Voyager Software Design Specification

Document Number: TBD

Document Revision History

Version	Comment	Author	Date
v0.1	- Initial version	SnapRoute	06/10/16
v0.2	- Added more details on Optic Daemon	SnapRoute	06/27/16
v0.3	- Updated format of all config and state objects - Removed references to web links	SnapRoute	09/29/16

Document Approval History

Name	Signature	Dept	Date

Table of Contents

1. Introduction

1.1 Purpose

The goal of this document is two fold :
(a) Provide a high level description of the software controlling the Voyager Platform, referred to as FlexSwitch
(b) Describe in detail the internal design of software subsystems of FlexSwitch relevant to the Voyager platform

1.2 Scope

The scope of this design document is strictly limited to discuss the general architecture of FlexSwitch and internal design of software subsystems that are relevant to performing management/control and monitoring of the Voyager platform.
This document does not include information regarding any applications that can be built using the infrastructure provided by FlexSwitch to manage and monitor the Voyager platform.

1.3 Definitions and Abbreviations

1.3.1 Definitions

FlexSwitch	Controlling software that runs on the COM-E module on Voyager

1.3.2 Abbreviations

FS	FlexSwitch
REST	Representational State Transfer
ASICd	FS Asic daemon. Software process controlling the packet processing ASIC
OPTICd	FS Optic daemon. Software process controlling the optical modules
FMgrd	FS Fault manager daemon.
Notifierd	FS Alarm notifier daemon. Software process responsible for publishing system alarms.
TCA	Threshold crossing alarms

2. FlexSwitch system architecture

FlexSwitch refers to the suite of software that controls the underlying Voyager platform. FlexSwitch is designed based on the micro service design pattern. The microservice architecture/design pattern implies that the software suite is designed to be a collection of independently deployable, modular services each serving a specific function.

The FlexSwitch software suite is essentially decomposed into the following categories of microservices/software daemons :

- Hardware/Platform management daemons (asicd, opticd, platformd)
- L2/L3 Protocol daemons (stpd, lacpd, bgpd, arpd, ribd etc.)
- Infrastructure daemons (sysd)
- Configuration management daemon (confd)
- Monitoring and performance management (fMgrd, notifierd)

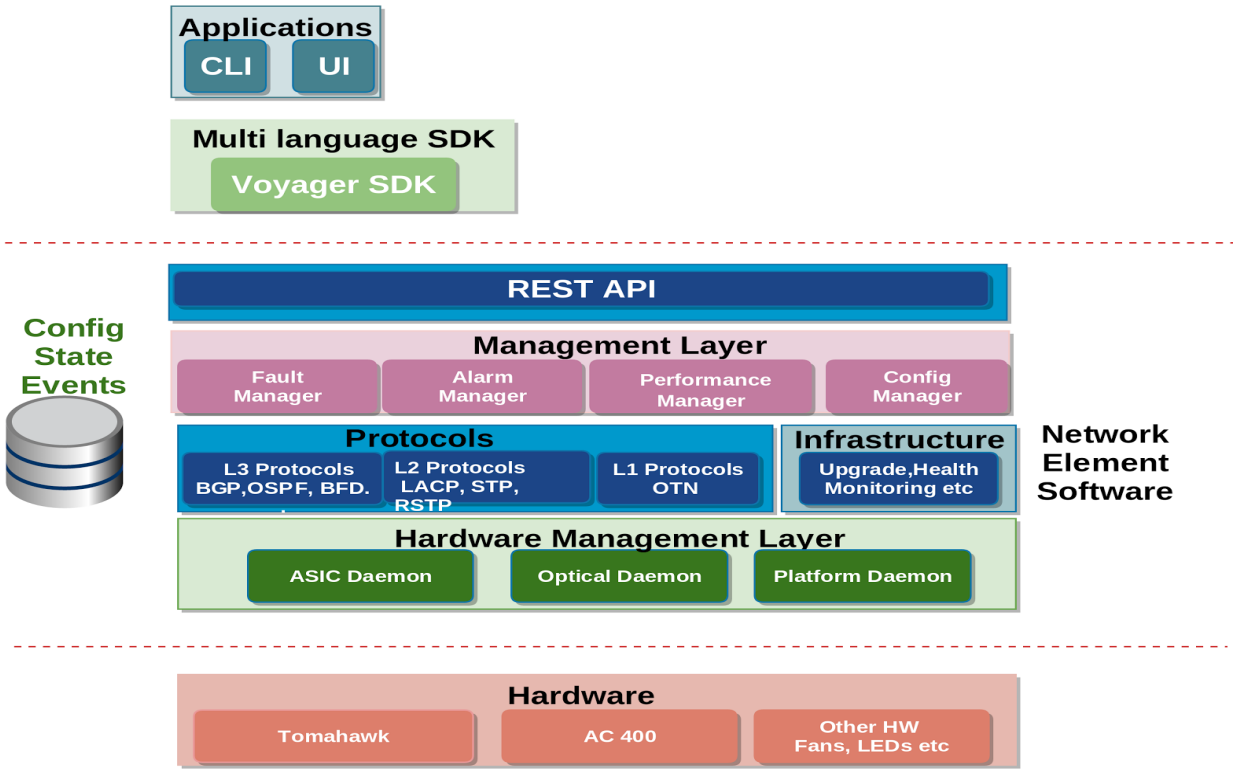


Figure 1: Diagram showing Flexswitch microservices

The diagram in Figure 1 shows a high level layered representation of components that constitute FlexSwitch's microservices. At the bottom of the network element software stack are the hardware/platform management daemons. These daemons allow configuration, management and monitoring of underlying hardware. They also provide service to higher layer software that allows provision of the hardware. The middle layer of the stack is comprised of the protocol and infrastructure daemons, that provide core switching and routing functionality. Above the protocol/infrastructure layer resides the management layer that allows configuration of various services, and also performs monitoring and reports the health of the system via events, faults and alarms. The configuration management daemon i.e 'configd' exposes REST APIs for the end user/application to manage the system.

A multi-language SDK layer is provided for making application development easy. Reference applications are provided as part of the FlexSwitch software suite, including applications like CLI, UI, that use the REST APIs.

The following sections provide a brief overview of FlexSwitch software components relevant to the Voyager platform

2.1. Config Daemon

Config Daemon, serves as the termination point for all REST API requests made to the FlexSwitch software suite. Once incoming REST APIs are terminated, further processing as required for the incoming request is dispatched to the appropriate FlexSwitch daemon. Once the request has been served, an appropriate HTTP response is provided.

2.2. Optical Daemon

The Optical daemon within the FlexSwitch software suite is the primary software process that controls the optical hardware subsystem on Voyager. Voyager employs Acacia AC400 optical modules that provide optical transport.

