Cloud Provisioning APIs with Self-Care Portals for Orchestration Software

Hosting Controller Whitepaper
www.hostingcontroller.com

Latest version of this white paper can always be downloaded from:
http://hostingcontroller.com/Resources/Whitepapers/Provisioning-APIs-with-Self-care-
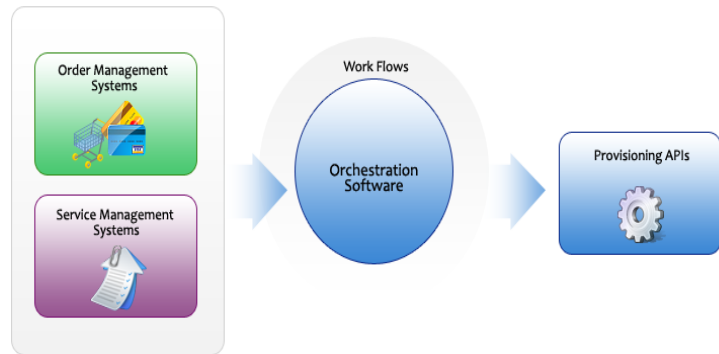**Portals-for-Orchestration-Software.pdf**

Table of Contents

Overview

Orchestration software is at the heart of any Service Provider's fulfillment processes. Its purpose is to execute the work flows required to fulfill provisioning requests for different resources. Work Flow orchestration is typically used behind:

1. **Order Management Systems:** To execute the work flows to provision resources for new orders or to reclaim resources from deleted orders.

2. **Service Management Systems:** To execute the work flows to manage ongoing services with use cases like Suspend / Degrade / Resume Service and so on.

The complexity for Orchestration software is growing in multiple dimensions, such as:

Bundled Product Offerings

Product catalogues of Service Providers now include multiple bundled individual services together as a single Product Offering.
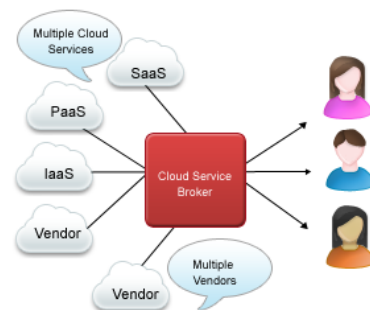
More sophisticated Service Realization models

Each individual service now has multiple network elements performing different functions of Policy Enforcement, Border Control, Service delivery, Service Management and others.

Distributed Deployment models

Services are orchestrated by brokering between multiple distributed (potentially cloud based) service delivery models.

Granular Monetization

Services could be monetized down to a single service feature and hence require interface with subscription servers carrying details of add-on subscriptions and features available in them.

For example, it is not uncommon for Internet Service Providers now to be offering E-Mail hosting, website hosting or some kind of VoIP in addition to basic Internet service and for each of these basic services to be realized through multiple network elements. More aggressive Service Providers are offering different Cloud Computing services in addition to basic Internet Access or telecom service that they provide.

This is the fusion of conventional Service Providers with the world of 'Cloud Computing'.

Service Providers & Cloud Computing

NIST (National Institute of Standards and Technology) has recently released their definition of "Cloud Computing" as having five essential characteristics:

1. Resource Pooling

2. On Demand Self-Service

3. Rapid Elasticity

4. Measured Service

5. Broad Network Access

Full report can be found at http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Let's discuss the role of correct provisioning for all the above characteristics one by one.

1. Resource Pooling

Resource Pooling is a core characteristic of any Cloud Computing deployment. This means that Resources of all types including Compute, Storage, Networking and Software are pooled among multiple users in a way that allows them simultaneous use but at the same time keeps a layer of separation among them. Better described as a Shared Multi-Tenant structure, as we will see, it is achieved differently for different services.

Shared Multi-Tenancy choices: Virtualization or Access Control
Virtualization has been the driving technology behind the rapid proliferation of Cloud based service delivery models. Virtualization makes creation of shared multi-tenant partitions possible and easier. But Virtualization is only viable in a narrow band in terms of size of

the application provided. It is not viable or even possible to use Virtualization out of that band both ways. This happens when:

1. The Application is too **small** for virtualization

2. The Application is too **large** for virtualization

Example of too small Application would be websites on a shared hosting server. It may not be viable to provision each website on its separate virtualized container. Same goes for bulk of web applications hosted in the cloud. They are usually too small to justify individual partition.

