Transitioning to Oracle Commerce Cloud

WHITE PAPER | MAY 29, 2019

ORACLE

Table of Contents

Introduction 3

Oracle Commerce Cloud Architecture 3

Oracle Commerce Cloud Concepts 4

Data Modeling 4

Data Migration 6

Integrations 6

Business Logic / Java Code 8

Customer-Facing Front End 11

Storefront 11

Personalization 12

Customer Service 13

Search 14

Reporting 16

Related Oracle Services 16

Conclusion 17

INTRODUCTION

Oracle Commerce customers have made significant investments in their applications for digital commerce and customer experience. The purpose of this document is to provide useful guidance on how to transition to Oracle Commerce Cloud from Oracle Commerce. It's important to understand every customer implementation is unique. Similar to how upgrades vary based on site build specifications and the level of customization, the level of effort and approaches vary when transitioning to Commerce Cloud.

ORACLE COMMERCE CLOUD ARCHITECTURE

Oracle Commerce Cloud is an API-first, cloud-native, SaaS application deployed within the Oracle Cloud. Oracle manages the hardware and operational components of the application while giving full control to businesses to create commerce experiences and applications unique to their needs. Oracle Commerce Cloud is a modern application with a focus on ease of use, speed-to-market, and developer flexibility,

SERVICES

Oracle Commerce Cloud is built upon a services-based model of components, which work together to deliver a high performance, scalable, extensible commerce platform. Each service's functionality is available from a public, documented REST API designed to be cloud-native. Upon these services, we built modern web applications for the storefront, administration (e.g., merchandising, search, experience management, reporting), and customer service.

UI EXTENSIBILITY

In the storefront and agent console, custom extensions can layer upon the available functionality and further fine tune it for specific needs. Developers code against the same versioned and backwards compatible APIs used by Commerce Cloud's native storefront, admin, and agent applications. This means the entire platform is an open set of services to build upon if desired. However, if not, there is already a complete solution from which to begin. Designers can use Oracle Commerce Cloud's OOTB "widgets" to quickly build and style their storefront, take a completely "headless" API approach to use commerce with existing web or native applications, or use an integrated approach combining the two methods.

SERVER-SIDE EXTENSIBILITY

Oracle Commerce Cloud allows developers to extend the platform. Server-side extensions let developers write code which will run in the Oracle Cloud. Developers write server-side extensions in Node.js which is executed securely away from shopper browsers. These extensions are used to create integrations, simple ETL transformations, or to implement entirely new functionality.

SCALABILITY

Oracle Commerce Cloud is designed to be horizontally scalable as traffic grows. The automated Oracle Cloud Infrastructure (OCI) allows for scaling hardware for the application as needed. Services are also purposely built as either single-tenant or multi-tenant. Core commerce services are single-tenant and dedicated to ensure noisy neighbors are not disrupting traffic flow while still ensuring seamless and automatic updates. Other services such as recommendations, wish lists, and experiments are built multi-tenant to allow for easier updates and reduced cost.

GLOBAL CDN

Global CDN is included in the platform. The CDN is pre-integrated and tuned to ensure shoppers worldwide will have fast access to content for their desktop or mobile device. APIs are also cached by the CDN. There is no need to configure and integrate a CDN separately.

ORACLE COMMERCE CLOUD CONCEPTS

Oracle Commerce Cloud introduces several terms and concepts found in more modern application frameworks not previously available in Oracle Commerce. These concepts make it easy to integrate and extend Commerce Cloud using open standards while ensuring new updates to the application can be delivered continuously without disrupting shoppers, business users, or developers. These include the following:

- **Webhooks**: User-defined REST HTTP callbacks where, within Oracle Commerce Cloud, a developer can configure a URL and other options to invoke at various points. Webhooks have two different types, event and functional.

  Event webhooks are asynchronous and do not block the shopper. They allow Commerce Cloud to initiate sending information to another system. An example of an event webhook is OrderSubmit. When an order is submitted, the order data is sent to the URL configured for the webhook. The URL will typically be the order management system (OMS) or a gateway service to accepts orders. The receiving system will send an HTTP response to indicate success or failure.

  Functional webhooks are synchronous for the shopper and require a response. This allows custom logic deployed outside of Commerce Cloud to be inserted into the commerce flow at required times. An example of a functional webhook is a shipping calculator. When a shipping calculation is required, a request containing relevant order data is sent to the configured URL and waits for a response (e.g., the calculated shipping price). This is one of the ways developers can customize and extend Commerce Cloud while maintaining the ability for Oracle to continuously update the application. For more information, please see the online manuals on webhooks.

