Windows Azure™ Security Overview

By Charlie Kaufman and Ramanathan Venkatapathy

Abstract

Windows Azure, as an application hosting platform, must provide *confidentiality*, *integrity*, and *availability* of customer data. It must also provide transparent *accountability* to allow customers and their agents to track administration of services, by themselves and by Microsoft.

This document describes the array of controls implemented within Windows Azure, so customers can determine if these capabilities and controls are suitable for their unique requirements. The overview begins with a technical examination of the security functionality available from both the customer's and Microsoft operations' perspectives - including identity and access management driven by Windows Live ID and extended through mutual SSL authentication; layered environment and component isolation; virtual machine state maintenance and configuration integrity; and triply redundant storage to minimize the impact of hardware failures. Additional coverage is provided to how monitoring, logging, and reporting within Windows Azure supports accountability within customers' cloud environments.

Extending the technical discussion, this document also covers the people and processes that help make Windows Azure more secure, including integration of Microsoft's globally recognized SDL principles during Windows Azure development; controls around operations personnel and administrative mechanisms; and physical security features such as customer-selectable geo-location, datacenter facilities access, and redundant power.

The document closes with a brief discussion of compliance, which continues to have ongoing impact on IT organizations. While responsibility for compliance with laws, regulations, and industry requirements remains with Windows Azure customers, Microsoft's commitment to providing fundamental security capabilities and an expanding range of tools and options to meet customers' specific challenges is essential to Microsoft's own success, and key to our customers' success with Windows Azure.

August, 2010

Table of Contents

1 Introduction

Windows Azure™ is a cloud services operating system that serves as the development, service hosting and service management environment for the Windows Azure platform. Windows Azure provides developers with on-demand compute and storage to host, scale, and manage web applications on the Internet through Microsoft® datacenters.

With Windows Azure, Microsoft hosts data and programs belonging to customers. Windows Azure must therefore address information security challenges above and beyond traditional on- or off-premises IT scenarios. This document describes the array of controls Windows Azure customers can use to achieve their required level of security, and determine if the capabilities and controls are suitable for their unique requirements.

1.1 Audience and Scope

The intended audience for this whitepaper includes:

- Developers interested in creating applications that run on Windows Azure
- Technical decision makers (TDMs) considering Windows Azure to support new or existing services

The focal point of this whitepaper is the Windows Azure "operating system as an online service" platform component, and does not provide detailed coverage of any of the related Windows Azure platform components such as Microsoft SQL Azure, AppFabric, or Microsoft Codename "Dallas."

The discussion is focused around Windows Azure's security features and functionality. Although a minimal level of general information is provided, readers are assumed to be familiar with Windows Azure basic concepts as described in other references provided by Microsoft. Links to further information are provided in the "References & Further Reading" section at the end of this document.

A Glossary is also included at the end of this document that defines terms highlighted in **bold** as they are introduced.

1.2 Security Model Basics

Before delving deeper into the technical nature of Windows Azure's security features, this section provides a brief overview of its security model. Again, this overview assumes readers are familiar with basic Windows Azure concepts, and focuses primarily on security-related items.

1.2.1 Customer View: Compute, Storage, and Service Management

Windows Azure is designed to abstract much of the infrastructure that typically underlies applications (servers, operating systems, web & database software, and so on) so that

developers can focus on building applications. This section provides a brief overview of what a typical customer "sees" when approaching Windows Azure.

