# Centralized Event Logging and Alerting ER Review Topics

The goal of this document is to coalesce related comments on the Centralized Event Logging and Alerting ER. Many of the comments in the document relate closely to one another, and we can therefore save time by referring to them as a group.

I moved to the “Accepted” state those comments that I felt could be remedied by simply correcting the ER. Such comments are not included here; rather, I have created compiled these in a separate PDF (indexed by author), and my replies should indicate the nature of the change that I made. You’re encouraged to review my replied to your changes, and please let me know if you think that the change does not adequately address your concern.

Comments in this document are first grouped by topic, then by the order in which they appear (page numbers shown are PDF page numbers, not document page numbers). The comments themselves are *italicized*, with my commentary in normal text after all of the comments for a topic.

## What constitutes a “system event”?

### Page 9 - Deepak Brahmavar

Is logging of execution exception conditions like data structure maxing out also permitted?

I think we should include some guidance to what conditions do or do not merit the generation of system events within the requirements section of the documents, and use this to refine the brief definition of a system event here in the introduction.

Conditions that I understand to be system events include:

* Object lifecycle events – a managed object is created or destroyed. What about modifications by the user?
* Object fault state changes – the fault state of an object has transitioned to a new value.

What shouldn’t go to the event log? Are we applying this criteria uniformly today (I believe BVL logs system events far more liberally than DPL, for example)? Is it okay to give the guideline, but let any existing variances stand?

## Comments related to detailed architecture.

### Page 9 – Yanling Qi

John, for the IO VM, we have two instances of IO VMs. Do you consider they are two event sources or one event source? The events for the both IO VMs should be duplicated.

For the NAS events, the scopes of events may be cluster level or node level.

### Page 9 – Yanling Qi

clustered application here implies that one instance of event router per cluster?

### Page 9 – Chris Vandever

Not if the event router is eventd. eventd runs on each node as peers. Are you replacing eventd with a Linux HA app?

### Page 9 – Deepak Brahmavar

Are any steps being taken to avoid split-brain syndrome in event processing?

### Page 10 – Chris Vandever

You're missing some components here, like ASD, SNMPD, EMRS who are the actual recipients of the events and who then generate the email alerts, snmp traps, and support bundle.

### Page 10 – Ray Jantz

*May be a nit, but seeing CIM indications and SNMP traps emitted by the event router suggests that the CIMOM and SNMP agent are "inside" the event router - is that the case?*

### Page 10 – Chris Vandever

So, is this eventd or the common logger or are you combining them into one? Please clarify this for your NAS readers. :)

### Page 12 – Oded Kellner

Did we consider storing the events in a database of soem sort instead of text file? this will make search/sort/filter operations as simple as a query to the events DB

### Page 12 – Yanling Qi

Does this mean

1. we will have the coupler driver in the service VM
2. we will pre-config BVL volume and map the volume to the coupler driver initiator of the service VMs (both service VMs)
3. since the BVL volume has controller-ownership model, we will use AVT for failover?

### Page 13 – Yanling Qi

Should we support event listener registration here? The event consumer adds itself for event notification.

### Page 13 – Yanling Qi

Should we add the communication mechanism between event source and event router and between event consumers and event router?

### Page 14 – Yanling Qi

Will the event router failover to the other service VM in the case of controller failure?

All of these comments are better directed to the detailed architecture, which doesn’t formally exist yet. That said, I’ll try to address these as completely as I can right now.

First, the diagram shown in the overview section depicts the entire event router service (including both log access and alert generation) as a black box; it is a logical view whose purpose is to show the relationships between the event sources, event router, and management clients. The “Reliable Operation” requirements covers the external behavior related to HA scenarios.

If we look inside the event router (the implementation here places the event logger in the SVC VM), things look like this:



I hope we can avoid spending a ton of time on this just yet, but the key details are:

* The event router consists of a client located in the event source, and a server and observers that may be distributed anywhere.
* The event router server is eventd, running on each node under the supervision of the NAS Process Manager (PM). The SNMP and EMRS event observers run on each node under the supervision of PM.
* The event logger and CIMOM/Provider run on a designated node under the supervision of Pacemaker. A cluster IP allows the server to forward events to these observers regardless of which controller is the designated node.

There is a similar functional model for an implementation based on an event logger observer that reuses the BVL logging code.

It should be noted somewhere in the ER or DA that the centralized event logging and alerting architecture implements the legacy interprocess event routing functionality for NAS. Eventd works as it does in the NAS gateway – event sources send only to the eventd running on the same controller, and the notion of local and global scope is preserved for NAS events (and anyone else that wants to use this).

The use of Pacemaker and Corosync, and how we avoid any possible split-brain and cluster amnesia scenarios, will be covered in the Serviceability Infrastructure AAD.

Regarding how we store data in the event log (do we use a database), I think we should first have a very clear picture of the Amelia requirements for event log access. Is the centralized event logging service responsible for executing queries? If so, how general may the queries be?

Whether we use coupling drivers or not to access a “system” BVL volume for log storage depends on whether we choose to use the BVL logging in the IOVM, or an event logger in the SVC VM. The storage infrastructure DA in the Serviceability Infrastructure DA will describe the interfaces and processes for allocating system volumes during out-of-box configuration, and will also cover failover mechanisms.

## External syslog support.

### Page 9 – Aaron Dailey

*Today, I believe we can generate syslog events for external clients. Is this still a requirement?*

### Page 10 – Ray Jantz

I believe we have a requirement to be able to send events to an external syslog server, as this is something we do today. I mention it because I don't see it in the picture.

The closest PR that I could find to this is LSIP200012460, but it doesn't actually specify external syslog support. I asked around about this:

Is it a requirement for the array to be able to send events to an external syslog server in Orion?

I can find no PR for this; the closest I can see is LSIP200012460, but it doesn’t specify anything about external syslog support. Up until now, the alerting mechanisms we’ve been considering are email, EMRS, CIM indications, and SNMP traps.

Chuck replied:

Adding Greg Yarnell…

I wasn’t aware that BVL had that capability, and I don’t know whether it was exposed in Flint.

From my perspective, the only reason we might have this requirement for UIP would be if the functionality were inherited from Flint. However, the UIP OSA architecture is different enough from Flint that we probably wouldn’t have to carry this requirement forward.

Greg replied:

Chuck,

I’m not sure if syslog support has been exposed for Flint. I discussed it with someone from Nesher 6 or more months ago, and at the time is was not planned for Flint (Nesher plans). Razi Inbar would probably know for sure, or could find out.

I replied:

Thanks, Greg. Based on the discussions here, I’m asserting that remote syslog support is a Pnot for Orion, though I’ve added Razi to confirm this.

