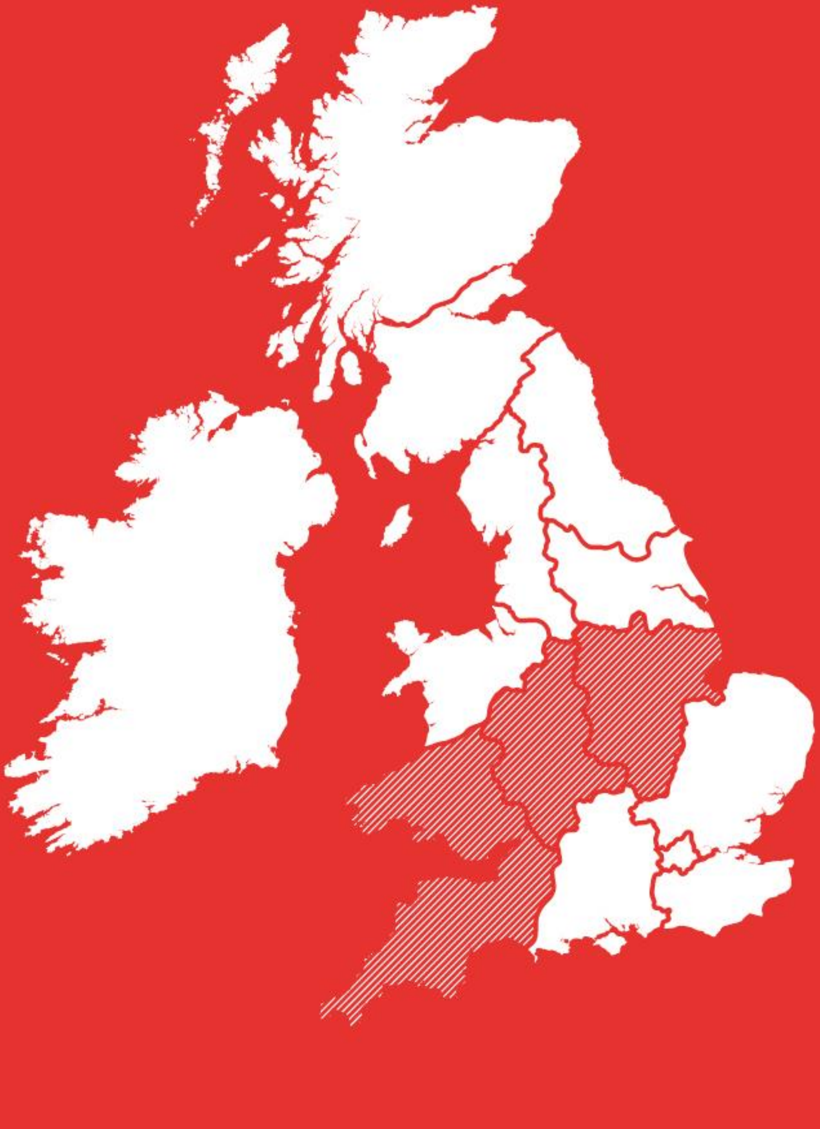
WESTERN POWER DISTRIBUTION

OPEN LV



OPENING UP THE SMART GRID

OPENLV SDRC 1

DETAILED DESIGN OF THE OVERALL OPENLV SOLUTION

ea technology

RIIO NIC NETWORK INNOVATION COMPETITION

Report Title	:	SDRC 1 - Detailed design of the overall OpenLV solution
Report Status	:	FINAL
Project Ref	:	WPD/EN/NIC/02 - OpenLV
Date	:	17.10.17

Document Control

	Name	Date
Prepared by:	Richard Potter	17.10.17
Reviewed by:	D.Hollingworth/M.Dale	17.10.17
Recommended by:	D.Roberts/R.Hey	17.10.17
Approved (WPD):	A.Sleightholm	17.10.17

Revision History

Date	Issue	Status
06.10.17	1.0	For WPD Review
17.10.17	1.1	FINAL

Contents

DISCLAIMER

Neither WPD, nor any person acting on its behalf, makes any warranty, express or implied, with respect to the use of any information, method or process disclosed in this document or that such use may not infringe the rights of any third party or assumes any liabilities with respect to the use of, or for damage resulting in any way from the use of, any information, apparatus, method or process disclosed in the document. © Western Power Distribution 2017

Telephone +44 (0) 1332 827446. E-mail wpdinnovation@westernpower.co.uk

Glossary

Term	Definition
ACL	Access Control List
API	Application Programming Interface
APN	Access Point Name
App	Software designed to run on smartphones and other mobile device
DSO	Distribution System Operator
DTR	Dynamic Thermal Rating
LV	Low Voltage
LV-CAP™	LV-CAP™ (Low Voltage Common Application Platform) is a hardware agnostic environment that in and of itself does not provide any direct functionality but instead makes available the environment in which various 'Apps', each designed to provide specific a network benefits (or benefits) can be deployed.
LV Feeder	The outgoing supply from the distribution substation.
LV Substation	LV substations step the electricity supply down from 11kV to 230 / 400V and distribute it along connected feeders.
Network Meshing	Joining adjacent normally separated networks by closing a Normally Open Point.
Normally Open Point	A Normally Open Point is generally located between two feeders, connected to different distribution substations. It allows the Distribution Network Operator (DNO) to reconfigure the network through closing the point to join the two networks together.
NIC	Network Innovation Competition
MQTT	MQ Telemetry Transport
Substation	A point on the network where voltage transformation occurs.
Successful Delivery Reward Criteria	The Project specific criteria set out in the Project Direction against which the Project will be judged for the Successful Delivery Reward.
Transformer	Device that changes the voltage of an a.c. current, without changing the frequency.
WPD	Western Power Distribution

1 Document Purpose

The OpenLV Project is funded by the Network Innovation Competition (NIC). The requirements for key project deliverables, as part of NIC Governance, are defined as Successful Delivery Reward Criteria (SDRC) and each SDRC has associated evidence criteria as defined in the Project Direction [Ref. 1].

It is confirmed that the SDRC and associated evidence requirements have been met and this is supported by the compliance matrix provided below:

Category	Detail	Criterion Met	Section(s)
Successful Delivery Reward Criterion	Detailed systems architecture	✓	Section 2
Successful Delivery Reward Criterion	Requirements specification for the OpenLV intelligent substation hardware	✓	Section 3.4 & Annex 2
Successful Delivery Reward Criterion	An assessment of the development of the intelligent substation control software to identify whether any changes are required to the planned deployment for the OpenLV project	✓	Annex 1
Successful Delivery Reward Criterion	Detail the approach for testing the overall solution ahead of wide scale deployment	✓	Section 3.3
Successful Delivery Reward Criterion	Factory and site acceptance test documentation	✓	Section 3.5 & Annex 3
Successful Delivery Reward Criterion	Factory testing results	✓	Section 3.5 & Annex 3
Evidence	the specification for the OpenLV solution	✓	Section 2 & Annex 2
Evidence	FAT and SAT documentation	✓	Annex 3 & Annex 4
Evidence	FAT test results	✓	Section 3.5 & Annex 3

2 Executive Summary

In this SDRC report we present the Specification, Design and Factory Testing results of the overall OpenLV solution.

Great Britain has about 1,000,000 Low Voltage (LV) feeders; these have largely been designed and operated on a fit-and-forget basis for the last 100 years, but things are set to change. The LV networks are expected to see radical change as we, as customers, alter our behaviour and requirements stemming from the vehicles we drive, to the generation and storage devices we put onto and into our homes.

The technology to be trialled as part of the OpenLV Project provides a new, open and flexible solution that will not only provide the Distribution Network Operator (DNO), Community Groups and the Wider Industry with data from the LV network; but will also enable these groups to develop and deploy apps within LV substations. The OpenLV Project is seeking to prove the technology and assess how the provision of LV network data and ability to develop and deploy apps can provide benefits to the DNO, Community Groups and the Wider Industry (See Figure 1).

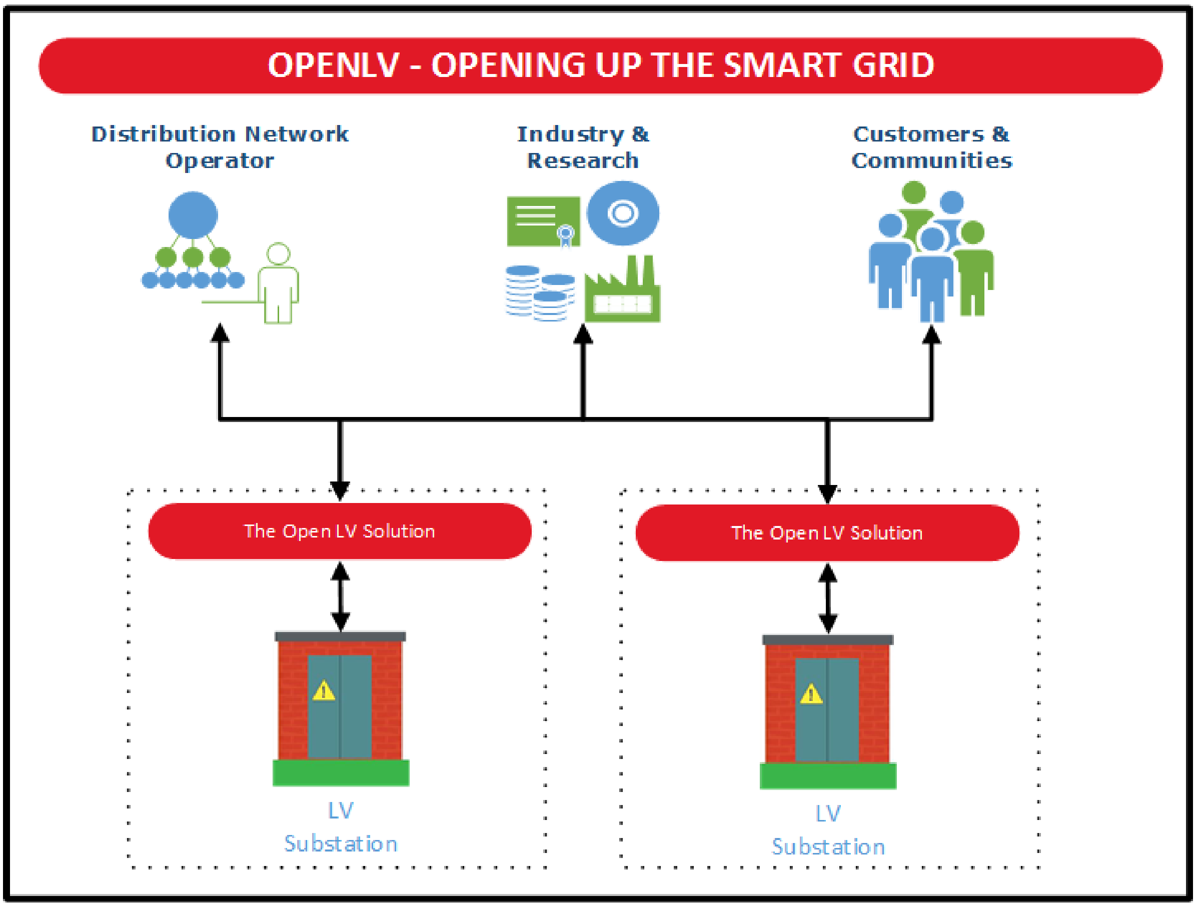


Figure 1: Overall OpenLV Solution

This report has been structured to meet the SDRC evidence criterion outlined in the OpenLV Project Direction [Ref. 1]. Chapter Two, Systems Architecture, outlines the design concepts for the overall OpenLV solution, the software components, the hardware components and associated central infrastructure and communications components.

Chapter Three, Building & Testing the OpenLV Solution, outlines the approach taken to build and test the OpenLV solution, outlines how the key requirements were captured and the results of the Factory Acceptance Tests (FAT) completed to date. It is confirmed that the OpenLV solution has passed the first two stages of FAT and that preparations are now being made to install the first 4 test units on WPD’s LV network.

Chapter Four, Key Learning Points, outlines the key learning points recorded at this stage of the Project in relation to the specification, design, build and testing of the overall OpenLV solution.

Chapters Two and Three are supported by document annexes including: 2) the OpenLV Solution Requirements Specification, 3) the Factory Acceptance Test documentation along with the results of the tests completed at FAT and 4) the proposed Site Acceptance Tests (SAT) documentation.

The approach taken to building and testing the OpenLV solution is robust and is on track to enable initial installation of the first 4 devices in LV substations in 2017 with full roll out of 76 further devices in 2018 to support the trials for all 3 methods:

1. The network capacity uplift trials;
2. The community engagement trials; and
3. The OpenLV extensibility trials.

3 Systems Architecture

3.1 High Level Overview

The high-level architecture of the OpenLV solution is shown in Figure 2.

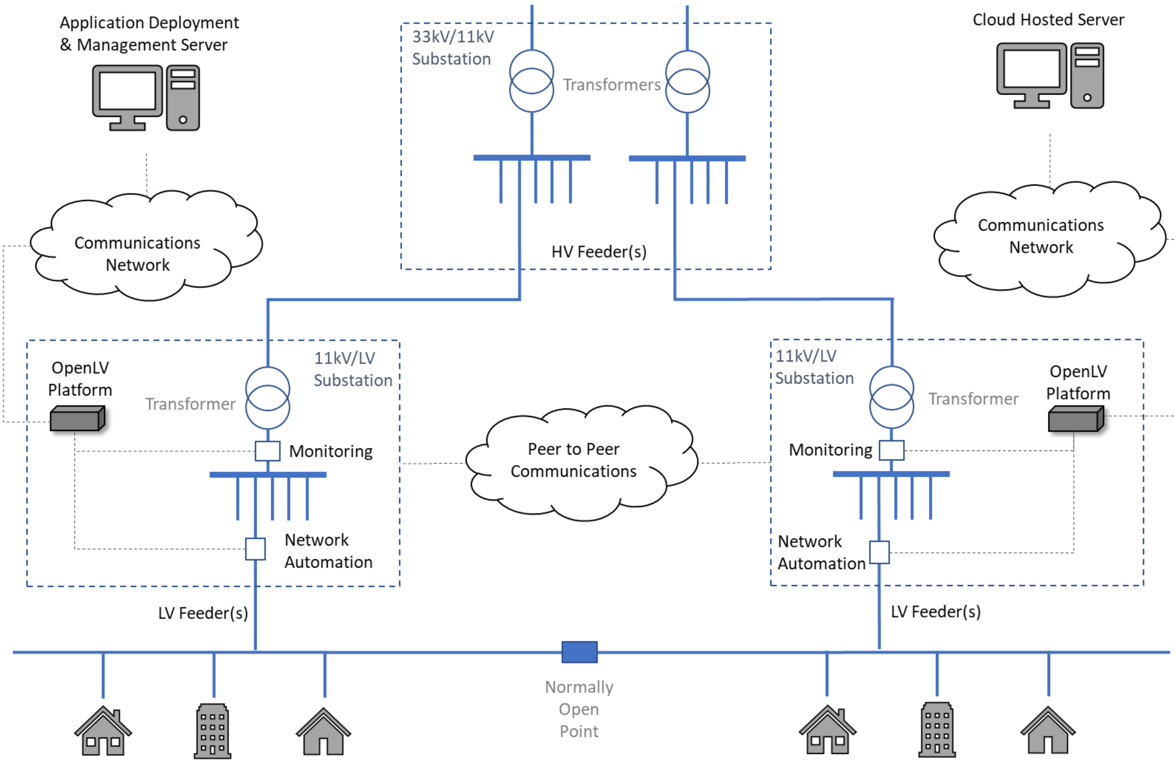


Figure 2: High Level Architecture

The key components of the solution are as follows:

- **LV Network Automation:** These devices enable automated meshing of the LV network via an app or app(s) installed on the platform.
- **LV Monitoring Equipment:** This monitoring equipment utilises sensors to take electrical measurements from the LV busbar, the transformer and the outgoing feeder(s). In addition, temperature measurements are also taken from the transformer, and inside and outside substation(s). The monitoring equipment provides this LV monitoring data to the OpenLV Platform.
- **OpenLV Platform:** Consists of a ruggedised PC with a Linux based operating system running the Low Voltage-Common Application Platform (LV-CAP™). This platform receives, stores and processes data from external LV monitoring equipment. These devices have sufficient computational power to store and run multiple apps and can provide relevant information out via a communications link to centralised server(s).

- **Application Deployment & Management Server:** Enables management of the OpenLV Platform(s) that will be installed as part of the project. This includes the deployment of app(s) to devices in the field. It will also be utilised to store relevant data to enable the OpenLV trials to be assessed.
- **Cloud Hosted Server:** Enables LV monitoring data to be collected, stored, shared and visualised to provide benefits to communities and the wider industry.

3.2 Design Concepts

The solution has been designed to be "open" in that it enables any individual or company to develop apps to be deployed on the platform. The OpenLV Platform is analogous to a smartphone. The LV-CAP™ software is analogous to an open operating system, for example Android, running on a smartphone.

The following key points should be noted regarding the systems architecture of the overall OpenLV solution. The solution, for the Project, has been designed to:

- Incorporate the LV-CAP™ platform;
- Interface to LV monitoring equipment to collect and share LV network data;
- Conform to a de-centralised systems architecture. This means that intelligence is built into the hardware that is installed in LV substations;
- Not to integrate with the Distribution Network Operator (DNO) Distribution Management System (DMS). This limits complexity of the Project trials;
- Provide data back to multiple vendor(s) back office systems; and
- Enable anyone to develop an app to run on the hardware installed in LV substations. Therefore, providing benefits to DNOs, community groups and the wider industry.

The core aim of the project is to prove the open nature of the platform through three core Methods:

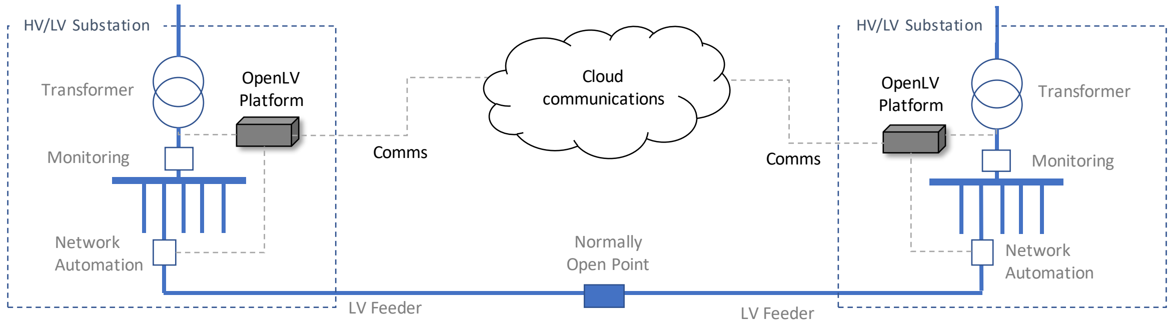
1. Network Capacity Uplift;
2. Community Engagement; and
3. OpenLV Extensibility.

These Methods are outlined in the following sub-sections of this report.

3.2.1 Method 1: Network Capacity Uplift

Figure 3 provides an overview of the systems architecture that will be deployed to complete Project trials for Method 1 – Network Capacity Uplift.

As part of the Project trials for Method 1 apps will be used to increase the capacity of existing LV assets through the application and implementation of Dynamic Thermal Rating of the LV Transformer and through meshing LV Feeder(s) on the LV network.



What

- Check network capacity against thermal rating of transformer; when breached, close two radial circuits to mesh the LV network
- Deploy two proven techniques
  - ‘Dynamic Thermal Ratings App’ and
  - ‘Network Meshing App’.
- Together with a ‘Network Control App’ to operate/configure the network

How

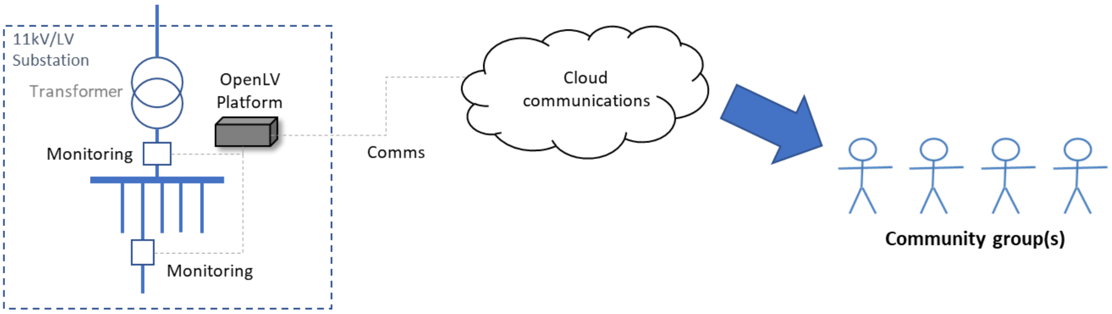
- Assess WPD’s network to identify candidate circuits
- Deploy LV-CAP™ to 60 substations
- Monitor how the solution would operate over several months
- Install actuators on 5 circuits (2 ends each) to prove end-to-end control
- Assess and report on performance

Figure 3: Method 1 – Network Capacity Uplift

3.2.2 Method 2: Community Engagement

Figure 4 provides an overview of the systems architecture that will be deployed to complete Project trials for Method 2 – Community Engagement.

As part of the Project trials for Method 2, Community Groups will, either make use of the LV network data provided by the OpenLV Platform, and/or develop and deploy apps to provide benefits to individual Communities.



What

- To work with key community groups to understand whether Apps can be developed and installed on the platform
- Identify funding sources that customers / communities can use to develop specific Apps

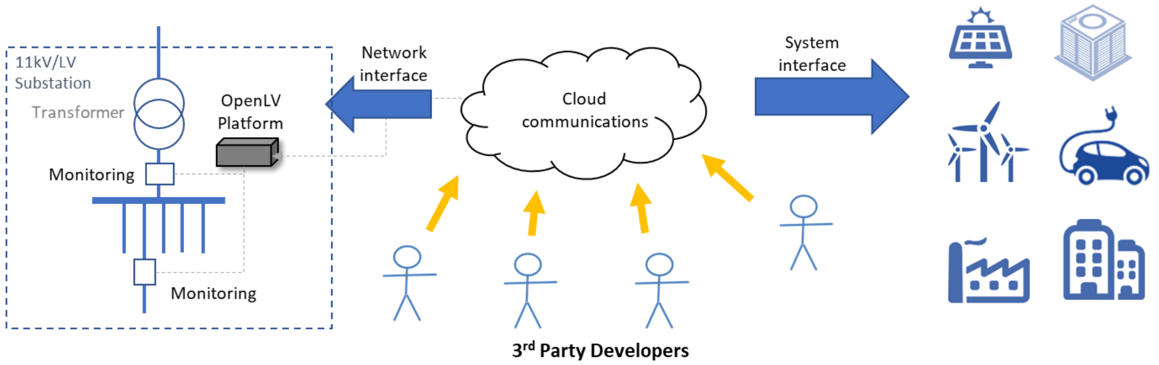
How

- Community engagement to promote availability of platform / LV network data
- Make available 10 LV-CAP™ units for deployment
- Provide communities access to relevant data
- Funding to develop specific Apps to be raised outside of the project budget, e.g. public funding / private sector

Figure 4: Method 2 – Community Engagement

3.2.3 Method 3: OpenLV Extensibility

Figure 5 provides an overview of the systems architecture that will be deployed to complete Project trials for Method 3 – OpenLV Extensibility. As part of the Project trials for Method 3, the Wider Industry will either, make use of the LV network data provided by the OpenLV Platform, and/or develop and deploy ‘apps’ to provide benefits to: DSOs, Platform Providers, 3rd Party Developers and Customers.



What

- To enable companies to develop innovative algorithms and applications for either the DNO, or it’s customers

How

- Publicise the opportunity to 3rd parties
- Make available standard App ‘container’ for third parties to use for their development
- Make available 10 LV-CAP™ devices for substation deployment
- Funding to develop specific Apps to be raised outside of the project budget, e.g. private sector

Figure 5: Method 3 – OpenLV Extensibility

3.3 Software Components

3.3.1 High Level Software Flow

The LV-CAP™ is a hardware agnostic operating system that enables cost effective deployment of smart grid products from multiple suppliers on a single set of hardware, the OpenLV Platform, as shown in Figure 6.

Apps can be developed by multiple manufacturers and generate bespoke datasets and/or control various unrelated network assets without any application being influenced or affected by another, although outputs can be shared. For example, data can be transmitted back to multiple vendor back office systems.

The apps that are being deployed have been coded in C++, Go and Java. It is also expected that Python will also be utilised for further apps.

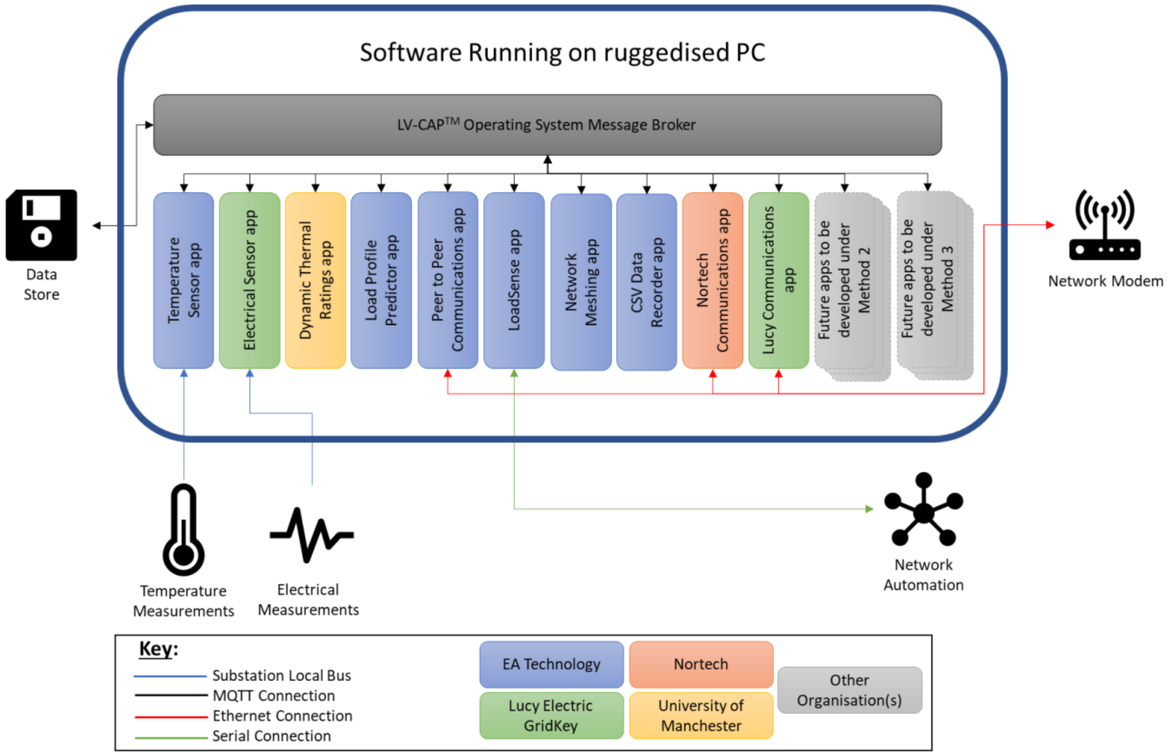


Figure 6: OpenLV Software Stack

3.3.2 Apps to be deployed

The solution has been designed to enable the deployment of apps developed by multiple companies. The apps to be deployed are as follows:

- **EA Technology:** The following apps developed by EA Technology will be deployed:
    - A 'Temperature Sensor' app to receive and process temperature measurement points from the transformer and within and outside the substation.
    - A 'Load Profile Predictor' app to assess historic load and predict the load profile of the LV transformer in the future.
    - A 'Peer to Peer Communications' app to enable load and capacity forecasts to be shared between 'adjacent' LV substations.
    - A 'LoadSense' app that, directly or indirectly, utilises the end outputs of the above elements to inform the decision of whether to mesh or de-mesh the network, or to leave it in the current state.
    - A 'Network Meshing' app to respond to the commands from the LoadSense app and sends commands to the automated network circuit breakers.
    - A 'CSV Data Recorder' app to enable storage of all data captured by the system, information generated by any applications and a record of any actions implemented.
- **University of Manchester:** A 'Dynamic Thermal Ratings' app utilising underlying IP owned by the University of Manchester will be deployed. This app enables dynamic rating rather than static rating of the LV transformer enabling additional capacity to be made available on the LV network.
- **Nortech:** A 'Nortech Communications' app that enables communications to the application deployment & management server will be deployed.
- **Lucy Electric GridKey:** The following apps developed by Lucy Electric GridKey will be deployed:
    - An 'Electrical Sensor' app that enables LV monitoring data to be transmitted from the LV monitoring hardware to the OpenLV Platform.
    - A 'Lucy Electric Gridkey Communications' app that enables communications to a server hosted in the cloud.

In addition to the apps listed above the Project will seek 3rd parties to develop new apps to be deployed as part of the trials to support community energy schemes and the wider industry (Methods 2 and 3).

3.4 Hardware Components

The following sub-sections provide technical information regarding the core hardware components.

3.4.1 LV Network Automation

EA Technology's ALVIN Reclose™ devices (see Figure 7) will be utilised to provide the 'network meshing' functionality. The Network Meshing app will be deployed on the LV-CAP™ platform allowing control of individual connected ALVIN Reclose™ devices. ALVIN Reclose™ devices can be deployed in place of 315A and 400A fuses on the LV network and operate automatically to protect the network in the event of a fault.

The devices monitor the voltage on either side of the LV fuse board and current passing through them, and can relay this information back to the LV-CAP™ platform. The LV-CAP™ platform is also able to control the relay within the ALVIN Reclose™ devices, enabling autonomous reconfiguration of the network.

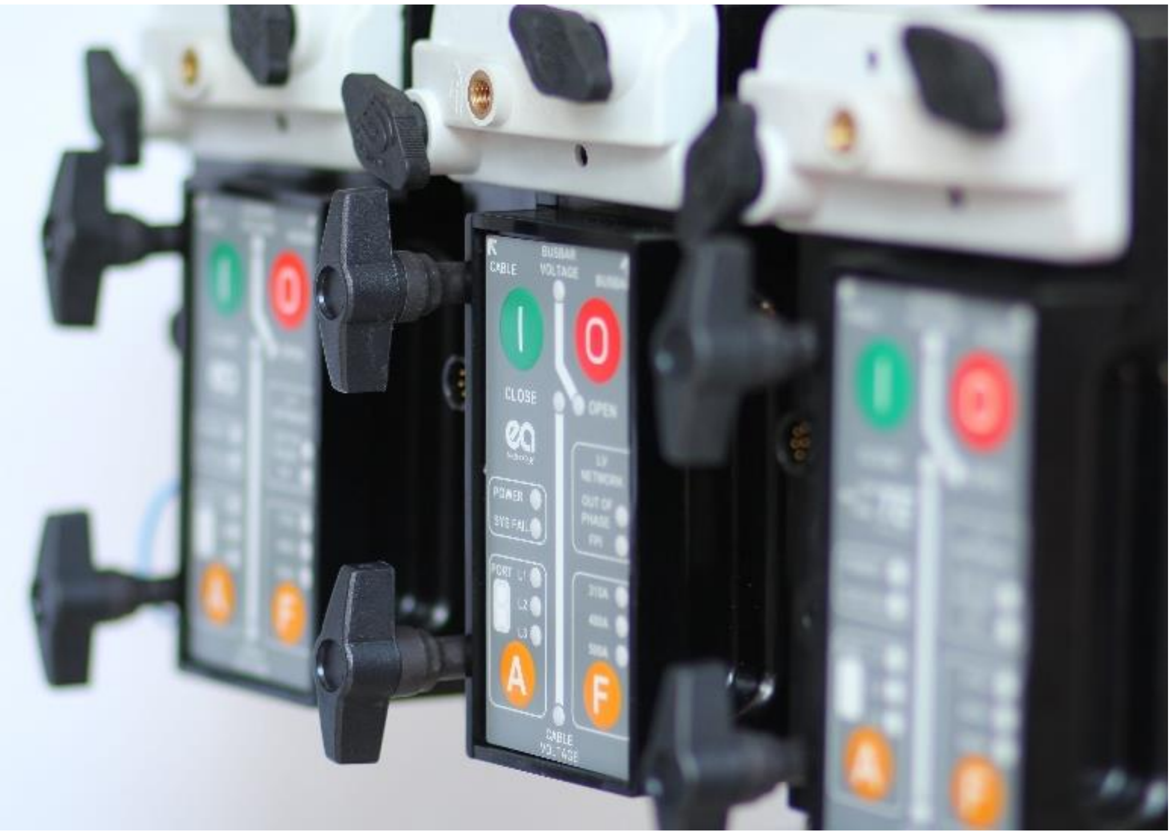


Figure 7: ALVIN Reclose™ devices

3.4.2 LV Monitoring Hardware

The LV monitoring equipment, installed in each LV substation, consists of a Lucy GridKey MCU520 LV monitoring system (see Figure 8). This device utilises sensors to measure the Voltage and Current and pass this data on to the OpenLV Platform through an Ethernet connection.

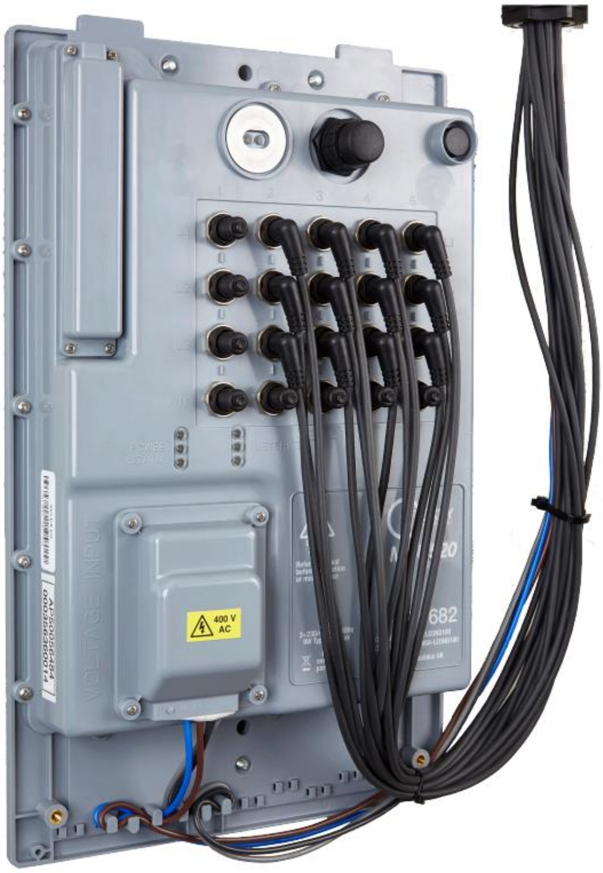


Figure 8: GridKey MCU520

3.4.3 OpenLV Platform

The OpenLV Platform consists of the Following:

- **Temperature Sensors:** Temperature measurements of the connected transformer is achieved through the use of an Ethernet to 8 Channel Isolated Thermocouple Input / 8 Channel Digital Output Module with Modbus TCP. In addition, temperature sensors will be deployed to provide temperature both inside and outside the substation (see Figure 9). The temperature data will be utilised by the DTR app and will also be made available to potential apps to be developed as part of Methods 2 and 3.
- **Intelligent Substation Devices:** The Intelligent Substation Devices consist of a ruggedised PC with a Linux based operating system running the LV-CAP™ software platform. These devices have sufficient computational power to run the LV-CAP™ operating system and multiple apps. These devices provide the capability to receive, store and process data being gathered by the monitoring sensors, and relay all, or part of the information back to a centralised location if required (see Figure 10 and Figure 11).
- **GSM Modem:** The GSM modem provides the LV-CAP™ platform access to the management and data servers via a dedicated private mobile network. Through this, the platform is able to receive over-the-air updates, signal alerts and transmit data and processed application outputs to designated storage servers (see Figure 10).

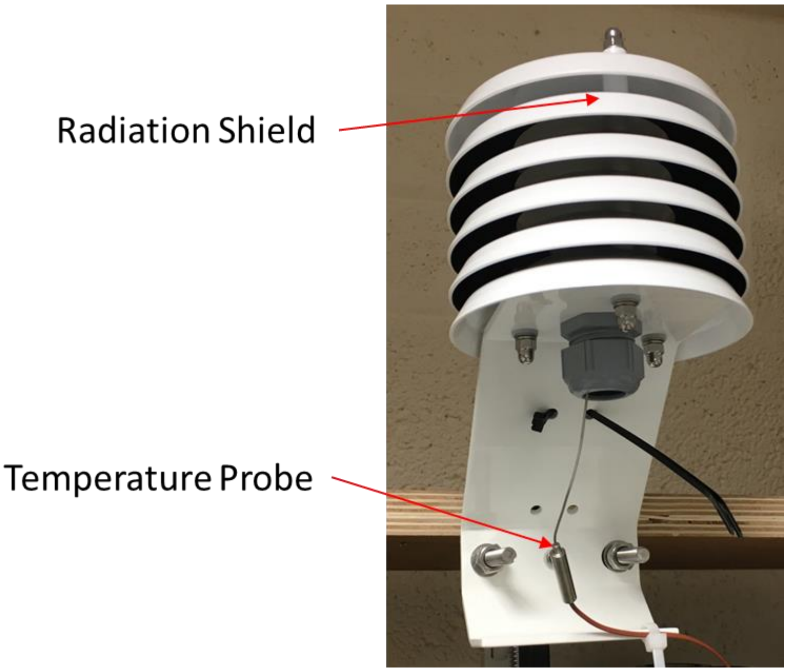


Figure 9: Temperature Sensor

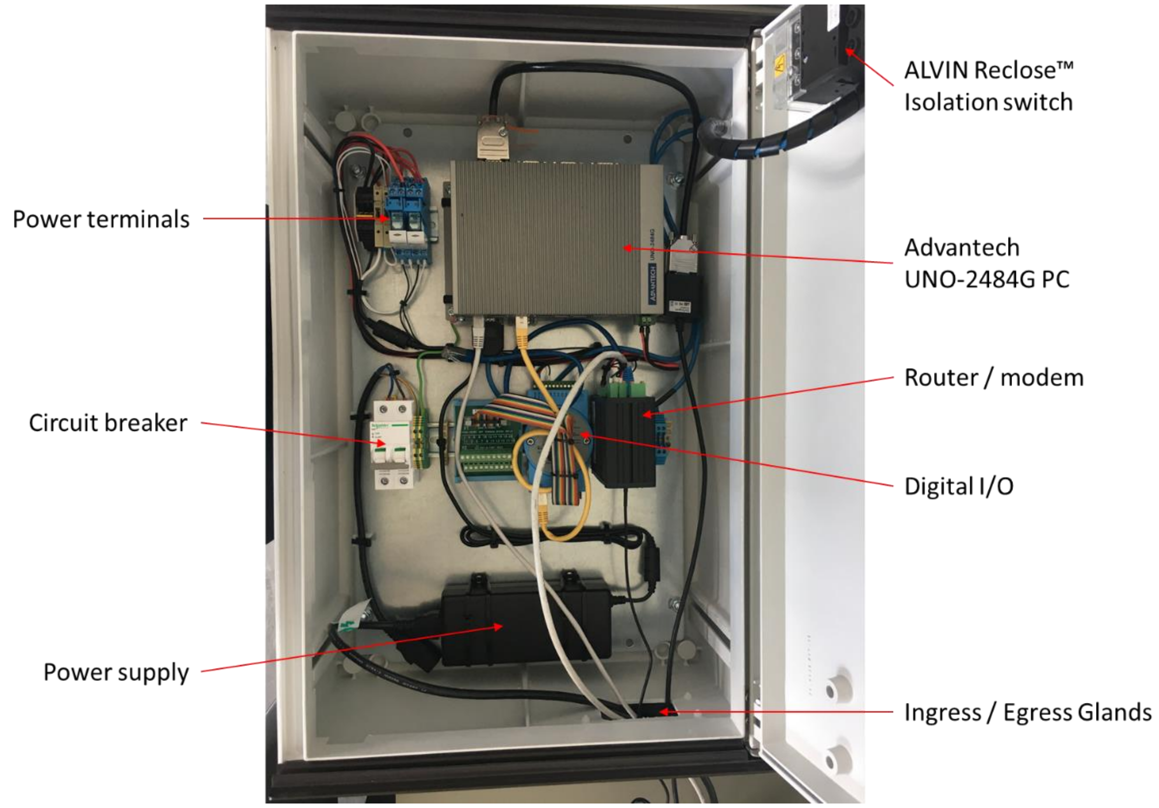


Figure 10: OpenLV Platform (inside)

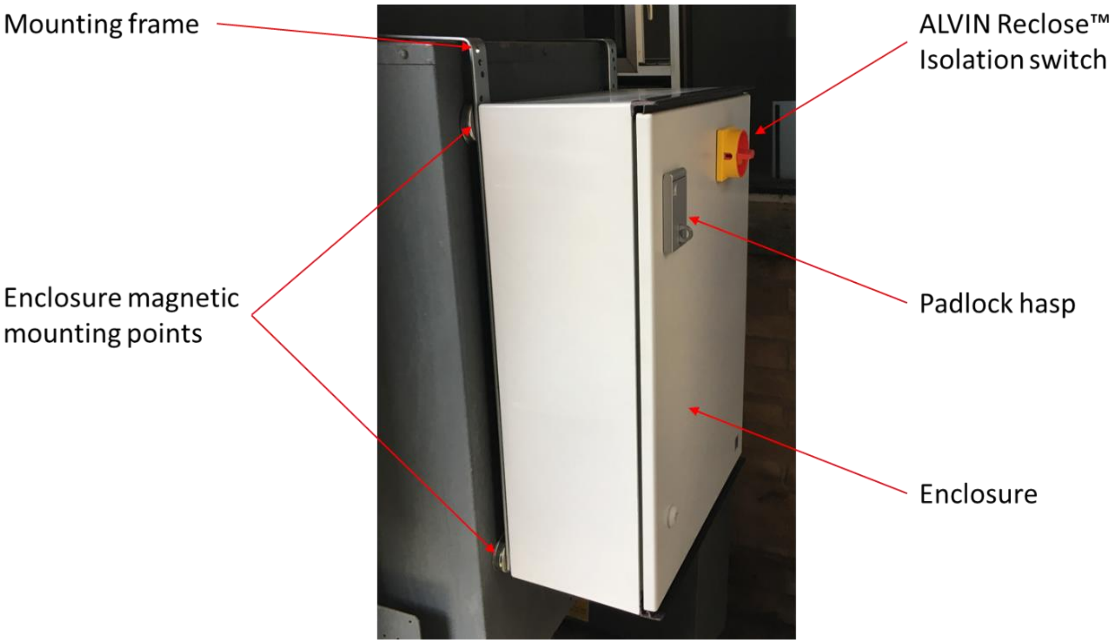

Mounting frame
ALVIN Reclose™ Isolation switch
Enclosure magnetic mounting points
Padlock hasp
Enclosure

Figure 11: OpenLV Platform (outside)

3.4.4 Application Deployment & Management Server

The Application Deployment and Management Server is an iHost server (see Figure 12), provided by Nortech and provides the LV-CAP™ platform management services as well as receiving the data gathered by all sensors connected to the deployed devices. This provides a secure, reliable central host platform, receiving and storing data from any number of remote sites using a variety of communication channels and, as part of the OpenLV solution the iHost system will enable the deployment and management of apps on the Intelligent Substation Devices and will store data to support the Project trials for Method 1 – Network Capacity Uplift.

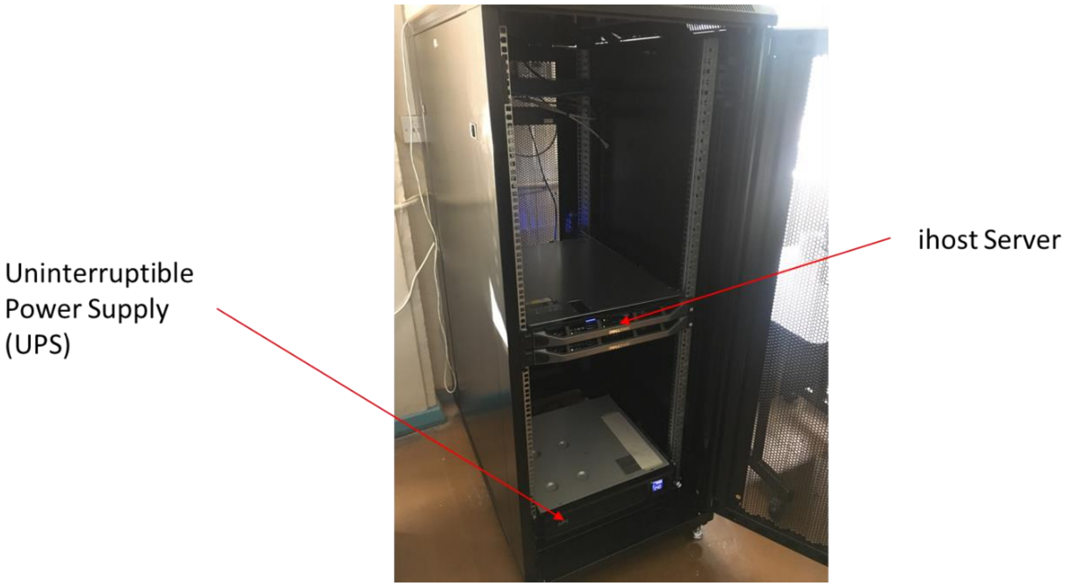


Figure 12: Nortech iHost Server & UPS

3.4.5 Cloud Hosted Server

This Server is a cloud based data storage and processing platform, provided by Lucy Electric GridKey. This provides a secure, reliable central host platform, receiving, and storing data from any number of remote sites using a variety of communication channels and protocols. As part of the OpenLV solution Lucy Electric's cloud storage system will store data to support the Project trials for Method 2 – Community Engagement and Method 3 – OpenLV Extensibility, providing community groups, academia and third-party companies access to data gathered by the deployed platforms. The platform will be separated, both physically and via firewalls, from Lucy Electric's BAU data provision systems.

3.5 Communications

The wide area communications links for the project will be provided over 3G / 4G mobile data networks. A dedicated private Access Point Name (APN) will be set up for the project trials, supporting roaming between three of the four UK Mobile Operators. Using a dedicated APN separates the project equipment from other mobile network users and provides secure communications between the OpenLV Platforms deployed in LV substations and both the Application Deployment & Management Server and Cloud Hosted Server.

4 Building & Testing the OpenLV Solution

4.1 Background

The core element of the OpenLV solution is the LV-CAP™ environment. The development of the LV-CAP™ was joint-funded by InnovateUK under the project name "Common Application Platform for Low Voltage Network Management".

This project involved the formation of a SME-led collaboration between EA Technology, Nortech and the University of Manchester to develop a novel, common, low cost, robust monitoring and management system for the LV network. As part of this project LV-CAP™ was tested in a laboratory environment.

The OpenLV project builds on this work, taking the platform out of the laboratory and into a real-world controlled trial, with a total of 80 devices to be installed as part of the Project trials. Information on the changes made to the LV-CAP™ environment, for implementation on the OpenLV project, is provided in Annex 1.

4.2 Requirements Specification

The OpenLV Requirements Specification is provided in Annex 2. This document provides a record of the requirements for the overall OpenLV solution that will be utilised to support Project trials for the three Methods outlined in the Full Submission Process (FSP) [Ref. 3].

4.2.1 Capturing & Prioritising Requirements

In order to define the requirements for the overall OpenLV Solution, the key hardware and software components were defined, and then the requirements for each component were identified.

In order to prioritise requirements, the MoSCoW approach was utilised. This approach is a well-known technique used in business analysis and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement.

Each requirement has been identified and prioritised using the MoSCoW approach, that stands for Must, Should, Could and Will not:

- M – Must have this requirement to meet the business needs;
- S – Should have this requirement if possible, but project success does not rely on it;
- C – Could have this requirement if it does not affect anything else in the project; and
- W – Will not deliver this requirement at the current time, but it could be delivered at a later date.

In addition to the above requirements the OpenLV Requirements Specification also includes a number of Information Statements that provide relevant context.

