Product Registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 30-03-2016 | Deepthi Shivakumar | All | Initial draft |
| 0.2 | 15-5-2016 | Deepthi Shivakumar | All | Incorporated Matthijs’ comments |

|  |  |
| --- | --- |
| **Author** | Deepthi Shivakumar |
| **Approved by** |  |
| **Email Id** | Deepthi.Shivakumar@Philips.com |

CONTENTS

1. DEFINITIONS & ABBREVATIONS 3

2. INTRODUCTION 3

2.1 Purpose 3

2.2 Scope 3

2.3 Target Audience 3

2.4 References 4

3. ARCHITECTURAL ANALYSIS 5

3.1 Product Registration Overview 5

3.2 System Context 5

3.3 Design Method 6

3.4 Platform dependency 6

3.4.1 Class names 6

3.4.2 Errors 6

4. STATIC DESIGN 7

4.1 Overall System View 7

4.2 Sequence 8

4.2.1 Product registration sequence ( Online/Offline ) 8

4.2.2 Product registration when user signs in 9

4.3 Layering Model 10

5. DYNAMIC DESIGN 12

5.1 Use cases 12

5.1.1 Use case: Fully automatic 12

5.1.2 Use case: Semiautomatic case 13

6. PRX backend 13

# DEFINITIONS & ABBREVATIONS

|  |  |
| --- | --- |
| API | Application Programming Interface |
| CDP2 | Connected Digital Platforms and propositions |
| HTTP | Hyper Text Transfer Protocol |
| OCDB | Online Consumer Database |
| SDK | Software Development Kit |
| SSL | Secure Socket Layer |
| UUID | Universal Unique Identifier |
| PPR | Philips Product Registration |

* User registration – The process where user creates an account with Philips using social service provider or traditional email/password mechanism.
* Product Registration – Process where-in the registered user links his Philips product with his account.
* Janrain – Platform that provides social service interface.
* Janrain SDK – Software Development Kit provided by Janrain for Mobile development.
* PRX – Platform that provides an interface to retrieve product information and also provides interface for product registration.

# INTRODUCTION

This document provides an overview of architecture and design for product registration component in Mobile applications.

## Purpose

The purpose of this document is to elaborate various building blocks used in product registration of Philips products using mobile applications. This document also explains how these building blocks can be put together to build a re-usable component.

## Scope

This document covers only product registration from mobile touch point. Also, the design does not cover the data sync mechanism between various backend components.

The component design is not specific to any particular platform though iOS and Android will be the first carrier platforms.

## Target Audience

CDP2 development teams

## References

* DI use of Janrain service – Bas Bergevoet
* PRX\_API\_v4.18.docx
* Janrain mobile documentation - <http://developers.janrain.com/documentation/mobile-libraries/>
* PRXClientDesign\_v0.4.docx

# ARCHITECTURAL ANALYSIS

## Product Registration Overview

Product registration is one the features of the mobile applications developed in Philips. It allows user to register a product that he/she owns after which it can be used for extending warranty.

As part of CDP2 Product registration is a common component that can be reused across various applications. The subsequent sections provide the detail of the product registration as a re-usable component.

## System Context

The following diagram shows interaction of Mobile application with various systems. The document concentrates on the direct touch points of mobile application.

