## Table of Contents

### INTRODUCTION

LOGICAL SECURITY ........................................................................................................ 3
- System Architecture .................................................................................................. 4
- Jitterbit Studio .......................................................................................................... 5
- Harmony Cloud Platform .......................................................................................... 6
- Management APIs and Web Management Console ...................................................... 8
- Users, Organizations and Roles .................................................................................. 8
- Environments and Control .......................................................................................... 9
- Cloud Agent Group .................................................................................................... 9
- Local Agent Group ...................................................................................................... 10
- Jitterbit Data Loader .................................................................................................. 11
- Data Storage .............................................................................................................. 11
- Security Topologies .................................................................................................... 13

PHYSICAL SECURITY ...................................................................................................... 15
- Infrastructure Compliance .......................................................................................... 16
- Physical and Environmental Security ......................................................................... 16
- Business Continuity Management .............................................................................. 17
- Secure Network Architecture .................................................................................... 19
- Transmission Protection ............................................................................................. 19
- Network Monitoring and Protection .......................................................................... 19
- Secure Design Principles ......................................................................................... 21
- Change Management ................................................................................................ 21
- Amazon Elastic Compute Cloud (Amazon EC2) Security ............................................. 22
- Multiple Levels of Security ....................................................................................... 22
- The Hypervisor .......................................................................................................... 22
- Load Balancing Security ........................................................................................... 23
- Data Storage .............................................................................................................. 23
- Data Durability and Reliability .................................................................................... 23

ORGANIZATIONAL SECURITY ......................................................................................... 24
- Confidentiality ............................................................................................................ 24
- Personnel Policy .......................................................................................................... 24
- Jitterbit Operations ..................................................................................................... 24
- Jitterbit Engineering ................................................................................................... 25
- Jitterbit QA Team ........................................................................................................ 25
- Jitterbit Harmony Trust Site ....................................................................................... 25
- Identity and Access Management ............................................................................... 26
- Incident Management ............................................................................................... 26
- Patch Management ..................................................................................................... 26
- Capacity Management ............................................................................................... 27
INTRODUCTION

Jitterbit Harmony delivers powerful integration tools and services through Jitterbit’s Multi-tenant Cloud Integration Platform called Jitterbit Harmony. Previous versions of Jitterbit required organizations to deploy and manage Jitterbit software and integration projects on internal network infrastructure or dedicated private clouds. With Jitterbit Harmony, Jitterbit can now assume any or all of these responsibilities.

While Jitterbit Harmony can drastically simplify most aspects of managing integration processes, the introduction of a multi-tenant cloud system raises security questions for customers and users. This document describes the following about Jitterbit’s security:

1. Logical Security
2. Physical Security
3. Organizational Security

This document takes a broad view of security and includes high availability and performance in addition to data protection.

This document reflects the Spring 14 version of Jitterbit Harmony. It is expected that Jitterbit will continually strive to improve its offering in these dimensions and so readers should always check that they are referencing the latest version of this document for the most current information.

LOGICAL SECURITY

Logical security is comprised of all the security measures taken within the Jitterbit software. This section will describe the following:

1. System Architecture
2. Major Components and Security
3. Jitterbit Harmony Users, Organizations, Roles
4. Jitterbit Harmony Environments and Access Control
5. Jitterbit Harmony Data
6. Jitterbit Harmony Security Topologies
Jitterbit enables users to design, test, deploy, run and manage their Jitterbit Integration Projects. Using Jitterbit Harmony, customers can run their Jitterbit integration processes completely in the cloud without the need to procure or manage any software or the infrastructure required to operate it.

Jitterbit recognizes that many customers and users need their integration processes to communicate with applications that operate behind corporate firewalls for various security and regulatory compliance reasons.

Jitterbit’s System Architecture caters for both scenarios: integration processes can run completely in the cloud, or integration processes can run inside corporate firewalls and ensure that no business data is exposed to the cloud. Users can also employ hybrid models where some integrations run in the cloud and others are behind firewalls.

While the system simplifies the provisioning, deployment and management of integration projects in Jitterbit Harmony, a multi-tenant cloud service, it also offers users the flexibility to run their integration operations using detachable Local Agent Groups. These are self contained...
subsystems that can be installed anywhere:

- Behind corporate firewalls
- On dedicated private clouds

The separation of integration designs, that are stored on Jitterbit Harmony, from integration runtime, that occurs on Agent Groups, enable customers to control access and flow of sensitive business data.

The following describes each major component from the system architecture and information pertaining to its security.

**Jitterbit Studio**

The Jitterbit Studio exposes the richest set of functionality for configuring and testing Jitterbit Integration Projects.

![Diagram of Jitterbit Studio](image)

The Jitterbit Studio is a Java Stand-alone client that can be installed on most Windows, Mac and Linux workstations. It simply requires a Java Runtime Environment and access to the internet on the workstation or laptop that it runs on. Communication through corporate proxy servers is fully supported.