4.3 Building the OpenLV solution

The overall approach for building and implementing the OpenLV solution is outlined in Figure 13. The green boxes represent stages that are complete, the orange boxes represent stages that are currently being completed and the grey boxes represent stage(s) that have not started yet.

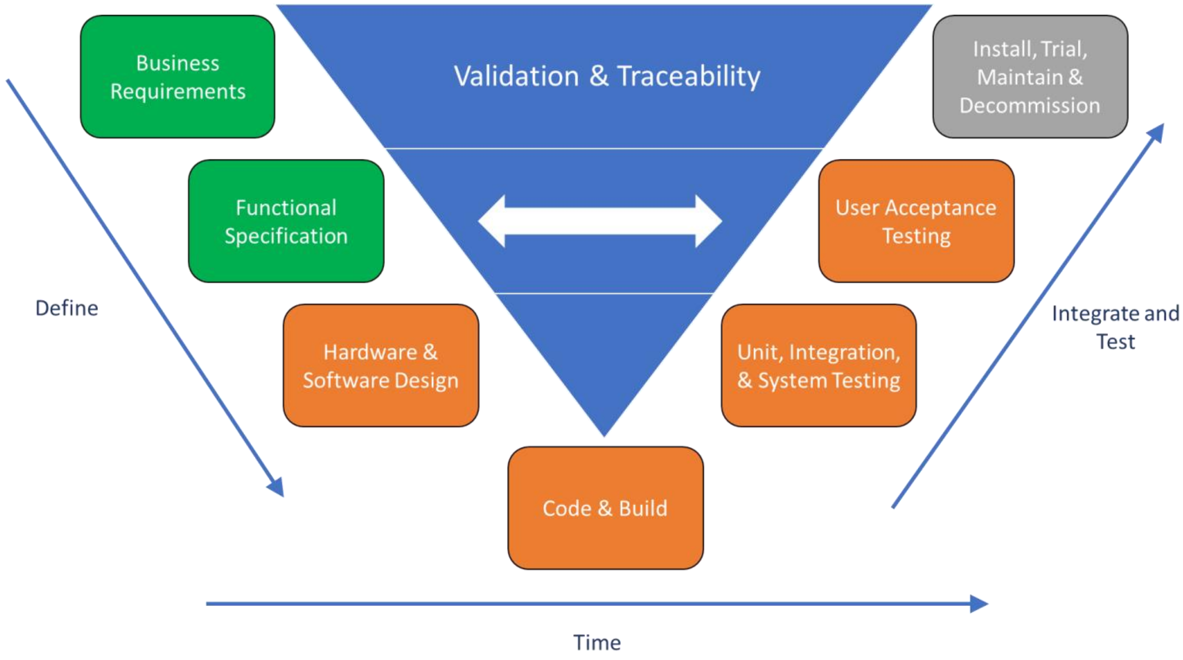

Business Requirements
Validation & Traceability
Install, Trial, Maintain & Decommission
Functional Specification
User Acceptance Testing
Define
Integrate and Test
Hardware & Software Design
Unit, Integration, & System Testing
Code & Build
Time

Figure 13: Building the OpenLV solution

An overview of each stage and associated key documentation are outlined below:

- **Business Requirements:** In this stage, the business requirements were defined. The business requirements for the OpenLV solution are outlined in the Initial Screening Process (ISP) and FSP documents [Ref. 2 and Ref. 3]. These documents were completed in line with NIC Governance Document [Ref. 4]. This stage is complete.
- **Functional Specification:** In this stage, the requirements for the overall solution were defined. Requirements are documented in the OpenLV Requirements Specification. This document is provided in Annex 2. This stage is complete.
- **Hardware & Software Design:** This stage entailed the hardware design for the equipment to be installed in LV substations and the software design for each of the software components that runs on the OpenLV Platform. This stage is currently being completed.
- **Code & Build:** This stage includes coding the individual software components and building the Intelligent Substation Devices to be installed in LV substations. This stage is currently being completed.

- **Unit Integration & System Testing:** This stage includes testing individual hardware and software components and also testing the overall system. System testing ensures that the individual hardware and software components work together. This stage is currently being completed.
- **User Acceptance Testing:** This stage includes testing the overall solution. The key tests that are completed in this stage are Factory Acceptance Tests (FAT), that are completed in a laboratory environment and Site Acceptance Tests (SAT), that are completed on equipment installed in the field. In both cases the tests defined in the test documents are traceable, in that they link back directly to each requirement. In both cases the tests are formally signed off by EA Technology and WPD representatives. This stage is currently being completed.
- **Install, Trial, Maintain & Decommission:** This stage includes the installation of 80 OpenLV Platforms in WPD's licence area(s), trials to prove the overall solution, maintenance of the devices and finally de-commissioning the devices at the end of the project. This stage has not started.

4.4 Approach to Testing

Testing the overall OpenLV solution has been completed through: 1) Unit Integration & System Testing and 2) User Acceptance Testing.

A staged approach has been taken to the development and testing of the overall OpenLV solution. This has enabled the project team to focus on building and deploying the core solution and then adding additional functionality.

As a result, the approach to testing has been to hold multiple formal acceptance testing FATs and multiple SATs have also been scheduled. The core components tested at each stage are outlined in the core component test matrix sub-section (below) and the dates for completed and scheduled acceptance testing are outlined in the timeline sub-section (below).

4.4.1 Core Component Test Matrix

The core components tested or scheduled to be tested at each formal test stage are shown in Table 1.

Component	Category	FAT 1 OpenLV Core System	FAT 2 OpenLV Network Meshing & Cloud Hosted Server	FAT3 OpenLV LoadSense & DTR app tests	SAT 1 OpenLV Core System	SAT 2 OpenLV Full Solution
LV Network Automation Hardware	Hardware	No	Yes	Yes	No	Yes
LV Monitoring Hardware	Hardware	Yes	Yes	Yes	Yes	Yes
OpenLV Platform	Hardware	Yes	Yes	Yes	Yes	Yes
Application Deployment & Management Server	Hardware	Yes	Yes	Yes	Yes	Yes
Cloud Hosted Server	Hardware	No	Yes	Yes	Yes	Yes
Communications Infrastructure	Hardware	Yes	Yes	Yes	Yes	Yes
LV-CAP Operating System	Software	Yes	Yes	Yes	Yes	Yes
Temperature Sensor app	Software	Yes	Yes	Yes	Yes	Yes
Load Profile Predictor app	Software	No	Yes	Yes	Yes	Yes
Peer to Peer Communications app	Software	Yes	Yes	Yes	Yes	Yes
LoadSense app	Software	No	No	Yes	No	Yes
Network Meshing app	Software	No	Yes	Yes	No	Yes
Dynamic Thermal Ratings app	Software	No	No	Yes	No	Yes
Nortech Communications app	Software	Yes	Yes	Yes	Yes	Yes
Electrical Sensor app	Software	Yes	Yes	Yes	Yes	Yes
Lucy Electric Gridkey Communications app	Software	Yes	Yes	Yes	Yes	Yes

Table 1: Core Component Test Matrix

4.4.2 Timeline

The high-level test and build timeline for the overall OpenLV Solution is outlined in Table 2.

Title	Task	Start Date	End Date	Status
Build & Install Hardware	Build Hardware	Sep-17	Jan-18	In Progress
	Install & Field Test First (4) Devices	Oct-17	Feb-18	Not Started
	Install Method 1 Devices (56)	Jan-18	Mar-18	Not Started
	Install Method 2 Devices (10)	Feb-18	Aug-18	Not Started
	Install Method 3 Devices (10)	Feb-18	Aug-18	Not Started
Central Infrastructure	Install & Set Up - Application Deployment & Management Server	Jun-18		Complete
	Set Up - Cloud Hosted Server	Sep-17	Oct-17	In Progress
	Set Up Communications Infrastructure	Aug-17	Oct-18	In Progress
Test	FAT 1 - OpenLV Core System	Aug-17		Complete
	FAT 2 - OpenLV Network Meshing & Cloud Hosted Server	Sep-17		Complete
	FAT 3 - OpenLV LoadSense & DTR app tests	Dec-17	Jan-18	Not Started
	SAT 1 - OpenLV Core System	Nov-17		Not Started
	SAT 2 - OpenLV Full Solution	Jun-18	Jul-18	Not Started
Trial	Method 1 Trials	Mar-18	Mar-19	Not Started
	Method 2 Trials	Sep-18	Jun-19	Not Started
	Method 3 Trials	Sep-18	Jun-19	Not Started

Table 2: High Level OpenLV Test & Build Timeline

4.5 Factory & Site Acceptance Testing

The tests completed and associated results from FAT 1 and FAT 2 are provided in Annex 3. It is confirmed that both FAT's were successful and the work required to install the initial 4 OpenLV Platforms is being completed. Once these initial devices are installed exact SAT dates will be scheduled and completed. The proposed tests to be completed at SAT are provided in Annex 4.

5 Key Learning Points

The key learning points regarding the specification, design, build and testing of the overall OpenLV solution, to date, are outlined in the below sub-sections.

5.1.1 Specification

The key learning points regarding the specification of the OpenLV solution are as follows:

- It is better to over specify core components, for example the ruggedised PC, when trialling new systems to ensure you have sufficient computational processing power and storage space to support Project trials.
- It is important to ensure the hardware specified fully supports the software you want to implement. In the case of LV-CAP™ operating system it is possible to run the software utilising an ARM chipset rather than an Intel chipset. However, the LV-CAP™ environment relies upon Docker, which is not yet fully supported on the (cheaper) ARM hardware. As a result, an Intel chipset was specified to reduce technical risks for implementation.
- What is seen as a single, simple, requirement from an end user perspective may require more than one Application to deliver it, and so trigger numerous technical requirements which must be cross-referenced.
- It is important to utilise known, existing, tried and tested techniques to capture software requirements. For OpenLV we utilised the MoSCoW approach.
- The sensors specified and the time intervals at which they are sampled will affect what applications it is possible to run on the system. It may be desirable to over-specify sensors to provide for future Application requirements.

5.1.2 Design

The key learning points regarding the design of the OpenLV solution are as follows:

- It is important to ensure that the systems deployed for innovation trials are sufficiently secure. In the case of OpenLV, NCC Group were awarded this role and part of their scope of works is to ensure that the cyber security elements of the proposed trial solution are fit for purpose.
- It is important to ensure that the hardware is designed to enable it to be installed in a number of different ways. The space available for hardware and the mounting requirements for the OpenLV Platform and associated LV monitoring hardware will vary on a site by site basis. As a result, the OpenLV Platform has been designed to be mounted in a number of different ways (magnetic, floor and wall mount).
- To reduce technical risks, off the shelf hardware has been used where possible. For example, the ruggedised PC is an off the shelf piece of hardware that is available from a number of suppliers. In addition, the LV monitoring hardware has already been deployed by WPD in a Business as Usual (BaU) scenario as have the ALVIN Reclose™ devices.

- Safety of on-site maintenance personnel is key and needs to be taken into account when designing new hardware to trial on innovation projects; with this in mind the OpenLV Platform enclosure has been designed to include an isolation switch for the ALVIN Reclose™ devices. This ensures that on site personnel can isolate these devices locally when working on site.
- The decision to utilise a dedicated private APN for the OpenLV Project trials was taken, rather than using a shared private APN. This improves the security of the overall solution.

5.1.3 Build

The key learning points regarding building the OpenLV solution are as follows:

- The approach to building the overall OpenLV solution was to focus on building the core functionality first and then adding additional functionality later. This is reflected in the approach to testing and implementation. This enabled the Project team to focus on delivering core functionality, testing it and then building on this foundation. This approach gets a core system built earlier which means testing can also start earlier in the programme reducing the technical risks of deployment.
- The LV-CAP™ operating system is based on a Docker systems architecture. This enables flexibility when building the overall solution. This architecture means that software, or in the case of OpenLV, Apps, from multiple vendors can be packaged into separate ‘containers’. The core advantage of this is that the containers are designed to run on a shared operating system.
- The LV-CAP™ environment enables developers to write apps in any programming language. This has enabled the overall platform to be built up quickly and easily utilising apps developed by multiple vendors using various programming languages (C++, Java and Go).
- Although LV-CAP™ allows the use of a wide range of programming languages, it still imposes restrictions on the memory usage, processor usage and storage space available to Applications. These restrictions must be clearly communicated to developers at an early stage.
- The main limit on the storage size of Applications is the reliability and cost of deploying them to all required sites over mobile data networks.
- Prior to the deployment of any trial system of this nature it is critical to complete a cyber security review of the proposed solution prior to installation. In the case of OpenLV, NCC Group has completed an assessment of the proposed solution and has confirmed that the OpenLV Platform can be deployed for field trials.

5.1.4 Test

The key learning points regarding testing the OpenLV solution are as follows:

- A dedicated test rig was built to enable testing of two development OpenLV Platforms. This test rig includes relevant sensors (temperature, voltage and current) to provide data inputs to the test system. This test rig was built as early as possible within the programme to enable components to be soak tested for as long as possible prior to installation.
- Having a controlled test rig in a laboratory environment allows defined inputs (currents, voltages and temperatures in this case) to be applied and the outputs verified. Where necessary scaling and unit issues can be resolved under laboratory conditions. This would be very difficult to achieve in a field situation on a live network.
- Formally defining the requirements for the overall solution is key to ensure that the FAT and SAT documents test each of the individual Project requirements. Both the FAT and SAT documents refer back to the specific requirements to ensure relevant tests are completed at each stage.

6 Summary

This SDRC report has presented the Systems Architecture for the overall OpenLV solution outlining the design concepts, key hardware components, key software components and associated central infrastructure and communications components.

The approach taken to build and test the overall OpenLV Solution has been outlined and supported by the provision of the OpenLV Requirements Specification, FAT documentation, and the associated results of the FAT tests completed to date. The approach to testing the overall solution has been outlined and this will enable the OpenLV Platforms to be tested in the field in late 2017 and rolled out at scale in 2018. This approach will ensure relevant learning can be generated from the Project trials under the 3 methods: 1) The network capacity uplift trials, 2) The community engagement trials and 3) The OpenLV extensibility trials.

In addition, the key learning points recorded at this stage of the Project in relation to the specification, design, build and testing of the overall OpenLV solution have also been recorded. It is confirmed that the SDRC and associated evidence requirements have been met and this is supported by the compliance matrix provided in Section 1.

References

1. OpenLV Project Direction, 16th December 2016, https://www.ofgem.gov.uk/system/files/docs/2017/02/open_lv_formal_project_direction.pdf
2. OpenLV Initial Screening Pro-forma, https://www.ofgem.gov.uk/system/files/docs/2016/04/electricity_isp_proforma_nic_wpd_openlv_pdf_0.pdf
3. OpenLV Full Submission Pro-forma, https://www.westernpower.co.uk/docs/Innovation/Current-projects/Open-LV/NON-CONFIDENTIAL-OpenLV-NIC-Bid-2016-WPD_EN_NIC_02.aspx
4. NIC Governance Document, Version 2.1, https://www.ofgem.gov.uk/publications-and-updates/version-2-1-network-innovation-competition-governance-documents
5. LV Common Application Platform Public API, July 2017, V04.03.00.

Annex 1. Assessment of the LV-CAP™ operating system for the OpenLV project

An assessment of LV-CAP™ operating system developed as part of the "Common Application Platform for Low Voltage Network Management", InnovateUK project, was undertaken. The purpose of this assessment was to identify what changes were required to the LV-CAP™ operating system to enable the deployment of the overall OpenLV solution required to complete Project trials. This assessment identified four security improvements that needed to be made each of which have been addressed. The four security improvements that have been made are as follows:

- **MQTT Broker Authentication:** The MQTT broker did not check whether application containers connecting to it supplied a username and password, i.e. anonymous connections are permitted. Therefore, any attacker would have been able to send network traffic to the MQTT broker, monitor the communication activity of the other operating applications, and impersonate them on the broker.
- **Shared MQTT Authentication Credentials:** The username and password for all application containers within the InnovateUK trials were the same. Consequently, any container could connect as though it was any other container. In the event the system was breached then every container would be impacted and would have required an update to restore system operation.
- **Symmetric MQTT Authentication Credentials in Mosquitto:** MQTT Broker connections are protected using a username and password, the password must be stored both in the container and in the authentication data for the MQTT Broker. In the Mosquitto implementation used as part of the InnovateUK trials, the Broker on the LV-CAP™ platform held the password text for all users. Although this was encrypted, there was a risk that theft of, or other unauthorised access to, an individual LV-CAP™ platform could provide an attacker with the passwords for all container users.
- **No Access Control Lists on MQTT Broker:** The MQTT Broker did not implement Access Control Lists (ACLs) controlling which topics a given user can publish on or subscribe to. Once connected to the MQTT Broker any container was able to view the topics published to by any other container and was able to impersonate any of them.

In addition the following changes were identified and have been amended in the latest revision of the Application Programming Interface (API) documentation [Ref. 5]:

- Changes to the software to enable performance issues with the database to be resolved; and
- Changes to the software to support multiple communication containers, as the OpenLV solution requires communication with multiple servers from independent suppliers to be supported.

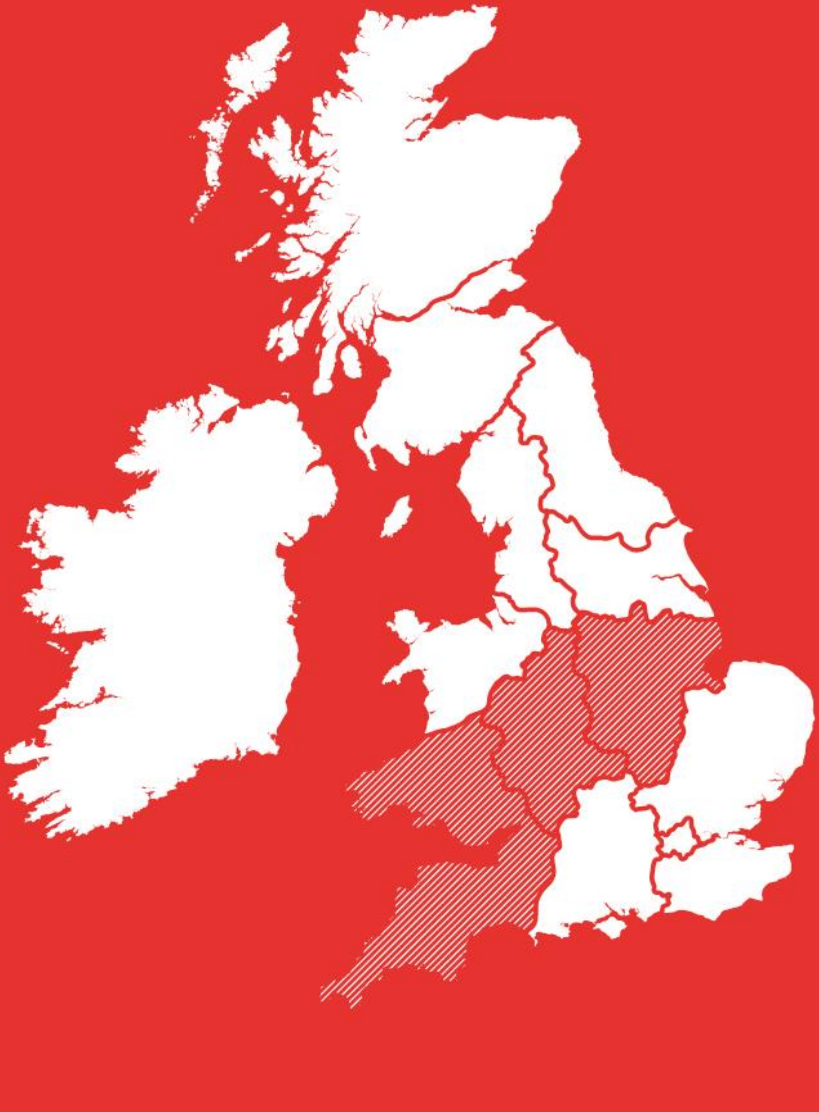
Annex 2. OpenLV Solution Requirements Specification

WESTERN POWER DISTRIBUTION

OPEN LV

OPENING UP THE SMART GRID

Requirements Specification



Report Title:	OpenLV Solution Requirements Specification
Report Status:	Issued
Project Ref:	WPD/EN/NIC/02 – OpenLV
Date:	20 September 2017

Document Control

	Name	Date
Prepared by:	Tim Butler	August 2017
Reviewed by:	Richard Ash	August 2017
	Richard Potter	August 2017
Recommended by:	Dan Hollingworth	20 September 2017
Approved (WPD):	Mark Dale	20 September 2017

Revision History

Date	Issue	Status
20 September 2017	1.0	Issued pre-FATs Part 2
15 August 2017	0.1	Draft issued for comment pre-FATs Part 1.

DISCLAIMER

Neither WPD, nor any person acting on its behalf, makes any warranty, express or implied, with respect to the use of any information, method or process disclosed in this document or that such use may not infringe the rights of any third party or assumes any liabilities with respect to the use of, or for damage resulting in any way from the use of, any information, apparatus, method or process disclosed in the document.

Telephone +44 (0) 1332 827446. E-mail wpdinnovation@westernpower.co.uk

Contents

Table of figures

Table of tables

1 Introduction

1.1 Purpose

The Purpose of this document is to provide a record of the requirements for the overall OpenLV solution that will be required to support Project trials for the three Methods outlined in the Full Submission Proforma (OpenLV Bid document).

1.2 Background

The FSP provides a high-level description of the overall OpenLV solution that will be required to support Project trials for the three Methods.

In order to define the requirements for the overall OpenLV Solution, the key hardware and software components have been defined, then the requirements for each component have been identified.

In order to prioritise requirements, the MoSCoW approach has been utilised. This approach is a well-known technique used in business analysis and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement.

Each requirement has been identified and prioritised using the MoSCoW approach, that stands for Must, Should, Could and Will not:

- M – Must have this requirement to meet the business needs;
- S – Should have this requirement if possible, but project success does not rely on it;
- C – Could have this requirement if it does not affect anything else in the project; and
- W – Will not deliver this requirement at the current time, but it could be delivered at a later date.

1.3 OpenLV Solution Overview

The OpenLV solution to be trialled within the OpenLV project, utilises a distributed intelligence device, built on a software platform, LV-CAP™, developed to enable multiple applications and solutions to be deployed in a single enclosure. The trials will, across three 'Method' areas, demonstrate that the platform can:

- monitor the network in real-time;
- process the collected data to determine the need for action to manage the network;
- implement that action if necessary;
- provide this data to third party companies and equipment; and
- provide this data to community groups.

1.3.1 Method 1: LV network capacity uplift

Method 1 will demonstrate the capability of the LV-CAP™ platform to perform measurements and control from within an HV/LV substation, in 60 substations (30x pairs).

To demonstrate this, the deployed trial hardware will utilise monitored data to predict future network load and when necessary, automatically share the feeder load between two transforms. This will be demonstrated through direct control of ALVIN Reclose™ circuit breakers installed in the substations at either end of the utilised feeder in a subset (5x pairs) of Method 1 installations.

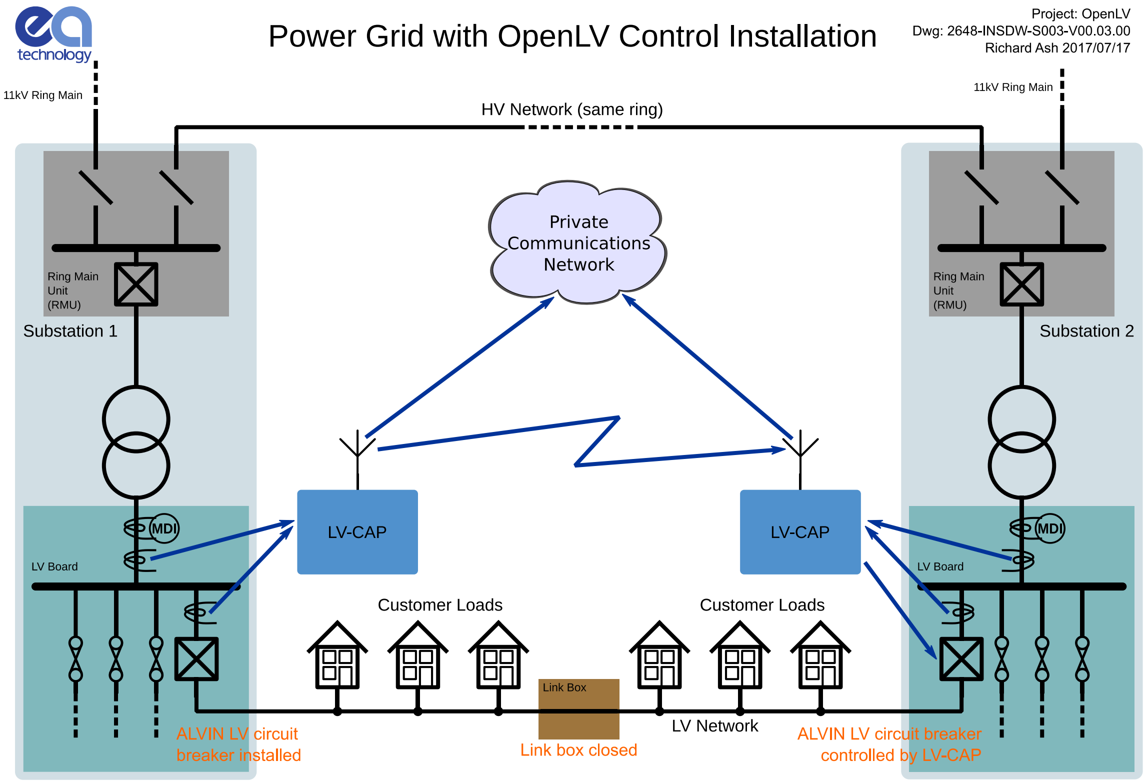


Figure 1: OpenLV Method 1 Overview

1.3.2 Method 2: Community engagement

Once deployed, the LV-CAP™ platform can be used to provide data to community groups or individual customers. LV-CAP™ platforms deployed for Method 2 implementation will be identical to those deployed for Method 1 but will not, in any situation, have ALVIN Reclose™ devices installed as well.

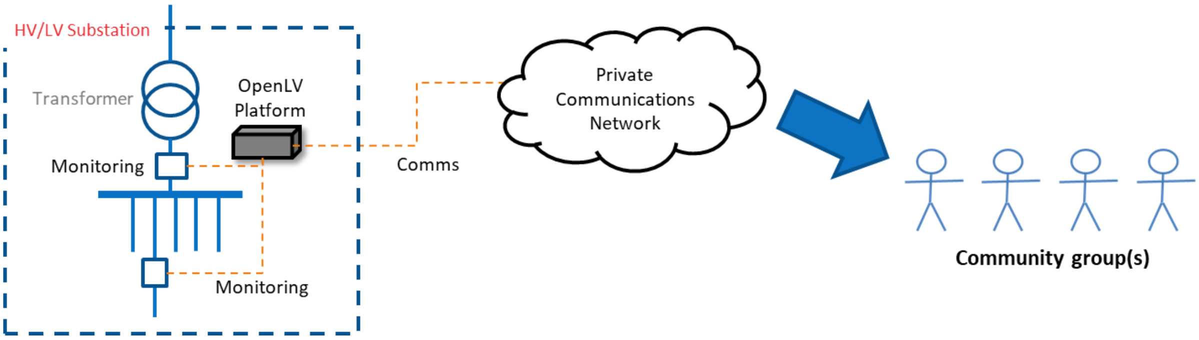


Figure 2: OpenLV Method 2 Overview

10 LV-CAP™ devices have been allocated for the implementation of Method 2 activities.

1.3.3 Method 3: OpenLV extensibility

Once deployed, the LV-CAP™ platform can be used to provide a secure platform for third parties to provide services to the DNOs, customers, and wider industry. This may take the form of pure Applications, or a combination of Applications and connected external devices. As with Method 2, however, Method 3 installations will not, in any situation, have ALVIN Reclose™ devices installed as well.

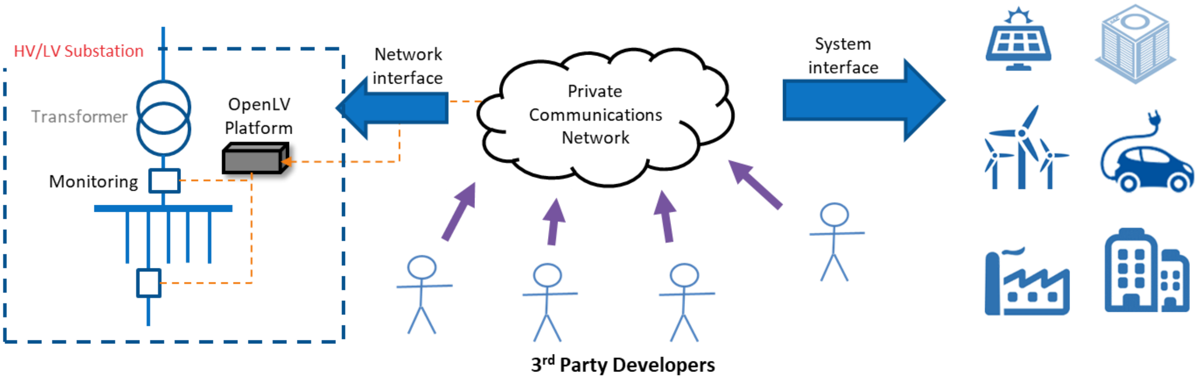


Figure 3: LV-CAP™ Method 3 Overview

10 LV-CAP™ devices have been allocated for the implementation of Method 3 activities.

1.4 LV-CAP™ Software Platform Overview

The LV-CAP™ software platform is designed to enable a single hardware deployment to monitor the network and make the gathered data available to multiple software Applications running on the platform. These Applications could be developed by multiple manufacturers and control various unrelated network assets without any application being influenced or affected by another.

This enables a single investment in the hardware to support deployment of multiple solutions to benefit the network.

The figure below demonstrates the deployment of software applications within the LV-CAP™ platform under the OpenLV Project.

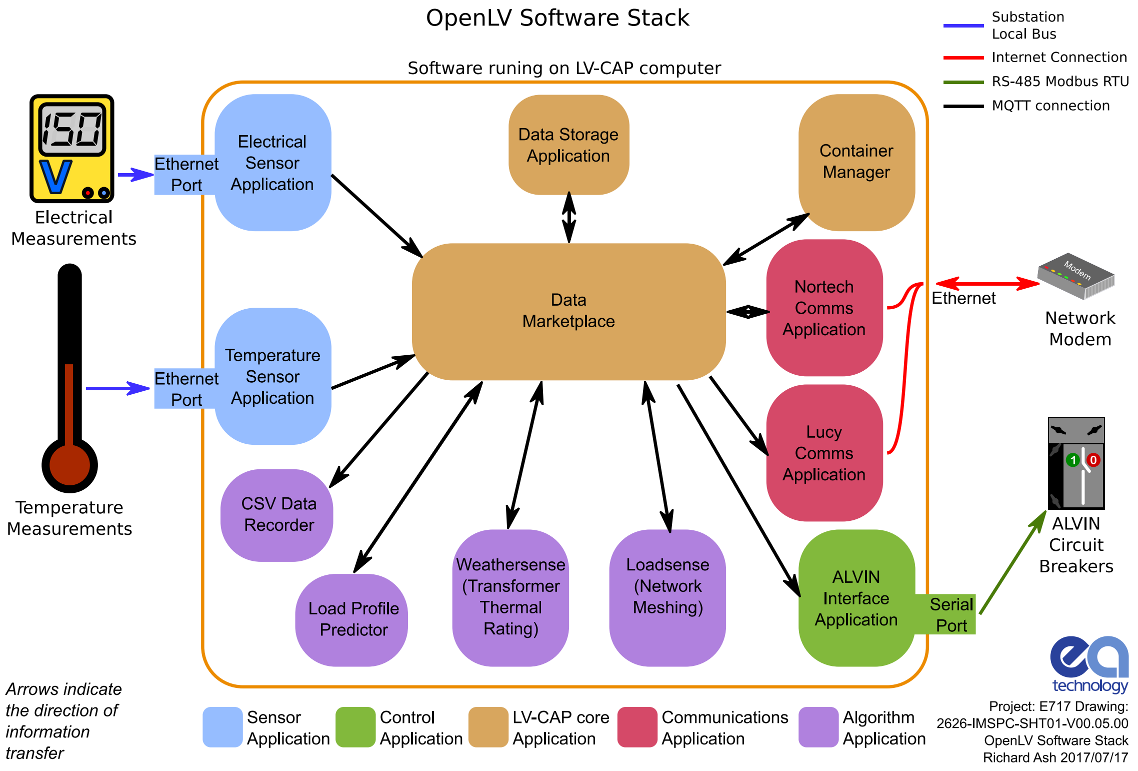


Figure 4: LV-CAP™ Software Platform Overview

1.5 Report Structure

The structure of this document is as follows:

- **Section 2: Grouping of the OpenLV Solution Requirements –** Provides an overview of the key components of the overall system.
- **Section 3: OpenLV Solutions Requirements –** Outlines the requirements for each component of the overall OpenLV solution.
- **Section 4: The Way Forward –** Outlines how the requirements will be used within the Project.

2 Grouping of the OpenLV Solution Requirements

The key components of the overall OpenLV solution have been assessed and requirements have been grouped under the titles outlined in

Table 1: Grouping of Requirements

Group	Description
Intelligent Substation Devices	The Intelligent Substation device is an enclosure, containing a ruggedised PC capable of being installed in harsh environments and interface connections to receive data from external sensors. Applications installed within the LV-CAP™ Software Platform gather data, store it locally, process if necessary and send requested information back to central servers.
LV-CAP™ Software Platform	The LV-CAP™ software platform runs on the intelligent substation devices. This is an operating system that enables multiple Applications to be installed in software containers on a single device. The software also provides Apps with access to data provided from the sensors which are installed in each LV substation.
LV Monitoring Equipment	LV network monitoring provides the core data to be utilised by all Apps that will be deployed on the Intelligent Substation Devices.
Temperature Sensing	Temperature sensors are required to monitor the temperature of the LV transformer and ambient air temperature. This data is provided to the relevant container 'Apps' within the LV-CAP™ software platform and can be used for Dynamic Thermal Rating (DTR) of LV transformers to release additional capacity from existing LV network assets.
LV Network Meshing	This section details the hardware to be installed and the associated software to enable meshing of individual feeders between two LV substations. This has the potential to release additional capacity from existing LV network assets.
Load profile predictor application	A load profile predictor application is required to utilise historical load on both the transformer and specific LV feeder and predict the likely load profile for the future.

Group	Description
CSV data recorder application	Storage of all data captured by the system, information generated by any applications and a record of any actions implemented are required to be stored on non-volatile memory within the ISD.
LoadSense	Loadsense is an application designed to respond to outputs from Weathersense relating to real time and predicted network loading. These outputs may trigger an immediate (relatively) response or be a prediction alert, effectively stating that "unless network load is less than predicted, the transformer is going to exceed its RTTR rating in x-hours." Loadsense would schedule a LV network meshing to occur prior to that time slot, if the networks to be connected also have sufficient capacity.
Communications	This section covers the applications relating to communications to and from the deployed trial devices. Communications are required between deployed devices and separate, centralised data servers and between individual LV-CAP™ platforms.
Centralised System Requirements	Centralised systems are required to enable 'Apps' to be deployed on the intelligent substation devices and to store data that is required for the assessment of Project trials.
Overall System Requirements	This group covers all other requirements not captured under the groups defined above.

The numbering system used to identify individual requirements within Section 3 maintains traceability of the individual requirements identified. The numbering system is as follows:

- I:XXX: Provides relevant information regarding the overall solution;
- M:XXX: Outlines a 'must have' requirement, these are requirements that are needed to ensure the success of the Project;
- S:XXX: Outlines a requirement that 'should' be provided, if possible, but project success does not rely on it;
- C:XXX: Outlines a requirement that 'could' be provided, but does not affect anything else in the project;
- W:XXX: Outlines requirements that 'will not' be delivered at the current time, but could be delivered at a later date.

3 OpenLV Solution Requirements

EA Technology undertook an InnovateUK Energy Catalyst project with the University of Manchester and Nortech Management Ltd to develop a Common Application Framework for LV Network Management. The core software platform developed under the project is called Low Voltage Common Application Platform (LV-CAP™) and consists of a number of Docker containers communicating with each other using MQTT.

This section outlines the functional requirements and necessary interfaces with 3rd party equipment and software for LV-CAP™. The OpenLV Project will deploy 80 LV-CAP™ devices to test the platform as outlined above.

The final 'product' delivered will be referred to as the OpenLV Solution, acknowledging that the specific combination of hardware and software that results from this requirements specification will not be implemented again outside of the OpenLV Project, although the requirements will likely be utilised as a base of core functionality for future deployments.

3.1 Overall System Requirements

The purpose of the OpenLV Project is to test the LV-CAP™ platform's capabilities and demonstrate its ability to provide benefits to the industry, communities and third-party companies. Therefore, the trial system deployed by the project must demonstrate that it meets the below requirements.

- gather voltage and current data relating to the LV network;
- store this data for a period of time suitable to meet the requirements of the applications running on the platform;
- make this data available to multiple applications running on the platform;
- enable multiple applications to process and manipulate this data;
- transmit the pertinent, generated information back to central location(s), without requiring the raw data to be transmitted as well;
- be able to transmit the raw data to a data repository if required;
- enable deployed applications to make decisions regarding the LV network, based on the monitored data;
- implement those decisions autonomously if deemed appropriate for the network based on decision processes agreed with WPD;
- Make data and information, not considered operationally sensitive by the DNOs, available to communities, academia and third-party businesses.

In order to achieve these above requirements, a number of applications are necessary with specific, additional requirements detailed in the Annexes to this document.

3.2 Intelligent Substation Devices (ISDs)

3.2.1 Overall Statement

I:001. The Intelligent Substation device is an enclosure, containing a ruggedised PC capable of being installed in harsh environments and interface connections to receive data from external sensors.

I:002. Applications installed within the LV-CAP™ Software Platform gather data, store it locally, process if necessary and send requested information back to central servers.

I:003. The ISDs are responsible for providing the LV-CAP™ platform and associated software containers with a computing environment suitable for implementation for the duration of the OpenLV Project.

I:004. The Operating System for LV-CAP is GNU/Linux. The LV-CAP™ core software will be installed on the operating system along with the following containers, designed to deliver the required OpenLV solution functionality.

- LV-CAP™ Software Platform
- LV Monitoring
- Temperature Sensing
- Data Upload Application
- Dynamic Thermal Ratings Application
- Load Profile Predictor
- LoadSense
- ALVIN Reclose™ Interface Application
- CSV Data Recorder
- Management Comms Application
- Peer to Peer Communications Application

I:005. The requirements for each element of the ISDs are detailed later in this section.

3.2.2 Services required

I:006. The services detailed here refer to those required to be provided by the ISD platform as a whole, rather than individual elements, either hardware or software, within the platform.

Processing capability

M:001. The computational hardware must be based upon an industrial PC architecture.

M:002. The computational hardware within the ISD must be capable of running LV-CAP™, the additional application containers detailed below and up to three others to be written by community groups and / or third-party companies.

I:007. Applications to be developed by community groups or third parties will be subject to ‘reasonable complexity’ restrictions, as determined solely by the LV-CAP™ team at EA Technology to ensure the hardware is capable of delivering all project requirements.

I:008. The system within each enclosure must be capable of ensuring that all applications defined in this document can operate simultaneously in order to meet their individual requirements.

Communications

M:003. The ISD must have a modem / router installed capable of providing the below functionality.

I:009. The ISD must be capable of communication, both incoming and outgoing, with a control server based on Nortech's iHost platform located at EA Technology's Capenhurst offices.

I:0010. The ISD must be capable of outgoing communications (i.e. initiated without external request) for the transmission of data and information to the iHost platform located at EA Technology's Capenhurst offices.

I:0011. The ISD must be capable of outgoing communications (i.e. initiated without external request) for the transmission of data and information to a separate cloud based data storage server owned and operated by Lucy Electric.

I:0012. The ISD must be capable of communication, both incoming and outgoing, with other ISD's located in a local, (geographically similar) location to transfer data between the two platforms in support of the network automation functionality.

I:0013. The ISD must be capable of receiving data pertaining to the monitoring of the LV network from monitoring hardware installed in the same substation as the ISD.

I:0014. The ISD must be capable of receiving data pertaining to the temperature of the associated transformer and ambient air temperature.

Storage Capacity

M:004. The internal, non-volatile storage of the ISD must be of sufficient capacity to store all data captured by the sensors, generated by installed application containers and received from other ISDs, for the duration of the OpenLV Project (minimum period of 18 months).

Security (Physical)

M:005. The enclosure must be physically secured from unauthorised access. It cannot be assumed that the trial equipment will be always installed at indoor, secured substations.

Environmental

I:0015. The ISDs will be installed in a mixture of indoor and outdoor substations and the enclosure must be suitable for use in either instance.

M:006. The enclosure therefore, must be suitably IP rated to adequately protect the internal equipment in all potential substation environments and installation methods.

Protection

M:007. In the situation where network meshing hardware (ALVIN Reclose™) is installed and connected, it must be possible for a maintenance engineer to isolate the LV-CAP™ platform from the Reclose™ devices enabling manual operation of the network meshing capability.

M:008. This communication isolation must be capable of being 'locked' in a given state, to ensure automated operation cannot resume unexpectedly.

M:009. The enclosure must be non-conductive to avoid potential earthing issues.

Watchdogs & reset capability

I:0016. It is essential that the requirement for manual (in person) resets in the event of loss of communications or loss or responsiveness is avoided wherever possible due to the range of locations in which the trial equipment is to be installed.

M:0010. It must be possible to remotely reset the platform without requiring physical access to the ISD through establishing remote access to the router to trigger a reset of the computational hardware.

I:0017. The ISD should have appropriate 'Watchdogs' to ensure the individual devices within the overall ISD reset automatically if necessary.

W:001. There should be a Watchdog to ensure the router / model is reset if it enters a non-responsive state.

S:001. There should be a Watchdog to ensure the computational hardware is reset if it appears to have entered a non-responsive state, as indicated by a lack of network communication within the ISD.

3.2.3 Mounting arrangements

M:0011. The enclosure must be capable of multiple mounting options, for example:

- direct wall mounting;
- magnetic mounting bracket on the side of switchgear equipment; or
- bolting in place on the floor.

I:0018. Individual arrangements on site will determine which approach shall be used.

3.2.4 Products required

M:0012. The ISD must include an enclosure for the necessary hardware (industrial PC, modem and ancillary connection elements for monitoring devices.

I:0019. The ISD must include an industrial, ruggedised PC capable of running the LV-CAP™ platform and associated application containers.

I:0020. The ISD must include LV monitoring equipment for the gathering of information about the associated LV network.

I:0021. The ISD must include a modem enabled for two-way data transmission, capable of communicating over multiple mobile networks.

M:0013. The PC hardware must be installed with the LV-CAP™ platform.

3.2.5 Dependencies

I:0022. The system must be capable of installation within electrical safety standards required by WPD for deployment of LV monitoring equipment on their network, specifically:

- Standard Technique: SP2KD

3.2.6 Performance measurement

I:0023. The ISDs must enable the LV-CAP™ platform, including subsidiary application containers and associated connected hardware, to meet all requirements as defined in sections below.

I:0024. The ISDs must be capable of performing these functions for at least the duration of the OpenLV Project trials.

M:0014. The overall system must include the necessary monitoring equipment to gather necessary data for each software application.

M:0015. The overall system must be capable of communicating with physically separate network monitoring hardware providing voltage and load data.

M:0016. The overall system must be capable of measuring incoming signals from the temperature monitoring hardware.

M:0017. The overall system must be capable of communicating with ALVIN Reclose™ devices (x3) if they are installed in the substation.

I:0025. ALVIN Reclose™ devices will only be installed in 10 locations within the trials.

I:0026. Every ISD installed must be capable of communication with ALVIN Reclose™ devices as sites determined as suitable for their installation will not be identified until later in the project.

M:0018. The overall system must be capable of communicating with other ISDs via non-dedicated cable connection methods such as the use of a modem-to-modem connection.

I:0027. This communication link will be used to share information relating to the LoadSense application between adjacent, linked, substation LV-CAP™ platforms.

3.3 LV-CAP™ Software Platform

3.3.1 Overall statement

I:0028. The purpose of the OpenLV Project is to demonstrate the LV-CAP™ platform is a capable of operating as a non-specific distributed intelligence platform for the LV network.

I:0029. The LV-CAP™ software platform runs on the ISDs. This is an operating system that enables multiple Applications to be installed in software containers on a single device.

I:0030. The software also provides Apps with access to data provided from the sensors which are installed in each LV substation.

All applications developed for and deployed to the LV-CAP™ platforms must conform to the LV-CAP™ API document (

I:0031. Appendix A – LV-CAP™ API).

I:0032. The exceptions to 0, are below and will not be implemented as part of the OpenLV Project:

W:002. Individual message signing (see section 8.1.2 for further information).

W:003. Signing of Docker Image files.

W:004. Only one instance of each Application will be run (see section 4.2) on LV-CAP.

As a result, Applications may continue to use legacy GUID identifiers.

W:005. To simplify TLS implementation, TLS keys and certificates will be built into Docker Image files. The end date of TLS certificates should be set beyond the end of the OpenLV project trials in September 2019. TLS implementation is mandatory.

W:006. The Priority feature of the data storage APIs will not be implemented, with all queries returning messages of all priorities. Applications are free to output Priority data, but it will not be parsed yet. Similarly, requests may be made with Priority key values, but the key will be ignored.

3.3.2 Services required

I:0033. The ISD requires a number of ‘services’ to be provided by the LV-CAP™ platform. For example, these include monitoring of the LV network, storing the associated data and making predictions based on the data gathered.

I:0034. Each service required by the ISD is provided by an individual, specific software application.

I:0035. Rather than providing directly measurable outputs, the LV-CAP™ platform enables the operation of the various applications, and makes the data gathered and generated available.

I:0036. The LV-CAP™ platform will be subjected to a cyber-security assessment, including penetration testing, architecture evaluation and code review.

I:0037. The LV-CAP™ platform must demonstrate an appropriate level of security within the system. This will be informed by the cyber-security review to be undertaken by NCC Group.

M:0019. The LV-CAP™ platform must ensure communications between applications and the message broker are encrypted and authenticated to prevent application impersonation.

3.3.3 Products required

Hardware environment

I:0038. The OpenLV project hardware consists of an industrial PC based around a dual-core Intel Core i3 processor with 8GB of RAM and a 512GB SSD.

I:0039. This PC provides the processing power and storage for the whole LV-CAP solution.

I:0040. It has two Ethernet ports for network communications, one of which is utilised to connect a stand-alone 4G router which provides wide area network communications.

Base operating system

I:0041. The ruggedised PC within the ISD will be running 64-bit Ubuntu Server 16.04 LTS with current updates applied.

3.3.4 Dependencies

I:0042. The LV-CAP™ software platform, as deployed within the OpenLV project is managed and controlled via an instance of Nortech's iHost server.

I:0043. The LV-CAP™ platform deployed within the OpenLV Project requires PC hardware, with an installed operating system, as defined above.

3.3.5 Performance measurement

M:0020. The LV-CAP™ software platform must be demonstrated to run all software applications deployed to the platform if those applications conform to the API documentation provided.

3.4 LV Monitoring Equipment

3.4.1 Overall statement

I:0044. LV network monitoring provides the core data to be utilised by all Apps that will be deployed on the Intelligent Substation Devices.

I:0045. The ISDs must have connected monitoring equipment for the collection of data pertaining to the LV Network.

I:0046. The equipment must provide the ISD with voltage and current measurements at sufficient resolution and granularity to enable the effective operation of each application on the platform to meet the requirements specified in this document.

3.4.2 Services required

M:0021. The complete ISD system must be capable of measuring the following from appropriate sensor hardware for all phases.

Voltage Measurements

I:0047. RMS Voltage phase to neutral (x3) at the substation busbars.

Current Measurements

I:0048. For each circuit measured:

- RMS current in each phase
- Power factor for each phase
- Real and Reactive power flow each phase (including direction, so reverse power is read as negative current)

Temperature Measurements

I:0049. Outdoor ambient air temperature must be measured.

I:0050. Indoor ambient air temperature (indoor substations) must be measured. In multiple room substations, this will be in the transformer chamber.

