STEP
Trailblazer

Solution Enablement

CUSTOMER MASTER DATA MANAGEMENT

STEP 9.0

StiboSystems

Table of Contents

Customer Master Data Management Solution Enablement

Customer MDM (CMDM) is a technology enabled discipline which is required to create and maintain an accurate, timely, complete, and unified view of a company or customer. CMDM will consolidate, cleanse, enhance, and govern customer data from disparate siloed systems enabling organization to identify, consolidate, and link customer data across heterogeneous data sources in a single, accessible customer view.

To use the solution implementation, please see:

- CMDM Solution Overview
- CMDM Solution Implementation Guidelines

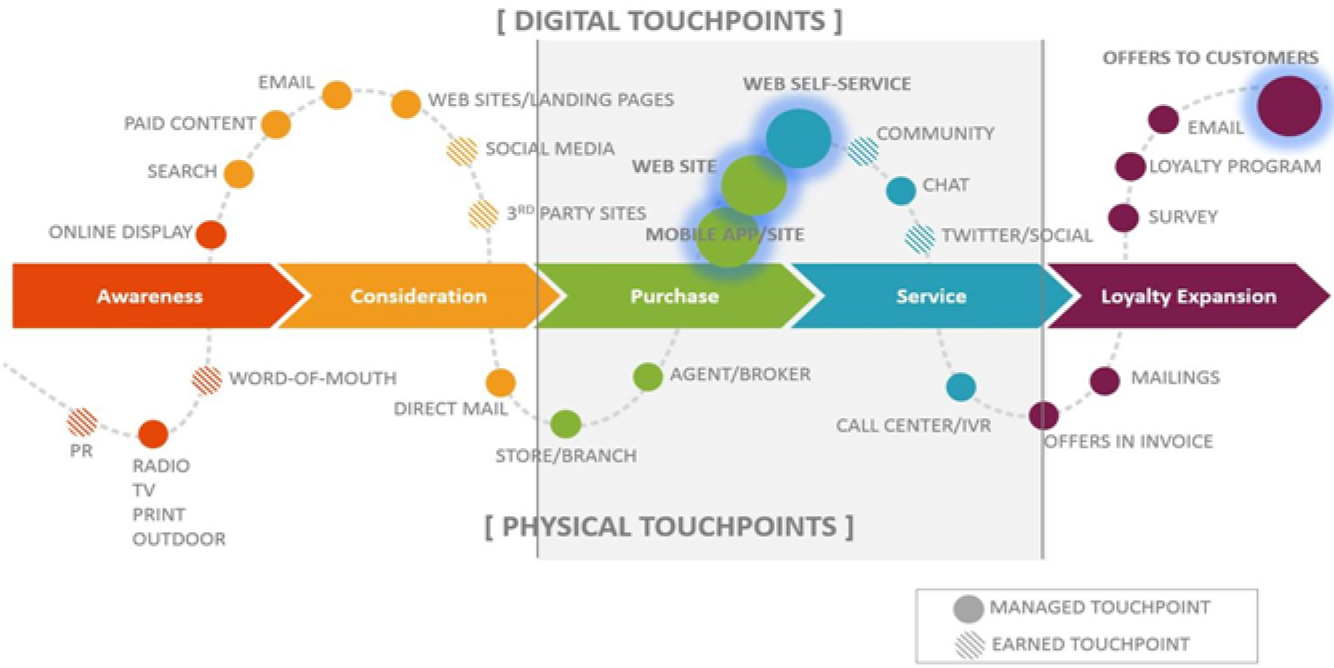
Customer Master Data Management Overview

The Customer Master Data Management solution provides organizations with the means to address complex customer data challenges and improve their overall customer experience.

Improving Customer Experience

Customer Experience is the holistic journey of a customer that begins the moment they first become aware of a product and continues until the customer is building their own loyalty platform with a product.

This documentation is focused on providing guidance to organizations in how to improve their customer data so that they can make informed decisions, and thus improve their customer experience.



Customer Data Management Challenges

Organizations are faced with four primary challenges when managing customer data:

- Data Quality
- Operational Inefficiencies
- Reactive Strategic Decisions
- Governance

Data Quality Challenges

Data quality issues typically arise when attempting to house and manage customer data across multiple platforms. Doing so often times leads to inaccurate, duplicated, and incomplete data.

Operational Inefficiencies

Operational inefficiencies can be characterized by an organization's inability to share accurate and up-to-date data across the business, severely impacting downstream systems, processes, and initiatives. For example, some of these inefficiencies can lead to: returned mail, increased costs, payment delays, and unacceptable response time.

Reactive Strategy Decisions

Many organizations struggle to achieve their Customer Relationship Management (CRM) vision. Considering its impact on an organization's success, a concrete CRM strategy is a high concern for business and IT leaders.

Bad data is the most common reason many CRM initiatives fail.

Governance

The most important part of data governance is to consider the availability, usability, integrity, and security of data within an organization. Poor data governance can lead to financial, reputational, and legal risks.

Implementation Methodologies

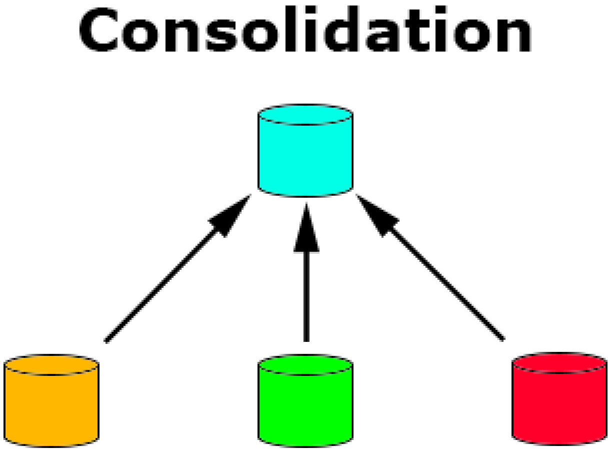
CMDM comprises four paths to implementation. These paths are:

- Consolidation Implementation
- Coexistence Implementation
- Registry Implementation
- Centralized Implementation

Consolidation Implementation

This methodology refers to an implementation where STEP is a hub for externally provided data that is funneled into single Golden Records, which are then made available to external systems. These records are created via matching and merging operations with validation checks throughout the process. The STEP data created from this process is read-only, which is often a necessity as the data originates outside of STEP in an upstream system. A consolidation style implementation does not synchronize data back to the contributing systems.

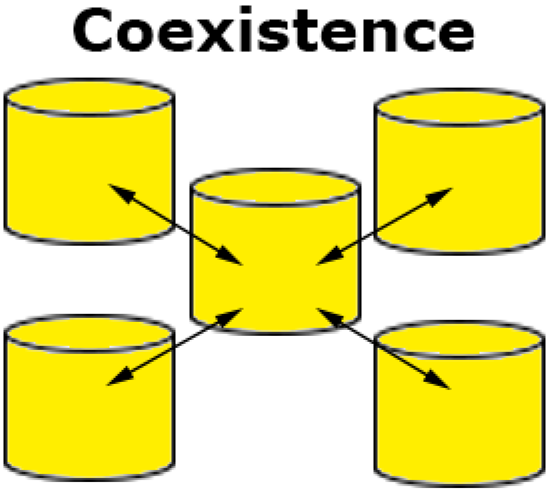
For example, a company wishes to combine all of their sales channel data into one record bank so that data analytics may provide meaningful information. Once all these records are consolidated, marketing initiatives can use this information to target different channels as well as different regions.



Coexistence Implementation

This methodology refers to a scenario where multiple databases containing the master data must operate at the same time. The customer data is authored and stored in various external sources while being synchronized to STEP. This process involves deduplication, conflict resolution, and validation operations, resulting in the creation of golden records. Finally, updates to the data can be done in the source system and any external systems that master customer data. STEP synchronizes the content of golden records back to the source systems, keeping every source up to date with data contributed from any source. This allows for a hybrid where DTA authoring may happen in the source system as well as in the MDM system.

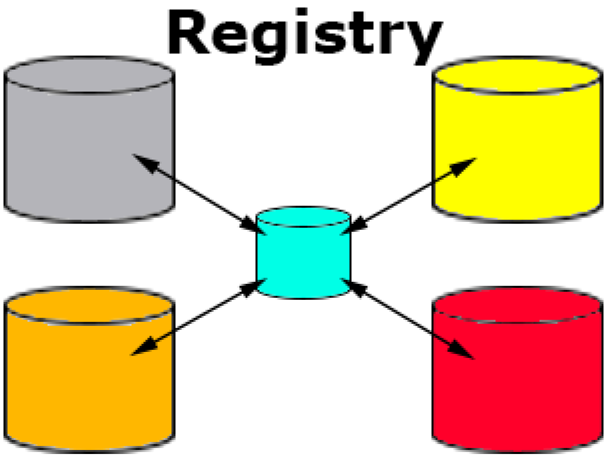
A common situation for coexistence is in companies where each department has their own customer database. For example, a service department that has data on customers is stored in a separate database than where customer data is stored for the sales and marketing departments. However, using a CMDM solution with the Coexistence model, if information is changed in the marketing department, it is changed in all other departments.



Registry Implementation

The registry method uses a simple database to reconcile identifiers for all customers across the enterprise. This method allows STEP to serve as a referenced, read-only source of mastered customer data for external systems with minimal data redundancy. STEP will only match on the bare minimum of needed data to confirm a unique record. All externally maintained data quality is the responsibility of the source systems.

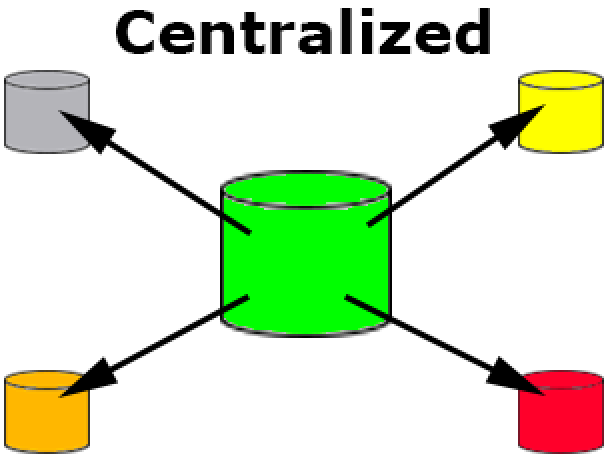
For example, a company maintains different types of customer information across various external systems. A registry is used to identify and associate these different types of customer data with external systems. STEP serves as this registry by storing unique records with identifiers that are different from one external system to the next. STEP will ONLY store these records and unique record identifiers, and deduplicate their contact information.



Centralized Implementation

The centralized implementation is based around a well-managed and governed central repository for all master data. This repository holds a set of Golden Records that are read-only by the operational and analytical systems throughout the enterprise. STEP is the centralized owner of customer data, serving as the system of record. This system of record is an organization’s single version of truth for customer data. Data is not only centralized in STEP but also created and stored there. The most important aspect of a centralized implementation is that it is a system of record instead of a system of reference.

For example, a company wants to use STEP as the system of record for onboarding, deduplicating, and enriching new suppliers. Over time, STEP could remove duplicate records of these suppliers, and then the golden supplier records could be provided to external systems.



This implementation guidelines focuses with a case that implements data with the **consolidation** and the **coexistence** methodologies.

Customer Master Data Management Implementation Guidelines

The Customer Master Data Management Implementation Guidelines provide organizations with baseline configurations and sample data to help get their STEP Systems off the ground and guide them in the creation of their Customer MDM solution.

This user guide offers a number of resources that can help achieve this goal:

- A generic Customer MDM use case that accounts for both B2C and B2B solutions, and provides context for system configuration and solution design. For more information, see the **Use Case Overview** section of this user guide.
- A series of component-specific use cases that tie into the generic Customer MDM use case mentioned above. For more information, see the **Use Case Appendix** section of this user guide.
- A comprehensive guide for how data flows in and out of the STEP system, including Matching. For more information, see the **CMDM Data Flow** section and the **Matching** section of this user guide.
- A comprehensive guide for modeling the data in STEP. For more information, see the **Data Modeling** section of this user guide.
- A comprehensive guide for data stewardship. For more information, see the **Data Stewardship** section of this user guide.
- For the configuration of a base CMDM solution and how to deploy it, see the **CMDM Configuration Guide** section of this user guide.

Important: This guide references supporting configuration documentation that exists in other sections of the STEP Online Help.

Use Case Overview

To illustrate how the CMDM solution can address common challenges an organization faces, the following use case presents a fictional company with typical business considerations and problems. Understanding this use case example will provide additional context to the configuration steps included in this guide, as well as the more granular use cases that concern specific CMDM components.

For more information on component-specific use cases, see the **Use Case Appendix** section of this guide.

ACME Company's Business to Customer Use Case

The ACME Company is a large clothing brand which markets and sells largely to North America. ACME sells directly to their consumers via various channels. As the company has grown, issues have developed due to poor data quality. Because of this, customer contact information is rarely in sync across business units. Furthermore, all customer shopping patterns are inaccurate because the data is not distributed and updated evenly. This asyncrhonization causes a missed opportunity to target customers for the best sales, promotions, and loyalty programs. Finally, due to these gaps in the data, the customer experience is quite poor as actions performed in one department by a customer are not known in another.

During an internal evaluation of the IT infrastructure, it was revealed that the customer data is owned and maintained by separate and distinct source systems. This setup causes data to not be updated across the enterprise, creating a large amount of conflicting duplicate customer records. Additionally, it was concluded that a lack of data governance has made it impossible to determine what degree of overlap and duplication is present within the customer database.

To improve customer data quality, ACME has targeted a few areas where data quality becomes less consistent:

- ACME's e-commerce system processing customers who self-register on the company website.
- Customers placing phone orders to a call center representative who then enters their data into ACME's ERP system.
- Customers signing up for the loyalty platform.
- Marketing uses customer data residing in ACME's CRM system for seasonal marketing campaigns.
- ACME uses loyalty points which can be redeemed for discounts and contain a monetary value. The financial data is owned by the CRM system which stores additional financial information.

Considerations

- North America is the primary market.
- ACME sells directly to customers via various channels.
- Multiple different source systems contain customer data.
- Currently, the customer data violates ACME's own data policies.

Difficulties and Problems

- Contact information not in sync across business units.
- Customer shopping patterns are of poor quality.

- Very poor user experience due to out of sync data across the business units.
- There is an excessive amount of duplicated customer records across the business.

ACME Company's Business to Business Use Case

ACME Company is a major clothing brand that not only sells directly to individual consumers, but also to both major and independent retailers and distributors across North America. To manage these various retailers, North America is divided up into several sales territories which are each managed by their own sales managers. Each sales territory contains several retailers, and information for each territory is managed by different instances of CRM. Since the same retailers may operate out of various different states or sales territories, ACME Company requires CMDM to not only deduplicate retail entities but also provide a trusted source of information for various source systems.

ACME also tracks and maintains contact persons' information for each retailer. These contact persons are designated sales representatives, account owners at each retailer, and purchasers in the procurement departments. The CMDM system must be able to account for contact persons that move from retailer to retailer, or those that get duplicated within the same organization, to ensure that accurate information is maintained.

Sales hierarchies are used to organize the customer portfolio in the sales staff and to aggregate bonus schemes in the sales staff, based on the revenue from each customer.

Considerations

- ACME sells to both major and independent retailers and distributors as well as to consumers.
- The company mostly services clients in North America.
- The North American region is divided into several sales territories with sales managers responsible for each territory.
- Contact persons are maintained in CRM as well as the ERP applications.

Difficulties and Problems

- There is an excessive amount of duplicate customer records.

Data Management Improvement Plan

The ACME Company has strategically planned the evolution of how their customer data will mature over the course of the next few years. To accomplish this long-term goal, an enterprise strategy plan is laid out in three phases, with each phase progressively building on top of the previous one.

Phase 1 – ACME would like to ensure their customer data is governed and can be trusted across the entire organization:

- Use MDM technologies to achieve a single, trusted customer view which is accurate, complete, and up to date.
- Establish baselines and improve analytical reporting accuracy for a better CRM and e-commerce experience.

- Identify customers who live under the same household for more efficient mail distribution and for better profiling of the customer.
- Reduce operational overhead cost by decreasing the amount of returned mail due to inaccurate mailing addresses.

Phase 2 – ACME would like to combine core customer master data with transactions, web site activity and social profiles:

- Enhance the customer experience by providing a personalized shopping experience.
- Utilize purchasing behavior analytics for increasing cross sell / up sell opportunities.
- Discover hidden associates and relationships to gather context for segmentation, targeted marketing, risk management, and regulatory reporting.

Phase 3 – ACME would like to further enhance the personalized shopping experience:

- Create an enterprise-wide single view of customers to serve all operational systems and use big data to generate insight and foresight based on advanced learning models.
- Contextualized customer data is then used to enhance customer experience through machine learning as well as digitally automating key business processes.

Use Case Appendix

The following use cases address a number of common challenges organizations face when handling customer data, and how they are solved by a CMDM solution.

Business to Customer Use Cases

- Merge Individual Customer Records via Automatic, Manual, or Advanced merging methods
- Rejecting Potential Duplicates (during Clerical Review)
- Customer Marriage and Relocation
- Trigger Clerical Review after Manually Editing Customer Record
- Address Validation

Business to Business Use Cases

- Identifying and Deduplicating Contacts Within the Same Organization and Handling Contacts across Different Organizations

Use Cases - Merging Individual Records

In these use cases the data steward is responsible for overseeing the deduplication of customer records via different merge processes.

Auto-Merge Individual Records

ACME has several external source systems which contain multiple records with separate identifiers (source record IDs). Despite these different identifiers, many of these records are in fact the same individual but with minor variations in demographic data.

These individual records are sent to STEP and automatically deduplicated via the configured matching logic, which identifies and merges identical records. These matched records are then merged and a single golden record is created.

Original source systems and respective source record IDs are maintained as part of the golden record for source traceability purposes.

Merge Individual Records

ACME has a number of records which are considered potential duplicates of one another, but the system cannot ultimately determine whether or not they are true duplicates without the input of a data steward. In these cases, a data steward must review potential duplicate records and corresponding demographics to intelligently determine whether they are a match or not.

Once STEP determines which records are potential duplicates of one another, a Clerical Review task list is presented to the data steward in the Web UI. Data stewards may review the demographics of each task to determine whether the records should be merged. If it is determined that the tasks are a match then the **Merge** action will merge the records into a single golden record. If it is determined that the records are not matches of each other then the **Reject** action will leave the records unmerged.

For matched records, a single resulting golden record will survive that contains the source system and source record ID of all records that were merged. For non-matched records, a Confirmed Non-Duplicate reference will be established to prevent future clerical review tasks from being created for the same records. The existing task will also be removed from the Clerical Review task list / workflow.

Advanced Merge Individual Records

As an alternative to the **Merge** function described in the above use case, data stewards can use the **Advanced Merge** option to manually dictate which demographics survive the merge between two or more potential duplicates. In addition they can choose to exclude individual records from the merge.

Once STEP determines which records are potential duplicates of one another, a Clerical Review task list is presented to the data steward in the Web UI. Data stewards may select a task, and by clicking **Advanced Merge**, they are presented with a table view of the associating records as well as corresponding demographics. As the data steward selects which demographics will survive the merge and be promoted to the golden record, a preview column displays the resulting merged golden record. Additionally, should the data steward determine that one or more records should not be considered as a potential duplicate, the option to **Exclude Record** is provided.

The resulting golden record reflects the demographics manually selected by the data steward. The task is removed from the Clerical Review task list once the merge is complete.

Use Cases - getSimilarObjects Acquisitions

Benny, from the sales team, is responsible for retailers and distributors that have contracts to resell ACME's products.

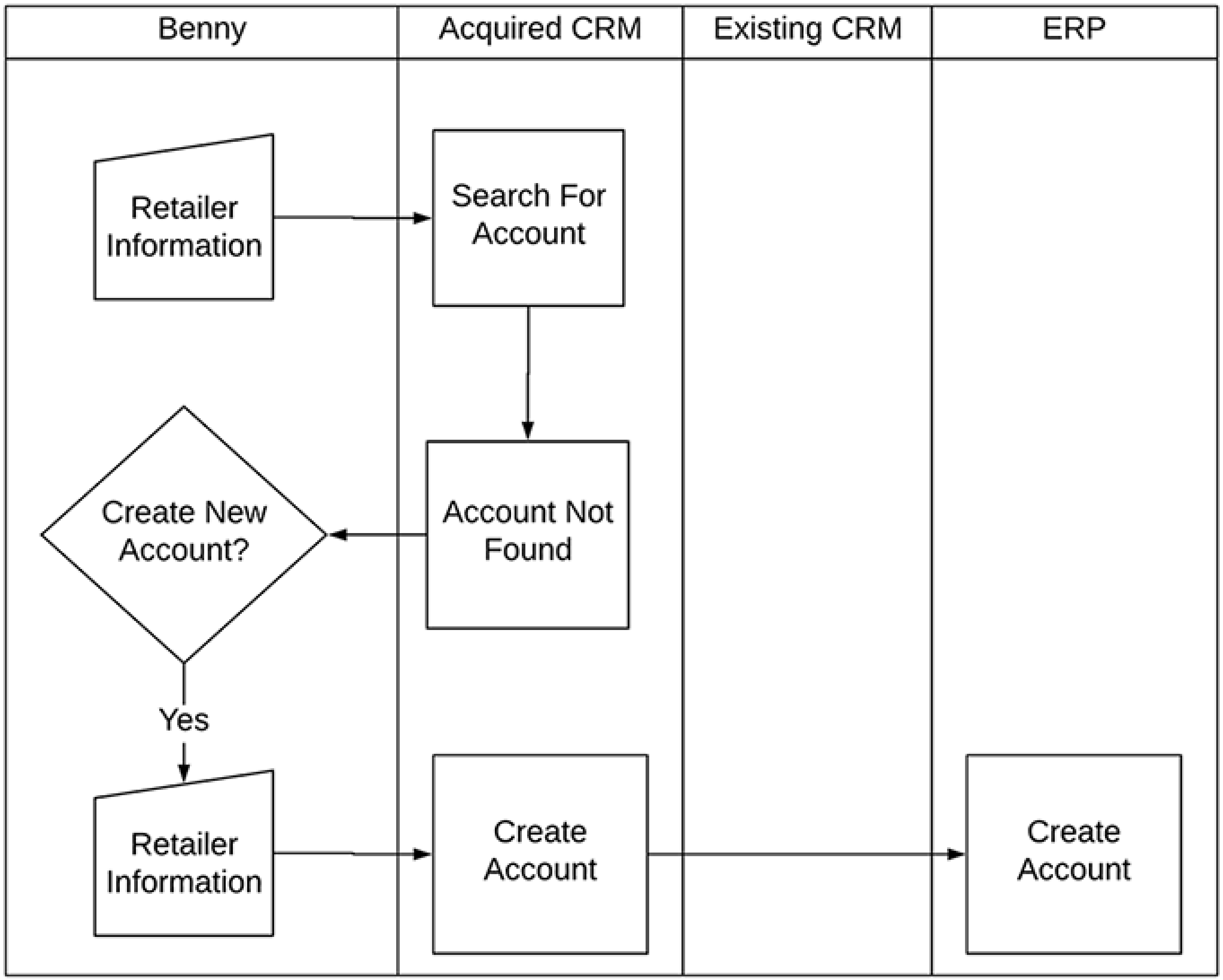
Benny needs to be able to check if a retailer or distributor already exists because ACME recently acquired a new company with a separate CRM.

Benny doesn't have clear visibility into other systems. He is unable to identify the record since the record does not exist in his CRM. Additionally, the other company's CRM cannot identify the record based on a fuzzy match if Benny cannot provide information for an exact match like a unique identifier.

Benny has difficulty merging records in the ERP system once they've already been created because they may have orders associated with them. It is also common that existing records from another system contain information that is valuable to Benny, in the process where he is creating the new record. i.e. the customer may have made a purchase recently, have an open support issue, or in the process of paying debts. It is therefore of high value to know of the existing record as well as to prevent creation of duplicate records.

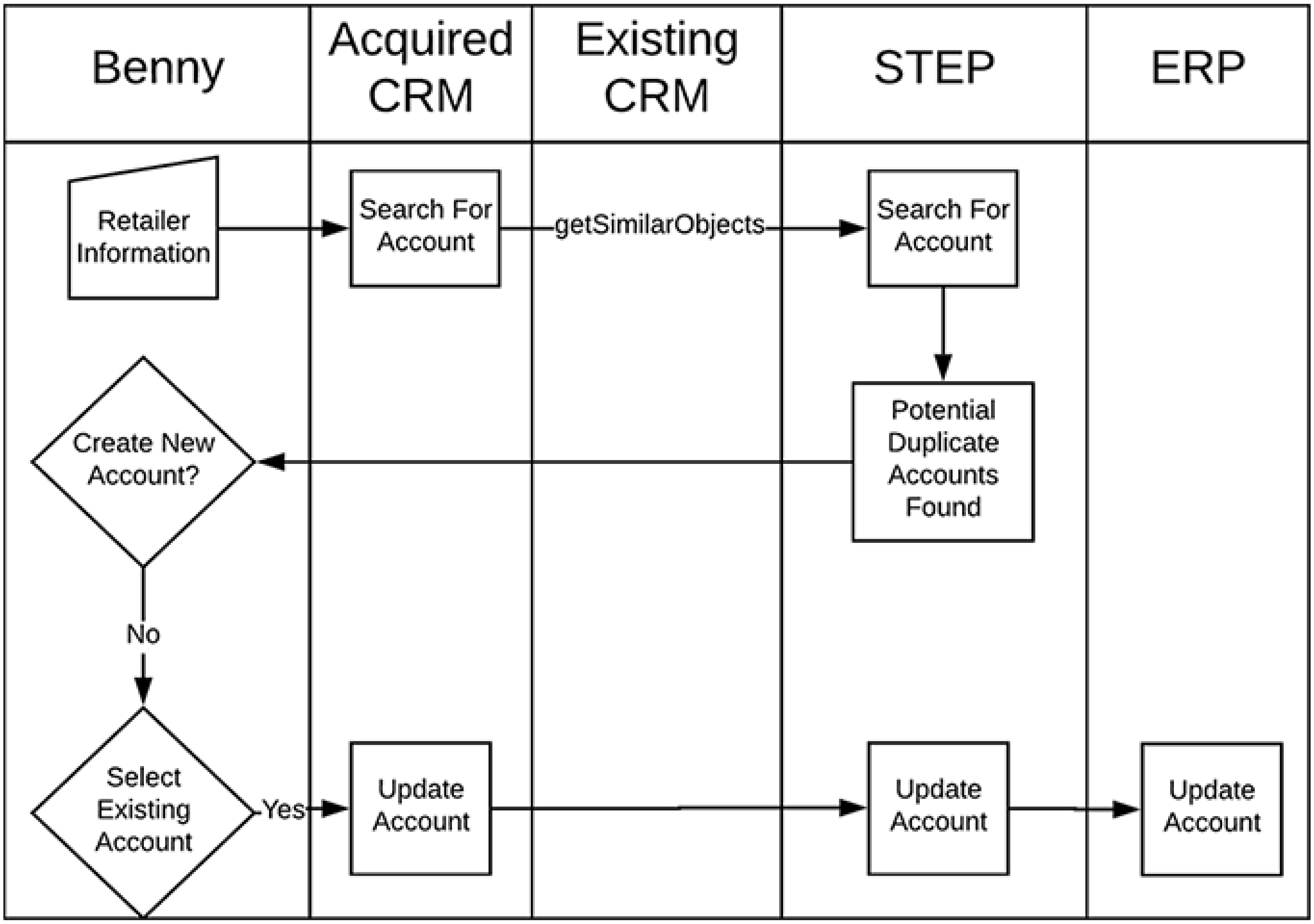
Benny provides a poor customer experience, due to his lack of knowledge of other engagements with the same customer. Additionally, he is increasing the risk of poor customer experiences in the future as he causes multiple copies of the same account with potential inconsistent data and increased complexity in merging the records at a later time.

Current Duplicate Search



With the current system landscape, Benny has a difficult time determining if the retailer already exists. This leads Benny to create a duplicate retailer in his ERP.

To-Be Duplicate Search



With a CMDM solution in place, all data is fed into STEP. This allows other systems to make getSimilarObjects calls to STEP to determine if potential duplicates already exist. Benny can then make an informed decision to either update an existing record accordingly or create a new record.

Use Cases - Marriage and Relocation

In these use cases, the data steward is responsible for overseeing customer record updates in the event that two customers get married or relocate.

Marriage

ACME must update their database in the event that two customers get married. A pre-existing customer named Catherine Yu got married to another pre-existing customer, Charles Johnson, and had her last name and address changed.

In this scenario, the change in status is initially updated in a source system by a customer service representative within a call center. The source system feeds this update to STEP, where it is then matched to an existing record via the source system record ID for Catherine Yu.

Once matched and updated, Catherine Yu is now Catherine Johnson and has a new address. Additionally, the household is updated to reflect that Catherine and Charles belong to the same household.

Relocation

ACME must update their database in the event that a customer moves to another address. In this scenario, Maxie Hadley was a pre-existing customer who moved from KY to UT. At some point a second record was created with the new address. STEP automatically identifies that the two records with different address are the same person.

Once flagged as potential duplicates, a data steward can handle the deduplication task via a clerical review. Concluding that they are in fact the same person, the data steward merges the two records into a single golden record.

Use Cases - Trigger Clerical Review after Manually Editing Customer Record

Sometimes ACME requires customer records to be manually edited in STEP to address errors in data quality. In this scenario, STEP identifies a new potential duplicate after the customer record for Kimberley Kaine had the spelling of her first name corrected. Once identified as a potential duplicate, a task is created in a clerical review workflow.

The data steward concludes that they are duplicate records and chooses to merge them into a single golden record.

