Instant Marketing Security Whitepaper

Contents

1. Security

1.1) Overview

We have a security strategy which enables us to highly mitigate the most common security risks.

Part of this strategy is for our team to "stand on the shoulders of giants" – using best-in-class expert services to avoid managing our own servers/VMs/database, to automatically audit dependencies, and a SSO-only strategy to mitigate account-management category risks.

By using fully managed services by Google Cloud Platform (the company who has the world-leading Project Zero cybersecurity team, and who successfully safeguards the huge volume of data for Google's own services), and MongoDB (the same team who maintain the database software itself) we are using amongst the best operational/security teams in the world.

We have been able to develop SIGNIFICANT mitigation against the most common cyber-security attack-vectors, in the following ways:

Risk Category	**Mitigation**
Misconfigured or unpatched database	By allowing MongoDB themselves to run our database (via. their MongoDB Atlas service – physically hosted in GCP Netherlands), we have the most competent database administrators possible (the makers of MongoDB itself). This heavily mitigates risks around unpatched or misconfigured databases. Please refer to their impressive security & operation model detailed in this whitepaper: https://webassets.mongodb.com/_com_assets/cms/MongoDB_Atlas_Security_Controls-v7k3rbhi3p.pdf
Misconfigured or unpatched server or VM	We use Google Cloud Platform, but more specifically we use their PAAS (platform-as-a-service) offerings AppEngine and Google Cloud Functions. This means Google takes all responsibility for patching, hardening, and securing the main virtual machine. They implement tremendous security-in-depth, which is detailed in this whitepaper: https://cloud.google.com/security/overview/whitepaper
Phishing, Account Management	There are no Just Add Features specific accounts. We only use SSO from Eloqua. This means we introduce zero additional risk as there are no accounts to compromise, or forget to deactivate, etc.
Outdated Dependencies	We utilize automatic dependency scanning built-into our regular dev/ops workflow to automatically audit dependencies for all known vulnerabilities Everything except dependencies is operated by Google Cloud Platform or MongoDB Atlas
Data Minification	We deploy extreme data-minification. We go so far as to scan user configurations each time they are changed, to detect the exact fields used in each record, and update Eloqua to only send us this minimum possible data

	We additionally only store the most sensitive data for 7 days (enough to process and provide our customers a brief monitoring window) before automatically removing it
Credential Handling	Instead of directly storing access to Eloqua as a username and password, we instead implement an OAuth flow where a limited credential is stored. This credential becomes invalid and must be replaced with every single API request

1.2) Encryption in Transit and At Rest

We utilize Google Cloud for application hosting, and MongoDB Atlas (in GCP) for data-hosting. These providers apply best-practice security. This section details the security/encryption protocols utilized by these providers.

The following two quotes from MongoDB Atlas' security whitepaper provide some insight into the encryption schemes employed in a MongoDB/Google Cloud Platform combination:

"TLS and authentication (SCRAM) are enabled by default and cannot be disabled. Traffic from clients to Atlas is authenticated and encrypted in-transit, and traffic between the customer's internally managed MongoDB nodes is also authenticated and encrypted in-transit using TLS."

…

"Encryption for data at rest is automated using GCP's transparent disk encryption, which uses Advanced Encryption Standard (AES) algorithm with 256-bit key length, in Galois/Counter Mode (GCM). This is implemented in the BoringSSL library that Google maintains. In addition to the storage system level encryption, data is also encrypted at the storage device level with AES-256 on solid state drives (SSD), using a separate device-level-key (different key than storage level). All keys are fully managed by GCP."

Source: https://webassets.mongodb.com/_com_assets/cms/MongoDB_Atlas_Security_Controls-v7k3rbhi3p.pdf

1.3) Backups

a) Database

We have “Continuous Backups” managed by MongoDB Atlas. This provides 2 snapshots per day, which can be restored via. a GUI in their dashboard. This also benefits from best-in-class operational procedures, since it is operated by the database-vendors themselves.

b) Application

Google Cloud Platform’s AppEngine service maintains previous versions as VM images, which gives the ability to quickly “roll back” to a previous version in the event of an issue.