Optical daemon software is designed to be independent of the underlying hardware and hence serves as a hardware abstraction layer for optical modules from different vendors. This is achieved by employing a plugin based design where all vendor specific code is imported at runtime based on the plugins that are instantiated.

2.3. Asic Daemon

The Asic daemon within the FlexSwitch software suite serves as the primary software process that controls the ethernet processing subsystem on Voyager. Voyager employs the Broadcom Tomahawk asic for ethernet packet processing.

Asic daemon software is designed to be independent of the underlying hardware, hence it essentially serves as a hardware abstraction layer (HAL) for ethernet packet processors. It provides a common northbound API interface is exposed to all protocol daemons. This interface allows provisioning a range of packet processing ASICs from different vendors.

2.4. L2/L3 Protocol Daemons

FlexSwitch software suite supports various layer 2 switching and layer 3 routing protocols. Information regarding these protocol daemons is outside the scope of this design document

2.5. Fault management Daemon

The FlexSwitch software suite has a fault management daemon that is responsible for monitoring system health and tracking all faults in the system. Fault manager is designed to be completely data driven, whose behavior can be customized by editing the control data contained in configuration files in the JSON format. Fault manager utilizes the alarm manager(library) to raise alarms.

3. Detailed design

The following sections delve into detailed design aspects of the FlexSwitch software daemons relevant to the Voyager platform. Each section also describes the configuration and state objects that are exposed by the corresponding daemon.

3.1. Config Daemon

The Config Daemon, serves as the front end, terminating all REST requests that are made to the FlexSwitch software suite.

A fundamental aspect of the configuration and management software design, employed by the FlexSwitch software suite, is the development of a data model or a data object. The data model is a collection of parameters that map to appropriate control inputs and observation points in each underlying physical resource or a physical resource group.

Each data model or model object has a primary owner daemon. The owner for each object supplies the following primitives as applicable to the object

- Create : Creates an instance of the object with the supplied parameters
- Update : Modifies all or a subset of attributes of the specified object
- Retrieve : Returns the values of a single object identified by a key or returns a list of objects of a particular type
- Delete : Deletes the instance corresponding to the object specified

Any incoming request from an external application is proxied by config daemon to the owner daemon based on the object referred to in the request.

In addition to serving as a front end for configuration requests, the Config daemon also ensures persistence of all user applied configuration. This is achieved by writing configuration requests to a DB and replaying configuration from DB when software is restarted.

Config daemon also provides support for functionality such as dumping/saving configured state of a device, bulk application of configuration and resetting all configuration to a default/known state.

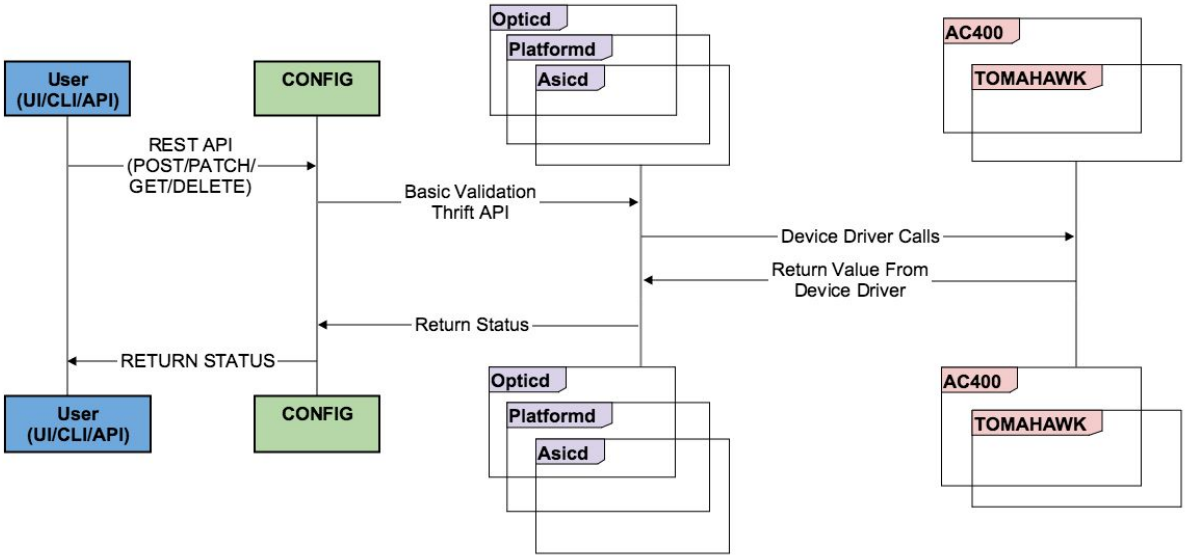


Figure 2: Transaction diagram for configuration requests

The diagram shown in Figure 2 above, illustrates an API call that is initiated by a GUI or CLI. This API call originates an appropriate HTTP request that will be terminated at the Config Daemon. Based on the incoming request, Config Daemon then dispatches this request to the appropriate owner (asicd, opticd, platformd etc.) which in turn executes one or more functions per the incoming request.

Once the request is handled by the appropriate daemon, a return status code is sent to the Config Daemon, which in turn sends a return status to the requester via an HTTP response.

3.2. Optical Daemon

As mentioned in earlier sections, the optical daemon is primarily responsible for configuration and management of the optical subsystem. Figure 3 provides a detailed representation of the internals of the optic daemon and it's interfaces to other software components within the FlexSwitch software suite.

