|  |
| --- |
| GE tagline.jpg  ***Global Professional Services*** |
| “Bringing to life GE Intelligent Platforms solutions, utilizing extensive experience, expertise, and capabilities, while following a disciplined and structured set of processes, helping our customers to achieve their business objectives.” |
| Functional Design |
| Low cost device integration to Cloud |
| |  |  | | --- | --- | | Customer: | GE | | Location: | Genova | | Project: | Predix hackathon | | Project #: |  | |
| [This document defines the general solution that will be used to meet or exceed the project deliverables defined in the Requirements Specification for the System to be detailed.] |

[This item and following Copyright © and Confidentiality Notices, as well as those contained within the document footer, may be removed prior to final delivery to our Customer. These items are provided for the express purpose of protecting the template.]

[All rights reserved. No part of this document may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of GE Intelligent Platforms, 2500 Austin Drive, Charlottesville, VA, 22911, USA.

**Confidentiality**

The information contained herein is confidential and proprietary to GE Intelligent Platforms. It may not be disclosed or transferred, directly or indirectly, to any third party without the explicit written permission of GE Intelligent Platforms.]**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Summary of Update |
| 0.1 | 11/04/2016 | Nicolandrea Costa | Draft |
| 1.0 | 06/05/2016 | Vamsi Kasturi | Release Version |
|  |  |  |  |

**Contributors**

|  |  |  |
| --- | --- | --- |
| Name | Organization | Title |
| Gisella Bennardo | GE Digital GPS EMEA | Hackathon Architect |
| Alberto Gorni | GE Digital GPS EMEA | Hackathon Architect |
| Nicolandrea Costa | GE Digital GPS EMEA | Hackathon Lead |
| Miroslav Pindjak | GE Digital GPS EMEA | Hackathon Developer |
| Daniel Batista | GE Digital GPS EMEA | Hackathon Developer |
| Vamsi Kasturi | GE Digital GPS EMEA | Hackathon Developer |
| Carlos da Rocha | GE Digital GPS EMEA | Hackathon Developer |

**Table of Contents**

[1 Audience 1](#_Toc450314998)

[2 Abbreviations and Acronyms 2](#_Toc450314999)

[3 Overall Description 3](#_Toc450315000)

[3.1 Conceptual Module 3](#_Toc450315001)

[3.2 Infrastructure Requirements 3](#_Toc450315002)

[4 System Architecture 4](#_Toc450315003)

[4.1 Hardware Architecture 4](#_Toc450315004)

[4.1.1 Overview 4](#_Toc450315005)

[4.1.2 Hardware Servers 4](#_Toc450315006)

[4.2 Software Architecture 6](#_Toc450315007)

[4.2.1 Software modules 7](#_Toc450315008)

# Audience

This document is intended for use by members of the Project Team, Key Stakeholders, and other members of the Global Professional Services team.

It is also intended to be available to all project staff in support of the overall objectives of the Global Professional Services team, emphasizing active participation in achieving quality delivery and execution goals.

# Table of Figures

[Figure 1: Hardware Architecture 4](#_Toc450315209)

[Figure 2: Network diagram 5](#_Toc450315210)

[Figure 3: Software Architecture 6](#_Toc450315211)

[Figure 4: Data Flow between the components 6](#_Toc450315212)

[Figure 5 : Software Component diagram 7](#_Toc450315213)

[Figure 6: Data stored in flat file in JSON Format 8](#_Toc450315214)

[Figure 7: Web Application showing the data in Timeline 8](#_Toc450315215)

# Overall Description

## Conceptual Module

The Goal is to design and develop a model where any low cost devices (< 100 USD) that support communication via http can connect to the Predix cloud. This set up can be used demo a quick understanding of software components and the data can be shown in the cloud with in few configurations.

## Infrastructure Requirements

These the components used for the purpose illustrated below:

**Hardware:**

|  |  |
| --- | --- |
| NodeMCU | It is IOT based platform to connect sensors and can communicate to devices using wireless and Ethernet port using Http communication |
| Low Cost sensor | Sensors to capture any physical property like Temperature, humidity , pressure, noise etc. |
| Raspberry PI | A low cost credit card sized computer. It is used to Receive the data from the NodeMCU, run the Predix machine and push the data to Predix Cloud |

**Software:**

* Predix Machine
  + Custom application on PM running as a web service
* Predix Cloud
* Predix Mobile Application

