Software Requirements Specification

Condenser: Data Logging Platform

Version 0.1 draft

Prepared by:
Jesse Blum
for the
Horizon Digital Economy Research Institute
Nov, 2011

Revision History

Name	Date	Reason For Changes	Version
SRS-DLP-0.1	11-2011	First draft	0.1

Contents

1 Scope **3**
1.1 Problem Statement . . . 3
1.2 Objectives . . . 3
1.3 Document Overview . . . 3

2 Requirements **4**
2.1 Actors . . . 4
2.2 Usage Scenarios . . . 4
2.2.1 Relate Project: Condenser Installation . . . 4
2.2.2 Relate Project: Oasis of Connectivity . . . 5
2.2.3 Energy Project: Trickle of Connectivity . . . 5
2.3 Functional Requirements . . . 6
2.4 Non-functional Requirements . . . 6
2.4.1 Reliability and Security . . . 6
2.4.2 Test Data . . . 7
2.4.3 Performance . . . 7
2.4.4 Supportability . . . 7
2.4.5 Implementation . . . 7
2.4.6 Interface . . . 7
2.4.7 Operation . . . 7
2.4.8 Packaging . . . 7
2.4.9 Legal . . . 8

3 Related Works **9**
3.1 Data Brokers . . . 9
3.1.1 Pachube . . . 9
3.2 Data Replication and Mirroring . . . 9
3.3 Distributed Filesystem . . . 9
3.4 Cloud Synchronization . . . 9
3.5 SQL Azure DB . . . 9
3.5.1 Unstructured Database Options . . . 9
3.5.2 Structured Database Options . . . 9
3.6 Policies and Rules . . . 9
3.7 Agents . . . 9
3.8 Sensor Metadata . . . 9

1 Scope

This Software Requirements Specification (SRS) identifies the requirements for the Condenser Data Logging Platform.

1.1 Problem Statement

A trend across various Horizon projects (Energy and Relate for instance) is to provide distributed (immutable) sensor data capture, process and storage facilities. The connection between the local environment and external control and storage environment, however, is intermittent and unreliable. This disconnection results in many disadvantages such as data loss. Horizon projects have heretofore provided bespoke solutions to the external connections, however these have become difficult to maintain with the increase of projects.

1.2 Objectives

The aim of the Condenser (meta)-project is to provide a configurable component for local logging of immutable data to a local server with intermittent synchronisation to external (possibly cloud-based) storage environments. The Condenser component will be made widely available and be data-type neutral.

1.3 Document Overview

Section 2 is divided into functional requirements and the non-functional requirements. The functional requirements explain the interactions between Condenser and its environment (including usage scenarios and cases). The non-functional requirements explain the quality constraints. Section 3 provides critical review of state-of-the-art existing options and existing components that may be leveraged in the delivery of Condenser.

2 Requirements

This section defines functional and non-functional requirements for the Condenser tool. It begins with an enumeration of usage scenarios that Condenser could be involved in. These are used to help determine the following requirements.

2.1 Actors

These actors are involved in the following usage scenarios:

- Horizon project staff (HPS)
- Active Ingredient staff (AIS)
- Member of the public (MP)
- Energy project participant (EPP)

2.2 Usage Scenarios

2.2.1 Relate Project: Condenser Installation

Participating Actors: Anne:AIS , Harry:HPS
Event Flow:

1. Anne prepares to run an energy and climate data collection exercise in the Brazilian rain forest using open mobile sensing kits (OMSK).
2. Harry installs Condenser on the local server component of the OMSKs.
3. Harry configures a Cloud-server to accept data uploads from each OMSK.
4. Harry uses the Condenser RESTful interfaces to configure them to log the data from their local stores to the cloud when connectivity is available. He also configures them to run in low-powered mode by setting their reconnection attempts to a low amount and to only send data in aggregated bursts every hour.

2.2.2 Relate Project: Oasis of Connectivity

Overview: Condenser's behaviour in long period without connectivity
Participating Actors: Anne:AIS, Miguel:MP
Event Flow:

1. Anne conducts a public art activity in the Brazilian rain forest.
2. Anne gives Miguel an OMSK.
3. Miguel activates the OMSK as part of the activity.
4. On start up the OMSK activates Condenser and begins to capture temperature, humidity, C02 and energy information.
5. No network connectivity is available to Condenser so it evaluates the expected data load and determines that no action needs to occur until the next connection attempt. [Note that this would be much more complicated if Condenser determined that the current data capture rates exceeded the storage capacity]
6. Miguel returns the OMSK to Anne, and she returns to her office with it.
7. Condenser successfully attempts to connect to the network after its timeout.
8. Condenser transfers the data from the OMSK to the Cloud-store.
9. Anne looks at the day's data using the web-based visualisation tools. These have immediate access to the Cloud-data.

