



# High Level Design

## Seagate Storage Platform Library

<http://goo.gl/t2bjc3>  
Seagate Technology PLC

### Security Notice

This drawing/document, its specification, and the data contained herein, are the exclusive property of Seagate Technology PLC, issued in strict confidence and shall not, without the prior written permission of Seagate Technology PLC, be reproduced, copied, or disclosed to a third party, or used for any purpose whatsoever, except for the manufacture of articles for Seagate. This drawing/document is prepared by Seagate Technology PLC © 2014.

## Revision History

| Revision | Date       | Author/Editor | Comments         |
|----------|------------|---------------|------------------|
| 0.1      | 2014-11-27 | Andy Nimmo    | Initial document |

# Table of Contents

[Revision History](#)

[Table of Contents](#)

[Introduction](#)

[SSPL - High Level Verbs \(SSPLHL\)](#)

[Configuration Verbs](#)

[Management Verbs](#)

[Telemetry Interface](#)

[SSPL - Low Level Verbs \(SSPL3\)](#)

[Actuators](#)

[Sensors](#)

[Use Cases](#)

[Temperature Reading](#)

[Outstanding Questions](#)

[Communication mechanisms for the following:](#)

[Interfaces to be decided upon:](#)

## Introduction

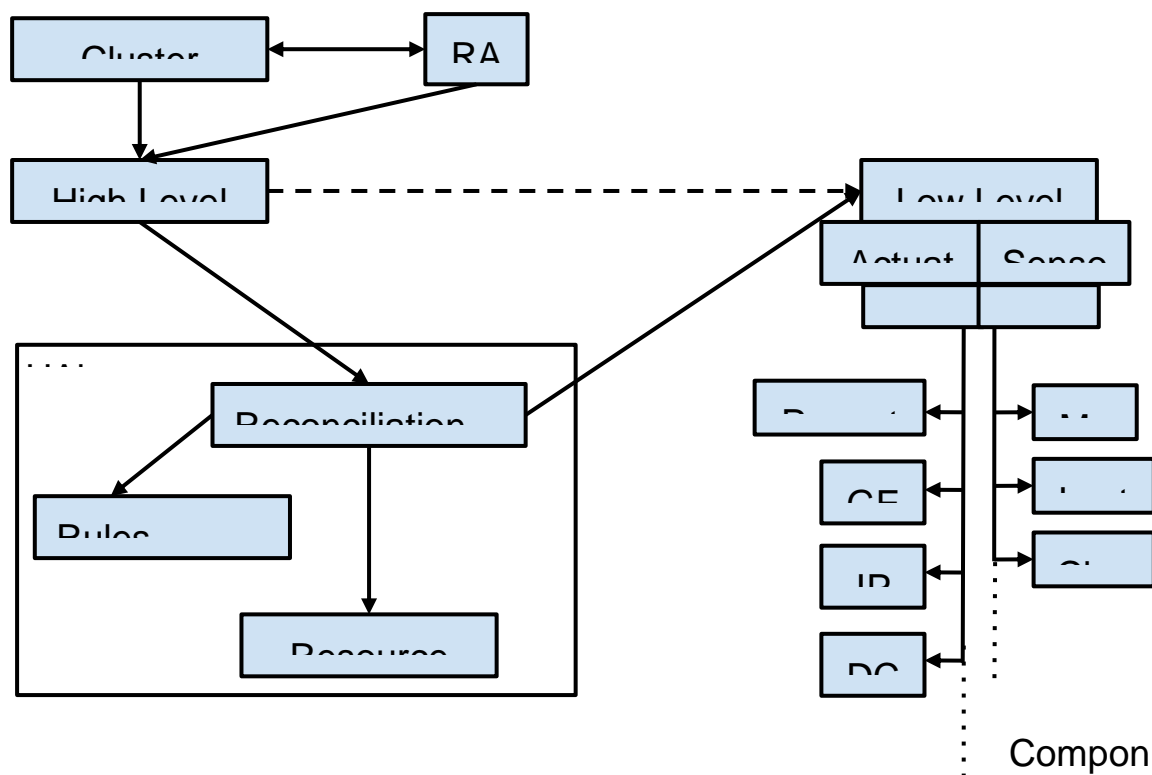
Platform functionality will be provided via the Seagate Storage Platform Library (SSPL), a library of functions designed to provide a multi-tiered common interface to the storage platform.

SSPL consists of two layers of interfaces (verbs) that together provide an abstracted interface between all the components of the Mero solution.

These two layers are the High Level (SSPLHL) and Low Level (SSPLLL) subcomponents which work together to provide the layered abstraction of hardware components and software stack with the aim of providing an agnostic interface to the entire platform.

The High Level (HL) component provides abstraction of the underlying cluster to a top level consumer such as a generic management UI/CLI and the Seagate RAS system.

The Low Level (LL) component provides abstraction from the underlying hardware and software stack that comprises the storage platform. This will be but isn't limited to the Core Mero software stack; the Puppet provisioning toolset and the RAS DCS collector subsystem.



