

# HamšČi



---

***PERSONAL SPACE WEATHER SYSTEM***  
***LOCAL HOST***  
**FUNCTIONAL SPECIFICATIONS**

---

Version Number: 0.76

Version Date: Sep. 25~~Aug. 19~~, 2019

## VERSION HISTORY

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
0.1	W. Engelke	<a href="#">5/10/2019</a>			
0.2	W. Engelke	5/13/2019			Added unattended restart
0.3	W. Engelke	5/24/2019			Changed text to consider that the Tangerine includes both the DE and the SBC; added more functionality and notes from Dayton 2019 meeting. Elaborated 2 use cases.
0.4	W. Engelke	6/4/2019			Added section numbers; added reference to TangerineSDR Requirements doc.
0.5	W. Engelke	7/9/2019			Added mention of throttling, personal propagation monitoring, magnetometer
<a href="#">0.6</a>	<a href="#">W. Engelke</a>	<a href="#">8/19/2019</a>			<a href="#">Clarified use cases; add to general requirements; added discussion of firehose mode</a>
<a href="#">0.7</a>	<a href="#">W. Engelke</a>	<a href="#">9/25/2019</a>			<a href="#">Added Use Case 3; added method of updating; added interfaces discussion; added future features concepts; added Section 508 disclaimer; added section on remote system updates</a>

## Table of Contents

<b>1. INTRODUCTION .....</b>	<b>5</b>
1.1. System Overview .....	6
<b>2. FUNCTIONAL REQUIREMENTS.....</b>	<b>7</b>
2.1. GENERAL REQUIREMENTS - overview.....	7
2.1.1. Operating Environment .....	8
2.1.2. Assumptions and Dependencies.....	8
2.2. User Interface Requirements .....	9
2.2.1. Guiding Principles .....	9
2.2.2 A User-friendly Interface .....	9
<b>3. MAJOR FUNCTIONALITIES.....</b>	<b>9</b>
3.1. Configuration and Setup .....	9
3.2. Startup and connecting to the Data Engine .....	10
3.3. Connecting to the Central Control System.....	10
3.4. Central Request / Response / Upload (Use Case 1: Data Collection) .....	10
3.4.1. Heartbeat .....	12
3.5. Large Local Server Data Collection (Use Case 2: Firehose) .....	13
3.6. Local Data Reduction (Use Case 3: Continuous Upload) .....	14
3.7. Local Signal Processing and User Notifications.....	14
3.8. Logging .....	15
3.9. Help .....	15
3.10. Interfaces .....	16
3.11. Remote System Updates .....	16
Updating the Local Host.....	16
Updating the DE .....	16
<b>4. FUTURE FEATURE CONCEPTS.....</b>	<b>18</b>
<b>5. TECHNICAL NOTES.....</b>	<b>19</b>
<b>6. LOGICAL DATA MODEL .....</b>	<b>20</b>
<b>7. REQUIREMENTS TRACEABILITY MATRIX .....</b>	<b>20</b>
<b>1. INTRODUCTION .....</b>	<b>4</b>
1.1. System Overview .....	4
<b>2. FUNCTIONAL REQUIREMENTS.....</b>	<b>5</b>
2.1. GENERAL REQUIREMENTS - overview.....	5
2.1.1. Operating Environment .....	5
2.1.2. Assumptions and Dependencies.....	6

2.2. User Interface Requirements ..... 6  
2.2.1. Guiding Principles ..... 6  
2.2.2. A User friendly Interface ..... 7  
**3. MAJOR FUNCTIONALITIES ..... 7**  
3.1. Configuration and Setup ..... 7  
3.2. Startup and connecting to the Data Engine ..... 7  
3.3. Connecting to the Central Control System ..... 7  
3.4. Central Request / Response / Upload (Use Case 1: Data Collection) ..... 8  
3.4.1. Heartbeat ..... 9  
3.5. Large Local Server Data Collection (Use Case 2: Firehose) ..... 10  
3.6. Local Signal Processing and User Notifications ..... 11  
3.7. Logging ..... 11  
3.8. Help ..... 11  
3.9. Localization ..... 12  
3.10. Remote System Updates ..... 12  
**4. TECHNICAL NOTES ..... 12**  
**5. LOGICAL DATA MODEL ..... 13**  
**6. REQUIREMENTS TRACEABILITY MATRIX ..... 13**

## Figures

Figure 1. Conceptual Overview. .... 6  
Figure 2. Use Case 1. Request/Response process model. .... 11  
Figure 3. Use case 2, Firehose. .... 13  
Figure 4. Use Case 3, Local data reduction + continuous upload. .... 14  
Figure 1. Conceptual Overview. — 4  
Figure 2. Use Case 1, Request/Response process model. — 8  
Figure 3. Use case 2, Firehose. — 10

## 1. 1. INTRODUCTION

This Functional Specification describes the capabilities of the Local Host computer (typically a Single Board Computer, or SBC) which works together with the Radio (single board SDR including ADC, FPGA and Data Engine (DE)) to form the Tangerine. The Tangerine is the local (primarily a receiver/data collector) subsystem for the Personal Space Weather ~~Station~~system (PSWS). Within the Tangerine, the Local Host communicates with the Radio portion through a Gigabit switch (GbS) built into the DE.