Example of too large Application would be Microsoft Exchange and its related suite of Enterprise Application Servers. Minimum deployment requires multiple heavy roles provisioned to work together as a single big whole. Creating a new set of minimal roles for each new tenant is just isn't practical.

Whenever Virtualization based Multi-Tenancy is not viable, 'Access Control' based multi-tenancy is applied. That requires understanding of the actual software and its realization model and provisioning resources accordingly.

Access Control Based Multi-Tenancy Requirements
Access Control List (ACL) based provisioning is not about simple creation of new resources. It requires detailed 'Application Specific' understanding of how access control is supported and how it is to be maintained during provisioning. There is no single formula that would apply to all applications.
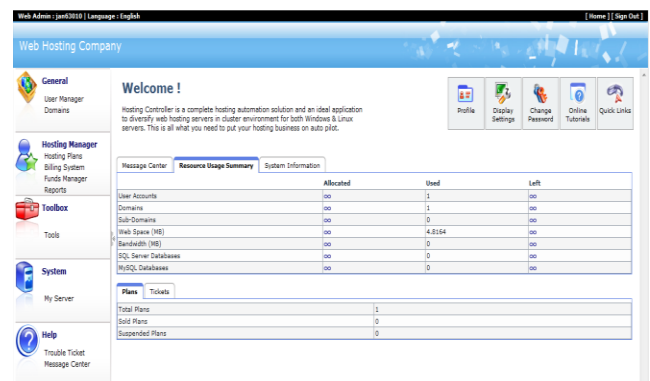
It may provide multi-tenancy by any of these or a combination thereof:

- Meta Data in the user database or active directory
- Folder isolation
- Database isolation
- Configuration file settings
- Web Application isolation

2. On Demand Self-Serve

On-Demand Self-Serve is the second basic characteristic of any Cloud Computing offering. Most applications come with an out of the box self-serve portal allowing users to do basic provisioning tasks on their own. There are reasons why out of the box self-serve portals cannot work in a Cloud enabled environment:

1. When multi-tenancy is imposed through Access Control Lists, the self-serve portals may not function, as they are not designed to keep different tenants separate from each other.

2. When quotas are to be imposed as part of subscriptions, there may be no way for out of the box self-serve portals to enforce quotas.
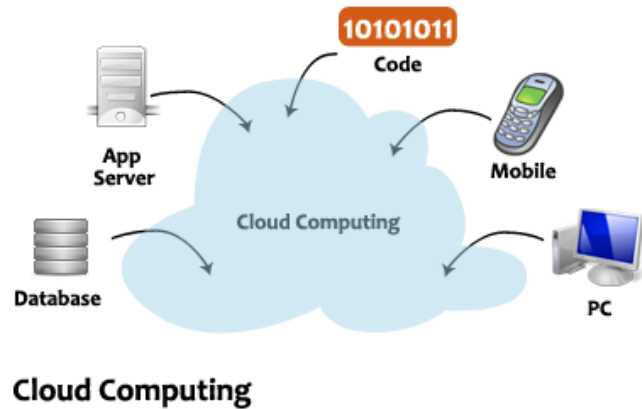
3. A transactional behavior may be needed behind each use case so as to keep databases updated, to do some consistency checks, support licensing compliance and others.

This requires custom built self-serve portals for different layers of users for each different software application supported in the cloud.

3. Rapid Elasticity

Rapid Elasticity is about the ability of the infrastructure and the platform to scale up and down on demand. This could be initiated by the customer service representatives or the customer itself through its self-serve portal. The elasticity use cases again need more than provisioning. It will need:

- A Resource Distribution Manager that keeps track of resource availability across infrastructure and its consumption thereof.
- A tight integration with the Product Offering to allow expansion as per business approved products that can then be tracked and charged accordingly.
- Feature Monetization support to allow the SP to allow granular expansion of the service and be able to charge for different features as per will.
- A dependency resolution rules engine that will make sure that any dependent resources are available when provisioning new resources.

While virtualization software have done a good job in allowing rapid elasticity for the infrastructure level provisioning, they cannot allow such elasticity in platform or software specific provisioning.

