A guide to understanding and mitigating switching costs when changing your Cloud Services Provider

Unpicking Vendor Lock-in

Unpicking Vendor Lock-in: A guide to understanding and mitigating switching costs when changing your Cloud Services Provider

Table of Contents

This whitepaper is for historical reference only. Some content might be outdated and some links might not be available.

Unpicking Vendor Lock-in

Publication date: **June 21, 2021** (*Document revisions*)

Abstract

Customers should be able to switch their Cloud Services Provider (CSP) if they want to.

This whitepaper looks at what customers should require from CSPs so they have the freedom to choose the innovative services they need, coupled with the ability to turn things off and move, should they decide to do so. It provides a practical approach to defining, understanding, and eliminating sources of vendor lock-in.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console (sign-in required), you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

A CSP, or vendor, should earn customer business by providing the best services and capabilities at the best price. If a CSP makes it difficult to switch away from them (the essential element of vendor lock-in), it suggests that their services are not earning customer trust through the value they bring, and that they are restricting customer choice.

Amazon Web Services (AWS), provides customers with full control, ownership,and portability of their data, and allows customers to quickly move to another CSP should they choose to. AWS never wants to trap customers with *lock-in* tactics such as fixed-price, mandatory long-term contracts, or technical hurdles to changing CSP that amount to vendor lock-in.

AWS wants customers to stay by choice, because AWS offers the broadest choice of the best cloud services. The AWS outlook is that customers are loyal right up until the moment that somebody else offers them a better service. This drives the AWS customer-obsessed approach to innovation, and ensures AWS earns customer trust on a continuous basis.

The commentary in this whitepaper is based on many years of experience in delivering a secure cloud infrastructure to millions of customers worldwide.

Unpicking vendor lock-in

Customers should have the freedom to choose what they like, when they like. This is a win-win scenario for customers and vendors—as customers have access to the things they need and want, and vendors keep their service offerings relevant and market-leading. That's why constant innovation, and constant improvement is good for the vendor and good for the customer. Many of the new services offered by AWS are created in response to specific customer needs—strengthening the AWS service offering for customers.

AWS wants to solve customer needs every moment of the day; this is imperative, given that AWS cloud services are provided on an on-demand, pay-as-you-go basis. AWS enjoys the challenge of ensuring that customers have the freedom to choose from the broadest range of innovative cloud services, so that customers continue to *want* to use AWS. Should a customer choose to move away, AWS makes this as simple as possible.

Offering customer choice and freedom are core principles throughout AWS. For example, where customers might have been locked into a legacy database provider before, AWS offers a large number of different types of databases, suitable for different types of data. There is no contractual obligation for customers to stay with one type of database. You can run databases from other vendors on AWS, and can change the type of instance your databases run on at any time. AWS makes it easy to manage your data so that you can take advantage of fit-for-purpose database solutions, including exporting your data out of AWS, if desired. Everything AWS does gives customers the freedom to choose the best-fit cloud services and features available. If you find your choices limited in this way, a cloud provider might be trying to lock you in.

Defining "Lock-In" and "Switching Costs"

Avoiding lock-in means that if you decide to move, you can do so, at speed, and without unreasonable difficulty. This does *not* mean that if you decide to move all of your workloads to a different CSP today, you will have everything up and running on them tomorrow. There are always some tradeoffs or *switching costs*, such as time to switch technology stacks, migrate data, or train staff on new vendor services. Switching "costs" include time, flexibility, functionality and sometimes financial cost.

The term lock-in can be misleading from a technology perspective. "Vendor lock-in" is really about switching costs which make it prohibitively difficult or risky to move away from a vendor, rather than being explicitly blocked from doing so.

Switching costs have existed throughout the history of IT. As soon as you decide on a technology vendor, product, or service, you will have switching costs should you decide to change. For example, if you choose Java and then migrate to Node.js, you will have a cost. You're not actually locked in, you simply have switching costs that may be large or small depending on the situation.