Before a user can start their work, then have to authenticate with Jitterbit Harmony using their Jitterbit Harmony user account. All communication with Jitterbit Harmony occurs over HTTPS.

Once authenticated, Jitterbit Harmony identifies all the organizations and environments that this user has access to. Jitterbit Harmony provides the user with a list of the integration assets they can work on and allows the user to create a new project in any environment where they have sufficient privileges.
The Jitterbit Studio captures integration configuration data in local project files. Each project will have its configurations stores in a local folder on the disk drive where the Jitterbit Studio runs. If the local project files are lost then downloading the project from Jitterbit Harmony will re-create them.

Jitterbit Studio will erase any credential configuration from the local project files once an end-point or project has been deployed to Jitterbit Harmony. This is to minimize the duration that end-point credentials reside on local Jitterbit Studio project files.

If an intruder were to access the user’s local project files and they could decipher the XML project configuration files, they would gain access to a project’s configurations. They would be able to understand the process logic and data transformation logic for a given project. Provided that the end-points, sources and targets have been deployed to Jitterbit Harmony, the intruder would not have access to any credentials to applications.

The intruder would also not be able to access Jitterbit Harmony, as they would need valid credentials for a given user that has access to the various integration assets.

**Jitterbit Harmony Cloud Platform**

Jitterbit Harmony includes Multi-tenant databases, Files, Services and a Service Messaging infrastructure that are used to deploy, manage and run integration projects. Jitterbit Harmony runs on Amazon Web Services.

**Project Repository**

Jitterbit Harmony stores all projects deployed in a multi-tenant project repository. Jitterbit backups these projects, which enables customers to download older versions of the project from Jitterbit Harmony. This project repository is built on a multi-tenant database architecture, which provides logical partitioning of projects by organization, and in most instances further, by environment.

Projects typically contain the credential information such as username/password used to connect to various end-points such as SAP, Oracle EBS. This information is stored in an encrypted manner in the multi-tenant repository.

The repository is replicated across regions using a Jitterbit developed multi-master algorithm to ensure data is replicated in almost real-time for high-availability. Each database is also backed up and can be restored if ever needed by the Jitterbit Operations Team.
Customers do not have any direct access to this repository. All Jitterbit components such as the Studio and Agents use APIs to access the repository. Once authenticated and access control is validated, all communication with the repository is done through various API layers. In addition to controlling edge API access via HTTPS and server side sessions, most APIs must validate user access control through environment-based and role-based Access Control Lists. These ensure that users can only view, act and change the system based on permissions granted by their organization’s administrator.

**Activity Database**

The Jitterbit Harmony rotating activity database stores all the run-time status information as well as logs of all running operations of all Jitterbit users.

The activity database is built on a multi-tenant architecture, and while activity data for all users resides in the same database, there is logical segmentation by organization and environment applied through software access control layer to ensure that users can only view activities that they have access to.

The activity database is also replicated across regions to ensure high-availability and is also backed up if there is a need for it to be restored.

Customers will not have any direct access to this activity database and all access is provided by a set of APIs. The Activity Log uses similar APIs and ACL infrastructure as the Project Repository.

**File Services**

Jitterbit Harmony includes a set of File services used to store files such as schemas, and customizations. All files stored in AWS S3 service and can only be accessed through Jitterbit Harmony’s software and cannot be access directly from S3 by any user.

**Schema Repository**

To support integrations from a variety of end-points Jitterbit Harmony needs to store various types of schemas such as WSDL, XSD, JTR, DTDs.

**Customizations Repository**

To support integrations from a variety of database end-points Jitterbit Harmony needs to store various types of drivers such as JDBC and ODBC drivers. Jitterbit also provides a framework where customers can customize the Jitterbit operations using “Plug-ins”. 
Plug-ins are designed to be plug-n-play and customers using their own custom Plug-ins are responsible for the security of the Plug-ins.

**Run-time Messaging Services**

Communication between various Jitterbit components such as the Jitterbit Studio, the Agents and Jitterbit Harmony is done using a set of messaging services based on the Java Messaging Services (JMS) API services included in the Java Platform. These APIs are internal to Jitterbit components and customers do not have any access to these APIs. Jitterbit Agents include listeners to this messaging service. All Agents' that listen for requests have to authenticate and have a valid session in the Jitterbit Harmony messaging network. They can only listen to requests for their particular Agent Groups or that are made to them directly via Jitterbit Harmony. Messages are never sent to Agents; Agents always pick them up over HTTPS. This enables Agents to run behind corporate firewalls and to remain protected without the need for opening ports that would allow incoming traffic from the Internet.

**Management APIs and Web Management Console**

The Jitterbit Web Management Console communicates with Jitterbit Harmony through a well-defined set of Management APIs. All users of these APIs must be authenticated with Jitterbit Harmony and all communication is handled securely over HTTPS.