I:0051. Transformer top oil temperature (or as close an approximation as can be managed).

3.4.3 Products required

I:0052. Lucy Electric GridKey’s MCU520 system must be utilised as the monitoring platform.

I:0053. 6x Flexible Rogowski Coils or Current Transformers, compatible with the GridKey MCU520 system must be provided, to monitor the total transformer load and the specific feeder connecting to an adjacent substation.

I:0054. Modified fuse carriers are the preferred method for connecting the GridKey platform to LV network.

I:0055. 3x modified fuse carriers are required for substations where there is sufficient capacity, (i.e. at least one empty fuse holder per phase), within the fuse board.

I:0056. G-Clamps are required for connection of the GridKey platform to the substation neutral busbar.

I:0057. In the event that sufficient capacity on the fuse board is not available for all phase connections, (I:0055), then G-Clamps should be used as with I:0056.

I:0058. An application container (GridKey Sensor Container) is required to provide the interface capabilities between the MCU520 and the LV-CAP™ platform.

I:0059. The GridKey Sensor Container will communicate directly with the GridKey MCU520 via a local Ethernet port.

W:007. The GridKey Sensor Container will not have access to the wide area communications network.

I:0060. The requirements for this application, enabling communication between the LV-CAP™ platform and Lucy Electric GridKey's MCU520 platform are detailed in Appendix C – Lucy Electric Application Container.

3.4.4 Dependencies

I:0061. For the application to function, an MCU520 must be procured from Lucy Electric and connected to the ISD hardware via the ethernet port.

3.4.5 Performance measurement

M:0022. The LV-CAP™ platform must be provided with timestamped readings of voltage and current readings from the GridKey Sensor Application.

S:002. It is desirable that synchronous sampling is implemented to aid in analysis of the gathered data.

I:0062. These readings must be provided at a frequency of once per minute sufficient to meet the requirements of other applications running on the platform.

M:0023. It must be demonstrated that data readings from the GridKey Sensor Application can be acquired at a rate of once every minute for an indefinite period.

M:0024. It is essential that the system demonstrate the capability of continuous data capture at a rate of once every ten (10) seconds for a period of at least one hour.

3.5 Temperature Sensing

3.5.1 Overall statement

I:0063. Temperature sensors are required to monitor the temperature of the LV transformer and ambient air temperature. This data is provided to the relevant container 'Apps' within the LV-CAP™ software platform and can be used for Dynamic Thermal Rating (DTR) of LV transformers to release additional capacity from existing LV network assets.

I:0064. It is necessary for the operation of the DTR application that the ambient temperature and specific temperatures relating to the transformer are collected and made available.

3.5.2 Services required

M:0025. The ISD must have the means to collect thermal readings as defined in I:0064, receive and store this data in a format readable by other applications.

3.5.3 Products required

I:0065. This specification requires, at a minimum:

- physical means for detecting the ambient temperature and specific transformer temperatures;
- software compatible with the LV-CAP™ platform for receiving and managing the data from these sensors.

I:0066. Therefore, the ISDs must be equipped with the necessary thermocouples to monitor the range of temperatures required by the DTR application.

I:0067. The ISDs must be equipped with the necessary interface equipment to connect the temperature monitoring equipment to the LV-CAP™ hardware.

M:0026. The temperature sensing application container must take the values provided by the thermocouple(s) and pass them to the LV-CAP™ system for storage in non-volatile memory.

I:0068. The temperature readings must be recorded at a rate of once every minute for an indefinite period.

3.5.4 Dependencies

I:0069. For the data to be provided, the application requires an appropriately sensitive thermocouple to be connected to the ISD via a suitable data port.

3.5.5 Performance measurement

M:0027. The LV-CAP™ platform must be provided with timestamped temperature readings from the Temperature sensing application at a rate of once per minute.

I:0070. These readings must be provided at one-minute intervals throughout the duration of the OpenLV Project.

I:0071. It is noted that future business-as-usual deployments may require the ability to vary the rate of data capture.

S:003. Therefore, it is desirable to demonstrate that the rate of data capture can be varied between 10-second and 10-minute intervals, in 10-second stages.

3.6 LV Network Meshing

3.6.1 Overall statement

I:0072. This section details the hardware to be installed and the associated software to enable meshing of individual feeders between two LV substations. This has the potential to release additional capacity from existing LV network assets.

I:0073. The OpenLV Project must demonstrate that autonomous control of network assets, based on pre-defined logic, is possible via a distributed intelligence platform (the ISDs).

I:0074. Within the OpenLV Project, this is to be demonstrated through direct control of ALVIN Reclose™ devices to mesh and de-mesh adjacent LV networks.

3.6.2 Services required

M:0028. The LV Network Meshing Application must enable communication capabilities between the LV-CAP™ platform and ALVIN Reclose™.

M:0029. The application must read the desired information from the ALVIN Reclose™ devices, and pass it to the LV-CAP™ platform for storage in non-volatile memory through the CSV data recorder.

- MIR_BUS_VOLTAGE_RMS
- MIR_CABLE_VOLTAGE_RMS
- MIR_LINK_CURRENT_RMS
- MIR_OPEN_OPERATIONS
- MIR_CLOSE_OPERATIONS
- MIR_WATCHDOG_FAULTS_DETECTED
- MIR_CHIP_TEMPERATURE
- MIR_REACTIVE_POWER
- MIR_ACTIVE_POWER
- MIR_UPTIME_HIGH
- MIR_SWITCH_TEMPERATURE
- MHR_SHADOW_FAULT_STATUS

M:0030. It must be demonstrated that data readings from the LV Network Meshing Application can be acquired at a rate of once every minute for a period of at least one hour.

I:0075. The variable MHR_SHADOW_FAULT_STATUS reads the current state of the circuit breaker within the connected ALVIN Reclose™ devices.

I:0076. LV Network Meshing Application must be able to trigger an opening or closing of the ALVIN Reclose™ device's circuit breaker, meshing, or de-meshing the network as applicable.

I:0077. For the purposes of the OpenLV Project trials, it is preferred that a record of the process is stored for project evaluation.

M:0031. The application must therefore store a record of reacting to a command, whether to initiate or break a network mesh, in non-volatile memory.

M:0032. The application must also store a record of the state of the connected ALVIN Reclose™ devices both before and after implementing of the command, i.e. open / closed.

I:0078. If there are no attached circuit breakers, an appropriate 'error code' must be provided instead.

M:0033. All control communications, whether acknowledged by a connected ALVIN Reclose™ device or not, must be stored in memory on the LV-CAP™ platform for later analysis if required.

M:0034. The ISD must be electrically isolated from ALVIN Reclose™ devices installed within the substation.

3.6.3 Products required

I:0079. This specification requires, at a minimum, provision of an application capable of communicating with ALVIN Reclose™ devices to deliver the above requirements

I:0080. The ALVIN Reclose™ devices will be procured by the OpenLV Project.

I:0081. The interconnection cable to interface the ISD with the ALVIN Reclose™ devices will be provided by EA Technology's LV Solutions team, in collaboration with EA Technology's HV59s team.

3.6.4 Dependencies

I:0082. The application requires a control input from another application (LoadSense) to determine whether to open or close attached circuit breakers.

M:0035. The LV Network Meshing Application must only respond to instructions to mesh or de-mesh the network through opening and closing of ALVIN Reclose™ circuit breakers from the LoadSense application.

3.6.5 Performance measurement

M:0036. The ALVIN Reclose™ devices must respond to an instruction to initiate an open or close operation.

M:0037. It must be demonstrated that the control signals for transmission to the ALVIN Reclose™ devices from the LV Network Meshing application are triggered whether an ALVIN Reclose™ devices is installed within the substation or not.

3.7 Load Profile Predictor Container 'App'

3.7.1 Overall statement

I:0083. A load profile predictor application is required to utilise historical load on both the transformer and specific LV feeder and predict the likely load profile for the future.

3.7.2 Services required

I:0084. This application must utilise historical load values to generate a forecast of future load on the transformer and individually monitored LV feeder.

I:0085. It must not utilise all the data available on the trial platform as future systems will not have access to 'unlimited' historical data due to local storage limitations.

I:0086. The duration of historical data utilised by the application should be confirmed with explanation of why that duration has been selected.

3.7.3 Products required

I:0087. This specification requires, at a minimum, an application that utilises the historical load data to create a predictive forecast for the network and asset in question.

3.7.4 Dependencies

I:0088. In order to predict future load profiles, the Load Profile Predictor application requires the historical data gathered by the GridKey MCU520 and stored in non-volatile memory.

3.7.5 Performance measurement

I:0089. This application must utilise a sufficient period of historical data to provide sufficient predictive assurance for the calculated outputs.

M:0038. The Load Profile Predictor application must output a load forecast at half-hourly intervals for the next 24-hour period.

3.8 CSV Data Recorder Application

3.8.1 Overall statement

I:0090. Storage of all data captured by the system, information generated by any applications and a record of any actions implemented are required to be stored on non-volatile memory within the ISD.

3.8.2 Services required

M:0039. This application must store all data output by each application container on the platform.

M:0040. All data must be timestamped such that raw data, and processed information derived from that data can be reconstructed at a later date if required.

M:0041. All data must be attributable to the application that created and published it.

3.8.3 Products required

I:0091. This specification requires, at a minimum, an application that monitors all communications traffic within the LV-CAP™ platform and stores it with a timestamp, and provides a record of which application published that item of data.

3.8.4 Dependencies

I:0092. This application requires other applications to be running on the LV-CAP™ platform to provide data and processed information for storage.

I:0093. The application must be granted sufficient authorisations within the platform to enable access to all data and information for storage.

3.8.5 Performance measurement

M:0042. It must be demonstrated that accurate data values are stored in non-volatile memory for each application on the LV-CAP™ platform that is providing measured or calculated data.

M:0043. This data must be stored at a frequency that matches the outputs of the individual applications.

3.9 LoadSense Container 'App'

3.9.1 Overall statement

I:0094. Loadsense is an application designed to respond to outputs from Weathersense relating to real time and predicted network loading.

I:0095. These outputs will trigger an immediate response to outputs from the Dynamic Thermal Rating application.

I:0096. The LoadSense application implements network meshing through the ALVIN Reclose™ devices and associated LV Network Meshing application.

3.9.2 Services required

I:0097. At present, the operational characteristics of the LoadSense application have not been agreed with WPD; consequently, this section will be updated in the future once the requirements have been determined.

3.9.3 Products required

M:0044. This specification requires, at a minimum:

- Provision of an application capable of utilising the outputs from the applications listed below in combination with decision processes agreed with WPD to determine if initiating a network meshing event is appropriate; and
- If such an event is required, the application must instruct the ALVIN Reclose™ interface application to commence network meshing procedures.

M:0045. The application must also determine when it is appropriate to de-mesh the networks, again based on decision processes agreed with WPD, and instruct the ALVIN Reclose™ interface application accordingly.

3.9.4 Dependencies

I:0098. Input is required from the below applications:

- Load profile predictor
- WeatherSense
- Peer-to-Peer communications

3.9.5 Performance measurement

M:0046. The application must be demonstrated to arrive at the correct decision given specific inputs and initiate the appropriate action of the ALVIN Reclose™ interface application as a result.

I:0099. In the event that ALVIN Reclose™ devices are installed as part of the project trials, it is possible to determine the condition of the device (i.e. circuit open or closed) from the visual indicator on the front.

3.10 Dynamic Thermal Ratings Application

3.10.1 Overall statement

The Dynamic Thermal Ratings (DTR) Application utilises the current transformer temperature, along with the forecast load profile from the Load Profile Predictor Application, to determine the temperature of the transformer asset being monitored over the next 24 hours.

3.10.2 Services required

M:0047. The DTR Application must determine up-to-date thermal ratings for the associated transformer.

M:0048. Based on the forecast load profiles generated by the Load Profile Predictor Application the DTR Application must determine the forecast temperature profile for the transformer.

I:00100. In both instances, this information must be output to the main LV-CAP™ platform.

3.10.3 Products required

I:00101. This specification requires, at a minimum, provision of an application, compatible with the LV-CAP™ platform, containing transformer DTR algorithms.

3.10.4 Dependencies

I:00102. The DTR application requires load data, temperature data and load profile predictions from the LV-CAP™ platform in order to operate.

3.10.5 Performance measurement

M:0049. The DTR application must generate outputs once each subsequent predicted load profile is available, based on that profile.

M:0050. Therefore, the DTR application must output a thermal rating forecast at hourly intervals for the next 24-hour period.

3.11 Centralised Systems

3.11.1 Overall Statement

I:00103. Within the OpenLV Project there is a requirement for two ‘centralised’ systems to enable management of the trial platforms and delivery of the project requirements.

I:00104. The trial system utilises a Nortech iHost server to manage the deployed hardware and store the gathered and processed data.

I:00105. A second server, to be provided by Lucy Electric will store the data gathered by the platform, and processed data generated by applications deployed under Methods 2 and 3.

I:00106. In both cases, separate communication applications are required although both will utilise the 4G modem within the ISDs.

3.11.2 Security

I:00107. In all cases, both for the iHost based control server and the Lucy Electric cloud based server, security must be paramount in keeping with the OpenLV Project’s Data Protection Strategy.

M:0051. User authentication via unique login and password must be enabled.

S:004. Two-factor authentication should be utilised wherever possible.

M:0052. Mutual authentication must occur for all communication between platforms.

I:00108. An independent cyber-security evaluation of the LV-CAP™ platform and associated control systems will be undertaken as part of the OpenLV Project.

I:00109. The system must implement any recommendations from this evaluation to ensure the safety of WPD’s network assets.

3.11.3 Application deployment and management server (Nortech)

Services required

I:00110. The OpenLV Project’s deployed ISDs require a central management and control system, this is provided by a Nortech iHost server.

M:0053. This system must be capable of deploying a new application container to a single device, a subset of devices, or all devices.

M:0054. This system must be capable of removing an application container from a device, a subset of devices, or all devices.

M:0055. This system must be capable of updating the application containers on a device, a subset of devices, or all devices.

M:0056. This system must be capable of changing configuration settings for any individual container on a specific device, a subset of devices, or all devices.

S:005. Identifying when a deployed platform, that has not been decommissioned, has not connected to the server for more than one (1) day, three (3) days and five (5) days, and trigger notification alerts in each instance.

M:0057. The Nortech Comms Application and the iHost server must mutually authenticate each other so that only authorised data uploads occur, and Man-in-the-Middle attacks are prevented. (This requirement is linked with I:00125.)

M:0058. Measures must be taken to ensure that the data uploaded remains confidential in transit, to comply with the OpenLV Project Data Protection Strategy. (This requirement is linked with I:00126.)

Products required

I:00111. Nortech's iHost server is utilised as the central command and control system for the LV-CAP™ platforms.

M:0059. The iHost server for the OpenLV Project must be installed behind a firewall to restrict unauthorised access as far as reasonably practicable.

Dependencies

I:00112. The server is installed at EA Technology and requires access to communications outside of the EA Technology corporate network to enable communications with the deployed LV-CAP™ platforms.

I:00113. The server requires each deployed platform to have a functional router modem and Nortech Communications application as defined in this document.

Performance measurement

M:0060. The iHost server must demonstrate the ability to receive all data uploaded from each connected LV-CAP™ platform.

M:0061. The platform must demonstrate the ability to deploy a new application container to a connected LV-CAP™ platform;

M:0062. The platform must demonstrate the ability to update an application container on a connected LV-CAP™ platform;

M:0063. The platform must demonstrate the ability to change configuration files for a software container on a connected LV-CAP™ platform;

M:0064. The platform must demonstrate the ability to remove an application container from a connected LV-CAP™ platform;

3.11.4 Cloud Based Hosted Platform (Lucy)

Services required

I:00114. The OpenLV Project's deployed ISDs require a public facing data management system to enable community groups and third-party companies access to network data, and outputs generated by their own applications.

M:0065. The Cloud Based Hosted Platform system must be capable of receiving data from the Data Upload Application installed on each LV-CAP™ enabled device deployed within the project.

M:0066. The Cloud Based Hosted Platform system must be capable of sharing this data with appropriate individuals via an API interface.

M:0067. The Cloud Based Hosted Platform system must be capable of sharing this data with appropriate individuals via a web-portal viewer interface.

M:0068. The (Lucy Electric) Data Upload Application and the associated Cloud Based Server must mutually authenticate each other so that only authorised data uploads occur, and Man-in-the-Middle attacks are prevented.

M:0069. Measures must be taken to ensure that the data uploaded remains confidential in transit, to comply with the OpenLV Project Data Protection Strategy.

Products required

M:0070. Lucy Electric to provide a separate, instance of their cloud based data server for use by the OpenLV Project.

Dependencies

I:00115. The server requires each deployed platform to have a functional router modem and the Lucy Electric Communications application as defined later in this document.

Performance measurement

I:00116. The platform must demonstrate the ability to receive a selected subset of data from each connected LV-CAP™ platform;

I:00117. The platform must demonstrate the ability to allow authorised individuals to access the information stored within the server, on a location (LV-CAP™ platform) basis.

3.12 Communications

3.12.1 Overall Statement

I:00118. The LV-CAP™ platform, as being deployed as part of the OpenLV Project, requires three separate communication applications, each to meet specific communication requirements for project delivery.

I:00119. It is necessary for each platform to have the capability to communicate with:

- The application deployment and management server;
- Cloud based, public facing data storage server; and
- Adjacent LV-CAP™ platforms for data sharing purposes.

3.12.2 Security

I:00120. In all cases, both for the iHost based control server and the Lucy Electric cloud based server, security must be paramount in keeping with the OpenLV Project's Data Protection Strategy.

I:00121. User authentication via unique login and password must be enabled.

I:00122. Mutual authentication must occur for all communication between platforms.

3.12.3 Management Comms Application

I:00123. The Nortech communications container is considered a core-element of the LV-CAP™ platform as Nortech's iHost server is utilised to manage and control all LV-CAP™ platform's (ISD's) deployed within the OpenLV Project.

Services required

M:0071. The application container must facilitate two-way communication between the LV-CAP™ platform and the iHost server.

I:00124. This must enable transfer of all desired data from the platform back to the iHost server. This data may be all monitored and calculated values or a selected subset thereof. In either case, the container must be capable of transferring the desired data.

M:0072. The data from each LV-CAP™ system running the Nortech Comms Application must be uploaded as a separate RTU (or multiple virtual RTUs) within the iHost server.

M:0073. Once successfully uploaded to the iHost server, data must be marked as 'uploaded' within the LV-CAP™ platform to prevent retransmission.

M:0074. The application must receive and implement new application containers for installation onto the LV-CAP™ platform.

M:0075. The application must receive and implement configuration files for the installed applications.

M:0076. The application must receive and implement instructions to remove application containers from the LV-CAP™ platform.

M:0077. The application must be capable of managing loss of communications during file upload and download, resuming once communications are restored.

I:00125. The Nortech Comms Application and the iHost server must mutually authenticate each other so that only authorised data uploads occur, and Man-in-the-Middle attacks are prevented. (This is linked with S:005.)

I:00126. Measures must be taken to ensure that the data uploaded remains confidential in transit, to comply with the OpenLV Project Data Protection Strategy. (This is linked with M:0058.)

I:00127. The volume of mobile data transferred must be managed to reduce the operating costs of the OpenLV system.

I:00128. The application must be configured via the standard LV-CAP™ configuration mechanism (see sections 8.2.1 and 9.5 of the LV-CAP™ API). The configuration is likely to be altered in the course of the OpenLV Trials, so the configuration settings available must be documented alongside the Application.

I:00129. The configuration is expected to cover the following areas:

- iHost server settings (included where to send the data, and authentication settings).
- Data Selection settings, i.e. which topics are to be uploaded to the iHost server.
- (Optionally) Where data is to be placed in the iHost structure.

I:00130. The requirements document provided to Nortech for this application container is located in Appendix B – Nortech Application Container.

I:00131. As part of the OpenLV Project, a Cyber-Security review of the LV-CAP™ platform and Applications deployed within the project is to be undertaken. The Cyber-Security supplier will be undertaking an audit of the LV-CAP™ platform and it should be expected that this will include an audit of the software Application and associated documentation created by Nortech as part of the project.

Products required

I:00132. This specification requires, at a minimum, a software container to manage the communication link between the LV-CAP™ platforms and the central iHost server in line with the required services above.

Dependencies

I:00133. The application will require access to:

- the connected router modem;
- the data stored on the platform.

Performance measurement

M:0078. The application must enable communications between an individual LV-CAP™ platform and the iHost command and control server.

I:00134. The application must demonstrate transfer of all data stored on the platform to the server.

I:00135. The application must demonstrate receipt and installation of a new application container.

I:00136. The application must demonstrate receipt and application of a revised configuration set for an application container.

I:00137. The application must demonstrate removal of an application container.

I:00138. The application must demonstrate ability to resume a download when communications are restored.

3.12.4 Data Upload Application

Services required

M:0079. This must enable transfer of all desired data from the platform back to Lucy Electric's cloud based data centre. This data may be all monitored and calculated values or a selected subset thereof. In either case, the container must be capable of transferring the desired data.

M:0080. The data from each LV-CAP system must be uploaded as a separate RTU (or multiple virtual RTUs) within the server.

M:0081. Once successfully uploaded to the server, data must be marked as 'uploaded' to prevent retransmission.

I:00139. The application must be capable of being configured via instructions received from the iHost platform control server.

M:0082. The GridKey Upload Container and the GridKey Data Centre must mutually authenticate each other so that only authorised data uploads occur, and Man-in-the-Middle attacks are prevented.

M:0083. Measures must be taken to ensure that the data uploaded remains confidential in transit, to comply with the OpenLV Project Data Protection Strategy.

I:00140. The requirements for the GridKey Data Upload container, to be provided by Lucy Electric are defined in a separate document located in Appendix C – Lucy Electric Application Container.

I:00141. As part of the OpenLV Project, a Cyber-Security review of the LV-CAP™ platform and containers deployed within the project is to be undertaken. The Cyber-Security supplier will be undertaking an audit of the LV-CAP™ platform and it should be expected that this will include an audit of the software container and associated documentation created by Lucy Electric as part of the project.

Products required

I:00142. This specification requires, at a minimum, a software container to manage the communication link between the LV-CAP™ platforms and the GridKey Data Centre in line with the required services above.

Dependencies

I:00143. The application will require access to:

- the connected router modem;
- the data stored on the platform.

Performance measurement

M:0084. The application must enable communications between an individual LV-CAP™ platform and the GridKey Data Centre.

M:0085. The application must demonstrate transfer of selected data stored on the platform to the server.

M:0086. The application must demonstrate ability to resume a download when communications are restored.

3.12.5 Peer to Peer Comms Application

Services required

I:00144. The LoadSense application in each ISD requires data relating to the status and operation of the linked transformer in order to ensure safe and effective operation.

I:00145. This data must include voltage and current, and the outputs from the load predictor and WeatherSense applications.

M:0087. This application must enable the transfer of the necessary data between the linked, adjacent devices.

M:0088. The data to be transferred between devices may change over the course of the project and consequently, the data must be configurable.

Products required

I:00146. This specification requires, at a minimum, an application to enable the transfer of information to allow the decision of whether to initiate network meshing. It must also be able to respond to equivalent requests.

Dependencies

I:00147. The application will require access to:

- the connected router modem;
- the data stored on the platform.

Performance measurement

I:00148. The Peer-to-Peer Communications application must demonstrate the ability to send and received the configured datasets with the assigned ‘partner’ LV-CAP™ platform.

3.13 Overall System

3.13.1 Overall Statement

I:00149. A magnetic mounting arrangement is preferred by the client DNO with wall or floor mounting acceptable as an alternative.

M:0089. Therefore, the enclosure must be capable of multiple mounting arrangements, including magnetic attachments, wall mounting bolts or ground placement.

3.13.2 Loss of Power

S:006. In the event of loss of power, the platform must, on completion of a successful reboot, determine from data logs how long it was offline, and consequently how much data has been lost.

This information must be stored within the system log files and must include:

- ISD serial number;
- Location;
- Time of last successful data record;
- Time of successful system restoration.

A notification should be issued to the OpenLV Project team, either via the iHost server or direct notification such as an e-mail.

M:0090. The ISD must be capable of self-restoration following a loss of power during boot-up-sequence.

S:007. The ISD is ideally required to withstand up to three loss-of power events within a period of five (5) minutes without suffering unrecoverable errors.

S:008. The ISD should respond appropriately to a loss of power during download of software or configuration updates, ensuring that the download is resumed / restarted and completed once the system is running.

W:008. The ISD must respond appropriately to a loss of power during an update procedure to the LV-CAP™ platform.

I:00150. The system should complete the process with the update / setting changes applied.

3.13.3 Controlled Access

M:0091. The ISD enclosure must be capable of being securely locked with a padlock.

I:00151. WPD will provide padlocks to restrict access only to those staff competent and authorised for LV Switching operations.

S:009. Access to the system software through direct ethernet connection must be restricted through methods such as digital signing, communication encryption and require a password to access the device.

3.13.4 Network deployment

M:0092. The cable connections (power, thermocouple and communications) must be suitable for implementation on WPD's network.

4 Appendix A – LV-CAP™ API

LV Common Application Platform Public API

UNRESTRICTED

Product: LV-CAP
Drawing: 2383-MANUL-V04.03.00
Date: July 2017

Version History

Date	Version	Author(s)	Notes
2015/11/06	1	James Slater/Siôn Hughes	First version of Third Party Developer API document
2015/11/12	2	James Slater/Siôn Hughes	Amendments made to first release of document
2017/07/26	04.03.00	Richard Ash, James Slater	Re-organise to reduce duplication, incorporate changes to meet 2362-RQSPC. Add MQTT security information and error reporting API. Updates from meeting with Nortech. Change Application identifiers and terminology, add concept of Priority for messages (future proofing). New release for OpenLV project.

CONFIDENTIAL - This document may not be disclosed to any person other than the addressee or any duly authorised person within the addressee's company or organisation and may only be disclosed so far as is strictly necessary for the proper purposes of the addressee which may be limited by contract. Any person to whom the document or any part of it is disclosed must comply with this notice. A failure to comply with it may result in loss or damage to EA Technology Ltd or to others with whom it may have contracted and the addressee will be held fully liable therefor.

Care has been taken in the preparation of this Report, but all advice, analysis, calculations, information, forecasts and recommendations are supplied for the assistance of the relevant client and are not to be relied on as authoritative or as in substitution for the exercise of judgement by that client or any other reader. EA Technology Ltd. nor any of its personnel engaged in the preparation of this Report shall have any liability whatsoever for any direct or consequential loss arising from use of this Report or its contents and give no warranty or representation (express or implied) as to the quality or fitness for the purpose of any process, material, product or system referred to in the report.

EA Technology Limited, Capenhurst Technology Park, Capenhurst, Chester, CH1 6ES;
Tel: 0151 339 4181 Fax: 0151 347 2404

http://www.eatechnology.com

Registered in England number 2566313

Contents

Figures

Tables

1. Introduction

The Common Application Platform for LV Networks (LV-CAP) is a software environment which facilitates the implementation of the Smart Grid at the lower distribution voltages. To drive down the cost of deploying Smart Interventions, the platform allows multiple algorithms to be deployed to one set of measurement and data processing hardware. The platform allows these algorithms to be designed and produced by independent third-party developers and packaged as stand-alone Applications which can be easily deployed by the distribution network operator without requiring bespoke software development.

This document details the Application Programming Interface (API) for developers intending to write Applications to run on LV-CAP. LV-CAP uses Docker to overcome dependency problems for third party developers, and helps to maintain and manage containers. It uses a MQTT messaging system for the communication of running containers and has a data storage functionality to persist data. This document has details on how a third-party Application can be set-up, run and interact with the core services on the platform.

2. Glossary

Term	Description
ACL	Access Control List, a list of the resources which a specific client may access. Used to control access to topics on the **MQTT broker**.
API	Application Programming Interface – a set of defined interfaces to be used by application developers.
APID	See **Application ID**.
Application	A **Docker Container** suitable for use with LV-CAP in accordance with this API document. All LV-CAP **Applications** are Docker Containers, but not all Docker Containers are suitable for use as LV-CAP Applications.
Application ID	The unique identifier for a specific version of an Application, by combining the **Vendor, Application Name** and **Application Version**. See Section 4.2.
Application Name	This is a string which identifies an **Application**. This is chosen at will by the Application developer. See Section 4.2.
Application Version	A string which indicates the version of **Application** in a **Docker Image**. Decimal points may be used to separate version numbers, e.g. 1.2.3. This is chosen by the Application developer. See Section 4.2.
BLOB	Binary Large OBject, a SQL database field which can store an arbitrary array of binary data.
Container Manager	The main process which controls all LV-CAP **Applications**.
Docker	Open source program that allows Linux applications and their dependencies to be packaged as a **Docker Image**.
Docker Container	An isolated environment in which a **Docker Image** is run under **Docker**. Multiple Docker Containers may be created from a single Docker Image and run simultaneously.

Term	Description
Docker Image	A file system image containing a packaged Linux program (with its dependencies) which can be deployed to run on a **Docker** system.
GUID/UUID	A Version 4 GUID/UUID is a universally unique 48-byte identifier which is generated using random numbers. Example:- 821b8e33-4eaa-480e-b205-30fa9572af1a
IID	See Instance ID
Instance	One running copy of an **Application**, which is separate from any other copy of the Application, and has its own independent configuration. See Section 4.2.
Instance ID	String identifying a specific **Instance** of an **Application**, which is unique only on a given LV-CAP system. See Section 4.2.
LV	Low Voltage. Used in this context to refer to the Low Voltage electricity distribution network which delivers power to domestic and commercial customers at 400/230V AC.
LWT	Last Will and Testament, in **MQTT** a message to be sent when to subscribers when a publisher disconnects unexpectedly.
MQTT	(Message Queue Telemetry Transport) a publish subscribe based lightweight messaging protocol, used on top of the TCP/IP protocol.
MQTT Broker	Process which is responsible for distributing messages to interested clients based on the topic of a message. The LV Common Application Platform runs a private instance of an MQTT Broker
MQTT Topic	Identifier within an **MQTT** message used by the broker to allow filtering and direction of messages. All messages are published to a topic, and clients receive them if they are subscribed to the topic.
Vendor	A string which identifies the developer of an Application. These are allocated by EA Technology to each party creating **Applications** to run on LV-CAP.

Table 1 - Glossary of Terms

3. Platform Overview

The LV Common Application Platform (LV-CAP) provides a framework for measurements to be made, processed through algorithms, and actions taken based on the results (Figure 1). All of these functions may be undertaken by Applications developed by EA Technology or third parties. LV-CAP provides a number of core services for third party Container developers to utilise. These are:

1. Container management (installation, configuration, starting and running of Applications, including multiple copies and versions.).
2. A Data Marketplace which allows all Applications on the platform to communicate with each other in a uniform manner.
3. A Data Storage mechanism which allows Application outputs to be stored for future use.

All other functionality is provided by Applications, but using standard interfaces so that different implementations can be swapped in and out without affecting other Applications. To achieve this Applications do not communicate directly but rather via the Data Marketplace using the messaging API described in Section 8. This is shown in Figure 2.

A key piece of the provided framework is the Container Manager. The Container Manager has ultimate control over the entire system ensuring everything runs as expected. Apart from the Container Manager, all core services on the platform run as Docker containers which the Container Manager is responsible for starting, stopping and updating. All Applications are packaged within Docker containers which the Container Manager will again start, stop and manage. A Docker container contains a GNU/Linux application and all its library dependencies except the Linux kernel itself. This allows each Docker Image created to be portable, easily updated and independent.

The Container Manager utilises the functionality within Docker to limit and share resources of a running container. This control allows the Container Manager to manage platform resources, giving Applications their requested resources and preventing them from consuming excess resources and starving others of resource. Developers of third party Applications must be aware that their application cannot use the entire resources of the system and that it must share processor, RAM and storage with other Applications running on the system.

As well as managing the start-up and shutdown of Applications, the Container Manager is responsible for ensuring that updated configuration files are delivered to the relevant containers, and that updates to containers are applied. Finally, it checks that Applications are still running correctly, handling any errors returned from Applications and dealing with Applications that have ceased to operate correctly.

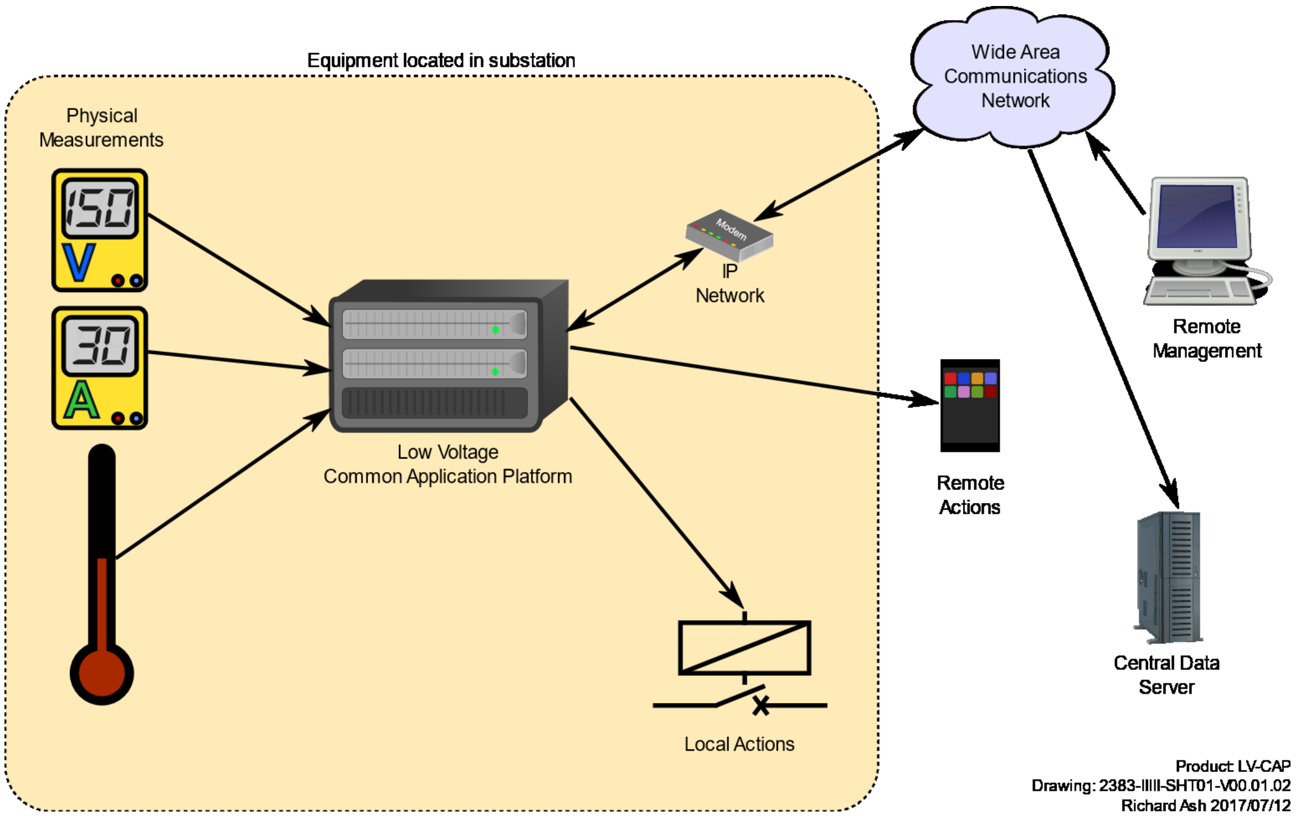


Figure 1 - LV-CAP System Concept

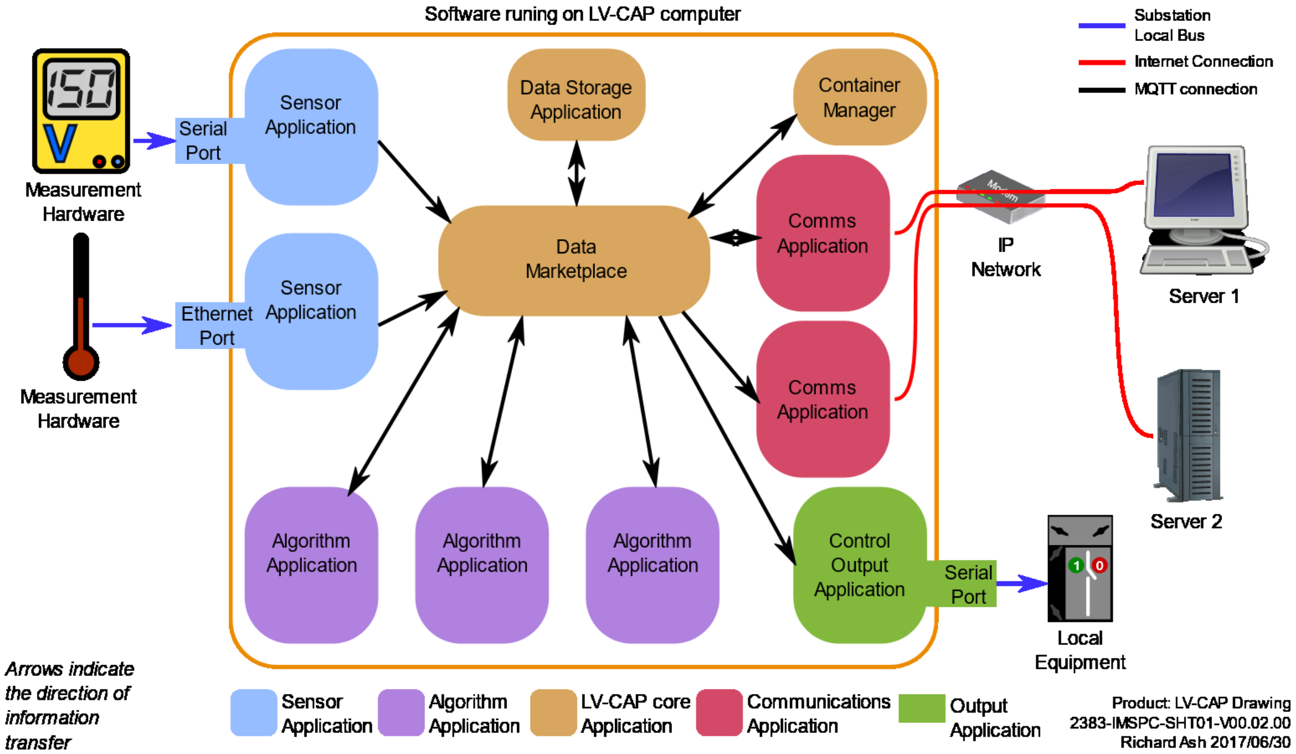


Figure 2 - LV-CAP Software Architecture

All communications between Applications in LV-CAP take place through the Data Marketplace (Figure 2). This uses Message Queue Telemetry Transport (MQTT) to transport messages. An MQTT broker, Mosquitto, is supplied as part of LV-CAP and is used by both core services and third-party containers. The message protocol for communicating on the MQTT broker and connection settings to the broker are documented in Section 8 of this document. Access Control Lists (ACLs) are used on the MQTT broker to secure it, preventing Applications from publishing on and subscribing to topics they should not. The ACLs are automatically configured by LV-CAP when a new Application is added to the system.

LV-CAP systems are configured with an internal IP network. The Data Marketplace operates on this internal network and all Applications are automatically connected to it. Applications are only connected to external networks when there is a clear requirement for such a connection, and the system administrator has permitted it.

The Data Storage Application provides a database connected to the Data Marketplace. As well as being used by the Container Manager, it stores the outputs of Applications so that they can be subsequently retrieved for external communication or further processing.

Applications running on LV-CAP will generally fulfil one of four roles. Some Applications may fulfil more than one role at the same time.

1. Sensor Applications are responsible for reading data from physical sensor hardware. The data read is sanity checked and published to the Data Marketplace in a standard format. The data is then available to any other Application to subscribe to. The set of sensors provided for any given LV-CAP installation, and hence the Sensor Applications required, will vary depending on the user's requirements. The data format is independent of the measurement hardware so that different supplier's hardware can be used without software alterations outside the related Sensor Application.
2. Algorithm Applications consume data from one or more sensors and perform calculations upon it, for instance calculating the real-time temperature of a Transformer or forecasting the localised demand for energy. The Applications read from the Data Marketplace and publish their outputs back to the Data Marketplace.
3. Output Applications are the mirror image of Sensor Applications. They respond to information on the Data Marketplace (created by Algorithm Applications) by controlling physical hardware connected to the LV-CAP system, for instance carrying out network switching or energy storage.
4. Communications Applications connect the LV-CAP platform to the outside world. LV-CAP provides an IP communications link to the outside world, which Communications Applications use to upload and download data. A Communications Application uploads selected data values from the Data Marketplace to a central data server, or downloads Application images and configuration files from a central management server.

The default Communications Application is provided by Nortech Management Ltd. to communicate with their iHost server product.

4. General Principals

This section includes some principals which have driven the design and operation of the LV-CAP system. An understanding of these will make it easier to navigate and comprehend the rest of this specification.

4.1 Architecture

LV-CAP is designed to work as a loosely-coupled data processing pipeline, in which measurement data from Sensor Applications feeds into one or more Algorithm Applications. The outputs of these Algorithm Applications may feed other Algorithm Applications. Ultimately data reaches either a Communications Application to be sent to an external system, or an Output Application to take local actions on the Smart Grid (Figure 3).

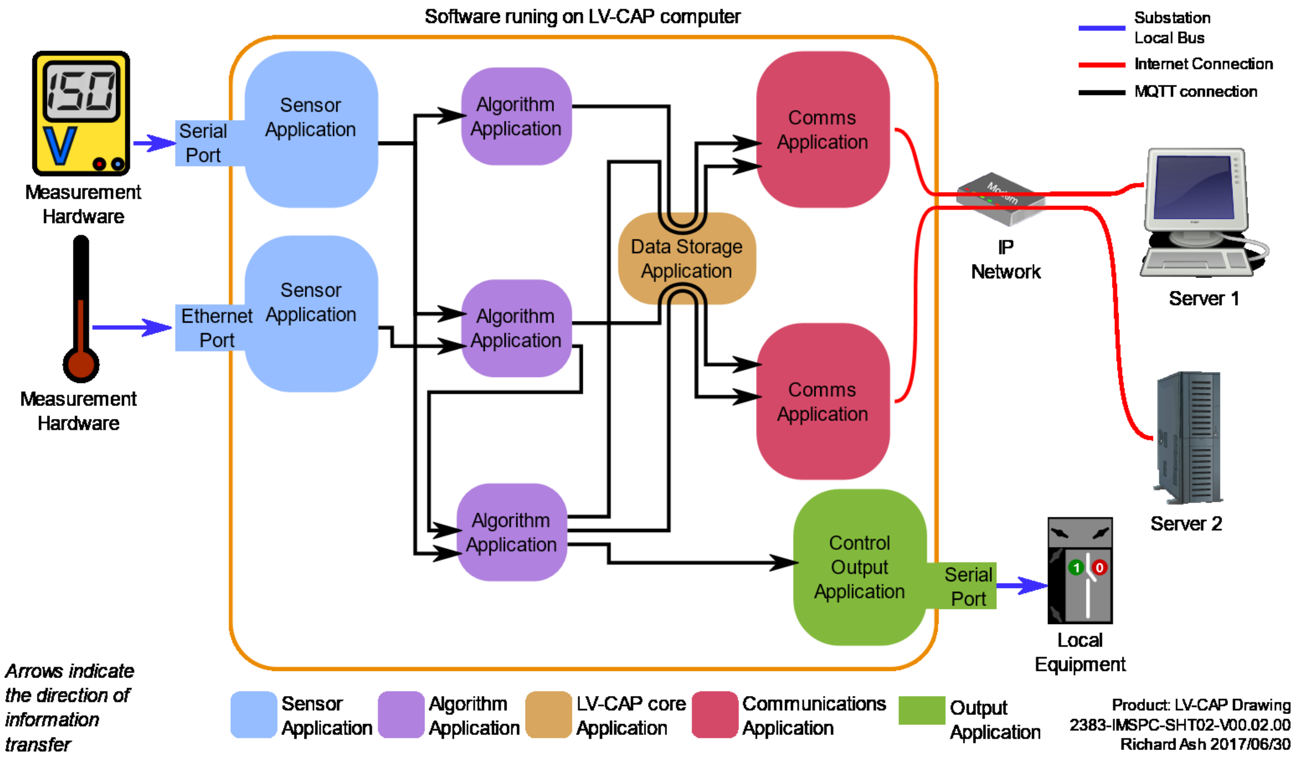


Figure 3 - Data Flow through an example LV-CAP system

Data is pushed through the pipeline from the Sensor Applications towards the outputs. The pipeline runs in approximately real time, although this is not enforced as in a true real-time system. If the workload of the LV-CAP system temporarily exceeds the available processing power then the system will lag behind before catching back up when resources allow.

The pipeline forms a tree structure, with each node being an input or output on a single topic in the Data Marketplace. Any Application may subscribe to any topic to make use of the data found there, with the delivery of messages to the various destinations handled by the MQTT broker. The expectation is that Applications will generally output onto fixed topic names (within their allocated sub-tree), whilst being freely configured (via the Configuration API) to read from whichever input topics the system operator requires.

The MQTT broker forming the Data Marketplace will only buffer a single message on each topic, so Containers must handle their input messages sufficiently quickly to be ready when the next message arrives on a topic. The Data Marketplace does not perform any rate adaption, so Applications need to be prepared to receive their input messages at whatever intervals the upstream Application produces them.

4.2 Application Identification

Applications must have uniquely defined identifiers. These fulfil a number of roles:

- To ensure that Applications will never encounter a name “clash” with another Application.
- To allow multiple copies of the same Application to run simultaneously, with separate configuration settings.
- To allow different versions of the same Application to be installed and run simultaneously.
- To allow system operators to unambiguously specify what Applications are to be run on any given system.

In this section, each word in **bold** is defined in the Glossary at the start of this document. To facilitate the selection and operation of **Application**s, each **Application**'s **Docker Image** has three unique pieces for information associated with it:

1. A **Vendor** string. This is a string which identifies the developer of the **Application.** These are allocated by EA Technology to each party creating **Application**s to run on LV-CAP.
2. An **Application Name** string. This is a string which identifies the **Application.** This is chosen at will by the **Application** developer. It should not container version information.
3. An **Application Version.** This is a string which indicates the version of the **Application.** Decimal points may be used to separate version numbers, e.g. 1.2.3. This is chosen by the **Application** developer.

This information enabled a system operator to specify exactly what **Application** they wish to run on LV-CAP. There are a number of constraints on the above fields which must be satisfied when they are chosen:

- The **Vendor** and **Application Name** must be valid Docker Names (see Reference 4):
  - Composed of valid ASCII characters.
  - Restricted to lower case letters, digits, periods and hyphens (no underscores).
  - May not start with a period or a dash.
- Each release or update of an Application must have a unique combination of **Vendor, Application Name** and **Application Version.**
- For an Application to be successfully updated, the update must have an **Application Version** which Docker considers to be different to the existing Application's **Application Version.**
- For compatibility, the total length of the **Vendor, Application Name** and **Application Version** must be less than 44 characters.

The **Vendor, Application Name** and **Application Version** are combined to form the **Application ID <APID>**. When used as a file name or **Topic Name** then these sections are separated with an underscore:

```
<Vendor>_<Application Name>_<Application Version>
```

When used as a tag for a **Docker Image** then they are combined according to the usual docker convention:

```
<Vendor>/<Application Name>:<Application Version>
```

The <**APID**> identifies a specific Application executable in a globally unique manner. In the future this will be enforced through the digital signing of **Application Images** and their <**APID**>.

When creating the *Docker Image*, these fields are specified to the -t option of the docker build command as follows:

```
docker build -t <Vendor>/<Application Name>:<Application Version>.
```

When an **Application** is to be executed on LV-CAP, a **Docker Container** is created from the **Docker Image.** Each **Docker Container** must have a unique name, and we must support creating multiple Containers from one Image. To enable this a fourth field is used, which is the **Instance.** Instance is a two digit number which is unique to this Instance of an **Application** on the LV-CAP system. **Instance** values are set up in the **Container Manager** configuration by the system operator. The **Instance** value of "00" is special and reserved for use by Applications which cannot have more than one instance running. A single instance of any other Application may use any other value between "01" and "99". There is no requirement for **Instance** numbers to be contiguous, and their numeric value has no significance.

