PCI COMPLIANCE REVIEW

MICROSOFT AZURE AUTOMATED FOUNDATIONAL ARCHITECTURE FOR PCI DSS-COMPLIANT ENVIRONMENTS v1.0

REVIEW AND GUIDANCE FOR IMPLEMENTATION

DAN STOCKER | CISSP, CISM, QSA

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | Coalfire.com

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This PCI Compliance Review examines the Microsoft Azure PCI Foundational Architecture (v1.0) solution that was joint developed with Avyan Consulting Corp (Microsoft MVP partner) and Microsoft. As Azure's PCI assessor, Coalfire Systems, Inc. (Coalfire) was consulted to evaluate the proposed architecture and compose guidance for Azure customers who will use it, in whole or in part, to build a PCI compliant environment.

The foundational architecture provides the steps necessary to stand up Azure resources in a manner that supports accepting credit card payments. This PCI Compliance Review outlines the topics necessary to build on the foundational architecture toward a full PCI-compliant business. These include payment processing, change management, and vulnerability management. Being aware of these dimensions in advance of further solution design will yield better results for an eventual PCI assessment.

It is Coalfire's judgment that the foundational architecture establishes a solid base for Azure customers to build upon, with the ultimate aim of attaining PCI compliance. Azure services have been designed to address the fundamental networking, logical access, and data protection controls in the PCI Data Security Standard (DSS). Avyan Consulting has packaged these with useful implementation guidance.

Azure customers leveraging this foundational architecture should also note that PCI compliance incudes more than technical controls. This paper also outlines necessary compliance topics that are wholly outside of Microsoft Azure, including scoping, policies and procedures, governance, and business as usual PCI program management. These will scale with the complexity of the business.

INTRODUCTION

The secret to getting ahead, is getting started. The secret of getting started is breaking down your complex overwhelming tasks into small manageable tasks, and then starting on the first one. – Mark Twain

Moving to "The Cloud" can be an intimidating prospect, if it is outside one's experience. Adding concerns around credit card payment security and compliance can make it seem overwhelming. Microsoft Azure and Avyan Consulting have collaborated on a foundational architecture for Azure customers, that is built with Payment Card Industry (PCI) compliance in mind. It is packaged with implementation guidance and helpful assistance for working with the many controls that Azure offers customers.

This PCI Compliance Review will evaluate the foundational architecture for key elements of PCI compliance. After initial implementation, there are further considerations that are relevant for businesses who are aiming at PCI compliance. These are outlined in the Considerations for Build Out section.

The architecture covers network design to meet PCI requirements for defense in depth, including a DMZ (implemented with Application Gateway and Application Service Envrionments) and internal Azure SQL zone. Vulnerability management requirements are met with a collection of Azure-native tools covering anti-malware, file-integrity, web-application firewall, intrusion defense and robust logging, monitoring and alerting (via OMS and Security Center).

Lastly, there are aspects of PCI compliance that are not covered by the foundational architecture, but will be useful in compliance planning. These are outlined in the Supplemental Considerations section. For organizations new to PCI, Coalfire recommends the Glossary of Common PCI Terms, at the end of this PCI Compliance Review.

Coalfire would like to thank Microsoft Azure and Avyan Consulting for the chance to evaluate this foundational architecture, and their vision in creating it. Coalfire would also like to emphasize that this evaluation is based on v1.0 of the PCI Foundational Architecture, which is intended to be updated over time. The current notes for each new release should be consulted carefully.

Microsoft Azure publishes the current Attestation of Compliance (AOC) and Responsibility Summary, and makes those available in the Microsoft Trust Center.

FOUNDATIONAL ARCHITECTURE REVIEW

INTENTION AND PURPOSES

The PCI Foundational Architecture was built to illustrate a PCI compliant foundation for a business using Azure services. What does that mean? Azure services have been composed to build the skeleton of a business that accepts credit card payments, in a PCI-compliant manner.

Any discussion of PCI compliance must start with scope, as that will determine what we are evaluating. PCI scope starts with any systems, physical areas, or processes that store, process or transmit cardholder data. That is known as the Cardholder Data Environment (CDE). Additionally, any systems that have a material impact on the security of the CDE are also in scope, to that degree.

The full foundational architecture is described below, broken out into topics that PCI is concerned with. Each topic is described, along with how the architecture addresses it. Readers should keep in mind that the foundational architecture is not a full solution. Several fundamental elements will be needed before credit card payments can be accepted. See Considerations for Build Out section. Those considerations do not detract from a well-constructed architecture that illustrates how Azure services can be leveraged for useful work in a PCI-compliant manner.

