Comparing Multicloud Management and Governance Approaches

Published: 6 March 2019    ID: G00375973

Analyst(s): Alan Waite

Organizations adopting multicloud architectures must decide how to transform in the areas of IT service brokering, governance, architecture and operations. This research helps I&O-focused technical professionals analyze the alternatives in each area and highlights the strengths of each approach.

Key Findings

- As organizations adopt public cloud services from multiple cloud providers, their IT processes must transform in four areas: service brokering, governance, architecture and operations.
- Not all organizations will adopt the same approaches in these areas. For example, one organization may reject a centralized cloud brokering approach while another embraces it.
- Multicloud management using comprehensive rules and restrictions is giving way to policy-based governance because public cloud adoption has become too complex for a single rule that applies to every eventuality.
- Choosing an approach for each area of multicloud management involves trade-offs between risk, agility, freedom-of-choice, complexity, cost, functionality and scalability requirements.

Recommendations

To successfully transform IT operations for multicloud environments, technical professionals should:

- Implement a coordinated, organizationwide approach to multicloud management and governance. Standardize cloud policies and clarify processes and ownership. The initial problems are political rather than technical.
- Decide on approaches in each area of service brokering, governance, architecture and operations — based on the organization’s priorities for risk, agility, freedom of choice, complexity, cost, functionality and scalability.
- Choose an enforcement style for each governance use case based on both preventive and retrospective controls. Separate governance policy definitions from enforcement style and provider-specific implementation.
Balance the competing requirements of efficient consistency across different cloud platforms with access to unique native functionality by rationalizing management tools and considering a central platform ops team.

Table of Contents

Comparison........................................................................................................................................... 3
  Comparing Approaches to Managing Multicloud Environments.......................................................... 3
  Multicloud Management Approaches and Alternatives........................................................................... 4
Analysis................................................................................................................................................11
  Central Brokering vs. Independent Teams.............................................................................................. 12
  “On the Side” Self-Service vs. “In the Way” Self-Service...................................................................... 15
  Programmatic Control Governance vs. Descriptive Guideline Governance......................................... 17
  Preventative Programmatic Control vs. Retrospective Programmatic Control........................................ 19
  Standardized Deployment Architecture vs. Custom Deployments......................................................... 21
  Use-Case-Based Silos vs. Composite Multicloud.................................................................................. 23
  Central Visibility and Control vs. Distributed Management Centers.................................................... 26
  Cross-Platform Tools vs. Native Tools................................................................................................... 29
Guidance..............................................................................................................................................31
  Brokering and Self-Service.................................................................................................................... 31
  Governance......................................................................................................................................... 32
  Architecture....................................................................................................................................... 32
  Operations......................................................................................................................................... 33
Gartner Recommended Reading............................................................................................................... 34

List of Figures

Figure 1. Service Brokering: Central Brokering vs. Independent Teams................................................... 4
Figure 2. Service Brokering: “On the Side” Self-Service vs. “In the Way” Self-Service.............................. 5
Figure 3. Governance: Programmatic Controls vs. Descriptive Policy....................................................... 6
Figure 4. Governance: Preventative Programmatic Control vs. Retrospective Programmatic Control........ 7
Figure 5. Architecture: Standardized Deployment vs. Custom Project Deployments................................ 8
Figure 6. Architecture: Use-Case-Based Silos vs. Composite Multicloud................................................ 9
Figure 7. Operations: Centralized Visibility and Control vs. Distributed Management............................ 10
Figure 8. Cross-Platform Tools vs. Native Tools..................................................................................... 11
Figure 9. Centralized Brokering vs. Independent Teams.......................................................................... 13
Comparison

Comparing Approaches to Managing Multicloud Environments

The eight different approaches analyzed in this research align to four areas of transformation: service brokering, governance, architecture and operations. The eight approaches are not necessarily mutually exclusive. In many instances, an organization will choose to adopt one approach for some use cases and an alternative approach for others.

For example, in the area of programmatic control in governance, an organization may choose preventative programmatic control for some use cases and retrospective control for others. This research will guide you to the correct approach for each use case by considering the strengths of each alternative.

The eight approaches considered in this research are:

- Service Brokering: Central brokering vs. independent teams
- Service Brokering: “On the side” self-service vs. “In the way” self-service
- Governance: Programmatic control vs. descriptive policy
- Governance: Preventative programmatic control vs. retrospective programmatic control
- Architecture: Standardized deployment vs. custom project deployments
- Architecture: Use-case-based provider silos vs. composite multicloud
- Operations: Centralized visibility and control vs. distributed management centers
- Operations: Cross-platform management tools vs. native platform-specific tools

A detailed description of each approach and the reasons to choose it are included in the Analysis section.
Multicloud Management Approaches and Alternatives

Figures 1 through 8 summarize the alternative approaches and the strengths of each alternative that drive a selection for an individual use case. More detailed recommendations for each approach are provided in the Guidance section.