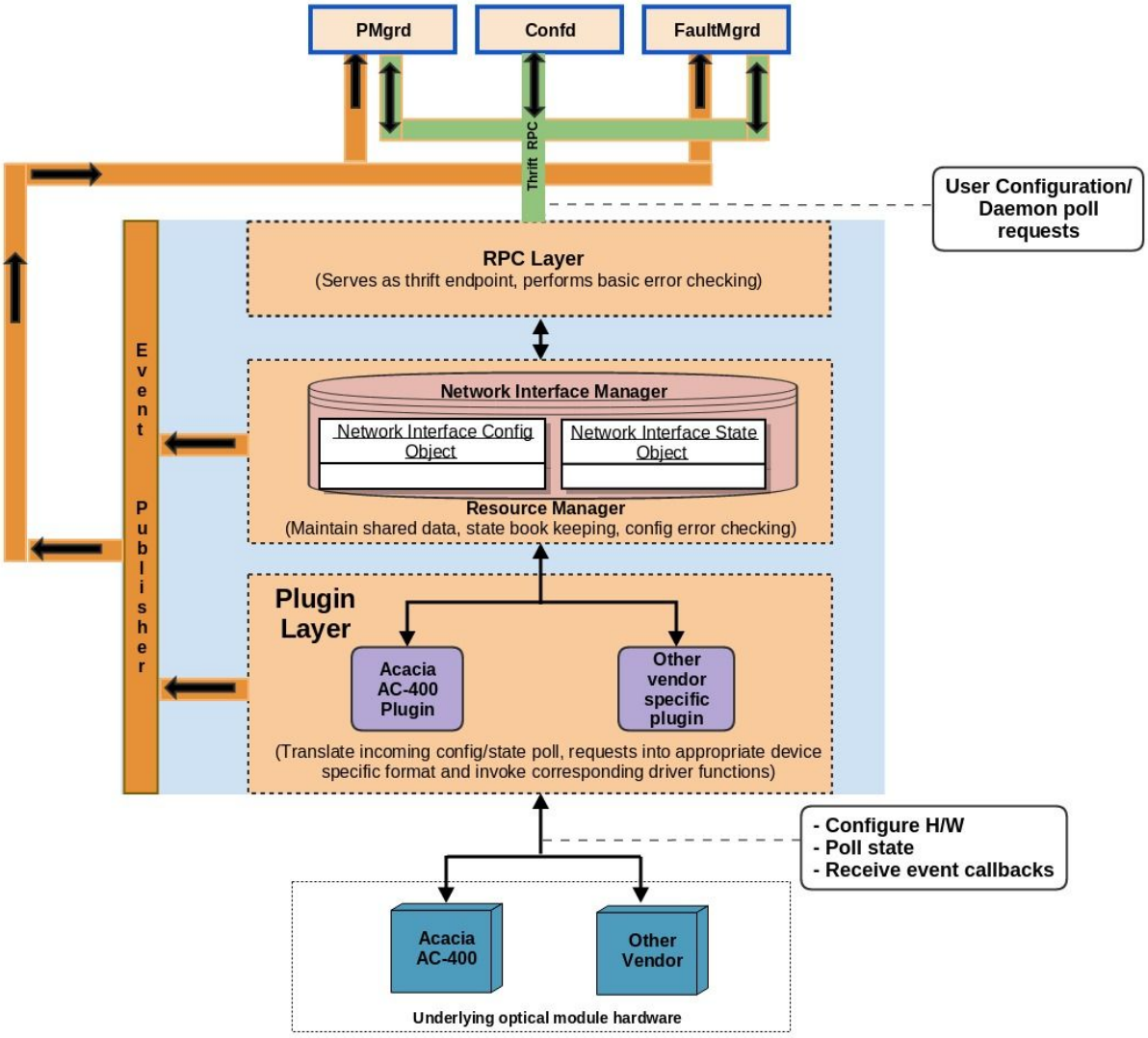


Figure 3 : Diagram illustrating internals of the optic daemon and its interfaces

The optical daemon exposes a northbound thrift API that provides services that allow configuring and monitoring the AC400 modules. The sections below describe the configuration

and state objects that are made available by the optic daemon to allow control and monitoring of the :

- AC400 module as a whole
- Individual network interfaces on each AC400
- Individual client interfaces on each AC400

3.2.1. Configuration Objects

The following section details the configuration objects that are provided to perform configuration/management of the optical subsystem

Voyager System Config Object - XponderGlobal		
Attribute	**Description**	**Possible Values**
XponderId	This is a system generated number identifying the transponder system as a whole	● 0 - 255
XponderMode	This attribute defines the configuration mode of the transponder system as a whole. Where defined, each mode when set, applies a predefined set of configurations to the various subsystems within the transponder	● InServiceWire ● InServiceRegen ● InServicePacketOptical ● OutOfService
XponderDescription	This attribute is a user configurable system description	● Custom string

In addition to the system level parameters, Optical Daemon also provides the following attributes to be configured independently for each of the AC400 module

AC400 Config Object - DWDMModule		
Attribute	**Description**	**Possible Values**
ModuleId	This is a system generated unique module identifier	● 0 - 255
AdminState	This attribute controls whether the optical module is held in reset/out of reset	● UP ● DOWN

IndependentLaneMode	This is a boolean attribute that controls whether the AC400 module operates as two independent lanes or in coupled lane mode	• true • false
PMInterval	This attribute determines the interval between successive ticks of the PM subsystem in the AC400 modules	• 1 -255
EnableExtPMTickSrc	This attribute specifies to the PM subsystem in the AC400 module that an external clock source is used to generate PM ticks	• true • false

The Optical daemon provides the following configuration object that allows independent control of the network interface

Network Interface Config Object - DWDMModuleNwIntf		
Attribute	**Description**	**Possible Values**
ModuleId	This is a unique system generated identifier for the AC400 module	• 0 - 255
NwIntfId	This is a unique system generated identifier for the network interface on the AC400 module	• 0 - 255
ModulationFmt	This attribute specifies the modulation format to be used on this network interface	• QPSK • 8QAM • 16QAM
TxPower	This attribute specifies the transmit power to use for this network lane in dBm	• -30 - 10
ChannelNumber	This attribute specifies the channel number to use for this network lane	• 1 - 100
FECMode	This attribute specifies the forward error correction mode to be applied to this network lane	• 15%SDFEC • 15%OvrHeadSDFEC • 25%OvrHeadSDFEC
DiffEncoding	This attribute specifies whether the given network lane uses differential encoding	• false • true
TxPulseShapeFltrType	This attribute specifies the type of pulse shaping filter to use for this network lane	• RootRaisedCos

		• RaisedCos • Gaussian
TxPulseShapeFltrRollOff	This attribute specifies the roll off factor to use for the pulse shaping filter	• 0 - 1
AdminState	This attribute specifies whether the given network lane is enabled or not	• UP • DOWN
EnableTxPRBS	This attribute controls whether PRBS generation is enabled on this network lane	• false • true
TxPRBSPattern	This attribute specifies the pattern to be used for PRBS generation on this network lane	• 2^7 • 2^15 • 2^23 • 2^31
TxPRBSInvertPattern	This attribute specifies whether the PRBS pattern generated on this network lane is to be inverted	• false • true
RxPRBSChecker	This attribute specifies the receive PRBS pattern to check against on this network lane	• 2^7 • 2^15 • 2^23 • 2^31
RxPRBSInvertPattern	This attribute specifies whether the PRBS pattern is to be inverted prior to the check being performed	• false • true
ClntIntfIdToTributary0Map	This attribute maps a specific client interface to tributary 0 of this network lane	• 0 - 3
ClntIntfIdToTributary1Map	This attribute maps a specific client interface to tributary 1 of this network lane	• 0 - 3
TxPowerRampdBmPerSec	This attribute controls/limits the rate of change of TX power on this network lane	

