ConversationalGeek

Conversational Key Management in the Cloud

Brien Posey (Microsoft MVP, Commercial Astronaut Candidate)

MINI Edition

Learn about:

- Ensuring apps work with your cloud providers' key management services
- How to use a uniform API for all your apps' cryptography-related functions, for all key stores

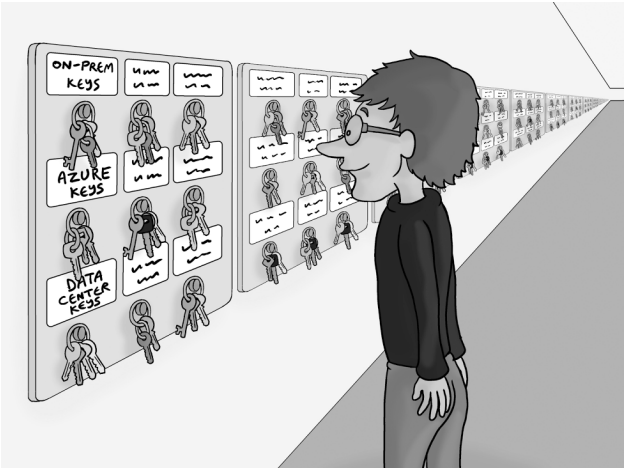
Sponsored by Unbound Security

Unbound Security is the global leader in cryptography and empowers enterprise customers worldwide to confidently secure, manage, and authenticate all critical business transactions, information, identity, and digital assets – anywhere, anytime. Unbound Security CORE is the enterprise platform of choice for secure key management, trusted by many of the world's largest banks and Fortune 500 companies. Unbound Security is a recent recipient of the Deloitte Fast 500 award and is headquartered in New York, with research and development facilities in Tel Aviv.

For more information visit
www.unboundsecurity.com

Conversational Key Management in the Cloud (Mini Edition)

by Brien Posey



ConversationalGeek®

Conversational Key Management in the Cloud (Mini Edition)

Published by Conversational Geek® Inc.

www.ConversationalGeek.com

Trademarks

Warning and Disclaimer

Additional Information

For general information on our other products and services, or how to create a custom Conversational Geek book for your business or organization, please visit our website at www.ConversationalGeek.com.

Publisher Acknowledgments

All of the folks responsible for the creation of this book:

Author:	Brien Posey
Project and Copy Editor:	Pete Roythorne
Content Reviewer(s):	Marcella Arthur
	Lucy Temprano
	Tova Dvorin

Note from the Author

Hi, I'm Brien. For those of you who don't know me (or know my work), I am a long-time Conversational Geek author, and 19-time Microsoft MVP.

In this book, I wanted to write about something that is easy to overlook – key management. Most organizations have a key management system in place, but those systems tend not to work for applications running in the cloud. If you want to migrate an application to the cloud, there is probably going to be some refactoring required in order to make the application work with the cloud provider's key management service. My goal in this book is to talk about some of the related challenges and offer a solution that can make your life easier.

Brien M. Posey

The “Conversational” Method

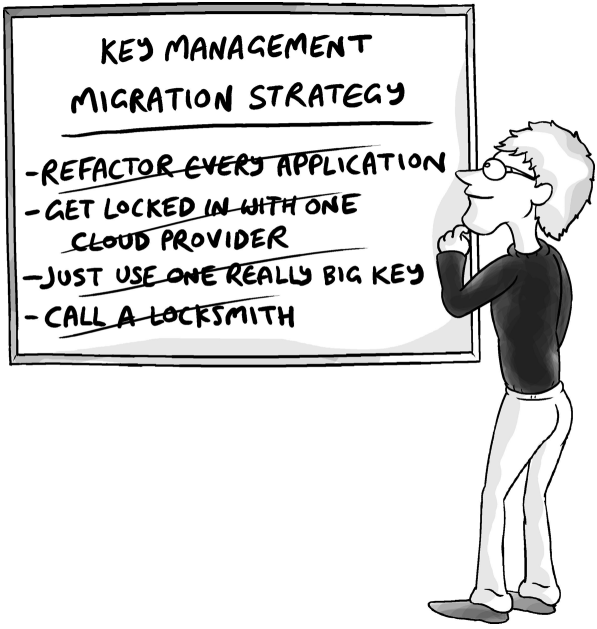
We have two objectives when we create a “Conversational” book. First, to make sure it’s written in a conversational tone so that it’s fun and easy to read. Second, to make sure you, the reader, can immediately take what you read and include it into your own conversations (personal or business-focused) with confidence.