Figure 1. Service Brokering: Central Brokering vs. Independent Teams

Central Brokering vs. Independent Teams

- Centrally governed application and service placement
- Consistent user experience for service requests
- Consolidated provider contracting, procurement and management
- Enables automated provisioning across multiple environments
- Easier to implement prescriptive governance

- Larger range of choices for developers and user groups
- Avoids process bottlenecks
- Avoids political issues (Who decides what runs where)
- No need for external broker agency or complex transformation to brokering tools, people and procedures

Source: Gartner (March 2019)
Figure 2. Service Brokering: “On the Side” Self-Service vs. “In the Way” Self-Service

“On the Side” Self-Service

- Greater agility
- Faster provisioning with no central process bottleneck
- Larger range of choices and functionality for developers and user groups
- No need for complex implementation and integration of self-service portal and service catalog to different cloud providers

“In the Way” Self-Service

- Guaranteed compliance and control
- Consistent user experience for service requests
- Lower skills requirement for end users
- Less repetition of effort by user groups
- Can include on-premises and noncloud private resources in same interface

Source: Gartner (March 2019)
Figure 3. Governance: Programmatic Controls vs. Descriptive Policy

Programmatic Controls vs. Descriptive Policy

- Programmatic Controls
  - Enforces governance requirements and compliance
  - Appropriate for critical or high-risk use cases
  - Lower risk of security breaches, loss of control and overspending on cloud resources

- Descriptive Policy
  - Good for defining and communicating high-level governance goals
  - Can be applied to anything without concern for practicality (actual code, tools, etc.)
  - Less work to implement
  - Better agility and choice for users
  - Does not require skills, tools and APIs to implement programmatic controls on each provider

Source: Gartner (March 2019)
Figure 4. Governance: Preventative Programmatic Control vs. Retrospective Programmatic Control

Preventative Controls vs. Retrospective Controls

- "Instant" compliance and control
- Prevention is usually easier than remediation
- Appropriate for use cases where no risk is tolerated
- Avoids complex implementation of tools, APIs and processes for detection and remediation

- Better agility and creative options for users
- Broader choice of services for users
- Allows for hard and soft remediation processes
- Enables policies that cannot be enforced at deployment time

Source: Gartner (March 2019)
Figure 5. Architecture: Standardized Deployment vs. Custom Project Deployments

**Standardized Deployments vs. Custom Deployments**

**Standardized Deployments**
- Improves economies of scale in a DevOps environment
- Provides abstraction across public cloud, private cloud or virtualized on-premises infrastructure
- Consistent, reusable, tested deployment models
- Compliance to regulations
- Avoids fragmentation of skill sets and tools across providers

**Custom Deployments**
- Broad choice and flexibility of cloud services for user teams
- Quicker to get up and running
- Does not require a skilled central team with PaaS and CI/CD tools
- Avoids lock-in to a single PaaS or orchestrator environment

Source: Gartner (March 2019)
Figure 6. Architecture: Use-Case-Based Silos vs. Composite Multicloud

Use-Case-Based Silos vs. Composite Multicloud

- Avoids the complexities of multicloud integration
- Broad choice and flexibility of cloud services
- Accommodates mergers and acquisitions
- Quicker to get up and running
- Starting point for most organizations

- Workload portability/exit strategy
- Exploits unique high-level services for components of applications
- Allows cost optimization
- Enables additional high-availability scenarios

ID: 375673

Source: Gartner (March 2019)
Figure 7. Operations: Centralized Visibility and Control vs. Distributed Management

Centralized Visibility and Control vs. Distributed Management

- Consistent visibility and control across cloud environments
- Those who understand operations are doing operations
- Lower skills requirement for LOBs/developers

- Unlimited choice of providers and services
- More agile teams
- Quicker implementation
- Fragmentation of skill sets and tools across silos
- Does not require skilled central team and new tools

Source: Gartner (March 2019)
Analysis

By 2022, about 75% of enterprise customers using cloud infrastructure as a service (IaaS) will adopt a deliberate multicloud strategy, up from 49% in 2017.1 As organizations begin adoption of cloud computing, they typically focus on one provider and build skills around the native services of that vendor. However, many Gartner clients are pursuing a multicloud strategy because they want to use best-in-class services from multiple providers, distribute risk across vendors, and/or accommodate mergers and acquisitions. Detailed information on multicloud architectures, benefits, risks and challenges can be found in “Technology Insight for Multicloud Computing.”

Those organizations that successfully adapt to a multicloud strategy transform IT in four areas: service brokering, governance, architecture and operations. Within these four areas, Gartner sees eight approaches that are commonly employed for different use cases:

- Service Brokering:
  - Central brokering vs. independent teams
  - “On the side” self-service vs. “In the way” self-service
- Governance:
Programmatic control vs. descriptive policy guidelines

Preventative programmatic control vs. retrospective programmatic control

Architecture:

- Standardized deployment vs. custom project deployments
- Use-case-based silos vs. composite multicloud

Operations:

- Centralized visibility and control vs. distributed management centers
- Cross-platform tools vs. provider native tools

Choosing an approach for each of your use cases involves trade-offs between risk, agility, freedom-of-choice, complexity, cost, functionality and scalability requirements, and these were summarized in the preceding section. The eight approaches are described in more detail in this section.

