**Customer Product Information display and Integration**

**Technical & Functional Documentation**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comment |
| 1.0 | 01-08-2024 | TROUILLET Valentin | Creation |

[1 Context 3](#_Toc173424963)

[1.1 Display Product Information on Case 3](#_Toc173424964)

[1.2 Allow Product Information to be accessed from an external source. 4](#_Toc173424965)

[2 Salesforce 7](#_Toc173424966)

[2.1 Global Value Sets 7](#_Toc173424967)

[2.2 Custom Fields 7](#_Toc173424968)

[2.3 Custom Metadata Types 7](#_Toc173424969)

[2.4 Lightning Record Page 8](#_Toc173424970)

[2.5 Lightning Web Component 8](#_Toc173424971)

[2.6 Custom Labels 9](#_Toc173424972)

[2.7 Layout 9](#_Toc173424973)

[2.8 Permission Sets 9](#_Toc173424974)

[2.9 Apex 10](#_Toc173424975)

[2.9.1 DisplayProductInfoController 10](#_Toc173424976)

[2.9.2 WSContactInformationManager 10](#_Toc173424977)

[2.10 Connected App 10](#_Toc173424978)

[2.11 Integration User 10](#_Toc173424979)

[3 External Platform 11](#_Toc173424980)

# Context

Customer data is composed of Cases, Contact and Product. As of now there is no efficient way to see all the necessary information on a Contact from a Case and Customer Service Agents are faced with delays in accessing information.

The goal of this project is to enhance the efficiency and effectiveness of customer service agents by leveraging Lightning Web Components and Apex Rest API. This enhancement will enable agents to access crucial customer information in real-time, facilitating informed and responsive customer interactions. The focus is on providing dynamic access to customer product information directly within the Salesforce Case page layout, improving the agent's ability to resolve customer inquiries quickly and accurately.

There are two parts to this project:

## Display Product Information on Case

The Lightning component (or Lightning Web Component) should be visible on the Case page layout.It should utilize two custom fields from the related Contact record: Product\_\_c and Home\_Country\_\_c.

Based on these values, the component will display product information.

This product information will be stored in Salesforce and will allow for easy modification and addition of new products. A Product depends on its Home Country and Product Tier as seen below.

**Cost per Calendar Month**

| **Tier** | **DE** | **FR** | **ES** | **IT** | **UK** |
| --- | --- | --- | --- | --- | --- |
| Standard | €0 | €0 | €0 | €0 | £0 |
| Black | €9.90 | €9.90 | €9.90 | €9.90 | N/a |
| Metal | €16.90 | €16.90 | €16.90 | €16.90 | N/a |

**ATM Fee in Other Currencies**

| **Tier** | **DE** | **FR** | **ES** | **IT** | **UK** |
| --- | --- | --- | --- | --- | --- |
| Standard | 1.7% | 1.7% | 1.7% | 1.7% | 1.7% |
| Black | Free | Free | Free | Free | Free |
| Metal | Free | Free | Free | Free | Free |

**Card Replacement Cost**

| **Tier** | **DE** | **FR** | **ES** | **IT** | **UK** |
| --- | --- | --- | --- | --- | --- |
| Standard | €6 | €6 | €6 | €6 | £6 |
| Black | €6 | €6 | €6 | €6 | £6 |
| Metal | €45 | €45 | €45 | €45 | £45 |

## Allow Product Information to be accessed from an external source.

Create an external system that will allow access to customer data stored in Salesforce.

The external system passes a unique identifier (UUID) mapped to each Contact record in Salesforce.

The external system can create and update Contact records in salesforce with this identifier.

Getting a customer product information will be done with the following URL : {endpoint}/services/apexrest/ContactInformation/{UUID} , with the endpoint being the current org’s base URL.