- **Widgets**: User interface (UI) assets used to build the Storefront and Agent UIs. Oracle provides over 80 widgets out-of-the-box, and developers can also create their own for completely unique and differentiating design and functionality. Widgets are built using standard web technologies such as CSS3 and JavaScript making it easy for modern web developers to adopt the platform. For more information on widgets, please see the online manuals on widgets.

  Server-side Extensions (SSEs): Services built to perform custom server-side logic using a node.js environment deployed in the Oracle Commerce Cloud instance. SSEs can create new REST endpoints and be invoked from webhooks or client-side JavaScript contained in widgets. An example of an SSE is a custom shipping calculator. For more information on working with SSEs, please see this page.

DATA MODELING

Oracle Commerce offers data modeling capabilities commonly referred to as repositories. Repository functionality in Oracle Commerce represents the single method for the application to define and access data. Oracle Commerce defines all the major data objects out-of-the-box (such as profile, catalog, order, etc.) and customers commonly extend these out-of-the-box objects or create entirely new data objects to meet their specific requirements. Repositories consist of a database schema and an associated XML configuration to describe the schema to the application. The process of extending

repositories is typically a development task and there are many options available to the developer to define the data model. While the flexibility is helpful, it also doesn't prevent developers from introducing performance issues with overly complex data structures.

Oracle Commerce Cloud provides a comparable level of flexibility with an approach designed to reduce the amount of effort and prevent performance implications when modifying the data model. Dynamic attributes can be added to the following objects:

- Product
- SKUs
- Profile
- Orders
- Order Line Items

The process for modifying the data model in Oracle Commerce Cloud is also much easier and quicker. All data model modifications can be accomplished via the API. In addition, Oracle Commerce Cloud gives business users the ability to perform simple data modeling tasks such as adding a new property to the catalog directly in the admin tool without the involvement of development teams. For more information about data modeling in Oracle Commerce Cloud, please review the online manuals.

Best practices for migrating data models

When migrating data models from Oracle Commerce to Oracle Commerce Cloud, Oracle recommends the following:

- Utilize the Commerce Cloud functions for data extensions as documented for adding fields to existing objects such as products, SKUs, profile, orders, order line items, etc. This includes creating new item types such as product and SKU types. For more information, visit the Oracle Commerce Cloud Help Center.
- For completely new repositories, utilize Oracle Database Cloud Service, or other PaaS services, for data storage. Data from external PaaS services can be accessed and incorporated into Commerce Cloud logic in a number of ways.
    - Data can be accessed directly from the storefront. The storefront allows for custom JavaScript to be inserted into pages and can connect to the database cloud service to access the data. The information can be presented to users or populated in order fields and other objects.
    - Data can be accessed from SSEs. An SSE allows the creation of new endpoints to offer the ability to bundle custom logic and custom data into simple-to-consume services for the storefront client. Applications running as SSEs can access external systems via standard REST service calls to those systems.

In Oracle Commerce, developers sometimes required the ability to create custom java-backed properties as it provides a convenient way to install data-oriented logic right into the data objects themselves. In some cases, this was done in order to simplify existing profile properties making it easier to write personalization rules against the properties. An example is a new property called displayOnSite. It's not uncommon to have logic determine whether a product on the site should be visible or not. A developer could create this property with logic that evaluates various other properties of the product to ultimately return a true or false value (display on site – true or false). To achieve similar custom functionality in Oracle Commerce Cloud, use an SSE.

DATA MIGRATION

Data migration consists of moving data from a source system to a target system. In this case, the source system is Oracle Commerce, and the target system is Oracle Commerce Cloud. A preliminary step as described above is to review and understand the data model of both the target and source systems and define the mapping between them. Once this is understood, the process of actually migrating the data can begin. One important consideration is whether the data migration will be done once or if it will be recurring. This will depend on the implementation, but at a minimum, the migration should be built with scripts that can be executed across multiple environments for the purpose of testing.

Best practices for migrating data

Oracle Commerce Cloud Consulting, as well as numerous partners, have developed strategies and tools to make one time or recurring data migrations faster and easier. These tools generally leverage available public APIs (including bulk load data APIs) from Oracle Commerce and Oracle Commerce Cloud to extract the data, format it for import, and then import it via the APIs. These tools can be modified to reflect extensions and customizations based on the underlying repository structure.