The **Vendor, Application Name** and **Instance** are combined to form the **Instance ID** (abbreviated in this document as <**IID**>) in the form

```
<Vendor>_<Application Name>_<Instance>
```

The <**IID**> identifies a specific instance of an **Application**, which is unique only on a given LV-CAP system, and may use any (specified) version of the **Application**. This is set up through the **Container Manager** configuration file.

Each Application Instance **Docker Container** created on LV-CAP will have the container name in Docker set to the <IID>.

The **Application Version** is deliberately omitted from the **Instance ID** so that the name does not change when newer versions of the **Application** are deployed. The **Instance ID** <IID> of a container is used as its handle, to identify the Container's area of file system space, MQTT topic namespace and so on.

4.2.1 Legacy Applications

Applications developed against older versions of this **API** (and the Innovate UK project) were identified by a single **GUID**. This 48-byte opaque string served as both **Application ID** <APID> (although it lacked version information) and **Instance ID** <IID> (although it lacked instance numbers). Applications using this form can still run one instance, using their legacy identifiers.

For these legacy containers, two instances of the same Container with the same GUID will never be run on the same LV-CAP installation. The GUID of a container was used as its handle, to identify the Container's area of file system space, MQTT topic namespace and so on.

4.3 Message Serialisation

All messages transmitted via the Data Marketplace are serialised in JavaScript Object Notation (JSON). Adding additional white space to JSON payloads to 'pretty print' them is discouraged. All messages sent via the MQTT Broker must be valid JSON.

Standardised JSON object structures are used wherever possible to maximise interoperability. These are defined in Section 9 of this document.

4.4 Topic Names

All messages exchanged through the Data Marketplace are published on MQTT topics. Whilst this API sets out specific topics for some purposes (e.g. interactions with the Container Manager) it is up to Application authors to choose suitable topics (and especially sub-topics) for the messages which their container produces. In order to use the standard JSON object structures defined in Section 9, unchanging information about the value has to be encoded in the topic name, rather than in the JSON payload itself. This also reduces the transmission of redundant (invariant) information where communications bandwidth is limited.

The MQTT standard itself places few restrictions on the choice of topic names, apart those specified in Section 4.7 of the standard. When choosing topic names however, the following guidelines should be borne in mind to make development and administration easier:

- Avoid spaces in topic names, as they are prone to confuse parsers of all sorts.
- Avoid non-ASCII characters in topic names, as they are prone to confuse users or their tools.
- Use topic levels to separate sections of your topic name. E.g. "output/transformer/forecast/4h/capacity" not "output/transformer-forecast_4hCapacity".
- Design in extensibility – it will be disruptive to change existing topic names to allow additional data to be published (e.g. more channels or intermediate calculation results).

To make it easier for system operators to understand what messages on a topic mean, the following general form of topic names is recommended:

<subtree>/<asset>/<parameter group>/<time>/<parameter>

Not all components will be required for a given topic name and may be omitted. This results in topic names like:

algorithm/data/0ca2eadb-b128-4dff-9bd7-cbb15e21b8b1/Number1Tx/state/hst

sensor/data/eatl_modbusrtusensor_01/Number1Tx/load/A

4.5 Units

All messages transmitted should have a timestamp (as shown in the preferred JSON formats in section 9). These timestamps are 64-bit UNIX timestamps, defined as the number of seconds since 1st Jan 1970 UTC. Where sub-second resolution is required, the fractional value should be stored as a separate field.

Wherever possible, Applications should use the time stamp fields from incoming messages in preference to referencing system time (explicitly or implicitly). This will make it much easier to test Applications in a reproducible manner by simply replaying a fixed sequence of input messages, regardless of the relationship between message time stamps and system time.

Values are always given in the base SI unit for the quantity being measured or calculated. For example, current is always given in Amps, never in milliamps or kiloamps. Temperatures are given in degrees Centigrade rather than Kelvin (in accordance with common engineering practice).

Metadata for the display of values may be passed between containers via Data Series Metadata Objects described in Section 9.4.

4.6 Text Encoding

UTF-8 is the preferred method of encoding text.

When including non-English text in JSON strings bear in mind that that the double-quote character must be escaped with a backslash, and other escape sequences are used for newline etc. control characters, as per the JSON specification.

4.7 Data Persistence

Applications have two options for persisting data:

1. Data which is output to the Data Marketplace can be stored in the Data Storage Application (Section 0). Any Application can then retrieve this data in the future (up to a time limit imposed by the removal of old data values).
2. Each Application is assigned a filesystem volume. This file system is private to the Application and not visible to any other container on the system. Data can be stored here by the Application, e.g. to save system state or history.

The rest of the Docker Container environment is ephemeral and will be lost when the Application is re-started, either by the Container Manager or because the whole LV-CAP platform is rebooted.

4.8 Data Flow and Valid Flags

LV-CAP is designed to work on the basis that data keeps flowing through the processing pipeline at all times. To support this, the standardised JSON object structures in Section 9 all contain a Valid key. When correct data is not available or cannot be calculated, Applications should continue to output messages to the Data Marketplace in the normal manner, but with the Valid key set to false. Applications subscribed to the topics will then be made aware that time is moving on, but that there is a problem with the data source.

Containers which fulfil the Sensor Application role (Section 0) should continue to output messages at the configured interval under all circumstances. This includes if the sensors are disconnected or producing out-of-range values. When data is not available or out-of-range, the "Valid" member in the output should be set to false. The actual sensor value sent in this case does not matter, because subscribed containers should not use the value when "Valid" is false. The timestamp field must be updated so that the subscribed containers can keep track of time.

Algorithm containers receiving input messages with Valid set to false should not use the Value in the received message, but may rely upon the time stamps. When the input timestamps reach the point that the algorithm is due to provide output, it must do so. It is up to the Application author to decide if there is sufficient Valid data to produce an output or not. If there is insufficient Valid data to produce a new result then the container should output, setting Valid to false and using the timestamp from the most recent input message (whether that message is valid or not).

4.9 Data Priority

The standard JSON formats described in Section 9 provide for a Priority field. This allows the upload of certain messages to be prioritised by Data Upload Applications, based on policy set by the system operator and priority information from Application authors.

Valid Priority values are integers between 1 and 5. A Priority value of 1 is the highest priority and 5 is the lowest priority. Any other Priority value is not valid and is treated the same as if Priority is not specified. These messages with no specified Priority have lower Priority than all messages with a specified Priority.

The data APIs in sections 8.4 and 8.5 support query by Priority.

5. Start-up Procedures

5.1 LV-CAP System Start

5.1.1 Start-up of Core Services

As discussed in section 0, the LV Common Application Framework consists of a number of core services for third party containers to use. These have a defined start-up order and only once these have all started up will any other container be started. The Core Services are started in the following order:

1. Container Manager
2. Data Marketplace
3. Data Storage Application

5.1.2 Start-up of Remaining Applications

The remaining Applications installed on a given LV Common Application Platform will be started automatically, once the platform's framework has successfully started up and entered the running state. All other Applications must be independent of each other (there is no concept of inter-Application dependencies) so that Applications can start in any order. Applications will be started by the Container Manager in order of their installation date (i.e. order of when they were added to Docker's available image list).

5.2 Application Start

Each Application on the LV Common Application Platform must perform certain actions when it is started by the Container Manager. Failure to do so is likely to result in the Application being shut down by the Container Manager.

Upon starting, a third-party Application must perform the following actions in the given order:

1. Connect to the Data Marketplace (see Section 8.1 for the MQTT Broker connection details).
2. Subscribe to topics listed in Table 2
3. Send a configuration request message to the Container Manager via the Data Marketplace.
4. Wait until a response is sent back by the Container Manager. This response will either contain the Application's configuration, or will include an error message if the Container Manager is not aware of any configuration for the Application.
5. If configuration data is received, the Application should process the configuration and apply it internally.
6. If the configuration is valid, the Application can start operating, sending a status update to the Container Manager indicating all is OK.
7. If no configuration is available or there is an error in the configuration, the Application must send a status update to the Container Manager indicating an error:
    - Sending STATUS_INITIAL will result in the Container Manager re-sending the configuration file, the Application should stay in a non-operating state awaiting configuration.
    - Sending STATUS_ERROR and will cause the Container Manager to restart the Application. See Section 8.2.2 for more information on status messages.

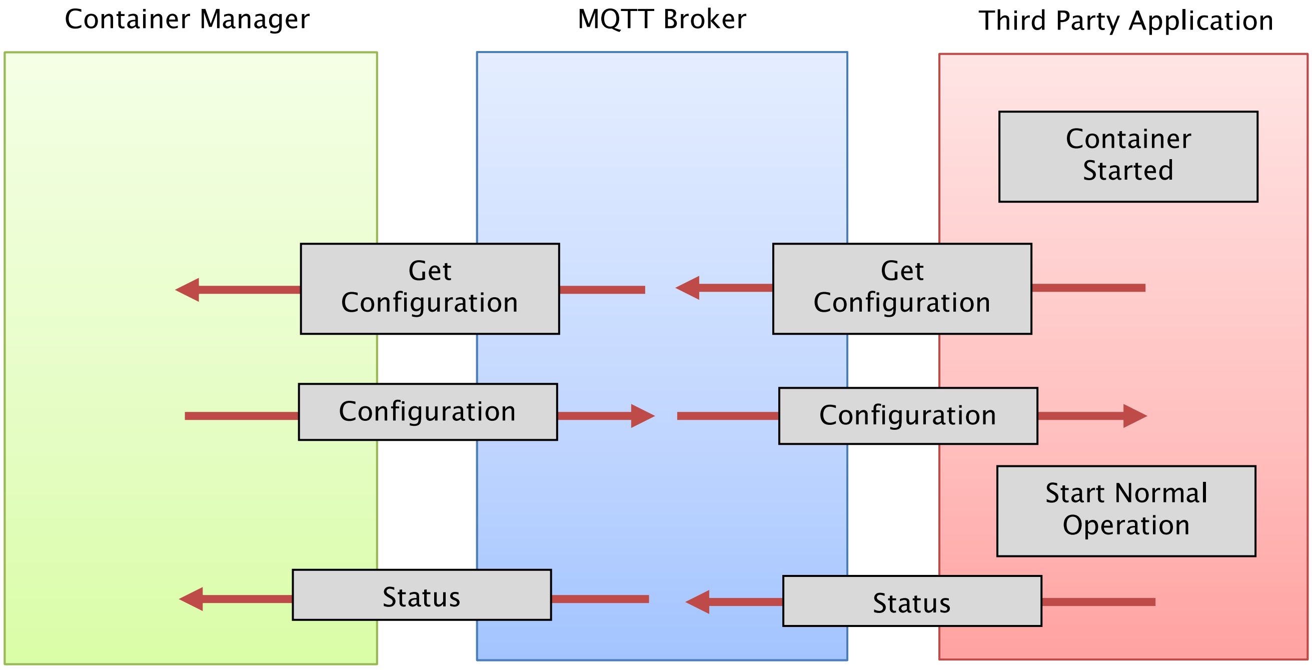


Figure 4 - Application start-up procedure

The above diagram shows the Application start procedure for a successful start.

Due to the Application start order (Section 5.1.2) it is possible that once a container starts its ‘normal operation’, other Applications it may want to communicate with may not yet be operating. In this situation the Application will have to wait until any dependencies are running. This is not normally a problem because the MQTT protocol allows publishing and subscription to occur in any order, with no requirement for topics to be configured or created in advance.

5.3 Required Subscriptions for all Applications

All Applications must subscribe to the following MQTT message topics in order to interact correctly with the Container Manager and remain running on the platform.

Topic	Purpose
status/request	Receives status requests from the Container Manager. Non-response to two consecutive status requests will lead to the Application being restarted without notice by the Container Manager.
config/response/<IID>	Receives Application configuration sent by the Container Manager. The IID is the Application's own unique IID as in section 4.2
command/<IID>	Integer, the command to execute.

Table 2 - Required Subscriptions by Third Party Applications

6. Shutdown Procedure

Similar to the start-up procedure (Section 5), the LV Common Application Platform has a defined shutdown procedure. This shutdown procedure is designed to allow Applications to shut down in a safe manner and avoid any data loss or corruption. The shutdown procedure not only applies to shutting down of the entire platform, but also occurs when an updated Application image is applied. Applications can request their own shutdown if required, but all Applications shall respond to a shutdown request from LV-CAP.

When LV-CAP requests shutdown of an Application, the procedure is defined as:

1. Container Manager sends a shutdown command to the Application via the MQTT broker (see section 8.2.3).
2. The Application handles the notification and performs its own internal shutdown procedure, which may include writing any data to disk, stopping all MQTT subscriptions including that of status requests, and any other work to perform a clean shutdown. Once completed, the Application must respond to the Container Manager using a *status/response* message with the STATUS_SHUT_DWN status.
3. The Container Manager will then shut down the container. If the Container Manager does not receive a status message from the Application to be shutdown which includes the STATUS_SHUT_DWN status for more than 1 minute, the container will automatically be shut down.

If the Application fails to shut down within the "status/request" (default 2 minutes) interval then Container Manager will shut it down forcibly by terminating the process.

In the event of an Application requesting its own shutdown by the Container Manager, the procedure is defined as:

1. The Application performs its own internal shutdown procedure, which may include writing any data to disk, stopping all MQTT subscriptions including that of status requests, and any other work to perform a clean shutdown.
2. Once completed, the Application must send a status to the Container Manager using a *status/response* message with the STATUS_SHUT_DWN status.
3. The Container Manager will then shut down the Application, and added to the stopped Application list. It will not be run again until the Container Manager configuration is altered or the Container Manager is re-started.

If an Application needs to be re-started by the Container Manager, the procedure is:

1. The Application prepares for being restarted, which may include writing any data to disk, stopping all MQTT subscriptions including that of status requests, and any other work to perform a clean restart.
2. Once completed, the Application must send a status to the Container Manager using a *status/response* message with the STATUS_RESTART status.
3. The Container Manager will then shut down the Application and start it back up again immediately.

7. Data Storage

The Data Storage Application allows persistence of data from Applications. Data stored in the Data Storage Application can also be configured by the system administrator to be uploaded by one or more Communications Applications. In this role, the Data Storage Application acts as a buffer so that data is uploaded to its destination reliably, even in the face of unreliable communications links.

The data stored on the platform will be placed in a database which can be accessed via the Data Marketplace (see section 8.3). The output of each Application Instance <IID> will be stored in its own table. This table is created when the Application Instance is first created by the Container Manager. All tables created for Applications will store records with the format documented in Table 3.

Field	Type	Description
ID	Opaque Integer	Each record stored will be assigned an ID by the Data Storage Application. This integer will be unique amongst the records currently stored in the Data Storage Application, but may be re-used over the life of the LV-CAP system as old data is purged from the database and new records added. There are no guarantees about the numeric value of this identifier.
Timestamp	Integer	The Unix timestamp at which the record was added to the Data Storage Container.
SubTopic	String	Part of the MQTT topic string on the Data Marketplace from which the record came will be stored in this field. Because tables are allocated by the Application Instance ID, the topic string up to the IID would be the same for all records. The common string is not stored, leaving only the Application's sub-topic string. If no sub topics are present this field will be null.
Data	BLOB	This is the MQTT JSON message sent to be stored. It is stored as a BLOB so that no alterations are made to the JSON object.

Table 3 - Containers Table schema

The payload an Application sends to be stored will retrieved unaltered from the Data Storage Application. The Data Storage Application will set the values of the other fields automatically. The API for retrieving records is documented in Section 8.5. Applications are strongly encouraged to output their data in one of the standard JSON payload formats described in Section 9.

8. Data Marketplace API

The main form of communication between Applications and the LV Common Application Framework is via the Data Marketplace. This section documents the MQTT message topics, their associated payload and includes examples of message payload. The API is broken up into sections according to the Application roles (Section 0) expected to use them. Application may (and will) use methods from more than one section of the API.

8.1 MQTT Broker

Each container wishing to operate on the LV Common Application Framework must connect and communicate using the provided MQTT broker. The LV-CAP system uses a secured MQTT broker, in order to support authentication of Application when they connect to the Data Marketplace. The connections settings required are shown in Table 4.

Setting	Value
Hostname	marketplace
Port Number	8883
Encryption	TLS v1.2 or higher
Authentication	X509 client certificate
Username	Set to the Application's Application ID
Client ID	Set to the Application's Application ID

Table 4 – Secured MQTT Broker Settings

EA Technology will operate a TLS Certificate Authority for the LV-CAP system. All client SSL certificates must be signed by this Certificate Authority, which will be trusted by the Data Marketplace. This Certificate Authority certificate will be issued to Application developers for inclusion in Application at build time, so they can authenticate the Data Marketplace.

Client certificates will be signed on request by the certificate authority, with the Common Name (CN) of the certificate set to the Application ID <APID> of the Application they are to be used by (see Section 4.2). This client certificate should be embedded in the Application so that it can be used to connect to the Data Marketplace. The client certificate and associated private key need to be embedded in the Application so that it can connect to the Marketplace. The private key should be encrypted to minimise the risk of it being extracted from the Application by a third party. These is no reason for EA Technology, or any other Application author, to know the Application's private key.

When the Application connects to the Data Marketplace it's certificate will be checked. If valid, and not revoked by the system operator, it will be allowed to connect. Access control lists will then allow the Application to publish on the topics set out in this API. In general, subscriptions will not be restricted.

A new certificate should be obtained whenever an updated version of the Application is produced. This both mitigates the fixed expiry date of certificates, and allows the certificate of specific Application versions to be revoked if the keys are compromised. This will also have to be done when an updated Certificate Authority Root Certificate is required.

8.1.1 Payload Descriptions

JSON does not have a concept of fixed-size (bit width) integers, however implementation in strongly typed languages is made much easier by defining the maximum size of integer fields wherever possible. In this documentation:

- Any key which is shown with type “Integer” will always fit into a 32-bit signed integer.
- Any key which is shown with type “Int64” will always fit into a 64-bit signed integer.

8.1.2 Security and Signing

The present implementation of LV-CAP provides only limited security between Applications, and so requires a high degree of trust in Application authors. To improve this situation in the future, an optional “signature” object has been added to all JSON payloads specified in this API. This member is reserved for the definition (in a future version of this API) of a mechanism for cryptographically signing each JSON payload.

The signing scheme is intended to use public (asymmetric) key cryptography. The source Application will sign all outgoing messages with a private key, which must be kept secret. Destination Applications receiving these messages can use the source Application’s public key (which does have to be kept secret) to verify that the messages received are indeed from the correct source container. It is intended that the public keys will be distributed to the relevant Applications via their configuration data.

A Application which does not implement signature verification will be able to receive future signed messages without modification, because it will ignore the signature object. Applications with signature validation implemented will have to decide on their policy for messages received without signatures.

At some future date, it may become mandatory to sign messages on some critical API topics when communicating with the LV-CAP core components. It will be up to other Application authors at what point they require signed input messages.

The signing of Docker Images will also be added in future to ensure that when system operators specify a particular Application ID <APID> (see section 4.2) only that specific version can be run.

8.1.3 Last Will and Testament

The MQTT broker supplied by the framework supports the Last Will and Testament (LWT) feature. This can be used to define, upon connection, a message which will automatically be sent by the broker to subscribers of the set topic upon the non-clean disconnection of a client. In order to manage the platform all Applications must provide a LWT on their status response topic (Section 8.2.2). The status response sent as the LWT must include the FAILED state within the payload. Applications may also set LWT’s on any topic they desire to inform others of their failed state.

8.1.4 Quality of Service

MQTT provides a Quality of Service (QoS) level feature, which defines how hard a broker or client will work to ensure a message is delivered. More details can be found in section 4.3 of the MQTT Standard.

MQTT QoS is a property of both the publishing and subscribing of a message, so a client can publish a message at any QoS and a client may subscribe to a topic at any QoS. The implemented QoS will be the lowest of the publishing and subscribing QoS levels. There are 3 QoS levels defined in MQTT:

- QoS 0 - At most once. The status request topic has a QoS of 0 as this regular heartbeat is not critical, and must be sent regularly.
- QoS 1 - At least once. The Container Manager sends out commands at QoS 1 as Containers can easily handle receiving the same command more than once.
- QoS 2 - Exactly once. This is used when querying the Data Storage Container, as multiple message delivery could have complex and undesirable affects upon the database.

8.2 LV-CAP Core API

The Core API is responsible for management of Applications. All Applications will need to use the Core API to register with and run on LV-CAP.

8.2.1 Configuration

The Configuration message topic is used to request and distribute configuration to Applications.

QoS: Messages on this these topics must be sent and received with QoS = 1. Applications must cope with multiple copies of their configuration information being delivered.

Retention: Messages sent on these topics must have the retention flag set to false.

Topic	Description	Sender	Receiver	Payload	Notes

config/request/
<IID>
	Message containing a request from a container to the Container Manager requesting it’s configuration	Any Application	Container Manager	`{`
				`    "Timestamp": <Int64>,`
				`    "Signature": {}`
				`}`
					No required payload.
					Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the configuration was requested. Required in signed payloads to protect against replay attacks.
					Signature: (Reserved) See Section 8.1.2.

Topic	Description	Sender	Receiver	Payload	Notes

config/response/
<IID>
	Message containing updated configuration for a specific Application Instance. Can be a response to a request, or a new set of configuration pushed to a Application Instance.	Container Manager	Application Instance with IID specified in topic name	`{`
				`"Configuration":`
				`    {`
				`      "<Key_1>": <Value_1>,`
				`      "<Key_2>": <Value_2>,`
				`      "<Key_n>": <Value_n>`
				`    },`
				`    "Timestamp": <Int64>,`
				`    "Signature": {}`
				`}`
					Configuration: JSON Object read directly from the Application Instance configuration file. The structure will be different for each Application, as described in Section 9.5.
					Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the configuration was requested. Required in signed payloads to protect against replay attacks.
					Signature: (Reserved) See Section 8.1.2.

Table 5 - Configuration MQTT topics

8.2.2 Status

The Status topic is used by the Container Manager to request the status of running Applications. The Container Manager will periodically request the status, and running Applications must respond to the request to confirm that they are operating correctly.

If an Application does not respond or responds with a status other than STATUS_MSG_OK or STATUS_INITIAL (see Table 7), it is considered to have failed the request. After three successive failed status requests the Application will be restarted by the Container Manager. If the Container still fails further status requests to reach a total of 5 consecutive requests, it will be permanently shut down, and this error logged in the database.

QoS: Messages on this these topics must be sent and received with the QoS shown in Table 6.

Retention: Messages on this these topics must have the retention flag set to false.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

status /request	0	Message to request status from all running containers.	Container Manager	All Applications	`{`
					`"Timestamp": <Int64>,`
					`"Signature": {}`
					`}`
						No required payload.
						Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the status was requested. Required in signed payloads to protect against replay attacks.
						Signature: (Reserved) See Section 8.1.2.

Status /response/ <IID>	1	Message containing a status update from the Application Instance identified by <IID>.	Any Application	Container Manager	`{`
					`"Status": <Integer>,`
					`"Message": "<message>"`
					`"Timestamp": <Int64>,`
					`"Signature": {}`
					`}`
						Status: (Required) One of the values from Table 7.
						Message (Optional): If the Message string is present it will be sent to the error Database.
						Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the status was requested. Required in signed payloads to protect against replay attacks.
						Signature: (Reserved) See Section 8.1.2.

Table 6 - MQTT Status Topic

The valid status response values are shown in Table 7.

Status Value	Meaning
1	STATUS_MSG_OK – the Application is running normally.
2	STATUS_MSG_FAIL – the Application has failed. The Container Manager will restart the container. If the key “Message” is present in the JSON object it will be stored in the Data Storage Application as an error message.
3	STATUS_MSG_ERR – the same as STATUS_MSG_FAIL for backwards compatibility.
4	STATUS_SHUT_DWN – the Application has completed its shutdown procedures and is ready to be stopped by the Container Manager. The container will not be restarted unless the Container Manager configuration is altered or the Container Manager is re-started.
5	STATUS_INITIAL – the Application is waiting to receive its configuration (and can do nothing until it does). The Container Manager will resend the Application's configuration.
6	STATUS_RESTART – the Application wishes to be re-started. It has completed any shutdown procedures and saving of state and is ready to be stopped and started again by the Container Manager.

Table 7 - Status Field Values

Status values other than STATUS_MSG_OK and STATUS_INITIAL are regarded as failure conditions. If the key “Message” is present in a JSON object with a failure status, the Message string will be stored in the Data Storage Application as an error.

8.2.3 Command

The MQTT command topic allows the Container Manager to send instructions to any running Application.

QoS: Messages on this these topics must be sent and received with QoS = 1

Retention: Messages sent on this topic must have the retention flag set to false.

Topic	Description	Sender	Receiver	Payload	Notes
command/<IID>	This is a command sent from the Container manager for the container to execute	Container Manager	Any Application	`{ "Command": <Integer>, "Timestamp": <Int64>, "Signature": {} }`	**Command**: (Required) One of the command values shown in Table 9 below. **Timestamp**: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the command was issued. Required in signed payloads to protect against replay attacks. **Signature**: (Reserved) See Section 8.1.2.

Table 8 – Commands MQTT

The command values in Table 9 are currently specified. In the future, more values may be added, so all LV-CAP Applications must check the payload of the message received is the expected value.

Command Value	Command
1	Shut Down. Currently the only implemented command. All Applications must implement this command. This command is used when an updated Application is deployed. The Container Manager will send a shutdown command for the running Application to stop everything it is doing before re-starting the Application. See Section 6 for more details.

Table 9 - Command Topic Command Values

8.2.4 Error

The MQTT error topic allows all containers to log any issue or internal error.

QoS: Messages on these topics must be sent and received with QoS = 1.

Retention: Messages sent on this topic must have the retention flag set to false.

Topic	Description	Sender	Receiver	Payload	Notes

storage/data/error/<IID>	Topic to log any external or internal errors to storage.	All Applications	Data Storage Application	`{`
				`  "Errno": <Integer>,`
				`  "Message": "String",`
				`  "Timestamp": <Int64>,`
				`  "Signature": {}`
				`}`
					Errno: (Required**)** One of the errno values shown in the
					Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the command was issued. Required in signed payloads to protect against replay attacks.
					Signature: (Reserved) See Section

Table 10 - Report Error Topic Table

Command Value	Errno Name	Description
1	ERRNO_JSON_INVALID	Payload from MQTT failed to Parse. Invalid JSON.
2	ERRNO_IO	Input/output Error
3	ERRNO_ACCESS	Permission denied
4	ERRNO_NO_DEVICE	No device found
5	ERRNO_FILE_DIRECTORY	Directory not found
6	ERRNO_MQTT_SUBSCRIPTION	Failed subscription to MQTT Topic
7	ERRNO_MQTT_PUBLISH	Failed Publish, this is only used when trying to publish a payload to. If failed to publish an error message use std::out. This will be saved by the Docker Log files and can be accessed later by Admin.
8	ERRNO_APPLICATION	Process failed due to Application error. The message to accompany this Errno is mandatory.

Command Value	Errno Name	Description
9	ERRNO_CONFIGURATION	Failed processing the incoming Config. This is if the contents of the configuration expected does not match or has the wrong types. (This could also be ERROR_JSON_INVALID if it's not valid JSON)
10	ERRNO_MQTT_CABLLBACK	Error occurred in the MQTT Call back. This can be when setting up the call back or an error within the call back with an incoming message
11	ERRNO_SENSOR	This can have two applications. The first, for any Sensor Container that has an error with reading a sensor it can output this Errno with the relevant message. The second is for any Algorithm Container reading in the Sensor Payload and the Payload is valid but any of the Key types is incorrect.
12	ERRNO_NETWORK	Error accessing the network.
13	ERRNO_PORT	Error opening or accessing a port.
14	ERRNO_PROFILE	Any Algorithm Application expecting a Profile Payload, the Payload is valid but any of the Key types is incorrect.

Table 11 - Errno Description Table

8.3 Sensor Data API

Applications which fulfil the Sensor Application role (see Section 0) will publish on the topics in the Sensor Data API. Applications in the Algorithm Application role will often subscribe to these topics to obtain their inputs. This data will not normally be stored.

8.3.1 Sensor Readings

Topics for transferring sensor reading data collected and published by Sensor Applications.

QoS: Messages on this these topics must be sent and received with QoS = 1.

Retention: Messages on this these topics must have the retention flag set to false.

Topic	Description	Sender	Receiver	Payload	Notes
sensor/data/<IID>/<sensorname>	New sensor readings	Sensor Applications	Any Application	Standard Scalar Object Format, Series Object Format or Co-ordinate Object Format.	See Section 9 for details of standard JSON formats.

Table 12 - Sensor Reading MQTT messages

See Section 4.4 for guidelines on choosing intelligible topic names for message output. Sensor Applications are responsible for publishing data and setting the Valid flag in messages (Section 9) in accordance with the guidelines set out in Section 4.8.

Readings will often be published at fixed time intervals. These intervals will start when the sensor Application receives its configuration, and so may not be aligned to "clock face" times. For instance, if the configuration was received at 09:05:00, setting a time interval of 20 seconds. The Sensor Application will output at 09:05:20 then at 09:05:45 and so on.

Depending on the properties of the sensor Application, there is a possibility that if many sensors have the same interval time and one sensor takes longer to read that this would delay the next sensor and so on. The start of the normal operation for the Sensor Application is most susceptible to this, however after a short time this will reach an equilibrium and each output will be at the prescribed interval. Applications consuming the messages must be equipped to cope with these timing variations.

8.3.2 Sensor Metadata

Topics for transferring sensor metadata published by Sensor Applications. Publishing on these Metadata topics is optional.

QoS: Messages on this these topics must be sent and received with QoS = 1.

Retention: Messages on this these topics must have the retention flag set to true.

Topic	Description	Sender	Receiver	Payload	Notes
sensor/data/<IID>/<sensorname>/metadata	Sensor reading metadata	Sensor Application	Any Application	Standard Data Series Metadata. Object Format.	See Section 9 for details of standard JSON formats.

Table 13 - Sensor Reading MQTT messages

8.4 Algorithm Data API

Applications which fulfil the Algorithm Application role (see Section 0) will publish on the topics described in the Algorithm Data API. Application in the Algorithm Application role may subscribe to these topics to obtain inputs. Applications in the Output Application role will normally subscribe to one or more of these topics to obtain inputs.

Data published on these topics may be stored in the Data Storage Application, depending on the latter's configuration and the "ToStore" flag set by the publishing Application. Only stored data will be available for upload by Communications Applications.

QoS: Messages on this these topics must be sent and received with QoS = 1.

Retention: Messages on this these topics must have the retention flag set to false.

Topic	Description	Sender	Receiver	Payload	Notes

algorithm/data/<IID>/<subtopic>	The main topic an algorithm Application will publish its data on	Algorithm Application	Any Application	Any valid JSON object.
				Applications are strongly encouraged to use one of the standard JSON Object Formats to improve interoperability.
				{
				`<Valid JSON Payload>`
				}
					Algorithm Applications may output on any sub-topic starting with "algorithm/data/<IID>" (where <IID> is the Application Instance's assigned identifier).

Table 14 - Algorithm Data Table

When choosing the sub-topic on which to output data, authors are encouraged to use a descriptive topic name (Section 4.4). This makes configuring systems easier and less error prone. For example, transformer capacity forecasts for the available capacity in transformer T1 over the next 4 hours might be output on topic

algorithm/data/<IID>/T1/forecast/4h/capacity

If the JSON payload is to be stored in the Data Storage Application it must have a KEY "ToStore" and the value set to true. If this is not present or is set to false then the data will not be stored. Only stored data will be available for upload by Communications Applications.

Payloads should have a KEY "Timestamp" containing the Unix timestamp the calculation refers to. Where the calculation covers a range of time, this should be the time stamp of the most recent time covered by the calculation.

Algorithm Applications may use optional metadata subtopics in exactly the same way as Sensor Applications, as documented in Section 8.3.2.

8.5 Data Upload API

The Data Upload API provides a means to access the data queued for upload in the Data Storage Application. Applications which fulfil the Communications Application Upload role (see Section 0) will use this API extensively. Applications using this API must be explicitly authorised by the system operator in the Data Storage Application configuration. A separate (virtual) queue is maintained for each Upload Application of data which is waiting for upload. Once a message has been uploaded the Upload Application must notify this fact back to the Data Storage Application via this API so that the queues can be updated.

This API operates on a pattern of separate topics for requests and response messages. When using this API, Applications should always subscribe to the response topic before publishing a request. This avoids a race between the response and the subscription which may cause the container to miss response messages.

All methods in this API work with the per-Application database tables described in Section 0. The SubTopic and Data columns are set from the received message. The other columns in the table will be set automatically by the Data Storage Application.

QoS: Messages on this these topics must be sent and received with the QoS shown in Table 15.

Retention: Messages on this these topics must have the retention flag set to false.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

storage/request/newdata/<IID>	2	A request for new data to be uploaded by Upload Application <IID>. The request will search all tables in the database which the Application is permitted to upload from.	Upload Application with identifier <IID>	Data Storage Application	`{`
					`"MaxLength": <Integer>,`
					`"StartTime": <Int64>,`
					`"EndTime": <Int64>,`
					`"PreferOldest": <Boolean>,`
					`“InstanceID”: <IID>`
					`"SubTopic": <String>,`
					`"MinPriority": <Int>,`
					`"MaxPriority": <Int>,`
					`"Timestamp": <Int64>,`
					`"Signature": {}`
					`}`
						The request Payload keys are documented in Table 16.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

storage/response /newdata/<IID>	2	The response to the above request.	Data Storage Application	Upload Application with identifier <IID>	`{`
					`"Status": <Integer>,`
					`"Response": [`
					`    {`
					`      "TableName": <IID>,`
					`      "TableRows": [`
					`        {`
					`        "ID":<Integer>,`
					`        "Timestamp": <Int64>,`
					`        "SubTopic": <string>,`
					`        "Data": <JSON Object>,`
					`        },`
					`        {rowN}`
					`      ]`
					`    },`
					`    {`
					`      "TableName": <IID>,`
					`       “TableRows”: [`
					`         {`
					`        "ID":<Integer>,`
					`        "Timestamp": <Int64>,`
					`        "SubTopic": <string>,`
					`        "Data": <JSON Object>,`
					`        },`
					`        {rowN}`
					`      ]`
					`    }`
					`  ]`
					`"Timestamp": <Int64>,`
					`"Signature": {}`
					`}`
						The response Payload keys are documented in Table 17.
						If an error occurs then the Payload will still have Status and Response members, but the Response array will be empty.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

storage/uploaded/<IID>	1	Indicates that messages have been uploaded by <IID> and they should be removed from the upload queue.	Upload Application with identifier <IID>	Data Storage Application	`{`
					`"NewData": [`
					`    {"<IID>": [<Integer>, <Integer>]},`
					`    {"<IID>": [<Integer>, <Integer>]},`
					`  ],`
					`"Timestamp": <Int64>,`
					`"Signature": {}`
					`}`
						NewData: (Required) Array of objects, one for each table to be updated.
						<IID>: (Required) Array of opaque integer identifiers of the messages which have been uploaded.
						Timestamp: (Optional) Standard LV-CAP timestamp (see Section 4.5) when the update was sent. Required in signed payloads to protect against replay attacks.
						Signature: (Reserved) See Section 8.1.2.

Table 15 – Communications Upload Container MQTT

Key	Status	Description
`MaxLength`	Optional	The maximum number of records to be returned from the database. This is subject to an upper limit set in the Data Storage Container configuration (see 8.5.1 below). If no value is given then the default value is 100 records.
`StartTime`	Optional	A UNIX timestamp. Only records added to the Data Storage Application after this time will be returned. If not supplied then records from the start of the database will be returned, unless the operator has imposed a tighter restriction.
`EndTime`	Optional	A Unix timestamp. Only records added to the Data Storage Application before this time will be returned. If not supplied then records up to the present time are returned.
`PreferOldest`	Optional	Flag indicating that if there are more than MaxLength records available, the oldest data should be returned rather than the default of returning the newest data.

Key	Status	Description
`InstanceID`	Optional	String identifying the Application Instance which data should be returned for. Only records which exactly match the given topic will be returned (no wildcards). This constrains the query to only return results from the specified table in the database. If this member is an empty string or omitted from the JSON then data from all Application Instances is returned.
`SubTopic`	Optional	String giving the sub-topic data is required for. This is the sub-topic below data/algorithm/<IID>. Only records which exactly match the given topic will be returned (no wildcards). To retrieve data from all sub-topics, do not include this key in the JSON payload. To request data only from the top-level topic (no sub-topics) then this key must be included in the JSON payload with an empty string value.
`MaxPriority`	Optional	Integer defining what priority messages are to be returned. If this JSON key is supplied, messages with priority equal to or numerically less than the value only will be returned. The special value of 6 can be used to return only messages which had no Priority value when stored. If neither this JSON key nor MaxPriority is specified then messages of all priorities will be returned.
`MinPriority`	Optional	Integer defining what priority messages are to be returned. If this JSON key is supplied, messages with priority equal to or numerically greater than the value only will be returned. If neither this JSON key nor MaxPriority is specified then messages of all priorities will be returned. If both keys are supplied then only messages which meet both criteria will be returned.
`Timestamp`	Optional	A UNIX timestamp when the request was sent. Required in signed payloads to protect against replay attacks.
`Signature`	Reserved	See Section 8.1.2

Table 16 – Request Object Keys

Key	Status	Description
`Status`	Always Present	Integer indicating whether the query succeeded or not. See Table 18.
`Response`	If Status = DSC_QUERY_OK	An array of objects containing data from different tables to be uploaded. Always an array even if data is only from one table.
Response/`TableName`	Always Present	Name of the table the data in this object is from.

Key	Status	Description
`Response/TableRows`	Always Present	An array of selected rows from the table (array even if only one row is selected). Each object in the array is an individual message from the source table.
`Response/TableRows/ID`	Always Present	Opaque integer identifier for the message. These have no meaning except as a handle to be passed back to the Data Storage Application when the message has been uploaded. ID values are only unique within a single table, and may be recycled after the database has been cleaned.
`Response/TableRows/Timestamp`		UNIX timestamp when the message was added to the Data Storage Application (see Section 0).
`Response/TableRows/SubTopic`		The subtopic (below algorithm/<IID>) on which this message was published.
`Response/TableRows/Data`		The original message JSON object stored in the Data Storage Application.
`Timestamp`	Optional	A UNIX timestamp when the response was sent. Required in signed responses to protect against replay attacks.
`Signature`	Reserved	See Section 8.1.2

Table 17 – Response Object Keys

Status Value	Code	Description
0		Never sent, an unanticipated error.
1	DSC_QUERY_OK	Query succeeded, the length of the complete results set is less than or equal to `MaxLength`. The result is returned in the `TableRows` array. See also DSC_QUERY_MORE.
2	DSC_QUERY_EMPTY	The query was valid, but found no records. The `TableRows` array will be empty.
3	DSC_QUERY_TABLE_DENIED	The query is against a table (Application Instance) which the sending container is not allowed to access.
4	DSC_QUERY_TOO_LONG	The query requested more data than the Data Storage Application is willing to provide, because the `MaxLength` field value was too large (see 8.5.1 below).
5	DSC_QUERY_TOO_BIG	The data requested by the query is too big to fit into the MQTT payload length restriction.
6	DSC_QUERY_TOO_OLD	The data requested in the query is from further in the past than the Data Storage Application is willing to provide.

Status Value	Code	Description
7	DSC_UNAVAILABLE	The Data Storage Application is unable to respond to this request, either because it is too busy or is in the process of shutting down.
8	DSC_QUERY_MORE	Query succeeded, there are more than `MaxLength results.` The first MaxLength results are returned in the `TableRows` array, but another query is needed to get more values.
9	DSC_QUERY_INVALID	The JSON query object is empty or not valid JSON and cannot be parsed.

Table 18 – Response Status Values

The Data Upload API is not designed to be re-entrant. After a request has been made, the container should wait for the response (there may need to be an exceptional time-out in case the Data Storage Application suffers an error). If a second request is made whilst the response is being produced, the response is undefined. The request does not modify the database at all, so if a second identical request is made after the first response is received, the same data will be returned.

The response status value is used to show whether there is more data available than was sent or not. There is no concept of a database cursor or response pagination. As a result, API users who need to upload all the available data must:

1. Request data for upload.
2. Upload the received messages (if any).
3. Update the database to mark the messages as uploaded.
4. Continue querying until a response of DSC_QUERY_EMPTY is received, at which point there is no more data to upload.

Although the JSON format for the "storage/response/newdata/<IID>" topic allows for messages from multiple tables to be sent in one message, this is not guaranteed. The Data Storage Container may opt to return data from only one table or topic in the response (where there is data to retrieve), and return data from other tables/topics when subsequent requests are received.

8.5.1 Limits

The Data Storage Application is a shared resource and excessively large queries have the potential to degrade the performance of LV-CAP for all users. To mitigate this risk, limits are imposed on the queries which will be accepted.

- **Maximum number of records requested in one query**. If `MaxLength` is not set then a limit of 100 records will be applied. A query for 100 records will always be permitted. This limit may be increased (up to a maximum of 15 000) by the LV-CAP operator, but Application should not depend upon larger queries being allowed on any given system. Note that the limit is on the requested size, not the actual number of records found (which is not known when the query is set up). Thus a request for 16 000 records will always fail (DSC_QUERY_TOO_LONG from Table 18), even if the table is empty.
- **Maximum query size.** Because the query result is sent as an MQTT message via the Data Marketplace, it is limited to a maximum of 256MB (268,435,455 bytes), as documented in section 2.2.3 of the MQTT 3.1.1 specification. If the query results in a message which is longer than this limit, then an error (DSC_QUERY_TOO_BIG from Table 18.) is returned instead. Applications must request fewer messages to reduce the returned message size below the limit.

- **Maximum data age**. The Data Storage Application will not be able to store data going back in time indefinitely. Old records will be purged by the Data Storage Application to control the database size. To manage database performance the LV-CAP operator may also impose a maximum age on queries. Any request for data older than this age will fail with DSC_QUERY_TOO_OLD from Table 18.

8.5.2 Examples

An example query payload requesting the oldest available data for upload, from all Applications, is shown in Figure 5. The query is not signed. Up to 100 records will be returned as there is no maximum length given. This query may fail:

- With status DSC_QUERY_TOO_OLD if the Data Upload Application does not allow queries indefinitely into the past.
- With status DSC_UNAVAILABLE if the Data Upload Application is shutting down or overloaded.
- With status DSC_QUERY_TOO_BIG if the results will not fit in a MQTT packet.

If it succeeds it could give status:

- DSC_QUERY_EMPTY if there is no data to be sent.
- DSC_QUERY_OK if there are between 1 and 100 messages to be sent.
- DSC_QUERY_MORE is there are more than 100 messages to be sent.

```
{
    "PreferOldest": True
}
```

Figure 5 – Example query payload

An example query payload requesting the latest available data from a specific Application Instance is shown in Figure 6. The query is not signed. Up to 50 records will be returned as requested. This query may fail:

- With status DSC_QUERY_TABLE_DENIED if the Data Storag Application does not allow this Data Upload Application to upload data from this Application Instance.
- With status DSC_UNAVAILABLE if the Data Storage Application is shutting down or overloaded.
- With status DSC_QUERY_TOO_BIG if the results will not fit in a MQTT packet.

If it succeeds it could give status:

- DSC_QUERY_EMPTY if there are no messages from this Application Instance.
- DSC_QUERY_OK if there are between 1 and 50 messages to be sent.
- DSC_QUERY_MORE is there are more than 50 messages to be sent.

```
{
    "MaxLength": 50,
    “InstanceID”: "eatl_profiler_04"
}
```

Figure 6 – Example query payload for a specific Application Instance

An example query payload requesting the latest, highest priority, data from a specific topic is shown in Figure 7. The query is not signed. Up to 10 records will be returned as requested. This query may fail:

- With status DSC_QUERY_TABLE_DENIED if the Data Storage Application does not allow this Data Upload Application to upload data from this Application Instance.
- With status DSC_UNAVAILABLE if the Data Storage Application is shutting down or overloaded.
- With status DSC_QUERY_TOO_BIG if the results will not fit in a MQTT packet.

If it succeeds it could give status:

- DSC_QUERY_EMPTY if there are no messages on this topic with priority equal to 1.
- DSC_QUERY_OK if there are between 1 and 10 messages on this topic with priority 1.
- DSC_QUERY_MORE is there are more than 10 messages on this topic with priority 1

```
{
    "MaxLength": 10,
    "InstanceID": "eatl_profiler_04",
    "SubTopic": "alarm/highhigh",
    "MaxPriority": 1
}
```

Figure 7 – Example query payload for a specific topic and priority.

8.6 Data Storage API

The Data Storage API provides a means to access the data persistently stored by the Data Storage Application. This API provides a mechanism for Applications to access data previously stored by Applications, e.g. where a system history is required.

This API operates on a pattern of separate topics for requests and response messages. When using this API, Applications should always subscribe to the response topic before publishing a request. This avoids a race between the response and the subscription which may cause the container to miss response messages.

All methods in this API work with the per-Application Instance database tables described in Section 0. The SubTopic and Data columns are set from the received message. The other columns in the table will be set automatically by the Data Storage Application.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

storage/data/<IID>/	2	Insert data into the Data Storage Application, in the <IID> table.	Any Application	Data Storage Application	{
					    "Key": Data,
					    "KeyN": DataN
					}
						Data messages on this topic can hold anything the sender wishes. The message payload will be stored unaltered as a BLOB.
						If messages are sent on a sub-topic below storage/data/<IID>/ then the sub-topic will be stored in the SubTopic column of the table.
						This is equivalent to publishing data on algorithm/data/<IID> with the ToStore flag true.

storage/request/<IID>	1	Request data by the Application Instance <IID>.	Any Application	Data Storage Application	{
					    "MaxLength": <Integer>,
					    "StartTime": <Int64>,
					    "EndTime": <Int64>,
					    "PreferOldest":
					<Boolean>,
					    "InstanceID": <IID>
					    "SubTopic": <String>,
					    "MinPriority": <Int>,
					    "MaxPriority": <Int>,
					, "Timestamp": <Int64>,
					    "Signature": {}
					}
						The Data Storage Application will return the requested data on the storage response topic below.
						The request Payload keys are documented in Table 16.

Topic	QoS	Description	Sender	Receiver	Payload	Notes

storage/response/<IID>	1	The response from the data storage container after a get request by Application Instance <IID>.	Data Storage Application	The <IID> Application which requested the table	{
					"Status": <Integer>,
					"Response": [
					{
					"TableName": <IID>,
					"TableRows": [
					{
					"ID":<Integer>,
					"Timestamp": <Int64>,
					"SubTopic": <string>,
					"Data": <JSON Object>,
					},
					{more rows}
					]
					}
					]
					}
						The response Payload keys are documented in Table 17.

Table 19 – Data Storage Container

Because of the automatic storing of Algorithm output described in Section 8.4, it will be unusual to need to explicitly store data using the "storage/data/" topic. Publishing on the "storage/data/<IID>" topic has exactly the same results as publishing on the "algorithm/data/<IID>" topic with the ToStore flag true.

The fields of the message on the request topic are documented in Table 16, and those of the response message on the response topic in Table 17. These objects are deliberately the same as those used by the Data Upload API. The same rules for InstanceID and SubTopic apply. Similarly, "storage/response/newdata/<IID>" and "storage/response/<IID>" use the same response format, although in this API there will only ever be data from one table and so only one element in the Response array. Note that the Instance ID <IID> in the topic names refers to the Application making the requests and receiving the data, not the table being accessed (except in the first topic documented, where they are the same).

9. JSON Object Structures

All messages passed through the Data Marketplace, and all Application Configuration data, is serialised as JSON objects. Whilst for some purposes bespoke JSON object structures are necessary, wherever possible use should be made of the standard JSON object structures defined in this section.

Using standard object structures ensures that data can be passed from any Application to any other Application without the need for bespoke software development. It minimises the need for Applications to cope with data from different sources in different formats. Applications which output in standard formats will be best placed to take advantage of facilities provided by LV-CAP and other Applications.

In applying these Object formats consideration should also be given to the general principals set out in Section 0.