Central Brokering vs. Independent Teams

The aim of a brokering function is to assist with selection of the right services for end users, as well as to enable and enhance accessibility and consistency of different cloud services. This should enable developers, consumers and lines of business to quickly access technology services while safeguarding the interests of the business through the application of centralized policies and procedures. A central brokering function can provide a consulting-style service to users to assist in selection of providers and services. Central brokerage also involves aggregation, integration and customization of the underlying technologies to add value to those underlying cloud services. Guidance on implementation of a central brokering function can be found in “Adapting IT to Become the Broker of Cloud Services.”

The different approaches involve either using a central broker function for all cloud services or allowing independent teams to choose their own workload placement, providers and services. For some use cases, a central brokering function is the best approach, and in others, the freedom of choice and agility provided by allowing teams to operate independently provide the greatest benefits. Note that it is possible to perform governance, standardize architectures or operate across multiple clouds without the implementation of specific central brokering function. However, most organizations will find that a central brokering function makes sense as they scale their use of public cloud services. Figure 9 depicts examples of the two approaches.
Figure 9. Centralized Brokering vs. Independent Teams

Centralized Brokering vs. Independent Teams

**Centralized Brokering**
- User Group
- Project Team
- Agency/Line of Business
- Project Team

**Independent Teams**
- IT Operations
- Project Team
- Agency/Line of Business
- Project Team

Source: Gartner (March 2019)
A central brokering function is the best option when your priorities include:

- **Centrally governed application and service placement**: Organizations that require a high level of control over application placement (due to security, compliance or similar issues) may require a central authority to approve and select the appropriate venue for an application or service. These organizations will prefer a central brokering function because it allows for complete control of the process.

- **Consistent user experience for service requests**: A brokering function provides a level of abstraction between different cloud and technology providers and the end users. This abstraction layer may be a requirement to simplify and standardize the experience for external groups or lines of business (LOBs).

- **Consolidated provider contracting, procurement and management**: The establishment of a central brokering function also enables a consolidation of provider management and streamlined procurement processes. This approach enables cost-saving opportunities for the organization as a whole by leveraging capabilities such as reserved instances from cloud providers.

- **Enablement of automated provisioning across multiple environments**: A central broker function may repeat deployment processes many times a day and develop expertise and automation across multiple environments. This allows for faster and easier deployments over time, rather than relying on disparate groups of users to learn how to deploy to each provider.

- **Easier to implement prescriptive governance**: A central brokering function can establish default deployment models and ensure policy is implemented as part of the provisioning process. Prescriptive governance is easier to implement in this model rather than allowing independent selection by each group of users.

Independent selection of cloud services should be used when you:

- **Prefer a larger range of choices for developers and user groups**: A central brokering function may limit the choices of users in terms of services and providers, based on the central team’s preferences or skills. Sometimes this is a good thing, but in other cases, you may wish to allow independent choice.

- **Need to avoid process bottlenecks**: A central broker function can introduce process bottlenecks, particularly if not performing optimally. In some situations, you may allow independent service choice for those groups that need speed and agility above all other considerations.

- **Need to avoid political issues**: The question of “who makes the call” in terms of what application runs where can be a hot political issue in some organizations. In this case, a central brokering function may not be politically possible, especially if the remote LOBs insist on complete independence.

- **Have no need for an external broker agency or complex transformation**: The implementation of brokering tools, skilled people and automated procedures can be complicated and expensive. Some organizations will forgo the implementation of a central
brokering function purely on the basis of cost and practicality. Examples would be small IT organizations or deliberately autonomous teams in a larger organization.

“On the Side” Self-Service vs. “In the Way” Self-Service

Self-service enablement helps realize the full benefits of public cloud services because it fosters user autonomy and supports business agility. Different self-service approaches enable varying degrees of autonomy, standardization, control and access to innovation. Each approach can distinctly address the requirements of different end-user personas in the organization. “5 Approaches for Public Cloud Self-Service Enablement and Governance” contains information on methods to implement self-service enablement for different use cases. However, at a high level, you must decide on the philosophy of an “on the side” approach or an “in the way” approach to self-service enablement.

An “on the side” approach stays out of the way of the deployment process, leaving the resource provisioning task to end users. It concentrates on assistance and ongoing monitoring of cloud resources to ensure compliance and efficiency.

Conversely, an “in the way” approach concentrates on executing the provisioning of cloud resources on behalf of the user as part of the request fulfillment.

Remember that these approaches are not mutually exclusive. Both approaches can coexist as part of an organization's self-service enablement strategy and will apply for different use cases. The approaches are depicted in Figure 10.
An “on the side” approach is indicated when these are your priorities:

- **Larger range of choices and functionality**: Developers and user groups may need access to the entire portfolio of services from the cloud providers. Providers are also innovating and adding new services at a tremendous rate. This approach allows end users to see and use all options from the provider’s portal as soon as they are made available.

- **Greater agility**: This approach avoids many central process bottlenecks and allows greater freedom for teams to leverage new technology to unlock new business opportunities.

- **Less complex cross-cloud provisioning systems**: This approach allows you to avoid the potentially complex implementation and integration of an in-house self-service portal and service catalog to different cloud environments and offerings.

