< **Device Management** > Business Requirements Document

Version 1.0

Votary Softech Solutions Pvt. Ltd.

Plot No: 76, Lumbini layout,   
Near Euro school,   
Gachibowli-I (V), Hyderabad,  
Telangana - 500032,   
India.

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version (x.y) | Date of Revision | Description of Change | Reason for Change | Affected Sections | Approved By |
| 0.1 |  | Initial Draft | New Document | All |  |
| 0.2 | 19/04/2016 | Team review and improvements |  |  |  |
| 1.0 | 21/04/2016 | More diagrams and implementation |  |  |  |

**Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version (x.y) | Prepared By | Reviewed By/Date | Approved By/Date |
| 1.0 | DM team | Mohan /Ashok |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[1 Define 4](#_Toc480805295)

[1.1 Introduction: 4](#_Toc480805296)

[1.2 Customer: 4](#_Toc480805297)

[1.3 Target Audience: 4](#_Toc480805298)

[1.4 Objective 4](#_Toc480805299)

[1.5 Deliverables 5](#_Toc480805300)

[1.6 Prerequisites 5](#_Toc480805301)

[1.7 Assumptions 6](#_Toc480805302)

[1.8 Limitations 6](#_Toc480805303)

[1.9 Business Risks 6](#_Toc480805304)

[1.10 Glossary 6](#_Toc480805305)

[2 Specify 7](#_Toc480805306)

[2.1 Scope 7](#_Toc480805307)

[2.2 Schedule and Milestones 7](#_Toc480805308)

[2.3 Resources 8](#_Toc480805309)

[2.4 Cost 8](#_Toc480805310)

[2.5 Market Analysis 8](#_Toc480805311)

[3 Architecture 10](#_Toc480805312)

[3.1 Block Diagram 10](#_Toc480805313)

[3.2 Internal Interfaces 10](#_Toc480805314)

[3.3 External Interfaces 10](#_Toc480805315)

[4 Design 11](#_Toc480805316)

[4.1 Flowcharts 11](#_Toc480805317)

[4.2 Development Environment 11](#_Toc480805318)

[4.3 IP Identified 11](#_Toc480805319)

[5 Implement 11](#_Toc480805320)

[5.1 System Requirements 11](#_Toc480805321)

[5.2 Revenue Realization Plan 15](#_Toc480805322)

[5.3 User Acceptance Criteria 15](#_Toc480805323)

[5.4 Quality Plan 15](#_Toc480805324)

[6 Validate 15](#_Toc480805325)

[7 Deploy 15](#_Toc480805326)

[8 Maintain 16](#_Toc480805327)

# Define

VITA Device management is IoT gateway firmware which enables application to discovery and engage IoT devices. DM is hardware & connectivity agnostic. This is an independent and pluggable module into any solutions.

Features:

* Detection of the various devices (BLE, IP based, WIFI etc)
* Connection of the devices
* Control/Read/Write of the devices
* Data parsing and providing data to other modules (such as storage management, smart app, custom specific)

## Customer:

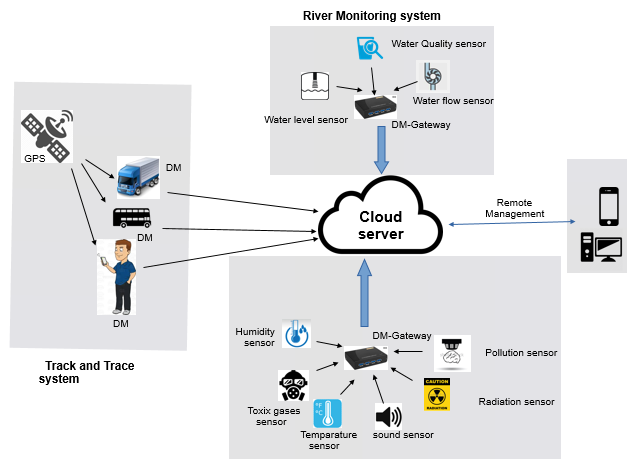
This is general purpose architecture of Votarytech. This will be baseline and roadmap for future projects in IOT and M2M space.

