Open Source Development Labs

Carrier Grade Linux
Requirements Definition

Version 2.0.2

Prepared by the Carrier Grade Linux Specifications Subgroup

Open Source Development Labs, Inc.
127 SW Millikan Way, Suite 400
Beaverton, OR 97005
USA
Phone: +1-503-626-2455
Copyright (c) 2004 by The Open Source Development Labs, Inc. This material may be
distributed only subject to the terms and conditions set forth in the Open Publication
License, v1.0 or later (the latest version is available at
http://www.opencontent.org/openpub/). Distribution of substantively modified versions of
this document is prohibited without the explicit permission of the copyright holder.
Other company, product or service names may be the trademarks of others.
Linux is a Registered Trademark of Linus Torvalds.

Contributors to each section of this Requirements Definition (in alphabetic order) include
the following:

<table>
<thead>
<tr>
<th>General Systems</th>
<th>Clustering</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anderson, Tim (MontaVista)</td>
<td>Anderson, Tim (MontaVista)</td>
<td>Badovinatz, Peter (IBM)</td>
</tr>
<tr>
<td>Aziz, Khalid (HP)</td>
<td>Badovinatz, Peter (IBM)</td>
<td>Campadello,</td>
</tr>
<tr>
<td>* Badovinatz, Peter (IBM)</td>
<td>Cahill, Ben (Intel)</td>
<td>Stefano(Nokia)</td>
</tr>
<tr>
<td>Chacron, Eric (Alcatel)</td>
<td>* Chacron, Eric (Alcatel)</td>
<td>Chen, Terence (Intel)</td>
</tr>
<tr>
<td>Chen, Terence (Intel)</td>
<td>Chen, Terence (Intel)</td>
<td>Cherry, John (OSDL)</td>
</tr>
<tr>
<td>Cherry, John (OSDL)</td>
<td>Cherry, John (OSDL)</td>
<td>Fleischer, Julie (Intel)</td>
</tr>
<tr>
<td>Cress, Andrew (Intel)</td>
<td>Cress, Andrew (Intel)</td>
<td>Haddad, Ibrahim (Ericsson)</td>
</tr>
<tr>
<td>Dague, Sean (IBM)</td>
<td>Dake, Steven (MontaVista)</td>
<td>Hussy, Florence (Alcatel)</td>
</tr>
<tr>
<td>Davis, Steven (Uliticom)</td>
<td>Dimartino, Joe (OSDL)</td>
<td>Jagana, Venkata(IBM)</td>
</tr>
<tr>
<td>Dunlap, Randy (OSDL)</td>
<td>Flaxa, Ralf (SuSE)</td>
<td>Jaskari, Johannes (Nokia)</td>
</tr>
<tr>
<td>Flaxa, Ralf (SuSE)</td>
<td>Haddad, Ibrahim (Ericsson)</td>
<td>Kathail, Pradeep (Cisco)</td>
</tr>
<tr>
<td>Fleischer, Julie (Intel)</td>
<td>Hussy, Florence (Alcatel)</td>
<td>Kierstead, Paul (Alcatel)</td>
</tr>
<tr>
<td>Grana, John J. (Performance</td>
<td>Jaskari, Johannes (Nokia)</td>
<td>Kylänpää , Markku (Nokia)</td>
</tr>
<tr>
<td>Technologies)</td>
<td>Kathail, Pradeep (Cisco)</td>
<td>Lavonius, Ville (Nokia)</td>
</tr>
<tr>
<td>Haddad, Ibrahim (Ericsson)</td>
<td>Kukkonen, Mika (OSDL)</td>
<td>Kukkonen, Mika (OSDL)</td>
</tr>
<tr>
<td>Howell, David (Intel)</td>
<td>Kukkonen, Mika (OSDL)</td>
<td>*Pourzandi,</td>
</tr>
<tr>
<td>Hussy, Florence (Alcatel)</td>
<td>Lavonius, Ville (Nokia)</td>
<td>Makan(Ericsson)</td>
</tr>
<tr>
<td>Jagana, Venkata (IBM)</td>
<td>Marowsky-Brée, Lars (SuSE)</td>
<td>Mäntylä, Janne (Nokia)</td>
</tr>
<tr>
<td>Jaskari, Johannes (Nokia)</td>
<td>McNeil, Daniel (OSDL)</td>
<td>Vinayakray-Jani, Preetida (Nokia)</td>
</tr>
<tr>
<td>Kathail, Pradeep (Cisco)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kukkonen, Mika (OSDL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lavonius, Ville (Nokia)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lynch, Rusty (Intel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mehaffey, John (MontaVista)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yarroll, La Monte H.P (Timesys)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thanks to everyone involved in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the CGL 2.0 sections. Many of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>these people have made editorial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contributions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Indicates lead editors for each requirements section

Comments on the contents of this document should be sent to cgl_discussion@osdl.org .
Preface .............................................................................................................................. 11
References ........................................................................................................................... 12

1 Carrier Grade Linux Context and Scope ................................................................. 13
   1.1 General Systems ............................................................................................... 13
   1.2 Clustering ......................................................................................................... 14
   1.3 Security ............................................................................................................ 15

2 Mainstreaming of CGL Requirements .................................................................... 16

3 Requirements Definitions ....................................................................................... 17
   1.3.1 ID .............................................................................................................. 17
   1.3.2 Name ........................................................................................................ 18
   1.3.3 Priority .................................................................................................... 18
   1.3.4 Category ................................................................................................. 18
   1.3.5 Security Level ......................................................................................... 20
   1.3.6 Maturity .................................................................................................. 20
   1.3.7 Default/Toggle ....................................................................................... 21
   1.3.8 Description ............................................................................................ 22
   1.3.9 POC References .................................................................................... 22

4 General Systems Requirements .............................................................................. 23
   STD.1.0 Linux Standard Base Compliance ........................................................ 23
   STD.2 POSIX Compliance .................................................................................. 24
   STD.2.1 POSIX Core Functionality .................................................................. 25
   STD.2.2 Barriers ............................................................................................... 25
   STD.2.3 Process CPU-Time Clocks .................................................................. 25
   STD.2.4 Clock Selection .................................................................................. 26
   STD.2.5 ISO-C Extensions ............................................................................. 26
   STD.2.6 IPv6 .................................................................................................... 26
   STD.2.7 Monotonic Clock .............................................................................. 26
   STD.2.8 Message Passing ............................................................................... 27
   STD.2.9 Realtime Signals Extension .............................................................. 27
   STD.2.10 Semaphores ..................................................................................... 28
   STD.2.11 Spin Locks ......................................................................................... 28
   STD.2.12 Thread CPU-Time Clocks ................................................................ 28
   STD.2.13 Threads ............................................................................................. 29
   STD.2.14 Timeouts ........................................................................................... 29
   STD.2.15 Timers ............................................................................................... 30
   STD.2.16 Thread Priority Inheritance .............................................................. 30
   STD.2.17 Thread Priority Protection ................................................................. 30
   STD.2.18 Thread Execution Scheduling .............................................................. 31
   STD.2.19 Thread Stack Address Attribute ...................................................... 31
   STD.2.20 Thread Safe Functions ..................................................................... 31
   STD.2.21 Thread Process-Shared Synchronization .......................................... 32
   STD.2.22 Thread Stack Address Size ............................................................... 32
   STD.2.23 XSI Extensions/ISO C Extensions ..................................................... 32
   STD.2.24 Synchronization and Scheduling Support ...................................... 33
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.3.0</td>
<td>SNMP Support Update</td>
</tr>
<tr>
<td>STD.4.0</td>
<td>Stream Control Transport Protocol (SCTP)</td>
</tr>
<tr>
<td>PLT.1.0</td>
<td>Persistent Device Naming</td>
</tr>
<tr>
<td>PLT.2.0</td>
<td>IPMI 1.5 Support</td>
</tr>
<tr>
<td>PLT.3.0</td>
<td>Service Availability Forum Hardware Platform Interface</td>
</tr>
<tr>
<td>AVL.1.0</td>
<td>Robust Mutexes</td>
</tr>
<tr>
<td>AVL.2.0</td>
<td>Software ECC Support</td>
</tr>
<tr>
<td>AVL.3</td>
<td>Software Live Installation and Upgrade</td>
</tr>
<tr>
<td>AVL.3.1</td>
<td>Software Remote Update and Installation</td>
</tr>
<tr>
<td>AVL.3.2</td>
<td>Software Live Upgrade Minimal Reboot</td>
</tr>
<tr>
<td>AVL.3.3</td>
<td>Software Live Upgrade RPM Version Check</td>
</tr>
<tr>
<td>AVL.3.4</td>
<td>Software Live Upgrade Log</td>
</tr>
<tr>
<td>AVL.3.5</td>
<td>Software Live Upgrade per-Node Custom Image and Configuration</td>
</tr>
<tr>
<td>AVL.3.6</td>
<td>Software Live Upgrade Manual Rollback</td>
</tr>
<tr>
<td>AVL.3.7</td>
<td>Software Live Upgrade Automatic Rollback</td>
</tr>
<tr>
<td>AVL.3.8</td>
<td>Software Live Upgrade Application Participation</td>
</tr>
<tr>
<td>AVL.3.9</td>
<td>Software Live Upgrade Fine-Grain Version Checking</td>
</tr>
<tr>
<td>AVL.4.0</td>
<td>Force Unmount</td>
</tr>
<tr>
<td>AVL.5.0</td>
<td>Linux Panic Handler Enhancement</td>
</tr>
<tr>
<td>AVL.6</td>
<td>Memory Overcommit Actions</td>
</tr>
<tr>
<td>AVL.6.1</td>
<td>VM Strict Over-commit</td>
</tr>
<tr>
<td>AVL.6.2</td>
<td>Replaceable OOM Killer</td>
</tr>
<tr>
<td>SVC.1</td>
<td>SNMP Support Update</td>
</tr>
<tr>
<td>SVC.1.1</td>
<td>SNMP Support for IPV4</td>
</tr>
<tr>
<td>SVC.1.2</td>
<td>SNMP Support for IPV6</td>
</tr>
<tr>
<td>SVC.1.3</td>
<td>SNMP Baseline MIBs</td>
</tr>
<tr>
<td>SVC.1.4</td>
<td>SNMP IPv6 MIBs</td>
</tr>
<tr>
<td>SVC.1.5</td>
<td>SNMP IPv6 Kernel Interface - Get</td>
</tr>
<tr>
<td>SVC.1.6</td>
<td>SNMP IPv6 Kernel Interface - Set</td>
</tr>
<tr>
<td>SCL.1.0</td>
<td>Efficient Low-Level Asynchronous Events</td>
</tr>
<tr>
<td>PRF.1.0</td>
<td>Soft Real Time Support Performance</td>
</tr>
<tr>
<td>PRF.2.0</td>
<td>Managing Transient Data</td>
</tr>
<tr>
<td>PLT.4.0</td>
<td>Boot Cycle Detection</td>
</tr>
<tr>
<td>PLT.5.0</td>
<td>Network Console Operation</td>
</tr>
<tr>
<td>AVL.7.0</td>
<td>Fault Isolation Enabling</td>
</tr>
<tr>
<td>AVL.8</td>
<td>Non-intrusive Monitoring of Processes</td>
</tr>
<tr>
<td>AVL.8.1</td>
<td>Process-level Non-intrusive Application Monitor</td>
</tr>
<tr>
<td>AVL.8.2</td>
<td>Kernel-level Non-intrusive Application Monitor</td>
</tr>
<tr>
<td>AVL.9.0</td>
<td>Disk Predictive Analysis</td>
</tr>
<tr>
<td>AVL.10.0</td>
<td>SA Forum HPI Certification</td>
</tr>
<tr>
<td>AVL.10.0</td>
<td>SA Forum HPI Certification</td>
</tr>
<tr>
<td>AVL.12.0</td>
<td>Multi-Path Access to Storage</td>
</tr>
<tr>
<td>AVL.13.0</td>
<td>NFS Client Protection across Server Failures</td>
</tr>
<tr>
<td>SVC.2.0</td>
<td>CIM</td>
</tr>
<tr>
<td>SVC.3</td>
<td>Kernel Dump Enhancements</td>
</tr>
</tbody>
</table>
5 Clustering Requirements ........................................................................................................... 68
  CFH.2.0 Cluster Node Membership with Failure Detection .............................................. 69
  CCM.1 Cluster Communication Service ............................................................................... 70
  CCM.1.1 Cluster Communication Service – Logical Addressing ...................................... 70
  CCM.1.2 Cluster Communication Service – Fault Handling ............................................... 71
  CCM.1.3 Cluster Communication Service – Quality of Service ........................................ 72
  CCM.1.4 Cluster Communication Service – Performance .................................................. 72
  CCM.1.5 Cluster Communication Service – Event Notification ........................................ 73
  CFH.3.0 Prevent Failed Node From Corrupting Shared Resources ..................................... 74
  CFH.4.0 LAN Failure / Node Failure Analysis ..................................................................... 75
  CFH.5.0 Application Fail-over Enabling .............................................................................. 75
  CSM.2.0 Storage Network Replication .................................................................................. 76
  CSM.3.0 Cluster-aware Volume Management for Shared Storage ..................................... 76
  CSM.4.0 Shared Storage Consistent Access ........................................................................... 77
  CSM.5.0 Shared Storage Mirroring ....................................................................................... 77
  CSV.1.0 Cluster-wide Resource Monitor ............................................................................... 78
  CSV.2.0 Cluster-wide Identified Core Dump ......................................................................... 78
  CSV.3.0 Cluster-wide Crash Dump Management ................................................................ 79
  CSV.4.0 Cluster-wide Log Collection .................................................................................... 79
  CCM.2 Cluster Message Service (SA Forum AIS) ................................................................. 80
  CCM.2.1 Cluster Messaging Service ....................................................................................... 80
  CCM.2.2 SA Forum AIS Certification – Messaging Service .................................................. 81

6 Security Requirements .............................................................................................................. 82
7 Requirements Carried Forward Unchanged From Version 1.1 to Version 2.0 .......................... 97
8 Requirements in Version 1.1 Not Included In Version 2.0 ........................................... 98
9 Requirements That Have Been Deferred ................................................................. 98

9.1 Deferred Clustering Requirements ........................................................................ 99
  CSV.dlm  Cluster Resource Locking ................................................................. 99
  CSV.ev   Cluster Event Service ................................................................. 100
  CSM.cfs  Cluster File System ................................................................. 100

9.2 Deferred Fast System Start Requirements ..................................................... 100
  AVL.fast  Fast System Startup ................................................................. 101
  AVL.fast.1 Fast System Boot ................................................................. 102
  AVL.fast.2 Fast SCSI Probe ................................................................. 102
  AVL.fast.3 Fast Linux Static Start ........................................................... 102
  AVL.fast.4 Fast Linux Dynamic Start ..................................................... 103

9.3 Deferred Hardware Requirements ...................................................................... 103
  PLT.atca aTCA Platform Support .......................................................... 104
  PLT.atca.1 Advanced TCA IPMI Support ............................................... 104
  PLT.atca.2 Advanced TCA Block Device Removal .................................... 104
PLT.atca.3    Advanced TCA Shutdown Systemcall Integrated With ATCA System Management ......................................................... 105
CSM.atca    Advanced TCA Multiple Host Synchronized Device Hotswap ................................................................................ 105

9.4 Deferred Diagnostics Requirements .................................................................................................................. 105

SVC.diag.0  Online Diagnostics ............................................................................................................................... 106

Appendices ................................................................................................................................................................. 107

A.1 General References ................................................................................................................................................ 107
A.2 General Systems References .................................................................................................................................. 107
A.3 Clustering References ............................................................................................................................................. 107
A.4 Security References ................................................................................................................................................ 108
A.5 CGL Changes Between 2.0 and 2.0.1 .................................................................................................................... 109
  A.5.1 Preface ............................................................................................................................................................ 109
  A.5.2 Incorrect References in Carrier Grade Linux Requirements Definition 2.0 .......................................................... 110
  A.5.2.1 Requirement “PLT.1.0” – Page 32 of the CGL 2.0 Requirements 110
  A.5.2.2 Requirement “AVL3.1” – Page 35 of the CGL 2.0 Requirements 110
  A.5.2.3 Requirement “AVL.3.2” – Page 36 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.4 Requirement “AVL.3.3” – Page 36 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.5 Requirement “AVL.3.4” – Page 37 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.6 Requirement “AVL3.5” – Page 37 of the CGL 2.0 Requirements 110
  A.5.2.7 Requirement “AVL.3.6” – Page 38 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.8 Requirement “AVL.3.7” – Page 38 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.9 Requirement “AVL3.8” – Page 39 of the CGL 2.0 Requirements 110
  A.5.2.10 Requirement “AVL.4.0” – Page 40 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.11 Requirement “PLT.4.0” – Page 46 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.12 Requirement “PLT.5.0” – Page 46 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.13 Requirement “AVL.8.0” – Page 50 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.14 Requirement “SVC.2.0” – Page 54 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.15 Requirement “SVC.5.1” – Page 56 of the CGL 2.0 Requirements ...................................................................... 110
  A.5.2.16 Requirement “SVC.5.2” – Page 56 of the CGL 2.0 Requirements ...................................................................... 111
  A.5.2.17 Requirement “PRF.3” – Page 57 of the CGL 2.0 Requirements 111
A.5.2.18 Requirement “PRF.3.1” – Page 57 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.19 Requirement “TLS.1.0” – Page 58 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.20 Requirement “TLS.2.1” – Page 59 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.21 Requirement “TLS.2.2” – Page 59 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.22 Requirement “PRF.4.0” – Page 62 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.23 Requirement “PRF.5.0” – Page 62 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.24 Requirement “PRF.8.0” – Page 65 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.25 Requirement “CFH.5.0” – Page 72 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.26 “Section 8, ID 3.1” – Page 95 of the CGL 2.0 Requirements 111
A.5.2.27 Requirement “CSV.dlm” – Page 96 of the CGL 2.0
Requirements ................................................................. 111
A.5.2.28 Requirement “CSV.ev” – Page 96 of the CGL 2.0
Requirements ................................................................. 112
A.5.2.29 Requirement “CSM.cfs” – Page 97 of the CGL 2.0
Requirements ................................................................. 112
A.5.2.30 Requirement “AVL.fast.1” – Page 99 of the CGL 2.0
Requirements ................................................................. 112
A.5.2.31 Requirement “SVC.diags.0” – Page 103 of the CGL 2.0
Requirements ................................................................. 112
A.5.3 Developer Notes for Carrier Grade Linux Requirements Definition 2.0 113
A.5.3.1 Requirement “SVC.1.4” – Page 44 of the CGL 2.0 Requirements 113
SVC.1.4 SNMP IPv6 MIBs ..................................................... 113
SVC.1.4 SNMP IPv6 MIBs ..................................................... 114
A.5.4 Erratum and Their Corrections of Carrier Grade Linux
Requirements Definition 2.0 ........................................... 115
A.5.4.1 Requirement “STD.2.20” – Page 32 of CGL 2.0 Requirements 115
STD.2.20 Thread Safe Functions .......................................... 115
STD.2.20 Thread Safe Functions .......................................... 115
A.5.4.2 Requirement “STD.4.0” – Page 32 of CGL 2.0 Requirements 116
STD.4.0 Stream Control Transport Protocol (SCTP) .............. 116
STD.4.0 Stream Control Transport Protocol (SCTP) .............. 116
A.5.4.3 Requirement “AVL.3.1” – Page 35 of CGL 2.0 Requirements 117
AVL.3.1 Software Remote Update and Installation ................ 117
AVL.3.1 Software Remote Update and Installation ................ 118
A.5.4.4 Requirement “AVL.11.0” – Page 52 of CGL 2.0 Requirements 119
AVL.10.0 SA Forum HPI Certification .................................. 119
AVL.10.0 SA Forum HPI Certification .................................. 120
A.5.4.5  Requirement “AVL.12.0” - Page 52-53 of CGL 2.0
Requirements................................................................. 121
AVL.12.0  Multi-Path Access to Storage................................. 121
AVL.12.0  Multi-Path Access to Storage................................. 121
A.5.4.6  Requirement “SVC.2.0” - Page 54 of CGL 2.0 Requirements 123
SVC.2.0  CIM........................................................................ 123
SVC.2.0  CIM........................................................................ 123
A.5.4.7  Requirement “CCM 1.2” - Page 68 of CGL 2.0 Requirements 124
CCM.1.2  Cluster Communication Service – Fault Handling .......... 124
CCM.1.2  Cluster Communication Service – Fault Handling .......... 125
A.5.4.8  Section 7 - Requirements Carried Forward Unchanged From
Version 1.1 to Version 2.0 – page 94 of CGL 2.0 Requirements ...... 126
A.5.4.9  Section 8 - Requirements in Version 1.1 Not Included In
Version 2.0 – page 95 of CGL 2.0 Requirements ....................... 129

A.6  CGL Changes Between 2.0.1 and 2.0.2................................. 132
A.6.1  Preface ........................................................................ 132
A.6.2  Requirement “STD.1.0”- Page 21 of the CGL 2.0 Requirements
132
A.6.2.1  Original Requirement .................................................. 132
STD.1.0  Linux Standard Base Compliance ............................... 132
A.6.2.2  New Requirement.......................................................... 132
STD.1.0  Linux Standard Base Compliance ............................... 133
A.6.3  Requirement “AVL.3.1”- Page 35 of the CGL 2.0 Requirements 133
A.6.3.1  Original Requirement .................................................. 133
AVL.3.1  Software Remote Update and Installation .................... 133
A.6.3.2  New Requirement.......................................................... 133
AVL.3.1  Software Remote Update and Installation .................... 134
A.6.4  Requirement “AVL.3.2”- Page 36 of the CGL 2.0 Requirements 134
A.6.4.1  Original Requirement .................................................. 134
AVL.3.2  Software Live Upgrade Minimal Reboot ....................... 135
A.6.4.2  New Requirement.......................................................... 135
AVL.3.2  Software Live Upgrade Minimal Reboot ....................... 136
A.6.5  Requirement “AVL.3.3”- Page 36 of the CGL 2.0 Requirements 136
A.6.5.1  Original Requirement .................................................. 136
AVL.3.3  Software Live Upgrade RPM Version Check .................. 137
A.6.5.2  New Requirement.......................................................... 137
AVL.3.3  Software Live Upgrade RPM Version Check .................. 137
A.6.6  Requirement “AVL.3.4”- Page 37 of the CGL 2.0 Requirements 138
A.6.6.1  Original Requirement .................................................. 138
AVL.3.4  Software Live Upgrade Log......................................... 138
A.6.6.2  New Requirement.......................................................... 138
AVL.3.4  Software Live Upgrade Log......................................... 138
Preface

Since the release of Version 1.1 of Open Source Development Lab’s (ODSL) Carrier Grade Linux (CGL) Requirements Definition in October 2002, we’ve extensively re-evaluated and revised each requirement presented. In addition, we’ve developed new requirements in the areas of clustering and security. This version combines these three efforts into a single document.

