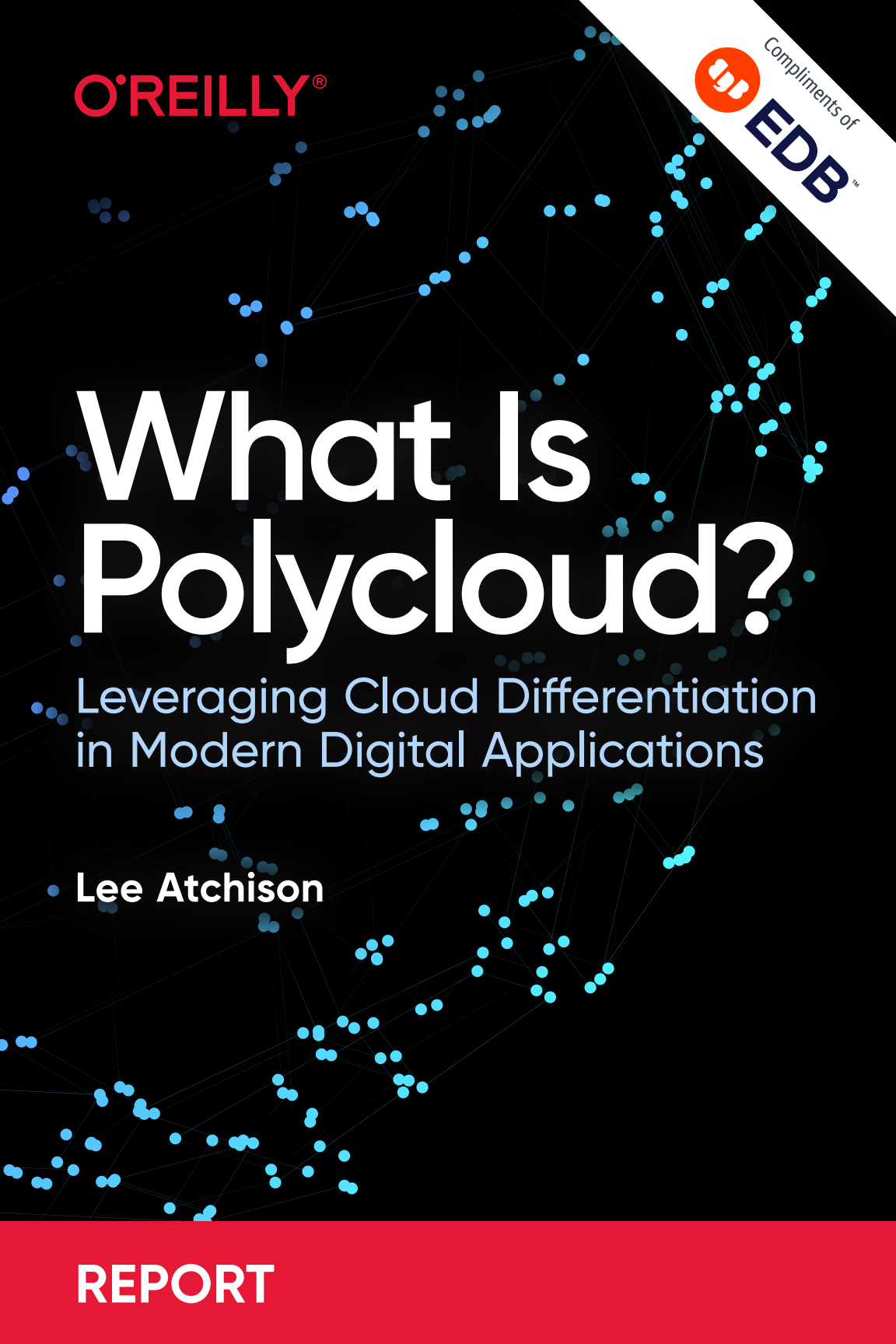
O'REILLY®
Compliments of
EDB™
What Is Polycloud?
Leveraging Cloud Differentiation in Modern Digital Applications
Lee Atchison
REPORT

What Is Polycloud?

Leveraging Cloud Differentiation in Modern Digital Applications

Lee Atchison

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

What Is Polycloud?
by Lee Atchison

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Jennifer Pollock
Development Editor: Jill Leonard
Production Editor: Kate Galloway
Copyeditor: Shannon Turlington
Proofreader: Sonia Saruba
Interior Designer: David Futato
Cover Designer: Randy Comer
Illustrator: Kate Dullea

May 2021: First Edition

Revision History for the First Edition
2021-04-28: First Release

978-1-098-15261-1

[LSI]

Table of Contents

CHAPTER 1
What Is Polycloud?

Cloud computing is central to most modern organizations' infrastructure operations, and cloud native applications are becoming more the norm for new application development. Additionally, as competition between cloud providers heats up and legitimate competitors to Amazon Web Services (AWS) begin to take hold, applications that span more than one cloud provider are becoming more typical.

The term *multicloud*, used to describe these cloud-spanning applications, has been in our vocabulary for many years. But the term has never been well-defined. It describes using more than one cloud provider, but it does not specify how those cloud providers are used. Over the years, many different models of multicloud have come into modern usage, and each of them has different sets of advantages and disadvantages. Yet the terminology used to describe each of these has not yet been defined in a standardized manner.

Enter *polycloud*. Polycloud is one particular model for using multicloud that has gained significant traction in recent years, especially as cloud providers have added higher-order capabilities above basic cloud infrastructure services such as storage, compute, and networking. These higher-order capabilities have encouraged cloud specialization and differentiation. Many applications aim to ignore those differentiations, but polycloud is an architectural model that encourages cloud specialization and differentiation.

In this report, we will define what polycloud is and how it compares to other types of multiple cloud deployments. Additionally, we will show how your application can evolve into a polycloud architecture, and we will examine some common examples of polycloud.

Types of Cloud Deployments

There are many types of multicloud deployments that all serve different purposes for the organizations that use them. Some were designed for strategic purposes, and others emerged from organic growth and individual business choices. Let's take a closer look at how these multicloud deployments work in practice.

Hybrid Cloud: Moving Off Premises

A typical first approach to a cloud strategy is for a company to move part of its application to the cloud, creating a hybrid cloud deployment. In a hybrid cloud, components of the application are on premises and components of the application run in a cloud provider using cloud services. Applications using this model typically will employ basic cloud services such as object storage, database, and other application building blocks. An example hybrid cloud deployment is shown in Figure 1-1.