## Target Audience:

This is baseline document for the VITA developers, Architects and stakeholders.

## Objective

DM gateway firmware shall enable application to discover, control IoT devices independent of how they connected to Gateway (BLE, IP-Ethernet, IP-Wifi)



**Market Need:**

This should be device independent module, which can be used as template for future (internal or external) product line.

## Deliverables

* VITA Device Management library on x86 and ARM
* User API documents [as reference]

## Prerequisites

**Software Prerequisites**

* Software tools
  + QT
  + GCC tool chain
  + GIT
  + g++

**Hardware Prerequisites**

* Hardware/boards

|  |
| --- |
| * RaspberryPI (ARM Arch) |
| * Desktop (x86 ) |
| * BLE-beacons * BLE-USB dongles   **Technology Prerequisites** |

* IOT industry specifications

## Assumptions

* DM APIs are called through SA with help of TL (Transport Layer)
* DM pushes data to SM using TL (Transport Layer) APIs

## Limitations

* DM can handle one application at a time.
* BLE range is 30 feet
* Wifi range is 60 meters

## Business Risks

**Market Risks**

* Hugely competitive market, therefore our product should have uniqueness and needs to be constantly upgraded to keep in pace with advanced technologies consistently.

**Resource Risks**

* Hardware and software tools
* Specification implementation
* Team experience

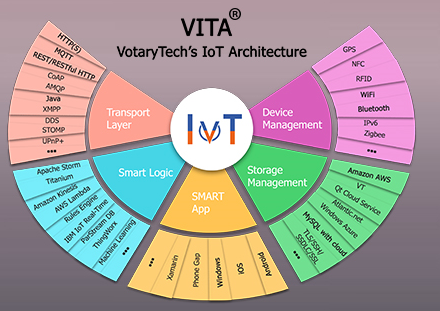
## Glossary

| Term | Definition |
| --- | --- |
| VITA | Votarytech IOT Architecture |
| BLE | Blue-tooth Low energy devices (Ex: Beacons) |
| IP | Internet protocol |
| DM | Device Management |
| OCF | Open Connectivity Forum |
| IoTivity | Open source project implemented based on OCF specification |
| QT | Tool to write/Debug the code |
| GCC | GNU Cross Compiler [Compiler to build C & Cpp files] |
| SL | Smart Logic |
| SA | Smart App |
| TL | Transport Layer |
| SM | Storage Management |

# Specify

At very high level VITA architecture is Votarytech’s “Integrated solution for M2M and IOT solution”.

It consists of the following modules



* Device management
* Storage Management – out of scope of this document
* Smart App – out of scope of this document
* Smart Logic – out of scope of this document
* Transport Layer – out of scope

**DM Specification:**

1. Develop a “Device Management” platform, which has communication and inter-operability framework for multiple market segments, , platforms, modes of communication, transports and use cases
2. Develop common protocols for discovery and connectivity
3. Opportunity for innovation and product differentiation (other framework vendors)
4. Multiple “Operating systems” support such as Linux, Android [Out of scope for this Phase]

## Scope

DM will provide common interface across various communication protocols and does management/control/monitor of the resources.

APIs are categorized in different section

* Discovery and control API
* Data storage ( with SM )
* Data Exchange (with cloud)

## Schedule and Milestones

**Schedules**

1. Schedule1 Third party lib identification and integration
2. Schedule2 Discovery, Connect, Engage –BLE devices
3. Schedule3 Discovery, Connect, Engage –Third party stack BLE/IP devices

**Milestones**

Porting Third party Stack into DM  12th May 2017

Desing & Implementation of DM  18th May 2017

DM interoperability to Iotivity and non Iotivity devices  21st May 2017

Resources

**High level resources**

* Project Manager /Technical lead
* 5 Developers
* 5 Testers

**Skill set**

* C, Linux, CPP
* Python

## Cost

Resource 5 Engineers + 1 Manager

Project duration: 6 Weeks

Project cost Rs 6,37,147/-

## Market Analysis

**Competitors:**

VITA is IOT architecture of Votarytech. This will be used for internal and external products architecture.

