

Software Architecture Base App  
<AjayDas20160629-01V01>

BMS-REA-214

Approval

Name	Function	Date	Signature
Bhargavi Upadhya	System Architect		

Revision History

Date	Revision number	Status	Author	CR/P R ID	Changes/Comments
2016-06-29	0.1	Draft	Ajay Das		First version
2016-07-20	0.2	Draft	Deepthi Shivakumar		Added micro app framework section
2016-09-23	0.3	Draft	Deepthi Shivakumar		Modified documents with latest updates. Moved micro app section to different document.
2016-11-12	0.4	Draft	Deepthi Shivakumar		Addressed review comments and added more details about flow manager.

Printed copies are uncontrolled unless authenticated

# Software Architecture Base App

<AjayDas20160629-01V01>

BMS-REA-214

## 1 Table of Contents

Approval .....	1
Revision History .....	1
<b>1 Table of Contents .....</b>	<b>2</b>
<b>2 Document Introduction .....</b>	<b>3</b>
2.1 Purpose and Scope .....	3
2.2 Intended Audience .....	3
2.3 References .....	3
2.4 Definitions, Acronyms and Abbreviations .....	3
<b>3 Product Overview .....</b>	<b>3</b>
3.1 Purpose and Intended Use .....	3
3.2 Network Security .....	4
3.3 Architectural Goals and constraints .....	4
3.3.1 Goals .....	4
3.3.2 Constraints .....	4
<b>4 Software Architecture .....</b>	<b>5</b>
4.1 Decomposition .....	6
4.1.1 App framework .....	6
4.1.2 uApp framework. ....	7
4.1.3 Common components and micro apps. ....	7
4.2 Third party and open source software .....	8
4.3 UI Flow .....	8
4.3.1 Configurable User interface .....	8
4.4 Development/Implementation view .....	9
4.5 Deployment view .....	10

## Software Architecture Base App

&lt;AjayDas20160629-01V01&gt;

BMS-REA-214

## 2 Document Introduction

The purpose of this document is to identify the architectural blocks and explain the top level design of Base App for iOS and Android. The document provides an overview of the various functional blocks of base app, design choices made and guidelines to few elements. This forms a reference for detailed design and implementation.

### 2.1 Purpose and Scope

This document describes the high level software architecture of the Base app. The contents of this document apply to both the iOS and Android.

The interface and detailed design for each component is out of scope of this document.

Below are the core components of Base app, it is built for Android phones with version 4.4 and above and IOS 8.4 and above.

- Data
- Navigation and Logic
- Content
- Common Components
- Platform centric approach

### 2.2 Intended Audience

CDP2 Architects and platform development team.

### 2.3 References

Reference	Identification	Title / additional remarks
[REQUIREMENT]		
[UAPP FRAMEWORK]		UAPPFramework_Guidelines_1.0.0
[APP_SIZE]		App size optimization version 0.3
[INTERNATIONALIZATION]		App UI Internationalization, Version 1.0
[CM_PLAN]		SCM Plan
[RULES_ENGINE]		

### 2.4 Definitions, Acronyms and Abbreviations

Term	Description
MVP	Model View Presenter
MVC	Model View Controller
UI	User Interface

## 3 Product Overview

### 3.1 Purpose and Intended Use

This document describes the high level software architecture of the Base app. The contents of this document apply to both the iOS and Android .

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 3 of 10	Last modified:	2016-12-07

## Software Architecture Base App

&lt;AjayDas20160629-01V01&gt;

BMS-REA-214

**3.2 Network Security**

To be updated.

**3.3 Architectural Goals and constraints****3.3.1 Goals****3.3.1.1 Extensibility**

The base app is intended as a platform for all next CDP2 propositions. As such the base app needs to be extensible with sensors, algorithms and user interfaces relevant for upcoming propositions while keeping functionality of micro apps. This is done by making use of state machine for user interface and standardized interfaces with respect to common components.