4. Measured Service

The charges for the Service could be on a purely subscription basis, purely usage basis or a combination thereof. All charging scenarios require a visibility of provisioned resources for the service and measurement of the actual usage. 'Usage' is again a very service specific thing and its meanings for 'Measurement' can only be defined in the context of one specific platform or software. Mostly usage is measured in multiple dimensions and has a link back to the billing and charging systems. 'Measurement Agents' are required in such cases that know the details of the specific platform or software and can send numbers back to the charging or billing engines in some common denominator terms in which the product offerings have been designed.

5. Broad Network Access

This is usually on the access side of the service provider and is not a core functionality to be handled in the cloud. That's why it remains out of scope for this white paper.

Orchestration Software Architecture:

While Orchestration software provides the work flow execution engine, it always needs a provisioning layer to perform the actual 'work' behind these work flows. Some Work APIs are provided as part of the major Orchestration Engines, for reasons mentioned above, they may not work for Service Provider scenarios. Some of the reasons where specialized APIs would be needed may include:
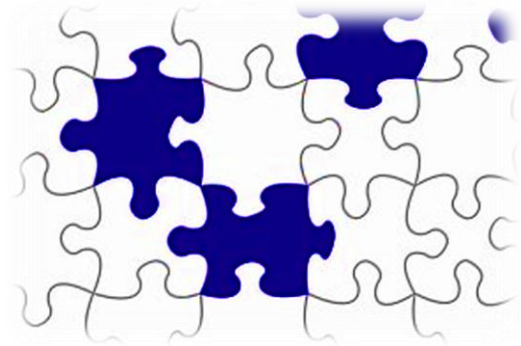
1. Support for third party software and systems not supported by Orchestration engines

2. Account Level Awareness

3. Missing use cases for Service Management

4. Detailed knowledge of Access Control Lists for Multi-Tenancy

5. Multi-Tenant aware self-serve portals

6. Policy and Quota enforcement through self-serve portals

7. Feature Level Productization

8. Usage Monitoring Agents

    a. Storage

    b. Bandwidth

9. Extensibility of the Platform

10. Topology hiding of actual infrastructure from Orchestration

11. Layers of required provisioning through infrastructure, platform and software

12. Keeping an up to date list of provisioned resources

13. Transactional APIs vs Raw work APIs

14. Dependency Resolution and Domain Knowledge

15. Availability of Resources

16. Licensing

17. Provisioned vs Allocated vs Consumed Resources

18. Reseller Channels B2B / B2C relations

19. Configuration Management Database

20. Operational Reporting and integration

21. Load Distribution through master console

22. Support for multiple clusters of servers

23. Allowing PaaS extension

24. IaaS use cases

25. PaaS use cases

26. SaaS use cases

27. Both Linux and Windows support

28. Requirement for ongoing patches and upgrades

29. Meeting business use cases of customers

30. Exposing raw APIs

Hosting Controller as the Provisioning Layer

Hosting Controller is an automation control panel that evolved for the Hosting industry in general. While it remains a standalone product, it provides APIs that can or should be used as the Provisioning or actual work execution layer behind any other work flow execution or orchestration engine.

The value additions offered by Hosting Controller as the Provisioning Layer are numerous and covers almost all the aspects mentioned above. Some of them are briefly discussed below:

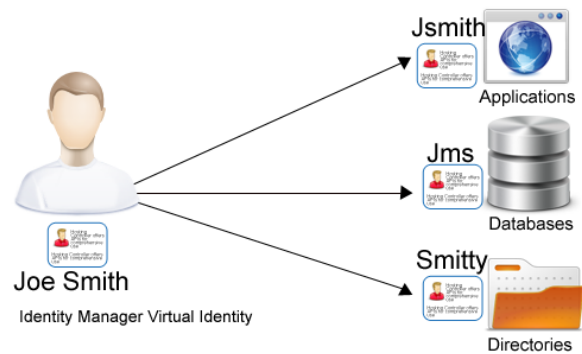## *Support for Large Number of Supported Softwares*

First and foremost is Hosting Controller's out of the box support for a very large range of existing and common Software, Platform and Infrastructure providing software servers. The list of supported software is long and can be seen on HC Website at:

- ●For Windows supported Software click here
- ●Linux Supported Software click here
- ●Microsoft Exchange and related Enterprise products click here
- ●Virtualization Products for both Windows and Linux click here

