Solution Design document

# Vision

* Customer 360
* Omnichannel Experience
* Real-time inventory visibility
* Efficient and personalized service
* Data-driven marketing journeys
* Scalability for growth

# Assumptions

* Segregating functionalities in phases.
* Existing systems are not to be sunset, at least in Phase 1.
* Lead and Opportunity management to be implemented in phase 2. SFMC will be used to run campaigns, gather leads and send to SF to be converted to Account.
* Account is the customer as it is an e-commerce application. Complexity of Business Account with multiple contacts is not considered for now.
* Order management, Inventory maintenance, sales transactions, financials, payments have still happened in existing e-commerce portal. SF is used more as a CRM tool rather than e-commerce platform, at least in phase 1.
* This is a first time Salesforce migration project started with fresh orgs with no accounts already present in SFDC.

|  |  |
| --- | --- |
| **Phase 1** | **Phase 2** |
| Customer Account/Single Contact Accounts | Lead management, Opportunity management. |
| Account Migration and sync | Customer portal replacing external e-commerce |
| Service cloud implementation | Inventory management and Financials |
| Marketing cloud implementation | Business Accounts support |
| Real time inventory check | Order Management organically |
| Order data load, integration and sync | Sunset obsolete integrations from phase 1 |
| Systems Integration | Service automation/Bots/AI |
| Data Migration |  |

# Salesforce Product Strategy & Licensing

## Customer 360

* + Use **Sales cloud + customizations** to build customer 360 view
  + **Use combination of sub-tab+custom+standard to give efficient and enriching view of all data in a single page view**.
  + Number of clicks can be reduced by total custom implementation using LWC components with greater effort. (Customer choice)
  + This is also a UX factor, based on which 360 page can be designed and implemented.

## Omnichannel Experience

* + This can be achieved **via Experience cloud** implementation of E-Commerce portal.
  + Customer, Sales Reps, Store reps and other employees all deal with Salesforce in this case.
  + Segregated for Phase 2.
  + **Real time inventory check is across clouds**. Can be exposed outside Salesforce if needed.

## Real-time inventory visibility

* + This is a component **that is available to be exposed in different views across clouds** so that Employees, Sales/service reps and managers can check for inventory.
  + **Uses LWC components, continuation, API callouts, graceful error handling.**
  + Implementation details are in this github project.
  + Some screenshots:

## Efficient and personalized service

* + Real time Case creation from different systems into Salesforce.
  + Using **service cloud** capabilities for the same.
  + Use standard email-to-case, call-to-case etc. for newer cases
  + Use integrations for Case intake from Different systems.
  + Service cloud provides activity history, Knowledge articles etc capabilities OOB.
  + Case summary, chat GPT etc. AI features can be used if costing allows Einstein for Service

