

Cisco UCS Manager Architecture

What You Will Learn

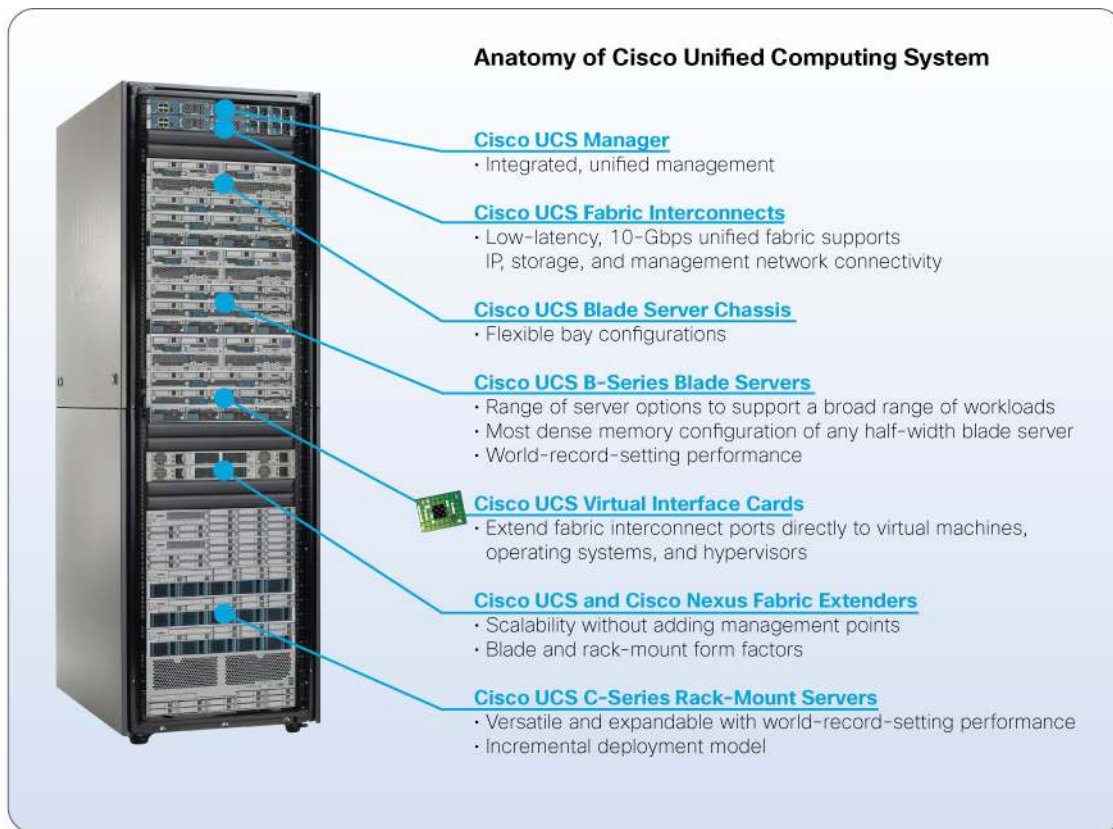
Cisco UCS™ Manager provides unified, embedded management of all software and hardware components of the Cisco Unified Computing System™ (Cisco UCS) across multiple chassis, rack-mount servers, and thousands of virtual machines. Cisco UCS Manager is a device manager that manages Cisco UCS as a single entity through an intuitive GUI, a command-line interface (CLI), or an XML API for comprehensive access to all Cisco UCS Manager functions.

The management solution of an integrated data center infrastructure must control the network and storage resources required by the computing platform. Server management software typically becomes more complex with the need for various products to control and maintain the integrated infrastructure. Each component vendor has its own management interface and communication protocols. However, Cisco UCS Manager, which runs embedded in the fabric interconnect, manages all components of Cisco UCS through a single pane. Managing the server, networking, and storage resources with a unified approach greatly decreases complexity, providing the capability to deploy and alter applications quickly with the innovative use of policies, role-based access control (RBAC), service profiles, and templates.