I never heard back from Razi.

## Fault state retrieval, needs-attention conditions, and Recovery Guru.

### Page 9 – Scott Hubbard

What about failure types for Recovery Guru-like procedures - is this handled as part of the centralized logging or something separate?

### Page 11 – Oded Kellner

Since the events text is for users, it should be in clear language that users can understand. The text should describe the problem, suggest root cause, and list possible corrective actions.

I think that it's the job of the Recovery Guru functionality to suggest root cause and corrective actions. I will add the guideline for user-targeted language, though.

### Page 13 – Eric Stanton

In general having "stateful" fault management has been a requirement for LSI systems. I really don't know how we are going to do this with the NAS subsystem. There are no provisions to support stateful fault behavior. Using the system log to imply stateful behavior is difficult proposition. There must be a careful correlation between "on" and "off" events. If the service vm misses the "off" event, then the condition stays on forever.

### Page 13 – Sridhar Balasubramanian

This section requires some more details on processing events (Combinatorial) that have inter-dependency on VMs. For example, a drive failure (DPL) constituting virtual volume (BVL) and filesystem failure (NAS) must have appropriate linkages to the event posted to central log.

### Page 13 – Chris Vandever

Excuse my ignorance, but what do you mean by "fault data"? I thought we send an event, flag it as "needs attention", <insert appropriate component> gets the event and sees the flag and invokes the appropriate recovery guru. Another event would then be sent to clear the "needs attention". Clearly, I'm missing something here.

### Page 13 – Martin Jess

What is the relationship between events and recovery guru procedures?

Are all CRITICAL events faults or do we have some kind of table mapping events to faults/recovery guru procedures?

### Page 13 – Ray Jantz

So this particular aspect is not centralized? The advantage of having it centralized is that we can more successfully implement the policy that "every guru event has an associated event log event"

### Page 13 – Elizabeth George

Is this (ed: sending an event for recovery from a persistent fault) to clear the previous critical event? Sometimes the recovery action maybe to reboot the controller, so the previous persistent fault maybe gone.

### Page 13 – Chris Vandever

With a few exceptions, we don't detect persistent faults. For example, after the retry count for onlining a volume has been exceeded, we persistently mark the volume offline and forget about the problem. The user has to explicitly online the volume. However, we can't detect that their action was because they had intentionally offlined the volume previously or was required to clear a previous problem.

### Page 13 – Eric Stanton

Are these alerts are redundant with the alerts generated by the log based system? Note that in BVL there are only a handful of persistent faults, but many alertable conditions that come from the logs.

### Page 13 – Scott Hubbard

In the past, we only triggered Needs Attention conditions when a Critical event occurred. However, with the increased number of Event severities, there will need to be investigation as to what severities cause failure types to be generated. It will be more than just critical events.

First, “fault data” is persistent fault or failure state information. Today, DPL (and BVL, I think) provide interfaces by which SANtricity can retrieve a list of non-optimal components. If SANtricity sees that the list is non-NULL, it deems that the array is in a ‘needs-attention’ state. The Recovery Guru function of SANtricity looks at the failed components and their failure types, and suggest recovery actions to the user. If recovery actions are successful, the failure state should go away, and this should be reflected in the failure list. At this point SANtricity can update its needs-attention state.

As with the mechanism for getting events from sources to the provider, we’re faced with a choice as to how we want to “centralize” failure state retrieval. It boils down to where we perform adaptation (from “native” to “centralized” interfaces).

* We can mandate that major subsystems (BVL, DPL, NAS, dom0) implement adapters, so that the aggregator needs to only deal with a single interface. This requires that adapters be developed for each environment.
* We can make the aggregator perform the adaptation, which means that the major subsystems require no modification, and a single team can work on getting the adaptations right.

In either case, it’s possible for a serviceability component (such as centralized event logging) perform the aggregation, or this work could be pushed into the CIM provider.

The approach that the ER currently advocates is that the aggregator performs the adapation (BVL, DPL, and NAS have to do as little as possible), and that the CIM providers are the aggregator (the CIM provider for each subsystem interacts with the failure state interface for that subsystem).

Ray proposes that a serviceability component do this, and this is also a viable option. It could be considered part of the centralized event logging subsystem, but I haven’t seen anything about the coupling between events and failure states that necessitates this. I’ve only seen a constraint that an event accompanies each failure state transition; I haven’t seen anything that says that Amelia would use event messages to determine fault state (I'm using the Alert and Event Notification DA (47557-00) as my source here).

Ray and I considered the approach of centralizing both event logging and failure state via the event logging interfaces through the use of ‘stateful’ events, where each event would have a failure state and failure type entry code. To make this work, the failure state should not be persistent across reboots; we wouldn’t want a failure to magically persist in the event subsystem even though the reboot/restart of another subsystem made the failure disappear (the subsystem wouldn’t necessarily know to send a clearing event in this case). I fear that this would be a very intrusive approach, though, and would slow deployment of any new OSA subsystem we’d ever want to do.

For NAS, doesn't the fact that the volume is offline count as fault state information? At the least, couldn't we look at the state of each volume, and drive Recovery Guru actions for volumes in non-optimal states?

Regarding Ray’s comment on the relationship between the relationship between a failure state transition and the requirement to generate an event with specific severity, should we change the requirement to remove the severity constraints? Would it be useful to have a flag in the event that notes that a failure transition occurred (so that a management client knows to check the failure state interface?)

## Recovery profile listener?

### Page 10 – Ray Jantz

Today there is another event consumer that is not accounted for in this picture. It is the "recovery profile listener." This host-based module listens for all virtual drive "life cycle" events (created, deleted, change of RAID level, change of segment size, etc.) and writes them to a log - this log has proven useful in reconstructing virtual drives that have lost their metadata. OSA needs to somehow account for this, as it is a key part of our serviceability story. Does that fall under your scope or someone else's? One way we could accomplish this is to use the "write events to external syslog server" feature. This would keep the lifecycle in a place other than the array, so they would still be available if the array lost metadata or DACStore.

This is the first time I've heard of this requirement. We can add this to centralized logging, or as an alternative, perhaps this could be done by mining EMRS data at the warehouse?

## Event data model (or lack thereof).

### Page 11 – Chris Vandever

I was told the severity and the 4 booleans following it would be compile time constants, dependent upon which OEM the product is built for. This is certainly more flexible, but extends the dependencies an order of magnitude.

### Page 11 – Martin Jess

Does this mean that the event source has to specify this (ed: severity)?

I think it would be best if the event source did not have to specify this, same for the SNMP trap, email message and EMRS.