## Data-driven marketing journeys

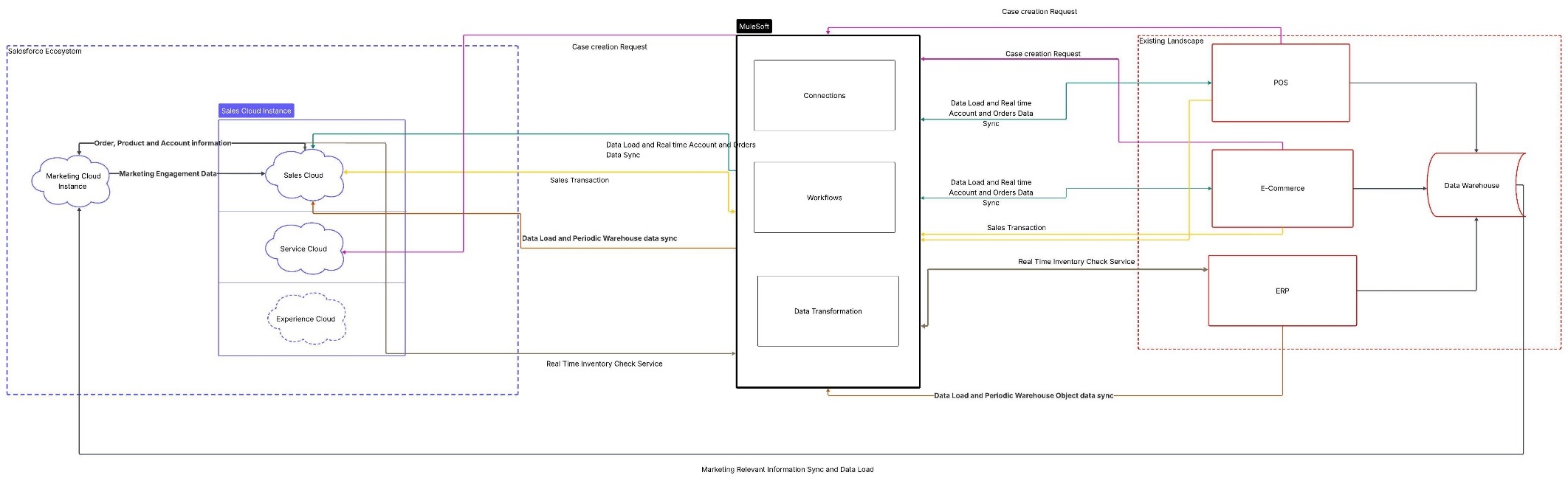
* + Use Salesforce **Marketing Cloud (Exacttarget)** as a solution product.
  + Use **Journey builder** for personalized customer journeys.
    - Existing customers
    - To tap new customers. (Future state.)
  + Use **Contact builder** for importing SFDC data.
  + Use Order Sync addon if costing permits, else sync using custom objects.
  + Using **automation studio, ftp server and DE**, build a pool of Product, Inventory and Order data that can be used for building journeys per customer.
  + Sync Marketing interactions back to SFDC using **ampscript** for customer 360 experience.
  + Better compliance with CAN-SPAM etc. regulations.

## Scalability for growth

* + Periodic archival of Orders upto ‘n’ years reduce stale data volume.
  + Legacy Order history and new order’s history to be dumped in Big Object OR
  + Use SF Connect for Legacy Order history if costing permits.
  + Bigger transaction to be segregated in separate asynchronous threads based on functionality
  + Platform Cache for efficient use of API limits.
  + More Details are covered in Data migration and Integration strategy.

|  |  |  |  |
| --- | --- | --- | --- |
| **Products** | **Add Ons** | **Licensing** | **Comment** |
| Sales cloud | Standard OOB | TBD |  |
| Big Objects | TBD | Either OR |
| SF Connect | TBD |
| Service Cloud | Standard OOB | TBD |  |
| Marketing Cloud | MCC | TBD |  |
| MCC for Order Sync | TBD |  |
| Mulesoft | TBD | TBD |  |
| Data Loader | NA | NA |  |
| Experience Cloud | Customer Community |  | Future state |
| VCS |  | TBD |  |