In the PSWS, the Tangerine collects spectrum data (in general I&Q); it can save these data one of ~~three~~ ways: (a) locally in a 24-hour ring buffer, with upload to a target system for storage, ~~or~~ (b) direct transfer to a high-capacity server on the same network as the Tangerine, or (c) continuous upload to Central Host of pre-processed, low bandwidth data such as magnetometer and FFT of WWV data, 5 bands at 10 Hz bandwidth each sent once per second. In the first case, data analysis can be on the Central Control System, and in the second, data analysis can be on the large local server; in the third case, preprocessed data is loaded directly into the Central Database. Refer also to *TangerineSDR Requirements* document (see [https://tangerinesdr.com/TangerineSDR\\_documents/](https://tangerinesdr.com/TangerineSDR_documents/)).

A parallel project with the Tangerine SDR is the CWRU low-cost PSWS; in several cases, that project will be referred to where it would be advantageous to use a common approach in both systems. This refers primarily to communications from the PSWS to Central Control.

All functionality documented here is for Phase 1 of the project. Numerous other concepts have been proposed that would significantly increase scope; for risk management, most of these are being designated for future phases (see section Future Feature Concepts).

### 1.1. SYSTEM OVERVIEW

As mentioned above, there is more than one way to for the DE to save or upload data; the Local Host connection is one of them; this specification documents the capabilities of that Local Host. An overview of the system in this configuration is shown in Figure 1.

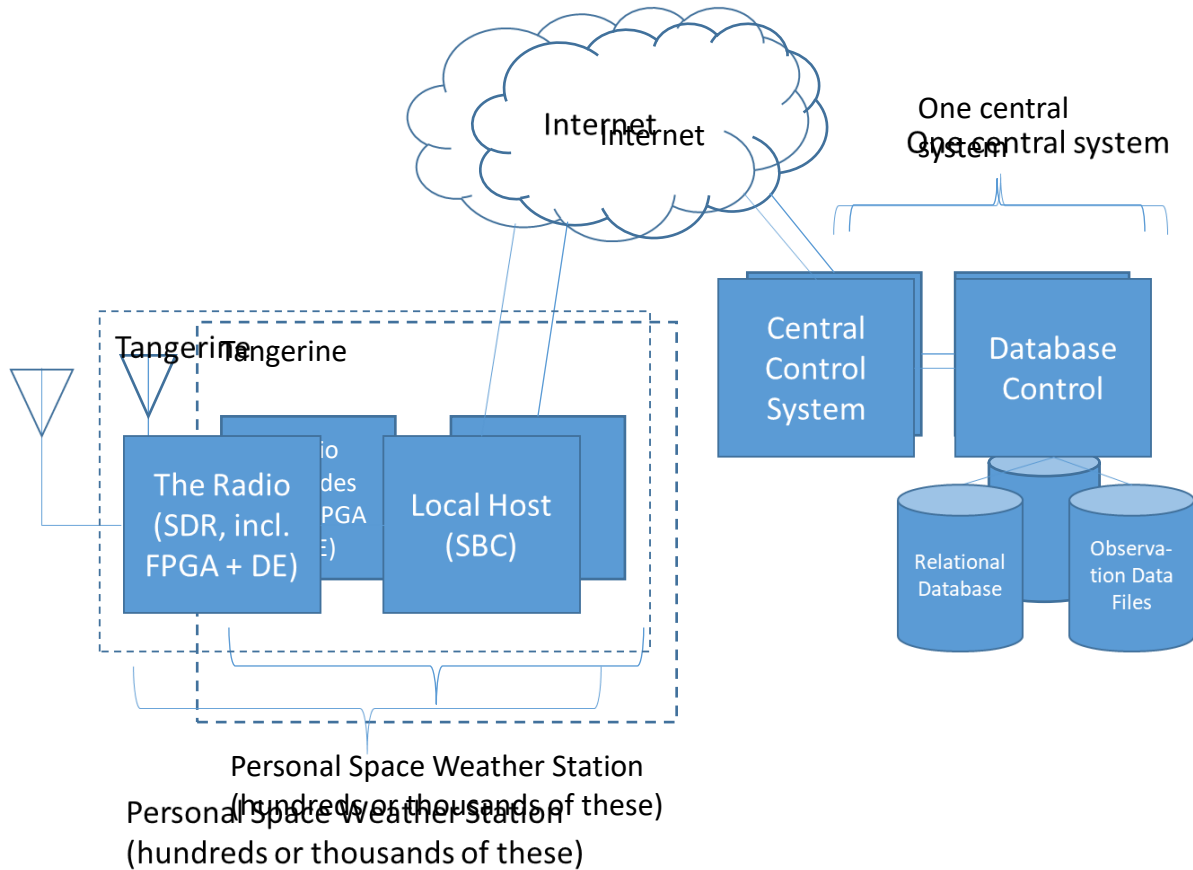


Figure 1. Conceptual Overview.