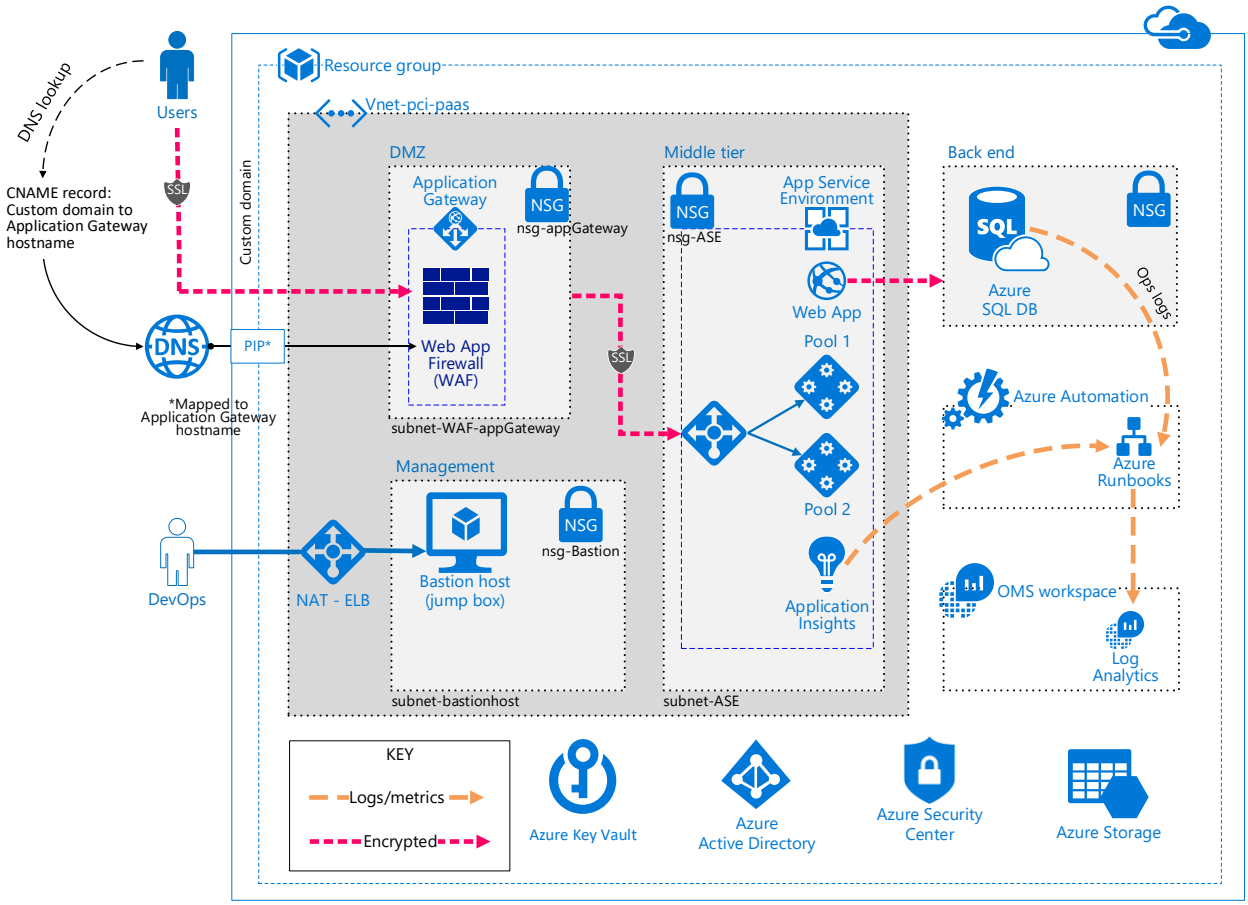


FIGURE 1: NETWORK DIAGRAM

NETWORK

Part of PCI compliance includes maintaining documentation of the Cardholder Data Environment (CDE). Figure 1 illustrates the initial diagram for the PCI Foundational Architecture. It is a useful guide to discuss networking elements of PCI compliance, from the outside inward.

Perimeter

Security at the perimeter is a best practice, as well as a PCI requirement. The Appplication Gateway service is a key layer of defense for the web application. In addition to offering a managed interface, it has the ability to act as a web application firewall. Since traffic must be inspected for threats, SSL is offloaded here.

A bastion host is also part of the perimeter, since it receives incoming connection for administrative and deployment purposes. That host will require hardening, which is discussed below.

DMZ

A DMZ is both a best practice, and a PCI requirement for applilcations receiving connections from the internet. In the foundational architecture, it is implemented using Azure Application Gateway and ASE. This packaging of several *a la carte* services offers integration benefits. A load balancer (ILB) is included, and networking security group (NSG) controls are available to limit traffic to and from the Azure SQL database (and the bastion). These would not be possible with a generic Web Application.

Internal Zones

PCI expects that databases with cardholder data be further isolated from the internet than just a DMZ. The foundational architecture uses the Azure SQL service, on a separate subnet. Azure SQL Firewall rules whitelist connections from the ASE web application.

Several other Azure services are used in the foundational architecture. Some are depicted in the diagram. Operations Management Suite (OMS) supports logging and monitoring. Connections to it are managed with Azure Automation runbooks (depicted). Key Vault is used to manage encryption keys used to secure the cardholder data in the Azure SQL database. Lastly, the Azure Security Center service is used to satisfy the intrusion defense requirement, and enhance monitoring and alerting.