# System Landscape Diagram



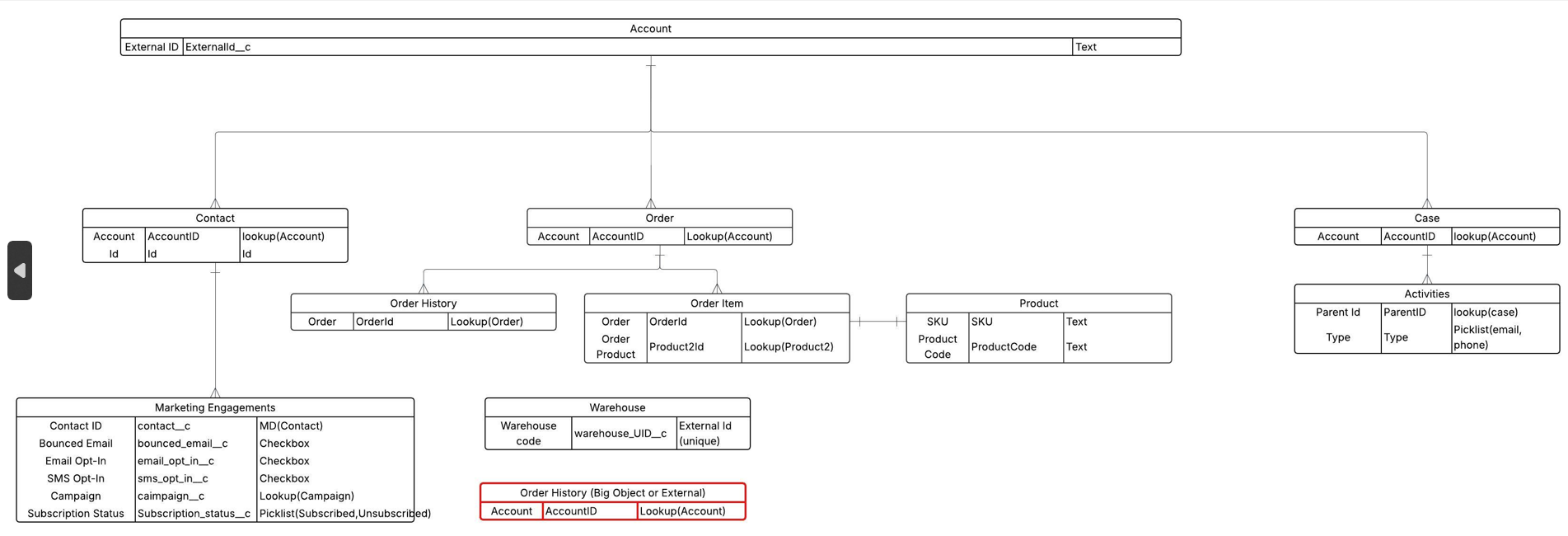
# Integration Strategy

* Note: For all API based integrations, mulesoft can be used as a central hub to manage integrations with different systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entities** | **What is being integrated?** | **Integration Pattern** | **Tool** | **Implementation Approach** |
| E-commerce/POS => Salesforce | Data: Account,Order, OI, Products  Process: Order cancellations, product modifications, Customer information updates | Remote Call-In  (API based) | Apex Webservices | * Data collation and transformation in MuleSoft. It send consolidated data to SFDC under 1 account. * REST based api call to salesforce with Client credentials flow. * Use POST, PUT/PATCH, DELETE methods to update orders and related line items and products * Real Time * If a Product is not already found in salesforce, create it and then associate with order. * Different order mergers to happen in Mulesoft under 1 account. * Real-time, Asynchronous |
| Salesforce => E-Commerce/warehouse | Data: Account,Order, OI, Products  Process: Order cancellations, product modifications, Customer information updates | REST Based API callout | Apex callout | * Call MuleSoft’s same endpoint with same credentials but the services for particular object update. * Real time, asynchronous |
| POS System to Salesforce for Sales Transactions |  | Data Virtualization (preferred) | SF Connect. Synchronous  OR Bulk API, scheduled. | * Why are Sales transactions needed in SFDC? It can be done in Phase 2 * Create a custom External Object for storing this information. * SF Connect can be used if cost permits. * If not, scheduled data sync using API and Batch |
| ERP ⬄ Salesforce Real Time Inventory check service | Data:  Warehouse Information | Platform Events | Mulesoft/Boomi Synchronous | * This is like Metadata, doesn’t need regular sync. After One time data load, platform events will sync any new record/changes. |
| ERP ⬄ Salesforce Real Time Inventory check service | Data:  Search results.  Process: None. | REST Based API callout |  | * SKUs is a multiselect picklist, warehouse is picklist. * Use this pair as parameter * REST API, GET Call. * If GraphQL is supported, prefer to use that. * Get real time results * Use techniques like Continuation, paging, CTA filters for better UX and performance. * Use Platform cache to fetch results from cache for same query within 10mins. |
| Marketing Cloud to Salesforce | Data:  Marketing engagements | Data Integration | MCC Synchronous | * TODO Using ampscript, send data to Sales cloud from SFMC. Create custom objects on SF side whose parent is contact that will hold interaction data of that contact like clicks, bounced, email subject, subscription information etc. * Display this consolidated information using LWC on the Tab of Account. |
| Salesforce to Marketing cloud | Data: Subscribers, Orders, products | Data Integration | Synchronized Data Extensions | * ContactID to be used as subscriber key in SFMC * 1 contact per account to be created in SFDC * As and when new customer is created/updated in SFDC from POS, sync it with SFMC. * Sync order data to SFMC using std. MCC. Need additional license to sync Order object. |
| Data Warehouse to Marketing cloud | Data:  Inventory and other as required for marketing. | Bulk Data Integration | FTP server and Automation studio. | * Option1 :FTP server can be used to update latest products and inventory details into SFMC where automation studio will consume the data to update the Data extensions. * Option 2: More efficiently, use integration that will update product information into SFMC w/o manually uploading to server. Needs 1 time data load for existing inventory. TODO |
| E-commerce/POS => Salesforce | Data:  Case  Process:  Case creation | Platform Events | Mulesoft/Boomi Synchronous, Real time. | * Case intake from multiple systems into SFDC. * Configure a PF whose trigger will create a case in SFDC and associate with appropriate order/account which will also be send in same PF. * Going ahead, case management will happen from SFDC centrally. |

