**High Level Solution Design and Project Estimate**



TSI Incorporated

Multi-Pollutant Monitor System, Phase 1 Planning

December 2013

**Table of Contents**

[1 Document Overview 4](#_Toc375123610)

[1.1 Purpose 4](#_Toc375123611)

[1.2 Project Goals 4](#_Toc375123612)

[1.3 Document References 4](#_Toc375123613)

[1.4 Data Dictionary 5](#_Toc375123614)

[2 Participants 6](#_Toc375123615)

[2.1 Workshops 6](#_Toc375123616)

[2.2 Digi/Etherios 6](#_Toc375123617)

[2.3 TSI 6](#_Toc375123618)

[3 Actor Analysis 7](#_Toc375123619)

[4 High Level Solution Architecture Options 8](#_Toc375123620)

[4.1 Local Independent Gateway Option 8](#_Toc375123621)

[4.2 Local Mesh Gateway Option 10](#_Toc375123622)

[4.3 Local Appliance Option 11](#_Toc375123623)

[4.4 Conclusion 12](#_Toc375123624)

[5 Scope and System Specifications 13](#_Toc375123625)

[5.1 Minimal Baseline Solution 14](#_Toc375123626)

[5.2 Simple Mesh Add-ons 21](#_Toc375123627)

[5.3 Advanced Mesh Add-ons 22](#_Toc375123628)

[5.4 Advanced Cloud Add-ons 23](#_Toc375123629)

[5.5 Validation, Training, and Deployment Support 25](#_Toc375123630)

[5.6 Deliverables 25](#_Toc375123631)

[5.7 Work Out of Scope 26](#_Toc375123632)

[6 Approach 27](#_Toc375123633)

[6.1 Implementation 27](#_Toc375123634)

[6.2 Assumptions 27](#_Toc375123635)

[7 Implementation Breakdown 29](#_Toc375123636)

[7.1 Resources 29](#_Toc375123637)

[7.2 Design and Development Tasks 29](#_Toc375123638)

[7.3 Potential Sprints 32](#_Toc375123639)

[7.4 Implementation High-Level Schedule and Cost Estimates 34](#_Toc375123640)

[7.5 Hardware 36](#_Toc375123641)

[7.6 Hosting 36](#_Toc375123642)

[8 Appendix 1: Actor/Use Case Analysis 37](#_Toc375123643)

[1. TSI Administrator Tasks (cloud) 37](#_Toc375123644)

[2. Customer Administrator Tasks (cloud & local) 37](#_Toc375123645)

[3. Site Installation and Management Tasks (cloud & local) 38](#_Toc375123646)

[4. TSI Support Tasks (cloud) 39](#_Toc375123647)

[5. Customer Operator Tasks (cloud & local) 39](#_Toc375123648)

[6. Internal Cloud Processing (non-UI) 39](#_Toc375123649)

[7. Internal Local Gateway App Processing 40](#_Toc375123650)

[8. Other 43](#_Toc375123651)

# Document Overview

## Purpose

The purpose of the high-level solution design deliverable is to clearly document the to-be solution based on requirements provided by TSI and requirement gathering sessions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/17/13 | 1 | Initial draft | Jeff Ryan, Lynn Linse, Leann Landry |
| 12/18/13 | 2 | Team review updates | Leann Landry |
|  |  |  |  |

## Project Goals

TSI is a precision measurement and instrumentation company with solutions spanning many different market segments and industries. TSI, their integrator partners and customers have a responsibility to measure, monitor and report on air quality on job sites where dust, debris, and particulates exist. TSI is looking to building a remote monitoring solution for their DustTrak product as a way of satisfying a market need, and staying in compliance with Local, State and Federal regulations.

This document describes the high-level design approach and alternatives to implement the MPM/DustTrak monitoring system where TSI and their customers may visualize field data, receive alerts, and perform offline data analysis. Key elements of the MPM system include:

* Intelligent Gateway Device Connectivity
* Intelligent Gateway Wifi App
* Intelligent Gateway Alerting
* Data Storage
* Cloud App
* Cloud Alerting

The key parts of this document include:

* Actor/Use Case Analysis
* High Level Solution Architecture
* System Specifications
* Implementation Breakdown – Schedule, Resources, Tasks, and Costs

## Document References

This document makes references to the following related project documents:

|  |  |
| --- | --- |
| **Document** | **Filename & Version** |
| MPM project requirements provided by TSI | Customer Requirements – MPM. doc; version 0.0 |

## Data Dictionary

The following terms and abbreviations are described here and may be used as a quick reference.

**ATEX:** European Union (EU) Directives commonly called “ATEX” cover electrical and non-electrical equipment that is used in potentially explosive atmospheres.The Directives are known as ATEX (from the French – ATmospheres EXplosibles)

**CE**: The CE marking is a mandatory European marking for certain product groups to indicate conformity with the essential health and safety requirements set out in European Directives. CE has no meaning as an abbreviation.

**CSA**: Canadian Standards Association. CSA marks mean a product has been tested and meets applicable standards for safety and/or performance, including the applicable standards written or administered by the American National Standards Institute (ANSI), Underwriters Laboratories (UL), Canadian Standards Association (CSA), National Sanitation Foundation (NSF), and others

**DustTrak**: Current DustTrak product offering that will be included in phase 1 of the MPM project

**Environmental DustTrak**: Future DustTrak product offering that will be addressed in phase 2 of the MPM project

**UL**: The UL Mark stands for Underwriters Laboratories Inc. (UL). The UL Mark on a product means that UL has tested and evaluated representative samples of that product and determined that they meet UL's requirements. Under a variety of our programs, products are periodically checked by UL at the manufacturing facility to make sure they continue to meet UL requirements. There are several types of UL Marks. Each has its own specific meaning and significance.

# Participants

This section identifies participants and contributors to the MPM phase 0 project.

## Workshops

|  |  |  |
| --- | --- | --- |
| **Session** | **Date** | **Group** |
| Kickoff and requirements review | 11/20/13 | Digi/Etherios and TSI |
| Requirement question feedback (via email) | 12/4/13 | Digi/Etherios and TSI |
| Requirement gathering | 12/11/13 | Digi/Etherios and TSI |

## Digi/Etherios

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Contact (Phone/Email)** |
| Jeff Ryan | Solution Architect | (952) 912-3208, jeff.ryan@digi.com |
| Lynn Linse | Technical Architect | (952) 912-3029 , lynn.linse@etherios.com |
| Leann Landry | Project/Engagement Manager | (214) 336-5669, llandry@etherios.com |

## TSI

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Contact (Phone/Email)** |
| John Wood | Executive Sponsor | (651) 490-2767, [john.wood@tsi.com](mailto:john.wood@tsi.com) |
| Jerry Bark | Executive Sponsor | (651) 490-2835, jerry.bark@tsi.com |
| Frank Pernat | Program Manager | (651) 490-4054, frank.pernat@tsi.com |
| Anthony (Tony) Hase | Technical Manager | (651) 490-4044, tony.hase@tsi.com |
| Brian Osmondson | Product Management | (651) 490-2842, brian.osmondson@tsi.com |
| Shawn Knepper | Technical Lead | (651) 490-4025, shawn.knepper@tsi.com |

# Actor Analysis

This section of the document explores who the users of the system are (actors) and what their primary tasks will be. By looking at the workflow from each user’s perspective, we aim to provide a fluid solution without missing important requirements. It helps crystallize feature and security requirements as it explains who needs the features and why.

The following table describes the direct and indirect actors of the TSI solution:

|  |  |
| --- | --- |
| **Actor** | **Description** |
| TSI Administrator | Person from TSI responsible for establishing/managing customer relationships. This could include Sales, order fulfillment, and billing. |
| Integrator | TSI channel partner that helps install, configure, and service TSI customers |
| Customer Administrator | Customer contact responsible for managing their sites and the access to them |
| Site Installer | Person from TSI or an approved integrator responsible for installing a new customer site |
| TSI Support | Person from TSI or an approved integrator handling customer support issues |
| Customer Operator | Daily user of the TSI equipment that monitors and responds to alarms and takes corrective action |
| *Internal* | Internal tasks expected to be performed by the system |

Appendix 1 describes the tasks involved in each role defined above.

# High Level Solution Architecture Options

This section describes the major components of the system and how we envision them to work together.

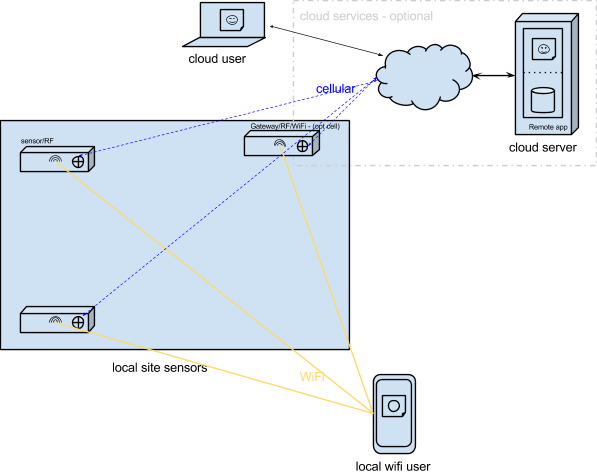
* Multiple site networks of gateways/devices
* Local wifi network on each site
* Wifi html app to manage that site
* Gateway applications
  + Connect, control, and aggregate data from TSI devices
  + Offer wifi html application to deploy, configure, and operate local site
  + Offer optional uplink and control channel to cloud application
* Optional cloud application
  + Deployment and maintenance considerations

We’ve identified three potential architectures that may suit TSI needs - each with their own pro’s and con’s.