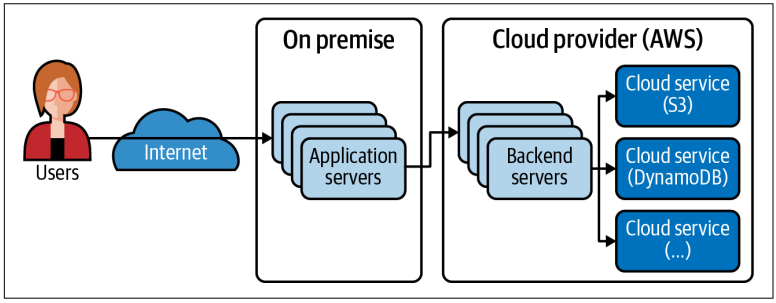

Figure 1-1. Hybrid cloud deployment

Cloud Native: Fully Off Premises

To avoid requiring on-premises components, many applications are moved to a full cloud deployment. They may either be migrated from on premises to a full cloud deployment model, as shown in Figure 1-2, or be developed in a cloud native environment. Either

way, the entire application runs in a full cloud or cloud native environment.

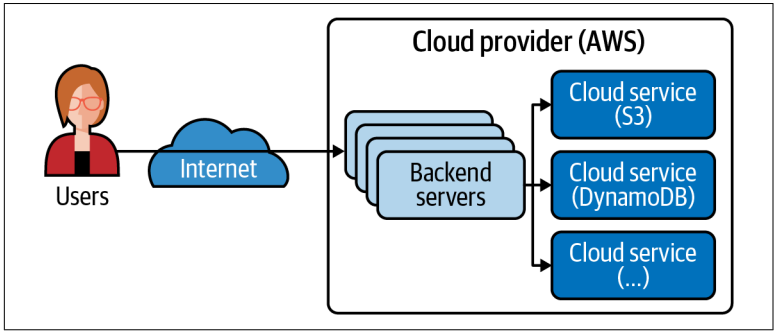

Figure 1-2. Full cloud and cloud native deployments

Multicloud: Leveraging Capabilities

As cloud providers expand their offerings, applications become more complex and companies become more cloud savvy. This leads many companies to adopt a simple multicloud architecture for their applications, such as the one shown in Figure 1-3.

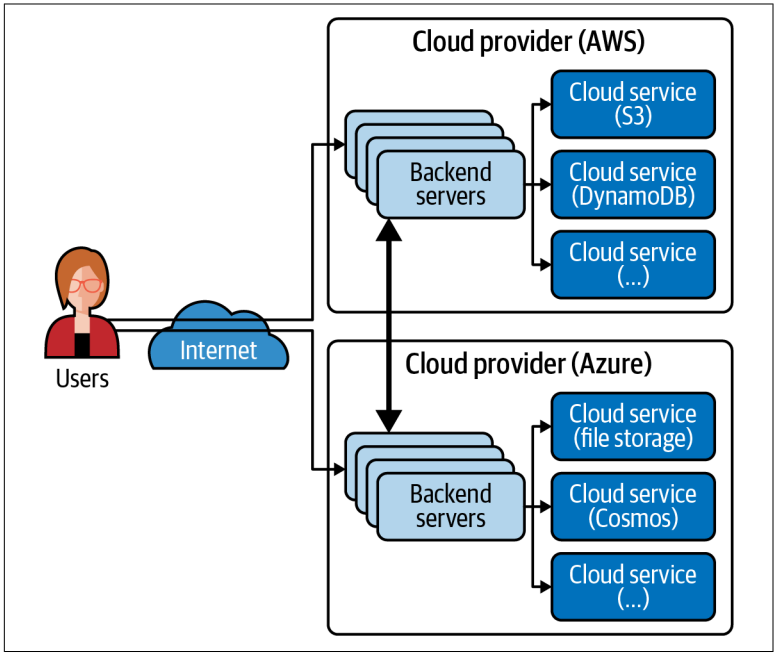

Figure 1-3. Simple multicloud deployment

In this model, either the entire application is duplicated across multiple providers or parts of the application are on one cloud provider and parts are on another cloud provider. In either case, typically no specialization exists. The application does not usually take advantage of feature optimizations provided by individual cloud providers. In fact, use of specialized cloud provider services may be discouraged or even disallowed for applications. Instead, only basic, standard cloud capabilities tend to be used.

Polycloud: Cloud Specialization

Cloud providers are creating more sophisticated services and are starting to differentiate these capabilities among themselves. These differentiations can take a variety of forms:

- The types of infrastructure provided, such as Microsoft Windows Servers versus Linux servers
- The types of services offered, such as specialized machine learning and artificial intelligence services
- The topics of analytics provided or auxiliary tools and services provided over and above the base capabilities

Large, complex applications take advantage of many of these specialized services and often will use different services from different cloud providers, based on the specific specialized services available. As such, an application may make use of multiple cloud providers to provide *different* service capabilities.

This is called polycloud and is illustrated in Figure 1-4. In this diagram, we see AWS providing basic cloud services for the application. We see Microsoft Azure providing Microsoft Windows–based servers for a portion of the application. And we see Google Cloud Platform (GCP) providing AI capabilities for the application. Each cloud provider is giving unique capabilities to the application.