INTEGRATIONS

When integrating Oracle Commerce to various external systems, there are many different options available. Developers could extend Oracle Commerce via coding and configuration, leverage most kinds of middleware, and use a variety of batch or real-time data feeds. This area is complex and tends to be challenging for customers since there are potentially hundreds of integration points for commerce sites. Integration points could include simple tag-driven integrations, such as Google Analytics. Others are more complicated, such as product / SKU import, order export, external tax calculation, and calling external business logic for things like product compatibility, punch outs, and others.

The integration approach of Oracle Commerce Cloud is much more streamlined and maintainable while remaining flexible. There are many integrations in Oracle Commerce Cloud already pre-built including numerous payment gateways (e.g., CyberSource, PayPal), numerous tax providers (e.g., Avalara, Vertex), and other Oracle applications (e.g., Responsys, Loyalty Cloud, CPQ) For more information, please review the online manuals about pre-built integrations in Oracle Commerce Cloud.

For system-level integrations, leverage the API's, webhooks, and Oracle Integration Cloud Service (OIC). For the shopper experience, many integrations can be done via custom widgets in the storefront designed to pull data directly from other systems as described in the migrating data models section of this document.

Oracle Commerce Cloud has been designed with an API- first, ensuring proper, public, efficient, granular, and documented APIs are available for all functionality. For events initiated by Commerce Cloud, such as the submittal of orders, or the creation of a new shopper profile, outbound REST webhooks call into middleware, such as OIC, or directly to the receiving system.

OIC allows for integrations to be created, maintained and monitored. These integrations are all created from an easy-to-use management console. The process of creating an integration is based upon dragging and dropping configurations and some simple wizard steps.

OIC provides a number of pre-built adapters to Oracle and third-party applications. Adapters can connect to cloud and on premise applications. The list for OIC adapters can be found here. Adapters, such as one available for Commerce Cloud, expedite connectivity by handling the underlying complexities of connecting to applications using industry-wide best practices. With the adapter, Commerce Cloud APIs and webhooks are all visible and available for use within the OIC UI.

To reveal the available APIs and webhooks, users create a connection by providing minimal connectivity information for each system. Users can leverage the drag-and-drop management console to easily create flows connecting one application's APIs to another application's APIs.

Once applications are integrated and activated to the runtime environment, a dashboard displays information about the available integrations so users can monitor the status and processing statistics for each integration. The dashboard measures and tracks the performance of transactions by capturing and reporting essential information, such as throughput, the number of messages processed successfully, and the number of failed messages. Users can also manage business identifiers that track fields in messages and manage errors by integrations, connections, or specific integration instances. For more information on OIC, see the Oracle Integration Cloud documentation.

Best practices for migrating integrations

Payment and Tax:

For common integrations such as payment gateways and tax services, use the out-of-the-box integrations with providers such as CyberSource, PayPal, PaymentTech, PayU Latam, Avalara, and Vertex. Other payment and tax engines can be integrated using the open APIs and frameworks to connect to any provider. Integration code can be implemented and run on the provided node.js platform as an SSE. This approach removes the need for another platform and associated PCI challenges.

Order Management Systems:

For other integrations with an OMS, use the standard, public APIs and webhooks provided. Oracle Commerce Cloud generates an outbound REST webhook call for submitted orders to be routed to the OMS. Oracle also provides native integration to Oracle order management.

Product Catalog and Inventory:

Similarly, Oracle Commerce Cloud provides APIs for import of products, SKUs, prices, inventory, etc. with both singular and bulk APIs. These webhooks and APIs can be accessed via OIC, custom integration code running on Commerce Cloud's node.js as an SSE, or any other middleware or integration code.

Pricing and Promotions:

Oracle Commerce Cloud provides APIs for import of pricing with both singular and bulk APIs. Promotions can be migrated directly from Oracle Commerce to Oracle Commerce Cloud.

In addition, Oracle Commerce Cloud provides the ability to use an external promotions engine and provides an open framework for pricing.

Third Party Extensions:

Oracle and its partners have provided integrations for use, which can be found in the Oracle Cloud Marketplace. Where such integrations exist, it's suggested to use them. Integrations found on the Cloud Marketplace include capabilities related to tag management, payments, user-generated content, marketplace development, and more. For presenting information to shoppers from multiple systems (such as ratings/reviews or bill presentment), utilize the storefront extension framework to add custom widgets to either call out directly to the source system, or call into custom endpoints on the SSE node.js server to pull data from other systems.