Vendor lock-in and data portability are not the only lenses that customers look at cloud services through. Ultimately, you want to benefit from the advantages that cloud services can bring to your organization. In many cases, customers select a cloud provider because they offer capabilities that their competitors do not provide.

Using CSP-native services

An early decision that customers need to consider is: "Am I going to make a commitment to a high-value service from a Cloud Service Provider for the business value that it returns; or am I going to take on the cost and complexity of building it myself (and often conforming to the lowest common denominator) in favor of perceived portability?"

The answer, of course, is that you should use the service which best meets your business needs. There is no need to spend resources to reinvent the wheel, if your needs can be met with a CSP's services, as long as there are no known egregious switching costs.

For example, if you need machine learning (ML) capabilities, choosing a CSP that allows for the use of multiple technology approaches gives you speed and agility, which can save time and money with respect to innovation and responding to changing business needs. You're not locking yourself into the vendor, you're making a decision to use the CSP-native service because it's the best fit for your needs. Trying to use cloud services across multiple providers without the technical justification to do so is likely to lead to greater complexity and less service depth.

AWS has made open source accessible to the widest possible audience, enabling customers to make their own technology choice at any point up the stack. For example, you might choose AWS-native services to manage massive datasets, but you might also choose to drop Apache Spark onto an Amazon Elastic Compute Cloud (Amazon EC2) instance to meet that need. It isn't always a case of having to pick a CSP-native service versus a service that a different CSP offers.

Defining what business outcomes you need to achieve, and doing due diligence of a CSP's offerings, can help you choose the best solutions for your organization's needs. Making everything easy to switch isn't the only consideration—it best to focus on adopting services which are best for your organization.

Remember to factor in switching costs into your evaluation of cloud providers so you don’t get locked in technically or contractually. Build a plan that anticipates the potential need to switch technology in the future, and evaluate vendors on how they help to minimize switching time and cost.

Six lock-in considerations

A CSP should succeed by the breadth, depth, global reach, innovation, agility, reliability, and security of its services. Unless you have access to the broadest range of cloud services, with a CSP that adds new capabilities and services at an accelerating pace, customers risk becoming locked-in to outdated technology. A CSP which is not retaining customer trust through the quality of its services, may tend to lean towards lock-in practices. Following are six lock-in considerations to help you understand what to ask of CSPs, what to avoid, and how to build a plan to switch technology, if desired.

Minimum commitments or long-term contracts

A CSP should not have *mandatory* minimum commitments, or *mandatory* long-term contracts. Long-term contracts should only be offered as a *choice* for the customer's convenience, and should typically come with a compensating feature such as reduced pricing or commitment to through-life improvement to the functionality and/or performance of the service.

Historically, customers have entered into contracts that last years or even decades— with technology often frozen in time for the duration of the contract, inhibiting innovation and experimentation. The key problem when it comes to these commitments or contracts in the cloud, is when they are mandatory. If you want to use cloud as pay-as- you-go, it should be available. Pay-as-you-go pricing with no contractual commitment provides customers with the ability to shut down their environment, export their data and virtual machines (VMs), and walk away without incurring further expense.

However, customers should still feel free to *choose* a longer-term contract, if that fits their business needs best. Customers should keep in mind that a longer-term contract should provide advantages over and above a pay-as-you-go style contract, whether this is reduced pricing, a commitment to in-contract upgrades or some other factor which benefits your organization. A CSP should give you the freedom to choose whichever pricing and contractual model suits them and their workloads best—this way you can adopt the cloud on your own terms.

Licensing

A CSP should provide a range of licensing options for customers. Examples include a marketplace with a curated digital catalog that helps reduce costs by not over- purchasing with an inperpetuity license, multiple database offerings, robust open-source software support, alternative virtualization environments to support hybrid cloud services, and tools to help you assess and reduce license costs.

AWS works with independent software vendors (ISVs) who have permitted the use of their product on AWS, enabling customers to bring their own license (BYOL) and apply it to the product on the AWS Cloud—which helps to reduce switching costs when moving away from the AWS Cloud. Additionally, the AWS Marketplace enables you to find, buy, deploy, and manage third-party software and services.