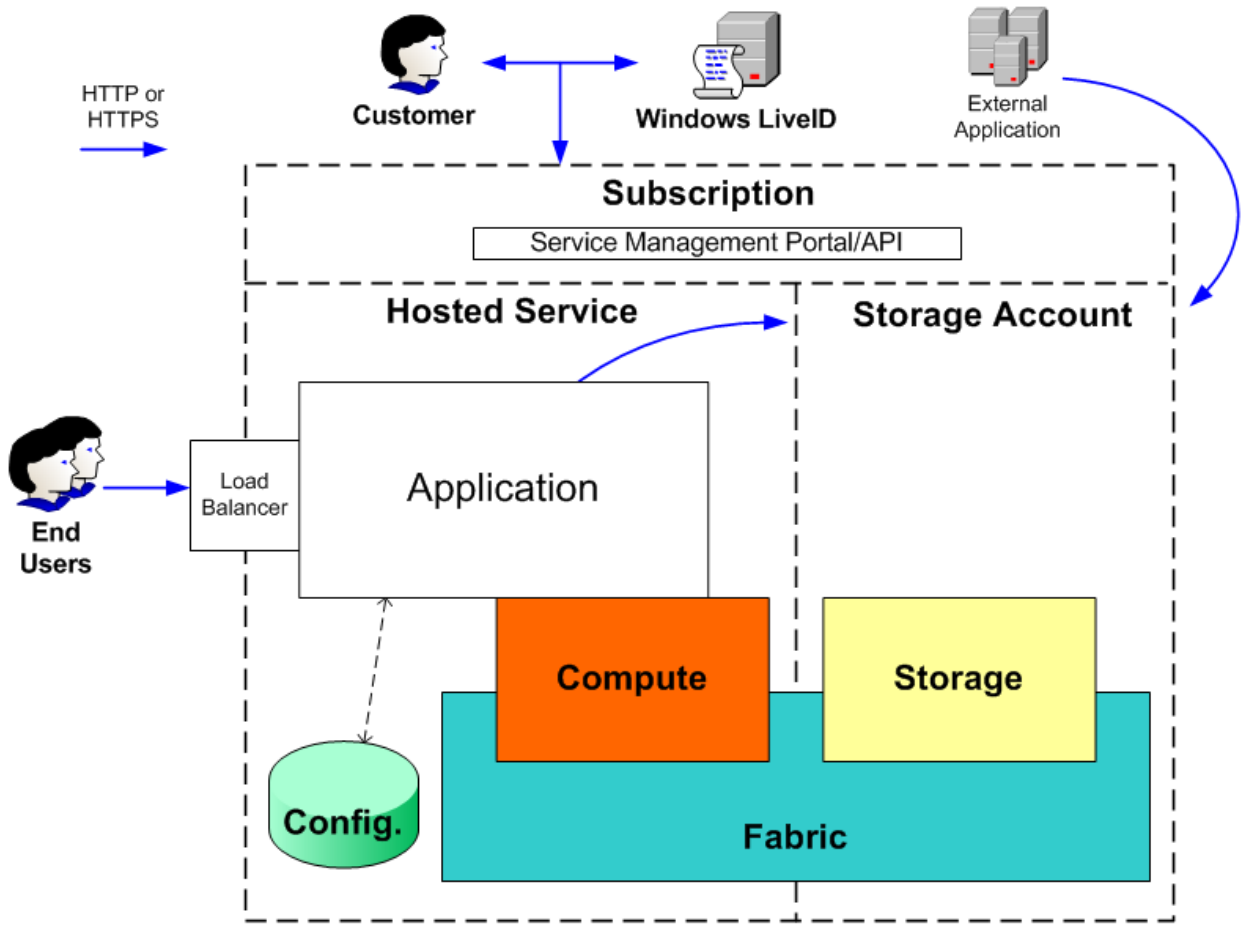

Figure 1: Simplified overview of key Windows Azure components.

As shown in Figure 1, Windows Azure provides two primary functions: cloud-based compute and storage, upon which customers build and manage applications and their associated configurations. Customers manage applications and storage through a **subscription**. A Subscription is created typically by associating new or existing credentials with a credit card number on the Subscription portal web site. Subsequent access to the Subscription is controlled by a Windows Live ID (https://login.live.com). Windows Live ID is one of the longest-running Internet authentication services available, and thus provides a rigorously tested gatekeeper for Windows Azure.

A subscription can include zero or more **Hosted Services** and zero or more Storage Accounts. A Hosted Service contains one or more deployments. A deployment contains one or more **roles**. A role has one or more instances. Storage Accounts contain blobs, tables, and queues. The Windows Azure drive is a special kind of blob. Access control for Hosted Services and Storage Accounts is governed by the subscription. The ability to authenticate with the Live ID associated

with the subscription grants full control to all of the Hosted Services and Storage Accounts within that subscription.

Customers upload developed applications and manage their Hosted Services and Storage Accounts through the **Windows Azure Portal** web site or programmatically through the Service Management API (SMAPI). Customers access the Windows Azure Portal through a web browser or access SMAPI through standalone command line tools, either programmatically or using Visual Studio.

SMAPI authentication is based on a user-generated public/private key pair and self-signed certificate registered through the Windows Azure Portal. The certificate is then used to authenticate subsequent access to SMAPI. SMAPI queues requests to the Windows Azure Fabric, which then provisions, initializes, and manages the required application. Customers can monitor and manage their applications via the Portal or programmatically through SMAPI using the same authentication mechanism.

Access to Windows Azure storage is governed by a storage account key (SAK) that is associated with each Storage Account. Storage account keys can be reset via the Windows Azure Portal or SMAPI.[1]

The compute and storage capabilities are further comprised of the fundamental functional units of Windows Azure. Figure 2 provides a more granular view, exposing these fundamental units and illustrating their relationships to the previously described components. All of the components described so far are summarized below:

- Hosted Services contain deployments, roles, and **role instances**
- Storage Accounts contain blobs, tables, queues, and drives

Each of these entities is defined in the Glossary, and further details about them can be found in general references on Windows Azure. They are introduced here briefly to facilitate further discussion of Windows Azure's security functionality in the remainder of the document.

The primary Windows Azure "subjects," "objects," and authentication mechanisms are summarized in Table 1.

[1] There are additional access control mechanisms exposed by Storage Accounts which are described in more detail in Section 7 of this document.

Table 1 - A summary of Windows Azure authentication mechanisms.

Subjects	Objects	Authentication Mechanism
Customers	Subscription (Compute & Storage)	Windows Live ID
Developers & Operators	Windows Azure Portal/API	Live ID (Windows Azure Portal) or Self-signed certificate (SMAPI)
Role Instances	Storage	Storage account key
External Applications	Storage	Storage account key
External Applications	Applications	Customer–defined