All the management functions provided to the WMC user are further controlled by an Access Control Layer model defined for each environment. As a result, any user using the WMC will be able to see only the data for which he has the permission to do so. Access controls are applied to any and all functions including searching for operation information, running operations, viewing logs etc.

**Users, Organizations and Roles**

In order to access Jitterbit Harmony, a user must provision their user account. Every User created within Jitterbit Harmony has their own personal account where personal integration tools and projects can be stored and kept private.

Users can create Organizations when they need to work in teams to design, build, test and manage their integration projects. Every organization has a role called “Administrator” that can access all assets belonging to the Organization.

Administrators can add new Roles to the Organizations and invite other Jitterbit Harmony users to join.
Environments and Control

All projects deployed to Jitterbit Harmony are deployed to Environments. Environments represent a given state of an integration project. Many projects exist in various different stages within different environments e.g. a common project lifecycle configuration would have three environments: “Development”, “Test”, “Production”. A project can exist in different states within each environment.

Organization Administrators would manage access control to each environment based on Role. For example, users in the “Developer” Role may have Read, Execute and Write privileges in the “Development” environment, but only “Read” access to “Test” and no access to “Production”.

The access levels for an environment include are:

Read Access

Only allows a user to read a project from a given Environment. This can be used to share project templates with various users, or to allow them to view but not affect projects deployed to critical environments such as Test or Production.

Execute Access

This access provides Read access and allows users to run operations within a given environment. This is a common access control for “Test” environments and is often granted to users who need to support an integration as they will need to test, run and view operation logs.

Write Access

Write access provides full control to a given environment. Users belonging to a Role with Write access can read, test, run and change projects within an environment.

Jitterbit Cloud Agent Group

Jitterbit provides its customers the option to run all their integrations in the cloud by providing a scalable, fault-tolerant clustered agent group maintained and managed by Jitterbit. This Jitterbit Cloud Agent Group consists of a set of Agents managed by Jitterbit. The number of Agents scales elastically based on the workload on the Cloud Agent Group Queue.
Users cannot control or configure the Jitterbit Cloud Agent Group or its Agents as they do their own Local Agent Groups. They have no direct access to any Agent within the Jitterbit Cloud Agent Group.

Jitterbit agents do not persist permanently any customer data locally when processing the operations.

When the Jitterbit Cloud Agent Group performs an integration, it will connect directly with the application that requires data integration. It will read and post data to these applications. It can also persist the user’s business data in temporary files (that persist for 24 hours) or the debug logs (that persist for 14 days).

Data persisted in the Jitterbit Cloud Agent Group is stored in S3 buckets that are not accessible to users directly. Each integration stores data in its environment’s bucket.

The Jitterbit Cloud Agent Group is not suitable for companies that need to access data and applications that reside within their firewall, or users that must perform integration under strict regulatory compliance that forbids data to travel or reside in the cloud.

**Local Agent Group**

Jitterbit provides the flexibility for customers to provision and manage their own Agent Groups and Agent within their corporate firewall or virtual private clouds. This allows customers to choose where their integration run-time environment operates and let’s them control which network their business data travels and resides in. Using Local Agent Groups, integrations can ensure that business data never flows through the cloud if there is internal access to applications that don’t operate in the cloud.

Jitterbit Agents belonging to Local Agent Groups authenticate and communicate with Jitterbit Harmony over HTTPS. Local Agent Groups deployed behind corporate firewalls can be configured to communicate via a corporate proxy server. There are no additional networking requirements such as opening ports within corporate firewalls.

While Local Agent Groups cater for stringent security requirements, the user or customer is responsible for installing and managing their on-premise agents. While Jitterbit Harmony provides out-of-the-box high availability and scalable configurations for these, the actual services run on the users server or virtual machines.

Users can host HTTP End-Points and SOAP End-Points designed in Jitterbit Studio on their Private Agents. This allows internal applications to trigger events that can make HTTP requests to Jitterbit.
For production Local Agent Groups that can consist of multiple agents, it is the responsibility of the user or the administrator to setup internal load balancing for managing the distribution of internal API requests to these Agents. The Agents run standard Apache Web Server through a service called Jitterbit Apache. Most load balancers that distribute traffic across Apache and other Web Servers will work.

**Jitterbit Data Loader**

The Jitterbit Data Loader is a free product from Jitterbit targeted towards Salesforce.com customers, which provides a limited set of functionality. It allows customers to move data in and out of salesforce to flat files or databases. The Jitterbit Data Loader product is built on the same underlying Jitterbit Harmony Platform.

In addition to limited functionality, the Jitterbit Data Loader installs a limited Local Agent. The agent is installed on the same machine where the Studio is installed and only supports one Environment and one Local Agent. It is not meant for scalable, highly available projects.