The table below details the fields names depending on the origin :

|  |  |  |
| --- | --- | --- |
| **External Field** | **Salesforce Field** | **Values** |
| externalReference | Product\_\_c | Standard,Black,Metal |
| lastName | LastName |  |
| firstName | FirstName |  |
| homeCountry | Home\_Country\_\_c | DE,ES,FR,IT,UK |

The response will be the following depending on success :

Code 200

{

"product": "Standard",

"lastName": "Ganz",

"homeCountry": "DE",

"firstName": "Bruno",

"costPerCalendarMonth": "0.0",

"cardReplacementCost": "6.0",

"atmFee": "1.7"

}

Or failure :

Code 400

There has been an error : Unable to find the contact

Upserting contacts will be done via the following URL : {endpoint}/services/apexrest/ContactInformation

Expected JSON will look like this :

[

{

"externalReference": "aaaa",

"lastName": "Mastroianni",

"firstName": "Marcello",

"product": "Metal",

"homeCountry": "IT"

},

{

"externalReference": "abc2",

"lastName": "Reno",

"firstName": "Jean",

"product": "Black",

"homeCountry": "FR"

},

{

"externalReference": "abc3",

"lastName": "Frisch",

"firstName": "Arno",

"product": "Standard",

"homeCountry": "DE"

},

{

"externalReference": "abc4",

"lastName": "Noriega",

"firstName": "Eduardo",

"product": "Metal",

"homeCountry": "ES"

},

{

"externalReference": "abc5",

"lastName": "Craig",

"firstName": "Daniel",

"product": "Metal",

"homeCountry": "UK"

}

]

A successful message will be :

Code 201

Successfully upserted 5 Contacts

A failure message would depend on the error, a bad JSON would return the following :

Code 400

There has been an error : Unexpected character ('w' (code 119)): expected a valid value (number, String, array, object, 'true', 'false' or 'null') at [line:1, column:25]

# Salesforce

## Global Value Sets

Two Global value sets are created to ensure data quality. They will be used in relevant picklists.

* Home Country
  + DE
  + ES
  + FR
  + IT
  + UK
* Product Tier
  + Standard
  + Black
  + Metal

## Custom Fields

Three custom fields are created on Contact.

* Product\_\_c
  + Uses the global value picklist Product Tier
* Home\_Country\_\_c
  + Uses the global value picklist Home Country

These 2 fields are restricted and will ensure data quality across the platform.

* External\_Reference\_\_c
  + Unique , External Id
    - Used to synchronise data with the external system.

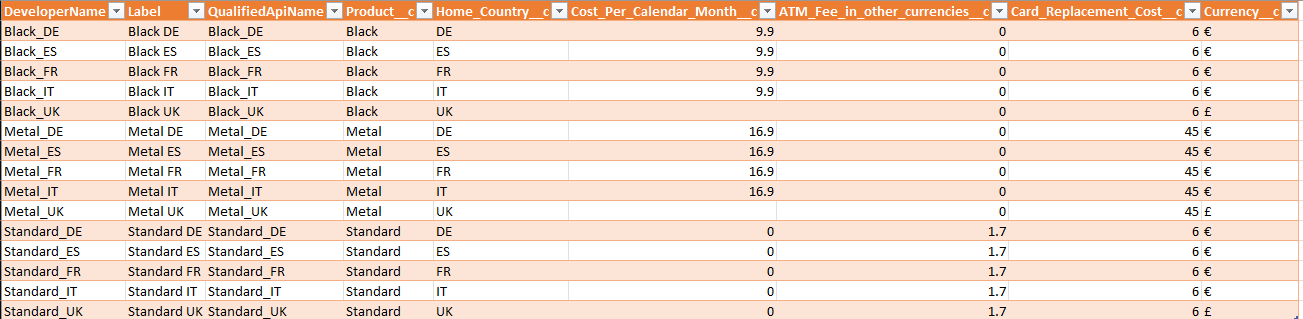
## Custom Metadata Types

A New custom metadata type has been implemented: Cost\_Per\_Product\_\_mdt.

It contains the following fields :

* Product\_\_c
* Home\_Country\_\_c
* Cost\_Per\_Calendar\_Month\_\_c
* ATM\_Fee\_in\_other\_currencies\_\_c
* Card\_Replacement\_Cost\_\_c
* Currency\_\_c

