Brought to you by Marketing Cloud Practice

Marketing Cloud Data Skills Development Session 1

Author: Daniel Kuo

Last Edited: 10/30/2023

Table of Contents

Introduction

Overview of concepts

Getting Started

What is this workshop about?

- A forum to work on common data model designs together
- Conversational workshop, so please feel free to ask questions
- Might be basic for some, difficult for others, so let’s help each other
- Remember, a pattern being suitable for one situation may not work for another

If you did not do so already, you should sign up for a MCDO learning Marketing Cloud account!

Data Model Basics in Marketing Cloud

Quick Review & White Boarding

Quick White Boarding

Let’s Start with the Basics

Whiteboard

Data Modelling Patterns in Marketing Cloud

Standard Data Models, Email History, Journeys

Introduction

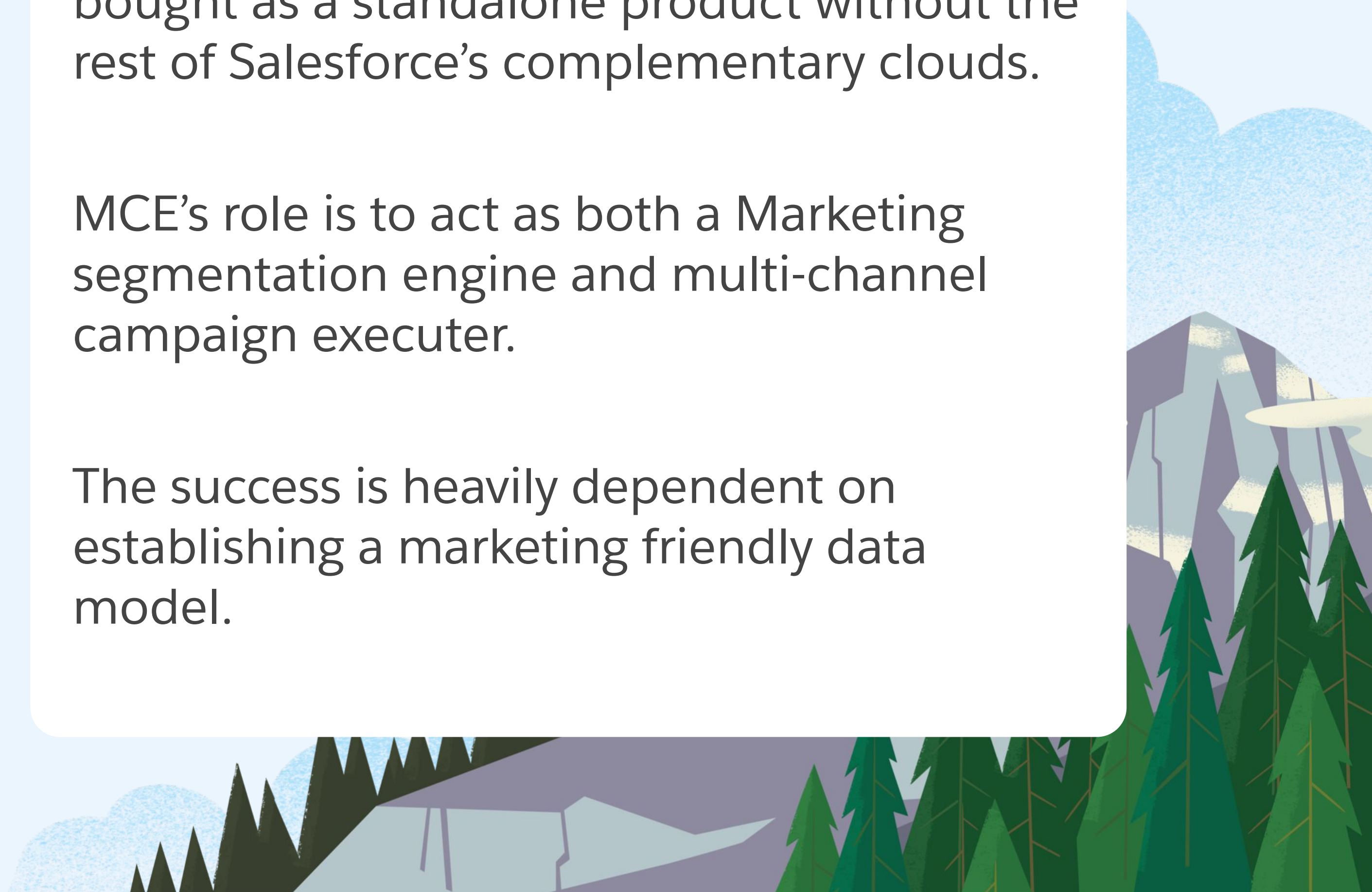
MCE ONLY ARCHITECTURE

- Only MCE is purchased
- No other accompanying SF Clouds
- MCE likely to be the marketing source of truth

This architectural pattern is for when MCE is bought as a standalone product without the rest of Salesforce's complementary clouds.