You can choose a utility pricing model with a support package, which is a pay-as- you-go license in which you do not incur any upfront licensing costs and only pay for the resources you consume. Some ISVs also offer their software as a service and charge a monthly subscription fee. AWS License Manager makes it easier to bring your existing software licenses from vendors such as Microsoft, SAP, Oracle, and IBM to AWS, and centrally manage these licenses across AWS and on-premises environments.

Alternatively, customers can also consider moving to a fully managed, open-sourced solution. For example, instead of paying long-term contracts and support fees for a commercial messaging broker, you can migrate to the fully managed, industry standards-based Amazon MQ service.

Data portability

A CSP should provide portability tools and services that enable customers to move data as needed on and off a CSPs storage *at any time*.

Cloud customers need detailed, clear, and transparent information regarding the processes, technical requirements, timeframes, and charges that will be applied should they want to switch to another provider or port data back to their own IT systems. This should also include the processes and locations of any data back-up, available data formats, the required IT configuration, and minimum network bandwidth. CSPs should also confirm the timeframe during which customer data will remain available for porting upon ending a contract.

The cloud-shared responsibility model is an important factor when it comes to data portability. Some key concepts regarding data ownership and management in the AWS Shared Responsibility Model include:

- Customers own their data.
- Customers have the ability to store content in the format they choose.

- Customers choose the geographic location(s) in which to store their data, and it doesn't move unless the customer decides to move it.
- Customers can download or delete their data whenever they like.

AWS services are built to support both data migration into and out of AWS. Additionally, AWS provides many tools and documented techniques to make it easy to do both. Many CSPs offer several tools to help move data between networks and technology partners. AWS services, for example, are built on numerous open standards like SQL, Linux, and Xen. This flexible foundation enables our customers to securely move information in and out of the cloud regardless of where that information is going, such as cloud-to- cloud or cloud-to-data center.

As a best practice, customers should request that CSPs demo their data portability tools and services as part of a cloud services/CSP evaluation process.

AWS understands that freedom of choice is a fundamental customer need. As mentioned previously, AWS customers always retain ownership and control of their data, including where it is stored, how it is stored, and who has access. AWS has contributed to industry initiatives such as the code of conduct on Switching Cloud Provider and Porting Data (SWIPO), a multi-stakeholder group facilitated by the European Commission, in order to develop voluntary Codes of Conduct for the proper application of the EU Free Flow of Non-Personal Data Regulation / Article 6 "Porting of Data".

Application portability

To minimize the risk of vendor lock-in, applications should be built or migrated to be as flexible and loosely coupled as possible.

Mitigating potential switching costs requires good architecture, standard deployment practices, and pre-planning. Your cloud services should be architected with transience in mind.

There are several ways to improve the portability of your applications:

- Use Docker containers that can be deployable virtually anywhere; build using microservices to reduce the "blast radius" of changes to parts of your application (enabling the testing of each one independently if you need to make changes on a large scale).
- Have loosely coupled services, especially when using a service specific to a CSP—building a façade for each CSP service so you can swap it out as transparently as possible.

- Build your cloud platform on open standards like Xen, SQL, KVM, and Linux.

Clearly demarcating services' interdependencies helps to create and maintain a dynamic back-out or reversibility plan. Depending on the level of risk you perceive, you can make this plan more or less detailed. In this plan, you can itemize what the major switching costs would be if you had to leave a CSP, and what steps you will take to manage them. You can also make a high-level project plan for how you would go about moving, as well as estimate these costs.

Hyperscale CSPs provide standardized services to millions of customers, and these standardized services are used by customers to build their unique cloud environments. A CSP cannot know what a customer environment will look like at any point in time, and a back-out or reversibility plan should not expect this kind of detail from a CSP. Instead, a CSP should provide detailed technical documentation for each of its standardized services, so that customers can use this information to create and maintain their own back-out plans.

