  For more information, see "About ML Ops – Model Management".

- **Prediction services**: Inference services that use deployed ML models.

  For more information, see "About Prediction Services".

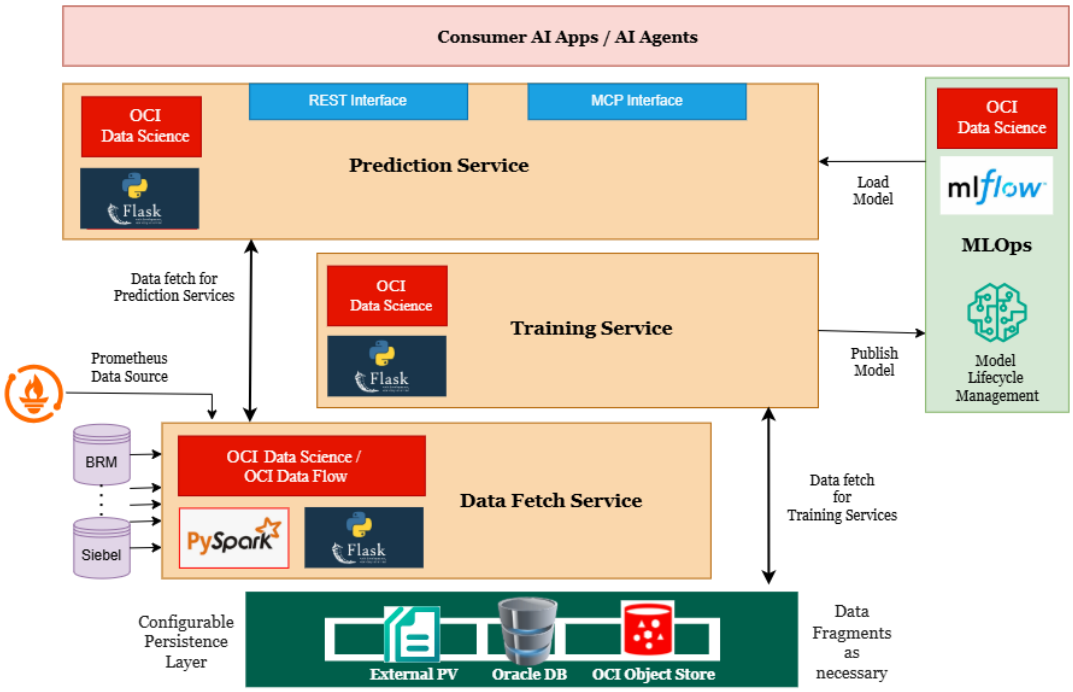
- **Client applications or AI agents**: End-user applications or automated agents that consume AI-driven services.

  For more information, see "About Client Applications and AI Agents".

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

Copyright © 2026, Oracle and/or its affiliates.

Figure 2-1 System Architecture Overview



In Figure 2-1, the data fetch service 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 use the published model to retrieve relevant cached input data from the data layer. They then use this data to respond to real-time prediction requests. These predictions are then consumed by the client applications or AI agents to get insights for decision making.

About the Data Layer

The data layer is the primary source that prepares, fragments, and supplies the required data for downstream ML tasks. This layer includes two data zones:

- **Bronze zone**: Contains raw, unprocessed data ingested from primary sources, such as Oracle DB or Prometheus.
- **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.

Copyright © 2026, Oracle and/or its affiliates.

The data service can collect data from multiple sources, such as Siebel, BRM, and Prometheus. For example, it can collect usage, account, product, and related data from BRM and collect customer, purchase, campaign, offer, and label data from Siebel.