Use Cases - getSimilarObjects Preventing Duplicates

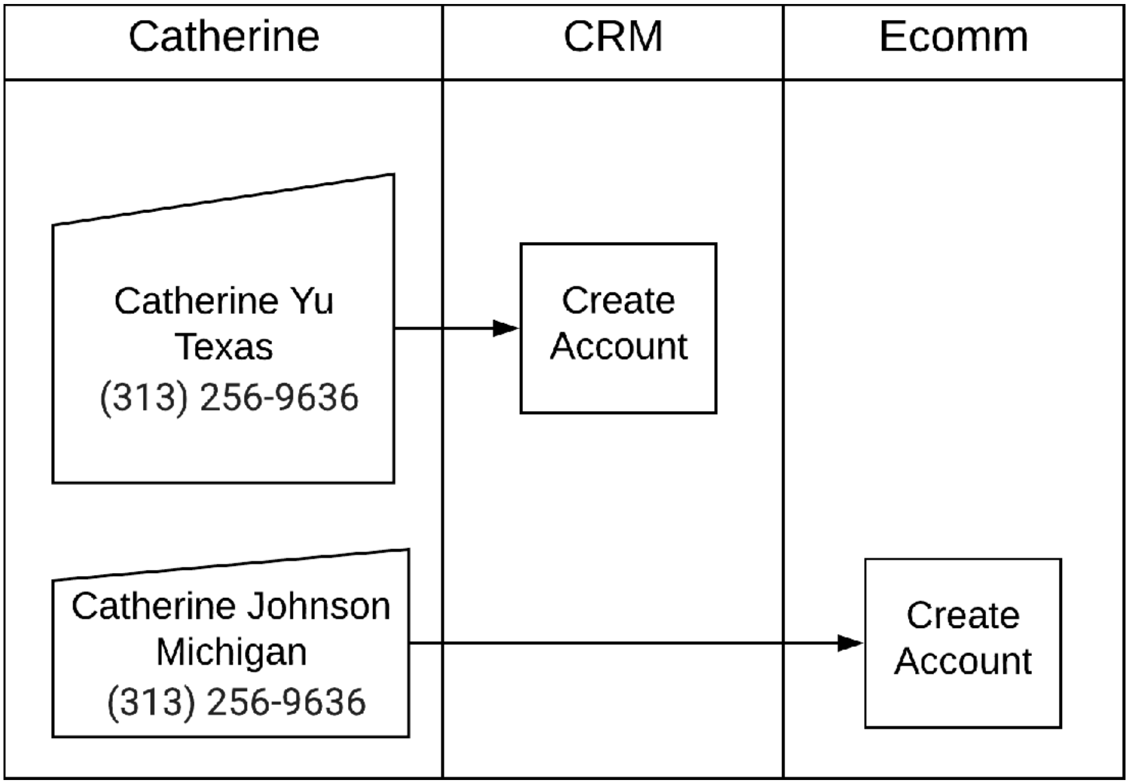
Catherine Yu signed up in-store for an ACME loyalty account to take advantage of an in-store promotion. Catherine is engaged and living in Texas. She is across the country from her fiancé who is in Michigan. Catherine signed up for the loyalty program using her maiden name, her Texas address, and her lifelong cell phone number.

After she's married, Catherine changed her name and moved to Michigan with her new husband. She goes shopping online and forgets she had signed up for the loyalty program and attempts to sign up again. Catherine uses her new last name, her Michigan address, and her lifelong cell phone number to sign up.

Without the exact same data, it is impossible for the CRM to find Catherine's loyalty account and previous purchases.

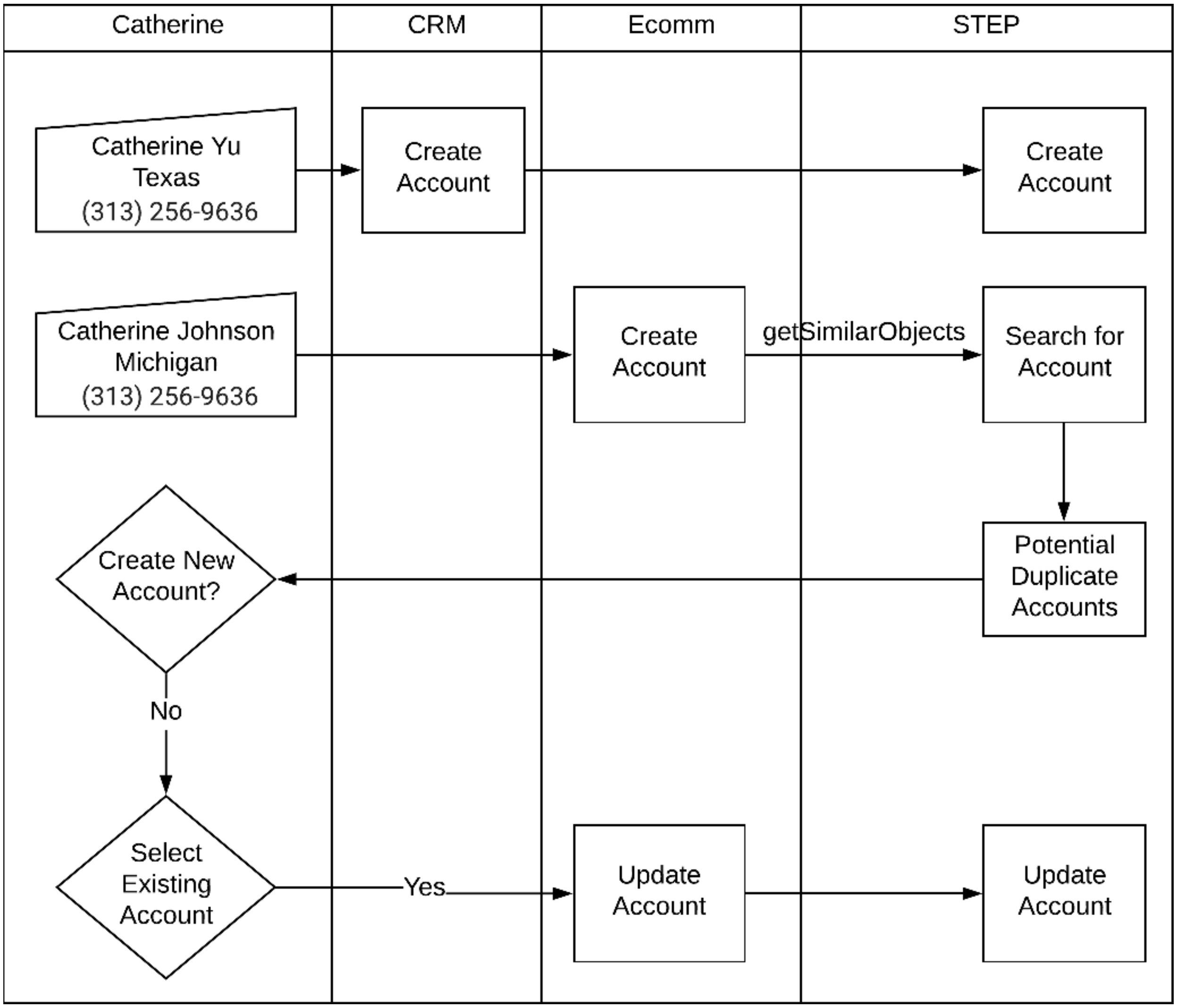
Catherine creates a duplicate loyalty account. Flyers and promotions are now being sent to two addresses. ACME missed cross sell and upsell opportunities because disjointed records make it impossible to identify Catherine's purchase history.

Current Duplicate Search



With the current system landscape, Catherine is unable to see that a loyalty account exists under her maiden name. ACME has a disjointed view of who Catherine is and the products she’s purchased.

To-Be Duplicate Search



The CMDM system can identify Catherine's existing loyalty account for other systems through the getSimilarObjects call. Catherine can identify and update her existing loyalty account preventing a duplicate loyalty account from being created.

Use Cases - Address Validation

ACME requires that address be validated via Loqate whenever a customer record is updated. In this scenario, Jason Kirby has moved to a new address. To account for this change, a data steward navigates to the customer record and manually updates the data. A Loqate request is sent, and new standardized data is returned based on the Jason Kirby's new address.

Typically, this kind of update would come through via a source system, in which case it would have already been validated upon import into STEP. In this scenario, there are no actions required by a data steward.

Use Cases - Rejecting Potential Duplicates

ACME may want to keep two matched records as separate records under certain circumstances. In such a scenario, a data steward can determine if two or more records in a clerical review workflow should merge or stay as separate entities. As an example, Jonathan Dorthy & Jenkin Doorthy are different people but were flagged as potential duplicates due to their similar names.

These potential duplicates are displayed via a clerical review task list in Web UI, where the data steward can reject the task to keep Jonathan Dorthy & Jenkin Doorthy from merging. Once rejected, the customer records remain as separate records.

Use Cases - Managing B2B Contacts

In these use cases, the data steward is responsible for managing organizational contacts (i.e. for vendors, distributors, etc.)

Identifying and Deduplicating Contacts Within the Same Organization

ACME must ensure that there are no duplicate contact records within the same customer organization. In this scenario, Sergio Bennett and Serge Bennet belong to the same organization and are identified as potential duplicates.

A source system sends a second record for Sergio Bennett with a different source record ID. STEP evaluates the incoming record and determines that it is in fact the same contact person for that specific vendor. Once STEP has determined that Sergio Bennett and Serge Bennet are one and the same, a data steward can merge them into one record via clerical review.

Handling Contacts Across Different Organizations

Emma Hoffman was previously a contact person for one of their suppliers, but now, works for another one ACME is associated with. Since the organizations are different, when the source system updates STEP with the new organization contact, Emma Hoffman's contact information will not be deduplicated across both organizations.

A new golden record contact for Emma Hoffman is created for that supplier and it is not deduplicated against her previously created golden record.

CMDM Data Flow

Having customer data authored and stored across separate source systems creates a barrier for achieving a unified and trusted view of customer data. When data has multiple sources, it can become inconsistent, inaccurate, outdated, and cause conflicts.

ACME Company has multiple source systems from which customer data originates. The flow of customer data for ACME Company begins at these points of entry, each of which is associated with a variety of channels. These channels include: the company website, phone direct, in-store, etc. (see Figure 1 below). ACME contains three source systems (Figure 2):

- E-commerce system for customers who place their orders and self-registers via the company's website.
- ERP system for customers who place orders by phone via ACME Company's call center.
- CRM system for driving seasonal marketing campaigns and financial data, which drives in-house financing as well as managing customer loyalty programs.

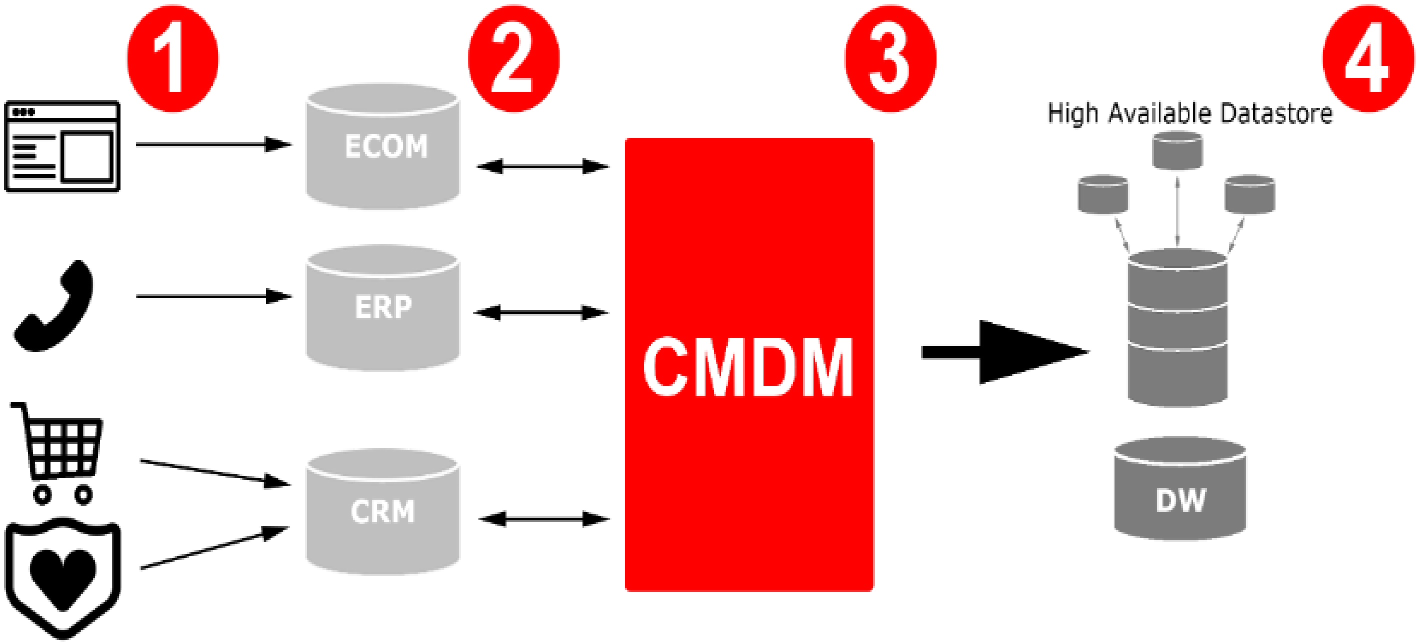
To understand how data flows into STEP and between applications, refer to the two implementation methodologies: consolidation and coexistence. For information on how the different implementation methodologies function, see the **Customer Master Data Management Overview** section of this documentation.

Data enters into STEP through external channels where customers will provide their information. In section 1 of the image below, these channels may be through a telephone order, a web order, when they sign up for a loyalty platform, or when they buy in-store.

After their data is recorded, it is saved to various source systems. A source system is any system that stores data to be delivered to the CMDM solution. In this use case, the source systems are the CRM, ERP, and E-commerce platforms (Figure 2). Once the data is inside of the CMDM application, data is standardized, matched with existing records, and golden records are created / maintained. Data will then be stored in a datastore / data warehouse.

For more details on what happens in Figure 3, see the Application path image following this section. Also, see the **Matching** topic and the **Loqate** topic of this documentation for more information.

In a coexistence implementation, the data matched and merged in the CMDM solution (Figure 3) will be updated back at the source systems (Figure 2) as well as downstream solutions (Figure 4).



Data Flow Use Cases

Use Case 1 - Business to Business (B2B)

Scenario 1

Benny, a representative from the sales team, is responsible for retailers and distributors that have contracts to resell ACME's products.

Benny needs to be able to maintain sales terms and conditions, as well as maintain relationships with appropriate contact person(s) for each retailer / distributor (sales reps, account owners, purchasers, etc.).

He requires his retailer / distributor information to be accurate in order to intelligently negotiate appropriate terms and conditions.

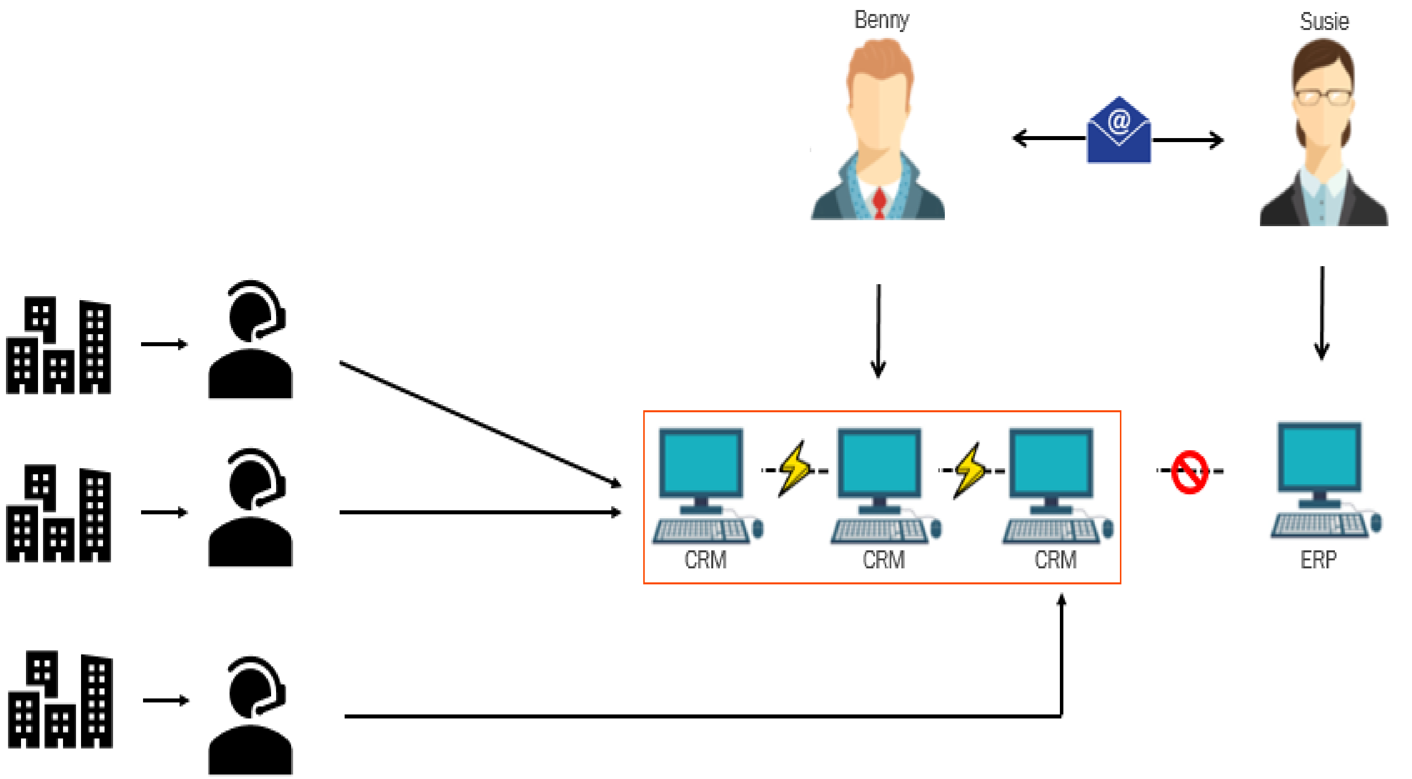
Susie from Financing reaches out to Benny to request the retailers / distributors contact information.

Benny primarily interfaces with ACME's CRM system to track and edit all relationships and conditions.

As field representatives enter contact information for their retailers / distributors, they may enter the same contact person multiple times. Since different CRM instances track sales and relationships across different regions of the country, this requires deduplicating across the various CRM platforms. This can be a complex process for Benny to look up the proper contact person for a retailer who's parent company information is maintained in a different CRM instance. This can be frustrating to Benny as he must first identify what the parent-child subsidiary relationship is and then has to determine which CRM instance to search in. Even so, there is currently no guarantee that the retailer information is stored in a single CRM instance nor is there any governance to ensure the data is up to date.

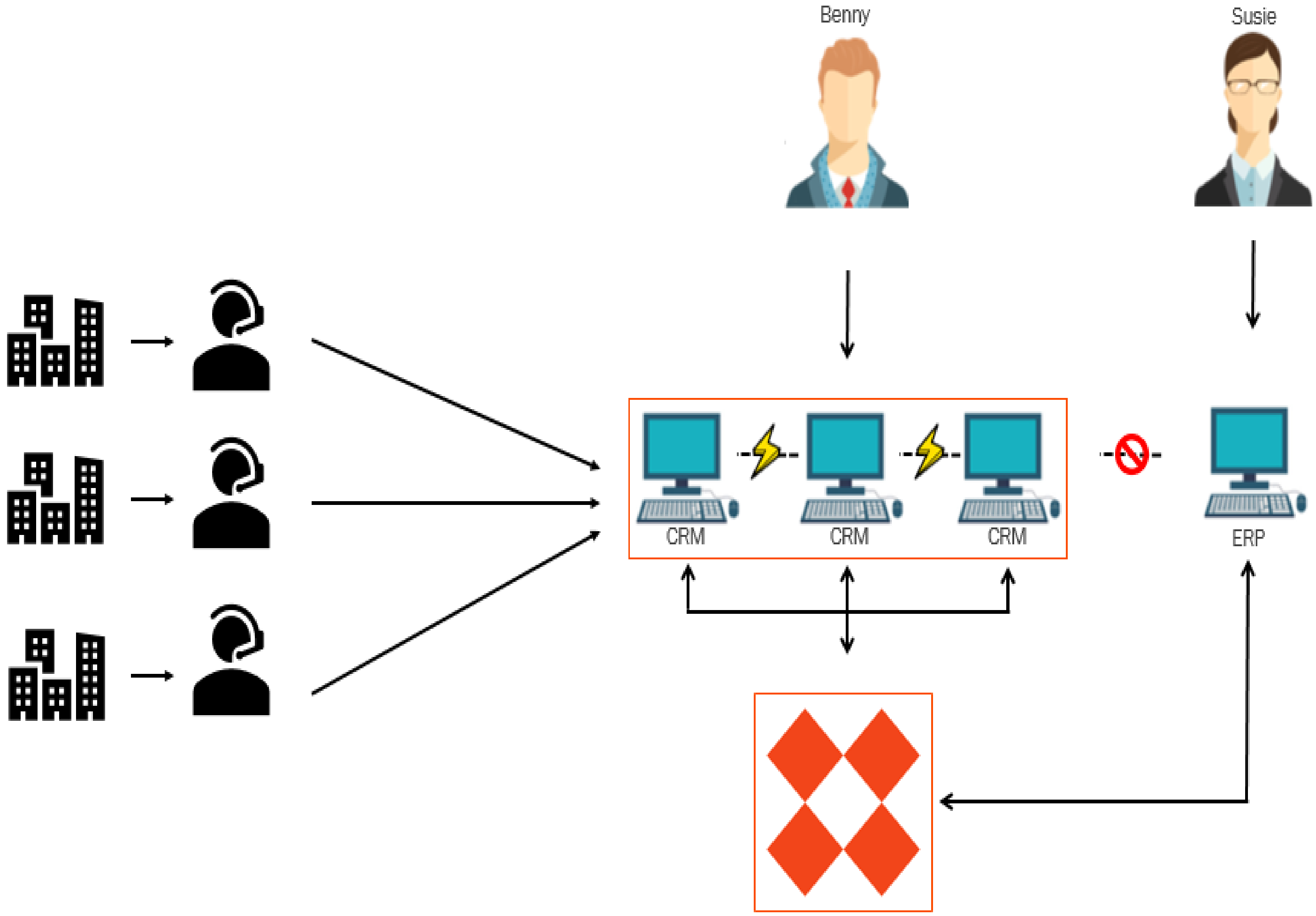
To account for these challenges, ACME requires a solution which assists in determining which Primary Contact is accurate, up to date, and appropriate to the inquiring department (Marketing contact for Marketing Analyst, Technical Contact for IT, etc.)

Current Data Flow:



Within the current system landscape Contacts are created and stored across many CRM systems. These systems do not interact frequently enough. This means Benny may find duplicated records across the systems, which makes it difficult to determine who the primary contact is. Susie works out of ERP and must notify Benny via email to have contacts or payment terms updated.

To-Be Data Flow:



With a CMDM solution in place, all data is fed into STEP. It is then synchronized back to the various source systems in a timely manner. The solution consolidates all contact persons from the various CRM systems and also deduplicates them within each organization.

This ensures that Benny has the most accurate and complete data available regardless of which CRM he is accessing.

Additionally, Susie is able to rely on the data in ERP because it is in constant communication with CMDM.

Scenario 2

Doris is also part of the sales team but works primarily in the field.

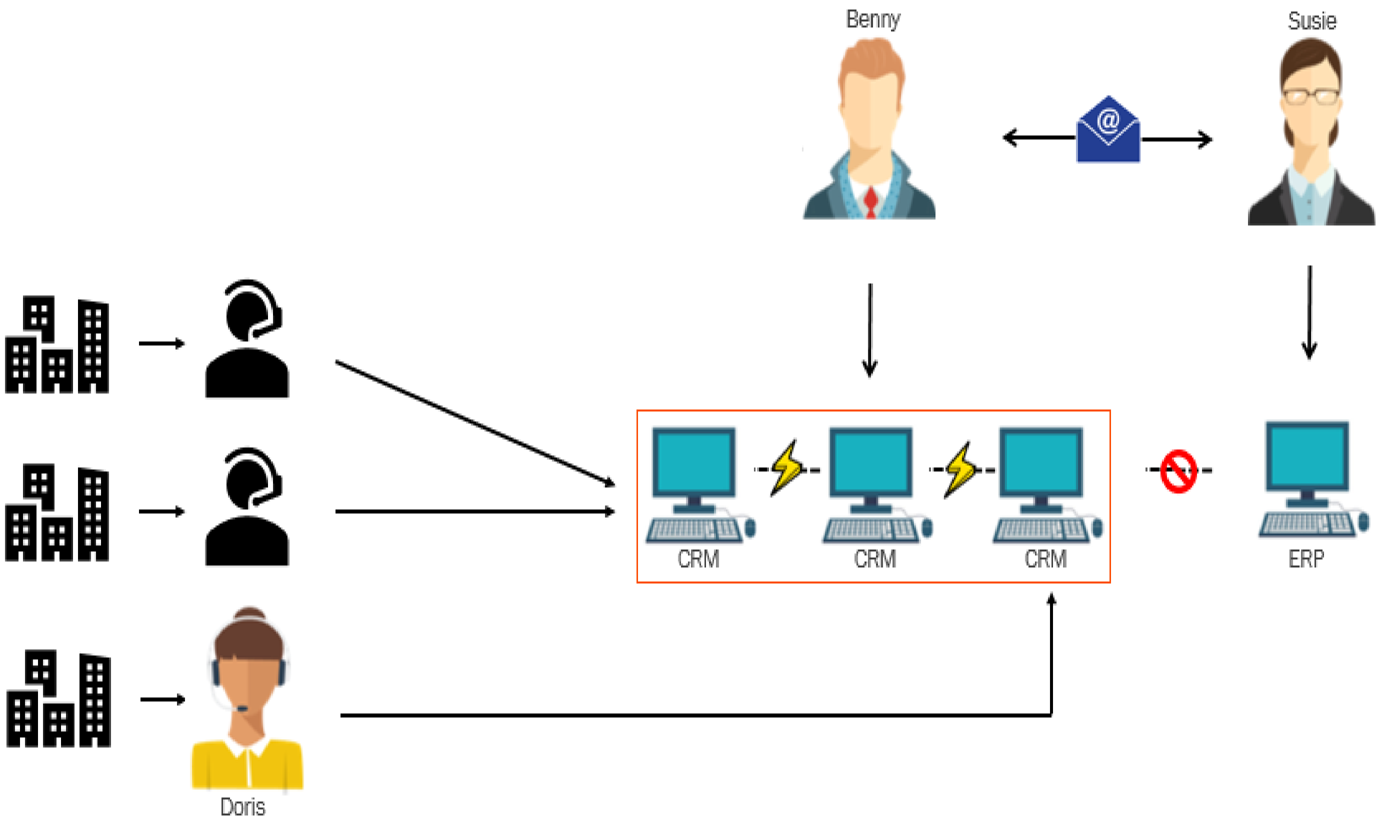
Her role requires her to meet with ACME's retailers / distributors to identify and provide relevant product offerings based on climate, demographics, and demand that are specific to individual sales regions and territories. In doing this, ACME's strategy is to increase brand awareness and up-sell / cross-sell opportunities by empowering their retailers / distributors to cater to the demand of the end-consumers.

As a saleswoman, Doris must have the most up to date information on which contacts representing retailers and distributors to invoice. To do this, Doris relies on her access to ACME's ERP system to track all transactions.

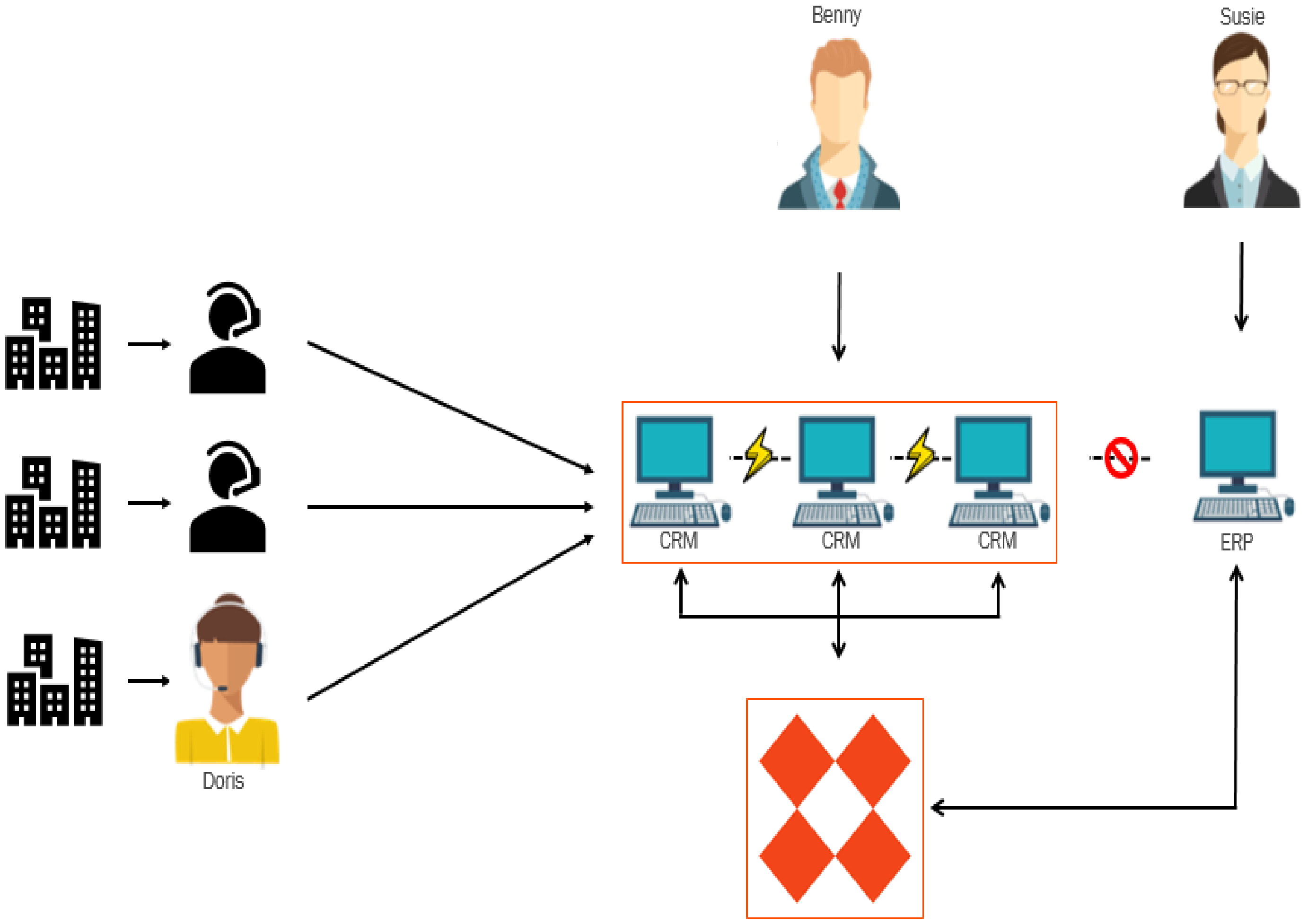
While Doris' job depends heavily on Benny's work, ACME's ERP system does not synch with their various CRM systems in a timely manner. Because of this, Doris may place orders without realizing that payment conditions have changed, leading to her and the customer operating under out-dated terms. This miscommunication leads to frustrating ACME's retailers / distributors when they sign up for one payment term but are invoiced according to another. End customers' shopping experience may also be negatively impacted when retailers pass along the cost differential to them.