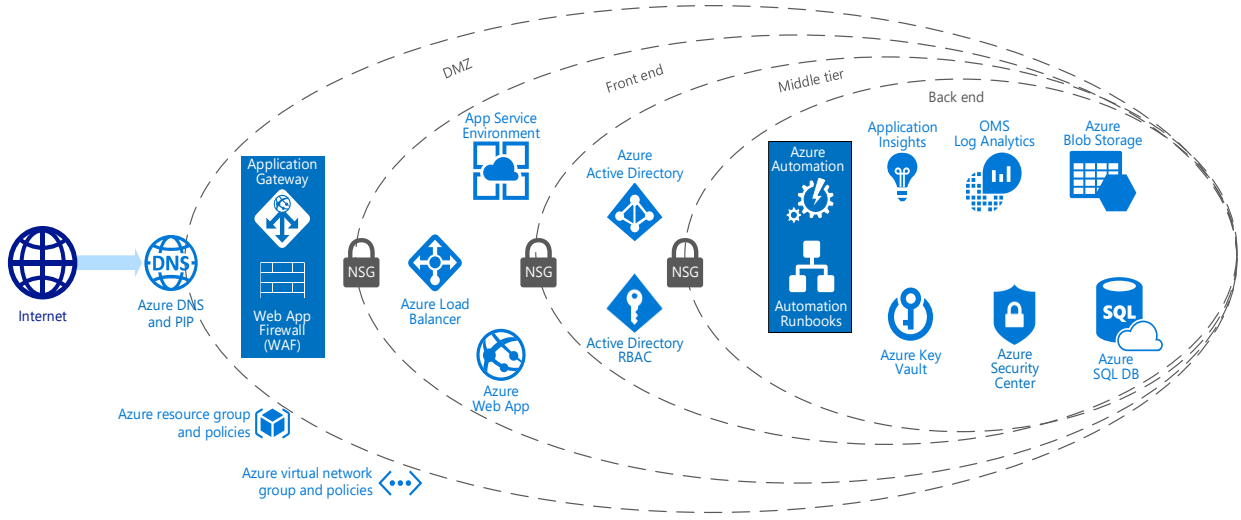


FIGURE 2: NETWORK DESIGN

DATABASE

The foundational architecture has chosen to use the Azure SQL service, instead of standing up a separate SQL Server on an IaaS instance. This is best practice for a foundational architecture that is intended to illustrate principles for solution building in Azure. Most organizations will benefit from reduced complexity.

Several aspects of the architecture deserve special mention:

- *Role-based Access Control (RBAC) with Azure Active Directory (AAD):* this is standard for Azure SQL.
- *Azure SQL Firewall:* in Azure currently, this is a crucial networking control to protect the database from attack, and to implement whitelisting for connections from the web application in ASE.
- *Data Masking:* full credit card numbers should only be viewable by personnel with a job-related need. This is discussed further in the next section. Masking of the full card number will enable other personnel to provide support functions without concern over security of the full Primary Account Number (PAN). The masking function is implemented in Blueprint code, and based on the role of the user receiving the data.
- *Transparent Data Encryption with column-level encryption:* Azure SQL enables businesses to set up an automatic column-level encryption of the full PAN, expriation, and card verification value

columns. The encryption keys used for this process are managed by Key Vault and used automatically by Azure SQL.

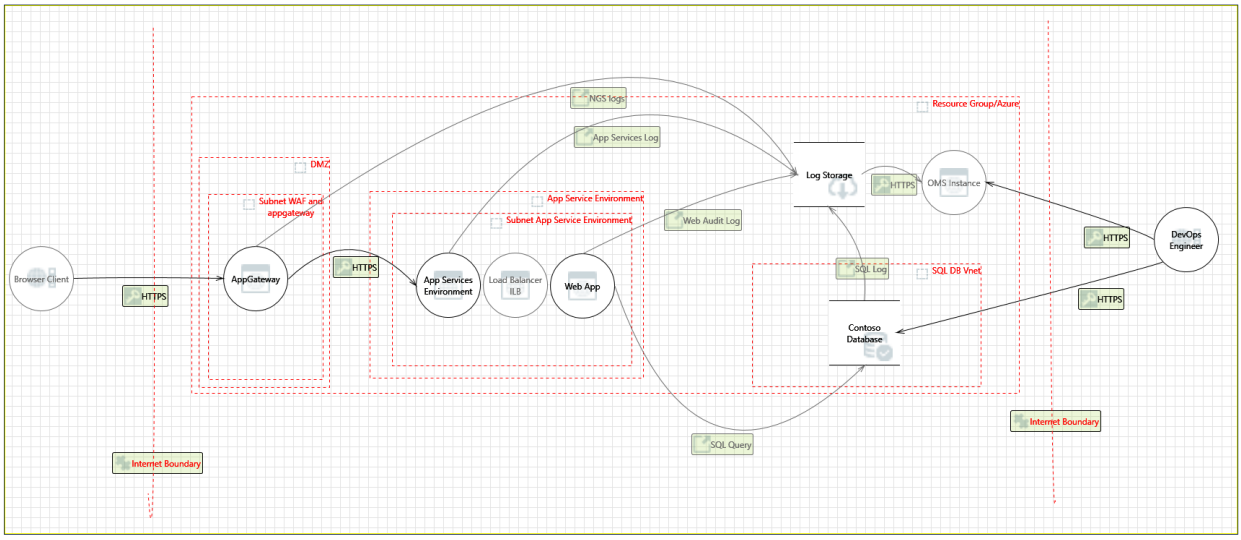


FIGURE 3: DATA FLOW DIAGRAM

LOGICAL ACCESS

Microsoft's Active Directory is an industry standard for managing logical access and authorization. The Azure-native service, Azure Active Directory (AAD), has a multitude of useful hooks for easing the management of a cloud deployment. This foundational architecture addresses the need for logical access using identities for both people (users) and processes (services).

There are several guidelines for how logical access should be managed for PCI compliance:

- *Users:* only those personnel who will be administering the solution, or deploying new software, should be provisioned with access.
- *Role-based Access:* permissions should be granted based on role, not identity. For example, Database Administrator (DBA) is a role that one or more people may be assigned. DBA permissions are assigned to the role, not users who serve in those roles.
- *Groups implement Role:* to assign a role to a user, they should be added to a group that represents the role. This process supports effective User Access Reviews, where group membership can be reviewed for continued authorization.