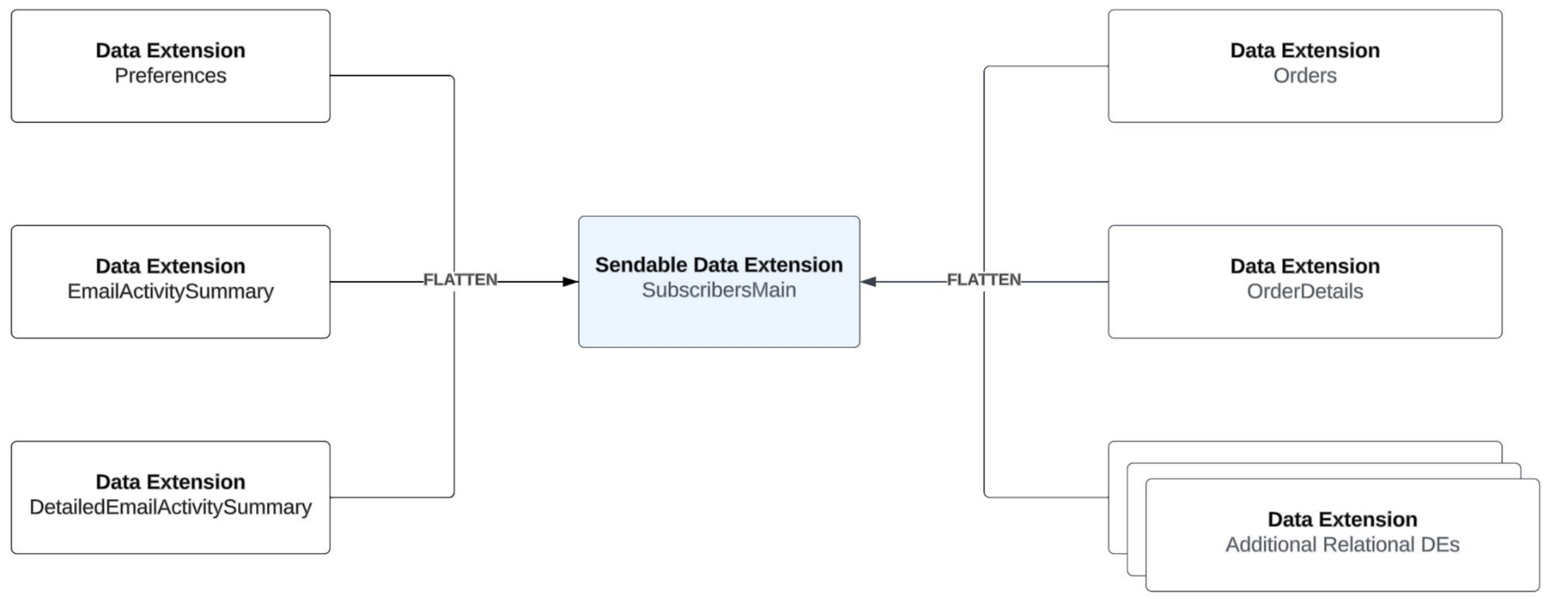
MCE's role is to act as both a Marketing segmentation engine and multi-channel campaign executer.

The success is heavily dependent on establishing a marketing friendly data model.



Architecture - Flat Primary Table

Overview



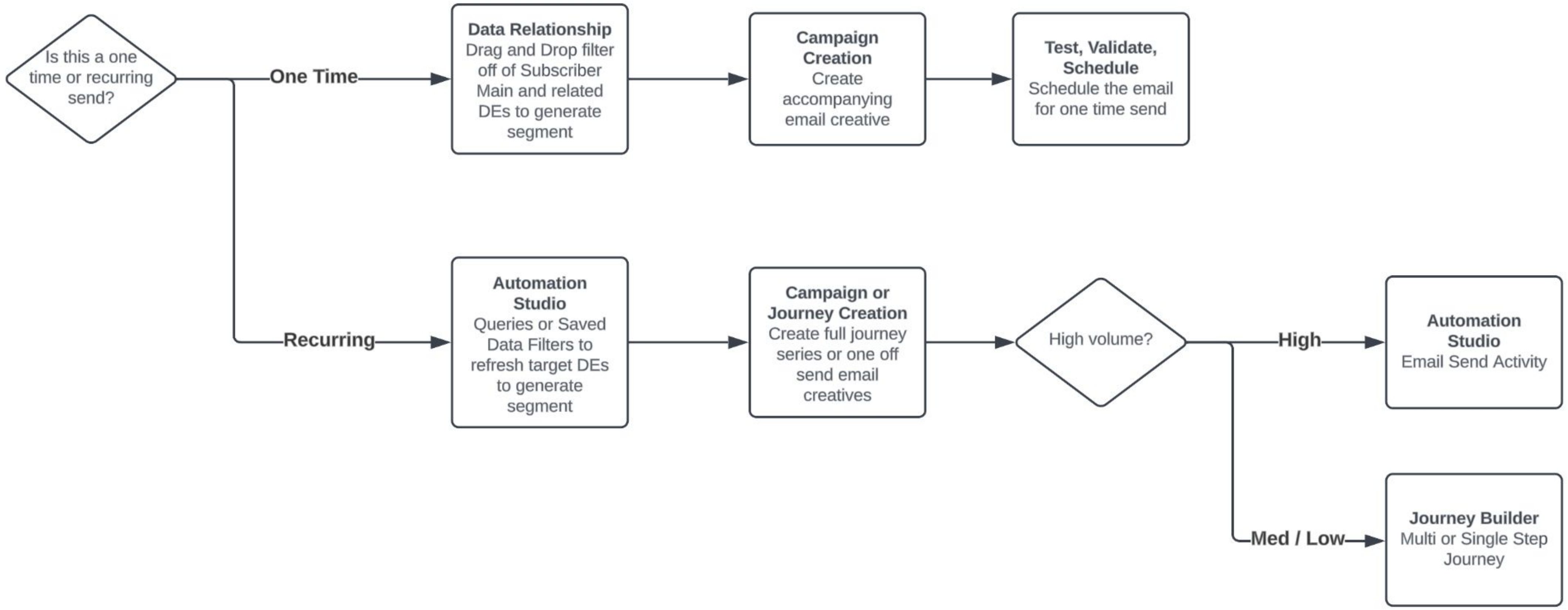
Alternative to Core Data Model, sometimes data is flattened into a single table.

- SubscribersMain remains, and has the universe of unique subscribers.
- All data in related table is flattened into corresponding fields in SubscribersMain
- 1:1 data writes in directly, 1:Many data is flattened; for example, order amounts are flattened into life time value
- Resulting in a large single table that drives all segmentations and Data Designer decisions