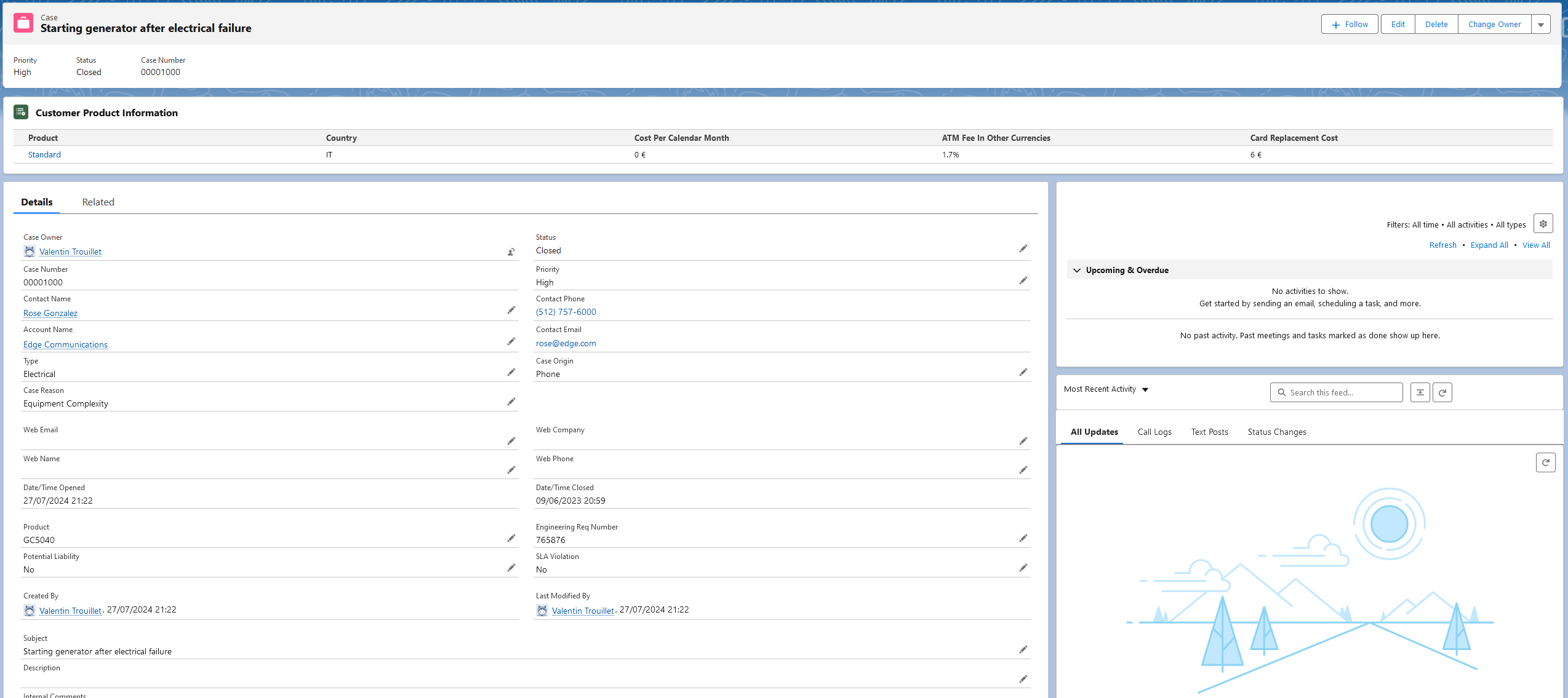
It contains the information regarding costs and fee per product tier and country. Adding custom metadata records is easily done and can be deployed as metadata.

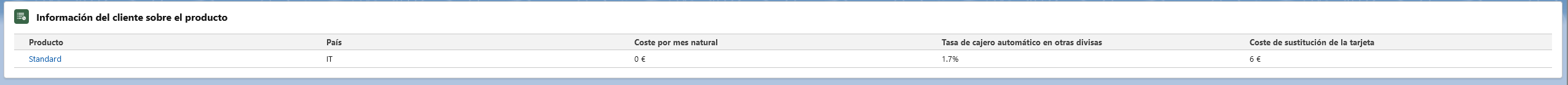
These are the current records:   


## Lightning Record Page

A new Lightning Record Page has been created for the Case Object. This is used to implement the LWC into the record page: Standard\_Case\_Page\_With\_Product\_Info

It contains the new Lightning Web Component and a standard layout for the Case Object.



****

## Lightning Web Component

A new Lightning Web Component has been developed. Located on a Case Record Page. This component will fetch the Product Information stored in the Cost Per Products metadata records depending on the related Contact’s Home Country and Product. It will then show this data in a table.

Custom Labels are used to ensure support agents can see the information in their own language. This LWC is displayProductInfo.