In summary, these issues lead to: low governance, payment risks, diminished customer experience, and increased costs and potentially lost revenue.

Current Data Flow:



With the current System landscape Doris does not have the most up to date information available to her. This may lead to Doris setting payment terms based on outdated data.



With CMDM in place Doris can be sure that she is working with the best possible data. This allows her to set appropriate payment terms and avoid duplicating contacts no matter what CRM she interacts with.

Scenario 3

In this scenario, Jack works with business analytics and is responsible for creating business reports. He succeeds when he can create accurate reports and insights into the state of the business.

More specifically, Jack needs to provide other departments (marketing, e-commerce, finance, and management) with information such as:

- How many customers do we actually have?
- Who are the best customers?
- How much and which products do they buy?
- Are my customers stand-alone or do they belong to a group?

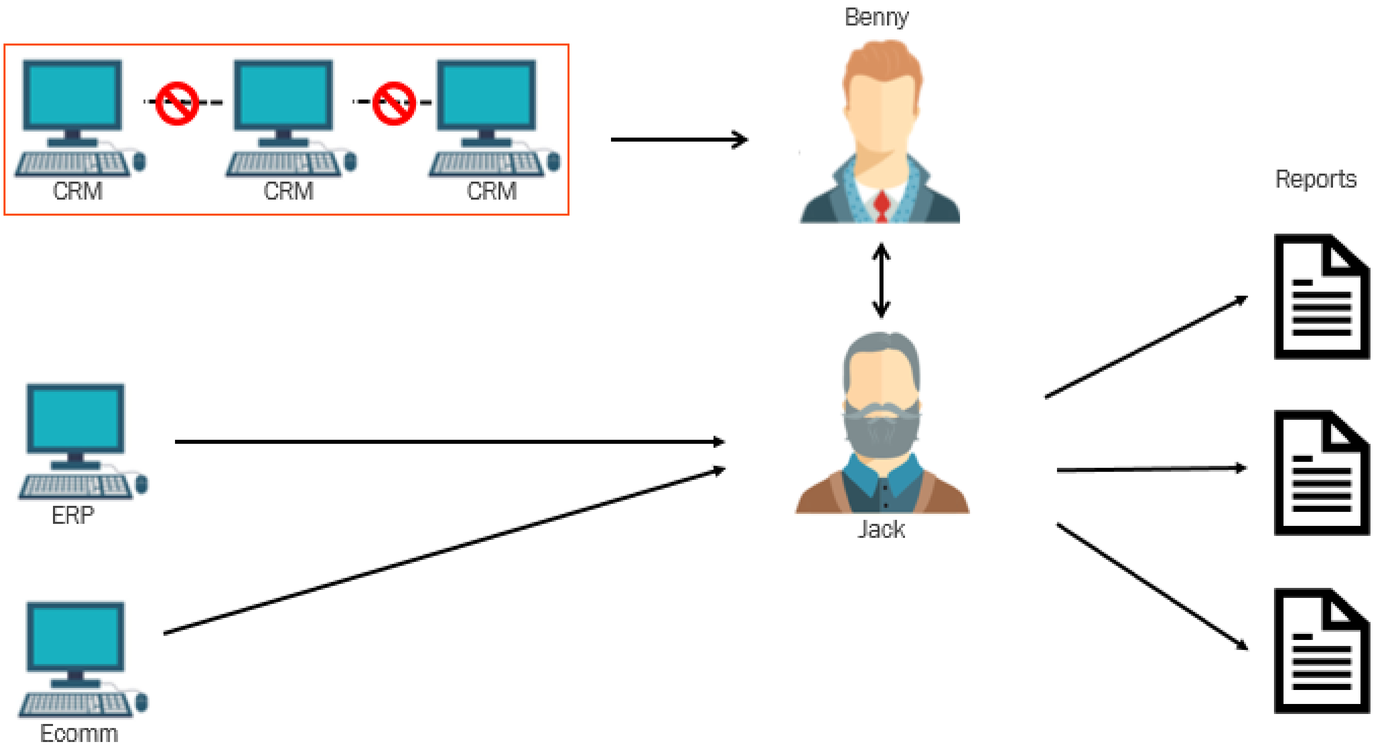
Jack creates all reports in the Business Intelligence system.

Unfortunately, ACME's current solution does not provide a good way for marketing (and other departments) to maintain and build their specific hierarchies, nor is it possible to ensure the availability, usability, and integrity of the data. When there is no consolidated, trusted view, Jack is unable to provide for them an accurate overview of the customer.

The low-quality data this solution generates is sent to the BI system from various sources. The chief issues with the data include: duplicates, incomplete records, conflicting data, no source of truth, and differing data formats.

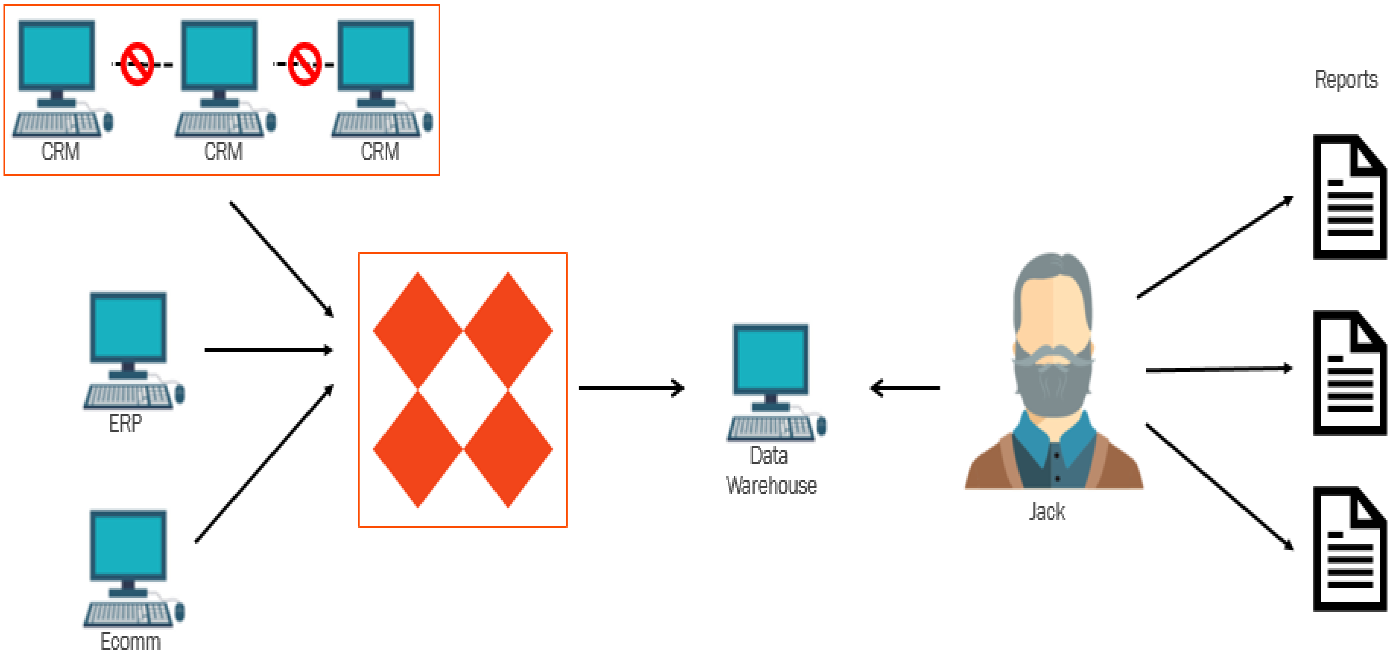
These issues result in reactive or wrong strategic decisions, and potential revenue in cross-sell is not realized.

Current Data Flow:



Jack must interact with Benny to receive data from the CRMs since he does not have access to them. This leads to conflicting or inaccurate data for the reports. With bad data, the reports do not generate the value they should.

To-Be Data Flow:



With the CMDM solution in place, all source systems are being read when generating the master data. This allows Jack to pull whatever data he needs from the CMDM system rather than multiple siloed source systems. This ensures that reports can be trusted because they are generated off the most accurate and up-to-date information. This includes golden customer records and golden company hierarchies. The golden data reports that Jack provides allows marketing to efficiently run campaigns. The golden company hierarchy allows Finance to more accurately run quarterly roll-up reports.

Use Case 2 - Business to Customer (B2C)

Scenario 1 (Marriage)

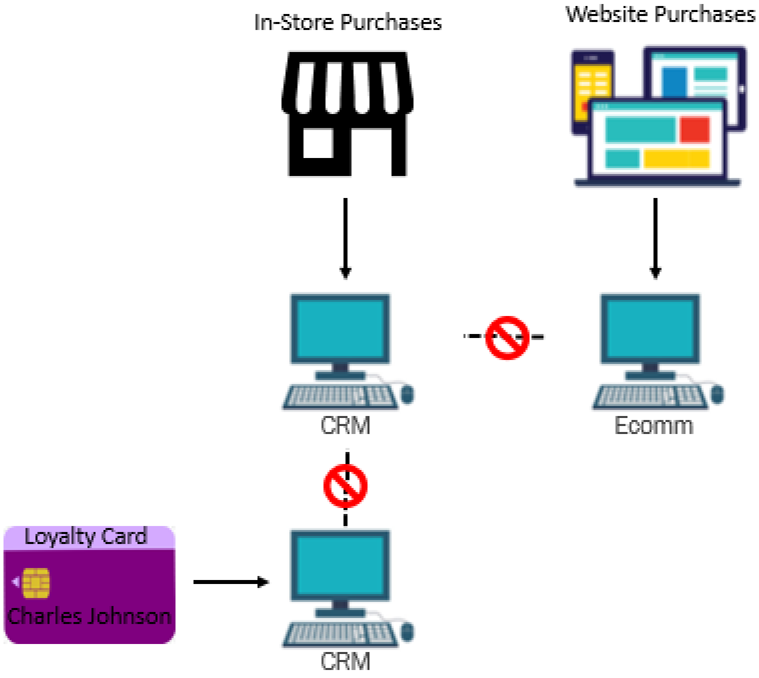
In this scenario, Catherine Yu lives in Texas and is a customer of ACME who enjoys the experience of discovering the latest clothing trends offered within ACME's physical (B&M) stores.

Catherine occasionally shops on ACME's e-commerce site due to the online exclusive promotions and friendly in-store return policy. Having recently gotten engaged, Catherine signed up for ACME's customer loyalty program so she can take advantage of additional loyalty perks and savings in order to save for the wedding and honeymoon.

Sandra manages Catherine's order from a B2C e-commerce site which contains no address validation, allows multiple accounts to be created for a single customer, and lacks any form of de-duplication of customer records. Lacking address validation leads to missed order deliveries which damages brand reputation. Allowing multiple customer accounts of a single record due to no deduplication leads to loss of cross sell opportunities as an email or sales catalog may be sent based off the information of an old account. Deduplicating and consolidating these records allow all contact information to be available on a single entity, eliminating errors caused by multiple conflicting entities

Charles Johnson, Catherine's fiancé, is from Michigan and is an occasional shopper of ACME brand clothing. As a loyalty program member, Charles has decided to take advantage of recent promotions in preparation for their wedding. Upon marriage, Catherine has chosen to take Charles' last name and moved to Michigan with Charles.

Customer data gathered in physical stores are managed by a CRM platform which associates credit card payments to specific customers to ensure faster processing and confirmation or receipt delivery, while customer data gathered from the e-commerce site is managed by an e-commerce system. Customer loyalty sign-ups are tracked by a different CRM platform that is owned and maintained by the marketing department.



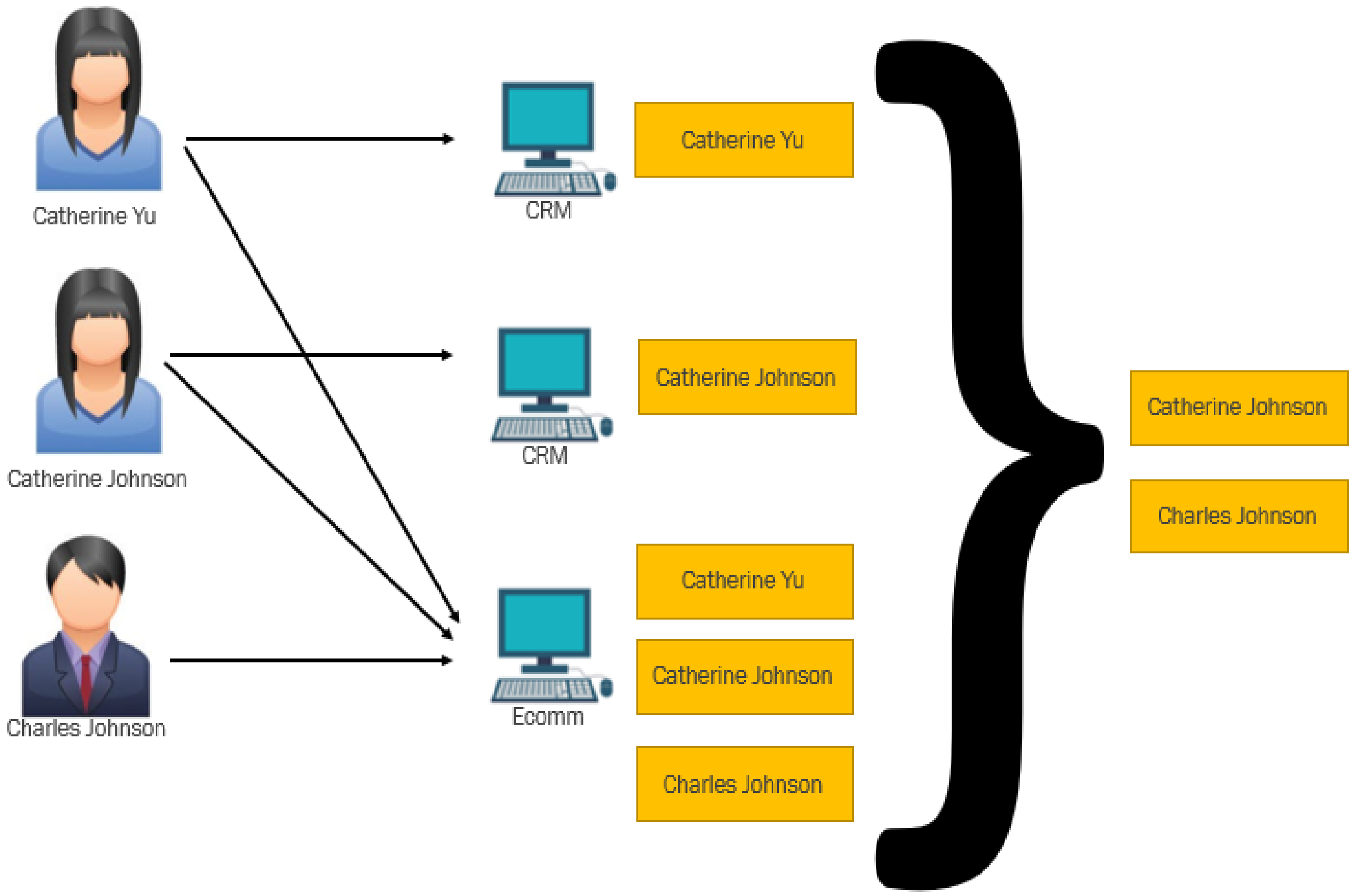
Because ACME's multiple CRM and SAP systems are siloed and are not synched with one another there is no automated capability for ACME to identify that Catherine Yu and Catherine Johnson are the same customer. ACME also does not have a straightforward way to identify customers who have relocated to a different address.

Furthermore, ACME would like to be able to identify customer groupings as households in order to minimize wasted marketing mailings and to better target buyer behaviors.

As these challenges are currently not addressed, several problems arise:

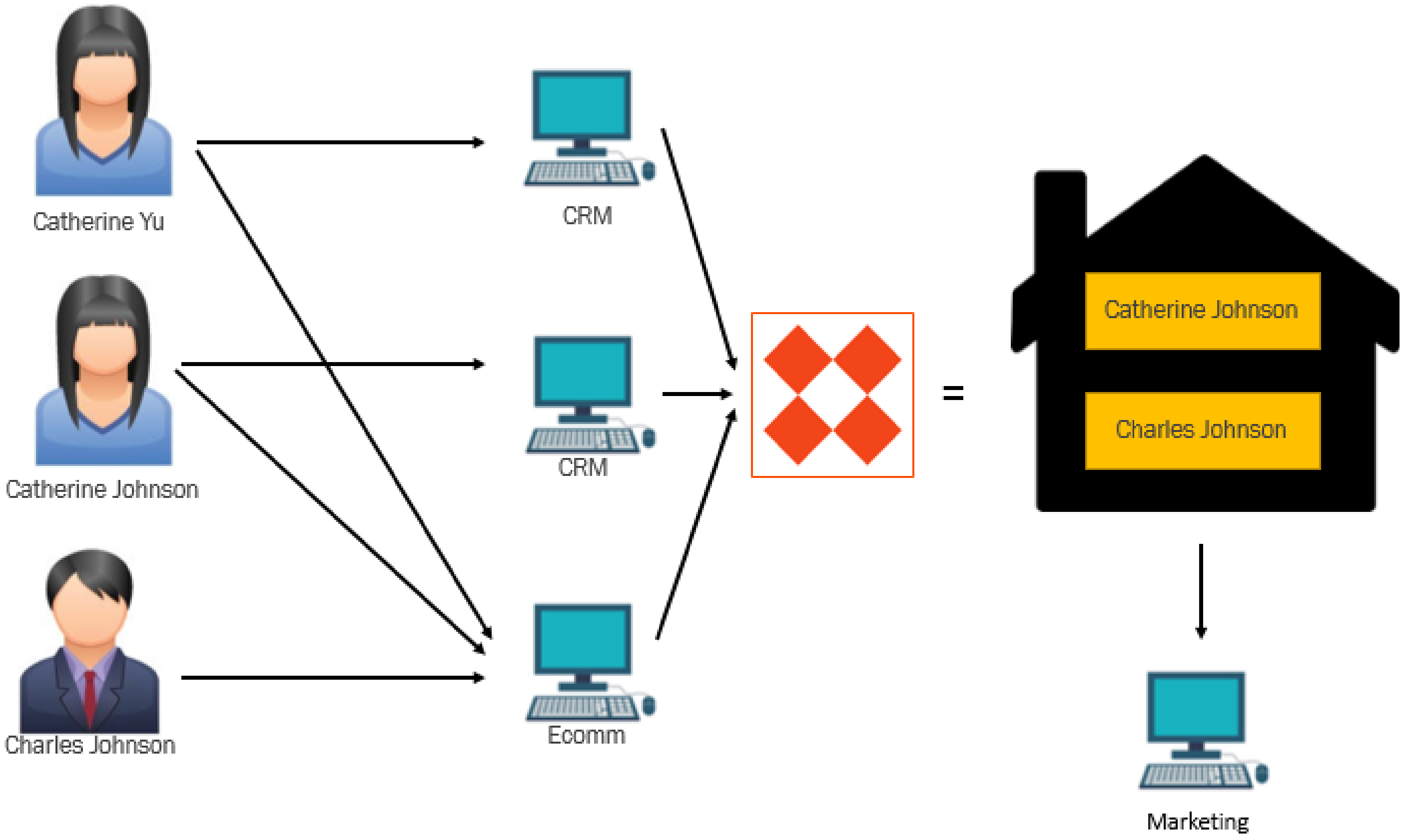
- For every seasonal marketing campaign, mailings that are sent to Catherine end up at her old address in Texas which causes her to miss out on certain promotions and even important rebate checks.
- Charles gets some of the same marketing mailings but since he's not a big shopper, he ends up throwing them out before Catherine sees them.
- Even after Catherine had to call into the ACME call center to update her address, ACME still sent double of every marketing mailing to Catherine and Charles even though they live together which adds to the frustration in Charles' and Catherine's customer experience.

Current Data Flow:



The e-commerce system is unable to identify that Catherine Yu and Catherine Johnson are the same customer. This leads to three separate records being created. No connection between Catherine and Charles has been identified.

To-Be Data Flow:



The CMDM system can identify that Catherine Yu is the same customer as Catherine Johnson and will merge them into a single golden record. Additionally, CMDM will be able to identify the Johnson Household based on a shared last name and address. The household concept allows marketing to avoid redundant mailings and better target offers and promotions.

Scenario 2 (E-Commerce)

Sandra is responsible for managing orders coming from the new B2C e-commerce site, as well as returns. She succeeds when she can deliver the right product to the right customer at the right time.

ACME Company’s goal is to make it as easy as possible for customers to buy their products. If customers do not remember their log in, they can simply create a new user and then place the order, or even place smaller orders without creating a user. Customers must also provide their delivery address.

Poor data quality has created several challenges for ACME Company, preventing them from achieving their goal:

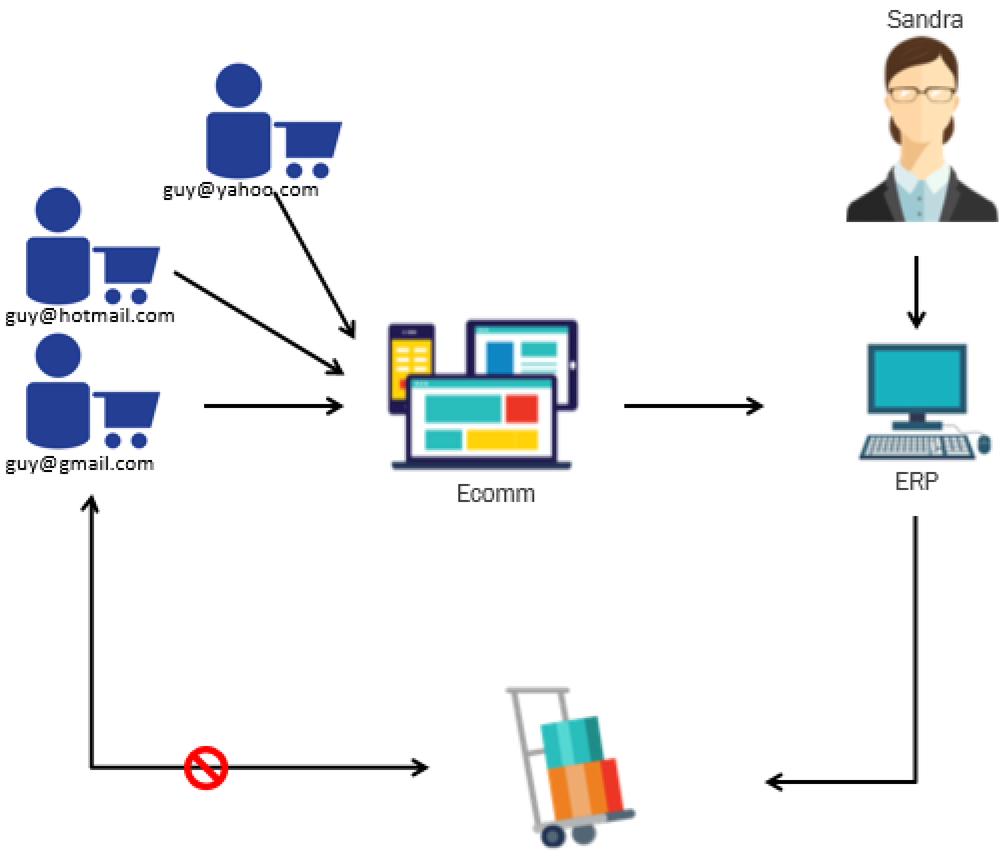
- Addresses are not validated before the order is sent to logistics, leading to a lot of products delivered to wrong addresses or not arriving to the right customer at the right time.
- Sometimes customers accidentally misspells their address or leave out vital information.
- Duplicates are and will continue to be created in the e-commerce and ERP systems.

- Neither the e-commerce system or ERP can check for duplicates, obfuscating how much a customer actually buys from ACME.
- Customers do not receive their orders in a timely manner (or at all).
- Inconsistent customer experience, as duplicate accounts may lead to orders not being associated between duplicates.

These issues result in:

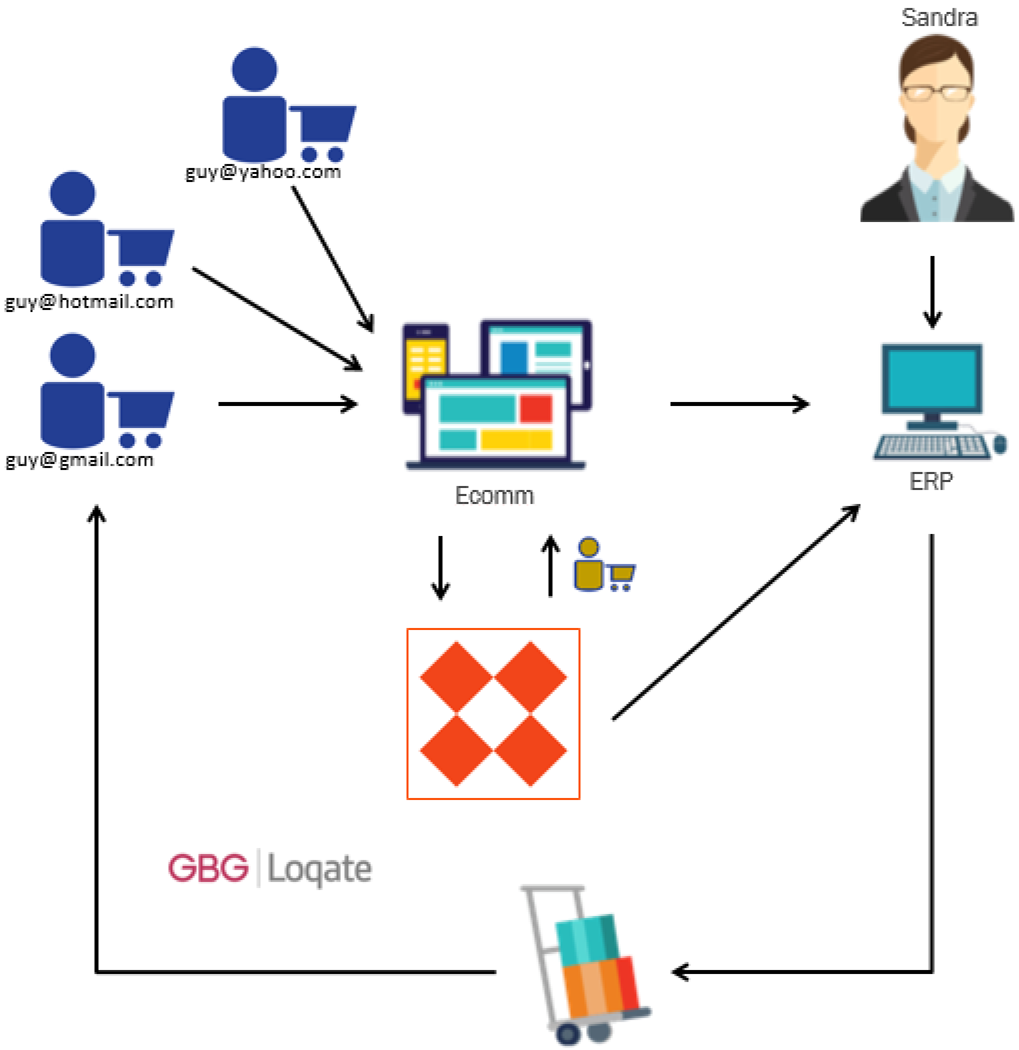
- Low data quality of delivery locations leads to instances of 'return to sender' and double shippings, causing increased costs.
- Poor customer experience due to orders not arriving in a timely manner.
- Lost revenue and loss of customers. If the products are not delivered in a timely manner, customers will buy products from elsewhere.
- Increased returns.
- Duplicate records carry operational and analytical consequences. Operationally, the correct delivery address may be available on a duplicate record. This leads to customer frustration if ACME is not considering that record when shipping. Analytically, reports may be skewed as duplicate records cause a single customer to be weighed multiple times during analytics.

Current Data Flow:



The customer places orders that are processed out of an e-commerce system. The customer and order details are sent to the ERP system for processing and to get shipments out. Currently, there are no validations being done on the provided address. This causes customers to not receive their purchases in a timely manner or not at all. Additionally, shipments may be returned to sender causing increased costs.