Client Interface Config Object - DWDMModuleClntIntf		
Attribute	**Description**	**Possible Values**
ModuleId	This is a unique system generated identifier for the AC400 module	• 0 - 255
ClntIntfId	This is a unique system generated identifier for the client interface on the AC400 module	• 0 - 255
TXFECDecDisable	This attribute controls whether the TX FEC decoder is enabled or disabled	• false • true
RXFECDecDisable	This attribute controls whether the RX FEC decoder is enabled or disabled	• false • true
HostTxEqLfCtle	This attribute specifies the host interface TX deserializer equalization, LELPZRC LF-CTLE LFPZ gain code	• 0 -8
HostTxEqCtle	This attribute specifies the host interface TX deserializer equalization, LELRC CTLE LE gain code.	• 0 - 20
HostTxEqDfe	This attribute specifies the host interface TX deserializer equalization, s-DFE, DFE tap coefficient	• 0 - 63
HostRxSerializerTap0Gain	Host RX Serializer tap 0 control, gain for equalization filter tap	• 0 - 7
HostRxSerializerTap0Delay	Host RX Serializer tap 0 control, delay for equalization filter tap	• 0 - 7
HostRxSerializerTap1Gain	Host RX Serializer tap 1 control, gain for equalization filter tap	• 0 - 7
HostRxSerializerTap2Gain	Host RX Serializer tap 2 control, gain for equalization filter tap	• 0 - 15
HostRxSerializerTap2Delay	Host RX Serializer tap 2 control, delay for equalization filter tap	• 0 - 7
AdminState	This attribute specifies whether the given network lane is enabled or not	• UP • DOWN
EnableTxPRBSChecker	This attribute controls the enable/disable state of TX PRBS checkers for all lanes of this client	• false

	interface	• true
TxPRBSPattern	This attribute specifies the pattern to be used for PRBS checker on this client interface	• 2^7 • 2^15 • 2^23 • 2^31
EnableRxPRBS	This attribute controls the enable/disable state of RX PRBS generator for all lanes of this client interface	• false • true
RxPRBSPattern	This attribute specifies the pattern to use for PRBS generation on this client interface	• 2^7 • 2^15 • 2^23 • 2^31
EnableIntSerdesNWLoopback	Enable/Disable serdes internal loopback, N/W RX is looped back to N/W TX	• false • true
EnableHostLoopback	Enable/Disable loopback on all host lanes of this client interface	• false • true
NwLaneTributaryToClntIntfMap	This attribute controls the network lane and tributary that maps to this client interface	• 0 - 3

3.2.2. State Objects

The Optic daemon provides the following state objects that contain several read only attributes which reflect the current state of various parameters of the optical subsystem.

AC400 Global State Object - DWDMModuleState	
Attribute	**Description**
ModuleId	This is a system generated unique module identifier
ModuleState	This attribute specifies the current MSA state of the AC400 module
ModuleVoltage	Module power supply voltage in Volts
ModuleTemp	Module temperature in degree Celsius

VendorName	Vendor identifier for optical module
VendorPartNum	Vendor provided part number of optical module
VendorSerialNum	Vendor provided serial number of optical module
VendorDateCode	Manufactured date code of optical module
ModuleHWVersion	Hardware version of optical module
ModuleActiveFWVersion	Firmware version of firmware loaded in the current/active partition
ModuleStandbyFWVersion	Firmware version for firmware loaded in standby/backup partition
ModuleActiveFWStatus	Status of firmware installed in activer partition
ModuleStandbyFWStatus	Status of firmware installed in standby/backup partition

Network Interface State Object - DWDMModuleNwIntfState	
Attribute	**Description**
ModuleId	This is a system generated unique module identifier
NwIntfId	This is a system generated unique identifier for the network lane
TxChanGridSpacing	Channel grid spacing used for this network interface in GHz
CurrentBER	Current value of BER on this network interface
MinBEROverPMInterval	Min value of BER on this network interface over the last PM interval
AvgBEROverPMInterval	Average value of BER on this network interface over the last PM interval
MaxBEROverPMInterval	Max value of BER on this network interface over the last PM interval
CurrUncorrectableFECBlkCnt	Current uncorrectable FEC block count on this network interface
UncorrectableFECBlkCntOverPMInt	Total uncorrectable FEC block count over the last PM interval
PRBSRxErrCnt	Received PRBS error count on this network interface
RxPower	Receive power value on this network interface in dBm
ChanFrequency	Transmit frequency corresponding to channel number for this network interface

CurrChromDisp	Current RX chromatic dispersion for this network interface
AvgChromDispOverPMInt	Average RX chromatic dispersion for this network interface over the last PM interval
MinChromDispOverPMInt	Min RX chromatic dispersion for this network interface over the last PM interval
MaxChromDispOverPMInt	Max RX chromatic dispersion for this network interface over the last PM interval

Client Interface State Object - DWDMModuleCIntIntfState	
Attribute	**Description**
ModuleId	This is a system generated unique module identifier
CIntIntfId	This is a system generated unique identifier for the client interface
PRBSTxErrCntLane0	Client interface host lane 0 PRBS TX Error count
PRBSTxErrCntLane1	Client interface host lane 1 PRBS TX Error count
PRBSTxErrCntLane2	Client interface host lane 2 PRBS TX Error count
PRBSTxErrCntLane3	Client interface host lane 3 PRBS TX Error count

3.3. ASIC Daemon

The Asic daemon, referred to as Asicd in the FlexSwitch software suite, is the primary process that controls the underlying packet processing asic. The Asic daemon is responsible for exposing configuration/management and monitoring capabilities of the packet processing asic on the Voyager platform. Figure 4 below shows a detailed illustration of the internal of the asic daemon and it's interfaces.

