**Auto Connectivity Workflow**

**Software Design Document**

By

*Ayala Barzilay*

Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Description** |
| 1.0 | November 20th, 2018 | Preliminary |
|  |  |  |
|  |  |  |

**Table of Contents**

[1. Scope 4](#_Toc398821515)

[2. Acronyms & Abbreviations 4](#_Toc398821516)

[3. Referenced Documents 4](#_Toc398821517)

[4. Class Diagram 5](#_Toc398821518)

[4.1. Project 5](#_Toc398821519)

[4.1.1. Plan .vs Project 6](#_Toc398821520)

[4.1.2. Save Project 6](#_Toc398821521)

[4.1.3. Load Project 8](#_Toc398821522)

[4.1.4. Export project 8](#_Toc398821523)

[4.1.5. Import project 9](#_Toc398821524)

[4.1.6. Recent SCBProjects project 9](#_Toc398821525)

[4.2. Donor and Small Cell 10](#_Toc398821526)

[4.3. SWWO – SWWOOut 10](#_Toc398821527)

[4.4. Connection 11](#_Toc398821528)

[4.4.1. Create Connection 12](#_Toc398821529)

[4.4.2. Promote Connection to link 12](#_Toc398821530)

[4.5. COWO - COWOOut 13](#_Toc398821531)

[5. Use cases 14](#_Toc398821532)

[5.1. Add connection 14](#_Toc398821533)

[5.2. Generate SWWO 14](#_Toc398821534)

[5.3. Pull SWWOResults from DB 15](#_Toc398821535)

[6. QlikView Integration 16](#_Toc398821536)

[6.1. Qlik View data 16](#_Toc398821537)

[6.2. Qlik View Tab’s content 16](#_Toc398821538)

[6.3. Licensing 16](#_Toc398821539)

[7. DB or XML 16](#_Toc398821540)

[8. DB Architecture 17](#_Toc398821541)

[8.1. Motivation 17](#_Toc398821542)

[8.2. Linq2 SQL vs EF 17](#_Toc398821543)

[8.3. Linq2SQL to EF migration 18](#_Toc398821544)

[8.4. Passing data Client - > Server 18](#_Toc398821545)

[8.4.1. Json.NET 19](#_Toc398821546)

[8.5. DB Tables 19](#_Toc398821547)

[8.5.1. Project 19](#_Toc398821548)

[8.5.2. Connections table 20](#_Toc398821549)

[8.5.3. SiteWalkWO table 20](#_Toc398821550)

[8.5.1. SiteWalkWOOut table 21](#_Toc398821551)

[8.5.2. SiteWalkWOOutPerFrequency table 21](#_Toc398821552)

[8.5.3. CommissioningWO table 21](#_Toc398821553)

[8.5.4. CommissioningWOOutput table 22](#_Toc398821554)

[9. GUI 23](#_Toc398821555)

[9.1. SCB Projects Tab 23](#_Toc398821556)

[9.2. SCP Project Settings: 25](#_Toc398821557)

[9.3. SCB Layers 26](#_Toc398821558)

[10. Assigning Technician to a project 27](#_Toc398821559)

[11. R-Sync program 27](#_Toc398821560)

[12. Open issues 27](#_Toc398821561)

# Scope

This document covers software design of auto connectivity.

# Acronyms & Abbreviations

The acronyms and abbreviations used within this document are detailed in the following table:

| **Acronym** | **Description** | **Acronym** | **Description** |
| --- | --- | --- | --- |
| CBW | Channel Bandwidth |  |  |
| HBS | Base Transceiver Station (a.k.a. BTS) |  |  |
| HSU | Subscriber Unit (a.k.a. CPE) |  |  |
| DL | Down-Link Traffic (HBS to HSU) |  |  |
| UL | Up-Link Traffic (HSU to HBS) |  |  |
|  |  |  |  |
| DB | Data Base |  |  |

**Important Terms**

| **Term** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Referenced Documents

The following referenced documents might be used for additional clarifications.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref No** | **Document** | **Author** | **Location** |
|  | [R-Planner - 3.11 System Specification](http://confluence:8090/display/SYS/R-Planner+-+3.11+System+Specification) | Ori Rotem | http://confluence:8090/display/SYS/R-Planner+-+3.11+System+Specification |
|  | [R-Planner 3.11- Auto-connect](http://confluence:8090/display/TOOL/R-Planner+3.11-+Auto-connect) | Moshe | http://confluence:8090/display/TOOL/R-Planner+3.11-+Auto-connect |