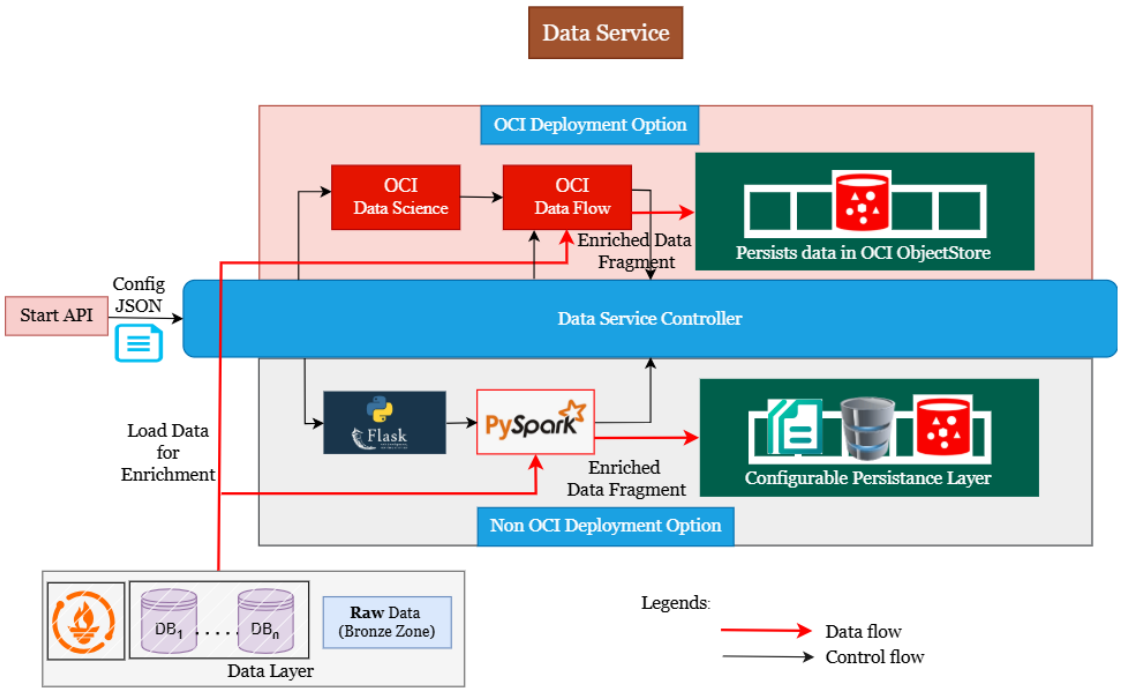
The data service processes each source and prepares refined data fragments for downstream services. Depending on the use case, BRM and Siebel data can remain as separate fragments or can be combined during preprocessing.

When configuring multiple sources in one request, you need to specify the source type, query configuration, and connection details in the same order. This ensures that the data service uses the correct connection details for each source.

Data services can be deployed on OCI or non-OCI environments. For OCI deployments, data engineering either uses OCI Data Flow directly or calls 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, OCI Object Storage, or Oracle DB, based on the configuration.

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

Figure 2-2 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.

Copyright © 2026, Oracle and/or its affiliates.

The data can be persisted in one of the following:

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

About Training Services

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

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.

Training service supports unsupervised algorithms like KNN and Cosine Similarity, supervised algorithms like DNN (for single classification) and multi-head DNN (for multiple classification), and XGBoost (for time series based prediction).

For example, training services support multi-head DNN models for use cases that require multiple predictions from the same customer data. A multi-head model uses one shared set of input features and creates a separate output head for each target label.

In this case, the training data must contain two or more label columns that use the LABEL_ prefix, such as LABEL_1 and LABEL_2. Each label column is encoded separately and is associated with a separate model output.

> **Note**
>
> If you are using models like DNN and XGBoost, there should be just one column named LABEL.