I think it should be up to the management client/user to configure what category of events result in what kind of notification. For example, a user may specify that a CRITICAL event should always result in an EMRS data dump, an email and a SNMP trap whereas a WARNING event may only result in a SNMP trap.

### Page 11 – Scott Hubbard

I don't think the product requirement allows for simply tying these fields to severity. They are not specified by the event source, though (that would be making these fields event-specific content and not event type content). The event configuration tables (in BVL, these are the master tables that generate the BVL and MEL event header files, etc.) specify these values per event type.

### Page 11 – Ray Jantz

The information is pulled from the table and used to populate the event messages that are sent?

### Page 12 – Chris Vandever

This needs to be coordinated with the contents of nfx\_eventMsg\_t. There is no reason to have two different structures for the same thing. Therefore, it also needs needAck, status, eventScope, srcAppId, and msg\_to\_gen\_autosupport. This last one is the text that will be logged in elog, regardless of the OEM's setting for the centralized log.

### Page 12 – Wayland Jeong

Will we need to keep both a source local ESN and a global ESN?

### Page 12 – Chris Vandever

If the event sources supply this (ed: ESN) there is no guarantee the event messages will arrive in sequence, nor that there won't be missing sequence numbers as errors occur and messages are dropped. I think the ESN would need to be allocated as part of a single function to post the event, which would mean a fair amount of rework since almost every app has it's own function to post events. Some use the common event\_post() provided by eventd, but others send the message directly to eventd via RMC or SendAgile(). If we allocate it in the "event router", I suspect it's defeating your purpose. Just FYI, in GNS we include an ESN in the event. (See cluster\_allocGnsEventCounter() in code/ssc-cluster/cluster-gns-api.c and vsd\_gnsEventDemonProc() in code/ssc-vsd/vs-daemon.c for how we allocate it, and more importantly gns\_updateEventCount() in code/sm-gns/gns-api.c for how we utilize the counter.)

### Page 12 – Scott Hubbard

How does this line up with the contents that is displayed today in the SANtricity? It seems that there are some missing items such as:

* Description of the event
* Have we abandoned the raw data that came from the DPL MEL log or is that at least still part of the support bundle?
* Event severity
* User (where applicable)

### Page 12 – Sridhar Balasubramanian

It would be good if you can add an illustration depicting these fields.

### Page 12 – Eric Stanton

I wonder if we need a unique event ID here also. Do we not need something like this for the ETT lookup? Each Event Source plugin would need to generate the UID for its own data source. This especially thorny for the NAS data as it has no unique identifier at all.

### Page 12 – Oded Kellner

What is the added value in the logging time? Why do users care?

### Page 12 – Chris Vandever

I’d prefer we not clutter up the global event fields with information that is only pertinent to specific events, even if it applies to multiple types of events, which I think is your position as well. Not to put words in your mouth or anything. J

### Page 12 – Scott Hubbard

Can you explain this (ed: the “variable length component” of the event-specific data) further? Not sure what this really means.

### Page 12 – Naveen BS

do we need a field indicating total length of event content as the event content is of variable-length?

how do we know the end of event content?

### Page 12 – Chris Vandever

Parent object name/ID or ESN? I’m assuming the event source would have to supply this? We have no concept of related events. An event source posts an event and forgets about it. If we add this field, it cannot be required.

### Page 12 – Elizabeth George

What format will these events be displayed on the management client? Is this something new to Orion or will it follow NAS or BVL format?

### Page 13 – Eric Stanton

Specific plugins will need to be written to handle the different record types that are stored within the DB. OR, a unified record format needs to be developed to house most of the queryable data so that it can be treated uniformly.

### Page 20 – Martin Jess

How does this new mechanism fit with the Flint event log mechanism, i.e., the IOVM? How do the Flint events map to these events?

And what about the NAS elog, how does that map into this?

And domain 0 syslog?

This is currently the weakest part of the ER, in my opinion.

First, the terms are confusing, so let’s come up with better ones for the entities we’re working with:

* Event type – an event type contains the information associates with a specific class of event. An event type has the following fields which apply to all system events of that type: severity, visibility, alert flags (SNMP, email, EMRS, CIM?).
* System event – per the definition in the introduction. The fields in Section 3.1.3 describe the content, but one confusing name is the ‘variable length content’. Let’s call this ‘type-specific content’ (though this might get confused with the event type attributes, so maybe there’s a better name?). The type-specific content contains additional data specific to a system event that conforms to a data structure defined by the event type. For this reason, a length field isn’t required.

Some of the biggest tasks are in this area, as evidenced by Elizabeth’s and Martin’s comments, among others:

* What is the data representation for the system event? If we use eventd, it stands to reason that the interface between event sources and the event router would adapt to a variation of nfx\_eventMsg\_t. Does this mean we need to adopt this structure for the system event returned via the log access API? I think that the Amelia team should guide the design of this interface. One factor in the decision may be the desire to have an interface that is similar or identical to an existing one (DPL SYMbol or BVL CORBA), another may be the desire to have an interface that best addresses Amelia’s application needs. As Sridhar points out, we need to depict the event type and system event definitions more clearly and formally (though perhaps this goes into the administrative/configuration interface section?).
* What external behaviors do we need to describe regarding how the canonical ECT/ETT gets transformed into all of the products needed by event sources, the event router, and management clients? The latter item seems more like a DA topic, but are there external aspects to this also?
* Per Eric’s comment, how do we provide query over type-specific data? We can define an interface that supports general ad-hoc queries, though providing this efficiently could make for a complicated DA.

Chris’ comment about minimizing the number of fields in the common part of the system event definition makes sense; we want to minimize the number of fields that might go largely unused.

What user-visible event fields are we missing?

* Event description (per Scott) – isn’t this just the event text for the event type? Or is this something else, like a full description?
* MEL raw data (per Scott) – since this applies to DPL, could this be type-specific data?
* User (where applicable, per Scott) – since it’s “where applicable”, isn’t this type-specific data?
* Message parameters – see the “Event Text Representation” section below.

Oded asked whether we needed to record the time at which the central event logging subsystem logged the event – this is a good question. The logging time information could be useful to support personnel in identifying whether there are problems with event delivery, but maybe this isn't user-visible information. Would management clients ever want to retrieve events by log time? If so, then log time would need to be recorded, and exposed externally.

I need to better describe what a client needs to do to send an event. It only needs to supply the fields required for the system event data (common fields and type-specific); it does not need to set severity, alert flags, etc., as these are fixed by the event type ID for the system event.