BUSINESS LOGIC / JAVA CODE

The primary development language used with Oracle Commerce is Java. Customers who have implemented Oracle Commerce will have some amount of Java code written to perform custom business logic for their site. Oracle Commerce Cloud does not require and does not expose Java as a development language for extensions. Instead, JavaScript is the primary language used when implementing Oracle Commerce Cloud for widget/extension development as well as server-side extensions. This approach provides a method that is more pervasive and easier to use. Using and learning a scripting language is much faster and easier than a compiled language like Java. Identifying, rationalizing, and determining what custom business logic should be implemented in Oracle Commerce Cloud using JavaScript is a key task for modernizing your commerce application. Below are the various forms of custom business logic used on many Oracle Commerce sites along with a recommended approach for similar logic using Commerce Cloud.

Form Handlers

In Oracle Commerce, developers extend or create new form handlers to perform custom logic upon interaction with forms. Form handlers are written in Java. They require code deployment and additional QA to get them into production because form handlers are almost always customized to some extent by Oracle Commerce customers. This means deployment must be restarted anytime a change is made, making it a lengthy process. The most common form handlers to be extended are for cart functions (add, remove, update, etc.), and profile management (log in, register, etc.).

One possible example is when someone registers on the site, the form handler takes the email address and performs a lookup in a backend customer database to see if he/she is an existing customer. Another possible example is when someone logs in, the form handler could make a call to a backend loyalty system to get the current point balance. In general, form handlers can be used to perform data validation, capturing/storing data, and handling error conditions.

There is no analogy to form handlers in Oracle Commerce Cloud. Custom logic when interacting with forms can be handled either as part of the widget containing the form. This allows for quick changes to be made. Another option is to use an SSE that exposes new endpoints which layer additional logic on top of the native REST endpoints and therefore providing additional security. Both methods dramatically reduce sprint cycles as compared to Oracle Commerce.

Droplets

Droplets in Oracle Commerce are reusable components used in display logic in a JSP-based front end. Oracle Commerce provides many pre-built droplets, but customers also have the ability to build their own.

Droplets in Oracle Commerce are built using Java code, which requires a full code deploy when a change is made. Any customer-built droplets should be rewritten in JavaScript and either added to out-of-the-box widgets or placed in custom widgets in Oracle Commerce Cloud. This offers an easier experience and a more agile site than using droplets in Oracle Commerce. The approach in Oracle Commerce Cloud also gives greater control of the site, as droplets in Oracle Commerce are proprietary.

Scheduled services

It is common for customers to create several scheduled services to perform a variety of tasks, often performing data import or export processes. Users could also perform a specific business task like calculating totals for custom reports. For these types of activities, Oracle Commerce can send the customer an email to notify them.

For customers looking to migrate to Oracle Commerce Cloud, any scheduled services will need to be defined and managed external to Oracle Commerce Cloud. For example, the customer could create a scheduled Java application running in Oracle PaaS which would be coded to use the Oracle Commerce Cloud APIs to perform its job.

Commerce pipeline

Commerce pipelines are used to define a flow of execution for certain tasks like repricing an order, performing tax calculation, or handling a submitted order. Oracle Commerce customers have full access to add and modify commerce pipelines. Some of the pipeline chains in Oracle Commerce are not relevant in Oracle Commerce Cloud, such as “add to cart” as it’s all handled on the client widget. This allows customers to extend via changes to storefront widgets which improves browser performance and decreases development time.

For relevant pipelines such as the Order Submit pipeline, customers may have customizations where they need to perform similar logic in Oracle Commerce Cloud. There are numerous pre-defined common points of extension where developers can use webhooks to call out to external logic or systems to perform tasks. For example, webhooks can be used to reserve inventory or validate the order. Other examples include calling out for external pricing, calling out to an external system for shipping, pricing, etc.

While these webhooks can be pointed at external systems, they can also be routed to custom code running on the Oracle Commerce Cloud SSE node.js platform. This allows for logic, such as replacing the out-of-the-box shipping calculator with a custom calculator, all within Oracle Commerce Cloud. SSEs simplify both business and technical overhead while empowering developers to build integrations and custom code on Oracle’s highly available infrastructure. SSEs are secure since they run server-side. They are built using JavaScript to stay consistent with the frontend design language. They also run on the same server as Oracle Commerce Cloud to optimize performance.