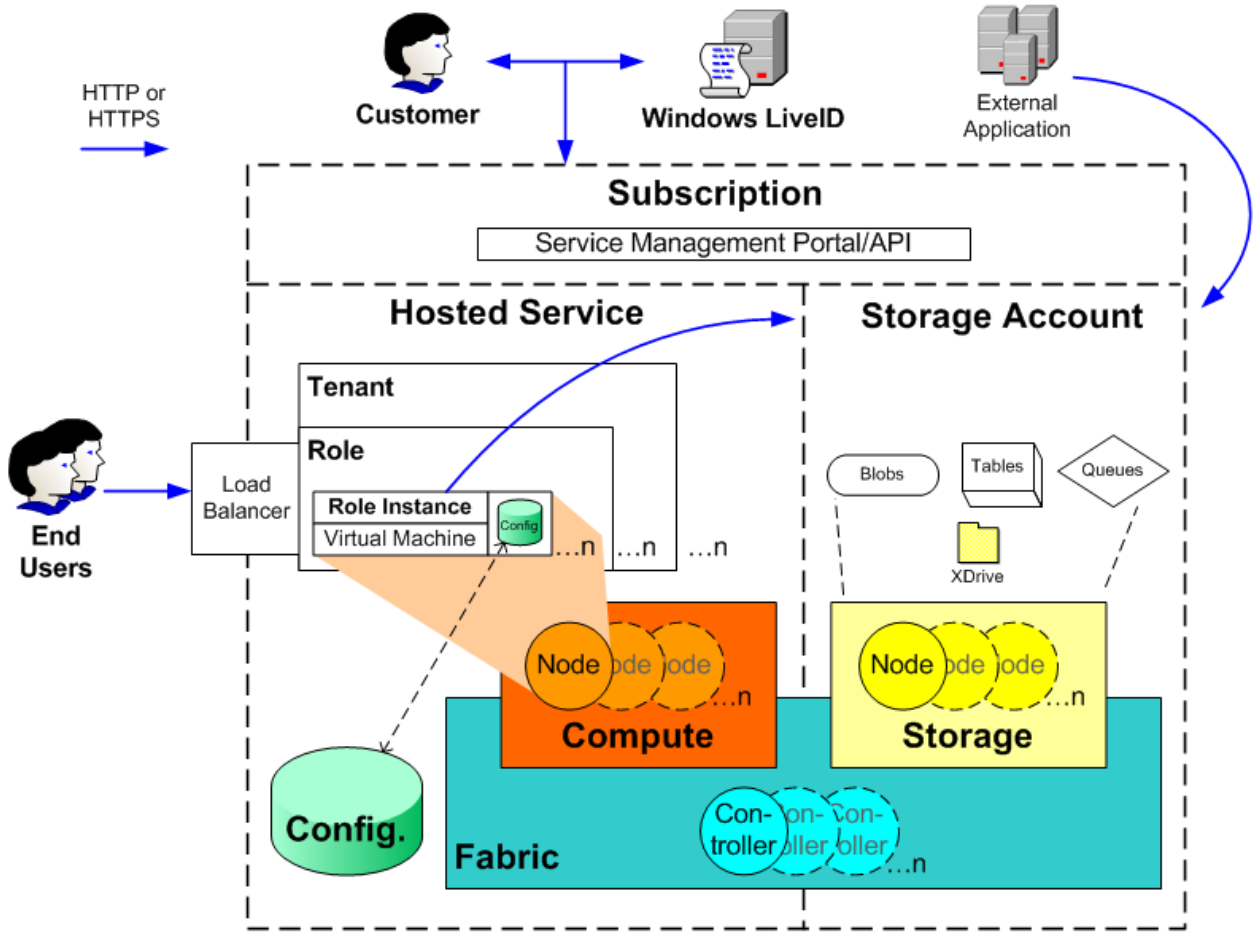

Figure 2: More granular illustration of Windows Azure components and relationships.

1.2.2 Windows Azure View: Fabric

Having described the high-level Windows Azure components manageable by customers, we'll now delve a bit deeper into the Fabric that underlies the basic compute and storage capabilities of Windows Azure. Although customers control aspects of the Fabric via defined management

interfaces as noted previously, the primary purpose of Windows Azure is to abstract management of this virtual infrastructure so that it simply appears as a consistent, scalable set of resources for customers. Put more simply, developers don't explicitly manage this virtual infrastructure – Microsoft does. This section introduces some of the basic components of the Windows Azure Fabric that Microsoft manages directly.

Based on the number of role instances specified by customers, Windows Azure creates a virtual machine (VM) for each role instance, then runs the role in those VMs. These VMs in turn run on a **hypervisor** that's specifically designed for use in the cloud (the **Windows Azure Hypervisor**). One VM is special: it runs a hardened operating system called the **root OS** that hosts a **fabric agent (FA)**. FAs are used in turn to manage **guest agents (GA)** within **guest OSes** on customer VMs. FAs also manage storage nodes. The collection of Windows Azure hypervisor, root OS/FA, and customer VMs/GAs comprises a **compute node**.

FAs are managed by a **fabric controller (FC)**, which exists outside of compute and storage nodes (compute and storage **clusters** are managed by separate FCs). If a customer updates their application's **configuration file** while it's running, the FC communicates with the FA, which then contacts GAs, which notifies the application of the configuration change. In the event of a hardware failure, the FC will automatically find available hardware and restart the VM there.

2 Cloud Security Design

Fundamentally, Windows Azure must provide *confidentiality*, *integrity*, and *availability* of customer data, just like any other application hosting platform. It must also provide transparent *accountability* to allow customers and their agents to track administration of applications and infrastructure, by themselves and by Microsoft. Drawing on the basic components and relationships described so far, this section will illustrate how Windows Azure provides these classical dimensions of information security.

2.1 Confidentiality

Confidentiality ensures that a customer's data is only accessible by authorized entities. Windows Azure provides confidentiality via the following mechanisms:

- Identity and Access Management - Ensures that only properly authenticated entities are allowed access.
- Isolation - Minimizes interaction with data by keeping appropriate containers logically or physically separate.
- Encryption - Used internally within Windows Azure for protecting control channels and is provided optionally for customers who need rigorous data protection capabilities.

More detail about how each of these data protection mechanisms is implemented in Windows Azure follows.

2.1.1 Identity and Access Management

The strongest security controls available are no protection against an attacker who gains unauthorized access to credentials or keys. Thus, credential and key management are critical components of the security design and implementation of Windows Azure.

All of the primary identities and authentication mechanisms have been introduced previously, and are summarized in Table 1. This section provides further details around vital elements including APIs, application privilege levels, key distribution, and authentication credentials for trusted subsystems like the fabric controller.

2.1.1.1 SMAPI Authentication

The Service Management API (SMAPI) provides web services via the **Representational State Transfer (REST)** protocol and is intended for use by Windows Azure tools provided to customer developers. The protocol runs over SSL and is authenticated with a certificate and private key generated by the customer. This certificate does not chain back to a trusted root certificate authority (CA). Rather, it is self-signed and its fingerprint is associated with the subscription via the Windows Azure Portal. As long as the customer maintains control of the private key and the Live ID used to create the account, this mechanism provides a high degree of assurance that only the customers' authorized representatives can access specific aspects of the service.

2.1.1.2 Least Privilege Customer Software

Running applications with "least privilege" is widely regarded as an information security best practice. To align with the principle of least privilege, customers are not granted administrative access to their VMs, and customer software in Windows Azure is restricted to running under a low-privilege account by default (in future versions, customers may select different privilege models at their option). This reduces the potential impact and increases the necessary sophistication of any attack, requiring privilege elevation in addition to other exploits. It also protects the customer's service from attack by its own end users.

2.1.1.3 SSL Mutual Authentication for Internal Control Traffic

All communications between Windows Azure internal components are protected with SSL. In most cases, the SSL certificates are self-signed. Exceptions are for any certificates for connections that could be accessed from outside the Windows Azure network (including the storage service), and for the fabric controllers.

Fabric controllers have certificates issued by a Microsoft CA that chains back to a trusted root CA. This allows FC public keys to be rolled over easily. Additionally, FC public keys are used by

Microsoft developer tools so that when developers submit new application images, they are encrypted with a FC public key in order to protect any embedded secrets.