USE CASE 1. Spectrum data collected by the DE is uploaded to the Local Host which maintains a ring buffer of 24 hours' worth of observations. The Central Control System is able to send the Local Host a request to upload selected observation data. Comment: need to avoid producing so much data that the system quickly (a few months) exceeds the data write lifetimes for reasonably priced devices – data collection capacity will need to be optimized for the trade-off between storage device cost and total data collection capability. Observations are sent as large files which are saved (as discrete files) and cross-referenced in the Central Database.

The Local Host configuration is to allow an individual PSWS to participate in crowd-sourced data collection in spite of a low-bandwidth Internet connection, and also to provide some local signal analysis and propagation reporting for the individual

PSWS owner. In this case, the Local Host initiates communication with the DE; the Local Host then collects data from the DE and saves them to the local ring buffer.

USE CASE 2. In an alternate configuration the DE uploads directly to a remote server. This is intended for use in institutional settings where the local area network is high bandwidth and locally attached computers are very fast and can handle the full stream of data being collected. In this case, the Local Host is used to put the DE into local server upload mode. Data archiving is handled by the local server in whatever form the user decides.

USE CASE 3. Real-time, low-bandwidth data stream to Central Control system. An example of using the system this way would be for continuous upload of a low-bandwidth, low data-rate series of measurements, such as magnetometer readings and FFT snapshots of small band segments around WWV frequencies for observing Doppler shifts. PHASE 1 Target: ability to upload FFT snapshots of WWV at 5, 10, 15 and 20 MHz plus magnetometer reading at one second intervals. Comment: this will provide enough data for useful science related to TIDs without posing a heavy load on the user's upload bandwidth, and should also leave enough system capacity for multiband propagation data collection (FT8, WSPR, etc.). Uploaded data are saved in the Central Database directly (not as files).

## 2. FUNCTIONAL REQUIREMENTS

Functional requirements capture and specify intended behavior of the system being developed. They define things such as system calculations, data manipulation and processing, user interface and interaction with the application, and other specific functionality that show how user requirements are satisfied.

### 2.1. GENERAL REQUIREMENTS - OVERVIEW

Requirements and capabilities of the overall PSWS system are described in a separate document. This Functional Specification covers only the Local Host, which shall have the following capabilities:

- Can connect to exactly one DE
- Send commands to the connected DE
- Provide a web browser and web-based interface for user interaction
- Receive uploaded spectrum (I&Q) data from connected DE
- Save uploaded spectrum data to a ring buffer for 24 hours, or upload spectrum data directly to large server on the same network<sup>1</sup>, or preprocess data (using GNURadio) and upload results continuously to Central Control
- Connect to Central Control System
- Respond to commands from the Central Control system to upload data
- Provide a variety of propagation and band-opening information of interest to the radio amateur, including reporting to WSPR, RBN, and PSK

---

<sup>1</sup> We specify "local" here primarily due to the need for continuous bandwidth of about 1 Gbps when running; this also simplifies security requirements

networks and email notification when selected (digital) stations are heard (to include FT4, FT8, WSPR; further details TBD)

- Can be updated from the Central Control System. Need to support two updating methods:
  - SD Card Swap (requires reboot)
  - Update while running - Presumably we need to let user select whether automatic updates from Central are allowed, or if user wishes to schedule these at a specific time
- Is able to push firmware updates to the DE for loading into the FPGA [??Need to understand how to do this; doesn't this require a USB Blaster?] (Note: need to add Security Concept for this QQQQ)

### 2.1.1. Operating Environment

The Local Host is distributed as an image to be burned to a ~~SD or~~ micro-SD card using an imaging utility such as Balena Etcher.<sup>2</sup> For Phase 1, the Odroid N2 SBC will be supported. (Logical compatibility with the CWRU low-cost PSWS will be maintained via a common interface to the Central Control system). UsersThe user is expected to supply a SBC compatible with [ Ubuntu, current plan (??) ] Support will be provided only for the [ Odroid (??)?]; users electing to install and use the system on other types of hardware are on their own.

The SBC should have the minimum features as follows:

- USB-3 port for attached high speed disk or SSD
- Gigabit Ethernet (GbE)
- ARM, 4 fast cores, 4 slow Quad-cores, 2GHz
- 4 GB RAM
- Refer to Odroid N2 specifications:
  - -

### 2.1.2. Assumptions and Dependencies

The assumptions and dependencies for the project are as follows:

- User-provided equipment includes the SBC, micro-SD card, attached USB-3 hard drive with at least 4 TB capacity (QQQQ need to determine required speed), GbE switch, Internet connectivity with user-configurable router, ability to burn image to micro-SD card
- The Local Host will be on a local network behind a router and not exposed to the public Internet (i.e., it does not need to have require a built-in firewall, but compatible Linux versions generally contain one)
- The user will provide a workstation (i.e. a HDMI compatible monitor, keyboard, and mouse) either directly connected to the Local Host or on another PC on the same network as the ~~SBC and~~ Tangerine
- Dependencies for the DE/FPGA documented elsewhere

---

<sup>2</sup> Fast micro-SD cards such as the SanDisk Extreme Plus are needed; more ambitious users may optionally elect to copy their O/S to eMMC, which can also be mounted on Odroid and booted via the MMC boot switch.



