Reference Model
and
Use Cases
▪
Part 2 of 2

Developed by:
Enterprise Grid Alliance
Reference Model Working Group

*** Version 1.5 ***
As Approved by EGA

10 March 2006

This Page Intentionally Blank

Copyright Notice

Author Information

Contributor	***Organization***
Mukund Buddhikot	Sun Microsystems, Inc.
Alan Clark	Novell, Inc.
Michael Enescu	Cassatt Corporation
Brajesh Goyal	Network Appliance, Inc.
Julian Hammersley	Advanced Micro Devices, Inc.
Masato Kataoka	NEC Solutions (America), Inc.
Raj Kumar	Hewlett-Packard Company
David Pearson	Oracle Corporation
Parviz Peiravi	Intel Corporation
Stephen Schleimer	Cisco Systems, Inc.
Michael B Schmitz	Oracle Corporation
Harald Skardal	Network Appliance, Inc.
Paul Strong	Sun Microsystems, Inc.
Kazunori Sudo	NEC
Bob Thome--editor	Oracle Corporation
Vinod Viswanathan	Optena Corporation

Revision History

Version	Date	Comments
1.5	10 March 2006	Created separate document for extensive set of new use cases.

Contents

Figures

Preface

This document is the second of two parts which together comprise the EGA Reference Model.

The first part contains the **Introduction**, **EGA Reference Model, Generic Use Cases, Glossary of Terms** and **Appendix A**.

This second part contains the specific **Provisioning Use Cases**.

Using the reference model as a foundation, the use cases in this document capture the processes and actors involved in the management of the lifecycle of various grid components. These use cases provide the basis for sharing enterprise grid requirements and context with the EGA's partners – vendors, end users and standards bodies. The use cases are intended to show how to derive basic specific use cases around management scenarios for simple grid components and how these may then be used and extended to derive more complex scenarios and use cases.

It should be noted that the purpose of the use cases is to capture processes that have a large manual component today, so that they may be automated in future. These use cases are expected to form the basis of the work carried out by various EGA working groups.

Comments regarding content and format are welcome, and may be directed to the EGA Reference Model working group at ega_referencemodelwg@mail.gridalliance.org.

Additional Notes

1) In the following use cases, when grid components are discovered, configured and started, their existence, state etc. are stored within the GME Repository. No assumptions are made about the repository within which the GME stores its data. The GME Repository, like the GME itself, is a single logical entity. It may be implemented as a single database instance, or as set of database instances within discrete components of the GME. It may be implemented as a federation of such databases.

2) The GME repository is directly analogous to the ITIL Configuration Management Data Base (CMDB).

3) Within the use cases, we describe actors who participate in the use cases. These actors are describe the roles found in today's typical data centers. As the tasks described in these use cases are automated, many of these roles may change. They may either be subsumed by the GME, or may be combined into more general roles that do not need the same level of technical expertise. The table below describes the roles within the use cases:

Actor Name	Role definition
Data Center Admin	Data Center Admin is responsible for establishing, coordinating, and maintaining a set of operational activities and services to enforce SLA and polices ensuring that the technical architecture supports the users expectations and requirements
System Admin	A system administrator is responsible for the OS installations, configurations and maintenance such as startups, shutdowns, data backup and restoration, administration of accounts and users, operational monitoring, and software installation

Data Base Admin (DBA)	Database Administrator is responsible for the operation and administration of the DBMS, and the creation of physical database designs and maintenance
Network Admin	Network Administrator is responsible for design and operation of network infrastructure such as installing , configuring, maintaining and troubleshooting network device and components
Storage admin	Storage administrator is responsible in developing storage management program which includes allocating storage volumes; enabling quotas; managing capacity plans, changing LUN sizes; security, zoning, and backup
Hardware technician	Hardware Technician, is responsible for hardware installation, configuration and maintenance
Security Admin	Security Admin is responsible for the implementation and management of the security policies, rules and procedures.
Gird Management Entity (GME)	GME is that ***logical*** entity that manages • The grid components. • The relationships between various grid components. • Their life cycles. It is responsible for ensuring that the various grid components meet their goals, i.e. that the high level services and applications (eg ERP, BI, CRM etc.), deployed on the enterprise grid, meet their service level objectives within their operational constraints.

References

1. Strong et al, “The Enterprise Grid Alliance Reference Model v1.0”, May 2005, Enterprise Grid Alliance.
2. “Sun Fire™ Midrange Systems Platform Administration Manual, Firmware release 5.19.0”, Part No. 819-1271-10, July 2005, Sun Microsystems, Inc.
3. “Solaris 10 Installation Guide: Basic Installations”, Part No. 817-0544-11, January 2005, Sun Microsystems, Inc.
4. “Solaris 10 Installation Guide: Network-Based Installations”, Part No. 817-5504-11, January 2005, Sun Microsystems, Inc.
5. “Solaris 10 Installation Guide: Custom Jumpstart and Advanced Installations”, Part No. 817-5506-11, January 2005, Sun Microsystems, Inc.

This Page Intentionally Blank

1 Provisioning Remotely Controllable Power Supplies

1.1 Provision Remotely Controllable Power Supply

1.1.1 Summary

A remotely controlled power switch is typically an intelligent switch with facilities to connect to physical grid components such as services or switches and be part of a data centre network. It will provide capabilities for such connected grid components to be switched on/off and rebooted through corporate networks or Internet connections. It will be compliant with standard facilities such as wake-on-lan.

This use case describes the process of provisioning a remotely controllable power supply in an enterprise grid environment. The remotely controllable power supply will enable the GME to remotely turn on and off power to the attached grid component (e.g., a server, switch, storage device).

1.1.2 Dependencies

1.1.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System Admin
- GME

1.1.4 Pre-Conditions

An IP based network is available for the remotely controllable power supply to be provisioned.

1.1.5 Initiation Of Use Case

A remotely controllable power supply is added to the data centre. It is provided with the required power supply and power redundancies.

1.1.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

Step 1: The Hardware Technician connects the unit to the network.

Step 2: The Hardware Technician powers the unit on.

Step 3: The system admin using the GME of the power supply, such as data centre management console or a user interface provided by the vendor is able to detect the IP address and show its properties.

Step 4: The Hardware Technicians attaches the grid components that needs to be managed to the unit.

Step 5: The system admin using the GME is able to register the IP addresses and MAC addresses of the grid components that are attached to the unit.

Step 6: The system admin using the GME is able to power on and power off the grid components that are attached to the unit.

Step 7: The system admin adds the power supply to the GME repository.

Alternative Flows

Exceptional Flows

1.1.7 Success Criteria

The system admin using the GME can remotely power on and power off the unit.

1.1.8 Post Conditions

The system admin using the GME can power on and off the grid component attached to the remotely controllable power supply.

1.2 Decommission Remotely Controllable Power Supply

1.2.1 Summary

A remotely controlled power switch is typically an intelligent switch with facilities to connect to physical grid components such as services or switches and be part of a data centre network. It will provide capabilities for such connected grid components to be switched on /off and rebooted through corporate networks or Internet connections. It will be compliant with standard facilities such as wake-on-lan.

This use case describes the process of decommissioning a remotely controllable power supply. Once deprovisioned, the GME will no longer be able to power on and off any attached grid component. Any attached grid components must be decommissioned prior to decommissioning the power supply.

1.2.2 Dependencies

1.2.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System Admin
- GME

1.2.4 Pre-Conditions

The grid components such as servers and switches connected to the power supply are decommissioned.

1.2.5 Initiation Of Use Case

The power supply decommissioning starts after the grid components connected to the power supply are decommissioned.

1.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

Step 1: The system admin updates the GME repository with the state of the power supply, the date it was decommissioned, the reason and archives all the information available about this component.

Step 2: The Hardware Technician powers the unit off.

Step 3: The Hardware Technician disconnects the unit to the network.

Alternative Flows

Exceptional Flows

1.2.7 Success Criteria

- The remotely controllable power supply is powered off

1.2.8 Post Conditions

- The remotely controllable power supply is no longer controllable via the GME, and is not providing power to any grid components.

2 Provisioning Physical Compute Devices (Servers)

2.1 Discover Physical Compute Device (Server)

2.1.1 Summary

This use case covers both the manual and automatic discovery/creation of a single server in the enterprise grid so that it is physically present, and exists within the GME repository. An operating system may or may not have already been installed on the server and the presence of a boot device for an operating system is not assumed. The actors one would expect to interact with the enterprise grid in the traditional data center are assumed, although it is indicated in the text where the GME could be substituted.

2.1.2 Dependencies

Use Case 1.1, Provision Remotely Controllable Power Supply

2.1.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System Admin
- GME

2.1.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The data center has sufficient physical space for the server.

2.1.5 Initiation Of Use Case

A previously requested server is delivered to the data center.

2.1.6 Manual Discovery Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The Hardware Technician places the server in the data center.
2) The system admin adds a new class of server to be managed to the GME repository. This may include adding information such as –
   - Manufacturer
   - Model type
   - Capabilities of model, which may include but is not necessarily limited to number of internal power supplies, number of I/O slots, networking ports, maximum and minimum number of processors, memory and internal discs etc.
   - Other server type specific information necessary [for the GME] to manage the type of server.
3) The system admin logs the addition of a new class of server in the GME event log.
4) The system admin adds a new instance of the server to be managed to the GME repository. This act essentially adds the server to the asset register. This may include adding information such as –
   - Physical location of server
   - Serial number of server and any appropriate sub-components of the server.
   - Host ID of server (if this exists).
5) The system admin sets the server state to unconfigured in the GME repository.
6) The system admin logs the discovery of the server in the GME repository event log.

[NOTE – The GME repository is more or less synonymous with the CMDB defined within ITIL. This is relationship is described more fully in the EGA Reference Model document[1].]

Alternative Flows

Steps 2 and 3: Steps 2 and 3 may be missed if a server of this class has previously been created/discovered.

Step 4: If the server type supports hardware partitions then this capability will be included.

Exceptional Flows

If any steps fail then they are either repeated until successful or until the server is removed from the data center.

2.1.7 Automatic Discovery Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The Hardware Technician places the server in the data center.
2) The Hardware Technician adds, removes and configures any hardware as appropriate.

3) The system admin provisions one or more remotely controllable external power supplies (see section 1.1, Provision Remotely Controllable Power Supply).

4) The system admin provisions one or more remotely controllable internal power supplies (see section 1.1, Provision Remotely Controllable Power Supply).

5) The system admin or network admin configures the management network to enable access to the console, service processor or other management port of the server. The management network may be the same physical network as the general network fabric.

6) The system admin logs the management network configuration change to the GME repository event log.

7) The Hardware Technician physically connects the server to the management network. The management network may be the same physical network as the general network fabric.

8) The system admin logs the connection to the GME repository event log.

9) The system admin starts any components that are physically connected to the server and which it depends upon, for example externally attached devices, such as non-network or SAN attached disks or arrays. Note that these dependencies would be represented by another DAG.

10) The system admin logs the starting of all depended upon devices to the GME repository event log.

11) The system admin (or GME) connects to the remotely controllable external power supplies and switches them on. This could be as simple as using a GUI or as complex as having the connection via a terminal concentrator and then initiate a serial connection to the power controller and then use a command line interface to power the server up.

12) The states of the external power supplies are updated in the GME repository.

13) The system admin logs the connection of the external power supplies to the GME repository event log.

14) The system admin (or GME) connects to the remotely controllable internal power supplies and switches them on.

15) The system admin updates the state of the internal power supplies in the GME repository.

16) The system admin logs the connection of the internal power supplies and their state change in the GME repository event log.

17) The GME detects the presence of the new server.

18) The GME adds a new class of server to be managed to the GME repository. This may include adding information such as –
    - Manufacturer
    - Model type
    - Capabilities of model, which may include but is not necessarily limited to number of internal power supplies, number of I/O slots, networking ports, maximum and minimum number of processors, memory and internal discs etc.
    - Other server type specific information necessary [for the GME] to manage the type of server.
19) The GME adds a new instance of the server to be managed to the GME repository. This act essentially adds the server to the asset register. This may include adding information such as –
    - Physical location of server
    - Serial number of server and any appropriate sub-components of the server.
    - Host ID of server (if this exists).
20) The GME updates the GME repository to reflect the fully, physically configured server.
21) The GME logs the discovery of the server to the GME repository event log.
22) The system admin updates the GME repository to associate the remotely controllable external power supply/supplies with the server.
23) The system admin updates the GME repository to associate the remotely controllable internal power supply/supplies with the server.
24) The GME connects to the remotely controllable internal power supplies and switches them off.
25) The GME updates the states of the internal power supplies in the GME repository.
26) The GME logs the changes of state of the internal power supply to the GME repository event log.
27) The GME connects to the remotely controllable external power supplies and switches them off.
28) The states of the external power supplies are updated in the GME repository.
29) The GME logs the turning off of external power supplies to the GME repository event log.
30) The GME sets the server state to unconfigured in the GME repository.
31) The GME logs the change in state to the GME repository event log.

Alternate Flows

Step 3: This step may be missed if a previously provisioned external power supply is to be used.

Steps 3, 11, 12, 13, 22, 27, 28 and 29: These steps may be missed if a remotely controlled internal power supply is to be used.

Steps 4, 14, 15, 16, 23, 24, 25 and 26: These steps may be missed if there are no remotely controllable internal power supplies or if only remotely controllable external power supplies are to be used.

Other alternate flows are possible, especially for automatic discovery, for example network fabric and storage fabric connectivity may be provided prior to powering up the server.

Exceptional Flows

If any steps fail then they are either repeated until successful or until the server is removed from the data center.

2.1.8 Success Criteria

The server is physically present in the data center and logically present within the GME repository.

2.1.9 Post Conditions

- The server is within the data center.
- The server has been added to the GME repository.
- The server state is set to unconfigured.

2.2 Configure Physical Compute Device (Server)

2.2.1 Summary

Configure a physical compute device, i.e. a server, so that it is logically manageable by the GME, i.e. manageable without the need for further physical manipulation by an administrator or Hardware Technician. Configuration could include, but is not necessarily limited to, setting physical switches, adding or removing hardware, or configuring the GME in advance of starting, i.e. powering up, the server.

2.2.2 Dependencies

The use case has dependencies on the following use cases:

Section 1.1, Provision Remotely Controllable Power Supply

2.2.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

2.2.4 Pre-Conditions

- The server exists within the data center.
- The server is not powered up.
- The server has been added to the GME repository.
- The server has inherent remote management capability via a console/network connection
- If required, a remotely manageable external power switch/supply is available. This may not be required for some servers as they have remotely manageable internal power supplies or service processors that allow the main system components to be powered up and down.
- A network fabric exists.
- The management network exists. This may be the same (at least physically) as the network fabric.
- A storage fabric exists. This may be the same as the network fabric.

2.2.5 Initiation Of Use Case

A previously requested server is delivered to the data center and is to be made available as part of the enterprise grid. It has been placed in the data center and added to the GME, but has yet to be made logically manageable by the GME.