The purpose of the ESN is to identify the event in the event log, so that management clients can address system events in the log. The ESN is assigned by the event router and not the client. It may be desirable to allow event sources to supply opaque handles to facilitate asynchronous completion processing, but this is an internal detail. Another possible use for client-specified identifiers would be for filtering dependent events, though we’d need to be able to ensure that clients generate globally unique IDs (Chris’ comment touched upon these mechanisms).

## Localization and customization.

### Page 11 – Yossi Hazoref

This information (ed: event type data such as severity, visibility, alert flags) should be visible for users in simple and clear text.

### Page 11 – Chris Vandever

My understanding was that LSI would be doing this customization. If the OEM does it, then we need a tool to build it and to sanity check it to guarantee they don't do something silly, like generate a "needs attention" for an event for which we have no recovery guru. This is a lot more work for something of limited utility. Speaking of recovery guru, we would need a "needs attention" boolean, also, if we go this route.

### Page 11 – Deepak Brahmavar

Would there be provision to prevent override of certain events?

### Page 11 – Scott Hubbard

Do we really need "Alternate" tables for the OEMs? Shouldn't we just have the flexibility to customize the ECT and AETT during build time.

### Page 11 – Eric Stanton

Just as a note: event text today is housed as an XML document which is the source for the "MEL Spec". For Flint it was expanded to include BVL sourced events too. For Orion it could be expanded again.

### Page 13 – Wayland Jeong

Do we have any requirements regarding how localized text is enabled/configured? Is it okay for this to be compiled or must it be dynamically configurable? Who maintains the localized text? Do we want this dictionary managed by Amelia? Currently with Flint, the event text is a compiled module managed within BVL.

### Page 13 – Ray Jantz

Could there be a provision for a default locale other than English? (I really don't know how locale gets established when you have a browser app. I'm assuming it gets set on the server, so the client doesn't have to know what it is.

### Page 14 – Oded Kellner

This (alert generation via the alert flags per event type) can be the default behaviour, but I would leave to user to configure what events to be send by SNMP, email, etc...

The main requirements are for the architecture to be localization-ready, but there are no P0s for specific localizations.

My preference would be for each user to be able to set their own locale and timezone. There would be a default locale/timezone that are system attributes (chosen by the user during out-of-box config) that are used by other services as well as centralized event logging. There could also be a separate locale for generated alerts if this is deemed useful (I’m inclined to leave this out, though).

From Rebecca:

The requirement is to be localization-ready. We have no plans to do actual translations, but need to use unicode, isolate text, etc.

Here’s the actual text of PR 39727:

Block - Yes File - Yes DAS - Yes All strings must be isolated and UTF-8 in order to accommodate double-byte character sets. Any graphic that has text content would need to be translatable also. Additional comments: - It is not a requirement to accommodate vertical or right-to-left languages; everything can display in left-to-right format. - There is a standing requirement from OEM to be able to display the correct foreign language EULA for the locale, regardless of what language the mgmt software is in. - It is likely that an English language version of Amelia might be running under a foreign-language browser. This is commonly done but should be a test case. - This PR is not a request for translations, only that the product be ready to translate if a partner chooses to do so. (updated based on historical knowledge and conversations with Oded--R. Holt)

Also FYI there is the below entry in the PRD. However it is classified as a P1 rather than a P0 and does not have a corresponding PR and is not part of the plan. Yet the isolation aspect needs to be taken into account in initial design:

CLI localization:

* CLI Commands do not translate. The same "English" command syntax is kept.
* Command output and error messages can be in local language.
* GUI text such as menu items, dialog strings etc… is "localizable".
* All events messages are "localizable"
* On Line Help can be localized
* Icons and splash screens are "localizable"

I need to document the external representation of the ETT and ECT in the next iteration. We need some canonical representation for this data, from which we derive everything that event sources, the event router, and management clients need. If we can adapt the existing “MEL spec” XML that to this purpose, that'd be fine by me.

Scott’s point about overlaying OEM customizations at build time makes sense – this way, every component that needs ECT/ETT information doesn’t need to also understand how to overlay tables at SOD. (though would event sources need to see event type information, other than to have access to the type-specific structure definitions?)

I do think we need to manage ETTs in distributed fashion. Can we partition ECTs and ETTs so that different development groups can easily define the events that pertain to their subsystems, and allow customers to define AECTs/AETTs as overlays to our event definitions? This seems easier than other approaches I’ve seen or heard about (having some authority gatekeep a single event file = huge bottleneck, having a web interface for OEMs to override events = cumbersome interfaces and IT projects we don’t need to undertake). We do need to consider that different groups may use different SCM systems – do we require that event definitions go into a single system, or that we distribute the files to the ‘native’ repositories?

Ideally, each event source wouldn’t need access to full ECT/ETT data – only to the type-specific structure and event text parameter definitions for the events specific to that source’s subsystem.

We could have a needs-attention flag as part of the type-specific data (Greg Yarnell's latest event AAD adds this), but the management clients would only use this as a hint for them to check the failure type entry interface, which is separate from the event log interface.

User-configurable alerting is not a PR, to my knowledge. If it becomes one, then the flag definitions in the AECT need to be configurable at runtime, and the event router needs to expose an interface for configuring these flags (and the GUI needs to allow the user to override any event!)

Disabling the ability to override an event is not a PR, as far as I know.

Yossi’s comment brings up the point that severity strings need to be localizable. Should we add this (retrieval of localized severity strings for values in some form) to the centralized event logging subsystem, or should management clients be themselves be responsible for mapping integers to strings?

## Severity level mappings.

### Page 11 – Scott Hubbard

I realize that this is just providing an exact list from the IETF but we don't plan to use all of these in the Amelia product. Don't we want to state which ones we plan to use??

### Page 11 – John Logan

The Management Events FFD defines six levels, the Alert and Event Notification AAD defines four.

### Page 11 – Oded Kellner

Do we realy need the 8 severity levels? 4 will be enough: Critical,Error,Warning, Info

### Page 11 – Chris Vandever

Chris V: levels, the only level I haven’t seen in our code base is EMERGENCY, but that’s no guarantee it’s not there somewhere.

We don't need them today but I believe that it's a good idea to conform our levels to a well-established industry standard.