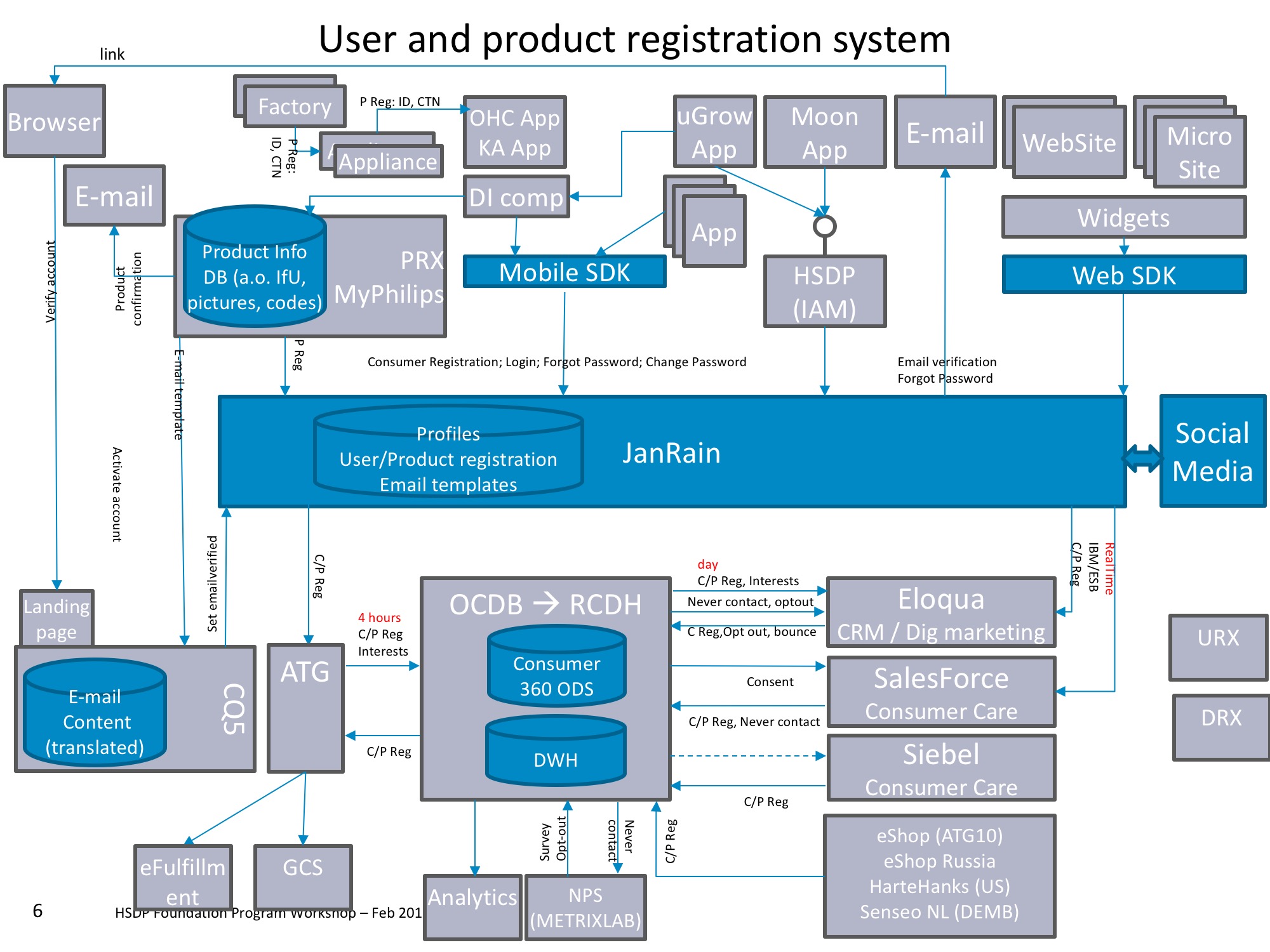


Figure 1: System context

## Design Method

The design used Object Oriented concepts which are supported by majority of the platforms like Objective C, Swift and Java. The design patterns like Façade, Delegation, Notification and Extensions are used.

All communication with server (Janrain, PRX) will be using SSL certificates. Plain HTTP communication will not be used.

The product registration Component will not expose any internal APIs and classes. This is to avoid applications bypassing the product registration component and directly invoking internal functions thus compromising the rationale behind horizontal component.

The components follow the layered approach. Following rules are enforced for how the layers interact with each other.

* **One way interaction (Top to down)**: Higher level layers can interact with layers below, but lower layers cannot interact with layers above. This is to avoid circular dependencies between layers.
* **Strict interaction**: Layers can only interact with layers below. This will make sure that modifications in one layer will only affect layers above.