VULNERABILITY MANAGEMENT

MS Anti-malware (Bastion)

The Azure Anti-malware extension is employed for the Bastion host. This extension will provide protection from malware, including viruses, trojans, rootkits and worms. Best of all, it is supported and kept up-to-date by Azure.

Web Application Firewall

PCI mandates that web applications be either protected by a web-application firewall (WAF) or assessed for vulnerabilities before being deployed to production. The foundational architecture offers WAF coverage that is part of Application Gateway. The Application Gateway allows a business to meet multiple PCI requirements in one step, including: disabling SSLv3 and early TLS, and dynamic protection for OWASP 3.0 vulnerabilities.

Log Monitoring and Alerting

Operations Management Suite (OMS) has been chosen as the Azure solution for logging, monitoring and alerting. While all elements of the solution must emit logs (including yet-to-be built components), the aggregation of those logs is a specific PCI requirement. Centralizing logs enables protecting them from alteration, and preserving them for (potential) forensic value in the event of an incident.

Additionally, Azure Security Center offers the ability to augment OMS functionality for monitoring and alerting. See the Considerations for Build Out section, below, for that discussion.

Hardening via Azure VM images

The Azure marketplace offers many options for virtual machine (VM) images. In the base foundational architecture, only a single IaaS instance will need a VM. That image is fairly generic, but will still need to be hardened by changing default configurations for:

- *Users:* all logical access should be managed via Azure Active Directory (AAD).
- *Networking:* access should be limited to known hosts within the under business. Coalfire recommends that personnel who work remotely use a Virtual Private Network (VPN) connection into the business before connecting to the bastion host. Only actively used ports and protocols should be enabled.
- *Software:* the standard theme of less is more applies. Only software necessary for the anticipated uses of the bastion host should be installed.
- Smaller businesses may be tempted to install other software, or use the bastion host for other computing purposes, but this would violate the PCI requirement that components in the CDE have a single purpose.

FIM (Change Tracking via OMS agent)

One benefit of using OMS is the availability of Change Tracking to detect when key files and configurations have changed. PCI expects two classes of files to be monitored for changes: system settings and files (i.e. parts of the operating system), and critical files (usually parts of in-scope applications or processes). OMS Change Tracking has the capability to track both classes.

IDS (Security Center)

Security Center[1] has broad capabilities for visualizing and alerting on the security posture of Azure resources. It is backed by Microsoft's Global Threat Intelligence. This provides a business the ability to leverage that expertise with modest effort. In the PCI world, that effort maps primarily to the requirement for Intrusion Detection, which generates alerts leading to incident management.

ENCRYPTION

Azure Key Vault is used for protection of cardholder data at rest, in the Azure SQL database, using column encryption. Use of Key Vault is a good choice, as the key management is simplified (managed automatically between Key Vault and Azure SQL) by avoiding the effort of managing key-encrypting keys (KEKs). This is

advantageous for security reasons as well, and can allow implementation of least privilege for roles that will need to view full Primary Account Numbers (PAN) (using Dynamic Data Masking).

PCI expects encryption keys used to protect cardholder data be changed periodically (a process known as "rotation"). The keys used to protect the card number, expiration, and card verification value columns in Azure SQL will need to be rotated, via Key Vault, as part of the key management procedures of the business. Keys used for BitLocker should not need to be rotated, unless there are files used to store PAN. See the Middle-Tier Processing discussion below, under Considerations for Build Out).

CUSTOMER RESPONSIBILITY MODEL

Overview

Given the complexity of PCI compliance, the foundational architecture is accompanied by a Responsibility Model that guides adopters on how elements of the architecture map to PCI requirements. Since most of the architecture is implemented with Azure PaaS services, and those tend to be more challenging for businesses to conceptualize their compliance obligations, the Responsibility Model is particularly helpful.

	IaaS	PaaS	SaaS
App Configuration	Azure Customer	Azure Customer	Azure Customer
Application	Azure Customer	Azure Customer	Azure
Platform	Azure Customer	Azure	Azure
OS	Azure Customer	Azure	Azure
Network	Azure	Azure	Azure

FIGURE 3: RESPONSIBILITIES PER CLOUD MODEL

Coalfire recommends a helpful general rule for understanding how PaaS services should be assessed: if you have control over a setting, you must answer for its compliance. This informs the Azure Responsibility Summary (for all Azure services) and is applicable here as further guidance. The table above illustrates the idea of responsibility for the "stack" of foundational elements used in the delivery of cloud services. A proactive PCI Compliance program will return to this idea regularly, when managing compliance between assessments, and when considering the impact of changes or new architecture.

Key Responsibility Notes

Non-technical PCI requirements are wholly the responsibility of the business. These include administrative, change management, vendor management, access control management, governance, human resources, and risk business functions. Most PCI requirements have both policy/procedure and technical control expectations.

Microsoft Azure has prepared a Responsibility Summary for the use of this foundational architecture. It is a supplemented version of the Microsoft Azure Responsibility Summary, with notes on the shared responsibility for elements of the foundational architecture. The Blueprint itself has several key points about responsibility for adopters, including business responsibility for setting up a valid SSL certificate, and the need to actively manage logical access (especially passwords).