Cisco UCS Architectural Components

Cisco UCS has many physical components, all managed by Cisco UCS Manager (Figure 1). Cisco UCS is a next-generation data center platform that unifies computing, networking, storage access, and virtualization resources into a cohesive system designed to reduce total cost of ownership (TCO) and increase business agility. The system integrates a low-latency, lossless 10 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. It is a scalable, multichassis platform in which all resources are integrated into a unified management domain.

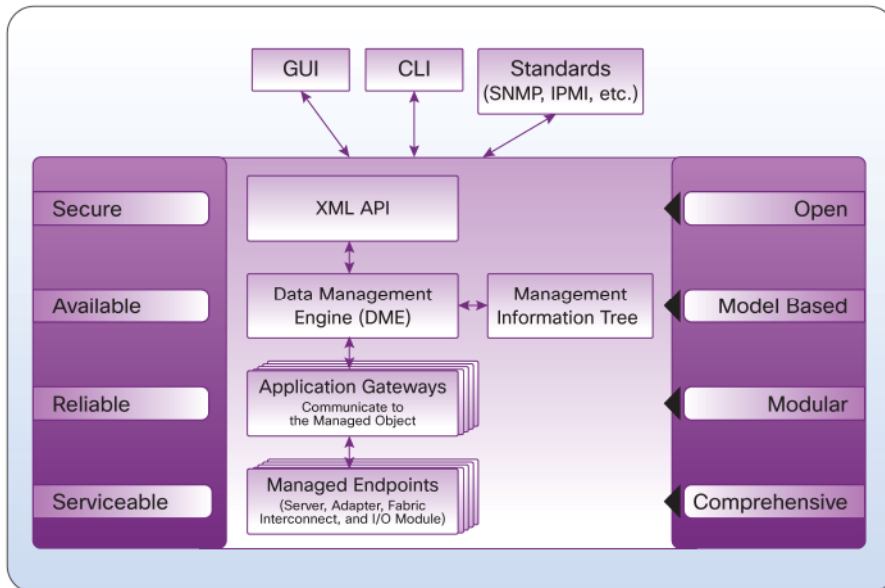
Figure 1. Cisco UCS Components



Cisco UCS Manager Architecture System Components

The architecture of Cisco UCS Manager consists of multiple layers with well-defined boundaries. External interfaces provide communication with the outside world (Figure 2). The data management engine (DME) is the central service that manages the components of Cisco UCS. Application gateways act as hardware abstraction layers between the DME and the managed endpoints. The endpoints are the actual devices or entities that are managed by Cisco UCS Manager, but they are not considered as part of Cisco UCS Manager itself.

Figure 2. Model-Based Framework in Cisco UCS Manager



Cisco UCS Manager exposes all of its functions through a powerful native XML API. All operations, including the Cisco UCS Manager’s GUI and CLI, use this XML API. The management information tree is exposed in its entirety through the API, permitting users to modify management information as needed. Cisco’s GUI and CLI for Cisco UCS Manager also use the XML API. Users can create their own front-end user interfaces with the API. For example, organizations that want to deploy Cisco UCS in a multi-tenant environment can create a custom GUI for each tenant. Others may want to use the API to connect to a configuration management database (CMDB) or to third-party management and provisioning tools. Cisco management ecosystem partners have built robust solutions using the API to integrate Cisco UCS management into their software products.

Data Management Engine

The DME is the central component in Cisco UCS Manager and consists of multiple internal services. The DME is the only component in Cisco UCS that stores and maintains states for the managed devices and elements. The DME is the authoritative source of configuration information. It is responsible for propagating configuration changes to the devices and endpoints in the form of managed objects. Managed objects contain the desired configuration and the current state of a corresponding endpoint.

Administrators make changes to MOs in the management information tree through the XML API. These changes are validated by the DME and propagated to the specific endpoint. For example, if an operator initiates a server “power-on” request through the GUI, when the DME receives the request, it validates it, and if the request is valid, the DME makes the corresponding state change on the server objects in the management information tree. This state change is then propagated to the server through the appropriate application gateway.