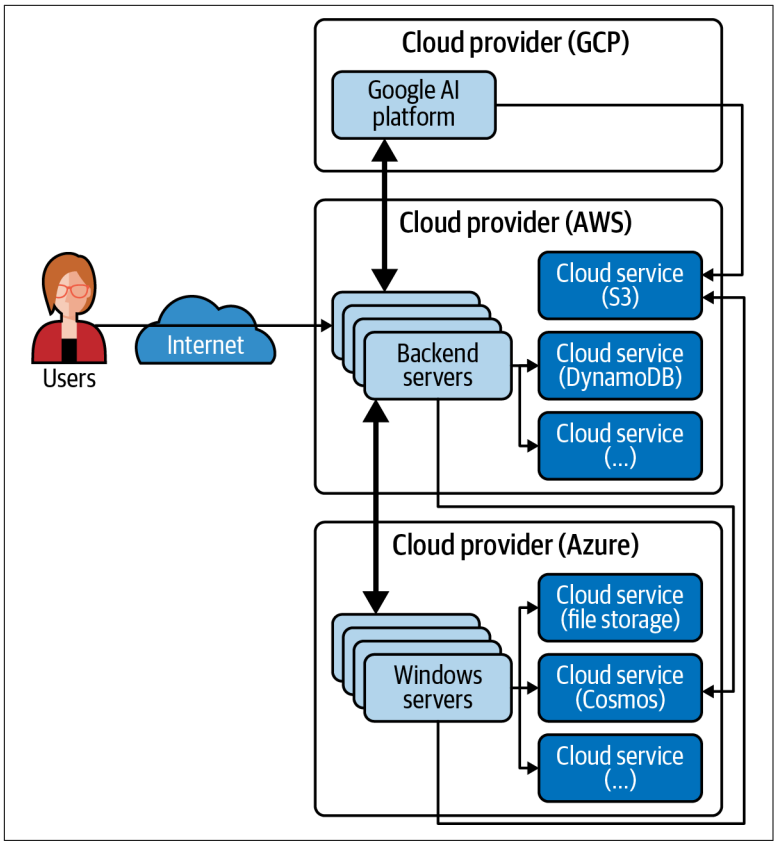

Figure 1-4. Polycloud deployment

While multicloud is about using multiple cloud providers in a cloud-agnostic manner, polycloud is about using specialized capabilities of multiple cloud providers.

Multicloud stresses the *generic* nature of cloud capabilities. Polycloud stresses the *specialization* of cloud providers. Multicloud tries to downplay the differences between the providers, while polycloud uses the differences to the benefit of the application.

Now that we have a clearer understanding of what various multicloud deployments look like, let's take a closer look at polycloud in particular and the various paths your organization can take to find itself working in this type of architecture.

CHAPTER 2

Paths to Polycloud

How do you get to the point where your application is operating in a polycloud environment? Some arrive at a polycloud architecture quite accidentally. As their applications, organizations, and cloud dependence grow, they find they are, before they even know it, operating in a polycloud environment. They experience all the advantages and disadvantages that polyclouds offer without actively making a decision to arrive there.

Other organizations take very deliberate and planned steps to build out a polycloud architecture. They do it because of the advantages that such an architecture brings, and they believe these advantages outweigh the potential disadvantages.

Let's look closer at the different routes that companies take in their journey toward polycloud architectures.

Managing Risk

Risk management is a popular, albeit often accidental route to polycloud.

Take a look at Figure 2-1. Here you see a typical cloud architecture that has been optimized for risk management. Multiple cloud providers are used to provide parallel services, with the hope that if a single cloud provider has an operational issue, the other cloud providers will not and traffic can be routed to those other providers.

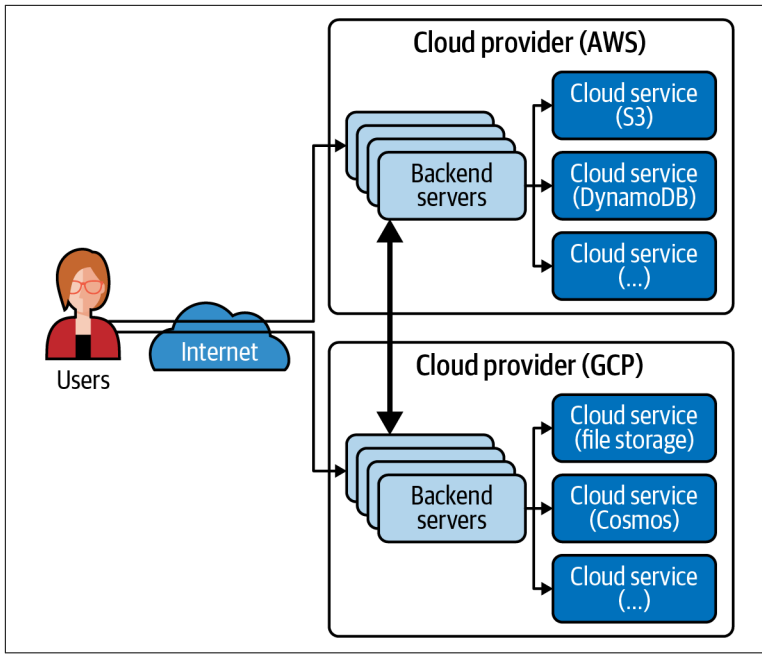

Figure 2-1. Multicloud for risk management

This is not a polycloud architecture, though. Instead, the creator of this architecture, as we learned previously, created a generic multicloud architecture. However, once multiple cloud providers are in use, it's very easy to start making use of specialized services from particular providers while still working toward a risk-tolerant approach to multicloud. This tendency toward specialized services leads to a polycloud approach to application architecture, even in the guise of a planning for a more simple multicloud architecture to reduce risk.

To illustrate, take for example Figure 2-2. Here you see a single application that is deployed uniformly across multiple providers in order to provide higher tolerance toward risk from provider-availability issues. However, in this example, each provider is using a unique set of cloud capabilities that is differentiated for that particular provider. The services running on AWS make use of Amazon SageMaker, while those on GCP use the Google AI platform. While the two services provide similar capabilities, their implementation, specific capabilities, and actions are radically different. They are, in all respects, unique offerings.

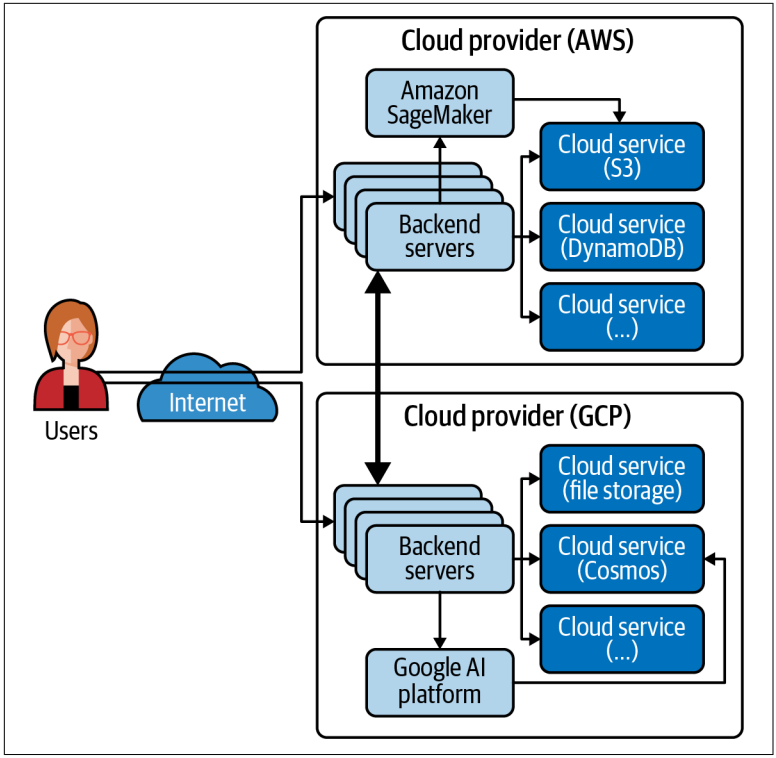

Figure 2-2. Multicloud risk management over time changes to polycloud

This differs from the classic multicloud approach in Figure 2-1, where a uniform product architecture is used across all providers. By using differentiated capabilities in each provider, you are creating a polycloud application even though your motivation is uniform diversity for risk management.