2.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The Hardware Technician adds, removes and configures any hardware as appropriate.
2) The system admin updates the GME repository to reflect the fully, physically configured server.
3) The system admin provisions one or more remotely controllable external power supplies (see section 1.1, Provision Remotely Controllable Power Supply).
4) The system admin updates the GME repository to associate the remotely controllable external power supply/supplies with the server.
5) The Hardware Technician physically connects the server to the designated external power supply/supplies.
6) The system admin provisions one or more remotely controllable internal power supplies (see section 1.1, Provision Remotely Controllable Power Supply).
7) The system admin updates the GME repository to associate the remotely controllable internal power supply/supplies with the server.
8) The network admin (or GME) configures the network fabric so that when the server is connected and powered up, it does not disrupt the network.
9) The Hardware Technician connects the server to the physical network(s).
10) The network admin (or GME) updates the network topology in the GME repository to reflect the new topology including the new server.
11) The storage admin (or GME) configures the storage fabric such that when the server is powered up it does not disrupt the fabric to which it is connected.
12) The Hardware Technician physically connects the server to storage fabric(s).
13) The storage admin (or GME) updates the storage topology in the GME repository to reflect the new topology including the new server.
14) The system admin or network admin configures the management network to enable access to the console, service processor or other management port of the server.
15) The Hardware Technician physically connects the server to the management network.
16) The system admin (or network admin or GME) adds the server to the management network topology in the GME to reflect the presence of the new server.

Alternative Flows

Steps 1 and 2: These steps may be missed if no physical reconfiguration is required.

Step 3: This step may be missed if a previously provisioned external power supply is to be used or if only an internal remotely controllable power supply is to be used.

Steps 3, 4 and 5: These steps may be missed if the server was automatically discovered, i.e. it is implied that to be discovered, the server must already have been connected to

Steps 6 and 7: These steps may be missed if only an external remotely controlled power supply exists or is to be used.

Steps 11 to 13: If the storage fabric is not separate from the network fabric then these steps may be missed.

Step 15: This step may be missed if there is no physically separate management network.

Steps 14, 15 and 16: These steps may be missed if they have already occurred as part of automatic discovery.

Exceptional Flows

If any steps fail then they are either repeated until successful or until the server is destroyed/removed from the data center.

2.2.7 Success Criteria

The server is fully connected to the various physical fabrics and the GME is fully configured.

2.2.8 Post Conditions

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server has been added to the GME repository.
- The server is manageable by the GME.

2.3 Start (Power-On) Physical Compute Device (Server)

2.3.1 Summary

This use case captures the starting or powering on of a physical compute device (server). Nothing is assumed about any subsequent flow that may automatically follow powering up, such as the booting of an operating system. The server is assumed to be powered down and configured prior to starting.

2.3.2 Dependencies

None

2.3.3 Actor(s)/Stakeholder(s)

- System admin
- GME

2.3.4 Pre-Conditions

The physical compute device, or server, has completed both the Discover Physical Compute Device (Server) use case (see section 2.1) and the Configure Physical Compute Device (Server) use case (see section 2.2). Thus

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server has been added to the GME configuration.
- The server is manageable by the GME.
- The server is not powered up.

2.3.5 Initiation Of Use Case

The physical compute device, or server must be powered on to be made available as part of the enterprise grid.

The power-on functionality could be used as a scheduled power management task. For example, an actor schedules the power-on operation of compute devices, or servers every Monday morning and power-off operation of compute devices or servers every Friday night to save energy costs during the weekend.

2.3.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin (GME) starts any components that are physically connected to the server and which it depends upon, for example externally attached devices, such as non-network or SAN attached disks or arrays. Or the system admin (GME) request that they be started.
2) The system admin (or GME) connects to the remotely controllable external power supplies and switches them on. This could be as simple as using a GUI or as complex as having the connect to a terminal concentrator and then initiate a serial connection to the power controller and then use a command line interface to power the server up.
3) The state of the remotely controllable external power supplies is updated in the GME repository.
4) The system admin (or GME) connects to the remotely controllable internal power supplies and switches them on.
5) The state of the remotely controllable internal power supplies is updated in the GME repository.
6) The system admin (or GME) updates the server state in the GME repository to active.

Alternative Flows

Steps 2 and 3: These steps may be missed if a remotely controlled internal power supply is to be used.

Steps 4 and 5: These steps may be missed if there are no remotely controllable internal power supplies or if only remotely controllable external power supplies are to be used.

Exceptional Flows

If any of the steps fail then the all completed steps are reversed to leave the server and its GME representation in its original state.

2.3.7 Success Criteria

The server is powered up and manageable via the management network.

2.3.8 Post Conditions

- The server is powered up.

2.4 Provision Physical Compute Device (Server)

2.4.1 Summary

Provision a single server to the enterprise grid so that it is physically connected to the network, storage and management fabrics, and is logically manageable. An operating system may or may not have already been installed on the server and the presence of a boot device for an operating system is not assumed. The actors one would expect to interact with the enterprise grid in the traditional data center are assumed, although it is indicated in the text where the GME could be substituted. This is an example of the provisioning a physical grid component.

2.4.2 Dependencies

This use case depends upon the following use cases –

- 2.1, Discover Physical Compute Device (Server)
- 2.2, Configure Physical Compute Device (Server)
- 2.3, Start (Power-On) Physical Compute Device (Server)

2.4.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

2.4.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The data center has sufficient power, cooling and physical space for the server.

2.4.5 Initiation Of Use Case

A previously requested server is delivered to the data center and is to be made available as part of the enterprise grid.

2.4.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Discover the server – see section 2.1.
2) Configure the server – see section 2.2.
3) Start (power-up) the server – see section 2.3.

Alternative Flows

Step 1: If the server has already been discovered/created then this step may be missed.

Steps 1 and 2: If the server has already been fully configured then steps 1 and 2 may be missed.

Steps 1: If the server is already running and needs to be reconfigured or re-provisioned then the stop physical compute device (server) and unconfigure physical compute device (server) use cases are called prior to beginning at step 2.

Exceptional Flows

If any steps fail then they are either repeated until successful or until the server is removed from the data center.

2.4.7 Success Criteria

The server is powered up and manageable via the management network by the GME.

2.4.8 Post Conditions

The grid component is active and available for the GME to deploy other grid components or services onto. This may include a login service. A login environment may thus be considered to be another service, and thus a grid component that depends on the operating system grid component, which needs to be separately provisioned.

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server is powered up.
- The server has been added to the GME repository.
- The server is manageable by the GME.

2.5 Stop Physical Compute Device (Server)

2.5.1 Summary

This use case captures the stopping of a server. Nothing is assumed about its state prior to the stop (power off) request, other than that the server has been successfully provisioned at some point in the past. For example the server may or may not have an operating system, a hypervisor or other grid components, such as an application, bound to it or running on it. This use case assumes that the GME has determined that the powering down of the server is allowable and that the unbinding or re-provisioning/remapping of all of the grid components that directly or indirectly depend upon it should be carried out.

2.5.2 Dependencies

None.

2.5.3 Actor(s)/Stakeholder(s)

- System admin
- GME

2.5.4 Pre-Conditions

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server has been added to the GME configuration.
- The server is manageable by the GME.
- The server is powered up.

2.5.5 Initiation Of Use Case

The physical server must be powered down for some reason. The reason may range from the need to replace a failed physical component, or physically reconfigure the server or connections to the various fabrics, to decommissioning of the server.

2.5.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin (GME) requests that each of the grid components that are bound to the server, if any, is stopped and potentially unconfigured. This may include, but is not necessarily limited to virtual machine monitors and operating systems.
2) The system admin (GME) connects to the remotely controllable internal power supplies and switches them off.
3) The system admin (GME) updates the state of the internal power supplies in the GME repository.
4) The system admin (GME) connects to the remotely controllable external power supplies and switches them off.
5) The system admin (GME) updates the states of the external power supplies in the GME repository.
6) The system admin (GME) sets the state of the server to configured in the GME repository.

Alternative Flows

Step 1: Step 1 may be missed if all of the grid components that depend upon it have already been stopped and/or unconfigured.

Steps 2 and 3: These steps may be missed if there are no remotely controllable internal power supplies or if only remotely controllable external power supplies are to be used.

Steps 4 and 5: These steps may be missed if a remotely controlled internal power supply is to be used.

Exceptional Flows

If a step fails then repeat until successful.

2.5.7 Success Criteria

The server is powered off.

2.5.8 Post Conditions

The server is powered off.

2.6 Unconfigure Physical Compute Device (Server)

2.6.1 Summary

This use case captures the unconfiguring of a server. The server has already been powered down (stopped).

2.6.2 Dependencies

This use case's primary flow assumes that the server has been stopped/powered down (see section 2.5, Stop Physical Compute Device (Server)).

2.6.3 Actor(s)/Stakeholder(s)

- System admin
- Storage Admin
- GME

2.6.4 Pre-Conditions

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server has been added to the GME configuration (ITIL CMDB).
- The server is manageable by the GME.
- The server is powered down.

2.6.5 Initiation Of Use Case

The physical server must be unconfigured for some reason. The reason may range from the need to replace a failed physical component, or physically reconfigure the server or connections to the various fabrics, to decommissioning of the server. Removing the existing configuration, rather then simply overwriting it may be a security requirement, so as to leave no trace of the previous configuration or identity.

2.6.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin disconnects all of the remotely controllable external power supplies from the server.
2) The system admin updates the GME repository to dissociate the remotely controllable external power supply/supplies from the server.
3) The system admin updates the GME repository to dissociate the remotely controllable internal power supply/supplies from the server.
4) The network admin (or GME) configures the network fabric so that when the server is disconnected, it does not disrupt the network.
5) The system admin disconnects the server from the physical network(s).
6) The network admin (or GME) updates the network topology in the GME repository to reflect the new topology excluding the decommissioned server.
7) The storage admin (or GME) configures the storage fabric such that when the server is removed it does not disrupt the fabric to which it is connected.
8) The storage admin physically disconnects the server from the storage fabric(s).
9) The storage admin (or GME) updates the storage topology in the GME repository to reflect the new topology excluding the decommissioned server.
10) The system admin or network admin configures the management network to remove access to the console, service processor or other management port of the server.
11) The system admin physically disconnects the server from the management network.
12) The system admin (or network admin or GME) removes the server from the management network topology in the GME to reflect the absence of the server.
13) The Hardware Technician adds, removes and configures any hardware as appropriate.
14) The system admin updates the GME repository to reflect the fully, physically unconfigured server and status of its removed components (if any).

Alternative Flows

Steps 1 and 2: These steps may be missed if only an internal remotely controllable power supply was used.

Step 3: This step may be missed if only an external remotely controlled power supply was used or managed.

Steps 7, 8 and 9: If the storage fabric is not separate from the network fabric then these steps may be missed.

Step 11: This step may be missed if there is no physically separate management network.

Steps 13 and 14: These steps may be missed if no physical deconfiguration is required.

Exceptional Flows

If any steps fail then they are either repeated until successful or until the server is destroyed/removed from the data center.

2.6.7 Success Criteria

The server is physically disconnected from the network, storage fabric, management network and power management fabric, and is in the unconfigured state.

2.6.8 Post Conditions

- The server is not physically connected to the power fabric.
- The server is not physically connected to the network fabric.
- The server is not physically connected to the storage fabric.
- The server is not physically connected to the management fabric.
- The server is no longer manageable by the GME.
- The server is powered down.
- The server is in the unconfigured state within the GME repository.

2.7 Destroy Physical Compute Device (Server)

2.7.1 Summary

This use case captures the removal of a server from the data center. This does not remove it from the GME repository as it is assumed that records of removed grid components and their history within the enterprise grid are retained.

2.7.2 Dependencies

None

2.7.3 Actor(s)/Stakeholder(s)

- System admin
- GME

2.7.4 Pre-Conditions

- The server is in an unconfigured state.
- The server is not physically connected to the power fabric.
- The server is not physically connected to the network fabric.
- The server is not physically connected to the storage fabric.
- The server is not physically connected to the management fabric.
- The server is no longer manageable by the GME.
- The server is powered down.
- The server is in the unconfigured state within the GME repository.

2.7.5 Initiation Of Use Case

A decision is made to remove the server from the data center.

2.7.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin removes the server from the data center.
2) The system admin marks the server as removed in the GME repository.

Alternative Flows

None.

Exceptional Flows

None

2.7.7 Success Criteria

The server is no longer present in the data center.

2.7.8 Post Conditions

- The server is no longer present in the data center.
- The server is marked as removed/destroyed in the GME repository.

2.8 Decommission Physical Compute Device (Server)

2.8.1 Summary

This use case captures the decommissioning phase of the life cycle of a physical server, including the unbinding or removal of any other grid components that are bound to it or associated with it.

2.8.2 Dependencies

- Use Case 2.5, Stop Physical Compute Device (Server)
- Use Case 2.6, Unconfigure Physical Compute Device (Server)
- Use Case 2.7, Destroy Physical Compute Device (Server)

2.8.3 Actor(s)/Stakeholder(s)

- System admin
- Hardware technician
- GME

2.8.4 Pre-Conditions

Nothing is assumed about the state of the server, other than that it has been provisioned in the past. At the point in time that the server decommissioning is requested, the server may be in use and may have services, operating systems, virtual machine monitors (hypervisors) and hardware partitions bound to it. It may also be a part of a pool of servers.

- The server is physically connected to the power fabric.
- The server is physically connected to the network fabric.
- The server is physically connected to the storage fabric.
- The server is physically connected to the management fabric.
- The server has been added to the GME configuration.
- The server is manageable by the GME.
- The server is powered up.

2.8.5 Initiation Of Use Case

Removal of the server is requested. This may be for any reason, ranging from the server having failed to its having become obsolete.

2.8.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin (GME) powers the server down – see section 2.5.
2) The system admin unconfigures the server – see section 2.6.
3) The system admin destroys/removes the server – see section 2.7.

Alternative Flows

Step 1: This step may be missed if the server has already been powered down.

Step 2: This step may be missed if the server has already been unconfigured.

Exceptional Flows

Each step is repeated until it is successful.

2.8.7 Success Criteria

The server is no longer present in the data center.

2.8.8 Post Conditions

1) The server is not physically connected to the power fabric.
2) The server is not physically connected to the network fabric.
3) The server is not physically connected to the storage fabric.
4) The server is not physically connected to the management fabric.
5) The server is no longer manageable by the GME.
6) The server is powered down.
7) The server is no longer present in the data center.
8) The server is marked as removed/destroyed in the GME repository.

3 Provisioning Firmware

3.1 Provision Firmware Media

This use case captures the actions associated with making firmware available to the GME for later installation. Examples of such firmware could include OPB or BIOS updates for servers, or hypervisor updates for virtual machine monitors, or updates for storage RAID controllers or for network routers. The use case covers the acquisition and copying of the software into the GME repository, including updating the GME as to how to install the firmware.

