Technical Details

slate by technolutions

SECURE HOSTING

Technolutions provides all hosting, network, and server management for Slate in secure, modern datacenters, through the use of the Amazon Web Services (AWS) cloud. Production services are hosted in the us-east-1 region in Northern Virginia, with services duplicated across two availability zones. Each availability zone consists of one or more discrete datacenters, each with redundant power, networking, and connectivity, housed in separate facilities, and physically and operationally isolated from the other availability zones. Each can take over from the other in case of an outage. All data and machine configurations are further replicated to the us-west-2 region in Oregon for disaster recovery. Institutions based outside of the United States have the opportunity to locate their data in the ca-central-1 region in Canada, the eu-west-1 region in Ireland, or the ap-northeast-1 in Tokyo, Japan. No sensitive data is stored outside of the institution's designated region, and content delivery network servers throughout the world cache static, non-sensitive resources. Datacenters undergo annual SOC audits, and no issues have ever been identified from these audits, nor has there ever been a data or security incident of any kind.

ENCRYPTION IN TRANSIT

All data is received and transmitted over TLS, using 2048-bit keys. Due to vulnerabilities with the SSL protocol, no connections using SSL are supported and only connections over secure versions of TLS may be initiated. 128-bit asymmetric encryption is enforced as the minimum, with 256-bit AES encryption available as the default for supported clients. Forward secrecy is supported within all modern browsers.

ENCRYPTION AT REST

All data is stored in encrypted databases on encrypted filesystems in secure datacenters, using 256-bit AES encryption.

SINGLE SIGN-ON

Slate integrates with institutional single sign-on, including CAS, LDAP / Active Directory, SAML / Shibboleth / AD FS / Azure AD, and other identity and authentication providers, with permission- and role-based authorization tables. Technolutions is a member of InCommon. Slate supports multifactor authentication.

PERMISSIONS AND ROLES

Slate provides field-level, function-level, and feature-level security, enabling granular control of access permissions and rights. An institution can create any number of custom permissions and roles, each of which may contain any number of standard or custom permissions. A user may be assigned to any number of roles and will assume the security permissions from each.

HIGH PERFORMANCE, HIGH AVAILABILITY

Slate utilizes redundant systems and resources at every level in the architectural stack. Should Slate become unavailable for any localized reason, we immediately begin a failover process that takes approximately 15 seconds to complete. We regularly add servers and computational capacity to provide real-time data with ever-increasing performance. Slate typically achieves "five nines" of availability, with downtime of less than 5 minutes for the entirety of the calendar year. A transaction enters the Technolutions network through redundant firewalls and load balancers, where all non-essential ports are closed and traffic is evaluated through deep packet inspection. The requests are answered by nodes in the web cluster, which pass requests to parameterized procedures through limited-rights accounts to nodes in the database cluster. The web, worker, and database clusters do not have publicly-routable addresses, and the web clusters are only accessible via the load balancers and application security firewalls. Capacity is regularly evaluated and added to support continued growth and resource utilization. We test and verify the redundancy of these systems quarterly and after major changes.

BUSINESS CONTINUITY AND DISASTER RECOVERY

Technolutions is headquartered in New Haven, Connecticut with a second office in Portland, Oregon. Each office provides staff and geographic redundancy, and employees may access tools remotely via secure, two-factor VPN connections and terminal servers in the event of limited physical access to our New Haven or Portland offices.

COMPREHENSIVE INSURANCE COVERAGE

Technolutions maintains, through Chubb (A.M. Best rated A++ Superior), policies in the amounts of $1,000,000 per occurrence for comprehensive business liability, $2,000,000 aggregate for comprehensive business liability, $4,000,000 per occurrence for excess liability, $1,000,000 per accident for workers' compensation, $1,000,000 disease policy limit for workers' compensation, and $7,000,000 for technology services errors and omissions liability.

ACCESSIBILITY AND COMPLIANCE

Slate maintains compliance with all laws and standards, including PCI compliance for all financial transactions, NACHA compliance for all ACH transactions, FERPA compliance for the protection of student information, GDPR compliance, and adoption of and adherence to Section 508 ADA accessibility guidelines, as implemented through the WCAG 2.0 accessibility guidelines.

