SHAMAN SECURITY

Shaman understands that the confidentiality, integrity, and availability of our customers' presentations are vital to their business success. Shaman takes security very seriously. We use the best engineering practices and tools to build our platform and integrate the highest security standards. All your content & data are backed up, stored securely and protected. This document provides detailed information on Shaman security on different levels:

- ShamanCloud, the online presentation management application
- Shaman App for iOS
- Cloud platform and hosting by Microsoft Azure

ShamanCloud

Validation

To access ShamanCloud, the user must log in first. Although the admin user of a company creates new users, the password to this login is never created by anyone other than the user itself, including the admin user. All of these users receive an email with a link that they can use to set a password of their preference, which only they will be aware of.

Password

The password to Shaman (both for ShamanCloud and Shaman App) goes through a 2-step procedure of encryption.

- The password is first concatenated with a random salt. This long string is then encrypted using SHA 256 hash function.
- All passwords on Shaman require passwords to include a minimum of 7 characters and must include alphanumeric characters.

The Shaman password expires every 6 months; the user then needs to take steps from shaman cloud to update their passwords accordingly.

When a user forgets their password, they just need to enter the email (advise: business email) they are signed up to Shaman with, instructions are then sent to this email if it is valid on how to reset the password.

Finally, the SHA 256 function is known to be a virtually irreversible hashing technique and so no Shaman employees or partners will be able to log into ShamanCloud with getting permission and authorization from the company itself.

Preventing SQL Injection

ShamanCloud uses the escape technique where characters and commands that have special meaning in SQL are escaped and everything entered is treated as plain text.

SSL

ShamanCloud integrates Digicert's EV (extended validation) certificate. More on what this certificate offers in terms of security and trust can be found here: https://www.digicert.com/ev-multi-domain-ssl.htm

Cookie Policy

ShamanCloud uses a policy of not using cross browser cookies. The cookie does not hold any username password information but currently does hold the user's session id.

Isolated database per company

When the Shaman administrator creates a new company, a new database itself is created for that particular company. This way, data of one company cannot be accessed by another incase of any incorrect queries being run as they are run on their independent databases.

Sessions

When a user logs in, they are provided with a session id, the user remains logged in as long as this session id is valid. Once the user logs out or does not visit shaman cloud for a period of time, the session id expires and the user needs to log in again.

Shaman API

The Shaman Api can only be called using an api token. The entity that uses the api currently is the Shaman App. The application can only get a token once a user logs in. Each api request must have the token as a parameter to it.

URL Protection

Links to resources on the Shaman service are encoded in base64 this makes sure that malicious users cannot guess links to resources for online handouts.

Privacy Policy

Shaman does only store the following information of Shaman users (salesmanagers, projectmanagers of the subscribed company): name, business email and phone number. Users are created and maintained by the admin user of the company, how has the responsibility for this information. The Shaman administrator cannot edit or access this information. ShamanCloud does only store the salution name for clients and prospects of the company that receive a handout. This name cannot be linked in any way to other data like email addresses or phone numbers of that same person.

Recovery

ShamanCloud uses Microsoft Azure's Geo Redundant data containers as a means of recovery (see more below on Microsoft Azure). This data is currently in the Europe East server (based in the Netherlands) and the replication is at the Europe North Data Center (based in Ireland).

A transaction is fully replicated on three different storage nodes across three fault domains and upgrade domains inside the primary location, then success is returned back to the client. Then, in the background, the primary location asynchronously replicates the recently committed transaction to the secondary location.

That transaction is then made durable by fully replicating it across three different storage nodes in different fault and upgrade domains at the secondary location. Because the updates are asynchronously geo-replicated, there is no change in existing performance for your storage account, though transactions are typically geo-replicated within a few minutes after they have been committed in the primary location.

ShamanCloud also operates a backup process of its own using Duplicity. This softwaretool fully automates the process to backup data to a storage container. Duplicity also supports incremental backups where backups can be made only of changed files in the system this helps conserve storage space.

Shaman will also give it's paying customers the option to export their library of slides (JPG).

Presentation Access

Presentations on ShamanCloud are accessible on successful login, this allows authorized users to view presentations here. Although, once handouts are shared, anyone with the handout url will be able to access it. Although as mentioned before, the parameters of the URL are encoded so it is not easy to regenerate this url as the server expects certain parameters.

User Access

There are a few different users on ShamanCloud:

- Admin: Can CRUD presentation / user data and read statistics
- Project Manager: Can CRUD presentation data but only read user data and statistics
- Project Manager + Sales Manager: Can CRUD presentation data but only read user data and statistics
- Sales Manager: Can read statistics

These roles are enforced so to perform certain actions, the right privileges are necessary.

Prevention of Hacking

Microsoft Azure offers a variety of technologies to help prevent hacking. A combination of other previously mentioned techniques of escaping, hashing, salting, using tokens, using an SSL certificate and data encryption where necessary are different techniques ShamanCloud uses to prevent hacking.