9.1 Scalar Object Format

The default choice of JSON Object for almost all sensor readings and many algorithm outputs will be the Scalar Object. It represents a single value at a single point in time, for instance a temperature or a power flow. To provide more metadata about the value and how it was arrived at, a separate Data Series Metadata Object should be used (see Section 9.4).

```
{
    "Timestamp": <Int64>,
    "Value": < >,
    "Valid": <Boolean>,
    "ToStore": <Boolean>,
    "Priority": <Int>,
    "Signature": {}
}
```

Figure 8 - Scalar Object Format Structure

Key	Status	Description
`Timestamp`	Required	Standard LV-CAP timestamp (see Section 4.5) when the reading was made.
`Value`	Required	The reading, converted to base engineering units. The reading can be of any scalar type (Boolean, Integer or Floating Point).
`Valid`	Required	A logical value, showing if the Value is within the expected range (configured).
`ToStore`	Optional	Flag used historically to indicate whether the data should be stored by the Data Storage Application or not. May be over-ridden by the Data Storage Application configuration.
`Priority`	Optional	A priority indicator as in section 4.9, which allows the upload of certain messages to be prioritised by Data Upload Applications.
`Signature`	Reserved	See Section 8.1.2

Table 20 - Scalar Object Format Keys

The Value is always given in the base SI unit for the quantity being measured or calculated, as in Section 4.5. The units can be given explicitly in the optional Data Series Metadata Object (Section 9.4).

9.2 Series Object Format

Where a series of closely related values are to be sent as a set then a Series Object provides a way to package the complete set of values in a single JSON Object. It can represent a time series of values (anything from a fault waveform recorder (sampling many times per mains cycle) to a load profile (hourly load values), or a frequency spectrum. The object contains fields to record what range of source data was used to produce series. To provide more metadata about the value and how it was arrived at, a separate Data Series Metadata Object should be used (see Section 9.4).

```
{
    "Timestamp": <Int64>,
    "StartPoint": <Int64 or float>,
    "Interval" : <float>,
    "Value": [< >],
    "Confidence": [< >],
    "Valid": <Boolean>,
    "TimestampStart": <Int64>,
    "TimestampEnd": <Int64>,
    "ToStore": <Boolean>,
    "Priority": <Int>,
    "Signature": {}
}
```

Figure 9 - Scalar Object Format Structure

Key	Status	Description
`Timestamp`	Required	Standard LV-CAP timestamp (see Section 4.5) when the series was produced.
`Interval`	Required	Interval between values in the series. For time series, this is the time between samples in seconds (or decimals of them), for frequency spectrums Hertz and so on.
`StartPoint`	Required	The x-axis co-ordinate of the first value in the series. For time series this is the timestamp of the first value, for frequency spectrums the frequency of the first bin and so on.
`Value`	Required	An array of series values in base engineering units. The values can be of any scalar type (Boolean, Integer or Floating Point).
`Confidence`	Optional	An array, the same size as the Value array, of values giving the confidence in the values. This may be used to represent the uncertainties caused by missing input data or inadequate system history.
`Valid`	Required	A logical value, showing if the Series as a whole is thought to be valid for further use.
`TimestampStart`	Optional	The standard LV-CAP timestamp of the earliest source data used to produce this series.
`TimestampEnd`	Optional	The standard LV-CAP timestamp of the latest source data used to produce this series.
`ToStore`	Optional	Flag used historically to indicate whether the data should be stored by the Data Storage Container or not. May be over-ridden by the Data Storage Container configuration.
`Priority`	Optional	A priority indicator as in section 4.9, which allows the upload of certain messages to be prioritised by Data Upload Applications.
`Signature`	Reserved	See Section 8.1.2

Table 21 - Scalar Object Format Keys

This is deliberately an abstract data format designed to be capable of accommodating a wide range of different data types. The Value are always given in the base SI unit for the quantity being measured or calculated, as in Section 4.5. The units can be given explicitly in the optional Data Series Metadata Object (Section 9.4). Some practical examples of its use are given in Figure 10 and Figure 11.

Figure 10 shows the Series Object Format used for a predicted load profile.

- It was calculated and published at 14:30:05 UTC on 8th March 2017.
- The first predicted load segment in the prediction starts at 14:30:00 UTC on 8th March 2017
- The prediction is composed of half-hourly (1800 seconds) load current values.
- The prediction is for 4 hours, so has 8 values of load current in amps.
- The predictor is confident in the prediction for the first two hours and the last one, but is aware of limitations in the data for the third hour (e.g. because there are problems with missing data in that hour). These reduce the confidence in those predictions.
- Overall the predictor thinks that this data is valid for use.
- The prediction is built on the previous 4 weeks of data, so the oldest data used was from 14:30:00 UTC on 8th March 2017.
- The most recent data used was from 15:00:00 UTC on 1st March 2017, the end of the half hour period one week ago.
- The prediction is not signed.
- The predictor does not stipulate whether this data is to be stored or not.

```
{
    "Timestamp": 1488983405,
    "StartPoint": 1488983405,
    "Interval" : 1800.0,
    "Value": [120.0, 122.5, 125.7, 130.9, 124.8, 121.4, 118.6, 115.3],
    "Confidence": [1.0, 1.0, 1.0, 1.0, 0.85, 0.75, 1.0, 1.0],
    "Valid": true,
    "TimestampStart": 1486564200,
    "TimestampEnd": 1488380400
}
```

Figure 10 – Example of a Scalar Object used for a load prediction

Figure 11 shows the Series Object Format used for a harmonic spectrum.

- It was calculated and published at 11:00:00 UTC on 5th March 2017.
- The first harmonic in the spectrum is 50Hz
- The spectrum is composed of values for each harmonic, so every 50 Hz.
- The spectrum is for the first 5 harmonics only.
- The spectrum is calculated, so no confidence values are given.
- Overall the calculation succeeded, so the data is valid for use.
- The spectrum was calculated from the previous 10 minutes of data, so the oldest data used was from 10:50:00 UTC on 5th March 2017.
- The most recent data used was from 10:59:59 UTC on 5th March 2017, the end of the 10-minute period.
- The publishing container thinks that this data should be stored in the Data Storage Container.
- The publishing container has assigned this data the lowest available priority.
- The spectrum is not signed.

```
{
    "Timestamp": 1488711600,
    "StartPoint": 50,
    "Interval" : 50,
    "Value": [76423.0, 122.5, 86.7, 57.9, 12.4],
    "Valid": true,
    "TimestampStart": 1488711000,
    "TimestampEnd": 1488711599,
    "ToStore": true,
    "Priority": 5
}
```

Figure 11 – Example of a Scalar Object used for a harmonic spectrum

9.3 Co-ordinate Object Format

Where a group of co-ordinates are produced then a Co-ordinate Object provides a way to package them in a single JSON Object. The co-ordinates may be in a real space (e.g. Latitude and Longitude for geographic position) or a conceptual one (real and imaginary power in a power flow vector diagram). To provide more metadata about the value and how it was arrived at, a separate Data Series Metadata Object should be used (see Section 9.4).

```
{
    "Timestamp": <Int64>,
    "Coordinates": [< >],
    "System": <String>,
    "Valid": <Boolean>,
    "ToStore": <Boolean>,
    "Priority": <Int>,
    "Signature": {}
}
```

Figure 12 - Co-ordinate Object Format Structure

Key	Status	Description
`Timestamp`	Required	Standard LV-CAP timestamp (see Section 4.5) when the reading was made.
`Coordinates`	Required	Array of co-ordinate values, in base engineering units. The co-ordinate values will be Integer or Floating-Point values.
`System`	Optional	The co-ordinate system being used, e.g. Cartesian (for x-y plots) or WGS84 latitude and longitude.
`Valid`	Required	A logical value, showing if the Value is within the expected range (configured).
`ToStore`	Optional	Flag used historically to indicate whether the data should be stored by the Data Storage Container or not. May be over-ridden by the Data Storage Container configuration.
`Priority`	Optional	A priority indicator as in section 4.9, which allows the upload of certain messages to be prioritised by Data Upload Applications.
`Signature`	Reserved	See Section 8.1.2

Table 22 - Co-ordinate Object Format Keys

No implementation of this Object Format yet exists.

9.4 Data Series Metadata Object Format

There will be various pieces of metadata (that is, information about the data) associated with the data published on a given topic. It may be desirable to transmit these in a machine-readable format, so that consuming Applications can make use of them. However, this metadata does not change from reading to reading, so it would be inefficient to transmit (and especially store) them alongside the readings themselves. Instead a separate /meta/ sub-topic is used for metadata, which is transmitted as Data Series Metadata Objects.

The metadata objects are sent only when an Application starts, or there is a change to the metadata. In order that subscribing Applications always receive this information, the messages are published with the Retained flag set to true. This means that the Data Marketplace will automatically send a copy of the latest metadata to any new client which subscribes the /meta/ sub-topic, without any effort from the publishing Application.

```
{
    "Name": <String>,
    "Units": <String>,
    "DisplayUnits": <String>,
    "SigFigs": <Integer>,
}
```

Figure 13 - Data Series Metadata Object Format Structure

Key	Status	Description
`Name`	Required	String to be used as the name for this data, e.g. labels on graphs
`Units`	Optional	String giving the units of the data sent on the topic. As the data will be in base engineering units (Section 4.5), this just gives the correct SI unit, for instance amps or metres per second.
`DisplayUnits`	Optional	String giving the units for display of the data. This may include the use of SI prefixes for convenient display, e.g. kVAr for reactive power flow.
`SigFigs`	Optional	Integer indicating how many significant figures the values should be displayed to, to avoid the spurious display of excess decimal places caused by binary floating-point representation.

Table 23 - Data Series Metadata Object Format Keys

An example of the Data Series Metadata for a topic carrying a voltage measurement is shown in Figure 14.

```
{
    "Name": "Line1 to Neutral",
    "Units": "Vrms",
    "DisplayUnits": "Vrms",
    "SigFigs": 3
}
```

Figure 14 - Scalar Object Format Structure

9.5 Application Configuration Format

All configuration files must be valid JSON objects. Application configuration data will differ significantly from Application to Application depending on their structure, and only be of use to the Application it is intended for. To accommodate this, the structure of the configuration file for each container is largely up to the Application author, but a standard top-level structure is required in order to deliver updated configuration information to the correct container. Application authors are free to structure the ContainerConfig object within their configuration in whatever way suits their application, provided that it is a valid JSON object.

```
{
      "ContainerName": "<IID>",
      "ContainerConfig":
      {
            "<examplekey1>": <configvalue1>,
            "<examplekey2>": <configvalue2>,
            "<examplekeyN>": <configvalueN>
      }
}
```

Figure 15 - Third Party Container Configuration File Example

Key	Status	Description
`ContainerName`	Required	IID of the Application the configuration is for.
`ContainerConfig`	Required	JSON object containing the Application Instance configuration. This object's contents and structure will change from Application to Application.

Table 24 - Third Party Configuration File Keys

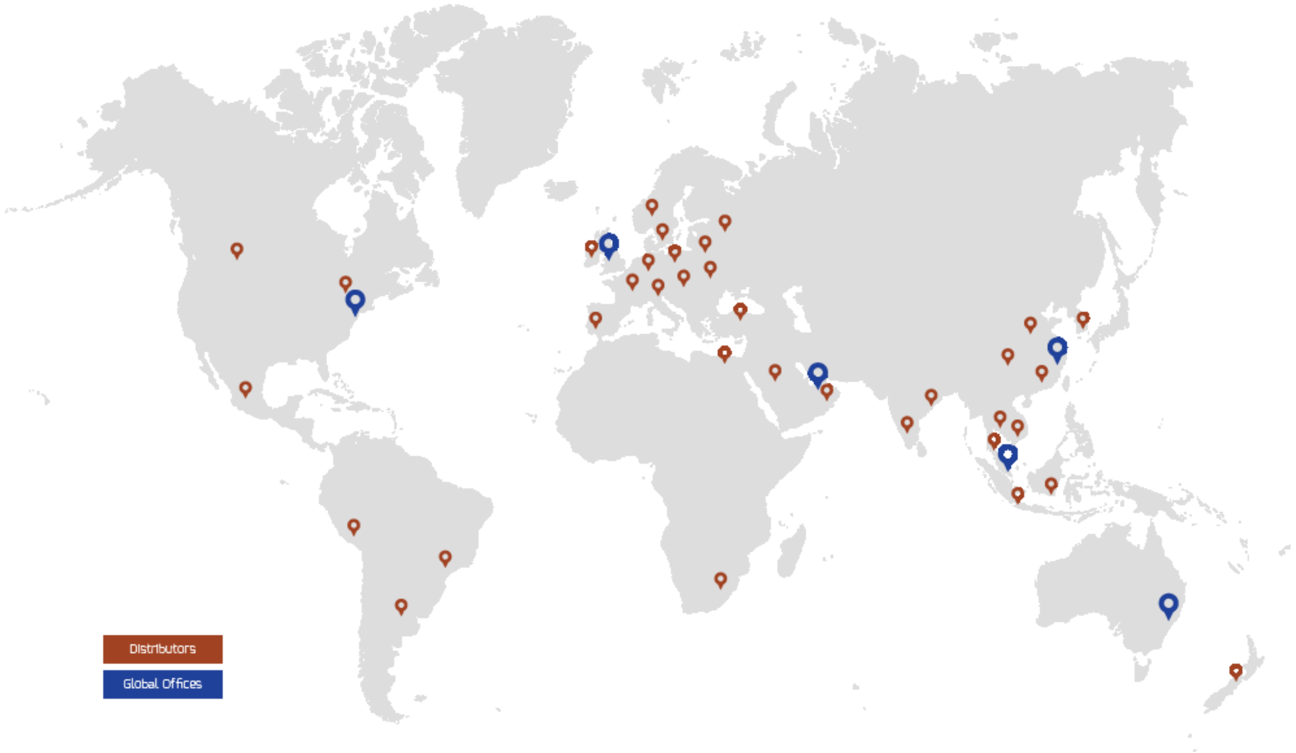
10. References

The following external resources provide more information to support this specification:

1. The MQTT Standard, version 3.1.1: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
2. ECMA Standard 404, "The JSON Data Interchange Format" http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf
3. Blog Post "MQTT Topics & Best Practices" http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices
4. Docker Documentation for docker tag command: https://docs.docker.com/v1.12/engine/reference/commandline/tag/ and https://docs.docker.com/v17.03/engine/reference/commandline/tag/

Global Footprint

We provide products, services and support for customers in 90 countries, through our offices in Australia, China, Europe, Singapore, UAE and USA, together with more than 40 distribution partners.



Our Expertise

We provide world-leading asset management solutions for power plant and networks.

Our customers include electricity generation, transmission and distribution companies, together with major power plant operators in the private and public sectors.

- Our products, services, management systems and knowledge enable customers to:
- Prevent outages
- Assess the condition of assets
- Understand why assets fail
- Optimise network operations
- Make smarter investment decisions
- Build smarter grids
- Achieve the latest standards
- Develop their power skills

Safer, Stronger,
Smarter Networks

5 Appendix B – Nortech Application Container

Requirement Specification

Drawing Number	**2626-RQSPC-SHT04-V00.01.01 Nortech Comms Container Requirements.docx**
Project Title	**OpenLV: Nortech Comms Application Requirements**
Electronics Systems Project No.	**E717**
Charge Code	**EX267**
Supplier	**Nortech**
Request Date	**2017/07/01**

1 Background

EA Technology undertook an InnovateUK Energy Catalyst project with University of Manchester and Nortech Management Ltd to develop a Common Application Framework for LV Network Management. The core software platform developed by EA Technology under the project is called LV-CAP and consists of a number of Docker containers communicating with each other using MQTT.

WPD has now been awarded a Network Innovation Competition (NIC) project to trial the LV-CAP platform on real networks. This project is known as OpenLV. The OpenLV project will use a Nortech iHost server to store data from the OpenLV hardware, and as the management platform for the LV-CAP software running on the OpenLV hardware. To send data to and from the iHost server, a communications Application written by Nortech will be used. This Application was created as part of the Innovate UK LV-CAP project, but some enhancements are necessary to meet the requirements of the OpenLV project. This specification details the enhancements to the Nortech Comms Application required to meet the needs of the OpenLV Trial Programme.

2 Requirements

2.1 Hardware Environment

The OpenLV project hardware consists of an industrial PC based around a dual-core Intel Core i3 processor with 8GB of RAM and a 512GB SSD. This PC provides the processing power and storage for the whole LV-CAP solution. It has two Ethernet ports for network communications:

1. Local Ethernet link to the GridKey MCU520, fitted with an additional Ethernet module.
2. Ethernet link to a stand-alone 4G router which provides wide area network communications.

The Nortech Comms Application will communicate with the iHost sever via the wide area network.

2.1.1 Networking

See EATL drawing 2626-IMSPC-SHT03-V00.01.00 for the expected network architecture.

The volume of mobile data transferred must be managed to reduce the operating costs of the OpenLV system.

2.2 Base Operating System

The OpenLV project PC will be running 64-bit Ubuntu Server 16.04 LTS with current updates applied.

2.3 LV-CAP Environment

The Nortech Comms Application is a "core" Application on the LV-CAP Platform which fulfils the Management Comms and Data Upload roles. This is described in the Public API document 2383-MANUL-V04.02.07 and the Internal API document 2362-MANUL-V04.01.05, hereafter referred to as "the LV-CAP API".

2.3.1 API Implementation Status

The following features of the LV-CAP API are not expected to be implemented in time for the OpenLV project:

1. Individual message signing (section 8.1.2) will not be implemented.
2. Signing of Docker Image files will not be implemented.
3. Only one instance of each Application will be run (section 4.2) on LV-CAP.
4. As a result, Applications may continue to use legacy GUID identifiers.
5. To simplify TLS implementation, TLS keys and certificates will be built into Docker Image files. The end date of TLS certificates should be set beyond the end of the OpenLV project trials in September 2019. TLS implementation is mandatory.
6. The Priority feature of the data storage APIs will not be implemented, with all queries returning messages of all priorities. Applications are free to output Priority data, but it will not be parsed yet. Similarly requests may be made with Priority key values, but the key will be ignored.

2.3.2 Application Identification

The Vendor string for Nortech is "nortech". The Application Name for this Application is "commscontainer". Each release of the Nortech Comms Application should be tagged as described in section 4.2 of the LV-CAP API. For example the Docker tags would be:

```
nortech/commscontainer:0.1.0
nortech/commscontainer:0.1.2
nortech/commscontainer:0.1.3
nortech/commscontainer:1.0.0
nortech/commscontainer:1.0.3
nortech/commscontainer:1.2.0
```

These tags are important as they are used by Docker to load and run the Application on the LV-CAP. Incorrect tags may result in the incorrect version of the image being deployed and run. The tag of each released image must be documented along with the released Docker Image file.

2.4 Download and Management

No major changes are required to the Application download or configuration download parts of the Application.

The maximum size of a Docker Image which can be deployed via the iHost management system will be increased to 300MB.

2.5 Data Upload

2.5.1 Data Input

The Nortech Comms Application will obtain the data to be uploaded from the Data Storage Application via the Data Upload API (see section 8.5 of the LV-CAP API). Once data has been successfully uploaded to the iHost server it must be marked as uploaded in the database so that it is not re-transmitted in future.

2.5.1.1 Data Format

No Change

2.5.2 Data Destination

The data from each LV-CAP system running the Nortech Comms Application must be uploaded as a separate RTU (or multiple virtual RTUs) within the iHost server.

2.6 Configuration

The Nortech Comms Application must be configured via the standard LV-CAP configuration mechanism (see sections 8.2.1 and 9.5 of the LV-CAP API). The configuration is likely to be altered in the course of the OpenLV Trials, so the configuration settings available must be documented alongside the Application.

The configuration is expected to cover the following areas:

- iHost server settings (included where to send the data, and authentication settings).
- Data Selection settings, i.e. which topics are to be uploaded to the iHost server.
- (Optionally) Where data is to be placed in the iHost structure.

2.7 Security

2.7.1 Authentication

The Nortech Comms Application and the iHost server must mutually authenticate each other so that only authorised data uploads occur, and Man-in-the-Middle attacks are prevented.

2.7.2 Confidentiality

Measures must be taken to ensure that the data uploaded remains confidential in transit, to comply with the OpenLV Project Data Protection Strategy.

2.7.3 Audit

As part of the OpenLV Project, a Cyber-Security review of the LV-CAP™ platform and Applications deployed within the project is to be undertaken. The Cyber-Security supplier will be undertaking an audit of the LV-CAP™ platform and it should be expected that this will include an audit of the software Application and associated documentation created by Nortech as part of the project.

3 Timescales

Delivery Date	**2017/07/22**
Prepared by	**Richard Ash**
Date	**2017/07/01**

6 Appendix C – Lucy Electric Application Container

Requirement Specification

Drawing Number	**2626-RQSPC-SHT02-V00.02.01 Lucy Gridkey Sensor Container Requirements.docx**
Project Title	**OpenLV: Lucy GridKey Sensor Container Requirements**
Electronics Systems Project No.	**E717**
Charge Code	**EX267**
Supplier	**Lucy Gridkey**
Request Date	**2017/06/05**

1 Background

EA Technology undertook an InnovateUK Energy Catalyst project with University of Manchester and Nortech Management Ltd to develop a Common Application Framework for LV Network Management. The core software platform developed by EA Technology under the project is called LV-CAP and consists of a number of Docker containers communicating with each other using MQTT.

WPD has now been awarded a Network Innovation Competition (NIC) project to trial the LV-CAP platform on real networks. This project is known as OpenLV. The OpenLV project will use the proven Lucy GridKey MCU520 measurement unit to make electrical measurements of the substation load. To integrate the GridKey MCU520 into the LV-CAP platform Lucy GridKey will create a sensor container for the LV-CAP platform to receive data from the GridKey MCU. This specification details the LV-CAP GridKey Sensor Container required to meet the needs of the OpenLV Trial Programme.

2 Requirements

2.1 Hardware Environment

The OpenLV project hardware consists of an industrial PC based around a dual-core Intel Core i3 processor with 8GB of RAM and a 512GB SSD. This PC provides the processing power and storage for the whole LV-CAP solution. It has two Ethernet ports for network communications:

1. Local Ethernet link to the GridKey MCU520, fitted with an additional Ethernet module.
2. Ethernet link to a stand-alone 4G router which provides wide area network communications.

The GridKey Sensor Container will communicate directly with the GridKey MCU520 via the local Ethernet port. It will not have access to the wide area communications network.

2.2 Base Operating System

The OpenLV project PC will be running 64-bit Ubuntu Server 16.04 LTS with current updates applied.

2.3 LV-CAP Environment

The GridKey Sensor Container must run as a "third party" container on the LV-CAP Platform. This is described in the Public API document 2383-MANUL-V04.02.00, hereafter referred to as "the LV-CAP API".

In order to be deployed via the iHost management system, the maximum size of the GridKey Sensor Container as an uncompressed TAR file is 100MB.

The GUID assigned to this container is "96d6f19b-7022-45f2-b753-cb5012626b4d"

The Docker "repository" string assigned to this container is "lucy/gridkeysensor". Each release of the GridKey Sensor Container should be tagged with this repository string and its version number, separated by a colon. The version number must monotonically increase with each release. For example:

lucy/gridkeysensor:0.1
lucy/gridkeysensor:0.2
lucy/gridkeysensor:1.0
lucy/gridkeysensor:1.1
lucy/gridkeysensor:1.3
lucy/gridkeysensor:2.0

These tags are important as they are used by Docker to load and run the Containers on the LV-CAP. Incorrect tags may result in the incorrect version of the container being deployed and run. The tag of each released container must be documented along with the released Container file.

2.4 Electrical Measurements

The following electrical measurements must be made available to the LV-CAP platform. Each measurement point should be updated at intervals of 10 seconds or less.

The three phases shall be designated "L1", "L2" and "L3" as on the MCU520 hardware. The current measurement channels shall be designated "Feeder1" through "Feeder5" as on the MCU520 hardware.

Other measurement data may be included if available.

2.4.1 Voltage Measurements

At the substation busbars:

- RMS Voltage phase to phase (x3)
- RMS Voltage phase to neutral (x3)

2.4.2 Current Measurements

For each circuit measured:

- RMS current in each phase
- Power factor for each phase
- Real and Reactive power flow each phase (including direction, so reverse power is read as negative current)

2.5 Output Messages

Each of the above measurements must be output as JSON messages on a separate MQTT topic, as described in the Sensor Data API (Section 8.3 of the LV-CAP API). The format of each JSON message shall be the LV-CAP Scalar Object

Format as described in Section 9.1 of the LV-CAP API. The output topic names should be chosen as described in Section 4.4 of the LV-CAP API. A suggested set of topic names is given in Section 4. of this document.

Output messages must be produced at all times as described in Section 4.8 of the LV-CAP API, with the valid flag set appropriately if there is a problem receiving data from the GridKey MCU.

The container may optionally provide measurement metadata as set out in Section 8.3.2 of the LV-CAP API.

2.6 Time

The LV-CAP system clock will be kept set accurately by LV-CAP using a combination of Network Time Protocol and GPS time reference information. The messages output from the GridKey Sensor Container must contain time stamps which are synchronised to this clock to avoid confusion.

2.7 Configuration

The GridKey Sensor Container must be configured via the standard LV-CAP configuration mechanism and a JSON configuration file as described in Section 9.5 of the LV-CAP API. This file can be used to store any relevant configuration parameters for the container. The following parameters must be configurable:

- The IP address of the GridKey MCU.

2.8 GridKey Firmware Update

The implementation of functionality to enable the firmware of the GridKey MCU to be updated in the field would be desirable, but is not required at this stage of the project. It would be acceptable to implement this feature in a future update to the Gridkey Sensor Container. The remote updating of containers on the LV-CAP platform is a core feature which will be proven before deployment of the system, so such an update can readily be deployed to operational LV-CAP systems.

In order to update the firmware on the MCU, two things need to happen:

1. The new firmware file is downloaded over the wide area network to the LV-CAP platform
2. The firmware is transmitted over the local Ethernet connection from the LV-CAP computer to the MCU.

The GridKey Sensor Container can access local file storage on the LV-CAP computer and the Ethernet link to the MCU. The second step (updating the MCU from the local file) should be carried out by the Gridkey Sensor Container.

The GridKey Sensor Container will not have access to the wide area network and so should not perform the first step (downloading firmware). There are several options for delivering the firmware update to the LV-CAP platform:

- The firmware file is packed within the GridKey Sensor Container image. When new firmware needs to be deployed, a new release of the GridKey Sensor Container is made and deployed via the normal LV-CAP mechanism. This will increase the size of the Container image however.
- The firmware file is separately downloaded via the normal LV-CAP mechanism and made available to the GridKey Sensor Container for installation. This would require changes to the LV-CAP APIs and so will be difficult to achieve for this project.
- The firmware file is downloaded by the Data Centre Communications Container, and transferred from that container to the GridKey Sensor Container for installation. This requires a mechanism for the two containers to transfer files between them to be designed and implemented, but does not affect the LV-CAP core.

3 Timescales

Delivery Date	**2017/06/23**
Prepared by	**Richard Ash**
Date	**2017/06/08**

4 Proposed Output Topics

The following output topic names are proposed to comply with the requirements in Section 2.5 of this document. They are based on the assumption that RMS values will be output once per second, and will need to be adjusted if this is not the case.

For the measurements in Section 2.4.1 of this document:

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L1-N_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L2-N_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L3-N_RMS

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L1-L2_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L2-L3_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Busbar/voltage/1s/L3-L1_RMS

For the measurements in Section 2.4.2 of this document:

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/current/1s/L1_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/current/1s/L2_RMS
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/current/1s/L3_RMS

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/power_factor/1s/L1
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/power_factor/1s/L2
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/power_factor/1s/L3

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/real_power/1s/L1
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/real_power/1s/L2
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/real_power/1s/L3

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/reactive_power/1s/L1
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/reactive_power/1s/L2
sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/Feeder1/reactive_power/1s/L3

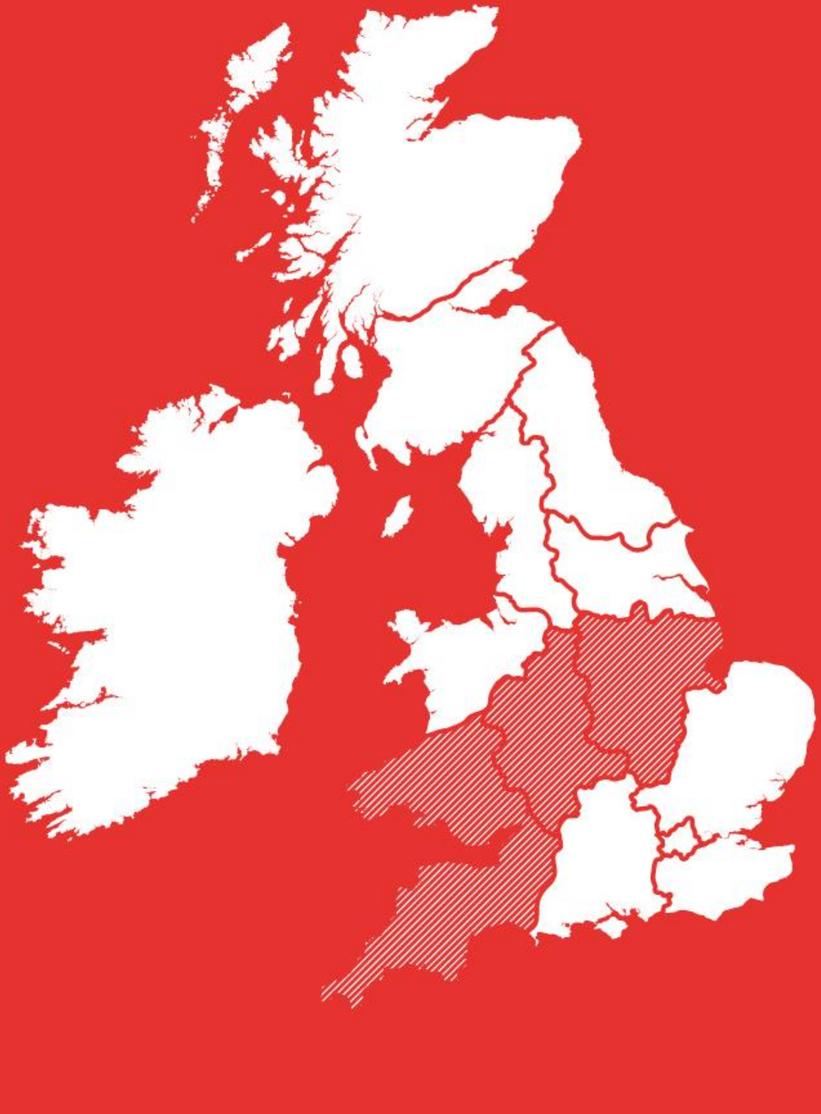
and similarly for each of the 5 feeders supported by the hardware.

Annex 3. Factory Acceptance Test (FAT) Documentation & Results

WESTERN POWER DISTRIBUTION

OPEN LV

OPENING UP THE SMART GRID



Factory Acceptance Tests

Results Documentation

Report Title:	Factory Acceptance Tests – Results Documentation
Report Status:	Issued
Project Ref:	WPD/EN/NIC/02 – OpenLV
Date:	21.09.2017

Document Control

	Name	Date
Prepared by:	Tim Butler	September 2017
Reviewed by:	Richard Ash	September 2017
	Richard Potter	September 2017
Recommended by:	Dan Hollingworth	September 2017
Approved (WPD):	Mark Dale	September 2017

Revision History

Date	Issue	Status
21.09.2017	2.1	Not yet issued – pending FATs Part 2.
16.08.2017	1.0	Issued

DISCLAIMER

Neither WPD, nor any person acting on its behalf, makes any warranty, express or implied, with respect to the use of any information, method or process disclosed in this document or that such use may not infringe the rights of any third party or assumes any liabilities with respect to the use of, or for damage resulting in any way from the use of, any information, apparatus, method, or process disclosed in the document.

Telephone +44 (0) 1332 827446. E-mail wpdinnovation@westernpower.co.uk

Contents

Table of figures

Table of tables

1 Introduction

1.1 Purpose

The testing of the OpenLV solution covers three distinct areas.

- **Factory Acceptance Tests** to verify the equipment meets the requirements detailed in the Requirements Specification.
- **Site Acceptance Tests** to verify the solution meets the requirements in realistic, non-laboratory / controlled environment conditions.
- **Cyber-security testing** to evaluate the cyber-security capabilities of the LV-CAP™ platform; these tests will be undertaken by a specialist provider.

This document details the Factory Acceptance Tests (FATs) for the overall OpenLV solution.

1.2 Scope

The tests in this document are based upon the functionality documented in the OpenLV Requirements Specification. Factory Acceptance Testing (FAT) will be performed using this specification.

1.3 Environment

A representative setup of the OpenLV solution was established in the testing laboratories at EA Technology’s Capenhurst offices.

This setup comprises two OpenLV solutions, connected in an equivalent manner to the planned trial deployments locations under Method 1 for the Project.

1.4 Test Data & Verification

Each test case lists the following:

- The objectives of the overall test;
- The initial conditions;
- A list of numbered actions with their corresponding expected results; and
- A test results record for the overall test including the result, date, name of the tester and name of the witness. White space is left after each test to allow for the recording of comments, issues, etc.

2 Factory Acceptance Tests (FATs)

The FATs are separated into discrete areas, those that test the OpenLV solution (the LV-CAP™ platform and associated hardware) and those that test the applications to be deployed for the provision of trial functionality.

The tests outlined below have been scheduled to minimise repeated tasks and wherever possible, to enable a single action, or sequence of actions to demonstrate that multiple requirements are met where appropriate to do so.

Each test clearly details which requirement(s) it is testing, and the system area(s) it applies to.

It is not intended to undertake tests relating to the cyber-security requirements at the same time as the hardware and functionality tests. Due to the nature of cyber-security testing, particularly penetration testing, specifically the duration required for effective evaluation, and the potential conflict of simultaneous tests being undertaken, these will be appraised separately by NCC Group, the OpenLV Project's cyber-security specialist.

Testing was separated into two sessions due to the availability of some testing elements. Not all tests undertaken in Part 1 were repeated in Part 2 as there was no requirement to repeat tests, given that nothing had changed within that part of the solution in the interim period.

Where further work on the solution between Part 1 and Part 2 FATs had the potential to affect the results, individual FATs were repeated for assurance purposes.

3 Factory Acceptance Tests – Part 1

The Part 1 FATs were conducted on August 16th, 2017.

3.1 Attendees

3.1.1 Western Power Distribution (WPD)

- Mark Dale (MD)

3.1.2 EA Technology

- Richard Potter (RP)
- Richard Ash (RA)
- Tim Butler (TB)

3.1.3 Nortech Management Ltd.

- Julian Brown (JB)
- Lee Fitzpatrick (LP)

3.1.4 Lucy Electric

- Simon Andrews (SA)
- Stuart Brady (SB)
- Jordan Griffiths (JG)

3.2 Setup Details

3.2.1 Login details

Direct access to the LV-CAP™ test platform requires a username and password. These are:

Username:	installer
Password:	LvCAP6wpd

In both cases, these are case sensitive.

To undertake the tests, the following additional computers, beyond the LV-CAP™ platform, will be required:

- 1x laptop to provide direct access to and control the router modems within the enclosure.
- 1x laptop to provide access to the iHost control server.
- 1x laptop containing development tools for the LV-CAP™ platform.

3.3 Hardware checks – Intelligent Substation Device (ISD)

This section covers tests relating to the physical architecture elements of the trial system.

Test	1	
Requirement No.	Must (M:) **059**	Should (S:)
Objective	To confirm the iHost server is installed behind a physical firewall restricting unauthorised access as far as is reasonably practicable.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security ✓	Overall System
Initial condition	The iHost server is physically located within a secure part of EA Technology's Capenhurst offices. The server rack is locked at all times with only select individuals having keys to access the server directly.	
Action(s)	1. Visually verify that a firewall is installed within the iHost server rack and has been connected to the primary and backup servers.	
Expected Result	Firewall confirmed to be present and operational.	
Pass / Fail	Pass	
Comments	No comments or queries. Explanation provided of iHost server and backup setup along with the firewall arrangement and purpose behind it.	

Test	2	
Requirement No.	Must (M:) **011, 089**	Should (S:)
Objective	To confirm the enclosure can be mounted through multiple means; direct mounting; magnetic mounting; securing to floor.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosure on test-bench for inspection of multiple mounting arrangements.	
Action(s)	1. Verify enclosure has capability for multiple mounting arrangements and confirm that brackets have been designed and verified as suitable where necessary. 2. Relocate enclosure to switchgear / transformer equipment to demonstrate magnetic mounting arrangement.	
Expected Result	To be provided with evidence of the wall mount suitability. For magnetic mounting, a frame will be necessary to support some of the weight of the enclosure. Floor mounting will utilise the same mounting points as wall mounting.	
Pass / Fail	Pass	
Comments	No comments or queries.	

Test	3	
Requirement No.	Must (M:) **006**	Should (S:)
Objective	To confirm the enclosure is suitably IP rated for the potential environments into which the trial equipment will be installed.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Not applicable – no change expected.	
Action(s)	1. Verify IP rating of enclosure and that work undertaken to install equipment, modules and glands has not affected the rating. • Data sheet for the enclosure is located in the appendices, Appendix B. • Data sheet for the enclosure gland fitting is located in Appendix C. • Data sheet for the isolation switch is located in Appendix D.	
Expected Result	Datasheet confirms enclosure is of a suitable IP rating (xxx). Installation of equipment, modules, and glands to have been undertaken in accordance with manufacturers' instructions.	
Pass / Fail	Pass	
Comments	Datasheets attached to appendices of this test document.	

Test	4	
Requirement No.	Must (M:) **009**	Should (S:)
Objective	To confirm the enclosure is non-conductive to avoid potential earthing issues.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Not applicable – no change expected.	
Action(s)	1. Verify the enclosure is non-conductive, either through direct demonstration or checking of the product's datasheet.	
Expected Result	Datasheet confirms enclosure is manufactured of a non-conductive plastic.	
Pass / Fail	Pass	
Comments	No comment or queries.	

Test	5		
Requirement No.	Must (M:) **016, 018, 087, 088**	Should (S:)	
Objective	To confirm the Peer-to-Peer Communications Application enables the transfer of selected data sets between two appropriately configured LV-CAP™ platforms and hence confirm that data can be transferred directly between platforms without the need for a centralised data platform.		
System Area	ISD	iHost Control Server	
	Lucy Data Server	LV-CAP™ Platform	
	LV Monitoring	Thermal Monitoring	
	LV Meshing	Load Profile Predictor	
	CSV Data Recorder	Loadsense	
	Dynamic Thermal Rating	Management Communications	
	Data Upload Communications	Peer-to-peer Communications	✓
	Cyber-Security	Overall System	
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server. User with appropriate privileges logged into the Data Centre server.		
Action(s)	1. Ensure the Peer-to-Peer Application is installed on two LV-CAP™ enabled devices / enclosures. 2. Utilise the iHost Control Server to verify the respective applications are configured appropriately to communicate with each other. 3. Ensure the Peer-to-Peer application is active on each platform and set to transfer the 1-minute monitored data between the platforms.		
Expected Result	1-minute data for both platforms will be verified as present on both platforms.		
Pass / Fail	Pass		
Comments	Verified, as pulling the temperature (transformer) data from the primary LV-CAP™ platform (OpenLV-6) to the secondary LV-CAP™ platform (OpenLV-5) at one-minute intervals.		

Test	6	
Requirement No.	Must (M:) **034**	Should (S:)
Objective	To confirm the ALVIN Reclose™ devices are electrically isolated from the ISD enclosure and contents.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. Demonstrate the design for connecting the ALVIN Reclose™ devices to the OpenLV ISD. 2. Verify that connection methodology for ALVIN Reclose™ communication link cable within the enclosure provides suitable isolation properties.	
Expected Result	The connection point for installation of an ALVIN Reclose™ communication cable within the enclosure is suitably isolated. Verified that the data cable connections within the ISD, provide suitable electrical isolation properties.	
Pass / Fail	Pass	
Comments	Brief conversation held around the potential for safe operation and isolation of ALVIN Reclose™ devices in fault conditions and manual network re-arrangements. No issues with the planned approach. Reference Test 8.	

Test	7	
Requirement No.	Must (M:) **003**	Should (S:)
Objective	To confirm presence of a 3G / 4G modem / router within the OpenLV solution, connected to the PC unit and energised appropriately.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Not applicable – no change expected.	
Action(s)	1. Visual check and verification of the module.	
Expected Result	Visual confirmation of the presence of a 3G / 4G modem / router.	
Pass / Fail	Pass	
Comments	No comments.	

Test	8	
Requirement No.	Must (M:) **008**	Should (S:)
Objective	To confirm the demonstrated isolation capability of the ALVIN Reclose™ devices can be 'locked on' in a safe state, such that engineers can be confident that the system will not operate autonomously until the lock is removed.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. Verify connections for ALVIN Reclose™ devices are routed through the isolation switch and that it is possible for this switch to be locked in the off position.	
Expected Result	Communication link can be deactivated without requiring access to the interior of the ISD enclosure and this cannot be overridden unintentionally.	
Pass / Fail	Pass	
Comments	Brief conversation held around the potential for safe operation and isolation of ALVIN Reclose™ devices in fault conditions and manual network re-arrangements. No issues with the planned approach. Reference Test 6.	

Test	9	
Requirement No.	Must (M:) **005, 091**	Should (S:)
Objective	To confirm the enclosure can be physically secured from unauthorised access.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Closed, locked.	
Action(s)	1. Verify that enclosure can be ‘locked’ with a standard T-bar key and can be padlocked.	
Expected Result	Unable to open enclosure without T-bar key and key to alternative means of securing the enclosure if applied.	
Pass / Fail	Pass	
Comments	No comment or queries. Reference Test 14 for details of agreed padlock key ownership to restrict access to enclosure interior.	

Test	10	
Requirement No.	Must (M:) **001**	Should (S:)
Objective	To confirm that the computational hardware is based on PC processing architecture and is within an industrialised PC unit.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Not applicable – no change expected.	
Action(s)	1. Visual check and verification of the platform. 2. Confirm on datasheet (Appendices, section 0), that the processing platform is appropriate architecture.	
Expected Result	Visual confirmation of the presence of an Advantech UNO_2484G_DS platform.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	11	
Requirement No.	Must (M:) **013, 020**	Should (S:)
Objective	To confirm the LV-CAP™ platform is installed and running successfully on the OpenLV industrial PC, and is running the applications deployed to the platform.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform ✓
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. Verify that the LV-CAP™ platform is running on the hardware. Run the command docker ps	
Expected Result	Demonstration that LV-CAP™ is operational: 1. nortech/commscontainer is running 2. lvcapcore/marketplace is running 3. lvcapcore/datastorage is running	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	12	
Requirement No.	Must (M:) **002**	Should (S:)
Objective	To confirm that the computation hardware is capable of running the LV-CAP™ platform, the application containers necessary for the OpenLV Project and those additional applications to be developed for Methods 2 & 3.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. As the software for methods 2 and 3 have not yet been developed, determine the level of processor and memory usage for running the current applications and determine that remaining, unused capability is sufficient for additional applications.	
Expected Result	The majority of CPU and RAM capability to be unutilised during standard operation.	
Pass / Fail	Pass	
Comments	Task manager of the platform shows: Overall CPU utilisation at c1%. Overall RAM usage at c1.5Gb of 8GB. Most of this is 'allocated'; in live use is less than 200mb.	

Test	13	
Requirement No.	Must (M:) **004**	Should (S:)
Objective	To confirm that the internal storage capability of the OpenLV solution is sufficient to store all data capture by the sensors and generated by application containers.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Some storage will be utilised already due to installation of the Linux OS and LV-CAP™ platform and applications. OS Partition is <15GB. Remainder of the drive (c440GB) is allocated to data storage. It is expected that for the LV-CAP™ platform to be deployed under Method 1 at the start of the project deployment phase, (i.e. no applications from community groups or third-party companies), no more than 200 GB of the HDD's 512GB capacity will be utilised.	
Action(s)	1. Check the available capacity on the HDD and verify that there is sufficient capacity for all data capable of being recorded over the 18-month period of the project trials. • On the command line, enter the following command df -h <Enter> 2. Verify that there remains a reasonable margin of additional capacity (in excess of 250GB) in the /home/ file system to allow for data from community and third-party applications.	
Expected Result	1. The internal HDD is partitioned into an OS drive and a database drive. 2. The OS drive is configured to a capacity of c15GB; this will not change and allows for sufficient expansion of the OS if required as part of future updates. 3. The remaining capacity of the HDD (c440 GB) is allocated to data storage. 4. High usage case data requirement estimates for individual platforms are c170GB for the duration of the project under Method 1. A minimum of 250 GB of available capacity on the LV-CAP™ platform's HDD is desirable. In excess of 400GB of space expected.	
Pass / Fail	Pass	

Test	13
Comments	Following initial soak tests at EA Technology and the initial deployment, data usage will be evaluated and checked against forecasts to confirm excessive data isn't being generated by the system. OS drive has 11GB capacity with 3GB in use. Data drive has 446GB capacity with 1.5GB in use with multiple test runs and repeated data samples.

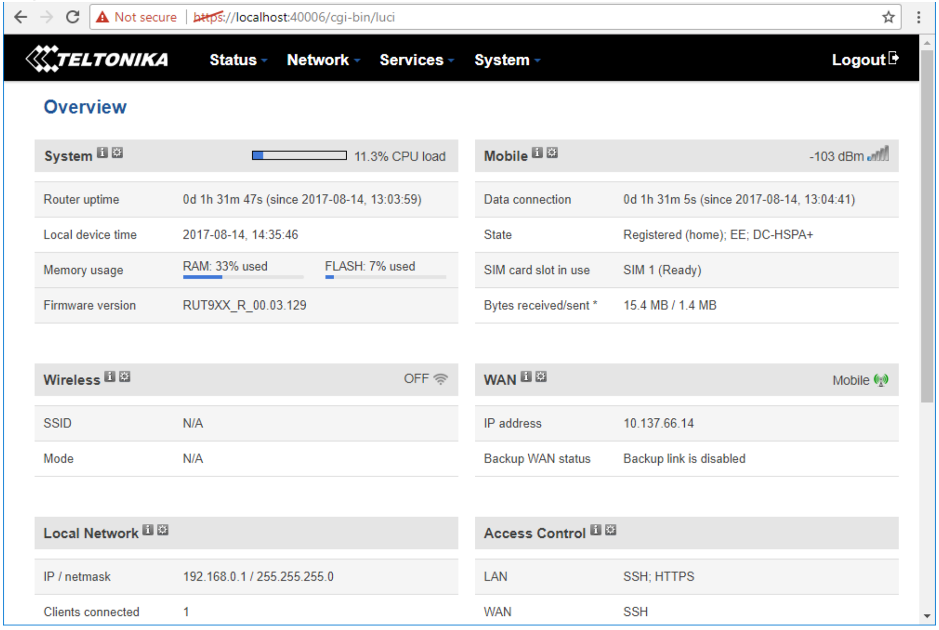
Test	14	
Requirement No.	Must (M:) **012**	Should (S:)
Objective	To confirm the enclosure utilised is suitable for the hardware necessary for the OpenLV Project: industrial PC, modem / router, ancillary connections.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosure on test-bench for inspection.	
Action(s)	1. Verify enclosure contains all necessary equipment, with sufficient space for safe working in and around the enclosure if required. 2. Additionally, confirm that all points for potential electrical hazard are protected from accidental touch.	
Expected Result	Enclosure to contain all required assets and be in a safe configuration, suitable for deployment on the LV network.	
Pass / Fail	Pass	
Comments	MD believes that the enclosure should not be accessible to WPD staff. Agreed that EA Technology shall hold the keys for access to each enclosure. Label on the front will be updated with contact details for key staff within WPD and EA Technology and details of isolation methodology (once agreed with WPD). Need to consider risk of vandalism to the ambient temperature sensor radiation shield.	