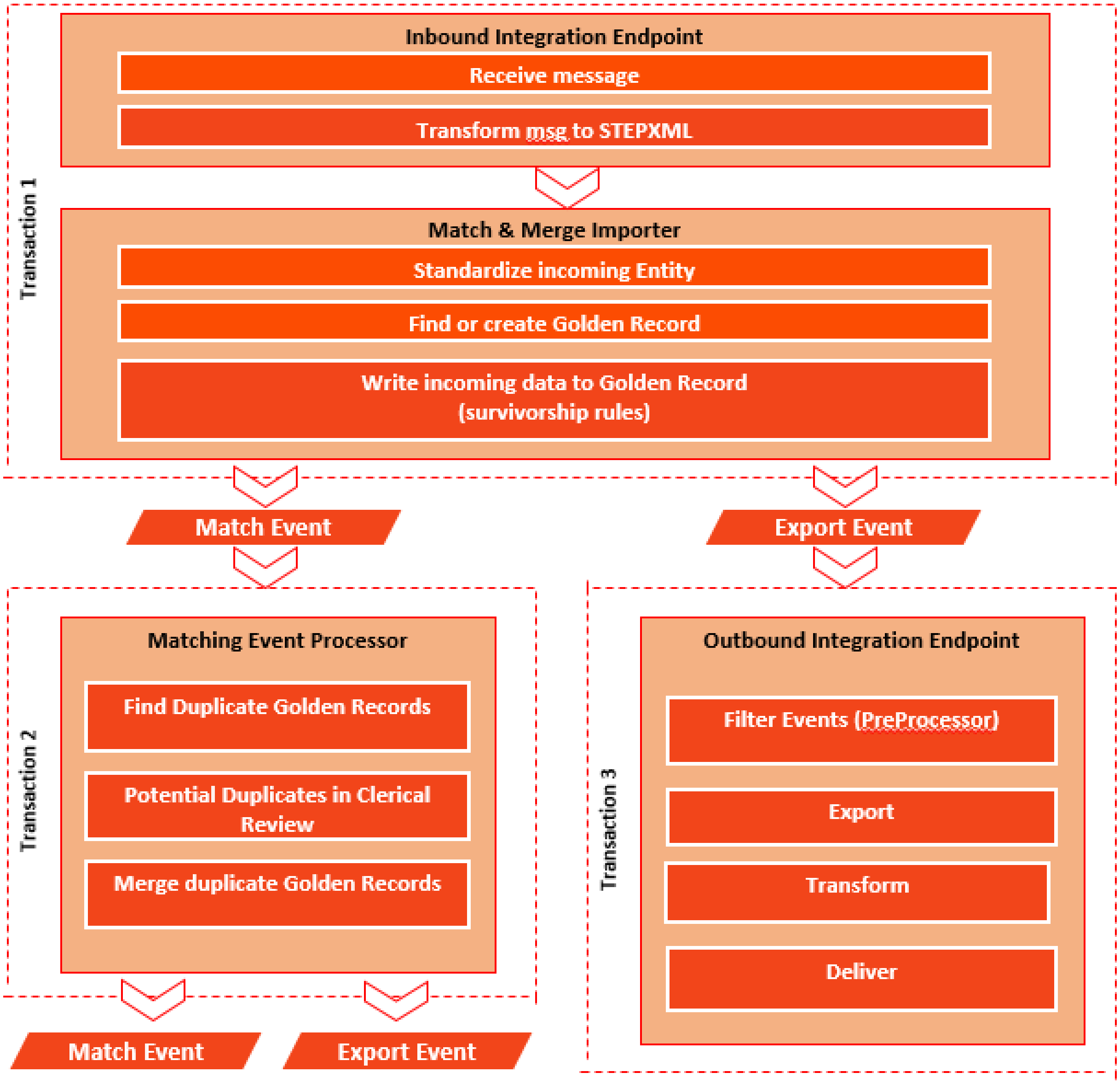
To-Be Data Flow:



The CMDM solution is integrated with Loqate to verify and standardize addresses. This increases address quality, reducing instances of 'return to sender' and deliveries shipped to the wrong address. CMDM will interact with the ERP, in place of the e-commerce system, to ensure only mastered data is sent to ERP for order processing.

Data Flow in STEP

This section provides a process overview of the primary data flow throughout the STEP system.


Transaction 1
Inbound Integration Endpoint
Receive message
Transform msg to STEPXML
Match & Merge Importer
Standardize incoming Entity
Find or create Golden Record
Write incoming data to Golden Record (survivorship rules)
Match Event
Export Event
Transaction 2
Matching Event Processor
Find Duplicate Golden Records
Potential Duplicates in Clerical Review
Merge duplicate Golden Records
Match Event
Export Event
Transaction 3
Outbound Integration Endpoint
Filter Events (PreProcessor)
Export
Transform
Deliver

Transaction 1: Inbound Integration Endpoint

An inbound integration endpoint will receive a message and transform it into STEPXML.

Once transformed, the Match & Merge Importer will:

- Execute standardizations and other associated business rules to cleanse the data. Cleansed data allows for the Match algorithm to more accurately identify duplicate records.
- Determine the target Golden Record. For more information, see the **IIEP - Configure Match and Merge Importer** section of the **Data Exchange** documentation.

- Survivorship rules will write any updates from the incoming data to the golden record. For more information, see the **Golden Records Survivorship Rules** section of the **Matching, Linking, and Merging** documentation.
- To the extent that a Golden Record was created or an existing Golden Record was updated, events will be written to event queues triggering on those changes. In this case, a Matching event processor as well as an Outbound Integration Endpoint.

Transaction 2: Matching Event Processor

When new golden records are created or an existing golden record is updated, this will trigger a matching event, with the purpose of comparing golden records to each other.

Prior to matching, the golden record under treatment will have its match codes updated.

Matching golden records may result in a clerical review workflow initiation or a merge, depending on the score.

For more information on Clerical Reviews, see the **Clerical Review** section of the **Matching, Linking, and Merging** documentation.

Transaction 3: Outbound Integration Endpoint

Whenever Golden Records are created or changed, an export event will trigger. This allows the mastered data to be synced back to the various source systems.

For more information about Outbound Integration Endpoints, see the **Outbound Integration Endpoint** section of the **Data Exchange** documentation.

Note: Clerical Review is not a blocking state.

Use Case 1 : New Golden Record

1. **Transaction 1 : Import**

   - IIEP: Receive message
   - IIEP: Transform message to STEPXML
   - M&M: Standardize Address
   - M&M: Check if Source System and Source Record ID is associated to existing, active Golden Record (it is not)
   - M&M: Generate Match codes for incoming data and look for matching record using Match Code and Matching Algorithm (no match is found)
   - M&M: Create new Golden Record
   - M&M: Survivorship Rules Write data to new Golden Record

2. **Transaction 2 : Handle duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - No duplicate golden records found
  - No potential duplicate golden records found

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 2 : Update Existing Golden Record with Known Source Record ID

1. **Transaction 1 : Import**

- IIEP: Receive message
- IIEP: Transform message to STEPXML
- M&M: Standardize Address
- M&M: Check if Source System and Source Record ID is associated to existing, active Golden Record (it is)
- M&M: Survivorship Rules Write data to existing Golden Record

2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - If an absolute match is found, auto-merge
      - MA: Survivorship Rules Write data to existing Golden Record
  - If potential duplicate golden records are found, initiate into clerical review

3. **Transaction 3: Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 2.1 : Update Existing Golden Record with Known Golden Record ID

1. **Transaction 1 : Import**

- IIEP: Receive message
- IIEP: Transform message to STEPXML
- M&M: Standardize Address
- M&M: Find existing Golden Record based on STEP ID (golden record id)
- M&M: Survivorship Rules Write data to existing Golden Record

2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - If an absolute match is found, auto-merge
      - MA: Survivorship Rules Write data to surviving Golden Record
  - If potential duplicate golden records are found, initiate into clerical review

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 3 : Add new source record to existing golden record by matching

1. **Transaction 1 : Import**

- IIEP: Receive message
- IIEP: Transform message to STEPXML
- M&M: Standardize Address
- M&M: Check if Source System and Source Record ID is associated to existing, active Golden Record (it is not)
- M&M: Generate Match codes for incoming data and look for matching record using Match Code and Matching Algorithm (match is found)
- M&M: Update Existing Golden Record with additional source record ID
- M&M: Survivorship Rules Write data to existing Golden Record

2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm

  - If an absolute match is found, auto-merge
    - MA: Survivorship Rules Write data to surviving Golden Record
  - If potential duplicate golden records are found, initiate into clerical review

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 3.1 : Add New Source Record to Existing Golden Record by Golden Record ID

This use case is relevant when the getSimilarObjects web service has been used to identify that a golden record for a specific customer already exists and a new source record needs to be associated with it.

1. **Transaction 1 : Import**

- IIEP: Receive message
- IIEP: Transform message to STEPXML
- M&M: Standardize Address
- M&M: Find existing Golden Record based on STEP ID (golden record id)
- M&M: Update existing Golden Record with new source record ID
- M&M: Survivorship Rules Write data to new Golden Record

2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - If an absolute match is found, auto-merge
    - MA: Survivorship Rules Write data to surviving Golden Record
  - If potential duplicate golden records are found, initiate into clerical review

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 4 : Add New Source Record to Existing Golden Record by Matching, Clerical Review

1. Transaction 1 : Import

- IIEP: Receive message
- IIEP: Transform message to STEPXML
- M&M: Standardize Address
- M&M: Check if Source System and Source Record ID is associated to existing, active Golden Record (it is not)
- M&M: Generate Match codes for incoming data and look for matching record using Match Code and Matching Algorithm
- M&M: Match Rules will not find certain matches
- M&M: Create new Golden Record
- M&M: Survivorship Rules Write data to new Golden Record

2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - Identify that newly created golden record is within clerical review threshold and put oldest record in clerical review workflow

3. **Transaction 3: Merge**

- User review duplicate records in clerical review
- User merge potential duplicates
- Source record references from non-surviving record gets written to surviving record
- References TO the non-surviving record gets moved to the surviving record
- Survivorship rules promote updates from non-surviving- to surviving golden record
- Non-survivor gets deactivated and Merged-Into reference is established from non-survivor to survivor

4. **Transaction 4 : Export**

The deactivated record will be exported, as the 'Merged-Into' reference was added and the 'Is Deactivated' was changed to 'true'. The surviving record will be exported, as source records were added to it. Any other object having a reference target moved from the non-surviving record to the surviving record will get exported, to the extent triggering definitions are defined for it.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 5 : Manually Modify Golden Record

1. **Transaction 1 : Manual Edit**
   - Data Steward: Manually makes edits on an existing Golden Record via the Web UI.
2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
  - If an absolute match is found, auto-merge
    - MA: Survivorship Rules Write data to existing Golden Record
  - If potential duplicate golden records are found, initiate into clerical review
    - MA: If match is confirmed, survivorship Rules Write data to existing Golden Record
    - MA: If match is rejected, create a 'Confirmed Non-Duplicate' reference between the two (or more) Golden Records.

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

Use Case 6 : Import Cross References by Source Record ID

1. **Transaction 1 : Import**
   - IIEP: Receive message
   - IIEP: Transform message to STEPXML
   - M&M: Standardize Address
   - M&M: Check if Source System and Source Record ID is associated to existing, active Golden Record (it is for the target entity)
   - M&M: Generate Match codes for incoming data and look for matching record using Match Code and Matching Algorithm
   - M&M: Match Rules will not find certain matches
   - M&M: Create new Golden Record (Source entities)
   - M&M: Survivorship Rules Write data to new Golden Record including creating cross references
2. **Transaction 2 : Handle Duplicate Golden Records**

The Matching Event Processor polls for events on a schedule.

- Update match codes for changed Golden Record and look for matching Golden Records using Match Code and Matching Algorithm
    - If an absolute match is found, auto-merge
        - MA: Survivorship Rules Write data to surviving Golden Record
    - If potential duplicate golden records are found, initiate into clerical review

3. **Transaction 3 : Export**

Outbound Integration Endpoints poll for events on a schedule. Notice that transaction 2 and 3 are independent.

- OIEP : No event PreProcessor in use
- OIEP: Export data to STEPXML
- OIEP: Transform to target message format
- OIEP: Deliver message to target

When the Match and Merge Importer reads cross references in the STEPXML, the target Entity may be specified with a source system and a source record ID, instead of a STEP ID. This allows import of references to objects, where the STEP ID is not known.

Matching

With a proper Matching strategy, organizations can consolidate their customer data from a variety of source systems and generate singular, definitive golden records for those individual customers.

Before creating a Matching configuration, it is important to get an understanding of the client organization's data and identify challenges the algorithm must account for. With this knowledge, a clear match strategy can be defined.

Once a strategy is defined, the Matching Algorithm is typically configured first, followed by the Match Codes. Both should be tailored to the client organization's data. The algorithm should be configured first in order to know what data points specific to the client organization will provide reliable matching. With the data points identified, you can define what attribute combinations will contribute to the match codes.

When the initial configurations are complete, they can be iterated upon via Matching Tuning sessions. The goal of these sessions is to gradually improve the accuracy of the matches being made, until you arrive at your final Matching configuration.

For more information on the Matching process and configuring the relevant components, see the **Matching, Linking, and Merging** documentation.

Matching Algorithm Configuration

The Matching Algorithm should be tailored to the data, and strive for precision. The CMDM Starter Package includes four pre-configured Matching Algorithms that can be used as a foundation to build a client organization's Matching solution:

- Individual Customer
- Organization Customer
- Contact Person
- Household

For more information, see the Algorithm & Match Codes topics.

When configuring the Matching Algorithm, it is important to consider the impact that Thresholds have on match results. If the Clerical Review threshold is set too high, a large amount of false negatives may be generated. Similarly, if the Auto Threshold is set too low, false positives could be generated. If the initial Matching configuration produces false negatives and/or false positives, the Thresholds should be reevaluated during the Match Tuning sessions.

For more information on configuring a Matching Algorithm, see the **Configuring Matching Algorithms** section of the **Matching, Linking, and Merging** documentation.

Match Codes Configuration

Match Codes should aim to ensure that anything the algorithm may score high is not left out, and that only those records that may score high get compared. The rules defined by the Match Criteria determine whether or not any given Match Code is relevant. The Match Codes should be considering the same data points that the Match

Criteria does. This ensures that the comparison pool that Match Codes generate is relevant to the data points the Match Criteria is matching on.

For example, if the Match Criteria is matching on a combination of Person Name and Address, it is not recommended to generate Match Codes based on unrelated attributes (e.g. Phone and Email).

Note: Configuring Match Codes is a development task, as it requires advanced scripting. However, the Starter Package comes with a library of scripts to get you started.

Example Match Codes

Typical attributes used to generate Match Codes for Individual (B2C) solutions include Name, Address, Email, and Phone. For Organization (B2B), these attributes potentially include Name, Address, Phone, DUNS (D&B Number), and Tax Identifier.

Email

Email is often used in combination with a person name for Match Criteria rules. In matching, no edit distance or phonetics is allowed. Email is potentially unique enough to warrant a high score on a match when included in a Match Criteria rule, removing the need for other Match Codes.

Example: A customer with the email 'HendrickBarnard@acme.com' becomes the Match Code 'EM~hendrickbarnard@acmecom'.

Phone

Phone shares many of the same benefits that email provides. However, it is common to have variations in whether country code is included or not. This can be solved by only including the last 7 digits of the phone number in the match code (for US phone numbers).

Example: A customer with the phone number '(337) 934-8392' becomes the Match Code 'PH~9348392'.

Person Name (First, Middle, and Last)

Person Name is often used in Match Criteria rules in combination with emails, phone numbers, or addresses. It has fairly low uniqueness compared to other Match Codes. When matching, techniques like edit distance, equivalent values and soundexes are used. Using person name as a Match Code requires a lot of simplification of the values in order to avoid false negatives and requires combination with other data elements (email, phone, etc.) to be sufficiently unique. For email and phone combinations, other Match Codes already capture the relevant candidates. It gets complicated when used in combination with addresses, as these suffer from some of the same complexities as person names.

Example: A customer with the name 'Hendrick Barnard' becomes the Match Code 'PN~h~PRNRT'.

Address

Address is often used in combination with person name or organization name. It is fairly unique if the full address is used, but has low uniqueness if only part of the address is used.

The accuracy of addresses varies (i.e. one entry may include suite no. while another may not). When matching, techniques like edit distance is used on city and street. Because of this, and the fact that accuracy varies, the full address does not make a good match code, as it will likely lead to false negatives.

Addresses are often abbreviated ('st' for street or station, etc.) in complex patterns that cannot be uniquely resolved easily. For high accuracy of address matching, it is therefore recommended to use STEP’s Address standardization capabilities, which are integrated to Loqate.

Example: A customer with the zip code '92627' and country code 'US' becomes the Match Code 'ZICC~92627~us'.

Name and Address

Combining elements of a person name and elements of an address is often a good way to create Match Codes that are sufficiently unique, without causing false negatives. However, multiple permutations are often required to avoid false negatives.

Example: A customer with the name 'Hendrick Barnard' and zip code '92627' becomes the Match Code 'ZINM~92627~h~PRNRT'.

Other examples of Name and Address combination Match Codes:

- Zip + first letter of first first name + metaphone3 of last last name
- Zip + first letter of last last name + metaphone3 of first first name
- Metaphone3 of city + first letter of first first name + metaphone3 of last last name
- Metaphone3 of city + first letter of last last name + metaphone3 of first first name
- In case first name equivalents exists, it is likely necessary to add permutations for these as well

For more information on configuring Match Codes, see the **Configuring Match Codes** section of the **Matching, Linking, and Merging** documentation.

Starter Package Match Codes

The CMDM Starter Package includes four pre-configured sets of Match Codes that correspond to the four Matching Algorithms mentioned in the above section:

- Individual Customer
- Organization Customer
- Contact Person
- Household

For more information, see the Algorithm and Match Codes topics.

Matching Algorithm Tuning

Algorithm Tuning begins during the build phase of the implementation. Early on in this process, it is common to find a large number of invalid matches making it past the Auto-Merge threshold while valid matches fall short of the Clerical Review threshold. Thus, the ultimate goal of the algorithm tuning sessions is to perfect the matching logic's accuracy so that good and bad matches score within the appropriate thresholds.

This goal can be achieved by:

- Adjusting the scoring method and weighting of each scored attribute.
- Adjusting the relative weighting of scoring across all the scored attributes.
- Adjusting the Auto-Merge and Clerical Review Thresholds.

Note: It is important to have both the data steward and data owner present during algorithm review sessions. In this context, the data steward is the business user who has been tasked with the formation and execution of policies for the management of data and metadata. The data owner is the business user who typically has a direct line of responsibility for a functional area.

Algorithm Tuning Process

The Matching Algorithm Tuning process is as follows:

1. Configuration: Utilize a Match Tuning configuration to generate a Data Profile. Using this Data Profile, identify key data points to consider when configuring a baseline algorithm (Matching Algorithm and Match Codes).
2. Generate Sample Pair: Once the baseline algorithm is configured, generate the Random Sample Pair spreadsheet via a Match Tuning configuration. This baseline configuration is just a 'best-guess' configuration based on the analysis of the customer data so far.
3. Review Sample Pair: Review the Sample Pairs with the client. Each individual pair gets either a 'Yes', 'No', or 'Not Sure' indicating whether or not they should be considered the same entity by the algorithm and linked together.
4. Tuning the Algorithm: Tune the algorithm based on feedback from the Sample Pair review, and generate a new set of Sample Pairs based on the updated algorithm. Repeat steps 2 and 3 for two more cycles (or more, as needed).
5. Finalize: Make a decision on the final Auto Merge Threshold and Clerical Review Threshold.

Before the sample pair review can kick-off, the raw data from the output file should be formatted to make it human readable. The Sample Pair Formatter Excel sheet can optionally be used on the output file.

Note: To obtain the Sample Pair Formatter Excel sheet, please refer to the separately bundled Starter Package configuration files. Detailed instructions on how to use this Excel sheet is included within the sheet itself.

Once the random sample pair spreadsheet is generated and formatted, it is vital to review the sample pairs to see how the algorithm evaluates them. The primary purpose of the review is to assess the confidence of each merge, and modify the thresholds if the scores appear inaccurate. During the review process, it is also important to consider the following:

- The organization should mark each set with a decision as to whether (based on the data available) the records should be considered the same entity.
- It is best to approach this task from a 'human' standpoint as opposed to creating logic to help you achieve a

certain score.

- This is not a data cleaning task.

The ultimate goal by the end of each Sample Pair Review session is to improve the quality of the matches found. It is much easier to identify false positives than false negatives in the pair export. Therefore, it is recommended to start with a wide net and narrow the Match Criteria during tuning.

Note: It is important to set expectations with the client organization that while those implementing the starter package can provide guidance, it is the organization's responsibility to determine which customer records should match and which should not.

The Sample Pair Review process can be a time-consuming task, but is critical in getting the algorithm tuned to meet requirements. Typically, 1,000+ Sample Pairs will need to be reviewed each cycle.

There are several personnel considerations to make during the sample pair review process:

- It is recommended that the consultant be present on-site for the Sample Pair Review sessions. Due to the highly interactive nature of these discussions, being on-site helps facilitate the process.
- Data Owners from different functional areas within the organization may have differing opinions on matching requirements. It may be difficult to get consensus among these various Data Owners.
- Some more sophisticated client organizations may introduce a Data Governance Board that can assist with reaching consensus among the Data Owners.

Other Considerations

It is recommended that the client organization re-engage with the applicable solution consultants post go-live for new Algorithm Tuning sessions when any of the following takes place:

- A new source is added that has different demographic or data quality.
- Substantial increase in False Positives or False Negatives.
- The Entity Size or Task Size metrics change significantly over time.

For more information on Matching Tuning and creating a Match Tuning configuration, see the **Match Tuning** section of the **Matching, Linking, and Merging** documentation.

Working with Customer Data

When developing a Matching strategy, it is important to consider the client organization's data and the potential challenges the algorithm will have to account for.

The following considerations and challenges are commonly encountered when implementing Matching:

- Algorithm Tuning is highly data-dependent, so real production data should be made available for analysis. As an example, start with 1000 records, eventually go to 100000, and in the end, at least 10% of the full data set should be taken in to consideration.
- Solution Consultants should expect delays in receiving customer data. They should work with the client organization early in the process to define the data to be delivered, and push to get it as early as possible.

There may be both technical and process reasons for holding up data delivery. The ETL team may have issues in staging data from the source systems, and the legal or security team may introduce their own delays.

- Solution Consultants should establish a delivery date that the client organization agrees on, and emphasize that delays to that date will delay critical-path tasks. Having access to production data is a critical dependency to starting the Algorithm Tuning tasks.
- Solution Consultants should work with the client organization to determine the level of security needed around the data they provide. Note that the client may be held to a higher degree of security due to the regulation in their industry.
- Before Algorithm Tuning starts, the customer data should be analyzed to determine its quality and characteristics. This analysis will help in determining the baseline algorithm configuration. The Solution Consultant should analyze the data for attribute validity (percent populated), bad (or anonymous) data values populated, and any patterns in the data that may be instructive for Algorithm Tuning. The quality and characteristics of data can differ between sources, so data from each source system should be analyzed.
- Solution Consultants should not expect a client organization to be able to articulate their matching requirements. To arrive at a baseline algorithm configuration, a discussion with the client should occur focusing on what their priorities are for generating Golden Records. The Sample Pair review sessions will help facilitate the discussion around finalizing matching requirements.

Customer Data is required at key points during the implementation:

- 10-100 records for Data Modeling
- 20% of total data volume for Algorithm Tuning
- 100% of total data volume for go-live

Data from all sources must be included:

- Data quality and characteristics can vary from source system to source system, so getting sample from all sources is critical
- Samples of all object types in scope
- Other considerations for sampling data:
  - Data that crosses regions
  - The age of records (recently created records vs records created 20 years ago)
  - Records updated recently vs records not updated in years
  - Data captured through different means (call center, web, mobile, etc.)

Algorithm & Match Codes - Individual Customer

Matching Algorithm

The Individual Matching Algorithm delivered as part of this Starter Package is designed as a match algorithm for Individual Customer solutions. The most common data to have available for matching Individual Customers are First Name, Middle Name, Last Name, Address, E-mails, and Phone numbers

This algorithm consists of four normalizers and four matchers, with an Auto Threshold of 90.0 and a Clerical Review Threshold of 60.0.

For more information on configuring a Matching Algorithm, see the **Configuring Matching Algorithms** section of the **Matching, Linking, and Merging** documentation.

Normalizers

Normalizers are used to standardize values that are being compared. This ensures equal formatting is applied, increasing the accuracy of the comparisons being made. For more information, see the **Decision Table Normalizers** section of the **Matching, Linking, and Merging** documentation.

Person Name Normalizer

The Person Name Normalizer is configured to normalize the corresponding first, middle, and last name attributes (e.g. FirstName, MiddleName, LastName).

For customers with a large, non-English speaking consumer base, it is recommended to normalize accents and diacritic characters. Such characters may not 'play well' with the phonetic encoding of words (e.g. soundex or metaphone3) during the match process.

Address Normalizer

Because it is recommended to model addresses as data containers, the Address Normalizer will need to be configured to normalize data container attributes as defined within the Address component model.

Email Normalizer

Because it is recommended to model emails as data containers, the Email Normalizer will need to be configured to normalize data container attributes as defined within the Email component model.

Phone Normalizer

Because it is recommended to model phone numbers as data containers, the Phone Normalizer will need to be configured to normalize data container attributes.

Matchers

For general information on configuring Matchers, see the **Decision Table Matchers** section of the **Matching, Linking, and Merging** documentation.

Person Name Matcher

The Person Name Matcher is largely left with the default settings. However, an alias table is used to provide an equivalent names table.

Third party sources may be leveraged to build and enhance the equivalent names table within STEP. However, it is recommended that the client's expertise with their customer data be consulted to account for industry or business specific patterns. For example, if a company is based in southwestern United States then their equivalent names table may contain an emphasis on Hispanic name equivalents. Furthermore, equivalent name values may also be added as a result of a match tuning exercise with the client's customer data set.

Regarding middle names, an evaluation exercise with the customer is recommended to review the quality of data they have for customer middle names. It is possible that middle names are not collected from the consumers, or only middle initials are required but rarely provided. In such cases, it is recommended to reconsider the weight of MiddleName.

Note: The names in the equivalent names table provided by the Starter Package target the US market.

Address Matcher

The Address Matcher utilizes default configuration values.

Email Matcher

The Email Matcher utilizes default configuration values.

Phone Matcher

The Phone Matcher utilizes default configuration values.

Rules

When considering match rules, the recommended strategy is to dissect the customer's information into the smallest possible portions of data. These rules should not weigh the sum of *all* the customer's input data, and should instead be split so that it is possible to optimize each rule. Careful analysis of the customer dataset is required in order to determine what combinations of attributes present the best chance of uniqueness.

Three rules are provided to calculate the final score of the Individual Matching Algorithm. According to the configured conditions, if all respective Matchers resolve to true (i.e. scores above 70), then the algorithm will take the highest scoring rule as the final score of the algorithm. The rules are comprised of combinations of each Name score, Address score, Email score, and Phone score. The rules are then standardized to resolve to a value between 0 and 100.

These rules specifically include:

- Name & Address
- Name & Email
- Name & Phone

In this scenario, Name is used in all three rules because it is common for family members who use the same email and/or phone number to live at the same address. Thus email, phone number, or address are not reliably unique by themselves. By adding Name to these rules we are ensuring the uniqueness of the individual's name is taken into consideration, in addition to the other contact information elements.

It is possible to extend a rule by including various other combinations of matchers. This should be considered if there are specific requirements or use cases which requires specific combinations of matchers.

By adding another parameter, you have the ability to identify false positives. Unique identifiers such as Social Security, Passport, or Driver's License Numbers may be used as veto rules to further enhance the quality of the match process.

Other extensions include Date of Birth (DOB), which can be used in combination with other rules to be less strict on equality. For example, name and address runs the risk of a father and son having the same name which would be resolved by considering the date of birth. Additionally, you can accept lower scores of names and addresses, if DOB is equal.

Survivorship

The following survivorship rules are used by the Individual Customer Algorithm:

- **Value: Most Recent**
  - **Attribute / Attribute Group:** Individual - Most Recent
  - **Last Edit Date Attribute:** Last Edit Date - Record
- **Data Container: Most Recent (Emails)**
  - **Business Condition:** DataContainer Survivorship Email
  - **Data Container Type:** Emails
  - **Last Edit Date Attribute:** Last Edit Date - Email
- **Data Container: Most Recent (Phones)**
  - **Business Condition:** DataContainerSurvivorshipPhone
  - **Data Container Type:** Phones
  - **Last Edit Date Attribute:** Last Edit Date - Phone
- **Data Container: Most Recent (Main Address)**
  - **Business Condition:** DataContainer Survivorship Address
  - **Data Container Type:** Main Address
  - **Last Edit Date Attribute:** Last Edit Date - Main Address

Note: Data Containers require their own survivorship rules. Additionally, each Survivorship rule requires a unique Last Edit Date attribute.

Match Codes

For the Individual Customer entity type, three separate Match Codes are being generated. While based on the demographics of the customer record, these Match Codes are composed of: Email, Phone Number, and a combination of Individual Name and Address.

For information on how to configure Match Codes in STEP, see the **Configuring Match Codes** section of the **Matching, Linking, and Merging** documentation.

Email Match Code

The Email Match Code is simply the normalized value of the email address attribute. Both the username (or local part) of the email and the domain are normalized to ensure variations of the same email address (as a result of differing cases or special characters) are accounted for.

It is recommended to add a discernible prefix to each Match Code so the end-user may easily identify what attribute(s) the Match Code is based off of. The email Match Code contains the prefix 'EM~'.

Phone Number Match Code

The Phone Number Match Code is the normalized value of the individual’s phone number attribute. The normalization removes any parenthesis and hyphenation in between numbers. Additionally, the area itself has been removed, leaving only the last 7 digits available as the Match Code. Phone number Match Code is prefixed by 'PH~'.

Address and Name Match Code

The Address and Name Match Code is a combination of elements of the individual’s name and address. For example, the provided Match Code within this starter package is composed of: zip code + the first letter of the individual’s first name + metaphone3 representation of the individual’s last name.

The Address and Name Match Code contains the prefix 'ZINM~'.

Configuration Considerations

It is worth considering the use of the Equivalent Values Lookup Table and Anonymous Value Lookup Table. The Equivalent Value Lookup Table is used by both Match Codes and the Match Criteria in order to ensure that values that mean the same thing are evaluated as such. Equivalent values will score appropriately high, as if the values were actually the exact same.

Example:

- Name: Matt = Matthew

Note: Equivalent Values are only used for person & organization names.

The Anonymous Values Lookup Table is also used by both Match Codes and the Match Criteria in order to ensure that values that are anonymous, or not meaningful, do not contribute to identifying potential duplicates. Determining what these values should be is highly dependent on the organization's dataset.

Typically, these values will be default values that users of a Source System enter when they don't have the correct value, or don't want to enter a value. The actual anonymous values are not included in the baseline build of the Customer MDM configuration.