- **End-user-friendly**: Some central IT organizations want to avoid the appearance of imposing non-negotiable restrictions on lines of business, or becoming the “department of no.” This approach allows the end-user groups to feel more in control of their destiny while still maintaining guardrails and audit checks on cloud usage.

An “in the way” approach should be used when you need:
Guaranteed compliance and control: The degree of standardization in this approach is much greater than in an “on the side” approach and potentially reduces risk exposure. By controlling the provisioning process from the central IT platform, unknown risks are avoided.

Consistent user experience for service requests: In some cases, the end users may be ill-equipped to learn the portals and systems of multiple individual providers. This approach allows central IT to provide a consistent interface for users and to mask the underlying complexity of provisioning to the different providers. The lower skills requirement for end users is an attraction of this approach.

Reproducible, standardized deployments: Centralizing the deployment processes means less “reinventing the wheel” by lines of business and user groups. Best practices can be created and leveraged by default in this approach.

Inclusion of on-premises or noncloud private resources: An “in the way” approach allows for in-house provisioning scripts or processes to be presented in the same interface alongside public cloud options. Organizations with a significant percentage of applications still hosted on-premises may prefer this approach for that reason.

Priority on prescriptive governance: This approach allows for easier, prescriptive-style governance because guardrails are automatically implemented as part of an “in the way” model. This approach may be preferable in those situations where prescriptive guardrails and preventative programmatic control are important.

Programmatic Control Governance vs. Descriptive Guideline Governance

Governance should prevent chaotic or reckless behavior on the part of the enterprise as a whole or any of its people individually. The aim of governance is to protect the interests of the business and ensure that risk and cost are managed effectively. Organizational governance principles will drive implementation either as programmatical controls or as descriptive organizational policies. Implementation of governance is covered in detail in “Solution Path for Enabling Governance of Public Cloud Computing.” To enforce governance policy, two approaches to consider are when to implement programmatic control and when to use descriptive guidelines (see Figure 11).
Programmatic control has the following strengths:

- **Guaranteed enforcement of governance requirements**: The problem with guidelines is that there is no guarantee that users will follow or even read them. Programmatic control ensures that default policies or retrospective enforcement is applied when cloud resources are provisioned.

- **Appropriate for critical or high-risk use cases**: Critical or high-risk use cases will rely on programmatic control and compliance checking; they cannot rely on guidelines.

- **Improved control of security and expenses**: Implementation of programmatic controls lowers the risk of security or compliance breaches and overspending on cloud resources.

Descriptive guidelines are useful for communicating high-level governance goals:

- **No concern for practical application**: Programmatic controls require implementation of code, tools and provider APIs. These are not always practical, in which case guidelines must suffice.

- **Easier implementation**: Publishing of guidelines does not require specialized skills, tools and APIs to implement programmatic controls on each provider.
Agility and choice for users: Programmatic controls can limit the choice or ability of users to access innovations from providers. In those scenarios where user groups must be allowed maximum freedom, it may be appropriate to not only provide them with guidelines, but also allow them unrestricted access in order to experiment.

Preventative Programmatic Control vs. Retrospective Programmatic Control

Preventative controls stop actions that are likely to be harmful to governance goals before they can execute. Preventative controls are most often associated with account provisioning, role management and access control.

Retrospective controls operate on policy violations detected in an audit that runs periodically, and an action is then performed to remediate that violation. Detailed guidance on implementation of both types of programmatic control can be found in “Implementing Governance for Public Cloud IaaS.” Figure 12 shows a graphical depiction of retrospective-style programmatic enforcement of policy.
Figure 12. Retrospective Programmatic Controls

Workflow for Retrospective Policy Controls

1. Change Happens to a Cloud Resource

2. Detect Change in Cloud Resource
   - Cloud Logging Services (e.g., AWS CloudTrail, Google Cloud Audit Logging, Azure Activity Log)

3. Check Compliance Against Policies
   - Reflect Change
   - Policy Compliance Database

4. If Policy Violated, Remediate
   - "Hard" Remediation
   - "Soft" Remediation
   - Cloud Audit and Compliance Services (e.g., AWS Config/Config Rules, Azure Policy)

Source: Gartner (March 2019)
Preventative style controls have these benefits:

- **“Instant” compliance and control:** These controls prevent unwanted or harmful action in your environment. Retrospective controls, on the other hand, allow an exposure time between the creation of an issue and the remediation, which is avoided when using preventative controls.

- **Simpler implementation:** The tools and processes required to audit, detect and remediate with retrospective controls can be complex and are often unique to each cloud provider. Preventative controls may be as simple as default templates or account policy restrictions, so they are usually easier to implement.

- **Minimized risk:** Preventative controls are required for use cases where no risk is tolerated.

Retrospective controls are appropriate when you need:

- **Better agility and choice for users:** By design, preventative controls limit user choice of functions and providers. In situations where you need to maximize agility and options for users, you may let them access services without restriction and concentrate on continuous auditing and remediation strategies.

- **Hard and soft remediation processes:** Preventative controls are by nature “hard” controls that enforce requirements in a non-negotiable fashion. In some situations, a “soft” remediation (for example, an alert message or email) may be preferred, and retrospective controls allow for this.