Management Information Tree

All the physical and logical components that comprise Cisco UCS are represented in a hierarchical management information model referred to as a management information tree. The Cisco UCS Manager information model is the logical abstraction of all hardware and software components managed in Cisco UCS. Each node in the information tree represents a managed object or group of objects that contains its configuration and operational

state. The state information for a managed object includes both its administrative state (how it is configured) and its operational state (how it is running). Hierarchically organizing the management information reflects a logical containment of the objects. It also facilitates parallel and asynchronous management operations, thereby improving the consistency of the operations and the efficiency of the transactions on individual managed objects and on subtrees of managed objects. For example, it enables administrators to configure a complete Cisco UCS instance in a single transaction.

Application Gateways

Application gateways are agents used by the DME to propagate changes to the endpoints. They also report the system state from the endpoints to the DME. An application gateway is a module that converts management information (such as configuration, statistics, and faults) from its native representation into the form of a managed object. The application gateway is the hardware abstraction layer that abstracts the object model from the managed device or entity. Application gateways implement the platform-specific details of each of the managed endpoints. In Cisco UCS Manager, the application gateways are implemented for the Cisco NX-OS Software chassis, blades, ports, host agents, and network interface cards (NICs).

Application gateways compare the administrative state of a managed object with the operational state of the managed endpoint and take appropriate action. Application gateways are stateless, and their stimuli (entities that can change the configuration state of a managed object) are unchanged in value when multiplied by themselves, making Cisco UCS Manager a robust framework for configuration management, particularly in the event of failures. When an application gateway fails to configure an endpoint, that failure is reflected in the operational state of the managed object. At the end of any transaction, whether it was successful or not, all affected objects are inspected, and appropriate managed object rules are applied.

With this model-based framework, Cisco UCS Manager separates business logic from platform implementation. This approach lets programmers develop the business logic independent of the platform implementation, thus permitting rapid support for new platforms. This approach also helps simplify software maintenance, since model logic errors should be easily distinguished from platform implementation errors.

Managed Endpoints

Managed endpoints are resources within Cisco UCS that are managed by the Cisco UCS Manager [Figure 3]. The DME within Cisco UCS Manager interacts with the endpoints through the application gateways. These resources are LAN, SAN, and server related.

With Cisco UCS Manager, all configuration changes are first applied to the managed object in the information model by the DME and are then applied to the actual managed endpoint by the appropriate application gateway. These deployments of management to managed objects are fully transactional and abide by the atomic, consistent, isolated, and durable (ACID) requirements of a standard database transaction. With the DME, the deployment of the administrative state to the managed endpoint is just a side-effect of an information model transaction. However, the deployment of the administrative state to the managed endpoint is not transactional in nature. This is an important distinction between the model and the managed endpoint implementations.