All these sources should be carefully considered by any business which builds on this foundational architecture. The following per-requirement notes summarize key points of Azure customer responsibility:

Requirement 1: As the business builds on the foundational architecture, attention should be paid to the networking configuration, to maintain segmentation of DMZ, database and middle tier. Regular review of those Azure networking rules should confirm that the allowed connections are authorized. Devices used for

administration of the cardholder data environment should be configured in accordance with requirement 1.4.

Requirement 2: The business should adopt and manage a hardening standard. Hardening should be applied to all IaaS instances, including the bastion host. Any in-scope wireless networks (outside Azure) will need active management. All components of the cardholder data environment should be inventoried and tracked.

Requirement 3: If the business collects Sensitive Authentication Data (SAD), a process must ensure that it is not retained after authorization. As the business builds out the payment collection and processing functions, attention should be paid to ensuring that Primary Account Number (PAN) is not readable by personnel who do not need access. Any storage of cardholder data (other than the architecture's use of Azure SQL) must ensure that PAN is unreadable. Encryption processes (including the Blueprint's design for Azure SQL) will require key management procedures.

Requirement 4: As noted in the Payment Processing solution, the business is responsible for securing and managing a valid certificate for use of HTTPS on the website. Any in-scope wireless networks (outside Azure) will need active management.

Requirement 5: Anti-malware on the bastion host is managed by Microsoft Azure. Any additional IaaS instances should be configured similarly.

Requirement 6: Most of requirement 6 will be managed by the business. These topics include adopting and maintaining a vulnerability management standard, change management, and secure software development and testing. Any IaaS instances (including the bastion host) will need patching on a regular schedule. Only the Web Application Firewall (provided by the Application Gateway) is managed by Microsoft Azure.

Requirement 7: The business should organize authorization using Role-based Access Control (RBAC), where permissions and capabilities are assigned to a role, and roles are then assigned to the personnel performing the role. The least privilege principle should be used to limit roles to just what they need to accomplish their remit.

Requirement 8: The PCI standard has a number of prescriptive requirements regarding authentication (e.g. password complexity, inactive session timeouts, etc.) These will require some attention to ensure proper configuration, using Azure Active Directory (AAD).

Requirement 9: The compliance of any media with cardholder data outside Azure (hard copy, other digital data stores) must be managed by the business. If the business also has card-present payment channel integration with point-of-interaction (POI) devices, they should address requirement 9.9 for management and periodic inspection of those devices.

Requirement 10: Finalize the design for centralization and management (including retention) of logs; responsibility for monitoring and alerting on logging.

Requirement 11: This Payment Processing solution has been pentested by the Azure team, as is, with no adverse findings. Quarterly internal vulnerability scans should be performed against IaaS instances and internal APIs delivered from PaaS-hosted services. Quarterly external scans, by an Approved Scan Vendor (ASV), must be performed against any external-facing endpoints (websites, APIs, etc.). Azure Security Center and OMS were noted above as useful tools for intrusion detection (IDS) and file integrity monitoring (FIM).

Requirement 12: Primarily non-technical controls, but does expect a technical control on how long a remote session may idle, before being terminated.

Readers may note that, in general, IaaS as an architectural choice will lead to greater compliance efforts, as the responsibility for management is more heavily borne by the business. PaaS services offer a balance of Azure-managed compliance and ample choice for building the middle-tier.

CONSIDERATIONS FOR BUILD OUT

SCOPING / CDE CONSIDERATIONS

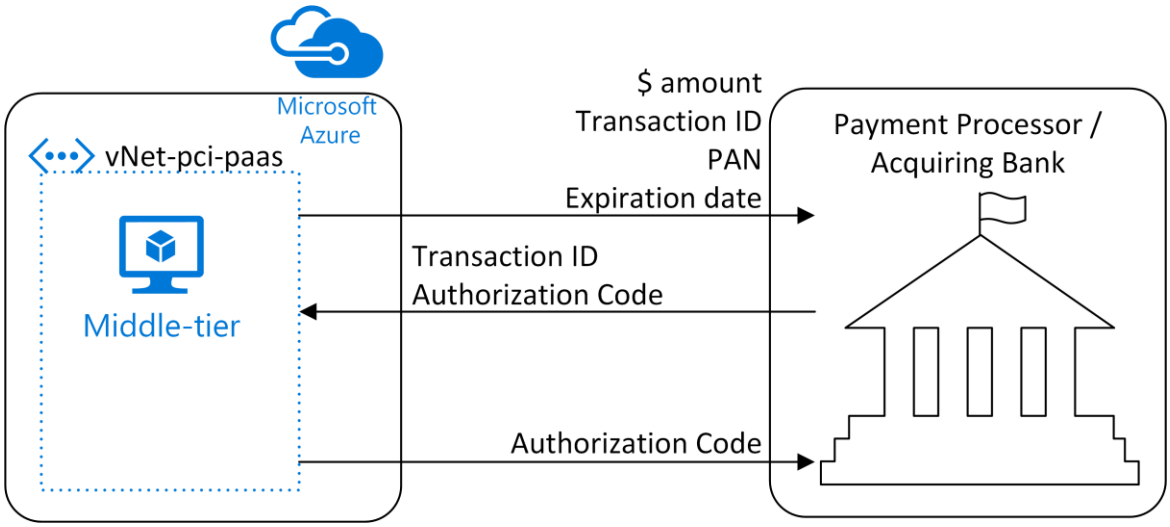
The Azure PCI Foundational architecture does not, by itself, stand up a business for accepting credit card payments. Businesses who start here should be mindful that the scope of PCI assessment starts with the Cardholder Data Environment (CDE), and includes anything that has a material impact on the security of the CDE. This process will fall into one of the following categories:

- *Azure Only:* the entire CDE will be implemented in Azure. This is most likely for either a new business, or an existing business that can partition their operations. A clean separation of Azure-based CDE and other areas of the business will reduce scoping complexity.
- *Hybrid:* the CDE will include the Azure footprint and other elements (e.g. physical locations, mail and telephone ordering, or other computing facilities (datacenters, clouds, Software as a Service). This is most likely where the foundational architecture will be used to move a payment channel into the cloud. There may be other aspects of the business that are relevant for PCI compliance.