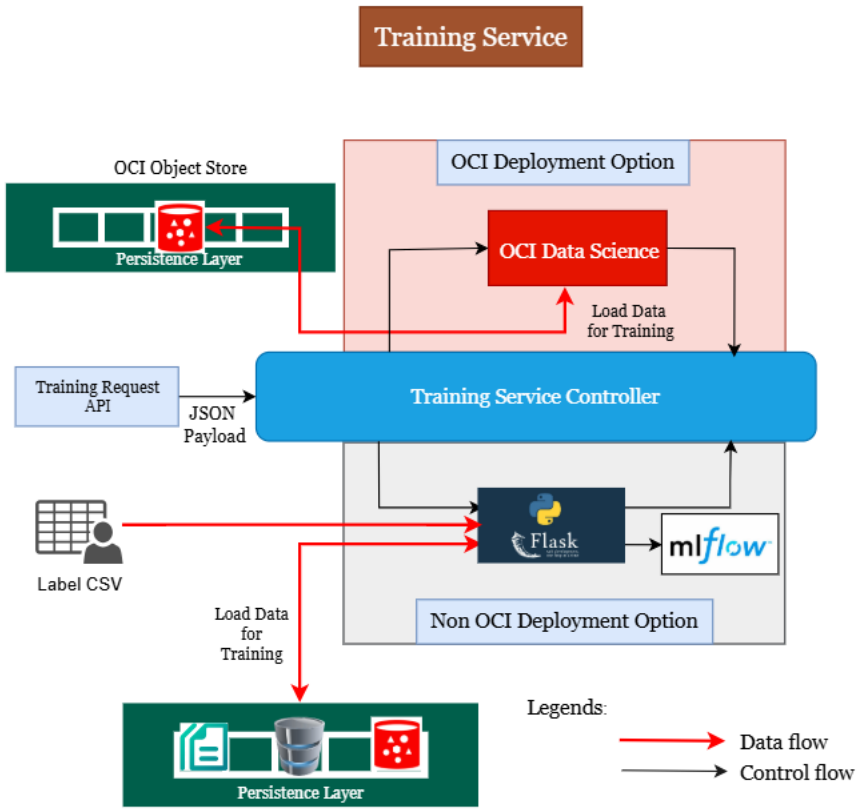
The training service preprocesses the data, trains the model, evaluates each output head, and stores the model with its metadata. The metadata can include feature definitions, label mappings, and model metrics. The prediction service uses this information to prepare input data and interpret prediction results.

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.

Copyright © 2026, Oracle and/or its affiliates.

Figure 2-3 Training 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.

The training service stores the model in ONNX format. Associated model files can include label-encoder metadata, feature-column metadata, training metrics, and evaluation metrics for each output model.

Before using a model, the user must manually specify the model version that the prediction service must load. The prediction service loads the specified model version and uses the associated metadata to prepare input data and interpret output values.

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

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.

Copyright © 2026, Oracle and/or its affiliates.

When the prediction service starts, it loads the specified trained model version from the model management layer. For each request, the service uses the relevant input data to generate predictions. Client applications or AI agents then consume and act upon these predictions.

For offer recommendation use cases, the prediction service retrieves cached customer data from data service and uses an approved multi-head model to generate recommendations. A prediction can return more than one result type. For example, it can return a recommended charge or product offer and a recommended discount offer for the same customer.

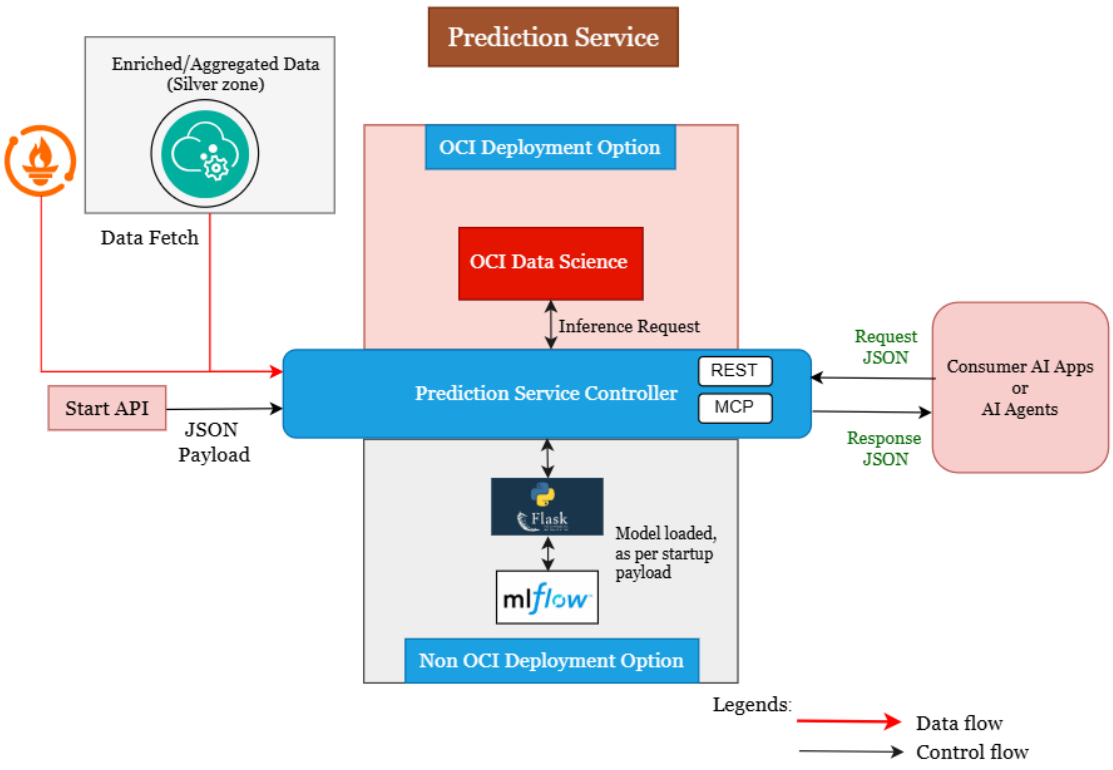
The prediction service returns structured results to the calling integration layer. The result include expected output and configured details, such as confidence score, prediction, and the used algorithm.