## Local Independent Gateway Option

This design considers a solution where each gateway operates independently of each other. In this situation the gateways do not talk with one another and act as independent WiFi HotSpots. Each gateway offers up a very simple web application that allows configuration of the sensor and viewing of its current data and alarms status. The gateway can optionally be configured to connect via cellular to the cloud where a more advanced remote web app can provide aggregated sensor and alarm management.

This scenario is simple and may be suitable in situations where there are only a small number of sensors on site. It may be cumbersome when many sensors are involved since the wifi app only connects to one sensor at a time and you must be within wifi range of the sensor.

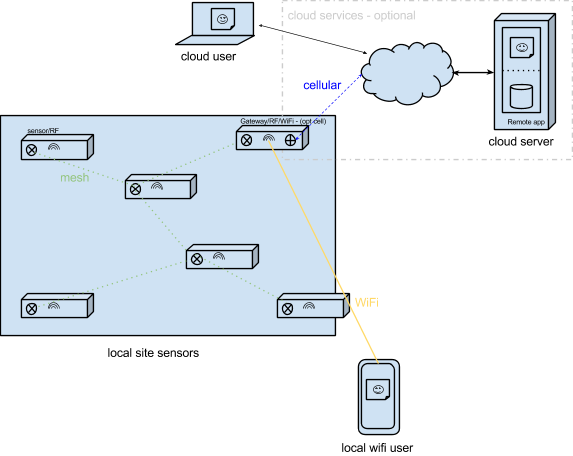


|  |  |
| --- | --- |
| **Pro’s** | **Con’s** |
| * Simple for single sensor site. WiFi user simply joins gateway and configures it * local WiFi interface provides access to gateway and attached sensor. User can check data and alert status as well as ack/reset alarms. * Optional Device Cloud reporting on whichever sensors require it | * Required site sizes prohibit WiFi so each gateway has unique WiFi network * Gateways cannot share information with one another * local WiFi phone must join to each gateway individually to setup and configure device * local WiFi phone must join to each gateway individually to check data and alert status as well as ack/reset alarms. * Cellular connection for each gateway means extra $$ * Cloud app and wifi app will be two different apps * Gateway WiFi web app capabilities are limited compared to cloud server app. |

## Local Mesh Gateway Option

This design considers a solution where gateways are connected together in a mesh. As gateways report sensor and alarm data they send that data through the mesh. One of the gateways in the mesh can be configured to connect via cellular to the cloud and can thus relay that data to a more advanced remote web app can provide aggregated sensor and alarm management. Each gateway still operates as a WiFi hotspot that serves up a simple configuration and management web app to local WiFi users.

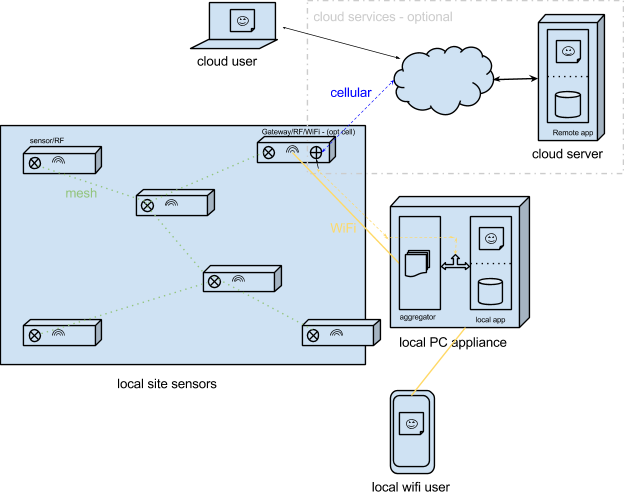
In this scenario, it may be possible to enhance the WiFi app to provide limited access to remote gateways through the mesh. Things like last sensor readings, current alarm conditions, etc. In general however this app will likely not have all the features of the cloud web app due to limited processing and storage capacity of the gateway.



|  |  |
| --- | --- |
| **Pro’s** | **Con’s** |
| * Ideal for sites with two or more sensors * Mesh means gateways can communicate with one another and share single uplink * Enhanced usability because local WiFi user can connect to any node and configure or view its detailed information as well as summary information on other remote nodes. * Local app could be further enhanced to offer full configuration and alert management of remote gateway sensors. * Optional Device Cloud uplink provides remote cloud monitoring and support | * Mesh network is additional cost for xbee sensors and drivers to support them - but this is offset by the reduced cellular uplink cost due to mesh sharing * Local wifi application is more complex resulting in higher development cost * Cloud app and wifi app will be two different apps * Gateway WiFi web app capabilities are limited compared to cloud server app. |

## Local Appliance Option

This design considers the insertion of a pc appliance that serves to concentrate communication with the local site sensors and supports a web based management app that is available to local WiFi users. The local app would support many of the same features and capabilities of the cloud app but is limited to interacting with sensors at the local site. If the sensor data and alarm processing needs of the site are great, some of this may be offloaded from the gateway into the pc appliance (see aggregator component).



|  |  |
| --- | --- |
| **Pro’s** | **Con’s** |
| * (Same pro’s as option #2 above) * More advanced web app capabilities due to processing power of local pc appliance * Opportunity to share much of web application between cloud and local | * Mesh network is additional cost for xbee sensors and drivers to support them - but this is offset by the reduced cellular uplink cost due to mesh sharing * Local wifi application is more complex resulting in higher development cost * Added expense of local pc appliance |

## Conclusion

On 12/11/2013 we presented the above options to TSI and discussed the pro’s and con’s of each option. We discussed that the local independent gateway option would be the simplest first step but that the mesh gateway option has many attractive attributes and would be a good next step. We feel that this could be a logical stepping stone approach since it is unlikely we can go directly to the full mesh solution in time for their first phase rollout.

The specification section below details a baseline solution that centers around the independent gateway option #1 with optional add-on’s that bring us to the mesh gateway solution #2. These solutions can be implemented in multiple phases to accommodate schedules and should be interoperable and either can be employed by TSI as they see fit.

It was felt that the local appliance option #3 was not necessary.

# Scope and System Specifications

This section takes a deeper dive into what will be delivered based on our improved understanding of the requirements and further exploration of architecture options. Ideally we would have very basic wireframes of the expected UI’s to provide TSI with an understanding of how the different roles would interact with the system to accomplish key tasks. The UI’s are certainly be subject to change however and would be refined in the early phases of the development project.

In this section we outline in more detail:

* What the major components of the system are
* What function will be delivered by each component
* An identification of the major work items for each component
* An estimate of the development effort involved to build the solution
* An estimate of the development effort for major features (if possible)
* Deployment and maintenance considerations
* Hosting & support costs

Considering both the Baseline solution and the more advanced mesh solution there are several major system components to consider. They are:

* Security Model
* Gateway
  + WiFi web UI
  + Sensor communication drivers
  + Application processing
* Cloud
  + UI
  + Cloud communications
  + Application processing

The security model considers the various actors and use cases examined above and defines a mechanism to control access to both system resources (i.e. customer data) as well as system function (i.e. alarm reset).

The Gateway and Cloud components define applications available both locally and remotely. They each contain user interface work as well as communication and internal processing.

Each of these will be described further in the sections below. First the minimal baseline solution option and then additional add-on components that can be pulled into the initial or follow-on phases depending on resource and schedule constraints.

## Minimal Baseline Solution

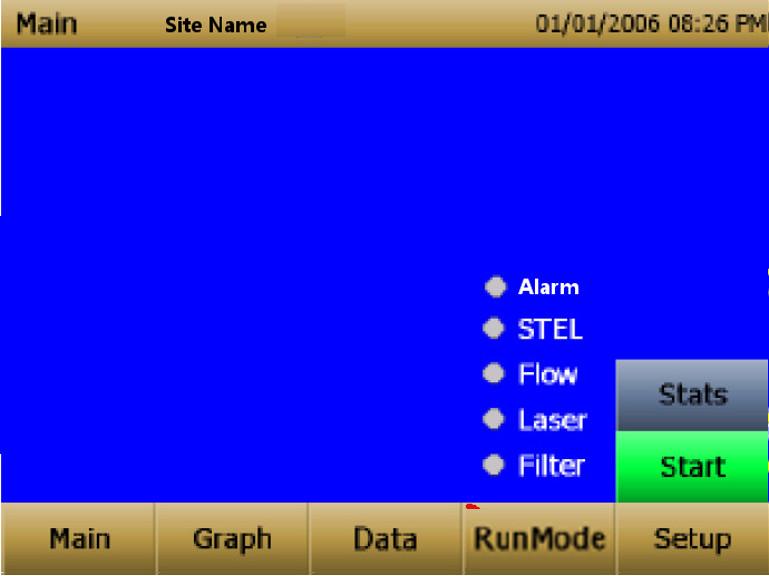
This solution is the simplest scenario where each gateway and cloud apps supports only critical features. Gateways operate independently and each gateway has optional cellular uplink to the cloud.