In the Hybrid case, any interconnections between the Azure footprint and the other in-scope areas must be designed for PCI compliance. Those will implicate network security, logical access, and data protection requirements, at a minimum, and non-Azure technical controls.

APPLICATION AND DATA MODEL

As constructed, the reference architecture doesn't process credit card payments. Once the data is collected, there needs to be a transaction with a payment processor. That transaction can be modeled simply as:



1. Send amount to be charged, transaction ID, PAN, expiration date (and optionally, Security Code) to the processor.
2. Processor resolves the payment with the appropriate counterparties, returning the transaction ID and an Authorization Code.

3. At settlement time, that Authorization Code can be used to settle the funds for the transaction.

None of this code exists in the foundational architecture, partly because the details can vary widely, depending on who is involved. One decision a business will make when building out this function will be whether or not to collect Sensitive Authentication Data (SAD). This category of cardholder data includes the Security Code (3-4 digit number printed on the front/back of the card), and the details encoded in the magnetic strip on the back.

Whether a business should collect SAD will depend, in part, on the relationship with their payment processor. In cases where it is entirely up to the business, the special compliance implications of SAD should be considered in the solution architecture. In particular, it may not be stored after the authorization is obtained, and it must be securely deleted.

The data flows for accepting and processing payments should be designed to minimize the length of time that SAD is stored before authorization is obtained. Unless further work is done to model the risks, any SAD collected should only be stored in the Azure SQL database. Happily, Azure SQL's default `delete` command is implemented in a way that is compliant with this requirement. In general, Coalfire recommends that unless SAD is required, it should not be collected.

All custom applications (including the above payment processing functions) must emit logs that are centralized and monitored in the same way that elements in the foundational architecture do. See the Logging, Monitoring and Alerting section, below.

MIDDLE-TIER PROCESSING

The payment processing function (described above) does not have a placeholder element in the foundational architecture. How a business chooses to implement that (and other necessary functions) will depend on a number of variables. Coalfire recommends a further network segment for middle-tier processing, and network security groups (NSGs) that limit connections to and from that segment.

One particular value of the foundational architecture is the process of how to stand up an IaaS instance in a PCI environment. The Bastion host, that is intended to support administrative and deployment operations, is configured with best practices for BitLocker, anti-malware and OMS support, using a machine image from Azure. This baseline is a great start for any supplemental systems that support middle-tier processing (including payment processing). Of course, other PaaS services (including further ASE instances) could also be used, and have noted advantages for easier management.

LOGGING, MONITORING AND ALERTING

All components in the CDE are expected to emit logs, that are centralized for monitoring and alerting. This particularly includes custom applications that implement (at a minimum) the payment processing function. These logs should be relayed to OMS via either the Azure Automation already set up in the foundational architecture, or via another instance of Application Insights set up for new ASE or PaaS services. Those logs will require some customization of the monitoring and alerting in OMS and Security Center to capture salient aspects of the custom code, and ensure that security considerations are covered.

All logs must be retained for at least one year, with 90 days worth available immediately. Businesses will have options to implement this requirement, based on relative cost structures for OMS and more generic Azure storage services. Access control to log archives should very limited, and read-only. They are intended to be immutable. It will be advantageous to consider how that can be documented in evidence for a PCI assessment.

While the foundational architecture can support any size business getting started for PCI in Azure, the value sweet spot will be for small and medium-sized businesses (SMB). These businesses often have personnel wearing multiple hats, out of necessity. While this needs special attention for the PCI expectation of role-based access control (discussed above), monitoring and alerting can be a particular challenge for small teams. Security Center (along with OMS) offer useful automation and scaffolding for security monitoring. Additionally, the foundational architecture has anticipated integration with Security Center, which allow for greater visibility into the security of the CDE.

Example: security alerts via email don't scale. Configuring a combination of OMS and Security Center with alerts custom-designed for the resources involved, anticipated threats (from the annual Risk Assessment), and centralizing those for a dedicated security team should be a project for any larger organization starting from this foundational architecture. Smaller organizations may not operate at the same project-level scale, but will be better able to manage PCI-compliant monitoring in this manner.

ENTERPRISES AND SOLUTION ARCHITECTURE

For larger enterprises with multiple service providers requiring access to parts of their CDE, special attention should be paid to building separate avenues of access. Options include an API to facilitate structured connections (implemented with an ASE), or a custom bastion host, with limited access to relevant functionality. In both cases, logical access should be managed with Azure Active Directory to establish role-based access control that can be managed via policies.