Another classic example where polycloud results from a risk management plan is in archival and backup storage. This is illustrated in Figure 2-3. This is a common situation, as different cloud providers are often used to store backups and duplicate copies of critical data in order to preserve against failure in the main cloud provider.

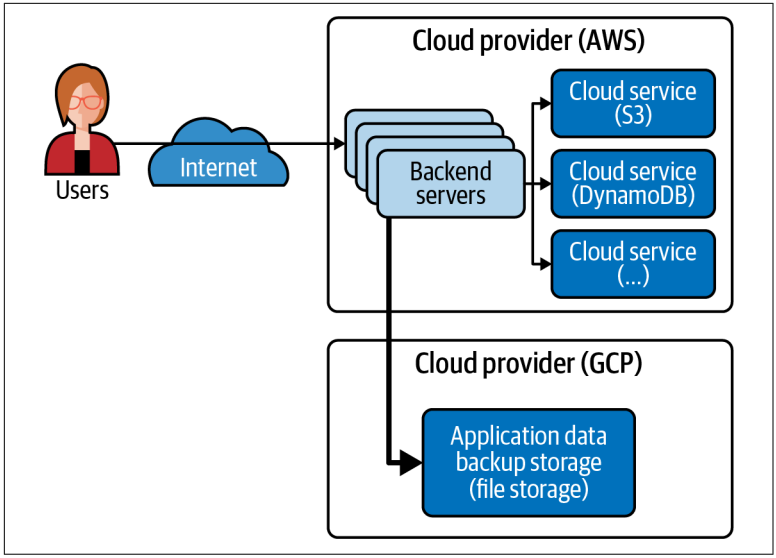

Figure 2-3. Polycloud backup management

This model allows backups to be kept independently and gives some ability to recover from the failures of a single cloud provider. This is a true polycloud model, because one cloud provider is being used for a specific capability—archival storage—while another cloud provider is being used for active application services.

Managing Costs

Managing costs is a common model used by organizations that leads to a polycloud application.

Many organizations, especially organizations with large operational budgets, will give higher weight during architectural decision-making to criteria based on cloud costs.

An example of this higher weighting is to determine that the cost of object storage may be cheaper in one cloud provider, like AWS, while the cost of server instances may be cheaper in another provider, like GCP.[1] As such, an application may become polycloud

1 This is a hypothetical example, not an indication of which cloud provider is actually cheaper or better.

simply to optimize cloud costs for a given application. This is illustrated in Figure 2-4.

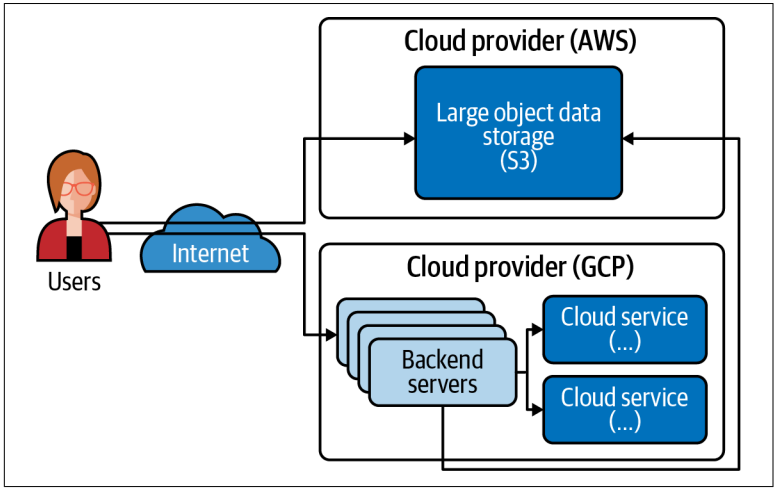

Figure 2-4. Cost-optimized polycloud

Care must be taken in this sort of operational plan, since data transfer costs between cloud providers may negate any specific service cost advantages. As such, you are more likely to see this model used with less active data, such as for archives and backups, where data transfer and interprovider communications may be more limited.

Selecting Features

Some cloud providers have specialties and provide better capabilities in some of their cloud services than other providers. This results in certain types of services being better in different cloud providers. For example:

- Amazon S3 is a solid object storage system that allows quick access and inexpensive deep storage all within the same infrastructure. If your application has large data requirements with varying accessibility needs, AWS is a cloud provider to consider.
- GCP provides solid management of running containers in a Kubernetes cluster. While other providers offer Kubernetes cluster management, Google has more Kubernetes knowledge and experience, and its offering is more mature than most other

providers'. If your application depends heavily on containers and container orchestration, Google's expertise is quite valuable.

- Microsoft Azure offers the most experience hosting Microsoft-based servers, resulting in the most capable and reliable hosting for Microsoft-based applications. An application or service that requires Microsoft capabilities is a good choice to operate on the Azure cloud.
- GCP has historically had very solid AI and machine learning (ML) capabilities, much more so than other cloud providers. However, lately, AWS has also been focusing in this area, and its AI/ML capabilities have expanded significantly in recent years, making AWS a good choice for services highly dependent on AI/ML capabilities.
- AWS Lambda is one of the best serverless computation platforms available today, allowing you to build specialized functions without having to be concerned about the server deployment and scaling requirements as much as you do with a server-based architecture. GCP provides similar capabilities, but AWS's offering is more mature.
- Microsoft Azure has invested heavily in Internet of Things (IoT) capabilities for its cloud offering. While AWS has recently started to expand its IoT capabilities, Azure's focus on IoT is notable and provides more complete capabilities.