**3.3.1.2 Compatibility**

The Base app should build modular UI components to handle upgrades and extensions. UI interfaces should be developed adopting the responsive design, density independent layouts and constraints based interface principle in order to support current and future smartphones/tablets with different properties.

**3.3.1.3 Design rules**

The design uses Object Oriented concepts which are supported by iOS and Android. The design patterns like State and MVP are used.

User Interface is built from different micro components, each user interface component or micro app in base app exhibits different behavior based on the state of the user interface. Flow manager manages all the states in the App flow. App flow consist of a set of states. App flow is a transition from one state to another.

uApp framework provides the basic interface that all micro apps will have to adhere to plugin in to base app.

**3.3.2 Constraints****3.3.2.1 Device constraints**

App Framework is optimized for smartphones and compatible to tablets. The below table lists various device constrains in App design and development.

Constraints	Risk	Mitigation
<b>OS version</b>	App incompatible with new versions of OS - inconsistent style - incompatible rights -	Minimum OS requirement iOS>=8.4 Android >=4.4  Performance: Check on first use/update if real time requirements can be met. Should guard for incorrect results, yet users are likely unhappy if their mobile is stated as unsupported.
<b>Screen size</b>	Incorrect rendering Incorrect use of real screen estate	Responsive layout Test on range of products
<b>Devices</b>	App incompatible with must support devices.	iOS: iPhone 5 and above Android: Huge device diversity. Test on hero list of devices and with different device + OS version combination.

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 4 of 10	Last modified:	2016-12-07

## Software Architecture Base App

<AjayDas20160629-01V01>

BMS-REA-214

<b>Performance</b>	Sluggish UI behavior	The response to the user actions should be completed in less than the human perceivable duration (Typically 200 msec). Operation that take more time (Ex: network operations) should not be executed in UI/Main thread.
<b>App size</b>	Huge app size	The guidelines provided in [APP_SIZE] shall be enforced.
<b>Localization/Internationalization</b>	App becomes inflexible to add new language or support more countries.	The guidelines provided in [INTERNATIONALIZATION]
<b>Security</b>	The data stored in the app can be viewed by hackers.	Encrypt the data stored locally in the app using the guidelines provided in TBD.

## 4 Software Architecture

The purpose of the base app is to provide a quick starting point for application development. The base app is a base application that pre-integrates all CDP2 common components. Base app demonstrates how to build configurable different UX flows that can be further configured by proposition to build their own UI flows.

Base app guarantees that the combination of integrated common component versions functions properly. Although the base app delivers a functional application, it is the proposition's responsibility to deliver an application that meets the intended user requirements. This can be achieved by configuring the framework, removing unrequired components, and adding proposition specific logic.

The following diagram illustrates the top level architectural view of App Framework with interfaces to external System.