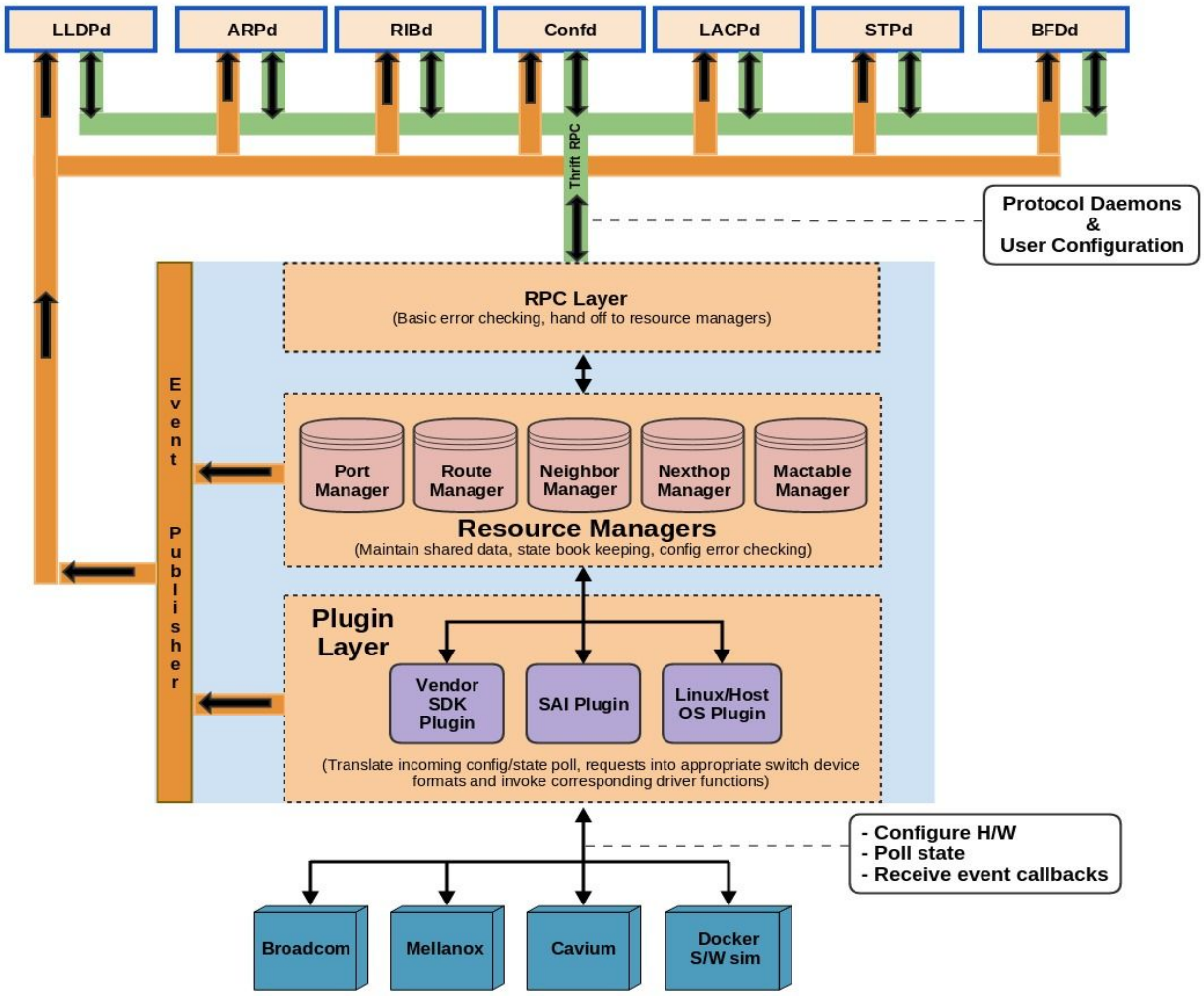


Figure 4 : Diagram illustrating the internal design of the asic daemon and it's interfaces

The sections below detail the configuration and state objects that are exposed by Asicd.

3.3.1 Configuration Objects

The Asic daemon provides the following configuration objects to allow configuration/management of the ethernet packet processors

Port Config Object - Port (R/W attributes)		
Attribute	**Description**	**Possible Values**
IntfRef	Unique interface reference/interface name	fpPort*
IfIndex	System generated unique identifier for this port	Int32 value
Description	Custom string description of this port	Custom string
PhyIntfType	Software discovered physical interface type for this port	SR4/CR4/SR2/CR2/XFI/SFI
AdminState	Enable/Disable state of this port	• UP • DOWN
MacAddr	Mac address associated with this port	
Speed	Speed of this port in Gbps	10/40/100Gbps
Duplex	Duplex type of this port	• HalfDuplex • FullDuplex
Autoneg	Enable/Disable state of autonegotiation on this port	• false • true
Mtu	The maximum tansmission unit size setting for this port	64B - 16KB
BreakOutMode	Break out configuration of this port	• 1x40 • 1x100 • 4x10
LoopbackMode	Loopback mode configuration for this port	• MAC • PHY • RMT
EnableFEC	Enable/Disable state of 802.3bj FEC on this port	• false • true
PRBSTxEnable	Enable/Disable state of PRBS transmission for this port	• false • true

PRBSRxEnable	Enable/Disable state of PRBS reception on this port	• false • true
PRBSPolynomial	PRBS polynomial to use for TX/RX	• 2^7 • 2^23 • 2^31

3.3.2 State Objects

The Asic daemon provides the following state objects that are read only values representing the current state of several parameters in the ethernet packet processor subsystem

Port State Object - PortState (RO attributes)	
Attributes	**Description**
IntfRef	Unique interface reference/interface name
IfIndex	System generated unique identifier for this port
OperState	Current operational state of this port
NumUpEvents	Number of times an operstate transition from DOWN to UP occured
LastUpEventTime	Timestamp corresponding to last operstate DOWN to UP transition
NumDownEvents	Number of times an operstate transition from UP to DOWN occured
LastDownEventTime	Timestamp corresponding to last operstate UP to DOWN transition
Pvid	Port/Default vlan id corresponding to this port
IfInOctets	Ingress octet count for this port
IfInUcastPkts	Ingress unicast packet count for this port
IfInDiscards	Count of packets discarded on ingress for this port
IfInErrors	Count of packets with errors ingressing this port
IfInUnknownProtos	Count of packets ingressing this port with unknown protocol identifiers

IfOutOctets	Egress octet count for this port
IfOutUcastPkts	Egress unicast packet count for this port
IfOutDiscards	Count of packets discarded on egress for this port
IfOutErrors	Count of packets egressing this port with errors
IfEtherUnderSizePktCnt	Count of undersized ethernet frames ingressing/egressing this port
IfEtherOverSizePktCnt	Count of oversized ethernet frames ingressing/egressing this port
IfEtherFragments	Count of fragmented ethernet frames ingressing/egressing this port
IfEtherCRCAlignError	Count of ethernet frames with CRC errors ingressing/egressing this port
IfEtherJabber	Count of jabber frames ingressing/egressing this port
IfEtherPkts	Count of all ethernet packets ingressing/egressing this port
IfEtherMCPkts	Count of multicast packets ingressing/egressing this port
IfEtherBcastPkts	Count of broadcast packets ingressing/egressing this port
IfEtherPkts64OrLessOctets	Count of ethernet packets of size 64 bytes or less ingressing/egressing this port
IfEtherPkts65To127Octets	Count of ethernet packets of size 65 to 127 bytes ingressing/egressing this port
IfEtherPkts128To255Octets	Count of ethernet packets of size 128 to 255 bytes ingressing/egressing this port
IfEtherPkts256To511Octets	Count of ethernet packets of size 256 to 511 bytes ingressing/egressing this port
IfEtherPkts512To1023Octets	Count of ethernet packets of size 512 to 1023 bytes ingressing/egressing this port
IfEtherPkts1024To1518Octets	Count of ethernet packets of size 1024 to 1518 bytes ingressing/egressing this port
ErrDisableReason	Error string displaying reason why port has been disabled
PresentInHW	Attribute indicating if port exists in hardware