2.1.1.4 Certificate and Private Key Management

To lower the risk of exposing certificates and private keys to developers and administrators, they are installed via a separate mechanism than the code that uses them. Certificates and private keys are uploaded via SMAPI or the Windows Azure Portal as **PKCS12** (PFX) files protected in transit by SSL. Those PKCS12 files may be password protected, but if so, the password must be included in the same message. SMAPI removes the password protection (if necessary) and encrypts the entire PKCS12 blob using SMAPI's public key and stores it in a secret store on the FC, along with a short certificate name and the public key as metadata.

The configuration data associated with any role within the same subscription specifies the certificates that should be made available to the role. When a role is instantiated on a VM, the FC retrieves the appropriate certificate, decrypts the PKCS12 blob, re-encrypts it using the FA's public transport key, and sends it to the FA on the node. The FA on the node sends it to the GA in the VM that is instantiating the role, and then the GA decrypts it and installs it in the operating system certificate store with a flag indicating that the private key can be used but not exported. After installation, all temporary copies of the certificates and keys are destroyed; if reinstallation is required, the certificates must be repackaged by the FC.

2.1.1.5 Hardware Device Credentials Used by the FC

In addition to application keys, the FC must maintain a set of credentials (keys and/or passwords) used to authenticate itself to various hardware devices under its control. The system used for transporting, persisting, and using these credentials is designed to make it unnecessary for Windows Azure developers, administrators, and backup services/personnel to be exposed to secret information. Encryption based on the FC's master identity public key is used at FC setup and FC reconfiguration time to transfer the credentials used to access networking hardware devices, remote power switches on the racks that are used to power cycle individual nodes, and other systems. The FC maintains these secrets in its internal replicated data store (still encrypted with its master identity public key). Credentials are retrieved and decrypted by the FC when it needs them.

2.1.1.6 Access Control in Windows Azure Storage

As discussed earlier, Windows Azure Storage has a simple access control model. Each Windows Azure subscription can create one or more Storage Accounts. Each Storage Account has a single secret key that is used to control access to all data in that Storage Account. This supports the typical scenario where storage is associated with applications and those applications have full control over their associated data. A more sophisticated access control model can be achieved by creating a custom application "front end" to the storage, giving the application the storage

key, and letting the application authenticate remote users and even authorize individual storage requests.

Two mechanisms support generalized access control scenarios. A portion of the data in a storage account can be marked as *publicly readable*, in which case requests to read that data are allowed without a shared key signature. The primary use of this feature is to access non-sensitive data such as web page images.

The other mechanism is called a *Shared Access Signature (SAS)*, where a process, knowing a given storage account key (SAK), can create a query template and sign it with the SAK. That signed URL can be given to another process which can then fill in the details of the query and make the request of the storage service. Authentication is still based on a signature created using the SAK, but it is sent to the storage server by a third party. Such delegations can be limited in terms of validity time, permission set and what portions of the Storage Account are accessible.

A Shared Access Signature may also reference a *Container-Level Access Policy*, which substitutes in the URL for some number of parameters (such as validity time or permission set). Those parameters are instead dictated by the named access policy, which is stored within Windows Azure Storage. Because a Container-Level Access Policy can be modified or revoked at any time, it provides greater flexibility and control over the permissions that are granted.

To support periodically changing SAKs without any breaks in service, a Storage Account can have two secret keys associated with it at the same time (where either key gives full access to all of the data). The sequence for changing the secret key is to add the new one as authorized to the storage service, then change the key used by all applications accessing the service, and finally remove the old key so that it will no longer be authorized. Changing the set of authorized storage keys associated with an account is done via SMAPI or the Windows Azure Portal using the subscription credentials.

2.1.2 Isolation

Beyond authenticating access to data, simply keeping different data appropriately segregated provides well-recognized protection. Windows Azure provides isolation at a number of levels, as discussed below.

2.1.2.1 Isolation of Hypervisor, Root OS, and Guest VMs

A critical boundary is the isolation of the root VM from the guest VMs and the guest VMs from one another, managed by the hypervisor and the root OS. The hypervisor/root OS pairing leverages Microsoft's decades of operating system security experience, as well as more recent learning from Microsoft's Hyper-V, to provide strong isolation of guest VMs.

2.1.2.2 Isolation of Fabric Controllers

As the central orchestrator of much the Windows Azure Fabric, significant controls are in place to mitigate threats to fabric controllers, especially from potentially compromised FAs within customer applications. Communication from FC to FA is unidirectional – the FA implements an SSL-protected service that is accessed from the FC and replies to requests only. It cannot initiate connections to the FC or other privileged internal nodes. The FC strongly parses all responses as though they were untrusted communications.

In addition, the FCs and devices incapable of implementing SSL are on separate VLANs, which limits exposure of their authentication interfaces to a compromised node that hosts VMs.

2.1.2.3 Packet Filtering

The hypervisor and the root OS provide network packet filters that assure that the untrusted VMs cannot generate spoofed traffic, cannot receive traffic not addressed to them, cannot direct traffic to protected infrastructure endpoints, and cannot send or receive inappropriate broadcast traffic.

Storage nodes run only Windows Azure-provided code and configuration, and access control is thus narrowly tailored to permit legitimate customer, application, and administrative access only.

Customer access to VMs is limited by packet filtering at edge load balancers and at the root OS. In particular, remote debugging, remote Terminal Services, or remote access to VM file shares is not permitted by default; Microsoft plans to permit customers to enable these protocols as an explicit option in the future. Microsoft allows customers to specify whether any connections are accepted from the Internet and from role instances within the *same* application.

Connections between role instances of *different* applications are considered to be Internet connections. Connectivity rules are cumulative; for example, if role instances A and B belong to different applications, A can open a connection to B only if A can open connections to the Internet and B can accept connections from the Internet.

The fabric controller translates the list of roles into a list of role instances, and from that to a list of IP addresses. This list of IP addresses is used by the FA to program the packet filters to only allow intra-application communication to those IP addresses. Roles are allowed to initiate communication to Internet addresses. This enables them to communicate with the Internet and send traffic to any other role with visibility from the Internet via their **VIPs**).