The Jitterbit Data Loader has been deployed in a production version of Jitterbit Harmony since summer of 2013. It has been adopted by thousands of users and has been used to test the security, scale and availability of Jitterbit Harmony in supporting thousands of users.

**Jitterbit Harmony Data Storage**

The following describes the type of information stored in Jitterbit Harmony:

**User Data**

When a user registers and subscribes to Jitterbit Harmony they have to provide the following information, which is stored in the Project Repository: First Name, Last Name, E-Mail and Phone Number. The following information can be supplied but is optional: Company, Company Address, Company Website.

**Project Data**

In order to run and manage an integration project, a user must deploy that project to Jitterbit Harmony. The project stores design and implementation details to instruct Agent Groups what activities they need to perform. This includes the following:

- Integration Operations that describe what a unit of integration will do, for example, synchronized all changes to customer data in the CRM system with customer data in the ERP system
• Transformations and Scripts that describe how that data is transposed from the source system to the target system. This includes any validation rules or data manipulation required to transfer the data successfully

• Interfaces that describe the various source and target object structures. These interfaces can be simple text structures or complicated XML, JSON or EDI object representations

• Sources, Targets and Connector End Points that describe how to access a particular system. While these can be hard coded values including system addresses and credentials, they can also be referenced through variables that can be stored in internal databases for customers that implement their own credential vaults.

• Schedules and Notifications that determine when batch operations need to run and what to do in the event of successful and failed outcomes.

• API End Points that inform Agents and Agent Groups how to expose APIs so that external system events can call and invoke Jitterbit integrations.

**Integration Activity Log**

When an Agent Group runs an integration operation, it synchronizes its logs to the Jitterbit Harmony Multi-tenant Rotating Activity Log. This includes the following information:

- **Status** - the state of an operation (e.g. pending, running, successful, failed).
- **Agent** - which Agent in the Agent Group ran the operation
- **Timing** - when the operation run was submit, started and completed
- **Submitted By** - who submitted the request to run the operation
- **Records Processed** - the number of records processed from the source system and how many records were posted to the target
- **Message** - any additional information that is relevant for troubleshooting a failed outcome, or summary information that a user explicitly tells Jitterbit to write to the log using the internal function “WriteToOperationLog()”

The Activity Data is stored in the cloud on a rotating daily partitioned system. The activities for each day are captured in that day’s partition and each partition is dropped after 31 days. So activity log data older than 31 days is permanently removed.

**Agent Logs**

Each Agent can generate additional data that can either be accessed via Jitterbit Harmony or can be stored on local customer file storage devices such as File Shares and SFTP and accessed from within a firewall. These are detailed logs, which include:
• Debug Logs
• Files of Successful Records Processed
• Files of Failed Records Processed

The Agent Groups and Agents do not automatically synchronize these with Jitterbit Harmony as they typically include business data. By using their own storage devices, customers can keep that data within their firewall or on their own private cloud infrastructures.

These logs are useful for detailed troubleshooting and auditing purposes. By default, an Agent Group will store this for 1 to 14 days. The Agent Group can be configured to cleanup this data at other intervals.

Test Data that flows through Jitterbit Harmony

In addition to data stored in Jitterbit Harmony, business data can flow through the cloud platform during integration design. This data is not persisted in any Jitterbit Harmony storage device. This data flows when performing functions such as:

• Load Source Data - this brings sample data into a transformation tree to assist a user in identifying elements in an interface and test transformations.
• Test Transformation - this shows the transformed target result for a given set of data that is loaded.
• Test Transformation Function or Script - this allows a user to test functions and script that could include a database select statement or view variables values returned from a web service API.
• Test Web Service Call and Test Operation - allows the user to run an integration and view all results on the screen.

The Jitterbit design services also enforce a limit of 100 KB on all test data, so this would limit data that flows through the cloud to a very small subset of data.

Jitterbit Harmony Security Topologies

Users should consider employing one of the following topologies to meet their data protection and security requirements:
1. Jitterbit Cloud Agent Group Topology

Customers who need to perform integrations where all their data sources are accessible via the cloud can deploy their projects to Jitterbit Harmony Environments and run their projects on the Jitterbit Cloud Agent Group.

With exception to the Jitterbit Studio that runs on the users laptop or desktop, this topology has no software footprint in the customer’s network and everything is readily available for immediate use in the cloud.

Here, the Jitterbit operated Multi-tenant Public Agent Group will access customer business data directly over the Internet. Jitterbit Agents within this Agent Group will process business data and post it to any required target system. The data will flow within the Jitterbit network. All data mentioned above could persist in Jitterbit Harmony’s secure operating environment for a period of time.

Customers that have more stringent security policies, or require excessive liability and indemnification guarantees should validate that Jitterbit Cloud End User License agreement, Privacy and Security policies comply with their needs pertaining to their industry regulations, customer provisions, customer indemnification and customer liabilities terms.
2. **Local Agent Group Topology**

In most enterprise integration scenarios, the Agent Group has to access internal as well as cloud applications. Here, users would deploy their projects to Jitterbit Cloud Environments, install their own Local Agent Groups within their networks that have access to their applications, and then manage those Agent Groups via Jitterbit Cloud.