ConfigMode	Attribute indicating the configuration mode of this port (L2/L3)
PRBSRxErrorCnt	Error count information from the prbs checker

3.4. Fault and Alarm Management

The FlexSwitch software suite provides facilities to monitor autonomous events that occur in the system. These events are further classified as faults based on whether the events have a detrimental effect in terms of the operational abilities of the system. Faults in turn can be classified as alarms that require an operator to be notified in order to have corrective measures implemented to restore normal device operation.

The Fault Manager daemon is designed to be completely data driven. The control input data for the fault manager is provided via text files that contain JSON formatted data. This JSON data indicates what events are classified as faults, whether a given fault translates into an alarm, what severity level is associated with the alarm etc. Examples of such JSON data are shown in Appendix-A.

The primary building blocks for the fault/alarm management subsystem include

- Event Library
- Redis based PUB/SUB
- Fault Management Daemon (fMgrd)
- Alarm Management Library

The following diagram depicts the relationship between Fault, Performance and alarm managers

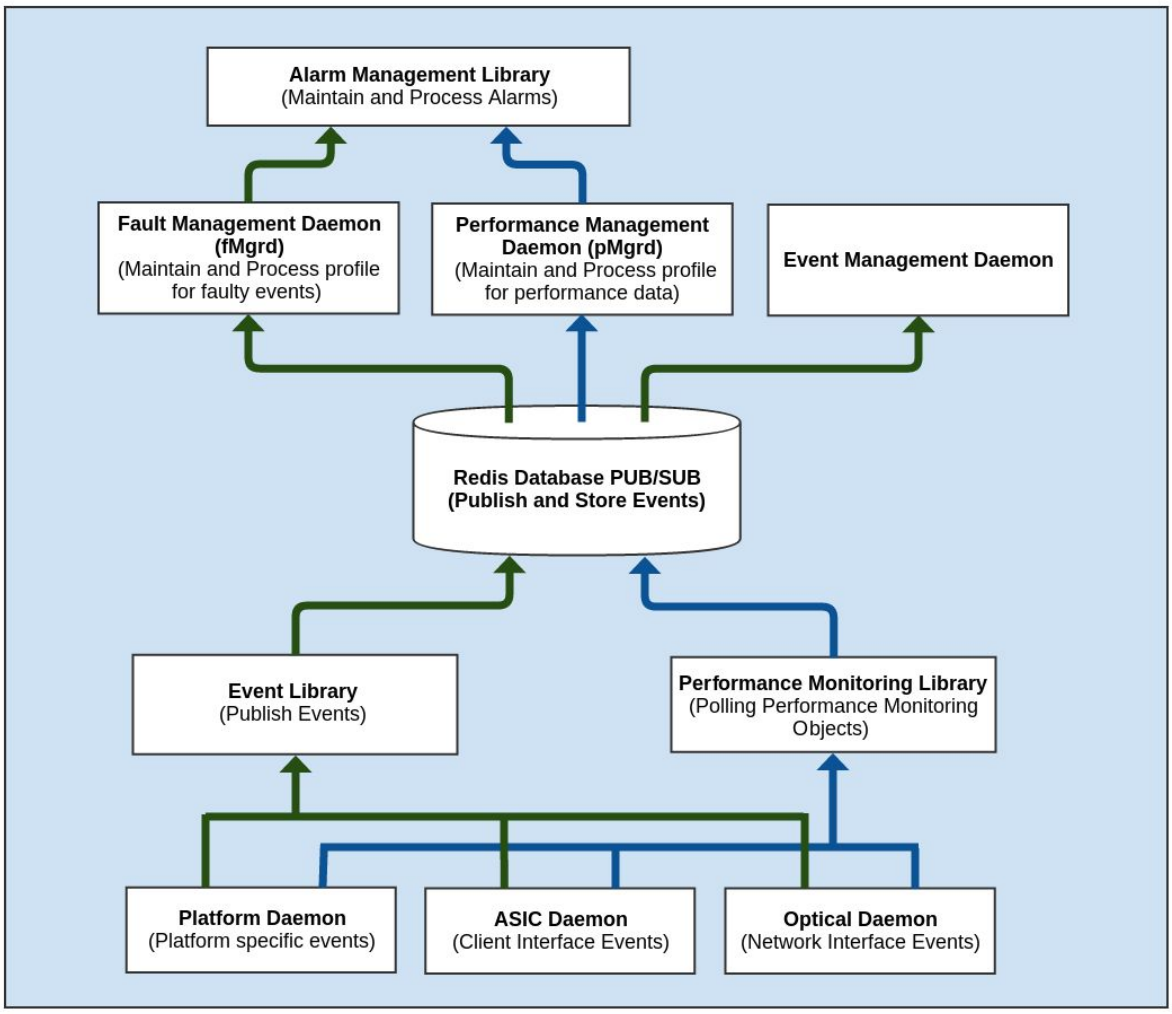


Figure 5 : Diagram illustrating the processing flow for events, faults and alarms

3.4.1 Event Library Software Design

Event library is a generic event generation framework integrated in FlexSwitch which is used by all daemon to publish events. Redis Pub-Sub feature is employed to publish events.