These are just some examples that depend on a number of variables. Specific applications will have specific requirements that will encourage the use of specific capabilities by specific providers. As such, it's easy to see how a large, complex application with multiple moving parts might make use of capabilities that are the specialties of different cloud providers, and hence might be suitable for a polycloud-hosted architecture.

Maintaining Development Team Independence

In a large organization, individual development teams sometimes have a level of independence that allows them to make some decisions about where and how their services are to operate.

Sometimes a development team will have specific guidance and constraints about how it operates its services in its own operations center but may have some flexibility for experimentation outside the systemic structures. This flexibility can lead to testing, experimenting with, and ultimately using specific cloud services in production that are unique and provided by specific cloud providers. When individual development teams are making independent decisions, not surprisingly, those decisions are often different.

Frequently, the net result is the organization ends up using a patchwork of services offered by different cloud providers for distinct parts of larger applications. This leads to individual applications relying on multiple cloud providers. More specifically, these applications have parts that depend on certain capabilities as provided by different cloud providers. These different services, or components of the application, are using different cloud providers, as shown in Figure 2-5.

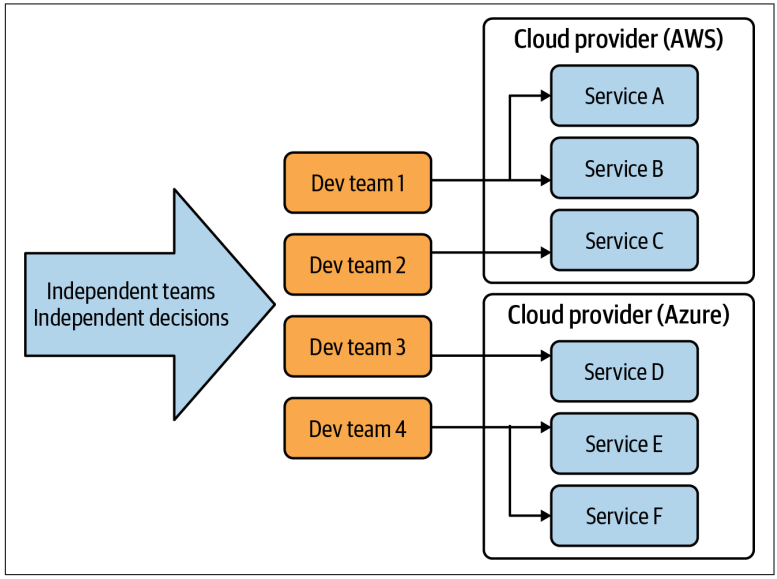

Figure 2-5. Independent development teams making independent decisions

This process, which leads to a polycloud architecture, isn't intentional but is the result of independent development teams operating independently. It is a side effect of the operational choice to allow teams to make their own decisions.

Allowing individual teams to make operational choices has many advantages, such as an improved sense of team ownership. This type of polycloud end result may appear suboptimal from a global standpoint—your application is using similar services from separate and distinct providers—but in fact may provide optimizations at the organizational and team level that far outweigh any perceived disadvantages.

Evolution Over Time

Rather than independent decisions being made by independent development teams, polycloud can result when independent decisions are made over time, over the life span of an application.

As an application grows and develops over time, and new and updated features and functionality are added to the application, the design and implementation of those pieces of new functionality are often architected and developed in a vacuum, independent of other decisions made at other times during the lifetime of the application.

Simply put, decisions usually occur like this:

> At this point in time, implementing *this* specific feature that is currently considered important…
> …is best served by using *this* capability…
> …offered by *this* specific cloud provider.

As time goes on, many such decisions like this are made, each decision independent from the others. This process often leads to an application with many individualized dependencies on different services, different capabilities, and, ultimately, different cloud providers.

In Figure 2-6, Service A may have been built at one time, Service G may have been built at another time, and Service F at a completely different point in time. At each point in time, different criteria led to different decisions and, ultimately, the use of different cloud providers.

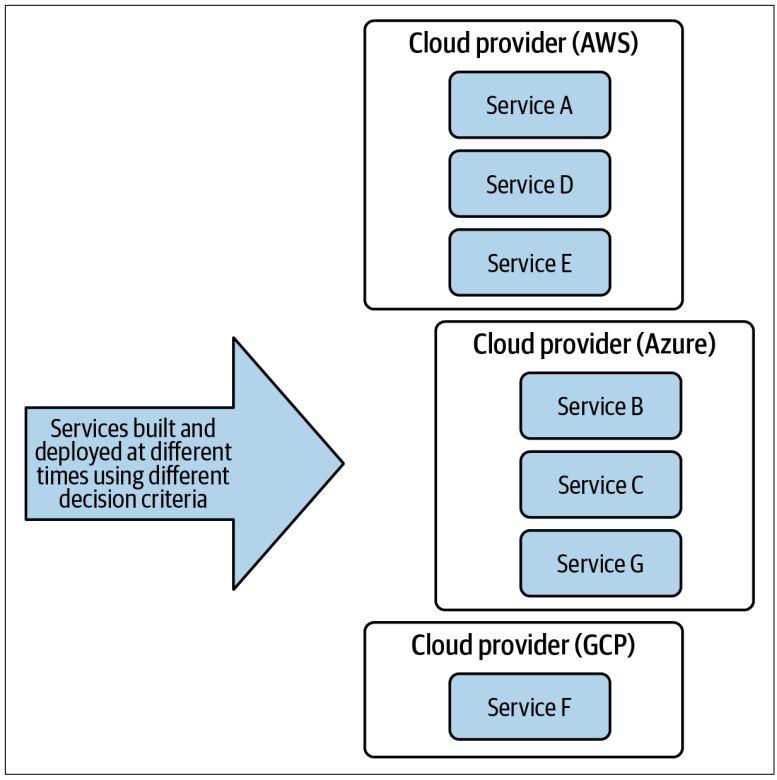

Figure 2-6. Independent deployment decisions made at different times (Service A through Service G)

This process, which leads to a polycloud architecture, was the result of independent decisions based on the criteria that happened to be important at that time. Each independent decision was a good decision at the time, but the result is a patchwork of independent service dependencies in multiple cloud providers.

Now that we've provided some definition and understanding of how your organization may find itself using a polycloud architecture, we'll dive deeper and look at some specific use cases.

CHAPTER 3

Examples of Service-Differentiated Polycloud Architectures

Differentiation based on features and capabilities is an important part of polycloud architectures whether or not the cloud providers were specifically selected for those features and requirements.