This topology enables users to provision and manage their Agent Groups using Jitterbit Harmony, but the Agent Group and any sensitive business data that it processes or persists reside within their network.

In this topology, the Local Agent Group can run on most Windows or Linux physical or virtual server environments.

**PHYSICAL SECURITY**

Jitterbit Harmony is hosted on AWS cloud infrastructure. Jitterbit chose AWS as it provides a platform that addresses Jitterbit Harmony’s scalability, availability and many of its security requirements.
**Infrastructure Compliance**

The IT infrastructure that AWS provides is designed and managed in alignment with best security practices and a variety of IT security standards, including:

- SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70 Type II)
- SOC 2
- SOC 3
- FISMA, DIACAP, and FedRAMP
- PCI DSS Level 1
- ISO 27001
- ITAR
- FIPS 140-2

In addition, integration projects deployed on Jitterbit Harmony can be configured to meet several industry-specific standards, including:

- HIPAA
- Cloud Security Alliance (CSA)

**Physical and Environmental Security**

Jitterbit Harmony is deployed across AWS data centers that are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means. Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff.

AWS only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee Amazon Web Services. All physical access to data centers by AWS employees is logged and audited routinely.

**Fire Detection and Suppression**

Automatic fire detection and suppression equipment has been installed to reduce risk. The fire detection system utilizes smoke detection sensors in all data center environments, mechanical
and electrical infrastructure spaces, chiller rooms and generator equipment rooms. These areas are protected by either wet-pipe, double-interlocked pre-action, or gaseous sprinkler systems.

**Power**

The data center electrical power systems are designed to be fully redundant and maintainable without impact to operations, 24 hours a day, and seven days a week. Uninterruptible Power Supply (UPS) units provide back-up power in the event of an electrical failure for critical and essential loads in the facility. Data centers use generators to provide back-up power for the entire facility.

**Climate and Temperature**

Climate control is required to maintain a constant operating temperature for servers and other hardware, which prevents overheating and reduces the possibility of service outages. Data centers are conditioned to maintain atmospheric conditions at optimal levels. Personnel and systems monitor and control temperature and humidity at appropriate levels.

**Management**

AWS monitors electrical, mechanical, and life support systems and equipment so that any issues are immediately identified. Preventative maintenance is performed to maintain the continued operability of equipment.

**Storage Device Decommissioning**

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent customer data from being exposed to unauthorized individuals. AWS uses the techniques detailed in DoD 5220.22-M ("National Industrial Security Program Operating Manual ") or NIST 800-88 ("Guidelines for Media Sanitization") to destroy data as part of the decommissioning process. All decommissioned magnetic storage devices are degaussed and physically destroyed in accordance with industry-standard practices.

**Business Continuity Management**

Jitterbit Harmony leverages AWS’s infrastructure to provide very high levels of availability. AWS has designed its systems to tolerate system or hardware failures with minimal impact.
Availability

Data centers are built in clusters in various global regions. All data centers are online and serving customers; no data center is “cold.” In case of failure, automated processes move customer data traffic away from the affected area. Core applications are deployed in an N+1 configuration, so that in the event of a data center failure, there is sufficient capacity to enable traffic to be load-balanced to the remaining sites.

Jitterbit Harmony is deployed across three geographic regions: US West, US East and EU. It is deployed in an availability zone within each region. Each availability zone is designed as an independent failure zone. This means that availability zones are physically separated within a typical metropolitan region and are located in lower risk flood plains (specific flood zone categorization varies by Region). In addition to discrete uninterruptable power supply (UPS) and onsite backup generation facilities, they are each fed via different grids from independent utilities to further reduce single points of failure. Availability zones are all redundantly connected to multiple tier-1 transit providers.

This provides high levels of resiliency for Jitterbit Harmony as it can tolerate most failure modes, including natural disasters or system failures without shutdown.

In the United States, in the event of a widespread catastrophic outage, we also can route all traffic destined for the problematic datacenter to a datacenter on the opposite coast.

Incident Response

Jitterbit’s Operations and Customer Support team work to identify any issues that may impact Jitterbit Harmony’s users. We monitor Jitterbit Harmony’s API usage, Databases, Services, Messaging Infrastructure and Jitterbit Cloud Agent Groups. Our support and operations team provide 24x7x365 global coverage to detect any critical issues and manage the impact and resolution of those incidents.

Jitterbit Harmony’s infrastructure is supported by the Amazon Incident Management team, which employs industry-standard diagnostic procedures to drive resolution during business-impacting events. Staff operators provide 24x7x365 coverage to detect incidents and to manage the impact and resolution.