- **Policies that cannot be enforced at deployment time:** For example, a policy such as “no EBS (Elastic Block Storage) snapshots should be maintained for longer than one year” cannot be enforced immediately (by definition, it takes a year to be violated) but requires retrospective audit and control.

Standardized Deployment Architecture vs. Custom Deployments

The approaches considered here relate to architecture roles and functions within the IT department. Deploying to multiple cloud providers involves architectural choices related to networks, storage, identity, security, databases and other areas. The choice to create a centralized architecture function (often as part of a cloud center of excellence [CCOE]) is considered here. The alternative is to allow different groups to architect and deploy their services as they see fit. See “Adapting IT to Become the Broker of Cloud Services” and “How to Build a Cloud Center of Excellence” for detailed guidance.

Organizations will adopt a standardized deployment architecture model when they prioritize:

- **Economies of scale in a DevOps environment:** As the use of DevOps practices increases within an organization, it often begins to see inefficiencies and duplication of operations effort across product teams. There is a difference between an organization with a few product teams and a larger enterprise with dozens of product teams. As these efforts scale up, it becomes more effective to centralize some aspects of deployment architecture and develop an internal platform for the use of the product teams. This is sometimes called platform ops and is described in detail in “Strengthen Your DevOps Capability With Platform Ops.”
Abstraction across public cloud, private cloud or virtualized on-premises infrastructures: Organizations operating across multiple public, private and on-premises environments may wish to implement consistent architectural practices relating to networks, security, identity and so on, and thus will institute a central cross-platform architecture effort for this purpose.

Consistent, reusable, tested deployment models: Different groups may be reinventing the wheel when deploying into a new cloud environment, and often, expertise and experience in that environment is already present somewhere else within the organization. Implementation of a standardized architecture allows for reusable and consistent deployment models that are easier to implement and support.

Compliance to regulations: Some environments require a controlled and standardized deployment architecture to ensure compliance with industry regulations or country-specific legislation.

Avoid fragmentation of skill sets and tools across providers: Most organizations that adopt cloud services in an ad hoc manner start to see fragmentation of skills and tools used in different parts of the organization. This approach avoids that fragmentation and allows for development of in-depth skills for each provider in one place.

A custom deployment architecture approach is used when:

Broad choice and flexibility of cloud services is important: User teams that rely on a central architecture function may be more limited in choice of services that can be used, depending on the expertise and preference of the central group. Organizations that value freedom of choice above other priorities will allow each group to deploy with its own architecture to providers.

Fast implementation is preferred: It is quicker for a user to get up and running without thinking about conforming to centralized standards or best practices. In situations where speed is valued above all else, you will adopt this approach. Keep in mind that this depends on the skill set of the user. If a user must spend a significant amount of time understanding and creating a deployment architecture, a central team may be a faster option.

A skilled central team is not required: It can be expensive and time-consuming to establish a person or group with architectural skills across multiple cloud and on-premises environments. Similarly, the adoption of central platforms based on common PaaS and continuous integration/continuous deployment (CI/CD) tools can be difficult and expensive. For these practical reasons, you may choose to use a custom architecture approach, rather than a central team, for each user group.

Lock-in to a single PaaS or orchestrator is a concern: Some organizations are concerned about lock-in to a single PaaS or orchestration platform. Although they can change or add cloud providers, the adoption of a centralized architecture may create too much reliance on that platform because all their processes and tools are based on a single vendor. In this case, they may deliberately adopt custom deployment approaches for the different infrastructures to which they deploy services.
Use-Case-Based Silos vs. Composite Multicloud

A use-case-based approach is where one set of applications is run on a particular cloud provider and a different set of applications is used on another, with strict separation between the environments. The consideration here is whether your multiple clouds are implemented with separate services and applications restricted to specific cloud providers, or if you are willing to architect integration and orchestration across cloud providers for applications and services. Use-case-based silos are an architecture based on using multiple providers but avoiding complex integration between the cloud provider environments. Specific applications or services are restricted to run only on particular infrastructure silos, and they are managed by using the tools of each silo or cloud provider. Figure 13 shows an example of this type of approach.

**Figure 13. Use-Case-Based Provider Silos**

The alternative is to implement true multicloud architectures for some use cases, where some applications span multiple cloud providers or may even be a composite of multiple different types of services and providers. Gartner identifies both composite and redundant architectures in these multicloud approaches, as shown in Figure 14.
Figure 14. Composite and Redundant Multicloud

Composite and Redundant Multicloud

Source: Gartner (March 2019)
A detailed discussion of multicloud architectures can be found in “Technology Insight for Multicloud Computing.”

Organizations will adopt use-case-based provider silos for these reasons:

- **As a starting point:** Most organizations start using cloud services based on use cases matched to each provider. As cloud strategy and skills mature, they begin to consider composite and redundant multicloud architectures.

- **To accommodate mergers and acquisitions:** Scenarios where merging business units have different cloud providers and strategies can be accommodated with this approach.