Shaman App

Validation

Like ShamanCloud, Shaman App also requires the user to login in order to gain access to the presentations. The password created for ShamanCloud can be used to access Shaman App as well.

Password

Login is performed via the API, the password is salted, encrypted and is sent via a secure channel. The API sends back a token, which is stored in the app database. Same password policies and rules apply to the Shaman App password as the password for ShamanCloud.

App Database

Once the user logs in successfully, Shaman App stores some user and company data on the local app database. Since offline login is a necessity, storing the password and the token is necessary.

The password is again encrypted and salted before being stored so it offers the same protection as the password stored in the cloud. The original password needs to be known in order to login and the hashed string cannot be used to login locally.

The token is stored so that the user can access the API on a regular basis. If the user logs out from the app, the token is destroyed and needs the user to log in again in order to get another token and access other API functions.

Only Shaman App is able to retrieve data and send information via the API, as it is currently the only entity that can retrieve a token from the API and a token is required in order to use the API.

Creating Handout URLs

Handout URLs require parameters to identify several pieces of information on the server side. Data here is encoded in base 64 to make the URLs unpredictable to replicate.

Uploading Meeting Data

When the device has internet connection the app can upload meeting data to the server. Each meeting is differentiated from the next using a hash of the current date and time, user id and presentation id, information that is not available outside the application environment.

iOS App Sandbox

For security reasons, iOS places each app (including its preferences and data) in a sandbox at install time. A sandbox is a set of fine-grained controls that limit the app's access to files, preferences, network resources, hardware, and so on.

As part of the sandboxing process, the system installs each app in its own sandbox directory, which acts as the home for the app and its data.

This way, an app cannot access files and data outside of it's own sandbox and likewise, other apps cannot access data inside the apps own sandbox.

IPA file

The file that installs on the app is an IPA file, which is an executable that contains all the required resources. The ipa file obfuscates the code and it cannot be reengineered easily.

Using Native Functions

The Shaman App mostly makes use of native iOS functionality and passes on responsibility to the iOS wherever possible so as to be protected by the security offered by apple. One major instance is making use of mailing provided by the device itself so that mails are sent using the mailing application set up by the user.

No Access to App Database

The app database cannot be retrieved currently as it sits in the app's documents directory that sits in the app sandbox. Auto synching to the iPad's data stores (e.g iCloud) is also disabled so no data or content can be retrieved from the app's sandbox, only can be viewed through the app. Once the app is deleted, the content and data in the sandbox is also deleted.

Microsoft Azure

Cloud Security Design

As the application hosting platform, Microsoft Azure provides confidentiality, integrity, and availability of customer data. It must also provide transparent accountability to allow customers and their agents to track administration of applications and infrastructure, by themselves and by Microsoft.

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.

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.

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.

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.

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.

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.

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.

Access Control in Windows Azure Storage

Microsoft 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.

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.

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.

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.

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**).

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.).

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.

Encryption

Encryption of data in storage and in transit can be used by customers within Microsoft 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.

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.

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 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.

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.

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.

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.

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.

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.

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.

Media Disposal

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

Resource Security

While the concern for DOS attacks from within the Data center in a real threat, Azure over comes this by using the fabric controller, “The fabric controller will make sure that all of the VMs deployed into the Azure fabric will get the resources they paid for. It’s not possible for an infected VM to consume unlimited resources and starve its neighbours.”

There are also additional layers designed to reduce the risk of a distributed denial of service (DDOS) attack. Azure has many custom built and proprietary method in-place to protext against DDOS attacks from the outside, they have standard mitigation processes in place and check for DDOS attacks from within the network and shut down the offending VM.

Geo-Replication

Transactions are typically geo-replicated within a few minutes after they have been committed in the primary location. This is mostly for recovery services for machines.

EU Policy

The E.U. Data Protection Directive (95/46/EC) contains strict requirements for the handling of personal data in the European Union. Under European law, our customer is the data controller of its Customer Data and Microsoft is the data processor. To allow for the flow of information required by international business (including cross border transfer of personal data), Microsoft adhere to the U.S.-EU Safe Harbor Framework developed by the Department of Commerce in coordination with the European Commission. The Safe Harbor certification allows for the legal transfer of E.U. personal data outside the E.U. to Microsoft for processing.

Microsoft also offers customers E.U. Standard Contractual Clauses that provide additional contractual guarantees around transfers of personal data for in-scope services. Microsoft's implementation of the E.U. model clauses has been validated by European Union data protection authorities as being in line with the rigorous privacy standards that regulate international data transfers by companies operating in its member states. Microsoft is the first company to receive joint approval from the E.U.'s Article 29 Working Party for its strong contractual commitments to comply with E.U. privacy laws no matter where data is located.

It is important to note that Microsoft will transfer E.U. Customer Data outside the E.U. only under very limited circumstances.

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.
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.
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.
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
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.
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.
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 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.