## Custom Labels

Custom Labels are used in both the display LWC and in messages sent back to the external users accessing the salesforce platform.

|  |  |
| --- | --- |
| **Name** | **Value** |
| [LWC\_ATM\_FEE\_IN\_OTHER\_CURRENCIES](javascript:srcUp(%27%2F101WU00000IyyYP%3Fisdtp%3Dp1%27);) | ATM Fee In Other Currencies |
| [LWC\_CARD\_REPLACEMENT\_COST](javascript:srcUp(%27%2F101WU00000IyyYQ%3Fisdtp%3Dp1%27);) | Card Replacement Cost |
| [LWC\_COST\_PER\_CALENDAR\_MONTH](javascript:srcUp(%27%2F101WU00000IyyYR%3Fisdtp%3Dp1%27);) | Cost Per Calendar Month |
| [LWC\_COUNTRY](javascript:srcUp(%27%2F101WU00000IyyYS%3Fisdtp%3Dp1%27);) | Country |
| [LWC\_CUSTOMER\_PRODUCT\_INFORMATION](javascript:srcUp(%27%2F101WU00000IyyYT%3Fisdtp%3Dp1%27);) | Customer Product Information |
| [LWC\_FREE](javascript:srcUp(%27%2F101WU00000IyyYU%3Fisdtp%3Dp1%27);) | Free |
| [LWC\_PRODUCT\_NAME](javascript:srcUp(%27%2F101WU00000IyyYV%3Fisdtp%3Dp1%27);) | Product |
| [WS\_CONTACTMISSINGINFO](javascript:srcUp(%27%2F101WU00000IyVGK%3Fisdtp%3Dp1%27);) | The contact does not have a Home Country or Product. |
| [WS\_ERROR](javascript:srcUp(%27%2F101WU00000IyahZ%3Fisdtp%3Dp1%27);) | There has been an error |
| [WS\_NOPRODUCTINFO](javascript:srcUp(%27%2F101WU00000IxWsk%3Fisdtp%3Dp1%27);) | Unable to find the product information |
| [WS\_NORECORD](javascript:srcUp(%27%2F101WU00000Iyfab%3Fisdtp%3Dp1%27);) | Unable to find the contact |
| [WS\_NOTHINGTOUPSERT](javascript:srcUp(%27%2F101WU00000Iyp5J%3Fisdtp%3Dp1%27);) | No contacts to upsert |
| [WS\_UPSERTSUCCESS](javascript:srcUp(%27%2F101WU00000IymH9%3Fisdtp%3Dp1%27);) | Successfully upserted |

Translations are available in German, French, Italian and Spanish.

## Layout

The standard cast layout Contact-Contact Layout has been modified, Product\_\_c and Home\_Country\_\_c have been added.

## Permission Sets

3 Permission sets have been created :

* Support Agent
  + Has access to Contacts,Cases and Cost Per Product
* Integration Access
  + Has access to the webservice apex class : WSContactInformationManager and Cost Per Product
* API Integration
  + Has access to Contact Records

## Apex

Two classes have been built for these features, one Controller for the LWC and one WebService, as well as the equivalent test classes

### DisplayProductInfoController

This apex class contains a method, getCostPerProduct, which is AuraEnabled and will be used the query the custom metadata based on the recordId it receives as parameter(A Case Id).

This method is called in the LWC displayProductInfo

### WSContactInformationManager

This class contains two methods:

#### getContactByExtId

Uses HttpGet. Awaits an external Id (Contact.External\_Reference\_\_c) as a parameter, and returns the product information depending on the customer’s product information.

Errors are handled and a relevant error message will be returned.

#### upsertContacts

Uses HttpPut. Awaits a JSON containing Contact Records with External References. Will upsert the contacts received, and return a success or failure message relevant to the error if needed.

## Connected App

A Connected App is created, using the Client Id, Client Secret, Token allows connection to the salesforce platform from outside.

It has to be manually created by hand before deploying the other components, on each environment.

## Integration User

Likewise, a User using the Salesforce API Only System Integrations profile will be created manually, that user with OAuth 2.0 can be used to send API Requests to the salesforce platform.

# External Platform

To connect to the REST API developed on the Salesforce platform, the external platform developers must connect with the Salesforce organization using Client\_Id/Client\_Secret/Token.