# System Architecture

## Hardware Architecture

### Overview

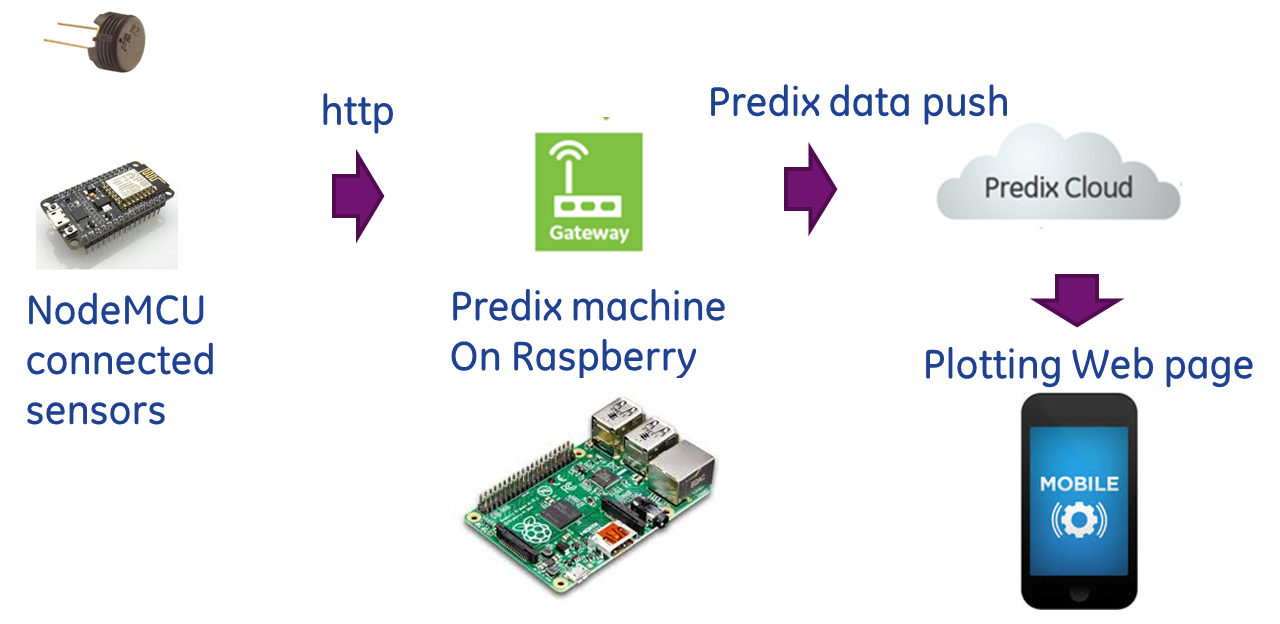


Figure : Hardware Architecture

### Hardware Servers

Rasperry PI

NodeNCU



Figure : Network diagram

## Software Architecture

The architecture includes several pieces of software to be deployed on different Predix Architecture components. The following are modules and the data interaction between them.

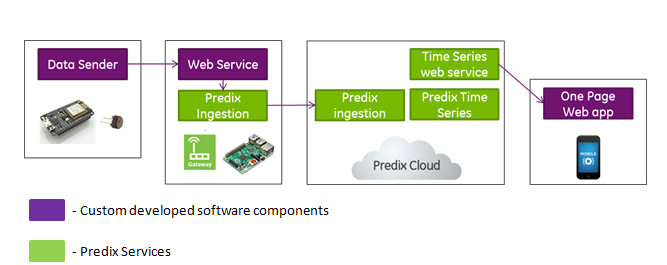


Figure : Software Architecture



Figure : Data Flow between the components

### Software modules

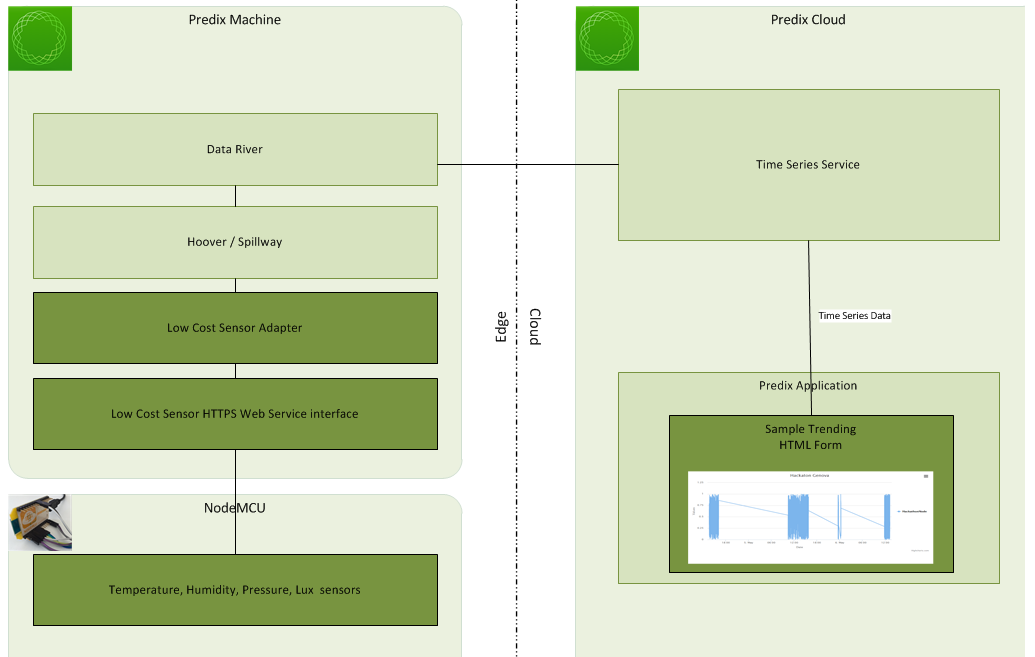
The following is the component diagram showing the interaction between the components of the system: 