## Platform dependency

While Object Oriented concepts are used in iOS and Android, the implementation details differ. The following sections provide a brief overview of such implementation details.

### Class names

The class names used here can be modified based on the platform. In iOS, class names are prefixed with product names (Ex: UserProduct class is implemented as PPRUserProduct. PPR indicates Philips Product Registration). In Android it is associated with package names like com.philips.cdp.prodreg.UserProduct

### Errors

The design here provides an abstract of error and not the implementation details.

# STATIC DESIGN

## Overall System View

The following diagram shows the over system architecture of product registration using PRX.

Application

Appliance

CTN,   
Serial nr.

CTN,

Serial nr.

Factory

Product registration

Email confirmation

Janrain mobile SDK

PRX

Janrain

CQ5

ATG

OCDB

Eloqua  
Digital marketing

SalesForce  
Consumer Care

Siebel  
Consumer Care

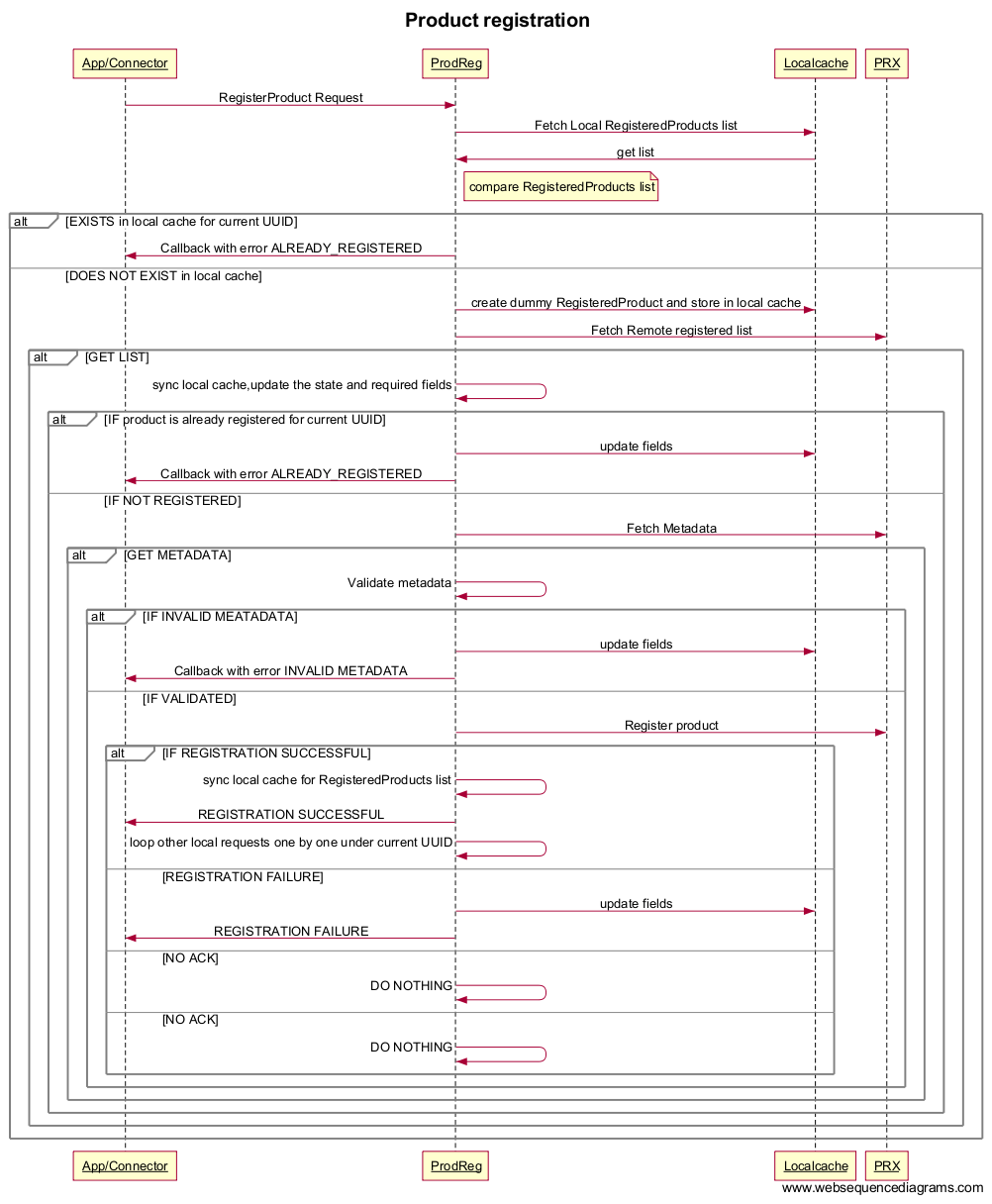
Figure 2: Product Registration system view