SESSION AUTHENTICATION

All requests to Slate resources are routed over HTTPS with a minimum grade of 128-bits enforced. Any request over HTTP is automatically redirected to HTTPS, including a browser-based non-network redirect to HTTPS based upon cached HSTS information. When a user authenticates with Slate, a new login entry is created for the user in the database, with a session ID as a 128-bit UUID, their user ID (also a 128-bit UUID), the IP address they are logging in from, a machine cookie that is set (to be able to uniquely identifier a particular computer/browser), the user agent (browser string), the login date/time, and the expiration date/time of the credentials (usually 60 minutes). Slate then sets a session cookie in the browser with the "HttpOnly" flag set (preventing it from being accessed by client-side script) along with the "secure" flag (preventing it from being sent over HTTP). The cookie has the value of the 128-bit session ID. These cookies are set as session cookies (no expiration set) and are deleted upon closing the browser. Upon every request to an authenticated resource, we check the session ID from the cookie against the persistent machine cookie and IP address stored, in addition to verifying that the session has not been expired (such as clicking "logout" which forces immediate expiration of the ticket in the system, or having the expiration date/time elapse). Each new authenticated page request extends the lifetime of the session. These session IDs are completely unique and cannot be guessed.

TRANSIT LAYER

- Requests over HTTP are redirected to HTTPS.
- HTTP Strict Transport Security headers set to prevent against HTTPS downgrade attacks.
- Only secure versions of TLS are supported and all versions of SSL are disabled.
- Request verbs are limited to GET, HEAD, and POST.
- Request content length is limited as appropriate.
- Cache expiration is enforced ("Expires: 0" and "Cache-Control: private" or "Cache-Control: no-cache") on all secured pages.
- Content is compressed using gzip or deflate if supported by the browser.
- Static content is cached server-side with non-immediate browser expiration and via edge-servers in content delivery network utilizing international datacenters for low latency access all around the world. Sensitive data is never accessed by or through the content delivery network.

APPLICATION LAYER

- No platform-specific file extensions or headers are used, helping to obfuscate the application and platform types.
- No source code is published to production servers. Application code is fully compiled.
- Application code is fully managed, and there is no native application code which might provide a vector for a buffer overflow.
- No detailed error messages are ever displayed externally.
- Unhandled exceptions are logged automatically for evaluation.
- Uploaded files are never committed to web-readable directories.
- Parameterized procedures are used for all transactions, eliminating the vector for SQL injection attacks.
- Pages are rendered by building an XML document and transforming that against an XSL transform, which escapes all data by default. Under no circumstances is output escaping ever disabled for user input.
- Sequential identifiers are never used. Only 128-bit UUIDs are used for primary record identifiers.
- Sessions are maintained by generating a UUID in a login table and assigning that UUID to a session cookie, and may be remotely terminated by an authenticated user or automatically upon session expiration.
- Unnecessary whitespace from all rendered pages is removed, all CSS are minified, and all Javascript resources are obfuscated and minified.
- Non-authenticated redirectors require a salted URL hash to prevent abuse.
- External user accounts are activated using the external user's email address and a 9-digit PIN that is transmitted by email, in addition to the birthdate for verification purposes (optional), which is not communicated by email. Upon activation, the external user must set a password of his/her choosing. A salted hash is stored instead of plaintext passwords.
- Administrative users are authenticated against institutional SSO. Passwords are never stored.

STORAGE LAYER

- All implementations of Slate use the same database schema that is centrally developed, tested, and administered.
- Each Slate instance has its own discrete database, and no institutional data is ever commingled with data from other institutions or stored in a database that could be accessed by users from another institution.
- Databases run with full transaction logging. Transaction logs are backed up every 3 hours and are held for at least 60 days, providing point-in-time restores for that duration. Full

backups are taken weekly and are held for at least 60 days. The outside duration of the Recovery Point Objective is 3 hours, and the outside duration of the Recovery Time Objective is 12 hours (the RTO for most issues would be measured in seconds), depending upon the severity of the issue.
- Document stores are versioned and all versions are automatically replicated throughout the us-east-1 region, with near real-time replication to the us-west-2 region.