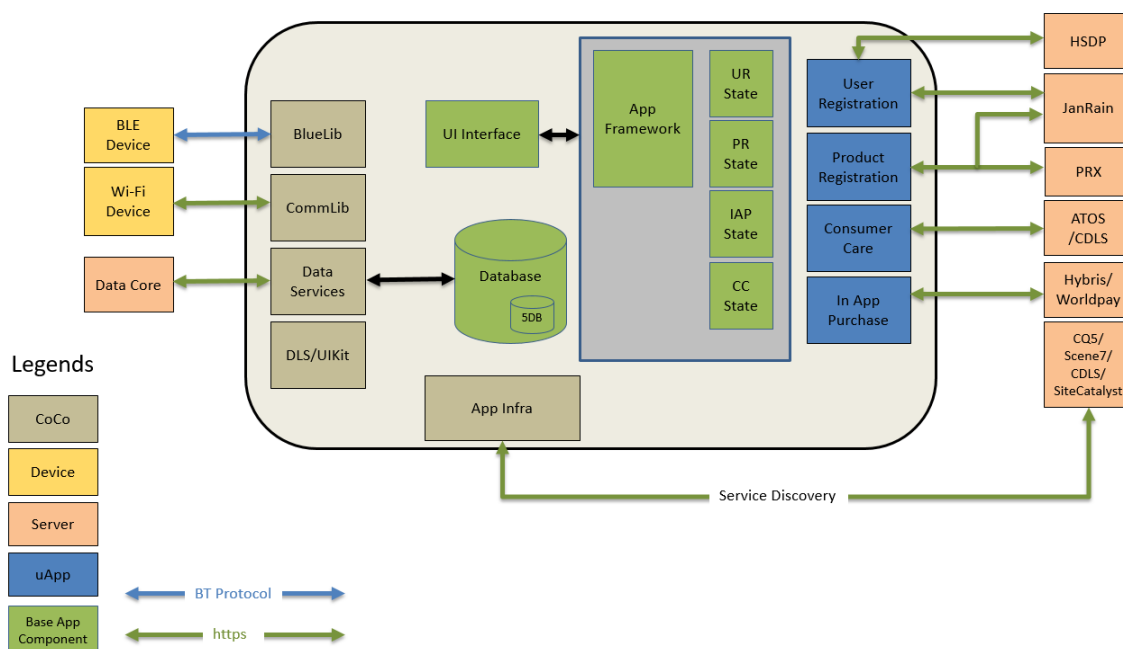


Figure 1: Top level architecture

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 5 of 10	Last modified:	2016-12-07

## Software Architecture Base App

&lt;AjayDas20160629-01V01&gt;

BMS-REA-214

User Interface in base app consists of the Model, view and the Presenter associated with the view. The User Interface sends user actions to the presenter. The user interface works on the principle of state machine. UX flows are composed of states and UI navigation is state transitions.

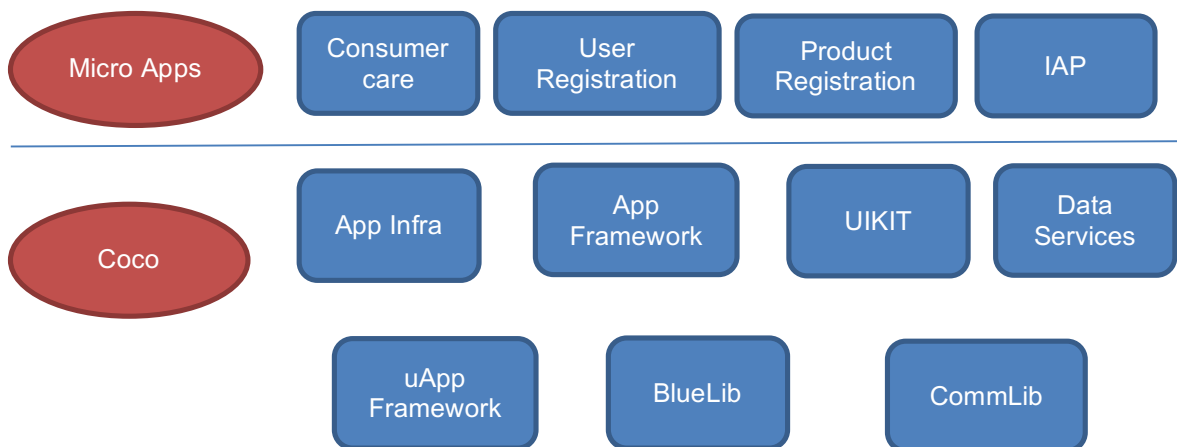
Each flow element is represented as state and base app works on the principle of state machine. Example: UR state represents user registration micro app which means base app will be in UR state when it launches user registration micro app.

Database is local to application which is used for storing moments data.

Each component talks to different backend system as represented in above diagram.

#### 4.1 Decomposition

The following block diagram provides a high-level overview of the functional components in the base app.



The core functionality of Base app is to provide a quick starting point for application development. It is a base application that pre-integrates all CDP2 common components.

##### 4.1.1 App framework

###### Main responsibility:

Currently app framework is not formed as a component yet and it is part of Base app.

It contains Base state class and Flow manager.

Any flow element can be represented as state. For example each micro app can be represented as UI state in an app.