### Security Model

The following security model will be used to control access to the information and the services offered up by the cloud application. Later parts of this specification will refer to the classes and roles required to access particular features of the application. Some classes and roles may not be used in the baseline solution but are leveraged in later advanced options. Still, we define them here to explain the model. [D2]

1. **Primary User Classes** - The user class defines the kind of user logged into the system and in general what system resources they can access.
   1. *TSI Admin*: TSI personnel who manage customer relationships, integrator partner relationships, and technical support for the entire system. They have access rights to all customers and integrator partners on the system.
      1. Roles: *Customer-Onboarding, User-Management, Partner-Onboarding, Site-Management, [Customer-Cloud-Access], [Customer-WiFi-Access]*
   2. *Integrator*: TSI channel partners who typically assist designated customers with site setup and support. These users can access the accounts of any customers configured in the their access control list.
      1. Roles: *Site-Management, [Customer-Cloud-Access], [Customer-WiFi-Access]*
   3. *Customer*: Uses cloud services for remote access to and management of the sensors in their site. Customers can only access resources tied to their own account.
      1. Roles: [*User-Management], [Customer-Cloud-Access], [Customer-WiFi-Access]*
2. **Roles** - Roles define what operations a given user can take against the resources they have access too. Based on their class, users will have one or more roles assigned to them. The following roles control access to various features in the system:
   1. *Customer-Onboarding*: A user with this role can create, edit, or delete customers in the system. The existing customers they can view are restricted by their access rights.
   2. *User-Management*: A user with this role can create, edit, or delete other users within the customer accounts that they are authorized to.
      1. The initial user created for a customer will have this role. They can create additional users within that customer account and can grant or revoke this role as desired.
   3. *Integrator-Onboarding*: A user with this role can create, edit, or delete integrators in the system.
   4. *Site-Management*: A user with this role can create, edit, or delete sites for any customer that they have access rights to. In addition, this user can add, configure, and remove equipment from a customer’s account.
      1. TSI has decided that Customer users will not have this role and must use TSI or a designated integrator partner to manage their sites
   5. *Customer-Cloud-Access*: a user with this role has both roles described below.
      1. For the baseline solution, Customer users will all have this role
      2. *Customer-Data-Access*: a user with this role can examine any customer data they have access rights to.
         1. Customer users are optionally granted this role when they are created. This role can be removed from an existing user.
         2. TSI Admin or designated Integrator users can be granted this role temporarily by customer during troubleshooting activity
      3. *Customer-Alert-Management*: a user with this role can receive alarm and alert notification as well as acknowledge and reset alarms.
         1. Customer users are optionally granted this role when they are created. This role can be removed from an existing user.
         2. TSI Admin or designated Integrator users can be granted this role temporarily by customer during troubleshooting activity
   6. *Customer-WiFi-Access*: a user with this role can log in and use the WiFi application on the customers site
      1. This role is not used in the baseline solution.

### Gateway Components/Tasks

1. **Smart Phone UI**
   1. Summary:
      1. Estimated 20 web pages, most with buttons or data entry.
      2. Use of a graphing plugin
      3. 4 mini-database managers
         1. user/password
         2. log configs
         3. file manager
         4. multi-language phrases
      4. Dynamic adjustment of screen elements for multi-language
   2. User uses normal smart-phone settings to locate SSID of desired box, entering the required WiFi keys, and connects. Standard IG Web UI shall be used to configure the WiFi Access Point informat (name, keys, etc). SSID and Wifi Keys will be in ANSI form always.
   3. HTML5 page support of username/password for local access defaults to off (assuming SSID with WiFi key blocks unauthorized access). Up to 10 username/password combinations can be entered on the Setup Screen, including a setting for view-only, or full access. Locally entered names are not synchronized; cloud service is required to synchronize user names on all gateways. Locally entered names have no cloud access rights. If cloud access is used, entering local names/passwords can be disabled from the cloud, giving cloud full control over security. Usernames and passwords will be in ANSI form always.
   4. TSI splash screen shall be shown for a fixed number of seconds.
   5. MAIN SCREEN is like DT II, with some details eliminated or adjusted as appropriate for the newer design. Also, layout may change slightly to accommodate screen geometry of mobile devices. A site name is displayed in top bar, and the date/time is only shown on local WiFi access showing estimated DT II time setting. Tony H asked to make the normal alarm and STEL display as alarms like flow/laser/filter.
   6. Pressing START will either show the values, or a time delay to log/test start (per Tony H, the new design will always run in one of the 5 configured log/test modes). We propose eliminating the STATS button and always showing the min/max/avg/TWA since by definition this is running in a hand-held device.
   7. GRAPH SCREEN will be much like the existing DT II, with any limitations imposed by HTML5 in terms of zooming or refresh rate.
   8. DATA SCREEN will be much like existing DT II. The IG is limited to 12 character file names, so for the same style (like log\_001), the base name is limited to 8 chars. All files name (even in Chinese/Asian language) will use standard ANSI chars. Total flash space used can be limited to a max with old files auto deleted by age - will default to 4MB. How files are transferred to the smart-device is to be determined, but will be limited to normal HTML5 transfers.
   9. RUNMODE allows selecting 1 of 5 log modes (Tony H said survey/manual modes did not need to be supported.)
   10. SETUP will be much like DT II, with following exceptions:
       1. ZERO CAL, as expected.
       2. no FLOW CAL
       3. USER CAL, as expected, data stored/retrieved from DT
       4. ALARM, as appropriate (no audible/visible alarm). Relay output to be the 28vdc 50mA FET output on the WR41.
       5. no ANALOG settings
       6. SETTINGS
          1. Date/Time can be set manually, or use SNTP via Internet
          2. IP defaults to DHCP (assigned by IG) Efforts will be taken to always assign the same IP to the DT
          3. Site Name
          4. User Name / Password management
   11. Note that IG wireless (WiFi and cellular) settings will be done by the IG web interface, as quite a bit of diagnostic information may be involved and vary by carrier.
   12. Basic multi-language (ANSI char set) will be by web page, allowing regional distributors to adjust and add local support. Asian/Chinese input will be an option. There will be no support for non-ANSI (non-arabic) numbering systems. Limitations in underlying hardware/firmware means not all text can be converted to multi-language.
2. **IG Python programming**
   1. Device driver speaking to one DK via TCP/IP, including modest effort to locate the DT by DHCP-assigned IP address.
      1. Support for protocol as published in document DustTrak\_DRX\_II\_Communication\_Manual- 6002481G-web.pdf
   2. Polling (non-STEL behavior)
      1. When one of the programmed log modes is active, poll the DT every second, calculating the min/max/avg/TWA stats over the configured time interval. Only the periodic stats are uploaded or logged.
      2. The IG will log the data to support the expected Data screen, including enough summary information to mimic existing DT II behavior when browsing. Only ANSI characters will be used in the log file.
      3. When the cloud is active, the periodic data shall be coalesced and uploaded at a configured period.
      4. Each second, the data shall be compared to the 2 alarm values.
      5. As required, enable auto-zero per config
   3. Alarms
      1. Alarm 2 (warning alarm)
         1. Can be enabled or disabled, has a fixed 5% hysteresis, and auto-resets
         2. No acknowledgement is required
      2. Alarm 1 (STEL)
         1. Can be enabled or disabled, and has no hysteresis
         2. Acknowledgement is required (and logged as ANSI text)
         3. Driving external relay signal can be enabled, disabled, inverted
         4. Causes STEL log file
         5. Functions as described - a 15-minute period, and if at the end of the 15 minutes the value is still above STEL setting, another 15-minute period is started.
      3. STEL log file shall record 1 minute data for 15 minutes, using a file name related to the running log mode. Data shall be CSV with ANSI character set.
      4. Alarm and STEL status shall be uploaded (attached) to the periodic sample uploaded.
      5. When cloud is enabled, alarms shall trigger an upload as fast as 5 seconds, with a demonstrable latency of 15 seconds to hosted app. (assuming a good cellular connection and active cloud socket. Of course, there can be no promise of an ability to upload by cellular at any specific time.)
      6. If enabled, the IG’s digital output shall be asserted (or inverted) during the STEL event.
   4. Web back-end
      1. fetching data from DT
      2. Auto-zero
      3. user calibration

### Cloud Components/Tasks [MB-CC]

1. **UI**
   1. Login page:
      1. Authenticate user, identify roles, and initiate web session accordingly [D2]
   2. Customer Management Page (*Customer-Onboarding* role)
      1. List of existing customers the logged in user has access to.
      2. Create new or edit existing customer
         1. Define customer info: customer name, address, contact, access level [D2]
         2. Designate integrators that can work with the customer
         3. Link to User Management to create users for this customer
         4. Link to Site Management to create sites for this customer
         5. Customer can be deactivated which will disable all device management and data reporting features. Historical data is still accessible
      3. Delete existing customer
         1. Deleting customer will delete all customer sites and associated data
   3. User Management Page (*User-Management* role)
      1. List of users for the currently scoped customer
      2. Create new or edit existing user
         1. Define user info: name, phone #, email, username, & password, ?
         2. The only role allowed will be cloud-app. WiFi users need to be defined when gateway is installed - see Gateway component for details.
         3. This user will have access to all the customers sites - in future this could be limited
      3. Delete existing user
   4. Site Management Page (*Site-Management* role)
      1. List of sites for the currently scoped customer
      2. Create new or edit existing site
         1. Define site info: location, description, and other pertinent info
         2. Manage list of gateways and sensors for the site
            1. Add new or edit existing gateway/sensor

Define gateway info: gatewayId, friendly name, description, location lat/long, ?