It is in-house product, which will help faster Go To Market. As such this is one-size fits all architecture.

**Years in business // Hardware partners//Hardware cost/Customer Base:**

TBD

**Substitutes in Market for Product**

Reference: [https://github.com/HQarroum/**awesome-iot**](https://github.com/HQarroum/awesome-iot)

* Eclipse IoT Projects including Kura, Mihini, Ponte, SCADA, Smarthome,
* AllJoyn by AllSeen Alliance.
* IoTivity by OIC.

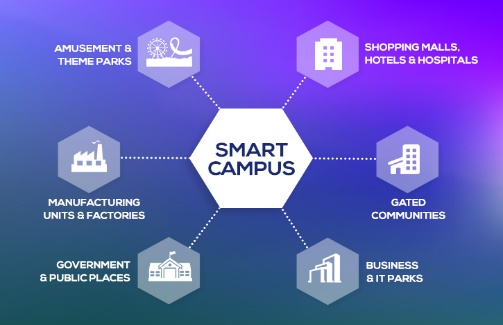
**Targeted Domains**

#### Domains

* Smart Transport: This is for end2end Smartt IOT connected transportation. This will be extension and movement from existing “Reach Safe” applications



* Smart Campus : IOT based sensors, apps and hardware agnostic frameworks accelerate the smart campuses and improve customer experience



* Track and Trace : Using IOT based solution, enterprises will be able to track and trace the mission critical assets in real-time.



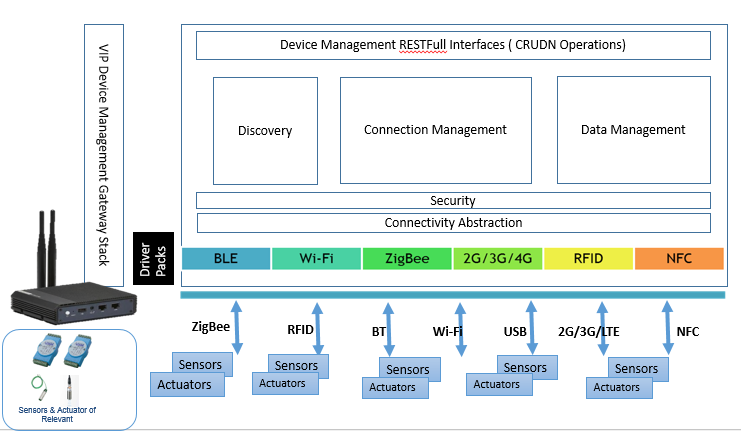
* Smart Applications: This applications use “data-driven features “, that deliver actionable insights to users ( Smart Watch)

#### How this aligns with Votarytech Business Strategy?

##### 

# Architecture

## Block Diagram



1. User Application communicates to “sensors/actuators” via DM gateway.

## Internal Interfaces

Will be defined SRS and other documents

## External Interfaces

Will be defined SRS and other documents

|  |  |
| --- | --- |
| Organization | Liaison/ Interface |
|  |  |

# Design

## Flowcharts

Will be defined SRS and other documents

## Development Environment

Mentioned in “section 1.6. Prerequisites”