Events are published in the format shown below:

Event Structure	
Parameter Name	**Description**
OwnerId	Unique Id corresponding to the Daemon which is raising the event. Value from events.json
OwnerName	Name of the daemon which is raising the event
EvtId	Unique Id corresponding to the event which is raised. Value from events.json
EventName	Name corresponding event. Value from events.json
TimeStamp	TimeStamp at which event has occurred
Description	Description about the event
SrcObjName	Name of the object for which event is raised
SrcObjKey	Object key value which will give the exact source of event

Shown in Figure 6 is a flowchart depicting the lifecycle of the event publisher within each FlexSwitch daemon

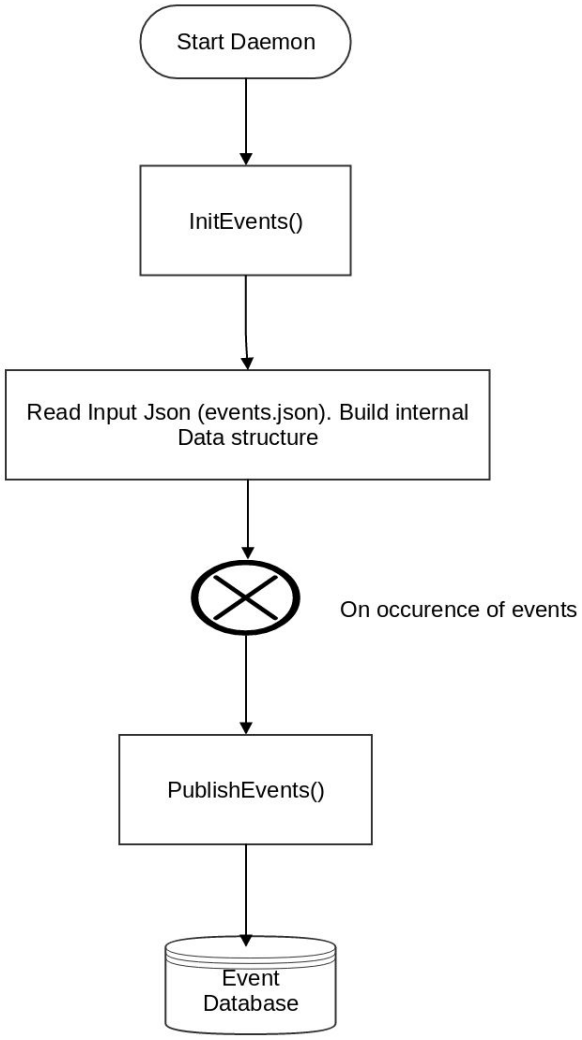


Figure 6 : Steps involved in event processing within a FlexSwitch daemon

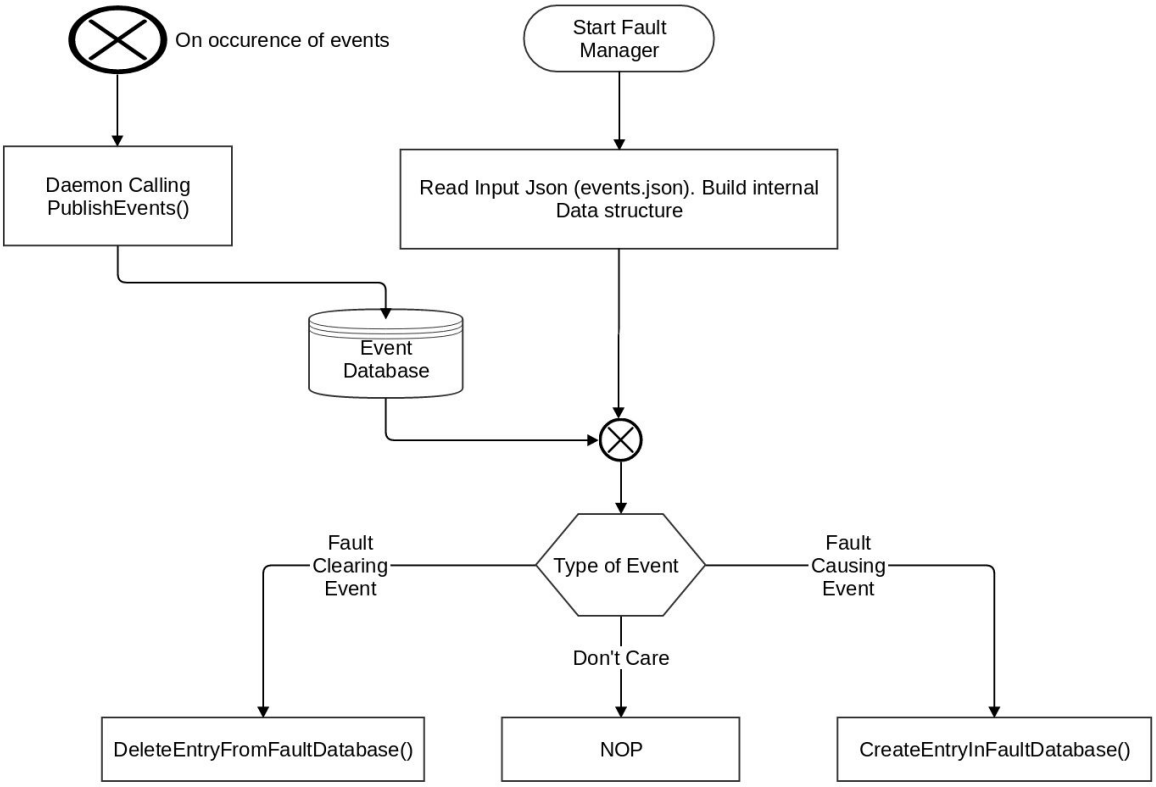
3.5 Fault Management Daemon

The fault management daemon, is responsible for monitoring all events in the system and then classifying the events as faults where applicable. Fault manager subscribes to events published by all daemons. Every event received by Fault Manager can be classified into 3 categories:

1. Faulty Events eg: PortOperStateDown
2. Fault Clearing Events eg: PortOperStateUp
3. Don’t Care Events eg: PortSpeedChange

Based on this classification of event received by Fault Manager following functions:

1. Faulty Events: Create an entry in the fault DB
2. Fault Clearing Events: Delete entry from fault DB
3. Don’t Care Events: NOP



Every entry in the fault database in stored in the given structural format:

FaultDatabaseEntry Structure	
Parameter	**Description**
OwnerId	Unique Id corresponding to the Daemon which is raising the event. Value from events.json
OwnerName	Name of the daemon which is raising the event
EvtId	Unique Id corresponding to the event which is raised. Value from events.json
EventName	Name corresponding event. Value from events.json
Description	Description about the event
SrcObjName	Name of the object for which event is raised
SrcObjKey	Object key value which will give the exact source of event
OccurranceTime	Fault Occurrence Time
ResolutionTime	Time at which Fault is resolved by Fault Clearing event

Fault Manager Config Object:

Fault Config Object	
Parameter	**Description**
EventId	Unique Id corresponding to the event which is raised. Value from events.json
OwnerId	Unique Id corresponding to the Daemon which is raising the event. Value from events.json
FaultEnable	True/False Control for enabling and Disabling Faults