3.1.1 Dependencies

None

3.1.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- GME

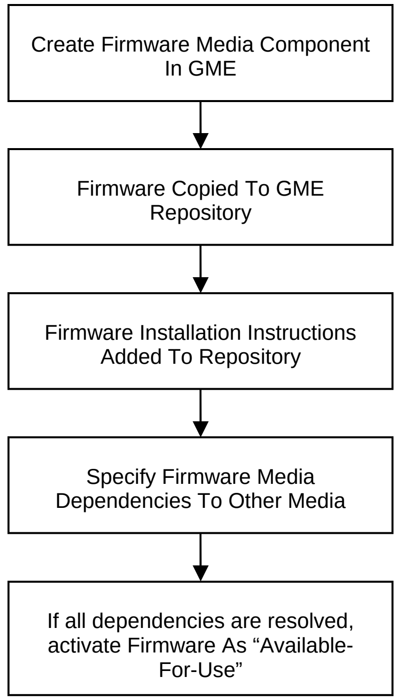
3.1.3 Pre-Conditions

- The GME Exists
- The GME has a repository to hold all firmware media that needs to eventually be provisioned on physical grid components.
- The GME has access to enough storage to store all firmware media.

3.1.4 Initiation Of Use Case

A new (or a upgrade version) of firmware media is available. This has to be provisioned on certain categories of grid components (e.g. BIOS on servers).

3.1.5 Flow Of Events (Basic, Alternative And Exceptional)



Basic Flow

1) Create/Discover firmware media
    a. Software admin creates new firmware media grid component in GME repository with appropriate name and version.
    b. Software admin places media (CD/DVD) in a drive that is accessible to the GME.
    c. Software admin (GME) copies firmware media into GME repository and associates it with the firmware media grid component.
    d. Software admin removes media from the drive.
2) Configure firmware media
    a. Software admin updates GME repository with information as to how to install the firmware, allowable targets, dependencies and so forth.
    b. Specify firmware media dependencies. Firmware media may have dependencies on other firmware versions. The GME database keeps track of the firmware media dependencies and all dependencies must be resolved before the firmware can be provisioned on a grid component.
3) Start firmware media
    a. If all firmware media dependencies have been resolved the software admin makes firmware available for use by marking it active in the GME repository.

Alternative Flows

Step 2: Replace with a copy from the Internet.

Exceptional Flows

If dependent firmware media is not available with GME then the firmware media cannot be set as available for use.

3.1.6 Success Criteria

The GME has successfully registered the firmware, uploaded the media into its repository and has accepted installation instructions and dependencies.

3.1.7 Post Conditions

Once successfully completed, the Firmware media is available in the GME repository for a Provision Firmware operation when needed.

3.2 Decommission Firmware Media

3.2.1 Summary

This use case captures the process associated with the decommissioning of firmware media. This equates to removing it from the GME repository. It is expected that the GME grid component associated with the media will remain to ensure appropriate auditing. The decommissioning may be broken down in to steps if appropriate. For example, one could “stop” it, thus making it unavailable for deployment, although the media will still exist (so to speak) in the GME repository.

3.2.2 Dependencies

None

3.2.3 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- GME

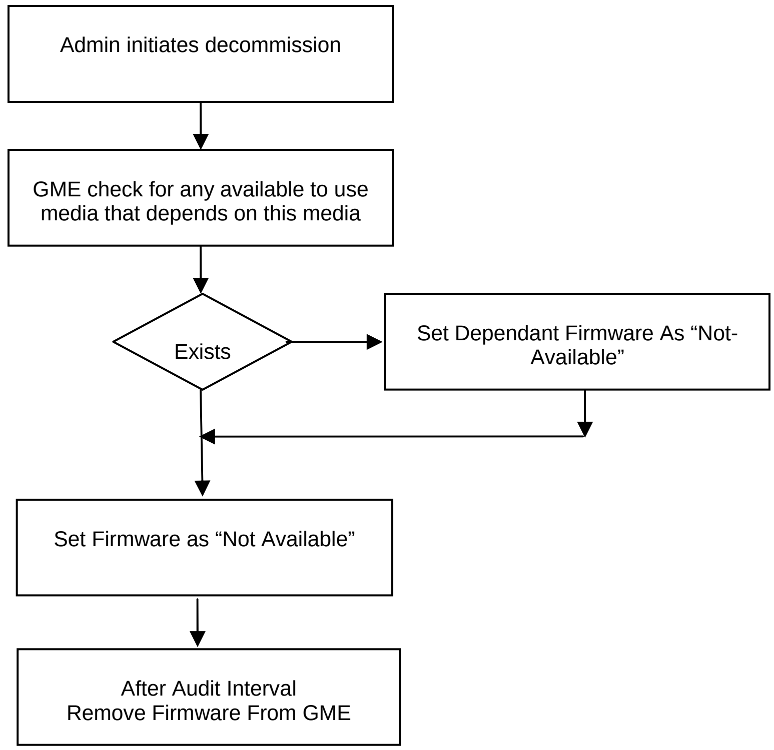
3.2.4 Pre-Conditions

- The GME Exists
- The GME has the firmware media provisioned on it.

3.2.5 Initiation Of Use Case

A firmware media that has been provisioned with the GME must be decommissioned. One of the administrators (Server, Network, or Storage Admin) initiates the decommissioning.

3.2.6 Flow Of Events (Basic, Alternative And Exceptional)



Basic Flow

1) Stop Firmware media
    a. Admin initiates the decommissioning, specifying the audit interval
2) Unconfigure Firmware media
    a. The GME checks if this firmware media is a dependency for any other “available-to-use” media
    b. If the dependency exists, the dependant firmware is set as not available
    c. The firmware media status is set to “Not-Available”.
    d. The decommission date and time is updated in the firmware media and dependent firmware for audit purposes.
3) Destroy/Remove Firmware media
    a. After the audit interval, the firmware media metadata and the media is removed from the GME database.

Alternative Flows

When other firmware media is dependant on the firmware media targeted for decommissioning, an alternative flow might be to check with the Admin before proceeding with the decommissioning. This avoids automatically decommissioning the dependant firmware if this was not intended or not desired. Thus at step 2 b) the admin would have the choice to set the dependant firmware as “Not-Available”, or to exit the decommissioning leaving both the targeted firmware and dependant firmware in their original (available) state.

Note there may be situations where firmware media is cross dependant (e.g. two firmware media modules are dependant on each other) and so neither can be decommissioned first and must both be decommissioned in the same operation.

3.2.7 Success Criteria

The firmware media status is set to “Not-Available”. After an audit interval has passed, the firmware media metadata and the media itself are removed from the GME database.

3.2.8 Post Conditions

Once successfully completed, the Firmware media is removed from the GME repository and is no longer available for a Provision Firmware operation.

3.3 Provision Firmware

3.3.1 Summary

This use case captures the process associated with the installation (provisioning) of firmware to a target grid component (the Device). Such a target could be a server (OBP, firmware or BIOS updates), a network Device (such as a router) or a storage Device (such as a RAID controller or SAN switch). No assumptions are made with regards to the prior provisioning of the firmware media, so that use case may need to be called.

3.3.2 Dependencies

- Use Case 3.1, Provision firmware media

3.3.3 Actor(s)/Stakeholder(s)

- System admin (Server Device)
- Network Admin (Network Device)
- Storage Admin (Storage Device)
- Security Admin
- GME

3.3.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The Device (Server, Network, Storage), is provisioned and running (Pre-OS state or OS state) and is manageable from the network. No service is bound to the device.
- It is assumed that the Device has a network manageable interface that allows full remote control of the Device that includes the necessary firmware provisioning mechanisms. Without such mechanisms a manual procedure must be used involving physical access to the Device by maintenance personnel
- If the Device has no firmware installed (or corrupt firmware) it may not be able to perform a Firmware Provisioning operation. If this is the case then the Device must be repaired or replaced before the Firmware Provisioning can be done.

3.3.5 Initiation of Use Case

Firmware Provisioning is initiated by a request from one of the administrators (Server, Network, or Storage Admin).

3.3.6 Flow Of Events (Basic, Alternative And Exceptional)

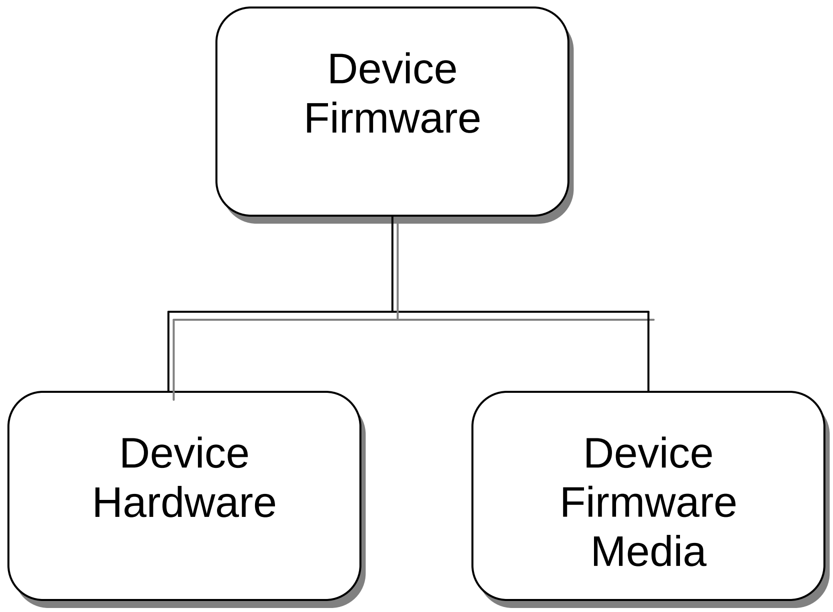


Figure 1 – Provision Firmware to Grid Device DAG

If the new Firmware Media is not already present in the GME repository then this step must be completed before attempting to Provision the firmware onto the Device.

If the Device is in service then it must be removed from service and placed in the Pre-OS state via the management interface.

Basic Flow

1) Establish connection to the Device with the necessary security privileges to perform Firmware Provisioning.
2) Save the current state of the Device in the GME repository so that the Device can be restored to the correct state after provisioning.
3) Verify compatibility of new Firmware media with current Device configuration.
4) Download new Firmware Media file to local temporary memory on the Device using the management interface and verify downloaded image is correct.
5) Backup previous copy of firmware, or verify that a valid master copy is available for recovery if needed.
6) Update the local Non-volatile firmware memory (e.g. Flash) with copy of new Firmware Media from local temporary memory.
7) Verify image correctly stored in local Non-volatile memory – must be placed into the appropriate location for proper execution on restart..
8) Restart Device using the new firmware.

9) Verify correct operation of Device (e.g. self-test)

10) Re-establish device/GME communication and report success of Firmware provisioning.

11) Update the GME repository to reflect the new Firmware is now provisioned.

12) Return the Device to the appropriate operational state it had before the provisioning operation

Alternative Flows

If the Device doesn't support the online management interface for firmware provisioning, then a manual procedure must be used. Instead of step 3) in the basic flow above, the Firmware media from the GME repository must be copied to the appropriate physical media and then loaded via the Device's media reader. When complete the status must be manually reported to the GME and the Device restarted.

In some Devices the Firmware may be located in a replaceable or socked chip or integrated circuit and may only be provisioned or re-provisioned by inserting the appropriate chip from the Device vendor, or replacing the current version of the chip with an updated one.

Exceptional Flows

If download fails or Non-volatile memory write fails, ensure previous or master copy of firmware is restored and used to restart the Device. Report the provisioning failure to the GME.

3.3.7 Success Criteria

The Device is successfully restarted and is now running the new Firmware Media as requested by the GME.

3.3.8 Post Conditions

The device is now updated with the new Firmware Media.

3.4 Decommission Firmware

3.4.1 Summary

This use case shows the process associated with the removal of firmware from the target grid component. In reality, such a process rarely exists, as firmware is usually overwritten rather than removed or erased and thus is really a special case of the Provision Firmware use case described previously with the content of the new Firmware to be Provisioned containing the default or minimal vendor firmware known in this case as the “base” Device Firmware.

[Once Firmware is actually removed or erased, the Device will usually be rendered inoperative and require hardware maintenance or a repair procedure to restore it to useful operation within the grid. Thus to make the Decommission Firmware a nondestructive step, the process must leave sufficient Firmware to restart and run the Provision Firmware operation.]

3.4.2 Dependencies

- Provision Firmware (section 3.3)

3.4.3 Actor(s)/Stakeholder(s)

- System admin (Server Device)
- Network Admin (Network Device)
- Storage Admin (Storage Device)
- Security Admin
- GME

3.4.4 Pre-Conditions

The preconditions are the same as the “Provision Firmware” case.

3.4.5 Initiation of Use Case

Firmware Decommissioning is initiated by a request from one of the administrators (Server, Network, or Storage Admins).

3.4.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Perform a “Provision Firmware” operation using the “base” Firmware for the Device.

3.4.7 Success Criteria

The Device is successfully restarted and is now running the “base” Firmware as requested by the GME.

3.4.8 Post Conditions

The device is now in is original state of running the Device “base” Firmware.

4 Provisioning Service Processors And System Controllers

4.1 Provision System Controller Media

This use case captures the actions associated with making system controller media available to the GME for later installation. The use case covers the acquisition and copying of the software into the GME repository, including updating the GME as to how to install the firmware.

4.1.1 Dependencies

None

4.1.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- GME

4.1.3 Pre-Conditions

- The GME exists
- The GME repository exists
- The GME has access to enough storage to store all system controller media

4.1.4 Initiation Of Use Case

A new (or an upgrade version) of system controller media is available.

4.1.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Software admin creates new system controller media grid component in GME repository with appropriate name and version.
2) Software admin places media (CD/DVD) in a drive that is accessible to the GME.
3) Software admin (GME) copies system controller media into GME repository and associates it with the system controller media grid component.

4) Software admin removes media from the drive.

5) Software admin updates GME repository with information as to how to install the system controller media, allowable targets, dependencies and so forth.

6) Specify system controller media dependencies. System controller media may have dependencies on other system controller versions. Before the system controller media can be provisioned on a grid component, the system controller media dependencies must be verified.

7) Software admin makes system controller media available for use.

Alternative Flows

Step 2: Replace with a copy from the Internet.

Exceptional Flows

Dependent system controller media not available with GME.

4.1.6 Success Criteria

The GME has successfully registered the system controller media, uploaded the media into its repository and has accepted installation instructions and dependencies.

4.1.7 Post Conditions

Once successfully completed, the system controller media is available in the GME repository for a Provision System Controller operation when needed.

4.2 Decommission System Controller Media

4.2.1 Summary

This use case captures the process associated with the decommissioning of system controller media. This equates to removing it from the GME repository. It is expected that the GME grid component associated with the media will remain to ensure appropriate auditing. The decommissioning may be broken down in to steps if appropriate. For example, one could “stop” it, thus making it unavailable for deployment, although the media will still exist (so to speak) in the GME repository.

4.2.2 Dependencies

This use case is an example of, and extends, the general use case for provisioning a grid component

4.2.3 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- GME

4.2.4 Pre-Conditions

- The GME Exists
- The GME has the system controller media provisioned on it.