## 2.2. USER INTERFACE REQUIREMENTS

### 2.2.1. Guiding Principles

- This system is intended for use by both amateurs and professionals, so it should have default configurations and easy-to-use settings to enable startup of the base use case system by a non-expert; but allow access to advanced functionalities by experts
- Security (access) features roughly equivalent to a modern desktop PC shall be provided (see Assumptions and Dependencies), e.g., a User-ID and password for access, where the system is delivered with standard defaults and the user has the option to change these
- Configurations will be saved in meaningfully/logically named files and containing human readable/editable settings.
- System functions will be logged for debugging, and the user will have the option to turn off logging or select circular logging to avoid using up all space
- Major functions and settings will be settable through a web interface
- The same system logos will be used for the Local Host and Central Control System so it will be clear to users that the systems are part of a whole
- Conventions used in related projects (HPSDR, GNUradio, etc.) should be used so users familiar with those systems will have a reduced learning curve

### 2.2.2 A User-friendly Interface

- The user should not have to remember things the computer knows. For example, whenever possible, choices should be shown as a drop-down list (rather than requiring user to key in).
- When the user attempts any action, the system must advise the user whether the action was successful or not, and if not, attempt to provide some useful information for troubleshooting.
- Intentionality: the user should take explicit action to apply or discard changes they make; avoid situations where the user can make a major update and then cannot change their mind about it before leaving the page. The user should have to hit "Save" or "Discard changes."

## 3. MAJOR FUNCTIONALITIES

### 3.1. CONFIGURATION AND SETUP

The Local Control System (as distributed and burned to SD card) will be preconfigured so that the simplest default configuration should be workable after simply connecting the system (assuming the user has procured the recommended hardware and connected it according to plan).

The user has two interface options:

1. Connect a HDMI-compatible monitor, keyboard and mouse directly to the SBC or

2. Use an existing PC on the same network and browser, and enter the URL of the SBC (which can be determined based on the MAC address of the SBC Ethernet interface).[??]

Once the user has a browser connected via http, they will see the welcome screen of the Local Host with default User id (UID and password (PW)).

### 3.2. STARTUP AND CONNECTING TO THE DATA ENGINE

In the initial Local Host web screen, the user will have a button to connect to the DE. When clicked, this button will trigger code which will handshake with the DE and receive DE status.

~~QQQQ NOTE: For PHASE 1:~~

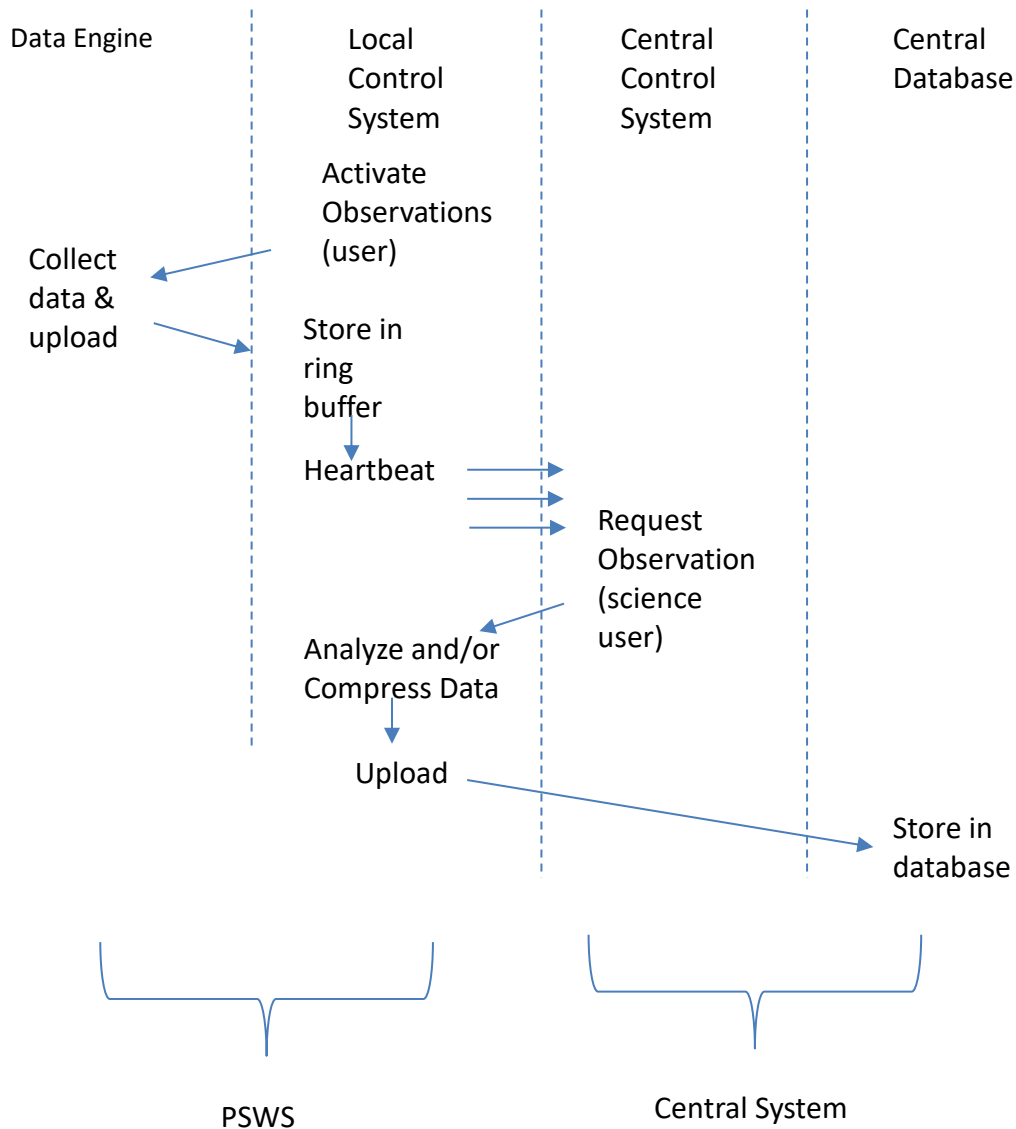
~~?? -- here we need to decide on the dDefault operation – should it be automatic monitoring of WWV on 2.5, 5, 10, 15 and 20 MHz, Use Case 3 (1-second FFT & magnetometer snapshots going to Central Control) – these should be stored directly into a table in the database to allow real-time monitoring by Central Control users. For other data collection ideas, see Future Feature Concepts section, with data going to the ring buffer?~~