### Communication between WPF application and Angular Application

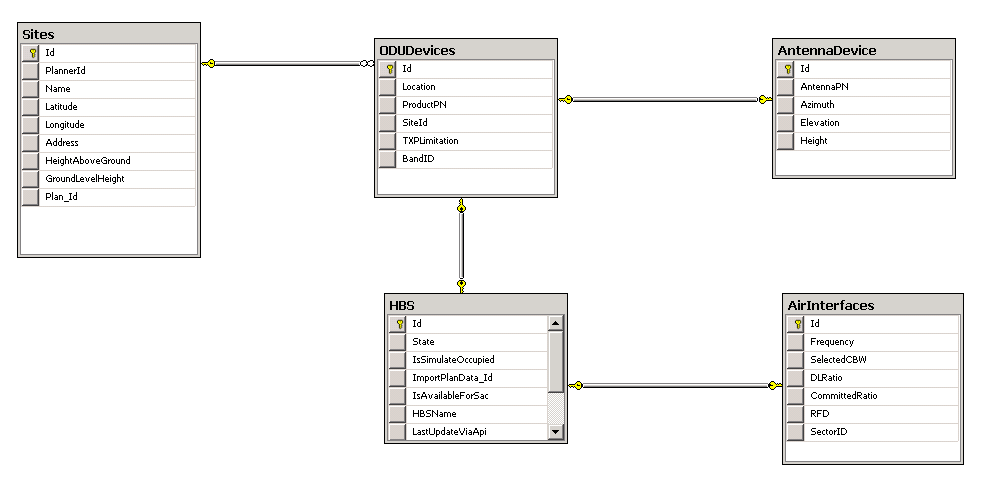
### Save plan to mongo DB

Planner client will serialize the plan's relevant data structures and will send it a JSON objects to RWS micro service which will save it in to the DB.

#### Project schema SQL

# 

#### HBS schema SQL



#### HSU schema

No

#### Link schema

No

Details

Currently the RPlanner have the following class hierarchy which is serialized to binary stream and saved on the server side with .prj suffix

Plan file

* Last modified
* version
* Plan file header
  + Name
  + Description
  + Plan size
* Hash table
  + Links
  + Sites
  + Plan

# Use cases

## Input wizard

## Pre-connectivity

Progress bar

### Auto-connectivity Validation

Currently the validation is done on the server one by one.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The progress is calculated on the client side. | Performance vise – the communication to the server is slow |

Read file stream (client) and send the file stream to the server (IPlannerREST endpoint)

Accept the stream on the server and read it in to the CSV objects (Move those classes to RPlannerServer)

* HSUInfo
* SiteAndHBSInfo

Perform the validation on the server

Image: use case - validation

### Auto-connectivity LOS

Image: use case – LOS check

### Auto-connectivity UL & DL utilization

### Auto-connectivity Algorithm

# QlikSence Integration

TBD

# Mongo or SQL or direct communication with Angular

## Direct communication with Angular

Project /HBS/ HSU / Links will be passed as JSON to the Angular client app

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| No DB management is required | This way the angular client is not standalone,  The information will not be reachable outside of RPlanneClient, short term solution |
| Existing plan - Passing lots of information between WPF client & Angular client will be faster | From file - The preconnectivity is done on server side, no need to parse it on client, so passing the information to RPlannerClient and then to Angular client is redundant. |
|  |  |

There is a need to share information between the following tools:

* RPlanner
* RWS

Currently the RPlanner save its data using binary serialization. In order to share the relevant information between the above application the RPlanner will save the required information in the DB.

Using DB is preferred than using xml serialization for the following reason:

## SQL

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Already has an infrastructure to handling pz\_users, project, site, HBS and working properly - saves few weeks of development. | More difficult to change the DB structure |
| Working with EF – entity framework is convenient. | Requires license |
|  | Server is expensive relative to the alternative |

### Authentication to SQL

TBD

## Mongo

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| It is a document DB, the structure can change in the future with no problem | New micro service is needed to handle CRUD operations on the Plan |
| Will be used in the future by the micro services environment |  |
| No license required |  |
| Will be dockerized – no server is required |  |

### Authentication to Mongo

TBD

# DB Architecture

It is important to understand, that currently the RPlanner mechanism of saving data by using binary serialization and it keeps being relevant the RPlanner Client save , load plan functionalities.

