**Tokenized Sending**

**Purpose:** Through this document we will go through about Tokenized Sending, how we can implement tokenized sending and how it can benefit us.

**Overview:** Tokenization is the process of replacing sensitive information elements with non-sensitive ones, called tokens, to ensure data security. With Tokenized Sending, an account stores a single token per send attribute. This token resembles a standard email address or phone number but doesn’t include any information about the contact.

**Introduction:** Tokenized Sending is used to send contact data that is too sensitive to store in Marketing Cloud Engagement database. We can take information from our own data systems and transmit it via API for use at send time.

**Use Cases:** To protect sensitive information across clouds, we can use tokenized sending with Marketing Cloud Connect for Marketing Cloud Engagement.

* **Across all kinds of Industry**, any Personally Identifiable Information (PII) can be stored in a customer data warehouse, and before using that data, it can be imported as tokens by SFMC, before using it to send any communication.
* For **Health Care Industry**, any Protected Health Information (PHI) can be used by tokenization for an added layer of encryption and protection.
* For **Financial Sectors**, any Payment Card Industry (PCI) data can be tokenized for security.

**Scope:** These are the following things that we will be covering in the document:  
> The prerequisites that we require before implementing tokenized sending.  
> Implementation of Tokenized Sending.  
> Token API Security  
> Tokenized Send Examples for Marketing Cloud Engagement  
> Token Formats  
> Supported Email Messaging Functionality  
> Supported MobileConnect Messaging Functionality  
> Unsupported Tokenized Sending Features  
> No Token Queue for MO SMS Messages  
> Get Token API Specification  
> Resolve Token API Specification

**Current Challenges:** Tokenized Sending lets you use data for your messages that is deemed too sensitive to store in Salesforce due to laws, regulations, or security policies. To protect sensitive information across clouds, use tokenized sending with Marketing Cloud Connect for Marketing Cloud Engagement. For example, sensitive information can include:

* Personally Identifiable Information (PII)
* Protected Health Information (PHI)
* Any other information you prefer to retain on your own servers.

**Goals and Objectives:** The tokenized sending process transmits information from your data systems to Engagement at send time using an API call. Data is never stored in Salesforce. Instead, your Salesforce org stores a single token, resembling a standard email address, per subscriber. Each token can contain a collection of attributes about that subscriber. Sensitive data received from the token exchange is discarded from memory and not retained in Marketing Cloud. However, all tracking and deliverability data is available in Engagement, identified by the subscriber key.

If tokenized sending is provisioned for your account, keep these considerations in mind when using it with Marketing Cloud Connect.

* Recommended for use only with new accounts.
* Works with all Engagement account types using subscriber keys.
* Contact, Lead, and Person Account records in Sales and Service Clouds must use the token as the main email address.

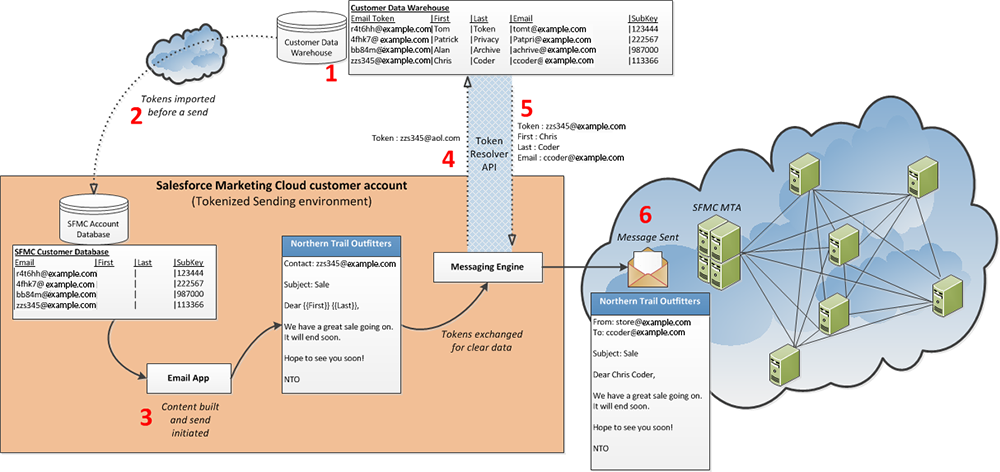
**Solution Description & Components:** These should be prerequisite knowledge.

* You have basic knowledge about token and security concepts.
* Contact an account executive to enable this feature when you create a new account. A new account ensures that no sensitive data is already stored.
* Tokenized Sending functions as part of any account type. The feature operates as an enterprise-wide configuration across parent and child business units.
* You need a separate tenant for Tokenized Sending.
* Develop REST API calls for your data system to send the correct information at send time. These calls don’t occur within REST API framework.
* Use the provided API specifications for getting and resolving tokens. Develop your code around these specifications for this feature to work.
* This feature requires that your REST API implementation must use HTTPS (SSL, port 443). Your SSL server certificate can’t be self-signed.
* Your REST API implementation handles up to 50 connections simultaneously and resolves batches of up to 500 tokens for each connection.
* Each REST API connection resolves tokens and returns a response within 60 seconds.
* Tokenized Sending uses token-to-send attributes as a 1:1 relationship. Create tokens, even when the subscriber is new or opting-in and maintain the mapping of tokens to clear subscriber data.
* All automatically processed STOP and UNSUBSCRIBE requests from your subscribers require tokens. If your Get Token API for SMS MOs is unavailable or doesn’t provide a token, you assume the responsibility to manually unsubscribe your subscriber.
* Log all request and response payloads.
* Monitor your production server and connectivity.
* Test your REST API implementation before deployment using the provided testing harness.
* To ensure proper connectivity, allowlist Engagement IP addresses.
* This feature requires a subscriber key value for each contact in your account.
* Email address or mobile number must be tokenized.
* This feature applies to new data only.