Custom Logic Summary

The above sections outline a range of different types of custom logic many Oracle Commerce customers have used to insert business-specific capabilities into their systems. With Oracle Commerce Cloud, the approaches are designed to increase developer flexibility, reduce development time, and increase agility. In summary, customers implementing Oracle Commerce Cloud should:

- Utilize widgets to insert custom JavaScript logic into the storefront. This logic can be used to validate data, call into additional systems, or apply additional logic.

- Widgets can contain custom logic as JavaScript, or can also call into new storefront API endpoints routed to custom logic running on the SSE platform (node.js) included with Oracle Commerce Cloud. The SSE code can perform custom logic and call into Oracle Commerce Cloud APIs or external system APIs.
- Leverage Commerce Cloud webhooks to insert custom logic for key functions, such as shipping calculations, order validation, inventory checks/reserve, etc. These webhooks can be routed to external systems, or to custom logic running as SSEs.
- For very complex logic, use an external PaaS solution, such as Oracle Java Cloud, and access it via webhooks, SSEs, or directly from the client via custom widgets.

Event handling

In Oracle Commerce, developers can code or configure any out-of-the-box events as well as create custom events. They can enable or disable events. When enabled, developers can write event listeners to look for those events and perform custom logic when they occur. This provides flexibility for the developer to hook into events and perform the required logic.

In Oracle Commerce Cloud there are a number of pre-defined event webhooks to be invoked when those events occur. This saves time and streamlines processes to speed time to market. For events which have an existing webhook, it is suggested to use the webhook to notify any systems listening for the event. For an activity without a pre-defined webhook, developers can extend storefront widgets to fire events to external systems directly, or via custom logic running as an SSE.

REST Usage

Oracle Commerce provides an extensible framework to expose any underlying logic as a REST endpoint. In addition, the framework includes a range of tools to chain various actors together, filter results and more. Implementers can expose existing logic as a REST service, chain together multiple underlying calls into a single service, or expose custom logic via REST.

Oracle Commerce Cloud streamlines this by providing a complete set of REST APIs to access all existing functionality in the Commerce Cloud platform. All API documentation is publicly available. There is no need to configure a REST layer to expose underlying logic. Everything is available via services, which is faster to set up. Oracle Commerce Cloud provides access to all functions. For situations where it is desirable to chain together multiple pieces of logic, or include logic from external systems, an SSE can be used. SSEs allow developers to write custom node.js applications and expose them as new, custom APIs. These APIs can then call a complex set of logic, which may call multiple Oracle Commerce Cloud endpoints, endpoints from external systems, or use native logic in the node.js application to provide the desired function.

Generic custom nucleus components

Oracle Commerce allows customers to define new custom components to perform different tasks. There is a configuration aspect and often Java code as well. Depending on the requirements this isn't required.

Oracle Commerce Cloud was built to optimize commerce functionality and get sites live quickly and efficiently. Any custom components need to be analyzed to determine their purpose and devise an approach as to how this functionality should be manifested in Oracle Commerce Cloud, as the functionality may already exist in the platform or can be leveraged through integrations or extension methods already mentioned.

CUSTOMER-FACING FRONT END

STOREFRONT

With Oracle Commerce, there are many different ways customers can design and architect their storefront. One way is to simply use JSP, leveraging the JSP tag libraries offered with Oracle Commerce. Some customers have elected to access Oracle Commerce via web services APIs and render the pages using whichever JavaScript framework they choose. Another common approach for developing the storefront is to use Experience Manager (XM) in combination with one of the technical platforms mentioned. With XM, the business user can use existing page templates and cartridges to structure the overall design of the site. Developers can create custom page templates and custom cartridges to provide unique functionality as required by the business.

Oracle Commerce Cloud includes a full-featured storefront, a drag and drop page layout tool called Design Studio, a collection of page building blocks called widgets, tools for A/B testing, and tools for personalization. Instead of XM page templates, there are layouts. Instead of XM cartridges, there are widgets. Custom layouts and widgets can be created as required by the business to deliver dynamic experiences based on unique needs. There are libraries and code techniques used in Oracle Commerce Cloud which are considered industry standard making it much easier to find developers who can step in and provide value very quickly.

In addition, the Oracle Commerce Cloud storefront is responsive out-of-the-box meaning no additional work is required to support mobile. The storefront can be easily configured by business users and can be customized and extended to meet branding and experience needs so customers have the most optimal experience on the site. The Design Studio is very intuitive; it is much easier to use compared to Experience Manager, giving business users the flexibility they need to get sites live quickly and efficiently with easy drag-and-drop page layouts and site design tools.