In summary, some important application portability considerations are:

- Cloud application components should be loosely linked with the application components that interact with them. You can do this by incorporating REST APIs with popular industry standards like HTTP, JSON, and OAuth to abstract your applications from the underlying proprietary cloud infrastructure.
- Any business logic should be separated from the application logic and be clearly defined and documented. This will avoid the need to decipher business rules in case a migration to a new CSP occurs.
- DevOps tools are being implemented to maximize code portability. Configuration management tools such as Chef and Puppet help you automate the configuration of the infrastructure on which your apps run. This allows you to deploy your application to diverse IT environments, which can reduce the difficulty of moving to a new CSP. These technologies reduce the lock-in risks that can come from proprietary configurations, and can ease the transition from one CSP to another.
- Container technology provided by companies such as Docker, Kubernetes, and CoreOS help isolate software from its environment and abstract dependencies away from the cloud provider. Since most CSPs support standard container formats, it should be easy to transfer your application to a new cloud vendor, if necessary.

Service availability and vendor innovation

In order to justify committing to a CSP, the CSP should be able to demonstrate a proven history of innovation and service availability. If a CSP offers the most innovative, highly available, flexible service on the market, and makes moving away easy and safe; then there's no reason not to choose their service from a vendor lock-in perspective.

Vendor lock-in is an important factor, but it's not the only consideration when adopting new technology. When you move to a CSP, you're not just solely focused on avoiding vendor lock-in. Ultimately, you want to unlock vendor capabilities and the benefits they bring to your organization. A CSP's proven pace of innovation reassures customers that the CSP intends to keep winning their business by offering them more service options, with the knowledge that they can move to another cloud provider if they choose.

AWS gives customers the services and features they want, and avoids locking them in to using services AWS *thinks* they want. The AWS culture of innovation provides customers with the ability to both tap into and influence AWS innovation, and the innovation of third- party services available via the AWS Partner Network (APN).

Unlocking vendor capabilities extends to ensuring that services are available. A CSP can offer peerless services, but if they're not reliable, you'll probably want to move to a different CSP that offers reliable services. Availability and reliability are key benefits of the cloud, and CSPs needs to show SLAs that clearly demonstrate the service commitment and compensation terms if those SLAs are not met. Customers can use AWS's distinct and geographically diverse Regions and Availability Zones (AZs) to protect applications from the failure of a single location. A 2018 IDC study of enterprises that migrated to AWS found that they experienced, on average, 94 percent less downtime as compared to their on-premises environments.

Cost

CSPs should be able to demonstrate a history of ensuring that operational efficiencies flow down to customers in the form of price reductions.

If a CSP increases prices, customers think about moving away. If a CSP has a history of increasing prices, it's likely they will try to make it difficult for their customers to move away by increasing switching costs, often through punitive licensing practices or arbitrary technological barriers.

AWS operates at a massive scale, offering standardized services in a self-service manner. This low-touch model enables us to focus on innovating on behalf of customers to provide more services, enhance services already available, and find operational efficiencies that can be passed back to customers in the form of price cuts. *AWS has reduced prices 107 times*, largely in the absence of competitive pressure to do so. AWS has tried to make it as easy as possible to leave if you want to, because AWS is confident that AWS services speak for themselves, and that service quality will make customers want to stay with AWS.

How the AWS Cloud helps to eliminate lock-in

This section provides examples of AWS features and services that customers can use to keep their data portable and avoid lock-in while keeping switching costs to a minimum.

Data migration

When moving data in or out of the cloud, you need to understand where you are moving your data, the type of data you are moving, and the network resources available— among other considerations. AWS offers a wide variety of services and partner tools to help customers migrate their datasets, whether they are files, databases, machine images, block volumes, or even tape backups.