DE Sample

Usage Pattern - Flat Primary Table

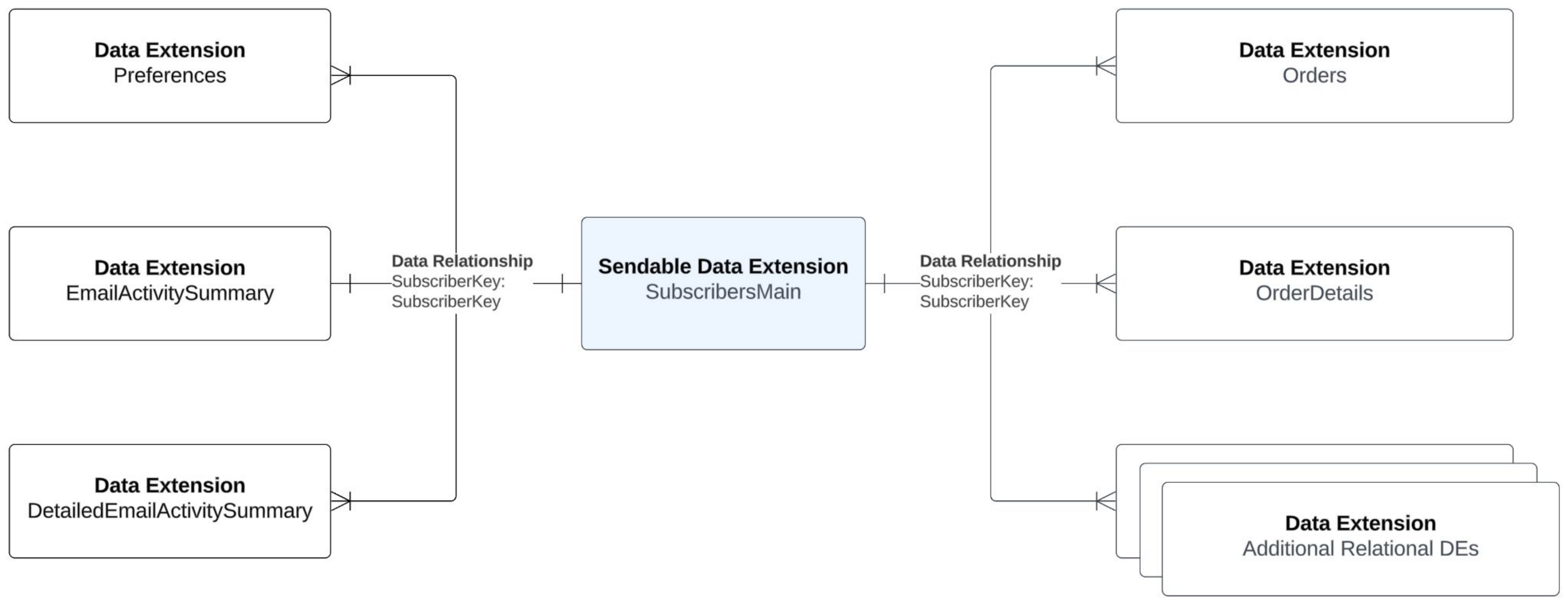
How day-to-day usage looks like


Is this a one time or recurring send?
One Time
Data Relationship
Drag and Drop filter off of Subscriber Main and related DEs to generate segment
Campaign Creation
Create accompanying email creative
Test, Validate, Schedule
Schedule the email for one time send
Recurring
Automation Studio
Queries or Saved Data Filters to refresh target DEs to generate segment
Campaign or Journey Creation
Create full journey series or one off send email creatives
High volume?
High
Automation Studio
Email Send Activity
Med / Low
Journey Builder
Multi or Single Step Journey

NOTE: Usage pattern and data update process identical to Core Data Model pattern, but with only 1 target DE to segment and drive campaigns out of.

Architecture - Relational Core Data Model

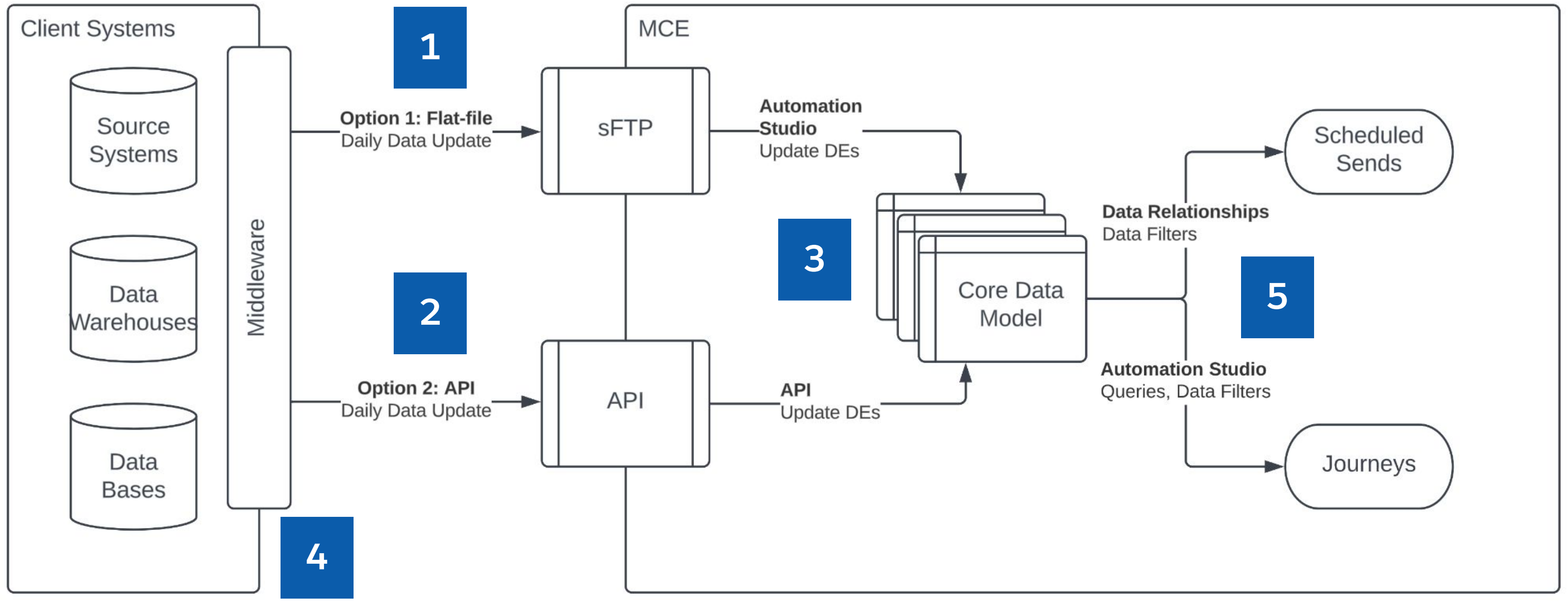
Overview



- The relational core data model principal centers around a single Sendable DE that has a distinct store of your main audience pool
- Each related DE is a store of a distinct category of data
- Each record relates to the main table on a common SubscriberKey / ContactKey; ideally some form of an enterprise ID / SFID
- The related DEs are structured to be only 1-layer deep to allow for drag and drop segmentation in Email Studio via Data Relationships (Audience Builder end-of-life'd)

Architecture - Relational Core Data Model

High-level Architecture



1. Through a combination of client systems, the most performant way is to update the core data model through flat-file drops + automation studio processing