The diagram only considers mobile application as the touch point for user and product registration. It is also possible to perform registration from website, consumer care and from products directly but that is out of scope of this document. It is considered that Philips IT will take care of handling those touch points.

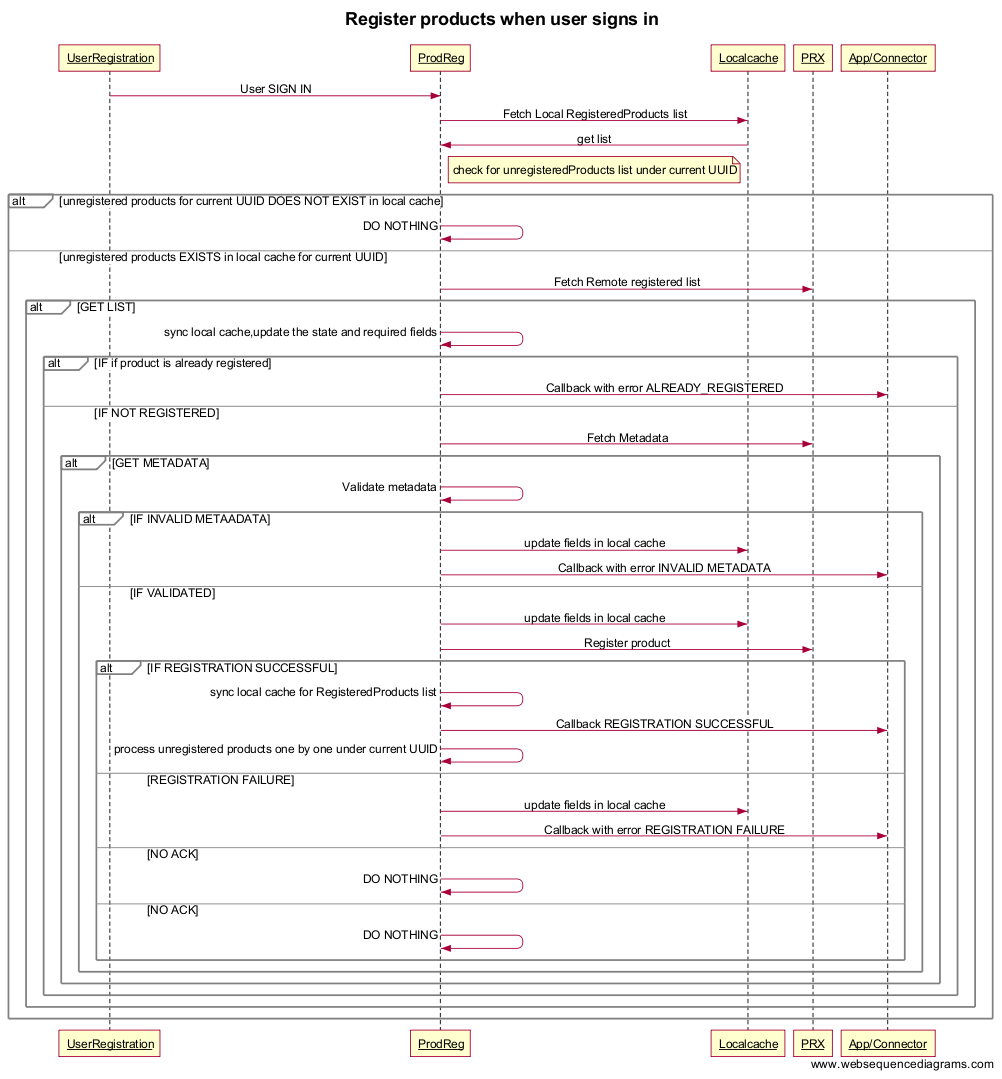
The mobile application interacts with various systems in order to perform user and product registration. The focus of this document is to “unify” all these components (PRX, Janrain SDK, backend API) into a re-usable mobile component that can be used across various mobile applications.