4.2.5 Initiation Of Use Case

A system controller media that has been provisioned with one or more GMEs must be decommissioned. The administrator (one of the admin actors) initiates the decommissioning.

4.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Admin initiates the decommissioning, specifying the audit interval
2) The GME checks if this system controller media is a dependency for any other “available-to-use” media

3) If the dependency exists, the decommissioning action is stopped and the admin informed of the dependency error
4) If there are no dependencies, the system controller media status is set to “Not-Available”. The decommission date and time is updated for audit purposes.
5) After the audit interval, the system controller media metadata and the media is removed from the GME database.

Alternative Flows

Exceptional Flows

Dependent system controller media preventing decomission.

4.2.7 Success Criteria

The system controller media being decommissioned has no other system controller media dependent on it. The system controller media status is set to “Not-Available”. After an audit interval has passed, the system controller media metadata and the media itself is removed from the GME database.

4.2.8 Post Conditions

Once successfully completed, the system controller media is removed from the GME repository and is no longer available for a Provision System Controller media operation.

4.3 Provision System Controller

4.3.1 Summary

This use case describes the process associated with provisioning a system controller associated with a server. The system controller media is assumed to have already been provisioned.

4.3.2 Dependencies

Use case 4.1, Provision system controller media

4.3.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

4.3.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The system controller media exists in the GME.

4.3.5 Initiation Of Use Case

Provisioning of a systems controller is requested.

4.3.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME checks whether the system controller media is compatible with the system on which it is to be provisioned.
2) GME makes the node(s) unavailable during the provisioning process, by updating the information repository.

3) GME stops any running applications, and updates repository.

4) GME restarts the specified node(s) with the right privilege level, if needed.

5) GME downloads the system controller media to the specified node(s).

6) GME configures the system, and the installation scripts for correct installation.

7) GME starts the scripts for installing the software.

8) GME checks the log files for correct installation.

9) GME repeats steps 4) - 7) if the installation was not successful as specified by a policy engine.

10) GME updates its repository.

11) GME makes provisioned system controller available for use.

12) GME sends a notification to the subscribed users about the new system controller.

Alternative Flows

Provision with a copy from the Internet.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

4.3.7 Success Criteria

The new system controller software has been installed on the specified node(s) in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

4.3.8 Post Conditions

The new system controller software has been installed on the specified node(s) in the grid, and relevant info added into the repository of the GME.

4.4 Decommission System Controller

4.4.1 Summary

This use case describes the process associated with the decommissioning of a system controller associated with a server.

4.4.2 Dependencies

None

4.4.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- System admin
- Security Admin

4.4.4 GME Pre-Conditions

- The GME exists.
- The GME repository exists.
- The system controller exists on the node(s).

4.4.5 Initiation Of Use Case

Decommissioning of a system controller is requested.

4.4.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME checks whether the system controller is already installed on the system on which it is to be decommissioned.
2) GME updates the information repository and makes the node(s) to be decommissioned as unavailable during the decommissioning process.
3) GME stops any running applications.

4) GME restarts the specified node(s) with the right privilege level, if needed.
5) GME configures the system, and the installation scripts for correct decommissioning.
6) GME starts the scripts for decommissioning the system controller.
7) GME checks the log files for correct decommissioning.
8) GME repeats steps 4) - 7) if the decommissioning was not successful as specified by a policy engine.
9) GME updates its repository.
10) GME makes decommissioned node(s) available for use.
11) GME sends a notification to the subscribed users about the decommissioned node(s).

Alternative Flows

Try a manual decommission of the system controller.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

4.4.7 Success Criteria

The system controller has been decommissioned on the specified node(s) in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

4.4.8 Post Conditions

The system controller has been decommissioned on the specified node(s) in the grid, and relevant info added into the repository of the GME.

5 Provisioning Hardware Partitions

5.1 Provision Hardware Partition

5.1.1 Summary

This use case describes the process by which a hardware partition is configured on a server. Hardware partitions are typically created by configuring the memory/IO switch which a set of boards (each typically including processors, memory and some IO) share, such that only a subset of components can share memory/IO. This subset then behaves as a single physical server. In this way, a rack of components that share a memory/IO switch may behave as a single physical compute device (server) or many physical compute devices (servers). This provisioning is usually undertaken through interaction with the service processor. Nothing is assumed about the state of the components that will comprise the hardware partition, such as whether they are part of another partition, or if an OS is already booted on them.

This use case assumes that the software required to manage hardware partitions and the service processor or controller on which it is installed has already been provisioned. This is viewed as a part of the GME.

This use case assumes that the partitionable resource set has a fixed and pre-assigned set of potential partitions, thus a new partition, in the logical sense, does not need to be created. Rather, a set of physical resources has to be assigned to it to make it usable.

5.1.2 Dependencies

This use case depends upon (calls) the following use cases:

- Use case 2.2, Configure Physical Compute Device (Server)
- Use case 2.3, Start (Power-On) Physical Compute Device (Server)

5.1.3 Actor(s)/Stakeholder(s)

- System Admin – the administrator responsible for man aging the rack(s) of partitionable resources.
- GME

5.1.4 Pre-Conditions

- The rack(s) containing the partitionable resources has been provisioned, see section 2.4, Provision Physical Compute Device (Server).
- The service processor or system controller has already been provisioned and is accessible via the management network.

5.1.5 Initiation Of Use Case

A new hardware partition is required to host a new operating system instance and application component.

5.1.6 Flow Of Events (Basic, Alternative And Exceptional)

This flow is based on the process used to configure domains (a.k.a. hardware partitions) on Sun's Sun Fire™ 15K server (see ref [2]).

Basic Flow

1) The platform admin (GME) connects to the system controller.
2) The platform admin (GME) logs into the platform management shell, rather than into any existing hardware partition.
3) The platform admin (GME) creates a new hardware partition.
4) The platform admin adds a new instance of the hardware partition to be managed to the GME repository. This act essentially adds the hardware partition to the asset register. This may include adding information such as –
    a. Physical location of the hardware partition, i.e. the set of resources within which the partition exists.
    b. Serial number of hardware partition and any appropriate sub-components of the partition.
    c. Host ID of hardware partition (if this exists).
5) The system admin sets the hardware partition state to unconfigured in the GME repository.
6) The platform admin configures the hardware partition - see section 2.2, Configure Physical Compute Device (Server). Configuration in this case specifically includes the following additional steps
    a. The platform admin (GME) adds each desired processor/memory board and i/o board to the selected hardware partition. This equates to steps 1 and 2 in the original Configure Physical Compute Device (Server) use case.
7) The platform admin (GME) logs into the hardware partition management shell on the system controller.
8) The platform admin (GME) sets the hardware partition date and time.
9) The platform admin (GME) sets the password for accessing the hardware partition via the system controller.
10) The platform admin (GME) sets up other partition specific parameters.
11) The platform admin (GME) saves the hardware partition configuration.
12) The platform admin (GME) starts any components that are physically connected to the hardware partition and which it depends upon, for example externally attached devices, such as non-network or SAN attached disks or arrays.
13) The platform admin (GME) logically powers up the hardware partition.

14) The platform admin (GME) updates the state of the power supply to the hardware partition in the GME repository.

15) The platform admin (GME) updates the hardware partition state in the GME repository to started.

Alternative Flows

Step 3: This step may be missed if the partitionable resources already have a fixed set of hardware partitions that exist logically even though physical resources may not have been assigned to them.

Exceptional Flows

In the event of failure, the steps are rewound to the last appropriate state, i.e. either discovered, configured or destroyed. The hardware partition cannot revert to the undiscovered state as the GME must retain a log of the failed attempt to provision the hardware partition.

5.1.7 Success Criteria

The hardware partition is powered up and manageable via the management network/system controller by the GME.

5.1.8 Post Conditions

- The hardware partition is physically connected to the network fabric.
- The hardware partition is physically connected to the storage fabric.
- The hardware partition is physically connected to the management fabric.
- The hardware partition is powered up.
- The hardware partition has been added to the GME repository.
- The hardware partition is manageable by the GME

5.2 Decommission Hardware Partition

5.2.1 Summary

This use case describes the decommissioning of a hardware partition. No assumptions are made as to the state of the partition, i.e. whether it is running an OS instance.

5.2.2 Dependencies

None

5.2.3 Actor(s)/Stakeholder(s)

- System Admin
- GME

5.2.4 Pre-Conditions

- The hardware partition is physically connected to the network fabric.
- The hardware partition is physically connected to the storage fabric.
- The hardware partition is physically connected to the management fabric.
- The hardware partition is powered up.
- The hardware partition has been added to the GME repository.
- The hardware partition is manageable by the GME.

5.2.5 Initiation Of Use Case

The hardware partition is to be removed as the resources it uses are to be reassigned.

5.2.6 Flow Of Events (Basic, Alternative And Exceptional)

This flow is based on the process used to configure domains (a.k.a. hardware partitions) on Sun's Sun Fire™ 15K server (see ref [2]).

Basic Flow

1) The platform admin (GME) requests that each of the grid components that are bound to the hardware partition, if any, is stopped and potentially unconfigured. This may include, but is not necessarily limited to virtual machine monitors and operating systems.
2) The platform admin (GME) connects to the system controller.

3) The platform admin (GME) logs into the hardware partition management shell on the system controller.

4) The platform admin (GME) logically powers down the hardware partition.

5) The platform admin (GME) updates the state of the power supply to the hardware partition in the GME repository.

6) The platform admin (GME) updates the hardware partition state in the GME repository to configured.

7) The platform admin (GME) stops any components that are physically connected to the hardware partition and which it depends upon, for example externally attached devices, such as non-network or SAN attached disks or arrays. Alternatively the platform admin (GME) requests that they be stopped.

8) The platform admin (GME) logs into the platform management shell, rather than into any existing hardware partition.

9) The platform admin unconfigures the hardware partition - see section 2.6, Unconfigure Physical Compute Device (Server). Unconfiguration in this case specifically includes the following additional steps

    a. The platform admin (GME) removes each processor/memory board and i/o board from the selected hardware partition.

10) The platform admin (GME) resets the password for accessing the hardware partition via the system controller.

11) The platform admin (GME) resets up other partition specific parameters.

12) The platform admin (GME) saves the hardware partition configuration.

13) The platform admin (GME) removes/deletes the hardware partition.

14) The system admin sets the hardware partition state to destroyed/removed in the GME repository.

Alternative Flows

Step 1: Step 1 may be missed if all of the grid components that depend upon it have already been stopped and/or unconfigured.

Step 7: Step 7 may be missed if the hardware partition has no external dependencies.

Step 13: Step 13 may be missed if it is not possible to delete a partition. In this case decommissioning merely results in a change of state to destroyed, but it can always be reprovisioned.

Exceptional Flows

The steps are repeated until successful or the flow is rewound to the nearest state – started, configured or discovered.

5.2.7 Success Criteria

The hardware partition is marked as destroyed/removed in the GME repository.

5.2.8 Post Conditions

- The hardware partition is powered down.
- The hardware partition is no longer physically connected to the network fabric.
- The hardware partition is no longer physically connected to the storage fabric.
- The hardware partition is no longer physically connected to the management fabric.
- The hardware partition state is set to removed/destroyed in the GME repository.

6 Provisioning Bootable Storage

6.1 Provision Boot Disk/LUN - Local Disk

6.1.1 Summary

This use case captures the provisioning of a boot LUN for use by a compute device for storing a bootable OS image. This use case captures the case where such a LUN is on a local, that is non-shared, disk. The disk may be connected to the server via IDE, EIDE, SCSI, FC-AL and so forth. This use case may include the physical connection of the disk, its configuration, for example through dip switches, and configuration of the server OBP or BIOS to point to it as the boot disk. It is not assumed that the disk already exists with the GME as a managed grid component.

6.1.2 Dependencies

None

6.1.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- GME

6.1.4 Pre-Conditions

- The Server is provisioned, and contains the required I/O devices to connect to the boot device.

6.1.5 Initiation Of Use Case

The Hardware Technician provisions a server. That server will require a boot disk.

6.1.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1. Perform the physical connection of the disk by preparing the system for installation and familiarizing yourself with the safety requirements. Then install the hardware controller and/or the new disk. Be sure to properly configure the disk setting the appropriate hardware switches, if available.

2. Configure the server OBP or BIOS to point to the boot disk. All BIOS's contain a setting that controls the search sequence for a boot drive. Once the BIOS identifies the target boot drive, the it looks for boot information to start the operating system boot process, searching the disk for a Master Boot Record (MBR) or boot sector.

   In the PC architecture the Master Boot Record (MBR), or partition sector, is the 512-byte boot sector, i.e. the sector on the logical beginning of a hard disk that contains the sequence of commands necessary for booting the operating system(s). The bootstrapping firmware contained within the ROM BIOS loads and executes the master boot record. The MBR of a drive usually includes the drive's partition table, which the PC uses to load and run the boot record of the partition that is marked with the active flag.
3. Create a Master Boot Record or appropriate boot sector and allocate or divide the disk into partitions for the operating system's use utilizing a partition table manipulator, typically known as the fdisk utility.
4. Format the partition(s), installing the appropriate file system on the media. Make a file system on the disk partition/partitions using the appropriate OS based tools.
5. Attach the partitions to the file system. The mechanism for performing this action is OS specific. An operation of connecting partitions (to be more exact - connecting file systems) in Linux is called mounting and, as such, uses the command 'mount' to handle this. When mounting a partition in Linux, it needs to be associated with a directory somewhere in the file system. This is called creating a mount point. System configuration data may also need to be updated to automate the mounting of the drive on system startup.

Alternative Flows

Exceptional Flows

6.1.7 Success Criteria

The Disk is mountable and ready to store a bootable OS image

6.1.8 Post Conditions

The Disk is mountable and ready to store a bootable OS image.

6.2 Decommission Boot Disc/LUN – Local Disc

6.2.1 Summary

This use case captures the decommissioning of a boot LUN for use by a compute device for storing a bootable OS image. This frees the LUN or disk for use by other grid components. Releasing the disk may require it to be scrubbed and for the GME repository to be updated.

6.2.2 Dependencies

None

6.2.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- GME

6.2.4 Pre-Conditions

- A local non-shared disk has been provisioned and is available for use by a compute device

6.2.5 Initiation Of Use Case

A local non-shared disk which is available for use by a compute device is available to be decommissioned.

6.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Inform the GME that this resource is unavailable.
2) Disconnect the partition. The mechanism for performing this action is OS specific. An operation of disconnecting partitions (to be more exact - disconnecting file systems) in Linux is called unmounting and, as such, uses the command 'umount' to handle this. When unmounting a partition in Linux, the association with the file system is removed at the mount point. System configuration data may also need to be updated to remove the auto-mounting of the drive on system startup.

3) Scrub the partition(s), using the appropriate OS based tools if available. If not specific tools are available, utilize a partition table manipulator (fdisk) and the partition formating tools to scrub the disk.

Alternative Flows

Exceptional Flows

6.2.7 Success Criteria

The Disk has been scrubbed.