2. Alternatively, API DE updates is sometimes (but rarely) used to update the DEs

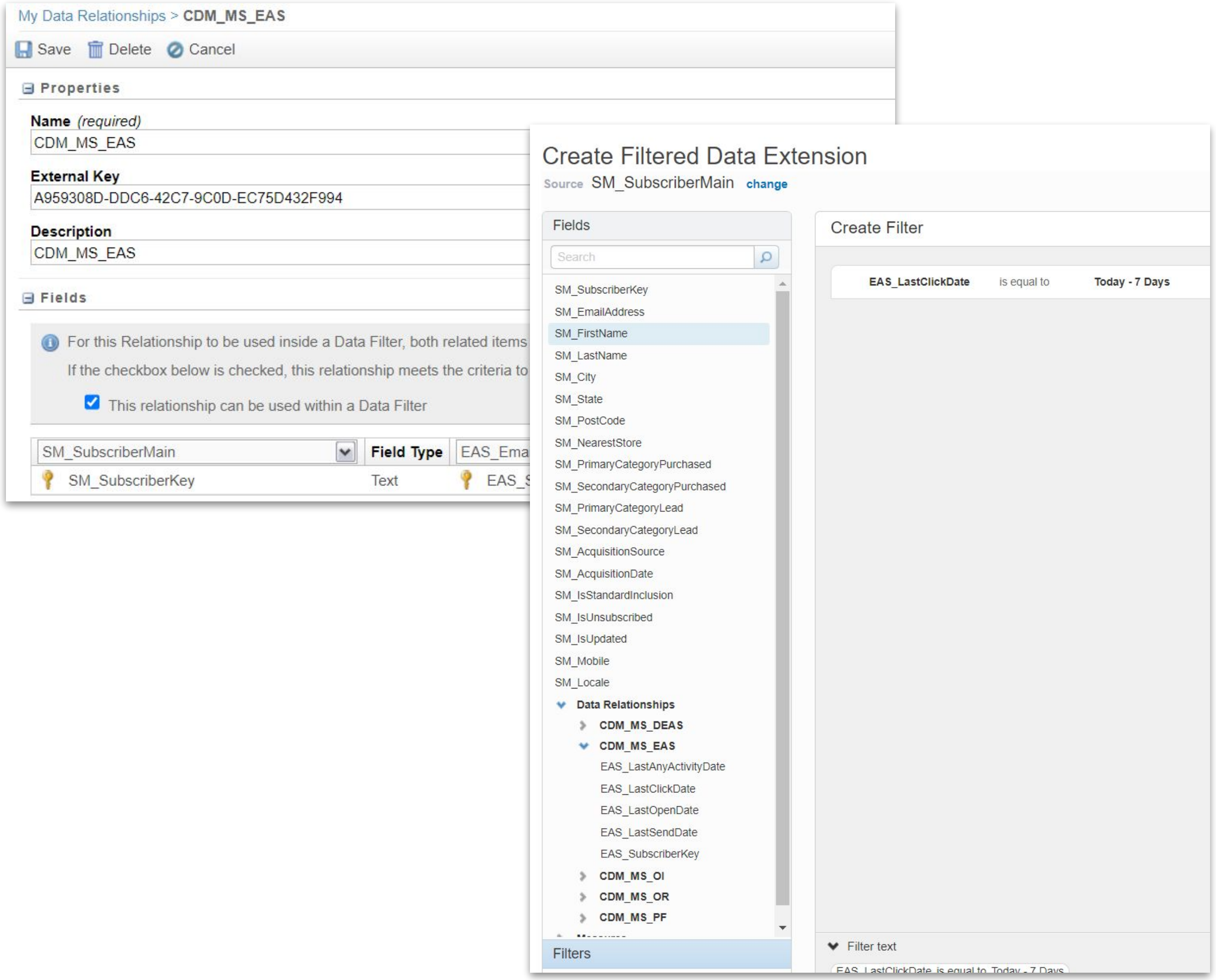
3. The daily updates is to ensure that the latest deltas come into the system to power the most up-to-date segmentations and journey functionalities

4. Typically, the client has tools that can natively support flat-file transfers or has middlewares to support this process; in the complete absence of options, tools like Mulesoft needs to be considered in facilitate the data transfer

5. Once the CDM is updating regularly, target segments can be created using Data Relationships (point-and-click) at an adhoc basis, or via Automation Studio Queries (scripts) and Data Filters at an ongoing basis

Architecture - Relational Core Data Model

Data Relationships



My Data Relationships > CDM_MS_EAS

Save | Delete | Cancel

Properties

Name *(required)*: CDM_MS_EAS

External Key: A959308D-DDC6-42C7-9C0D-EC75D432F994

Description: CDM_MS_EAS

Fields

For this Relationship to be used inside a Data Filter, both related items

If the checkbox below is checked, this relationship meets the criteria to

☑ This relationship can be used within a Data Filter

SM_SubscriberMain	Field Type	EAS_Ema
SM_SubscriberKey	Text	EAS_S

Create Filtered Data Extension

Source SM_SubscriberMain change

Fields

- SM_SubscriberKey
- SM_EmailAddress
- SM_FirstName
- SM_LastName
- SM_City
- SM_State
- SM_PostCode
- SM_NearestStore
- SM_PrimaryCategoryPurchased
- SM_SecondaryCategoryPurchased
- SM_PrimaryCategoryLead
- SM_SecondaryCategoryLead
- SM_AcquisitionSource
- SM_AcquisitionDate
- SM_IsStandardInclusion
- SM_IsUnsubscribed
- SM_IsUpdated
- SM_Mobile
- SM_Locale
- Data Relationships
  - CDM_MS_DEAS
  - CDM_MS_EAS
    - EAS_LastAnyActivityDate
    - EAS_LastClickDate
    - EAS_LastOpenDate
    - EAS_LastSendDate
    - EAS_SubscriberKey
  - CDM_MS_OI
  - CDM_MS_OR
  - CDM_MS_PF

Filters

Create Filter

EAS_LastClickDate is equal to Today - 7 Days

Filter text

EAS_LastClickDate is equal to Today - 7 Days

To allow for codeless segment output, the **only method** you have in an MCE only environment is to set up Data Relationships in Email Studio using the Core Data Model DEs.