## Sequence

### Product registration sequence ( Online/Offline )



### Product registration when user signs in



## Layering Model

The following diagram shows the top level design of product registration block.

User Registration

Product Registration

PRX

Janrain

Figure 3: Top level design of Product Registration Component

The design consists of following blocks.

* Application – This represents any vertical application which uses product registration component.User registration component is mandatory for product registration.
* User Registration – This component exposes the classes for applications to invoke various user registration features where in user can register/login in Philips using My Philips account or any social providers account.
* Product Registration– This component exposes API or user interface where in user can register the products which he/she owns.
* Janrain – This platform provides mobile sdk both on Android and IOS and also exposes REST APIs to facilitate user/product registration.

#### Class diagram

The following figure shows the class diagram for the important classes used inside product registration.

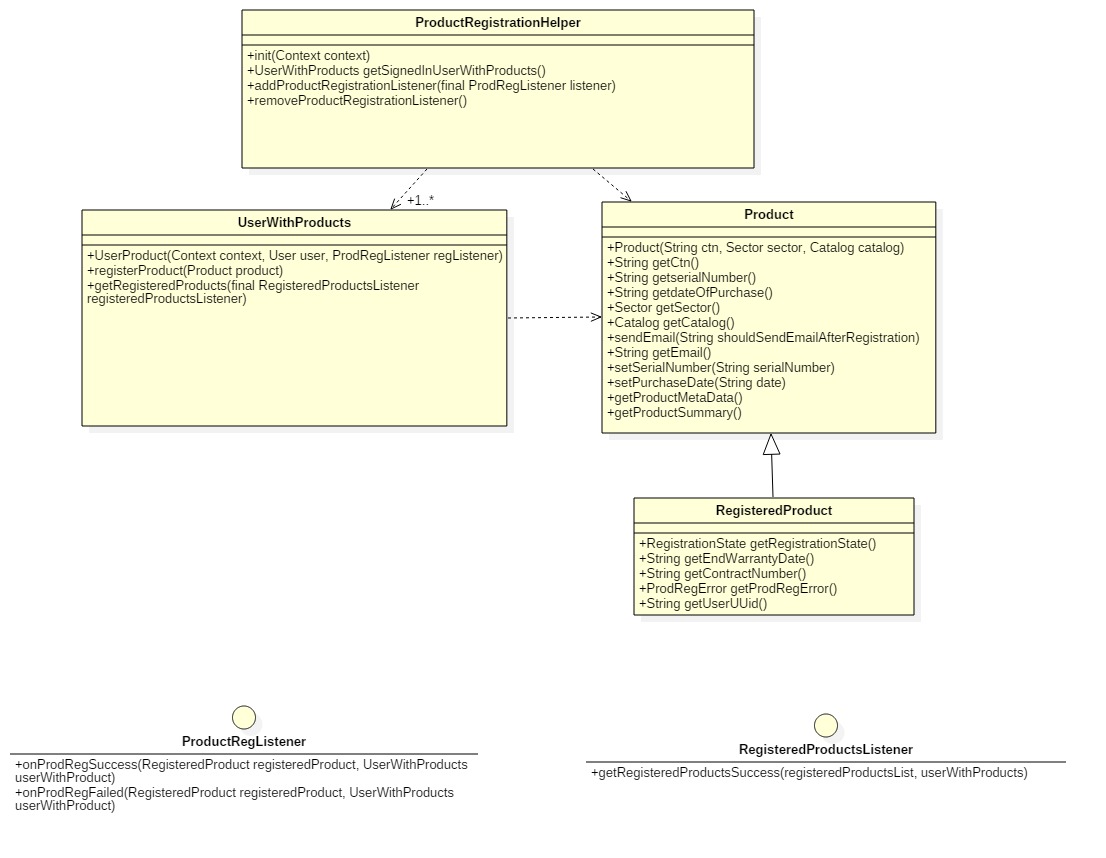


Figure 4: Product registration component class diagram

The product registration component depends on PRX client component for execution of PRX API requests.

Please refer PRXClientDesign\_v0.4.docx for more details.

**ProdRegistrationHelper:** It is a class consisting of public interface APIs exposed for applications.

**UserWithProducts:** A class which associates a user with the product which needs to be registered.It also provides list of registered products for a specific user.