- **To avoid complexities of multicloud integration:** Separating the infrastructure silos, and not attempting integration across silos, allows you to avoid the major problem of multicloud implementation: complexity of infrastructure integration across cloud providers.

- **As a broad choice and to benefit from flexibility of cloud services:** Implementation and integration of a new cloud provider in a multicloud scenario involves significant effort, so adding a new provider to a multicloud environment is not to be taken lightly. By taking a siloed approach to the use of cloud providers, it is much easier to implement a new provider for a specific use case because the complex integration to that provider is not necessarily required.

- **To fully leverage native capabilities in silos:** This approach allows you to take advantage of unique capabilities offered by providers in each silo rather than resorting to a “lowest common denominator” approach across all the infrastructure types. An example would be leveraging a machine learning capability from one cloud provider while using another provider for other components of the application.

As the use of multiple cloud providers increases, organizations may mature to composite or redundant multicloud architectures. Some examples of these architectures are included in “Assessing the Strengths and Weaknesses of High-Value IaaS and PaaS Multicloud Use Cases.”

Organizations will choose this approach for specific use cases when they require:

- **Workload portability/exit strategy:** Gartner expects that most applications will not migrate between cloud providers, but some use cases require portability between public cloud services and private environments, or between multiple public cloud providers. In addition, some organizations want the flexibility to be able to leave a provider by choice, whether due to dissatisfaction or simply a desire to improve negotiating leverage.

- **Exploitation of unique high-level services for components of applications:** An organization that has concentrated on a strategic provider for most applications may find that a specific component of an application requires a specific service that is not available from that provider. For use cases where immediate access to the newest innovations is of critical importance, a multicloud strategy allows exploitation of new capabilities as soon as they are introduced into the market. In this instance, a component of an application may be based on a different provider in order to exploit that unique capability.
- **Cost or performance optimization:** Some use cases benefit from an ability to move quickly between providers to take advantage of cost changes. For example, a provider may offer a new instance size or pricing model that allows for better optimization of performance or cost profiles. Similarly, load balancing between providers may be desirable for some applications, depending on geographic location of users or changes in traffic patterns.

- **High-availability scenarios and reduced risk:** Both active/active and active/passive scenarios across cloud providers are potential architectures in a multicloud scenario. This can result in improved application resilience by reducing the likelihood that systemic failure of any given cloud provider will result in application failure. This can mitigate other vendor-related risks as well.

Central Visibility and Control vs. Distributed Management Centers

Many organizations already have multiple cloud platforms in use, often commissioned by groups outside of IT. This can rapidly lead to “management sprawl” and a proliferation of tools and processes. The adoption of cloud management platforms and tools is an important part of consistent governance and operation in the multicloud environment for most organizations. Some organizations will plan a very centralized approach, with the cross-platform span of management tools a relatively important factor. Some will be happy with a more decentralized approach and may allow different groups to manage different cloud platforms with their own toolsets. These approaches are shown in Figure 15.
Figure 15. Centralized Management vs. Distributed Management

Centralized Management vs. Distributed Management

<table>
<thead>
<tr>
<th>Central Management</th>
<th>Distributed Management Centers</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Operations</td>
<td>IT Operations</td>
</tr>
<tr>
<td>Project Team</td>
<td>Project Team</td>
</tr>
<tr>
<td>Agency/ Line of Business</td>
<td>Agency/ Line of Business</td>
</tr>
<tr>
<td>Project Team</td>
<td>Project Team</td>
</tr>
</tbody>
</table>

Centralized Management Toolset

- Standardized Management Toolset
- Native Tools
- Private Virtualization
- Private IaaS/PaaS
- Public IaaS No. 1
- Public IaaS No. 2

Distributed Management Centers

- Central/Legacy Toolset
- Toolset No. 1
- Toolset No. 2

Native Tools
- Private Virtualization
- Private IaaS/PaaS
- Public IaaS No. 1
- Public IaaS No. 2

Source: Gartner (March 2019)

Organizations will move to a centralized and standardized management tool approach when they prioritize:

- **Consistent skill set and improved efficiency:** A centralized approach allows a single skilled group to concentrate on reducing overhead and duplication in management tools. This group can develop enhanced capabilities because it specializes in operations for all parts of the organization. The same skills do not have to be developed in multiple parts of the organization for each new use case.

- **Operations functions performed by experts:** Those with experience in infrastructure and operations issues are performing operations in this model. Developers or end users in a line of business trying to perform operations may not understand or allow for problems and exposures that experienced operators will avoid.

- **Lower skills requirement for lines of business or developers:** A central management approach allows for the central team to “shield” the users from the underlying complexity of multicloud operations. A layer of abstraction can be added between the users and the infrastructure to improve end-user or developer productivity.

Organizations will use a distributed management center approach when they prioritize:

- **Choice of providers and services:** Forcing a line of business to use only the services supported by the central management team may stifle creativity and agility. Those organizations that strive to allow maximum freedom for developers and user groups will have to allow for multiple centers of management if those groups choose to use providers or services outside the set supported by central IT.

- **Agile teams:** In addition to service choice, some organizations value the ability for product teams to operate at their own pace, completely independent of any central IT authority. These types of requirements may also lead to the independent team running its own management toolsets or provisioning environments.