Account Level Awareness

Hosting Controller APIs are aware of the User Account and keep track of all resources provisioned under that User Account. This is done in a database layer that is kept up to date with all provisioning.
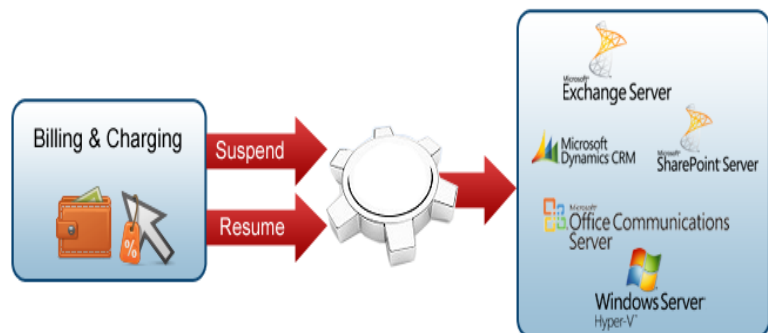

Use Cases for Service Management

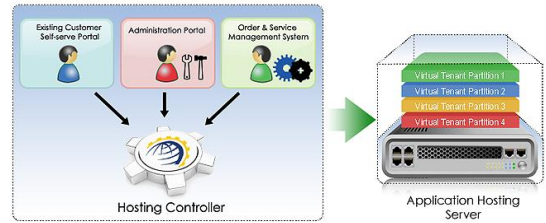Hosting Controller offers APIs for comprehensive use cases of Service Management. They include:

- ● Suspend Services
- ● Resume Services
- ● Degrade Service (for selected products)

Services can be suspended and resumed at a Resource level or an Account level.

Application Dependent Access Control

Hosting Controller APIs are aware of the way multi-tenancy is imposed on the otherwise monolithic software applications. This varies from application to application and HC offers native support using best practices for access control based on each products configuration options.

Multi-Tenancy Aware Self-Serve Portals

Hosting Controller offers comprehensive self-serve portals for all the applications that it supports. The self-serve is made considering the design of target application and allows all provisioning done to be compliant with multi-tenancy requirements.

Policy and Quota Enforcement through Self-Serve Portals

In addition to implementing access control, the self-serve portals also need to consult Policy and Quota manager servers to 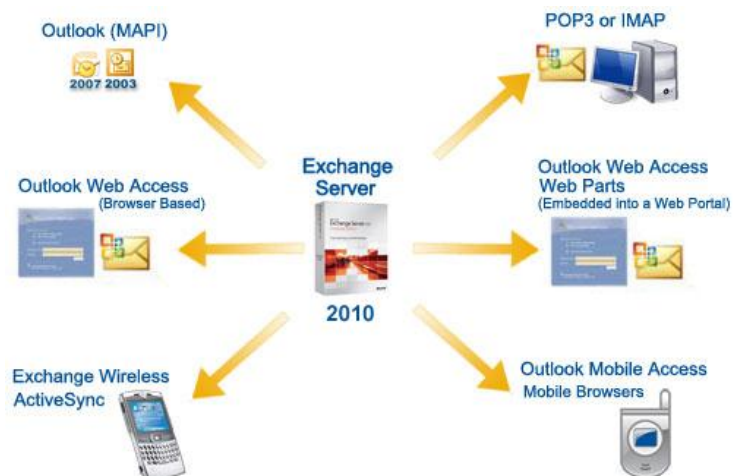make sure any new provisioning request is within allowed quotas. For commercial software, it may also have an implication on licensing purchased. Hosting Controller comes with a Plan and Quota manager that can be provisioned with the allowed blanket values of all Resources and Features within those resources. When users run out of quotas and attempt provisioning beyond their limits, Hosting Controller offers useful use cases like 'Restrict and Alert' or 'Allow with Invoice' to be integrated with billing engines.

Feature Level Productization

The whole catalogue and product management revolves around the ability to provision the services and resources down to 'Features' available in the Application. If the provisioning only allows an 'all or none' type provisioning, then the product offerings based on it can also not go beyond basic flat charges. Hosting Controller allows provisioning and hence monetization down to individual features in most supported application. Once Hosting Controller identifies a feature, it not only allows productization based on those features but also takes care of all further provisioning through self-serve based on those features. Example for Microsoft Exchange monetization features include:

- IMAP
- POP
- MAPI
- OWA /HTTP
- Active Sync
- Distribution List
- Public Folders
- Mailboxes
- Domains

Usage Monitoring Agents

Provisioning can only enforce policies or resource restrictions that can be applied at a point of time. There are other monetization features which are 'counted' over a period of time and Cloud based monetization models excessively use such features. Examples include storage space and bandwidth used.