If a customer wants to use other tools rather than the built-in tools provided with the storefront mentioned above, Oracle Commerce Cloud can be run in a headless mode leveraging the full set of REST APIs provided from whatever front end platform is desired.

Best practices for migrating storefront

Since the two storefront paradigms are different, there is no direct way to migrate one to the other. Transitioning from Oracle Commerce to Oracle Commerce Cloud is an opportunity redesign the storefront and decouple the frontend from the backend. This results in greater agility and a better experience for business users and developers.

- Leverage Oracle provided out-of-the-box widgets to provide core commerce functionality and a starting storefront site. Modify these widgets as needed for look and feel (i.e., branding) and for any custom functionality to quickly build and style the storefront.
- Create new widgets for the storefront using the extensible, standards-based framework for new functionality or integrations not provided by out-of-the-box widgets or via the Oracle Cloud Marketplace.

PERSONALIZATION

Oracle Commerce offers a powerful range of personalization capabilities including:

- Segments: Define a set of shoppers with similar characteristics.
- Content Targeters: Define which content groups to show to shoppers in specified segments at specified times.
- Scenarios: Define a sequence of time-based events and content to show to shoppers for each event.
- Experience Manager Rules: Define page content and layout based on the shopper's membership in Segments, and the state of the shopping session.
- Promotions: Define discounts that can be restricted to certain segments of customers.

Oracle Commerce Cloud offers a streamlined and rich set of capabilities which are more powerful than the capabilities in Oracle Commerce. They include:

- Audiences: Provide the ability to easily define sets of shoppers based on rules against out-of-the-box or custom properties of anonymous and/or registered shoppers or B2B account contacts. Audiences are analogous to segments in Oracle Commerce.
- Shopper lifetime attributes: Track values such as lifetime spend, average order value, and the dates shoppers registered, last visited the site, and made a purchase. Values are tracked automatically for B2C shoppers. These attributes provide a way to quickly set up common audiences without the need to define custom code, as would have been the case with content targeters in Oracle Commerce. This allows businesses to more precisely target different audiences and provide personalized and relevant experiences to them.
- Promotions: Define discounts that can be restricted to certain audiences of customers.
- Content Variation Slots: Allow merchants to define a set of storefront widgets to be shown to shoppers in specified audiences within defined time periods. Slots are similar to targeters in Oracle Commerce. Slots are more powerful than targeters because they allow merchants to control entire sections of the page layout easily from Design Studio. This approach increases agility by making it very simple to convert audience definitions into shopper experiences.
- Experiments: Allow merchants to conduct A/B tests throughout the entire site experience. Experiments is the integrated A/B testing solution available out-of-the-box with Oracle Commerce Cloud. Tests can be associated with audiences so merchants can test out new content with an audience using commerce-specific data without custom integration. Merchants gain greater insight to what drives desirable shopper behavior, and can optimize these high-value elements of their sites. With experiments merchants can get statistical insight to understand what will resonate with shoppers in order to increase engagement, conversion rates, and order values.

Overall, Oracle Commerce Cloud's redesigned UIs and additional capabilities make personalization much easier. Oracle Commerce Cloud vastly improves the availability of actionable data out-of-the-box, including data for both registered and anonymous shoppers like earliest visit date, geolocation, total number of visits, referring site and URL, UTM campaign parameters, and much more. Tasks are now streamlined using the actionable data which previously took a variety of skills, time, and IT involvement in Oracle Commerce. This approach provides more valuable data for merchants through a single UI to hyper-personalize each experience for their customers without custom integration.

For example, a common task is to identify people who have placed at least three orders on the site, but have not ordered anything within the last two months, and offer them a promotion on their next purchase with added messaging on the site to make them aware of the offer.

In Oracle Commerce, this requires a whole range of skills to accomplish. These skills include a:

- Database developer to extend the database schema to capture last purchase date, number of purchases, etc.
- Developer who can write the XML to extend the profile repository schema to create these profile properties.
- Java developer or scenario author to provide the code for capturing the data about number of orders, last order date, etc. and add it to the profile.
- Frontend JSP developer to modify the site pages to add messaging about the promotion offer.
- Merchandiser to use the BCC to create the audience and the promotion, and associate the promotion with the audience.

The update would then have to be deployed to the website's code and pages. This process could take a few weeks, and require non-trivial IT involvement, wasting valuable time to target customers.