Yes. I have updated as follows:

An integer indicating the severity level for events of the type. Valid severity levels conform to IETF RFC 5424, and have the following meanings:

* EMERGENCY (0) – Reserved for future use.
* ALERT (1) – Reserved for future use.
* CRITICAL (2) – An error has occurred that needs to be addressed immediately. Information has been lost or will be lost if the error is not immediately corrected. An example of a CRITICAL event is a volume transitioning to a failed state due to drive failures.
* ERROR (3) – Reserved for future use.
* WARNING (4) – An error has occurred that has degraded the performance or the array's ability to recover from another error. Information has not been lost, but the error should be corrected to prevent information loss if another error occurs. An example of a WARNING event is a single drive failure on a volume configured to RAID 5. The array can operate in this degraded state, but another drive failure on this volume will lead to lost information.
* NOTICE (5) - Reserved for future use.
* INFORMATIONAL (6) – An event has occurred that does not affect the normal operation of the system, but provides useful information to the user of a system. Examples of INFORMATIONAL events include notifications when the background data scrub operation starts and stops, or when the array transitions from degraded to optimal state after the replacement of a failed fan canister.
* DEBUG (7) - An event has occurred that provides information that will help troubleshoot or debug the steps or states that led to an error case. This information should not typically be visible to the user, but should be available to a support or development engineer to help identify and fix problems.

## Filtering events by user or role.

### Page 11 – James Wendel

Will there be any assistance in filtering events that specific users can't see. Lets say we provision some storage where user X can only see a few of the volumes on the system. Will there be a way to know this even can or can't be shown to them. (the bigger question may be if the above will even happen with OSA with NAS).

Does this correspond to the filtering on the ‘user’ field for some events that Scott referred to?

## Event text representation.

### Page 11 – Chris Vandever

This is not compatible with our existing product. Our event text may vary according to event sub-type, for example, EVENT\_VOL\_ONLINE and EVENT\_VOL\_OFFLINE are both NFX\_EVENT\_VOLUME type events, but the text logged is different. Also, we use printf-style text, so we fill in parameters like volume name or ID, etc., which differ for each message. Also, we should note that the OEM supplied text only applies to system event log messages, trace log messages for the same event will be as supplied by LSI. Otherwise, numerous tools for the data warehouse and automated testing will break.

### Page 11 – Chris Vandever

Most of our code doesn't support UTF8, so we'll need to make sure this is well contained. (This has been a well-known and accepted constraint since day 1.)

### Page 11 – Yanling Qi

How does it work? Since an event will have variables such as drive in enclosure x, slot y failed, the event text will contain format information?

### Page 11 – James Wendel

So we only plan on allowing for compile-time variation in event strings?

### Page 12 – Itay Szekely

How do you handle event data that includes variables? E.g., "Added volume X to host Y".

In the DPL the strings are in printf format, and the variables are passed as event arguments. How does it translate to this model, where the string is fixed and the additional arguments are passed as a single variable-length component?

We need to reconcile the various approaches to text. BVL and DPL both use text that's fixed to the event type. BVL embeds printf-style parameters in the text and the management client is responsible for performing the substitutions. With NAS supplying text per event, we've got significant disparity between approaches. Could NAS use parameter substitution or a type-specific field, instead of supplying custom text as a common system event parameter?

Does DPL use parameter substitution in event text, or is it only BVL that does this today? We’ll need to support this capability; BVL has a huge number of event messages that use this.

Which components that need access to the event text cannot handle UTF-8? Do any NAS components (or other event sources) need to access actual event text? If not, then is UTF-8 compatibility really an issue?

Does BVL parameter substitution require that all localizations of a given message follow the same order and number of parameter subsititutions?

## Related-object identification and attributes.

### Page 12 – Chris Vandever

The event source may not always know what object ID Amelia is using.

### Page 12 – Wayland Jeong

Do we want to force all of the event sources to have access to a unique identifier (i.e. chassis serial number of somesuch) to indicate the local controller? Seems like, we can infer the controller at the event aggregator via the IP address. That way we don't force the event sources to find out the source identifier.

### Page 12 – Wayland Jeong

How are object IDs assigned? By source type? By source instance? How do these identifiers get generated?

### Page 12 – Chris Vandever

I assume this is like a 64- or 128-bit value. I don't think we want to bloat the event with unnecessary strings.

### Page 12 – Ray Jantz

Is there any consideration of having object location (e.g., tray/slot) information in the event. I think we have this in MEL.

### Page 12 – Jeremy Birzer

Jeremy Birzer: When describing event specific content it would recommend recording of the serial number of field replaceable unit events.

Example: It would be useful within when a drive fails to know the serial number of that drive within the log. It would also be useful to know the serial number of the device that replaced it.

This may also prove to be useful if parsing for errors relative to a specific device.

### Page 12 – Ray Jantz

Also, the object type seems useful. I think the client would want to be able to show all disk drive events for example.

The source is responsible for specifying related object data. Since there could be different representations of related objects for different event types, I think it makes the most sense for the related object (as well as location, serial number, etc.) to be contained in the type-specific data.

Ray’s comment suggests an alternative – if we want to be able to readily query for ‘all events of a certain type’, we may want to have a common field that is equivalent to a tagged union, with the type as the tag. This has the drawback of bloating the related object ID to the size of the largest possible object name, and uses space even for events where there is no related object.

The CIM provider is responsible for correlating the related object in the system event to a CIM reference.

## Event correlation and dependent-event hiding.

### Page 12 – Yossi Hazoref

The Parent object name\ID - for example the pool name for which the volume name belong to and is the issue of the event. This relation will save time to correlate events that start from an object and propagate to all it's child objects

### Page 12 – Deepak Brahmavar

Are there any plans to combine identical events into a single event ? For example, there may be a single failure condition which is repeatedly reported by the Event Source which could be collapsed into a single event

### Page 13 – Wayland Jeong

Is this one of the "DELL" requirements? Who can we talk to to clarify what it is that DELL wants with this capability?

### Page 13 – Oded Kellner

Filtering is a generic mechanism. Why it needs to know about the specifics of DPL,BVL or NAS?

Here’s what Rebecca wrote regarding dependent event filtering:

You are correct that Dell has requested this more than once, and it makes sense for anyone. However, it sounds non-trivial and not isolated to host software. So our first goal is to not be worse than EOS 10. Though being better would obviously be ideal.

Regarding EOS 10, Leslie is pretty sure this has not been agreed to for Chessie 2, but she thought Rod might know more about what may have been agreed to with Dell.

Here’s what Rod wrote:

This is related to some work that we are doing for Dell in Chessie 2, but I am not aware that anyone is assigned to this directly. We need to make sure we account for the Dell Requirements – this is being done by Patrick Flynn. Copying Patrick. It would be good to have a discussion with him. He may be able to help us here as well.

How do we achieve filtering in Flint today – does it use a generic mechanism, or is it more hard-coded? Can we simply have the event source set a ‘hidden’ flag if the event relates to a dependent object?

Unless I can see examples of what needs to be filtered, I'm not sure if I can give you a good generic mechanism for how that filtering will take place. Certainly we can try an implementation where every event includes an antecedent ESN, and try to ensure that all event sources track antecedent ESNs and incude them in their dependent events. Does it sound easy do do this for every possible case? I don't even know what all the cases are - we need requirements here, and that's why there's an architectural note asking for the scenarios that we need to support.