- **Fast implementation:** For organizations that lack an existing central management team, the time frame required to select tools and train a central management group may be unacceptable for lines of business that need to progress digital business initiatives quickly. In these situations, distributed management centers may be set up. This can be a temporary arrangement, until central management comes online at a later date, or a more permanent approach.

- **Avoidance of organizational challenges:** Political issues within the organization can favor a distributed management approach. Because budgets and expense lines are typically owned by lines of business, it can be challenging to dictate a centralized management approach. Additionally, if management skill sets and tools are distributed across silos of the organization already, consolidating to a skilled central team and rationalizing toolsets may present organizational challenges that are insurmountable.
Cross-Platform Tools vs. Native Tools

Broad cross-platform cloud management tools are also known as cloud management platforms (CMPs). Gartner defines the functionality expected from cloud management tools in “Evaluation Criteria for Cloud Management Platforms and Tools.” A tool can span across multiple cloud platforms and/or functional categories.

These “broad” tools can be:

- Cross-platform (providing functionality across several cloud platforms)
- Multifunctional (addressing more than one functional category of cloud management)
- Both cross-platform and multifunctional

In contrast, deep tools have a narrower focus, concentrating on specific cloud platforms or functionality. Deep tools are necessary when broad tools cannot address all the functional capabilities within a platform, or when new features are introduced on a cloud platform faster than the broad tool can accommodate. Deep tools are typically (but not always) offered from a cloud provider itself, in which case these tools may also be called provider-native tools or simply native tools.

Figure 16 shows the concepts of broad and deep tools and the requirement for each type.
The choice of an approach involves finding a balance between the competing requirements of efficient consistency across different cloud platforms and access to different native functionality within individual cloud platforms.

The reasons to consider an approach based on broad cross-platform tools are:

- **Consistency**: The key value proposition of a broad cross-platform tool is consistency across cloud environments or functions. A broad tool can offer a single view across multiple cloud platforms for areas such as monitoring, cost management, alerting and inventory. By abstracting cloud platforms’ proprietary APIs, a consistent process can be used for provisioning or security. A common service portal for users across environments is another example of the value of a cross-platform tool.

- **Usability**: Rather than training in multiple tools and dealing with many user interfaces, a cross-platform tool allows a “single pane of glass” approach across multiple cloud platforms.

- **Feature parity**: Broad tools can provide capabilities that are not provided by native tools to augment the features of a cloud platform. Cross-platform tools can provide feature parity across all managed environments.
Reduction of cloud provider lock-in: Heavy reliance on platform-native tools makes it increasingly difficult to exit from that provider. Some organizations rely on cross-platform tools to reduce their dependence on a particular cloud provider.

The reasons to consider an approach based on deep/native tools include:

- **Access to unique cloud platform capabilities:** Cloud providers are working hard to provide value-added features with unique capabilities in order to attract users. These features may not be accessible via a broad tool that only addresses a lowest-common-denominator style of management. For example, a broad tool may provide the ability to provision VMs, block storage and Amazon Virtual Private Cloud (VPC), but have no capability for services like Amazon S3, Amazon CloudFront, Amazon Relational Database Service (RDS), Amazon DynamoDB and so on. To access these value-added services on Amazon Web Services (AWS), an administrator would need to use native AWS tools in addition to the broad tool.

- **Provider choice:** Broad tools limit the choice of providers and services to those that they support. Native tools will be introduced as part of the adoption of any new service. Also, reliance on a third-party tool for management functions across cloud platforms means that your processes, scripts and workflows are locked in to that vendor’s tool specifically. The risk of committing to a single tool for a management function must be balanced against the benefits of a consistent cross-platform tool.

- **Direct access to platform APIs:** Enforcing consistency and provisioning through a single broad tool rather than directly through native cloud provider APIs may make you dependent on the tool’s APIs and vendor instead. This is not acceptable for many developers or users, who may need direct access to platform-native APIs.

- **Additional software costs:** Third-party tool implementation can be expensive and complex. In particular, the deployment of CMPs tends to be costly and lengthy. In many cases, a more limited function tool (for example, for cost management or security compliance) can be a better choice. Conversely, native tools are generally available as part of the cost of the service.

**Guidance**

This section summarizes Gartner’s guidance for selecting the best management approach for each multicloud use case.

**Brokering and Self-Service**

A centralized brokering function should act in a consultative role to central IT, business unit IT and consumers of cloud services. Too much intermediation by a brokering function causes users to avoid the brokering function and go straight to the providers — too little and it adds no value to the organization. Implement a centralized brokering function if your need for consistency and efficiency outweighs the cost and reduced freedom of choice involved in creating, training and equipping such a structure within your organization.
Self-service of computing capabilities is desired by most organizations as they mature their cloud computing initiatives. The autonomy of users and developers translates into better business agility, faster reactivity to market conditions and greater competitiveness for the business. However, ungoverned self-service access to cloud services can introduce the risk of security breaches, loss of data, compliance issues or running over budget. Implement “in the way” self-service only when you have strong requirement for control or to include noncloud infrastructure. Otherwise, try to allow direct access from user groups to cloud services with “on the side” self-service.