For such features, 'usage monitoring agents' are needed to keep counting the usage as it happens and report to any billing manager or quota manager about excesses. The problem is compounded when counter quotas are segregated among many applications.

Hosting Controller comes with Usage Monitoring Agent for different resources. With its awareness of a 'user' and all the resources that belong to a user, it has the capability to count and add usage from different places and report them together.

Disk Space Utilization

1. Mailbox Size

2. Database Size

3. SQL Database Size

4. SharePoint Database Size

5. FTP Folders

6. Virtual Hard Disc

Bandwidth Utilization

1. VM Bandwidth

2. Web Server Bandwidth

    a. IIS

    b. Apache

3. FTP

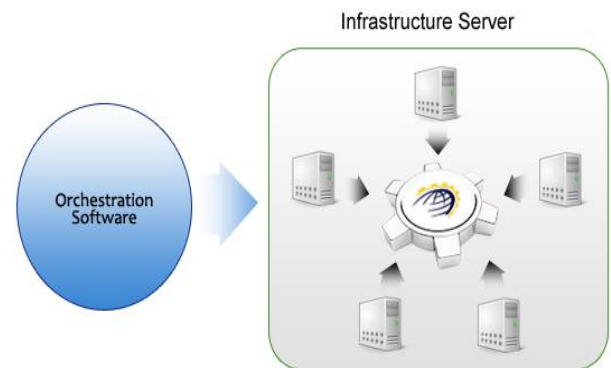## *Extensibility of the Platform*

Hosting Controller offers the Service Provider to extend the systems to make it into Platforms that can be offered as a Service. Hosting Controller offers Click & Install APIs that are extensible through simple XML based scripting and allows you to use the offered platform for custom web applications.

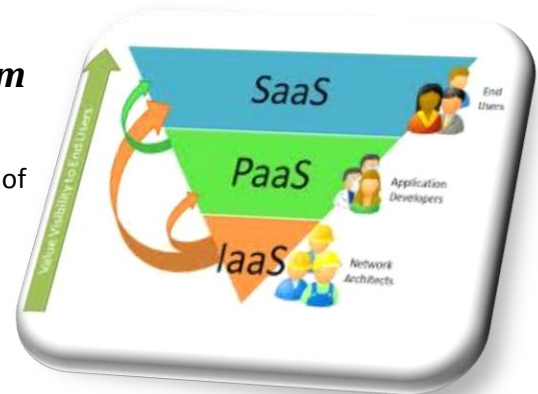For further information regarding HC's Click & Install API, click here

Topology Hiding of Actual Infrastructure from Orchestration

Hosting Controller provides an infrastructure database. Another web based application allows administrators to mention all their deployment topology here and assign roles to different servers. Infrastructure Selection Policy chooses the infrastructure where resources are actually provisioned based on policies dictated by the Administrator. This totally hides the deployment topology from Orchestration or other order management systems taking substantial complexity out of them. They can simple call HC's exposed APIs and resources are provisioned on the back end.

Provisioning through Infrastructure, Platform and Software

Hosting Controller exposes a simple API that does provisioning of layers of resources as and when needed. This may include infrastructure provisioning, platform provisioning or simple software provisioning.

List of Provisioned Resources

Hosting Controller is user aware and maintains a list of all provisioned resources on all the distributed infrastructure into a single database. This single database can be queried to get a full 360° view of the provisioned resources across distributed infrastructure and their relation to 'users'.

Transactional APIs

All Hosting Controller APIs are transactional i.e. they are not raw work APIs. Transactional APIs do the following:
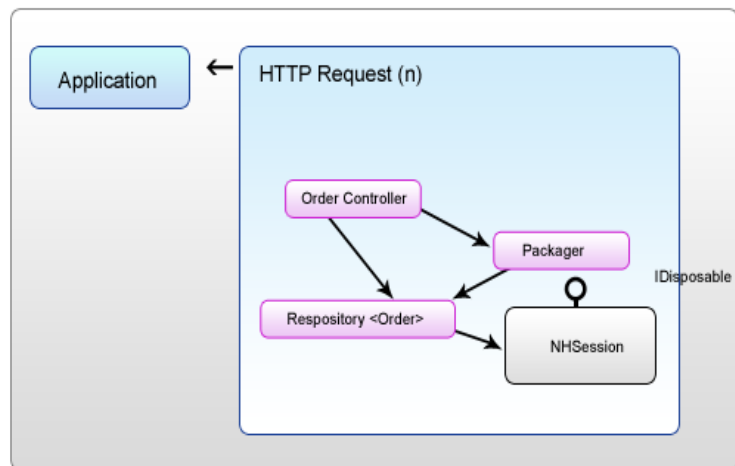
1. Check quotas before execution