I’m not sure that including the parent object ID allows for correlation on an event-by-event basis, only on an object-by-object basis. Would it be better to have a field for "parent ESN" that identifies the event that is the antecedent of the new event? The problem here is that it might be hard to properly communicate ESNs assigned by the event router to the event source that generates a dependent event.

There are no plans to coalesce repeated events at this point.

## Controller swap and disk adoption support.

### Page 12 – Wayland Jeong

The requirement as written forces us to persist longs on spinning disk via the IOVM. If we relax and/or change this requirement we could consider persisting logs to flash only which could simplify central logging.

Yes, but then we would need to mirror the flash volumes that store the log messages, or implement database replication, so this complexity must be taken into account.

## Log retention and “pruning”.

### Page 12 – Chris Vandever

Remember that we will collect the log on a daily basis via EMRS. Any fancy retention scheme will make it harder to use the logs for support and debugging, as we won’t know when “less important” entries have been pruned, and that they would likely be available in a log from an earlier date. I’d rather know I have everything that happened between the timestamps of the first and last entries. An entry telling me the log may have lost something (like on controller failure, etc.), which Aaron suggested later, would be very helpful.

### Page 12 – Wayland Jeong

Where did this value come from? Might we consider keeping logs of all severity up to a certain number then keep only high severity logs thereafter to extend our log time coverage? IOW we can keep this value but be smart about which events are persisted.

### Page 12 – Oded Kellner

Where this number came from? Doesn't it make more sense to keep the events for a defined period of time (1 day? 1 week? 1 month?) instead of # of messages?

### Page 12 – Chris Vandever

Are you using a single file or multiple files with logrotate? EMRS expects the log to grow virtually unbounded between the regularly scheduled (daily) data collection to be sent to the DW. It collects and sends the entire log. If you "wrap", events will be lost.

### Page 12 – Sridhar Balasubramanian

Does this (event log capacity) include core dumps?

### Page 12 – Scott Hubbard

Will this be configurable by the user as far as the purge behavior?

### Page 12 – Scott Hubbard

What do you mean by "virtually" trim the event log?

Today, in SANtricity, we do have a Clear Log function available to the user that does, in fact, remove entries from the event log.

### Page 12 – Ray Jantz

(Ed: Virtual “trimming” of event log per user) By being able to select events based on a particular criteria such as time range or object type?

I figured that it would be better to specify capacity at the user-visible level (num of events) instead of the storage level (MB of event capacity), so I used 50MB, and roughly 256 bytes per event. To me, it makes sense to retain as many messages as we can and only get rid of them when we absolutely must.

Core dump storage and event storage are unrelated. Crash dumps probably need to go on a dedicated volume; application core dumps could possibly share the same volume as the event log.

If we implement such a requirement for fancy retention policies, I'd suggest that using a database would make it much easier to implement. Do we really need this though? We should look at the "typical" event generation rate, and the event log capacity, to determine how much history we get, and decide whether that's enough.

The reason I suggested the approach of users “virtually” trimming the event log is so that one user’s actions in this regard do affect another’s. What is the benefit to prematurely purging events from the persistent store? The approach I suggest allows management clients to decide how much of the event log to retrieve and use:

* There can be a global value that limits how much information can be displayed in the event browser.
* There can be per-user limits that get set when a user says "purge the event log". Rather than purging the actual event log, this simply constrains the user's view of the log to events that occur after the purge.

## Event log retrieval interface.

### Page 12 – Eric Stanton

Need clarification here about whether the incoming raw log data is "decoded" before being stored in the DB. My guess is that they are not for space saving and localization issues. The filtering capability though presents an interesting challenge though if text filtering is allowed. Each log record would need to be decoded to its text representation, then the filter applied.

### Page 13 – Ray Jantz

Is the range of ESNs currently occupying the log known to the client? Seems it may not be if the log has wrapped.

### Page 13 – Deepak Brahmavan

No provision to retrieve by event type / severity ?

### Page 13 – Oded Kellner

how is this different from the filetr described in 3.2.4.3? Seems like retriving by time range is just one example of specific filter.

### Page 13 – James Wendel

Will this (ed: CIM RecordLog support) hook in with the CIMOM we will be using for Amelia (SFCB with the LSI providers)? Can you further explain how these hook together?

### Page 13 – Sridhar Balasubramanian

How about filtering based on event time stamp?

### Page 13 – Ray Jantz

The filtering is implemented in the event router, and therefore the CIM provider does not have to do it, right?

Filtering (or ad-hoc query, and which covers searching by type and severity) is implemented in the event router in this ER (the specific component within the event router would be the event logger). I need to combine the range-based search and filtering into a single section, it seems. I had the range retrieval broken out because it specifically provides the foundation for incremental browsing of the event log.

ESNs are 64-bit and would never wrap themselves - they should monotonically increase forever. In an earlier version of this doc, I think I included a bit about the client being able to retrieve the current lowest and highest ESNs; I'll make sure to add this back.

Regarding how we store data in the database, and where “decoding” (joining of event text to actual events, and parameter substitution), that's also more a concern of the detailed architecture, though I think the BVL approach requires the client to perform the "decoding" (look up event text and substitute parameters, in the BVL case). If we did a DB-based implementation, we'd likely perform a JOIN of the ETT and ECT with the selected events for a given request, and then perform the filtering.

The CIM-level interface to the event log is via the RecordLog subprofile. I’m assuming that if there’s some capability that we need that RecordLog doesn’t provide, we would subclass the appropriate classes within the profile and add these capabilities.

One thing I know that we need in order to support RecordLog is the ability to supply the schema for the type-specific component of an event retrieved from the RecordLog.

The Amelia application logic is going to go the CIMOM for everything, right? That is, it would not bypass the CIMOM to go directly to something like the log access API.

## Event timestamps and timezone.

### Page 13 – Scott Hubbard

Is there a requirement for the Amelia Client to provide a way for the user to specify the time zone?

### Page 13 – Chris Vandever

The mgmt client may be in a different TZ than the system. Our elogs use local time. It would be useful for support and engineering to be able to coordinate the two without having to jump through hoops because they're running in a different TZ than the customer.

### Page 13 – Chris Vandever

NAS requires the timezone for file timestamps for CIFS and/or NFS.

My preference is to use GMT for the stored event timestamps, and have clients perform timezone conversions (this is a presentation layer function, IMO). This allows different users to have different views based on their timezone. The stored timestamp should be invariant over changes to the default timezone of the box and daylight time shifts.