## SSPL - High Level Verbs (SSPLHL)

The purpose of the HL component of the SSPL is to provide a management and configuration interface for carrying out such actions as on-the-fly Mero configuration changes; service and filesystem control and supply telemetry and platform metrics.

It is expected that most commands accessing the HL library will not be acted upon directly by the platform library but will in fact be sent to the HALon High Availability subsystem for processing as a cluster wide event in order to ensure that the only commands executed are those that can be completed without detrimental impact of the operation of the system as a whole.

It is reasonable for us to expect that not all commands should flow via the HALon subsystem and for those outside cases the requirement to utilise the HALon subsystem will be relaxed such that those commands can progress immediately to the SSPL (Low Level) library. Examples of such a command could be the querying of a system metric not recorded or monitored by the HALon subsystem for example Puppet manifest data or control of some other 3rd party application and data.

In the above cases of non-HALon derived actuation of high level requests or commands the SSPLHL should be able to ascertain the programmatic path for command/control realisation. This programmatic path discovery should not rely on HALon's notion of the ground truth of the system as it is highly likely HALon will have no awareness at any level of the software or data being referenced.

All activity through these HL verbs should be logged, this logging information should include both the verb called; the variables passed to that verb and the response(s) received.

## Configuration Verbs

There are currently expected to be two considerations for the configuration interface part of the HL verbs, these are configuration of Mero for the initial deployment and setup of a filesystem and additional post-deployment configuration activities such as the addition of further storage resources (SSUs, SATI's etc). The removal of such entities is also expected behaviour that this interface should be able to manage. All of these commands are to be sent for execution via the HALon subsystem without exception.

The interface definition for the currently planned Core Mero management capabilities can be found [here](#). This document should form the basis of expected management commands that will need to be accessed via SSPL for Core Mero. The document does not cover other aspects of the system that may need to be configured also such as HALon; DCS collector etc

Due to the nature of these commands they will need to be handled in a transactional manner with return values representing the success or failure of the command.

## Management Verbs

Under the expected design for SSPL there is the requirement for a set of management capabilities which will be utilised by external applications to control those aspects of the storage system that we wish to expose. The sort of management controls we will be needing to provide include but are not limited to tasks such as the starting/stopping of services and filesystems; powering off/on controllers and handling servicing scenarios such as planned disk removals.

As with the above [Configuration Verbs](#) these commands received from a management UI should only be passed onto the HALon component and not handled directly through any other low level access.

## Telemetry Interface

As part of the standard functions of any management tool the ability to request metrics about the performance and state of the underlying system is a necessity. The HALon subsystem holds what is considered to be the ground truth of the state of the system thus for all metrics and status information it holds within its resource graph the query should be processed by HALon itself.

It is foreseen that there will be scenarios in which information is being queried that is being recorded by the DCS Collector but about which HALon has no awareness, in these cases the results should be acquired directly from the node in question through the Low Level verbs. With this information not being held in HALon's resource graph there is a requirement that the HL will be able to identify and access the necessary local node Low Level instance.

## SSPL - Low Level Verbs (SSPL3)

The Low Level verbs are intended to provide complete abstraction from the underlying hardware and software stacks and to be the interface through which actions on the system are carried out.

Low Level functionality is to include the ability to query underlying hardware and software as well as instruction those components to carry out actions directly related to High Availability; configuration changes and metrics gathering.

It is desired that there be an active component to this library such that upstream applications can register an interest in various resources abstracted away from them by the library. The HALon system is relying on the Low Level verbs to generate and transmit a stream of events the HALon stream collectors.

There are three capabilities to be offered by the Low Level verbs these are:

**Action** - Use the actuator code to carry out an operation on the system

**Read** - Access a locally stored value and statically read that back

**Listen** - Register an interest in notification of state changes; all listeners notified when state change detected

Both actuators and monitors are modular in design and should plug into the LL verbs framework. This is so that future actuators and monitors can be added to extend overall SSPL capabilities without upstream changes being required.

### Actuators

The actuators are the parts of the Low Level verbs that are designed to carry out actions on a system and return to the caller the result of these calls.

Services started with optional configuration file paths: Handles scenarios where an alternative config file might be a valid cause for restart of a node service.

### Sensors

Three capabilities are to be offered by the monitor component of the Low Level verbs. This is expected to provide the hardware/software abstraction layer to hide the underlying environment from the higher level systems such as HALon.

The current available metrics available from DCS Collectors can be found [here](#).

SSPL will also likely need to provide an API for access the Mero stats service.

A consumer of the sensors is expected to carry out one of 3 functions, these are as follows:

Register as a listener to be notified

# Outstanding Questions

## Communication mechanisms for the following:

- CLI to SSPLHL
- GUI to SSPLHL
- SSPLHL to HAIon
- HAIon to SSPL3

## Interfaces to be decided upon:

- Interface definition for SSPLHL
  - RESTFUL for future proofed GUI support?
  -
- Interface definition for HAIon
  - How do we handle comms between SSPLHL and SSPL3?

Interaction mechanisms to be supported by SSPL:

passive polling

active callbacks (callback registration, etc)

directives (commands)