Examples:

- Phone: 999999999
- Address: DO NOT USE

Algorithm & Match Codes - Organizational Customer

Matching Algorithm

The Organization Matching Algorithm delivered as part of this Starter Package is designed as a match algorithm for Organization Customer solutions. This algorithm is most relevant in a Business to Business (B2B) style implementation and consists of two normalizers and two matchers, with an Auto Threshold of 90.0 and a Clerical Review Threshold of 60.0.

For more information on configuring a Matching Algorithm, see the **Configuring Matching Algorithms** section of the **Matching, Linking, and Merging** documentation.

Normalizers

Normalizers are used to standardize values that are being compared. This ensures equal formatting is applied, increasing the accuracy of the comparisons being made. For more information, see the **Decision Table Normalizers** section of the **Matching, Linking, and Merging** documentation.

Organization Name Normalizer

The Organization Name Normalizer for the Organization Algorithm is configured to normalize only the Legal Name (LegalName) attribute for organization entities.

Address Normalizer

Because it is recommended to model addresses as data containers, the Address Normalizer will need to be configured to normalize data container attributes as defined within the Address component model.

Matchers

For general information on configuring Matchers, see the **Decision Table Matchers** section of the **Matching, Linking, and Merging** documentation.

Organization Name Matcher

The Organization Name Matcher is used by the algorithm to match on organization names. This matcher is left with default settings, however it is recommended that the Organization Name Matcher is tuned to the specifics of the organization.

Address Matcher

The Address Matcher utilizes default configuration values.

When matching organizations, keep in mind that multiple business entities may reside at a singular address. For example, Office Buildings where no suite number is provided.

Rules

When considering match rules, the recommended strategy is to dissect the customer's information into the smallest possible portions of data. These rules should not weigh the sum of *all* the customer's input data (which is likely inefficient), and requires a careful analysis of the customer dataset in order to determine what combinations of attributes present the best chance of uniqueness.

There is only one Rule associated with the Organization Algorithm. This Rule evaluates the scores of the Name and Address matchers. The resulting score is then weighed against the defined algorithm thresholds to determine whether the organization record should be created, auto-merged, or if it requires a clerical review.

Note: Common attributes to match for organizations on include Address, DUNS (D&B Number), and Tax Identifier.

Survivorship

The following survivorship rules are used by the Organization Algorithm:

- **Value: Most Recent**
  - **Attribute / Attribute Group:** Organizations - Most Recent
  - **Last Edit Date Attribute:** Last Edit Date - Record
- **Data Container: Most Recent**
  - **Business Condition:** DataContainer Survivorship Address
  - **Data Container Type:** Main Address
  - **Last Edit Date Attribute:** Last Edit Date - Main Address

Note: Data Containers require their own survivorship rules. Additionally, each Survivorship rule requires a unique Last Edit Date attribute.

Match Codes

For the Organization Customer entity type, three separate Match Codes are being generated. While based on the demographics of the organization record, these Match Codes are composed of the Email, Phone Number, and a combination of Legal Name and Address.

Since initial demographic data for organizations may be limited prior to enrichment (for example, Dun & Bradstreet detailed profile), generating these three Match Codes will help ensure proper identification of organizations during the match process.

For information on how to configure Match Codes in STEP, see the **Configuring Match Codes** section of the **Matching, Linking, and Merging** documentation.

Email Match Code

The Email Match Code is simply the normalized value of the organization's main contact email address. Both the username (or local part) of the email and the domain are normalized to ensure variations of the same email address (as a result of differing cases or special characters) are accounted for.

It is recommended to add a discernible prefix to each Match Code so the end-user may easily identify what attribute(s) the Match Code is based off of. The Email Match Code contains the prefix 'EM~'.

Phone Number Match Code

The Phone Number Match Code is the normalized value of the organization’s main contact phone number. The normalization removes any parenthesis and hyphenation in between numbers. Additionally, the area itself has been removed, leaving only the last 7 digits available as the Match Code. Phone number Match Code is prefixed by 'PH~'.

Organization Address and Name Match Code

The Organization Address and Name Match Code is composed of: the zip code + metaphone3 representation of the organization’s legal name.

The Organization Address and Name Match Code contains the prefix 'ZONM~'.

Algorithm & Match Codes - Contact Person

Matching Algorithm

The Contact Person Matching Algorithm delivered as part of this starter package is designed as a match algorithm for Contact Person records. This algorithm is relevant in a Business-to-Business (B2B) style implementation and consists of four normalizers and four matchers with an Auto Threshold of 90.0 and a Clerical Review Threshold of 60.0.

For further details regarding the Contact Persons use case, see the **Data Modeling** and **Use Case Overview** sections of this documentation.

For more information on configuring a Matching Algorithm, see the **Configuring Matching Algorithms** section of the **Matching, Linking, and Merging** documentation.

Normalizers

Normalizers are used to standardize values that are being compared. This ensures equal formatting is applied, increasing the accuracy of the comparisons being made. For more information, see the **Decision Table Normalizers** section of the **Matching, Linking, and Merging** documentation.

Person Name Normalizer

The Person Name Normalizer is configured to normalize the corresponding first, middle, and last name attributes (e.g. FirstName, MiddleName, LastName).

For customers with a large, non-English speaking consumer base, it is recommended to normalize accents and diacritic characters. Such characters may not 'play well' with the phonetic encoding of words (e.g. soundex or metaphone3) during the match process.

Address Normalizer

Because it is recommended to model addresses as data containers, the Address Normalizer will need to be configured to normalize data container attributes as defined within the Address component model.

Email Normalizer

Because it is recommended to model emails as data containers, the Email Normalizer will need to be configured to normalize data container attributes as defined with the Email component model.

Phone Normalizer

Because it is recommended to model phone numbers as data containers, the Phone Normalizer will need to be configured to normalize data container attributes.

Matchers

For general information on configuring Matchers, see the **Decision Table Matchers** section of the **Matching, Linking, and Merging** documentation.

Person Name Matcher

The Person Name Matcher is largely left to the default settings. However, an alias table is used to provide an equivalent names table.

Third party sources may be leveraged to build and enhance the equivalent names table within STEP. However, it is recommended that the client's expertise with their customer data be consulted to account for industry or business specific patterns. For example, if a company is based in southwestern United States then their equivalent names table may contain an emphasis on Hispanic name equivalents. Furthermore, equivalent name values may also be added as a result of a match tuning exercise with the client's customer data set.

Regarding middle names, an evaluation exercise with the customer is recommended to review the quality of data they have for customer middle names. It is possible that middle names are not collected from the consumers, or only middle initials are required but rarely provided. In such cases, it is recommended to reconsider the weight of MiddleName.

Note: The names in the equivalent names table provided by the Starter Package target the US market.

Address Matcher

The Address Matcher utilizes default configuration values.

Email Matcher

The Email Matcher utilizes default configuration values.

Phone Matcher

The Phone Matcher utilizes default configuration values.

Rules

When considering match rules, the recommended strategy is to dissect the customer's information into the smallest possible portions of data. These rules should not weigh the sum of *all* the customer's input data, and should instead be split so that it is possible to optimize each rule. Careful analysis of the customer dataset is required in order to determine what combinations of attributes present the best chance of uniqueness.

Three rules are provided to calculate the final score of the Contact Person Matching Algorithm. According to the configured conditions, if all respective matchers resolve to true (i.e. scores above 70), then the algorithm will take the highest scoring rule as the final score of the algorithm. The rules are comprised of combinations of each Name score, Address Score, Email Score, and Phone score. The rules are then standardized to resolve in a value between 0 and 100.

- Name & Address
- Name & Email
- Name & Phone

In this scenario, Name is used in all three rules because it is common for family members who use the same email and/or phone number to live at the same address. Thus email, phone number, or address are not reliably unique by themselves. By adding Name to these rules we are ensuring the uniqueness of the individual's name is taken into consideration, in addition to the other contact information elements.

It is possible to extend a rule by including various other combinations of matchers. This should be considered if there are specific requirements or use cases which requires specific combinations of matchers.

By adding another parameter, you have the ability to identify false positives. Unique identifiers such as Social Security, Passport, or Driver's License Numbers may be used as veto rules to further enhance the quality of the match process.

Other extensions include Date of Birth (DOB), which can be used in combination with other rules to be less strict on equality. For example, name and address runs the risk of a father and son having the same name which would be resolved by considering the date of birth. Additionally, you can accept lower scores of names and addresses, if DOB is equal.

Survivorship

The following survivorship rules are used by the Contact Person Algorithm:

- **Value: Most Recent**
  - **Attribute / Attribute Group:** Surv Contact Most Recent
  - **Last Edit Date Attribute:** Last Edit Date - Record
- **Data Container: Most Recent (Main Address)**
  - **Business Condition:** DataContainer Survivorship Address
  - **Data Container Type:** Main Address
  - **Last Edit Date Attribute:** Last Edit Date - Main Address
- **Data Container: Most Recent (Emails)**
  - **Business Condition:** DataContainer Survivorship Email
  - **Data Container Type:** Emails
  - **Last Edit Date Attribute:** Last Edit Date - Email
- **Data Container: Most Recent (Phones)**
  - **Business Condition:** DataContainerSurvivorshipPhone
  - **Data Container Type:** Phones
  - **Last Edit Date Attribute:** Last Edit Date - Phone
- **Reference: Most Recent (Organization)**
  - **Reference Type:** Contact Person Organization
  - **Last Edit Date Attribute:** Last Edit Date - Org Ref

Note: Data Containers require their own survivorship rules. Additionally, each Survivorship rule requires a unique Last Edit Date attribute.

Match Codes

For the Contact Person entity type, three separate Match Codes are being generated. While based on the demographics of the Contact Person record, these Match Codes are composed of the Email, Phone Number, and a combination of Contact Person's Name and Address.

To ensure the algorithm will only match Contact Persons within the same organization, the Match Codes should contain the ID of the reference organization for a given Contact Person. This ensures that only Contact Persons within the same organization will be deduplicated.

For information on how to configure Match Codes in STEP, see the **Configuring Match Codes** section of the **Matching, Linking, and Merging** documentation.

Email Match Code

The Email Match Code is simply the referenced organization's ID and the normalized value of the email address attribute. Both the username (or local part) of the email and the domain are normalized to ensure variations of the same email address (as a result of differing cases or special characters) are accounted for.

It is recommended to add a discernible prefix to each Match Code so the end-user may easily identify what attribute(s) the Match Code is based off of.

The email Match Code contains the prefix 'EM~'.

Phone Number Match Code

The Phone Number Match Code is the referenced organization's ID and the normalized value of the Contact Person's phone number attribute. The normalization removes any parenthesis and hyphenation in between numbers. Additionally, the area itself has been removed, leaving only the last 7 digits available as the Match Code. Phone number Match Code is prefixed by 'PH~'.

Address and Name Match Code

The Address and Name Match Code is a combination of the referenced organization's ID and elements of the Contact Person's name and address. For example, the provided Match Code within this starter package is composed of: the zip code + first letter of the individual's first name + metaphone3 representation of the Contact Person's last name.

The Address and Name Match Code contains the prefix 'ZINM~'.

Configuration Considerations

It is worth considering the use of the Equivalent Values Lookup Table and Anonymous Value Lookup Table. The Equivalent Value Lookup Table is used by both Match Codes and the Match Criteria in order to ensure that values that mean the same thing are evaluated as such. Equivalent values will score appropriately high, as if the values were actually the exact same.

Example:

- Name: Matt = Matthew

Note: Equivalent Values are only used for person & organization names.

The Anonymous Values Lookup Table is also used by both Match Codes and the Match Criteria in order to ensure that values that are anonymous, or not meaningful, do not contribute to identifying potential duplicates. Determining what these values should be is highly dependent on the organization's dataset.

Typically, these values will be default values that users of a Source System enter when they don’t have the correct value, or don’t want to enter a value. The actual anonymous values are not included in the baseline build of the Customer MDM configuration.

Examples:

- Phone: 999999999
- Address: DO NOT USE

Algorithm & Match Codes - Household

Matching Algorithm

The Household Matching Algorithm delivered as part of this Starter Package is designed as a match algorithm for Household entities. This algorithm is most relevant in a Business to Consumer (B2C) style implementation and consists of two normalizers and two matchers, with an Auto Threshold of 90.0 and a Clerical Review Threshold of 80.0. The Clerical review threshold is both higher and tighter to ensure only customer records that are extremely likely to belong to the same household are considered.

It is worth noting that the Household Matching Algorithm utilizes the Link Golden Record solution engine rather than the Merge Gold Record approach. This is to ensure that individual customers that are determined to be of the same household are linked to a separate household entity object rather than merged.

For details regarding the Household entity and its purpose, see the **Data Modeling** section of this documentation.

For more information on configuring a Matching Algorithm, see the **Configuring Matching Algorithms** section of the **Matching, Linking, and Merging** documentation.

Normalizers

Normalizers are used to standardize values that are being compared. This ensures equal formatting is applied, increasing the accuracy of the comparisons being made. For more information, see the **Decision Table Normalizers** section of the **Matching, Linking, and Merging** documentation.

Last Name Normalizer

The Last Name Normalizer uses the Words Normalizer, and only evaluates the Last Name attribute of an individual record. This is because households are referred to only by the Last Name, or household name.

Address Normalizer

Because it is recommended to model addresses as data containers, the Address Normalizer will need to be configured to normalize data container attributes as defined within the Address component model.

Matchers

For general information on configuring Matchers, see the **Decision Table Matchers** section of the **Matching, Linking, and Merging** documentation.

Name Matcher

The Last Name Matcher is largely left with the default settings. An equivalent names look-up table is not utilized for Household entities since last names generally do not have shortened versions or a nickname alternative.

Address Matcher

The Address Matcher utilizes default configuration values.

Rules

When considering match rules, the recommended strategy is to dissect the customer's information into the smallest possible portions of data. These rules should not weigh the sum of *all* the customer's input data (which is likely inefficient), and requires a careful analysis of the customer dataset in order to determine what combinations of attributes present the best chance of uniqueness.

There is only one rule associated with the Household algorithm, which evaluates the scores of the Name and Address matchers. The resulting score is then weighed against the defined algorithm thresholds to determine whether the individual record should be created, auto-linked to an existing record, or it requires clerical review.

The Starter Package matches on Last Name and Address for households. Matching on Last Names may be advantageous in identifying family units within a large population such as a retirement home or a school. However, in some cases a married couple within a household may not have the same Last Name. It may be beneficial to make Address weigh more in scenarios like this.

Conversely, sometimes a large number of customer records with the same Address should not be considered the same household. This can include college campuses, retirement homes, and similar institutions. These scenarios should be handled on a case-by-case basis. However, one possible approach is to utilize an anonymous words table to anonymize aforementioned addresses, eliminating addresses from the matching logic for such records.

Survivorship

The following survivorship rules are used by the Household Algorithm:

- **Value: Most Recent**
  - **Attribute / Attribute Group:** Last Name
  - **Last Edit Date Attribute:** Last Edit Date - Record
- **Data Container: Most Recent**
  - **Business Condition:** DataContainer Survivorship Address
  - **Data Container Type:** Main Address
  - **Last Edit Date Attribute:** Last Edit Date - Main Address

Note: Data Containers require their own survivorship rules. Additionally, each Survivorship rule requires a unique Last Edit Date attribute.

Match Codes

The Household Match Code is generated on the Individual Customer entity type along with the Individual Match Codes. Since a household is identified by its last name, the Household Match Code is composed of: the zip code + metaphone3 representation of the individual's last name.

The Household Match Code contain the prefix 'ZILNM~'.

getSimilarObjects

Business users in external systems throughout the enterprise may need to check if customer records already exist. By using the getSimilarObjects SOAP API call, the business can search to see if similar records exist and prevent duplicates from being created at the point of entry.

Considerations

The getSimilarObjects call employs the matching functionality to produce the potential duplicate results returned in the response. For this reason, certain aspects of the solution must be appropriately paired for the call to function.

For more information, see the **Search Before Create** topic of the **Matching, Linking, and Merging** documentation.

Configuration

The matching configuration may need to be altered slightly to support the getSimilarObjects call including even creating additional Match Codes and Match Algorithms. It is important to examine the current matching landscape to determine the alterations needed.

Determine the Match Algorithm to be used

The Match Algorithm being used to create and manage golden records may not always be the best Match Algorithm to use for the getSimilarObjects call. Consider if the business needs of the external system align with the business needs of the existing governing Match Algorithm.

Who should make the decision when reviewing potential duplicates, a person or a machine? If a person such as a Customer Service Representative in an external system will be asking customers confirming questions about the data that is found, a looser Match Algorithm may be desired. If a machine is making the decision, there is no need for a looser Match Algorithm than the existing governing Match Algorithm because the decision will be made solely on the score.

To configure match codes and match algorithms to support getSimilarObjects, see the get SimilarObject Match Codes and Match Algorithms of the **Search Before Create** topic of the **Matching, Linking, and Merging** documentation.

Sample getSimilarObjects Calls

Sample calls for an Organizational Customer, a Contact Person, and an Individual Customer are included as part of the starter package. Households have been excluded as they are generally not created in systems external to the CMDM system.

Organizational Customer

The Organizational Customer getSimilarObjects call uses the Organization Matching Algorithm to identify potential duplicate Organization Customers. For the incoming values to be compared, an Email, a Phone, or an Organization Name and a Zip Code must be provided.

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://stibo.com/step/ws/step/1.0"
xmlns:ns1="http://stibo.com/step/ws/types/1.0"
xmlns:out="http://www.stibosystems.com/step/outputtemplate">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getSimilarObjectsRequest>
      <ns:accessContext>
        <ns1:userName>StiboCMDM</ns1:userName>
        <ns1:password>StiboCMDM</ns1:password>
        <ns1:contextUrl>step://context?id=Context1</ns1:contextUrl>
        <ns1:workspaceUrl>step://workspace?id=Main</ns1:workspaceUrl>
      </ns:accessContext>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=LegalName</ns1:propertyURL>
        <ns1:value>Saval Foods</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputStreet</ns1:propertyURL>
        <ns1:value>6740 DORSY</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCity</ns1:propertyURL>
        <ns1:value>Elkridge</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputState</ns1:propertyURL>
        <ns1:value>MD</ns1:value>
      </ns:values>
```

-- Continued on next page --

```
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputZip</ns1:propertyURL>
        <ns1:value>21075</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCountry</ns1:propertyURL>
        <ns1:value>US</ns1:value>
      </ns:values>

<ns:objectTypeURL>step://objecttype?id=OrganizationCustomer</ns:objectTypeURL>

<ns:matchingAlgorithmURL>step://matchingalgorithm?id=OrganizationMatchingAlgori
thm</ns:matchingAlgorithmURL>
       <ns:searchThreshold>1</ns:searchThreshold>
       <ns:maxCount>10</ns:maxCount>
     </ns:getSimilarObjectsRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:getSimilarObjectsResponse
xmlns="http://www.stibosystems.com/step/outputtemplate"
xmlns:ns2="http://stibo.com/step/ws/types/1.0"
xmlns:ns3="http://stibo.com/step/ws/step/1.0"
xmlns:ns4="http://www.stibosystems.com/step">
      <ns3:similarObjects>
        <ns2:id>144767</ns2:id>
        <ns2:url>step://entity?id=144767</ns2:url>
        <ns2:title>Organization Customer Test</ns2:title>
        <ns2:type>entity</ns2:type>

<ns2:objectType>step://objecttype?id=OrganizationCustomer</ns2:objectType>
        <ns2:rank>100.0</ns2:rank>
      </ns3:similarObjects>
      <ns3:similarObjects>
        <ns2:id>144575</ns2:id>
        <ns2:url>step://entity?id=144575</ns2:url>
        <ns2:title>(144575)</ns2:title>
        <ns2:type>entity</ns2:type>

<ns2:objectType>step://objecttype?id=OrganizationCustomer</ns2:objectType>
        <ns2:rank>68.625</ns2:rank>
      </ns3:similarObjects>
    </ns3:getSimilarObjectsResponse>
  </S:Body>
</S:Envelope>
```

Contact Person

The Contact Person getSimilarObjects call uses the Contact Person Matching Algorithm to identify potential duplicate records. For the incoming values to be compared, an Organization ID and one of the following must be provided:

- an Email,
- a Phone,
- a First Name, Last Name, and a Zip Code

The GetSimilarContactOrgID must be bound to the match code for Organization IDs to be compared.

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://stibo.com/step/ws/step/1.0"
xmlns:ns1="http://stibo.com/step/ws/types/1.0"
xmlns:out="http://www.stibosystems.com/step/outputtemplate">
<soapenv:Header/>
  <soapenv:Body>
    <ns:getSimilarObjectsRequest>
      <ns:accessContext>
        <ns1:userName>StiboCMDM</ns1:userName>
        <ns1:password>StiboCMDM</ns1:password>
        <ns1:contextUrl>step://context?id=Context1</ns1:contextUrl>
        <ns1:workspaceUrl>step://workspace?id=Main</ns1:workspaceUrl>
      </ns:accessContext>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=FirstName</ns1:propertyURL>
        <ns1:value>Sergio</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=LastName</ns1:propertyURL>
        <ns1:value>Bennett</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputStreet</ns1:propertyURL>
        <ns1:value>525 STATE</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCity</ns1:propertyURL>
        <ns1:value>York</ns1:value>
      </ns:values>
```

-- Continued on next page --

```
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputState</ns1:propertyURL>
        <ns1:value>PA</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCountry</ns1:propertyURL>
        <ns1:value>US</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=EmailField</ns1:propertyURL>
        <ns1:value>campbell@outlook.com</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=PhoneNumber</ns1:propertyURL>
        <ns1:value>5323010724</ns1:value>
      </ns:values>
      <ns:values>

<ns1:propertyURL>step://attribute?id=GetSimilarContactOrgID</ns1:propertyURL>
        <ns1:value>144581</ns1:value>
      </ns:values>
      <ns:objectTypeURL>step://objecttype?id=ContactPerson</ns:objectTypeURL>

<ns:matchingAlgorithmURL>step://matchingalgorithm?id=ContactPersonMatchingAlgor
ithm</ns:matchingAlgorithmURL>
      <ns:searchThreshold>1</ns:searchThreshold>
      <ns:maxCount>10</ns:maxCount>
    </ns:getSimilarObjectsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:getSimilarObjectsResponse
xmlns="http://www.stibosystems.com/step/outputtemplate"
xmlns:ns2="http://stibo.com/step/ws/types/1.0"
xmlns:ns3="http://stibo.com/step/ws/step/1.0"
xmlns:ns4="http://www.stibosystems.com/step">
      <ns3:similarObjects>
        <ns2:id>149102</ns2:id>
        <ns2:url>step://entity?id=149102</ns2:url>
        <ns2:title>(149102)</ns2:title>
        <ns2:type>entity</ns2:type>
        <ns2:objectType>step://objecttype?id=ContactPerson</ns2:objectType>
        <ns2:rank>100.0</ns2:rank>
      </ns3:similarObjects>
    </ns3:getSimilarObjectsResponse>
  </S:Body>
</S:Envelope>
```

Individual Customer

The Individual Customer getSimilarObjects call uses the IndividualMatchingAlgorithm to identify potential duplicate Individual Customers. For the incoming values to be compared, one of the following attributes must be provided:

- an Email
- a Phone
- First Name, Last Name, and a Zip Code

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://stibo.com/step/ws/step/1.0"
xmlns:ns1="http://stibo.com/step/ws/types/1.0"
xmlns:out="http://www.stibosystems.com/step/outputtemplate">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getSimilarObjectsRequest>
      <ns:accessContext>
        <ns1:userName>StiboCMDM</ns1:userName>
        <ns1:password>StiboCMDM</ns1:password>
        <ns1:contextUrl>step://context?id=Context1</ns1:contextUrl>
        <ns1:workspaceUrl>step://workspace?id=Main</ns1:workspaceUrl>
      </ns:accessContext>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=FirstName</ns1:propertyURL>
        <ns1:value>Aarone</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=MiddleName</ns1:propertyURL>
        <ns1:value></ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=LastName</ns1:propertyURL>
        <ns1:value>Kirk</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputStreet</ns1:propertyURL>
        <ns1:value>6770 West Via Tres Casas</ns1:value>
      </ns:values>
```

-- Continued on next page --

```
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCity</ns1:propertyURL>
        <ns1:value>Tuson</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputState</ns1:propertyURL>
        <ns1:value>AZ</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputZip</ns1:propertyURL>
        <ns1:value>85743</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=InputCountry</ns1:propertyURL>
        <ns1:value>US</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=EmailField</ns1:propertyURL>
        <ns1:value>fringilla.est@eu.org</ns1:value>
      </ns:values>
      <ns:values>
        <ns1:propertyURL>step://attribute?id=PhoneNumber</ns1:propertyURL>
        <ns1:value>(310) 401-1771</ns1:value>
      </ns:values>

<ns:objectTypeURL>step://objecttype?id=IndividualCustomer</ns:objectTypeURL>

<ns:matchingAlgorithmURL>step://matchingalgorithm?id=IndividualMatchingAlgorith
m</ns:matchingAlgorithmURL>
      <ns:searchThreshold>1</ns:searchThreshold>
      <ns:maxCount>10</ns:maxCount>
    </ns:getSimilarObjectsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:getSimilarObjectsResponse
xmlns="http://www.stibosystems.com/step/outputtemplate"
xmlns:ns2="http://stibo.com/step/ws/types/1.0"
xmlns:ns3="http://stibo.com/step/ws/step/1.0"
xmlns:ns4="http://www.stibosystems.com/step">
      <ns3:similarObjects>
        <ns2:id>152087</ns2:id>
        <ns2:url>step://entity?id=152087</ns2:url>
        <ns2:title>(152087)</ns2:title>
        <ns2:type>entity</ns2:type>

<ns2:objectType>step://objecttype?id=IndividualCustomer</ns2:objectType>
        <ns2:rank>100.0</ns2:rank>
      </ns3:similarObjects>
      <ns3:similarObjects>
        <ns2:id>151889</ns2:id>
        <ns2:url>step://entity?id=151889</ns2:url>
        <ns2:title>(151889)</ns2:title>
        <ns2:type>entity</ns2:type>

<ns2:objectType>step://objecttype?id=IndividualCustomer</ns2:objectType>
        <ns2:rank>84.8</ns2:rank>
      </ns3:similarObjects>
      <ns3:similarObjects>
        <ns2:id>152081</ns2:id>
        <ns2:url>step://entity?id=152081</ns2:url>
        <ns2:title>(152081)</ns2:title>
        <ns2:type>entity</ns2:type>
    -- Continued on next page --
```