The suite of data transfer services created by AWS includes many methods that help you migrate your data more effectively. They are broken into two categories:

- **Online data transfer and hybrid cloud storage** —These methods make it simple to create a network link to your VPC, transfer data to AWS, or use Amazon S3 for hybrid cloud storage with your existing on- premises applications. These services can help you both lift and shift large datasets once, as well as help you integrate existing process flows like backup and recovery or continuous data streams directly with cloud storage.
- **Offline data migration to Amazon S3** — One should never underestimate the bandwidth of a semi-truck filled with 100 petabytes of hard drives, or a 100 terabytes suitcase-sized device. These offline data migration services that use shippable, ruggedized devices are ideal for moving large archives, data lakes, or in situations where bandwidth and data volumes cannot pass over your networks within your desired time frame.

Table 1 – Some online AWS data transfer options

If you need an online link to AWS Cloud storage services so you can:	Consider using:
Privately connect your data center with a network link directly to your VPC in an AWS region	AWS Direct Connect

If you need an online link to AWS Cloud storage services so you can:	Consider using:
Copy or replicate file system data into Amazon S3 or Amazon EFS	AWS DataSync
Connect existing on-premises applications to cloud storage • **for files stored as objects in Amazon S3** • **for tape backups** • **for block volumes**	AWS Storage Gateway, File Gateway Tape Gateway, Volume Gateway (or) Partner Products
Migrate databases to AWS quickly and securely, with minimal downtime	AWS Database Migration Service
Migrate running machine images into Amazon EC2 with their data	CloudEndure
Read and write data to Amazon S3 over long geographic distances	Amazon S3 Transfer Acceleration
Collect and ingest multiple streaming data sources	Amazon Data Firehose
Push backups or archives to the cloud with minimal disruption	Technology Partnerships

Table 2 - Some offline AWS data transfer options

If you need an offline way to move data in large batches so you can:	Consider using:
Physically transport petabytes of data in batches to the cloud	AWS Snowball Edge

If you need an offline way to move data in large batches so you can:	Consider using:
Physically transport petabytes of data in an appliance with on-board storage and compute capabilities	AWS Snowball Edge Edge
Migrate tens of petabytes to exabytes of data in batches to the cloud	AWS Snowmobile

Containers

Containers isolate code from the IT environment it's stored in, making it more portable and less disruptive to your existing environment when it's removed. This improvement in portability reduces the switching costs of moving your application to another CSP.

AWS works closely with the open-source community, contributes to hundreds of open-source projects, and builds services that are compatible with open source tools. You can use AWS to run sophisticated open -source software for containers such as Docker, Kubernetes, Mesos, Swarm, Envoy, Linkerd, and Prometheus.

In the Kubernetes community, AWS is an active contributor, leading SIG-AWS and helping to maintain projects that make it easier to run Kuberentes on AWS, such as etcd, Kubernetes CSI, AWS IAM authenticator, CNI plugin, and External DNS.

AWS offers the widest range of services for storing, managing, and running containers which enable customers to choose the right AWS service for their workload, from small experiments to critical production applications.

Table 3 - Some AWS container options

If you want to...	Consider using:	Reason:
Store, encrypt, and manage container images	Amazon ECR	ECR compresses and encrypts your container images, making them fast to start and available to run anywhere.

If you want to...	Consider using:	Reason:
Run containerized applications or build microservices	Amazon ECS	ECS eliminates the need for you to install and operate your own container orchestration software, or manage and scale a cluster of virtual machines.
Manage containers with Kubernetes	Amazon EKS	EKS provisions and manages a conformant, upstream Kubernetes control plane and is integrated with many AWS services to provide scalability and security for your applications.
Run containers without managing servers	AWS Fargate	Fargate scales and manages the infrastructure required to run your containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing.
Run containers with server-level control	Amazon EC2	EC2 virtual machines give you control of your server clusters and provide a broad range of customization options.

VMWare on AWS

VMware on AWS is a native, fully managed VMware environment on the AWS Cloud that can be accessed on an hourly, on-demand basis or by subscription. It includes the same core VMware technologies that customers run in their data centers today including vSphere Hypervisor (ESXi), Virtual SAN (vSAN), and the NSX network virtualization platform to allow them to continue to leverage investments in VMware without continuing to buy and maintain hardware.