**Product:** A class denotes a product consisting of product details like CTN, serial number, date of purchase, sector and catalog.

**RegisteredProduct:** This is a subclass of product class which contains registered product information.

**ProdRegListener:** An interface for providing call backs to application after registration request. This is specific to Android otherwise the completion blocks contains the methods of this class as per diagram.

**RegisteredProductsListener:** An interface which provides callbacks after fetching the list of registered products for a user. This list includes products registered both locally and remotely.

# DYNAMIC DESIGN

## Use cases

### Use case: Fully automatic

DI comm/Bluelib

Application

CTN, Serial nr.

CTN, Serial nr.

Factory

CTN, Serial nr.

Appliance

DIComm/Bluelib

Connector

Product

registration

Packaging

### Use case: Semiautomatic case

Factory

Application

CTN, Serial nr.

Appliance

CTN, Serial nr.

CTN, Serial nr.

Packaging

Product registration

.

# PRX backend

PRX backend component provides the product registration features. This component contains the APIs to interact with REST interfaces.

The following functions are provided by PRX backend.

* Register the product
* Fetch the registered products w.r.t user
* Fetch the product summary
* Fetch the product metadata

All services exposed by PRX and Philips backend uses https.

The diagram below shows how the app registers product using PRX APIs. Mobile app makes a call to PRX server to register product using access token.

*(Picture reference: C020\_4-Identity\_Access management-Product registration.pptx)*

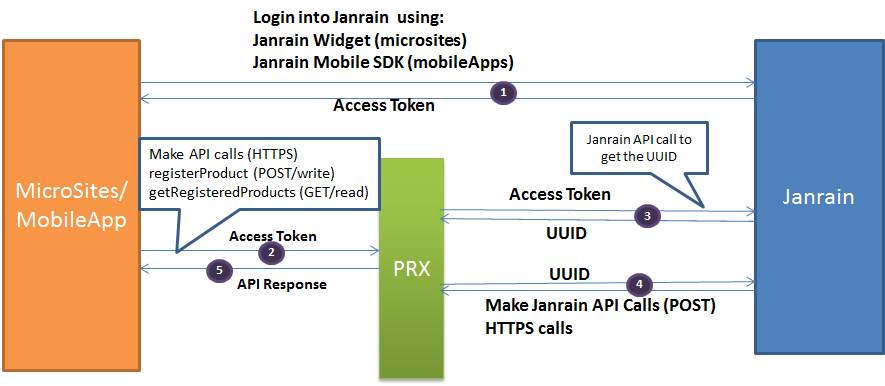


Figure 6: Usage of Philips service for product registration