What are some cloud provider differentiated features that lead to polycloud architectures? There are many examples that are worth considering. The examples here use different cloud providers for different parts of the application because they provide service characteristics that are better than, or simply different from, other providers:

Cost of storage
: When an application requires significant amounts of storage capabilities, the cost of that storage can be a major driver in overall cloud infrastructure costs. This can lead to storage cost and/or the availability of different classes of storage being important selection criteria.

Cost of computation
: All major cloud providers offer computation, and most of them provide cost-effective options for computation. However, an application's specific computation needs may best be served by one or more specific cloud providers, and that provider may offer cost options that are more competitive for your particular application.

Service architecture specialization
: There are other types of service specializations that are unique to or at least enhanced in one provider over another provider. GCP, for example, has the deepest integration of Kubernetes in its cloud infrastructure. Microsoft Azure has the deepest integration of Microsoft Windows–based servers into its cloud infrastructure. You can use either service with any of the major cloud providers, yet GCP's Kubernetes and Microsoft Azure's Windows Servers have some unique depth to them. As such, a specialty in a particular type of technology may be an important criterion for cloud selection.

Regionalized requirements
: All major cloud providers offer services around the world; however, for a specific location or region, a given cloud provider may offer a deeper set of services or a better location with better performance or other unique geographic advantages. The location an application is called from may very well impact which provider your application should run in, and a large global application deployment may require multiple cloud providers because of unique geographic requirements around the world. This is especially true when you consider applications that require edge locations in many high-traffic locations that provide better performance capabilities for specific locations.

Specialized AI/ML services
: AI and ML are becoming increasingly critical in large and complex modern applications. All major cloud providers offer some form of AI/ML, but the specific capabilities vary dramatically. Application requirements for specific capabilities and performance, or simply preferences in technology and the approach to the technology, can be important selection criteria for a modern application.

In this chapter, we will show examples of each of these types of differentiation.

Storage Requirements: Photograph Management Application

Service architectures make use of polycloud by dividing an application into distinct service needs and having distinct services facilitated using the capabilities of potentially different cloud providers.

Consider a photo management cloud application. This application would have some server-side capabilities making use of computation within a cloud environment. The actual storage of the photographs themselves would be put into a cloud data storage service. Figure 3-1 shows what this application might look like.

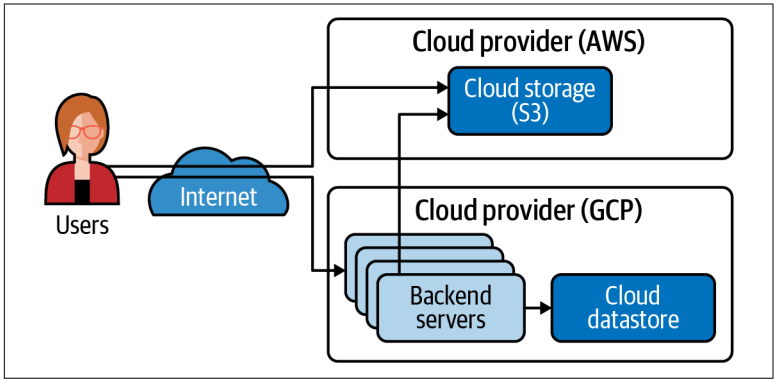

Figure 3-1. Photograph management application using polycloud

There isn't necessarily a strong need for the storage provider of the photographs to be the same as the computation provider. Even considering data transfer costs, it's very possible that the cost of storage itself is a primary driver for this photo management application, and making a distinct decision on who should provide that storage could be an important consideration that leads to a polycloud architecture.

In this diagram, you see the application itself is running on backend servers that are operating in GCP. This application is making use of Google Cloud Datastore. However, for the photographs themselves, which need a large amount of fixed space, you may find that Amazon S3 provides relatively inexpensive high-capacity storage that has fast access to the internet, making it a perfect solution for hosting the actual photographs. So, the application uses Amazon S3 for managing the photographs despite running on GCP.

Specialized Services: Microsoft Software and Applications

There is a simple truth in the world of software developers. Developers are divided into two primary camps: those who prefer to develop and deploy applications on Microsoft technology stacks, and those who prefer not to use Microsoft technology.

Those in the Microsoft camp often will develop applications that are deployed and operated on Microsoft Windows Servers. Those in the non-Microsoft camp tend to prefer developing applications that are deployed and operated on Linux servers.

Most major cloud providers offer servers that run either software environment. However, those who deploy to Microsoft Windows Servers may find that Microsoft Azure provides an experience more in line with the rest of their Microsoft environment.

As such, it is not uncommon for Microsoft technology versus non-Microsoft technology to be a major driver in determining which cloud provider is preferred for a given service.

Some large-scale applications, however, require both Microsoft and non-Microsoft technology to operate the applications. This is most likely because some specific and unique capability of the application requires a Microsoft environment or because consumers of the application require a Microsoft infrastructure for some use cases. We can see that illustrated in Figure 3-2.

In this diagram, the majority of the application is running on Linux servers in AWS. However, parts of the application must be run on Microsoft Windows Servers, and these servers are running in Microsoft Azure, creating a polycloud environment.

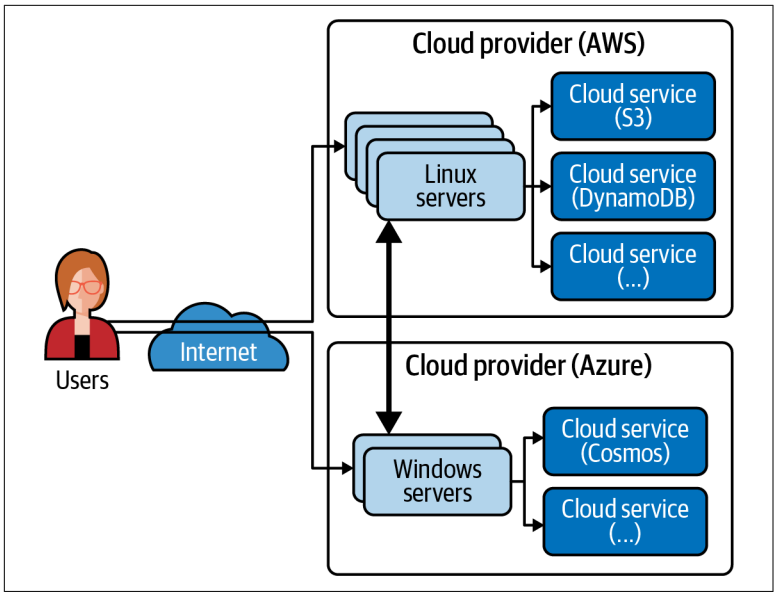

Figure 3-2. Application leveraging some Microsoft Windows Servers

Yes, AWS does offer Microsoft Windows Server instances, but there may be tooling, cost, or other advantages to running the Windows Server instances in a native Microsoft Azure environment.