Figure 3. Power On Sequence

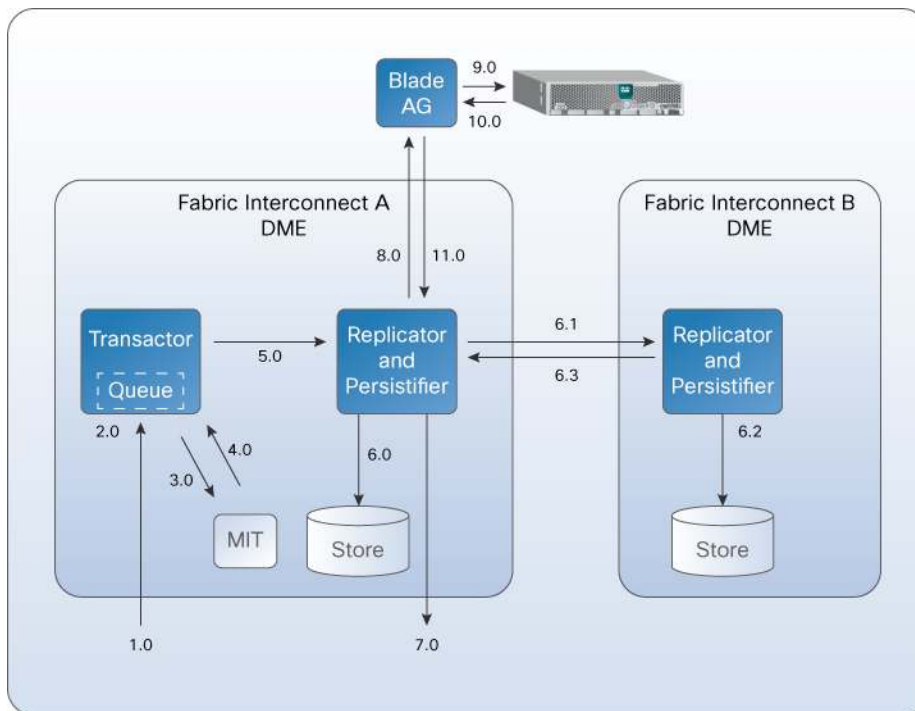


Table 1. Server Boot Request

| Step | Command/Process | Administrative Power State of MO (Server) | Operational Power State of MO (Server) |
|------|-----------------------------------------------------------------|-------------------------------------------|----------------------------------------|
| 1.0 | CMB Request: Boot Server | Down | Down |
| 2.0 | Request gets queued | Down | Down |
| 3.0 | State change in Model Information Tree | Up | Down |
| 4.0 | Transaction complete | Up | Down |
| 5.0 | Pass change information and boot request stimuli | Up | Down |
| 6.0 | Persistify the state change of MO to local store | Up | Down |
| 6.1 | Send state change information to peer DME | Up | Down |
| 6.2 | Persify the state of MO to peer's local store | Up | Down |
| 6.3 | Reply with success (replication and persistification) | Up | Down |
| 7.0 | CMO: Response and external notification | Up | Down |
| 8.0 | Apply reboot stimuli | Up | Down |
| 9.0 | Instruct CIMC to power on server | Up | Down |
| 10.0 | Reply from CIMC, server power on success | Up | Down |
| 11.0 | Reply, reboot stimuli success, pass new power state information | Up | Up |

Security

This flexibility in client interfaces facilitates role-based management because it enables IT administrators from network, storage, and server domains to work with the terminology with which they are most familiar. Users may

choose to adapt their tools for specific needs. For example, one administrator may be responsible for servers, one for SAN configuration, one for LAN configuration, and one for managing the rest of the Cisco UCS infrastructure. In large-scale environments, teams of users may be responsible for managing each of these components across the entire organization. Within these teams, some individuals may specialize in the management of subcomponents (for example, on the IT management team, one user may be responsible only for virtual NIC [vNIC] configuration). Standard roles are built into Cisco UCS Manager, and you can also create custom roles. Each role has a corresponding set of privileges that control write access to server configuration, internal and border LAN configuration, internal and border SAN configuration, and configuration of other Cisco UCS components, including configuration of RBAC itself.

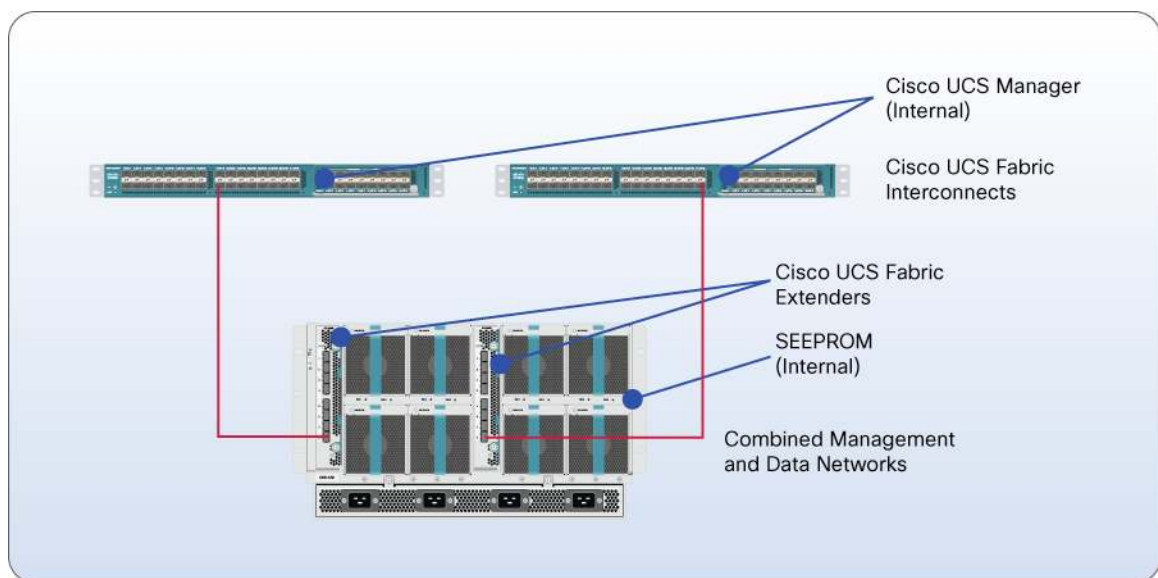
Larger-enterprise customers can divide their organizations hierarchically into suborganizations. A locale is a region within that hierarchy. A combination of locales and privileges can determine to what resources users have rights within suborganizations.