Communication

Jitterbit implemented various methods of internal communication at a global level to coordinate all critical communication across Jitterbit’s Operations, Customer Support, Engineering, QA and
Service Teams. These teams work across the globe with presence across the US, Asia and Europe. Our employees understand their individual roles and responsibilities and know when to communicate significant events in a timely manner.

Jitterbit has standard daily meetings between the various teams, which includes team managers and company officers, to highlight any known issues and ensure that there are no bottlenecks within the organization preventing fast resolution.

**Network Security**

Jitterbit Harmony resides within the AWS network that has been architected to provide the level of security and resiliency required for Jitterbit Harmony to support high service levels. Jitterbit Harmony is geographically dispersed, with a fault-tolerant architecture supported in all core services. Jitterbit Harmony relies on AWS world-class network infrastructure that is carefully monitored and managed. This includes:

**Secure Network Architecture**

Network devices, including firewall and other boundary devices, are in place to monitor and control communications at the external boundary of the network and at key internal boundaries within the network. These boundary devices employ rule sets, access control lists (ACL), and configurations to enforce the flow of information to specific information system services.

**Transmission Protection**

The only external communication with Jitterbit Harmony available is via HTTPS using Secure Sockets Layer (SSL), a cryptographic protocol that is designed to protect against eavesdropping, tampering, and message forgery.

**Network Monitoring and Protection**

Jitterbit Harmony leverages AWS utilization of a wide variety of automated monitoring systems to provide a high level of service performance and availability. AWS monitoring tools are designed to detect unusual or unauthorized activities and conditions at ingress and egress communication points. These tools monitor server and network usage, port scanning activities, application usage, and unauthorized intrusion attempts. The tools have the ability to set custom performance metrics thresholds for unusual activity.

Systems within AWS are extensively instrumented to monitor key operational metrics. Alarms are configured to automatically notify operations and management personnel when early warning
thresholds are crossed on key operational metrics. An on-call schedule is used so personnel are always available to respond to operational issues. This includes a pager system so alarms are quickly and reliably communicated to operations personnel.

Jitterbit operations and support teams work with engineering in handling any incidents or issues related to Jitterbit developed software or infrastructure. All critical issues are identified and discussed during daily calls between the teams. Post-mortems are documented after any significant operational issue, regardless of external impact, and Cause of Error (COE) reports are drafted so the root cause is captured and preventative actions are taken in the future.

Jitterbit Operations leverages AWS security-monitoring tools help identify several types of denial of service (DoS) attacks, including distributed, flooding, and software/logic attacks. When DoS attacks are identified, the AWS incident response process is initiated. In addition to the DoS prevention tools, redundant telecommunication providers at each region as well as additional capacity protect against the possibility of DoS attacks.

Jitterbit Harmony gains the benefits of the AWS network, which provides significant protection against traditional network security issues. The following are a few examples:

- Distributed Denial Of Service (DDoS) Attacks. Jitterbit Harmony API endpoints are hosted on large, Internet-scale, world-class infrastructure that benefits from the same engineering expertise that has built Amazon into the world’s largest online retailer. Proprietary DDoS mitigation techniques are used.
- Man in the Middle (MITM) Attacks. All of the Jitterbit Harmony APIs are available via SSL-protected endpoints, which provide server authentication.
- IP Spoofing. The Amazon EC2 instances that run Jitterbit Harmony cannot send spoofed network traffic. The AWS-controlled, host-based firewall infrastructure will not permit an instance to send traffic with a source IP or MAC address other than its own.
- Port Scanning. All Jitterbit Harmony EC2 instances inbound ports are tightly controlled by Jitterbit Operations. We have strict controls of security groups that further mitigate the threat of port scans. Jitterbit Harmony uses SSL for any open port, together with server side sessions, to protect listening services from discovery by an unauthorized port scan.

In addition to monitoring, regular vulnerability scans are performed on the host operating system, web application, and databases using a variety of tools.
Secure Design Principles

Jitterbit Harmony’s development process follows secure software development best practices, which include formal design reviews by the Jitterbit Security Team that validate Jitterbit software is designed and developed to prevent error messages from transmitting sensitive information and ensure that software services reject unauthorized access and misuse.

Change Management

Routine, emergency, and configuration changes to existing Jitterbit Harmony infrastructure are authorized, logged, tested, approved, and documented in accordance with industry norms for similar systems. Updates to Jitterbit Harmony’s infrastructure are done to minimize any impact on the customer and their use of the services. The Jitterbit Harmony Trust Site provides a public facing dashboard that lists any outages and system performance degradation periods.

Software

Jitterbit Engineering applies a systematic approach to managing change so that changes to customer-impacting services are thoroughly reviewed, tested, approved, and well communicated. The change management process is designed to avoid unintended service disruptions and to maintain the integrity of service to the customer. Changes deployed into production environments are:

- Reviewed: Peer reviews of the technical aspects of a change are required.
- Tested: Changes being applied are tested by a separate QA team to ensure they will behave as expected and not adversely impact performance.
- Approved: All changes must be authorized in order to be rolled out by Engineering, QA and Customer Support.