Modern day PC hardware enables most users to no longer care about the nitty-gritty details of computer hardware, while it still allows those who do care to tinker with bits to achieve an optimal solution. It’s my sincere belief that Linux will lead the same kind of technical revolution on the operating system front. I believe this specification is an important step toward that goal.

I’m often asked, “What is Carrier Grade Linux?” My answer is that it’s an enhancement of vanilla Linux, aimed at the communications network industry. The word "carrier" itself is typical of the compromises the ODSL CGL working group has made to accommodate all team members. Carrier refers to the big telecom companies who provide the communications services. The word "grade" represents the Linux enhancements needed to add important features the communications industry requires. These features are spelled out in this specification.

I’ve often acknowledged version 1 of this document was far from perfect in technical content. After all, we did get it out in six months, which must be a world record or something. When I now compare this document to the previous version, I see we’ve made significant progress in defining the scope while at the same time raising the level of the quality. I think this is simply an amazing feat.

On the other hand, it’s never been our goal to produce a “perfect” specification. To me, the specification is a tool that will enable the ecosystem consisting of Linux distribution vendors, their customers and various hardware and middleware providers to prosper and grow, especially in the current harsh economic situation. However, this document is bound to contain some glitches. If you spot one, please let us know.

To wrap this up, I wish to thank all of the people (too numerous to mention here) who have contributed to this specification. Your input has been deeply appreciated! I also want to thank Peter Badovinatz, our Specifications subgroup chair, for his amazing multi-tasking capabilities, as well as Makan Pourzandi and Eric Chacron, the editors of the Security and Clustering sections respectively, for the selfless sacrifice of their time for the group. Finally I wish to thank OSDL staff and sponsors for enabling this work group to exist.

I sincerely hope that you, the reader of this document, will find it useful. Cheers!

Mika Kukkonen
Roadmap Coordinator, CGL-WG
mika@osdl.org
References

Background information useful to readers of this document can be found in the following places:

Open Source Development Labs (OSDL) home page: http://www.osdl.org

The Carrier Grade Linux web page on the OSDL Web site:
http://www.osdl.org/projects/cgl

The OSDL “Requirements Definition, Version 1.1”:
http://www.osdl.org/projects/cgl/osdl_cgl_requirements_definition_1_1.pdf
1 Carrier Grade Linux Context and Scope

OSDL Carrier Grade Linux (CGL) requirements are organized into three broad classifications: general systems, clustering and security. The scope of these requirements is limited to Linux as an operating environment, and it includes the Linux kernel and libraries, like glibc and libpthreads, which are key to the operation of a Linux-based system.

The scope of CGL also includes a number of utilities and tools needed in a carrier-grade system, such as fault-tolerant installation and updating of a Linux distribution. However it is not our purpose to mandate specific deployment processes, but rather to allow distributors to create and maintain their own distribution mechanisms.

The clustering requirements in this document are aimed at supporting clustered applications in a carrier-grade environment as an effective way to achieve highly-available services inside a network element. We specifically have not addressed the use of clustering to maximize performance, such as in the Beowulf and Mosix solutions.

The security requirements are aimed at maintaining a certain level of security while not endangering our goals of high-availability and performance. With this in mind, a four-tier security classification has been created so users of this document can choose a security level with which they’re comfortable (in terms of tradeoffs between performance and security).

1.1 General Systems

The general systems requirements apply to the Linux kernel, core libraries and essential tools that are needed in a carrier-grade system. This section contains a wide range of requirements ranging from user-space tools, like a kernel debugger, to hardware support requirements, like an Intelligent Platform Management Interface (IPMI).

Another important part of the general systems requirements is the linkage to other specifications, like POSIX, Linux Systems Base (LSB) and Service Availability (SA) Forum HPI. General systems requirements define the priority and level of support required for these specifications, and we’ve taken special care not to create requirements that conflict with those specifications.

Several key requirements address deployment support, such as installing and updating software and providing complete support for SNMP for IPv6 installations. These requirements have limited impact on the Linux kernel, but they have broad impact on the overall carrier-grade environment. Fault management is also addressed in the general systems requirements.

Several requirements enhance the Linux kernel to better support carrier-grade applications. Some enhance controls for managing and monitoring applications, while others allow applications to access functionality that normally requires super-user privileges, such as pinning memory pages.
1.2 Clustering

Network service providers must continually update their network infrastructure to keep up with customer demands for high quality services that take advantage of ever-increasing Internet bandwidth. Providers prefer a solution that’s scalable and cost effective, where scalability means adding more computation and networking bandwidth to existing devices.

Clustering multiple computing resources addresses system scalability by allowing new nodes to be added to an existing cluster to meet performance and scalability demands. Network equipment must be capable of providing 5 to 6 nines availability and should be able to withstand hardware, software, and firmware failures on individual compute nodes. Hardware redundancy of components, such as fans, power supplies, network connections, and RAID storage, represents a subset of failures that would otherwise impact the availability of a service. Clustering addresses redundancy at a higher level and contributes to service availability: the goal is to eliminate any single point of failure in the cluster.

Clustering also implies distribution of computing resources. Clustering requirements for CGL are defined in terms of availability, maintenance, and performance. The main objectives of clustering architectures are as follows:

- **Improved product performance.** Clustering provides a framework that allows multiple computing resources to provide services.
- **Increased product availability.** Clustering increases service availability by preventing an instance of the operating system from being a single point of failure.
- **Enhanced product maintenance.** Clustering provides a framework for maintenance tasks like node replacement and version upgrades (both hardware and software), enabling a cluster of individual systems to be managed as if it were a single system.
- **More cost efficient solutions.** Clustering enables cost efficiency due to the sharing and pooling of resources within a cluster.

Other objectives for clustering include the reusability of standard interfaces, such as those defined by the Open Cluster Foundation (OCF) and the Service Availability Forum (SA Forum). While it’s likely CGL will leverage the SA Forum Application Interface Specification (SA Forum AIS) and OCF APIs, requirements in this document are written to be read without referring to these SA Forum or OCF standards.

CGL clustering requirements are defined to be independent of any clustering model. However, clustering models deployed by companies in the communications market were the inspiration for the clustering requirements. See A.3 Clustering References for more details.

The ODSL white paper “Carrier Grade Linux Clustering Model Version 1.0” (http://www.osdl.org/docs/cgl_clustering_model_whitepaper.doc) was the primary reference for the clustering requirements. The whitepaper defines availability and covers topics like early fault detection, failure confinement, fault localization, and failure notification. It provides physical model descriptions, like descriptions of node types, boot models,
redundant local networks, storage replication, shared storage, and shared file systems. It also provides logical model descriptions, like descriptions of active/standby, active/active, hotstandby/coldstandby, distributed data access, communication services and maintenance.

### 1.3 Security

Security mechanisms are required for CGL systems to help protect confidential data and to help ensure high availability and reliability by reducing downtime caused by security issues. Demands for more secure systems are driven by a diversity of clients ranging from large corporations to governmental organizations to small e-businesses. The need for security has increased as telecom servers move from closed systems to IP-based systems connected to the Internet, thereby increasing the risk of security problems due to the more open environment of the Internet.

Security requirements must take into account that an external network connection is a more likely approach for an attack against a system, since systems are typically in physically secure locations. However, an attack from an internal intranet is also a possibility. So security requirements need to be created to protect the system against attacks from both the Internet and intranets.

Security requirements must also take into account that no system is 100% secure, so systems should be designed in a way that minimizes potential damage. An attacker can perpetrate systematic, deliberate and repetitive attacks against a system. Defending against these systematic attacks requires a different approach than that for defending against undesirable events like hardware faults, which occur unpredictably. For example, redundancy can be used to make a system less vulnerable to unpredictable events, but it doesn’t reduce vulnerability to a systematic attack. The attacker simply needs to repeat the attack to penetrate the system.

Particular constraints related to security for carrier-grade Linux systems include the following:

- **Minimized performance degradation** – Many existing solutions have been developed for the military and similar organizations where security is a top priority and long response-times are acceptable. These solutions are often not acceptable for use by the communications industry, which has established demanding requirements for response-time and system availability. Any security solution that can’t satisfy these requirements won’t be accepted by the industry.
  - The introduction of security features must not impose significant performance penalties. Performance can be expected to degrade slightly due to initial deployment of a security mechanism; however, the impact of any subsequent accesses must be negligible. The impact on connections that aren’t secure must be null or insignificant (less than 0.5%).

- **Availability of Quality of Protection (QoP) options** – Some clients will pay for high quality protection due to the type of transactions their systems handle, yet many won’t want to pay extra for added security. Therefore, a range of
configurable security services must be available, and the system must be capable of simultaneously providing different clients with different qualities of protection, without compromising the overall security of the server and other clients.

Rather than specify all security requirements for a carrier-grade Linux system, the intent of this document is to define necessary add-ons to commercially-available Linux distributions in order to bring them to the carrier-grade level of security. For example, firewalls aren’t mentioned in this document, because valid firewall mechanisms already exist for Linux.

This document covers security requirements for node-level systems. It does not cover the specific requirements related to cluster computing.

2 Mainstreaming of CGL Requirements

This document is meant to identify requirements rather than specify solutions. This is done with an understanding of the nature of Linux and the cooperative, open source development model. In a proprietary development environment, requirements are used to guide ongoing operating systems releases. In a cooperative, open source world, Linux advances when developers address items of interest to them in ways that complement the overall system.

The CGL working group has constrained the requirements to those for which at least one valid open source project or patch exists. We believe this approach maximizes our chances of seeing these requirements become a part of a Linux distribution and accepted into mainstream Linux. Carrier-class servers, oriented to high availability, security, remote access and low-latency performance, have much in common with other enterprise uses of Linux. Taking advantage of this commonality, the CGL working group is striving to identify potential solutions based on existing open source projects.

Carrier Grade Linux is not intended to be a distribution by itself, but rather a part of a commercial Linux distribution. However, carrier-class systems may be configured in specific ways, which is true of any sensitive server, be it a Web server, banking database or VoIP (Voice over IP) network element.

To avoid reinventing the wheel, while at the same time not locking down implementations by defining a strict set of API's, the CGL working group has tried to identify and encourage existing solutions that are consistent with our requirements. For example, two competing projects, Next Generation POSIX Threads (NGPT) and Native POSIX Threads Library (NPTL), are addressing the issue of POSIX threads support. For us it does not matter which solution predominates, as long as the requirement for POSIX thread support is met.

In areas less well-defined than POSIX threads, we hope to see projects converge into a de facto standard that will serve as a basis for CGL-compliant implementations. We believe this is the best way for solutions to be accepted into mainstream Linux.


### 3 Requirements Definitions

The requirements are split into three sections: general systems, clustering, and security requirements. The requirements in all three sections apply to the carrier-grade Linux operating system environment.

Within each section, the requirements are ordered by priority, with the highest priority items listed first. This aids in the assessment of system impacts and benefits. Lower priority requirements can then be considered in the context of higher priority requirements.

Seven header fields and two text fields are used to describe each requirement. These fields are listed below and they’re described in the following sections:

**Header fields**
- ID
- Name
- Priority
- Category
- Security Level
- Maturity
- Default/Toggle

**Text fields**
- Description
- Proof of Concept (POC) References

#### 1.3.1 ID

The ID is a unique set of letters and numbers associated with a specific requirement. The three-letter prefix of the ID identifies the category to which a requirement belongs, as shown in the following list:

<table>
<thead>
<tr>
<th>ID Prefix</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.x.x</td>
<td>Standards</td>
</tr>
<tr>
<td>PLT.x.x</td>
<td>Platform</td>
</tr>
<tr>
<td>AVL.x.x</td>
<td>Availability</td>
</tr>
<tr>
<td>SVC.x.x</td>
<td>Serviceability</td>
</tr>
<tr>
<td>SCL.x.x</td>
<td>Scalability</td>
</tr>
<tr>
<td>PRF.x.x</td>
<td>Performance</td>
</tr>
<tr>
<td>TLS.x.x</td>
<td>Tools</td>
</tr>
<tr>
<td>CFH.x.x</td>
<td>Cluster – Fault Handling</td>
</tr>
<tr>
<td>CCM.x.x</td>
<td>Cluster – Communication and Messaging</td>
</tr>
<tr>
<td>CSM.x.x</td>
<td>Cluster – Storage Management</td>
</tr>
<tr>
<td>CSV.x.x</td>
<td>Cluster – Services</td>
</tr>
<tr>
<td>AUT.x.x</td>
<td>Security – Authentication</td>
</tr>
<tr>
<td>ACC.x.x</td>
<td>Security – Access Control</td>
</tr>
<tr>
<td>CON.x.x</td>
<td>Security – Confidentiality and Integrity</td>
</tr>
<tr>
<td>AUD.x.x</td>
<td>Security – Auditing</td>
</tr>
</tbody>
</table>
SMN.x.x  Security – Security Management

The second digit of the ID identifies either a stand-alone requirement or a requirement summary for a set of related sub-requirements. The third digit is “0” for a stand-alone requirement. The third digit is not present for a requirement summary. The third digit is a counter that uniquely identifies each sub-requirement under a requirement summary.

1.3.2 Name

The name of the requirement indicates the purpose of the requirement.

1.3.3 Priority

Priority means the importance of the requirement for a carrier-grade system. The priority is designated as shown below:

1  Mandatory in a carrier-grade system compliant with this specification.
2  Desirable in a carrier-grade system, but not vital.
3  Not currently considered important, but in future may become so.

For easy reference, when a requirement from the OSDL “Requirements Specification, Version 1.1” has been included, the v1.1 priority is shown in parentheses below the current priority. Several v1.1 requirements have been upgraded from priority 2 to priority 1 or from priority 3 to priority 2. We expect priorities to continue to change in future releases of this document.

1.3.4 Category

Each requirement is assigned to a category. Although most requirements will fit into several categories, the assigned category is intended to indicate the core nature of the requirement. Categories include:

- **Standards** A requirement that references specifications controlled outside the CGL workgroup that are important to carrier server systems.
- **Platform** A requirement that supports interactions with the hardware platforms making up carrier server systems.
- **Availability** A requirement that supports greater availability of carrier server systems, such as improving the robustness of software components or support for recovery from a hardware or software failure.
- **Serviceability** A requirement that supports servicing and managing hardware and software on carrier server systems.
- **Scalability** A requirement that supports vertical and horizontal scaling of carrier server systems such that the addition of hardware resources results in acceptable increases in capacity.
- **Performance** A requirement that supports performance levels necessary for the environments expected to be encountered by carrier server systems.
Tools
A requirement that supports ancillary capabilities not directly involved in normal execution of carrier server systems, such as debuggers used to develop modules, drivers or applications.

Clustering
A requirement that supports the use of multiple carrier server systems to provide higher levels of service availability through redundant resources and recovery capabilities and to provide a horizontally-scaled environment supporting increased throughput.

Clustering requirements are further categorized according to the following mechanisms:

- **Cluster Fault Handling (CFH)**: The detection of faults/failures in clustered systems and the response to those faults/failures.

- **Cluster Communication and Messaging (CCM)**: Low-level communication between cluster nodes and higher-level cluster messaging mechanisms.

- **Cluster Storage Management (CSM)**: Cluster-wide management of shared or replicated storage.

- **Cluster Services (CSV)**: Cluster-wide services like global resource locking, event services, clustered file systems, cluster logging and debugging tools.

Security
A security requirement. Security requirements are further categorized according to the following mechanisms:

- **Authentication (AUT)**: A mechanism that helps ensure that users are the persons they claim to be.

- **Access control (ACC)**: A mechanism that helps to ensure that the information is not accessed, destroyed, or modified by an unauthorized person.

- **Confidentiality and integrity (CON)**: A confidentiality mechanism ensures the information is not read by an unauthorized person. An integrity mechanism ensures information is not altered by an unauthorized person in a way not detectable by authorized users.

- **Auditing (AUD)**: A mechanism that detect attacks and aids in forensic analysis of attacks.

- **Security management (SMN)**: A mechanism that provides a fast, reliable and efficient means of detecting an attack and reacting to it.
1.3.5 Security Level

In CGL, it must be possible select and enable a level of security. Four levels of security have been defined as described below. The requirement for each level includes the requirements for the levels numerically below it. For example, a system at security level 3 means that all the requirements have been implemented for security levels 1, 2, and 3, and they are enabled.

Security levels include:

1. **Default**: No additional security mechanisms are added to the defaults provided by major Linux distributions.

2. **Perimeter Defense**: Basic perimeter defenses are provided, like firewall functionality. Operations are logged and limited alarm functionality is provided. For any system that resides on a network not 100% trusted, this level should be considered the absolute minimum.

3. **Secured Communication, System Integrity Protection**: No addition to perimeter defense is provided, but measures to prevent attacks on management protocols are introduced. Measures are introduced to prevent abuse of the system by insiders or outsiders.

4. **Intrusion Mitigation, Insider Controls**: At this level, extensive measures are introduced to control insider access and to mitigate the effects of a successful intrusion.

The security level for communications between network elements is determined by the element with the lowest level of security. For example, if a system contains elements at security level three and elements at security level 2, the overall system security level will be at level 2.