While less popular, there are other vendor-specific examples of this model, including using Apple macOS Servers on the backend. Additionally, older, large-scale, monolithic mainframe applications still exist and operate in cloud and noncloud environments, often requiring specialized capabilities to connect these large applications to a more modern application infrastructure.

Cost of Compute: High Performance Computation

One value of the cloud is that you can use computation that is focused on very specialized needs, and this computation is available at your fingertips. Besides general-purpose computation instances, you can find instances that are optimized for high performance computation, large memory applications, and I/O performance. Depending on the specific needs of your application, different types of computation are available in very diverse portfolios.

And, as you might expect, different cloud providers have different computation options with different capabilities optimized. The specific requirements you have for a given application may require—or be best optimized for—computation that is available from more than one cloud provider.

Imagine, for example, an application that requires general-purpose "frontend" computers to interact with the end user but requires very high performance computation instances to perform a backend task. You can easily imagine a situation where the high performance instances might be best provided by one vendor (say, for example, AWS), while the general-purpose computation might be best provided by another vendor (say, for example, GCP). This is illustrated in Figure 3-3.

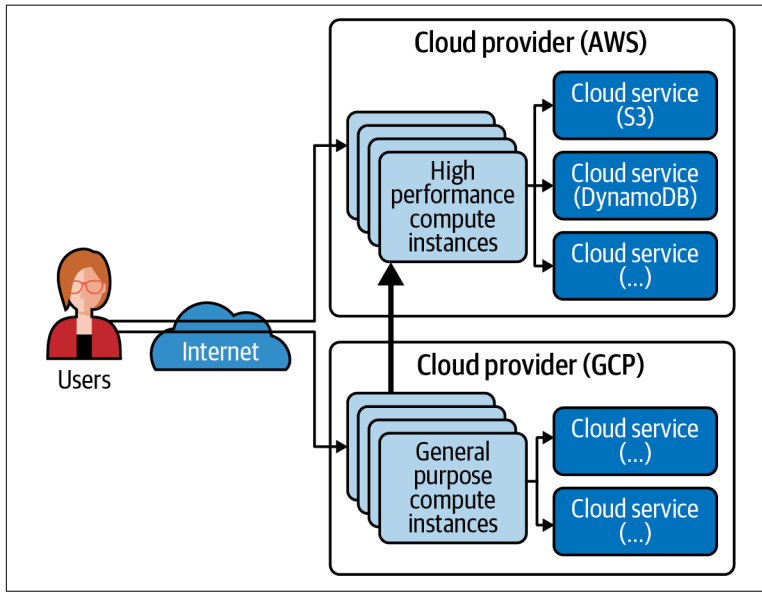

Figure 3-3. Application leveraging specialized computation from different providers

In this diagram, general-purpose computation instances are readily available with the majority of the application running on GCP. Yet high performance computation instances, such as those offered by AWS, may be best suited for the backend portion of the application.

It's not uncommon for computation to be treated as a simple commodity, and thus price becomes the most important selection criteria. Yet computation specialization can also create opportunities for diverse decisions that create a polycloud-architected application.

Regionalization: Edge Data Ingestion

Geography often impacts performance of cloud services. Many cloud applications make use of edge locations located at specific geographic locations in order to provide specific performance-sensitive application capabilities nearer to the end user. These edge locations make use of cloud locations in cities near where the users of the application are located. Some cloud providers provide edge locations that may be in a better geographic location than another cloud provider to serve a specific group of users, and the preferred cloud provider can vary from one location to another. This leads to a polycloud application by distributing parts of the application geographically, making use of multiple cloud providers. This is illustrated in Figure 3-4.

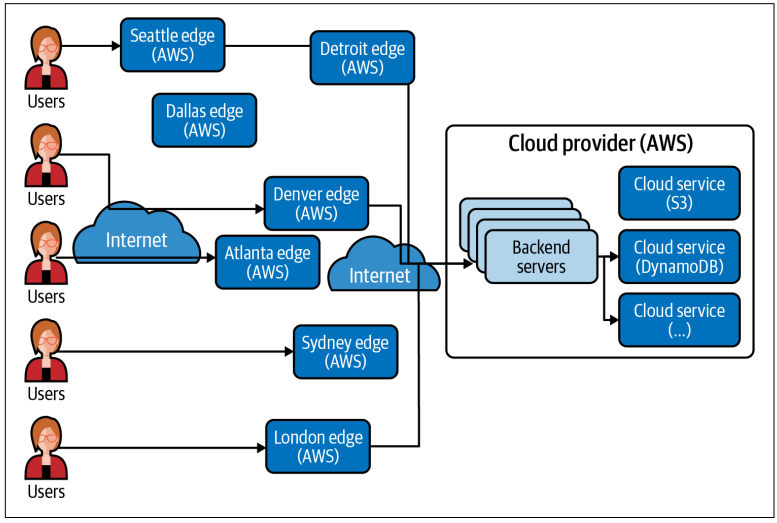

Figure 3-4. Edge locations by polycloud providers

In this diagram, some edge locations are in cities where AWS provides good edge capabilities, and some locations are in cities where GCP and Microsoft Azure provide better edge capabilities. For example, Azure provides more complete overall coverage to

European cities, while AWS provides solid coverage in the Asia-Pacific region, primarily because of how the different cloud providers focused on rollouts in various parts of the world. Meanwhile, Azure's IoT focus gives it a specific advantage in many locations for edge deployments.

When including edge computing in the architecture of an application, as shown in Figure 3-4, while the backend application itself may be monocloud, the application as a whole is polycloud.

AI/ML: Chat Pattern Processing

AI and ML are becoming very important capabilities in most modern applications. Sometimes, sophisticated AI/ML capabilities are required to perform complex data analysis. Sometimes, though, AI/ML is simply needed to look for patterns and trends in simple chat conversations.

Consider the application in Figure 3-5. This application is designed to mostly run on GCP services. However, the application has a chat capability. On the surface, you can imagine this capability would run just fine using generic GCP services. However, it was determined as an early application requirement that the chat conversations needed to be analyzed in order to look for trends, patterns, and even fraud. This is where AI/ML services can be quite useful.