For time-series anomaly detection, the prediction service retrieves current metrics directly from Prometheus. The service prepares the metrics using the deployed model metadata and returns the classification for the latest timestamp in the requested time window.

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

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

Figure 2-4 Prediction Service Layer Architecture



About Client Applications and AI Agents

You can use client applications or AI agents as the interface between end users and AI-ML models. These may include Cloud Scale Monetization applications or AI-driven agents that invoke prediction services for real-time or batch inference.

Copyright © 2026, Oracle and/or its affiliates.

For client applications, you can invoke prediction services through REST APIs. For AI agents, you can request recommendations through a MCP server.

The MCP server provides a controlled interface between the client and the prediction service. The client sends the required request context to the MCP server. The MCP server validates the request, calls the prediction service, and returns the result as a structured MCP tool response.

The MCP server enables clients to use prediction capabilities without requiring direct access to data sources, model artifacts, or model-management systems. Access to these resources is controlled by the configured monetization services and deployment security policies.

> **Note**
>
> If you use batch inference, the client must handle the batch processing because there is no dedicated API for this.

Copyright © 2026, Oracle and/or its affiliates.

3

AI Reference Implementations

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

Topics in this document:

- About the AI Reference Implementation
- Next Best Offer Recommendation
- Integrating Next Best Offer Recommendation with BRM and Siebel
- Time-Series Anomaly Detection
- Billing Care AI Agent for Bill Explanation

About the AI Reference Implementation

Cloud Scale Monetization 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. The models integrate with Oracle Monetization Suite products and you can train them 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. For example, an offer recommendation model can use customer features such as country, state, currency, and usage pattern. A time-series anomaly detection model can use operational metrics collected over time.

> **Note**
>
> This release offers only one AI-based feature because the AI-based microservices and associated features are still in an experimental phase.

Next Best Offer Recommendation

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

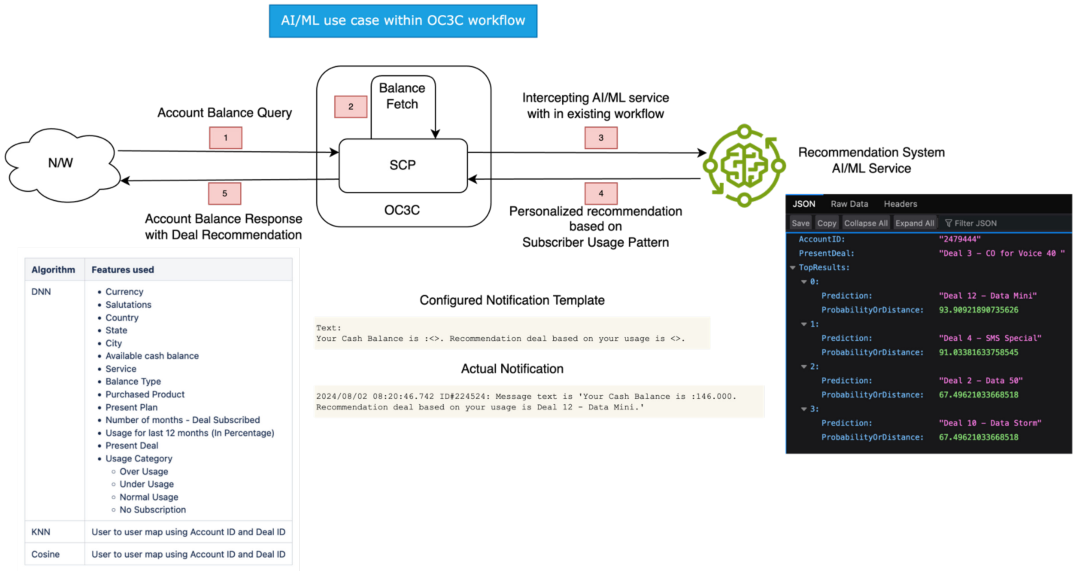
The recommendation engine analyzes a combination of customer profile attributes, engagement history, product catalog information, and past purchase behavior to suggest the most appropriate offers for each customer. You can use these recommendations to target customers with highly relevant offers and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