1.3.6 Maturity

In OSDL Carrier Grade Linux Requirements Specification, Version 1.1, priority was used to indicate both the desirability of the functionality and the maturity of existing Open Source solutions. In this version of the document we have created a separate field for maturity to avoid conflicts between these two attributes.

Values for maturity are:

- **None**: The CGL working group is not aware of an existing Open Source solution.

- **Started**: A project attempting to address this requirement exists, but it is too new or uncertain in direction to be considered a suitable candidate.

- **Experimental**: One or more projects exist and are viable candidates but more work is needed to fully satisfy the requirement.

- **Production**: A project fully satisfies the requirement and is ready for deployment.
The maturity values assigned to specific requirements are somewhat subjective. The projects considered by the CGL working group when the Maturity value was assigned are described in the POC References section below.

In certain deployment scenarios, a less mature solution may be quite adequate. While we are trying to provide information and incentive in support of these projects, distributors, vendors and customers must decide which solution is appropriate for their needs.

### 1.3.7 Default/Toggle

Every priority 1 requirement is mandatory in that the required functionality is expected to be present in any CGL system. However, it is also expected that across all possible deployments of CGL systems, tradeoffs will need to be made among the available functions, acceptable levels of system complexity, security needs, and customer preferences. To accommodate these tradeoffs, each requirement specifies a default setting and toggle control. The values in the Default/Toggle field apply to the functional implementation that fulfills the requirement.

The Default setting indicates whether the functional implementation is expected to be functionally enabled when a CGL system is created in a default configuration. Values are:

- **On**: The function must be functionally enabled, or “turned on”, by default.
- **Off**: The function must be functionally disabled by default but must still be available to be enabled.

Toggle Control indicates whether a function must be implemented in such a way that it can be toggled between the enabled and disabled states without rebuilding. Rebuild in this context means (re)compiling source code. It is strongly encouraged that toggling the state should be online and should not require rebooting the kernel or restarting an application. Values are:

- **Yes**: The function must support the capability of being toggled.
- **No**: The function is not required to support the capability of being toggled.

A requirement with Default set to "On" and Toggle set to "No" is valid for all deployments. In this case, a Toggle setting of "No" does not disallow such control, but should be seen as indicating that a function that will always be present and enabled.

CGL does not specify how toggle control is to be implemented other than that it should be realizable without a rebuild of the system and, if possible, without rebooting the kernel or restarting an application. It is assumed that the toggle control will typically consist of normal system commands applied against modules, drivers, daemons and other embodiments of the functional implementations of the requirements.

Toggle control may require disruption of highly-available services on a node in a clustered environment if those services can be continued on other systems within the cluster.
**1.3.8 Description**

The description section contains the description of the requirement.

**1.3.9 POC References**

The Proof of concept (POC) references listed for each requirement specify Open Source projects considered by the CGL working group to embody work toward implementation of the requirement. However, such an implementation doesn’t necessarily mean the requirement has been met. The list of references provided is not meant to be exclusive or complete. We are very open to being directed to new and active projects in these areas.
4 General Systems Requirements

This section contains requirements that apply to the Linux kernel, core libraries and tools essential to a carrier-grade system. Requirements related to clustering and security can be found in the following sections.

Requirements are generally arranged according to priority. However, if a requirement summary is followed by sub-requirements with different priorities, the requirement summary and sub-requirements are placed at the priority level of the highest priority sub-requirement.

**Priority 1 Requirements**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.1.0 (1.1.0 in v1.1)</td>
<td>Linux Standard Base Compliance</td>
<td>1 (P1 in v.1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a system supporting CGL requirements shall remain compliant with the Linux Standard Base (LSB) 1.3 or later. Some of the features outside the current LSB definition may become part of subsequent LSB definitions.

**POC References**

Linux Standard Base: http://www.linuxbase.org/
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2</td>
<td>POSIX Compliance</td>
<td>Mixed</td>
<td>Standards</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide POSIX-compatible interfaces to support direct porting of common carrier-grade applications. CGL shall follow the IEEE POSIX standards as described in:

- IEEE Std 1003.1-2001. Technical Corrigendum No. 1

Note: The first three standards above have been combined into IEEE Std 1003.1, 2003 Edition (or Single UNIX Specification, Version 3).

All references to “the POSIX specification” in the following sub-requirements refer to these standards.

The POSIX specification uses margin codes to identify functional areas. These margin codes are referenced in the sub-requirements below. POSIX functionality required by CGL includes the following:

- Core POSIX functionality
- POSIX Timers functionality
- POSIX Signals functionality
- POSIX Threads functionality
- POSIX Message Queues functionality
- POSIX Semaphores functionality

**POC References**

For specific implementations, see the POC References in the sub-requirements below.

As a general reference eventually covering POSIX support as a whole, refer to the Open POSIX Text Suite for more POSIX conformance data: [http://posixtest.sf.net](http://posixtest.sf.net)
### STD.2.1 POSIX Core Functionality

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.1</td>
<td>POSIX Core Functionality</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide all items defined in the POSIX specification that have no margin code are considered core POSIX functionality and shall be implemented by CGL.

**POC References**
The latest stable version of the Linux kernel.

### STD.2.2 Barriers

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.2</td>
<td>Barriers</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the BAR (barriers) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**
See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)
See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

### STD.2.3 Process CPU-Time Clocks

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.3</td>
<td>Process CPU-Time Clocks</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Started</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the CPT (process CPU-time clocks) margin code in the POSIX specification. This functionality is used in POSIX Timers.

**POC References**
High resolution-timers project: [http://high-res-timers.sf.net](http://high-res-timers.sf.net) – not in stable kernel yet.
### STD.2.4 Clock Selection

**Description**: OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the CS (clock selection) margin code in the POSIX specification. This functionality is used in POSIX Timers.

**POC References**

Linux kernel greater than 2.5.63

High resolution-timers project: [http://high-res-timers.sf.net](http://high-res-timers.sf.net) – for user space.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.4</td>
<td>Clock Selection</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

### STD.2.5 ISO-C Extensions

**Description**: OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the CX (ISO C extensions) margin code in the POSIX specification. This functionality is part of the core POSIX functionality.

**POC References**

The latest stable version of the Linux kernel.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.5</td>
<td>ISO-C Extensions</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

### STD.2.6 IPv6

**Description**: OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the IP6 (IPv6) margin code in the POSIX specification is required to be implemented by CGL. This functionality is not part of the six POSIX functionality areas defined above, but it is part of the CGL requirements. See the IPv6 requirements for more details.

**POC References**

[USAGI](http://www.linux-ipv6.org/)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.6</td>
<td>IPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

### STD.2.7 Monotonic Clock

**Description**: OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the MON (monotonic clock) margin code in the POSIX specification. This functionality is used in POSIX Timers.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.7</td>
<td>Monotonic Clock</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>
**POC References**

Linux kernel greater than 2.5.63

High resolution-timers project: [http://high-res-timers.sf.net](http://high-res-timers.sf.net) – for user space.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.8</td>
<td>Message Passing</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the MSG (message passing) margin code in the POSIX specification. This functionality is the core POSIX Message Queues functionality.

**POC References**

POSIX message queues for Linux by Wronski/Benedyczak: [http://www-users.mat.uni.torun.pl/~wrona/posix_ipc/](http://www-users.mat.uni.torun.pl/~wrona/posix_ipc/)

POSIX message queues userspace and filesystem based implementation by Peter Waetchler: [http://homepage.mac.com/pwaechtlr/linux/mqueue.tgz](http://homepage.mac.com/pwaechtlr/linux/mqueue.tgz)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.9</td>
<td>Realtime Signals Extension</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the RTS (realtime signals extension) margin code in the POSIX specification. This functionality is used in POSIX Signals.

**POC References**

The latest stable version of the Linux kernel.
### Description:
OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the SEM (semaphores) margin code in the POSIX specification. This functionality is the core POSIX Semaphores functionality.

**POC References**


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the SPI (spin locks) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TCT (thread CPU-time clocks) margin code in the POSIX specification. This functionality is used in POSIX Timers and POSIX Threads.

**POC References**

Linux kernel greater than 2.5.63

High resolution-timers project: [http://high-res-timers.sf.net](http://high-res-timers.sf.net) – for user space.
### Description
OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the THR (threads) margin code in the POSIX specification. This functionality is core POSIX Threads functionality.

### POC References


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

---

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.13</td>
<td>Threads</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

### Description
OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TMO (timeouts) margin code in the POSIX specification. This functionality is used in POSIX Semaphores and Message Queues.

### POC References


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

POSIX message queues for Linux by Wronski/Benedyczak: [http://www-users.mat.uni.torun.pl/~wrona/posix_ipc/](http://www-users.mat.uni.torun.pl/~wrona/posix_ipc/)

POSIX message queues userspace and filesystem based implementation by Peter Waechtler - [http://homepage.mac.com/pwaechtlr/linux/mqueue.tgz](http://homepage.mac.com/pwaechtlr/linux/mqueue.tgz)
### STD.2.15 Timers

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TMR (timers) margin code in the POSIX specification. This functionality is core POSIX Timers functionality.

**POC References**
- Linux kernel greater than 2.5.63
- High resolution-timers project: [http://high-res-timers.sf.net](http://high-res-timers.sf.net)

### STD.2.16 Thread Priority Inheritance

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TPI (thread priority inheritance) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**
- See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)
- See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

### STD.2.17 Thread Priority Protection

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TPP (thread execution scheduling) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**
- See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)
- See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)
### ID: STD.2.18
#### Name: Thread Execution Scheduling
#### Priority: 2
#### Category: Standards
#### Security Level: N/A
#### Maturity: Experimental
#### Default/Toggle: On/No

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TPS (thread execution scheduling) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**

See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

### ID: STD.2.19
#### Name: Thread Stack Address Attribute
#### Priority: 2
#### Category: Standards
#### Security Level: N/A
#### Maturity: Experimental
#### Default/Toggle: On/No

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSA (thread stack address attribute) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**

See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

### ID: STD.2.20
#### Name: Thread Safe Functions
#### Priority: 2
#### Category: Standards
#### Security Level: N/A
#### Maturity: Experimental
#### Default/Toggle: On/No

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSF (thread-safe functions) margin code in the POSIX specification.

**POC References**

See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.21</td>
<td>Thread Process-Shared Synchronization</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSH (thread process-shared synchronization) margin code in the POSIX specification. This functionality is used in POSIX Threads.

**POC References**


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.22</td>
<td>Thread Stack Address Size</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSS (thread stack address size) margin code in the POSIX specification. This functionality is used in POSIX Threads functionality.

**POC References**


See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.23</td>
<td>XSI Extensions/ISO C Extensions</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the XSI (XSI extensions/ISO C extensions) margin code in the POSIX specification. This functionality is used in POSIX Timers, Signals, and Threads.

**POC References**

The latest stable version of the Linux kernel.
### ID STD.2.24

<table>
<thead>
<tr>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization and Scheduling Support</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the margin codes in the POSIX specification that are required to be implemented by CGL to support real-time middleware and communications applications include:

- FSC (File Synchronization)
- SHM (Shared Memory Objects)
- ML (Process Memory Locking)
- PS (Process Scheduling)

**POC References:**

Linux 2.5 kernels and glibc 2.3

### ID STD.3.0 (1.5.0 in v1.1)

<table>
<thead>
<tr>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP Support Update</td>
<td>1 (P1 in v1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide Simple Network Management Protocol (SNMP) support. All three versions of SNMP agent (SNMPv1, SNMPv2, and SNMPv3) shall be supported. This includes support of SNMP for IPv4 environments as well as for IPv6 environments.

See the requirement SVC.1.2 SNMP Support for IPv6 for details about requirements for SNMP support for IPv6.

**POC References**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.4.0</td>
<td>Stream Control Transport Protocol (SCTP)</td>
<td>1 (P2 in v.1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for the following core Stream Control Transmission Protocol (SCTP) IETF RFCs:
- RFC 2960

**POC References**
Linux kernel greater than 2.5.39

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.1.0</td>
<td>Persistent Device Naming</td>
<td>1 (P1 in v.1.1)</td>
<td>Platform</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality such that a device’s identity shall be maintained when it is removed and reinstalled even if it is plugged into a different bus, slot, or adapter. “Device identity” is the name of the device presented to user space, and this identity is assigned based on policies set by the administrator, e.g., based on location or hardware identification information.

This requirement was called “Hot Device Identity” in an earlier version of these requirements. The current name better describes the desired capability.

**POC References**
### Description:
OSDL CGL specifies that carrier grade Linux shall provide the low-level hardware controls specified in the Intelligent Platform Management Interface (IPMI) hardware control/monitoring specification. CGL shall also provide interface drivers and support for higher-level software, such as that described in the SA Forum Hardware Platform Interface (HPI) specification.

### POC References
- OpenIPMI project: [http://sourceforge.net/projects/openipmi/](http://sourceforge.net/projects/openipmi/)
- IPMI specifications: [http://www.intel.com/design/servers/ipmi/spec.htm](http://www.intel.com/design/servers/ipmi/spec.htm)

---

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.2.0</td>
<td>IPMI 1.5 Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for the Service Availability Forum Hardware Platform Interface (HPI) specification. The HPI specification calls for a common interface for all manageable components on a platform including basic components, like fans and temperature sensors, as well as platform-unique components.

The nature of HPI makes it impossible to require support for every type of manageable hardware component, so it is assumed that the HPI implementation will include a modular mechanism for adding support for specific hardware devices. It is further assumed the solution provider will need to configure the HPI implementation and possibly provide additional hardware access modules unique to the target platform.

This requirement specifically calls for a common interface as seen by an HPI caller and does not assume a specific target platform; therefore, it does not require specific hardware capabilities.

### POC References
- OpenHPI project: [http://sourceforge.net/projects/openhpi](http://sourceforge.net/projects/openhpi)
**AVL.1.0** Robust Mutexes 1 Availability N/A Production On/No

**Description:** OSDL CGL specifies that carrier grade Linux shall provide an enhancement to the POSIX Thread implementation that provides robust mutexes support. Robust mutexes support shall permit a mutex to synchronize threads, either in the same process or in different processes, even when processes exit or abort unexpectedly.

A robust mutex is initialized with robust mutex attributes. It must be an inter-process shared mutex, allocated in a shared memory segment mapped into the processes that use it. Applications using a robust mutex shall be able to see various return codes that indicate whether the previous holder of the mutex terminated, and also the recovery status of the state of the mutex. The new holder of the robust mutex shall be able to detect a failure, perform cleanup actions, and re-initialize the mutex for continued use.

If a cleanup of the state protected by the mutex can't be completed, the mutex shall be marked "inconsistent" so that any future attempts to lock it will generate a status indicating that it is inconsistent. The following two modes for setting the mutex to an inconsistent state shall be provided:

- Automatically mark the mutex “inconsistent” when the owner dies and a subsequent lock is done.
- Provide an advisory to the next owner that the mutex needs to be explicitly marked inconsistent.


**POC References**


Robust Mutexes: [http://developer.osdl.org/dev/robustmutexes/](http://developer.osdl.org/dev/robustmutexes/)


**AVL.2.0** Software ECC Support 1 Availability N/A Production Off/Yes

**Description:** OSDL CGL specifies that carrier grade Linux shall provide error correction codes to allow hardware error correction for detecting and/or recovering from memory errors and detecting and reporting single-bit and multi-bit ECC errors from the memory subsystem.

**POC References**


Error Correcting Codes (ECC) Site: [http://www.eccpage.com/](http://www.eccpage.com/)
### AVL.3: Software Live Installation and Upgrade

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a software live update feature. It shall include functions that allow kernel and application software to be upgraded, while minimizing downtime of the system. Specific downtime targets will need to conform to product-specific system requirements based on the application for which the system is being used.

Features that support installations and upgrades must interact with and support significant capabilities for identifying and tracking the software installed on the system.

To allow flexibility in supporting a wide range of platforms and applications, CGL does not specify a specific time limit for downtime.

**POC References**

- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE - YaST

### AVL.3.1: Software Remote Update and Installation

**Name Clarification:** Remote Package Update and Installation

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a remote software package update feature. It shall include functions that allow kernel modules and application software to be installed or upgraded remotely, while minimizing downtime of the system.

The usage of the term “remotely” does not imply a central package management platform, nor does it preclude such a system. This requirement only necessitates that a single device may be upgraded without requiring the administrator to be physically at the device.

Due to the wide range of platforms and applications, CGL does not specify a specific downtime limit metric. Downtime targets will vary based on the system application.

**POC References**

- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE - YaST
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.2</td>
<td>Software Live Upgrade</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
<tr>
<td></td>
<td>Minimal Reboot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3.8.1 in v1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Name Clarification:** No System Reboot

**Description:** OSDL CGL specifies that carrier grade Linux shall provide CGL shall ensure that remote software installation and software upgrade mechanisms that requiring no system reboots:

- No reboot shall be required to upgrade kernel module components (such as device drivers)
- No reboot shall be required to upgrade user-space applications provided by CGL system software.

It is noted that certain custom applications may require a reboot, but the CGL system software should not.

Remote software installation and upgrade mechanisms will not require any more reboots than the same upgrade done via console.

**POC References**


Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE – YaST

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.3</td>
<td>Software Live Upgrade RPM Version Check</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
<tr>
<td>AVL.3.4</td>
<td>Software Live Upgrade Log</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Name Clarification:** Version and Dependency Checking via RPM

**Description:** OSDL CGL specifies that carrier grade Linux shall provide CGL remote software installation and upgrade capabilities shall that include provisions for version compatibility and dependency checking at the RPM package level.

The upgrade process shall allow the coexistence of new and old executables, shared libraries, configuration files, and data.

It is reasonable that this requirement be implemented as a combination of the installer and the chosen package management system.

**POC References**


Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE – YaST

---

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.4</td>
<td>Software Live Upgrade Log</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Name Clarification:** Upgrade Log

**Description:** OSDL CGL specifies that carrier grade Linux shall provide CGL remote software installation and upgrade shall mechanisms that perform transaction logging of dates, times, changes and the identity of the user performing the change.

**POC References**


Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE – YaST
### Description:
OSDL CGL specifies that carrier grade Linux shall provide software remote upgrade mechanisms that support multiple versions of applications and kernels on image target nodes. The new version shall be installable without interfering with the execution of the old version or with the old version's configuration, logs and other files or information.

An image server is the node on which all of the code to be installed resides. An image target is any node installed or updated from an image server. Nodes can be either servers or targets, such as when multiple nodes are installed by a single server, or they can switch roles over time.

A machine can be switched between different versions to allow it to execute on either the old or the new version. Switching between versions can require:

- Stopping and restarting applications
- Rebooting to run on a new kernel

In addition, each node can be customized, using post-install scripts or other techniques, to allow node-specific data, such as hostnames, IP addresses, and application configuration information, to be configured on the remotely installed nodes. This data is maintained on the server and automatically configured.

### POC References
None

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.5</td>
<td>Software Live Upgrade per-Node Custom Image and Configuration</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that carrier grade Linux shall provide mechanisms that allow manual rollback to a previous version of the software without having to reinstall the previous version.

This can be done in the following ways:

- Reinstall from the master node
- Switch from one version to another on the client node. (Requires the Software Live Upgrade Multiple Client Versions requirement to be supported.)

### POC References
None
Description: OSDL CGL specifies that carrier grade Linux shall provide mechanisms that allow automatic rollback to a previous version of software without having to reinstall the previous version. Configuration capabilities must be provided that allow automatic rollback conditions to be defined to take place without the intervention of the system administrator.

POC References
Rollback failed RPM transactions: http://lee.k12.nc.us/~joden/misc/patches/rpm/

Description: OSDL CGL specifies that carrier grade Linux shall provide that applications and the kernel be able to be integrated with the installation and upgrade mechanisms such that they can be informed of and participate with the installer in upgrading the software.

For carrier-class systems, on-line upgrading of operating systems and application software is a high priority. It must be completed without any system interruption or downtime.

To clarify, downtime refers to downtime of the services being provided rather than downtime of a specific machine. Therefore, achievement of this requirement requires that more than one node share the load in a clustered system.