The plan saved in the DB will be relevant new functionalities in RWS as Auto-connectivity & SET, etc.

## DB Tables

### Project

|  |  |
| --- | --- |
| ID | Key (int) (Is actually the planID) |
| LastUpdate | DateTime |
| UserContactID | nvarchar(MAX)\* |
| Name | nvarchar(1024) |
| Customer Name | nvarchar(1024) |
| City | nvarchar(1024) |
| Description | nvarchar(MAX) |
| Required Uplink Tput | int |
| Required Downlink Tput | int |
| BrandID | Foreign key(int) |
| CatalogID | Foreign key(int) |
| FrequencyBandId | FrequencyBandId |
| Default DONOR product PN | nvarchar(128) |
| Default SmallCell product PN | nvarchar(128) |
| Allowable Channels | nvarchar(1024) ( 11011000101001 …) \*\* |
| Default Channel Bandwidth (MHz) | int |

\* **UserContactID** Actually this field should be foreign key to the sfContactTable, but due to the fact that the RADWIN, RAD and Ceragon users’ contactID reside on different tables, It would be a regular field of type (nvarchar(MAX)).

While querying the DB the RPlanner would join sfContactTable and Project tables.

While querying AirmuxPlanner we join Projects and RadUsers table.

**\*\*Allowable channels** field is of type string and build of “1” & “0” characters.

1 is available

0 is close

According to the “Default Product” and to the “Default Channel Bandwidth” the allowable channels could be restored.

### Project Table

### HSU table

### HBS table

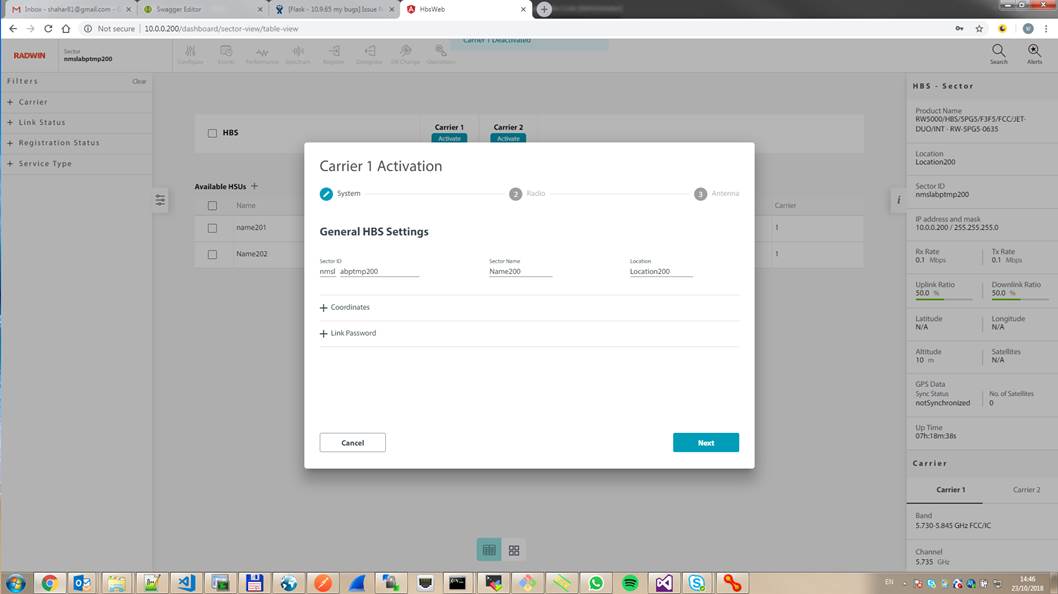
### Link table

|  |  |
| --- | --- |
| ID | Key (int) |
| ProjectID | Forigen key (int) |
| SMALL CELL ID | int |
| DONOR ID | int |
| Frequency | float |
| Channel BW | int |
| Required Throughput Uplink | float |
| Required Throughput Downlink | float |

### Site table

# GUI

Wizard



## 

TBD, fine-tuning to the wizard, reuse of the Web-UI

### Projects Tab

### Auto-connectivity Layers

Some new layers which enable the user to see: (multiple choice)

* accomplished Site survey
* accomplished Commissioning

# Open issues