In addition, authentication of users can be performed from an external system such as Lightweight Directory Access Protocol (LDAP) (Active Directory [AD]), RADIUS, or TACACS; or internally using local accounts. Authorization is performed by the Cisco UCS Manager after the user has been authenticated.

Availability

Cisco UCS supports the use of two interconnected fabric interconnects for management redundancy and increased fabric interconnect throughput. A valid high-availability configuration includes a pair of interconnected fabric interconnects and two fabric extender modules per chassis, with one fabric extender attached to one fabric interconnect, and the other fabric extender attached to the other fabric interconnect, as shown in Figure 4.

Figure 4. High Availability Using Management Redundancy



Both Cisco UCS Manager instances can run as either the active or standby instances. (The data plane always remains active-active during normal production.) All Cisco UCS Manager processes running in these instances are monitored and restarted if they terminate abnormally by a monitoring process. A floating management IP address is configured on the active instance so that all GUI and CLI connections and management operations are forced to

initiate there. Configuration and operational state changes are then propagated over the private network from the active instance to the standby instance so that management information is synchronized. The Cisco UCS Manager instance acting as the active instance will remain so either until it fails over or until the administrator requests a role change (a switchover). Each instance periodically sends a heartbeat message to the other to indicate that it is healthy. Cisco UCS Manager detects and resolves heartbeat discontinuities by using the serial electronic erasable programmable read-only memory (SEEPROM) on the chassis.

Reliability and Serviceability

Since the DME is aware of the administrative and operational states of all managed endpoints, it notifies all interested client programs about state and configuration changes as they occur. The DME was designed to publish notifications to clients so that clients can maintain a consistent and up-to-date view of management information without having to poll Cisco UCS Manager for updates. GUI, CLI, and XML API clients can subscribe to events so that they reflect the most up-to-date state and configuration of the system at all times.

Cisco UCS Manager uses generic object notifications to manage events and faults. An event is a representation of something that momentarily occurred in the system; for example, insertion of a blade into a chassis is represented as an event. By contrast, a fault represents something that failed in the system; the failure of a fan module is represented as a fault. Events and faults are themselves objects that are managed by Cisco UCS Manager and are subject to the same base set of rules as other managed objects in Cisco UCS Manager. However, events and faults have additional rules that specifically apply to them. For example, event managed objects are immutable because they usually correspond to a momentary, nonpersistent condition that should be logged.

By contrast, fault managed objects are mutable because the operational state of a managed endpoint may transition between a failed state and a functioning state. Such a fault is known as a cleared fault. Cisco UCS Manager has a user-configurable policy for handling cleared faults. Cisco UCS Manager can automatically delete or retain cleared faults for a certain amount of time. For both events and faults, Cisco UCS Manager supports the concept of expiry or termination in the lifecycle of their corresponding managed objects.