2.1.2.4 VLAN Isolation

VLANs are used to isolate the FCs and other devices. VLANs partition a network such that no communication is possible between VLANs without passing through a router, which prevents a compromised node from faking traffic from outside its VLAN except to other nodes on its VLAN, and it also cannot eavesdrop on traffic that is not to or from its VLANs.

There are three VLANs in each cluster:

- The main VLAN – interconnects untrusted customer nodes
- The FC VLAN – contains trusted FCs and supporting systems
- The device VLAN – contains trusted network and other infrastructure devices

Communication is permitted from the FC VLAN to the main VLAN, but cannot be initiated from the main VLAN to the FC VLAN. Communication is also blocked from the main VLAN to the device VLAN. This assures that even if a node running customer code is compromised, it cannot attack nodes on either the FC or device VLANs.).

2.1.2.5 Isolation of Customer Access

The systems managing access to customer environments (the Windows Azure Portal, SMAPI, and so on) are isolated within a Windows Azure application operated by Microsoft. This logically separates customer access infrastructure from customer applications and storage.

2.1.3 Encryption

Encryption of data in storage and in transit can be used by customers within Windows Azure to align with best practices for ensuring confidentiality and integrity of data. As noted previously, critical internal communications are protected using SSL encryption. At the customer's option, the Windows Azure SDK extends the core .NET libraries to allow developers to integrate the .NET Cryptographic Service Providers (CSPs) within Windows Azure. Developers familiar with .NET CSPs can easily implement encryption, hashing, and key management functionality for stored or transmitted data. For example, using the .NET CSPs, Windows Azure developers can easily access:

- Recognized encryption algorithms like AES that have years of real-world exposure and testing, avoiding the classic mistake of attempting to "roll your own crypto" for applications.
- A full array of cryptographic hash functionality including MD5 and SHA-2 to verify data correctness, create and validate digital signatures, and create non-identifiable tokens in place of sensitive data.
- The RNGCryptoServiceProvider class to generate random numbers sufficient to seed the high level of entropy required for strong cryptography.
- Straightforward key management methods that enable simple manipulation of custom encryption keys within Windows Azure Storage.

For more detailed descriptions of how to leverage cryptographic capabilities provided by Windows Azure, please see "References & Further Reading" at the end of this document.

2.1.4 Deletion of Data

Where appropriate, confidentiality should persist beyond the useful lifecycle of data. Windows Azure's Storage subsystem makes customer data unavailable once delete operations are called. All storage operations including delete are designed to be instantly consistent. Successful execution of a delete operation removes all references to the associated data item and it cannot be accessed via the storage APIs. All copies of the deleted data item are then garbage collected. The physical bits are overwritten when the associated storage block is reused for storing other data, as is typical with standard computer hard drives. Section 4.4.3 discusses disposal of physical media.

2.2 Integrity

Customers seeking to outsource their data compute and storage workloads to Windows Azure obviously expect it to be protected from unauthorized changes. Microsoft's cloud operating system provides this in a number of ways.

The primary mechanism of integrity protection for customer data lies within the Fabric VM design itself. Each VM is connected to three local Virtual Hard Drives (VHDs):

- The D: drive contains one of several versions of the Guest OS, kept up-to-date with relevant patches, selectable by the customer.
- The E: drive contains an image constructed by the FC based on the package provided by the customer.
- The C: drive contains configuration information, paging files, and other storage.

The D: and E: virtual drives are effectively read-only because their ACLs are set to disallow write access from customer processes. Since the operating system may need to update those read-only volumes, they are implemented as VHDs with delta files. The initial VHDs for all role instances in an application generally start out identical. The delta drive for the D: drive is discarded any time Windows Azure patches the VHD containing the OS. The delta drive for the E: drive is discarded any time the VHD is updated with a new application image. This design strictly preserves the integrity of the underlying operating system and customer applications.

Another primary integrity control is of course the configuration file, which is stored on the read/write C: drive. The customer provides a single configuration file specifying the connectivity requirements of all roles in the application. The FC takes the subset of that configuration file appropriate for each role and places it in the C: drive for each role instance. If the customer updates the configuration file while the role instances are running, the fabric controller (FC) – through the fabric agent (FA) – contacts the guest agent (GA) running in the VM's guest OS and instruct it to update the configuration file on the C: drive. It can then signal the customer's application to re-read the configuration file. The contents of the C: drive are not discarded for this event, which means that the C: drive appears to the customer's application to be stable

storage.[2] Only authorized customers accessing their Hosted Services via the Windows Azure Portal or SMAPI (as described earlier) can change the configuration file. So, by the inherent design of Windows Azure, the integrity of the customer configuration is protected, maintained, and persisted constantly during an application's lifetime.

As for Windows Azure Storage, integrity is dictated by applications using the simple access control model described earlier. Each Storage Account has two storage account keys that are used to control access to all data in that Storage Account, and thus access to the storage keys provide full control over the associated data.

Finally, the integrity of the Fabric itself is carefully managed from bootstrap through operation. As noted earlier, the root OS that runs on VM hosting nodes within the Fabric is a hardened operating system. After a compute node is booted, it starts the fabric agent (FA) and awaits connections and commands from the fabric controller. The FC connects to the newly booted node using SSL, authenticating bi-directionally via SSL as described previously. FC communication with FAs is via one-way push, making it difficult to attack those higher in the chain of command because they cannot make requests of directly to those components. Combined with the many mechanisms described above, these features help maintain the Fabric in a pristine state for customers.

2.3 Availability

One of the main advantages provided by cloud platforms is robust availability based on extensive redundancy achieved with virtualization technology. Windows Azure provides numerous levels of redundancy to provide maximum availability of customers' data.

Data is replicated within Windows Azure to three separate nodes within the Fabric to minimize the impact of hardware failures.

Customers can leverage the geographically distributed nature of the Windows Azure infrastructure by creating a second Storage Account to provide hot-failover capability. In such a scenario, customers may create custom roles to replicate and synchronize data between Microsoft facilities. Customers may also write customized roles to extract data from storage for offsite private backups.