Code is managed in a source-control system with local and remote copies stored securely.

In a disaster-recovery scenario all accounts would be restored simultaneously, as they are part of the same backup. (Our application in general is the highest priority.)

1.4) Certifications

As much of our technical operations as possible is managed by our PAAS vendors (Google Cloud, MongoDB Atlas). They take responsibility for the physical hardware, operating system/system-level patches, database administration, network design/maintenance, VM-hardening/patching/multi-tier security, etc. This means our operational responsibility is primarily limited to our own codebase/application-design.

Therefore, in this section, we list the certifications of these vendors:

a) ISO/IEC 27001

a. Google Cloud ISO/IEC 27001 compliance page
https://cloud.google.com/security/compliance/iso-27001/
b. MongoDB Atlas ISO/IEC 27001 compliance page
https://www.mongodb.com/cloud/trust/compliance/iso

b) ISO/IEC 27017

a. Google Cloud ISO/IEC 27017 compliance page
https://cloud.google.com/security/compliance/iso-27017/

c) ISO/IEC 27018

a. Google Cloud ISO/IEC 27018 compliance page
https://cloud.google.com/security/compliance/iso-27018/

d) SOC 1/2/3

a. Google Cloud SOC 1/2/3 compliance page
https://cloud.google.com/security/compliance/soc-2/
b. MongoDB Atlas SOC 2 compliance page
https://www.mongodb.com/cloud/trust/compliance/soc

e) PCI DSS

a. Google Cloud PCI DSS compliance page
https://cloud.google.com/security/compliance/pci-dss/
b. MongoDB Atlas PCI DSS compliance page
https://www.mongodb.com/cloud/trust/compliance/pci-dss

f) CSA STAR

a. Google Cloud CSA STAR compliance page
https://cloud.google.com/security/compliance/csa-star/

g) GDPR

a. MongoDB Atlas GDPR compliance page
https://www.mongodb.com/cloud/trust/compliance/gdpr

1.5) Log Retention

Our main log aggregator is Google Cloud Platform's Stackdriver service. Stackdriver has the following retention periods:

a) Normal logs are retained for 30 days.

b) Audit logs/system event audit logs/access transparency logs are retained for 400 days

Please see https://cloud.google.com/logging/quotas

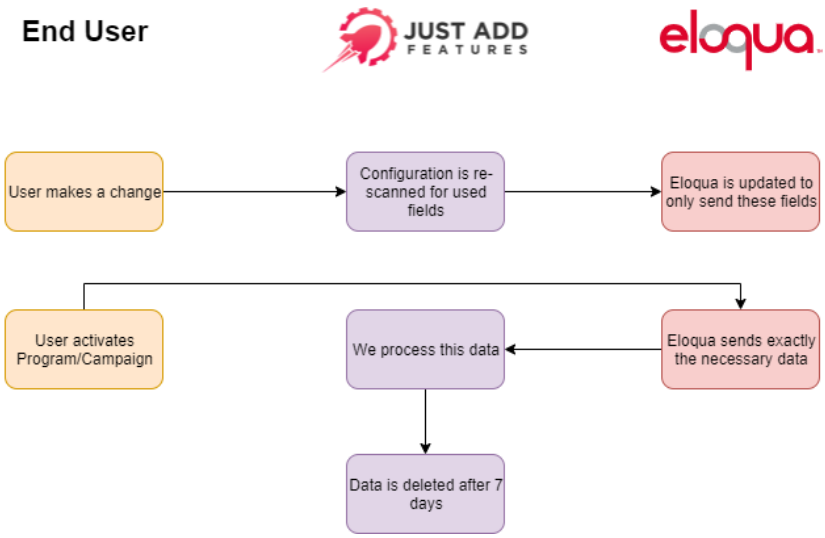
c) We additionally maintain transactional data including outcome and some logging for a period of 7 days in our database.