VMware Cloud on AWS runs directly on the physical hardware to avoid nested virtualization, while still taking advantage of a host of network and hardware features designed to support our security-first design model. Also, the entire roster of AWS compute, storage, database, analytics, mobile, and IoT services can be directly accessed from your applications.

VMware on AWS can be consumed either on-demand, or over a multi-year subscription, depending on the customer's choice. This ensures that any contractual commitments made by the customer are a conscious choice for the benefit of the customer, rather than an unavoidable switching cost.

DevOps tools to maximize code portability

AWS provides a set of flexible services designed to enable organizations to more rapidly and reliably build and deliver products using AWS and DevOps practices. These services simplify provisioning and managing infrastructure, deploying application code, automating software release processes, and monitoring your application and infrastructure performance.

These tools help build, deploy, and manage services in a decoupled way that makes them easy to move if necessary.

Table 4 - Some AWS DevOps tools to maximize code portability

If you want to:	Consider using:	Because:
Run and manage web apps	AWS Elastic Beanstalk	AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. You can simply upload your code and Elastic Beanstalk automatically handles the

If you want to:	Consider using:	Because:
		deployment, from capacity provisioning, load balancing , auto-scaling to applicati on health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.
Automate deployment	[AWS CodeDeploy]()	AWS CodeDeploy automates code deployments to any instance, including Amazon EC2 instances and on-premis es servers. AWS CodeDeplo y makes it easier for you to rapidly release new features, helps you avoid downtime during application deploymen t, and handles the complexity of updating your applications.

If you want to:	Consider using:	Because:
Unify CI/CD projects	[AWS CodeStar]()	AWS CodeStar enables you to quickly develop, build, and deploy applications on AWS. AWS CodeStar provides a unified user interface, enabling you to easily manage your software development activities in one place. With AWS CodeStar,

If you want to:	Consider using:	Because:
		you can set up your entire continuous delivery toolchain in minutes, allowing you to start releasing code faster.
Use Chef for configuration management	AWS OpsWorks	AWS OpsWorks is a configura tion management service that uses Chef, an automatio n platform that treats server configurations as code. OpsWorks uses Chef to automate how servers are configured, deployed, and managed across your Amazon Elastic Compute Cloud (Amazon EC2) instances or on- premises compute environments. OpsWorks has two offerings, AWS Opsworks for Chef Automate, and AWS OpsWorks Stacks.

If you want to:	Consider using:	Because:
Configuration management	AWS Systems Manager	AWS Systems Manager is a management service that helps you automatically collect software inventory , apply OS patches, create system images, and configure Windows and Linux operating systems. These capabilit ies help you define and track system configurations, prevent drift, and maintain software compliance of your EC2 and on-premises configurations.
Private Git hosting	AWS CodeCommit	AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

Conclusion

A major benefit of the cloud is that it provides customers with the ability move at any time to another vendor. CSPs should not place any impediments on this benefit through making it technically or contractually difficult to switch away from them. Following the recommendations in this paper will help customers ensure that CSPs raise the bar in terms of offering broad service choice, while also keeping switching costs low.

We are confident that AWS cloud services speak for themselves in comparison to other vendors, and that our customer obsession keeps us focused on giving customers the freedom to choose the tools and services they want. This includes the tools and services that allow customers to move away, should they wish.

Contributors

Contributors to this document include:

- Anthony Kelly

Further reading

For additional information, refer to:

- AWS Blog Post: Switching Costs and Lock-in
- AWS Blog Post: Busting the Myth of Vendor Lock-in
- Principles of Fair Software Licensing for Cloud Customers
- AWS Data Migration
- Containers on AWS
- AWS DevOps Services
- IDC Whitepaper, sponsored by AWS, “Fostering Business and Organizational Transformation to Generate Business Value with Amazon Web Services” February 2018

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper reviewed	Content reviewed and verified still accurate - no changes.	February 27, 2023
Initial publication	Whitepaper first published	June 21, 2021

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS's current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS's products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS's responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.