In Oracle Commerce Cloud, a single merchandiser can do all of this from the Admin tool in literally minutes without involving IT or deploying new code.

Best practices for migrating personalization

Personalization is heavily dependent on shopper profile data. Please follow the guidelines outlined earlier in this document related to migrating the data model and data for the shopper profile.

- Leverage the streamlined Oracle Commerce Cloud Admin tool to quickly create audiences previously designed as Oracle Commerce segments. This is an valuable opportunity to perform system clean up and determine which segments are appropriate going forward.
- Create Content Variation Slots using Design Studio to display the appropriate content to a particular audience. It is also recommended to consider using Content Variation Slots at a higher level than previous implementations of Content Targeters to control the overall page design and content. This is similar to the manner Experience Manager may have been used with Oracle Commerce.

CUSTOMER SERVICE

Oracle Commerce customers have the option to use an application called Customer Service Center (CSC). CSC is designed as a call center application for call center agents to manage customers, orders, returns, exchanges, appeasements, etc. Some customers extended the UI and functionality of CSC to accommodate specific customer service and order management requirements.

With Oracle Commerce Cloud, there is an application similar to CSC called Agent. The Agent application provides the same core commerce functionality CSC provides. In addition, it allows agents to browse the site on behalf of customers. The out-of-the-box Agent application leverages the same documented APIs of Oracle Commerce Cloud.

Best practices for migrating customer service

Core CSC functionality in Oracle Commerce exists in the Agent application of Oracle Commerce Cloud. However, necessary call center functionality can vary greatly from merchant to merchant. To accommodate custom functionality, the Agent UI can be customized and extended using the same frameworks and tools as the storefront. Oracle Commerce Cloud Design Studio can be used to create Agent experiences specific to each merchant's needs. Custom functionality can be incorporated using the same development principles outlined earlier in this document (e.g., custom widgets, SSEs).

In addition, all Agent functionality is exposed via REST APIs and webhooks. This allows you to incorporate Agent functionality into any customer service application, or build your own.

SEARCH

Oracle Commerce customers have access to the full capabilities of Endeca. In addition, there is no definitive integration model for Oracle Commerce customers, and implementations vary based on the following:

- Site development guiding pattern (e.g., Commerce Reference Store (CRS/CSA)
- Custom data integration (e.g., direct-to-database, export-based, etc.)
- Custom data patterns
- Custom "division of labor" between ATG and Endeca (e.g., search repository IDs for products and then retrieving information from the repository vs. search driving entire product listing page)
- Use of Experience Manager for merchant management of pages
- Use of custom cartridges to encapsulate application logic
- Use of existing assets for controlling the data model (e.g., Product Catalog Deployment Template, Forge Configuration Manager, etc.)

Another factor for consideration is which application (ATG or Endeca) was implemented first. Typically, ATG customers adding Endeca will have a more CRS-based approach for implementation, whereas Endeca customers adding ATG may have a more complex and customized Endeca implementation.

Regardless of which application was implemented first, merchants will find value in how search has been incorporated into Oracle Commerce Cloud. Oracle Commerce Cloud includes search capabilities built into the solution and is configured to easily index and search catalog data. In addition, many of the same controls for managing search experience and results are provided within the Commerce Cloud Admin UI. The experience in Oracle Commerce Cloud is streamlined to focus on core commerce tasks, such as ordering products and facets. The UI is intuitive, allowing business users to set it up themselves, without any developer assistance.

Features available in Oracle Commerce Cloud (as of 19B release)

A number of key features are available in Oracle Commerce Cloud as of the current release:

- Spell correction and 'Did-you-Mean' suggestions
- Dynamic stemming
- Thesaurus management
- Keyword redirects

- Type-ahead
- Facet management
- Facet ordering/display
- AI Search
- Dynamic Curation
- Search Data API / Indexing Non-Catalog Content
- Boost and Bury
- Type-ahead Framework

Non-Catalog Content

With Oracle Commerce, customers can implement the Content Acquisition System (CAS) crawler to browse file systems or URLs. This is deprecated in Oracle Commerce v11.3.1, and will not be mirrored in Oracle Commerce Cloud. Instead, customers will be encouraged to handle the crawling of content independently and transform into structured data and upload this structured data to the Non-Catalog Content API (Oracle Commerce Cloud).