SUPPLEMENTAL CONSIDERATIONS

CHANGE MANAGEMENT

The DSS mandates a robust process for Change Management, for a couple key reasons. First, changes in the components of a CDE (whether infrastructure or code), will have an impact on compliance. Second, a well-defined change management process is the best defense to unintended effects that could introduce vulnerabilities.

Among the key elements of a compliant change management program are:

- Business Impact Analysis for all proposed changes;
- Testing of proposed changes for both security and compliance; and
- A workflow for all changes, that considers the impact of the change, and results of testing before approval by management.
- Updating key documentation to reflect environmental changes (e.g. network diagrams).

The Attestation of Compliance (AOC), that an officer of the business will sign, includes a representation that the business will maintain PCI compliance. Practically speaking, this is only possible with a robust change management process.

PERIODIC VULNERABILITY MANAGEMENT

Vulnerability scanning is a DSS expectation, on a quarterly basis. Scans must be performed both inside the network and from the internet (by an ASV). These scans will be judged by different criteria. Internal scanning is judged by the business itself, and can be thought of as a risk tolerance. The ASV program has established an objective standard for vulnerabilities that all PCI compliance scans are measured against.

Annual penetration testing is expected, from multiple perspectives, to provide a backstop for overall vulnerability management efforts. The standard internal and external perspectives are required.

Additionally, internet-facing applications must be tested, and network segmentation that supports scoping must be validated.

The DSS includes three other noteworthy periodic vulnerability management requirements. An annual Risk Assessment is mandated. This report should model the risks of the organization and provide support for prioritization of security processes. Semi-annually, the technical controls that manage network segmentation (and the perimeter of the CDE) must be reviewed for technical correctness, and continued business need. These would include the NSGs and any whitelisted IP addresses. Finally, regular patching is required. The OMS Update Management Solution will alert on critical security updates applicable to the Bastion host.

DATA RETENTION

Once in possession of cardholder data, PCI expects businesses to have a policy for retention of that data. There must be a time period beyond which the data is securely deleted. The foundational architecture has the advantage of using Azure SQL, which can be used to periodically delete records older than a certain age. A common data retention period is three (3) years, but Coalfire recommends that, if the records have no operational value (e.g. chargebacks), that they should be deleted, or transformed into a safe (but useful) alternative.

Example: the data retention period is set at 1 year, considering that the return policy is 90 days. At the year mark, a stored procedure in the Azure SQL database copies the last 4 digits of the full PAN, securely deletes the PAN, and writes the last 4 back into the Credit Card Number field (e.g. XXXX-XXXX-XXXX-1234). This number looks masked, but it is actually truncated, a stronger form of redaction. PCI does not consider the last 4 digits of a PAN to be cardholder data (given that it cannot be mapped to the full PAN in the environment, which it cannot, since that data has been deleted).

INCIDENT RESPONSE

Incident response is a basic function for businesses that are exposed to PCI compliance. In addition to having an Incident Response Plan, businesses are expected to test that plan, evolve it over time (based on industry trends and results from prior exercises), and regularly train personnel who support incidents. In any public-facing business, there will be a steady stream of alerts for potentially-impactful incidents. The incident response function is intended to address this risk exposure.

PCI PROGRAM MANAGEMENT

Oversight of processes that make up and support PCI compliance is a necessary function. At the most mundane level, this includes periodic review of policies and procedures to ensure they are current. Change management will have more touchpoints, and the periodic vulnerability management expectations will be times where the organization focuses on execution.

Collectively, these are known as Business As Usual. This regular rhythm of activities is best addressed by an explicit role, even in small businesses where people wear many hats.

VENDOR MANAGEMENT

If Microsoft Azure is the lone service provider for a business build using this foundational architecture, that will simplify matters somewhat. Azure has attained PCI compliance, as documented in its Attestation of Compliance (AOC). In that case, the Azure Responsibility Summary should be consulted to ensure that the business is aligning its PCI compliance with Azure guidance for each in-scope service.

Where other service providers are used, the business must practice due diligence in evaluating vendors with PCI compliance exposure, before using them. Additionally, those service providers will have

Responsibility Summaries which will require consultation to obtain a holistic picture of how the business documents compliance.

CONCLUSION

The Azure PCI Foundational Architecture, as built and documented by Avyan Consulting, is a good foundation on which to build a PCI compliant business.

The major dimension of building out from the foundational architecture is how to implement payment processing. Those decisions will drive many of the following choices, including use of PaaS services, networking controls to segment internal zones, and increased complexity of logging and monitoring via OMS and Security Center.

Above and beyond the technical implementation, businesses will need administrative, governance, and operations management to track changes, manage periodic vulnerability management operations, manage vendors, and respond to incidents. While some of these will exist in an ongoing business, PCI requirements will need to be assessed for what further maturities are necessary.

Ultimately, PCI compliance is more than just technical controls. A successful business will also require processes that support ongoing technical and security compliance and address process maturity.

GLOSSARY OF COMMON PCI TERMS

These definitions are drawn from the Payment Card Industry (PCI) Security Standards Council Glossary of Payment and Information Security Terms, with additional entries helpful in the review of the Microsoft Azure PCI Foundational Architecture.