The implementation used to satisfy this requirement must achieve the following:

- It must provide mechanisms to upgrade a running application such that the system will be able to handle an old and a new version of the application running simultaneously. The new version of the application will be servicing new requests and the older version will be completing old requests and not taking any new requests.

- It must provide a change from one version of the Linux kernel to another without any system interruption when the kernel is upgraded to a new version.

POC References
None
### AVL.3.9 Software Live Upgrade Fine-Grain Version Checking

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.9</td>
<td>Software Live Upgrade Fine-Grain Version Checking</td>
<td>3</td>
<td>Availability</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide remote software upgrade mechanisms shall include provisions for version compatibility and dependency checking at a fine-grain file or application level. The upgrade process shall allow the coexistence of new and old executables, shared libraries, configuration files and data.

The ability must be provided to identify the levels of specific binaries and files to ensure that the kernel, packages, applications and files an expected version inventory on the system and can be identified, and that compatibility information can be built and maintained.

**POC References**

None

---

### AVL.4.0 Force Unmount

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.4.0 (4.10 in v1.1)</td>
<td>Force Unmount</td>
<td>1 (P2 in v1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for forced unmounting of a file system. The unmount shall work even if there are open files in the file system. Pending requests shall be ended with the return of an error value when the file system is unmounted.

**POC References**

### AVL.5.0 Linux Panic Handler Enhancement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.5.0 (4.11.0 in v1.1)</td>
<td>Linux Panic Handler Enhancement</td>
<td>1 (P3 in v1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide enriched capabilities in response to a system panic. Currently the default system panic behavior is to print a short message to the console and halt the system. CGL shall provide a set of configurable functions, including logging the panic event to the system event log, as well as options to reboot, power down, or cycle power, when a panic event occurs.

CGL shall support enhanced kernel panic reporting, at a minimum supporting proper resolution of in-kernel symbols. This will make kernel panic reports useful to administrators that do not have access to the kernel for which the report was generated.

**POC References**

- kksymoops: [http://lwn.net/Articles/10796/](http://lwn.net/Articles/10796/)
- Linux 2.6 kernel + OpenPMI [http://openipmi.sf.net](http://openipmi.sf.net)

---

### AVL.6 Memory Overcommit Actions

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.6</td>
<td>Memory Overcommit Actions</td>
<td>Mixed</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide the ability to configure a global limit on RAM utilization. This limit is a combination of physical memory and swap space. In addition, adequate information and an interface must be provided to allow a middleware component to take action before the system runs out of memory. This is in addition to or a replacement for the kernel out-of-memory killer.

**POC References**

### Description:
OSDL CGL specifies that carrier grade Linux shall provide the ability to control kernel virtual memory allocation adjustments based on the specific needs of the system. Control of virtual memory shall include but not be limited to the following:

- **Strict over-commit** - The total address space commit for the system is not permitted to exceed swap + a configurable percentage (default is 50%) of physical RAM.
- **Heuristic over-commit** - Obvious over-commits of address space are refused. Limited to free physical memory + free swap.

### POC References


---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide mechanisms to allow the replacement of the out-of-memory (OOM) killer algorithm within the kernel. In an environment in which an application is made up of many processes, the act of killing any single process may prevent the application from continuing to provide service while leaving its remaining processes and preventing proper recovery. Hence it must be possible to provide a replacement algorithm that can take the relationships between processes into account when determining which ones to slay. By default the current algorithm in the kernel is used, and the new algorithm can be activated by loading the relevant kernel module.

### POC References

None

---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide consistent support for IPv4 and IPv6 environments and provide a baseline set of MIBs.

### POC References

**NET-SNMP:** [http://www.net-snmp.org](http://www.net-snmp.org)
### Description: OSDL CGL specifies that carrier grade Linux shall provide support for SNMP for IPv4 environments. This includes operation of SNMP across IPv4 links as well as including MIBs that extend functionality or provide unique functionality for IPv4 environments.

### POC References

NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)

The Net-SNMP source tree is the only SNMP agent of this lineage that will be enhanced with new functionality, including the required IPv6 support. IPv6 support is in development and will not be available in the UCD-SNMP source tree.

### Description: OSDL CGL specifies that carrier grade Linux shall provide support for the SNMP protocol over UDP/IPv6. This support would allow an SNMP manager, running natively over IPv6 network, to retrieve and configure information sent from or to the hosts directly over IPv6.

### POC References

NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC.1.3</td>
<td>SNMP Baseline MIBs</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide at minimum the following baseline set of MIBs in all environments:

- RFC1213-MIB: MIBII, mandatory
- RFC-1215: Trap types for MIBII
- IF-MIB: New and better interface building on MIBII
- IP-MIB: Updates to IP and ICMP from MIBII
- TCP-MIB: Update to TCP management in MIBII
- UDP-MIB: Update to UDP management in MIBII
- HOST-RESOURCES-MIB: Management of server host resources.
- UCD-SNMP-MIB: Very complete UCD system, proc, disk, mem, load, etc.
- NET-SNMP-AGENT-MIB: Control and monitoring NetSNMP agent.
- NET-SNMP-MIB: Top-level registration point for NetSNMP MIB extensions
- NOTIFICATION-LOG-MIB: SNMP Notification (traps) logging
- SNMP-MPD-MIB: Agent SNMP message stats
- SNMP-NOTIFICATION-MIB: Agent notification configuration and filtering
- SNMP-TARGET-MIB: Agent management target config for notifications
- SNMP-USER-BASED-SM-MIB: Management of user-based security model
- SNMP-VIEW-BASED-ACM-MIB: Management of view-based access control
- SNMPv2-CONF: Definitions for v2 conformance
- SNMPv2-MIB: v2 updates to MIBII
- TUNNEL-MIB: Management of IP tunnels
- DISMAN-EVENT-MIB: Framework for event MIBs
- MTA-MIB: Management of Transfer Agents (mail and directory)

**POC References**

NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)

The Net-SNMP source tree is the only SNMP agent of this lineage that will be enhanced with new functionality, including the required IPv6 support. IPv6 support is in development and will not be available in the UCD-SNMP source tree.
### Description:
OSDL CGL specifies that carrier grade Linux shall provide required support for all IPv6 MIBs as specified by IETF IPv6 WG except for a very primitive level support for the general MIB group as described in the documents below:

- IP Version 6 MIB for the Transmission Control Protocol (RFC 2452)
- IP Version 6 MIB for the User Datagram Protocol (RFC 2454)
- MIB for IP Version 6: Textual Conventions and General Group (RFC 2465)
- MIB for IP Version 6: ICMPv6 Group (RFC 2466)

This means that internally within the kernel IPv6 stack, statistics and/or information corresponding to MIB data must be gathered and saved for ultimate retrieval by an SNMP Agent.

There is currently an ongoing effort within IETF to combine IPv4 and IPv6 MIBs into unified MIBs. The developer may choose to implement RFC2011, RFC2012, RFC2013 instead of RFC2452, RFC2454, RFC2465, and RFC2466.

### POC References
NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)
### Description

**OSDL CGL** specifies that carrier grade Linux shall provide an efficient capability for handling a large number of essentially simultaneous asynchronous events arriving on multiple channels like multiple sockets or other similar paths.

The motivation behind this mechanism is to enforce the system scalability and soft real-time responsiveness by reducing contentions appearing at the kernel level, especially under high load.

### POC References


epoll() (being added to kernel versions beginning with 2.5.46) in combination with libevent from [http://monkey.org/~provos/libevent/](http://monkey.org/~provos/libevent/).

### Description

**OSDL CGL** specifies that carrier grade Linux shall provide the capability of configuring the scheduler to provide soft real time support so the soft real time scheduling latency of a given task will not exceed a target defined by the vendor. Based on commodity hardware commonly supported by Linux, latency responses in the range of 10 to 15 milliseconds should be considered reasonable and likely.

### POC References

Preemption and a scalable scheduler are provided in the latest stable kernel version

### Description

**OSDL CGL** specifies that carrier grade Linux shall provide support for a self-resizing file system for transient data, which can be limited to a maximum size.

### POC References

/tmpfs implementation in the kernel

## PRIORITY 2 REQUIREMENTS

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.4.0</td>
<td>Boot Cycle Detection</td>
<td>2</td>
<td>Platform</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>(2.3 in v1.1)</td>
<td></td>
<td>(P1 in v.1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for detecting a repeating reboot cycle due to recurring failures and going to an offline state if this occurs.

**POC References**

None

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.5.0</td>
<td>Network Console Operation</td>
<td>2</td>
<td>Platform</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>(2.6.2 in v1.1)</td>
<td></td>
<td>(P2 in v.1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that Linux shall provide support for a connection to a system console via a network port in addition to providing support for a serial port.

**POC References**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.7.0</td>
<td>Fault Isolation Enabling</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support to report anomalies it has detected on a compute node. The objective in reporting these anomalies is to provide data for fault isolation mechanisms. Software-related failures may require actions like the restart or termination of a process or the unloading and reinstallation of a kernel module. Hardware-related failures may require actions to restart, turn off or isolate a failing device.

OSDL CGL specifies that carrier grade Linux shall provide mechanisms to isolate faulty software or hardware components. These mechanisms can be activated by management middleware fault isolation policies.

**POC References**

OpenIPMI: [http://openipmi.sf.net](http://openipmi.sf.net)
ID | Name                                      | Priority | Category | Security Level | Maturity | Default/ Toggle |
---|------------------------------------------|----------|----------|----------------|----------|----------------|
AVL.8 | Non-intrusive Monitoring of Processes   | 2        | Availability | N/A            | N/A      | N/A            |

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a range of capabilities to enable non-intrusive monitoring of processes. To enable monitoring, some configuration actions may have to be taken to specify which processes are to be monitored. Capabilities may be limited in certain cases, as long as the limitations are known. Capabilities to be provided include the following:

- Processes must be manageable and controllable even if the actual process code cannot (or will not) be changed to exploit a specified API.
- Processes must be manageable and controllable even if they are not a direct child process of the tools and mechanisms provided to enable these capabilities. A carrier system consists of middleware and processes from many sources, which may be difficult to run from a single parent process as they will usually require different userids, capabilities, permissions, etc.
- The latency of event detection while processes are being monitored must be as low as possible, preferably occurring immediately upon complete failure of a process.
- The overhead of monitoring the processes should be as low as possible.
- initramfs does not provide sufficient capabilities to meet this requirement. Therefore, enhancements to initramfs must be provided to address the following limitations:
  - Monitors only processes initramfs starts
  - Limited reactions to process death
  - No healthcheck capabilities for non-terminating processes
  - No controls on respawn loops of processes

**POC References**
See POC References in the sub-requirements below.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.8.1</td>
<td>Process-level Non-intrusive Application Monitor</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Started</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide control and management capabilities for processes that cannot be changed to any specific API. Such capabilities are known as non-intrusive monitoring. These capabilities must be implemented programmatically using commands or scripts.

Another issue for many such processes is that the start script itself may spawn a child that is then the actual process that makes up the application. This process then is not an actual child of the control and management process. This sub-requirement assumes that this does not happen, and the child process remains under the control of the management entity.

Capabilities required:

- The following capabilities must be enabled for controlling processes:
  - The ability to start a process (or a list of processes)
  - The ability to stop a process (or a list of processes)

- The following capabilities must be enabled for monitoring processes:
  - The ability to detect the unexpected exit of a process
  - The ability to configure a set of actions in response to an unexpected exit of a process

- The following services must be provided beyond those currently provided by initab:
  - The ability to configure whether to restart the application if the process dies
  - A configurable amount of time to wait before restarting the application
  - A limit on the number of times to restart the application

**POC References**
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/</th>
<th>Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.8.2</td>
<td>Kernel-level Non-intrusive Application Monitor</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>None</td>
<td>On/No</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** CGL shall support methods to non-intrusively monitor processes at the kernel level. For example, if requirement AVL.8.1 Process-level Non-intrusive Application Monitor is implemented, then monitoring must continue even if the monitoring process fails.

One approach is to restart the monitor process using `init`, since `init` won’t fail unless the system fails.

A second approach is to use the notification mechanism in the kernel that allows registration of events of interest regarding processes. Events of interest, like the following, could be monitored using this mechanism:

- Process creation (fork())
- Process exit
- Process calls exec()
- Creation of (user and/or kernel) threads
- Exit of (user and/or kernel) threads

The implementation of this second approach needs to provide high performance with low overhead. Monitoring will preferably be driven directly by process events in the kernel to ensure very low latency access to the interesting events. Such an implementation will require kernel changes, because a solution such as polling the set of processes listed in `/proc` is far too slow, resource expensive, and error prone.

The calling kernel module registers a notification routine that is executed within the kernel during some point during the processing of the events of interest. The caller must be able to specify, at minimum, different notification routines for different events.

It is likely that a feature such as this would be desirable for cluster services, like a distributed lock manager, to monitor client processes that use their services. A distributed lock manager cannot be the parent of processes using it. However, it must perform lock cleanup actions if a process owning locks should fail.

**POC References**

None
### Description:
OSDL CGL specifies that carrier grade Linux shall provide capabilities to assist in predictive analysis of disks. The aim of this support is to assist in predicting situations likely to lead to failure of disks. This allows preventive action to be taken to avoid the failure and resulting disruption of service.

Note that this could be considered a subset of the requirement **SVC.4.0 Online Diagnostics**, but since isolated mechanisms to support this requirement currently exist, it is listed as a separate requirement.

### POC References
Some disk devices offer support for the SMART protocols for direct queries to disk devices. Based on feedback and comments, it appears this functionality is often problematic. However, it is mentioned here because it does exist, and it is available in some limited fashion:


An alternative approach is tied to Performance CoPilot (PCP) that measures defect growth, SCSI resets, parity errors, etc. over time to form a better prediction mechanism.


---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide an HPI implementation that is certified in accordance with the SA Forum HPI certification process.

Note that the SA Forum has not yet established an HPI certification process, but it is the understanding of the CGL working group that such a certification process is a high priority to the SA Forum and will become available soon after the release of this document.

### POC References
Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)
**Description:** OSDL CGL specifies that carrier grade Linux shall provide a data checkpointing service. This service provides contextual state replication on a single node or across cluster nodes and provides service continuity upon node failure. The targets for data checkpointing are applications, not middleware. The checkpointing service shall include the following:

- Data checkpoints that contain user-specified data.
- The same API should work for a single node as well as when replicating checkpoints across nodes. In essence, a single node checkpoint is one with no replicas.
- A checkpoint needs to persist if the process using it dies, so that a recovering process can grab it. The checkpoints should either have a time-to-live expiration time limit and be automatically reclaimed, or provide a mechanism (c.f. ipcrm) to clean up old ones.
- A checkpoint is not persistent across a reboot of a node. Thus, unless it has a replica on another node, rebooting the node causes it to disappear.
- A capability to replicate checkpoints in the cluster either synchronously or asynchronously.
- Checkpoints that are transactional (no partial checkpoints) to provide atomicity (either the data unit is completely checkpointed or it is not checkpointed at all).
- Checkpoints need to be secured such that only approved users can access them, e.g., based on uid and/or gid of the creator.
- Writing a checkpoint from one node, and reading on another node for 4K of checkpoint data should take less than 5 ms.

Checkpoint integrity is provided by the checkpointing service.

**External Specification Dependencies:**

The Service Availability Forum Application Interface Specification (SA Forum AIS) addresses data checkpointing (*SA Forum AIS - Chapter 7*) and may be leveraged to define the interface for this feature.

**POC References**

Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)

Data checkpointing project as part of Linux-ha, see linux-ha-dev mailing list, available through [http://linux-ha.org](http://linux-ha.org). This will use the Service Availability Forum Checkpoint Service API for guidance.
**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable multiple access paths from a single cluster node to storage devices. The software shall determine if multiple paths exists to the same port of the I/O device and with configurable controls it will balance I/O requests across multiple host bus adapters. If multiple paths exist to the same device over two separate device ports on the same host bus adapter, those I/Os will not be balanced.

Handling a path failure must be automatic. A mechanism must be provided for the reactivation of failed paths, which can be placed back into service. It must be possible to automatically determine and configure multiple paths. Automatic configuration shall allow automatic multi-path configuration of complete disks and partitions located on those disks.

**POC References**
The latest stable Linux kernel currently supports multi-path I/O via the md layer. The problem with this approach is that there is no easy way to configure it, and incorrect configuration can lead to data loss.

IBM has an open source project for SCSI mid-level multipath implementation. The documentation for this project is at:

http://www-124.ibm.com/storageio/multipath/scsi-multipath/docs-0.1/

The uSDE project (see PLT.1.0 on page 34) provides automatic configuration of the MD for multipath disk access.

http://source.mvista.com/sde

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.13.0</td>
<td>NFS Client Protection across Server Failures</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide mechanisms that allow an NFS server to have failover capability to provide service continuity upon a node failure. The NFS service has to be resumed on another node without any impact on NFS clients other than the retransmission of pending requests (open files must remain open). Authenticated clients on the old server must remain authenticated by the new server.

**POC References**

NFS failure discussion:

Linux-HA ([http://www.linux-ha.org](http://www.linux-ha.org)) plus patch described at
**Description:** OSDL CGL specifies that carrier grade Linux shall provide a diagnostics framework and shall provide interfaces to support third-party CIM Provider and CIM Object Manager (CIMOM) applications so these applications can use the CGL framework to perform diagnostic functions for available system devices. CIM Provider and CIMOM specifications are based on the following:

- CIM Specifications V2.2 or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Schema V2.8 or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Diagnostic Model Specification V 1.012 or higher (available at [http://www.dmtf.org/education/whitepapers.php](http://www.dmtf.org/education/whitepapers.php))

**POC References**

---

**Description:** OSDL CGL specifies that carrier grade Linux shall provide enhancements to current kernel dumping mechanisms, including a way to define to which device the dump is generated and a way to decrease the size of the dump to contain only important information.

**POC References**
- Source RPM: [ftp://rpmfind.net/linux/rawhide/1.0/SRPMS/SRPMS/netdump-0.6.9-1.src.rpm](ftp://rpmfind.net/linux/rawhide/1.0/SRPMS/SRPMS/netdump-0.6.9-1.src.rpm)
- Software Suspend – the combination of power management and software suspend could be leveraged for a mainline kernel dump mechanism.
### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for producing and storing kernel dumps as follows:

- It must be possible to store kernel dumps to disk, across a network or to memory.
- Regardless of specific dump target, dumps must be preserved across the next system boot.
- Information pertaining to all CPUs and all kernel threads must be preserved.
- A kernel dump will provide the ability to capture enough of the kernel state to permit post-crash analysis of the conditions that led to a server problem, which may be a hardware, software, resources or configuration issue.

### POC References
See base SVC.3 requirement.

### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for producing summary dumps containing only specified data structures. A way must be provided for a system administrator to specify which data structure to include must be provided.

### POC References
See base SVC.3 requirement.

### Description:
OSDL CGL specifies that carrier grade Linux shall provide the ability to take a dump image of the system and applications running on the system.

### POC References
See the POC References in the sub-requirements below.
**Description:** OSDL CGL specifies that carrier grade Linux shall provide mechanisms to allow a dump image to be taken of a running application (process). While this capability would typically be used during a system debugging session, it must also be possible to use it during production. It is expected that the dump would affect application performance as follows:

- The application must be frozen during the time the dump image is being created to ensure that a consistent image is stored.
- While the application is frozen, it will not be able to respond to events like messages or signals.

This capability is important for situations like the following:

- If an application crashes that relies heavily on another application, the state of the other application may be needed to debug the problem.
- When debugging problems that do not impact service, like memory leaks.

**POC References**


---

**Description:** OSDL CGL specifies that carrier grade Linux shall provide mechanisms to allow a dump image to be taken of a running system. The system must be frozen during the time the dump image is being created to ensure a consistent image is stored. When the system is frozen, it will not be able to respond to external events.

**POC References**

None
### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for fully loading an application before beginning execution. `mlock()` alone does not meet this requirement because it requires superuser privileges.

#### POC References

See subrequirements.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRF.3 (6.3 in v1.1)</td>
<td>Application (Pre)loading Capability</td>
<td>2</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for the preloading of applications even when they are not executing as root. A configuration capability must exist to allow the system loader to determine applications eligible for preloading.