Test	15	
Requirement No.	Must (M:) **026, 039, 040, 041, 042 & 043**	Should (S:)
Objective	To confirm the CSV Data Recorder Application monitors the message broker and records all data gathered by the sensors and information published by the other applications on the platform in non-volatile memory.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform ✓
	LV Monitoring	Thermal Monitoring
	LV Meshing ✓	Load Profile Predictor
	CSV Data Recorder ✓	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. After the system has been running for a period of time (at least 48-hours), and other tests as part of the FATs have been undertaken, verify that the data is available within the platform. 2. Run the next test, then return to stage three. 3. Verify the monitored data previously present in the platform is still avail	
Expected Result	CSV files pertaining to the 'soak test and FATs' generated and stored within the LV-CAP™ platform. All data is timestamped appropriately, matching the output rate of the originating application. The originating application can be identified for all data within the file. The CSV file is stored in non-volatile memory.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	16		
Requirement No.	Must (M:)	**010, 090**	Should (S:)
Objective	To confirm that the LV-CAP™ computational hardware within the ISD can be reset without requiring physical access to the enclosure.		
System Area	ISD	✓	iHost Control Server
	Lucy Data Server		LV-CAP™ Platform
	LV Monitoring		Thermal Monitoring
	LV Meshing		Load Profile Predictor
	CSV Data Recorder		Loadsense
	Dynamic Thermal Rating		Management Communications
	Data Upload Communications		Peer-to-peer Communications
	Cyber-Security		Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.		

Action(s)

1. Connect to Wireless Logic SSL VPN (NetExtender) using user ssl_lv
2. Connect SSH session to 4G router using Putty:
   a. IP: see SIM records (10.x.x.x)
   b. Port: 8192
   c. Username: root
3. Forward local port “40006” (e.g.) to “localhost:443”
4. Open a web browser and connect to https://localhost:40006
5. Accept the security error for the self-signed router SSL certificate
6. Log in to web interface with user Admin



Overview

System	11.3% CPU load
Router uptime	0d 1h 31m 47s (since 2017-08-14, 13:03:59)
Local device time	2017-08-14, 14:35:46
Memory usage	RAM: 33% used FLASH: 7% used
Firmware version	RUT9XX_R_00.03.129

Mobile	-103 dBm
Data connection	0d 1h 31m 5s (since 2017-08-14, 13:04:41)
State	Registered (home); EE; DC-HSPA+
SIM card slot in use	SIM 1 (Ready)
Bytes received/sent *	15.4 MB / 1.4 MB

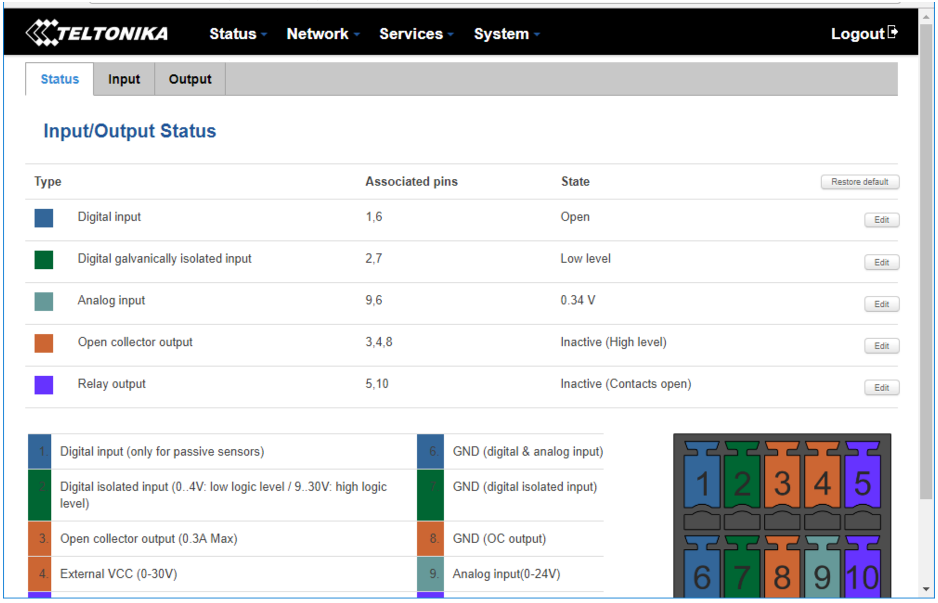
Wireless	OFF
SSID	N/A
Mode	N/A

WAN	Mobile
IP address	10.137.66.14
Backup WAN status	Backup link is disabled

Local Network	
IP / netmask	192.168.0.1 / 255.255.255.0
Clients connected	1

Access Control	
LAN	SSH; HTTPS
WAN	SSH

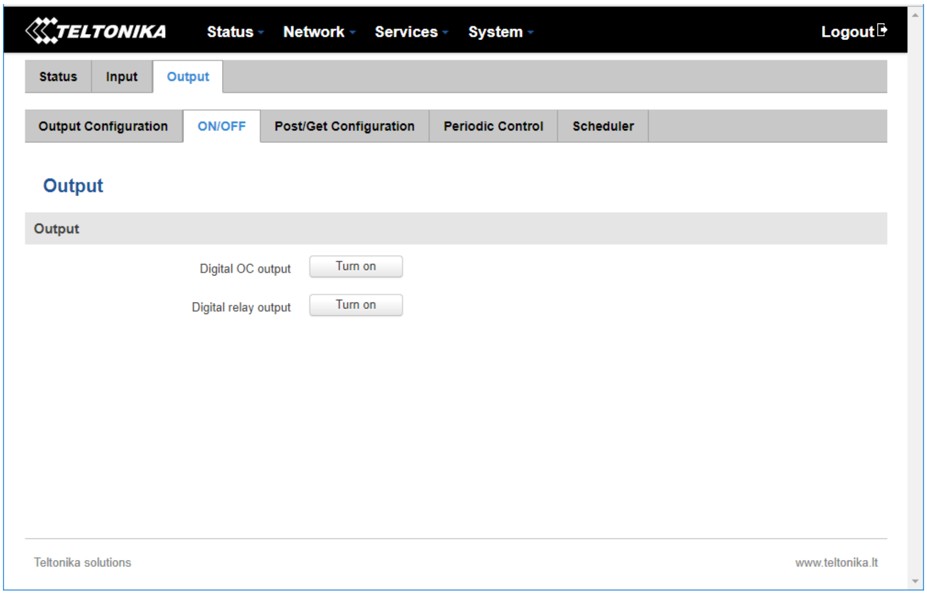
7. Click on Services > Input/Output.



Input/Output Status

	Type	Associated pins	State
🟦	Digital input	1,6	Open
🟩	Digital galvanically isolated input	2,7	Low level
🟦	Analog input	9,6	0.34 V
🟧	Open collector output	3,4,8	Inactive (High level)
🟪	Relay output	5,10	Inactive (Contacts open)

1.	Digital input (only for passive sensors)	6.	GND (digital & analog input)
2.	Digital isolated input (0..4V: low logic level / 9..30V: high logic level)	7.	GND (digital isolated input)
3.	Open collector output (0.3A Max)	8.	GND (OC output)
4.	External VCC (0-30V)	9.	Analog input(0-24V)

Test	16
	8. Click on the Output sub-tab and then the ON/OFF sub-tab.  TELTONIKA Status Network Services System Logout Status Input Output Output Configuration ON/OFF Post/Get Configuration Periodic Control Scheduler Output Output Digital OC output Turn on Digital relay output Turn on Teltonika solutions www.teltonika.lt 9. Click the “Digital OC output” Turn On button. This will remove power from the PC. 10. After the required delay, click the Turn Off button. This will re-apply power to the PC and allow it to start up.
Expected Result	Ruggedised PC will experience a hard shut down, before restarting once power is restored.
Pass / Fail	Pass
Comments	JB noted that it is possible to automate much of the above login process into iHost making accessing the router, if required, much easier.

3.4 Other, non-requirement specific tests

Test	17	
Requirement No.	Must (M:) **092**	Should (S:)
Objective	To confirm that cable connection arrangements (power, thermocouple, and communications) are suitable for use on WPD's network	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection.	
Action(s)	1. Demonstrate the proposed arrangements for providing power to the OpenLV system enclosure. • Mains cable – 1.5mm^2 TRS. Outer diameter 8-9mm. • Ethernet cable. • PVC coated thermocouple cables.	
Expected Result	Proposed cables confirmed as acceptable for use on WPD's LV network as part of the OpenLV trials.	
Pass / Fail	Pass	
Comments	Fused spur would be preferred for powering the enclosure and equipment rather than a 3-pin socket. It should be planned to not use 3-pin sockets even if available. No issues with the proposed power cables (non-armoured) with the expectation that it (along with GridKey data cable) will be routed within trunking, ducting or similar depending on site-specific requirements and availability. Potential for the use of access cylinder to the transformer oil pocket, in situations where selected sites have suitable transformers. MD to take a thermocouple to verify the maximum suitable length of the thermocouple sensor.	

3.5 LV-CAP™ system checks

This section covers tests relating to the LV-CAP™ platform and associated command and control elements.

3.5.1 Control server

Test	18	
Requirement No.	Must (M:) **051a**	Should (S:)
Objective	To confirm that access to the centralised 'command and control system' (the project's iHost server) requires a unique login and password.	
System Area	ISD	iHost Control Server ✓
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security ✓	Overall System
Initial condition	No user logged into the iHost server, web portal onscreen.	
Action(s)	1. Demonstrate that access to the system is unsuccessful without using the correct username and password combination.	
Expected Result	Correct password combination required before access is granted to the iHost interface.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	19			
Requirement No.	Must (M:)	**054, 055, 056, 061, 062, 063, 064, 074, 075, 076**	Should (S:)	
Objective	To confirm it is possible to update and remove application containers to any combination of devices.			
System Area	ISD		iHost Control Server	✓
	Lucy Data Server		LV-CAP™ Platform	✓
	LV Monitoring		Thermal Monitoring	
	LV Meshing		Load Profile Predictor	
	CSV Data Recorder		Loadsense	
	Dynamic Thermal Rating		Management Communications	✓
	Data Upload Communications		Peer-to-peer Communications	
	Cyber-Security		Overall System	
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server. Keyboard and monitor attached to OpenLV system for monitoring, logged in as “installer” user. LV-CAP development tool set available to monitor traffic.			
Action(s)	1. Verify the Modbus Communication Container is running by subscribing to its output topic. Temperature is reported every 10 seconds. Command make sub-tcp 2. Save the Container Manager configuration file from the iHost web interface (we will want it back later). 3. Edit the Container Manager configuration file on iHost and remove the Modus Communication Container section. Save the file. 4. Allow 2 minutes for the configuration file to be downloaded and applied. 5. The temperature messages will stop being reported. 6. Re-upload the original Container Manager configuration file to iHost in the “containers” folder, over-writing the existing file. Edit the file and make a non-change to ensure the timestamp is updated. 7. Allow 2 minutes for the configuration file to be downloaded and applied. 8. The temperature messages will start being reported again.			
Expected Result	1. Console shows steady stream of readings from the temperature sensors. 2. – 3. – 4. Comms Application downloads updated CM configuration. 5. Console output stops. 6. – 7. Comms Application downloads updated CM configuration. 8. Console shows steady stream of readings from the temperature sensors			

Pass / Fail	Pass
Comments	No comment or queries

3.5.2 LV-CAP™ Platform

Test	20			
Requirement No.	Must (M:) **053, 077**		Should (S:) **008**	
Objective	To confirm the OpenLV system's ability to handle a loss-of-power event during a download of an application or configuration updates. To confirm the OpenLV system's ability to successfully download and deploy an application container to the LV-CAP™ platform.			
System Area	ISD		iHost Control Server	✓
	Lucy Data Server		LV-CAP™ Platform	
	LV Monitoring		Thermal Monitoring	
	LV Meshing		Load Profile Predictor	
	CSV Data Recorder		Loadsense	
	Dynamic Thermal Rating		Management Communications	✓
	Data Upload Communications		Peer-to-peer Communications	
	Cyber-Security		Overall System	✓
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server.			
Action(s)	1. Upload a newer (copy) of the Modbus TCP sensor application onto iHost, folder containers and overwriting existing file. 2. Edit the modification time entry in the Container Manager configuration file to reflect the modification time of the new Application image. 3. Watch the log of the Nortech Comms application for the start of the application download. "Downloading new container …" 4. Whilst download is in progress, perform a hard-shutdown of the system through removal of power. 5. After a period of at least 30 seconds re-activate the power and allow the system to restart normally.			
Expected Result	The system should resume or restart the download once communications are re-established and then apply changes once the download is complete.			
Pass / Fail	Pass			
Comments	System took three attempts to download the file on subsequent restart due to time-out on network signal.			

3.5.3 Monitoring capability

Test	21	
Requirement No.	Must (M:) **014**	Should (S:)
Objective	To confirm the OpenLV hardware includes the monitoring equipment to gather the necessary data for project delivery.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System ✓
Initial condition	Enclosure and ancillary equipment on test-bench for inspection.	
Action(s)	1. Verify that ancillary monitoring equipment (GridKey MCU520 and temperature monitoring hardware) is present.	
Expected Result	Appropriate data cables utilised to securely connect each device.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	22			
Requirement No.	Must (M:) **015**		Should (S:)	
Objective	To confirm the LV-CAP™ hardware connects to the monitoring hardware: • MCU520 device; • Temperature monitoring probes.			
System Area	ISD	✓	iHost Control Server	
	Lucy Data Server		LV-CAP™ Platform	✓
	LV Monitoring	✓	Thermal Monitoring	✓
	LV Meshing		Load Profile Predictor	
	CSV Data Recorder		Loadsense	
	Dynamic Thermal Rating		Management Communications	
	Data Upload Communications		Peer-to-peer Communications	
	Cyber-Security		Overall System	
Initial condition	Enclosure and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.			
Action(s)	1. Verify that the MCU520 is connected to the enclosure and subsequently to the LV-CAP™ platform and providing data into the system. 2. Verify that the temperature monitor is connected to the enclosure and subsequently to the LV-CAP™ platform and providing data into the system.			
Expected Result	1. MCU520 connected to the LV-CAP™ platform and providing data to the platform. 2. Temperature monitor connected to the LV-CAP™ platform and providing data to the platform.			
Pass / Fail	Pass			
Comments	No comment or queries			

Test	23	
Requirement No.	Must (M:) **021, 022, 023, 025, 027**	Should (S:)
Objective	To confirm the ISD can monitor: • Voltage (RMS phase to neutral) for three phases; • RMS current for each phase in each circuit monitored; • Power factor for each phase; • Real and reactive power flow in each phase; • Ambient air temperature (indoor and outdoor); • Transformer top oil temperature (if acceptable to WPD). Also, to confirm this data is recorded by the platform.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring ✓	Thermal Monitoring ✓
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	OpenLV hardware energised, running in LV-CAP™ configuration with monitoring equipment connected.	
Action(s)	1. View the data feed within the platform to demonstrate live feed data is being provided by the sensors. 2. Thermal variation of the temperature probes to enact changes. • Use of hot and chilled water to generate temperature swings. 3. Varying the provided current and voltage. 4. Verify the readings generated are time-stamped appropriately.	
Expected Result	1. Output of sensors scrolling at one-minute intervals. 2. Measured temperature for the affected probe rises and falls significantly in accordance with expectations. 3. Monitored voltage and current increase and decrease in accordance with expectations. 4. Each reading is time-stamped appropriately.	
Pass / Fail	Pass	

Comments	Readings produced by the GridKey Platform and verified by separate meters. Temperature readings for thermocouples varied through use of hot and cold water. Pre-deployment commissioning to include verification of the impact of phasing on current readings. Commissioning process must include verification of phase angle at point of installation.

Test	24			
Requirement No.	Must (M:) **024**		Should (S:)	
Objective	To confirm that the complete system (LV-CAP™ platform and MCU520) can record the monitored values at a constant rate of once every ten (10) seconds for a period of at least one hour.			
System Area	ISD	✓	iHost Control Server	
	Lucy Data Server		LV-CAP™ Platform	✓
	LV Monitoring	✓	Thermal Monitoring	
	LV Meshing		Load Profile Predictor	
	CSV Data Recorder		Loadsense	
	Dynamic Thermal Rating		Management Communications	
	Data Upload Communications		Peer-to-peer Communications	
	Cyber-Security		Overall System	
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.			
Action(s)	1. Demonstrate that readings can be monitored and recorded at an (actual and timestamped) rate of six times per minute (once every ten seconds) and that this can be undertaken for a period of not less than one hour.			
Expected Result	Data being monitored by the OpenLV solution, (voltage & current) is stored within the platform at the same time and timestamped appropriately.			
Pass / Fail	Pass			
Comments	Verified that MCU520 platform is outputting at 10-second intervals and being recorded within the system memory.			

Test	25	
Requirement No.	Must (M:)	Should (S:) **003**
Objective	To confirm that the rate of temperature monitoring by the OpenLV solution can be varied from once per minute to once every 10-seconds, 10-second intervals.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform ✓
	LV Monitoring	Thermal Monitoring ✓
	LV Meshing	Load Profile Predictor
	CSV Data Recorder ✓	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. Demonstrate that readings can be monitored and recorded at an (actual and timestamped) rate of six times per minute (once every ten seconds) and that this can be undertaken for a period of not less than one hour.	
Expected Result	Data being monitored by the OpenLV solution, (temperature) is stored within the platform at the same time and timestamped appropriately.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	26	
Requirement No.	Must (M:) **038**	Should (S:)
Objective	To confirm the Load Profile Predictor application generates forecast load profiles for the next 24-hour period, at a frequency of once per hour.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform ✓
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor ✓
	CSV Data Recorder ✓	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	1. Load the system with sufficient data (more than 28-days) of load data. 2. Continue to provide data to replicate the availability of load data for the application. 3. Verify that load profiles have been generated on an hourly basis over the period in question.	
Expected Result	Load profiles generated on an hourly basis and were published on the message broker within the LV-CAP™ platform for use by other applications and storage in non-volatile memory.	
Pass / Fail	Not tested – data not gathered sufficiently to generate calculated outputs.	
Comments	Not tested – data not gathered sufficiently to output calculate outputs. Test to be repeated in FATs part 2.	

Test	27	
Requirement No.	Must (M:) **060, 071, 072, 073, 078**	Should (S:)
Objective	To confirm that data uploaded to the iHost Control Server is stored such that the original source of the data (deployed LV-CAP™ platform) is readily identifiable. Uploaded data should be marked at 'uploaded' to prevent unnecessary data usage.	
System Area	ISD	iHost Control Server ✓
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications ✓
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server.	
Action(s)	1. Verify that data within the iHost system details the LV-CAP™ platform the data originated from. 2. Install SQLite client on the LV-CAP™ platform to enable direct access to the database storing files on the platform. (This is not required for standard BAU operation but is required to undertake this test.) • Install SQLite client with the command sudo apt install sqlite3 • Open the DB sudo sqlite /home/CM/Database/ContainerDB.db • SQL Command - File is attached, now read the contents .read ./uploaded_flag_test.txt 3. This will only show data that has been uploaded to the iHost server since the last output.	
Expected Result	Data is uploaded to the iHost server by the LV-CAP™ platform on a regular basis. This will be available within the iHost server once each periodic upload is complete.	
Pass / Fail	Pass	
Comments	No comment or queries	

Test	28	
Requirement No.	Must (M:)	Should (S:) **007**
Objective	To confirm the OpenLV system's ability to withstand multiple power outages in close succession as may occur in the case of an intermittent fault.	
System Area	ISD ✓	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server.	
Action(s)	1. Ensure the system is operational and online. 2. Perform a hard shut-down through removal of power from the ISD enclosure. 3. Wait approximately 90 seconds then re-activate the power for 5 – 10 seconds, before de-activating the power again. 4. Wait approximately 90 seconds then re-activate the power for 5 – 10 seconds, before de-activating the power again. 5. Wait approximately 90 seconds then re-activate the power for 5 – 10 seconds, before de-activating the power again. 6. Wait for the system to fully restart and begin normal operation before commencing access via the iHost Control Server. 7. Verify the platform has restarted successfully and has begun monitoring again.	
Expected Result	System should restart after final sequence and resume normal operation.	
Pass / Fail	Pass	
Comments	No comment or queries	

3.6 Post-FATs meeting.

1 RP emphasised, supported by MD, that the SDRC document must be issued to WPD by the end of September 2017.

2 LUCY: RP requires that the Lucy Electric Cloud Data Upload Communication Application be completed and having passed all FAT tests for inclusion within the SDRC documentation.

3 RP, RA, TB to discuss Method 1: Loadsense & Weathersense applications.

a) ACTION: Weathersense: RP to agree commercial terms with University of Manchester or commit to implementing an equivalent application internally.

b) ACTION: Loadsense: TB to provide indicative, planned logic to MD.

4 ACTION: TB to provide proposed method statements for installation to MD.

5 ACTION: MD to provide example labels for use on the enclosures.

6 ACTION: MD to test maximum temperature probe length for uses with transformers.

7 ACTION: RA to verify all configuration files for each platform pre-issue.

8 Agreed that multiple attendees required for a meeting with Mike Gees 'penetration test experts' at some point in last two weeks of October.

a) EA Technology: RP, **RA** & TB

b) Nortech Management Ltd: JB, **SH**

c) Lucy Electric: To be confirmed

9 ACTION: RP, RA & TB to agree appropriate cyber-security tests are undertaken prior to equipment deployment.

MD & Andy Hood coming to Capenhurst on September 11th and 12th; ideally want to talk to them about the Loadsense logic if practical to achieve in time available.

3.7 Sign-off and acceptance

It is acknowledged by all those in attendance at the Factory Acceptance Tests (FATs) undertaken on the OpenLV LV-CAP™ Trial system at EA Technology's Capenhurst offices on 16th August ~~17th~~, 2017, that the results and comments detailed against each test in this document are a true record of the tests outcome.

The tests were witnessed by representatives of the below companies:

- Western Power Distribution
- EA Technology
- Nortech Management Ltd.
- Lucy Electric

Name	Company & Role	Signature
Mark Dale	Innovation and Low Carbon Networks Engineer Western Power Distribution	M. c Dale.
Richard Ash	Senior Consultant EA Technology	
Richard Potter	Principal Consultant EA Technology	
Tim Butler	Senior Consultant EA Technology	
Julian Brown	Managing Director Nortech Management Ltd.	
Simon Andrews	Senior Software Engineer Lucy Electric	

4 Factory Acceptance Tests – Part 2

The Part 2 FATs were conducted on September 21st, 2017.

4.1 Attendees

4.1.1 Western Power Distribution (WPD)

- Mark Dale (MD)

4.1.2 EA Technology

- Richard Potter (RP)
- Richard Ash (RA)
- Tim Butler (TB)
- Stephen Need (SN) / Piotr Przesmycki (PP)

4.1.3 Lucy Electric

- Stuart Brady (SB)

4.2 Setup Details

4.2.1 Login details

Direct access to the LV-CAP™ test platform requires a username and password. These are:

- Username: installer
- Password: LvCAP6wpd

In both cases, these are case sensitive.

To undertake the tests, the following additional computers, beyond the LV-CAP™ platform, will be required:

- 1x laptop to provide direct access to and control the router modems within the enclosure.
- 1x laptop to provide access to the iHost control server.
- 1x laptop containing development tools for the LV-CAP™ platform.

4.3 LV-CAP™ system checks

4.3.1 Monitoring capability

Test 26, scheduled for the Part 1 FATs was unable to be implemented on August 16th due to insufficient data gathered by the platform in advance of the tests, preventing the load profile predictor application from generating forecast profiles. The test is repeated to be undertaken as part of the Part 2 FATs.

Test	29	
Requirement No.	Must (M:) **038**	Should (S:)
Objective	To confirm the Load Profile Predictor application generates forecast load profiles for the next 24-hour period, at a frequency of once per hour.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform ✓
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor ✓
	CSV Data Recorder ✓	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration.	
Action(s)	Follow the process defined in Appendix G - Load profile test.	
Expected Result	Load profiles generated on an hourly basis and were published on the message broker within the LV-CAP™ platform for use by other applications and storage in non-volatile memory.	
Pass / Fail	Pass	
Comments		

4.3.2 Lucy Electric 'cloud based' data server

Test	30	
Requirement No.	Must (M:) **070**	Should (S:)
Objective	To confirm the cloud based data server provided by Lucy Electric is a separate system to their business-as-usual operational systems.	
System Area	ISD	iHost Control Server
	Lucy Data Server ✓	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	A representative Open LV remote cloud based data server is up and running. Access is provided to a browser with an appropriate extension to allow for authorisation headers to be specified.	
Action(s)	1. Access the API using "https://test.gridkey.uk/v1" 2. Take note of the returned Keyspaces 3. Note that the URL does not contain a domain reserved for the business-as-usual GridKey Data Centre	
Expected Result	The list of Keyspaces configured on the Cassandra node will be output. Each customer or project has their data stored in its own unique Keyspace in a Cassandra database. Therefore, as this is a separate system to the GridKey business-as-usual server, the only Keyspaces available should be the one allocated to "OPENLV01". Also, the domain is "test.gridkey.uk" is not the same one allocated to the business-as-usual customer GridKey Data Centre, which is "customer.gridkey.co.uk"	
Pass / Fail	Pass	
Comments		

Test	31	
Requirement No.	Must (M:) **065, 066, 067, 079, 080, 081, 084, 085, 086**	Should (S:)
Objective	To confirm that it is possible to upload a subset of the data, and all of the data, on an LV-CAP™ platform to the Lucy Electric GridKey Data Centre and hence confirm the GridKey Data Centre can receive the data provided by Data Upload Applications installed on the disparate LV-CAP™ enabled platforms deployed for the trials. To confirm that data stored within the Lucy Data Centre can be accessed via an API interface or a web-portal. To confirm the Data Upload Application can manage a loss of communications during file upload to the Data Centre Server, and resume the data transfer once communication links are restored. Once successfully uploaded to the Lucy Data Centre, data should be marked as ‘uploaded’ within the platform by the Management Communications Application and hence will not be automatically uploaded again. It should be possible to demonstrate that data uploaded to the Data Centre Server is stored such that the original source of the data (deployed LV-CAP™ platform) is readily identifiable.	
System Area	ISD	iHost Control Server
	Lucy Data Server ✓	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications ✓	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration. User with appropriate privileges logged into the controlling iHost server and route / modem.	
Action(s)	1. Verify the configuration settings for the Lucy GridKey Data Upload Application and identify the data fields marked for upload to the data server. 2. Then select all data stored on the platform and mark it for upload to the server. 3. Check the data centre to confirm the selected data is being uploaded. 4. During the upload process, terminate the mobile connection through a remote reset of the router modem then allow the platform to reconnect automatically. 5. Verify success of data upload. a. Access the data server via the API interface and verify the presence of the uploaded data. b. Access the data server via the web portal and verify the presence of the uploaded data. 6. Verify it is possible to determine which RTU uploaded the data. 7. Ensure that subsequent data uploads do not repeat data transmission.	

Test	31
Expected Result	Once the modem has re-established a connection to the mobile network, the Data Upload application will resume the transfer of both data and application. Identified data fields present within the Lucy Data Centre after a reasonable period of time to allow for data transfer. It should be possible to identify, select and manipulate data uploaded to the server on a basis of individual or multiple specific LV-CAP™ platforms. Access to the uploaded data will be possible via both the API interface and web portal.
Pass / Fail	Pass. All objectives met except the ability to recover from a loss of comms (the original approach was to remove power rather than just comms) as this caused reboot issues.
Comments	Identified a failure of the previously successful Test 16. The test to determine the system ability to handle loss of communications, identified an issue during system recovery from a hard reboot where the operating system layer requests user input for file recovery and restoration on reboot. This did not occur during previous tests where a hard reboot was initiated and the overall system must be capable of self-recovery in the event this happens in the field. EA Technology to undertake evaluation of the root cause and means to ensure the system can successfully restart autonomously in the future. Test was repeated utilising the approach of removing the network connection between the PC and router then restoring the connection. This approach to the test was successful.

4.3.3 ALVIN Reclose™ connectivity

Test	32	
Requirement No.	Must (M:) **017, 028**	Should (S:)
Objective	To confirm the LV-CAP™ hardware connects to the ALVIN Reclose™ devices. To confirm that the LV Network Meshing Application enables communications between the LV-CAP™ platform and connected ALVIN Reclose™ devices.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing ✓	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration with ALVIN Reclose™ devices disconnected from the other equipment.	
Action(s)	Follow the process defined in Appendix H - Data Reading Test.	
Expected Result	ALVIN Reclose™ devices successfully connected to the LV-CAP™ platform and providing data to the platform. The LV-CAP™ Platform demonstrates a working connection between the OpenLV solution equipment and the ALVIN Reclose™ devices.	
Pass / Fail	Pass	
Comments		

<table>
<tr><td><b>Test</b></td><td colspan="2">33</td></tr>
<tr><td><b>Requirement No.</b></td><td>Must (M:) <b>029</b></td><td>Should (S:)</td></tr>
<tr><td><b>Objective</b></td><td colspan="2">To confirm that the LV Network Meshing Application enables the transfer of selected sets of data from the ALVIN Reclose™ devices where connected.</td></tr>
<tr><td rowspan="8"><b>System Area</b></td><td>ISD</td><td>iHost Control Server</td></tr>
<tr><td>Lucy Data Server</td><td>LV-CAP™ Platform</td></tr>
<tr><td>LV Monitoring</td><td>Thermal Monitoring</td></tr>
<tr><td>LV Meshing ✓</td><td>Load Profile Predictor</td></tr>
<tr><td>CSV Data Recorder</td><td>Loadsense</td></tr>
<tr><td>Dynamic Thermal Rating</td><td>Management Communications</td></tr>
<tr><td>Data Upload Communications</td><td>Peer-to-peer Communications</td></tr>
<tr><td>Cyber-Security</td><td>Overall System</td></tr>
<tr><td><b>Initial condition</b></td><td colspan="2">Enclosures and ancillary equipment on test-bench for inspection.
OpenLV hardware energised, running in LV-CAP™ configuration with ALVIN Reclose™ connected.</td></tr>
<tr><td><b>Action(s)</b></td><td colspan="2">Follow the process defined in Appendix H - Data Reading Test.
Variable details below.
<table>
<tr><th>Register Name</th><th>Quantity</th></tr>
<tr><td>MIR_BUS_VOLTAGE_RMS</td><td>Busbar voltage</td></tr>
<tr><td>MIR_CABLE_VOLTAGE_RMS</td><td>Cable voltage</td></tr>
<tr><td>MIR_LINK_CURRENT_RMS</td><td>Current through ALVIN</td></tr>
<tr><td>MIR_OPEN_OPERATIONS</td><td>Number of times the circuit breaker has opened</td></tr>
<tr><td>MIR_CLOSE_OPERATIONS</td><td>Number of times the circuit breaker has closed</td></tr>
<tr><td>MIR_WATCHDOG_FAULTS_DETECTED</td><td>Number of times the ALVIN watchdog has operated</td></tr>
<tr><td>MIR_CHIP_TEMPERATURE</td><td>ALVIN CPU temperature</td></tr>
<tr><td>MIR_REACTIVE_POWER</td><td>Reactive power</td></tr>
<tr><td>MIR_ACTIVE_POWER</td><td>Active power</td></tr>
<tr><td>MIR_UPTIME_HIGH</td><td>Uptime counter</td></tr>
<tr><td>MIR_SWITCH_TEMPERATURE</td><td>ALVIN CB temperature</td></tr>
<tr><td>MHR_SHADOW_FAULT_STATUS</td><td>Fault status flags</td></tr>
</table></td></tr>
<tr><td><b>Expected Result</b></td><td colspan="2">Data for the variables detailed above is provided to the message broker.</td></tr>
</table>

Test	33
Pass / Fail	Pass`
Comments	

Test	34	
Requirement No.	Must (M:) **030**	Should (S:)
Objective	To confirm that the data from the ALVIN Reclose™ devices is capable of being recorded at a frequency of at least once per minute, in line with other data values.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing ✓	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Enclosures and ancillary equipment on test-bench for inspection. OpenLV hardware energised, running in LV-CAP™ configuration with ALVIN Reclose™ connected.	
Action(s)	Follow the process defined in Appendix H - Data Reading Test	
Expected Result	Variables provided by the LV Network Meshing Application are stored within the platform at the same time and timestamped appropriately.	
Pass / Fail	Pass	
Comments		

Test	35	
Requirement No.	Must (M:) **007, 031, 032, 033, 036 & 037**	Should (S:)
Objective	To confirm that the LV-CAP™ platform can cause the ALVIN Reclose™ relay to open and close as required. To confirm the OpenLV system's ability to autonomously control connected ALVIN Reclose™ devices can be disabled.	
System Area	ISD	iHost Control Server
	Lucy Data Server	LV-CAP™ Platform
	LV Monitoring	Thermal Monitoring
	LV Meshing ✓	Load Profile Predictor
	CSV Data Recorder	Loadsense
	Dynamic Thermal Rating	Management Communications
	Data Upload Communications	Peer-to-peer Communications
	Cyber-Security	Overall System
Initial condition	Connect an LV-CAP™ platform enclosure to the ALVIN Reclose™ test rig and energise the system. OpenLV hardware energised, running in LV-CAP™ configuration with ALVIN Reclose™ connected and circuit breakers in 'open' configuration.	
Action(s)	Follow the process defined in Appendix I - ALVIN Control Test.	
Expected Result	That: 1. The light bulb within the ALVIN test rig will go out as power is removed from the circuit. 2. The circuit will be re-energised, demonstrated by the light bulb being turned on again.	
Pass / Fail	Pass	
Comments		

4.4 Actions from the FATs Part 2

1. EA Technology to identify the root causes and fixes to the additional errors identified under Teste 31.

4.5 Sign-off and acceptance

It is acknowledged by all those in attendance at the Factory Acceptance Tests (FATs) undertaken on the OpenLV LV-CAP™ Trial system at EA Technology's Capenhurst offices on September 21st, 2017, that the results and comments detailed against each test in this document are a true record of the tests outcome.

The tests were witnessed by representatives of the below companies:

- Western Power Distribution
- EA Technology
- Lucy Electric

Name	Company & Role	Signature
Mark Dale	Innovation and Low Carbon Networks Engineer Western Power Distribution	M. e Dale.
Richard Ash	Senior Consultant EA Technology	
Richard Potter	Principal Consultant EA Technology	
Tim Butler	Senior Consultant EA Technology	
Stuart Brady	Principal Software Engineer Lucy Electric	

5 Appendices

The appendices to this document are:

- Appendix A - Factory Acceptance Test setup
- Appendix B - Data sheet – ISD Enclosure
- Appendix C - Data sheet – Enclosure glands
- Appendix D - Data sheet – Isolation switch
- Appendix E - Data sheet – LV-CAP™ PC platform
- Appendix F - Container Manager Manual – Document reference 2358
- Appendix G - Load profile test

The aim of the Load Profile configuration is to produce a predicted half hourly load current forecast for the next 24 hours. The method adopted for each half hour is to calculate the RMS load during the same half hour on the same day of the week in the previous four weeks. Thus the forecast for 00:00 to 00:30 on day twenty-nine is calculated as the RMS of the load currents between 00:00 and 00:30 on days one, eight, fifteen, twenty-two. The calculations are carried out per-phase for each of the three phases.

Inputs

The Application will take its data input by subscribing to MQTT topics

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l2/current-mean

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l3/current-mean

Payloads in Scalar Object Format (as defined in API section 9.1) will be published on these topics every 60 seconds, containing the measured RMS load current in each phase in Amps (API section 4.5). This is the measurement data from the Lucy GridKey Sensor Application

For testing data will be sent on the same topics but in faster than real time. The following data will be sent, the same for each day on each phase:

- L1: Load Curve G, peak current 100A
- L2: Load Curve 33A, peak current 200A
- L3: Constant current of 300A

This data is contained in the 28 day input data file input-28day.csv and the 1 day input data file which follows it input-1day-w5d1.csv.

Procedure for creating the test data set (for reference only, use above files for testing).

1. Open ODF data generator spreadsheet in LibreOffice Calc
2. Adjust settings to set desired curves, peak currents and start date/time (in UTC)
3. Make sure export sheet is selected sheet.
4. File > Save a Copy …
5. Change Save as type to Text CSV

6. Enter file name and click Save
7. In export dialogue:
    a. Character set: Western Europe
    b. Field delimiter: ,
    c. Text delimiter: "
    d. Tick Save cell content as shown and untick the others
    e. Click OK
8. When it warns only the selected sheet was saved, click OK.
9. This produces a 7 day input file, repeat for more days and edit files together, or chop out the required section for shorter files.

To feed the 28 day input into the system at 3600 times real time (one hour per second) use the command

```
./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt
--key eatl_tlsdevtools.key --no-store -i input-28day.csv -f 3600
```

To feed the 1 day input into the system at 60 times real time (one minute per second) use the command

```
./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt
--key eatl_tlsdevtools.key --no-store -i input-1day-w5d1.csv -f 3600
```

Configuration

Reference: 2662-MANUL-S001-Vxx.yy.zz

For the Load Profiler the following global settings are used:

Debug: true (turn debug output on)

RefreshRate: 2 (This determines how often outputs can be produced. This setting is in real time regardless of the timestamps in input data).

MaxDataWait: 300 (Data will come in much more often than this in testing).

One profile is configured:

ProfileName: PredictedTxLoad (name for output data)

ProfileCircuit: Transformer(what is this for?)

ProfileTimeSlotAlgorithm: SquaredAverage (we want to use the RMS / geometric mean of the load current values to give the correct heating effects in rating calculations).

ProfileLateDataLimitInSeconds: 120 (we will be feeding in 60 second interval data)

ProfileDailyTimeSlots: 48 (to give 30 minute interval slots)

ProfileWindowLengthDays: 28

Inputs:

value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l2/current-mean

value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l3/current-mean (three inputs for three phases, from the sensors defined above)

Output Settings:

ProfileOutputTopicString: "tx-load/predict/byday/l1/30/30/1440" (name of the sub-topic the data will be published on)

ProfileOutputIntervalMinutes: 30 (output a new profile every 30 minutes)

ProfileOutputPredictionLengthMinutes: 1440 (24 hour prediction)

ProfileOutputSmoothing: RMS (we want to calculate the RMS / geometric mean of the days)

ProfileOutputValues: value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean (only one input, from the sensor defined above)

The same arrangement is repeated for l2 and l3.

LV-CAP is configured to run the Load Profiler and CSV logger applications only. The CSV logger is configured to log the sensor topics to a single CSV file

Outputs

The output profiles must be published on the topics:

- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l1/30/30/1440
- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l2/30/30/1440
- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l3/30/30/1440

The Application must publish profiles in the standard LV-CAP time series JSON format to the above MQTT topic. Each profile produced must start at approximately the current time (based upon the time stamps of the incoming data, not the host system clock). Each profile produced will have 48 elements, each of 30 minutes duration.

To convert timestamps into human-readable UTC time use a command line

```
date -u -d '@1503966600' "+%Y/%m/%d %H:%M:%S %Z"
```

Test Steps

1. Start from a clean LV-CAP core system with no data files or load profile data present.
2. Deploy the TLS Development tools, which include the play_csv.py script. Install any necessary python packages to run the script against the LV-CAP core.
3. Deploy and configure the Load Profiler and Data Storage applications onto the system.
4. Use make sub-sensor to monitor sensor data on the system

5. Play the 28-day history file into the system to build up the profile data (this takes ~12 minutes):
   ./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key --no-store -i input-28day.csv -f 3600
6. When the 28-day data set completes, also monitor the algorithm output with make sub-alg
7. Continue to play back the 1 day file at slower speed to see the profiles being produced:
   ./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key --no-store -i input-1day-w5d1.csv -f 3600
8. Examine the profiles produced and check as documented below.

Expected Outputs

The 1 day data file we are using for testing contains data for Tue 29 Aug 2017, so we are looking for the outputs from this day to be produced.

The first three outputs will be produced almost immediately, for l1, l2 and l3 respectively.

- Each will have the same TimeStamp values of 1503966600, which is 2017/08/29 00:30:00 UTC, the first profile of the day.
- TimeStampStart will have a value of 1501547400, which is 2017/08/01 00:30:00 UTC, the date and time 28 days earlier when the oldest data used in the profile was collected.
- On the L1 output, the values are for load curve G at 100A peak, which starts with an hour of 100A, then 96.8A for the next hour and then 93A, ending up with a hour of 99.2A
- On the L2 output, the values are for load curve 33A at 200A peak, which starts with 187A for one hour (two profile steps), then 177.4A and 168.2A, ending up at 200A
- On the L3 output, the values are all 300A because the load does not change!

Now move on to the second set of profiles produced.

- Each will have a TimeStamp value of 1503968400, which is 2017/08/29 01:00:00 UTC, half an hour after the previous profiles.
- TimeStampStart has also moved on to 1501549200 (2017/08/01 01:00:00 UTC).
- On L1 output, things have moved up by half a hour, so we get only one step at 100A, two each at 96.8A and 93A. We end with two at 99.2A and then the 100A value wrapped from the start.
- The same on L2, one step at 187A, two each at 177.4A and 168.2A, ending with two at 200A and one at 187A
- On the L3 output, the values are all 300A because the load does not change!

We can keep doing this for as long as you like, but you get the picture!

- Data Reading Test
- Appendix I - ALVIN Control Test

Appendix A. Factory Acceptance Test setup

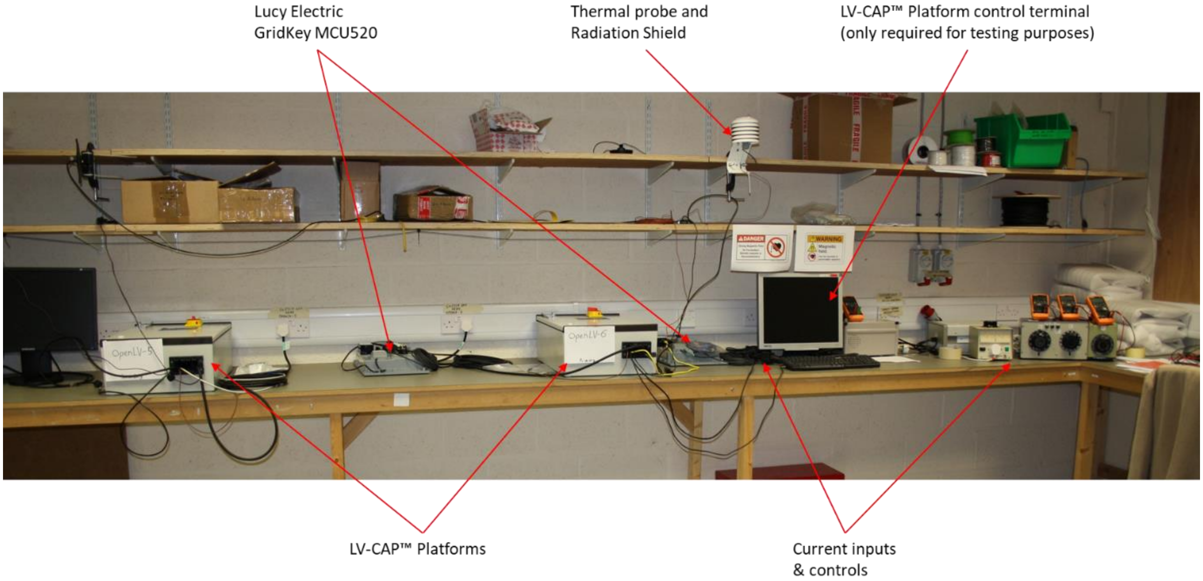


Figure 1: FAT Setup

Appendix B. Data sheet – ISD Enclosure

Enclosure_DataSheet.pdf

Plastic enclosures KS

System accessories Page 507 **KS for potentially explosive areas** Page 194

Material:
- Enclosure and door: Fibreglass-reinforced unsaturated polyester
- Door: All-round foamed-in PU seal
- Mounting plate: Sheet steel
- Viewing window: Glazed acrylic, 3.0 mm with all-round rubber cable clamp strip

Surface finish:
- Enclosure and door: Dyed plastic with no after-treatment
- Mounting plate: Zinc-plated

Colour:
- Similar to RAL 7035

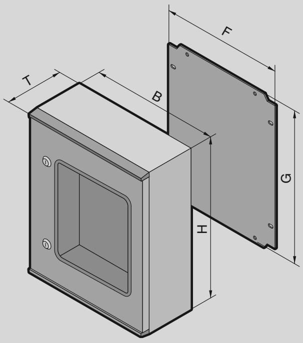
Supply includes:
- Enclosure with hinged door, of all-round solid construction, 3 mm double-bit lock
- Mounting plate
- Twin seal on the top and bottom edges of the door as integral rain protection strip
- Press-fitted C sections at the sides for infinitely variable mounting plate depth adjustment

Please note the product-specific scope of supply.

Note:
- Under the influence of long-term UV radiation (sunlight) in conjunction with wind and rain, the surface finish may become visually impaired. This does not affect the protection of the installed electrical components in any way. If the enclosures cannot be protected from UV radiation, we recommend that they should be painted with a PUR paint. Ambient temperature -30°C...+75°C.