2. Basic input sanitation and conflict resolution

3. Synchronizing the database with actual provisioning done

4. Execute the intended provisioning work

This makes sure that some raw work is not done on the network without a trace in the database or without going through quota checks. Quota checks can be turned off if required.

Dependency Resolution and Domain Knowledge

Hosting Controller provisioning evaluates all dependency rules which are specific to the application in question while executing all provisioning orders. e.g. when provisioning a mailbox for exchange Hosting Controller APIs make sure that a 'Mail Domain' is available within the quota of the user. This dependency resolution is domain specific and HC is aware of all such rules for all the software that it supports.

Licensing

Hosting Controller generates detailed reports of licensing to be used by Service Provider for compliance reasons. Some reports are generated through the database and others are generated through actual querying of configuration files of software servers. For some specific licensing reporting requirements like Microsoft SPLA program, Hosting Controller offers out of the box reporting managers that generate licensing data in excel sheets and XML formats to be used for such reporting.

Provisioned Vs Allocated Vs Consumed Resources

Hosting Controller keeps a current snapshot of all consumed resources versus all available resources in the infrastructure. This allows clear visibility for operational needs on the current state of all resources in the infrastructure.
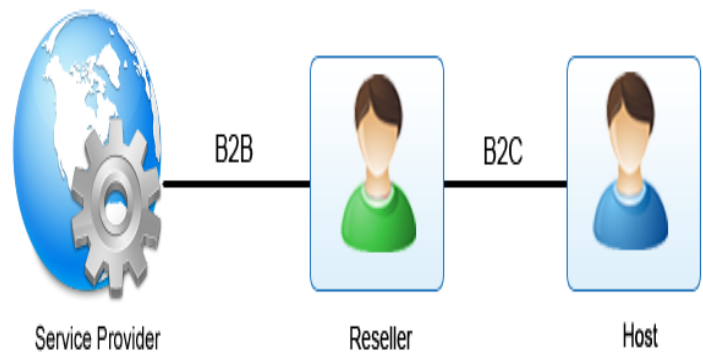
Hosting Controller's transactional APIs make sure that the database keeps the most current image of all provisioned resources. It not only keeps count of Provisioned Resources, it also keeps count of how many resources have actually been allocated in terms of quota and how many are actually consumed.

Reseller Channels B2B / B2C relations

Hosting Controller offers full support for 'Resellers'. These are entities that keep a wholesale or B2B relationship with the Service Provider and in turn keep a B2C relationship with their customers. Hosting Controller APIs are fully aware of this extra B2B layer and all APIs and thus do the followings:

- Keep ownership information of 'Reseller' along with the end user if required.
- Enforce B2B quotas (between Service Provider and Reseller) in addition to normal B2C quotas (toward end user).

Configuration Management Database

HC keeps full details of deployed infrastructure and its capabilities in terms of roles that it can assume. This optimizes licensing costs and otherwise keeps track of what 'roles' can be performed by available infrastructure and the provisioning more easily defined in terms of these roles.

Operational Reporting and Integration

HC allows the Service Provider to query its database directly and get the latest information about a reseller or hosts allocated, provisioned and used resources through a single query. The database can be easily integrated into SPs external CRM or other OSS systems for operations queries.