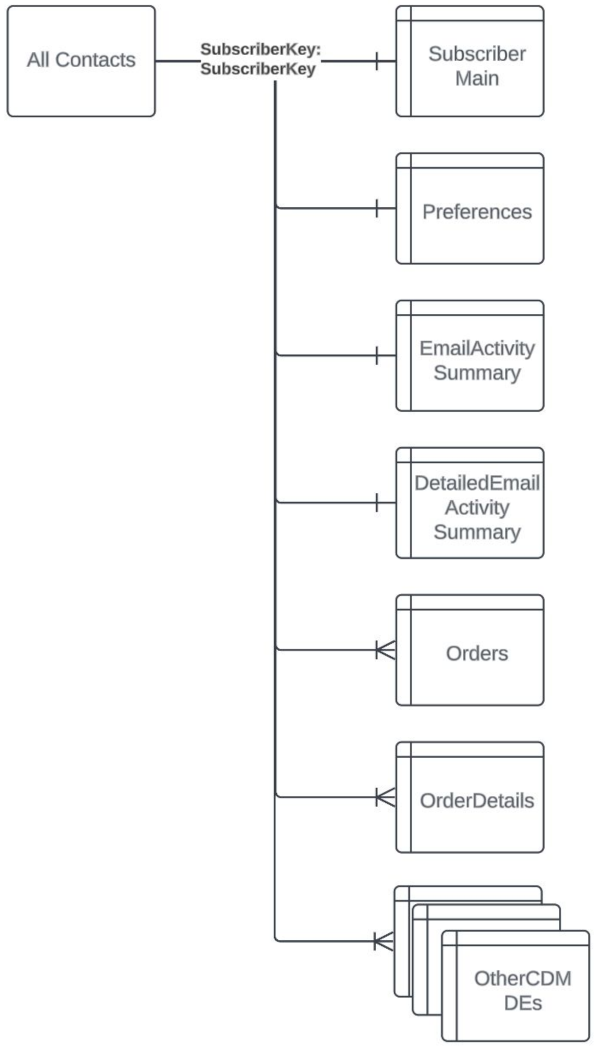
NOTE: This used to be supported by Audience Builder (discontinued), and now should be done in Data Cloud (if licensed).

- The “left” table will always be SubscriberMain, and the “right” table will be your related DEs
- Data Relationships work best when you have a clearly defined single field SubscriberKey as Primary Key across all tables
- Data Relationships works best when there is a main DE to hang off of (eg. Subscribers Main)

CAUTION: This function has not been updated for roughly 7 years. So, while stable, it handles complex multi outer joins incorrectly and is unlikely to get updated. Use with caution.

Architecture - Relational Core Data Model

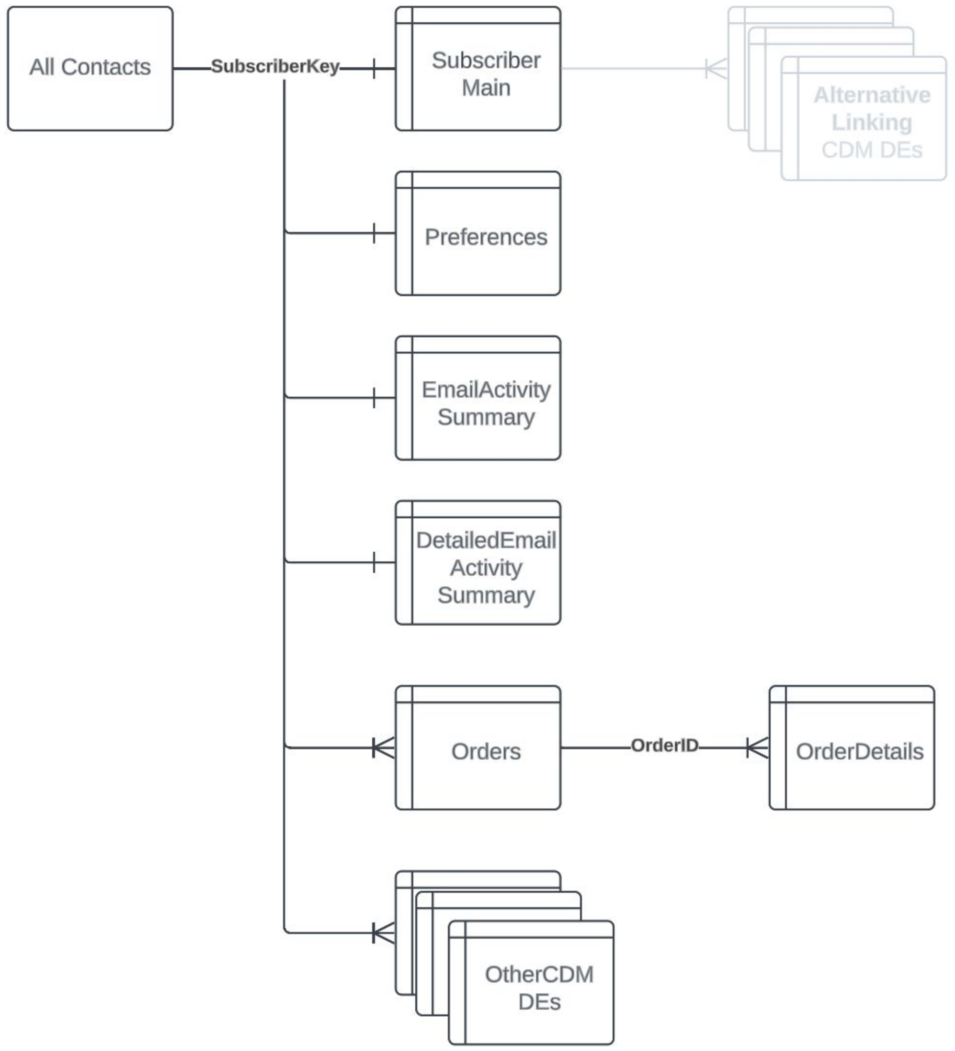
Data Relationships



- Data Relationships works only **one level deep**, unlike Data Designer
- It is the only codeless segmentation output method for out-of-box MCE

Architecture - Relational Core Data Model

Data Designer



CDM DEs also have to be configured in Data Designer in addition to Data Relationships to allow for cross channel data references in Journey Builder.