The action of preloading an application must not overload the system memory. The configuration capability must provide a control that allows the application to specify what is to be done if it can't be pre-loaded. Options are as follows:

- Load anyway as a normal (page-able) application.
- Fail and don't load the application.

Regardless of the action taken, any failure to pre-load the application must be logged.

#### POC References

Description: OSDL CGL specifies that carrier grade Linux shall provide mechanisms to avoid overloading a system when preloading applications, specifically, it must be possible to specify the total amount of memory reserved (pinned) by preloading applications.

POC References
None

Description: OSDL CGL specifies that carrier grade Linux shall provide support for tools to enable enhanced analysis of kernel dumps. These enhancements must include, but not be limited to, the following capabilities:

- Accessing kernel structures
- Virtual to physical address translation
- Module access

POC References

Description: OSDL CGL specifies that carrier grade Linux shall provide a fault injection capability that can be used as the test harness for injecting faults into device drivers and their resource areas for testing purposes.

POC References
Fault Injection Test Harness Project:
- Project home page: [http://sf.net/projects/fault-injection](http://sf.net/projects/fault-injection)
## Description:
OSDL CGL specifies that carrier grade Linux shall provide a kernel level infrastructure for defining, controlling and signaling software simulated faults. This framework is meant to enable the development of a wide variety of software-simulated fault conditions as separate kernel modules, and to allow a user-space test harness to be implemented without special knowledge of all possible fault simulation modules (fault injection points).

### POC References

Fault Injection Test Harness Project:

## Description:
OSDL CGL specifies that carrier grade Linux shall provide fault points for the following commonly utilized driver resources:
- Memory mapped I/O
- Programmable I/O
- Interrupts

These fault injection points shall be built on top of the kernel injection infrastructure (see requirement TLS.3.1 Fault Injection Framework) to act as both a general purpose set of injection points useful for testing of most device drivers and as examples to seed the development of a richer set of fault injection points.

### POC References

Fault Injection Test Harness Project:
- Project home page: [http://sf.net/projects/fault-injection](http://sf.net/projects/fault-injection)
### TLS.3 (5.5 in v1.1)

**Name**: Kernel Profiling

**Priority**: 2 (P2 in v1.1)

**Category**: Tools

**Security Level**: N/A

**Maturity**: Experimental

**Default/Toggle**: Off/Yes

**Description**: OSDL CGL specifies that Linux shall support profiling of the running kernel to identify bottlenecks and other interesting information. There are a number of methods extant for providing these features, but none are accepted as mainstream Linux methods.

**POC References**


### TLS.3.1 (5.5.1 in v1.1)

**Name**: Kernel Flat/Graph Execution Profiling

**Priority**: 2 (P2 in v1.1)

**Category**: Tools

**Security Level**: N/A

**Maturity**: Experimental

**Default/Toggle**: Off/Yes

**Description**: OSDL CGL specifies that carrier grade Linux shall provide support for profiling of the running kernel using a prof or gprof style recording of trace information during system execution.

**POC References**

See references in base requirement TLS.3.

### TLS.3.2 (5.5.2 in v1.1)

**Name**: Kernel Sampling for Profiling

**Priority**: 2 (P2 in v1.1)

**Category**: Tools

**Security Level**: N/A

**Maturity**: Experimental

**Default/Toggle**: Off/Yes

**Description**: OSDL CGL specifies that carrier grade Linux shall provide support for profiling of the running kernel by providing profiling based on interrupt sampling, for example:

- Take an interrupt.
- Record execution point.
- Repeat on rapid (micro- or milli-second multiples).
- Analyze to build up profile of system execution history.

**POC References**

See references in base requirement TLS.3.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS.4.0 (5.6 in v1.1)</td>
<td>System Tools to Analyze Execution Profiles</td>
<td>2 (P2 in v.1.1)</td>
<td>Tools</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for interfaces to allow utilities to collect data from trace kernel and application level activity.

**POC References**
### Description:
OSDL CGL specifies that carrier grade Linux shall provide support for forcing processes to execute on a specified CPU to avoid interfering with interrupt handling.
- This should be configurable by the systems administrator to set for each application.

### POC References
**CPU affinity kernel patches:** [http://www.kernel.org/pub/linux/kernel/people/rml/cpu-affinity/](http://www.kernel.org/pub/linux/kernel/people/rml/cpu-affinity/)
Also see “Syscall-interface for CPU affinity” at [http://www.tech9.net/rml/linux/](http://www.tech9.net/rml/linux/)

### Description:
OSDL CGL specifies that carrier grade Linux shall provide for a choice of process scheduling policies to the user. OSDL CGL specifies that carrier grade Linux shall provide a framework that allows multiple process scheduling policies to be plugged into the kernel easily, allows a user to select one of the available policies when configuring the kernel for compilation, and provides a developer’s guide to help developers implement new scheduling policies.

### POC References

### Description:
OSDL CGL specifies that carrier grade Linux shall provide mechanisms that ensure the scheduler can avoid priority inversion blocking of lower-priority processes blocking higher-priority processes. This is analogous to requirement STD.2.16, which addresses threads within a process, but applied to processes.

### POC References
See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)
See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)
In the context of Carrier Grade Linux Requirements Definition Version 2.0, the document outlines requirements for interruptless Ethernet delivery. Here are the relevant sections:

### PRF.7 - Interruptless Ethernet Delivery

**Description:** OSDL CGL specifies that carrier grade Linux shall provide for the capability for Ethernet drivers that contain a pure polling mode in which they do not generate interrupts for arriving frames. This is to prevent interrupt-storms from consuming too many CPU cycles. This is primarily an issue for gigabit Ethernet.

**POC References**
See information from /pub/Linux/net-development/NAPI:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRF.7.1</td>
<td>Interruptless Ethernet Delivery Kernel Framework</td>
<td>2</td>
<td>Performance</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a framework to allow the capability for Ethernet drivers that contain a pure polling mode, in which they do not generate interrupts for arriving frames. This is to prevent interrupt-storms from consuming too many CPU cycles.

**POC References**
See information from /pub/Linux/net-development/NAPI:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRF.7.2</td>
<td>Interruptless Ethernet Delivery Driver Support</td>
<td>2</td>
<td>Performance</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide for Ethernet drivers that take advantage of the kernel capabilities to operate in a pure polling mode, in which they do not generate interrupts for arriving frames. This is to prevent interrupt-storms from consuming too many CPU cycles.

**POC References**
See information from /pub/Linux/net-development/NAPI:
# Priority 3 Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRF.8.0</td>
<td>Page flushing</td>
<td>3</td>
<td>Performance</td>
<td>N/A</td>
<td>None</td>
<td>On/No</td>
</tr>
<tr>
<td>(6.8 in v1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide mechanisms to allow either application- or operator-controllable parameters to modify page-flushing operations. This capability must be configurable on a per-process or per-application basis and also as a global setting. Note that this requirement may have security implications.

Existing functions, such as fsync() and fdatasync(), are possible starting points for a solution for this requirement. These functions apply to files and need to be executed by the application itself rather than by an administrator or a manager program that monitors the system and adjusts for different requirements. For this requirement, the system also needs to be able to flush application memory pages onto swap space.

From a functional standpoint, this requirement is meant to allow the system to directly control system memory usage on a very granular basis. During different periods, different applications will be pinned into memory, and the system can be reconfigured to force out some applications and pin others.

**POC References**

fsync() and related functions as possible starting points
5 Clustering Requirements

This section contains requirements specific to supporting clustering. Most are new in this version of the OSDL “Carrier Grade Linux Requirements Specification.” A few requirements have been carried forward from the OSDL” Carrier Grade Linux Requirements Specification, Version 1.1”. For these, the version 1.1 priorities have been included in parentheses below the current priority.

Not all CGL deployments in the communications industry will be for clustered systems. For this reason, all clusters requirements are deployed by default as OFF. All of the clusters requirements have a toggle property of YES, and it must be possible to enable clustering capabilities and services as a unit.

Priority 1 Requirements
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFH.2.0</td>
<td>Cluster Node Membership with Failure Detection</td>
<td>1</td>
<td>Clustering – Cluster Fault Handling</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a fast, LAN-based cluster node failure mechanism that is reflected in a cluster membership service, which provides at minimum indications of the nodes that are currently active in the cluster. The membership change must result in a membership event that can be monitored by cluster services, applications, and middleware that register for membership events.

The faster this can be done, the more applicable to a broader range of carrier grade applications, with preference for sub-second indication, although the targeted use of the cluster will affect the decision to use any specific implementation.

Fast node failure detection must not depend on a failing node reporting that the node is failing. However, self-diagnosis may be leveraged to speed up failure detection in the cluster.

This requirement does NOT address the issue of how to prevent failing nodes from accessing shared resources.

Fast node failure detection shall include the following capabilities:

- Ability to provide cluster membership health monitoring through LAN(s).
- Support for multiple, redundant LAN-based communication paths to check the health of other nodes.
- Use of only a small percentage of the total LAN bandwidth for membership health monitoring. The guideline is that the bandwidth used by the health monitoring mechanism shall be linear for the number of bytes per second per node.
- Support for fast failure detection. The guideline is a maximum of 250ms for failure detection. Since there is tradeoff between fast failure detection and potentially false failures, the health-monitoring interval must be tunable.
- Ability to provide a cluster membership change event to middleware and applications.

**External Specification Dependencies:**

- The SA Forum AIS addresses cluster membership (Chapter 6 – SA Cluster Membership Service API) and event services (Chapter 8 – Event Service API) and may be leveraged to define the interfaces for this feature.
- The OCF addresses cluster membership and event services and may be leveraged to define the interfaces for this feature.
POC References
Heartbeat: http://linux-ha.org/heartbeat/
SGI Linux Failsafe: http://oss.sgi.com/projects/failsafe/
OSDL Telecom Inter Process Communication (TIPC): http://developer.osdl.org/dev/tipc
OpenGFS: http://opengfs.sourceforge.net – failure detection components

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1</td>
<td>Cluster Communication Service</td>
<td>Mixed</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Description: OSDL CGL specifies that carrier grade Linux shall provide a fast and reliable intra-cluster communication service that provides advantages beyond TCP/IP. It must provide the necessary infrastructure for cluster, network and software management communication. It must provide support for designing platform-independent, scalable, distributed, highly-available and well-performing communications applications.

POC References:
Heartbeat: http://linux-ha.org/heartbeat
TIPC: http://sourceforge.net/projects/tipc
OSDL TIPC: http://developer.osdl.org/dev/tipc
LAM/MPI distributed messaging: http://www.lam-mpi.org

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1.1</td>
<td>Cluster Communication Service – Logical Addressing</td>
<td>1</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

Description: OSDL CGL specifies that carrier grade Linux shall provide a cluster communication service with a socket-based interface that provides logical addressing for point-to-point and multipoint communication. The communication service must hide the cluster’s physical topology from application programs with this logical addressing scheme. Mapping between logical and physical addresses must be performed transparently. In addition, there must be no user-level distinction between inter- and intra-node communications or between user-space and kernel-space messages. Connection-oriented and connectionless modes must be supported.
### POC References

Heartbeat: [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)

TIPC: [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)

OSDL TIPC: [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)

### ID

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1.2</td>
<td>Cluster Communication Service – Fault Handling</td>
<td>1</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

### Description:

OSDL CGL specifies that carrier grade Linux shall provide a reliable communication service that detects connection failures, aborts the connection and reports connection failures. An established connection must react to and report a problem to the application within 250 ms upon any kind of service failure, like a process or node crash.

The connection failure detection capability must offer controls that allow it to be tailored to specific conditions in different clusters. An example would be the time-outs or number of lost packets before declaring a connection failed.

### POC References

Heartbeat: [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)

TIPC: [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)

OSDL TIPC: [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)

---

**Priority 2 Requirements**
### Description:
OSDL CGL specifies that carrier grade Linux shall provide a reliable communication service that shall guarantee in-sequence (a configurable option), non-replicating, uncorrupted, and loss-free message delivery in a connection-oriented mode. In case of destination unavailability, an error indicator will be returned to the sender along with information to describe which messages that could not be delivered, an additional configurable option is for such messages to be returned to the sender.

### POC References
- **Heartbeat:** [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)
- **TIPC:** [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)
- **OSDL TIPC:** [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)

---

### Description:
OSDL CGL specifies that carrier grade Linux shall provide a fast communication service when transferring data between nodes in a cluster. It shall provide throughput and latency performance improvements when compared to the performance of TCP mechanisms. The transport protocol should take advantage of the cluster-specific physical model and must provide stable and bounded transmission delays.

### POC References
- **Heartbeat:** [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)
- **TIPC:** [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)
- **OSDL TIPC:** [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1.5</td>
<td>Cluster Communication Service – Event Notification</td>
<td>2</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a reliable communication service that can relay Inter-Process Communication (IPC) events, like failure events, to interested applications. IPC events are published by the communication service and applications that subscribe to these events are notified when an IPC event occurs.

**POC References**

Linux Asynchronous Event Mechanism (AEM) project: [http://sourceforge.net/projects/aem](http://sourceforge.net/projects/aem)

OSDL TIPC: [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFH.3.0</td>
<td>Prevent Failed Node From Corrupting Shared Resources</td>
<td>2</td>
<td>Clustering – Cluster Fault Handling</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a way to fence a failed or errant node from shared resources, such as SAN storage, to prevent the failed node from causing damage to shared resources. Since the surviving nodes in the cluster will want to failover resources, applications, and/or middleware to surviving nodes in the cluster, the cluster must make sure it is safe to do the failover.

Killing the failed node is the easiest and safest way to provide this protection. If support for a remote power supply control is available, it could be used to restart a node or to turn it off completely. However, I/O fencing using a method like fibre channel zoning or reservation could be used with shared storage devices.

The requirement is to provide a mechanism to prevent a failed node from corrupting shared resources. The cluster cannot depend on the failing node to alter the cluster when it is failing, so the cluster must be proactive in protecting shared resources.

**External Specification Dependencies:**

This capability is dependent on hardware to provide a mechanism to reset or isolate the failed node(s).

**POC References**

Shoot The Other Node In The Head (STONITH): [http://linux-ha.org/stonith.html](http://linux-ha.org/stonith.html)


Description: OSDL CGL specifies that carrier grade Linux shall provide cluster health monitoring that has the ability to detect the difference between a LAN failure and a node failure. The faster this can be done, the more applicable to a broader range of carrier grade applications, with preference for sub-second indication, although the targeted use of the cluster will affect the decision to use any specific implementation.

External Specification Dependencies:
Implementation may be dependent on a redundant health monitoring mechanism, like the use of serial lines, hardware backplane signals, or redundant LAN connections.

POC References:
Linux System Event Log (panic handler): http://panicsel.sourceforge.net/
The panic handler might try to send information to other nodes before rebooting to ease membership and node failure detection.
OpenIPMI: http://openipmi.sourceforge.net/
Ethernet bonding (link aggregation and failover) – in stable kernel

Description: OSDL CGL specifies that carrier grade Linux shall provide mechanisms for failing over applications in a cluster from one node to another. Applications and nodes (CFH.2.0) are monitored and a failover mechanism is invoked when a failure is detected. Once a failure is detected, the application failover mechanism must determine which rules/policy apply to this failover scenario and then begin the process to start a standby application or initiate the respawn of an application within 1 second.

This functionality should be integrated with, or work in conjunction with, the functionality specified by requirement 3.3 Application Heartbeat Monitor listed on page Error! Bookmark not defined..

Note: The full application failover time is dependent upon application/node failure detection, the time to apply the failover rules/policy, and the time it takes to start or restart the application. The aggregate failover time for an application must allow the cluster to maintain carrier grade application availability.

POC References
Linux-HA Heartbeat: http://www/linux-ha.org/heartbeat/
### Description:
OSDL CGL specifies that carrier grade Linux shall provide a mechanism for storage network replication. This storage network replication shall provide the following:

- A network replication layer that enables RAID-1-like disk mirroring, using a cluster-local network for multi-node replication of data.
- Multi-node resynchronization of replicated data after node failure such that replicated data remains highly available during resynchronization.

### POC References

**DRBD:**
- [http://www.complang.tuwien.ac.at/reisner/drbd/](http://www.complang.tuwien.ac.at/reisner/drbd/)

**Network Block Device (NBD):**
- [http://www.it.uc3m.es/~ptb/nbd/](http://www.it.uc3m.es/~ptb/nbd/)
### CSM.4.0  
**Name:** Shared Storage Consistent Access  
**Priority:** 2  
**Category:** Clustering – Cluster Storage Management  
**Security Level:** N/A  
**Maturity:** None  
**Default/Toggle:** Off/Yes

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a consistent shared storage access method from different nodes to ensure partition information isn't changed on one node while a partition is in use on another node that would prevent the change. This protects administrators from accidentally changing a partition size when they believe all nodes in the cluster are no longer using the disk or partitions that surround the partition in question, which would result in data corruption.

**POC References**  
None

### CSM.5.0  
**Name:** Shared Storage Mirroring  
**Priority:** 2  
**Category:** Clustering – Cluster Storage Management  
**Security Level:** N/A  
**Maturity:** Started  
**Default/Toggle:** Off/Yes

**Description:** OSDL CGL specifies that carrier grade Linux shall provide cluster-wide data mirroring with shared storage. Each node in the cluster with access to this shared storage must have the same view of mirrored storage. Shared storage must be able to be managed by any node in the cluster. Mirrored shared storage can be provided by either a hardware RAID-1 architecture storage subsystem or by a software RAID-1 implementation.

**POC References**  
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV.1.0</td>
<td>Cluster-wide Resource Monitor</td>
<td>2</td>
<td>Clustering – Cluster Services</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a means to access clusters resources from a centralized location to facilitate easy performance analysis of the whole cluster and statistics collection.

**POC References:**

PCP offers remote access and tools to retrieve data, but it does not have the notion of a cluster. Remote access is configured to an arbitrary set of nodes.


<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV.2.0</td>
<td>Cluster-wide Identified Core Dump</td>
<td>2</td>
<td>Clustering – Cluster Services</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster-aware core dump that uniquely identifies which node produced the core dumps. For instance, if a diskless node dumps core files to network storage, the core dump will be uniquely identified as originating from that node.

**POC References**

Core dump file name control: [http://www.sinz.org/Michael.Sinz/Linux/core.html](http://www.sinz.org/Michael.Sinz/Linux/core.html)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV.3.0</td>
<td>Cluster-wide Crash Dump Management</td>
<td>2</td>
<td>Clustering – Cluster Services</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster-aware kernel crash dump that uniquely identifies which node produced the crash dump. For instance, if a diskless node dumps crash data to network storage, the data will be uniquely identified as originating from that node.

**POC References**


Software Suspend – the combination of power management and software suspend could be leveraged for a “mainline” kernel dump mechanism.


<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV.4.0</td>
<td>Cluster-wide Log Collection</td>
<td>2</td>
<td>Category – Cluster Services</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster-wide logging mechanism. A cluster-wide log shall contain node identification, message type, and cluster time identification. This cluster-wide log may be implemented as a central log or as the collection of specific node logs.

**POC References**


# Priority 3 Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.2</td>
<td>Cluster Message Service (SA Forum AIS)</td>
<td>3</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster messaging service that conforms to the SA Forum Application Interface Specification (AIS) and is certified in accordance with the SA Forum AIS certification process.

**POC References**
Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.2.1</td>
<td>Cluster Messaging Service</td>
<td>3</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>None</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster messaging service that conforms to the SA Forum Application Interface Specification (AIS). The Message Service specifies a buffered message-passing system based on the concept of a message queue. Load balancing is also supported via message queue groups. With message queues, the Message Service uses the model of n senders to one receiver whereas, with message queue groups, the Message Service uses the model of M senders to N receivers.

**External Specification Dependencies:**

**POC References**
Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.2.2</td>
<td>SA Forum AIS Certification – Messaging Service</td>
<td>3</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>None</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide an implementation of the Message Service supplied with CGL shall be certified in accordance with the SA Forum AIS certification process.

Note that the SA Forum has not yet established an AIS certification process.