6.2.8 Post Conditions

The Disk has been scrubbed and available to be provisioned.

7 Reserved for future

8 Reserved for future

9 Provisioning Operating Systems

9.1 Provision OS Media

9.1.1 Summary

This use case describes the process associated with making OS media available to the GME prior to the creation of an OS instance and binding it to another grid component. The example below is based on the creation of install images suitable for the installation of operating systems to servers across a local area network.

9.1.2 Dependencies

The installation server, a part of the GME and/or GME repository, must be configured to use a naming service, such as NIS, NIS+, DNS or LDAP, before starting the primary flow.

9.1.3 Actor(s)/Stakeholder(s)

- System Admin
- GME

9.1.4 Pre-Conditions

None

9.1.5 Initiation Of Use Case

A new version of an operating system is needed within the data center.

9.1.6 Flow Of Events (Basic, Alternative And Exceptional)

This flow is based on the use of the Sun Jumpstart mechanism for Solaris, see ref [4]. Similar mechanisms exist for the network based provisioning of Linux, HP/UX, etc.

Basic Flow

1) The system admin adds a class of operating system to be managed in the GME repository. This may include adding information such as
   - Manufacturer
   - Part number

- Capabilities or attributes of the media.
- Other operating system media specific information necessary [for the GME] to manage the type of operating system media.

2) The systems admin (GME) adds a new instance of the operating system media to be managed to the GME repository. This act essentially adds the operating system media to the asset register. This may include information such as
   - Physical location of media
3) The systems admin sets the operating system media instance state to unconfigured in the GME repository.
4) The systems admin (GME) logs in as super user onto the server (GME repository) that will provide network installation services.
5) The systems admin inserts the operating system distribution media into a CD or DVD drive that is accessible by the installation server.
6) The systems admin (GME) creates a directory in which to store the operating system installation image.
7) The systems admin (GME) copies the operating system distribution media image into the installation images directory.
8) The systems admin (GME) sets the operating system media instance state to configured in the GME repository.
9) The systems admin removes the operating system media from the CD/DVD drive and replaces it in its usual storage location.
10) The systems admin patches the install image as necessary.
11) The systems admin (GME) makes the installation images directory accessible via the network, for example through the use of NFS.
12) The systems admin (GME) sets the operating system media instance state to active.
13) The systems admin (GME) logs of the installation server.

Alternative Flows

Step 1: Step 1 may be missed if an operating system media type for this instance already exists.

Exceptional Flows

Steps 4 to 8: If any of these steps fail then the steps are backed out back to step 4. Then step 9 is executed.

Steps 10 to 11: If any of these steps fail then the steps are backed out to step 10.

9.1.7 Success Criteria

The operating system media instance state is set to active in the GME repository.

9.1.8 Post Conditions

- The operating system media image is available from the install server via NFS.

9.2 Decommission OS Media

9.2.1 Summary

This use case captures the process associated with the decommissioning of OS media. Whilst the copy of the software is removed, the GME grid component associated with it, including all of the records of all actions associated with it should remain.

9.2.2 Dependencies

None

9.2.3 Actor(s)/Stakeholder(s)

- GME

9.2.4 Pre-Conditions

- The operating system media instance state is set to active.

9.2.5 Initiation Of Use Case

The operating system instance and/or type is deprecated from the enterprise grid and is no longer depended upon by any other grid components.

9.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The systems admin (GME) logs onto the installation server (GME repository) as super user.
2) The systems admin (GME) configures the installation server to ensure that the specified installation image is no longer associated with any potential clients/targets.
3) The systems admin (GME) makes the installation images directory inaccessible via the network.
4) The systems admin (GME) sets the operating system media instance state to configured.
5) The systems admin (GME) removes the operating system distribution media image from the installation images directory.
6) The systems admin (GME) removes the directory in which the operating system installation image was stored.

7) The systems admin (GME) sets the operating system media instance state to unconfigured in the GME repository.

8) The systems admin removes the operating instance media from its storage location and destroys/removes it.

9) The systems admin sets the operating system media instance state to destroyed/removed in the GME repository.

10) The systems admin (GME) logs off from the installation server (GME repository).

Alternative Flows

Step 3: Step 3 is optional. It may be missed out of the shared directory tree contains other directories which need to remain remotely accessible.

Exceptional Flows

Steps 1 to 3: If any of these steps fail then the state of the operating system media instance in the GME remains active and the flow ends.

Steps 4, 7 and 9: If any of these steps fail then the flow terminates immediately with and error being logged in the GME repository event log.

9.2.7 Success Criteria

The operating systems media instance state is set to destroyed/removed in the GME repository.

9.2.8 Post Conditions

The operating systems media instance state is set to destroyed/removed in the GME repository.

9.3 Provision An OS Instance To A Server

9.3.1 Summary

To realize the benefits of grid computing, it is important to be able to provision and de-provision services, as your business needs change. This use case deals with provisioning a server (including an operating system) in an enterprise grid in order to meet the SLOs and/or constraints of a service or business application.

9.3.2 Dependencies

This use case is dependent on, i.e. it calls the following use cases:

- Provision OS Media: Use case 9.1
- Provision Physical Compute Device (Server): Use case 2.4
- Provision Boot Disk/LUN – Local Disk: Use case 6.1

9.3.3 Actor(s)/Stakeholder(s)

- Primary actor is GME
- Secondary actors which may act on behalf of the GME include:
    - Data Center System Admin
    - Data Center Admin
    - Data Center Manager
    - Data Center DBA
    - Data Center Network Admin
    - Data Center Storage admin
    - Data Center System admin
    - Data Center Security Admin

9.3.4 Pre-Conditions

- The various grid components, as enumerated in the DAG are manageable by the GME. For a physical component (e.g. server, drive array, network switch/router, software distribution media), this means that it must be at the enterprise grid's physical location. For all other grid components, it means that they must be logically manageable by the GME.
- All SLOs and constraints for the grid components have been defined. Depending on the particular type of grid component this may range from simple parameters such as the number of processors to be used by an application, to the high level business SLOs (e.g. average

transaction response time) which map directly to the SLA for the grid component which is a complete business application. It is the role of the GME to derive component goals from those of aggregated sets of components or more abstract components such as business applications or services.

- Use of the grid component and its intended deployment has been appropriately licensed or will be appropriately licensed in parallel to the provisioning process.
  Note – it may be appropriate for future iterations of this document to more clearly capture the licensing process as this is a key, long-term enabler for utility computing environments and may be closely bound to the utility accounting process.

9.3.5 Initiation Of Use Case

The GME determines an additional server, and associated operating system, is required to host a service in order to meet SLOs.

9.3.6 Flow Of Events (Basic, Alternative And Exceptional)

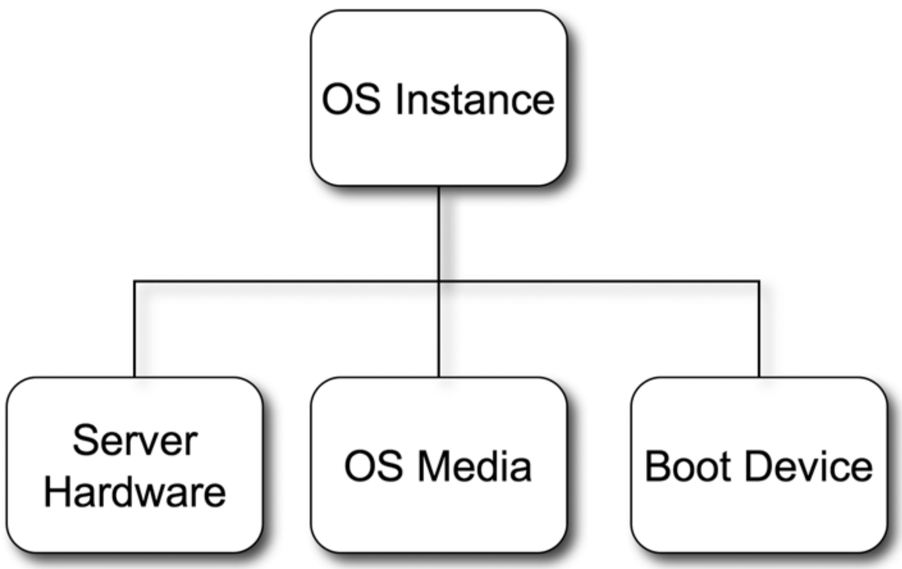


Figure 2 – Provision OS Instance to Server DAG

Basic Flow

Referring to Figure 1, Provisioning OS Instance to Server DAG, above. In order to provision an active OS instance, the DAG must be traversed as described in the general grid component provisioning use case above.

1) ***Provision Physical Compute Device (Server)***

Follow procedures in use case 2.4

2) ***Provision OS media grid component***

Follow the steps in use case 9.1, Provision OS Media.

3) ***Provision Boot Disk/LUN***

- If boot disk/LUN is local, follow steps in use case 6.1
- If boot disk/LUN is on a SAN, follow steps in a future use case
- If boot disk/LUN is NAS, follow steps in a future use case

4) ***Create/Discover OS instance Grid Component***

   Create new OS instance grid component in GME—status is unconfigured.

5) ***Configure OS Instance Grid Component***

   a) Create bootable image on boot device for server component.
   b) Configure OS management network, not necessarily same as console port and network.
   c) Configure OS instance with appropriate policies etc.
   d) Update GME to reflect OS Instance grid component is inactive.

6) ***Start OS Instance Grid Component***

   a) Activate OS instance.
   b) Update GME to reflect OS Instance Grid Component is active.

Alternative Flows

Many other flows are possible depending on the starting states of the various grid components. For example the boot device could be a local storage device within a server and thus may need different configuration steps/actions.

Also some of the state changes for individual grid components may be interleaved as their creation, configuration and activation may be carried out in parallel.

Exceptional Flows

Exceptions would be generated if the appropriate grid components are not available, or any of the steps generate errors, or the performance metric objectives are not met.

9.3.7 Success Criteria

The specified grid component is activated.

9.3.8 Post Conditions

The grid component is active and available for the GME to deploy other grid components or services on to.

9.4 Reboot OS

9.4.1 Summary

This use case will reboot the operating system from a server.

9.4.2 Dependencies

This use case is dependent on the following use cases (for exception handling only):

- Decommission OS Instance from Server: See section 9.5
- Provision OS Instance from Sever: See section 9.3

9.4.3 Actor(s)/Stakeholder(s)

- Primary actor is GME
- Secondary actors which may act on behalf of the GME include:
    - Data Center System Admin
    - Data Center Admin
    - Data Center Manager
    - Data Center DBA
    - Data Center Network Admin
    - Data Center Storage admin
    - Data Center System admin
    - Data Center Security Admin

9.4.4 Pre-Conditions

The workload or service hosted by the server has terminated

9.4.5 Initiation Of Use Case

System Admin or GME must determines system must be rebooted.

9.4.6 Flow Of Events (Basic, Alternative And Exceptional)

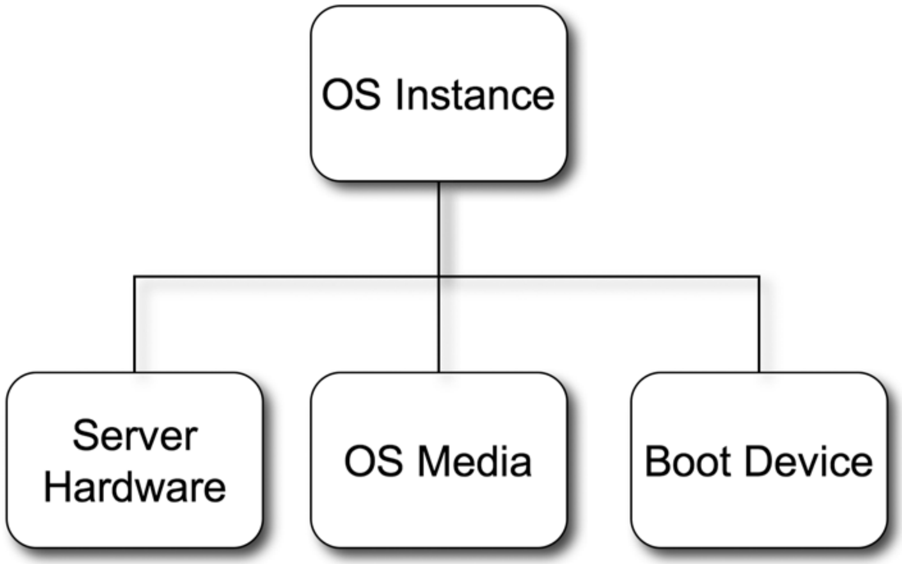


Figure 3 – Reboot Server DAG

Basic Flow

1. Update GME to reflect OS instance is inactive
2. Run reboot script for OS instance
3. Update GME to reflect OS Instance is Active

Alternative Flows

Many other flows are possible depending on the starting states of the various grid components. For example the boot device could be a local storage device within a server and thus may need different configuration steps/actions.

Also some of the state changes for individual grid components may be interleaved as their creation, configuration and activation may be carried out in parallel.

If no reboot script exists, then run the shutdown script and restart the system.

If the OS Instance is stopped, start the system.

Exceptional Flows

Exceptions would be generated if the appropriate grid components are not available, or any of the steps generate errors, or the performance metric objectives are not met.

If either step 1 or step 2 fail, then follow this exception flow.

1) Decommission the OS Instance as described in use case 9.5
2) Commission a new OS Instance as described in use case 9.3.

9.4.7 Success Criteria

The specified grid component is activated.

9.4.8 Post Conditions

The grid component is active and available for the GME to deploy other grid components or services onto.

9.5 Decommission OS Instance From Server

9.5.1 Summary

This use case will decommission the operating system from a server that is no longer needed to host a service. This will prepare the grid components for reuse by another service.

9.5.2 Dependencies

None

9.5.3 Actor(s)/Stakeholder(s)

- Primary actor is GME
- Secondary actors which may act on behalf of the GME include:
  - Data Center System Admin
  - Data Center Admin
  - Data Center Manager
  - Data Center DBA
  - Data Center Network Admin
  - Data Center Storage admin
  - Data Center System admin
  - Data Center Security Admin

9.5.4 Pre-Conditions

- The workload or service hosted by the server has terminated
- There is no longer a reservation for this server in the GME repository

9.5.5 Initiation Of Use Case

The GME detects that the reservation for this server that assigned it to a service has been cancelled.