Approvals:
- TÜV
- Germanischer Lloyd
- Russian Maritime Register of Shipping
- Lloyds Register of Shipping
- Bureau Veritas

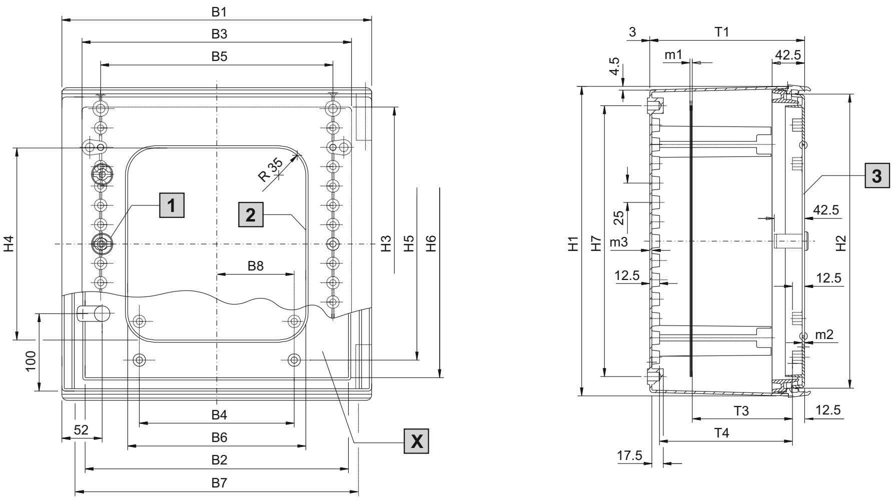
Technical details:
Available on the Internet



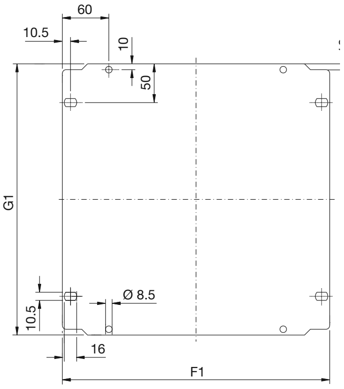
Width (B) mm	Packs of	200	250	300	400	400	400	400	500	Page
Height (H) mm		300	350	400	400	400	600	600	500	
Depth (T) mm		150	150	200	200	200	200	200	300	
Mounting plate width (F) mm		145	195	245	345	345	345	345	417	
Mounting plate height (G) mm		250	300	350	350	350	550	550	450	
Mounting plate thickness mm		2.0	2.0	2.0	2.5	2.5	2.5	2.5	2.5	
Model No.	1 pc(s).	**1423.500**	**1432.500**	**1434.500**	**1444.500**	**1448.500**	**1446.500**	**1449.500**	**1453.500**	
Weight kg		3.4	4.3	5.9	7.9	8.0	11.5	11.2	13.5	
Protection category IP to IEC 60 529		IP 66	IP 66	IP 66	IP 66	IP 56	IP 66	IP 56	IP 66	
Protection category NEMA		NEMA 4X	NEMA 4X	NEMA 4X	NEMA 4X	NEMA 12	NEMA 4X	NEMA 12	NEMA 4X	
Product-specific scope of supply										
Door(s)		1	1	1	1	1	1	1	1	
Door hinged on the right, may be swapped to the left		■	■	■	■	■	■	■	■	
Removable centre bar, lockable door on right		–	–	–	–	–	–	–	–	
Viewing window		–	–	–	–	■	–	■	–	
Cam lock		1	1	2	2	2	2	2	2	
3-point lock system		–	–	–	–	–	–	–	–	
Accessories										
Mounting plate adjustment bracket	4 pc(s).	1481.000	1481.000	1481.000	1481.000	1481.000	1481.000	1481.000	1491.000	594
Threaded inserts M6	20 pc(s).	1482.000	1482.000	1482.000	1482.000	1482.000	1482.000	1482.000	1482.000	625
Pole clamp	1 set(s)	2584.000	2584.000	2584.000	2584.000	2584.000	2584.000	2584.000	2584.000	589
Wall mounting bracket	4 pc(s).	1483.010	1483.010	1483.010	1483.010	1483.010	1483.010	1483.010	1483.010	588
Lock systems		from page	from page	from page	from page	from page	from page	from page	from page	560

Plastic enclosures KS

KS 1423.500, KS 1432.500
with only one cam lock in the centre



Mounting plate



B7 = Separation width for wall mounting hole

H7 = Separation height for wall mounting hole

1 Only for KS 1423.500, KS 1432.500

2 Viewing window only with KS 1448.500, KS 1449.500, KS 1454.500, KS 1467.500

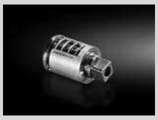
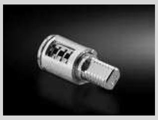
3 Material thickness of viewing window: 3 mm

X Door interior view

Model No. KS	Width dimensions mm								Height dimensions mm							Depth dimensions mm			Material thickness mm			Mounting plate mm	
	B1	B2	B3	B4	B5	B6[1]	B7	B8	H1	H2	H3	H4[1]	H5	H6	H7	T1	T3	T4	m1	m2	m3	F1	G1
1423.500	200	140	150	–	100	–	150	25	300	280	256	–	200	245	250	150	80 – 110/117	119	2.0	3.0	3.0	145	250
1432.500	250	190	200	75	150	–	200	50	350	330	306	–	250	295	300	150	80 – 110/117	119	2.0	3.0	3.0	195	300
1434.500	300	240	249	100	200	–	250	50	400	380	355	–	300	345	350	200	80 – 160/167	169	2.0	3.0	3.0	245	350
1444.500/ 1448.500	400	340	348	200	300	230	350	100	400	380	354	250	300	345	350	200	80 – 159/166	168.5	2.5	3.2	3.2	345	350
1446.500/ 1449.500	400	340	348	200	300	230	350	100	600	580	554	450	500	545	550	200	80 – 158/165	168	2.5	3.5	3.5	345	550
1466.500/ 1467.500	600	540	548	400	500	430	550	200	600	580	554	450	500	545	550	200	80 – 158/165	168	2.5	3.5	3.5	545	550
1453.500/ 1454.500	500	440	434	300	400	330	450	150	500	480	454	350	400	445	450	300	80 – 258/265	268	2.5	3.5	3.5	417	450

[1] Only in enclosures with viewing window

Lock cylinder inserts

Version	B	C
		
	With lock no. 3524 E[1]	With lock no. 3524 E[1]
Material	Die-cast zinc	Die-cast zinc
	Model No.	
	2571.000	**2525.000**

[1] With two keys

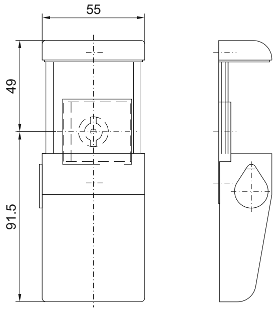
Semi-cylinder lock

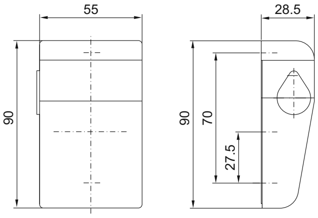
for compact enclosures AE
For retro-fitting to all single-door enclosures with cam lock.
The cover plate is locked by semi-cylinders with a total length of 40/45 mm (to DIN 18 252). An additional cover protects the cylinder against dirt. The protection category of the enclosure is not impaired. Supplied without semi-cylinder.

Material:
- Die-cast zinc

Version	Model No.
RAL 7035	**2534.100**
Nickel-plated (matt)	**2534.500**

Accessories:
- Semi-cylinder, see page 565





Lock cover

for padlocks or multiple locks
For retrospective mounting on all compact enclosures AE, sheet steel, with cam lock.

Material:
- Die-cast zinc

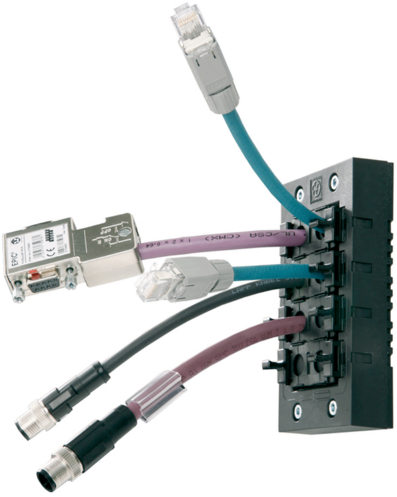
Surface finish:
- Nickel-plated

Packs of	Model No.
1 pc(s).	**2493.000**

Appendix C. Data sheet – Enclosure glands

Cube_Gland_DataSheet.pdf

Variable range of clamping force
Vibration-safe module fixation
Strain relief
Oil resistance
Simplified servicing due to easy assembling and disassembling



CE RoHS ✓

Wide clamping range

Assembly time

Optimum strain relief

Space requirement

Connector with standard housing unit

Info

Innovative multi-cable bushing system with variable clamping ranges for high flexibility in assembling.
When disassembling, the frame can remain on the housing and the plug-in module remains securely on the cable.

Application range

For installation of harnessed cables
Used in areas where cables and wires need to be safely inserted into housings
Apparatus and switch cabinet construction
Electronic installations
Automation technology

Design

The SKINTOP® CUBE system consists of the SKINTOP® CUBE FRAME and the clip modules SKINTOP® CUBE MODULE.
For cut-outs for industrial connectors with standard defined boreholes.
For cut-outs for 16-pin industrial connectors (36 x 86 mm)
For cut-outs for 24-pin industrial connectors (36 x 112 mm)

Note

SKINTOP® CUBE MODULE 20x20 BLIND can be used as a blind module and for clamping ranges 1 - 3 mm

Included

SKINTOP® CUBE FRAME including mounting material

Remark

Photographs are not to scale and do not represent detailed images of the respective products.

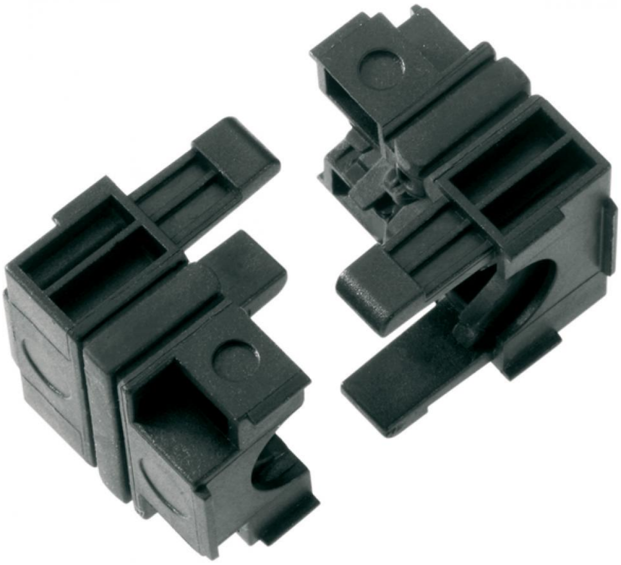
Technical Data

Approvals:	UL pending
Material:	Frame: glass fibre-reinforced polyamide Frame seal: CR Clip module: special polypropylene Clip module seal: LSE 2
Protection rating:	IP 64 NEMA 12
Temperature range:	-20°C to +80°C

Photographs are not to scale and do not represent detailed images of the respective products.

Part number	Article designation / size	Clamping range ØF (mm)	Max. number of executions
SKINTOP® CUBE Frame			
52220000	SKINTOP® CUBE FRAME 16		8
52220001	SKINTOP® CUBE FRAME 24		10
SKINTOP® CUBE clip modules			
52220004	SKINTOP® CUBE MODULE 20x20 BLIND	1 - 3	
52220002	SKINTOP® CUBE MODULE 20x20 SMALL	4 - 6	
52220003	SKINTOP® CUBE MODULE 20x20 LARGE	6 - 9	
52220040	SKINTOP® CUBE MODULE 20x20 AS-I BUS		
52220005	SKINTOP® CUBE MODULE 40x40 SMALL	9 - 12	
52220006	SKINTOP® CUBE MODULE 40x40 LARGE	12 - 16	
52220007	SKINTOP® CUBE MODULE 40x40 BLIND		

Photographs are not to scale and do not represent detailed images of the respective products.



Photographs are not to scale and do not represent detailed images of the respective products.

Appendix D. Data sheet – Isolation switch

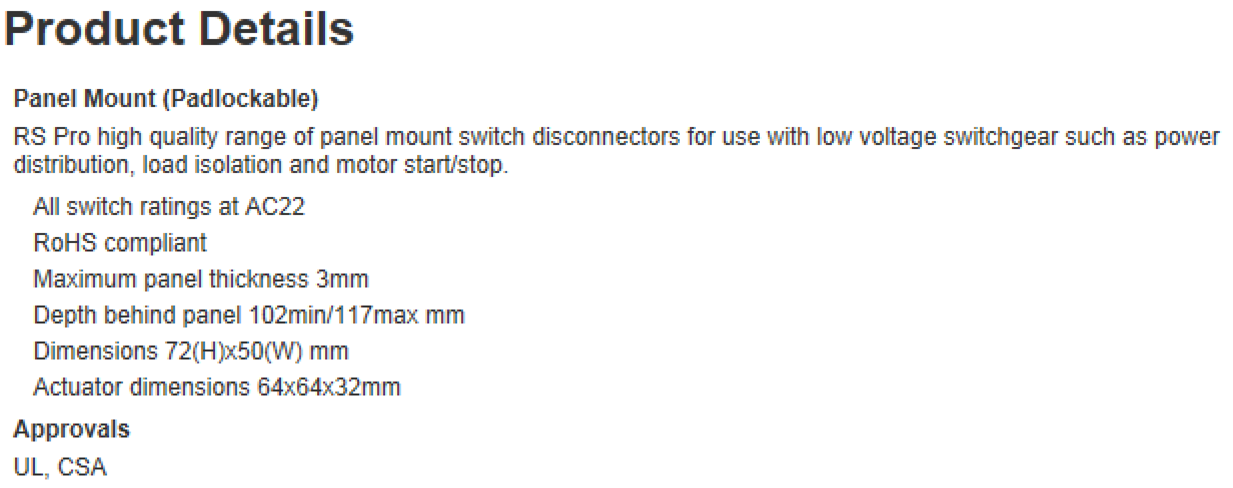
Isolator_Switch_DataSheet.pdf

ENGLISH

Datasheet

Stock No: 466-148

RS Pro 3 Pole Front Panel Mount Non-Fused Switch Disconnector, 25 A, 11 kW, IP65



Product Details

Panel Mount (Padlockable)

RS Pro high quality range of panel mount switch disconnectors for use with low voltage switchgear such as power distribution, load isolation and motor start/stop.

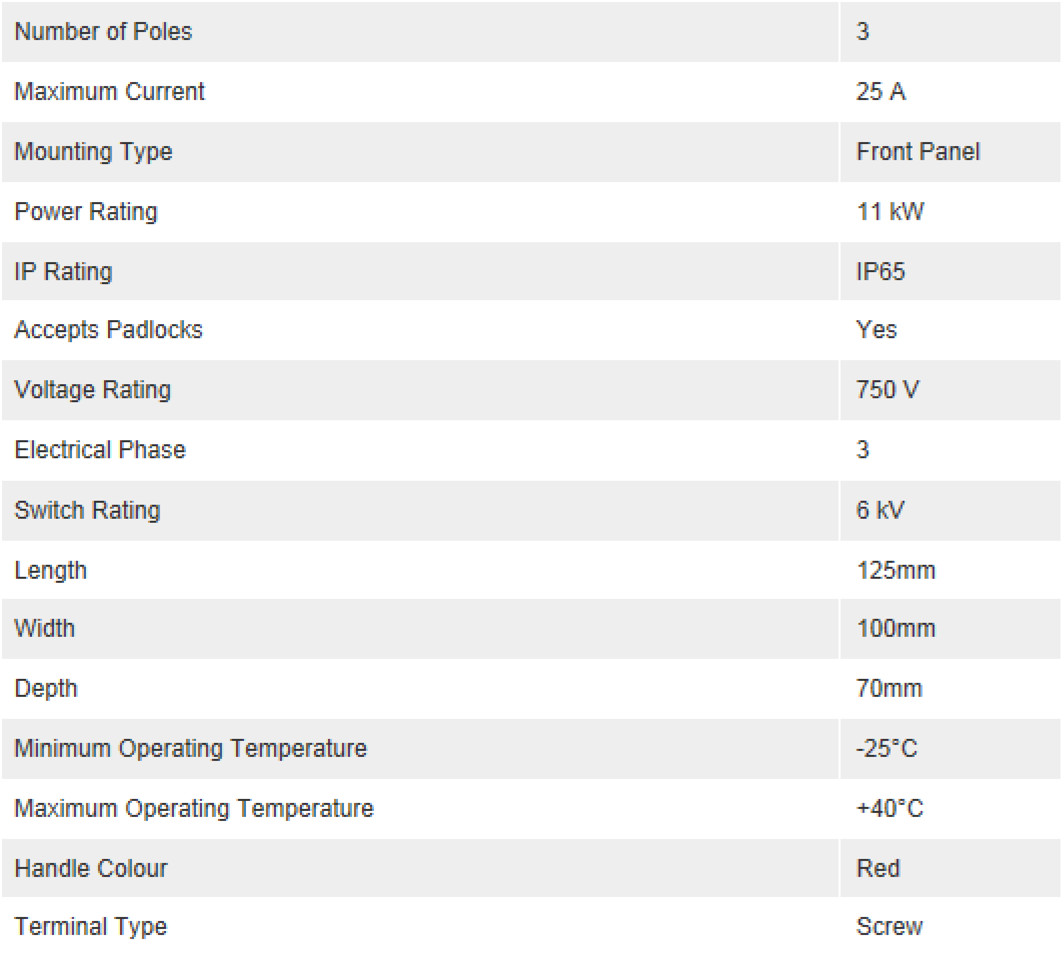
- All switch ratings at AC22
- RoHS compliant
- Maximum panel thickness 3mm
- Depth behind panel 102min/117max mm
- Dimensions 72(H)x50(W) mm
- Actuator dimensions 64x64x32mm

Approvals

UL, CSA

ENGLISH

Specifications:



Number of Poles	3
Maximum Current	25 A
Mounting Type	Front Panel
Power Rating	11 kW
IP Rating	IP65
Accepts Padlocks	Yes
Voltage Rating	750 V
Electrical Phase	3
Switch Rating	6 kV
Length	125mm
Width	100mm
Depth	70mm
Minimum Operating Temperature	-25°C
Maximum Operating Temperature	+40°C
Handle Colour	Red
Terminal Type	Screw

Appendix E. Data sheet – LV-CAP™ PC platform

UNO-2484G_DataSheet.pdf

UNO-2484G

Intel® Core™ i7/i5/i3 Regular-Size Modular Box Platform (MBP) with 4 x GbE, 1 x mPCIe, HDMI, DP

Features

- Intel® Core™ i7/i5/i3 Processor up to 2.6 GHz with 8GB DDR4 built-in Memory
- 4 x GbE, 4 x USB 3.0, 1 x HDMI, 1 x DP (4K), 4 x RS232/422/485
- Stackable 2nd layer for up to 4 iDoor extension or customize for domain applications
- Compact Fanless Design
- Ruggedized by Zero cable and lockable I/O design
- Optional TPM2.0 for Cyber Security
- Chassis Grounding Protection
- Diverse system I/O and Isolated Digital I/O by iDoor Technology
- Supports Fieldbus Protocol by iDoor Technology
- 3G/GPS/GPRS/Wi-Fi Communication by iDoor Technology
- Supports 30+ iDOOR combination with four main categories

Introduction

Advantech's new generation UNO-2000 series of Embedded Automation Computers are Fanless with highly ruggedized with embedded operation system. New UNO-2000 series implement Universal, Customized, and Domain idea into modular design concept which provides flexible and time-to-market support in variety of applications. The series also includes iDOOR technology which supports automation feature-extensions such as Multiple I/O Peripheral, Industrial Fieldbus, Smart I/O Communication, and Signal Communication. New UNO-2000 series including pocket, small, and regular-size form-factors with indicated market segments in terms of different kinds of smart factory application categories such as Equipment Connectivity (EC), Process Visualization (PV), Environment Management (EM), and Dispatch Management (DM) solution. New UNO enables smart factory and fulfill all application.

Specifications

General

Certification	CE, FCC, UL, CCC, BSMI
Dimensions (W x D x H)	200 x 140 x 40 (7.8" x 5.6" x 1.6"), Optimized UNO-2484G 200 x 140 x 70 (7.8" x 5.6" x 2.8"), Universal/Domain/Customized UNO-2484G
Form Factor	Regular Size with stackable design
Enclosure	Aluminum Housing
Mounting	Stand, Wall, VESA (Optional), DIN-rail (Optional)
Weight (Net)	1.2KG (2.65lb)
Power Requirement	10 - 36 V_{DC}
Power Consumption	55W (Typical), 85W (Max)
OS Support	Microsoft® Windows 10, Windows 7, Advantech Linux

System Hardware

BIOS	AMI EFI64 Mbit
Watchdog Timer	Programmable 256 levels timer interval, from 1 to 255 sec
Processor	6th Gen Intel Core i7-6600U, 2.6GHz 4M Smartcache 6th Gen Intel Core i5-6300U, 2.4GHz 3M Smartcache 6th Gen Intel Core i3-6100U, 2.3GHz 3M Smartcache
Memory	Built-in 8GB DDR4 1866/2133 MHz
Graphics Engine	Intel® HD Graphics 520
Ethernet	Intel i210-IT GbE, 802.1Qav, IEEE1588/802.1AS, 802 Intel i219-IT GbE
LED Indicators	LEDs for Power, HDD, LAN (Active, Status), RTC Battery low
Storage	One mSATA (co-lay mPCIe) Two drive bay for SATA 2.5" SSD/HDD CFast by iDOOR Technology (Optional)
Expansion	1 x full-size mPCIe slot on first stack Optional 3 x full-size mPCIe slots on 2nd extension I/O card

I/O Interfaces

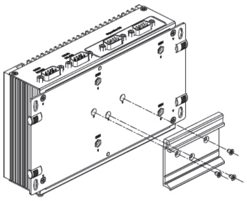
Serial Ports	4 x RS 232/422/485
LAN Ports	4 x RJ45, 10/100/1000 Mbps IEEE 802.3u 1000Base-T Fast Ethernet
USB Ports	4 x USB3.0
Displays	1 x HDMI supports 1920x1080, 1 X DP supports 4K
Audio	Line-out
Power Connector	1 x 2 Pins, Terminal Block
Grounding Protection	Chassis Grounding

Environment

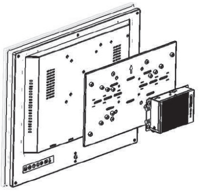
Operating Temperature	-20 ~ 60°C (-4 ~ 140°F) @ 5 ~ 85% RH with 0.7 m/s airflow
Storage Temperature	-40 ~ 85°C (-40 ~ 185°F)
Relative Humidity	10 ~ 95% RH @ 40°C, non-condensing
Shock Protection	Operating, IEC 60068-2-27, 50G, half sine, 11 ms
Vibration Protection	Operating, IEC 60068-2-64, 2 Grms, random, 5 ~ 500 Hz, 1hr/axis (mSATA)
Ingress Protection	IP40

Installation Scenario

DIN-rail Mount Illustration



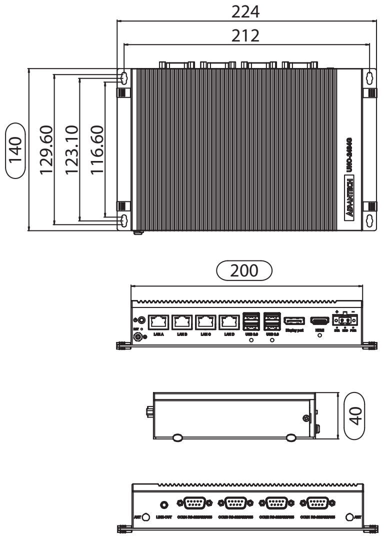
VESA Mount Illustration



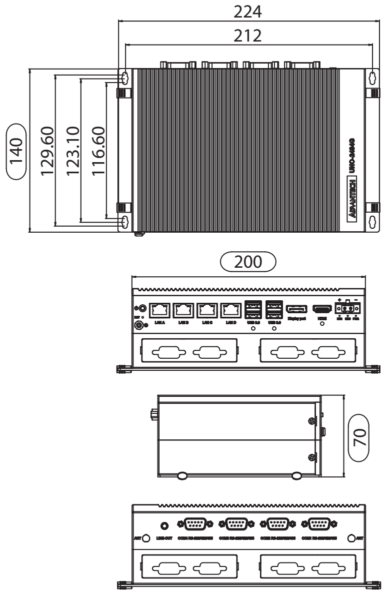
Dimensions

Unit: mm

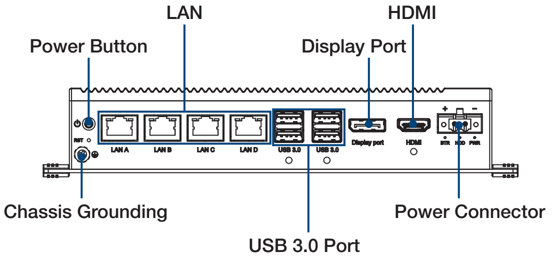
UNO-2484G Optimized UNO



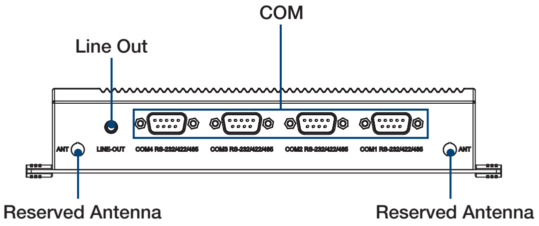
UNO-2484G Universal UNO



Front I/O View



Rear I/O Veiw



Ordering Information

UNO-2484G-6731AE	Intel Core i7-6600U, 2.6GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 1 x mPCIe
UNO-2484G-6531AE	Intel Core i5-6300U, 2.4GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 1 x mPCIe
UNO-2484G-6331AE	Intel Core i3-6100U, 2.3GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 1 x mPCIe
UNO-2484G-6732AE	Universal Intel Core i7-6600U, 2.6GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 4 x mPCIe
UNO-2484G-6532AE	Universal Intel Core i5-6300U, 2.4GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 4 x mPCIe
UNO-2484G-6332AE	Universal Intel Core i3-6100U, 2.3GHz, 8G RAM, 4 x GbE LAN, 4 x USB3.0, 4 x COM, 4 x mPCIe

iDoor Modules

PCM-24S2WF-AE	802.11 a/b/g/n 2T2R w/ Bluetooth4.0, Half-size mPCIe, 2-port SMA
PCM-24S23G-AE	6-band HSPA Cellular Module, GPS, SIM holder, SMAx2
PCM-24R2GL-AE	2-Port Gigabit Ethernet, mPCIe, RJ45
PCM-24D2R2-AE	2-Port Isolated RS-232 mPCIe, DB9
PCM-24D4R4-AE	4-Port Non-Isolated RS-422/485 mPCIe, DB37
PCM-24R1TP-AE	1-Port Gigabit Ethernet, Intel® 82574L, mPCIe, RJ45
PCM-27D24DI-AE	24-Channel Isolated Digital I/O w/ counter mPCIe, DB37

Optional Accessories

1702002600	Power Cable US Plug 1.8 M (Industrial Grade)
1702002605	Power Cable EU Plug 1.8 M (Industrial Grade)
1702031801	Power Cable UK Plug 1.8 M (Industrial Grade)
1700000596	Power Cable China/Australia Plug 1.8 M (Industrial Grade)
UNO-2000G-VMKAE	UNO-2000 VESA Mount Kit
UNO-2000G-DMKAE	UNO-2484G DIN RAIL Kit
TBD	Windows 10 Enterprise image for UNO-2484G
TBD	WES7P image for UNO-2484G MUI

Appendix F. Container Manager Manual – Document reference 2358

The content in this appendix is extracted from the Container Manager Manual, document reference 2358, on Monday August 14th and will not be kept up-to-date with future updates to that document.

iHost

website: https://test.nortechonline.net/ihost/

Login

Request a login from one of the following:

1. Richard Ash
2. James Slater
3. Jake Williams
4. Sion Hughes
5. Ben Cossins

New Image & Config upload

Files needed:

- New Image file to be uploaded (<IID>.tar)
- The New Images config (<IID>.json)
- Container managers config (75e81145-e85f-42ff-b992-d9d12c865c0e.json) get from iHost, see steps below.

Information needed:

- Unix timestamp of when the image was created
- File location (if required by the container) to save files internally.
- The image Repository name and version (Tag)

Step:

1. Login to iHost
2. On the left go to the System/RTU you wish to update.
3. Click on the 'Upload files' button
4. Upload:
    a. Folder – 'containers'
    b. Should not need to tick the Overwrite radio button
    c. Choose the image file
    d. Fill in the description
    e. Click upload
5. Repeat the above for the New images Config as well
6. Before uploading the Container manager config:
    a. Go to 'Files' -> 'containers' folder
        i. You should now see your new <IID>.tar file and <IID>.json files

        ii. Convert the date created for the <IID>.tar file to a Unix timestamp and confirm it's the same as (or newer) what you have. The image timestamp in the Container manager's config should be older than what is in iHost.

    b. Download and Update the Container Managers 'Containers' array with the new containers information:

```
{
  "containerName": "<IID>",
  "File": "<IID>.tar",
  "imageTimestamp": <The timestamp from step 6>,
  "DockerParams": {
    "containerName": "<IID>",
    "imageID": "<IID Repository>:<Version/Tag>",
    "containerVolume": </full/path/to/data/folder>",
    "containerPrivileged": false
  }
}
```

Table 1 - Image JSON object for the Container Managers Config

    c. Extra information can be found in the Public API Document
    d. Save the file and overwrite the original.
    e. Follow step 4 and upload the new Container Manager config file, but click the 'Overwrite' radio button

7. After a few cycles of the Comms Container checking iHost and status/request cycles the container will be downloaded with its config and started.

Update Image upload

Files needed:

- Updated Image file to be uploaded (<IID>.tar)
- The updated Image config (<IID>.json) if needed
- Container managers config (75e81145-e85f-42ff-b992-d9d12c865c0e.json) get from iHost see steps below.

Information needed:

- Unix timestamp of when the image was created
- File location (if required by the container) to save files internally.
- The image Repository name and version (Tag)

Step:

1. Login to iHost
2. On the left go to the System/RTU you wish to update.
3. Click on the ‘Upload files’ button
4. Upload:
    a. Folder – ‘containers’
    b. Will need to tick the Overwrite radio button
    c. Choose the image file
    d. Fill in the description
    e. Click upload
5. Repeat the above for the New images Config as well only if needed
6. Before uploading the Container manager config:
    a. Go to ‘Files’ -> ‘containers’ folder
        i. You should now see your new <IID>.tar file and <IID>.json files
        ii. Convert the date created for the <IID>.tar file to a Unix timestamp and confirm it’s the same as (or newer) what you have. The image timestamp in the Container manager’s config has to be older than what is in iHost.
    b. Download and Update the Container Managers ‘Containers’ array with the updated containers information:
        i. Only change the:
            1. imageTimestamp
            2. imageID
            3. containerVolume (If Changed) See Table 1 above
    c. Extra information can be found in the Public API Document
    d. Save the file and overwrite the original.
    e. Follow step 4 and upload the new Container Manager config file
7. After a few cycles of the Comms Container checking iHost and status/request cycles the container will be downloaded with its config and started.

Update Container Config

Files needed:

- The updated Image config (<IID>.json)

Steps:

1. Login to iHost
2. On the left go to the System/RTU you wish to update.
3. Click on the ‘Upload files’ button
4. Upload:
    a. Folder – ‘containers’
    b. Will need to tick the Overwrite radio button
    c. Choose the config file (<IID>.json)
    d. Fill in the description
    e. Click upload

5. After a few cycles of the Comms Container checking iHost the config will be downloaded and sent out by the Container Manager

Errno Output

MQTT Topic

storage/data/error/75e81145-e85f-42ff-b992-d9d12c865c0e

Errno Table

Err no	Name – inside	Description
10	MQTT_SUBSCRIBE_ERRNO	Used when trying to subscribe to a topic and it fails.
11	FILE_MANAGER_FAILED_MOVE	A failed move of either a config file or a .tar (Container Image) file. From (Generally but this can change) /tmp/LVCAP_config or /tmp/LVCAP_image to /home/CM/LVCAP_config or /home/CM/LVCAP_image In V7 not used.
12	JSON_PARSE_ERROR_FILE	A failed read of the Container Managers config from file.
13	FILE_MANAGER_READ_ERROR	A failed attempt at opening or reading a file on the system.
14	FILE_MANAGER_WRITE_BAD	After a successful open of the file, it fails writing data to the file.
15	FILE_MANAGER_WRITE_NOT_OPEN	A failed attempt at opening a file to write to.
16	FILE_MANAGER_COPY_BAD	After a successful open of the file the write of the copied data is bad.
17	FILE_MANAGER_COPY_FILE_NOT_EQUAL	The check that each file is the same size fails

Err no	Name – inside	Description
18	FILE_MANAGER_COPY_NOT_OPEN	Could not open the destination file to copy to.
19	FILE_MANAGER_IMAGE_MOVE_ERROR	After trying to move an Image file (.tar) the file does not exist at the intended destination.
20	MQTT_MANAGER_STATUS_MSG_ERROR	Error parsing the JSON message from any status/response/<GUID> topic
21	FILE_MANAGER_MKDIR	Can be multiple different causes. • ENAMETOOLONG • ENOSPC • ENOTDIR • EACCES For more information see: http://pubs.opengroup.org/onlinepubs/009695399/functions/mkdir.html
22	MQTT_MANAGER_STATUS_KEY_ERROR	The status JSON Payload is correct but is missing the Key “Status” or the value it holds in not an integer.
23	MQTT_MANAGER_STATUS_DEFAULT_ERROR	The status Integer does not match any that is present in the External API Document.
26	MQTT_MANAGER_CONFIG_MSG_PARSE_ERROR	Not used?
27	MQTT_MANAGER_TP_CONF_FILE_ERROR	If a Container config (Not the Container Managers) is empty or invalid JSON.
29	MQTT_MANAGER_TP_CONF_KEY_ERROR	No Key "ContainerConfig" found in the config, or the value does not contain a JSON Object
30	MQTT_MANAGER_INCOMING_STATUS_FAIL	Status Fail was sent by a running container.
31	MQTT_MANAGER_INCOMING_STATUS_ERROR	Status Error was sent by a running container.
32	MQTT_MANAGER_INCOMING_STATUS_SHUT_DWN	Status Shut Down was sent by a running container

Err no	Name – inside	Description
33	MQTT_MANAGER_CONFIG_DWN_LOAD_MSG_ERROR	The notification payload for a new config downloaded is empty or invalid JSON.
35	MQTT_MANAGER_CONFIG_DWNLD_EMPTY_FILE	The downloaded config is empty or invalid JSON
36	MQTT_MANAGER_CONFIG_DWNLD_NOT_USED	The file was found and valid JSON but the contents does not match what is needed for Container or the Manager. This is then logged as it is not used.
37	MQTT_MANAGER_CONFIG_DWNLD_CONTENTS	Config downloaded notification message not empty & valid JSON but does not contain correct Key/Value as a downloaded file.
38	DOCKER_INTERFACE_PROCESS_CONTAINERS	Processing the containers inside the Container Managers config. This is if the config is not valid JSON or empty.
39	DOCKER_INTERFACE_START_CONTAINERS	A failed start of a container. This is trying to start a pre-loaded container.
40	DOCKER_INTERFACE_STOP_CONTAINERS	A failed stop of a running container.
41	DOCKER_INTERFACE_RESTART_CONTAINER	A failed restart of a container.
42	DOCKER_INTERFACE_LOAD_CONTAINER	A failed Load of an docker image.
43	DOCKER_INTERFACE_PARSE_CMD	Running a command failed to parse the response.
44	DOCKER_INTERFACE_FAILED_STAUTS_REPLY	Container has failed to respond to status/request and is to be shutdown.
50	SYSTEM_COMMAND_POPEN_ERROR	Error doing popen(), failed to fork / pipe / malloc etc.
51	SYSTEM_COMMAND_PCLOSE_ERROR	Error back from pclose(), the process died on a signal etc.

Appendix G. Load profile test

The aim of the Load Profile configuration is to produce a predicted half hourly load current forecast for the next 24 hours. The method adopted for each half hour is to calculate the RMS load during the same half hour on the same day of the week in the previous four weeks. Thus the forecast for 00:00 to 00:30 on day twenty-nine is calculated as the RMS of the load currents between 00:00 and 00:30 on days one, eight, fifteen, twenty-two. The calculations are carried out per-phase for each of the three phases.

Inputs

The Application will take its data input by subscribing to MQTT topics

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l2/current-mean

sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l3/current-mean

Payloads in Scalar Object Format (as defined in API section 9.1) will be published on these topics every 60 seconds, containing the measured RMS load current in each phase in Amps (API section 4.5). This is the measurement data from the Lucy GridKey Sensor Application

For testing data will be sent on the same topics but in faster than real time. The following data will be sent, the same for each day on each phase:

- L1: Load Curve G, peak current 100A
- L2: Load Curve 33A, peak current 200A
- L3: Constant current of 300A

This data is contained in the 28 day input data file input-28day.csv and the 1 day input data file which follows it input-1day-w5d1.csv.

Procedure for creating the test data set (for reference only, use above files for testing).

10. Open ODF data generator spreadsheet in LibreOffice Calc
11. Adjust settings to set desired curves, peak currents and start date/time (in UTC)
12. Make sure export sheet is selected sheet.
13. File > Save a Copy …
14. Change Save as type to Text CSV
15. Enter file name and click Save
16. In export dialogue:
    a. Character set: Western Europe
    b. Field delimiter: ,
    c. Text delimiter: "
    d. Tick Save cell content as shown and untick the others
    e. Click OK
17. When it warns only the selected sheet was saved, click OK.

18. This produces a 7 day input file, repeat for more days and edit files together, or chop out the required section for shorter files.

To feed the 28 day input into the system at 3600 times real time (one hour per second) use the command

```
./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt
--key eatl_tlsdevtools.key --no-store -i input-28day.csv -f 3600
```

To feed the 1 day input into the system at 60 times real time (one minute per second) use the command

```
./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt
--key eatl_tlsdevtools.key --no-store -i input-1day-w5d1.csv -f 3600
```

Configuration

Reference: 2662-MANUL-S001-Vxx.yy.zz

For the Load Profiler the following global settings are used:

Debug: true (turn debug output on)

RefreshRate: 2 (This determines how often outputs can be produced. This setting is in real time regardless of the timestamps in input data).

MaxDataWait: 300 (Data will come in much more often than this in testing).

One profile is configured:

ProfileName: PredictedTxLoad (name for output data)

ProfileCircuit: Transformer(what is this for?)

ProfileTimeSlotAlgorithm: SquaredAverage (we want to use the RMS / geometric mean of the load current values to give the correct heating effects in rating calculations).

ProfileLateDataLimitInSeconds: 120 (we will be feeding in 60 second interval data)

ProfileDailyTimeSlots: 48 (to give 30 minute interval slots)

ProfileWindowLengthDays: 28

Inputs:

value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean

value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l2/current-mean

value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l3/current-mean (three inputs for three phases, from the sensors defined above)

Output Settings:

ProfileOutputTopicString: "tx-load/predict/byday/l1/30/30/1440" (name of the sub-topic the data will be published on)

ProfileOutputIntervalMinutes: 30 (output a new profile every 30 minutes)

ProfileOutputPredictionLengthMinutes: 1440 (24 hour prediction)

ProfileOutputSmoothing: RMS (we want to calculate the RMS / geometric mean of the days)

ProfileOutputValues: value: sensor/data/96d6f19b-7022-45f2-b753-cb5012626b4d/gridkey-mcu520/60/feeder/1/l1/current-mean (only one input, from the sensor defined above)

The same arrangement is repeated for l2 and l3.

LV-CAP is configured to run the Load Profiler and CSV logger applications only. The CSV logger is configured to log the sensor topics to a single CSV file

Outputs

The output profiles must be published on the topics:

- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l1/30/30/1440
- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l2/30/30/1440
- algorithm/data/5414c8fd-4924-4d08-a56a-c7e553b40e3e/tx-load/predict/byday/l3/30/30/1440

The Application must publish profiles in the standard LV-CAP time series JSON format to the above MQTT topic. Each profile produced must start at approximately the current time (based upon the time stamps of the incoming data, not the host system clock). Each profile produced will have 48 elements, each of 30 minutes duration.

To convert timestamps into human-readable UTC time use a command line

```
date -u -d '@1503966600' "+%Y/%m/%d %H:%M:%S %Z"
```

Test Steps

9. Start from a clean LV-CAP core system with no data files or load profile data present.
10. Deploy the TLS Development tools, which include the play_csv.py script. Install any necessary python packages to run the script against the LV-CAP core.
11. Deploy and configure the Load Profiler and Data Storage applications onto the system.
12. Use make sub-sensor to monitor sensor data on the system
13. Play the 28-day history file into the system to build up the profile data (this takes ~12 minutes):
    ./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key --no-store -i input-28day.csv -f 3600
14. When the 28-day data set completes, also monitor the algorithm output with make sub-alg
15. Continue to play back the 1 day file at slower speed to see the profiles being produced:
    ./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert

eatl_tlsdevtools.crt --key eatl_tlsdevtools.key --no-store -i input-1day-w5d1.csv -f 3600

16. Examine the profiles produced and check as documented below.

Expected Outputs

The 1 day data file we are using for testing contains data for Tue 29 Aug 2017, so we are looking for the outputs from this day to be produced.

The first three outputs will be produced almost immediately, for l1, l2 and l3 respectively.

- Each will have the same TimeStamp values of 1503966600, which is 2017/08/29 00:30:00 UTC, the first profile of the day.
- TimeStampStart will have a value of 1501547400, which is 2017/08/01 00:30:00 UTC, the date and time 28 days earlier when the oldest data used in the profile was collected.
- On the L1 output, the values are for load curve G at 100A peak, which starts with an hour of 100A, then 96.8A for the next hour and then 93A, ending up with a hour of 99.2A
- On the L2 output, the values are for load curve 33A at 200A peak, which starts with 187A for one hour (two profile steps), then 177.4A and 168.2A, ending up at 200A
- On the L3 output, the values are all 300A because the load does not change!

Now move on to the second set of profiles produced.

- Each will have a TimeStamp value of 1503968400, which is 2017/08/29 01:00:00 UTC, half an hour after the previous profiles.
- TimeStampStart has also moved on to 1501549200 (2017/08/01 01:00:00 UTC).
- On L1 output, things have moved up by half a hour, so we get only one step at 100A, two each at 96.8A and 93A. We end with two at 99.2A and then the 100A value wrapped from the start.
- The same on L2, one step at 187A, two each at 177.4A and 168.2A, ending with two at 200A and one at 187A
- On the L3 output, the values are all 300A because the load does not change!

We can keep doing this for as long as you like, but you get the picture!

Appendix H. Data Reading Test

Configuration

Configure the Modbus RTU Sensor Application to read all the identified registers to the topics as given in 2404-RQSPEC-S002, except that any parameter configured to read less often than every 60 seconds should be re-configured to read every 60 seconds. This configuration file ….

Configure the CSV Data Recorder Application to record the values of all parameters read from each of the three circuit breakers every 10 seconds to one file prefixed "ALVIN_load", and all other parameters (read from each of the three circuit breakers every 60 seconds) to a second file prefixed "ALVIN_status". The configuration file ….

Switch the ALVIN control switch on the front of the OpenLV Enclosure to the ON position.

Ensure that the Circuit Breakers are in the closed position.

Set up a voltmeter within the test enclosure to measure the L1 phase to neutral voltage, and a CT and ammeter to measure the L1 load current.

Test Steps

1. Apply power to the CBs and pass load current through them.
2. Switch the ALVIN control switch on the front of the OpenLV Enclosure to the ON position.
3. Run the command `make sub-alvin-load` in the LV-CAP TLS development tools directory.
4. Verify that, every 10 seconds, output are produced, on the following topics:
    a. sensor/data/eatl_sensorcontainer_00/feeder1/l1/busbar_voltage
    b. sensor/data/eatl_sensorcontainer_00/feeder1/l1/cable_voltage
    c. sensor/data/eatl_sensorcontainer_00/feeder1/l1/load_current
    d. sensor/data/eatl_sensorcontainer_00/feeder1/l1/reactive_power
    e. sensor/data/eatl_sensorcontainer_00/feeder1/l1/active_power
5. Examine the contents of the ALVIN_load CSV file and confirm that it contains a line every 10 seconds and a total of 31 columns (timestamp, 5 data points and 5 valid flags for each phase).
6. Confirm that the busbar_voltage and cable_voltage topics show the same voltage as the voltmeter.
7. Confirm that the load_current topic shows the same current as the ammeter.

8. Run the command `make sub-alvin-status` in the LV-CAP TLS development tools directory.

9. Verify that every 60 seconds, output are produced, on the following topics:
    a. sensor/data/eatl_sensorcontainer_00/feeder1/l1/open_operations
    b. sensor/data/eatl_sensorcontainer_00/feeder1/l1/close_operations
    c. sensor/data/eatl_sensorcontainer_00/feeder1/l1/watchdog_count
    d. sensor/data/eatl_sensorcontainer_00/feeder1/l1/cpu_temperature
    e. sensor/data/eatl_sensorcontainer_00/feeder1/l1/uptime
    f. sensor/data/eatl_sensorcontainer_00/feeder1/l1/switch_temperature
    g. sensor/data/eatl_sensorcontainer_00/feeder1/l1/fault_flags
10. Examine the contents of the ALVIN_status CSV file and confirm that it contains a line every 60 seconds and a total of 43 columns (timestamp, 7 data points and 7 valid flags).
11. Note the values of open_operations and close_operations and that they do not change.
12. Confirm that the cpu_temperature and switch_temperature are close to but above ambient temperature.
13. Confirm that the uptime value in seconds increments in real time.

14. Switch the ALVIN control switch on the front of the OpenLV Enclosure to the OFF position.
15. Run the command `make sub-alvin-load` in the LV-CAP TLS development tools directory.
16. Verify that, every 10 seconds, 15 lines of output are produced, 5 for each phase.
17. On L1 phase, verify that outputs are produced but with the Valid flag set to False on the following topics:
    a. sensor/data/eatl_sensorcontainer_00/feeder1/l1/busbar_voltage
    b. sensor/data/eatl_sensorcontainer_00/feeder1/l1/cable_voltage
    c. sensor/data/eatl_sensorcontainer_00/feeder1/l1/load_current
    d. sensor/data/eatl_sensorcontainer_00/feeder1/l1/reactive_power
    e. sensor/data/eatl_sensorcontainer_00/feeder1/l1/active_power
18. Run the command `make sub-alvin-status` in the LV-CAP TLS development tools directory.
19. Verify that every 60 seconds, 21 lines of output are produced, 7 for each phase.
20. On L1 phase, verify that outputs are produced but with the Valid flag set to False on the following topics:
    a. sensor/data/eatl_sensorcontainer_00/feeder1/l1/open_operations
    b. sensor/data/eatl_sensorcontainer_00/feeder1/l1/close_operations
    c. sensor/data/eatl_sensorcontainer_00/feeder1/l1/watchdog_count
    d. sensor/data/eatl_sensorcontainer_00/feeder1/l1/cpu_temperature
    e. sensor/data/eatl_sensorcontainer_00/feeder1/l1/uptime
    f. sensor/data/eatl_sensorcontainer_00/feeder1/l1/switch_temperature
    g. sensor/data/eatl_sensorcontainer_00/feeder1/l1/fault_flags

Appendix I. ALVIN Control Test

Configuration

Configure the Modbus RTU Sensor Application to control the ALVIN Reclose™ Circuit Breaker from the topic, ‘sensor/data/1’.

Also configure it to read the value of the MHR_SHADOW_SWITCH_STATUS register in each CB every minute and log these three values to topics

- sensor/data/eatl_sensorcontainer_00/feeder1/l1/switch_state
- sensor/data/eatl_sensorcontainer_00/feeder1/l2/switch_state
- sensor/data/eatl_sensorcontainer_00/feeder1/l3/switch_state

Configure the CSV Data Recorder Application to record the values of the above algorithm topics to one file prefixed "ALVIN_command", and the sensor topics (read from each of the three circuit breakers every 60 seconds) to a second file prefixed ALVIN_result. The configuration file ….

Switch the ALVIN control switch on the front of the OpenLV Enclosure to the ON position.

Ensure that the Circuit Breaker is in the closed position.

Test Steps

1. Apply power to the CBs and pass load current through them.
2. Switch the ALVIN control switch on the front of the OpenLV Enclosure to the ON position.
3. Run the command `make pub-mesh-open-l1` in the LV-CAP TLS development tools directory.
4. Observe that the L1 CB (only) opens. Note the time of this.
5. Wait at least one minute.
6. Run the command `make pub-mesh-close-l1` in the LV-CAP TLS development tools directory.
7. Observer that the L1 CB (only) closes. Note the time of this.
8. Switch the ALVIN control switch on the front of the OpenLV Enclosure to the OFF position.
9. Run the command `make pub-mesh-open-l1` in the LV-CAP TLS development tools directory.
10. Observer that none of the CBs opens, there is no change. Note the time of this.
11. Examine the contents of the ALVIN_command CSV file. Observe that both successful and unsuccessful commands are recorded at the times noted above.
12. Examine the contents of the ALVIN_result CSV file. Observe that only successful commands cause the ALVIN state to change at the times noted above.

Examine the contents of the ALVIN_status CSV file. Confirm that the values of open_operations and close_operations have increased from the ones noted before.

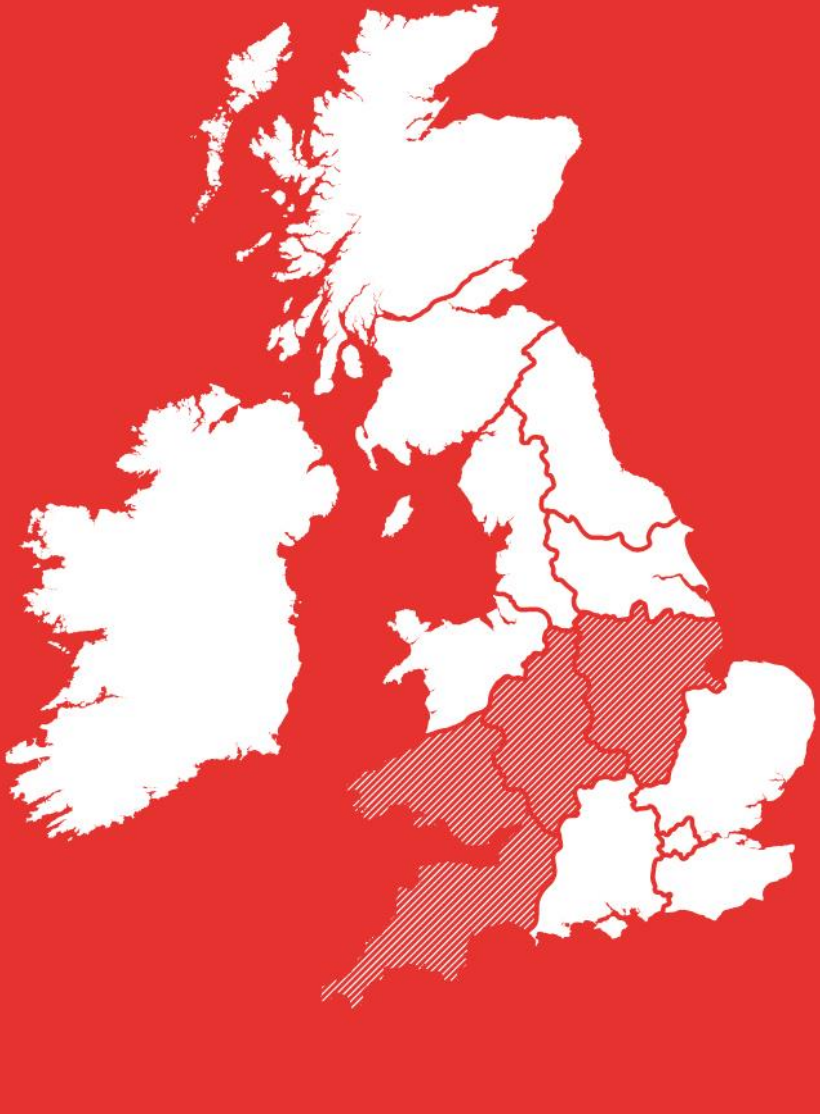
Annex 4. Site Acceptance Test (SAT) Documentation

WESTERN POWER DISTRIBUTION

OPEN LV

OPENING UP THE SMART GRID

Site Acceptance Tests



Report Title:	Site Acceptance Tests
Report Status:	Issued
Project Ref:	WPD/EN/NIC/02 – OpenLV
Date:	23.10.2017

Document Control

	Name	Date
Prepared by:	Tim Butler	10.10.2017
Reviewed by:	Richard Potter	11.10.2017
Recommended by:	Dan Hollingworth	12.10.2017
Approved (WPD):	Mark Dale	16.10.17

Revision History

Date	Issue	Status
23.10.2017	1.1	Issued
16.10.2017	1.0	Issued
27.09.2017	0.1	Draft

DISCLAIMER

Neither WPD, nor any person acting on its behalf, makes any warranty, express or implied, with respect to the use of any information, method or process disclosed in this document or that such use may not infringe the rights of any third party or assumes any liabilities with respect to the use of, or for damage resulting in any way from the use of, any information, apparatus, method or process disclosed in the document.