When possible, changes are scheduled during regular change windows. Emergency changes to production systems that require deviations from standard change management procedures are associated with an incident and are logged and approved as appropriate.

Infrastructure

Jitterbit Harmony runs inside a Virtual Private cloud (VPC) and includes the following services within each region:

1. Apache-Tomcat Elastic Load Balancer (ELB) that ensure requests to Jitterbit Harmony services and APIs scale and are highly available together with Apache-Tomcat Cluster
where Jitterbit Harmony services run. The number of nodes per cluster scales dynamically as request volumes scale up and down.

2. Terracotta Server cluster for managing user sessions. This is designed so that if any Terracotta server or service fails to function, the cluster has redundancy built in to ensure that a user’s session is not affected.

3. ActiveMQ broker network that manages requests for Agents. This ensures that there is a highly available redundant network between the Jitterbit Harmony and all Agents.

4. PostgreSQL database server with “near real-time” asynchronous replication across regions. This ensures that all Project designs and Activity Data is available across regions in the event that an entire region becomes unavailable.

AWS services are architected to work efficiently and securely with all AWS networks and platforms. Each service provides extensive security features to enable you to protect sensitive data and applications.

**Amazon Elastic Compute Cloud (Amazon EC2) Security**

Jitterbit Harmony makes extensive use of AWS Elastic Compute Cloud (EC2), which provides resizable computing capacity using server instances in AWS’s data centers.

**Multiple Levels of Security**

Jitterbit Harmony leverages the security within Amazon EC2 that is provided via: the virtual instance OS, firewall. External API access is only available on the Jitterbit Harmony HTTPS servers. All other services are protected behind the firewall.

**The Hypervisor**

Jitterbit Harmony Amazon EC2 currently utilizes a highly customized version of the Xen hypervisor, taking advantage of paravirtualization. Because paravirtualized guests rely on the hypervisor to provide support for operations that normally require privileged access, the guest OS has no elevated access to the CPU. The CPU provides four separate privilege modes: 0-3, called rings. Ring 0 is the most privileged and 3 the least. The host OS executes in Ring 0. However, rather than executing in Ring 0 as most operating systems do, the guest OS runs in a lesser-privileged Ring 1 and applications in the least privileged Ring 3. This explicit virtualization of the physical resources leads to a clear separation between guest and hypervisor, resulting in additional security separation between the two.

Each Jitterbit Harmony Virtual EC2 instance is controlled by the Jitterbit Operations Team. All
Jitterbit Harmony instances are hardened and utilize certificate-based SSHv2 to access the virtual instance. All key pairs are generated by Jitterbit Operations in order to guarantee that they are unique, and not shared outside Jitterbit Operations.

**Load Balancing Security**

Amazon Elastic Load Balancing is used to manage traffic on a fleet of Amazon EC2 instances. Elastic Load Balancing has all the advantages of an on-premises load balancer, plus several security benefits:

- Takes over the encryption and decryption work from the Amazon EC2 instances and manages it centrally on the load balancer
- Provides a single point of contact, and also serves as the first line of defense against attacks on your network
- Supports end-to-end traffic encryption. The SSL server certificate used to terminate client connections can be managed centrally on the load balancer, rather than on every individual instance.

**Data Storage**

Jitterbit Harmony uses Amazon S3 for file data storage. This data includes transformation schemas, database Drivers, Plugins and in certain cases, temporary and log files.

Jitterbit Harmony uses the Amazon S3 Encryption Client to encrypt data before uploading to Amazon S3. Amazon S3 uses one of the strongest block ciphers available – 256-bit Advanced Encryption Standard (AES-256). With Amazon S3, every protected object is encrypted with a unique encryption key. This object key itself is then encrypted with a regularly rotated master key. Amazon S3 provides additional security by storing the encrypted data and encryption keys in different hosts.

**Data Durability and Reliability**

Amazon S3 is designed to provide 99.999999999% durability and 99.99% availability of objects over a given year. Objects are redundantly stored on multiple devices across multiple facilities in an Amazon S3 region. To help provide durability, Amazon S3 PUT and COPY operations synchronously store customer data across multiple facilities before returning SUCCESS. Once stored, Amazon S3 helps maintain the durability of the objects by quickly detecting and repairing any lost redundancy. Amazon S3 also regularly verifies the integrity of data stored using checksums. If corruption is detected, it is repaired using redundant data. In addition,
Amazon S3 calculates checksums on all network traffic to detect corruption of data packets when storing or retrieving data.