9.5.6 Flow Of Events (Basic, Alternative And Exceptional)

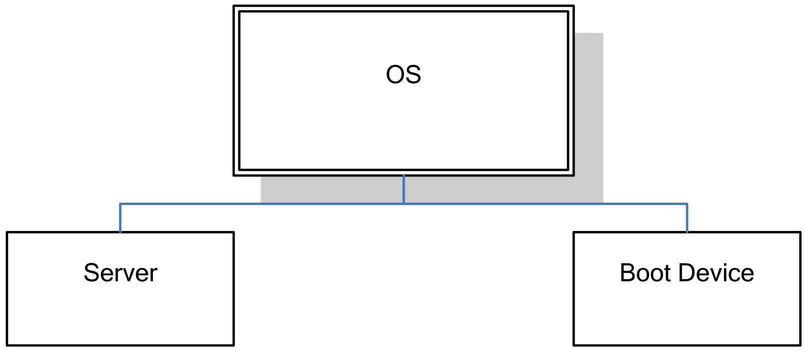


Figure 4 - Decommission OS Instance from Server DAG

Basic Flow

1. **Stop OS Instance Grid Component**
   a. Run shutdown script for server
   b. Update GME to reflect OS Instance inactive
2. **Unconfigure OS Instance Grid Component**
   c. Remove the bootable image from the boot device
   d. Update GME to reflect OS Instance unconfigured
3. **Destroy OS Instance Grid Component**

   Update the server configuration in the GME repository to reflect the OS instance grid component is destroyed.

Alternative Flows

- If a server is already shut down, then skip step 2.

Exceptional Flows

- GME will verify no services are active on any server. If a server is hosting an active service, the operation on that server will fail.
- If the OS fails to shutdown, then power cycle server and continue with step 3.
- If any steps fail, flag the server for diagnostic tests.

9.5.7 Success Criteria

The server has been shut down, and is not associated with an operating system

9.5.8 Post Conditions

The server is ready to be provisioned to another operating system and service, or has been flagged disabled due to hardware fault.

10 Provisioning Patches

10.1 Provision Patch Media

10.1.1 Summary

This use case describes the process associated with provisioning of patch media for an operating system, application or other grid component to the GME repository prior to application to a target grid component. The media may be on CD, DVD or a download from the Internet.

Dependencies

None

10.1.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

10.1.3 Pre-Conditions

- The GME exists.
- The GME repository exists.

10.1.4 Initiation Of Use Case

Provisioning of patch media is requested. This may be for any reason, ranging from a notification from a software vendor to an in-house patch for some known problems.

10.1.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Software admin creates new patch media grid component in GME repository with appropriate name and version.
2) Software admin places media (CD/DVD) in a drive that is accessible to the GME.

3) Software admin copies patch media into GME repository and associates it with the patch media grid component.
4) Software admin removes patch media from the drive.
5) Software admin updates GME repository with information as to how to install the patch, allowable targets, dependencies and so forth.
6) Software admin makes patch available for use.
7) Software admin sends a notification to the subscribed users about the new patch.

Alternative Flows

Step 2: Replace with a copy from the Internet.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

10.1.6 Success Criteria

The specified patch media has been added to the GME repository, GME records have been updated, and a notification sent to the system admin.

10.1.7 Post Conditions

The new patch media has been added to the repository of the GME, and relevant info added into the repository of the GME.

10.2 Decommission Patch Media

10.2.1 Summary

This use case captures the process associated with the decommissioning of patch media. Whilst the copy of the software is removed, the GME grid component associated with it, including all of the records of all actions associated with it should remain.

Dependencies

None

10.2.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

10.2.3 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The patch media exists in the GME.

10.2.4 Initiation Of Use Case

Decommissioning of patch media is requested. This may be for any reason, eg, a notification from a software vendor about an updated patch.

10.2.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME marks the patch media grid component in GME repository as decommissioned for future use, and the relevant reasons.
2) GME updates GME repository with information as to the affected targets, dependencies and so forth.

3) GME makes patch media unavailable for use.
4) GME sends a notification to the subscribed users about the decommissioning of this patch media.

Alternative Flows

Step 2: Replace with a copy from the Internet.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

10.2.6 Success Criteria

The specified patch media has been decommissioned from the GME repository, GME records have been updated, and a notification sent to the system admin.

10.2.7 Post Conditions

The new patch media has been decommissioned, and relevant info added into the repository of the GME.

10.3 Provision Patch

10.3.1 Summary

This use case describes the process associated with the application of a patch to an operating environment, application, or other grid component. Nothing is assumed about the starting state of the component or about the grid components bound to the OS or which depend upon it. The patch media is assumed to have already been provisioned.

Dependencies

- Provision patch media – see use case 10.1.

10.3.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

10.3.3 Pre-Conditions

- The GME exists.
- The GME repository exists.

10.3.4 Initiation Of Use Case

Provisioning of a patch is requested. This may be for any reason, ranging from a notification from a software vendor to an in-house patch for some known problems.

10.3.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

There are two basic types of patches. Online patches can be discovered and configured while the grid component is active, and the component only needs to be restarted (stopped and started) in order to active the patch. Offline patches cannot be discovered and configured while the target grid component is active. It must be stopped prior to starting discovery, and started upon completion of patch configuration. For this use case, offline patches will form the basic flow, and online patches will be treated as an alternative flow.

1) Create/Discover the patch
    a. GME checks whether the patch is compatible with the component on which it is to be provisioned.
    b. GME checks for dependencies—for example, other grid components that must also be patched. An example would be other components that make up a clustered service. All instances of the service may need to be patched at the same time.
    c. GME downloads the patch to the specified node(s).
    d. GME stops any running applications, and updates repository. (This may be another use case.)
    e. GME makes the node unavailable during the provisioning process, by updating the information repository.
    f. GME starts the specified node(s) with the right privilege level, if needed.
    g. Patch is now unconfigured
2) Configure the patch
    a. GME configures the system, and the installation scripts for correct installation.
    b. GME starts the scripts for installing the patch.
    c. GME checks the log files for correct installation.
    d. GME repeats steps 4) - 7) if the installation was not successful as specified by a policy engine.
    e. GME updates its repository.
    f. Patch is now inactive
3) Start the patch
    a. GME makes patched component available for use. This usually requires restarting the component, but not always
    b. GME sends a notification to the subscribed users about the new patch.
    c. Patch is now active
4) GME restarts dependent applications on the patched node. (This may be another use case.)

Alternative Flows

If the patch media does not exist, provision the patch media as in use case 10.1.

If patch is online, then steps 1e and 1f are moved to before step 3a.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

10.3.6 Success Criteria

The new patch has been installed on the specified nodes in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

10.3.7 Post Conditions

The new patch has been installed on the specified nodes in the grid, and relevant info added into the repository of the GME.

10.4 Decommission (Roll back) Patch

10.4.1 Summary

This use case describes the process by which a patch is rolled back. This is usually done through rolling back to a backed up version of the OS. Nothing is assumed about other grid components that are bound to the OS instance or which depend upon it. Nothing is assumed about the state of the OS instance (bound, not bound, running, configured etc.).

Dependencies

None

10.4.2 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

10.4.3 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The patch exists on the node.

10.4.4 Initiation Of Use Case

Decommissioning of a patch is requested. This may be for any reason, e.g., software compatibility of an existing application.

10.4.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME checks whether the patch is already installed on the system on which it is to be decommissioned.
2) GME updates the information repository and makes the node to be decommissioned as unavailable during the decommissioning process.

3) GME updates information repository about dependencies, relevant reasons etc.
4) GME stops any running applications, and updates repository. (This may be another use case.)
5) GME starts the specified node(s) with the right privilege level, if needed.
6) GME configures the system, and the installation scripts for correct decommissioning.
7) GME starts the scripts for decommssioning the patch.
8) GME checks the log files for correct decommissioning.
9) GME repeats steps 6) - 8) if the decommissioning was not successful as specified by a policy engine.
10) GME updates its repository.
11) GME makes decommissioned node available for use.
12) GME sends a notification to the subscribed users about the decommissioned patch.
13) GME restarts applications on the patched node. (This may be another use case.)

Alternative Flows

Try a manual decommission of the patch.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

10.4.6 Success Criteria

The patch has been decommissioned on the specified node in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

10.4.7 Post Conditions

The patch has been decommissioned on the specified node in the grid, and relevant info added into the repository of the GME.

11 Provisioning Pools Of Servers (Physical & Logical)

11.1 Create Pool Of Compute Devices (Servers) Object

11.1.1 Summary

This use case describes the process of creating a pool of servers. The pool is explicitly created to enable group functions to be undertaken on the members. The pool may be short or long lived depending on the function or set of functions to be applied to its members. Membership of the pool may be completely open and based solely on choices made by the system admin, or qualification through some set of shared properties or shared state may be stipulated.

The create pools use case does not result in any components or servers being members of the pool. It merely results in the pool existing as an abstract entity within the GME. The servers may be physical or logical, but all must be essentially identical in terms of behaving in the same way when a specific verb or action (the reason for aggregation) is applied to them, for example provision OS, or start OS or whatever.

11.1.2 Dependencies

None

11.1.3 Actor(s)/Stakeholder(s)

- System admin
- GME

11.1.4 Pre-Conditions

- The GME exists.

11.1.5 Initiation Of Use Case

A system admin needs to create a pool of components, in this case a pool of servers (physical or logical compute devices, i.e. virtual machines hosted by a VMM), in order to manage them collectively, thus realizing management efficiency gains.

11.1.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin creates the pool object within the GME repository.

2) The system admin defines the initial properties of the pool. These could include minimum and maximum number of pool members for example, a user friendly name and so forth.

3) The system admin saves the set of initial pool attributes and properties to the GME repository.

4) The system admin defines a set of membership criteria for the pool. These could range from none, so that an appropriately authorized admin may add any server they like to the pool, to restrictions on specific server properties, such as processor architecture, memory capacity, storage connectivity and so forth. These criteria could be used to filter servers and create a candidate set from which an admin could choose, or they could be used as a cross check once initial selections have been made.

5) The system admin saves the set of membership criteria to the GME repository.

6) The system admin defines the set of verbs that may be applied to the members of the pool, for example provision firmware or provision operating system etc. The choice of verbs may be tightly coupled with the membership criteria. For example, if firmware media provisioning were to be enabled, then all of the members of the pool would need to support firmware upgrades.

7) The system admin saves the set of verbs to the GME repository.

8) The system admin defines the set of roles associated with managing the pool. Such roles could include roles for adding and removing members of the pool, and roles associated with applying specific verbs to the pool.

9) The system admin saves the set of roles to the GME repository.

10) The system admin defines the set of policies associated with managing the pool. Such policies could include, for example, what actions to take in the event of exceptions during aggregate operations in the pool membership.

11) The system admin saves the set of policies to the GME repository.

12) The system admin updates the GME repository to indicate that the pool is now available for use.

Alternative Flows

Steps 2 and 3 – These may be ignored and default values assumed.

Steps 4 and 5 – These may be ignored and default values assumed.

Steps 6 and 7 – These may be ignored and default values assumed.

Steps 8 and 9 – These may be ignored and default values assumed.

Steps 10 and 11 – These may be ignored and default values assumed.

Exceptional Flows

If any of the steps fail then the pool object is not available for use. Steps may be repeated until the system admin decides to back out, removing the pool object from the GME repository.

11.1.7 Success Criteria

The GME repository indicates that the pool is available for use.

11.1.8 Post Conditions

- The server pool object exists in the GME repository.
- The properties and constraints of the pool have been set in the GME repository.
- Any membership criteria, beyond simple admin assignment, have been set in the GME repository.
- The set of verbs that may be applied to the pool has been defined in the GME repository.
- The roles associated with management of the pool have been defined in the GME repository.
- The policies associated with managing the pool have been defined in the GME repository.

11.2 Decommission Pool Of Compute Devices (Servers) Object

11.2.1 Summary

This use case describes the process of decommissioning a pool of compute components object, not necessarily the decommissioning of the servers themselves. When this use case is initiated the pool may or may not have members. Whilst the pool may no longer be available for actions to be undertaken on the membership, it will persist within the GME for record keeping and with respect to logging. Thus the decommissioning may be incomplete

11.2.2 Dependencies

- Remove Physical Server From Pool Object, see section 11.5.

11.2.3 Actor(s)/Stakeholder(s)

- System admin
- GME

11.2.4 Pre-Conditions

- The pool is available.

11.2.5 Initiation Of Use Case

This use case is initiated when the system admin no longer requires the pool to exist, for example because the pool is empty or is no longer relevant as a unit of manageability.

11.2.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) The system admin selects the pool within the GME repository.
2) The system admin removes each remaining member of the pool. See section 11.5, Remove Physical Server From Pool Object.
3) The system admin marks the pool object as unavailable in the GME repository.

Alternative Flows

Step 2: Ignore if there are no members of the pool.

Exceptional Flows

Repeat steps until 3) succeeds or the system admin gives up.

11.2.7 Success Criteria

The pool is marked as unavailable.

11.2.8 Post Conditions

- The pool object exists within the GME repository.
- The pool object has no members.
- The pool object is no longer available – that is no members may be added or removed, and no verbs may be applied to the pool.

11.3 Monitor & Manage Pool Object

11.3.1 Summary

This high-level use cases captures the general monitoring and management of the pool object and not the aggregate functions on the membership of the pool. Supported monitoring and management functions supported include, but are not necessarily limited to –

- Find grid object
- Set/get pool object properties/attributes
- Set/get pool membership constraints
- Set/get pool related policies
- Set/get pool verbs
- Set/get pool roles

11.3.2 Dependencies

11.1 Create Pool Of Compute Devices (Servers) Object

11.2 Decommission Pool Of Compute Devices (Servers) Object

11.4 Add Physical Server To Pool Object

11.5 Remove Physical Server From Pool Object

11.6 Grant Admin Rights To Pool

11.7 Rescind Admin Rights From Pool

11.8 Set Pool Derived Policy

11.9 Set Pool Constraints

11.10 Set Pool Configuration

11.11 Get Pool Policy

11.12 Get Pool Configuration

11.13 Get Pool Billing & Accounting

11.14 Select From Pool Members

11.15 Provision Firmware To Pool

11.16 Decommission Firmware On Pool

11.17 Provision OS To Pool

11.18 Decommission OS On Pool

11.19 Set State Of Pooled Servers

11.20 Get State Of Pooled Servers

11.21 Provision Patch To Pool

11.22 Decommission Patch From Pool

11.3.3 Actor(s)/Stakeholder(s)

- Installation Engineer
- Server Admin
- Network Admin
- Server Admin
- Security Admin
- GME

11.3.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.

11.3.5 Initiation Of Use Case

The required daemons are started/configured on the server, and metadata about valid pool definitions is added to the GME repository. Pools to be managed are operational with proper patches and firmware and their policies, configurations and constraints set appropriately.

11.3.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Initiate desired monitor/manage functionality (such as get/set/change pool configurations and policies).
2) Update the GME information repository to reflect the new Pool State.
3) Send a notification to the system admin

Alternative Flows

Step 1: If the necessary monitor/manage operation is already executional, then the corresponding steps may be skipped and a message is sent to the system admin.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin, as specified by a policy engine.

11.3.7 Success Criteria

Desired monitor/manage functionality (such as get/set/change pool configurations and policies) is effectively delivered and the GME information repository is updated to reflect the modified Pool.

11.3.8 Post Conditions

Pool/s with new specified configurations/policies are operational and are being monitored effectively.

11.4 Add Physical Server To Pool Object

11.4.1 Summary

This use case describes the process involved in adding a physical compute device (server) to a pool.