2. Data

Just Add Features as an extension to the Oracle Eloqua platform. Please note that the information in this document refers to the handling of data stored directly by Just Add Features – not data held within the Eloqua platform itself.

2.1) Data Minification

We utilize an extreme form of data-minification. This is summarized in this diagram:



Each time a user makes a change, we scan their input to determine the minimum set of data required to process this, and we update Eloqua with our new requirements.

When a campaign/program is activated, Eloqua uses these requirements to send only the exact minimum necessary data.

Additionally, we delete our operational data every 7 days.

2.2) Data Retention and Removal

Our primary operational data (e.g. records we are processing) is actively minified (see above section) and is deleted every 7 days. This gives us enough time to process these records, and to provide the End User with a period where they can monitor the outcome in detail.

A non-personally identifiable set of this data is stored permanently (or semi permanently) in our analytics database, so things like campaign outcome are permanently available, but things like individual sent messages are not.

2.3) Data Processing Agreement

End-user data must never be extracted from the MongoDB database, e.g., to make a local copy is forbidden.

Should there be an operational need to justify this, our staff would obtain explicit consent to process customer data, for a specified duration and purpose.

Our Data Processing Agreement can be found on https://instant.marketing/compliance

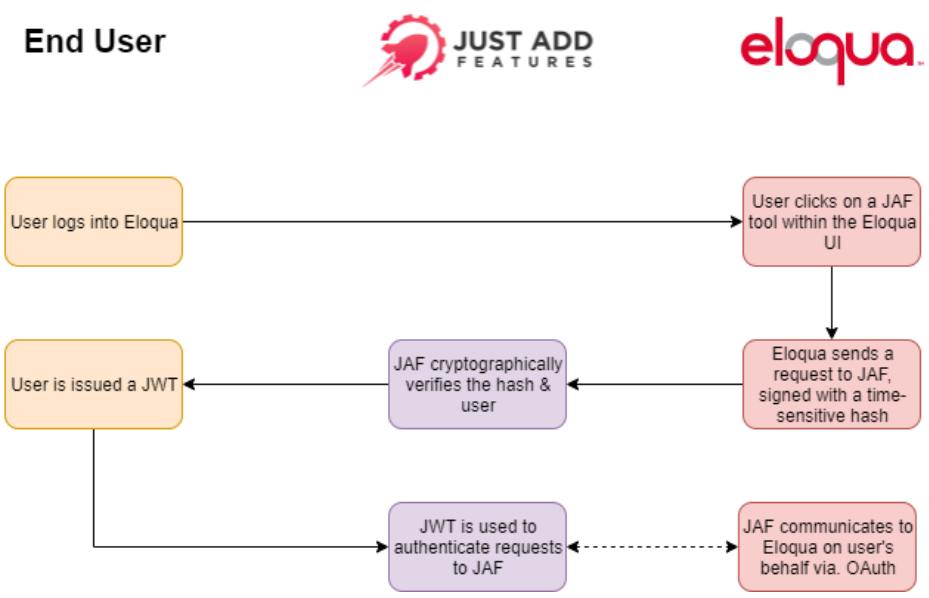
3. Application Design

3.1) User Access & SSO

Since our application functions as an extension to Oracle Eloqua, we do not manage "Just Add Features" accounts. Instead, Oracle Eloqua itself is used as a Single Sign-On provider.

This means that there is no need (and therefore risk) to provision and manage additional accounts.

- Anyone who is deactivated within Oracle Eloqua is automatically deactivated also within Just Add Features
- Anyone who has added to Oracle Eloqua is also automatically added to Just Add Features – *however* they will not automatically be given any access permissions. These roles must be explicitly set (see the next section)
- By extension, if you configure SSO for Eloqua itself, that would in-turn authenticate Just Add Features. (Whatever authentication strategy is used for Eloqua also automatically applies to Just Add Features)



Interaction between JAF and Eloqua is also handled via. best-practice OAuth token-delegation, rather than by less secure methods (e.g. storing a username & password).

The Eloqua OAuth implementation is particularly secure, as it uses a fast-rotating token which changes with every request.