**POC References**
Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)
6 Security Requirements

This section contains requirements specific to supporting security. These requirements are new in this version of the OSDL “Carrier Grade Linux Requirements Specification”.

### PRIORITY 1 REQUIREMENTS

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUT.1.0</td>
<td>Password Integrity Checking</td>
<td>1</td>
<td>Security – Authentication</td>
<td>2</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide tools to check passwords to ensure they cannot be easily cracked. These tools shall support at least the DES cipher text format and allow the user to specify rules for rejecting passwords.

**POC References**

Pluggable Authentication Module (PAM):
- [http://pam.sf.net](http://pam.sf.net)

is a prerequisite for the two PAM module projects below.


pam_cracklib PAM module:
- [http://pam.sf.net](http://pam.sf.net)


Software does not appear to be licensed, except that it is free, but donations are allowed.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.1.0</td>
<td>Support for Generic Kernel Security Modules</td>
<td>1</td>
<td>Security – Access Control</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide full Linux Security Module (LSM) support. The LSM project provides a lightweight, general-purpose framework for access control.

In order to allow security modules to be added or replaced based on the security requirements of the end-user, a generic kernel framework must be provided. Of the methods available, the LSM project appears to be flexible enough and has a good chance of being placed in the main kernel stream. This framework allows a wide variety of security models to be implemented in Linux with numerous modules already available. Note that LSM does not implement any access control mechanisms by itself but rather provides hooks for them to be added.

**POC References**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.1.0</td>
<td>IPSec for IPv4</td>
<td>1</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide IPsec support to provide network level confidentiality and integrity for IPv4 (it is already mandatory for IPv6). The implementation shall conform to RFC 2401, 2402, 2406 and any relevant algorithm. Specific RFCs, for example RFC 2451. Please note that this requirement refers to ESP/AH, not to key management, which is addressed in a separate requirement.

**POC References:**
IPSec is supported in Linux kernel 2.5.49 and higher.
### ID: CON.2.0  Support for IKE

<table>
<thead>
<tr>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide an Internet Key Exchange (IKE) daemon to perform standards-based key exchange for IPsec. The daemon shall conform to RFC 2409.

This key exchange mechanism will permit different implementations and platforms to interoperate and is critical in many situations to securely perform IPsec.

**POC References**
The Linux 2.5.49 and higher implementations work with KAME IKE daemon.


### ID: CON.5.0  PF_KEY Support

<table>
<thead>
<tr>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Experimental</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide PF_KEY support (RFC 2367) for key management for the IPsec module and the IKE daemon. For full IPsec support, PF_KEY must be extended to support IPsec specific features. As explicit standards for PF_KEY policy and extensions have not been established, it is recommended that the extensions used by KAME in the NetBSD implementation be used for CGL.

**POC References**
The Linux 2.5.49 and higher implementations work with KAME IKE daemon, which supports PF_KEY.
### Description
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable a periodic checking of the integrity of files at user-level. Files to be checked are both binary files, which should not change after installation, and text files, such as configuration files, which may change.

The implementation may involve adding a digital signature to the binary files.

File integrity checks shall be able to be scheduled at any time of the day. The checking mechanism shall be able to send alarms to security authorities when inconsistencies, such as invalid signatures, are detected.

### POC References
- Tripwire: [http://www.tripwire.org](http://www.tripwire.org) and [http://sourceforge.net/projects/tripwire](http://sourceforge.net/projects/tripwire)
- Bsign: [http://packages.debian.org/testing/admin/bsign.html](http://packages.debian.org/testing/admin/bsign.html)

### Description
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to check log files to ensure they have not been tampered with (message integrity). In addition, OSDL CGL specifies that carrier grade Linux shall provide a mechanism to verify the source of a log message received (origin authentication).

Syslog-sign (currently in IETF draft status) provides integrity over any transport.

### POC References
- SDSC Secure Syslog: [http://security.sdsc.edu/software/sdsc-syslog](http://security.sdsc.edu/software/sdsc-syslog)
- Modular Syslog (Msyslog): [http://sourceforge.net/projects/msyslog](http://sourceforge.net/projects/msyslog)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUD.2.0</td>
<td>Confidentiality Of Log Information</td>
<td>1</td>
<td>Security – Auditing</td>
<td>4</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide secure transport of log information to the log files.

A variety of methods can be used to achieve this:

- syslog (RFC 3195) over TLS
- syslog-ng daemon over TCP, allowing the use of stunnel (to tunnel any TCP connection over SSL) or ssh port forwarding
- IPsec

**POC References**

SDSC Secure Syslog: [http://security.sdsc.edu/software/sdsc-syslog](http://security.sdsc.edu/software/sdsc-syslog)


<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/ Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUD.3.0</td>
<td>Automated Log Analysis</td>
<td>1</td>
<td>Security – Auditing</td>
<td>2</td>
<td>Experimental</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism for automatically analyzing log files. This mechanism shall be able to generate alarms if any suspicious log is detected.

**POC References**


LogDog: [http://caspian.dotconf.net/menu/Software/LogDog/](http://caspian.dotconf.net/menu/Software/LogDog/)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.11.0</td>
<td>PKI CA Support</td>
<td>1</td>
<td>Security – Confidentiality and Integrity</td>
<td>2</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a basic PKI CA, which shall conform to the IETF PKIX standards, specifically RFC 2527, 3279 & 3280. Certification Revocation List (CRL) Support is required, although a specified delivery mechanism is not specified. HTTP/FTP (RFC 2585) is recommended at a minimum. Certificate Management/Request protocols are not a requirement.

**POC References**

OpenSSL [http://www.openssl.org/](http://www.openssl.org/) plus one of:

- OpenCA project: [http://sourceforge.net/projects/openca/](http://sourceforge.net/projects/openca/)
- pyCA: [http://www.pyca.de/](http://www.pyca.de/)
- TinyCA ([http://tinyca.sm-zone.net/](http://tinyca.sm-zone.net/))
### PRIORITY 2 REQUIREMENTS

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUT.2.0</td>
<td>Support For One Time Passwords</td>
<td>2</td>
<td>Security - Authentication</td>
<td>2</td>
<td>Experimental</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism that allows passwords that are different at each login as an alternative to the traditional Unix fixed-password authentication mechanism. Potential implementations may be a one-way function, like OPIE, or a hardware token-based one-time password mechanism, like RSA SecurID.

**POC References**

- One Time Passwords in Everything (OPIE): [http://inner.net/opie](http://inner.net/opie)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.2.0</td>
<td>Support Of Capabilities For File Systems</td>
<td>2</td>
<td>Security – Access Control</td>
<td>3</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide the “file system capabilities” functionality that is described in the POSIX 1003.1e draft. The POSIX 1003.1e draft categorizes the various access checks, such as file permission override, network configuration, and raw device access, into a discrete set of groups called capabilities. Implementation in CGL need not comply with the draft, but must implement the functionality defined in the draft.

Note that POSIX 1003.1e is a POSIX draft that has been withdrawn. Currently, the Linux kernel implements capabilities for processes but not for file systems. By default, all the capabilities are available to root-owned processes. However, capabilities can be removed from the process. This allows a least privilege approach for the root-owned process.

**POC References**

- Patch for 2.5.60: [http://www.olafdietsche.de/linux/capability/](http://www.olafdietsche.de/linux/capability/)
- Inserting capabilities info into the ELF files: [http://www.goop.org/~jeremy/caps/](http://www.goop.org/~jeremy/caps/)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.3.0</td>
<td>Buffer Overflows Protection Mechanisms</td>
<td>2</td>
<td>Security – Access Control</td>
<td>3</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide at least one mechanism to protect against the exploitation of software bugs that give an attacker arbitrary access to a task’s address space by using buffer overflows. Possible counter mechanisms may be a non-executable stack, kernel stack randomization user-space stack randomization, and compiler mechanisms.

**POC References**
PaX kernel Patch: [http://pageexec.virtualave.net/](http://pageexec.virtualave.net/)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.4.0</td>
<td>Access Control List Support for File Systems</td>
<td>2</td>
<td>Security – Access Control</td>
<td>3</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide the access control list (ACL) capabilities on file systems described in the POSIX 1003.1e draft. Implementation need not comply with the draft but must implement the functionality defined in the draft. This may be implemented with Linux file systems that support extended attributes that can be used to store ACLs.

Note that some restrictions apply when ACLs are used with NFS and backup mechanisms.

**POC References**
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.6.0</td>
<td>Secure Integrity Verification Of Binaries Before Loading Them</td>
<td>2</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide the capacity to check the integrity of the binary before loading it into the memory. In the case that the verification fails, the system must not load the binary.

Refer to the requirement CON.3.0 “Secure Integrity Check At User-level, to see the need for security integrity checks.

The signatures for binaries shall be verified each time they are loaded from hard disk, such as when execv is executed.

**POC References**

CryptoMark: [http://www.immunix.org/cryptomark.html](http://www.immunix.org/cryptomark.html)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.7.0</td>
<td>Generic Hardware Crypto Device Support</td>
<td>2</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for generic hardware crypto devices. These generic hardware crypto devices may assist either kernel components, such as IPsec or CIFS, or user-space applications, such as SSL or IKE, by unloading costly cryptographic operations from the system.

**POC Referrals**


<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.8.0</td>
<td>IPSec Hardware Support</td>
<td>2</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide an IPsec module that has support for a cryptographic API callout for external (to the module) cryptographic support.

**POC References**

### Description:
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to verify the signature of the kernel image and other configuration files downloaded when booting the system. PKI does not need to be supported as part of this requirement. The certificates for the boot server shall be stored locally on the booting system in its compact flash memory or on a floppy or CD used to boot.

**POC References**

Linux Secure Network Boot: [http://www.linux.ericsson.ca/dsi/secure_boot](http://www.linux.ericsson.ca/dsi/secure_boot)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.9.0</td>
<td>Secure Boot Signed Kernel Image</td>
<td>2</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable security of core dumps, including ensuring that only authorized users have access to core dump information and ensuring the system supports confidentiality for the core dumps generated. Note that fine-grained access control mechanisms must already be in place for this requirement to be implemented.

**POC References**


LKCD provides a reliable method for detecting, saving, and examining system crashes. Core dumps are currently not secure, but this project could be augmented to allow for secure core dumps.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUD.4.0</td>
<td>Support For Event Logging Mechanisms</td>
<td>2</td>
<td>Security – Auditing</td>
<td>3</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide fine-grained events logging for non-repudiation purposes. User account information must be logged for each transaction, making it possible to trace the actions of each individual user, identify the originators of messages and provide proof of transmission and receipt.

**POC References**


### PRIORITY 3 REQUIREMENTS

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.5</td>
<td>Mandatory Access Control</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide Mandatory Access Control (MAC) functionality as opposed to the Discretionary Access Control (DAC) functionality currently on Linux systems.

In DAC, it is at the discretion of the user to define access control privileges for user resources. In MAC, the system administrator defines the access control policy of the system. Because of this, users cannot grant privileges that can be used as an exploit. When used correctly, this will restrict the privileges of the super-user to create a more secure Linux system.

**POC References**
See the POC References in the sub-requirements below.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.5.1</td>
<td>MAC Based On Domain-Type Approach</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide MAC functionality using a domain-type approach. A domain-type approach divides a system into two logical entities: subjects and objects. Subjects are active entities, such as processes, and objects are passive entities, like files. A domain is associated with a subject and a type is associated with an object. Access-control permissions are associated with both domains and types, including domain-domain permissions, like create and kill, and domain-type permissions, such as read, write, and execute.

**POC References**
Domain and Type Enforcement (DTE) for Linux: [http://www.cs.wm.edu/~hallyn/dte/](http://www.cs.wm.edu/~hallyn/dte/)
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.5.2</td>
<td>MAC Based On Role-Based Access Control</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide MAC functionality using role-based access control.

Role-based access control is a form of mandatory access control where a user is granted or denied access to an object based on the roles the user performs in an organization. Users cannot pass on this access to others at their discretion.

**POC References**


<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.5.3</td>
<td>MAC Based On Access Control Lists</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide MAC functionality using Access Control Lists (ACL). ACL is a form of mandatory access control that uses control lists that state which access privileges, such as read, write, or execute, a user has on a given system object, like files and directories. Users cannot modify these access lists at their discretion.

**POC References**


SubDomain: [http://www.immunix.org/subdomain.html](http://www.immunix.org/subdomain.html)
### ACC.5.4 Process Level Granularity

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.5.4</td>
<td>Process Level Granularity</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide process level granularity for authentication and access control. Support for MAC is a pre-requisite for this requirement.

Current Linux security mechanisms are based on user privileges and do not support authentication checks for interactions between two processes belonging to the same user. CGL contains systems dedicated to running a few predefined applications, generally for long periods of time, and hosting only a few users with predefined roles. With Linux security mechanisms, all of these processes would have the same security privileges. For example, the same access control privileges would be given to the statistical software component and the database handling software component. Therefore, if intruders could breach into statistical software, they could modify database contents, too.

Process level granularity for authentication and access control will allow authentication and authorization between different tasks accessing different resources.

**POC References**


---

### ACC.6.0 Pre-emptive Security Mechanisms

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC.6.0</td>
<td>Pre-emptive Security Mechanisms</td>
<td>3</td>
<td>Security – Access Control</td>
<td>4</td>
<td>None</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide pre-emptive security mechanisms. That is, any changes in the security context shall be reflected immediately on the running security services. Whenever the security context of a subject changes, the system shall re-evaluate its current use of resources against this new security context.

**POC References**

None
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON.12.0</td>
<td>Secure Boot Communications</td>
<td>3</td>
<td>Security – Confidentiality and Integrity</td>
<td>3</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable secure communications at boot time to download files, such as configuration files, authentication files, and access control privilege information.

**POC References**

Linux Secure Network Boot: [http://www-linux.ericsson.ca/dsi/secure_boot](http://www-linux.ericsson.ca/dsi/secure_boot)
7 Requirements Carried Forward Unchanged From Version 1.1 to Version 2.0

These requirements were all designated priority 1 (P1) in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1”.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.1</td>
<td>IPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.2</td>
<td>IPSECv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.3</td>
<td>MIPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Hot Plug Insertion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Hot Plug Deletion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.2</td>
<td>Remote Boot Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.4</td>
<td>Loading Proprietary Modules</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.5</td>
<td>Diskless Systems</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Serial Console Connection</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.21</td>
<td>Watchdog Time Interface</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.3</td>
<td>Application Heartbeat Monitor</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Ethernet Link Aggregation</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Ethernet Link Failover</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.5.1</td>
<td>RAID 1 Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.6</td>
<td>Resilient Filesystem Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.7</td>
<td>Disk and Volume Management</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.3</td>
<td>Kernel Message Structuring</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.4</td>
<td>Dynamic Debug / Probe Insertion</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>Platform Signal Handler</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.6</td>
<td>Remote Access to Event Log</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>User-Level (gdb) Debug Support for Threads</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Kernel Debugger</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>6.2</td>
<td>RAID 0 Support</td>
<td>1</td>
<td>Performance</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>2.8</td>
<td>Hyperthreading of CPUs</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
<tr>
<td>3.2.2</td>
<td>Watchdog Timer Pre-Timeout Interrupt</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Multithreaded Core Dump Support for Threaded Applications</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
</tbody>
</table>
8 Requirements in Version 1.1 Not Included In Version 2.0

These requirements were included in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1” but are not currently considered as CGL requirements. However, they may serve as a basis for future requirements.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Device Driver Hardening Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5.7</td>
<td>Debugging Support for Fork</td>
<td>N/A (P3 in v.1.1)</td>
<td>Tools</td>
<td>N/A</td>
<td>N/A</td>
<td>On/No</td>
</tr>
<tr>
<td>35.2</td>
<td>Drivers Required to be Hardened</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2.1.3.1</td>
<td>System Device Enumeration Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Sample Hardened Device Driver</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Resource Monitor Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Serviceability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P1 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.6</td>
<td>Kernel I/O Performance Analysis</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

9 Requirements That Have Been Deferred
The following requirements have been deferred to a future version of this document. Deferred requirements are listed to indicate features felt to be useful to be specified for carrier grade Linux, but lack sufficient analysis at this time to be listed with a specific priority. These act as markers to ensure consideration of these items in the next release of the OSDL CGL requirements specification.

### 9.1 Deferred Clustering Requirements

This first set of the features described in these requirements are useful and necessary clustering capabilities but are considered out of scope for this version. These features could still be provided as part of a supplier's value-added offering.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV.dlm</td>
<td>Cluster Resource Locking</td>
<td>N/A</td>
<td>Clustering</td>
<td>N/A</td>
<td>Started</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a locking mechanism for multiple cluster nodes to compete for and safely access shared resources in a cluster.

**External Specification Dependencies:**

SA Forum AIS – Chapter 10 (SA Lock Service API)

**POC References:**


A distributed lock manager (DLM) provides advisory locking services used to coordinate access to any arbitrary shared resource. A DLM does not enforce good sharing behavior, but rather provides cooperating applications with information regarding the state of a shared resource. All locks are advisory, that is, voluntary. The system does not enforce locking. Instead, applications running on the cluster must cooperate for locking to work. An application that wants to use a shared resource is responsible for first obtaining a lock on that resource before attempting to access it. Applications and services that can benefit from using a distributed lock manager are transaction-oriented, such as a database, a file system, or a resource controller/manager.
### CSV.ev

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster event service that enables asynchronous communication between event publishers and event subscribers. The cluster event service is local to a cluster node, but may subscribe to clustering events, such as membership changes, failure notifications, or resource state changes.

**External Specification Dependencies:**

- SA Forum AIS – Chapter 8 (SA Event Service API)

**POC References:**


### CSM.cfs

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a cluster-wide file system. A clustered file system must allow simultaneous access to shared files by multiple computers. Node failure must be transparent to file system users on all surviving nodes. A clustered file system must provide the same user API and semantics as a file system on private, single-node storage.

**POC References:**

- Lustre: [https://projects.clusterfs.com/lustre](https://projects.clusterfs.com/lustre)

## 9.2 Deferred Fast System Start Requirements

These requirements are intended to describe a set of capabilities to allow rapid booting and initialization of a system running carrier grade Linux. It is felt that although these are quite valuable, the necessary level of diligence to understand and describe them has not yet been accomplished. These and additional requirements for these capabilities will be addressed in better detail and care in the next major version of the OSDL CGL requirements document.
### Description:
OSDL CGL specifies that carrier grade Linux shall provide a variety of capabilities to allow a single system to move from power-on to ready for applications in as short a time as possible.

Startup sequence divided up as follows:

- **power-on**
  - Boot
- Linux loading –[taken care of already]
- Linux started (COLD: BIOS -> OS handoff, WARM: OS -> OS handoff)
  - Linux start
- init started (OS -> init handoff)
  - init start
- applications started

Startup types can be divided as follows:

- Warm restart = restart in which the memory of the node is kept intact (basically requires kexec/bootimg/...).
- Cold restart = a restart in which the memory of the node is not kept.
- Intact (ye olde reboot).

The ways with which data about the system is passed on to the OS can be divided as follows:

- Static = based on hardwired data, eg. the boot loader informing the OS that no SCSI devices are connected to the node.
- Dynamic = based on data amassed by a previous instance of itself, e.g. persistent copy of sysfs that describes the SCSI devices connected to the node. Dynamic behavior is feasible only in a warm restart.

### POC References
See specific references in the sub-requirements.
### Description
OSDL CGL specifies that carrier grade Linux shall provide mechanisms that allow for a fast boot sequence (e.g. without RAM checking) that can complete its tasks in 20 seconds. This fast sequence is complemented by a slow sequence in which any number or required diagnostics can be run. There exists a mechanism with which the boot software can decide whether to run the fast or slow sequence.

### POC References
- LinuxBIOS: [http://www.linuxbios.org/index.html](http://www.linuxbios.org/index.html)

---

### Description
OSDL CGL specifies that carrier grade Linux shall provide for faster methods to perform probing of SCSI adapters and devices. Possible methods include running the probes in parallel or decreasing the timeouts (which are quite long to cater for slow legacy devices).