```
<ns2:objectType>step://objecttype?id=IndividualCustomer</ns2:objectType>
        <ns2:rank>71.34</ns2:rank>
      </ns3:similarObjects>
    </ns3:getSimilarObjectsResponse>
  </S:Body>
</S:Envelope>
```

getSimilarObjects Use Cases

The following use cases are examples of how the getSimilarObjects can be used to create unique records.

Business-to-Business (B2B) Use Case - Scenario 1 B2B Acquisitions

Business to Customer (B2C) Use Case - Scenario 1 B2C Preventing Duplications

Loqate

Loqate validations can occur at three different points during the customer record life cycle:

- During Import of Source Records
- After surviving attribute values have been determined for a newly created Golden Record
- When updating existing Golden Records

Note: It is recommended to validate address values upon Import and after Survivorship.

For more general information on Loqate, see the **Loqate Integration** section of the **Data Integration** documentation.

Validate on Import

By calling Loqate during Import, each Source Record will have its address values standardized before matching takes place. Raw values taken from Source Systems commonly include typos or inconsistent formatting. By standardizing these values beforehand, the matching algorithm is able to match records more accurately.

Validate on Survivorship

Validating addresses after survivorship ensures that the surviving address values are standardized. However, because address matching is less reliable when matching raw values taken from Source Systems, it is recommended to validate on Import as well.

Validate Updates to Existing Records

A Data Steward can trigger Loqate for single records or when running a bulk update for a collection. By using either of the maintenance methods listed below, address values can be standardized when records are updated.

Maintenance Validation Methods:

- A 'Run Business Rule' button on a Node Details screen
- Running a Bulk Update configuration (an Address Validation business rule can be configured for bulk updates)

By using Bulk Update, existing records can have their address values standardized whenever their data is updated.

Data Modeling

With the help of this guide, users should be able to model the following common customer data challenges in STEP:

- Individual and Organization Customers
- Individuals as Contact Persons
- Contact Persons in relation to Organization Customers
- Addresses and Contact Information specific to a Customers and Contact Persons

Note: Additional data may very well be applicable to include in a STEP CMDM solution, but is not covered by these guidelines

Customer Object Types

What is a Customer?

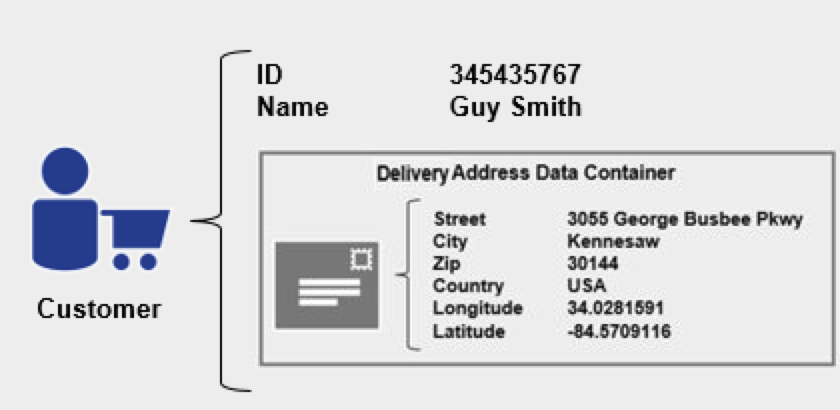
A customer is any individual or organization that purchases from a business. That said, any individual or organization could potentially interact with a business in another capacity (e.g. as a supplier or employee). Because of this, ‘Customer’ is defined as a role an organization or an individual has in the interaction with a business.

However, businesses do not typically manage information about individuals and organizations in a shared model across vastly different business processes, like procurement (supplier) and sales (customer), for simplicity’s sake. To account for this, ‘Customer’ is used as a catch-all entity type that owns all information about individuals and organizations. For more about customer types, see details about contact persons below.

Individual Customer

For an individual customer, the typical customer attributes are:

- First Name
- Last Name
- Addresses
- Date of Birth
- Gender
- Phone Numbers
- Fax numbers
- Emails
- Social Media IDs
- Loyalty Card Information
- Government IDs (social security, passport, driver's license)

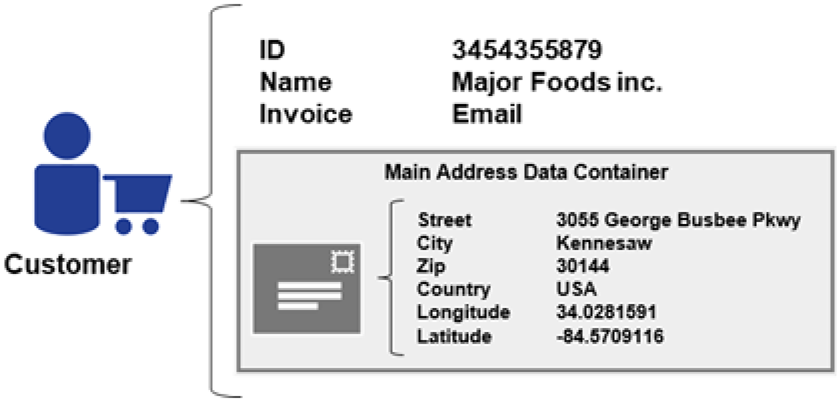


Organization Customer

For an organizational customer, the typical customer attributes are:

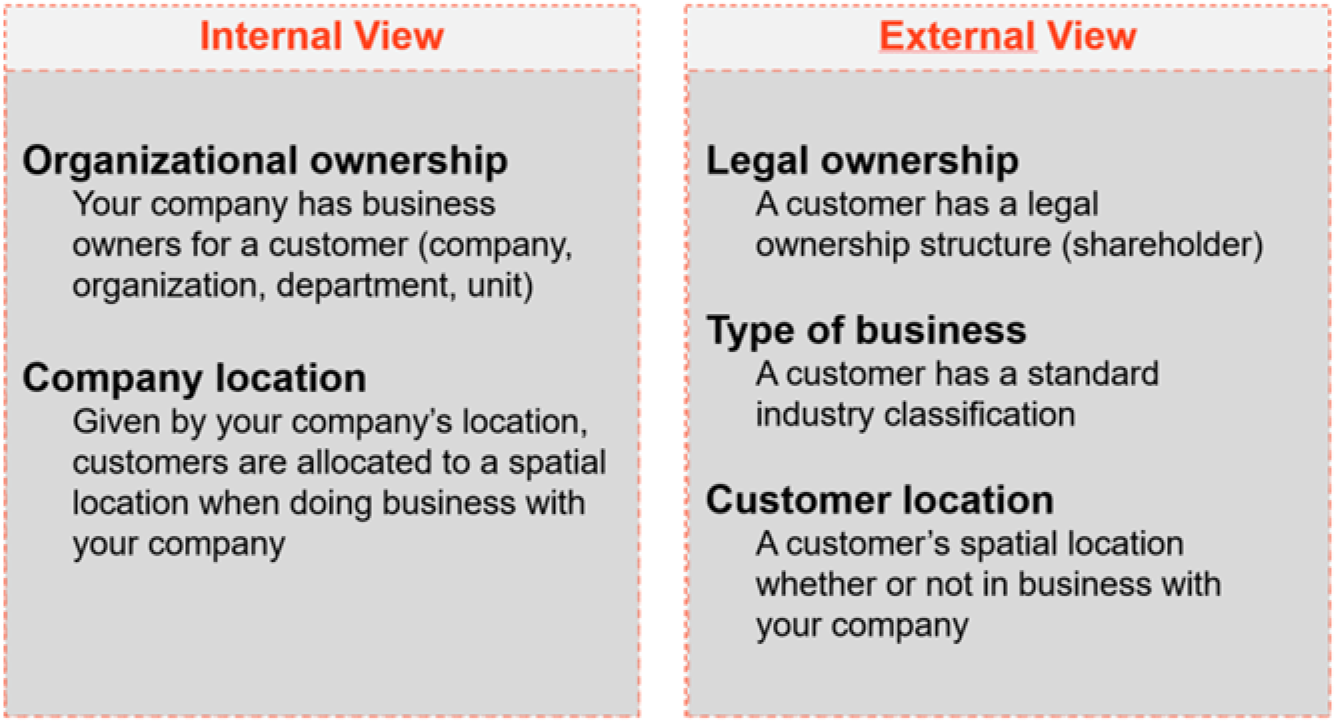
- Legal Name of the Organization
- Date of Formation
- Address
- Contact Information
- Main Phone
- Fax
- Main Email
- Website URL
- GLN / ILN
- DUNS Number
- Tax Identification Number
- Risk Category

Organization customers are often organized in hierarchies. See section about organization hierarchies below:



Customer Hierarchies

The primary purpose for a customer hierarchy in CMDM is for proper grouping of customers. There are two avenues for application: internal view and external view.



Internal View

Organizational ownership
Your company has business owners for a customer (company, organization, department, unit)

Company location
Given by your company's location, customers are allocated to a spatial location when doing business with your company

External View

Legal ownership
A customer has a legal ownership structure (shareholder)

Type of business
A customer has a standard industry classification

Customer location
A customer's spatial location whether or not in business with your company

With internal view, organizational ownership shows business owners as the customers and allocate customers to actual locations when doing businesses with your company. This affords a view from your own company's perspective as well as the ability to group customers according to your internal structure.

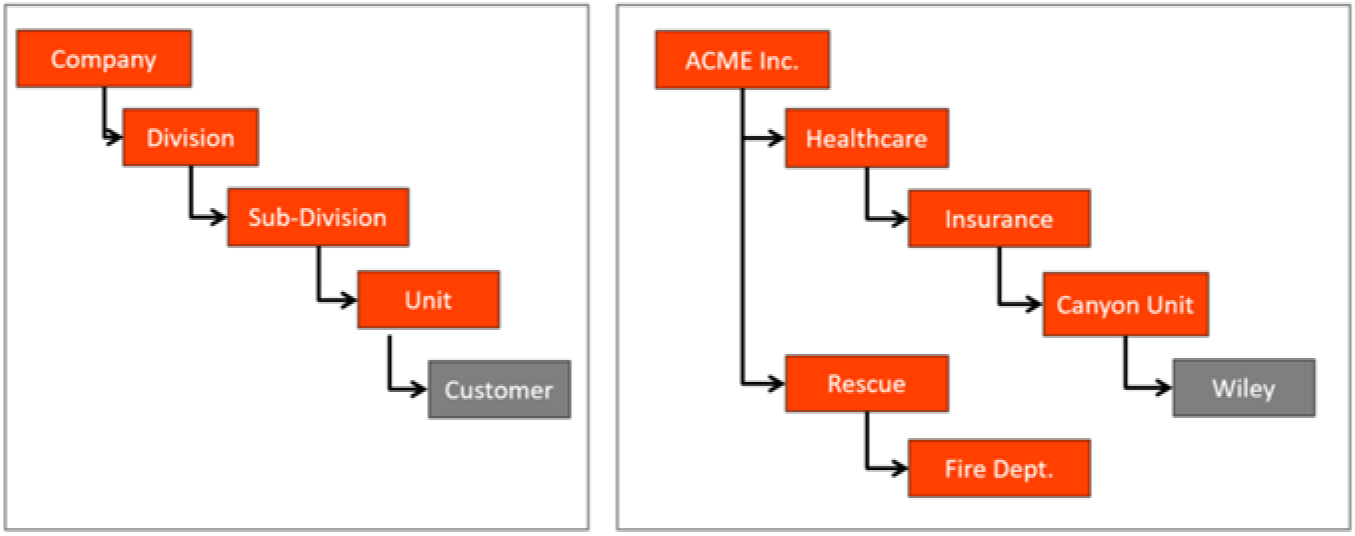
With external view, the customer has a legal ownership structure, works as a standard industry classification, and it shows the customer location, without regard to the company's location. This affords a view on customers from their company's perspective.

Hierarchy Benchmark

When considering best practices in constructing customer hierarchies, customers must pass the 'is grouped by' test. This is a general analysis of grouping based on common characteristics. Some examples of characteristic groupings are as described in the following sections.

Organizational Characteristics

Organizational hierarchies may be characterized by customers that are owned by the same organizational unit, such as with a Financial or Sales organization. Organizational hierarchies are typically used by the business to organize customers according to the organizational setup of their company (division, sales organization etc.) and may also be used in sales transactions, revenue reporting, and/or commissioning. The same customer may also be referenced in different locations within this hierarchical structure.

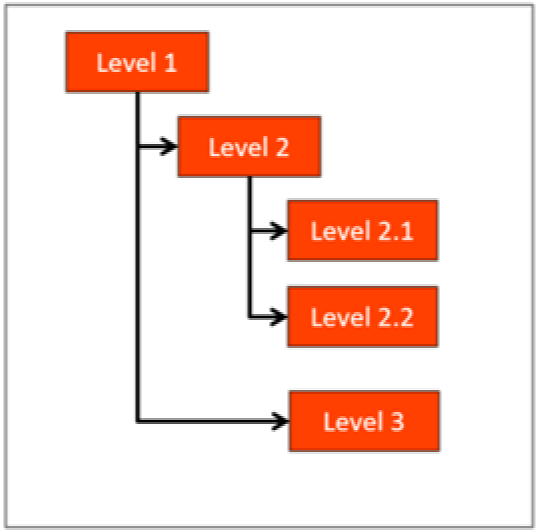


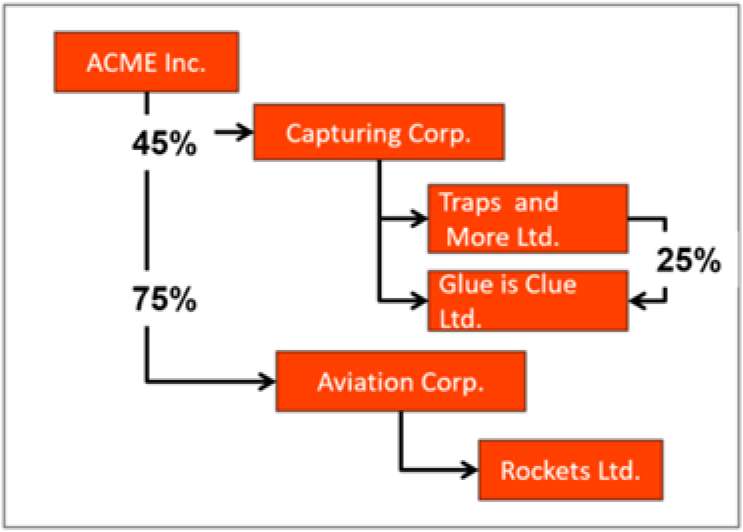
A division is regarded as an organizational unit for sales and distribution with a specific product portfolio to a customer. For every division, customer-specific agreements can be made like for partial deliveries, pricing, and terms of payment. A division can create an exclusion list of products for other divisions which can only be sold to a customer from that particular division.

In large organizations, one or more divisions may be responsible for a customer, such as if the company is selling cleaning equipment such as disinfectant dispensers (healthcare division) and laundry machines (textile care division) to the same hospital.

Legal Characteristics

Legal hierarchies may be characterized by customers that are owned by the same legal entity. Legal hierarchies are typically used by the business to categorize customers according to legal ownership, i.e. between holdings and subsidiaries and/or shareholders. A common usage is for legal reporting and analytics such as whitespace, risk analysis and credit checks (ex. roll up from subsidiary to holding). The same customer is not typically placed under different locations.



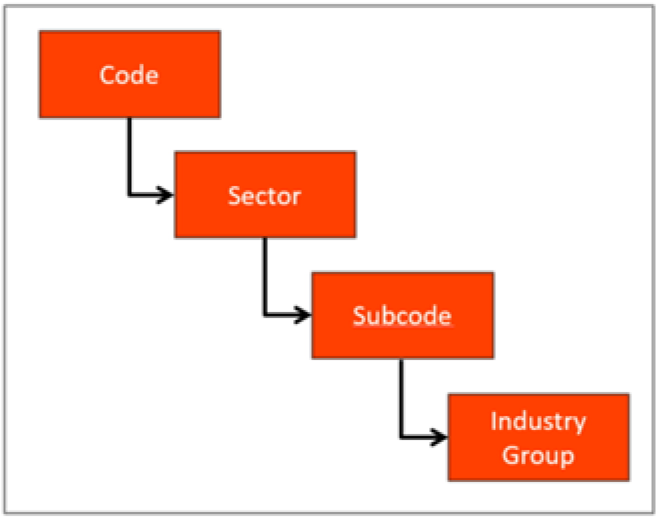


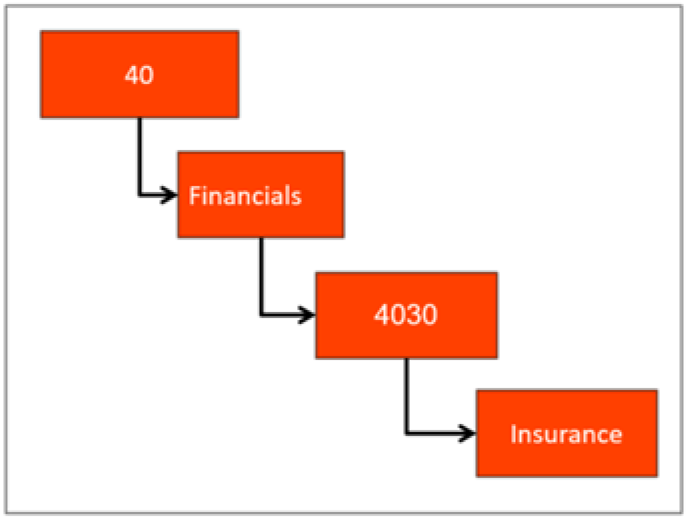
The above example illustrates a legal hierarchy for ACME Inc.

Industry Characteristics

Industry-specific hierarchies may be characterized by customers that are conducting the same type of business. Industry hierarchies are typically used by the business to categorize customers by type of business using the Standard Industry Classification (SIC). This hierarchy may be used for analytics, i.e. customer/ market segmentation (customer allocated to industry, organizational unit allocated to industry).

The same customer is not typically placed under different locations.





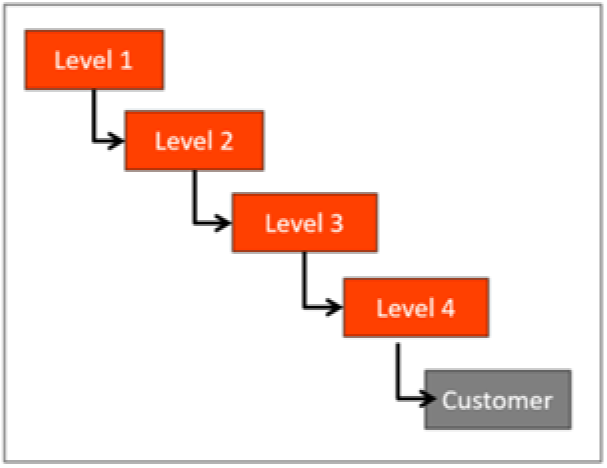
Companies of the same industry are assigned the same SIC code. For example, General Motors, Ford Motor Corporation, Honda, and Toyota were all assigned the SIC code of 3711.

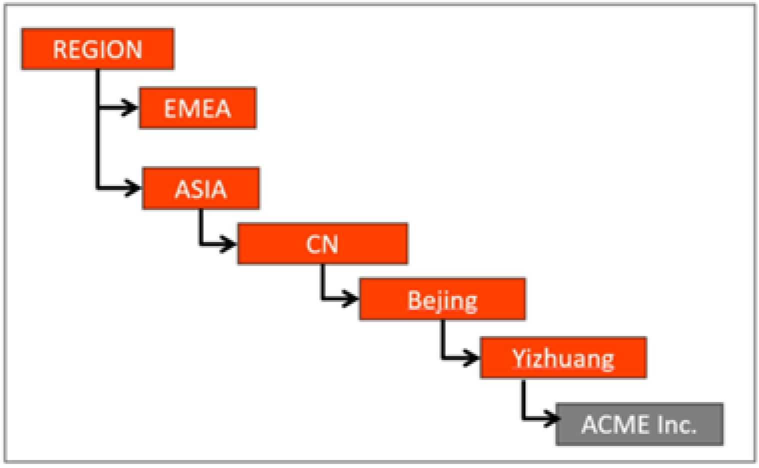
Note: Entities have limited functionality when using in combination with Classifications:

- Entities are not shown as 'sub-products' (or children) of the Classification node
- Attribute values cannot be inherited from the Classification node to the Entity object.
- Classification references to Entities are not inherited from parent Classification to child Classification.

Spatial Location

Location-based hierarchies may be characterized by customers that are allocated at the same spatial location. Location hierarchies are typically used by the business in sales transactions (ordering address), delivery planning (routing), regional revenue reporting, whitespace analysis, segmentation (demographics). This hierarchy may mix different customers under the same location.





In this example, the hierarchy is purely spatial:

- The Region contains lower categories called 'EMEA' and 'ASIA'
- Below 'EMEA' you'd find countries of Europe and Middle East
- Below 'ASIA,' you would have countries like 'CN,' 'JPN,' etc.
- Below 'CN' you could have provinces like 'Beijing' and cities like 'Yizhuang,' etc.
- Below the 'Yizhuang' node, you have all customers which are allocated on this location

When grouping customers, consider what the usage of the hierarchy is (organizational, legal, spatial or other). Furthermore, hierarchy names should be designed to organize customers used in that specific context.

Customer hierarchies are a way of structuring customer data so that the data is related to and connected to other objects. This isn't the same structuring as with classifications for products, nor will it be the same structure as a web hierarchy where each sales channel has their own web sites. Finally, customer hierarchies are not structured for searching, sorting, and browsing data, though these can be built to apply to customer hierarchies.

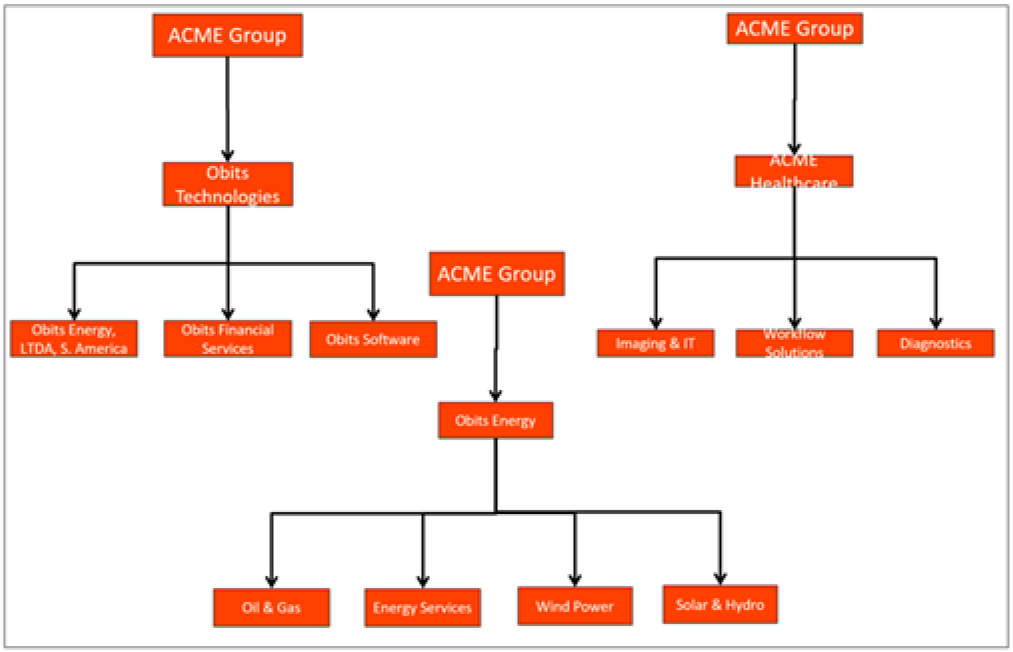
ACME Group Company Hierarchy

Currently, Acme Group consists of fourteen daughter companies across the Americas region. Among these, the relationships across sister companies may or may not exist. In each of these companies there may also exist subsidiaries which operate independently of the each other and of the parent company. These subsidiaries are a combination of acquisitions and spin-offs of individual divisions / business units:

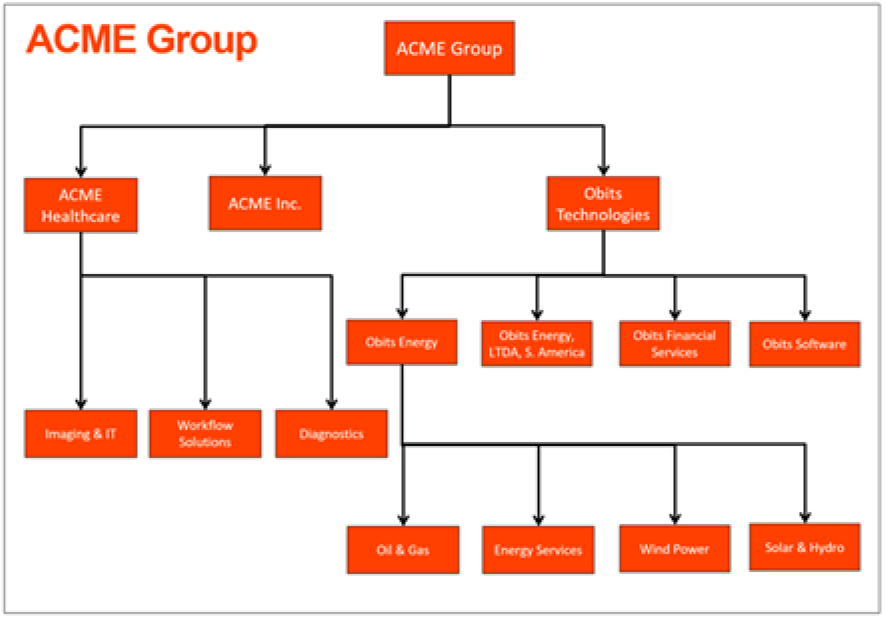
- **ACME Group** - Parent holding company with multiple subsidiary
- **ACME Inc** - Apparel Manufacturer and Distributor
- **ACME Healthcare** - Manufacturer and Distributor of healthcare equipment and supplies
- **Imaging & IT** – Focus on healthcare imaging systems
- **Workflow Solutions** – Focus on business workflow solutions either as a standalone solution or to compliment an ERP system.
- **Diagnostics** – Focus on healthcare chemical diagnostic supplies
- **ACME Healthcare Supplies LTDA, Brazil** - Healthcare Supply Division headquartered in Brazil and caters to South America
- **Obits Technologies** – Subsidiary focusing on enterprise software
- **Obits Financial Services** – Focus on financial sector: banking, insurance, wealth management
- **Obits Energy Innovations** – Think tank for innovations with a focus on energy generation, usage, and consumption
- **Wind Power** – Leader in wind turbine technology
- **Solar & Hydroelectric** – Leader in alternative energy solutions for businesses and municipalities
- **Oil & Gas** – Focus on technology to safely transport crude oil and natural gas
- **Energy Service** – Energy Solutions Professional Service offerings
- **Obits Energy LTDA, Brazil** – Energy division headquartered in Brazil and caters to South America

Walter is an Level I analyst for Acme Group, the primary holding company of Acme Inc and all daughter companies and subsidiaries. In alignment with a new initiative, one of Walter's tasks is to assist with the quarterly reporting for Acme Group. To do so, Walter requires running consolidated (roll-up) reports to provide an aggregate view of all of Acme Group's holdings and investments.

The challenge that Walter faces today is due to the growing number of holdings and subsidiaries over time, each individual company may have implemented their own accounting solutions to track various performance metrics and KPIs. As such, each company has different views of the truth this makes it very difficult to track and report on the relationships between daughter-to-subsidiary, daughter-to-daughter, subsidiary-to-subsidiary companies.

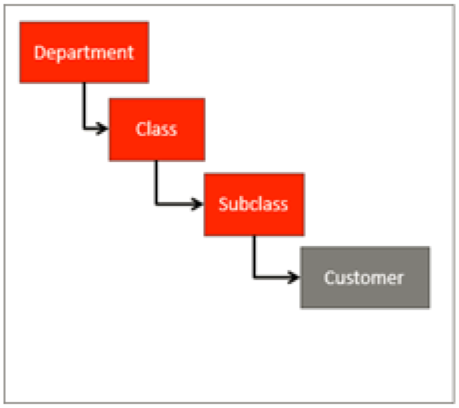


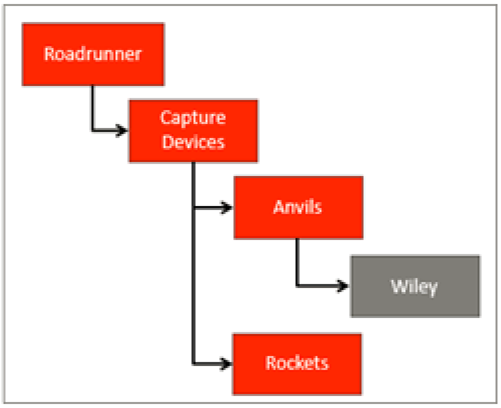
Every quarter, Walter is tasked with generating a quarterly financial report broken down by industry and subsidiaries. However, with no centralized view of the corporate hierarchy and since finances are maintained in separate, isolated ERPs based on industry and divisions. Walter is operationally challenged due to lack of governance.



However in using CMDM, Walter is now able to utilize the hierarchy maintenance tools to visualize and maintain a golden hierarchy which models the complete company hierarchy for Acme Group. By taking in consolidating the various information provided by each company's source system, CMDM may now establish the proper relationships between companies within Acme Group.

This golden hierarchy information may be synchronized with Acme Group's accounting application so Walter now has an accurate and complete view of all holdings and daughter companies. Walter is able to confidently select which subsidiaries to be included for various roll-up reports to drive Acme Group's quarterly analysis.

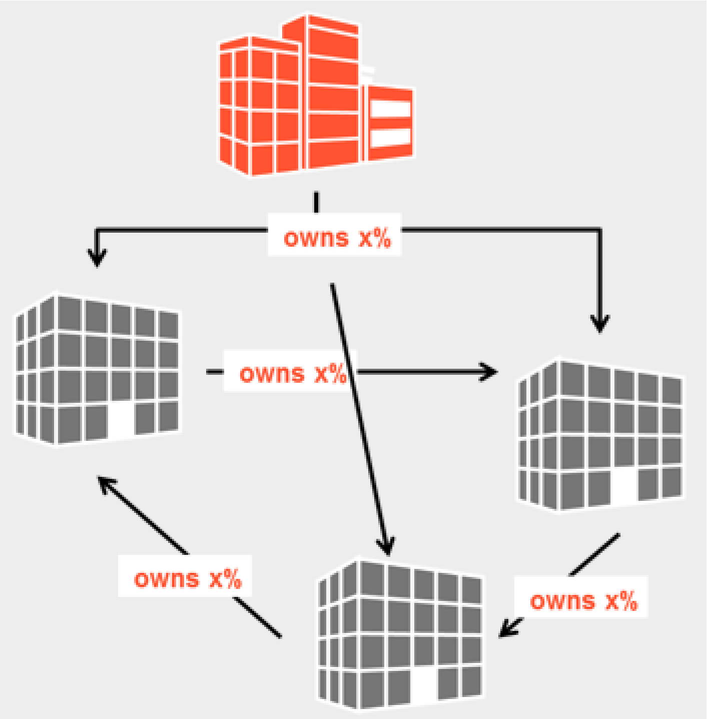




Best Practices for Hierarchies

Referenced Based Hierarchy

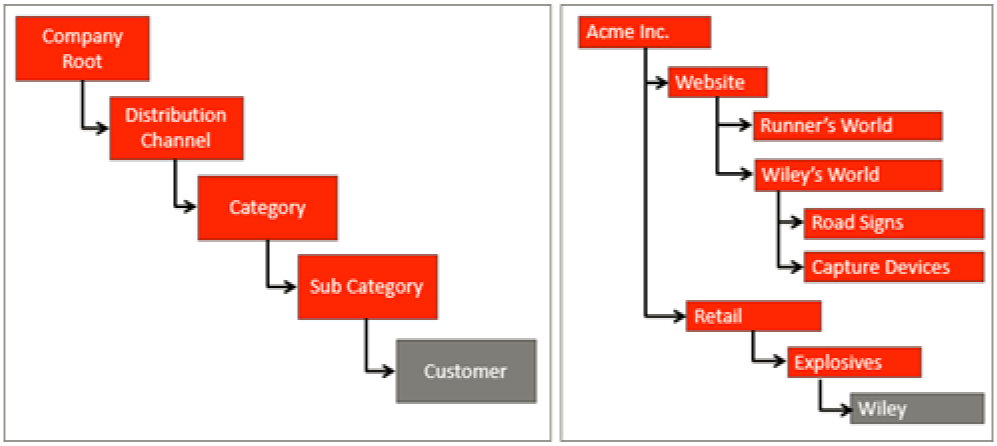
Referenced based hierarchies are structured to where customers reference other customers. Customers that are assigned to higher customer levels specified by the reference type used to appropriate attribution. One benefit of this hierarchy over the parent-child hierarchy is that reassigning a customer doesn't not require related customers to have to be moved, though the relationship change does apply to them.



Web Hierarchy

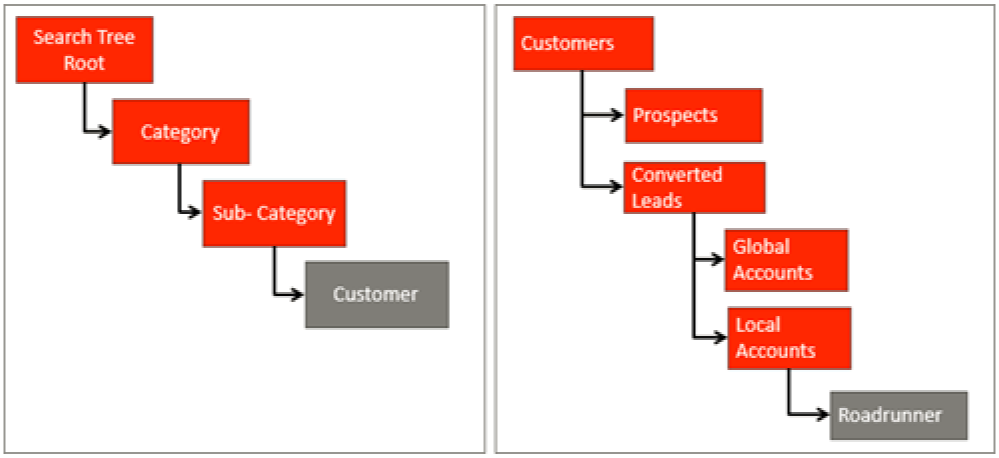
It is recommended to keep the web or sales hierarchy for products clean from customer data. A web or sales hierarchy for products is not intended to be customer specific but rather to represent the product portfolio intended for a specific sales channel.

Customers are typically assigned to a distribution channel in ERP in sales transactions. The distribution channel is an entity through which sales items (products or services) reach customers. They may include wholesale, retail and direct sales. Sales organization is the legal entity that structures the company according to its sales requirements (selling unit) and defines the basic sales and distribution structure. A sales organization is responsible for selling products and services.



Search Hierarchy

Customer hierarchies is not structured a searching, sorting, or browsing hierarchy. If necessary these structures may be built in addition to customer hierarchies.

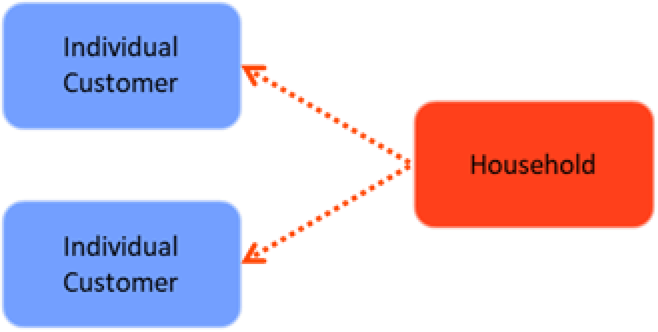


What is a Household?

A household is a collection of individuals, which is typically a family, living in the same home. Each member of the household is considered an individual customer.

The purpose of these household constructs are to help businesses target customers that share a household either as a family unit or as individual customers. This allows the company to limit extraneous mailing campaigns and aggregate buying patterns.

In the CMDM Starter Package, Households are discovered by matching and created via Link Golden Records.



Accurately identifying the correct members of a household comes with natural limitations, as the definition of household is quite vague and vary from case to case. The data available to determine if specific people are actually accurately identified as part of the same household is often rather sparse.

Address and last name are the two most common matching criteria used to determine households. For more information, see the **Algorithm & Match Codes - Household** section of this documentation.

This approach is intended for situations where accuracy must aim to be good but both false positives and false negatives can be accepted, such as with the case of analytics and mailing campaigns.

Handling Composite Records with Data Containers

A customer record comprises multiple objects. It may have multiple addresses, emails, and phone numbers, which are all considered part of the record by users, as well as surrounding systems.

This is not to be confused with hierarchical structures of separate entities with each containing their own life cycle.

Examples of separate entities may include:

- contact persons for organization customers
- store branches
- warehouse locations

Such records are complex structures in themselves that may even reference or be referenced by other entities.

For more information, see the **Data Containers** topic in the **System Setup** documentation.

Addresses

An address comprises several attributes that define a location. STEP offers specialized functionality around addresses, such as address verification via the Address Component Model.

Modeling addresses using data containers makes it possible to manage more than one address on the same customer / contact person. Data containers simplify the data modeling by providing a reusable address definition across multiple object types. When displayed in Web UI, addresses modeled using data containers can appear as one formatted value, despite being made of several attributes.

Addresses have metadata associated with them via the Address Component Model, the CASS address component model, and potentially solution specific meta data.

Email Addresses and Phone Numbers

Email addresses and phone numbers should be modeled as data containers, as it is typical to have multiple phone numbers and email addresses for the same customer or contact person. It is also common to have meta data associated to each phone number or email address.

Handling Confidential Information

Credit Card Information

Credit Card information should not be stored in or pass through STEP due to compliance requirements. It is simpler to store 'Recurring Charge Subscription IDs' when integrating to third party services.

Bank Account Information

Handling bank account information does not typically require more security than name and address information.

Privileges

STEP Privileges to control who has access to what attributes and/or attribute groups.

Security

This guide does not cover information security in STEP in a broader sense, like infrastructure recommendations, encryption strategies, etc.

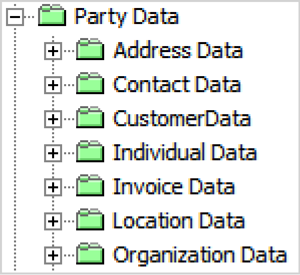
Note: The sample data provided in the Starter Package provides only main addresses. There are no delivery addresses.

System Setup

The following section details what attributes are used to create the data model. For information on everything in the system setup, see the **System Setup** documentation.

Attribute Groups

Attributes should be organized into domain specific attribute groups. Typically, these domains include:



- Party Data
  - Address Data
  - Contact Data
  - CustomerData
  - Individual Data
  - Invoice Data
  - Location Data
  - Organization Data

- Entity Type: **Address Data Containers**; Domain Folder: **Address Data**

Attribute Name	Attribute ID
Formatted Address	CalcFormattedAddress
Country ISO Code	CountryISOCode
City	InputCity
Country	InputCountry
State	InputState
Street	InputStreet
Zip	InputZip
Latitude	Latitude
Longitude	Longitude
Quality	Quality

Attribute Name	Attribute ID
Quality Index	QualityIndex
Standardized City	StandardizedCity
Standardized Country	StandardizedCountry
Standardized Country ISO Code	StandardizedCountryISOCode
Standardized Formatted Address	StandardizedFormattedAddress
Standardized State	StandardizedState
Standardized Street	StandardizedStreet
Standardized Zip	StandardizedZip
Validation Hash	ValidationHash
Validation Integration Status	ValidationIntegrationStatus
Validation Response	ValidationResponse
Validation Time	ValidationTime

- Entity Type: **Individual Customer**; Domain Folder: **Individual Data**

Attribute Name	Attribute ID
Birth Date	BirthDate
CalcAQI	CalcAQI
CalcHouseholdMembers	CalcHouseholdMembers
Calc Individual Name	CalcName
Name on Collection	CalcNameCollection
Creation Date	CreationDate
Credit Limit	CreditLimit
Deactivated Record	DeactivatedRecord
First Name	FirstName
Gender	Gender

Attribute Name	Attribute ID
Golden Record ID	GoldenRecordID
Good Person Flag	GoodPersonFlag
Income	Income
Income Update Date	IncomeUpdateDate
Last Name	LastName
LastUpdated	LastUpdated
Matching Source	MatchingSource
Middle Name	MiddleName
Nationality	Nationality
Past Due Days	PastDueDays
Past Due Dollars	PastDueDollars
Risk Flag	RiskFlag
Sibling Delinquency Flag	SiblingDelinquencyFlag

- Entity Type: Organization Customer; Domain Folder: Organization Data

Attribute Name	Attribute ID
Account Type	AccountType
CalcAQI	CalcAQI
Name on Collection	CalcNameCollection
CompanyVATNumber	CompanyVATNumber
Creation Date	CreationDate
Deactivated Record	DeactivatedRecord
Employee Size	EmployeeSize
Golden Record ID	GoldenRecordID
LastUpdated	LastUpdated

Attribute Name	Attribute ID
Legal Name	LegalName
NAICS Code	NAICSCode
Revenue Size	RevenueSize
SIC Code	SICCode

- Customer Source System Reference Type

Attribute Name	Attribute ID
Source Record ID	SourceRecordID

- Entity Type: **Household**

Attribute Name	Attribute ID
Last Name	LastName

Note: Data container types should be created under the domain with which they are used.

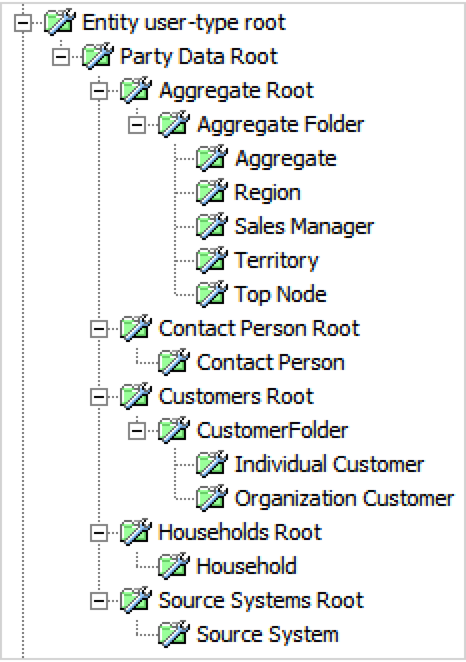
These attributes can be used across different object types, like where Individual and Organizational Customer records, as well as Contact data on Individual Customer, Organization Customer, and Contact Person records.