Figure : Software Component diagram

#### Data sender

Data Sender



Module that collect data from sensor with a configurable period and send immediately to Predix machine REST web service (http POST call) by calling the POST Method on the web service

#### Web Service

Web Service

The web service developed in Java running on the predix machine has two methods:

* POST: This method is called by the NodeMCU to send Data to the web service. The received from the sensor is stored in a flat text file with in the predix machine (raspberry Pi).
* GET: This method is called by the Http Adapter to receive the data stored in the flat text file in JSON format. Once the data is received the data in the flat text file is cleared.

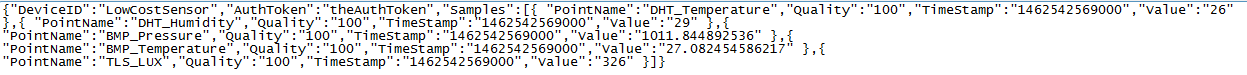


Figure : Data stored in flat file in JSON Format

#### Http Adapter:

The Http Adapter running on the predix machine calls the GET method of the web service and “OnPushData” call, the data is forwarded to the Predix data ingestion (Spillway) .

http adapter

#### Hoover Spillway:

The Data River service allows data to be passed through the Spillways for processing. Thus the Spillway will forward the collected data to the data river.

**Spillway**

The Predix data ingestion service should be configured to enable the store and forward functionality.

#### Data River

The Http Data River collects the data arriving from the Spillway and by HTTP sender service the data is pushed to the cloud

**Data River**

#### Predix Time Series

Data should be stored by the Time series service. That provides also a REST API for queries.

**Data River**

#### Web Application

This simple web application trends the data in the Timeline. The users can zoom in to see the data at a particular time frame. The application can be accessed from any PC, tablet or mobiles and require no resizing.

Web Application

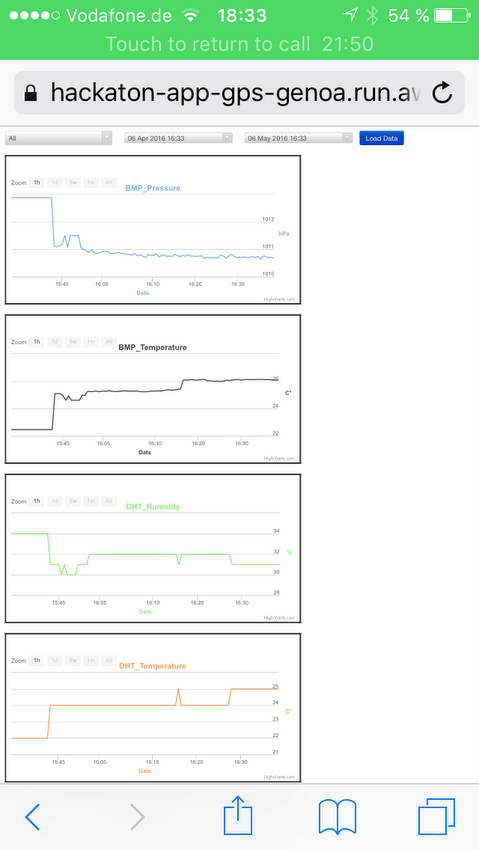


Figure : Web Application showing the data in Timeline

**Template Revision History**

| Version | MIRF | Date | Author | Summary of Update |
| --- | --- | --- | --- | --- |
| 0.1 |  |  |  | Baseline |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Contributors**

|  |  |  |
| --- | --- | --- |
| Name | Organization | Title |
| Nicolandrea Costa | Global Professional Services – EMEA | Technical Leader |
|  |  |  |
|  |  |  |