“Geek in the Mirror” Boxes

We infuse humor and insight into our books through both cartoons and light banter from the author. When you see one of these boxes it’s the author stepping outside the dialog to speak directly to you. It might be an anecdote; it might be a personal experience.

Within these boxes I can share just about anything on the subject at hand. Read ’em!

Key Management in the Cloud: The Challenge



Among the many changes brought about by the COVID pandemic is that, for many organizations, it has accelerated migration to the cloud. After all, the

pandemic and the subsequent lock downs have made it difficult to maintain any sort of physical presence in the corporate datacenter. So, migrating applications to the cloud seemed like the obvious solution.

As with so many other things in the world of IT, however, the devil is in the details. It's one thing for an organization to say that it needs to move its line of business applications to the cloud. It's quite another thing to actually do it. Even a simple cloud migration of an application comes with any number of challenges. Some of these challenges are apparent right from the start, but others are easy to overlook until the migration planning is well underway. Key management is one such challenge.

As I'm sure you already know, public and private key pairs are used to provide cryptography for applications and their data. At one time, not all that long ago, key management really wasn't that big of a deal. All of an organization's applications ran in the data center, and an enterprise key management infrastructure was well equipped to provide cryptographic services for all of those applications.

Today things are very different because almost no one's applications are confined to a single data center anymore. Enterprise-class organizations commonly leverage multiple sites and multiple clouds. This makes key management extremely challenging because enterprises lack the ability to manage their entire cryptographic system across all of these various locations. Organizations often have keys scattered across multiple locations, and it can be really difficult to figure out where the keys are located and how they are being used.

The real problem lies in the fact that each of these locations likely has its own cryptographic service provider. Perhaps more importantly, there is no consistency from one location to the next. The key management system used in the Amazon cloud, for example, is different from the one used in the Microsoft Azure cloud.

So, consider what this means for an organization that has a line of business application running in its own datacenter, and who wants to migrate that application to the cloud. The application is presumably already using a cryptographic service provider and keys that were issued by that provider.

If you were to simply move that application to the cloud without making any changes, the application would attempt to leverage a cryptographic service provider that does not exist in its new location. In other words, the application would not work because of a broken dependency on an external service.

The good news is that all of the big cloud providers offer key management services in the cloud. The bad news is that making the application work in the cloud probably is not going to be as simple as releasing the previously used keys and using a new set of keys that have been issued by the cloud service provider. More than likely, the application is going to have to be refactored to make it work with the cloud provider's key management system.

Refactoring refers to having a developer rewrite a portion of the application's source code. This process tends to be both expensive and time consuming.

On the surface, the need for refactoring an application to make it work in a cloud environment might not seem that big of a deal. Yes, there is a certain amount of effort involved in refactoring an application, but it's a one-time task. Once it's done it's done. Besides, when developers re-factor one application to make it work with a cloud provider's key management system, they gain knowledge that they can use in the future. Subsequent modifications to other applications might not be as time-consuming because the developers already know how to make the required changes.

Unfortunately, this logic does not hold up when you look at the bigger picture. Most enterprise-class organizations strive to be cloud agnostic. These organizations use a variety of public cloud providers and place workloads based on which cloud is going to do the best job at the lowest cost. As previously mentioned however, each public cloud provider has its own unique way of dealing with cryptographic keys. So, with that said, let's go back to the example from a moment ago.