- Each DE in the CDM can be linked directly to All Contacts, ideally on common SubscriberKeys
- In Data Designer, it does support multi-layer relationships (eg. Orders > OrderDetails), so you sometimes want to use another key when connecting secondary / tertiary relationships
- In **Journey Builder**, regardless of how you hookup the relationships, it will only qualify it as 1:1 (ie. down to a single row); the cardinality is a legacy function for Audience Builder (which has been end of life'd)
- You can alternatively route all tables through SubscriberMain (see greyed out section), but just ensure that SubscriberMain is updated properly; missing record would break the link to all subsequent DEs

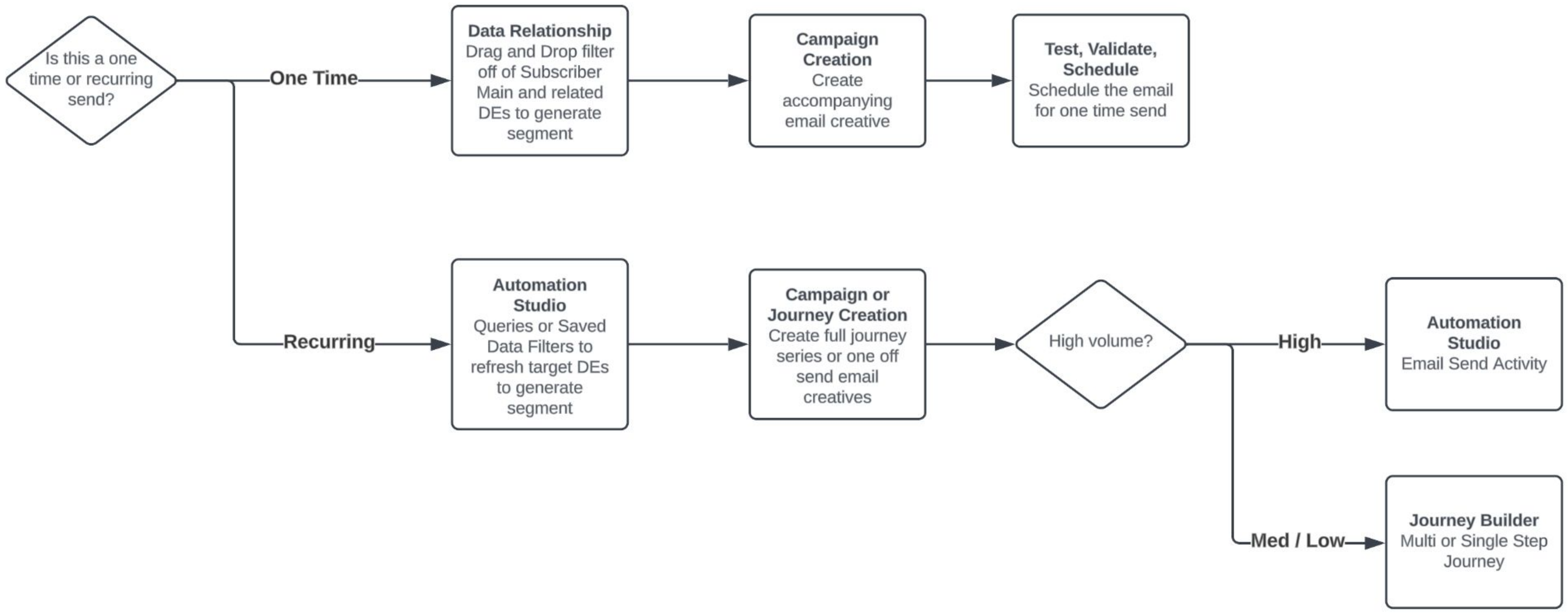
Architecture - Relational Core Data Model

List of Common DEs and Example Data Specifications

Data Extension	Description	Sample Specification
SubscriberMain	This is your primary DE that is unique on SubscriberKey / ContactKey. Each SubscriberKey is paired with all the main flattened customer attributes (eg. FirstName, LastName, EmailAddress, IsStandardInclusion). The paired attributes should also include all the typical fields you use for personalization.	https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=0
EmailActivitySummary	This is a DE that is unique on SubscriberKey / ContactKey, with the summary data of email activity. This would include LastSendDate, LastClickDate, and as needed, counts of time ranges (eg. ClicksLast90D). This allows email activity to be leveraged across Journeys for entry, decisions, goals and exits.	https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=1600745612
DetailedEmailActivitySummary	This is a DE that is unique on SubscriberKey / ContactKey AND Email Send (job ID + email name + etc), with the summary data of per email level activity. This allows cross journey decisions and suppressions (eg. do not enter Journey A if received email X from Journey B). This is a standard feature in competing tools that is not natively supported. IMPORTANT: Depending on send volume, a 90 day (or less) data retention policy should be put in place.	https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=1556473447
Preferences	Where there is a custom preference center, you'll want to include a DE that has the optins of each preference (not permission) and the last updated date. The population process of this DE may differ based on your preference center design.	https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=934321988
Additional Related DEs	Other related DEs such as Orders, Transactional Summaries, Events Registrations should be built as business requirements dictate. The main guiding principle is to separate out distinct buckets of data so an error in a single process does not cascade out to the whole model.	https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=2110069649 https://docs.google.com/spreadsheets/d/1DrWhJdLSIVEPOmbK4YLWI0-_CncngO3Dv8nQPtbMiEg/edit#gid=826996906

Usage Pattern - Relational Core Data Model

How day-to-day usage looks like


Is this a one time or recurring send?
One Time
Data Relationship Drag and Drop filter off of Subscriber Main and related DEs to generate segment
Campaign Creation Create accompanying email creative
Test, Validate, Schedule Schedule the email for one time send
Recurring
Automation Studio Queries or Saved Data Filters to refresh target DEs to generate segment
Campaign or Journey Creation Create full journey series or one off send email creatives
High volume?
High
Automation Studio Email Send Activity
Med / Low
Journey Builder Multi or Single Step Journey

FAQ - Relational Core Data Model

Common questions and topics

Why only one level deep in Data Relationships?

Data Relationship is an older feature, and multi-level linking is simply not supported. It also has issues doing complex outer joins across multiple DEs, so more complex segments should still be done in Queries.

Why even use Data Relationships?