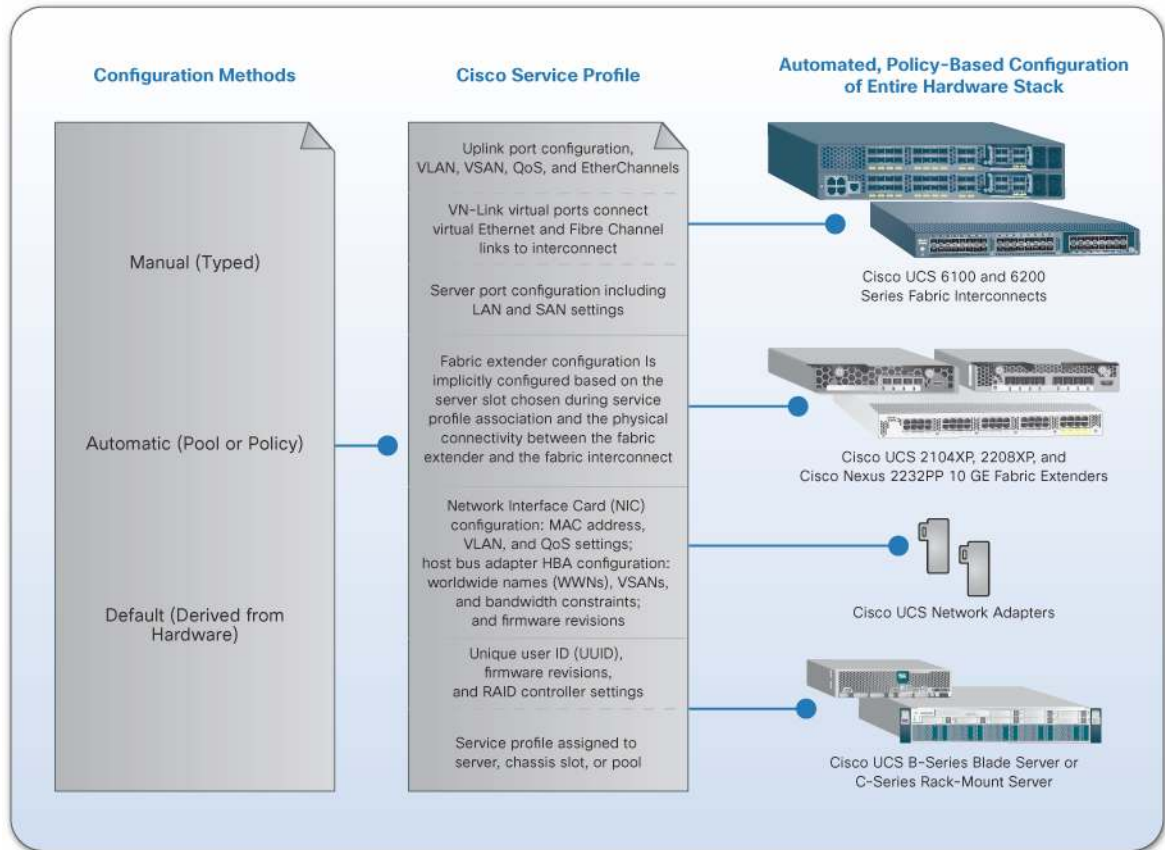
Modular Management

Cisco UCS Manager software is an embedded device manager for all Cisco UCS components. In essence, Cisco UCS Manager creates a unified management domain that serves as the central nervous system of Cisco UCS. In one sense, Cisco UCS Manager takes the place of several system management tools associated with traditional computing infrastructure by integrating computing, networking, storage access, and virtualization resources in a single cohesive system. However, Cisco UCS Manager leaves cross-system and heterogeneous device management to other system management tools and provides different APIs for those tools.

Policy-Based Management

A policy-based management approach allows Cisco UCS Manager to use the metadata of servers to abstract the state of the hardware. For example, the administrative state of blades is managed with service profiles. A service profile contains values for a server's property settings, including vNICs, MAC addresses, boot policies, firmware policies, and other elements. By abstracting these settings from the physical server to a service profile, you can deploy a service profile to any physical computing hardware in Cisco UCS. Furthermore, the service profile can, at any time, be migrated from one physical server to another. Figure 5 summarizes the information that is included in a service profile and shows how that information is populated in the service profile and ultimately associated with a server.

Figure 5. Service Profile Elements



A service profile is therefore the description of a logical server, and there is a one-to-one relationship between a service profile and a physical server. A service profile template is the blueprint for creating new service profiles. Using policies and pools that are defined by functional administrators, server managers can create service profiles. For example, a network administrator can define a pool of MAC addresses and policies such as quality of service (QoS) for a VLAN. A server administrator can use a MAC address from the pool to create a service profile.