# Data Architecture and Migration

## Object Data Model



## Data Migration Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Source** | **Destination** | **Approach/Tool** | **Frequency** |
| Account | Multiple | SFDC | 1 time Data load and real time sync with integration | 1 time followed by Data Integration as highlighted above. |
| Account/Contact | SFDC | SFMC | MCC | As required |
| Products | Data warehouse/ERP | SFDC | 1 time Data load and real time sync with integration | 1 time followed by Data Integration as highlighted above. |
| Order/OI | Multiple | SFDC | For past ‘n’ years, get all orders (and OI), use ETL to sort them under accounts, prepare OI under Orders etc. | 1 time followed by Data Integration as highlighted above. |
| Order History | Multiple | SFDC | 1. If Data virtualization is used this migration is not needed. 2. Big Objects can be used to store read-only data. Use long text area etc fields to store multiple products, invoice/financial data etc. Use ETL to collate data in this format. | 1 time data load as new orders will come into salesforce going ahead.  Use standard Order history going ahead. |
| Order | SFDC | SFMC | MCC | Scheduled Sync |
| Product | SFDC | SFMC | 1 time Data load and then scheduled sync | 1 time followed by scheduled sync |
| Inventory | ERP | SFMC | 1 time Data load (using FTP server) and then scheduled sync | 1 time followed by Data Integration as highlighted above. |
| Warehouse Information | ERP/Data warehouse | SFDC | Data Loader. | 1 time followed by Data Integration as highlighted above. |

* Reason to not use big objects:
  + Limited field support
  + Can’t be rendered on standard lightning UI. Need LWC/VF page
  + Needs bulk API to query/retrieve
  + We don’t need huge orders data for all accounts of all time at any given point. Hence, better to use data virtualization.
* Large Data Volume challenge mitigation

|  |  |  |
| --- | --- | --- |
| **Object** | **Challenge** | **Resolution** |
| Account | Skew problem for child objects. | * Split child records(contacts in our case) into 2 identical accounts is more than 10000. * Sort by account ID before migrating child objects. |
| Orders | Huge data leads to slowness of org while accessing them.  Inefficient performance of reports | * Import orders limited to ‘n’ years. * If older data needed, check order history. |
|  |  |  |

### Data Migration Strategy

* Identify limited pilot set of accounts and do complete migration of all children for these.
* Dry run in full copy sandboxes with near-production type limits/data/processes.
* Monitor and improve.
* Repeat for bigger set of accounts.
* As this is first time migration, no risk for live systems. Data migration can span over few days.
* Once migration is completed, 1 iteration for delta records is needed which should be quick.
* After delta migration, enable live data to flow in with integrations.
* Manually monitor for any missing data for a few days.