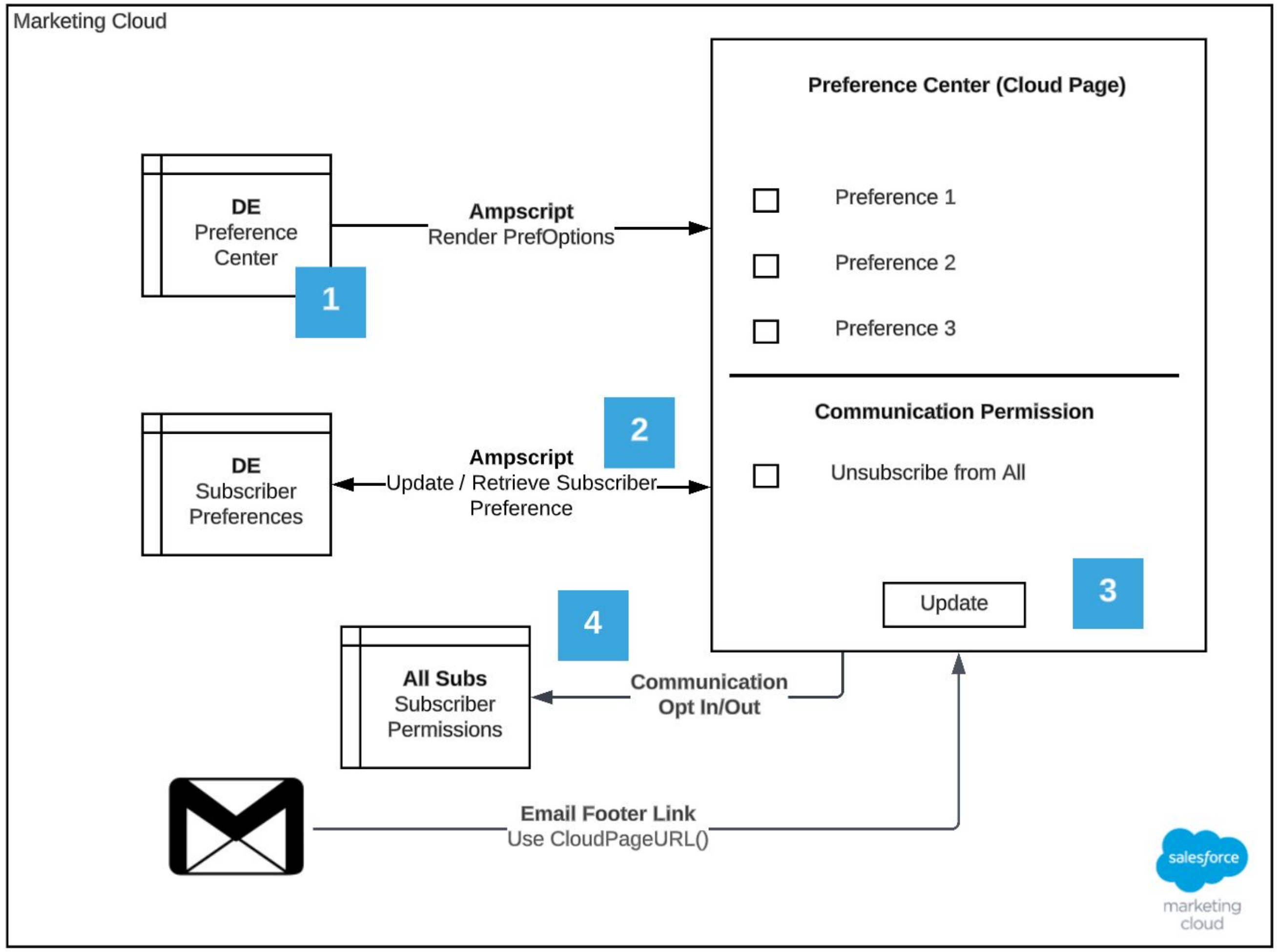
This is the only way to facilitate drag and drop segmentations without any additional SKUs or 3rd party tools. MCE would be a really hard sell to non-technical clients without this functionality.

Why don’t we use the API to feed the Core Data Model?

The API works fine but a lot of the functionalities still reside with the legacy SOAP API. The REST API is not fully featured in certain areas. Lastly, flat file feed is generally the most performant and fully featured.

Architecture - MCE Centric Custom Preference Center

Overview



Marketing Cloud
DE Preference Center
1
Ampscript Render PrefOptions
Preference Center (Cloud Page)
Preference 1
Preference 2
Preference 3
Communication Permission
Unsubscribe from All
Update
3
DE Subscriber Preferences
2
Ampscript Update / Retrieve Subscriber Preference
4
All Subs Subscriber Permissions
Communication Opt In/Out
Email Footer Link Use CloudPageURL()
salesforce marketing cloud

Custom Preference Centers are often needed for MCE only implementations due to the limitations of the out-of-box preference center. The design philosophy is to create a self-servable preference center and avoid hard coding values.

1. Custom Preference Center (CPC) reads the PreferenceCenter DE to render the preferences onto the page. Each row represents a preference option, and this can be added in Contact Builder by end users.

2. CPC reads the SubscriberPreferences DE matching on SubscriberKey to return the subscriber's existing preferences. In the absence of records, default opt in to preferences to allow for rendering.

3. On "Update" button submit, update the SubscriberPreferences DE. Records are created per preference along with OptinStatus and LastUpdatedDate.

4. Lastly, whenever an email unsub/resub occurs, the Preference Center should update All Subs directly as it is the source of truth system table for MCE.

Package Manager Reuse Link:
https://drive.google.com/file/d/16uN5Fo39BL35tsvl3dYety9RdKwDqdN3/view?usp=drive_link