A service profile can be associated with a physical server either manually or automatically (using a policy or the XML API). When a service profile is associated with a server, the DME performs all necessary changes to fulfill the request. Specifically, a thread of the DME process, referred to as the “doer” thread, creates a service profile association workflow in the form of a finite state machine (FSM). The doer thread modifies the managed objects in the management information tree according to the settings in the service profile. It then calls the appropriate application gateways to actually deploy the configuration changes to the managed endpoints. The application gateways transition through platform-specific workflows to help ensure that all endpoints are modified appropriately.

The doer thread does not wait for the application gateway to complete its task. Instead, the doer thread processes the next task in its queue. When the application gateway completes its task, it signals the doer thread, at which point the doer thread transitions to the next task in the FSM. The DME orchestrates all management stimuli serially and transactionally to help guarantee that all requests to change the management information tree are

successfully performed. Cisco UCS Manager implements a policy-based management using service profiles to help automate provisioning and increase agility.

Standard Interfaces and Clients

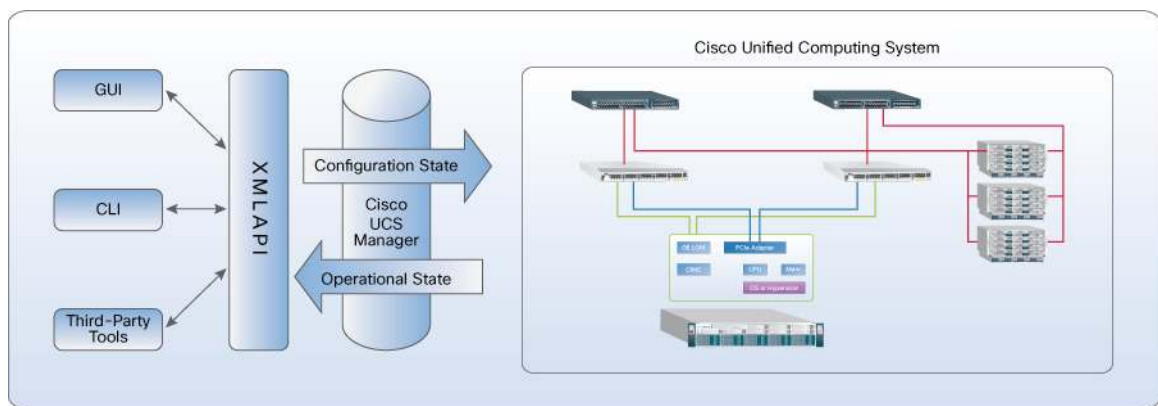
Cisco UCS Manager abstracts the back-end platform implementation from the business logic of the information model and is designed with independent front-end user interfaces. Configuration state changes that originated in the user interface are then applied to the managed object in the information model, and finally they are deployed to the managed endpoint. Conversely, operational state changes originating in the managed endpoint are applied to the managed object in the information model, and they then finally appear in the user interface.

The two primary interfaces for user interaction with Cisco UCS Manager are the buffered CLI and the GUI (Figure 6). Both the CLI and GUI are written using the XML API. The GUI provides a very simple and intuitive interface for users to manage Cisco UCS. The Java Web Start executable code is pushed to the administrative client at the time of connection to the management port, helping ensure that the most current version of the GUI is deployed. One of the benefits of using the GUI is that it provides several wizards to guide users through complex configuration tasks.

Client interfaces, such as the CLI and GUI, generate stimuli in Cisco UCS that can change the configuration state of a managed object. A less obvious example of a stimulus generator is another managed object. For example, if a fabric extender fails, then the operational state of the network interface ports connected to that fabric extender will also be affected.

The CLI provides an interface for users who prefer working at the command line. One of the benefits of using the buffered CLI is that it provides the capability to group multiple configuration changes in a single transaction. The CLI also allows the user to perform multiple commands and to store and run scripts. For example, if users perform a set of management operations on a regular basis, the associated commands can be stored in a script. Users can, whenever they want, run the script with a single command. In this way, users can effectively create their own custom configuration wizards by writing scripts. The CLI also comes equipped with standard enterprise features such as a command history and syntactically guided command completion.