Fault Manager State Object

Fault State Object	
Parameter	**Description**
OwnerId	Unique Id corresponding to the Daemon which is raising the event. Value from events.json
EventId	Unique Id corresponding to the event which is raised. Value from events.json
OwnerName	Fault owner daemon name Value from events.json
EventName	Fault event name Value from events.json
Description	Description about the Faulty event
SrcObjName	Name of the object for which event is raised
SrcObjKey	Object key value which will give the exact source of event
OccurranceTime	Fault Occurrence Time
ResolutionTime	Time at which Fault is resolved by Fault Clearing event

Appendix - A

Example events.json file:

```
{
	"DaemonEvents" : [
		{
			"DaemonID" : 1,
			"DaemonName" : "ASICD",
			"Enable" : true,
			"EventList" : [
				{
					"EventID" : 1,
					"EventName": "PortOperStateUp",
					"Description" : "Port Operational State UP Event",
					"SrcObjName" : "Port",
					"Enable" : true,
					"IsFault" : false
				},
				{
					"EventID" : 2,
					"EventName": "PortOperStateDown",
					"Description" : "Port Operational State DOWN Event",
					"SrcObjName" : "Port",
					"Enable" : true,
					"IsFault" : true,
					"Fault" : {
						"RaiseFault" : true,
						"ClearingEventId" : 1,
						"ClearingOwnerId" : 1,
						“IsAlarm” : true,
						“Alarm” : {
							“Enable” : true,
							“Severity” : “Critical”
						}
					}
				},
				...
				{
					"EventID" : n,
					"EventName": "PortSpeedChange",
					"Description" : "Port SPEED change Event",
					"SrcObjName" : "Port",
					"Enable" : true,
					"IsFault" : false
				}
			]
		},
		{
			"DaemonID" : 2,
			"DaemonName" : "PLATFORMD",
			"Enable" : true,
			"EventList" : [
				{
					"EventID" : 1,
					"EventName": "PSUUp",
					"Description" : "Power Supply Unit State UP Event",
					"SrcObjName" : "PSU",
					"Enable" : true,
```

```
                        "IsFault" : false
                },
                {
                        "EventID" : 2,
                        "EventName": "PSUDown",
                        "Description" : "Power Supply Unit State DOWN Event",
                        "SrcObjName" : "PSU",
                        "Enable" : true,
                        "IsFault" : true,
                        "Fault" : {
                                "RaiseFault" : true,
                                "ClearingEventId" : 2,
                                "ClearingOwnerId" : 1,
                                "IsAlarm" : true,
                                "Alarm" : {
                                        "Enable" : true,
                                        "Severity" : "Critical"
                                }
                        }
                },
                ...
                {
                        "EventID" : n,
                        "EventName": "FanUnitDown",
                        "Description" : "Fan unit state Down event",
                        "SrcObjName" : "FanUnit",
                        "Enable" : true,
                        "IsFault" : true,
                        "Fault" : {
                                "RaiseFault" : true,
                                "ClearingEventId" : n-1,
                                "ClearingOwnerId" : 2,
                                "IsAlarm" : true,
                                "Alarm" : {
                                        "Enable" : true,
                                        "Severity" : "Major"
                                }
                        }
                }
        ]
}
{
        "DaemonID" : 3,
        "DaemonName" : "OPTICD",
        "Enable" : true,
        "EventList" : [
                {
                        "EventID" : 1,
                        "EventName": "NormalSignal",
                        "Description" : "Normal Signal on a network lane",
                        "SrcObjName" : "NetworkLane",
                        "Enable" : true,
                        "IsFault" : false
                },
                {
                        "EventID" : 2,
                        "EventName": "LossOfSignal",
                        "Description" : "Loss of Signal on a network lane",
                        "SrcObjName" : "NetworkLane",
                        "Enable" : true,
                        "IsFault" : true,
```

```
                        "Fault" : {
                                "RaiseFault" : true,
                                "ClearingEventId" : 1,
                                "ClearingOwnerId" : 3,
                                "IsAlarm" : true,
                                "Alarm" : {
                                        "Enable" : true,
                                        "Severity" : "Critical"
                                }
                        }
                },
                ...
                {
                        "EventID" : n-5,
                        "EventName": "NormalVoltage",
                        "Description" : "Normal Voltage value of AC400",
                        "SrcObjName" : "AC400",
                        "Enable" : true,
                        "IsFault" : false
                },
                {
                        "EventID" : n-4,
                        "EventName": "VoltageCrossingMaxMajor",
                        "Description" : "Voltage of AC400 module exceeded
maximum major threshold",
                        "SrcObjName" : "AC400",
                        "Enable" : true,
                        "IsFault" : true,
                        "Fault" : {
                                "RaiseFault" : true,
                                "ClearingEventId" : n-5,
                                "ClearingOwnerId" : 3,
                                "IsAlarm" : true,
                                "Alarm" : {
                                        "Enable" : true,
                                        "Severity" : "Major"
                                }
                        }

                },
                {
                        "EventID" : n-3,
                        "EventName": "VoltageCrossingMaxMinor",
                        "Description" : "Voltage of AC400 module exceeded
maximum minor threshold",
                        "SrcObjName" : "AC400",
                        "Enable" : true,
                        "IsFault" : true,
                        "Fault" : {
                                "RaiseFault" : true,
                                "ClearingEventId" : n-5,
                                "ClearingOwnerId" : 3,
                                "IsAlarm" : true,
                                "Alarm" : {
                                        "Enable" : true,
                                        "Severity" : "Minor"
                                }
                        }

                },
                {
```

```
                                    "EventID" : n-2,
                                    "EventName": "VoltageCrossingMaxMajor",
                                    "Description" : "Voltage of AC400 module below
minimum minor threshold",
                                    "SrcObjName" : "AC400",
                                    "Enable" : true,
                                    "IsFault" : true,
                                    "Fault" : {
                                           "RaiseFault" : true,
                                           "ClearingEventId" : n-5,
                                           "ClearingOwnerId" : 3,
                                           "IsAlarm" : true,
                                           "Alarm" : {
                                                  "Enable" : true,
                                                  "Severity" : "Minor"
                                           }
                                    }

                             },
                             {
                                    "EventID" : n-1,
                                    "EventName": "VoltageCrossingMinMajor",
                                    "Description" : "Voltage of AC400 module is below
minimum major threshold",
                                    "SrcObjName" : "AC400",
                                    "Enable" : true,
                                    "IsFault" : true,
                                    "Fault" : {
                                           "RaiseFault" : true,
                                           "ClearingEventId" : n-5,
                                           "ClearingOwnerId" : 3,
                                           "IsAlarm" : true,
                                           "Alarm" : {
                                                  "Enable" : true,
                                                  "Severity" : "Major"
                                           }
                                    }
                             }
                      ]
               }
        ]
}
```