Note, the gateway and sensor may have been in use previously in another site and/or customer. The system must be able to handle this case and separate the data appropriately (i.e. data must be bound to cust/site not just gateway or sensor id).

**Note:** In baseline solution, gateway and sensor configuration are done in the WiFi app. This includes alarms, alert, and logging config

* + - * 1. Delete existing gateway/sensor from site

Removing a gateway will not delete the historical data that was reported by sensors attached to that gateway. This data remains with the site until the site is deleted.

* + 1. Delete existing site
  1. Dashboard Summary view. Tabular list of all sensors at a site (sorted by alarm state summary) - (*Customer-Alert-Management* or *Customer-Data-Access* role) [H.1]
     1. Site selector dropdown - customer can pick from the list of sites they have configured
     2. Table columns include:
        1. SensorId
        2. User assigned friendly name [H.12]
        3. Alarm state summary [D.10]
        4. Sensor operational status summary [D.7]
        5. Gateway operational status summary [D.8]
        6. Last reported sensor data statistics [D.4, D.5]
     3. Selecting a sensor from the list, the user can drill in to see details of that sensor
        1. Alarm state details (*Customer-Alert-Management* role)
           1. Option to display STEL data related to alarm
           2. Acknowledge/reset alarm
        2. Sensor details view that shows additional sensor information (model #, serial #, calibration date, firmware version) [D.17]
        3. Historical view of the following of a user specified time window [D.6,D.11,H.1]
           1. sensor data statistics
           2. operational status levels
           3. alarm/alerts
  2. Time Chart view [H.3]
     1. Site selector dropdown - customer can pick from the list of sites they have configured
     2. Plot historical data of up to two sensors from any site on a single chart
     3. User configures sensors, time window, and linear log axis
  3. Export data to CSV or equivalent [H.4]
     1. Site selector dropdown - customer can pick from the list of sites they have configured
     2. Sensor reading data. User specify time window & sensors of interest
  4. System status events [H.5]
     1. Site selector dropdown - customer can pick from the list of sites they have configured
     2. data transfer failures
     3. gateway/device logs
     4. alarm states
  5. Language Support
     1. Provide initial English
     2. Provide capability for TSI to define/install additional language packs
        1. Chinese, German

1. **Database/ORM work**
   1. Design database schema
      1. Multi-tenant: partition data by customerid
      2. Security: include necessary ownership/access right information to enable security model
      3. Major data elements: Customer, User, Site, Gateway, Sensor, Sensor channels, Sensor data/history, Alarm config, Alarm status/history, Alert status/history, operational status
   2. DAO/ORM work
      1. software layer to control access between database and upper layers
      2. maps data from db into native objects
      3. enforces security model
2. **Automated/Internal Processing**
   1. Time
      1. Data, Alarm, and Alert reports all in UTC
      2. Application time kept in UTC - application server synced using ntp to DeviceCloud
      3. Reports and UI can be rendered in UTC or users locale
   2. Gateway initial connect processing
      1. When gateway connects to cloud for the first time we will interrogate it and record some pertinent gateway information. Things like: location lat/long, firmware level, mesh addr, name, etc
      2. Record gateway information in database
   3. Sensor discovery processing
      1. When the gateway is connected to a new sensor the installer may need to configure the gateway to communicate with that sensor. Once the gateway/sensor configuration is complete, the gateway will generate a new sensor report and send it to the cloud. This report will include sensor information such as model #, serial #, calibration date, firmware version.
      2. Receive sensor discovery reports from gateway via HTTP push
      3. Record new sensor information in database
   4. Sensor data report processing
      1. Receive sensor data reports from gateway via HTTP push
      2. Sensor data channels will be discovered as gateway reports data. Data reports will include: timestamp, channel, interval, min, max, avg, twa
      3. Record sensor information and/or data in database
   5. Operational status report processing
      1. Receive periodic sensor and gateway operational status information from gateway via HTTP push
      2. Receive related error logs from sensor and/or gateway via HTTP push
      3. Record operational status info in database
      4. Lookup status notification rules and send notifications to configured users
   6. Alarm/Alert report processing
      1. Receive alerts from gateway via HTTP push
      2. Receive related STEL files from gateway via HTTP push
      3. Record alert info in database
      4. Lookup alert notification rules and send notifications to configured users
         1. Re-notify alerts until acknowledged
   7. Notification processing
      1. **Note:** in the baseline release notification will be by email only
      2. **Note:** In the baseline solution, alert notification will automatically be sent to all the customers users with the *Customer-Alert-Management* role.
3. **Device Cloud Communication**
   1. Setup initial TSI development and production accounts
   2. All DeviceCloud activity done under TSI account
   3. Group API for creating new customers
   4. DeviceCore API’s for provisioning gateways & sensors
   5. SCI for configuring gateways & service ops like reboot
   6. SCI DIA commands for data channels & dia commands
   7. HTTP Monitor API for reliable, real-time push of data and alarm activity into the Web App.
      1. DIA, DataPoint, or FileData for data, alerts, and alarm reports. FileData would probably be simplest to use for all of these.
      2. FileData API for transferring STEL files associated with alarms
      3. FileData API for transferring sensor and gateway error logs [D.7]
   8. Cellular configuration and status feedback is to be done through normal IG web ui - no special pages shall be provided.
4. **Production: client requirements**
   1. Establish domain, SSL certs
   2. Heroku deployment and initial configuration/setup
   3. Device Cloud account & subscription

## Simple Mesh Add-ons

These add-on items enable using a shared cellular uplink to reduce cellular costs. This would be basic data routing but still relatively simple UI.

1. **Basic Mesh Operations**
   1. Since the WR41 does not include internal XBee/mesh support, the mesh hardware would be connected by serial RS-232
      1. We’d require a simple API-wrapper to handle the xbee comms, plus adding multi-client support would allow transparent support of DIA data and remote SCI/RCI commands to share the mesh.
      2. Multi-client may require use of localhost sockets
   2. Use a simple discovery protocol to identify the IG in the mesh
      1. A minimum set of manual configuration will be required, such as coordinator/router (for ZB), fixed PAN or Vendor ID.
      2. IG should then be able to auto-detect other IG and handle any internal data changes
   3. RCI proxy
      1. Allow host-app or web-services to move non-DIA SCI through the mesh, using an IG python app.
      2. Modest effort (beyond multi-client) since the IG proxying the request does not need to understand.
2. **Enable Virtual Sensors**
   1. Because the DT II has limited data points, the easiest solution to a mesh of 10 or 20 IG is to enable remote virtual (or mirror) sensors to be managed.
      1. Example, a site has 7 IG, 6 with mesh, and 7th with cellular/broadband uplink, then mirror the 6 mesh-based sensors on the 7th IG, which appears to have 7 sensors
      2. This model fits fairly cleanly into common hosted-app design, as get/set commands targeted at the virtual sensors can automatically proxy across the mesh as appropriate.
      3. To support this at a faster pace, each IG could unicast (or broadcast) its fresh data periodically, so the 7th IG ‘mirroring’ the other six IG basically just retains a ‘last seen’ collection of data. Data changes can be queued for upload in a natural way.
3. **Local WiFi web pages**
   1. Support a simple, one screen display which can show the ‘last seen’ data from any local node, including alarm/STEL status. To do anything more, the user must walk to the appropriate IG and connect by WiFi.
4. **What is NOT supported**
   1. One cannot acknowledge alarms on different IG - only see alarm status
   2. One cannot configure other IG; one must use the cloud or go to each gateway one by one.
   3. File Transfer across the mesh is not supported - to update firmware or Python code, one would need to link the IG to a PC by Ethernet.

## Advanced Mesh Add-ons

These add-ons provide a richer experience for the local wifi app. We can provide options here like distributed configuration and alarm management.

1. **Add local support for multiple IG**
   1. Allow WiFi+ Smart-Device to acknowledge alarm for any IG from any other
   2. Allow WiFi+ Smart-Device to configure any IG from any other
2. **Add simple file transfer**
   1. Assuming a file exists on one IG, allow it to be shared or synchronized with any other IG at site
   2. Perhaps by XMODEM or FTP?
   3. Modifications to SCI/RCI proxy should allow IG firmware update indirectly.

## Advanced Cloud Add-ons

These add-ons provide a richer experience for the cloud app. We can provide options here like advanced user management, graphing, and reporting.