Copyright © 2026, Oracle and/or its affiliates.

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 using Convergent Charging Controller as the product. You can configure AI to integrate with other Oracle Monetization products in the Cloud Scale Monetization as needed.

Copyright © 2026, Oracle and/or its affiliates.

Integrating Next Best Offer Recommendation with BRM and Siebel

You can integrate the next best offer recommendation feature with BRM and Siebel by using the AI Services Framework.

In this integration, the data service collects the required customer and usage data from BRM and Siebel. The training service uses the prepared data to train a multi-head DNN offer recommendation model. The model can generate multiple related recommendation outputs, such as a charge or product offer and a discount offer.

At run time, a Siebel client or other client application sends a recommendation request through the REST API, while an AI agent uses the configured Model Context Protocol (MCP) interface The MCP server validates the request and calls the Prediction Service. The prediction service retrieves the required cached customer data, loads the approved model, and generates the recommendation results.

The MCP server returns the results to Siebel as a structured response. Siebel can use the results to display recommendations to users or to support configured next best offer workflows.

Time-Series Anomaly Detection

You can use the time-series anomaly detection feature to identify unusual behavior in operational metrics by using the AI Services Framework.

The workflow begins with the data service collecting historical metrics from Prometheus. The training service uses the prepared metric data to train a time-series anomaly detection model and identify any change in regular behavior. For example, the supplied sample uses an XGBoost model that specifies 0 to represent normal metrics and 1 to represent potential outage detection.

At run time, a client application sends an anomaly detection request through the REST API, while an AI agent uses the configured MCP interface. The MCP server validates the request and calls the prediction service. The prediction service retrieves the required current metrics from Prometheus, loads the deployed model, and generates a classification for the latest timestamp in the requested time window.

The MCP server returns the classification result as a structured response. A client application can use the result to identify anomalous operational behavior and support configured monitoring workflows.

You can train the model with additional labels when your use case requires different anomaly types. The meaning of each label is defined by the training data used for the model.

Billing Care AI Agent for Bill Explanation

The Billing Care AI Agent enables you to ask supported questions about billing information in natural language. The helps you understand billing information that is available in Billing Care and BRM.

For example, you can ask the agent to explain a bill, show a previous bill, compare bills, show bill charges, or identify supported payment and accounts receivable actions.

The Billing Care AI Agent includes the following components:

Copyright © 2026, Oracle and/or its affiliates.

- **Billing Care Chat**: It provides the user interface for sending a billing question to the agent.
- **Billing Care AI Agent**: It validates the request, invokes the AI model in OCI, calls approved MCP tools, and returns a response.
- **AI Model in OCI (LLM)**: It interprets the request and assists with selecting an approved tool.
- **Billing Care MCP Server**: It validates a tool request and calls approved Billing Care REST APIs.
- **Billing Care and BRM**: They provide the billing information to be returned to the agent.

The workflow begins when Billing Care sends the requested question and available account context to the agent. The agent uses OCI LLM to interpret the request and select an approved Billing Care MCP tool. The MCP server validates the request and calls the applicable Billing Care REST API to retrieve the required billing information.

The MCP server returns the billing information as a structured response. The agent uses the retrieved information to generate a summarized answer and returns the answer to the Billing Care chat interface.

Copyright © 2026, Oracle and/or its affiliates.