The guest agents (GAs) on every VM monitor the health of the VM. If the GA fails to respond, the FC reboots the VM. In the future, customers can optionally choose to run more sophisticated health monitoring processes adapted to a customized continuity/recovery policy. In case of hardware failure, the FC moves the role instance to a new hardware node and reprograms the network configuration for the service role instances to restore the service to full availability.

[2] All three drives will revert to their initial states if the role instance is ever moved to a different physical machine; thus, customer applications should only cache data to the C: drive as a performance optimization.

As noted earlier, each VM has a D: drive containing customer-selectable versions of the Guest OS. The customer can either manually move from one build of the Guest OS to another or choose to let Microsoft move their applications as new builds are released. This system maximizes availability throughout regular maintenance events with minimal customer interaction.

FCs adhere to similar principles of high availability through redundancy and automatic failover that are used for a customer's services, resulting in continuous availability of FC manageability capabilities. During an upgrade of the Windows Azure platform or a customer's service software, FCs utilize a logical partition called a*n update domain* to change a portion of a given service's role instances at a given time while the remaining instances continue to serve requests. FCs are also aware of potential hardware and network points of failure through specification of *fault domains*. For any service that has more than one role instance, Windows Azure ensures that these instances are deployed across multiple update and fault domains (unless specified otherwise by the customer) in order to maintain full availability of the service through updates and isolated network hardware failures.

2.4 Accountability

Because cloud computing platforms are effectively an outsourced computing environment, they have to be able to *demonstrate* safe operation to customers and their designated agents on a regular basis. Windows Azure implements multiple levels of monitoring, logging, and reporting to provide this visibility to customers. Primarily, the **monitoring agent (MA)** gathers monitoring and diagnostic log information from many places including the FC and the root OS and writes it to log files. It eventually pushes a digested subset of the information into a pre-configured Windows Azure Storage Account. In addition, the **Monitoring Data analysis Service (MDS)** is a freestanding service that reads various monitoring and diagnostic log data and summarizes/digests the information, writing it to an integrated log.

3 Security in the Development Lifecycle

Microsoft employs widely recognized tools and techniques to provide security assurance within Windows Azure's development processes and around the design and implementation of the service itself.

Windows Azure fully integrates Microsoft's Security Development Lifecycle (SDL) guidelines, recognized worldwide as a model for software security assurance programs (more information on SDL can be found in "References & Further Reading").

In particular, Microsoft scrutinizes places where data from a less-trusted component is parsed by a more trusted component. For example:

- When the Windows Azure hypervisor and root OS processes requests for disk I/O and network I/O from customer controlled VMs.
- When Windows Azure portal and SMAPI processes requests coming over the network from sources controlled by customers.
- When the fabric controller (FC) parses customer configuration data passed via SMAPI.

In addition to careful design and implementation, these components are developed using managed programming languages like C# that reduce the likelihood of well-known memory manipulation exploits, and are subjected to extensive testing of the interfaces before the Fabric is switched to production mode. Microsoft continues these practices when upgrading or modifying code that handles external requests.

Microsoft's SDL guidance is also promoted extensively to customers of Windows Azure, since the security of applications hosted on Windows Azure depends a great deal on the customers' development processes. A companion to this document, *Security Best Practices For Developing Windows Azure Applications*, is also available on microsoft.com (see "References & Further Reading").

Assuming both Microsoft and customers follow SDL, there still remains a remote probability of compromise between development and deployment to Windows Azure. As discussed previously, customers provision applications directly via SMAPI, which uses certificate authentication, and HTTPS-protected channels for code transfer, among other controls.

4 Service Operations

The people and processes that operate Windows Azure are perhaps the most important security feature of the platform. This section describes features of Microsoft's datacenter infrastructure that help enhance and maintain security, continuity, and privacy.

4.1 Microsoft Operations Personnel

Windows Azure developers and administrators have, by design, been given sufficient privileges to carry out their assigned duties to operate and evolve the service. As noted throughout this document, Microsoft deploys combinations of preventive, detective and reactive controls including the following mechanisms to help protect against unauthorized developer and/or administrative activity:

- Tight access control to sensitive data
- Combinations of controls that greatly enhance independent detection of malicious activity
- Multiple levels of monitoring, logging, and reporting

Additionally, Microsoft conducts background verification checks of certain operations personnel, and limits access to applications, systems, and network infrastructure in proportion to the level of background verification.

Microsoft operations personnel follow a formal process when they are required to access a customer's account or related information, and this is only done at the customer's request.

4.2 Security Response

Microsoft security vulnerabilities can be reported to the Microsoft Security Response Center (http://www.microsoft.com/security/msrc/default.aspx) or via email to secure@microsoft.com. Microsoft follows a consistent process to assess and respond to vulnerabilities and incidents reported via the standard facilities.

4.3 Network Administration

The networking hardware that connects all of the Windows Azure components is clearly a critical component of the platform. This section describes some of the security measures employed by the service at this layer.

As noted previously, the Windows Azure internal network is isolated by strong filtering of traffic to and from other networks. This provides a "backplane" for internal network traffic that is high-speed and at low risk from malicious activity generally.

The configuration and administration of network devices such as switches, routers, and load balancers is performed only by authorized Microsoft operations personnel, and generally only at major changes (such as when the data center itself is reconfigured). The virtualization provided by the Windows Azure Fabric makes such changes practically invisible to customers.

Furthermore, any hardware that does not implement adequate communications security features (such as SSL) is administered over a separate LAN that is isolated from nodes that are exposed to the Internet, or customer access.

4.3.1 Remote Administration of Fabric Controllers

Fabric controllers have an RPC-accessible API that accepts commands from SMAPI, and from Windows Azure administrators. Policy-level decisions are enforced by SMAPI at the level of the application, which will only generate requests concerning a customer's own resources based on the authenticated identity of the customer. FCs make finer-grained access control decisions, befitting of their role as the low-level central provisioning and management facility.

Connections to FCs are via SSL where the client is authenticated with a client certificate, and the certificate fingerprint determines if a caller has the appropriate access level to make a given request.