**ORGANIZATIONAL SECURITY**

Jitterbit strives to apply the operational best practices of leading cloud-computing providers around the world. This includes the following:

**Confidentiality**

Jitterbit Harmony’s confidentiality measures work to protect sensitive customer data from unauthorized access. In addition to the physical and logical security layers provided by our software and physical infrastructure, our internal policies dictate:

- All access to Jitterbit Harmony’s production system is only available to the Jitterbit Operations team. All work to the production environment must be applied by the Jitterbit Operations team.
- Within the Jitterbit Operations Team, access is restricted to the various Jitterbit Services on an as needed basis. The team knows which employee has access to which Jitterbit Harmony production resource at any point in time, and can revoke that access as needed.

**Personnel Policy**

Jitterbit’s personnel policy is designed to maintain a high level of employee trustworthiness and to keep employees aware of key aspects of information security and privacy. Employees must comply with a code of conduct that emphasizes confidentiality, ethics, and professionalism in all interactions with Jitterbit’s users, partners, and competitors. All employees sign a confidentiality agreement that protects Jitterbit’s customer data.

**Jitterbit Operations**

The Jitterbit Operations team is responsible for defining and executing procedures for application release management, hardware and operating system upgrades, system’s health monitoring, and other activities required for the maintenance of Jitterbit Harmony.

The team’s responsibilities include:

- Reviewing the security of cloud infrastructure design and implementation.
- Implementing procedures that follow security standards, such as ISO 27002.
• Defining and implementing identity and access management policy, including procedures for assigning unique and trackable identities to each member of the Jitterbit Operations team.
• Defining data confidentiality classifications that require employees who access Jitterbit Harmony customer information to do so in a prescribed manner that limits the possibility of unauthorized access.
• Identifying and implementing technologies that secure customer information, including encryption technologies for data in transit and data at rest.
• Conduct information security assessments that are based on penetration tests.
• Monitoring Jitterbit Harmony for possible security issues

Jitterbit Engineering

The Jitterbit Engineering team is responsible for designing, implementing, and testing the software services provided by Jitterbit Harmony. The Engineering team works closely with the Information Security and Operations teams to identify security concerns, develop monitoring procedures, and implement protective technology. The security responsibilities of the Engineering team include:

• Defining and implementing secure design and coding practices.
• Conducting design reviews to identify possible security concerns prior to coding.
• Conducting code reviews to identify code that could be exploited to grant unauthorized access to customer data.
• Conducting code reviews to identify code that could negatively impact availability.
• Performing load tests in pre-production environments to verify that availability requirements have been met.

Jitterbit QA Team

The Jitterbit QA team is responsible for carrying out new and existing regression tests on all software release by engineering to ensure no security or functional issues are introduced with changes in the software. The Jitterbit QA Team performs its function in a separate environment that closely resembles production configurations.

The Jitterbit QA Team must approve any software release before the Jitterbit Operations team can deploy that software to the Jitterbit Harmony production environment.

Jitterbit Harmony Trust Site
Jitterbit Harmony availability and security statuses are monitored 24 hours a day, seven days a week by the Jitterbit Operations Team. The data pertaining to such monitoring is published on the Jitterbit Harmony Trust Site (Trust.Jitterbit.com) giving users and the general public transparent visibility into our operations.

**Identity and Access Management**

Identity and access management policy requires that all Jitterbit personnel that have access to Jitterbit Harmony production environments be provisioned with unique and trackable identities in the form of a User ID. Identity and access management policy enforces the principle of least privilege, which restricts personnel to the minimum level of access required to complete their assigned tasks.

Virtual instances, firewalls, database servers, and other infrastructure software and hardware are protected by user identities that have been granted a limited set of permissions. Permission grants are regularly reviewed by the Operations team and revoked when an employee leaves the company. The Operations team enforces a password policy throughout Jitterbit Harmony production environments that requires strong passwords, regular password expiration, and restrictions on password reuse.

**Incident Management**

The goal of the Jitterbit incident management policy is not only to quickly and effectively close incidents, but to collect and distribute incident information so that processes are continuously improved and future responses are driven by accumulated knowledge.

Incident management includes initial diagnosis, classification, prioritization, escalation, and closure. All incidents that do not affect users of Jitterbit Harmony are recorded in the engineering issue tracking system. Any issues that affect users are recorded in the Customer Support system so that any affects on SLAs are tracked.

**Patch Management**

Jitterbit is continually strengthening its products as new threats to security emerge. In addition, the software infrastructure we use is also being strengthened.

In order to keep software used current, the Operations team works with the Engineering and QA team following a detailed patch management policy that covers the discovery, testing, and deployment of security patches.
The Operations team actively monitors vendor security advisories and subscribes to new patch release notifications.

**Capacity Management**

Jitterbit Harmony currently supports thousands of active users who perform various integration processes. The Jitterbit Harmony platform has been developed to scale dynamically. The core services that expose APIs to our tools and users run on Apache, Tomcat. Our systems track current usage rate and automatically provision and stop EC2 instances as required.