3.2) Role-Based Authorization

Single-sign-on with JWTs is used to authenticate users, but this simply establishes identify. To be able to perform actions within the system, roles must also be granted to these users.

Administrators can add/remove user roles using the application interface.

3.3) Audit Trail & RTSM

Audit Logging and Real-Time-Security-Management integration is not included by default but can be configured and customized for an additional fee.

Some of the possibilities include:

- Logging all successful and unsuccessful login attemps
- Logging all system-configuration changes
- Logging all content-configuration changes
- Logging all 404 errors from invalid URLs
- Access to view audit logs through the UI

We do not monitor access ourselves as part of our standard contracts.

3.4) Application LifeCycle

We have 3 distinct environments (development, staging/test, and production), each with their own set of applications and their own distinct database. The stages below represent typical practice – but sometimes these steps may be out of sequence (for example if something involves multiple components/microservices it may go to the test-server early and often).

a) Development

Applications are developed against a local (development) version which runs on localhost and connects to a specific development database. Dependencies are scanned for vulnerabilities during the development stage.

b) Manual/Automated Local Testing

A combination of manual and local testing on the development machine is used. For automated testing we use some combination of the Jest framework and Cypress. Automated testing is used frequently but is not used in all cases. At this stage test-specs will frequently be updated.

c) Code-Review

In many (but not all) cases, code will go through a review with one or more other developer, once it is in release-candidate stage. This is looking for code quality issues or security issues.

d) Test Server

Code is deployed to the test-server at least once. This step can happen multiple times and can happen earlier in the development. It should always go to test server as a final step before production. This lets us test in a near-production environment, and helps catch certain classes of issues (e.g., interactions in the infrastructure, build failures and many others). The test server also allows multiple people to test the release if required

e) Production

Code is deployed to production.

f) Monitoring

We have various monitoring and alerting tools to ensure application health

3.5) Microservices & Separation

Microservices are implemented (using Google Cloud Platform's managed "Cloud Functions"), which enable enhanced scalability and security for various parts of the application.

Many of these microservices are configured to be isolated from the public internet.

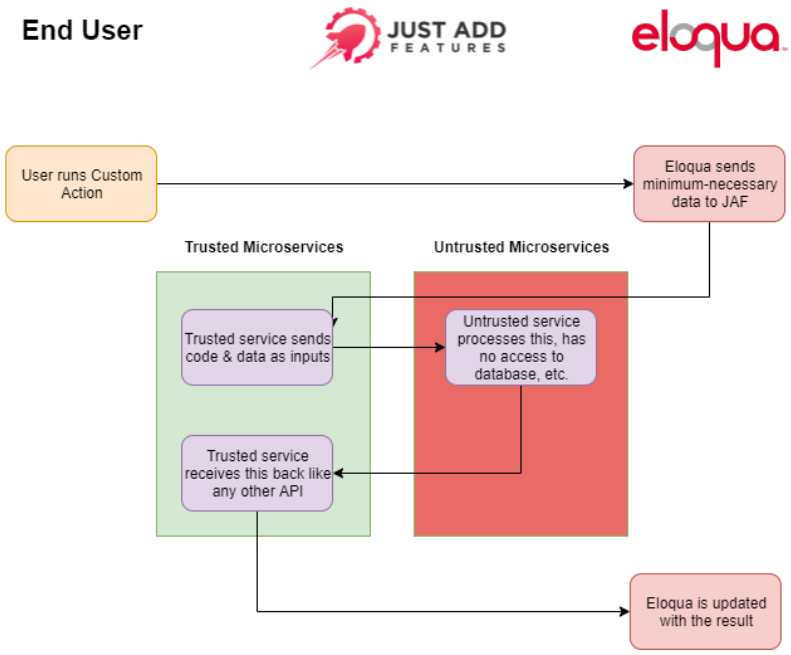
Generally, backend microservices for processing are not available on the public internet. Microservices in general have isolated, self-contained roles.

One of the most important uses of microservices for separation comes when we run untrusted code.