Telephone +44 (0) 1332 827446. E-mail wpdinnovation@westernpower.co.uk

Contents

1 Introduction

1.1 Overall testing approach

The overall approach to testing the OpenLV solution has been defined in 'SDRC 1 – Specification Design & Testing'.

There are three distinct areas of testing for the OpenLV solution.

- **Factory Acceptance Tests** to verify the system and its components function correctly, including the operation and configuration of software components, and consequently that the overall system meets the requirements detailed in the Requirements Specification.
- **Site Acceptance Tests** to verify the solution meets the requirements in realistic, non-laboratory / controlled environment once the complete system has been installed on location in its final configuration. These tests verify that no damage occurred to the hardware during shipment and installation.
- **Cyber-security testing** to evaluate the cyber-security capabilities of the LV-CAP™ platform; these tests will be undertaken by a specialist provider.

The Factory Acceptance Tests (FATs) were undertaken in two stages and completed successfully, demonstrating the LV-CAP™ platform and ancillary equipment has been functionally tested and approved for deployment in the OpenLV Project trials.

As with the FATs, there will be two phases of Site Acceptance Tests (SATs), the first for the core OpenLV solution and the second to comprise the full solution, including the network automation elements; refer to Table 1 below for details of the components covered in each set of tests.

The current version of this document only details the tests for the Phase 1 SATs, and will be updated prior to the deployment of the network automation hardware.

SATs will only be undertaken on the first iterations of equipment installed for the project. For example, SATs for the 'Core System' will be undertaken for the first two pairs of systems deployed. The full solution SATs will only be undertaken on the first pair of sites upgraded to implement autonomous network control.

A set of detailed commissioning tests will be utilised on all site installations, including the sites where the SATs will also be undertaken.

Table 1 - Summary of Site Acceptance Testing

Component	Category	SAT 1 OpenLV Core System	SAT 2 OpenLV Full Solution
LV Network Automation Hardware	Hardware	No	Yes
LV Monitoring Hardware	Hardware	Yes	Yes
OpenLV Platform	Hardware	Yes	Yes
Application Deployment & Management Server	Hardware	Yes	Yes
Cloud Hosted Server	Hardware	Yes	Yes
Communications Infrastructure	Hardware	Yes	Yes
LV-CAP Operating System	Software	Yes	Yes
Temperature Sensor app	Software	Yes	Yes
Load Profile Predictor app	Software	Yes	Yes
Peer to Peer Communications app	Software	Yes	Yes
LoadSense app	Software	No	Yes
Network Meshing app	Software	No	Yes
Dynamic Thermal Ratings app	Software	No	Yes
Nortech Communications app	Software	Yes	Yes
Electrical Sensor app	Software	Yes	Yes
Lucy Electric Gridkey Communications app	Software	Yes	Yes

Cyber-security testing is a multi-phased process, and will not be completed until after the deployment of the trial hardware and a period of operation to evaluate performance. At the time of writing, prior to initial hardware deployment, a cyber-security evaluation has been undertaken to confirm the suitability of the LV-CAP™ platform for deployment as part of the trials.

NCC Group, the cyber-security specialised contracted to the Project have confirmed cyber-security tests and overall evaluation will be undertaken in parallel with the equipment deployment and will inform updates to the platform where necessary.

It is not intended to undertake tests relating to the cyber-security requirements as part of the hardware and functionality tests. Due to the nature of cyber-security testing, particularly penetration testing, the duration required for effective evaluation, and the potential conflict of simultaneous tests being undertaken, these requirements will be appraised separately by NCC Group, the OpenLV Project's cyber-security specialist.

2 Site Acceptance Tests Methodology

At the point of equipment deployment, elements of the overall OpenLV solution, (e.g. the LV-CAP™ platform code, the management and control server, and software containers), are unchanged from the testing undertaken within EA Technology's and project partner facilities to the point of deployment to site. Furthermore, with the majority of system components being software based, these are unlikely to be affected by a physical change in the hardware's location.

Therefore, the SATs detailed in this document focus on ensuring the system continues to operate as expected despite no longer being based in a controlled environment and following the disconnection, relocation and reconnection of the system's primary hardware and ancillary devices. The SATs therefore comprise a number of tests to verify the system's overall functionality is unaffected following transport and installation.

Due to the interconnected nature of the system and components, rather than replicating the full range of individual tests undertaken as part of the FATs, the SATs can be undertaken through a reduced number of tests by utilising targeted testing. This ensures the full process of operation, from the point of data gathering from the LV network to uploading the processed information, generated from the data, to the connected servers and therefore, that each individual component, comprising predominantly of software containers, is operating correctly.

The tests outlined below have been scheduled to minimise repeated tasks and wherever possible, to enable a single action, or sequence of actions to demonstrate that multiple requirements are met, and consequently multiple modules and software components are functioning as expected.

2.1 Setup Details

2.1.1 Login details

To undertake the tests detailed in this document, two forms of access to the Intelligent Substation Device (ISD) are required.

1. Local access via a laptop; and
2. Remote access via the modem within the ISD.

The local connection will be achieved through a laptop and a direct ethernet connection, that requires the use of a unique secure encryption key.

Remote access to the modem within the ISD, via a Virtual Private Network (VPN) connection, for staff based at EA Technology's offices will also be required.

To undertake the tests, the following additional computers, beyond the LV-CAP™ platform, will be required:

- 1x laptop on-site to provide direct access to and control the router modems within the enclosure.
- 1x computer in EA Technology offices to provide access to the iHost control server.

The tests will verify correct installation of monitoring sensors and consequently, the below devices will also be required:

- Handheld ammeter, appropriately rated to allow measurement of the LV network assets (busbar feed and individual feeder phases).
- Thermometer capable of undertaking ambient temperature readings.

2.2 Pre-deployment work

Prior to the deployment of equipment to the network for the trials, each set of hardware will be set-up and tested by EA Technology to confirm it is operating as expected. These tests will cover:

- Correct functionality of the LV-CAP™ platform;
- Correct functionality of the GridKey MCU520 platform;
- Connectivity between the MCU 520 and the LV-CAP™ platform;
- Connectivity between the LV-CAP™ platform and router modem;
- Successful bi-directional communications through the router modem;
- LV-CAP™ access to the iHost Command and Control Server; and
- LV-CAP™ access to the Lucy Electric Cloud Data Server.

All sets of LV-CAP™ equipment will be tested and verified as being fully operational before shipment to site, with the SATs and commissioning tests designed to verify the shipping and installation process have not inadvertently introduced issues.

Between testing at EA Technology and installation on-site, the following elements will be changed:

- **Location**: Whilst the LV-CAP™ platforms will communicate via the 3G / 4G network through the internal router modem, the units will be reactivated onsite for connection to different cell towers. This requires verification that the system can establish communication links to the rest of the Project systems.
- **Thermocouple sensors**: The thermocouples will be disconnected to ensure safe transit and will be installed and reconnected on-site. It will be necessary to verify that the thermocouples are still operating correctly, and have been connected to the correct inputs.
- **GridKey sensors**: the GridKey sensors will be disconnected for safe transit before being installed onsite and reconnected back to the LV-CAP™ platform. It will be necessary to verify that the sensors have been installed and reconnected correctly.
- **iHost Command and Control Server**: It is necessary to verify that the LV-CAP™ platform can connect to the iHost server and receive data from the server.
- **Lucy Electric Cloud Data Centre**: It is necessary to confirm that the LV-CAP™ platform can upload data to the cloud based server.

On completion of the pre-deployment testing process, the sensor container in the LV-CAP™ platforms will be configured to output data at ten-second intervals to expedite testing on-site once the equipment is energised and online. Once on-site testing, (SATs and commissioning as required) is complete, the rate of sensor reading will be reverted to one-minute intervals.

Some sites will require two rounds of SATs, with the second necessary at locations where ALVIN Reclose™ devices are to be installed. In these instances, when the ALVIN Reclose™ devices are deployed, they will have been fully tested prior to issuance from EA Technology, and will be tested with the LV-CAP™ platform in-situ to verify successful communications and relay operation.

2.3 Phase 1 Site Acceptance Tests – Initial deployment

For the avoidance of doubt, some sites in which the ISD enclosures and monitoring equipment are installed will be upgraded later in the project with the installation of ALVIN Reclose™ devices and separate SATs will be undertaken to verify the successful installation and operation of those components and associated software. This document will be updated to detail the Phase 2 tests in advance of the deployment of the ALVIN Reclose™ devices to implement the network automation functionality.

All equipment will be installed in accordance with the agreed method statements prior to commencing these tests.

Prior to commencing any SATs, the enclosure shall be installed in the location agreed during the site surveys, as will the MCU520 platform, and all ancillary equipment (Rogowski coils and thermocouples). All cables shall be routed using cable ducts or equivalent, as available, and appropriate within the substation in question.

The enclosure shall not be energised and made live until all work elements are complete. Once energised, the ISD will be fully connected and energised, being left for a period of at least several minutes to enable it to initialise all software applications and communication links in preparation for the tests to commence.

2.3.1 On-site testing approach

The on-site testing approach will demonstrate the following high-level capabilities of the system once installed.

A system overview, highlighting the key hardware elements and the interconnecting communication links is shown below.

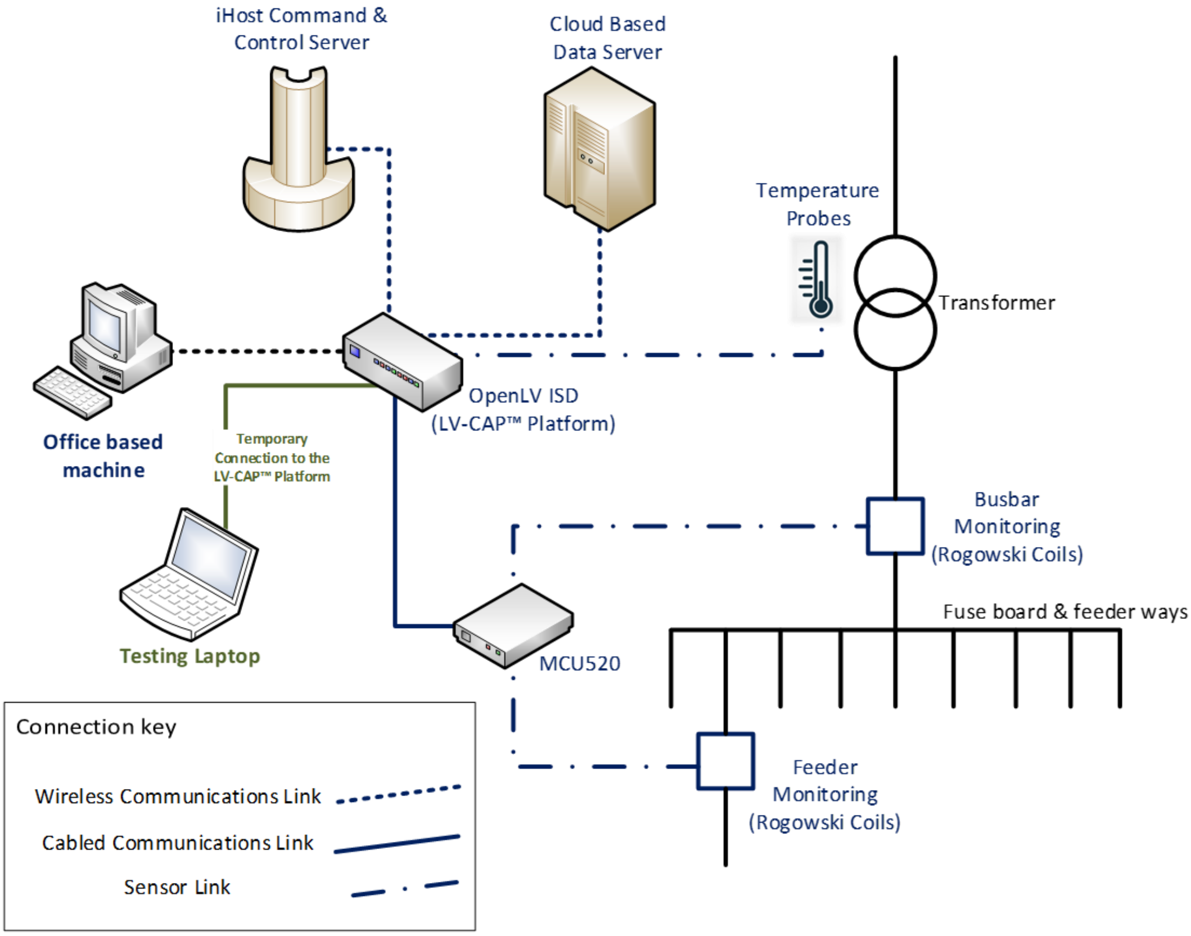

iHost Command & Control Server
Cloud Based Data Server
Temperature Probes
Transformer
Office based machine
Temporary Connection to the LV-CAP™ Platform
OpenLV ISD (LV-CAP™ Platform)
Busbar Monitoring (Rogowski Coils)
Fuse board & feeder ways
Testing Laptop
MCU520
Feeder Monitoring (Rogowski Coils)
Connection key
Wireless Communications Link
Cabled Communications Link
Sensor Link

Figure 1 – OpenLV Trial System Overview

Each sequence of tests detailed in Section 3 are designed to test a specific part of the system shown above: either individual components, or interconnections between them, and outline:

- The objectives of the overall test sequence;
- The initial required conditions; and
- A list of numbered actions with their corresponding expected results.

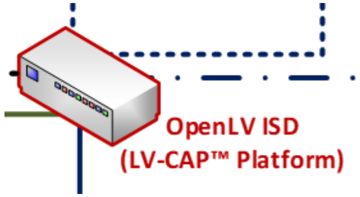
If successful, these sequence of tests, demonstrate the hardware, communication links and associated software containers previously verified as operational in the FATs and pre-deployment tests, continue to operate as expected in the new location of the equipment.

Consequently, successful completion of the FATs confirms that:

1. The system has been successfully energised with the LV-CAP™ platform running as expected;
2. The GridKey MCU520 has successfully energised and is communicating with the LV-CAP™ platform;
3. The sensors have been installed correctly and are successfully communicating with the MCU520;
4. The thermocouples have been installed correctly and are providing accurate data to the LV-CAP™ platform;
5. The router / modem is operational and communicating successfully to both the LV-CAP™ platform and the mobile data network;
6. That remote access can be achieved directly to the router / modem;
7. That remote access can be achieved to the LV-CAP™ platform through the router / modem;
8. That the LV-CAP™ platform is successfully transmitting data to the iHost and the Cloud Data Servers; and
9. That the LV-CAP™ platform can be updated remotely using the router / modem mobile data connection.

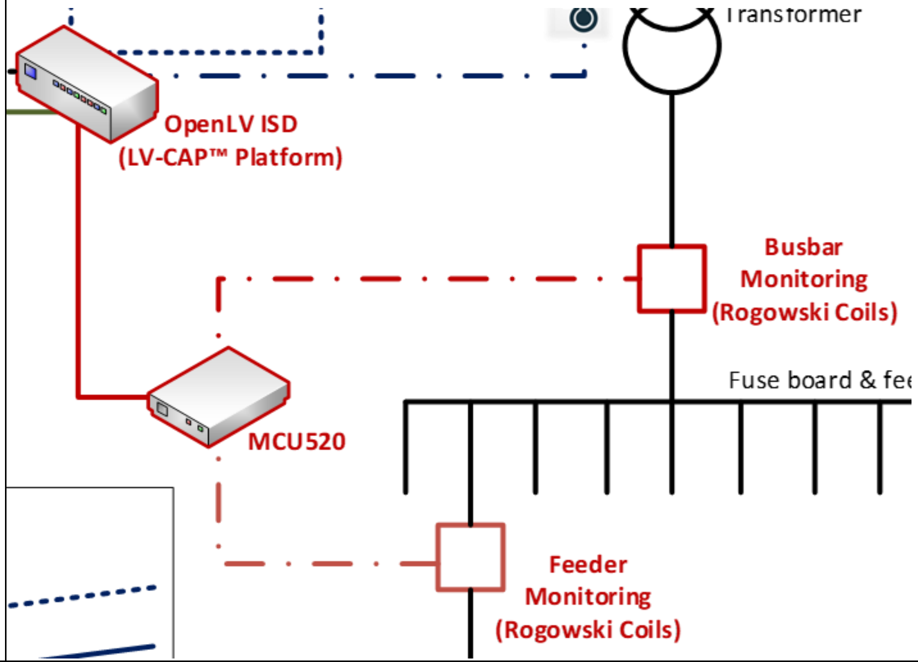
3 Site Acceptance Tests

Test: SAT 1.01

Objective:		To confirm the industrial PC is operational, has activated successfully and that the LV-CAP™ platform is operational.		
Elements under test:		 This test sequence is concerned with verifying the Ruggedised PC within the Intelligent Substation Device remains operational. In order to be successful, the following elements must be functioning correctly: • Nortech Communications Container • Load Profile Predictor • CSV Data Recorder • Lucy Electric Sensor Container • Lucy Electric Communications Container • Temperature Sensing Application		
Starting condition:		The ISD enclosure shall be installed in line with the method statement and specific on-site requirements. It will have been energised and left for a period of several minutes to enable the system to start-up without interference.		
Test sequence:		***Action***	***Expected result***	***Pass / Fail***
	1	Open the enclosure and connect the testing laptop to an ethernet port on the industrial PC.		
	2	Log into the LV-CAP™ platform.	A secure connection should be established. If the system is not online, disconnect the testing laptop, restart the ISD enclosure then restart this test sequence.	Pass ☐ Fail ☐

	3	Access the management console for the LV-CAP™ platform and verify that expected software containers are operational.	The following software containers should be active: • Nortech Communications Container • Load Profile Predictor • CSV Data Recorder • Lucy Electric Sensor Container • Lucy Electric Communications Container • Temperature Sensing Application	Pass ☐ Fail ☐
Comments:				

Test: SAT 1.02

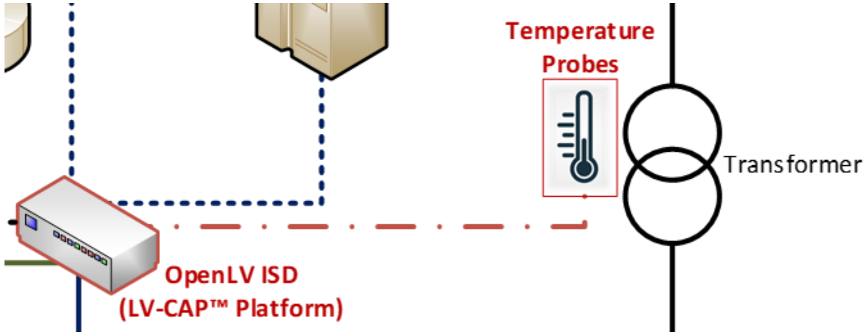
Objective:	To verify the MCU520 is communicating successfully with the LV-CAP™ platform and that the sensors providing data to the MCU520 have been installed correctly.
Elements under test:	 Transformer OpenLV ISD (LV-CAP™ Platform) Busbar Monitoring (Rogowski Coils) Fuse board & fee MCU520 Feeder Monitoring (Rogowski Coils) This test sequence is concerned with verifying the MCU520 and associated monitoring devices (Rogowski Coils) are operational and correctly installed. In order to be successful, the following elements must be functioning correctly: • Rogowski Coils; • MCU520; • Wired communications link between the MCU520 and the LV-CAP™ platform; and • MCU520 Sensor Container.
Starting condition:	The ISD enclosure shall be installed in line with the method statement and specific on-site requirements. This test sequence shall follow from the previous test, 1.01, and consequently, the testing laptop shall be already connected. If for any reason the testing laptop is not connected, follow step 1 of Test SAT: 1.01 before commencing the steps of Test SAT: 1.02.

Test sequence:		*Action*	*Expected result*	*Pass / Fail*	
	1	Using the testing laptop, monitor the MQTT message broker and confirm data readings are provided by the MCU520 platform at 10-second intervals.	Data should be published onto the message broker at 10-second intervals, providing the voltage at the busbars and current readings from both the Rogowski coils.	Pass ☐	Fail ☐
	2	Identify a suitable approach to use the portable ammeter to measure the current in each phase on the connection to the busbars.			
	3	Using the portable ammeter, monitor the current in the first phase of the power connector from the transformer and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the busbar Rogowski Coils are installed correctly (right direction) and connected to the correct terminal on the MCU520.	Pass ☐	Fail ☐
	4	Using the portable ammeter, monitor the current in the second phase of the power connector from the transformer and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the busbar Rogowski Coils are installed correctly (right direction) and connected to the correct terminal on the MCU520.	Pass ☐	Fail ☐
	5	Using the portable ammeter, monitor the current in the third phase of the power connector from the transformer and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the busbar Rogowski Coils are installed correctly and connected to the correct terminal on the MCU520.	Pass ☐	Fail ☐

	6	Identify a suitable approach to use the portable ammeter to measure the current in each phase on the feeder being utilised in the project.		
	7	Using the portable ammeter, monitor the current in the first phase of the feeder cable and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the feeder Rogowski coils are installed correctly (right direction) and connected to the correct terminal on the MCU520.	Pass ☐ Fail ☐
	8	Using the portable ammeter, monitor the current in the second phase of the feeder cable and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the feeder Rogowski coils are installed correctly (right direction) and connected to the correct terminal on the MCU520.	Pass ☐ Fail ☐
	9	Using the portable ammeter, monitor the current in the third phase of the feeder cable and compare this reading against the equivalent reading published to the message broker.	The two readings should be the same, within a reasonable margin for error. If not, verify that the feeder Rogowski coils are installed correctly (right direction) and connected to the correct terminal on the MCU520.	Pass ☐ Fail ☐
	10	Compare the three phase angles reported to the message broker.	The three phases should be separated by approximately 120 degrees, within the margins of reasonable monitoring error.	Pass ☐ Fail ☐

	11	The ability to complete the above tests provides assurance that the sensors, connections to the MCU520 platform and communication link between the MCU520 and LV-CAP™ platform are operating correctly.	It is expected that LV network readings will be published to the message broker allowing the previous stages to be successfully completed.	Pass ☐ Fail ☐
Comments:				

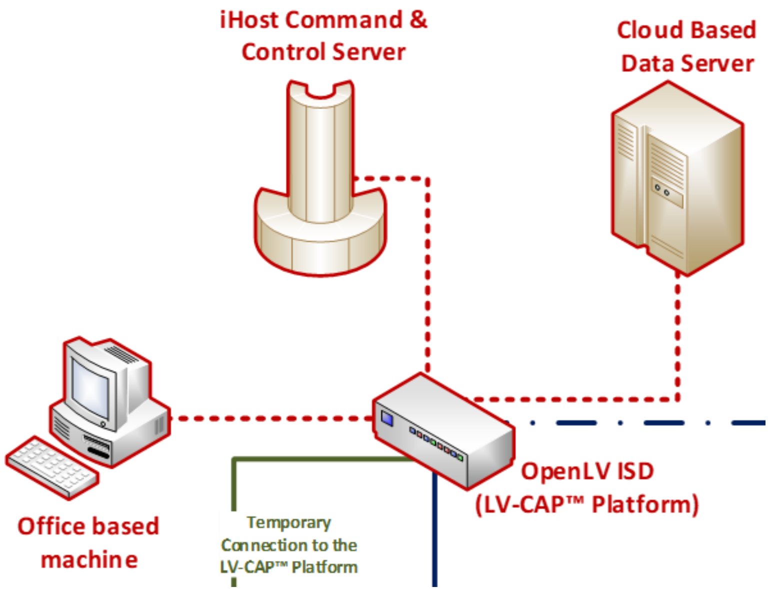
Test: SAT 1.03

Objective:	To verify that the thermocouples are providing reasonably accurate data into the LV-CAP™ platform.			
Elements under test:	 Temperature Probes Transformer OpenLV ISD (LV-CAP™ Platform)	This test sequence is concerned with verifying the thermocouples are connected correctly to the LV-CAP™ platform and are functioning as expected. In order to be successful, the following elements must be functioning correctly: • Thermocouples (multiple, but quantity site dependent); • Digital I/O module; and • Thermal monitoring software container.		
Starting condition:	The ISD enclosure shall be installed in line with the method statement and specific on-site requirements. This test sequence shall follow from the previous test, 1.01, and consequently, the testing laptop shall be already connected. If for any reason the testing laptop is not connected, follow step 1 of Test SAT: 1.01 before commencing the steps of Test SAT: 1.03.			

Test sequence:		***Action***	***Expected result***	***Pass / Fail***
	1	Using the testing laptop, monitor the MQTT message broker and confirm data readings are provided by the thermal monitoring software at 10-second intervals.	Data should be published onto the message broker at 10-second intervals, providing temperature readings for each connected thermocouple.	Pass ☐ Fail ☐
	2	Using the handheld thermometer, determine the outdoor air temperature of the substation in close proximity to the location of the thermocouple installed within the radiation shield.	The reading provided by the thermocouple should be reasonably close to the separately monitored value.	Pass ☐ Fail ☐

	3	If installing in an indoor substation: Using the handheld thermometer, determine the ambient air temperature of the substation in close proximity to the location of the thermocouple installed within the substation.	The reading provided by the thermocouple should be reasonably close to the separately monitored value.	Pass ☐ Fail ☐ N/A ☐
	4	In comparison to the temperature readings determined in steps 3 and 4 above, determine if the measured reading from the transformer thermocouple appears reasonable.	The reading provided by the transformer thermocouple should be warmer than the ambient air temperatures, although site specific conditions will determine by how much.	Pass ☐ Fail ☐
Comments:				

Test: SAT 1.04

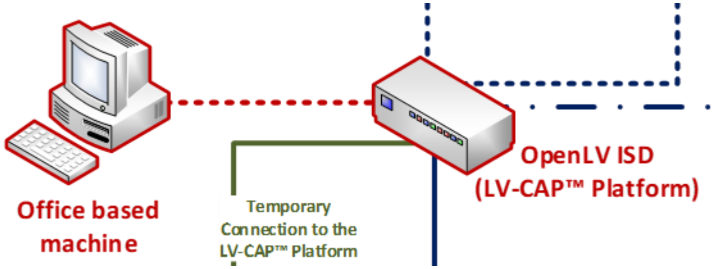
<table>
<tr><td><b>Objective:</b></td><td colspan="4">To verify that that data is being received by the iHost control server and the Cloud Data Server.</td></tr>
<tr><td><b>Elements under test:</b></td><td colspan="4">

iHost Command & Control Server
Cloud Based Data Server
Office based machine
Temporary Connection to the LV-CAP™ Platform
OpenLV ISD (LV-CAP™ Platform)

This test sequence is concerned with verifying the router / modem is correctly recorded as being installed at the location in question and that the mobile data connection is capable of allowing remote access to the unit.
In order to be successful, the following elements must be functioning correctly:
• Router / modem module;
• Mobile data network;
• Connection to iHost Command and Control Server; and
• Connection to Cloud Based Data Server.</td></tr>
<tr><td><b>Starting condition:</b></td><td colspan="4">The ISD enclosure shall be installed in line with the method statement and specific on-site requirements.
This test does not require the use of the testing laptop, if it is still connected from the previous test sequence, disconnect it from the LV-CAP™ platform but do not close and seal the enclosure.</td></tr>
<tr><td colspan="2"><b>Test sequence:</b></td><td><b><i>Action</i></b></td><td><b><i>Expected result</i></b></td><td><b><i>Pass / Fail</i></b></td></tr>
<tr><td></td><td>1</td><td>On-site testing crew to telephone a designated, office-based colleague.</td><td></td><td></td></tr>
</table>

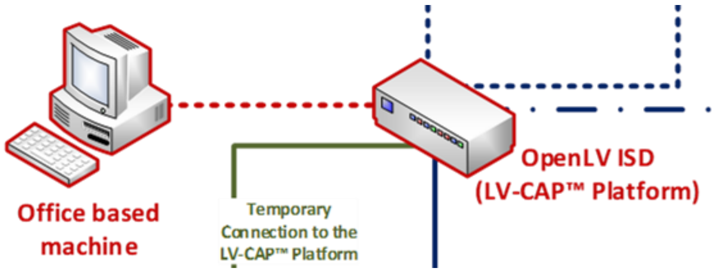
	2	Office-based test crew to login to the iHost Command and Control Server and verify that the LV-CAP™ platform in question has connected to the server and transferred data, at 10-second intervals.	iHost server contains data from the system having been uploaded since installation.	Pass ☐ Fail ☐
	3	Office-based test crew to login to the iHost Command and Control Server and verify that the LV-CAP™ platform in question has connected to the server and transferred data, at 10-second intervals.	Cloud based data server contains data from the system having been uploaded since installation.	Pass ☐ Fail ☐
Comments:				

Test: SAT 1.05

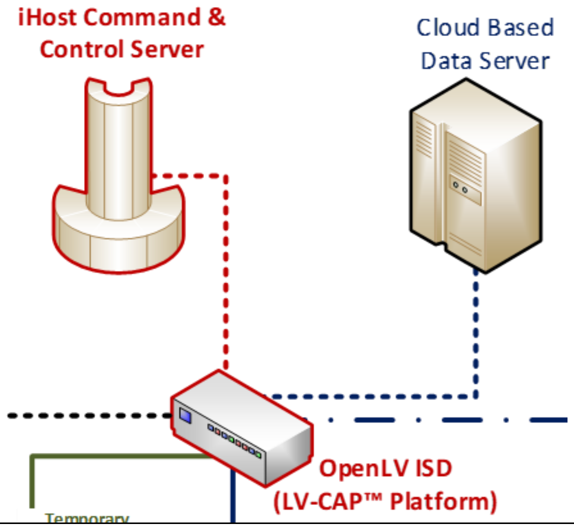
<table>
<tr><td colspan="2"><b>Objective:</b></td><td colspan="3">To verify that:
1. The router – modem and SIM card combination installed within the ISD is correctly recorded;
2. The router – modem can be remotely accessed; and
3. That data transfer to the servers resumes after a system restart.</td></tr>
<tr><td colspan="2"><b>Elements under test:</b></td><td colspan="3">
Office based machine
Temporary Connection to the LV-CAP™ Platform
OpenLV ISD (LV-CAP™ Platform)
This test sequence is concerned with verifying the router / modem is correctly recorded as being installed at the location in question and that the mobile data connection is capable of allowing remote access to the unit.
In order to be successful, the following elements must be functioning correctly:
• Router / modem module; and
• Mobile data network.</td></tr>
<tr><td colspan="2"><b>Starting condition:</b></td><td colspan="3">The ISD enclosure shall be installed in line with the method statement and specific on-site requirements.
This test does not require the use of the testing laptop, if it is still connected for any reason, disconnect it from the LV-CAP™ platform but do not close and seal the enclosure.</td></tr>
<tr><td colspan="2"><b>Test sequence:</b></td><td><b><i>Action</i></b></td><td><b><i>Expected result</i></b></td><td><b><i>Pass / Fail</i></b></td></tr>
<tr><td></td><td>1</td><td>On-site testing crew to telephone a designated, office-based colleague if not still connected to them from previous test-sequence.</td><td></td><td></td></tr>
<tr><td></td><td>2</td><td>Office-based testing crew to access the router / modem using the process detailed in 0.</td><td>Office-based crew should be able to access the router / modem command console.</td><td>Pass ☐ Fail ☐</td></tr>
</table>

	3	Whilst the on-site test crew watch the console, the office-based test crew should trigger a hard-rest of the platform.	If successful, the power relay for the LV-CAP™ platform will illuminate, confirming the correct modem was being communicated with and that remote reset capability is available for the site. Site-based test crew will confirm successful shutdown of the unit through lack of LED activity on network ports.	Pass ☐ Fail ☐
	4	Office-based test crew to re-energise the LV-CAP™ platform.	Site-based crew, will observe the relay light deactivate, demonstrating power has been restored the LV-CAP™ platform. Network ports on the platform will begin flashing signifying restoration of power to the unit.	Pass ☐ Fail ☐
	5	Site-based test crew to seal and lock the enclosure.		
Comments:				

Test: SAT 1.06

Objective:	To verify that the LV-CAP™ platform resumes communication with the iHost and Cloud Data Servers following a system restart.			
Elements under test:	 Office based machine Temporary Connection to the LV-CAP™ Platform OpenLV ISD (LV-CAP™ Platform) This test sequence is concerned with verifying the router / modem is correctly recorded as being installed at the location in question and that the mobile data connection is capable of allowing remote access to the unit. In order to be successful, the following elements must be functioning correctly: • Router / modem module; and • Mobile data network.			
Starting condition:	The ISD enclosure shall be installed in line with the method statement and specific on-site requirements. This test does not require the use of the testing laptop, if it is still connected for any reason, disconnect it from the LV-CAP™ platform then close and seal the enclosure.			
Test sequence:		***Action***	***Expected result***	***Pass / Fail***
	1	Office-based test crew to login to the iHost Command and Control Server and verify that the LV-CAP™ platform in question has re-connected to the server and is transferring data, at 10-second intervals.	System will restart following the restoration of power and following a period of a few minutes, recommence uploading data to the servers.	Pass ☐ Fail ☐
Comments:				

Test: SAT 1.07

Objective:	To verify that remote update capability of the LV-CAP™ platform configuration can be achieved.
Elements under test:	 iHost Command & Control Server Cloud Based Data Server OpenLV ISD (LV-CAP™ Platform) Temporary This test sequence is concerned with verifying the router / modem is correctly recorded as being installed at the location in question and that the mobile data connection is capable of allowing remote access to the unit. In order to be successful, the following elements must be functioning correctly: • IHost Command & Control Server; • Router / modem module; and • Mobile data network.
Starting condition:	The ISD enclosure shall be installed in line with the method statement and specific on-site requirements. This test does not require the use of the Testing Laptop; if it is still connected for any reason, disconnect it from the LV-CAP™ platform then close and seal the enclosure. The LV-CAP™ platform shipped following testing with the sensors configured for 10-second reporting and the outputs of those sensors assigned for upload to the iHost and Cloud Based Data Servers.

Test sequence:		***Action***	***Expected result***	***Pass / Fail***
	1	Office-based test crew to login to the iHost Command and Control Server and change the configuration settings for the site in question such that sensor reporting is only required at one-minute intervals.		Pass □ Fail □
	2	Office-based test crew to verify that data reporting back to the iHost and Cloud Based Data Servers reduces from a rate of once every ten seconds to once every minute.	The LV-CAP™ platform will reduce the rate of data capture and this will be reflected in a reduction in the data uploaded to the two servers.	Pass □ Fail □
Comments:				

OpenLV system tested elements

The below diagram, shows each element, (highlighted in red) of the deployed system that has been tested, as part of the Phase 1 SATs.

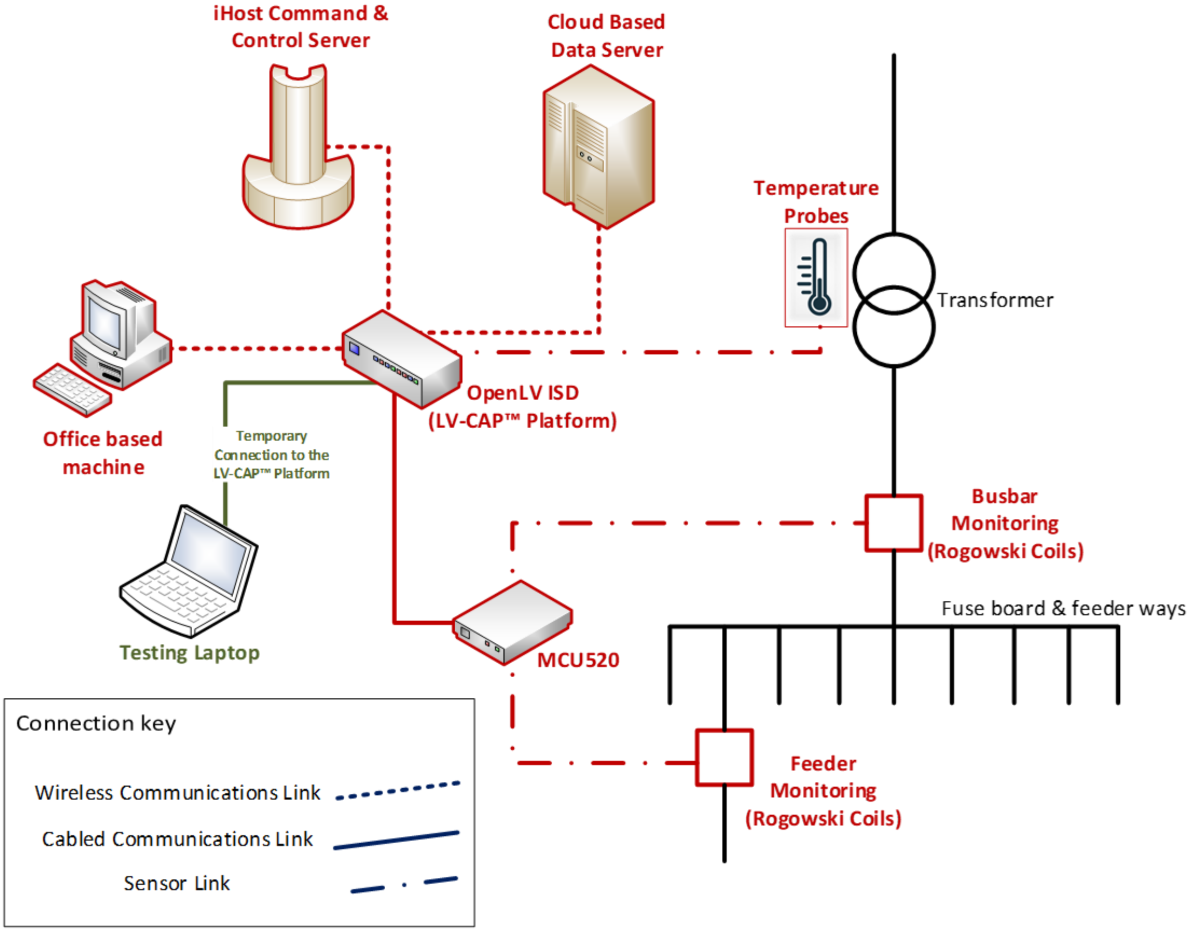
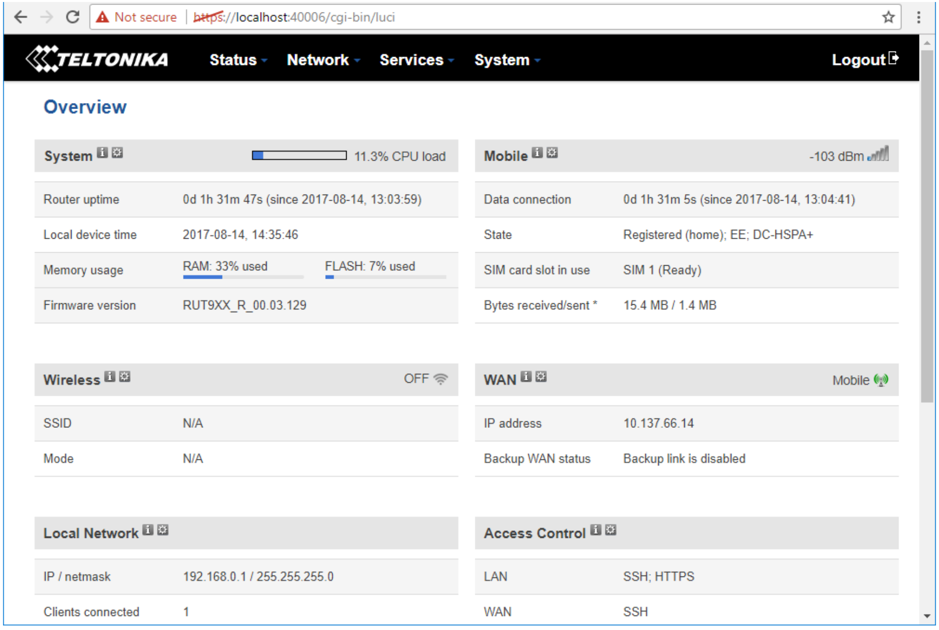

iHost Command & Control Server
Cloud Based Data Server
Temperature Probes
Transformer
Office based machine
OpenLV ISD (LV-CAP™ Platform)
Temporary Connection to the LV-CAP™ Platform
Busbar Monitoring (Rogowski Coils)
Fuse board & feeder ways
Testing Laptop
MCU520
Feeder Monitoring (Rogowski Coils)
Connection key
Wireless Communications Link
Cabled Communications Link
Sensor Link

Figure 2 - OpenLV Trial System - Tested Elements

Appendix A. Router / modem access routine

1. Connect to Wireless Logic SSL VPN (NetExtender) using user ssl_lv
2. Connect SSH session to 4G router using Putty:
   a. IP: see SIM records (10.x.x.x)
   b. Port: 8192
   c. Username: root
3. Forward local port “40006” (e.g.) to “localhost:443”
4. Open a web browser and connect to https://localhost:40006
5. Accept the security error for the self-signed router SSL certificate
6. Log in to web interface with user Admin



Overview

System	11.3% CPU load
Router uptime	0d 1h 31m 47s (since 2017-08-14, 13:03:59)
Local device time	2017-08-14, 14:35:46
Memory usage	RAM: 33% used FLASH: 7% used
Firmware version	RUT9XX_R_00.03.129

Mobile	-103 dBm
Data connection	0d 1h 31m 5s (since 2017-08-14, 13:04:41)
State	Registered (home); EE; DC-HSPA+
SIM card slot in use	SIM 1 (Ready)
Bytes received/sent *	15.4 MB / 1.4 MB

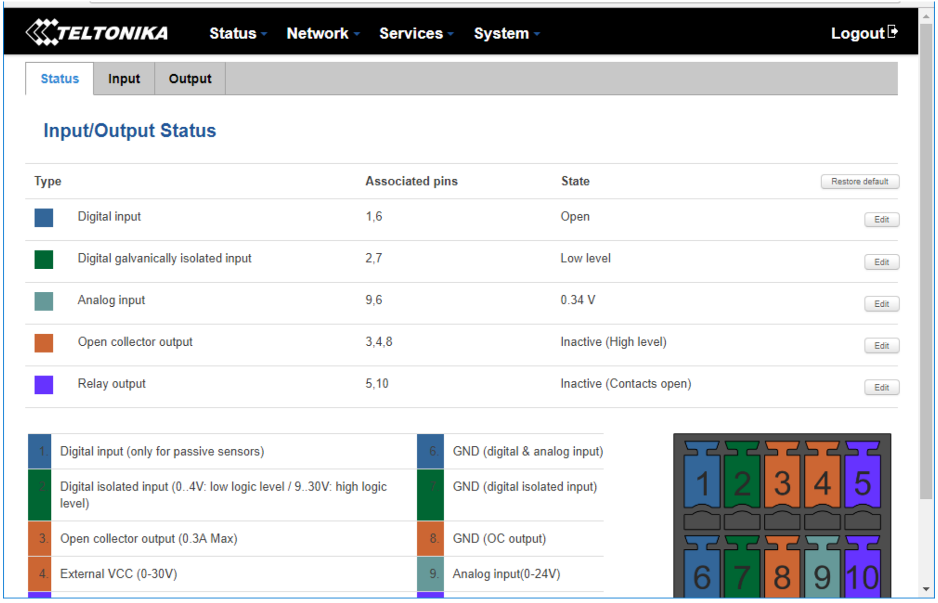
Wireless	OFF
SSID	N/A
Mode	N/A

WAN	Mobile
IP address	10.137.66.14
Backup WAN status	Backup link is disabled

Local Network	
IP / netmask	192.168.0.1 / 255.255.255.0
Clients connected	1

Access Control	
LAN	SSH; HTTPS
WAN	SSH

7. Click on Services > Input/Output.

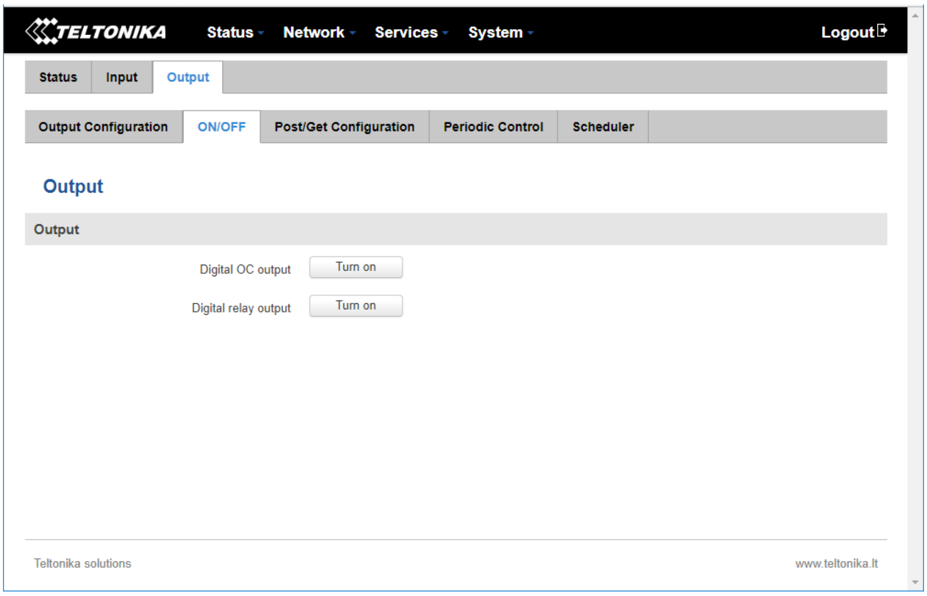


Input/Output Status

	Type	Associated pins	State	Restore default
■	Digital input	1,6	Open	Edit
■	Digital galvanically isolated input	2,7	Low level	Edit
■	Analog input	9,6	0.34 V	Edit
■	Open collector output	3,4,8	Inactive (High level)	Edit
■	Relay output	5,10	Inactive (Contacts open)	Edit

1.	Digital input (only for passive sensors)	6.	GND (digital & analog input)
2.	Digital isolated input (0..4V: low logic level / 9..30V: high logic level)	7.	GND (digital isolated input)
3.	Open collector output (0.3A Max)	8.	GND (OC output)
4.	External VCC (0-30V)	9.	Analog input(0-24V)

8. Click on the Output sub-tab and then the ON/OFF sub-tab.



9. Click the “Digital OC output” Turn On button. This will remove power from the PC.

After the required delay, click the Turn Off button. This will re-apply power to the PC and allow it to start up.