Suppose that an organization has a line of business application that it wants to migrate to the Amazon cloud. The organization's developers re-factor the application, and then the application is migrated. Suppose that later on, however, evolving business needs demand that the application be migrated to a competing cloud such as Microsoft Azure. In order to migrate the application, it is going to have to be re-factored again, but in a completely different way. Herein lies the problem.

It probably doesn't make a lot of sense for organizations to re-factor applications in a way that would allow them to work with a multitude of different cloud providers. Doing so would be a very complex and time-consuming undertaking for the development staff. Besides, any one of the cloud providers could change the way that it does things at any given time, necessitating the need for the applications to be re-factored yet again. Remember, the applications are hard-coded to work with a very specific key management architecture.

On the other hand, simply refactoring applications to work with a single cloud probably isn't the best option either. The risk here is that developers

become really comfortable with one particular cloud and so the organization begins to suffer from cloud service provider lock-in, because all of its re-factored applications are designed to work with that one single cloud service provider. This would mean that the organization would effectively lose all of the advantages that led it to adopt a multi-cloud strategy in the first place.

In the past, there was no easy solution to this problem. In fact, cryptography related challenges are one of the big reasons why some applications have remained on premises. Fortunately, there is a better way for an organization to accomplish its goals. Before I explain what this new approach looks like, let's take a step back and examine what a best-case situation would look like.

As previously mentioned, most larger organizations have adopted a multi-cloud strategy. The ultimate goal for most organizations is to be truly cloud agnostic. This means running a service or an application on whichever cloud provider offers the most compelling business case for doing so. At the same time though, cloud providers evolve, and so organizations should ideally be able to seamlessly

move applications from one cloud provider to another on an as-needed basis.

The big question is what would it take to actually be able to accomplish these goals? When you consider the differences in the way that the various cloud service providers handle cryptography, making applications truly cloud agnostic would seem like an impossible (or at least cost-prohibitive) task. However, the solution is easier than you might think.

Rather than trying to refactor your applications to make them work with a variety of cloud services (any of which could change at any time), why not use an intermediary? Instead of building the application to interface directly with a cloud provider's key management system, you could design the applications to communicate instead with an intermediary service by using some sort of standard protocol such as a RESTful API or one of the standard cryptographic libraries commonly in use today.

The intermediary service's job is to handle all of the backend intricacies involved in interacting with the individual cloud-based key management systems on

your behalf. There are three distinct advantages to using this approach.

The first of these advantages is that it greatly simplifies the application refactoring process. Developers can use a single API, regardless of which cloud will ultimately end up hosting an application. This API can be used for all of an organization's applications and can significantly decrease the amount of time and effort involved in cloud migrations.

The second advantage is that this method is future proof. If a cloud service provider changes the way that its key management service works, then it is the intermediary, not you, that will have to adapt to the change.

The third advantage is that using an intermediary service can help an organization to become truly cloud agnostic. An organization can migrate its application off of one cloud and on to another without having to refactor the application.

The Big Takeaways

One of the biggest challenges associated with migrating an application to the cloud is that of making the application work with the cloud provider's key management service.

Doing so normally requires a significant amount of development effort. It also means that the application will only work with one specific cloud provider. It is possible however, to configure applications to relay cryptography-related functions through an intermediary.

This allows for the use of a standard API, and the process works in exactly the same way regardless of which cloud an application is being hosted on.

NOTES

NOTES

Cloud Economy vs. Cloud Economy.

Always First Class to the Cloud with **Unbound.**

unbound security

unboundsecurity.com

For many organizations the pandemic has accelerated cloud adoption. However, even a simple application migration can come with a number of challenges that are easy to overlook. Key management is one. This ebook will help you ensure you can effectively manage cryptographic keys in the cloud.

About Brien Posey

Brien Posey is a 19-time Microsoft MVP, a published author and conference speaker with 20+ years of IT experience, and a Commercial Scientist Astronaut candidate.