Performance/Scalability

For larger Oracle Commerce sites, performance and scalability was addressed using common best practices required for highly performant on premise applications. This often included extensive consulting engagements involving architects and engineers to plan out environments and hardware requirements for peak. They were also responsible for configuring all aspects of the system (e.g., database, export routines, data volumes, etc.).

With Oracle Commerce Cloud, search has undertaken a number of initiatives to provide the required scale with less effort. These include the following:

- Wide-record indexing to reduce the need to use a “variant producer” which results in a larger number of indexed records.
- Optimization of indexing times
- Optimization of export capabilities (e.g., price groups)

Best practices for migrating search

Given the variations described above, transitioning to Oracle Commerce Cloud can take a number of different forms. Consideration needs to be given to the functions deployed, and how to ensure partners, developers and merchants can replicate those functions (ideally through custom scripts that call Oracle Commerce Cloud APIs). A few guidelines include using the following:

- Out-of-the-box Commerce Cloud Search capabilities for indexing key data such as the commerce catalog.
- Admin tool to manage the search experience and results. The Admin provides the ability to manage key search aspects such as keyword redirects, thesaurus entries, index fields, weighting, facets, and more.
- Search Data API to index non-catalog content.

Once the data is loaded into the catalog, merchants should do the following:

- Choose which properties should be facets and which should be multi-select.
- Choose which properties should be available for search.
- Add the properties to search, in order of importance, to the Searchable Field Ranking “All”.
- Add key properties to match for type-ahead. These should be added in order of importance to the Searchable Field Ranking “Typeahead”.
- Add the facets to appear in navigation to the Facet Order entry “Default” and enable “Include Remaining Facets” so any new facets appear automatically.
- Review any existing thesaurus entries and add these to Commerce Cloud.
- Set up the “Default” entry for Dynamic Curation.

REPORTING

Oracle Commerce provides a high-performance reporting solution and includes extensive out-of-the-box reports, as well as a flexible tool for custom and ad-hoc reporting, Oracle Business Intelligence.

Best practices for migrating reporting

Oracle Commerce Cloud provides an extensive set of out-of-the-box reports to help continually monitor and measure site performance. In addition, Oracle Commerce Cloud leverages the same front-end tool, Oracle Business Intelligence, for generating custom and ad-hoc reports, though the underlying data model used for report generation is different. Custom reports created in Oracle Commerce using Oracle Business Intelligence can be recreated in the Oracle Commerce Cloud Oracle Business Intelligence instance using the revised data model. This gives merchants the insights they need to optimize the site experience for their customers.

Most Oracle Commerce Cloud customers have implemented some sort of web analytics solution to supplement their out-of-the-box reports or to create reports customized for their own reporting needs. This flexibility allows customers to tailor Oracle Commerce Cloud to their business and get the most valuable insight.

RELATED ORACLE SERVICES

Oracle Commerce Cloud is part of Oracle’s extensive cloud services. The broad offerings from Oracle allow merchants to leverage the functions and services they need to create compelling customer experiences, and to integrate and operate within unique organizations.

It is beyond the intent of this document to cover all the Cloud services provided by Oracle. However, a number of these services provide excellent synergies with Oracle Commerce Cloud and are worth noting. These include:

- Custom Development: Oracle Java Cloud, Oracle Container Cloud, Oracle Serverless Cloud
- Workflow and Approvals: Oracle Process Cloud
- Mobile app and Chatbots: Oracle Mobile Cloud, Oracle Visual Builder Cloud

- Integrations: Oracle Autonomous Integration Cloud Service, Oracle Identity Cloud, Oracle Database Cloud
- Identity and Single Sign On: Oracle Identity Cloud
- Database: Oracle Database Cloud
- Content & Digital Asset Management: Oracle Content & Experience Cloud
- Loyalty: Oracle Loyalty Cloud
- Configure, Price, Quote: Oracle CPQ Cloud
- Marketing Automation: Oracle Marketing Cloud
- Sales Automation: Oracle Sales Cloud
- Service: Oracle Service Cloud

CONCLUSION

Transitioning from Oracle Commerce to Oracle Commerce Cloud is an opportunity to modernize architectures, reduce customizations, increase agility, and ultimately create a better experience for shoppers, merchants, and developers. This document serves as a guide for mapping various functions and approaches in Oracle Commerce to Oracle Commerce Cloud and addresses the most common needs and topics. For additional information, please contact your Oracle representative.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com/oracle facebook.com/oracle twitter.com/oracle

Integrated Cloud Applications & Platform Services