Connection of DE directly to a server – details documented elsewhere QQQQ

### 3.3. CONNECTING TO THE CENTRAL CONTROL SYSTEM

The user will use a browser with Internet access to connect to the Central Control system. (This activity is independent from configuring and controlling the Local Host, and serves to let the user make themselves known to the Central system and get an authentication token; complete documentation on this is in ~~another document QQQQ~~–“Central Control System Functional Specification” and “Central Control Detailed Design.” The user will create an account, create a “station,” be issued a token for that station, and then copy their (unique) token to the Local Host of that station. This will identify the Local Host to the Central Control system. Once the token is entered, the user will have a button to click which checks in with the Central Host and advises the user if the Host connection is complete. Users who have multiple stations (under the same account) will have one unique token for each station.

### 3.4. CENTRAL REQUEST / RESPONSE / UPLOAD (USE CASE 1: DATA COLLECTION)



**Figure 2. Use Case 1. Request/Response process model.**

Refer to Figure 2.

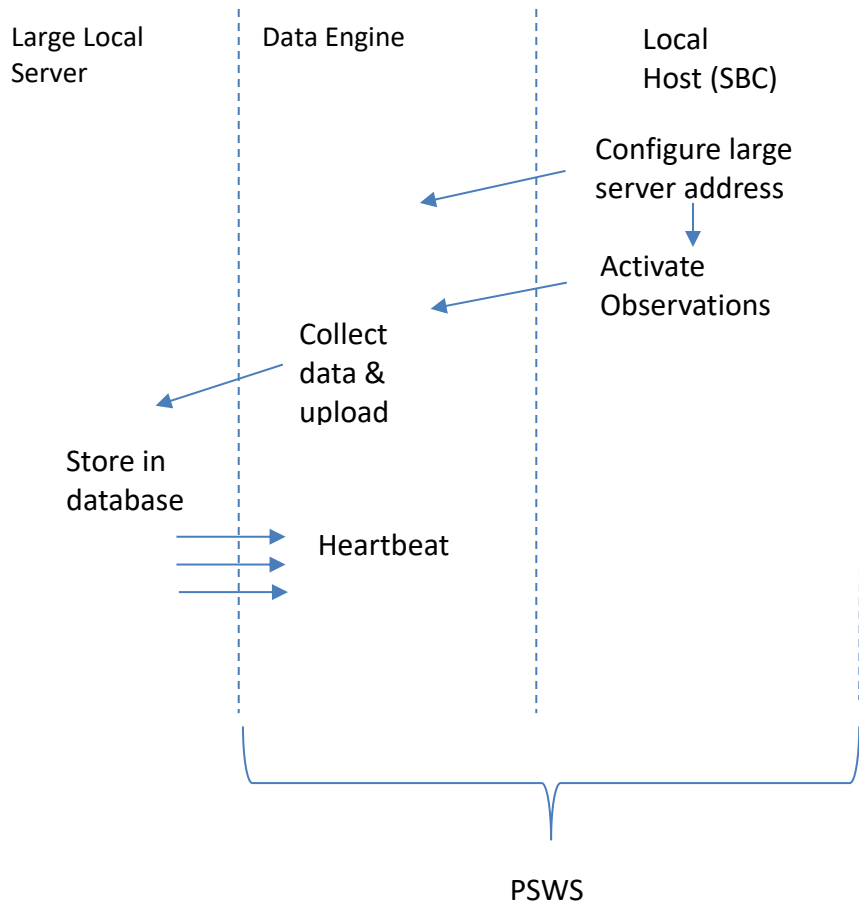
1. Once everything is configured (including user account established, token issued and entered into Local Host configuration), the user Activates Observations (thru the Local Host web interface).
2. The Local Host starts sending heartbeats to Central Control. The user can look at their profile on Central Control and see that their Local Host is connected to Central. The heartbeat includes data to tell the Central System that the Local Host is up, whether the DE is connected, whether data is being uploaded from DE to Local Host.
3. The Local Host triggers the DE to start sending observations.
4. The Local Host receives I&Q data and stores into its ring buffer.