PAYMENT CARD INDUSTRY (PCI)

- **Payment Processor/Gateway / Merchant Bank / Acquirer** – Entity engaged by merchants to handle payment card transactions on their behalf. While payment processors typically provide acquiring services, payment processors are not considered acquirers (merchant banks) unless defined as such by a payment card brand.
- **QSA** – A company approved by the PCI Security Standards Council to validate an entity's adherence to PCI DSS requirements.
- **Cardholder Data (CHD, PAN)** – At a minimum, card data includes the primary account number (PAN), and may also include cardholder name and expiration date. Cardholder data also includes sensitive security data, defined below.
- **Sensitive Authentication Data (SAD)** – Security-related information used to authenticate cardholders and/or authorize payment card transactions, stored on the card's magnetic stripe or chip. Also includes the Security Code (3-4 digit number printed on front or back of cards).
- **Cardholder Data Environment (CDE)** – The collection of physical areas, networks, computers, data, and code that Store, Process and Transmit cardholder data.
- **Authorization** – In a payment card transaction, authorization occurs when a merchant receives transaction approval after the acquirer validates the transaction with the issuer/processor.
- **Service Provider** – A business entity that provides various services to merchants or other service providers. Typically, these entities store, process, or transmit card data on behalf of another entity (such as a merchant) OR are managed service providers that provide business process or IT-related services.

TECHNICAL TERMS

- **Network Segmentation** – Technical controls limiting access between two computer networks. **Azure**-specific technologies include: Network Security Groups (NSGs), virtual networking, whitelisted IP addresses.
- **Demilitarized Zone** – The perimeter of a network, set up with limited access, to facilitate ingress and egress. **Azure**-specfic technologies include: Application Gateway, Application Service Envrionment (ASE).
- **Authentication** – Process of verifying identity of an individual, device, or process.
- **Multi-factor Authentication (MFA)** – Authentication typically occurs through the use of one or more authentication factors such as: *Something You Know*, such as a password or passphrase, *Something You Have*, such as a token device or smart car, or *Something You Are*, such as a biometric.
- **Remote Access** – Access to a computer network from a location outside of that network. Remote access connections can originate either from inside the company's own network or from a remote location. An example of technology for remote access is a virtual private network (VPN). Remote access can be either internal (e.g. IT support) or external (e.g., service providers, third-party agents, integrators/resellers).
- **Encryption** – Process of using cryptography to mathematically convert information into a form unusable except to holders of a specific digital key. Use of encryption protects information by devaluing

it to criminals. **Transparent Data Encryption (TDE)** is the management of encryption by a database, so that data is stored encrypted and decrypted when retrieved. The encryption key is managed by the database. The PCI Foundational Architecture uses **Always-Encrypted Columns** which protect a particular column. These keys are stored in Key Vault, and managed by the business.

- **Tokenization** – A process by which the primary account number (PAN) is replaced with a surrogate value called a token. Tokens can be used in place of the original PAN to perform functions when the card is absent like voids, refunds, or recurring billing. Tokens also provide more security if stolen because they are unusable and thus have no value to a criminal.

VULNERABILITY MANAGEMENT

Vulernabilities are weaknesses which allow an attacker to reduce a system's information assurance. They require three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw. Managing vulnerabilities is a broad theme in PCI compliance.

- **Malware** – Malicious software designed to infiltrate a computer system with the intent of stealing data, or damaging applications or the operating system. Such software typically enters a network during many business-approved activities such as via email or browsing websites. Malware examples include viruses, worms, Trojans (or Trojan horses), spyware, adware, and rootkits.
- **Intrustion Detection/Prevention (IDS/IPS)** – A device or software application that monitors a network or systems for malicious activity or policy violations. Systems with response capability (to mitigate the threat), are known as Intrustion Prevention Systems (IPS). Systems are further classified as either network or host-based.
- **File Integrity Monitoring (FIM)** – A process that performs the act of validating the integrity of operating system and application software files using a verification method between the current file state and a known, good baseline. This comparison method often involves calculating a known cryptographic checksum of the file's original baseline and comparing with the calculated checksum of the current state of the file.
- **Vulnerability Scanning (internal, ASV)** – A process of detecting vulnerabilities from either inside the network, or from a public (external) perspective. External vulnerability scanning is mandated to be performed by a third-party who is qualified in the Approved Scan Vendor (ASV) program.
- **Penetration Testing (internal, external, application, segmentation)** – An authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data. PCI mandates testing from a variety of perspectives, including internally, externally, application-specific, and to validate network segmentation.
- **Risk Assessment** – Process of estimating risk from known threats.
- **Secure Development Lifecycle** – a process for planning, creating, testing, and deploying an information system, with particular attention to security.

ABOUT THE AUTHORS

Dan Stocker | Payments Practice Director | Cloud & Tech

Dan started his career on Wall Street and in the telecommunications industry. He currently specializes in advising and assessing large service providers, with an emphasis on major Cloud Service Providers. His PCI experience also includes large international merchants, both ecommerce and Bricks and Mortar. He has also worked with financial institutions on GLBA and FINRA assessments.

In his 11 years at Goldman Sachs, he held lead technical positions in Trading Technology and Tech Risk, including Business Continuity. At AT&T, Dan was a principal SME at the worldwide Frame Relay NOC. He holds an MBA, and a MS in Computer Science.

Published September 2017.

ABOUT COALFIRE

Coalfire is the cybersecurity advisor that helps private and public sector organizations avert threats, close gaps, and effectively manage risk. By providing independent and tailored advice, assessments, technical testing, and cyber engineering services, we help clients develop scalable programs that improve their security posture, achieve their business objectives, and fuel their continued success. Coalfire has been a cybersecurity thought leader for more than 16 years and has offices throughout the United States and Europe. Coalfire.com