I don't know if there's a PR for this, but I think that it's clear that we should allow for a default timezone configuration in both out-of-box config (virtually every computer system includes this as part of installation), and the user should be able to modify this setting via the GUI/CLI.

Allowing users to override the default in their settings (as with locale) is almost certainly a Pnot, though it would be nice to have, and we can certainly allow for it in the architecture.

## Alert handling.

### Page 14 – James Wendel

Does this mean that this centralized logging system will push events out through the already existing providers/cimom? Couldn't this collide with the existing indication system within our providers?

### Page 14 – Chris Vandever

In this case, configuration of what's CRITICAL will need to be coordinated with SNMP and possibly ASD. An OEM cannot flag an event as CRITICAL if SNMPD doesn't support that level for the event.

### Page 14 – Yanling Qi

do we fire snmp trap from one service VM or both service VM?

### Page 14 – Yanling Qi

Do we send email only from the service VM in which the event router is actively running?

### Page 14 – Chris Vandever

We don't currently include all event fields in the email, nor do I believe we should, although we should probably include more than we do.

### Page 14 – Yanling Qi

The EMRS sometime needs to loadup OS specific trace/dump so that the trace/dump needs to be collected on a specific instance of the NAS VM or service VM

### Page 14 – Eric Stanton

We will need to define some boundaries around how often the data dumps are created. Retrieving the data required is a very intensive activity.

Follow-up from Chris regarding SNMPD knowledge of specific event types:

Eventd is aware of what types of events must be sent to SNMPD, and SNMPD must know how to process each event. It gets some information from the event structure itself, but then it will often look up various things in the clusDb to associate names, which are more meaningful to customers, with the numbers and IDs that are stored in the events. Thus, we could do an OEM-specific build so that SNMPD will NOT trigger a trap for a specific event, but we cannot trigger a trap for events we don’t know about.

SNMP and ASD (email and EMRS) run on both controllers; these alerts are locally generated. Logging and the CIM providers that handle indication generation are services that run on a designated controller behind internal cluster IP addresses.

Regarding CIM indications, I thought that event dispatch formed the basis for the indication system. Does Amelia assume some other method for generating CIM Indications? If a component fails or recovers, or if there's a lifecycle event (an object is created or destroyed), don't we use the event to generate the correponding CIM indication?

Chris, how do we ensure that we don’t run amok with event-driven EMRS dumps? I think there is a queuing mechanism for these dumps, so it’s possible that if there are too many requests, they just fall off the back of the queue.

## Delayed alerts.

### Page 14 – Chris Vandever

Just to be clear, a delayed alert is configured **not** to send traps, email, etc., while a different event, which is CRITICAL, is configured **to** send traps, email, etc.? Otherwise, don't we need some sort of flag on the event indicating whether it's delayed (used by the originating component to not send traps, email, etc.) or not (used by the embedded EM server to send the "delayed" traps, email, etc.)? How does this coordinate with OEM configurability?

### Page 14 – Ray Jantz

It seems this (ed: delayed) event needs to be distinguishable from the first one in some way that's obvious to someone browsing the log, e.g., it could retain the original text, but append "condition persisted beyond delay period"

A delayed alert is handled the management client, which has access to failure state information and when components transition in and out of failure state. The management client detects that a failure has occurred and starts a timer. If the condition has not cleared before the timer pops, the management client itself generates the event that generates the alerts. The alert flags, event severity, and event text can be customized in the standard way; if the delay value needs to be customized, this is something that the management client needs to handle.

Agreed that we should be clear about clearly distinguishing the initial (non-alerting) event text from the delayed (alerting) event text.

## Reliable event delivery.

### Page 12 – Ray Jantz

I assume logged events could be buffered in RAM for a period before being written to disk. Are there strategies for recovering these if the controller crashes before they are persisted?

### Page 14 – Chris Vandever

Our apps make a best attempt at delivery of events to eventd. In turn, eventd directly calls all of our existing clients to inform them of the event. This includes elog, asd (for support email), snmpd (for traps), and emrs (which may queue the request for later processing or drop it if the queue is full).

### Page 16 – Chris Vandever

Erm... (embarrassed blush here) while we define a needAck value, waiting and sending the acknowledgement was never implemented in eventd as all callers set the field to FALSE. If this is actually needed, it will mean more work than you were probably anticipating. :(

### Page 16 – Wayland Jeong

Does this imply that the event interface must support a sync and an async mode?

What is a "best attempt" at delivery in the NAS event system? If the system (or a component, such as eventd) fails, will some events be dropped?

We need to work out the details on what the reliability expectations for event logging. If we're trying never, ever to drop an event, we're in for a lot of work regardless. If we can compromise on reliability a little bit, we might be able to speed up implementation quite a bit.

I believe that the BVL logger either has capabilities for protecting events buffered in RAM, or plans to do so. We could have the logger write synchronously to disk, or perform frequent flushes, though this could impact our event logging throughput. For a sqlite-based approach, the Service VM won't have any access to pCache in the near term, and even if so, it would likely be very difficult to make use of it.

## Controller failure, lockdown and event router failure handling.

### Page 14 – Ray Jantz

It would be good if this document could more directly descirbe the HA model for centralized logging. What conditions apply when both controllers are healthy? What conditions apply when only one is up? What happens in the transitions between these two states?

### Page 14 – Aaron Dailey

How would the user know that an event was dropped, if not proactively examining the trace buffers? Can we add an event that says events may have been dropped (I wasn't clear on the meaning of the event listed in 5.1, may be this is already it).

### Page 15 – Wayland Jeong

Does this imply that the event router is restartable? IOW, is this requirement meant to handle the case where the event router or the Service VM as a whole crashes and is restarted? What happens in an OnStor Cougar gateway if eventd crashes? Does the entire controller get rebooted?

### Page 15 – Chuck Nichols

Can you elaborate, at least in the comments, on this (ed: controller reset following extended unavailability or inability to successfully restart a router component)? Are you saying that event sources will drive the controller reset or that some monitor task/daemon will identify that the router has been down for too long and reset? Is a full-blown controller reset necessary? Would it be possible simply to restart the service?

### Page 15 – Dan Sokolov

Service Mode is a condition about the alternate controller, the local controller places the alternate in and out of service mode and maintains connectivity to the management SW.

Service Mode is entered without rebooting, so IFF the local event router can claim control without disruption then entering Service Mode should not cause disruption.

When exiting Service Mode, the affected controller reboots but the survivor (which must have the active event router does not. So there should be no disruption to the current event router.

### Page 15 – Wayland Jeong

What is service mode? Is this a mode of the controller when VMs are not running?