### POC References
None.

---

### Description
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to speed up operating system initialization. The increase in speed shall be implemented in a static fashion, i.e. the behavior of the operating system shall not be determined by the behavior of a previous instance, rather than kernel parameters.

### POC References
The parameter interface is in plain vanilla kernel, but implementations of useful speed increases need work. The boot loader (e.g. LILO) can be enhanced to carry the device topology information.
### Description:
OSDL CGL specifies that carrier grade Linux shall provide a mechanism to speed up operating system initialization. The increase in speed shall be implemented in dynamic fashion, i.e., the behavior of the operating system shall be determined by data stored by a previous instance.

### POC References
None.

#### 9.3 Deferred Hardware Requirements

These requirements are incomplete references to classes of hardware applicable to carrier grade server installations. It is felt that the discussion of hardware platforms needs to be expanded and more carefully considered than has been possible to do for this version of the document. These and additional requirements for various hardware platforms will be addressed in better detail and care in the next major version of the OSDL CGL requirements document.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.atca</td>
<td>Advanced TCA IPMI Support</td>
<td>N/A</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that Linux shall provide support an Advanced TCA (aTCA) hardware platform when it is installed and executing on an appropriate platform. aTCA features are not required to be simulated or provided on non-aTCA platforms. It is the responsibility of applications that are sensitive to platform differences to detect and handle the differences.

**POC References**
None.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.atca.1</td>
<td>Advanced TCA IPMI Support</td>
<td>N/A</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that Linux shall provide programming APIs specific to the new IPMI commands, data structures, and sensors defined in the aTCA specification.

**POC References**
None.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLT.atca.2</td>
<td>Advanced TCA Block Device Removal</td>
<td>N/A</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that Linux shall allow removal of a block device while it is in use without degrading the reliability of the system. The block device shall be removable even if in use by an open file (fdisk /dev/sda), is a member of raid 1 volume, or a file system is mounted on the device, or permutations thereof. If a file is in use and it cannot be serviced by a mirrored disk, the operating system shall return an error to the system calls referencing that file.

**POC References**
None.
### Description:
OSDL CGL specifies that Linux shall ensure that the shutdown system call uses the ATCA system management IPMI interface to power down the cpu blade and light the hotswap indicator.

### POC References
None.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSM.atca</td>
<td>Advanced TCA Multiple Host Synchronized Device Hotswap</td>
<td>N/A</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that Linux shall support that when multiple hosts are using the same block or character device, and a user requests to remove the device, the device's blade won't be powered off and if a hotswap indicator available, lit, until all operating systems in the collection of cpu nodes using the device have removed all references to the device in the operating system.

### POC References
None.

### 9.4 Deferred Diagnostics Requirements

It has been widely discussed that online and offline diagnostics capabilities are essential for carrier grade environments. For that reason, this requirement has existed since the initial OSDL CGL requirements specification. However, we do not yet have sufficient detailed information to support specifying this as a prioritized requirement, and this will be handled more thoroughly in the next major release of the OSDL CGL requirements specification.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_diags.0</td>
<td>Online Diagnostics</td>
<td>N/A (P2 in v.1.1)</td>
<td>Serviceability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide an online diagnostics capability that supports a set of system health checks that can be run to determine system health, including operating system health and hardware health. The diagnostic check capability must be implemented so that it can be run remotely or locally.

Online diagnostics generally refers to diagnostic checks that run while the system is in service. They may execute at various times, such as:

- At the behest of a system administrator
- When triggered by pre-defined events
- As part of handling failures, like core dumps or panics

Offline diagnostics may be executed when a system is out of service, such as after it has been rebooted from a failure but before it is reintegrated into a cluster of running machines. A key difference between the offline and online diagnostic environment is that an online environment places stricter requirements on the diagnostics framework to minimize resource consumption and unpredictable effects. At the very least, these impacts must be well-documented.

**POC References**

Linux Diagnostics project: http://linux-diag.sourceforge.net/
Appendices

A.1 General References

The Carrier Grade Linux Web page on OSDL Web site: http://www.osdl.org/projects/cgl

A.2 General Systems References

POSIX:
- http://www.opengroup.org/
- http://www.unix.org/online.html
- http://www.opengroup.org/onlinepubs/007908799/
- http://posixtest.sf.net for more POSIX conformance data on Linux.

Linux Standard Base, Free Standards Group:
- http://www.linuxbase.org/
- http://www.freestandards.org/

Service Availability Forum:
- http://www.saforum.org/

Open Cluster Framework:
- http://www.opencf.org

IETF:

A.3 Clustering References

• Service Availability Forum: http://www.saforum.org/
• Open Cluster Framework: http://www.opencf.org
• OSDL Cluster Framework: http://osdcluster.sourceforge.net/

OSDL Carrier Grade Linux Clustering Model Version 1.0 white paper:
http://www.osdl.org/docs/cgl_clustering_model_whitepaper.doc

A.4 Security References

• RFC 3195 - Reliable Delivery for syslog, http://www.faqs.org/rfc/rfc3195.html
• POSIX 1003.1e - http://wt.xpilot.org/publications/posix.1e/download.html
A.5  CGL Changes Between 2.0 and 2.0.1

A.5.1  Preface

Most standards include some minor defects that make them un-implementable. The Carrier Grade Linux Requirements Definition 2.0 includes some of these minor defects. The defects in the Carrier Grade Linux Requirements Definition 2.0 are corrected in the full document of the Carrier Grade Linux Requirements Definition 2.0.1. This appendix serves as a guide to those changes for those that have read the Carrier Grade Linux Requirements Definition 2.0. The Proof of Concept committee of the CGL working group has created three classes of changes to the Carrier Grade Linux Requirements Definition 2.0 that are binding on the registration process.

There are a significant number of reference errors in the CGL 2.0. These errors reference the OSDL Requirements Definition Version 1.1 specification incorrectly. A distribution may choose to register without the binding effects of the reference corrections.

The second class of changes are developer notes. The developer notes were created to address situations in which a reasonable implementation that implements the spirit of the requirements exists, but implements a slightly newer draft or ratified standard or technology then defined in the Carrier Grade Linux Requirements Definition V2.0. A distribution may choose to register without the binding effect of the developer notes.

The final class of changes are erratum. The proof of concept committee has defined erratum as spelling errors, grammatical errors, difficulty in understanding the requirement, or any requirement that cannot be implemented or makes the requirements un-registerable. A distribution must register with the binding effect of the erratum.

Each changed erratum requirement is presented with the original text and the new text. The page numbers of the original text from the Carrier Grade Linux Requirements Definition 2.0 are presented to aid in reference. Finally a rationale is provided that explains the thought process of the Proof of Concept Committee of the CGL-WG.

Steven Dake
Proof of Concept Committee Chair, CGL-WG
sdake@mvista.com
A.5.2 Incorrect References in Carrier Grade Linux Requirements Definition 2.0

A.5.2.1 Requirement “PLT.1.0” – Page 32 of the CGL 2.0 Requirements
The ID field references 2.1.0 in v1.1 spec. The correct reference is 2.1.3 in the v1.1 spec.

A.5.2.2 Requirement “AVL3.1” – Page 35 of the CGL 2.0 Requirements
The ID field references 3.8.1 in v1.1 spec. There is no valid reference to the v1.1 spec.

A.5.2.3 Requirement “AVL.3.2” – Page 36 of the CGL 2.0 Requirements
The ID field references 3.8.2 in v1.1 spec. The correct reference is 3.8.1 in the v1.1 spec.

A.5.2.4 Requirement “AVL.3.3” – Page 36 of the CGL 2.0 Requirements
The ID field references 3.8.3 in v1.1 spec. The correct reference is 3.8.2 in the v1.1 spec.

A.5.2.5 Requirement “AVL.3.4” – Page 37 of the CGL 2.0 Requirements
The ID field references 3.8.4 in v1.1 spec. The correct reference is 3.8.3 in the v1.1 spec.

A.5.2.6 Requirement “AVL3.5” – Page 37 of the CGL 2.0 Requirements
The ID field references 3.8.5 in v1.1 spec. There is no valid reference to the v1.1 spec.

A.5.2.7 Requirement “AVL.3.6” – Page 38 of the CGL 2.0 Requirements
The ID field references 3.8.6 in v1.1 spec. The correct reference is 3.8.4 in the v1.1 spec.

A.5.2.8 Requirement “AVL.3.7” – Page 38 of the CGL 2.0 Requirements
The ID field references 3.8.3 in v1.1 spec. The correct reference is 3.8.4 in the v1.1 spec.

A.5.2.9 Requirement “AVL3.8” – Page 39 of the CGL 2.0 Requirements
The ID field references 3.8.8 in v1.1 spec. There is no valid reference to the v1.1 spec.

A.5.2.10 Requirement “AVL.4.0” – Page 40 of the CGL 2.0 Requirements
The ID field references 3.10.0 in v1.1 spec. The correct reference is 4.10 in the v1.1 spec.

A.5.2.11 Requirement “PLT.4.0” – Page 46 of the CGL 2.0 Requirements
The ID field references 2.3.0 in v1.1 spec. The correct reference is 2.3 in the v1.1 spec.

A.5.2.12 Requirement “PLT.5.0” – Page 46 of the CGL 2.0 Requirements
The ID field references 2.6.0 in v1.1 spec. The correct reference is 2.6.2 in the v1.1 spec.

A.5.2.13 Requirement “AVL.8.0” – Page 50 of the CGL 2.0 Requirements
The requirement’s description references AVL.7.1. The correct reference is AVL.8.1.

A.5.2.14 Requirement “SVC.2.0” – Page 54 of the CGL 2.0 Requirements
The ID field references 4.1.0 in v1.1 spec. The correct reference is 4.9.3 in the v1.1 spec.

A.5.2.15 Requirement “SVC.5.1” – Page 56 of the CGL 2.0 Requirements
The ID field references 4.12.1 in v1.1 spec. There is no valid reference to the v1.1 spec.
A.5.2.16 Requirement “SVC.5.2” – Page 56 of the CGL 2.0 Requirements
The ID field references 4.12.2 in v1.1 spec. The correct reference is 4.12 in the v1.1 spec.

A.5.2.17 Requirement “PRF.3” – Page 57 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 6.3 in the v1.1 spec.

A.5.2.18 Requirement “PRF.3.1” – Page 57 of the CGL 2.0 Requirements
The ID field references 6.3.1 in v1.1 spec. The correct reference is 6.3 in the v1.1 spec.

Requirement “PRF.3.2” – Page 57 of the CGL 2.0 Requirements
The ID field references 6.3.2 in v1.1 spec. There correct reference is 6.3 in the v1.1 spec.

“A.5.2.19 Requirement “TLS.1.0” – Page 58 of the CGL 2.0 Requirements
The ID field references 5.2.0 in v1.1 spec. The correct reference is 5.2 in the v1.1 spec.

A.5.2.20 Requirement “TLS.2.1” – Page 59 of the CGL 2.0 Requirements
The ID field references 5.4.1 in v1.1 spec. The correct reference is 5.4.2 in the v1.1 spec.

A.5.2.21 Requirement “TLS.2.2” – Page 59 of the CGL 2.0 Requirements
The ID field references 5.4.2 in v1.1 spec. The correct reference is 5.4.3 in the v1.1 spec.

A.5.2.22 Requirement “PRF.4.0” – Page 62 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 6.9 in the v1.1 spec.

A.5.2.23 Requirement “PRF.5.0” – Page 62 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 6.10 in the v1.1 spec.

A.5.2.24 Requirement “PRF.8.0” – Page 65 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 6.8 (P2 in v1.1) in the v1.1 spec.

A.5.2.25 Requirement “CFH.5.0” – Page 72 of the CGL 2.0 Requirements
The ID field references 9.17.0 in v1.1 spec. The correct reference is 3.9 in the v1.1 spec.
The requirement’s description references page 64. The correct reference is page 94, ID 3.3.

A.5.2.26 “Section 8, ID 3.1” – Page 95 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 3.1.1 in the v1.1 spec.

A.5.2.27 Requirement “CSV.dlm” – Page 96 of the CGL 2.0 Requirements
The ID field references 9.11.0 in v1.1 spec. There is no valid reference to the v1.1 spec.
A.5.2.28 Requirement “CSV.ev” – Page 96 of the CGL 2.0 Requirements
The ID field references 9.12.0 in v1.1 spec. There is no valid reference to the v1.1 spec.

A.5.2.29 Requirement “CSM.cfs” – Page 97 of the CGL 2.0 Requirements
The ID field references 9.13.0 in v1.1 spec. The correct reference is 3.12 in the v1.1 spec.

A.5.2.30 Requirement “AVL.fast.1” – Page 99 of the CGL 2.0 Requirements
The ID field doesn’t reference the v1.1 spec. The correct reference is 4.7 (P2 in v1.1) in the v1.1 spec.

A.5.2.31 Requirement “SVC.diags.0” – Page 103 of the CGL 2.0 Requirements
The ID field references 4.9.0 in v1.1 spec. The correct reference is 4.9.2 in the v1.1 spec.
A.5.3   Developer Notes for Carrier Grade Linux Requirements Definition 2.0

A.5.3.1 Requirement “SVC.1.4” - Page 44 of the CGL 2.0 Requirements

A.5.3.1.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC.1.4</td>
<td>SNMP IPv6 MIBs</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide required support for all IPv6 MIBs as specified by IETF IPv6 WG except for a very primitive level support for the general MIB group as described in the documents below:

- IP Version 6 MIB for the Transmission Control Protocol (RFC 2452)
- IP Version 6 MIB for the User Datagram Protocol (RFC 2454)
- MIB for IP Version 6: Textual Conventions and General Group (RFC 2465)
- MIB for IP Version 6: ICMPv6 Group (RFC 2466)

This means that internally within the kernel IPv6 stack, statistics and/or information corresponding to MIB data must be gathered and saved for ultimate retrieval by an SNMP Agent.

**POC References**

NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)
## Description

OSDL CGL specifies that carrier grade Linux shall provide required support for all IPv6 MIBs as specified by IETF IPv6 WG except for a very primitive level support for the general MIB group as described in the documents below:

- IP Version 6 MIB for the Transmission Control Protocol (RFC 2452)
- IP Version 6 MIB for the User Datagram Protocol (RFC 2454)
- MIB for IP Version 6: Textual Conventions and General Group (RFC 2465)
- MIB for IP Version 6: ICMPv6 Group (RFC 2466)

This means that internally within the kernel IPv6 stack, statistics and/or information corresponding to MIB data must be gathered and saved for ultimate retrieval by an SNMP Agent.

*There is currently an ongoing effort within IETF to combine IPv4 and IPv6 MIBs into unified MIBs. The developer may choose to implement RFC2011, RFC2012, RFC2013 instead of RFC2452, RFC2454, RFC2465, and RFC2466.*

## New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC.1.4</td>
<td>SNMP IPv6 MIBs</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**POC References**

NET-SNMP: [http://www.netsnmp.org](http://www.netsnmp.org)
A.5.4 Erratum and Their Corrections of Carrier Grade Linux Requirements Definition 2.0

A.5.4.1 Requirement “STD.2.20” – Page 32 of CGL 2.0 Requirements

A.5.4.1.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.20</td>
<td>Thread Safe Functions</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSF (thread-safe functions) margin code in the POSIX specification. This functionality is used by POSIX Timers.

POC References

See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

A.5.4.1.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.2.20</td>
<td>Thread Safe Functions</td>
<td>2</td>
<td>Standards</td>
<td>N/A</td>
<td>Experimental</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide functionality marked by the TSF (thread-safe functions) margin code in the POSIX specification.

POC References

See discussion about “Linux: Native POSIX Threading Library (NPTL)” at [http://kerneltrap.org/node.php?id=422](http://kerneltrap.org/node.php?id=422)

See additional documentation about NPTL at [http://people.redhat.com/drepper/](http://people.redhat.com/drepper/)

A.5.4.1.3 Rationale

An error was made during spec creation believing that asctime_r, ctime_r, gmtime_r, and localtime_r were part of the POSIX spec. As they are not, there is no need to reference these “thread-safe” functions in the specification.
### A.5.4.2 Requirement “STD.4.0” – Page 32 of CGL 2.0 Requirements

#### A.5.4.2.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.4.0 (1.8 in v1.1)</td>
<td>Stream Control Transport Protocol (SCTP)</td>
<td>1 (P2 in v1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for the following core Stream Control Transmission Protocol (SCTP) IETF RFCs:

- RFC 2960
- RFC 1112

**POC References**

Linux kernel greater than 2.5.39

#### A.5.4.2.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.4.0 (1.8 in v1.1)</td>
<td>Stream Control Transport Protocol (SCTP)</td>
<td>1 (P2 in v1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide support for the following core Stream Control Transmission Protocol (SCTP) IETF RFCs:

- RFC 2960

**POC References**

Linux kernel greater than 2.5.39

#### A.5.4.2.3 Rationale

RFC 1112 is not a SCTP RFC.
A.5.4.3 Requirement “AVL.3.1” - Page 35 of CGL 2.0 Requirements

A.5.4.3.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.1</td>
<td>Software Remote Update and Installation</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

Description: OSDL CGL specifies that carrier grade Linux shall provide software remote update and installation capability allowing all software on the system to be installed, updated, and configured remotely over the network.

- The nodes can be remotely booted using network booting.
- The software images can be maintained on a server that has network access to the nodes to be installed but may otherwise be close to or far remote from the target nodes.

POC References
Distribution-specific installation tools:
- RedHat - Kickstart
- SuSE – YaST

A.5.4.3.2 New Requirement
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.1</td>
<td>Software Remote Update and Installation</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software remote update and installation capability allowing all software on the system to be installed, updated, and configured remotely over the network.

- The nodes can be remotely booted using network booting.
- The software images can be maintained on a server that has network access to the nodes to be installed but may otherwise be close to or far remote from the target nodes.

**POC References**


Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE - YaST

**A.5.4.3.3 Rationale**

The extra bullet in the description of the original requirement was not intended by the Carrier Grade Linux Working Group and is removed.
A.5.4.4 Requirement “AVL.11.0” - Page 52 of CGL 2.0 Requirements

A.5.4.4.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.11.0</td>
<td>Checkpointing</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a data checkpointing service. This service provides contextual state replication on a single node or across cluster nodes and provides service continuity upon node failure. The targets for data checkpointing are applications, not middleware. The checkpointing service shall include the following:

- Data checkpoints that contain user-specified data.
- The same API should work for a single node as well as when replicating checkpoints across nodes. In essence, a single node checkpoint is one with no replicas.
- A checkpoint needs to persist if the process using it dies, so that a recovering process can grab it. The checkpoints should either have a time-to-live expiration time limit and be automatically reclaimed, or provide a mechanism (c.f. ipcrm) to clean up old ones.
- A checkpoint is not persistent across a reboot of a node. Thus, unless it has a replica on another node, rebooting the node causes it to disappear.
- A capability to replicate checkpoints in the cluster either synchronously or asynchronously.
- Checkpoints that are transactional (no partial checkpoints) to provide atomicity (either the data unit is completely checkpointed or it is not checkpointed at all).
- Checkpoints need to be secured such that only approved users can access them, e.g., based on uid and/or gid of the creator.
- Checkpoint read/write access time that is less than 5 ms for each 4K page for two nodes.

Checkpoint integrity is provided by the checkpointing service.

**External Specification Dependencies:**

The Service Availability Forum Application Interface Specification (SA Forum AIS) addresses data checkpointing (SA Forum AIS - Chapter 7) and may be leveraged to define the interface for this feature.