5. A science user triggers a data collection event. Central Control sends a command to active PSWS systems. System stability note: this can trigger heavy system loading – privileges on this to be tightly restricted
6. The Local Host selects the data as requested from the ring buffer, compresses and uploads.
7. Uploads must be able to start again from where they left off in case of interruption.

#### **3.4.1. Heartbeat**

When the Local Host is connected to Central Control, it will send a status message to Central every ~~–120XX –seconds~~ss QQQQ. The status message will include data on the Local Host's connection to the Tangerine, whether the Tangerine is actively uploading spectrum data, data availability in the ring buffer, etc. (Documented separately – the status of a user's PSWS will be reflected in real time by status information in the user's profile in Central Control).

**3.5. LARGE LOCAL SERVER DATA COLLECTION (USE CASE 2: FIREHOSE)**

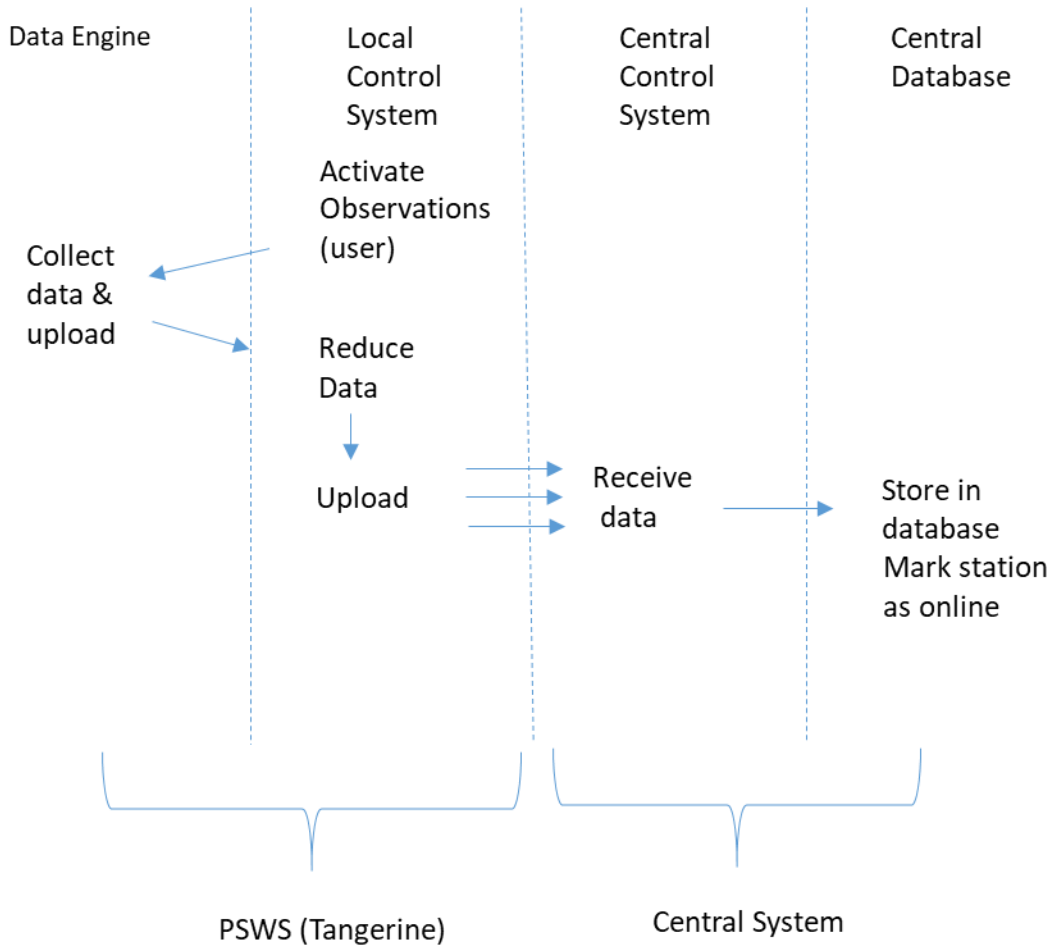


**3.5.**

*Figure 3. Use case 2, Firehose.*

1. Once everything is configured, the user enters the address of the large local server to receive data. (This is a server-to-server case, where the Tangerine acts as a spectrum data server to the large local data collection and analysis server).
2. The SBC shakes hands with the large local server.
3. The SBC sends a command to the DE including the large local server’s address and data collection parameters.
4. The DE starts sending data directly to the large local server. Note that this bypasses the SBC and no data is stored to the ring buffer.
5. The large local server heartbeat (every QQQQ msec) maintains the DE in continuous data upload mode.
6. QQQQ NOTE: there may be an additional functionality or use case where Tangerines are discoverable on the network. If so, need to determine the security aspect of that.