# Scalability and Performance

|  |  |  |
| --- | --- | --- |
| **Functionality** | **Challenge** | **Resolution** |
| Real time inventory check | Performance | Use lazy loading, offsets. If possible use GraphQL which will facilitate this.  Use platform cache for quick retrieval for a similar search query within a session/time frame. |
| Real time inventory check | User experience | Use continuation with LWC.  Spinners and progress indicators  Graceful handling of Timeout/API errors |
| Real time inventory check | API limits | Platform cache.  Check limits with Apex Database methods before API callout  Graceful handling with relevant messages |
| Order data reporting | Slow reports  Costly data retrieval | Use orders till ‘n’ years, rest archive in Order history.  Use Data virtualization for order history. |

# Security Model

* Index

|  |  |
| --- | --- |
| C | Create |
| R | Read |
| U | Update |
| D | Delete |
| VA | View All |
| MA | Modify All |
| NA | Not applicable |
| No UI access | This is to prevent impersonation using this profile |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Object** | **Persona** | | | | | | | **Access Levels** |
| **Sales Rep** | **Service Rep** | **Account Manager** | **Employee (assuming store managers)** | **System Admin** | **Integration User** | **Data Migration User (accesses are for limited time window)** |
| Account | RU | RU | RU | R | MA | CRUD | CRUD | Object Level |
| RU | RU | RU | R | MA | CRUD | CRUD | Field Level |
| Share based on a criterion | Share based on a criterion | Share based on a criterion | R | NA | No UI access | No UI access | Record Level |
| Read | | | | | | | OWD |
| Case | Hidden | CRUD | R | Hidden | MA | Hidden | Hidden | Object Level |
| Hidden | CRUD | R | Hidden | MA | Hidden | Hidden | Field Level |
| Hidden | Share based on a criterion | Share based on a criterion | Hidden | NA | NA | NA | Record Level |
| Read | | | | | | | OWD |
| Order | Read | Read/Write | Read/Write | Read | MA | CRUD | CRUD | Object Level |
| Read | Read/Write | Read/Write | Read | MA | CRUD | CRUD | Field Level |
| Read | Share based on a criterion | Share based on a criterion | Read | NA | No UI access | No UI access | Record Level |
| Read | | | | | | | OWD |
| Contact | RU | R | CRUD | R | MA | CRUD | Hidden | Object Level |
| RU | R | CRUD | R | MA | CRUD | Hidden | Field Level |
| Share based on a criterion | Share based on a criterion | Share based on a criterion | R | NA | No UI access | NA | Record Level |
| Read | | | | | | | OWD |
| Warehouse (access needed for selecting values in picklist only) | Read | Read | Read | Read | MA | CRUD | Hidden | Object Level |
| Only Warehouse Code and Name | Only Warehouse Code and Name | Only Warehouse Code and Name | Only Warehouse Code and Name | MA | CRUD | Hidden | Field Level |
| Read | Read | Read | Read | NA | No UI access | NA | Record Level |
| Read | | | | | |  | OWD |
| Order History | Read | Read | Read | Read | MA | CRUD | CRUD | Object Level |
| Read | Read | Read | Read | MA | CRUD | CRUD | Field Level |
| Read | Read | Read | Read | NA | No UI access | No UI access | Record Level |
| Read | | | | | | | OWD |
| Marketing Engagements | Read | Read | Read | Read | MA | CRUD | Hidden | Object Level |
| Read | Read | Read | Read | MA | CRUD | Hidden | Field Level |
| Read | Read | Read | Read | NA | No UI access | NA | Record Level |
| Read | | | | | | | OWD |

* Apart from this, Customer 360 view visibility can be handled using visibility options