4.4 Physical Security

A system cannot be more secure than the physical platform on which it runs. Windows Azure runs in geographically distributed Microsoft facilities, sharing space and utilities with other Microsoft Online Services. Each facility is designed to run 24 x 7 and employs various measures to help protect operations from power failure, physical intrusion, and network outages. These data centers comply with industry standards for physical security and reliability and they are managed, monitored, and administered by Microsoft operations personnel. They are designed for "lights out" operation. Further details of Windows Azure's physical security are discussed below.

4.4.1 Facilities Access

Microsoft uses industry standard access mechanisms to protect Windows Azure's physical infrastructure and datacenter facilities. Access is limited to a very small number of operations personnel, who must regularly change their administrative access credentials. Datacenter access, and the authority to approve data center access, is controlled by Microsoft operations personnel in alignment with local data center security practices.

4.4.2 Power Redundancy and Failover

Each datacenter facility has a minimum of two sources of electrical power, including a power generation capability for extended off-grid operation. Environmental controls are self-contained and remain operational as long as the facility and contained systems remain online.

Physical security controls are designed to "fail closed" during power outages or other environmental incidents. In case of fire or situations that could threaten life safety, the facilities are designed to allow egress without remaining exposed.

4.4.3 Media Disposal

Upon systems end-of-life, Microsoft operational personnel follow rigorous data handling procedures and hardware disposal processes.

5 Compliance

The importance of business and regulatory compliance has increased dramatically with the proliferation of global standards including ISO 27001, Safe Harbor and many others. In many cases, failure to comply with these standards can have a dramatic impact on organizations, up to and including catastrophic financial penalties and damage to reputation.

Any of the previously discussed threats can have an impact on compliance, but there are also threats that are directly related to failure to adhere to recognized practices, provide representation of compliance to independent auditors, support e-discovery, and otherwise facilitate reasonable efforts by customers to verify alignment with regulatory, legal, and

contractual requirements. Microsoft provides customers with the information they need to decide whether it is possible to comply with the laws and regulations to which they are subject within the context of Windows Azure and the tools to demonstrate that compliance when it is possible. Some of the ways Windows Azure assists customers with compliance are discussed below.

5.1 Customer-Selectable Geo-location

One of the key challenges inherent to Windows Azure is balancing compliance requirements against one of the key economic drivers behind cloud services: segmenting customer data and processing across multiple systems, geographies, and regulatory jurisdictions. Windows Azure addresses this challenge in a very simple way: customers choose where their data is stored. Data in Windows Azure is stored in Microsoft datacenters around the world based on the geo-location properties specified by the customer using the Windows Azure Portal. This provides a convenient way to minimize compliance risk by actively selecting the geographic locations in which regulated data will reside.

5.2 Compliance Controls

At the level of discrete controls, this document has illustrated Windows Azure's alignment with recognized compliance practices in many dimensions. To review some of these key security features that enable compliance:

Table 2: Compliance enabling features

Domain	Relevant Sections	Overview
Access control	1.2	Windows Azure provides numerous access control capabilities to protect against unauthorized administrative or end-user access.
Encryption	2.1.3	Encryption of data in storage and in transit can be defined by customers within Windows Azure to align with best practices for ensuring confidentiality and integrity of data.
Availability	2.3, 4.4	Customers can write customized roles to provide backups. Windows Azure's physical infrastructure is housed in managed geo-redundant facilities.
Privacy	2.1.4	Windows Azure Storage is designed to ensure customer-deleted data is faithfully and consistently erased.

5.3 ISO 27001 Certification

Trusted third-party certification provides a well-established mechanism for demonstrating protection of customer data without giving excessive access to teams of independent auditors that may threaten the integrity of the overall platform. Windows Azure operates in the Microsoft Global Foundation Services (GFS) infrastructure, portions of which are ISO27001-certified. ISO27001 is recognized worldwide as one of the premiere international information security management standards. Windows Azure is in the process of evaluating further industry certifications.

In addition to the internationally recognized ISO27001 standard, Microsoft Corporation is a signatory to Safe Harbor and is committed to fulfill all of its obligations under the Safe Harbor Framework.

While responsibility for compliance with laws, regulations, and industry requirements remains with Windows Azure customers, Microsoft remains committed to helping customers achieve compliance through the features described above.

6 References & Further Reading

The following resources are available to provide more general information about Windows Azure and related Microsoft services, as well as specific items referenced in the main text:

- Windows Azure Home – general information and links to further resources about Windows Azure; http://www.microsoft.com/windowsazure/
- Windows Azure Developer Center – main repository for developer guidance and information; http://msdn.microsoft.com/en-us/windowsazure/default.aspx
- Security Best Practices For Developing Windows Azure Applications – http://download.microsoft.com/download/7/3/E/73E4EE93-559F-4D0F-A6FC-7FEC5F1542D1/SecurityBestPracticesWindowsAzureApps.docx
- Crypto Services and Data Security in Windows Azure – http://msdn.microsoft.com/en-us/magazine/ee291586.aspx
- Microsoft's Security Development Lifecycle -- SDL is Microsoft's security assurance process that is employed during the development of Windows Azure; www.microsoft.com/security/sdl/
- Microsoft's Global Foundation Services Security – the group accountable for delivering the trustworthy, available online operations environment that underlies Windows Azure; http://www.globalfoundationservices.com/security/
- Microsoft GFS' ISO 27001 certification – http://www.bsigroup.com/en/Assessment-and-certification-services/Client-directory/CertificateClient-Directory-Search-Results/?pg=1&licencenumber=IS+533913&searchkey=companyXeqXMicrosoft
- Microsoft Security Response Center -- Microsoft security vulnerabilities, including issues with Windows Azure, can be reported to http://www.microsoft.com/security/msrc/default.aspx or via email to secure@microsoft.com.