**3.6. LOCAL DATA REDUCTION (USE CASE 3: CONTINUOUS UPLOAD)**



**Figure 4. Use Case 3, Local data reduction + continuous upload.**

Refer to Figure 4. In this case, the user puts the system into low-speed data collection with local data reduction (using GNURadio) to extract spectrum FFT, and continuous low bandwidth data upload (typically one data package per second). (Comment: it is anticipated that such data should be stored directly into the database, rather than be stored as operating system files; this will allow rapid Central System use of the data instead of having to wait for files to close).

6. \_\_\_\_\_

**3.6.3.7. LOCAL SIGNAL PROCESSING AND USER NOTIFICATIONS**

**(Additional Local Host capabilities TENTATIVE/OPTIONAL)**

- Ability to manually run GNURadio on SBC and process signals up to capacity of SBC
- ~~Standard GNURadio function to run FFT on I&Q data from Tangerine; upload FFT data (?)~~
- Personal Propagation Monitoring: Ability to run a multi-band receiver (up to 8 bands simultaneously) to support one or more of the following
  - FT8 signal copying and reporting to PSKreporter
  - WSPR signal copying and reporting to WSPRNet
  - Watch aggregate noise level to determine when a band might be open
  - Plug-in for CW Skimmer (?)<sup>3</sup>
- ~~Ability to set SBC to copy and upload WSPR, RBN, PSK (one at a time) on up to 8 (??) bands at once (+ CW Skimmer?)~~
- Ability to set SBC to watch for a specific callsign to be copied (or perhaps a list of up to 10 callsigns) (on digital mode. CW??) to be copied (previous item) and send an email to selected email address when call is detected
- Ability to define a rule (rule set?) for user notification when a band opens (need to define what constitutes “open” and how to detect it)
- ~~Personal Propagation Monitoring: Ability to run a multi band receiver to support one or more of the following~~
  - ~~FT8 signal copying and reporting to PSKreporter~~
  - ~~WSPR signal copying and reporting to WSPRNet~~
  - ~~Watch aggregate noise level to determine when a band might be open~~
  - ~~Plug in for CW Skimmer (?)~~
- Monitoring of locally connected magnetometer (need to develop details on this)
- Ability to “throttle” upload so as to manage bandwidth consumption, especially of interest to home stations

**3.7.3.8. LOGGING**

Local Host software will optionally output logging/debugging information to a configurable location. Each major step in operation will output to the log.

**3.8.3.9. HELP**

Describe how Help will be available to user

---

<sup>3</sup> CW Skimmer is a licensed product, expected to require a special separate installation and configuration.

### 3.9.3.10. INTERFACES LOCALIZATION

~~Localization allows an application to respond to the user's location, language, and similar factors that affect how the application should respond.~~

~~Describe DE to SBC (this is WA2DFI's ICD), SBC to Central (this to be harmonized with CWRU system, working with AC8YV)~~

~~SUGGEST TO MAKE THIS A PHASE 2 REQUIREMENT~~

### 3.10.3.11. REMOTE SYSTEM UPDATES

#### Updating the Local Host

~~A local update agent will be included to allow updates to SBC software to be done in automated or semi-automated (triggered by user) manner. This applies to PSWS application modules (not to the operating system; Linux updates are to be handled via built-in Linux update functionality). An example of an open source package that provides for this can be found at <http://mender.io> . DESIGN PATTERN NOTE: optimal use of this updating technique is best achieved by building the application to be automatically updated using multiple small, isolated services.~~

#### Updating the DE

~~This refers to updating FPGA firmware. The FPGA selected for the Tangerine has a fail-safe mode to which the device can fall back to in case of a failed firmware update. A standard method for updating the regular firmware by automated or semi-automated means will be included. No special hardware will be required for re-flashing FPGA.~~

~~Need to define what system must be able to do here — e.g., possibly — on startup, handshake with Central Control; if an update package is available, download and install. (Security requirements here) — what level of control should the user have over this? This should be supported only for a standard SBC and O/S — QQQQ~~

~~Need to determine/define what type of updating (if any) the Local Host is able to do to the Data Engine — (seems like an easy way to brick the system) — reflashing of the FPGA(?)~~

### 3.11.3.12. Accessibility

~~If federally funded, there may be some requirements for use by the disabled, i.e., user interfaces to trigger narrator or other voice response, high contrast display option, etc. Check on this~~  
~~Disclaimer: this project is a proof of concept. It is not deemed to be covered by Federal Section~~



508 accessibility or other ADA compliance regulations, unless the PI is formally notified otherwise. Such compliance is out of scope for phase 1.

## 4. FUTURE FEATURE CONCEPTS

- Localization which allows an application to respond to the user's location, language, and similar factors that affect how the application should respond
- Ability to make the DE broadcast data directly to one or more recipients on network (thus bypassing SBC)
- Data collection mode: advanced ionospheric sounding mode, details TBD; It is possible it will be based on some of the Ethan Miller's CODAR receiving work, or make use of existing ionosondes and SuperDARN radar transmissions. We need to have a discussion among project scientists in regards to what is most useful and feasible here.
- Monitoring the carrier of distant FM stations to watch for pings that indicate atmospheric meteorite impacts (similar to existing effort by RMOB.org)
- Monitoring lightning strikes to look for correlations of large electrical storms and effects on propagation and ionosphere

### Comment from Ward Silver, N0AX:

Standalone single-channel instrumentation, either as a PSWS module or a separate device, such as a magnetometer, irradiation meter, event counter (Geiger counter), or other useful devices related to space weather and general ionospheric studies. I often use the apocryphal "second grade science project" as our lowest common denominator and I think there is a lot of interest in citizen science at this level. Witness the proliferation of personal weather stations (from which the PSWS concept was created) operated by interested individuals. At any rate, a series of simple instruments that plug into the PSWS either physically or virtually via a data interface somewhere in the architecture would be very useful. There is no reason to limit these instruments to geophysical / ionospheric measurements, either. Any earth science variable would benefit from a distributed monitoring network - for example, electric fields and earthquakes.