Flow manager is basically a class part of app framework which helps in transitioning from one state to another state in the app flow.

The decision to move to next UI state in the flow is taken from JSON file named "AppFlow.json"

There are set of states in which an event can occur which leads to transition in new state based on few conditions.

Example : When you are in User registration state, you get logged in as an event and you would move to next state based on conditions as mentioned below.

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 6 of 10	Last modified:	2016-12-07

## Software Architecture Base App

&lt;AjayDas20160629-01V01&gt;

BMS-REA-214

An application need to define the app states, possible events in those events and conditions to move to next state.

Conditions are executed sequentially, if the first condition is met, app would move to the next state as defined in the flow otherwise iterate through next conditions.

**Constraints:**

- To check for multiple conditions through json is not supported yet.
- Backward navigation through json is responsibility of propositions where in platform considers back as an event.
- Creation of app flow via cloud is not tested yet though APIs are supportive.

## 4.1.2 uApp framework.

**Main responsibility:**

There is a separate library or framework defined to set guidelines and rules for micro app development.

uApp framework provides the basic set of interfaces that all uApps will have to adhere to plugin in to platform.

Please refer to uAppFramework\_Guidelines\_1.0.0 document to know more about app and micro app roles and responsibilities.

uApp interface provides a common interface to plugin any micro app in base app wrt the below,

- Common interface for performing init and launch of uapp.
- Unified way of handling action bar/Navigation bar
- Handling Back key
- Handling dependencies
- Handling Configuration
- Common interface or protocol for a micro app interface

## 4.1.3 Common components and micro apps.

Common component	Description
<b>User Registration</b>	The User Registration component provides a standard user experience for the user registration flow including social login. Base app integrates this component which abstracts the interaction towards the HSDP identity management service and various SDKs for social login and Janrain SDK.
<b>BlueLib</b>	BlueLib handles the communication between App and Bluetooth device. The services and characteristics specific to device are implemented using BlueLib plugin.
<b>InApp Purchase</b>	IAP is the component that provides interface to purchase products, services and accessories via mobile app. It provides the User Interface, connection to payment gateway and store.
<b>Philips UI Kit</b>	Philips UI Kit component provides the interface to create UI elements that are complaint with Philips design guidelines for Mobile Apps.
<b>CommLib</b>	Comm lib is a component that manages the app to device communication via wifi using dicomm protocol.
<b>Consumer Care</b>	The consumer care component is an off the shelf component providing consumer care functionality like customer care contact information,

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 7 of 10	Last modified:	2016-12-07

## Software Architecture Base App

&lt;AjayDas20160629-01V01&gt;

BMS-REA-214

	product information, product FAQs, service centers details, reviews and ratings etc.
<b>Product Registration</b>	The Product Registration component provides a standard user experience to register product under a registered user.
<b>App Infra</b>	App infra provides functionality that is common for most of the propositions and common components. It provides a base layer for optimal app development. App Infra provides logging , tagging , secure storage , service discovery and many other functions.
<b>Data Services</b>	It is developed as a common component that helps in syncing moments data between local data base and data core cloud.

## 4.2 Third party and open source software

A list of all used third party software and relevant licenses is described in the configuration management plan [CM\_PLAN]

## 4.3 UI Flow

### 4.3.1 Configurable User interface.

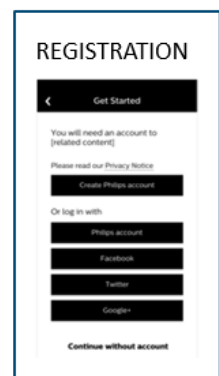
The User interface is build on the concept of micro app. The micro app boundaries are well defined. The microapps do not communicate with each other. The microapps communicate through the flow manager. The flow manager manages all the flow in the app. Navigation across micrapps is based on the concept of state machine.

The flow manager creates the state and navigates the app to the next state as defined.