1. **UI**
   1. Integrator Management Page (*Integrator-Onboarding* role)
      1. Create, edit, delete TSI channel partners (integrators)
         1. Create, edit, delete users for this partner.
            1. Define username, password, description, & notification email
            2. User will be of class *Partner* and will have roles: *Site-Management*. Customer can temporarily grant user additional roles to access customer data and alarms.
         2. Specify customers that partner is authorized to access. TSI Admin can edit this at any time to update which customers a partner has access to.
         3. Deleting Partners will delete all associated partner user accounts. Customers will not be removed.
         4. When deleting a partner TSI could optionally specify a new partner to take over responsibility for the deleted partners customers. This would add those customers to the new partners authorized customer list
   2. Customer Management Page (*Customer-Onboarding* role)
      1. Enhance customer user capabilities
         1. Add sms phone # to user info for notification
         2. add support for WiFi-access user role. Sends information down to gateways to support WiFi logins with those credentials.
         3. Option to disable local WiFi created users. Only cloud created WiFi users will be allowed access on site.
   3. Site Management Page (*Site-Management* role)
      1. Gateway Configuration
         1. Per Gateway Configuration
            1. Data Collection:

sample interval (fixed at 1sec?)

stats aggregation interval

status upload interval

logging parameters: log name, start date, interval, auto-zero interval, length

local raw logging options: location, duration, enable/disable

* + - * 1. Alarm 2 (warning)

trigger threshold (value, time/sample count?, hysteresis func)

reset threshold

Enable/Disabled

mode: Auto-Reset, No Ack required

* + - * 1. Alarm 1 (STEL)

trigger threshold

reset threshold

Enable/Disable

mode: Ack Required, STEL period.

DIO output?

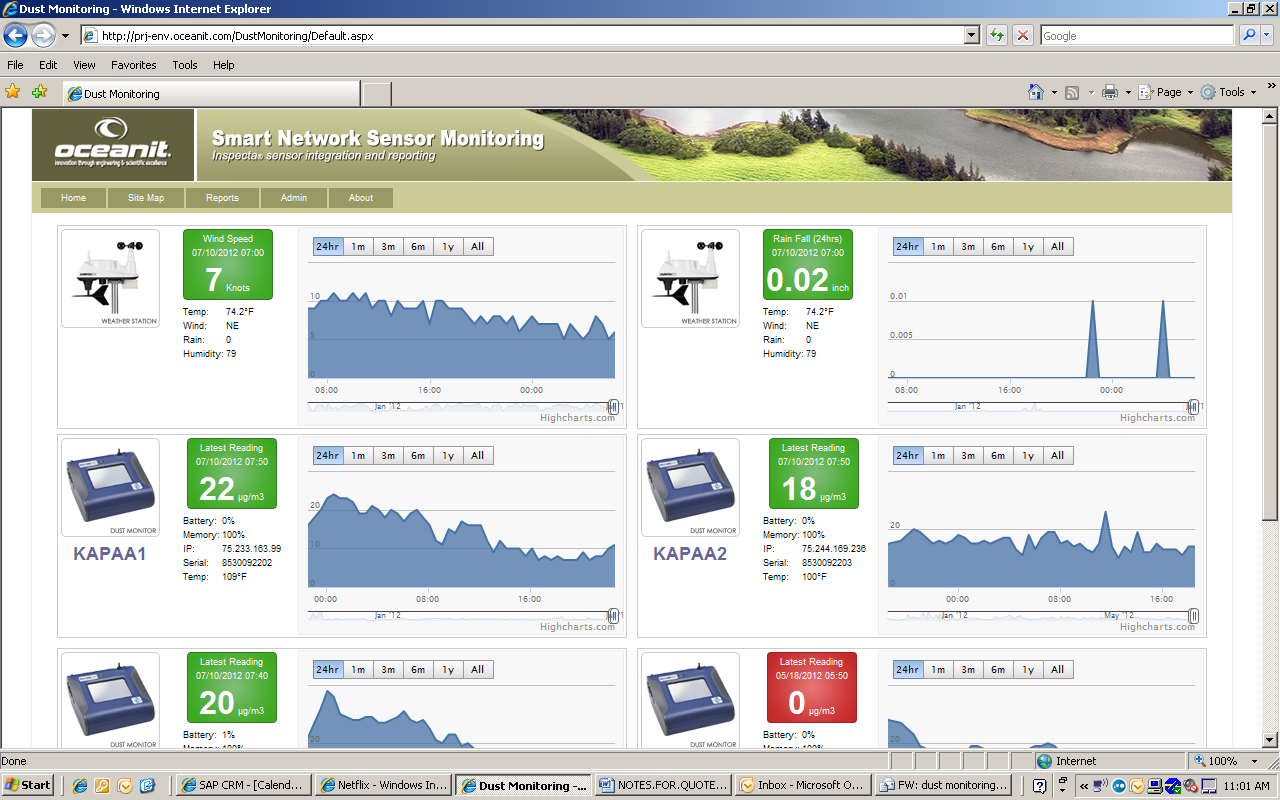
* + - * 1. Alerts:

Alarms, Sensor Status, Gateway Status

* + - 1. Site-Wide Configuration
         1. Configuration applied to all site gateway/sensors
    1. Notification Configuration
       - 1. Site notification default to users with *Customer-Alert-Management* role.
         2. Customer can override to email/sms #’s of choice
    2. Data Retention Configuration
    3. Gateway/Sensor Service Operations
       1. Auto-zero
       2. Calibration
       3. Reboot
       4. Firmware upgrade
       5. Connectivity test
       6. Service/Replace gateway and/or sensor equipment at a site
  1. Dashboard Summary view (per site) - (*Customer-Alert-Management* or *Customer-Data-Access* role)
     1. Geographic map view of site
        1. Current reading of all site sensors
        2. Current alarm status of all site sensors
        3. Plot information on google map overlay using gateway’s reported lat/long
        4. Links to move to sensor view page
     2. Advanced Plot Graphing
        1. User selects which channels and sensors to graph over a specified time window
        2. Other options on graphing, or graphical dashboard (see competitive charting below)
  2. Support view
     1. Grant temporary access. Customer user can temporarily grant TSI or their designated integrator access to their data: *Customer-Alert-Management* or *Customer-Data-Access* roles. Access is revoked after a configurable amount of time or when customer explicitly revokes it.

1. **Language Support**
   1. Provide capability for TSI to define/install additional language packs
      1. Chinese, German
2. **Automated/Internal Processing**
   1. Alarm/Alert report processing
      1. Support for SMS Alert notification
   2. Distributing/Synchronizing configuration with remote gateway
      1. Alarm & alert configurations
      2. Logging configurations

### Competitive Advanced Charting example:



## Validation, Training, and Deployment Support

* Conduct end-to-end System Testing of the solution
* Support of User Acceptance Testing for the resolution of critical defects
* Development of “Train the Trainer” Materials
* Conduct one Train-the-Trainer training session and administrator training session

## Deliverables

| **Deliverable Name** | **Deliverable Description** | **Deliverable Acceptance Criteria** | **Phase** |
| --- | --- | --- | --- |
| Project Plan  (Schedule and Timeline) | Living document that identifies the Project’s resources, tasks, deliverables, timeline, and dependencies. (Excel workbook, salesforce.com custom application, or MS Project document). | Customer accepts that the execution of the plan as stated is consistent with Customer’s resource, budget, and timeline requirements; Customer accepts roles and tasks assigned. (Formal email acceptance). | Plan |
| Requirements Workbook | Defines, documents, and prioritizes customer’s business and functional requirements (Excel workbook) | Customer accepts that this is the complete set of business requirements that will be used as the basis for designing and testing the system. (Formal email acceptance) | Analyze |
| Solution Design | Description of all high-level system components and business processes required to implement Customer’s requirements including some or all of: object model, to-be user processes, automation techniques, integrations, data management, 3rd party partners, and any other system design considerations such as SOX compliance (Word document) | Customer accepts the high level design components that are used to satisfy the agreed upon in scope requirements. The detailed design will be built based on this high level design. (Formal email signoff) | Design |
| Configured Application and Data Builds | Application build and data loads are conducted with frequent, iterative reviews. | Interim configurations and data loads, progress recorded in weekly status reports with formal email signoff on each report | Build |
| Final Validation Results Document | Record of the results of the executed test scripts (e.g. pass/fail, bugs identified, prioritized and resolved, etc.) (Excel workbook or custom salesforce.com testing application report) | Customer accepts that the Validation Results represent execution of the agreed-upon Validation Strategy and the outcomes are within the parameters specified in the Validation Strategy. (Formal email signoff) | Validation |

## Work Out of Scope

The following features are out of scope for Phase 1:

* Environmental DustTrak Monitoring (Phase 1 includes DustTrack Monitoring)
* Gas Monitor (C.7 in Customer Requirements document)
* Weather Station (C.8 in Customer Requirements document)
* Thermal Management Station (C.9 in Customer Requirements document)
* DustTrak Configuration (D.12 in Customer Requirements document)
* Multi-Language (D.13 and H.10 in Customer Requirements document)
* Gas STEL Alert (E.5 in Customer Requirements document)
* Gas Failure Alert (E.6 in Customer Requirements document)
* Aggregate Alerts (I.3 in Customer Requirements document)

# Approach

## Implementation

The Project Approach includes six key Phases: Plan; Analyze; Design; Build; Test/Validate; and Deploy. Each phase includes a checkpoint to confirm Scope, Schedule, Budget, and Resources.