## IP Identified

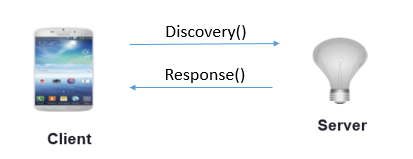
# Implement

## System Requirements

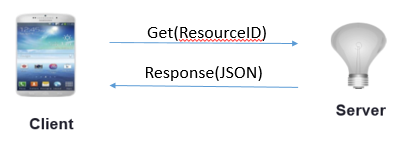
|  |  |  |  |
| --- | --- | --- | --- |
| SL # | Requirements | Description | Identification Method |
| 1 | REQ – 001 | DM shall be transport independent | DM should abstract Transport Technology  (How IoT device connected ) to calling Application |
| 2 | REQ – 002 | DM shall be OS independent | 1. DM should run on Linux 2. DM should run on Android (Phase 2) |
| 3 | REQ – 003 | DM shall be hardware agnostic | 1. *DM should run on RaspberryPI (ARM Arch)* 2. *DM should run on Desktop (x86 )* 3. *DM should run on Android Mobile* |
| 4 | REQ – 004 | Supporting simultaneous access from multiple clients not allowed |  |
| 5 | REQ – 005 | Communication between Application to DM should be secure |  |
| 6 | REQ – 006 | Communication between DM and IoT devices should be secure |  |
| 7 | REQ – 007 | DM shall provide RestFull APIs | Discover, Get, Put, Observe, Delete |
| 8 | REQ – 008 | DM shall communicate/engage BLE-IoT devices |  |
| 9 | REQ – 009 | DM shall communicate/engage **IoTivity** BLE-IoT devices |  |
| 10 | REQ – 010 | DM shall communicate/engage IP (Ethernet & WiFi) **IoTivity** Devices |  |
| 11 | REQ – 011 | DM shall take all configuration parameters from application | 1. *VITA Filter information* 2. *SM Server IP* |
| 12 | REQ – 012 | DM shall be able to Discover IoT devices around it | 1. *DM shall be able to Discover BLE-IoT devices within 30Feet* 2. *DM shall be able to Discover IoTivity-IP devices* 3. *DM shall be able to Discover IoTivity-BLE devices within 30Feet* |
| 13 | REQ – 013 | DM shall be able to provide interface to Read device Attributes in key-value pair |  |
| 14 | REQ – 014 | DM shall be able to provide interface to Write device Attributes in key-value pair |  |
| 15 | REQ – 015 | DM shall be able to provide interface to Observe any/few Attributes |  |
| 16 | REQ - 016 | DM shall out put its results to SM | 1. Discovery Result to be stored in SM 2. Get Result to be stored in SM 3. Observe Result to be Stored in SM |
| 17 | REQ – 017 | If Application running remotely, in this case its communicates toDM through TL-MQTT |  |
| 18 | REQ – 018 | Application and DM may be running in same System (Hardware), in this case its directly call DM APIs through TL |  |
| 19 | REQ – 019 | DM shall be able to monitor/observe (notify if any change in device attributes change) IoT device | 1. DM shall be able to monitor continuously to any IoT device 2. DM shall provide interface to cancel observe 3. DM shall monitor multiple IoT devices 4. DM shall monitor multiple attributes in IoT Device |
| 20 | REQ – 020 | DM shall be modular | Discovery alone shall be built independently |

**Sequence diagram**

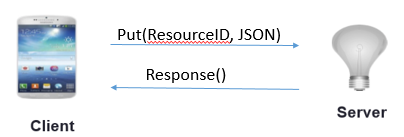
Resource Discovery



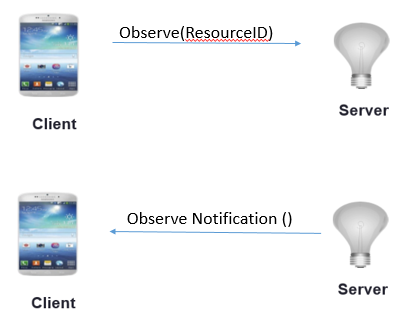
Resource Get operation



Resource Put operation



Resource Observe operation



## Revenue Realization Plan

## User Acceptance Criteria

Since DM is a technology, production Environment will vary as per the solution, when integrated with it.

## Quality Plan

Quality Assurance Plan of Votary Softech Solutions will be followed. Please refer the Quality Assurance plan of Votary Softech Solutions Pvt. Ltd.

# Validate

**References to “Review log” of the BRD**

**References to “V-Gates tracker” wrt BRD**

# Deploy

<Share BRD for signoff, references to emails sent and signoffs received and baseline the BRD>

**BRD Sign-off**

**References to emails sent**

**References to email received**

# Maintain

**Maintain versions**

Version history will be maintained in SVN