In accordance with the general guidelines for organizing attributes in attribute groups, separate attribute groups should also be created for:

- Displaying in Web UI, to ease Web UI component setup.
- Each survivorship rule, to provide a clear definition of survivorship rules.
- Each data quality policy, to provide a clear overview of which attributes are part of each data quality metric.
- Each Object Type for each export definition, to provide a clear definition of what is exported to where.
- In accordance to the guidelines around privilege configuration.
- Each Object Type, to describe what is included in data profiling, to ease the profiling configuration.

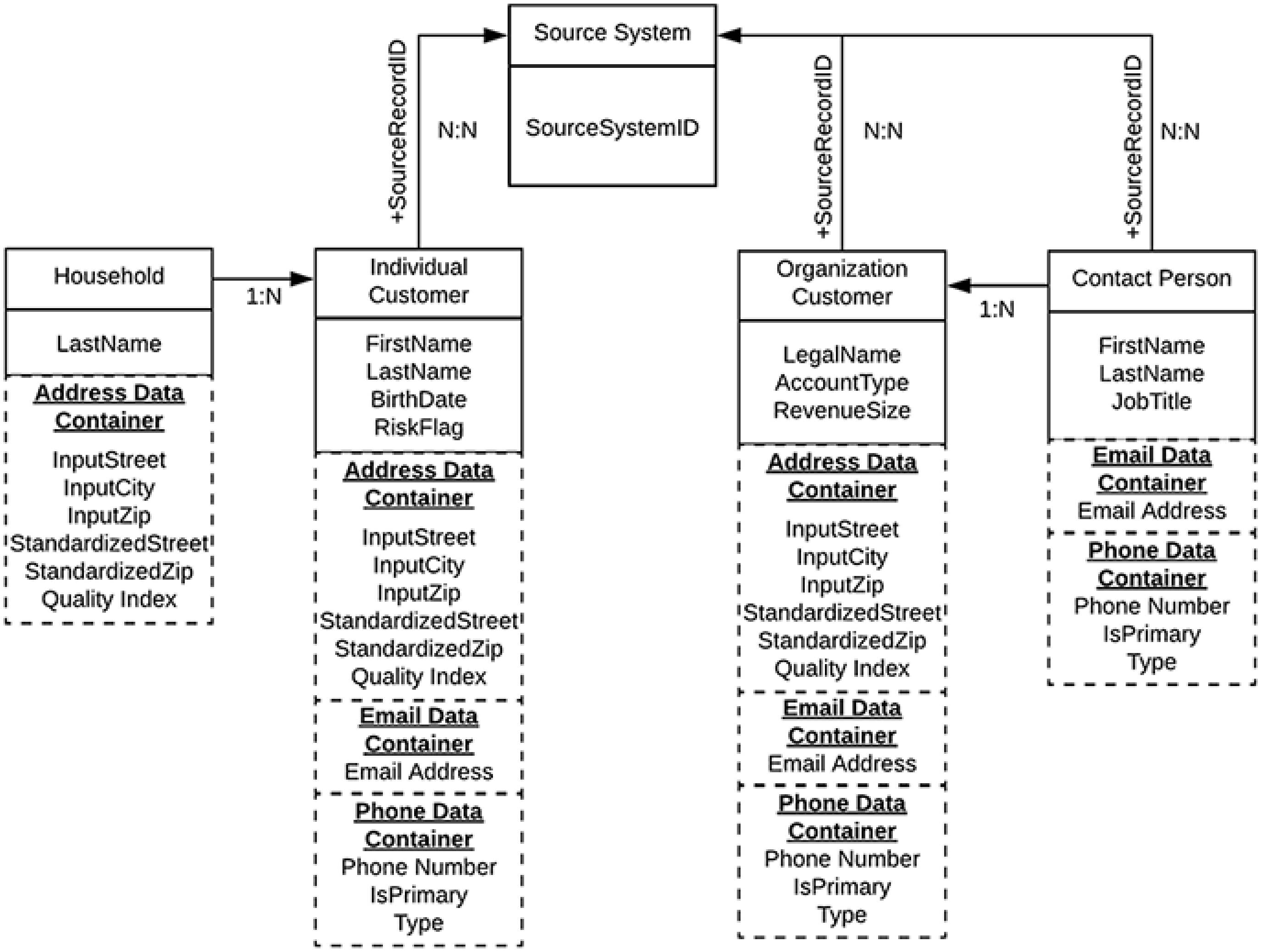
Object Types

Domain Folder constraints are less important than hierarchy constraints, as it typically does not require a user to select a folder. Due to this, it is recommended to limit complexity of Domain Folder Object Types. To preserve a basic integrity of data, it is recommended to create separate Object Types per data domain.



- Entity user-type root
  - Party Data Root
    - Aggregate Root
      - Aggregate Folder
        - Aggregate
        - Region
        - Sales Manager
        - Territory
        - Top Node
    - Contact Person Root
      - Contact Person
    - Customers Root
      - CustomerFolder
        - Individual Customer
        - Organization Customer
    - Households Root
      - Household
    - Source Systems Root
      - Source System

The collected data model is as follows. Each of the dotted boxes is a data container type.



Entity Super Types

STEP uses four types of object super types to model data: products, classifications, assets, and entities. Entities are the suggested object for modeling customer, contacts, and household data. Some benefits of entity object types include:

- May be configured as globally revisable, which requires no approvals or workspace revisions, and instantaneous updates.
- References can be configured as parent-child relationships.
- Support for Matching and Merging.
- Support for Dun & Bradstreet integration.
- Support for Experian Email Validation integration.
- Support for Policy Data Monitoring

Some limitations on using entities are that they do not support inheritance, translations, specification attributes, and specification data container types.

For more information on entities in STEP, see the **Entity Maintenance** topic in the **Getting Started** documentation.

Domain Folders

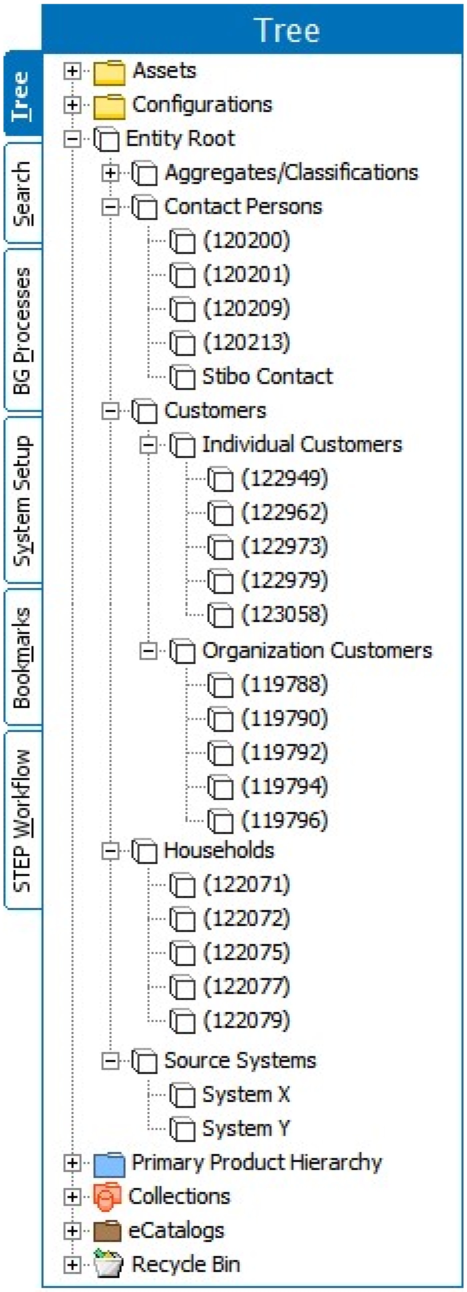
In STEP, any Entity must have a parent Entity. In a CMDM solution, this relationship does not provide any semantic meaning. Therefore, very little effort should be put in to organizing parent/child relationships.

The recommended approach is to create 'Domain Folders.'

For the purpose of traceability in the Workbench tree and when searching for records, it is recommended to split records governed by different matching algorithms into separate domain folders.

Source System Entities

Place all source systems into one folder as there are often very few to consider.



- Assets
- Configurations
- Entity Root
  - Aggregates/Classifications
  - Contact Persons
    - (120200)
    - (120201)
    - (120209)
    - (120213)
    - Stibo Contact
  - Customers
    - Individual Customers
      - (122949)
      - (122962)
      - (122973)
      - (122979)
      - (123058)
    - Organization Customers
      - (119788)
      - (119790)
      - (119792)
      - (119794)
      - (119796)
  - Households
    - (122071)
    - (122072)
    - (122075)
    - (122077)
    - (122079)
  - Source Systems
    - System X
    - System Y
- Primary Product Hierarchy
- Collections
- eCatalogs
- Recycle Bin

Dimensions and Contexts

Dimension specific attribute values and references should not be used in a CMDM solution. However, a language dimension can be used for the purpose of translating data definitions as Attribute names, Reference Type names, Object Type names, and Data Container Type names in Web UI and workbench. This also means that the use of multiple Contexts only allows for translation of data definitions in the UI.

ID Patterns

Use auto-generated STEP IDs for Entities and Data Containers managed by STEP. For information on auto IDs, see the **Autogenerate Using Name Pattern and ID Pattern** topic of the **System Setup** documentation.

Important Considerations for CMDM Data Modeling

These Implementation Guidelines assumes that customers, contact persons, and organizations are created and maintained by the Merge Golden Record solution. Households are created and maintained by using the Link Golden Record solution.

Data Stewardship

Data stewardship in a CMDM application is performed by two distinct roles: data stewards and business users. Each role requires the configuration of different Web UI screens to perform certain tasks. In this topic, these two roles (and the tasks performed by each) will be defined. Web UI configuration guidelines will also be provided.

Note: When configuring any Web UI, consult the Web UI documentation. This documentation contains details on how to add various components of the Web UI.

Role of the Data Stewards

Experienced data stewards must be provided with the proper tools to ensure data quality.

These tasks include:

- Clerical Review - Data Stewards can review potential duplicates.
- Manually maintaining records - Data Stewards update and edit records as needed.
- Monitoring / Data Quality - Data Stewards should review customer data quality in the Web UI, and monitor Source system performance.

Data Steward Tasks

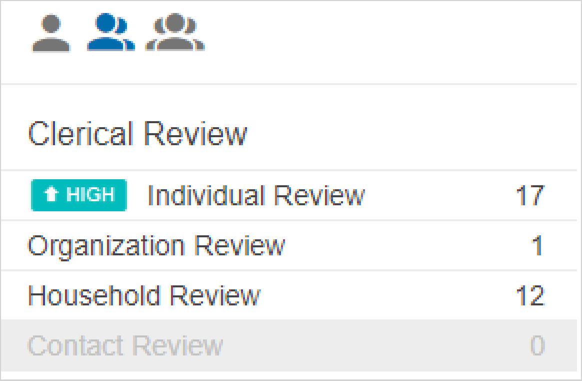
Typically, Data Stewards will need to perform the following tasks: clerical review, ad-hoc data stewardship, data governance, hierarchy maintenance, and attribute maintenance.

Note: The online help topics for these tasks are extensive and cover configuration and typical use.

Clerical Review

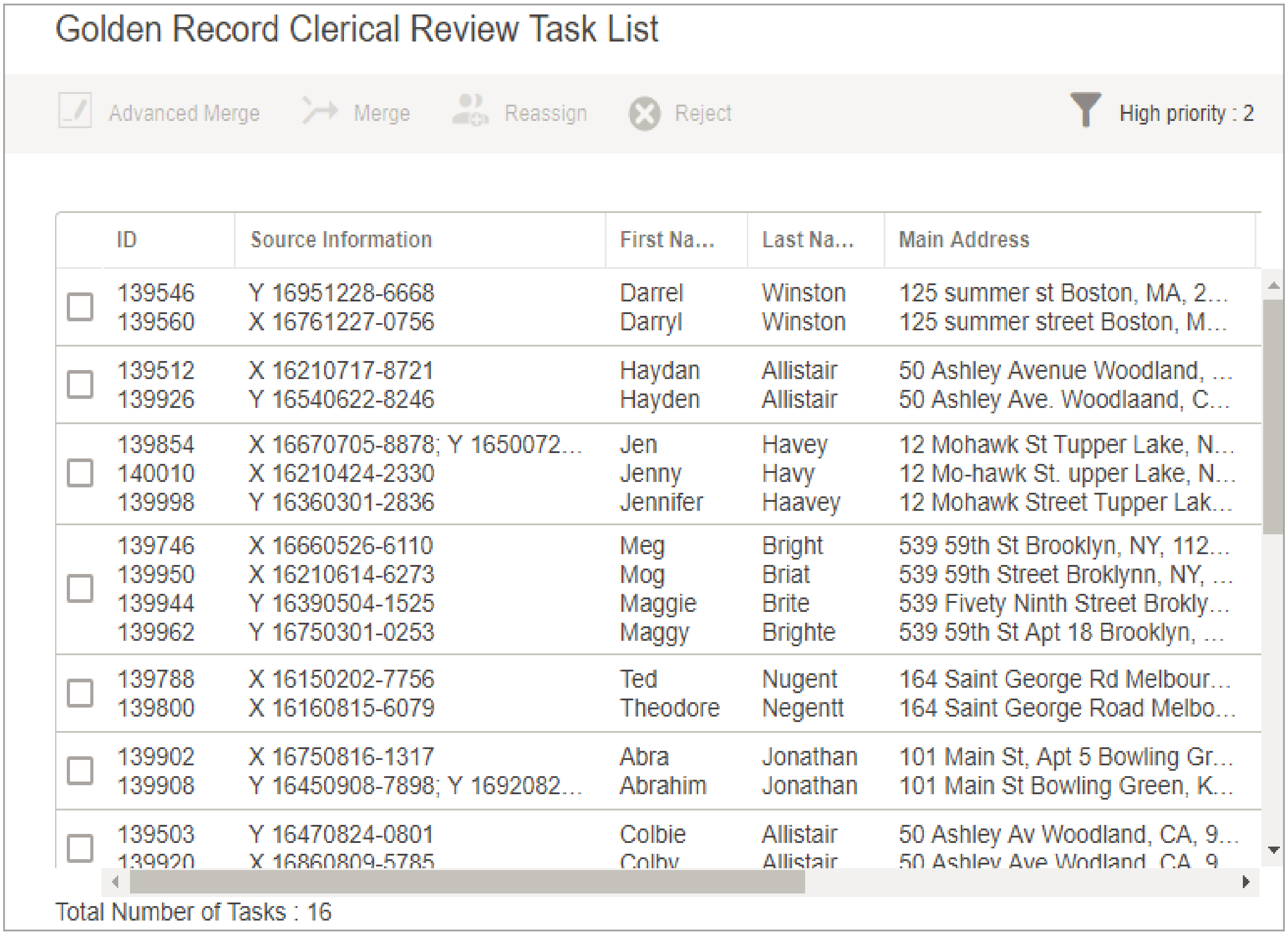
Clerical reviews allow a data steward to approve matched duplicates, reject erroneously matched records, and reassign tasks that are better suited for other users.

To assist data stewards in identifying critical issues, Clerical Review task lists can be filtered to display only high priority tasks. To flag tasks as high priority, a workflow status flag and accompanying business condition must be configured on the relevant matching algorithm.



Clerical Review

⬆ HIGH Individual Review	17
Organization Review	1
Household Review	12
Contact Review	0



Golden Record Clerical Review Task List

	ID	Source Information	First Na...	Last Na...	Main Address
☐	139546 139560	Y 16951228-6668 X 16761227-0756	Darrel Darryl	Winston Winston	125 summer st Boston, MA, 2... 125 summer street Boston, M...
☐	139512 139926	X 16210717-8721 Y 16540622-8246	Haydan Hayden	Allistair Allistair	50 Ashley Avenue Woodland, ... 50 Ashley Ave. Woodlaand, C...
☐	139854 140010 139998	X 16670705-8878; Y 1650072... X 16210424-2330 Y 16360301-2836	Jen Jenny Jennifer	Havey Havy Haavey	12 Mohawk St Tupper Lake, N... 12 Mo-hawk St. upper Lake, N... 12 Mohawk Street Tupper Lak...
☐	139746 139950 139944 139962	X 16660526-6110 X 16210614-6273 Y 16390504-1525 Y 16750301-0253	Meg Mog Maggie Maggy	Bright Briat Brite Brighte	539 59th St Brooklyn, NY, 112... 539 59th Street Broklynn, NY, ... 539 Fivety Ninth Street Brokly... 539 59th St Apt 18 Brooklyn, ...
☐	139788 139800	X 16150202-7756 X 16160815-6079	Ted Theodore	Nugent Negentt	164 Saint George Rd Melbour... 164 Saint George Road Melbo...
☐	139902 139908	X 16750816-1317 Y 16450908-7898; Y 1692082...	Abra Abrahim	Jonathan Jonathan	101 Main St, Apt 5 Bowling Gr... 101 Main St Bowling Green, K...
☐	139503 139920	Y 16470824-0801 X 16860809-5785	Colbie Colby	Allistair Allistair	50 Ashley Av Woodland, CA, 9... 50 Ashley Ave Wodland, CA, 9

Total Number of Tasks : 16

For more information on clerical reviews, see the **Golden Record Clerical Review Task List** topic in the **Using a Web UI** documentation.

Use Case

Arthur is a senior data steward responsible for the carrying out several data management tasks, and must interact with many areas of the Web UI to do so.

When Arthur begins his day, he will first handle any clerical review tasks that may have been generated from the previous day's activity. Some new customers coming into CMDM have a risk flag set to 'yes', this requires Arthur's immediate attention. Arthur is made aware of these high priority tasks through a visual cue (see below). Arthur is then able to enter the task list and filter based on priority. This allows Arthur to immediately address the high priority clerical reviews, before continuing with his other tasks.

Components

Arthur utilizes the following components to address the story presented above:

1. The Primary Navigation Panel - Allows Arthur to see all clerical review tasks grouped in one central location. For more information, see the **Primary Navigation Panel Component** section of the **Web UI** documentation.
2. Golden Record Clerical Review Task List – Allows Arthur to address deduplication tasks. More specifically, this screen allows for merging duplicate records, task reassignment, and rejecting potential duplicates.
3. Advanced Merge – Allows Arthur to manually dictate survivorship by selecting individual attribute values.

Ad-hoc Data Stewardship

Data stewardship requires users to deal with voluminous amounts of data. To make it easy to manage this data quickly, collections are used. Collections are created by performing an advanced search then saving the results. These collections can be used to perform bulk updates as well as access all these records at will.

For information on collections and how to create queries for searching, see the **Using Advanced Search** topic in the **Web UI** documentation. To use the collections with ad-hoc bulk updates, see the **Bulk Update Operations** topic of the **Bulk Update** documentation.

Use Case

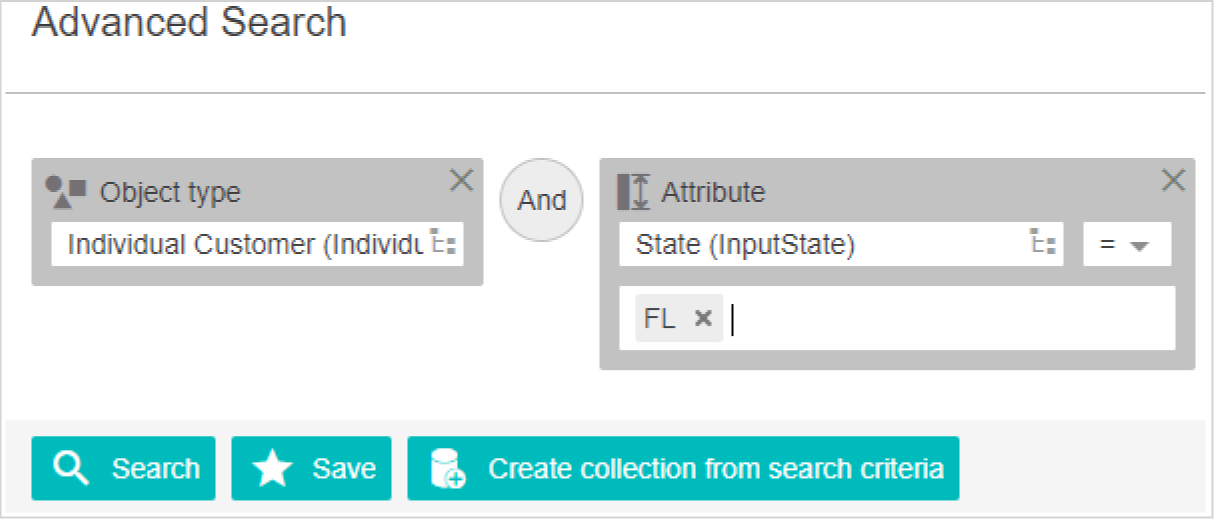
Arthur has been tasked with identifying all customers who live in Florida so they can create a new campaign to market ACME’s new swimwear. Utilizing an advanced search, Arthur can identify these customers and save them as a collection. Arthur can then export this collection and send it to the appropriate marketing specialists.

Arthur has been asked to investigate customer email quality to ensure a successful marketing campaign. Arthur realizes that no validations have been run to verify email addresses. Utilizing a bulk process, Arthur can have all customer emails validated (i.e. Experian). This allows marketing to avoid sending emails to customers with invalid email addresses. These customers can then be sent a physical letter with the promotion.

Components

Arthur utilizes the following components to address the story above:

1. Advanced Search Screen - A link from the homepage takes Arthur to an advanced search screen, allowing him to customize a search based on a variety of options. Collections can be saved once a search is made. For more information, see the **Advanced Search** section of the **Web UI** documentation.



Object type
Individual Customer (Individu

Attribute
State (InputState) =
FL

2. Node Details Screen – Once a collection is saved it can be accessed and viewed via a node details screen. Manual edits of the customer data may also be made from this screen. For more information, see the **Node Details Screen** section of the **Web UI** documentation.
3. Export – Data exports may be made on search results for reporting purposes.
4. Bulk Updates – Bulk updates may be executed on a collection for mass editing.
5. Collection Content – Sub screen used to display a list of records in a collection.
6. Source Record Traceability – Sub screen that allows Arthur to see which source system individual attribute values were retrieved from. For more information, see the **Golden Record Source Traceability Screen** section of the **Web UI** documentation.

Data Governance

Data governance is the practice of maintaining data quality within set parameters. This process is accomplished by setting policies to which data must conform, or alerts will be triggered for those in data stewardship roles. Policies can be configured to look at specific groups of attributes on specific types of records.

Arthur logs into the Web UI and is notified that an address quality policy has been breached via a homepage widget. He is able to navigate directly to the policy in order to view: current score of the policy, deviation within the last 24h which caused the breach, and the tendency over a 30 day period. This information is visualized in a graph format allowing Arthur to easily see that over the last 30 days address quality has been steadily declining. Arthur then reaches out to the source system administrators to further investigate what has caused address quality to continually decline. Through the monitoring of data governance, Arthur was able to proactively reach out to the appropriate people to address a data quality issue.

Components

Arthur utilizes the following components to address the story above:

1. Primary Navigation Panel – Allows Arthur to navigate to the policy list. For more information, see the **Primary Navigation Panel Component** section of the **Web UI** documentation.

2. Policy List - Displays all Policies in a list while providing key metrics in an overview. Individual policies can selected to view specific information. For more information, see the **Web UI Configurations for Policies** section of the **Data Governance** documentation.
3. Policy Details – Provides the performance of the selected policy.
4. Policy Widget – Provides number of breaches including which specific polices were breached.

For more information on using data policies to maintain quality data, see the **Data Governance** documentation.

Hierarchy Maintenance

Maintaining the company hierarchy is key to ensuring that data can be correctly represented across the enterprise. Using a hierarchy visualization implementation helps represent the flow of companies and how their ownership functions.

For more information on setting up and using a company hierarchy, see the **Company Hierarchy Visualization and Maintenance** topic of the **Web UI** documentation.

Use Case

A volcanic eruption has destroyed an ACME subsidiary. Arthur has been asked to remove the subsidiary from ACME's corporate hierarchy.

Arthur navigates the subsidiary in question and selects the hierarchy tab. From this screen, Arthur can remove the link to its parental organization using built in functions.

Components

Arthur utilizes the following components to address the story above:

1. Global Search – Arthur is able to navigate to the subsidiary based on source record ID.
2. Company Data Hierarchy Screen – This screen allows Arthur to add, edit, and remove links throughout the hierarchy. Several different views are also available for each hierarchy.

Attribute Maintenance

Data stewards must create and maintain various attributes and attribute groups.

For more information on attributes, see the **Attribute and LOV Creation and Maintenance Configuration** topic of the **Web UI** documentation.

Use Case

Arthur has been notified by management that an upstream SAP system is now tracking additional financial attributes (Income Verified Flag). Arthur would like to create this attribute in CMDM, so he will have access to the most complete view of the customers.

To do so, Arthur will utilize the attribute maintenance screens. First, Arthur will need to identify the correct party data attribute group. Since this is new financial data, Arthur decides it belongs best in the IncomeData group. Arthur selects this group and can create the attribute using the buttons on the screen.

Components

Arthur utilizes the following components to address the story above:

1. Attribute Group Management Screen – Using this screen, Arthur can view all attributes within the attribute group and judge whether the new attribute he wants to create belongs there. For more information, see the **Attribute Group Management Screen** section of the **Web UI** documentation.
2. Attribute Link Editor Screen - Arthur can establish or remove links between attributes via this screen. For more information, see the **Attribute Link Editor Screen** section of the **Web UI** documentation.
3. Attribute Management Screen – Arthur can create new attributes or edit existing ones via the Attribute Management Screen. He can edit the attribute details, change the attribute validity, and maintain attribute links via various tabs configurable on this screen. For more information, see the **Attribute Management Screen** section of the **Web UI** documentation.

Additional Considerations when Performing Attribute Maintenance:

Import configurations will need to be updated with the appropriate mappings. Otherwise the new incoming data will not be written to any attribute.

If the attribute does not fit into any existing attribute groups, a new group should be created first. If the attribute fits into a current display group, no web UI configuration is needed. If it is a stand alone attribute, it will need to be manually added to the appropriate details screens.

If the attribute should be considered during matching, the algorithms and/or match codes will need to be adjusted. If the intention is for the attribute to 'pool' similar customers together, it should be added as a match code function. If the attribute is intended to be considered by the algorithm, a new corresponding normalizer and matcher must be added. Additionally, the rules must be updated to include how the final score provided by the algorithm is affected. In order for survivorship to properly execute, the attribute must be placed in the relevant survivorship groups.

An onboarding process may be considered to verify that all considerations have been accounted for prior to finalizing new attribute(s).

If the attribute should be monitored via data governance, required policies and widgets will need to be updated to account for it.

Role of Business User

Experienced business users need to be able to accurately search for customers to verify information, and export customer data.

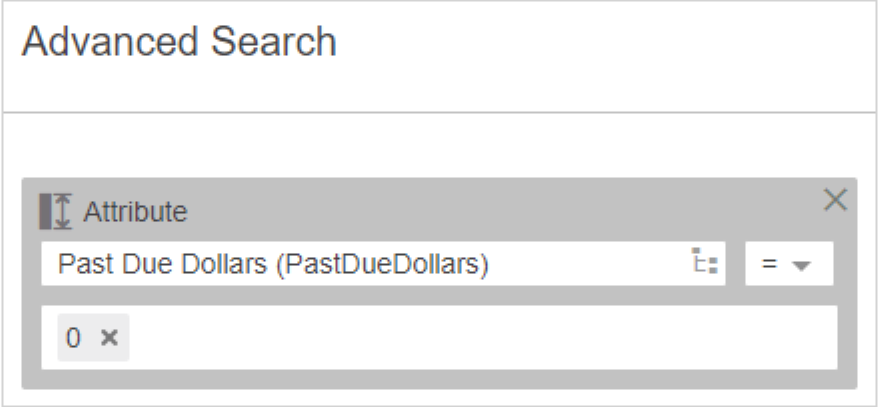
Business User Tasks

Business users need to be able to maintain small areas of the database, mostly involving searching for customers to find records and exporting this data.

Advanced Search and Data Export

Business users need to be able to navigate the large amount of data in the system. To do this, they need to create customized, granular searches using a myriad of search criteria.

Nikki, a business user, has been asked to export a list of customers who currently have outstanding payments. To do so, Nikki utilizes an advanced search. Nikki then saves the results to a collection and uses the built-in buttons to export.



To build these search queries, see the **Using Advanced Search** topic in the **Web UI** documentation.

Once the data is isolated for external support, the data will need to be exported. For more information, see the **Export Manager** topic of the **Data Exchange** documentation.

Manual Edits

Business users may also manually edit customer information within the Web UI. One example is a business user working within a call center may receive an updated contact number from a customer. The business user may edit an existing phone number or add an additional number to the customer's record.

While working in a call center, Nikki receives a phone call from a customer. The customer has notified Nikki that they are moving and would like to update their address to continue receiving promotional mailings. Nikki is able to look this customer up based on a provided loyalty card number and manually update the address. Once Nikki saves the address update, Loqate will trigger and standardize the new address.

Manual edits within the Web UI is generally carried out within the Node Details Screen.

Hierarchy Maintenance

Business users may also be responsible for maintaining company hierarchies.

Maintaining the company hierarchy is key to ensuring that data can be correctly represented across the enterprise. Using a hierarchy visualization implementation helps represent the flow of companies and how their ownership functions.

For more information on setting up and using a company hierarchy, see the **Company Hierarchy Visualization and Maintenance** topic of the **Web UI** documentation.

Data Governance

Data governance is the overall management of the availability, usability, integrity, and security of data used within an enterprise. A sound data governance program is driven by a governing body or council, which encompasses stakeholders from across the business, including data stewards. It includes a defined set of procedures, policies, and a plan to execute on those procedures.

Realization of corporate data governance is largely motivated by a desire to improve business operations and performance by gaining better oversight and management of corporate information. While a data governance program institutes policies and processes designed to produce more accurate and consistent data throughout an organization, it primarily becomes the job of the data steward to put those policies and processes into practice by ensuring compliance. It’s through governance and enforcement of said policies where you are ultimately supporting business process integrity, which in turn drives positive business outcomes.

Data Quality Policies

Data policies allow users like data stewards to define thresholds, and monitor breaches and deviations in the quality of the existing data as well as incoming data.

Data quality policies apply Metrics on Datasets to measure the quality of data. Thresholds define when users must be notified.

Policies enable data stewards to proactively monitor data. Data stewards can define policies to ensure data completeness, uniqueness, accuracy and more.

For more information, see the **Data Policies** topic in the **Data Governance** documentation.

Existing Data vs Incoming Data

Existing Data Policies evaluate data that exists in STEP each night. Incoming Data Policies only evaluates the incoming data from an Inbound Integration Endpoint of the Merge Golden Record type, allowing early warnings if the source system starts sending bad data.