The following graphic depicts the implementation methodology.



## Assumptions

* Digi can assist with deploying a pilot site, but deployment of additional sites will result in additional cost
* For the Gateway application, the solution will support the last two major releases available at time of delivery of these browsers: Chrome and Safari as well as iPhone and Android device support
* For the cloud application, the solution will support the last two major releases available at the time of delivery for these browsers: Chrome, Safari, Internet Explorer, Firefox
* Cellular performance cannot be guaranteed. Although Digi can design the system which demonstrates a particular alarm latency from IG to cloud, we cannot promise consistent alarm latency – especially over cellular. Some actions which might (when cell link is active and radio bandwidth allocated by the tower) take less than a second could take several minutes in unfortunate cellular conditions. These long lags are usually less than 0.5% of the time, but it is not a bug if on rare occasions an action over cellular takes up to 5 minutes.
* Customer will assign an internal day-to-day point of contact who will collaborate on confirming tasks, timelines, facilitate SME involvement, and track issues and risks throughout the duration of the engagement.
* Work will be performed at either the Customer site or one of the Etherios Solution Centers. The impact of travel to Customer locations that is deemed necessary will be determined during the first week of the project.
* For work performed on-site, as mutually agreed upon between Customer and Consultant, Consultant will be provided appropriate work facilities (e.g., work space, internet connectivity) to conduct project activities
* Customer is responsible for all third party relationships and procurement related to any integration solutions or the implementation of any integration solutions. Consultant does not recommend or warranty third party products or services.
* Language translation – Digi can supply a system supporting multi-language, but will not translate or define the foreign phrases. TSI will be responsible to define all translations.

# Implementation Breakdown

## Resources

* **Solution Architect (SA)**
  + Full time resource
  + Responsible for overall solution including requirements, design, documentation, validation, and deployment
  + Understands the customer business strategy and processes being translated into the solution
* **Technical Architect (TA)**
  + Part time resource
  + Responsible for technical and hardware design, architecture, and development oversight
* **Developer (D) - 4**
  + Part time resource
  + Responsible for development and unit testing
* **QA Analyst (QA)**
  + Part time resource
  + Responsible for documenting and executing test scripts
  + UAT support
  + Deployment testing
* **Project Manager (PM)**
  + Full time resource
  + Responsible for project timeline, schedules, budget, status, and team communications
  + Responsible for risk and issue identification and mitigation
* **Engagement Manager (EM)**
  + Part time resource
  + Provides immediate escalation point for the customer at any point in the program timeline
  + Oversees risk and issue management
  + Reviews and approves project deliverables, working with the project team to validate work plans and effort estimates
  + Provides subject matter expertise specific to the solution as well as significant experience with delivering complex projects that are similar in nature

## Design and Development Tasks

### Baseline Solution

**Gateway Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| Base WebPages | An estimated 20 web pages with input fields and multi-language hooks | Developer |
| Graphing Plugin | Adding graphing plug-in (High-Chart?) | Developer |
| Data management | Adding Data Management web pages (log file, other settings) | Developer |
| Chinese Input | Adding Chinese char input to smart-device | Developer |
| Manage Ethernet | locate DT when using fixed IP or DHCP, Ethernet comms | Developer |
| Polling, Logging | Basic polling mode, logging, auto-zero | Developer |
| Alarms STEL | managing alarms, STEL file, sync local and cloud | Developer |
| Design, doc system | design and document the gateway code | Technical Architect |
| Design, doc data xchg | design and document the message exchange for data, config, alarm/status ack - work closely with cloud designers | Solution Architect |

**Cloud Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| Schema | Design DB entities, tables, and relationships | Solution Architect |
| Dao/ORM | Develop dao layer and security controls | Solution Architect, Developer |
| UI:Skeletal app/login | Establish web app, login security, framework for remaining UI | Developer |
| UI:Cust Mgmt | Cust List, create/edit/delete dialogs, assign integrator, links to user/site pages, deactivate, purge data on delete | Developer |
| UI:User Mgmt | User list, create/edit/delete dialogs | Developer |
| UI:Site Mgmt | Site list, create/edit/delete dialogs, gateway list, add/edit/delete gw dialogs | Developer |
| UI:Dashboard | Sensor status list, sensor detail view, Alarm details, ack/reset actions, STEL data view, historical data, status, alert view | Developer |
| UI:Time Chart | Graph like sensor data for site sensors over time. select sensors, channel, window | Developer |
| UI:Data Export | Export stats data and/or STEL data to CSV for select sensors, channel, window | Developer |
| UI:Status Events | Display status events for selected site sensors over given time window | Developer |
| IP:connect/discovery | Detect initial gateway connect and discover connected devices | Technical Architect, Developer |
| IP:report processing | receive upcoming gateway reports for data, sensor status, gw status, alarms, and alerts | Developer |
| IP:notification | notify alarm/alert recipients of events via email | Developer |
| DC communications | Driver to communicate with device cloud | Developer |

**Other Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| Detailed Design Specs |  | Solution Architect |
| Domain & Certs |  | Technical Architect |
| Cloud App Hosting |  | Solution Architect |
| App Deployment & Config |  | Technical Architect |
| Device Cloud acct & subs |  | Solution Architect |
| Site Testing | 2 days of actual site testing, both gateway and cloud people involved, including gauging usability and user reaction. Perhaps one day early on in the development process, and second day near end of process. | Technical Architect |
| User Documentation | Draft out 2 manuals. First is an installation/maintenance guide; second is end user guide. | Solution Architect |

### Simple Mesh Add-Ons

These add-on items enable using a shared cellular uplink to reduce cellular costs. This would be basic data routing but still relatively simple UI.

**Gateway Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| Design | Design the gateway code | Technical Architect |
| Mesh Comms | Assuming serial link to Xbee 900 or 2.4, wrap and process packet stream, supporting multiple clients (mix host-app, device cloud, and DIA traffic) | Developer |
| Virtual Sensors | Enable any IG to mirror sensors on other IG, including mesh comms | Developer |
| IG Mesh Management | Allow IG in mesh to auto-detect the layout and uplink IG | Developer |
| Info web page | Single web page to browse/display the basic real-time (last period) data of remote IG on the present IG | Developer |
| SCI/RCI proxy | Allow uplink IG to forward normal Digi SCI/RCI to other gateways. | Developer |

### Advanced Mesh Add-Ons

These add-ons provide a richer experience for the local wifi app. We can provide options here like distributed configuration and alarm management.

**Gateway Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| Sitewide ACK/Reset | Allow a smart-device linked to WiFi on one IG to acknowledge/reset other IG | Developer |
| Sitewide Config | Allow a smart-device linked to WiFi on one IG to browse and configure other IG | Developer |
| File Transfer | Allow Python code (or any file) to be transferred between IG | Developer |

### Advanced Cloud Add-Ons

These add-ons provide a richer experience for the cloud app. We can provide options here like advanced user management, graphing, and reporting.

**Cloud Tasks:**

| **Task** | **Description** | **Resource** |
| --- | --- | --- |
| UI:Integrator Mgmt |  | Developer |
| SMS notification |  | Developer |
| WiFi users |  | Developer |
| Remote Gateway Config | Remote config of data, alrms 1 & 2, alerts | Developer |
| Remote Default Config | Apply default config to all gateways/sensors added to site | Solution Architect, Developer |
| Notification Config | Notification rules to override default users (ie specific email,sms) | Solution Architect, Developer |
| Data Retention Config | Data retention options and pruning logic | Developer |
| Gateway/Sensor Service Operations |  | Developer |
| UI:Geographic Maps |  | Developer |
| UI:Service/Support View |  | Developer |
| IP:Syncing services | Sync configurations with gateway | Solution Architect, Developer |

## Potential Sprints

These are potential sprint stages. Generally they are going to be 2-4 week sprints and are meant to be quick, show progress, and get early feedback on. We can give periodic demonstrations with TSI if we want. The other goal of this approach is to reduce risk by attacking the trickiest aspects of the design first rather than leaving them till the end. This reduces the chance of surprise slips in the development cycle. Note: These are potential sprints that will be re-evaluated during the Plan phase.