Automated signal monitoring and tracking, such as an RBN node, WSPR receiver, PSK Reporter node, etc. We are just at the beginning of what automated receivers can do and how they will integrate with both scientific research and amateur radio. This function can range from simple, single-mode decoders to wide-band spectrum capture. The upper end of this function is probably not sufficient for ionospheric research but would be targeted toward operational use and low-end event monitoring - like the Solar Eclipse QSO Party data and intruder alerts watching for OTH radar and things like that which pop up from time to time.

Noise characterization on a long-term basis beyond what the ARRL is envisioning with their Noise Study. We need to get some long-term baseline monitoring in place before wireless power transfer becomes widespread. The increasing noise floor and unintentional radiators are putting all amplitude-modulated modes under pressure, from broadcasting to point-to-point comms. Without data, we will have a hard time making a case against it.

## 4.5. TECHNICAL NOTES

Notes from discussions at Dayton, May 2019. To be incorporated into Detailed Design Specification.

- The plan is to store data into the ring buffer using the Digital \_RF HDF5 module from MIT. (Note that it still needs to be determined that this package can handle all the metadata we have in mind. )
- The DE will have a certain data packet type to transfer metadata to the SBC.
- Once a metadata variable's content is transferred to the SBC, it will be stored associated with the following signal data, until a new value for that variable is received from the DE.
- The signal data is called the payload, and is planned to be transferred from the DE to the SBC in blocks of 1024 bytes. This number is chosen to make it convenient to combine data for FFT, which requires the input data length to be a power of 2.
- Data will be transferred and handled as I & Q pairs.
- Time stamps. Each data block will contain the time stamp of the first pair of IQ samples in the payload. (Unix time) (This is metadata and is not included in the 1024 byte payload).
- Data format. ~~Tentatively, the plan is for the payload data to be in 16-bit integers (of these, 14 bits used).~~ I & Q data elements will be 32-bit floating point. QQQQ Must decide on the Endian-ness of these data. ??

### Notes from Discussions at Detroit DCC, Sep. 2019

- Need a layer of abstraction between the (browser-based) UI in the Tangerine and function processes; this will allow the use of (in unix terminal) command-line interface to control system (useful for testing, power users). Will look at interprocess communication methods for this.
- Related to the above point - The Local Host application should be implemented as a collection of small, isolated services with standardized intercommunication; this will allow for better remote updating. See section 3.11.

-----  
Remaining sections are standard Functional Spec stuff, may not be needed for this project

## 5.6. LOGICAL DATA MODEL

*[Include the Logical Data Model as an appendix.]*

See Appendix E - <Project Name> Logical Data Model.

## 6.7. REQUIREMENTS TRACEABILITY MATRIX

*[Include the Traceability Matrix as an appendix. In the Requirements Analysis phase, the matrix is populated with requirements identified in the Requirements Definition. It is a living document that should be populated with information throughout design, construction, and test phases, etc.]*

See Appendix F - <Project Name> Requirements Traceability Matrix.

### 7.8. Appendix A: Functional Specifications Definition Approval

The undersigned acknowledge that they have reviewed the **<Project Name> Requirements Definition** and agree with the information presented within this document. Changes to this **Requirements Definition** will be coordinated with, and approved by, the undersigned, or their designated representatives.

*[List the individuals whose signatures are desired. Examples of such individuals are Business Owner, Project Manager (if identified), and any appropriate stakeholders. Add additional lines for signature as necessary.]*

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_  
Title: \_\_\_\_\_  
Role: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_  
Title: \_\_\_\_\_  
Role: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Print Name: \_\_\_\_\_  
Title: \_\_\_\_\_  
Role: \_\_\_\_\_

## APPENDIX B: REFERENCES

The following table summarizes the documents referenced in this document.

Document Name	Description	Location
<a href="#">Tangerine SDR Requirements V0.3.pdf</a>	<a href="#">System requirements</a>	<a href="https://tangerinesdr.com/TangerineSDR_documents/">https://tangerinesdr.com/TangerineSDR_documents/</a>

## APPENDIX C: Business Process Model

The Business Process Model is attached as a separate document. (Master Flow diagram to be included and/or flow charts from EFOTM).

## APPENDIX D: Logical Data Model

The Logical Data Model is attached as a separate document.

## APPENDIX E: Requirements Traceability Matrix

*The Requirements Traceability Matrix is attached as a separate document.* (Refer to following spreadsheets):

1. Combined Use Case List for Intervention Release 1-8-10.xls
2. MCA - Intervention Business Requirements and Rules 12-29-09.xls