- **Navigation and Logic**

- > State machine to control navigation flow.
- > UI is represented as different states and navigation flows from one state to another.
- > Navigation and business logic are separated from the user interface.
- > States are configurable and UI is configurable.

- Enum State
- {SPLASH,
- WELCOME\_STATE1,
- WELCOME\_STATE2,
- REGISTRATION,
- HOME,



1. A particular flow in the app is managed by the flow manger, A flow is a transition across the states.
2. UIStateBase is the base class for the UIState.
3. UIState is the container that holds the state, It holds the navigation and logic for UIview.

Version:	<0.1>	Classification:	For internal use only	Author:	Ajay Das
Status:	<Proposed>		Page: 8 of 10	Last modified:	2016-12-07



Software Architecture Base App  
<AjayDas20160629-01V01>

BMS-REA-214

The figure below shows the class diagram of Configurable UI.

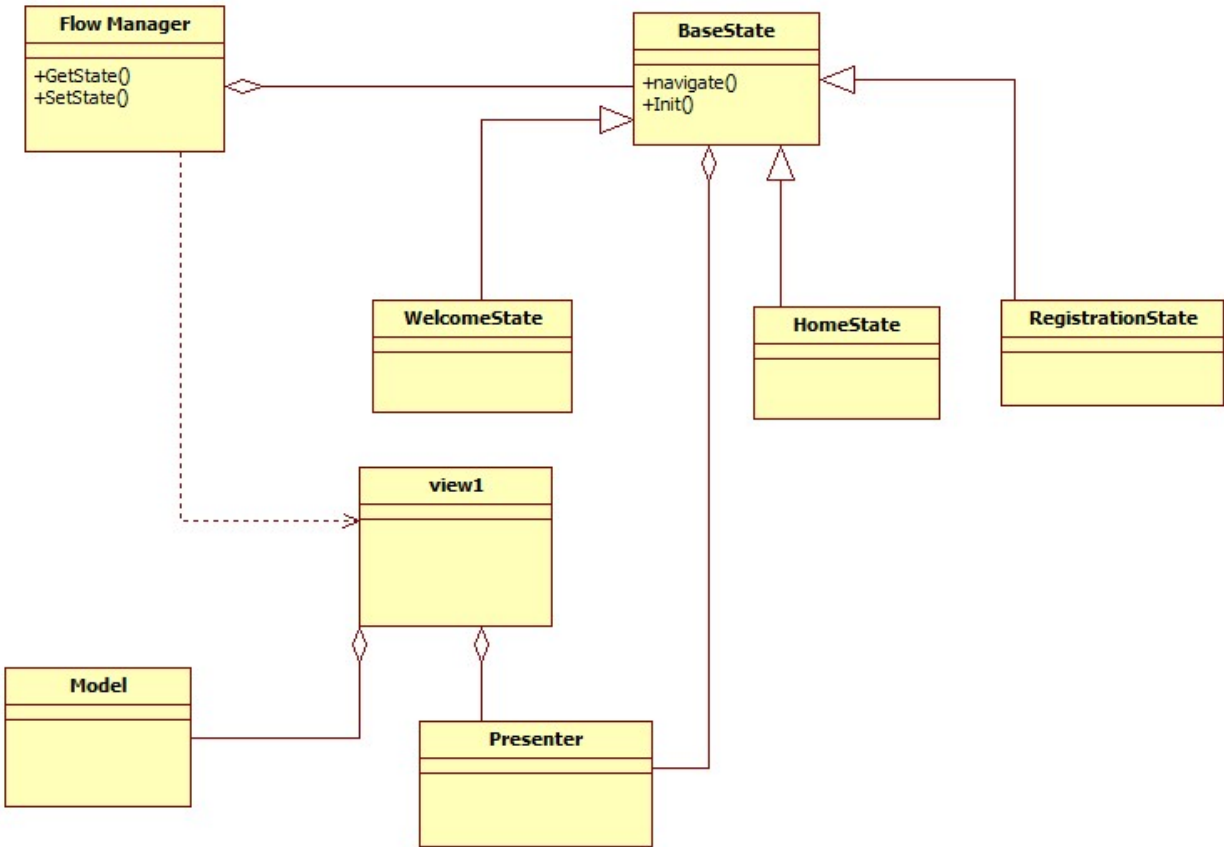


Figure 1: Configurable UI design

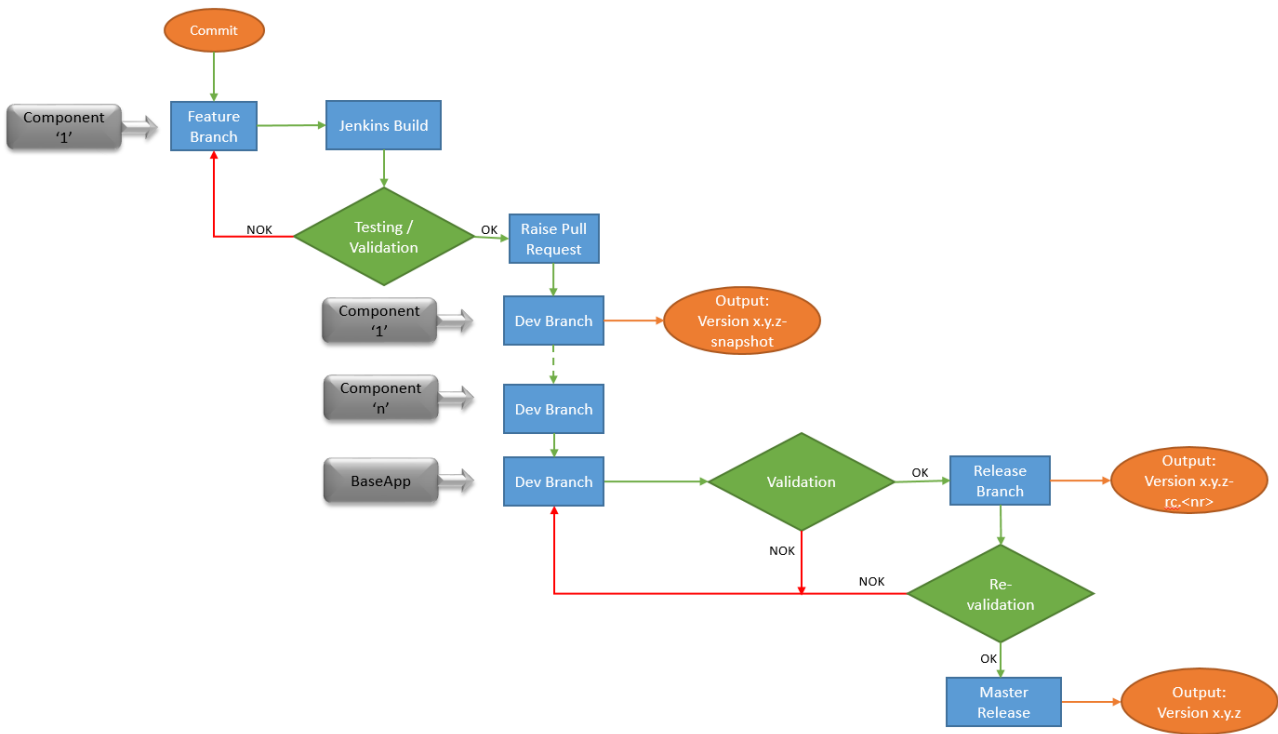
4.4 Development/Implementation view

Refer to Configuration Management document [CM\_PLAN].

Printed copies are uncontrolled unless authenticated

Software Architecture Base App  
<AjayDas20160629-01V01>

BMS-REA-214



4.5 Deployment view

Base app is released as a platform in terms of source code with appropriate versioning which comes with bundle of cocos and uApps pre integrated and reference UI flow which represents common user journey among propositions.

Vertical apps will use base app as starting point and will be deployed via Apple App Store and Google Play Store by adding proposition specific functionalities.

Printed copies are uncontrolled unless authenticated