The execution microservice is a fully isolated service which is called with code & data as an incoming payload and returns an output. It has no access to database credentials or anything outside of this explicit payload. Essentially – it is being called as an API. The only interface this untrusted service has is to receive a payload and return a JSON response.

The above is an extreme example, but the principles of isolation are applied in other places. Each Cloud Function has its own set of dependencies, access rights, environment variable access, etc. depending upon what each service needs. Where possible we use Google Cloud Platform's secure message bus "Pub/Sub" to communicate between microservices, which greatly limits the interfaces each microservice has with the rest of the application.

This also allows for faster, safer code-deployment, as each microservice can be updated independently.



4. Internal Procedures

4.1) Incident Management

Just Add Features has never had a security incident. However, in the case that there is a suspected breach or critical vulnerability, clients should report this to their account manager. Our organizational policy is to immediately escalate this to our CEO and senior engineering staff.

If a breach is confirmed, we will notify all affected customers by email and/or phone, within 24 hours at the latest.

Efforts would immediately be taken to identify & remedy the situation, to preserve logs, and to help customers with their mitigation as best as we can.

4.2) Disaster Recovery

Most aspects of disaster recovery are handled by our underlying providers (Google Cloud Platform, MongoDB Atlas). They have redundancy built into their platforms, and they both independently publish details of these measures.

Critical operations (application management/updates, database administrations, email, etc.) are all possible remotely, so if the Just Add Features office was physically unavailable, we would be able to continue to run our business.

4.3) Physical Security

The Just Add Features office does not contain any servers. Our data-centre site is Google Cloud Platform in the Netherlands (this is also where our MongoDB Atlas cluster is physically located).

The Just Add Features office is protected by two key-locks and a keypad on the internal office door, a keyfob for entry on the external office door, and CCTV cameras throughout the building.

All devices used are password controlled, and data is generally stored in cloud services.

We are too small a company to specifically survey for unauthorized equipment, require ID badges, etc: our office holds less than 10 people, so we all know everyone and all the equipment.

All company devices must be returned when an employee leaves. If these devices are being re-issued, they can be reformatted/reset. If these devices are being disposed of, the hard drives must be securely wiped or physically destroyed.

4.4) Vulnerability Scanning

We have security scanning built into every stage of our development workflow. Dependencies are automatically checked against the Node Security Project database for vulnerable versions. https://www.npmjs.com/advisories

Severity “moderate”, “high”, or “critical” vulnerabilities are immediately patched where possible. In most cases patches are available and can be implemented.

In the even that there are vulnerabilities for which no patch is available, or a patch cannot be immediately implemented:

- “High” vulnerabilities for which no patch is implementable, senior technical staff must be alerted.
- “Critical” vulnerabilities for which no patch is implementable, the CEO must be alerted

4.5) Support & Incident Notification

Support is available through your named Technical Account Manager. Incidents should also be reported through this channel.

Support is on a “best effort” basis and we have no formal SLA, unless one is contractually specified (for example, if you purchase a support contract).

We will provide support where our product is deficient. Other cases (for example, end-user code in our Instant Extensions products, configuring campaigns, solution design, etc.) may fall outside of the remit of product support. The Technical Account Manager will decide what constitutes product support. In cases where the customer wants support which goes beyond basic support (or what is included in their contract) we also offer paid consultancy/professional services.

5. Additional Security & Isolation Package (Enterprise)

5.1) Additional Security & Isolation Package

This is not included within our standard contracts, but the Additional Security & Isolation Package provides an even higher level of security, by further isolating each tenant.

Our security models are based upon logical security at the application-level, but with multi-tenant application and database instances.

The Additional Security & Isolation Package gives clients their own dedicated MongoDB instance, their own dedicated Google Cloud Platform project, including their own dedicated AppEngine instance, environment variable sets, and each of the 50+ microservice Cloud Functions (if that Cloud Function has database access).

This offering carries a significant cost, due to the significant maintenance and hardware inefficiencies incurred.

This offering is not necessary, given the strong security design in-place. However, some clients may need this level of separation to comply with organization policy.