Governance

Governance in the world of cloud computing involves auditing, guidance and control of user activity. As public cloud usage evolves, the idea of comprehensive rules and restrictions gives way to principle-based governance because public cloud adoption has become too complex for a rule that applies to every eventuality. A governance team oversees functions such as cloud computing policy, cloud provider selection and relationships, cloud solution architecture, and workload placement. It provides both guidelines and guardrails that improve outcomes and manage risks. Focus initially on creating policies and guidelines before trying to select programmatic governance tools. Prioritize enabling business agility and self-service in a sustainable fashion. Flexibility is extremely important to ensure that the organization can innovate. The tendency for central IT is to overcompensate with too much heavy-handed governance. Destroying the agility and choice of services provided by the cloud can be almost as bad as allowing cost overruns or compliance exposures. Allow users some freedom initially, and fine-tune governance controls over time.

Publish descriptive usage guidelines for all cloud services, and establish prescriptive programmatic control, especially for those services with higher-risk profiles. Develop and continually review overall organizational governance principles, which will then drive implementation either as programmatically enforced controls or organizational policies (when programmatic control may not be feasible).

Implement preventative controls to stop users from creating resources or performing some action on an existing resource (for example, you must have the role “network admin” to create a new virtual network). Other policy violations emerge in audits or monitoring, and an action is then performed to remediate that violation. These are known as “retrospective controls.” Implement preventative programmatic controls for those services where no risk is tolerated. Complement preventative controls with retrospective controls and remediation strategies when tools are available to do so.

Architecture

An emerging set of technical roles involves architecture and engineering across the silos of technology used by an organization. These roles are sometimes called cloud architect and cloud engineer, and they are discussed in more detail in “Analyzing the Role and Skills of the I&O Professional in DevOps.” A centralized, cross-cloud architecture group is the knowledge center that aids in navigating the sea of technology choices in a multiprovider world. This team can glue together disparate technology components to deliver services effectively and securely. The technical people in this group need to cover traditional IT and data center technologies as well as public cloud. The requirement is not only to enable cloud adoption, but also to create and document
a general application and service placement framework across all technology silos: traditional IT, private cloud and public cloud. Agreement on the application and service placement framework throughout the organization aids with political issues and improves efficiency of multicloud operations. Everyone must understand the reasoning behind placement decisions, and they must buy in to the centralized framework.

In an environment where product delivery teams are responsible for deployment and operations of their own applications, a shared platform may also be built in-house. This multicloud platform cuts down on duplication of technology and focuses expertise. In addition to centralized architecture, a platform ops team can be created that is responsible for providing a self-service development, deployment and operational platform that enables multiple software delivery teams to build and operate their own products. Consider creating a dedicated platform ops team when economies of scale are needed in DevOps environments. Otherwise, allow DevOps freedom with appropriate governance and monitoring.

Implement composite multicloud architectures when specific portability, cost optimization or high-availability functionality is required. Explicitly decide whether your application architecture should be multicloud. Applications and services are dependent on the platform components they use. In a multicloud environment, the components of each application may be entirely portable, entirely proprietary or somewhere in between. Examine the nature of each application and service to determine whether portability should be a priority, and to what extent you’re willing to trade off portability for productivity or capability gains. In addition, you’ll need to consider which applications will require multicloud deployment and what type of multicloud architecture they’ll need:

- Redundant multicloud architecture, in which complete applications are replicated across multiple cloud providers
- Composite multicloud architecture, in which components of a single application are distributed among multiple providers
- Redundant and composite multicloud architecture, in which components of a single application are replicated and distributed among multiple providers

Operations

Most organizations begin their cloud journey with ad hoc adoption of cloud services by multiple groups, with each group managing its own environment to some extent. Thus, it is natural to operate with distributed management initially and then evolve to central control by training and equipping a specialized operations team when required. In situations where autonomous groups require independence from a central operations team, you will maintain a distributed management structure. However, most organizations will see significant benefit in cost and efficiency of operation by consolidating tools and skills into a central group of experts.

When operating in a multicloud environment, administrators must balance the competing requirements of efficient consistency across different cloud platforms with access to different native functionality within individual cloud platforms. Organizations will combine platform-specific tools
with cross-platform tools, depending on requirements. Expect to use multiple tools — there is no single perfect solution.

First, implement platform-specific tools to familiarize yourself with each cloud platform’s native functionality. Add cross-platform tools later to support broad cross-platform consistency or to provide incremental functionality to some cloud platforms. Invest in broad capability management tools only when you have centralized cloud operations and are operating each provider effectively with native tools. Build do-it-yourself (DIY) tools as a last option only.

Gartner Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

“Solution Path for Enabling Governance of Public Cloud Computing”

“Implementing Governance for Public Cloud IaaS”

“Technology Insight for Multicloud Computing”

“Adapting IT to Become the Broker of Cloud Services”

“5 Approaches for Public Cloud Self-Service Enablement and Governance”

“Strengthen Your DevOps Capability With Platform Ops”


Evidence

1 “Market Insight: Multicloud Becomes Essential for Cloud IaaS Offerings”