### Page 15 – Elizabeth George

Would it be appropriate to have an additional requirement to specify what happens to events of clients, before central logging facility is available during

1) bootup and 2)SVC VM reboot? 3)setup volume unavailable etc under Reliability?

something like:

logging facility unavailability handling.

### Page 15 – Dan Sokolov

In dual lockdown cases, there will be no event router running, so it cannot respond to access requests or submissions. The management interface isn't even available.

### Page 15 – Greg Yarnell

Event router may not be running if both controllers are locked down.

### Page 15 – Dan Sokolov

The loss of events (ed: in lockdown) should mimic the event router failure case, it cannot be predicted when a lockdown occurs or what will/will not happen to in process events. Maybe this is just confusing me with single and dual lockdowns defined together? Please clarify.

### Page 15 – Dan Sokolov

I don't know what you mean by exiting a lockdown state. I think in terms of external behavior, lockdowns are a dead-end and the controller needs service.

There are concepts of configuration errors that can recover if the error condition is removed, but the current FFDs don't really describe them well. In any case, the controller doesn't really exit the lockdown state in a configuration error, rather it will reboot and try again if the condition may have been removed.

I think we need to do a very critical review of the reliability approach I've proposed here - I'm trying to keep things simple and not overengineer for rare cases, but we need to agree that the reliability described here is sufficient.

Per Aaron’s suggestion, where we can detect conditions in which we can lose events (such as an event router component restart), we should generate an event following recovery that indicates that some events may be lost. I’ll look at adding this.

A NAS node does not restart if eventd (or another component) crashes. According to Jonathan:

PM notifies rmc clients and servers that someone has gone down so they can retransmit messages. But only with a single OS/VM. A pm running in the service vm will not notify apps in another VM as far as the current implementation goes.

Regarding controller reset following extended unavailability of the event router or the inability to restart it, I do need to document who performs this action. My thought was that the applicable HA/monitoring subsystem (such as PM, Pacemaker, monit, etc.) would be responsible for restarting a failed component, and for triggering a controller reset if it detects an unrecoverable failure (that is, if the service cannot be restarted). We could opt not to reset the controller and leave the event router down, but how then do we diagnose this condition?

The plan is to treat SOD as a period of “temporary unavailability” per section 3.5.3, though this is not specifically called out in the ER (but should be). My thought was that the event router client component would buffer and retry events while the event router server was down, so this mechanism would be in effect during SOD and other periods of SVC VM unavailability.

Setup volume unavailability is a different case. There are a few ways we're looking at addressing this. One possibility is to mirror the event log volume to flash on both controllers so we just keep working should the BVL volume become unavailable. Or, we could punt and say that if the log volume is unavailable on both controllers, then event logging stops working (this is what I prefer).

Greg provided the following clarifications on lockdown:

 The event log is not accessible, and event routing ceases to operate, when both controllers in a pair are locked down. In this state, the event router returns error responses to both event log access requests and submissions from event sources.

If both controllers in a pair are locked down, the event router may not be running at all, so it can't return error messages to access requests. This is of course dependent on the lockdown mode. I'm not up to speed on how OSA lockdown will work, but we currently have at least two lockdown modes in DPL. If there are persistent ECC errors, the controller is locked down with an indicator on the 7-segment LED. The network device is not enabled.

We also have a less restrictive lockdown that allows network access (SYMbol), but does not spin up the drives. The intent here is to allow firmware download to fix whatever problem occurred (mismatched firmware version on a simplex array after controller swap, for example).

So there could be scenarios for OSA lockdown where only certain VMs are locked down, and the event router could respond to queries. There could also be scenarios where the lockdown is at a low enough level that the VMs are not running.

Probably the best place to find this information is in Controller Lockdown Mode Reporting, 29087-00. This is an FFD that Patrick Flynn wrote.

Also Controller Module Indicators FFD, 34533-00. This doc has the 7-segment codes and the lockdown reasons.

Dan and Greg, could I work with you offline to iron out the kinks related to these sections?

### Page 19 – Yanling Qi

The centralize logging should not delay the boot process of the service VM if the BVL log volume doesn't exist. We need to boot the controllers without drives.

My preference is that event logging stops working if BVL storage becomes unavailable.

## Simplex support.

### Page 16 – Chris Vandever

This is good information, but last I heard POR is dual-controller only. Should this be an architectural note instead?

Let's confirm this and if so, I'll demote this to an architectural note.

## Administrative and configuration interfaces.

### Page 17 – Chris Vandever

These are already defined in the existing product, although SNMP will need to be extended for block events (for which we will need to receive a list), and EMRS will need to be extended to optionally send to an OEM's DW in addition to LSI's. I am unaware of any additional config changes necessary for email.

### Page 17 – Martin Jess

Is this where we would describe the event viewing UI in Amelia too? (not the actual UX design, but the functionality also described in section 3.2.4.1)

If the configuration interfaces for SNMP and EMRS are documented elsewhere, then maybe I can just xref to the appropriate documents. Otherwise, I was planning to summarize these interfaces here.

Martin, when you say ‘event viewing UI’, what specifically are you referring to? My AAD inexperience is showing here.

## Error handling and event notification.

### Page 18 – Chris Vandever

Perhaps I'm interpreting this too literally, but I wasn't aware that the event router started any components, and thus has no control over restarting them. Do you mean some new event that indicates some component was restarted was sent to the event router? It looks like you're only talking about controllers, not apps or VMs, true?

Yes, it's really the relevant HA/monitoring components that do the monitoring and restarting. The thing I wanted to capture here was that an abnormal condition occurred that resulted in some component of the event router being restarted.

## System event log serviceability.

### Page 19 – Chris Vandever

FYI, for our existing logs the events are formatted into syslog text messages, and the files are gzipped by log rotate. EMRS gzips the current log file, bundles it together with the rotated files into a single gzip file, which it then sends to the DW. No special utilities are necessary. That said, I've heard that MEL events are huge, and thus we might want to log the binary event rather than expanding it into text, which means tools are a requirement.

Exactly - what I'm trying to do here is capture the requirement that we be able to get human-readable logs for support purposes.

## Database / event versioning.

### Page 12 – Jeremy Birzer

The event type does not have a event version or revision #.

When creating parsers that review of the data a working revision # would be very useful.

This makes review of logs via parsing utilities easier to handle changes that might exist between two CFW versions.

### Page 20 – Chris Vandever

The NAS elog is independent.

### Page 20 – Chris Vandever

In this case we may want to add a schema version number to the event structure.

Do we need to handle runtime mismatches of event schema version? Or is it sufficient to perform a migration on the event database as needed following a firmware upgrade?