INFRASTRUCTURE LAYER

- Administrator-level permissions are closely controlled, and no generic accounts are ever permitted for server-level access.
- Audit logs are held for at least 1 year and are replicated nightly to off-site storage.
- Servers run recent versions of their operating systems (Microsoft Windows Server 2019) and the latest versions of the database servers (Microsoft SQL Server 2019).
- Servers are patched regularly every month and as necessary for critical 0-day exploits.
- Secure connections are brokered through Windows Domain accounts using NTLM authentications, and SQL authentication is never used or permitted, nor are passwords permitted to be stored within application code.
- Forest and domains run at the highest functional levels, with all insecure protocols and encryption algorithms disabled.
- Services run under limited-access accounts.
- No servers have publicly-routable addresses, with all public IPs held by the firewalls and load balancers and only specific ports routed to private IPs.
- Remote Desktop access is limited to the connections from within the VPN.
- VPN access requires two-factor authentication, where one factor is the password followed by a token, and the second factor is human-approved response from an authorized mobile phone.
- Group policies are employed to limit wireless and remote storage access.

DATA INTEGRATION AND MIGRATION

Slate supports the bi-directional transfer of data between Slate and external systems, including student information systems such as PeopleSoft, Banner, Colleague, and Jenzabar, financial aid systems such as PowerFAIDS, search lists and score data files, and homegrown systems. Data integrations are achieved through several different mechanisms, enumerated below:

Data Export (Slate to an external system)

Batched exports These are built in the query tool and involve the generation of flat files (fixed-width, delimited, XML, JSON, etc.) on a scheduled frequency. Any code and value translations can be configured within Slate but outside of the query, so the query can be stable and immutable even when new entry terms, majors, and other code changes are introduced. This also ensures that the process on the campus system (e.g., SIS, ERP, etc.) side remains stable year-over-year, too. These exports can be cumulative, incremental, or differential. This differential option uses notification queues to track which records have changed since the export was last run, so full rows are returned for only the students for whom there has been some change to their record. This is typically the most appropriate option for exports to a campus system. The exports can be scheduled to occur on a daily frequency (or more frequently, as desired) and can be run on-demand for those instances throughout the year where you may need to update the campus system more immediately after adding/changing a batch of decisions. Exports can be pushed out to our SFTP servers or to a remote SFTP server. For exports to a campus system, we generally advocate for the use of batched exports, using delimited differential files, on a nightly overnight frequency, and to our SFTP server, where a school can then poll the /outgoing/ directory periodically to pull down and load any files. There are several major benefits to the batched export option. First, if exporting to our own SFTP server, we can ensure that it remains online/operational during the export. Second, there is the inherent logging of all exports that occurs just by preserving/archiving the exported files. This provides a great resource for troubleshooting if ever necessary. Third, the exports are scheduled to occur within a delivery window, but the particular execution time within that window (e.g., 2:00am until 4:00am) can be moved around based upon server maintenance activity and load. Fourth, we can provide automated email notifications upon successful, late, and/or failed generation and delivery of scheduled exports. Fifth, if using a delimited file structure, depending upon the capabilities of your system, new fields may be added to the export without it breaking anything on the campus system end.

Web services The same query that would be built for a batched export can be made available as a web service at any time simply by enabling web services on the query. If a specific XML structure was desired, there would be some modifications to the query, but they would generally be reasonably minor. When coupled with notification queues, these would enable the school to poll the web service on a frequency that they define and pull down just the new/changed records each time. Web services are a fine and production-appropriate option, but they usually don't offer enough additional business value to outweigh the benefits offered by batched exports. Typically, the data is not so volatile and the business need to see that data reproduced in a campus system so great that frequent web service polling would even be necessary, so a batched export usually achieves all the business process requirements. Web services are also much more difficult to troubleshoot, should the need arise, since the request and response are transient.

Other options Batched exports can be utilized for a nightly feed and coupled with web services for an update/incremental feed of perhaps a limited set of data points, individual records could be requested using APIs or custom web services, or we could post data to a remote web service, so there exists an array of additional options, but 95% of all data integrations typically use the batched file approach.