In this application, the developers decided, by using whatever decision criteria were important to them, that Amazon SageMaker was the ML tool they wanted to use. This decision led to the need to have the chat data that was to be processed by SageMaker to be located in Amazon S3, which led to the decision that the chat-processing infrastructure as a whole might work best over on AWS.

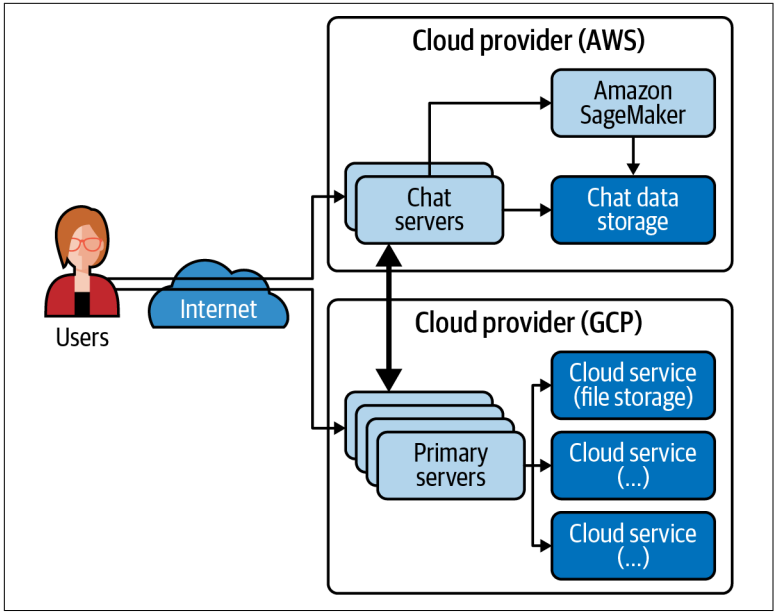

Figure 3-5. AI/ML capabilities offered by specific cloud providers can lead to polycloud

Note the entire application didn't have to be moved over to AWS to support this decision, nor did just the AI/ML processing have to be moved. Instead, a subset of the application was moved over completely to provide best optimization of the cloud services utilized. In this example, the chat capability—all of it—was moved to operate on AWS, while the rest of the application was free to operate on GCP.

This is a strong example of a well-thought-out architecture plan that makes use of polycloud. Decisions were actively made to determine which cloud provider would be best for which parts of the application, and the application architecture was optimized to make use of the specific cloud providers. This polycloud architecture is making effective use of the capabilities of multiple cloud providers.

CHAPTER 4
When Should You Use Polycloud?

While we talked earlier about different paths to polycloud, and some of these paths were planned and others were much more accidental in nature, there is one important question that we should ask: When should you intentionally implement a polycloud architecture for your application?

There is, of course, no easy answer to that question. However, there are a number of questions that you can ask yourself that can help you make the decision as to whether you should implement a polycloud architecture in your application or if you should stick to a monocloud architecture. Answering these questions will make answering the base question much more straightforward:

Tooling costs
: What is involved in maintaining tooling for both the development and operations environments for multiple cloud environments? Does adding a second cloud environment dramatically increase your tooling costs? Have you considered these costs in your overall architectural decision?

Expertise
: Do you have the proper expertise in-house for all the cloud providers you are planning on using? If you are adding support for a new cloud provider, do you need to hire additional expertise? Do you need to add scope to your existing operations training processes? What about development expertise?

Cognitive bandwidth
: Supporting a second cloud provider takes additional effort, in both the development teams and the operations teams. Do you have the team bandwidth it will take to keep up with the ongoing challenges and improvements that each cloud provider requires?

Relationship and trust
: Do you have a proper level of trust for the capabilities of the additional cloud provider? Do you have the appropriate relationships with the additional cloud vendors? What about the additional relationships with incidental and support organizations that provide resources to your company and the new cloud vendors?

Fully loaded costs
: When you are making decisions involving the costs of a particular service, are you considering the fully loaded costs of those services? In this context, *fully loaded* means not just the cost of the service itself, but the cost of the operations, development, training, and management of those services.

Performance
: Are the performance characteristics of the cross-cloud communications channels that will be required sufficient for the needs of the application? Will this remain true as your application scales?

Data bandwidth
: Transferring data into and out of a cloud provider costs money; therefore, transferring data from one cloud provider to another costs money. Have you architected a system that minimizes the need to send large quantities of data between cloud providers? If you are transferring data between providers, have you incorporated the costs, both real costs and performance costs, of those data transfers?

Architecting a polycloud application takes additional architecture knowledge, expertise, experience, and planning beyond the normal cloud application architecture needs. Make sure before you consider an intentional polycloud application architecture decision that you have considered all the architectural impacts. This includes the advantages that are driving you toward this potential strategy, along

with the potential costs and disadvantages, both visible and hidden, of implementing such a strategy.

Future of Polycloud

In the early days of the cloud, AWS ruled most of the cloud-computing world and provided most of the cloud opportunities. As time has gone on, both GCP and Microsoft Azure, along with IBM and other cloud providers, have gained market share and become a larger part of the overall cloud market.

As a result, using multiple cloud providers in simple applications is becoming significantly more practical. Additionally, as competitive pressures have mounted, differentiation in cloud services is increasing and becoming more important. When we last looked, AWS had more than two hundred distinct and unique cloud services, many of them providing a high degree of specialization.

The natural result is that the use of differentiated services will expand, and hence polycloud architectures will become more prevalent. This is normal and good for the industry, and good for consumers of the cloud.

Understanding polycloud—what it looks like, how you get there, and the decisions you make along the way—is a critical aspect of determining your cloud strategy for modern applications.

About the Author

Lee Atchison is a recognized industry thought leader in cloud computing, and the author of the best selling book *Architecting for Scale* (O'Reilly), currently in its second edition. Lee has 34 years of industry experience, including eight years at New Relic and seven years at Amazon.com and AWS, where he led the creation of the company's first software download store, created AWS Elastic Beanstalk, and managed the migration of Amazon's retail platform to a new service-based architecture. Lee has consulted with leading organizations on how to modernize their application architectures and transform their organizations at scale. Lee is an industry expert and is widely quoted in publications such as *InfoWorld*, Diginomica, IT Brief, Programmable Web, CIO Review, and DZone. He has been a featured speaker at events across the globe from London to Sydney, Tokyo to Paris, and all over North America.