Data Quality Dimensions

With Data Quality there are dimensions that need to be considered which are key cogs in driving the definition of Data Quality Policies.

Note: Data Quality dimensions are not to be confused with language dimensions, country dimensions, etc. that are platform-specific concepts.

Data Quality policies help organizations to ensure that data quality complies with the business' expectations.

Data quality policies use logical metrics on customer data to test the quality threshold. These thresholds show a simplified view of the metric performance. If data quality does not comply with the policy, a data policy breach will be recorded. These policies will also update when the data quality returns to normal expectations.

With these policies, data stewards are able to proactively monitor, control, and maintain customer data from within CMDM. Data stewards can build policies to ensure data completeness, uniqueness, accuracy, and more.

Examples of data quality dimensions are:

- Accuracy– Is the data verified, accurate and up to date? Our 3rd party integrations can now verify aspects of party data using the very latest trusted reference sources.
- Completeness– What data is missing or unusable? Is all the necessary data present?
- Timeliness - The degree to which data represents reality from a required point in time - Is data available at the time needed?

Metrics define the specifics of how to measure data quality.

ACME Holding Group Example Case

The parent company Acme Group operates across three business units; ACME Healthcare, ACME Inc., and Obits Technologies. Each business unit spans across multiple countries.

The three business units operate independently, but have a large overlap in customer base. It is a company strategy to align the business units in a customer centric way around one central brand that services the customers end-to-end. As part of this journey, customer master data must be aligned between the three business units across the various countries. For that purpose, ACME Group is turning to Stibo's CMDM solution to help address their needs.

There is not much overlap in the customer base across countries. Because of this, information stewardship is divided between each country, as there are many country-specific quirks in the data that local people can more easily deal with. To facilitate this organizational stewardship, Acme Group has established a governance organization with members from each business unit in each country. This organization will define a set of local policies for many different things. They expect that the MDM platform can monitor and report on these policies.

While customer data across various countries may adhere to different standards of completeness, it is widely accepted by ACME that the data being considered while evaluating completeness of a record shall remain the same. As such, ACME has determined that they require scoring metrics to be agreed upon across the organization. From there, individual policies may be built on top of the metrics to better define the country-specific thresholds that are appropriate.

Furthermore, since the focus of each business unit and country may be different, it is not easy for the team to understand what they should monitor. Therefore, some metrics may not be shared and must be considered as separate metrics for each business unit-country combination. In other instances, some metrics may be shared, however, the policy thresholds may be different.

Use Case 1 –Email Completeness Policy (Shared)

Dylan and Sonya are both data stewards within ACME Group whose team is responsible for analyzing and reporting on organizational governance across the globe. Dylan is responsible for ACME Healthcare in the US, whereas Sonya is responsible for ACME Inc. in Brazil.

To assist in planning for an upcoming promotional campaign, ACME Group would like to analyze the current quality of contact information of their customers across the two business units. Since the digital age means email is the engine that drives all digital campaigns, having reliable email addresses of their customers is key to achieving their quarterly fiscal goals.

To accomplish this goal, ACME Group would like to evaluate email completeness over a period to identify trends in data quality and devise internal process improvements to improve email quality.

Use Case 2 – Individual Completeness (Local)

Dylan and Sonya are both data stewards within ACME Group whose team is responsible for analyzing and reporting on organizational governance across the globe. Dylan is responsible for ACME Healthcare in the US, whereas Sonya is responsible for ACME Inc. in Brazil.

Having trusted customer data is critical to an organization's ability to confidently identify who their customers are in driving revenue and strategy. To achieve this level of maturity, ACME Group would like to obtain an accurate view of completeness of the data they have on their customers based on business unit and country.

The challenge they face is that there is no way for ACME to have meaningful visibility in assessing the quality of their customer data based even on a small set of core attributes over a period. The limitations stem from the siloed structure of the various CRM platforms for business units, countries, and subsidiaries operating today.

ACME Group has chosen to leverage Stibo CMDM's data quality offering to master their customer data. Typically, performance metrics will vary from country to country due to differences in consumer behavior, culture, standards, regulations, etc.

To achieve a level of uniformity in performance indicators across business units and countries, it therefore makes sense for Dylan and Sonya to define a common set of attributes which defines a completeness metric from which to gauge customer data. These core attributes consist of:

- First name
- Last name
- Address Line
- City
- State/Province
- Postal Code
- Country

Since the quality of customer information gathered may vary from country to country, Dylan and Sonya have realized that while the US and Brazil may share the same metrics, there is a need to track the quality of data within each country by using separate policies. This provides both Dylan and Sonya visibility into how their respective countries are "performing" over time.

Since ACME is interested in the overall quality of data on their customers, the data quality policy that Dylan and Sonya are responsible for will execute and profile against the existing customer data set as opposed to monitoring incoming data from specific source systems.

The result is a single completeness metric that is utilized by multiple policies between the US and Brazil (and potentially other countries) with thresholds unique to each country. As such, Dylan and Sonya can view and edit policy thresholds for their respective countries. They are also able to view the policy's performance via the data quality dashboard. Should a policy breach occur for their respective countries, Dylan and Sonya will be notified via email.

Use Case 3 – Source System Performance (Stream)

Bobby is a senior data analyst who is responsible for monitoring and evaluating the overall performance and quality of customer data for the many source systems (CRM platforms) within ACME Group subsidiaries.

To help support the upcoming season, Bobby is assisting the marketing department in streamlining the promotional mailings process and reduce overhead costs. To do so, Bobby requires the monitoring of address quality ACME Group possesses for their customers, particularly to identify quality within source systems to better identify and address the root problems.

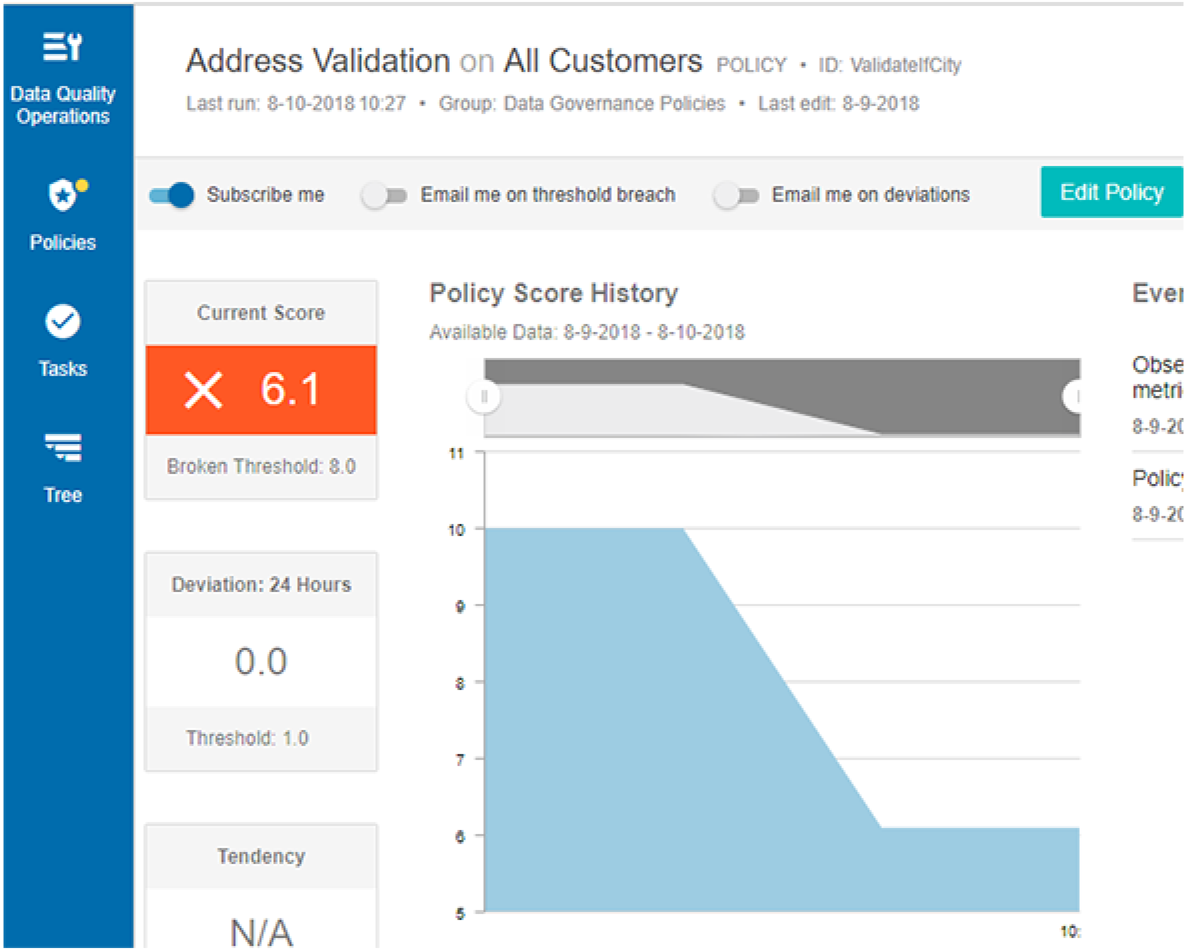
While an existing data quality policy allows for Bobby to gauge the quality of addresses among the existing data set within ACME Group, should there ever be a change in how data is captured in one of ACME Group's many source systems, there is not a good way for Bobby to realize this in a timely fashion. As such, a stream policy which monitors data as they come in from individual source systems would allow Bobby to more easily monitor the behavior/performance of each source system.

With CMDM, a value metric allows for evaluating of attribute values and mapping to specific scores. In the scenario of ACME Group's address quality policy, all customer addresses are verified and standardized via the Loqate integration which returns an Address Quality Index (AQI). The AQI is a letter score which of the quality of input address that was validated against Loqate's database. This letter score is evaluated and resolved as a corresponding numeric score for the policy to profile and display against.

The following table shows the evaluation of the AQI:

Score	Quality	Description	Policy Score
A	EXCELLENT	Verifiable to at least Premise level without changes	10
B	GOOD	Verifiable to at least Thoroughfare level with minor changes	8
C	AVERAGE	Verifiable to at least Locality level with moderate changes	6
D	POOR	Only verifiable to at least Locality level with more than moderate changes	4
E	BAD	N/A	2
[null]	N/A	No Address Validation / Quality Index	0

With this value metric defined, Bobby can configure individual policies to evaluate source system performance of customer addresses. He is also able to be notified of breaches and view the performance of a particular source system via the policy dashboard:



Address Validation on All Customers POLICY • ID: ValidateIfCity

Last run: 8-10-2018 10:27 • Group: Data Governance Policies • Last edit: 8-9-2018

Current Score

6.1

Broken Threshold: 8.0

Deviation: 24 Hours

0.0

Threshold: 1.0

Tendency

N/A

Policy Score History

Available Data: 8-9-2018 - 8-10-2018

Eve

Obse
metri
8-9-2(

Polic
8-9-2(

CMDM Configuration Guide

The purpose of this document is to provide an implementation party a detailed guide to the installation of the starter package configurations in a freshly installed STEP environment. This document should be interpreted as a complimentary peer to the collective enablement collateral provide by the CMDM Implementation Guidelines.

STEP System Components

This document assumes the target system is on the latest release of STEP 9.0.

The following components are required to be activated in your STEP system prior to beginning configuration within the application:

1. Party Data Matching (partydatamatching)
2. Company Data Visualization (companydata-visualization)

Note: Other CMDM-related features require dedicated components which are not in the scope of the CMDM Implementation Guide. Dedicated documentation for these components is available separately:

- Local Loqate (local-loqate)
- Experian Email Validation (experian)
- Dun & Bradstreet Company Match and Profiling (dnb-integration)

Deployment of STEP components require adequate working knowledge of patching STEP environments. For deploying STEP components it is recommended to solicit the assistance of a Stibo Systems developer or Technical Services representative.

STEPXML Configuration Files

The following is the complete list of the STEPXML import files required to provide a clean STEP system with the necessary configurations for a functional, base CMDM solution.

File Name	File Content
sharedconfig.properties	System Configuration Properties
Link, Reference, and Object Types.xml	Link, Reference, and Object Types
List of Values.xml	List of Values
Attributes and Attribute Groups.xml	Attributes and Attribute Groups
Setup Entities.xml	Setup Entities
Setup Groups.xml	Setup Groups

File Name	File Content
Data Containers.xml	Data Containers
Entity Root Nodes.xml	Entity Root Nodes
Classifications.xml	Classifications
Business Rules and Libraries.xml	Business Rules and Libraries
Component Models.xml	Component Models
Import Configurations.xml	Import Configurations
Transformation Lookup Tables.xml	Transformation Lookup Tables
Match Codes.xml	Match Codes
Match Algorithms.xml	Match Algorithms
Integration Endpoints.xml	Integration Endpoints
Users and User Groups.xml	Users and User Groups
Event Processors.xml	Event Processors
Collection Definitions.xml	Collection Definitions
Workflows.xml	Workflows
WebUI.xml	WebUI
SampleContacts.xlsx	Sample Contacts Data
SampleIndividuals.xlsx	Sample Individuals Data
SampleOrganizations.xlsx	Sample Organization Data
Sample_Pair_Formatter_ver3.1	Sample Pair Formatter Excel Sheet

STEPXML Import Sequence

Due to the nature of STEP configuration dependencies, to properly import the STEPXML configuration files a strict sequence of import must be adhered to. Please note the sequence of tasks presented in this section contains both STEPXML files to be imported as well as manual configuration tasks that are required but not captured by STEPXML.

Furthermore, it is strongly recommended to only import the latest released version of each STEPXML file. To determine whether a newer version is available, please contact Stibo Systems.

1 - Shared Config Properties File

Each STEP environment contains unique properties specific to the environment, hence, no sharedconfig.properties file will be provided as part of this Starter Package. The following CMDM properties need to be manually added to the existing sharedconfig.properties file.

The following are standard Loqate Cloud properties that must be added in order for Loqate Cloud integration to function properly. These properties assume the deployment of the Loqate Cloud solution. For details on configurations for the Loqate Local solution, please refer to the Loqate set-up guide.

Address.AddressQualityPlugin=LoqateAddressQuality

Address.Service.Loqate.License=[Please obtain a License key for Loqate Cloud]

Address.Service.Loqate.GeocodeAll=[true or false]

Note: The Loqate.Geocode property is Optional. The default value is true. Specifies if all addresses should be geocoded. A geocode is a pair of coordinates for the latitude and longitude of a location.

Additionally, for the STEP environment to access Loqate servers for validation through the cloud, the following URL must be added to the firewall to ensure connectivity:

http://saas.loqate.com

Once these properties are in place, a restart of STEP is required. Finally, if a pre-existing instance of STEP is already running on your local computer then it is recommended to relaunch STEP.

2 - Link, Reference, and Object Types

The first step after modifying the sharedconfig properties file is to ensure basic object types are available in the system.

This file includes not only references but also metadata attributes associated with references. Since attributes are not yet available in the system, it is expected that the import process will encounter some errors pertaining to missing attribute IDs which may be ignored. The reason for these errors is that Object Types must be present in order for attribute validity within the *Attributes and Attribute Groups* file to properly import. Once Attributes have been imported, a re-import of Object Types will take place to establish the metadata attribute associated with references.

3 - List of Values

This file contains all CMDM-specific Lists of Values.

4 - Attributes and Attribute Groups

Attributes, including all metadata attributes, will be imported via this file.

5 - Link, Reference, and Object Types

This is a re-import of the same file from import sequence #2 above. The reason is to ensure all metadata attributes for references are accounted for now that attributes are present in the system.

6 - Setup Groups

Setup groups are the root nodes within the System Setup tab. All root nodes must be present before dependent Setup Entities are imported.

7 - Setup Entities

This file contains CMDM-specific Setup Entities.

8 - Data Containers

Data containers utilize a number of attributes that should now be present in the system.

9 - Entity Root Nodes

This file provides a base entity structure within the Tree Navigation tab of the STEP application. Also included are basic entities such as default source systems and hierarchy visualization aggregate objects.

10 - Classifications

A specific classification folder is required to support transformation look-up tables.

11 - Business Rules and Libraries

All business rules and business rule libraries required for the starter package.

12 - Component Models

Included in the Component Models STEPXML file are components for Address, Party Data Matching, Linked Golden Record, and Merged Golden Record components. Importing the Component Models STEPXML file means the user is no longer required to deploy each individual component model as all necessary attributions are included. However, if it is determined that separate attributes are more desirable to a client / on-going project, then manual deployment of each component model is also available.

13 - Import Configurations

Three import configurations are provided and are associated with each of the three customer entity types: Individuals, Organizations, and Contact Persons. These import configurations provide attribute mappings for the corresponding sample files and are utilized by the match and merge importer.

Since Household entities are generated within the CMDM solution, there is no import configuration for Households.

14 - Transformation Lookup Tables

An equivalent names table is provided with basic and common name equivalents. It is recommended that clients and implementation parties maintain and expand this table with new entries as needed.

15 - Match Codes

Match codes for the four customer entity algorithms.

16 - Match Algorithms

Four algorithms for Individuals, Organizations, Household, and Contact Persons.

17 - Integration Endpoints

Upon importing the Integration Endpoints STEPXML file, the endpoints will need to be manually invoked in order for the hotfolders to be created on the filesystem.

18 - Users and User Groups

A dedicated 'Data Stewards' user group is provided. However, no specific privileges are given to this user group. It is recommended the clients and/or implementation parties determine the specific privileges the data stewards require.

19 - Event Processors

A matching and address standardization event processor is provided. Event processors and queue status' must be manually set to 'Enable' and 'Read,' respectively.

20 - Collection Definitions

A number of collection definitions are provided to support data profiling within the WebUI

21 - Profile

Data Profile configurations are not supported by STEPXML, hence a manual task to configure the completeness metrics with each collection is required.

The following configurations are done from the data profiles tab of a collection after entering the settings. Dashboard and profile configurations both require manual configuration.

All Individual Collection

Profile Configuration

1. Add individual Completeness under 'Select Completeness Metrics to profile'
2. Add Individual Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness

   a. Right click > Configure > Set completeness metric to Individual Completeness and Intervals to 10 (Or desired interval)

All Objects Collection

Profile Configuration

1. Add General Metric Completeness under 'Select Completeness Metrics to profile'
2. Add Individual Complexness, Organization Completeness, Household Completeness, Contact Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness
   a. Right click > Configure > Set completeness metric to General Metric Completeness and Intervals to 10 (Or desired interval)
7. Attribute Values Profile
   a. Right click > Configure > Set Attribute ID to 'CalcAQI' and chart type to Pie Chart

Recently Updated Contact Persons Collection

Profile Configuration

1. Add Contact Completeness under 'Select Completeness Metrics to profile'
2. Add Contact Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness
   a. Right click > Configure > Set completeness metric to Contact Completeness and Intervals to 10 (Or desired interval)

Recently Updated Households Collection

Profile Configuration

1. Add Household Completeness Metric under 'Select Completeness Metrics to profile'
2. Add Household Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness
    a. Right click > Configure > Set completeness metric to Household Metric Completeness and Intervals to 10 (Or desired interval)

Recently Updated Organizations Collection

Profile Configuration

1. Add Organization Completeness under 'Select Completeness Metrics to profile'
2. Add Organization Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness
    a. Right click > Configure > Set completeness metric to Organization Completeness and Intervals to 10 (Or desired interval)

Recently Updated Individuals Collection

Profile Configuration

1. Add Individual Completeness under 'Select Completeness Metrics to profile'
2. Add Individual Completeness attribute group under 'Include Data in Attribute Groups'

Dashboard Configuration

3. Object type count
4. Least complete attribute
5. Most complete attribute
6. Average Object completeness
    a. Right click > Configure > Set completeness metric to Individual Completeness and Intervals to 10 (Or desired interval)

22 - Workflows

A total of four clerical review workflow are provided; one for each of the four customer entity types.

23 - Web UI

A basic Web UI is provided to assist data stewards with their day to day responsibilities of clerical review, manual maintenance of customer data, and data monitoring.

24 - Sample Data Load Files

Three sample data Excel files are available for each of the three customer entity types to be imported via the match and merge import configuration (i.e. Individuals, Organizations, and Contact Persons).

The names of these sample data files are:

- SampleContacts.xlsx
- SampleIndividuals.xlsx
- SampleOrganizations.xlsx

Note: Users should import SampleOrganizations.xlsx before importing SampleContacts.xlsx.

Sample data files may be imported with the Match and Merge Importer via the Asset Importer Widget available within the Web UI.

CMDM Starter Package Configuration Delivery Mechanism

All STEPXML configuration and sample data files will be provided by Stibo Systems CMDM team in the form of a zip file. Subsequent updates to the configuration files will be made available upon request.

Important: Any CMDM Starter Package configuration files received from sources other than the Stibo Systems CMDM team should not be trusted as new revisions may be available. Please contact the Stibo Systems CMDM team for the latest versions.