**POC References**

Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)

Data checkpointing project as part of Linux-ha, see linux-ha-dev mailing list, available through [http://linux-ha.org](http://linux-ha.org). This will use the Service Availability Forum Checkpoint Service API for guidance.
### A.5.4.4.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.11.0</td>
<td>Checkpointing</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Started</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a data checkpointing service. This service provides contextual state replication on a single node or across cluster nodes and provides service continuity upon node failure. The targets for data checkpointing are applications, not middleware. The checkpointing service shall include the following:

- Data checkpoints that contain user-specified data.
- The same API should work for a single node as well as when replicating checkpoints across nodes. In essence, a single node checkpoint is one with no replicas.
- A checkpoint needs to persist if the process using it dies, so that a recovering process can grab it. The checkpoints should either have a time-to-live expiration time limit and be automatically reclaimed, or provide a mechanism (c.f. IPC) to clean up old ones.
- A checkpoint is not persistent across a reboot of a node. Thus, unless it has a replica on another node, rebooting the node causes it to disappear.
- A capability to replicate checkpoints in the cluster either synchronously or asynchronously.
- Checkpoints that are transactional (no partial checkpoints) to provide atomicity (either the data unit is completely checkpointed or it is not checkpointed at all).
- Checkpoints need to be secured such that only approved users can access them, e.g., based on uid and/or gid of the creator.
- Writing a checkpoint from one node, and reading on another node for 4K of checkpoint data should take less then 5 ms.

Checkpoint integrity is provided by the checkpointing service.

**External Specification Dependencies:**

The Service Availability Forum Application Interface Specification (SA Forum AIS) addresses data checkpointing (*SA Forum AIS - Chapter 7*) and may be leveraged to define the interface for this feature.

**POC References**

Service Availability Forum: [http://www.saforum.org](http://www.saforum.org)

Data checkpointing project as part of Linux-ha, see linux-ha-dev mailing list, available through [http://linux-ha.org](http://linux-ha.org). This will use the Service Availability Forum Checkpoint Service API for guidance.

### A.5.4.4.3 Rationale

The original requirement did not clearly explain the desired performance characteristics of the checkpointing service. Grammatical errors should be resolved to reduce confusion of the specification.
**A.5.4.5 Requirement “AVL.12.0” - Page 52-53 of CGL 2.0 Requirements**

### A.5.4.5.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.12.0</td>
<td>Multi-Path Access to Storage</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable multiple access paths from a single cluster node to storage devices. The software shall determine if multiple paths exists to the same port of the I/O device and with configurable controls it will balance I/O requests across multiple host bus adapters. If multiple paths exist to the same device over two separate device ports on the same host bus adapter, those I/Os will not be balanced.

Handling a path failure must be automatic. A mechanism must be provided for the reactivation of failed paths, which can be placed back into service. It must be possible to automatically determine and configure multiple paths. Automatic configuration shall allow automatic multi-path configuration of complete disks and partitions located on those disks.

**POC References**

The latest stable Linux kernel currently supports multi-path I/O via the md layer. The problem with this approach is that there is easy way to configure it, and incorrect configuration can lead to data loss.

IBM has an open source project for SCSI mid-level multipath implementation. The documentation for this project is at:

http://www-124.ibm.com/storageio/multipath/scsi-multipath/docs-0.1/

The uSDE project (see PLT.1.0 on page 34) provides automatic configuration of the MD for multipath disk access.

http://source.mvista.com/sde

---

### A.5.4.5.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.12.0</td>
<td>Multi-Path Access to Storage</td>
<td>2</td>
<td>Availability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>
**Description:** OSDL CGL specifies that carrier grade Linux shall provide a mechanism to enable multiple access paths from a single cluster node to storage devices. The software shall determine if multiple paths exist to the same port of the I/O device and with configurable controls it will balance I/O requests across multiple host bus adapters. If multiple paths exist to the same device over two separate device ports on the same host bus adapter, those I/Os will not be balanced.

Handling a path failure must be automatic. A mechanism must be provided for the reactivation of failed paths, which can be placed back into service. It must be possible to automatically determine and configure multiple paths. Automatic configuration shall allow automatic multi-path configuration of complete disks and partitions located on those disks.

**POC References**

The latest stable Linux kernel currently supports multi-path I/O via the md layer. The problem with this approach is that there is no easy way to configure it, and incorrect configuration can lead to data loss.

IBM has an open source project for SCSI mid-level multipath implementation. The documentation for this project is at:

http://www-124.ibm.com/storageio/multipath/scsi-multipath/docs-0.1/

The uSDE project (see PLT.1.0 on page 34) provides automatic configuration of the MD for multipath disk access.

http://source.mvista.com/sde

**A.5.4.5.3 Rationale**

Grammatical errors should be resolved to reduce confusion of the specification.
A.5.4.6 Requirement “SVC.2.0” - Page 54 of CGL 2.0 Requirements

A.5.4.6.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC.2.0</td>
<td>CIM</td>
<td>2</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a diagnostics framework and shall provide interfaces to support third-party CIM Provider and CIM Object Manager (CIMOM) applications so these applications can use the CGL framework to perform diagnostic functions for available system devices. CIM Provider and CIMOM specifications are based on the following:

- CIM Specifications V2.2 or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Schema V2.6 or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Diagnostic Model Specification V 1.012 or higher (available at [http://www.dmtf.org/education/whitepapers.php](http://www.dmtf.org/education/whitepapers.php))

**POC References**


A.5.4.6.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC.2.0</td>
<td>CIM</td>
<td>2</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a diagnostics framework and shall provide interfaces to support third-party CIM Provider and CIM Object Manager (CIMOM) applications so these applications can use the CGL framework to perform diagnostic functions for available system devices. CIM Provider and CIMOM specifications are based on the following:

- CIM Specifications V2.2 or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Schema **V2.8** or higher (available at [http://www.dmtf.org/standards](http://www.dmtf.org/standards))
- CIM Diagnostic Model Specification V 1.012 or higher (available at [http://www.dmtf.org/education/whitepapers.php](http://www.dmtf.org/education/whitepapers.php))

**POC References**

A.5.4.6.3 Rationale

The description included a grammatical error. The WBEMsource POC reference refers to an initiative, and as it is not a POC reference, has been removed.

A.5.4.7 Requirement “CCM 1.2” - Page 68 of CGL 2.0 Requirements

A.5.4.7.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1.2</td>
<td>Cluster Communication Service – Fault Handling</td>
<td>1</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a reliable communication service that detects connection failures, aborts the connection and reports connection failures. An established connection must react to and report a problem to the application immediately upon any kind of service failure, like a process or node crash.

The connection failure detection capability must offer controls that allow it to be tailored to specific conditions in different clusters. An example would be the time-outs or number of lost packets before declaring a connection failed.

**POC References**

Heartbeat: [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)
TIPC: [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)
OSDL TIPC: [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)
### A.5.4.7.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM.1.2</td>
<td>Cluster Communication Service – Fault Handling</td>
<td>1</td>
<td>Clustering – Cluster Communication and Messaging</td>
<td>N/A</td>
<td>Experimental</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a reliable communication service that detects connection failures, aborts the connection and reports connection failures. An established connection must react to and report a problem to the application *within 250 ms* upon any kind of service failure, like a process or node crash.

The connection failure detection capability must offer controls that allow it to be tailored to specific conditions in different clusters. An example would be the time-outs or number of lost packets before declaring a connection failed.

**POC References**
- **Heartbeat:** [http://linux-ha.org/heartbeat](http://linux-ha.org/heartbeat)
- **TIPC:** [http://sourceforge.net/projects/tipc](http://sourceforge.net/projects/tipc)
- **OSDL TIPC:** [http://developer.osdl.org/dev/tipc](http://developer.osdl.org/dev/tipc)

### A.5.4.7.3 Rationale

The term immediately is measured as zero seconds. An OSDL distribution could not comply with this requirement since it would be impossible to detect a failure in zero seconds.
### A.5.4.8 Section 7 - Requirements Carried Forward Unchanged From Version 1.1 to Version 2.0 – page 94 of CGL 2.0 Requirements

#### A.5.4.8.1 Original Section

These requirements were all designated priority 1 (P1) in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1”.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.1</td>
<td>IPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.2</td>
<td>IPSECV6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.3</td>
<td>MIPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Hot Plug Insertion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Hot Plug Deletion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.2</td>
<td>Remote Boot Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.4</td>
<td>Loading Proprietary Modules</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.5</td>
<td>Diskless Systems</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Serial Console Connection</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.21</td>
<td>Watchdog Time Interface Requirements</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.3</td>
<td>Application Heartbeat Monitor</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Ethernet Link Aggregation</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Ethernet Link Failover</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.5.1</td>
<td>RAID 1 Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.6</td>
<td>Resilient Filesystem Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.7</td>
<td>Disk and Volume Management</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.3</td>
<td>Kernel Message Structuring</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.4</td>
<td>Dynamic Debug / Probe Insertion</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>Platform Signal Handler</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.6</td>
<td>Remote Access to Event Log</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>User-Level (gdb) Debug Support for Threads</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Kernel Debugger</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>6.2</td>
<td>RAID 0 Support</td>
<td>1</td>
<td>Performance</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>
A.5.4.8.2  **New Section**

These requirements were all designated priority 1 (P1) in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1”.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.1</td>
<td>IPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.2</td>
<td>IPSECv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>1.4.3</td>
<td>MIPv6</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Hot Plug Insertion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Hot Plug Deletion</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.2</td>
<td>Remote Boot Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.4</td>
<td>Loading Proprietary Modules</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.5</td>
<td>Diskless Systems</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Serial Console Connection</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.21</td>
<td>Watchdog Time Interface Requirements</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.3</td>
<td>Application Heartbeat Monitor</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/Yes</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Ethernet Link Aggregation</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Ethernet Link Failover</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.5.1</td>
<td>RAID 1 Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.6</td>
<td>Resilient Filesystem Support</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>3.7</td>
<td>Disk and Volume Management</td>
<td>1</td>
<td>Platform</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.3</td>
<td>Kernel Message Structuring</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.4</td>
<td>Dynamic Debug / Probe Insertion</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>Platform Signal Handler</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>4.6</td>
<td>Remote Access to Event Log</td>
<td>1</td>
<td>Serviceability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>User-Level (gdb) Debug Support for Threads</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Kernel Debugger</td>
<td>1</td>
<td>Tools</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>6.2</td>
<td>RAID 0 Support</td>
<td>1</td>
<td>Performance</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
<tr>
<td>2.8</td>
<td>Hyperthreading of CPUs</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
<tr>
<td>3.2.2</td>
<td>Watchdog Timer Pre-Timeout Interrupt</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Multithreaded Core Dump Support for Threaded Applications</td>
<td>1</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
<td></td>
</tr>
<tr>
<td>Version 1.1 ID</td>
<td>Name</td>
<td>Rationale</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.3.1</td>
<td>Hyperthreading of CPUs</td>
<td>This was an oversight of the CGL group. This feature has since become mainlined in 2.4, 2.6 Linux kernels.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2.2</td>
<td>Watchdog Timer Pre-Timeout Interrupt</td>
<td>This was an oversight of the CGL group. This feature has since become mainlined in 2.4, 2.6 Linux kernels.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Multithreaded Core Dump Support for Threaded Applications</td>
<td>This was an oversight of the CGL group. This feature has since become mainlined and is definitely required.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A.5.4.9 Section 8 - Requirements in Version 1.1 Not Included In Version 2.0 – page 95 of CGL 2.0 Requirements

A.5.4.9.1 Original Section

These requirements were included in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1” but are not currently considered as CGL requirements. However, they may serve as a basis for future requirements.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Device Driver Hardening Specification</td>
<td>N/A</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5.7</td>
<td>Debugging Support for Fork</td>
<td>N/A</td>
<td>Tools</td>
<td>N/A</td>
<td>N/A</td>
<td>On/No</td>
</tr>
<tr>
<td>35.2</td>
<td>Drivers Required to be Hardened</td>
<td>N/A</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
A.5.4.9.2 New Section

These requirements were included in the OSDL “Carrier Grade Linux Requirements Specification, Version 1.1” but are not currently considered as CGL requirements. However, they may serve as a basis for future requirements.

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Device Driver Hardening Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5.7</td>
<td>Debugging Support for Fork</td>
<td>N/A (P3 in v.1.1)</td>
<td>Tools</td>
<td>N/A</td>
<td>N/A</td>
<td>On/No</td>
</tr>
<tr>
<td>35.2</td>
<td>Drivers Required to be Hardened</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2.1.3.1</td>
<td>System Device Enumeration Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Platform</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Sample Hardened Device Driver</td>
<td>N/A (P1 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Resource Monitor Specification</td>
<td>N/A (P1 in v.1.1)</td>
<td>Serviceability</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P1 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Concurrent Thread Scaling Behavior and Report</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Enhancements for High Concurrent Timer Scaling</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6.6</td>
<td>Kernel I/O Performance Analysis</td>
<td>N/A (P2 in v.1.1)</td>
<td>Performance</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### A.5.4.9.3 Rationale

<table>
<thead>
<tr>
<th>Version 1.1 ID</th>
<th>Name</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.3.1</td>
<td>System Device Enumeration Specification</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Sample Hardened Device Driver</td>
<td>Requirement for driver hardening specification removed in CGL 2.0.</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Resource Monitor Specification</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Concurrent Timers Scaling Behavior and Report</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Concurrent Timers Scaling Behavior and Report</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Concurrent Timers Scaling Behavior and Report</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Enhancement for High Concurrent Timer Scaling</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Enhancement for High Concurrent Timer Scaling</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Enhancement for High Concurrent Timer Scaling</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Enhancement for High Concurrent Timer Scaling</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
<tr>
<td>6.6</td>
<td>Kernel I/O Performance Analysis</td>
<td>The CGL Requirements Definition 2.0 excluded drafting of requirements documents for individual features or documents describing behavior.</td>
</tr>
</tbody>
</table>
A.6  CGL Changes Between 2.0.1 and 2.0.2

A.6.1  Preface

The Carrier Grade Linux Requirements Definition 2.0.1 was the first 2.0.x specification that was used for registration. During the process of registration, clarification was requested on the Software Live Installation and Upgrade requirements. Clarifications were added to:

- AVL.3.1 Software Remote Update and Installation
- AVL.3.2 Software Live Upgrade Minimal Reboot
- AVL 3.3 Software Live Upgrade RPM Version Check
- AVL 3.4 Software Live Upgrade Log

Compliance testing also changes over time. STD.1.0 (Linux Standard Base Compliance) was changed to reflect that LSB 1.3 will cease to available for LSB compliance testing and that the registration process would accept LSB 1.3 or later LSB compliance versions.

John Cherry
CGL Roadmap Coordinator
cherry@osl.org

A.6.2  Requirement “STD.1.0”- Page 21 of the CGL 2.0 Requirements

A.6.2.1  Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.1.0</td>
<td>Linux Standard Base Compliance</td>
<td>1 (P1 in v.1.1)</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a system supporting CGL requirements shall remain compliant with the Linux Standard Base (LSB) 1.3. Some of the features outside the current LSB definition may become part of subsequent LSB definitions.

**POC References**
Linux Standard Base: http://www.linuxbase.org/

A.6.2.2  New Requirement
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD.1.0</td>
<td>Linux Standard Base Compliance</td>
<td>1</td>
<td>Standards</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a system supporting CGL requirements shall remain compliant with the Linux Standard Base (LSB) 1.3 or later. Some of the features outside the current LSB definition may become part of subsequent LSB definitions.

**POC References**

---

**A.6.3 Requirement “AVL.3.1”- Page 35 of the CGL 2.0 Requirements**

**A.6.3.1 Original Requirement**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.1</td>
<td>Software Remote Update and Installation</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software remote update and installation capability allowing all software on the system to be installed, updated, and configured remotely over the network.

- The nodes can be remotely booted using network booting.
- The software images can be maintained on a server that has network access to the nodes to be installed but may otherwise be close to or far remote from the target nodes.

**POC References**
Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE - YaST

---

**A.6.3.2 New Requirement**
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.1</td>
<td>Software Remote Update and Installation</td>
<td>1 (P2 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>Off/Yes</td>
</tr>
</tbody>
</table>

**Name Clarification:** Remote Package Update and Installation

**Description:** OSDL CGL specifies that carrier grade Linux shall provide a remote software package update feature. It shall include functions that allow kernel modules and application software to be installed or upgraded remotely, while minimizing downtime of the system.

The usage of the term “remotely” does not imply a central package management platform, nor does it preclude such a system. This requirement only necessitates that a single device may be upgraded without requiring the administrator to be physically at the device.

Due to the wide range of platforms and applications, CGL does not specify a specific downtime limit metric. Downtime targets will vary based on the system application.

**POC References**


Distribution-specific installation tools:

- RedHat - Kickstart
- SuSE - YaST

A.6.4 Requirement “AVL.3.2”- Page 36 of the CGL 2.0 Requirements

A.6.4.1 Original Requirement
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.2</td>
<td>Software Live Upgrade Minimal Reboot</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

### Description:
OSDL CGL specifies that carrier grade Linux shall provide remote software installation and upgrade mechanisms to allow software to be upgraded without increasing the number of reboots required for the software upgrade:

- No reboot shall be required for upgrade of kernel components like device drivers or kernel modules.
- Upgrades of applications in user-space software shall not require a reboot of the platform. A characteristic within the application may require a reboot, which should be specified for the application.

This requirement means remote mechanisms will not require any more reboots than the same upgrade done locally by an administrator at the console.

### POC References

- **Distribution-specific installation tools:**
  - RedHat - Kickstart
  - SuSE – YaST

**kexec:** [http://www.osdl.org/archive/andyp/bloom/Code/Linux/Kexec/](http://www.osdl.org/archive/andyp/bloom/Code/Linux/Kexec/)

### A.6.4.2 New Requirement
### Name Clarification:
No System Reboot

### Description:
OSDL CGL specifies that carrier grade Linux shall provide CGL shall ensure that remote software installation and software upgrade mechanisms that requiring no system reboots:

- No reboot shall be required to upgrade kernel module components (such as device drivers)
- No reboot shall be required to upgrade user-space applications provided by CGL system software.

It is noted that certain custom applications may require a reboot, but the CGL system software should not.

Remote software installation and upgrade mechanisms will not require any more reboots than the same upgrade done via console.

### POC References


Distribution-specific installation tools:
- RedHat - Kickstart
- SuSE – YaST


---

**A.6.5 Requirement “AVL.3.3”- Page 36 of the CGL 2.0 Requirements**

**A.6.5.1 Original Requirement**
### Description:
OSDL CGL specifies that carrier grade Linux shall provide remote software upgrade capabilities that include provisions for version compatibility and dependency checking at the RPM file level. The upgrade process shall allow the coexistence of new and old executables, shared libraries, configuration files, and data.

It is reasonable that this requirement be implemented as a combination of the installer and the chosen package management system. The RPM file format is required by the LSB.

### POC References
- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE – YaST

---

### A.6.5.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.3 (3.8.2 in v1.1)</td>
<td>Software Live Upgrade RPM Version Check</td>
<td>1 (P2 in v.1.1)</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
</tbody>
</table>

**Name Clarification:** Version and Dependency Checking via RPM

**Description:** OSDL CGL specifies that carrier grade Linux shall provide CGL remote software installation and upgrade capabilities shall that include provisions for version compatibility and dependency checking at the RPM package level.

The upgrade process shall allow the coexistence of new and old executables, shared libraries, configuration files, and data.

It is reasonable that this requirement be implemented as a combination of the installer and the chosen package management system.

### POC References
- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE – YaST
A.6.6 Requirement “AVL.3.4”- Page 37 of the CGL 2.0 Requirements

A.6.6.1 Original Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.4</td>
<td>Software Live Upgrade Log</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
<tr>
<td>(3.8.3in v1.1)</td>
<td></td>
<td>(P2 in v.1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software upgrade mechanisms that include logging of dates, times, changes and the identity of the user.

**POC References**
- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE – YaST

A.6.6.2 New Requirement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Priority</th>
<th>Category</th>
<th>Security Level</th>
<th>Maturity</th>
<th>Default/Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL.3.4</td>
<td>Software Live Upgrade Log</td>
<td>1</td>
<td>Availability</td>
<td>N/A</td>
<td>Production</td>
<td>On/No</td>
</tr>
<tr>
<td>(3.8.3in v1.1)</td>
<td></td>
<td>(P2 in v.1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Name Clarification:** Upgrade Log

**Description:** OSDL CGL specifies that carrier grade Linux shall provide CGL remote software installation and upgrade shall mechanisms that perform transaction logging of dates, times, changes and the identity of the user performing the change.

**POC References**
- Distribution-specific installation tools:
  - RedHat - Kickstart
  - SuSE – YaST