Load Distribution through Master Console

Hosting Controller's database contains Resource Provisioning Policy that is queried at the time of actual provisioning. This allows distributing provisioning requests among available servers through a centralized policy and load can be balanced across available infrastructure.
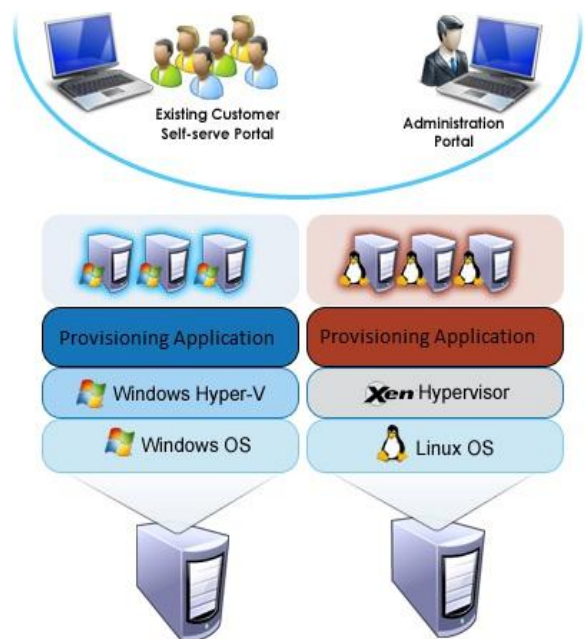
Load balancing is based on multiple roles that are assigned to infrastructure servers and realize the service

Support for Windows and Linux

HC exposes a single set of APIs to provision both Windows and Linux based servers. It completely hides the actual details of provisioned servers from the calling application. It has connectors for both types of servers and allows real-time connections towards all those servers for provisioning requests. It uses a secure tunnel channel between the master console serving APIs and the actual infrastructure servers.

Different flavors of Linux and all new versions of Windows server software are supported as well as multiple platforms and software on both popular operating systems.

Provisioning of Infrastructure

HC exposes APIs to provision virtualized Infrastructure both on Windows and Linux. It not only allows basic creation of virtualized partitions, it also allows further more granular control over it and also exposes Service Management use cases (Suspend, Resume). For further information, click here

PaaS Provisioning

HC offers provisioning of Platforms where users can extend the platforms for their needs. HC's Click & Install APIs allow exposing a full platform for Web Applications where users can install personal applications of their choice and have them automatically configured and provisioned.

A basic XML based work-flow language is included that is capable of provisioning most common ways new web applications are provisioned. This turns these APIs into a powerful web application platform that can be served as a PaaS offering. For further information, click here

SaaS Provisioning

HC offers provisioning of multiple software to be offered as a service. To view list of Windows Supported Software click here. For Linux Supported Software, click here.

In addition to this, HC provides extensive support for Microsoft Exchange suite of products. To view the list click here.

Requirement for Ongoing Patches and Upgrades

Hosting Controller continuously upgrades the realization of its APIs to support newer versions and service packs of products from vendors. This requires ongoing changes to actual realization but HC keeps the signatures of its APIs fixed (as much as possible) so that an integration done once over an HC API, it expected to work in future when newer versions are deployed without any rework on part of Orchestration providers. This layer of isolation saves tremendous ongoing effort on part of Service Provider who can depend on HC to continue to upgrade its APIs for all supported products.

Meeting Business Use Cases of Customers

Business use cases for Service Providers are getting complex and supporting those complex business models require specially created support on a product by product basis. HC products are used by more than 3,000 hosting companies worldwide and the exposed APIs make sure that most if not all business models over those applications are handled by HC.

Exposing raw APIs

Although HC only exposes Transactional APIs that execute fixed work flows behind them, HC can expose all raw Work APIs on demand by any Service Provider if need.

Summary

Hosting Controller is one out of the box product that can take care of most, if not all of the provisioning requirements of Service Providers using any kind of Orchestration for their service fulfillment. With its broad range of support for tens of popular software, its complete provisioning run-time with its local database, its support for Windows and Linux based operating systems and others, Hosting Controller is safely provides the 'Work' behind the work flows

About Hosting Controller:

Founded in 1999, Hosting Controller is a complete hosting automation solution for web hosts and cloud based service providers. It allows them to manage both Windows & Linux servers simultaneously as part of a single cluster. In addition to shared hosts, HC offers a full solution suite for hosted Enterprise Applications creating a shared multi-tenant environment for automatic provisioning of Exchange 2007/2010, BlackBerry, SharePoint, OCS and Dynamics CRM. It also offers a full automation solution for Infrastructure and Virtualization providers offering virtualized partitions on both Windows and Linux based servers through Hyper-V technology for Windows & Xen Hypervisor technology for Linux!

Hosting Controller Inc. is based in Ontario, Canada and has more than 2,000 customers in 125 countries worldwide.

See more details on http://HostingController.com