11.4.2 Dependencies

This use case depends upon the following use cases –

- Use case 2.4, Provision Physical Compute Device

11.4.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.4.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.

11.4.5 Initiation Of Use Case

The required daemons are started/configured on the server, and metadata about addition of the specified server is added to the GME repository.

11.4.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Provision the server as in use case 2.4

2) Configure, and run necessary daemons on the server node

3) Update the GME information repository

4) Send a notification to the system admin

Alternative Flows

Step 1: If the server is provisioned, this can be skipped.

Step 2: If the necessary daemons are already configured/running, then the corresponding steps may be skipped. If the server is already a member of the specified pool, this info is logged, and a message is sent to the system admin. If the server is already a member of another pool, an error message is generated, and the info is logged, and a message sent to the system admin.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin, as specified by a policy engine.

11.4.7 Success Criteria

The specified physical server has been added to the specified pool object, required daemons have been configured/started on the server, the GME repository updated, and a notification sent to the system admin.

11.4.8 Post Conditions

The server is now a part of a pool.

11.5 Remove Physical Server From Pool Object

11.5.1 Summary

This use case describes the process involved in removing a physical server from a Pool Object.

11.5.2 Dependencies

This use case depends upon the following use cases –

- Use case 9.5, Decommission Operating System Instance

11.5.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

11.5.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.5.5 Initiation Of Use Case

The applications running on the server are stopped, and the metadata about the removal of the specified server is updated in the GME repository.

11.5.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.

2) Verify that the physical server to be removed belongs to the Pool.

3) Gracefully shut down the Physical server to be removed, as described in use case 9.5

4) Remove the Physical Server from the Pool.

5) Update the Pool Configuration State Vector in the GME repository.

6) Send a notification to the system admin.

Alternative Flows

Step 1: If the physial server is non-operational already due to hardware/power failure, update the Pool Configuration State vector in the GME repository and start the revovery procedures, if mandated by the SLAs.

Exceptional Flows

If any steps (1-6) in the basic flow fail, a notification is sent to the system admin and GME, as specified by a policy engine.

11.5.7 Success Criteria

The Pool Configuration State Vector (which denotes the constituent resources and their states) has been updated to the GME repository and a notification sent to the system admin.

11.5.8 Post Conditions

The physical server is removed from the Pool and the Pool State vector is updated.

11.6 Grant Admin Rights To Pool

11.6.1 Summary

Pool of compute devices is formed. This use case grants administration rights such as pool membership/s, access and operations rights to the Pool.

11.6.2 Dependencies

None

11.6.3 Actor(s)/Stakeholder(s)

- GME
- System admin

11.6.4 Pre-Conditions

- GME exist
- GME repository exist
- The various grid components, as enumerated in the DAG are manageable by the GME.
- All SLOs for application services have been defined
- All SLOs and constraints for the grid components have been defined
- Use of the grid component and its intended deployment has been appropriately licensed or will be appropriately licensed in parallel to the provisioning process
- The pool has been created and is available.

11.6.5 Initiation Of Use Case

After the Pool of compute devices is formed, the Pool is set to get admin rights to facilitate operations such as adding servers, applying OS, firmware and patches and pool state management.

11.6.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.

2) Make sure the Pool is operational.
3) Update Pool Metadata to reflect new Pool Admin Capabilities.
4) Add the updated Pool Metadata to the GME repositoty.
5) Send a notification to the system admin.

Success Criteria

- Pool has desired Admin Rights.
- GME repository is updated

11.6.7 Post Conditions

Pool Admin is capable of carrying out essential pool maintainence and upgrade activities

11.7 Rescind Admin Rights From Pool

11.7.1 Summary

Pool of compute devices is formed and has been granted the admin rights. This use case rescinds administration rights such as pool membership/s, access and operations rights from the Pool, rendering the Pool to be administered by an entity other than itself.

11.7.2 Dependencies

None

11.7.3 Actor(s)/Stakeholder(s)

- GME
- System admin

11.7.4 Pre-Conditions

- GME exist
- GME repository exist
- The pool has been created, is available and has administration rights.

11.7.5 Initiation Of Use Case

Pool is being set to be administered by a process such as GME or another Pool Admin.

11.7.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Update Pool Metadata to reflect cancellation of Pool Admin Capabilities.
4) Add the updated Pool Metadata to the GME repositoty.
5) Send a notification to the system admin.

11.7.7 Success Criteria

- Pool has Admin Rights annuled.
- GME repository is updated.

11.7.8 Post Conditions

Pool is not self-administered and is taken care by an entity (such as GME or another Pool administrator) other than itself.

11.8 Set Pool Derived Policy

11.8.1 Summary

GME receive high priority service call from a mission critical application. GME identifies the pool of existing objects that match's the application service SLO requirements and repurpose the pool object to run high priority application services call on existing OS media and associated GME Grid components .

11.8.2 Dependencies

None

11.8.3 Actor(s)/Stakeholder(s)

- GME
- System admin

11.8.4 Pre-Conditions

- GME exist
- GME repository exist
- The various grid components, as enumerated in the DAG are manageable by the GME.
- All SLOs for application services have been defined
- All SLOs and constraints for the grid components have been defined
- Use of the grid component and its intended deployment has been appropriately licensed or will be appropriately licensed in parallel to the provisioning process
- The pool has been created and is available

11.8.5 Initiation Of Use Case

The GME derive the desire components SLO and determine to commission the existing pool object and associated operating systems, etc that required hosting a service in order to meet the application services component SLOs.

11.8.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME derive the application services component SLOs and search GME repository for required resources
2) GEM identify existing pool object matching the application services component SLOs (Performance, scalability and HA)
3) GME decommission the current service, no change to OS media
4) GME commission new services to pool object
5) GME update the repository to reflect the services is now running on the pool
6) In the event of failure during commissioning new service GME should restart the process or remove the failed component from the pool and add new component to match the desired SLOs

11.8.7 Success Criteria

- New services is commissioned and operational
- GME repository is updated
- Required services SLOs are achieved

11.8.8 Post Conditions

- GME to monitor the pool Object to ensure that new service is running on pool object and service SLOs are being met
- Report any service SLOs violation

11.9 Set Pool Constraints

11.9.1 Summary

GME receive application services request with specific SLOs, this application required 10 instances of database running on minimum set of 5 and maximum set of 10 servers equipped with 2 CPU and 8 gigabytes of Memory with associated OS and other Grid components. GME determine new pool of compute device is required to meet the application service SLOs.

11.9.2 Dependencies

- Discover Physical Compute Device (Server)
- Configure Physical Compute Device (Server)
- Start (Power-On) Physical Compute Device
- Provision Physical Compute Device
- Create Pool Of Compute Devices (Servers) Object
- Provision OS to pool object
- Commission Service

11.9.3 Actor(s)/Stakeholder(s)

- GME
- System admin
- Database Administrator

11.9.4 Pre-Conditions

- GME exist
- GEM repository exist
- All SLOs and constrain for application services has been defined
- All SLOs and constraints for the grid components have been defined

11.9.5 Initiation Of Use Case

GME receive new application services call, search the repository to find pool object (Server) with 5 t0 10 dual CPU and 8 GB RAM). After the search the GME determine to create pool of computing device in order to meet the application service call SLOs.

11.9.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME receive application service call with specific constrain
2) GME check the repository to find matching compute pool/device
3) GME determine to create pool of compute device based on set of application services components constrain (Describe above)
4) See “Create Pool Of Compute Devices (Server) section 7.43
5) System admin provision the OS Media to Pool of Compute Device
6) GME commission the service
7) GME check ensure that services is up and running and service SLOs constrain has been met
8) GME update the repository

Alternative Flows

GME determine that there are available set of pool object that meet the application services call SLO’s, GME decommission the service currently running on pool and commission the new services, see section 7.52

11.9.7 Success Criteria

The pool of new Compute Device (Server) created, OS media deployed and all 10 instances of database are running on servers configured to meet the application service SLOs.

11.9.8 Post Conditions

- GME verified that application service SLOs are met
- GME update the repository with pool configuration and mark this pool object commissioned/unavailable

11.10 Set Pool Configuration

11.10.1 Summary

This use case describes the process involved in Setting Pool Configuration.

11.10.2 Dependencies

This use case depends upon the following use cases –

- 11.4 Add Physical Server To Pool Object
- 11.6 Grant Admin Rights To Pool
- 11.8 Set Pool Derived Policy
- Set Pool Constraints

11.10.3 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.10.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- The pool has been created.

11.10.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.10.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Add metadata about pool and its constituents to the GME repository.
4) Create Pool Configuration State Vector with State Vector Values.
5) Add the Pool Configuration State Vector to the GME repository.
6) Send a notification to the system admin.

Alternative Flows

Step 1: If the GME repository already has the correct Pool Configuration Information, then the corresponding steps may be skipped. If the Pool Configuration State Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin and GME.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.10.7 Success Criteria

The Pool Configuration State Vector (which denoted the constituent resources and their states) has been added to the GME repository and a notification sent to the system admin.

11.10.8 Post Conditions

The pool configuration information is available to the GME.

11.11 Get Pool Policy

11.11.1 Summary

This use case describes the process involved to Get Pool Policy Information.

11.11.2 Dependencies

None

11.11.3 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

11.11.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- The pool has been created.

11.11.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.11.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Present the Get Pool Policy Query to the GME.
4) Obtain Pool Policy State Vector with State Vector Values.

5) Send Pool Policy Vector Information to the requesting admin/s.

Alternative Flows

Step 1: If the Pool Policy Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin/s and GME after the successful upgrade of the Pool Configuration State Vector.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.11.7 Success Criteria

The Pool Policy information is obtained and sent to the requesting admin/s. Pool Policy vector consists of policy information related to Security policy, performance administration policy and policy information related to billing and accounting amongst others.

11.11.8 Post Conditions

The pool policy information is available to the requesting admin/s.

11.12 Get Pool Configuration

11.12.1 Summary

This use case describes the process involved to Get Pool Configuration.

11.12.2 Dependencies

None

11.12.3 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.12.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.12.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.12.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Present the Get Pool Configuration Query to the GME repository.
4) Obtain Pool Configuration State Vector with State Vector Values.

5) Send Pool Configuration State Vector Information to the requesting admin/s.

Alternative Flows

Step 1: If the Pool Configuration State Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin/s and GME after the successful upgrade of the Pool Configuration State Vector.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.12.7 Success Criteria

The Pool Configuration State Vector (which denoted the constituent resources and their states) information is obtained and sent to the requesting admin/s.

11.12.8 Post Conditions

The pool configuration information is available to the requesting admin/s.

11.13 Get Pool Billing & Accounting

11.13.1 Summary

This use case describes the process involved to Get Pool Billing & Accounting.

11.13.2 Dependencies

None

11.13.3 Actor(s)/Stakeholder(s)

- Hardware Technician
- System admin
- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.13.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.13.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.13.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.

3) Present the Get Pool Billing & Accounting Query to the GME.
4) Obtain Pool Billing & Accounting.
5) Send Pool Billing & Accounting Information to the utility admin/s.

Alternative Flows

Step 1: If the Pool Configuration is being modified due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin/s and GME after the successful upgrade of the Pool Configuration State Vector.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.13.7 Success Criteria

The Pool Billing & Accounting information is obtained and sent to the requesting admin/s.

11.13.8 Post Conditions

The Pool Billing & Accounting information is available to the requesting utility admin/s.

11.14 Select From Pool Members

11.14.1 Summary

This use case describes the process involved in Setting Pool Configuration.

11.14.2 Dependencies

None

11.14.3 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.14.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.14.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.14.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Pool Components/Resources Required Query to the GME.
4) GME Runs Get Pool configuration Query.

5) GME obtains the information about available Pool Components.

6) Select the Pool Components/Resources Required.

7) Send a notification to the system admin.

Alternative Flows

Step 1: If the GME repository already has the correct Pool Configuration Information, then the corresponding steps may be skipped. If the Pool Configuration State Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin and GME.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.14.7 Success Criteria

GME obtains the Pool Configuration Information and presents a list of available Pool Components with Desired Characteristics; the Application Server/s select from the available Pool Components.

11.14.8 Post Conditions

Selection of Pool Components by the Application Servers.

11.15 Provision Firmware To Pool

11.15.1 Summary

This use case captures the process associated with the installation (provisioning) of firmware to a pool of target grid components. Such a target could be a server (OBP, firmware or BIOS updates), a network device (such as a router) or a storage device (such as a RAID controller or SAN switch). No assumptions are made with regards to the prior provisioning of the firmware media, so that use case may need to be called.

11.15.2 Dependencies

This use case builds upon the following use cases:

- Provision Firmware: See section 3.3.

11.15.3 Actor(s)/Stakeholder(s)

- Primary Actor is the GME
- Secondary actors which may act on behalf of the GME include:
  - System Admin
  - Data Center Admin
  - Network Admin
  - Storage admin
  - Security Admin

11.15.4 Pre-Conditions

- The various grid components, as enumerated in the are manageable by the GME.
- All SLOs and constraints for the grid components have been defined.
- Use of the grid component and its intended deployment has been appropriately licensed or will be appropriately licensed in parallel to the provisioning process.

The use case also assumes the following pre-requisites:

- The pool has been created as described in Create a Pool of Components. See section 11.1.
- Physical grid components have been added to the pool as described in Add Physical Server (grid component) to Pool Object. See section 11.4.
- Administrative rights have been granted as described in Grant Admin Rights to Pool. See section 11.6.

11.15.5 Initiation Of Use Case

The GME determines the grid components in the pool require new or updated firmware.

11.15.6 Flow Of Events (Basic, Alternative And Exceptional)

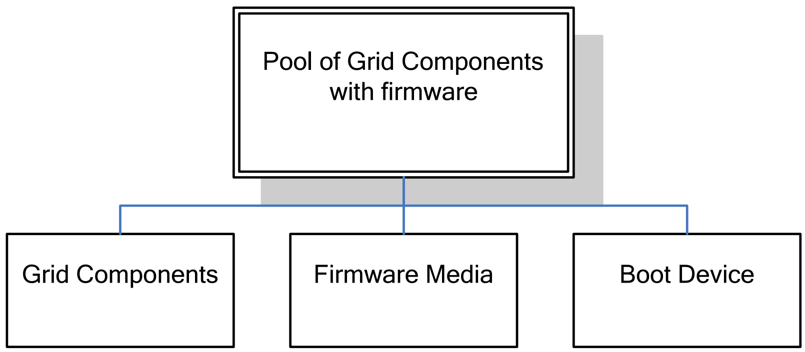


Figure 5 - Provision Firmware to Pool DAG

Basic Flow

Referring to Figure 1 – Provision firmware to Pool DAG above. For each grid component in the pool, refer to the flow described in the relevant use cases as described below.

1) Follow the flow in use case Provision Firmware (section 3.3)
2) Update Pool Configuration in GME

Update the pool configuration in the GME repository to reflect the new firmware running on each grid component in the pool

Alternative Flows

None

Exceptional Flows

In the event provisioning firmware to a grid component fails, the GME should respond by removing that grid component from the pool and adding a replacement component to the pool. The GME would then provision firmware to the replacement grid component. If no suitable replacement components are available, the operation should either fail, or succeed in a limited manner, depending on pool policies.