6.1 Disclaimer

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

7 Glossary

Term	Definition
application	A collection of roles that, when instantiated on VMs, provide a Hosted Service.
cluster	A collection of hardware modules under the control of a single fabric controller.
compute node	The collection of hypervisor, root OS/FA, and customer VMs/GAs comprises a compute node.
configuration file	The customer provides a single configuration file specifying the connectivity requirements of all roles in the application. The FC takes the subset of that configuration file appropriate for each role and places it in the C: drive for each role instance/VM. If the customer updates the configuration file while the role instances are running, the Fabric instructs all VMs to update their configuration files, and then signals the customer's application to reread the configuration file.
customer	In the context of this document, the customer is the party who is buying resources on Windows Azure from Microsoft for the purpose of running some application. The term customer includes internal Microsoft groups who deploy their applications to Windows Azure.
end user	End users are the people who access services deployed on the Windows Azure Fabric. They could be employees or customers of customers (as defined above). They generally access these services over the Internet (except for the case where the end user is a different Windows Azure customer, in which case requests may come from within the Windows Azure Fabric but are treated as coming from the Internet). End users are by design not trusted by the Windows Azure infrastructure or by our default customer configuration, and so the infrastructure provides mechanisms to protect against end users and for our customers to secure their services against them.
FA (fabric agent)	A component of the root OS that opens an SSL port that accepts incoming connections and requests from the fabric controller and performs local configuration actions on the node including creation and deletion of VMs and updates to the locally stored OS images and itself.
FC (fabric controller)	The software that executes the algorithms to manage and provision physical hardware, allocate disk resources, CPU resources, RAM, and VMs to customers, deploy application and OS images to nodes, and program the packet filters to control connectivity within a Fabric. It also participates in the node initialization process by serving the OS images for remote network boot via Intel's Preboot eXecution Environment (PXE) framework.
hosted service	A customer-defined, cloud-based service hosted by Windows Azure on behalf of Microsoft's customers.
GA (guest agent)	A Windows Azure-provided agent that runs within the Guest VM and provides services like role health measurement and the installation of certificates and private keys. This agent communicates with the outside world through a private connection to the FA in the root partition. While GAs are provided by Windows

	Azure, they run within security context of a application, and are thus considered application code within the Windows Azure security model.
guest OS	An operating system tested for compatibility with Windows Azure that runs on a VM on behalf of a customer. Each Guest OS is designed to be generally compatible with a specific release of Windows Server.
hypervisor	The software component used to isolate all customer code that runs in Windows Azure. It runs directly over the hardware and divides a node into a variable number of VMs. Together with the root OS, it enforces constraints on outside communications and divides resources.
load balancer	A hardware networking device that accepts Internet traffic coming into Windows Azure and forwards it to an appropriate IP address and port within the Fabric. In the common case where there are several different machines or VMs that can handle a given request, the load balancer allocates the connections in a way that balances the load among them. The load balancer's routing tables must be updated as VMs are created, deleted, and moved from one piece of hardware to another.
MA (monitoring agent)	An agent that runs in many places including the FC and the Root OS and gathers monitoring and diagnostic log information and writes it to log files. It eventually pushes a digested subset of the information into a pre-configured Windows Azure Storage Account.
MDS (monitoring data analysis service)	A freestanding service that reads various monitoring and diagnostic log data and summarizes/digests the information, writing it to an integrated log.
packet filter	A network policy enforcement mechanism implemented by the root partition of a node that enforces IP connectivity restrictions within the Windows Azure Fabric.
PKCS12	One of the Public-Key Cryptography Standards (PKCS), published by RSA Laboratories, which defines a file format commonly used to store X.509 private keys with accompanying public key certificates, protected with a password-based symmetric key.
REST (representational state transfer)	An RPC protocol running over SOAP used for many interactions within the Windows Azure Fabric and with Windows Azure customer development environments.
role	A process within an application that is comprised of two or more identical role instances distributed across multiple nodes to provide scalability and fault tolerance. Every Hosted Service has at least one role, and most have two or three. Complex services could have many roles. The term "role" is also sometimes used to refer to the collection of code and configuration settings that define the role's behavior and are used to instantiate single-node role instances.
role instance	A process running in a virtual machine implementing one single instance of a role's portion of a Hosted Service. For scalability and availability, there are typically several instances of a given role running at a time. If a particular Hosted Service is not running at some particular time, then there would not be any role instances for any of its roles. The term "role instance" is also sometimes used to

	refer to the entire VM instance that hosts a single role instance. Role instances typically correspond one-to-one with internal/NAT'd Windows Azure IP addresses.
root OS	A hardened operating system that runs in the first VM on a compute node, and hosts the fabric agent. This reduced-footprint operating system includes only those components necessary to host VMs. This is done both to improve performance and to reduce attack surface.
SMAPI (service management API)	The Hosted Service that implements the programmatically accessible API to Windows Azure customer developers. Windows Azure developers access SMAPI using the REST protocol running over SSL-authenticated with a certificate provisioned using the Windows Azure Portal.
subscription	A Windows Azure account set up by a customer to aggregate the billing associated with a collection of Hosted Services and Storage accounts.
VHD (virtual hard disk)	An image file that stores operating systems, customer software, and temporary state in a unitary format that mirrors a single computer hard disk.
VIP (virtual IP address)	An externally visible IP address through which clients communicate with services hosted on Windows Azure. The VIP is implemented by load balancers, which allocate communications to specific endpoints (primarily, roles).
VM (virtual machine)	A software-only computer emulation running within a virtual memory manager (VMM, or hypervisor) that behaves as if it is a physical computer.
Windows Azure hypervisor	(see *Hypervisor*)
Windows Azure Portal	Customers manage Hosted Services and Storage Accounts through the Windows Azure Portal web site.
Windows Azure Drive	Windows Azure Drive provides a durable NTFS volume for Windows Azure VM instances to mount and use. The Windows Azure Drive is actually a blob, where all writes to the drive are made durable to the Storage Account's blob. If the VM with the mounted drive fails over, then the drive still exists as a blob and it can be remounted elsewhere without loss of data. The octets in an Windows Azure Drive are typically formatted like an NTFS image on a physical disk, and Windows Azure VMs can mount them as disks and access them as file systems. The Windows Azure code aggressively caches the data from the Windows Azure Drive on its local disk to avoid a substantial performance penalty for reads. While Storage blobs, tables, and queues are designed to be open and updated by multiple independent VMs, a Windows Azure Drive can only be mounted read-write by a single VM, but snapshots of Drives can be mounted read-only by any number of VMs - , making it more difficult to update for distributed replicated processes. Windows Azure Drives exist primarily for compatibility with applications that are designed to natively access NTFS volumes, and to simplify durable migration of state for single-master roles.