Data Import (from an external system into Slate)

Batched imports and data migrations These are essentially the reverse of the batched export process, but configured and managed using different tools within Slate. Most frequently, the school delivers files to import to an /incoming/ directory on our SFTP server, which we poll frequently (at least once every 15 minutes) and load any files matching a specified filename mask. We can poll a remote SFTP server, too, but to the point of SFTP server availability, since we can ensure that our servers remain highly available, the experience is usually most reliable when using our infrastructure. These files can be delimited, fixed width, XML, etc., and we typically recommend that delimited files with column headers be used, since you can add/remove columns at any time without breaking the import process within Slate. This allows for asynchronous changes to the data feed specifications. The files are routed into our Upload Dataset tool, where the format can be predefined to handle all value and code translations, ensuring that the year-over-year changes to accommodate new fields or values is straightforward and can be handled by non-technical end users. This same Upload Dataset tool is used for importing data from search lists, score data files, and historical data migrations.

Transactional web services Web services can be created within Slate to update individual properties on individual records. These are appropriate for purely transactional events, but since they would execute synchronously, there could be some record locking while changes are committed. These also wouldn't be appropriate under high volume, since thousands of writes is not nearly as efficient as a single batched write to thousands of records. These typically require custom SQL, so their maintainability is not as straightforward.

Batched web services (synchronous) Like the transactional web services, an XML post could be submitted containing updates to be made to a lot of records. Since these would be executing synchronously for a group of records, the likelihood of record locking as changes are made and committed is greatly increased. These also typically require custom SQL, so maintainability is potentially an issue here.

- *Batched web services (deferred)* This option provides a web service way to post files into Slate that are then processed by the Upload Dataset mechanism, just as if the files were transferred via SFTP. These could include XML posts but can also include delimited data. Since the updates are processed through our Upload Dataset mechanism, the changes can be queued, batched, and run in the most efficient manner possible which will minimize or eliminate any potential for observable record locking.

Remote web services (scheduled) Slate can also consume data from remote web services and, like pulling down a file from a remote SFTP server, can route the data into the Upload Dataset import system. There is a time and place for this option, but it's not frequently utilized.

Document Import

Document Imports should occur over SFTP, since a ZIP archive containing thousands of PDFs could potentially be quite large. Files are typically sent using a DIP file approach, wherein a ZIP archive is generated containing PDFs/TIFFs of the documents to be imported along with an index file containing the filename of each document and any associated metadata parameters (an SIS ID and document type, for example). Slate can then extract the documents and index file and import the documents onto the appropriate student records. Slate can also extract metadata from within a filename, so you could have files with an SIS ID and document code in the filename and obviate the need for an index file, but usually the DIP approach is best here. We always recommend encapsulating all of these documents into a ZIP file, since SFTP is much more efficient with the transmission of a single file (e.g., a ZIP archive) instead of with the transmission of thousands of individual files. We also prefer PDFs to TIFFs, since a digital PDF of non-scanned data would be a fraction of the size of a TIFF, since a TIFF is a rasterized/bitmapped image that won't contain any digital text content and thus cannot be enlarged beyond the original resolution without a loss of fidelity.

Document Export

Like document imports, Slate also supports the scheduled and on-demand export of any set of document types for any population of records. These are typically delivered in a ZIP archive to our SFTP servers.

Direct SQL Access

We support direct SQL access from workstations/servers at the institution into their Slate database, but this is for ad hoc use only and is not intended for data integrations. This is for a few reasons. First, if the data integration exists entirely outside of Slate, Technolutions will not be aware of it and will not be able to proactively address the potential effects of schema changes. Second, direct SQL access is not schedulable on our end since the queries are being remotely initiated. Third, a poorly written SQL query could over-consume resources and create performance issues that are much more difficult to troubleshoot without visibility into what or how something was being used. Fourth, the access is all or nothing, so any user with direct SQL access has complete read-only access to the database.

Additional information is available upon request.

Technolutions New Haven
157 Church Street, Fl 22
New Haven, CT 06510
203.404.4835

Technolutions Portland
1211 SW 5th Avenue, Ste 2600
Portland, OR 97204
503.765.7500

www.technolutions.com