| **Sprint** | **Description** |
| --- | --- |
| **Project Setup** | **Initial setup stages of the project. Detailed specification of deliverables and development schedule** |
|  | Project Repositories |
|  | Detailed Design Specifications |
|  | Gateway/Cloud App Protocol Design |
|  | CL:Database Schema Work |
| **Development Bootstrap** | **Establish buildable, testable, deployable project frameworks and get running for gateway and cloud application** |
|  | Build, test, & deployment tooling |
|  | Gateway app skeleton |
|  | Cloud app skeleton |
| **Simple GW/Cloud Transport** | **Demonstrates exchange of mock data reports and commands per protocol design. This sets the state for concurrent development of both applications** |
|  | Gateway/Cloud communication drivers |
|  | skelton gateway report dispatchers and command handlers |
|  | skeleton cloud app report handlers and command dispatchers |
| **Sensor Data Reporting** | **Demonstrates the ability to talk to the sensor, take basic data readings, and send them up to the cloud app** |
|  | GW:rudimentary sensor driver capabile or reading data |
|  | GW:Rudimentary aggregation and reporting |
|  | CL:database implementation |
|  | CL:Rudimentary data report processing and insertion into db |
| **Skeletal UI** | **Demonstrates the ability to login to each app and view rudimentary sensor data** |
|  | GW: login, current sensor data |
|  | CL:login, skeleton dashboard with basic sensor data |
| **Robust Data Logging** | **Demonstrate full control over data collection, uplaoding, and local/remote display** |
|  | GW: Data Logging Config |
|  | GW: Full Data Collection & aggregation |
|  | GW: Full data report uploading |
|  | CL:Full data report history processing |
|  | CL:Dashboard data history |
| **Alarm Reporting** | **Demonstrate alarm configuration, processing, reporting, display, and acking (big sprint)** |
|  | GW: local alarm configuration |
|  | GW: alarm processing |
|  | GW: alarm reporting |
|  | GW: alarm ack/reset processing |
|  | CL:receive and store alarms |
|  | CL:Dashboard alarm status display |
|  | CL:Alarm ack/reset action |
| **Sensor Info & Operational Status** | **Demonstrate collection, reporting, and display of detailed sensor info and sensor/gateway operational status** |
|  | GW: sensor info collection and reporting |
|  | GW: gateway and sensor status collection & reporting |
|  | GW: local web display of sensor info and operational status |
|  | CL: receipts & storage of sensor info and operational status |
|  | CL: Dashboard display of sensor info and operational status |
| **User Management** | **Demonstrate management of multiple local and remote users** |
|  | GW: config 1-10 local users |
|  | CL: customer & user mgmt |
|  | CL: site mgmt |
| **Export, Graphing, and Reports** | **Demonstrate graphic and tabular reports of data as well as export capability** |
|  | GW: data graphing & export |
|  | CL: basic data graphing, export, & reports |
| **Notification** | **Demonstrate notification of alerts and alarms** |
|  | CL: Alert/Alarm email processing |
|  | CL: Notification & ReNotification logic |
| **Deployment** | **Demonstrate deployment of solution. Test Site Pilot** |

## Implementation High-Level Schedule and Cost Estimates

Note: The multi-language support requirement (including Chinese) nearly doubled the various web-page designs, which are already complex due to the need to support PC, tablet, and phone (with drastically different screen sizes and geometry).

## Hardware

* WR41-U8A3-WV1-XH at MSRP: Q1 - $1214, Q5k - $650
* 1 Ethernet 4 RS232 Gobi cellular Encryption/5 tunnels WiFi B/G USB Extended Temperature
* A 900Mhz mesh node (with up to 28-mile between nodes) P/N XM-C91-2P-UA is $145 Qty 1 (with antenna, power supply) and $125 Qty 1
* 2.4Ghz alternative (with up to 2 miles between nodes) would be $110 Qty 1 (with antenna, but without power supply) We actually have multiple P/N which would work at same price.
* Additional cell module (Gobi 3K) is a $300 adder

## Recurring Costs

## Hosting

**Initial Cellular Traffic Estimates (subject to change as functionality solidifies)**

* For each cellular-enabled IG, start with a basic overhead of about 1MB per month to be always connected to the cloud.
* For each DustTrak, assuming 1-minute interval, hourly upload, add another 1MB.
* For each DustTrak, assuming 5-minute interval, hourly upload, add another ¾MB.
* Assuming a few alarms per month, we don’t expect that to have a significant impact in this (maybe 0.01Mb per event?)
* For example, a site with 5 meshed boxes would be about 6MB of traffic for all, or without mesh it would be 5 cell-plans each IG with about 2MB traffic or 10MB for all if cellular carrier supports ‘pooled’ data plans.
* Reloading the IG Python code will cost about 1MB of traffic; reloading the full IG firmware may be from 4 to 6MB.

# Appendix 1: Actor/Use Case Analysis

Based on the actors described in section 3, the following section describes tasks for each actor. The tasks are linked back to the points in the customer requirement doc using the notation [CR-ID] where CR-ID denotes the requirement ID.

## TSI Administrator Tasks (cloud)

* 1. Manage customer accounts. New customer accounts are created when a TSI customer signs up for remote monitoring services. Once service agreements are in place the TSI administrator will create a new cloud account and grant the customer access to that account. From that point on, the customer can log in and access their account.
     1. Create a new customer with basic info, username, password, [D.2]
     2. Edit existing customer info
     3. Configure customer wide storage management options [H.8]
     4. Configure customer wide Cellular MNVO options [H.9]
     5. Manage customer access (activate/deactivate)[G.4]
     6. Delete customer account[G.4]
  2. Order fulfilment. When TSI ships orders to customers it has the opportunity to pre-provision the equipment directly into the customers cloud account.
     1. Pre-provision equipment to customer site? [H.7?]
     2. Repurposing equipment?

## Customer Administrator Tasks (cloud & local)

* 1. Manage remote work site. Once the customer has access to their cloud account, they can set up one or more work sites and manage the sensors at those sites.
     1. Create new work site with location and other pertinent info
     2. Edit existing work site info
     3. Oversee work site installation - *see Site Installer tasks which must be done by either TSI support personnel or by a designated integrator.*
  2. Manage remote work site access. The customer administrator can create additional user ids with limited access to one or more of the customer’s sites.
     1. Create/edit/delete additional users for this customer [D.2,G.3]
     2. Grant temporary access to TSI or integrator personnel for support assistance
  3. Remote reporting. The customer administrator (and authorized customer users) can access data reports of the remote customer sites.
     1. Dashboard showing key status information for each site customer has access to [H.1]
        1. Current alarm status [D.10]
           1. Acknowledge/reset fired alarms
        2. Current readings and statistics of all site sensors[D.4, D.5]
        3. Current operational status of all site sensors [D.7]
        4. Current operational status of all site gateways [D.8]
        5. Drill down historical view of sensor readings, statistics, and alarms [D.6,D.11,H.1]
        6. Ability to drill in on a sensor and get instrument identification information (model #, serial #, calibration date, firmware version). This information all comes from the device.[D.17]
     2. Geographic view of site [H.2]
        1. Current alarm status of all site sensors
     3. Time Chart view [H.3]
        1. plot historical data of up to two sensors from any site on a single chart
        2. User configures sensors, time window, and linear log axis
     4. Export data to CSV or equivalent [H.4]
        1. Sensor reading data. User specify time window & sensors of interest
     5. System status events [H.5]
        1. data transfer failures
        2. gateway/device logs
        3. alarm states
  4. Manage local work site (via wifi app)
     1. Cellular details:
        1. carrier selection
        2. APN (etc) is required
        3. Provisioning status (if required)
     2. Basic connection status - similar to X2e where we show internet and cloud access
     3. Manual entry to GPS coordinates.
  5. Manage local work site access (via wifi app)
     1. Local access is by a read-only and read-write user only. No larger collection of users. No user name? Use password to select R/RW?
     2. For security, local user details (name or password) cannot be used in cloud app - cloud users are not synchronized to local. Local user requires WiFi access.
  6. Local reporting (via wifi app)
     1. See basic graph showing data from gateway (last 30 days?)
        1. Change graph type or select a time/date range
        2. Due to lack of database tool, limited manipulation
     2. See event/alarm log as text
     3. One screen showing some real-time status (last readings, alarm/event status)
     4. In mesh design, show same ‘real-time status’ screen for any meshed units, but no other logs or history.
  7. Local alarm handling
     1. Can only acknowledge/reset from local box or from cloud.

## Site Installation and Management Tasks (cloud & local)

* 1. Installation & Management of work site via cloud app
     1. Initial new site setup
        1. Plan site layout and required infrastructure
        2. Physically set-up gateways and sensors
        3. Configure gateway to recognize sensor
     2. Add new equipment to cloud work site[G.1,H.7]
        1. Define new equipment (gateways, sensors, other)
        2. Prepare equipment for use
           1. Basic discovery and network config
           2. Assign meaningful names [H.12]
           3. Calibration & other special initial config [D.16]
           4. Configure gateway communication options[G.2]
           5. Verification
     3. Configure site data reporting (channels, intervals, etc...). Device is in streaming mode in which it continuously sends data to gateway. Gateway will need to parse incoming stream into channels, compute reporting period statistics, and apply alarm & reporting rules to the data. Using DIA framework gateway will batch up data for sending to the server on specified intervals and/or thresholds. Some of this is custom framework code.
     4. Define site alarms
        1. Alarm trigger & reset thresholds for various data streams
           1. Gateway alarms [D15]

Trigger/Reset criteria

STEL action enabled?

* + - * 1. Server alarms

Gateway connectivity alarms [E.7]

DataPoint missing alarms?

* + - * 1. notification mechanisms (gateway & server) [I.1]

email: list of email addresses

SMS: list of #’s (need sms gateway support… which carriers required)

* + 1. Configure site sensor logging. *Note, it is unclear yet if this needs to be set per sensor or site-wide.[D.14]*
    2. Service existing equipment at work site
    3. Remove old equipment from work site. Previously recorded data must remain available.
  1. Installation & Management of work site via local wifi app
     1. Assuming no master app: Initial new site setup

## TSI Support Tasks (cloud)

* 1. Service Management [H.6]
     1. Reboot Gateway
     2. Update Gateway Firmware
     3. Remote Gateway/Device tests (connectivity, etc)
     4. Data storage archive/deletion? See question on H.6 below
     5. Remote calibration?