11.15.7 Success Criteria

The grid components in the pool are all running the required firmware versions. All components in the pool are activated. Service level metrics for the pool are met.

11.15.8 Post Conditions

The grid components in the pool are active and available for the GME to deploy services onto.

11.16 Decommission Firmware On Pool

11.16.1 Summary

This use case shows the process associated with the removal of firmware from a pool of target grid components. In reality, such a process rarely exists, as firmware is usually overwritten.

11.16.2 Dependencies

This use case builds upon the following use cases:

- Decommission Firmware: See section 3.4

11.16.3 Actor(s)/Stakeholder(s)

- Primary actor is GME
- Secondary actors which may act on behalf of the GME include:
  - Data Center Admin
  - Network Admin
  - Storage admin
  - System admin
  - Security Admin

11.16.4 Pre-Conditions

- The workload or service hosted by the pool has terminated
- There is no longer a reservation for this pool in the GME respository

11.16.5 Initiation Of Use Case

The GME detects that the reservation for this pool that assigned it to a service has been cancelled, and policy for this pool states that firmware should be removed from a grid component in the pool once it is no longer in use.

11.16.6 Flow Of Events (Basic, Alternative And Exceptional)

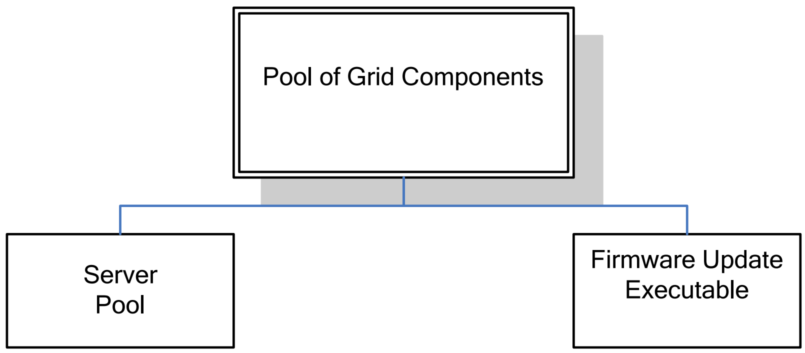


Figure 1 - Decommission Firmware from Pool DAG

Basic Flow

1) Decommission firmware for each grid component in pool as described in use case 3.4
2) Update Pool Configuration in GME

   Update the pool configuration in the GME repository to reflect the OS is no longer running on any servers in the pool.

Alternative Flows

None

Exceptional Flows

GME will verify no services are active on any grid component. If a grid component is hosting an active service, the operation on that grid component will fail.

11.16.7 Success Criteria

All grid components in the pool have been shut down, and none are have any firmware.

11.16.8 Post Conditions

The grid components in the pool are ready to be provisioned with new firmware

11.17 Provision OS To Pool

11.17.1 Summary

Servers are added to a pool with the intent of running some workload within the pool. This will require the servers be booted with an operating system suitable for the workload to be supported. This use case describes the activity associated with booting the servers in the pool with an appropriate operating system (type, version, and patches).

11.17.2 Dependencies

This use case builds upon the following use cases:

- Use case 9.3, Provision an OS Instance to a Server

11.17.3 Actor(s)/Stakeholder(s)

- Primary Actor is the GME
- Secondary actors which may act on behalf of the GME include:
  - Data Center Admin
  - Network Admin
  - Storage admin
  - System admin
  - Security Admin

11.17.4 Pre-Conditions

- The various grid components, as enumerated in the DAG are manageable by the GME.
- All SLOs and constraints for the grid components have been defined.
- Use of the grid component and its intended deployment has been appropriately licensed or will be appropriately licensed in parallel to the provisioning process.

The use case also assumes the following pre-requisites:

- The pool has been created as described in Create a Pool of Components.
- Physical servers have been added to the pool as described in Add Physical Server to Pool Object.
- Administrative rights have been granted as described in Grant Admin Rights to Pool.

11.17.5 Initiation Of Use Case

The GME determines additional servers, and associated operating systems, are required to host a service inorder to meet SLOs.

11.17.6 Flow Of Events (Basic, Alternative And Exceptional)

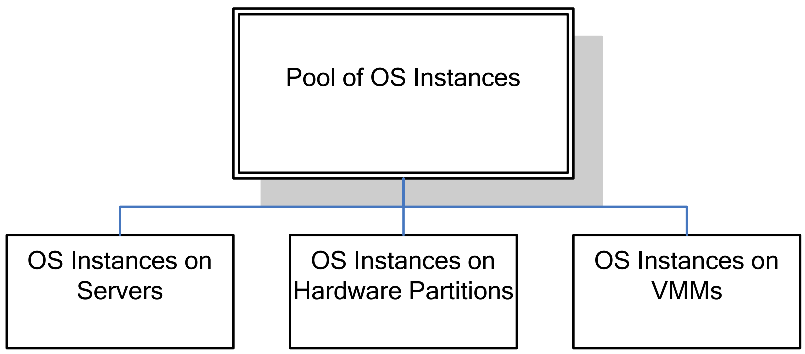


Figure 6 - Provision OS to Pool DAG

Basic Flow

Referring to Figure 6 – Provision OS to Pool DAG above. For each server in the pool, refer to the flow described in the relevant use cases as described below.

1) Provision OS to each server in pool as described in use case 9.3
2) Update Pool Configuration in GME

   Update the pool configuration in the GME repository to reflect the OS is now running on additional servers in the pool.

Alternative Flows

Some servers in the pool may already be running the appropriate OS, and may only need to be restarted.

Exceptional Flows

In the event provisioning an OS to a server fails, or the server fails to meet metric objectives or generates other errors, the GME should respond by removing that server from the pool and adding a replacement server to the pool. The GME would then provision an OS to the replacement server. If no suitable replacement servers are available, the operation should either fail, or succeed in a limited manner, depending on pool policies.

11.17.7 Success Criteria

The servers in the pool are all running the required OS types, versions and patches. All servers in the pool are activated. Service level metrics for the pool are met.

11.17.8 Post Conditions

The serves in the pool are active and available for the GME to deploy services onto.

11.18 Decommission OS On Pool

11.18.1 Summary

This use case will decommission the operating system from a pool of servers that are no longer needed to host a service. This will prepare them for reuse by another service.

11.18.2 Dependencies

This use case builds upon the following use cases:

- Use case 9.5, Decommission OS Instance From Server:

11.18.3 Actor(s)/Stakeholder(s)

- Primary actor is GME
- Secondary actors which may act on behalf of the GME include:
  - Data Center Admin
  - Network Admin
  - Storage admin
  - System admin
  - Security Admin

11.18.4 Pre-Conditions

- The workload or service hosted by the pool has terminated
- There is no longer a reservation for this pool in the GME respository

11.18.5 Initiation Of Use Case

The GME detects that the reservation for this pool that assigned it to a service has been cancelled.

11.18.6 Flow Of Events (Basic, Alternative And Exceptional)

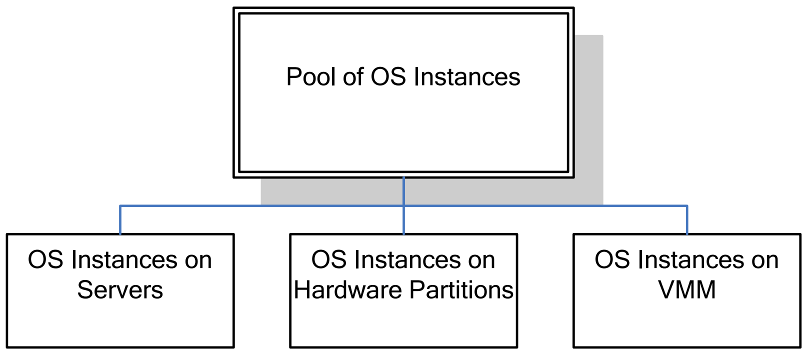


Figure - Decommision OS from Pool DAG

Basic Flow

1) Decommission OS for each server in pool as described in Use case 9.5
2) Update Pool Configuration in GME

   Update the pool configuration in the GME repository to reflect the OS is no longer running on any servers in the pool.

Alternative Flows

Some servers in the pool may already have had their operating systems decommissioned and may be shut down.

Exceptional Flows

GME will verify no services are active on any server. If a server is hosting an active service, the operation on that server will fail.

11.18.7 Success Criteria

All servers in the pool have been shut down, and none are associated with an operating system

11.18.8 Post Conditions

The servers in the pool are ready to be provisioned to another operating system and service.

11.19 Set State Of Pooled Servers

11.19.1 Summary

This use case describes the process involved in Setting state of Pooled Servers.

11.19.2 Dependencies

None

11.19.3 Actor(s)/Stakeholder(s)

- System admin
- Network Admin
- Storage Admin
- Security Admin
- GME

11.19.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.19.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.19.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Formulate Pool State vector information consisting of states of the servers from the Pool.

4) Update the Pool State Vector to the GME repository.

5) Send a notification to the system admin.

Alternative Flows

Step 1: If the GME repository already has the correct Pool Servers State Information, then the corresponding steps may be skipped. If the Pool Configuration State Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin and GME.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.19.7 Success Criteria

The Pool Servers State Information (which denotes the constituent servers and their states) has been added to the GME repository and a notification sent to the system admin.

11.19.8 Post Conditions

The Pool Servers State Information is available to the GME.

11.20 Get State Of Pooled Servers

11.20.1 Summary

This use case describes the process involved to Get State Of Pooled Servers.

11.20.2 Dependencies

None

11.20.3 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.20.4 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The GME daemons exist on the device.
- Desired Pool is formed with proper resources.

11.20.5 Initiation Of Use Case

The required daemons are started/configured on the grid.

11.20.6 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) Verify the Pool Definition and its current implementation.
2) Make sure the Pool is operational.
3) Present the Get State Of Pooled Servers Query to the GME.

4) Obtain Pool Servers State Vector with State Vector Values.
5) Send Pool Servers Vector Information to the requesting admin/s.

Alternative Flows

Step 1: If the Pool Servers State Vector is being upgraded due to changes to the Pool Structure, this info is logged, and a message is sent to the system admin/s and GME after the successful upgrade of the Pool Servers State Vector.

Exceptional Flows

If any steps fail then they are repeated until successful, or a notification is sent to the system admin and GME, as specified by a policy engine.

11.20.7 Success Criteria

The Pool Servers State information is obtained and sent to the requesting admin/s. Pool State vector consists of information related to Pool constituent servers and their states.

11.20.8 Post Conditions

The pool policy information is available to the requesting admins.

11.21 Provision Patch To Pool

11.21.1 Summary

This use case describes the process associated with the application of a patch to a Pool. Nothing is assumed about the starting state of the component or about the grid components.

Dependencies

- Use case 10.3, Provision patch

11.21.2 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.21.3 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The patch media exists in the GME.
- The pool exists

11.21.4 Initiation Of Use Case

Provisioning of a patch to the Pool is requested. This may be for any reason, ranging from a notification from a software vendor to an in-house patch for some known problems.

11.21.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME updates the information repository and makes the Pool to be decommissioned as unavailable during the decommissioning process.
2) GME stops any running applications, and updates repository. (This may be another use case.)

3) GME starts the specified Pool with the right privilege level, if needed.
4) GME updates information repository about dependencies, relevant reasons etc.
5) For each server in the pool, follow the steps of use case 10.3, provision patch.
6) GME checks the log files for correct provisioning.
7) GME repeats steps 5) - 6) if the decommissioning was not successful as specified by a policy engine.
8) GME updates its repository.
9) GME makes Pool available for use.
10) GME sends a notification to the subscribed users about the provisioned patch.
11) GME restarts applications on the patched Pool.

Alternative Flows

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

11.21.6 Success Criteria

The new patch has been installed on the specified Pool in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

11.21.7 Post Conditions

The new patch has been installed on the specified Pool in the grid, and relevant info added into the repository of the GME.

11.22 Decommission Patch From Pool

11.22.1 Summary

This use case describes the process by which a patch is rolled back. This is usually done through rolling back to a backed up version of the OS.

Dependencies

- Use case 10.4, Decommission patch

11.22.2 Actor(s)/Stakeholder(s)

- Network Admin
- Storage Admin
- System admin
- Security Admin
- GME

11.22.3 Pre-Conditions

- The GME exists.
- The GME repository exists.
- The patch exists on the Pool.
- The pool exists

11.22.4 Initiation Of Use Case

Decommissioning of a patch is requested. This may be for any reason, e.g., software compatibility of an existing application.

11.22.5 Flow Of Events (Basic, Alternative And Exceptional)

Basic Flow

1) GME updates the information repository and makes the Pool to be decommissioned as unavailable during the decommissioning process.

2) GME stops any running applications, and updates repository. (This may be another use case.)
3) GME starts the specified Pool with the right privilege level, if needed.
4) GME updates information repository about dependencies, relevant reasons etc.
5) For each server in the pool, follow the steps in use case 10.4, Decommission patch.
6) GME checks the log files for correct decommissioning.
7) GME repeats steps 5-6 if the decommissioning was not successful as specified by a policy engine.
8) GME updates its repository.
9) GME makes decommissioned Pool available for use.
10) GME sends a notification to the subscribed users about the decommissioned patch.
11) GME restarts applications on the patched Pool.

Alternative Flows

Try a manual decommission of the patch.

Exceptional Flows

If any steps fail then they are repeated for a specified number of times until successful, or a notification is sent to the system admin, as specified by a policy engine.

11.22.6 Success Criteria

The patch has been decommissioned on the specified Pool in the grid, and relevant info added into the repository of the GME, and a notification sent to the subscribed stakeholders.

11.22.7 Post Conditions

The patch has been decommissioned on the specified node in the grid, and relevant info added into the repository of the GME.

Future Use Cases

Some of the topics planned for future versions of this document include:

- Additional sections of Chapter 6 (Provisioning Bootable Storage)
    - Provision Boot Disc/LUN—SAN
    - Decommission Boot Disc/LUN—SAN
    - Provision Boot Disc/LUN—NAS
    - Decommission Boot Disc/LUN—NAS
- Provisioning Virtual Machine Managers
- Provisioning Virtual Machines
- Additional sections of Chapter 9 (Provisioning Operating Systems)
    - Provision OS to Virtual Server
    - Decommission OS from Virtual Server
- Additional sections of Chapter 11 (Provisioning Pools of Servers)
    - Add Virtual Server to Pool Object
    - Remove Virtual Server from Pool Object
    - Provision VMM to Pool
    - Decommission VMM to Pool
- Provisioning Operating System Level Clustering
- Provisioning Databases
- Provisioning Application Servers
- Provisioning Web Servers
- Provisioning Load Balancers
- Provisioning Firewalls
- Provisioning Database Clusters
- Provisioning Application Server Clusters
- Provisioning Database Server Tiers
- Provisioning Web Server Tiers
- Provisioning A Multi-Tier Service