Figure 6. Separation of Business Logic and Hardware Implementation



In addition to the native open XML API, Cisco UCS Manager complies with Intelligent Platform Management Interface (IPMI); Serial-over-LAN (SoL); keyboard, video, and mouse (KVM) connection; Simple Network

Management Protocol (SNMP); syslog; Systems Management Architecture for Server Hardware Command-Line Protocol (SMASH-CLP); and Common Information Model XML (CIM-XML) requirements.

Evolution of Cisco UCS Manager

Cisco UCS was introduced in July 2009, and as of December 2011 it has been used by more than 9000 customers globally. The first version of Cisco UCS Manager introduced the concepts of service profiles, service profile templates, policies, pools, and access through the XML API for agile management of Cisco UCS components. Since then, new features and capabilities have been added, with regular input from customers. Updates have been made for scalability, to accommodate hardware upgrades, and to improve usability. The December 2010 release brought significant enhancements in most domains, including monitoring, security, and network and rack-mount server management. The September 2011 release of Cisco UCS Manager 2.0 brought support for next-generation fabric hardware and various software innovations. In the future, Cisco will continue to introduce innovations throughout the Cisco UCS infrastructure.

Conclusion

Cisco UCS Manager is an embedded unified manager for Cisco UCS. It enables a policy-based management approach through the use of service profiles and just-in-time provisioning of physical resources. The model-based framework of Cisco UCS Manager separates the operational logic from platform implementation, making the software easier to maintain. The Cisco UCS Manager information model is hierarchical, which makes modifying the management information tree more intuitive for XML API users. Managed objects can be modified either through configuration state changes initiated by an administrator or through operational state changes that occur in the managed endpoint, both of which occur as ACID-compliant transactions in the DME. Users can use various industry-standard interfaces to manage Cisco UCS or use the GUI or CLI to interact with Cisco UCS Manager. The XML API makes the infrastructure programmable and can be used to write custom management interfaces as well as perform integration with existing environments. Cisco UCS provides high availability to Cisco UCS Manager by running two fabric interconnects in the environment.

Cisco Unified Computing Services

Using a unified view of data center resources, Cisco and our industry-leading partners deliver services that accelerate your transition to a unified computing architecture. Cisco Unified Computing Services help you quickly deploy your data center resources, simplify ongoing operations, and optimize your infrastructure to better meet your business needs. For more information about these and other Cisco Data Center Services, visit <http://www.cisco.com/go/unifiedcomputingservices>.

Why Cisco?

Cisco UCS continues Cisco's long history of innovation in delivering integrated systems for improved business results based on industry standards and using the network as the platform. Cisco UCS is a critical component of the Cisco Unified Data Center, which provides a complete data center infrastructure architecture. The Cisco Unified Data Center combines computing, storage, networking, security, and management resources into a fabric architecture that delivers outstanding performance for physical and virtualized business applications. Created to help companies evolve to cloud computing environments, the Cisco Unified Data Center embeds automation and simplified operations at the server, network, and cloud-services layers.

For More Information

Contact your local Cisco representative or visit:

-
- **Cisco Unified Computing System**
<http://www.cisco.com/go/unifiedcomputing>
 - **Cisco Unified Computing System: A Complete Reference Guide to the Cisco Data Center Virtualization Server Architecture**
<http://www.amazon.com/Cisco-Unified-Computing-System-Center/dp/1587141930>
 - **Cisco Developer Network**
<http://developer.cisco.com/web/unifiedcomputing/home>
 - **Cisco UCS Manager product page on Cisco.com**
<http://www.cisco.com/en/US/products/ps10281/index.html>
 - **Cisco UCS Platform Emulator (UCSPE) download**
<http://developer.cisco.com/web/unifiedcomputing/ucsemulatordownload>
 - **Cisco UCS Manager Advantage Video Series**
http://www.cisco.com/en/US/prod/ps10265/ucs_advantage_video_library.html
 - **Cisco IT Solutions**
http://www.cisco.com/web/about/ciscoatwork/data_center/index.html



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)