2.2.3 Energy Project: Trickle of Connectivity

Overview: Condenser's behaviour during intermittent connectivity
Participating Actors: Harry:HPS, Ernie:EPP
Event Flow:

1. Harry is conducting an energy study.
2. Harry sets up a Shiva Plug computer with an energy monitor.
3. Harry installs Condenser on the Shiva Plug.
4. Harry configures Condenser to attempt to reconnect whenever possible and to send discrete data points to a rack-mounted server using particular authentication details.

5. Harry installs the energy monitor in Ernie's home and configures it to use Ernie's home router.

6. Condenser attempts to transmit data as the energy monitor receives them, but connectivity is intermittent owing to Ernie's unstable Internet connection.

2.3 Functional Requirements

- Condenser is a data logging infrastructural component for use in software projects requiring intermittent synchronization with external environments.

- Functionality of Condenser is configurable through a RESTful interface. Clients can use HTTP calls to configure where and how often data is stored. Condenser can be used with cloud-based storage or more traditional server storage.

- Condenser is capable of handling disconnected operation by ensuring data are stored locally until network connection is resumed. Condenser evaluate expected data needs while offline and can be configured to clean, aggregate or delete data to handle limited local drive space.

- Condenser does not have (or need) a graphical user interface.

- Condenser will automatically provide standard metadata.

- Condenser will log its own performance to an external repository (with configurable repository and verbosity).

- Condenser should run as a background service that turns on automatically at startup

2.4 Non-functional Requirements

2.4.1 Reliability and Security

Condenser should continue to work reliably during periods of network disconnect. Error conditions that arise during periods of disconnect should be logged in a similar fashion to sensor data and transmitted externally (to a configurable store) upon resumption of network connection. Configurable options should be made available to allow for secure data transmission and storage. Data store connection settings must be encrypted.

2.4.2 Test Data

Even though Condenser will be data-type neutral, it will be tested using temperature, humidity, energy and C02 sensors data. Data sets will include: time-series data, discreet values and aggregate information. Condenser will be tested with up to 2GB of data.

2.4.3 Performance

Condenser' s external storage performance during times of reliable network connection will be configurable is order for an administrator to choose the amount of bandwidth will be taken up transferring data offsite. Condenser should support the offsite data transfer of up to 4095 sources as well as one more for its own logging information.

2.4.4 Supportability

Condenser tests, code, installation, administration and usage will be well documented.

2.4.5 Implementation

The local logging Condenser component will be operational on a Plug computer. Off the shelf technology may be (and indeed is encouraged to be) used to get a working version of Condenser built as soon as possible. New developments should be test-driven with an emphasis on documentation.

2.4.6 Interface

Condenser should support a RESTful interface for its activation, configuration and data transmission. Condenser should be compatible with other Relate project.

2.4.7 Operation

Condenser will be managed by Horizon staff or project clients.

2.4.8 Packaging

Condenser will be installed initially by Horizon project staff, with a view that client system administrators will be supported in the future. Condenser will initially be rolled-out for beta-testing in February, 2011.

2.4.9 Legal

Condenser will be licensed under The GNU Affero General Public License (AGPL 3). No liability will be assumed by Condenser's developers for losses incurred through its use since it is experimental and research driven software. Only Free (libre) software will be used for the development of Condenser.

3 Related Works

3.1 Data Brokers

3.1.1 Pachube

- RESTful
- Environment, Datastream and Datapoint model
- Push and Pull capabilities with “live” and “frozen” status
- Supports HTTPS/SSL
- Authentication is handled using API keys.

3.2 Data Replication and Mirroring

3.3 Distributed Filesystem

3.4 Cloud Synchronization

[1]

3.5 SQL Azure DB

3.5.1 Unstructured Database Options

- CouchDB
- MongoDB

3.5.2 Structured Database Options

3.6 Policies and Rules

3.7 Agents

3.8 Sensor Metadata

References

[1] S. Uppoor, M.D. Flouris, and A. Bilas. Cloud-based synchronization of distributed file system hierarchies. In *Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS), 2010 IEEE International Conference on*, pages 1–4. IEEE, 2010.