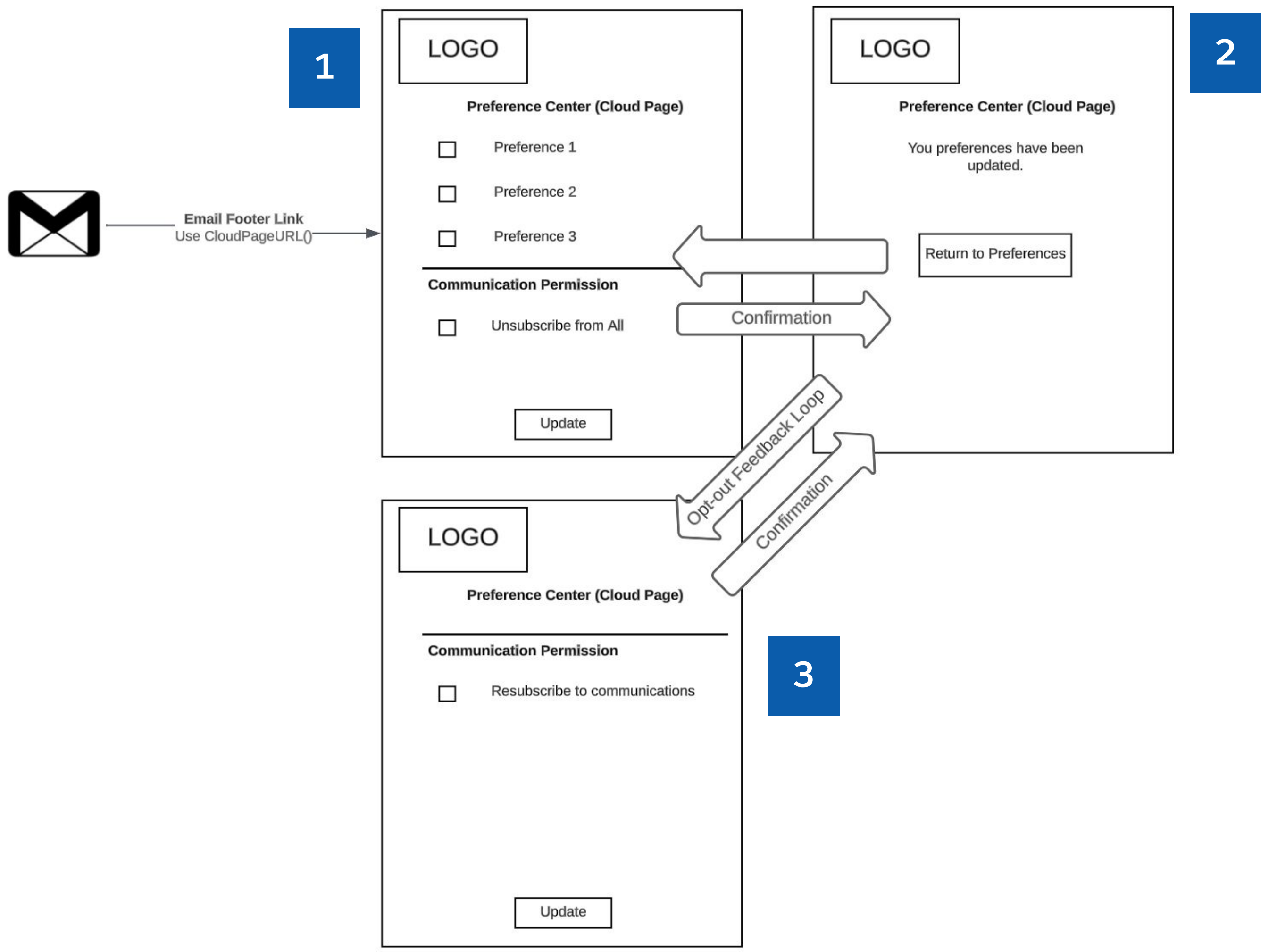
Architecture - MCE Centric Custom Preference Center

Data Extension Structures

Data Extension	**Description**	**Sample Specification**
PreferenceCenter	This is your DE that stores the options to be rendered, along with basic meta data like SortOrder, PreferenceDescriptions, IsActive. This is meant to be self-servable without code via adding rows in Contact Builder.	https://docs.google.com/spreadsheets/d/17gwVbIuZmm8sxVFy2sDF5PBPHIwJnWwni1CkOq9ZfTw/edit#gid=0
SubscriberPreferences	This is a DE that stores the Preferences of a subscriber upon accessing the custom preference center and updating their permissions.	https://docs.google.com/spreadsheets/d/17gwVbIuZmm8sxVFy2sDF5PBPHIwJnWwni1CkOq9ZfTw/edit#gid=699827839

Usage Pattern - MCE Centric Custom Preference Center

How day-to-day usage looks like



The Custom Preference Center should be accessed by customers at the email footer as a CloudPageURL() encrypted link to prevent PII exposure.

1. Customers would land on the CPC. They'd be introduced to the list of options as defined in the "Preference Center" configuration DE.
2. On "Update" button submit, a confirmation is presented along with a Return to Preferences feedback loop button.
3. If Unsubscribe from All is selected, Preferences are removed to avoid UI clutter until resubscription.
4. The customers preferences are then segmentable out of the "SubscriptionPreferences" DE for campaigns and journeys.

FAQ - MCE Centric Custom Preference Center

Common questions and topics

Why don’t we hardcode the preferences??

Inevitably, the client will eventually want to change the preferences. Typically, at that point, the client has forgotten context around the implementation, and this ends up coming back to us and incurs overhead costs that we don’t get to charge for.

Why is the SubscriptionPreferences DE structured vertically / without explicit columns for each preferences?

Structuring vertically with 1 row per user per preference ensures that the end user will never have to go in and update this table. If we had explicit columns, each time preferences are changed, the DE has to be updated, and it runs the risk of the entire preference center breaking due to human errors.

Why don’t we write this data to Sales / Service Cloud / CRM stack?

We can, and we should. However, it has different considerations that is covered the MCE to CRM section in this deck.

More Notable Mentions

Other Patterns and Tricks to Deploy, Good Resources

1. Email / Journey Activity Summaries
2. Dedicated Journey Exit Tables
3. New, consolidated “Platform Limits” Document Available Soon (link to come)

[Whiteboard]

FAQ - Flat Primary Table

Common questions and topics

How do we choose a flat vs relational model?

The flat model is seldomly used, but is quite useful for smaller clients / end users that have trouble grasping relational data models. As it's a single table to segment out of, it's quite easy to use in day-to-day tasks. However, a lot of data granularity is lost as you can't store 1:many data.

How "wide" should the flat DE be?

"Wide" refers to how many fields can you put into the single DE to segment and run campaigns off of. Generally, we wouldn't want to go more than 50 fields AND they should relatively small in string limit per field (ie. 50 - 254 chars, not 4000). Also depends on the total number of rows, which should be in the hundreds of thousands to low millions max. The general issue is that too much data will eventually cause the DE's performance to degrade significantly.

Can we create real-time data in the flat table method?

You "can", but it's not recommended. Real-time data writes into a single table while heavy nightly calculations are updating the DEs would typically lead to data loss, performance issues, among other problems. Recommended to separate real-time data into it's own dedicated DEs, then write them in with hourly automations.

Reviewing Use Cases

Round Table Discussion

Anyone with a Recent Scenario?

Open Discussion, or Past Use Case Review as Applicable

Simple Exercise

- Review any recent use cases
- I can pull up some past use cases to review, which applying our concepts here (ALDO, SurveyMonkey / Momentive, etc)

Q&A

Any questions?