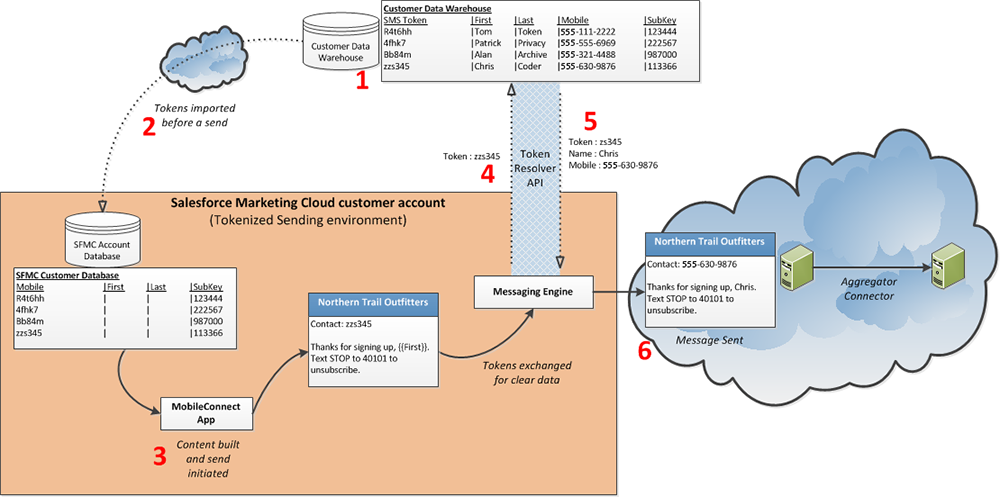
**Implementation:** Before you enable Tokenized Sending in Marketing Cloud Engagement, review requirements and test the functionality. Note that some features don't work when Tokenized Sending is enabled but not yet implemented.

1. To ensure successful implementation, review the tokenized sending prerequisites and the Get and Resolve Token API specifications.
2. Develop your REST APIs for getting and resolving tokens.
3. Configure your APIs.
   1. Allowlist applicable IP addresses.
   2. From Setup, under Security, enter the endpoint and authentication credentials.
4. To test your configuration, save your endpoint and authentication credentials in the Security section. Review results of the test and fix your configuration accordingly. Improper configurations aren't saved.
5. Create a sample list with sample tokens and import it.
6. Perform a test send using the sample list.
7. After a successful test send, your account is ready for production sends.

**Get Token API Specification:**   
Use this information to provide token representation of plain text values. Marketing Cloud Engagement uses this API method to retrieve the token representation of a plain text value.  
Engagement sends an HTTP POST request to the URI for the token server you provide as part of a tokenized sending process. This request contains a list of token values, and your token server responds with the list of token values and associated tokens.  
If the request is sent without a token, the response JSON in the tokens array does not contain any data. If there is only a single tokenValue in the request and your token server does not provide a token for it, the response includes an empty array for the tokens property in the JSON response.  
The API gets the token representation of phone number data. The token is stored in the database instead of the source phone number from an inbound SMS message.

**Email Send:** This example demonstrates an email send using Tokenized Sending in Marketing Cloud Engagement.

(1) Contact information is stored in the customer data warehouse.  
(2) Tokens are imported before a send.  
(3) User initiates a send. Content building occurs now.  
(4) Marketing Cloud Engagement initiates a call to the customer data warehouse for contact information.  
(5) Customer data warehouse returns contact information.  
(6) Marketing Cloud Engagement uses returned contact information for send.

**MobileConnect Send:** This example demonstrates using Tokenized Sending for a MobileConnect Send in Marketing Cloud Engagement. 

(1) Contact information is stored in the customer data warehouse.  
(2) Tokens are imported before a send.  
(3) User initiates a send. Content building occurs now.  
(4) Marketing Cloud Engagement initiates a call to the customer data warehouse for contact information.  
(5) Customer data warehouse returns contact information.  
(6) Marketing Cloud Engagement uses returned contact information for send.

Conclusion: Tokenized Sending basically swaps out your sensitive data for a token, so that sensitive data does not get stored in your Marketing Cloud account at all. At point of send, a token resolve runs, looking up the token and populating the required information. It fully prevents any sensitive data from residing in your Marketing Cloud account (if that's what you're trying to achieve i.e. for data sovereignty purposes). We just need to create a new Marketing Cloud account to use it and it is encrypted in the UI, meaning some features might not be possible i.e. segmenting based on encrypted fields. It supports only email and SMS resolve logic so things like MobilePush, which uses a device ID so 'send', are not supported.  
  
References:   
(1) <https://help.salesforce.com/s/articleView?id=mktg.mc_overview_tokenized_sending1.htm&type=5>  
(2) <https://help.salesforce.com/s/articleView?id=mktg.mc_co_tokenized_sending.htm&type=5>