## Customer Operator Tasks (cloud & local)

* 1. Access to Reporting Dashboard described under Customer Administrator. This gives operator access to current status of site as well as access to any alarm conditions that may need to be responded to. Operator can acknowledge or reset firing alarms as needed. *Confirm need for this role: does it need further restriction or expansion or should we combine with customer administrator role. [D.10]*

## Internal Cloud Processing (non-UI)

* 1. Alarm processing
     1. Receive alarm from remote gateway and record alarm state in cloud
     2. Broadcast detected alarm to configured destinations
     3. Rebroadcast alarm every 30 minutes until acknowledged [I.2]
  2. Data Retention Policy (for each data point):
     1. Define short-term handling - for example, all data for first 24-hours
     2. Define mid-term handling - for example an hourly average after 24-hours with original data purged
     3. Define long-term handling - for example, after X months, export to customer as Y format, purge all old data

## Internal Local Gateway App Processing

* 1. Device communication
     1. Driver that understands device communication protocol
     2. TCP/IP socket on port 3602 (*See questions: protocol document is unclear if IP is supported, or raw Ethernet packets are required*).
  2. Custom DIA Sensor Driver:
     1. A sensor is defined by a globally unique string, concatenation of
        1. gateway device-id (from Digi device)
        2. sensor model (from TSI protocol)
        3. sensor serial number (from TSI protocol)
     2. In addition, each sensor may have:
        1. mesh address (may be for 1 sensor if 1-to-1 with Xbee-232, or for a gateway then shared by all sensors on that gateway)
        2. sensor index in gateway (1 to N)
        3. short-local name (may be like D01, D02, D03, etc)
     3. A Sensor instance with channels
        1. interval (time in seconds used for the avg/min/max - Rd/Wr)
        2. minimum seen in last time interval (as floating point, RdOnly)
        3. maximum seen in last time interval (as floating point, RdOnly)
        4. average seen in last time interval (as floating point, RdOnly)
        5. TWA seen in last time interval (as floating point, RdOnly)
        6. alarm config (as string point, Rd/Wr)
           1. a command line which can set variable tweaks - levels, hysteresis, time-delay.
           2. Because is a string, can add things in future.
           3. reading returns the existing config as text string
        7. alarm (as enum, Rd/Wr) read shows status, write attempts ‘move’
           1. 0x00 = normal/no alarm
           2. 0x01 = abnormal/alarm (ack/reset not required)
           3. 0x02 = returned to normal, waiting for ACK
           4. 0x03 = still abnormal, waiting for ACK
           5. 0x04 = returned to normal, was ACKd, waiting for RESET
           6. 0x05 = still abnormal, was ACKd, waiting for RESET
           7. 0x06 = returned to normal, waiting for ACK, waiting for RESET
           8. 0x07 = still abnormal, waiting for ACK, waiting for RESET
           9. Writing 0x02 to alarm is ACK
           10. Writing 0x04 to alarm is RESET
           11. Writing 0x06 to alarm is ACK and RESET
           12. (Note that after the write, return value may not be as expected. ACK will always register, but RESET is ignored if field condition is still abnormal.)
        8. alarm\_field (T/F as to the actual field status of the alarm without concern for ACK/RESET state.)
        9. alarm\_relay (T/F if external alarm should be active or not)
        10. online (T/F if data is being received as expected, RdOnly)
        11. model (per the TSI protocol response, RdOnly)
        12. serialnumber (per the TSI protocol response, RdOnly)
     4. Data collection and statistical reduction
        1. Data is collected at a high rate and held as collection
           1. May need a lower limit (1 second?)
        2. At configured interval
           1. We will want min/max reasonable interval. Too short causes problems over cellular. Too long may cause memory problems for the average calculation.
           2. Calculate pure average, minimum, maximum
           3. Calculate TWA per TSI formula
           4. Can enable/disable synchronization with clock (so 5 minutes means on the hour, at :05, :10, etc)
           5. If synchronized to clock, need a randomizing component for cloud upload of normal non-alarm data.
        3. Optionally log raw data to USB flash drive (CSV?) retain based on flash drive size
        4. Must retain min/max/avg on gateway for 30-days
     5. Alarm definition & trigger/reset condition
        1. Configure base set-point for alarm
        2. Configure hysteresis for alarm (default?)
        3. Configure time-delay for alarm trigger (default 2 seconds? need 2 samples in alarm?)
        4. Configure alarm mode
           1. Is high or low?
           2. Is ACK required?
           3. Is RESET required?
           4. Is RingBack required? (return-to-normal)
           5. Assert digital output?
           6. Is STEL event?
        5. definition can be done locally in wifi app or remotely via device cloud
     6. STEL File?
        1. STEL = Short-Term-Exposure-Limit
        2. 15 minute file? is this raw 1-second data? What is format?
     7. Alarm processing
        1. Alarm is evaluated on raw high-speed data (aka: once per second)
        2. Alarm going True is queued for upload at high-priority
        3. If RingBack is True, return to Normal is treated as high-priority, else may be normal priority
        4. If required, remote ACK/RESET are processed
        5. If required, alarm output is handled as OR of all sensor alarms
        6. Locally, no external push notification beyond digital output (aka: lamp) is supported.
        7. Via local WiFi one can see alarm status. If ACK or RESET, then no cloud record?
     8. Keep local flash log of time of alarm, return to normal, ACK, RESET?
  3. Other-Node data - dummy local sensor image
     1. If mesh is supported, support dummy drivers which allows any gateway to have a copy of another gateway sensor objects
     2. assume gateways without upload (no cellular) unicast or broadcast their sensor data at time interval. All gateways with a copy of dummy data save last data including time-stamp, etc
     3. No alarm processing is done in dummy driver - values shown are ‘last seen’ and may be out of date
     4. Any alarm ACK/RESET or config hitting dummy driver is forwarded back to correct gateway, so dummy driver must include full address information
     5. Optionally a function to get/refresh the data manually
  4. Custom Alarm Output driver
     1. OR the ‘alarm\_relay’ outputs of any sensor instances on this device
     2. interface to WR41 output (or DTII output).
     3. support manual on/off for testing
  5. Upload
     1. Normal interval/event data to be done at configured interval, such as
        1. Once per sensor interval
        2. Once per hour (etc)
        3. Support enable/disable of gateway going offline between uploads
        4. Must support time randomization function since normally sensor readings are time synchronized to the clock.
     2. Alarm data to be uploaded faster
     3. If this node is an uploader, then ‘other-node’ data is queued for upload, including higher priority for alarms.
     4. If cellular upload link is not available, then save on gateway for X days? X samples? TBD
  6. Local Data Log
     1. Only the interval data is logged for 30-days
     2. The log won’t be updated more than once per 5 minutes, so interval rate faster than this will cause a small RAM cache to be used
     3. Optionally, data might be saved in a fashion which can be read by TSI TrakPro
     4. Optionally, if USB flash drive available, all data could be logged in daily files.
  7. Local HTML5 web pages
     1. Users can log in as read-only or read-write. No individual users.
     2. Data Pages (read-only and read-write access)
        1. View sensor history (per local 30-day max log)
        2. View real-time data (last sensor reading, status of alarms)
        3. View event/alarm log
        4. View STEL log(s)
     3. Control pages (read-write user can enable/disable these for read-only users)
        1. Ack/reset alarms
        2. Adjust alarm settings
     4. Settings pages
        1. WiFi settings
        2. access passwords
        3. log settings (interval, etc)
        4. cellular settings
     5. In mesh design, limited pages of other boxes data. Note: to support cloud config, we do need to define the protocol (or grammar) used for remote read/write of all settings. However that does not mean we need to build these into the local HTML5 pages.
  8. Meshed Box Support - only 1 of “N” boxes on a site need a cellular account
     1. A setting to define a box as an ‘uploader’ or ‘meshed’. Some auto-detect might be valid such as if a WR41 has no cell module (or no active account)
     2. Python code to send out a broadcast on mesh, query for the upload node MAC address, which could be saved locally in flash (or re-located each reboot?)
     3. Setting up local dummy sensors (see DIA driver section above)
        1. The upload node creates one (or ‘N’) dummy sensor instances for each local mesh node.
        2. Optionally, every node creates such instances for all other nodes. Assuming nodes periodically broadcast data status, all nodes could be updated in 1 message
     4. Mesh nodes would need to periodically send out a snap-shot of their real-time data state - likely at the interval setting and if any alarm state changes
     5. The upload node would also require an forwarding mechanism for settings and alarm ack/reset. Assuming the cloud sees these dummy-instances as real sensors, then writing a setting change to the dummy instance would cause a forward by unicast. feedback to cloud is critical, as forwarding might fail

## Other

* 1. nls capable - including support for Chinese, Japanese, etc.
     1. Perhaps the unicode could be Web only? I am not sure what would be involved in making Chinese web pages on a WR41, but the WR41 setup and config itself will NOT support Chinese.
     2. How complex to have smart-phone data/alarm displays in Chinese?
  2. Languages required in phase 1
     1. English
  3. Languages required in phase 2