Software Architecture Document

Version: 1.4, Feb 2024

Classification: Public

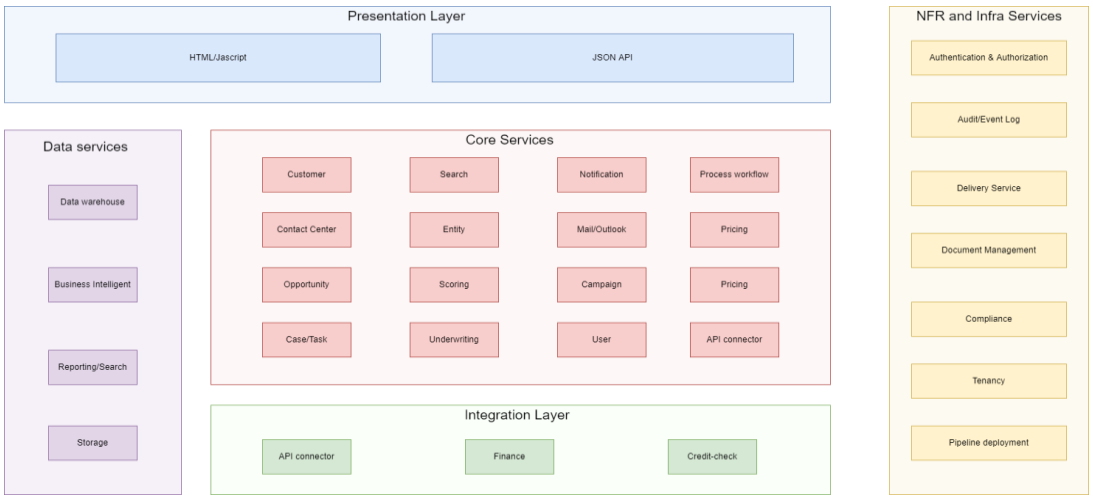
TABLE OF CONTENTS

1. Application Architecture

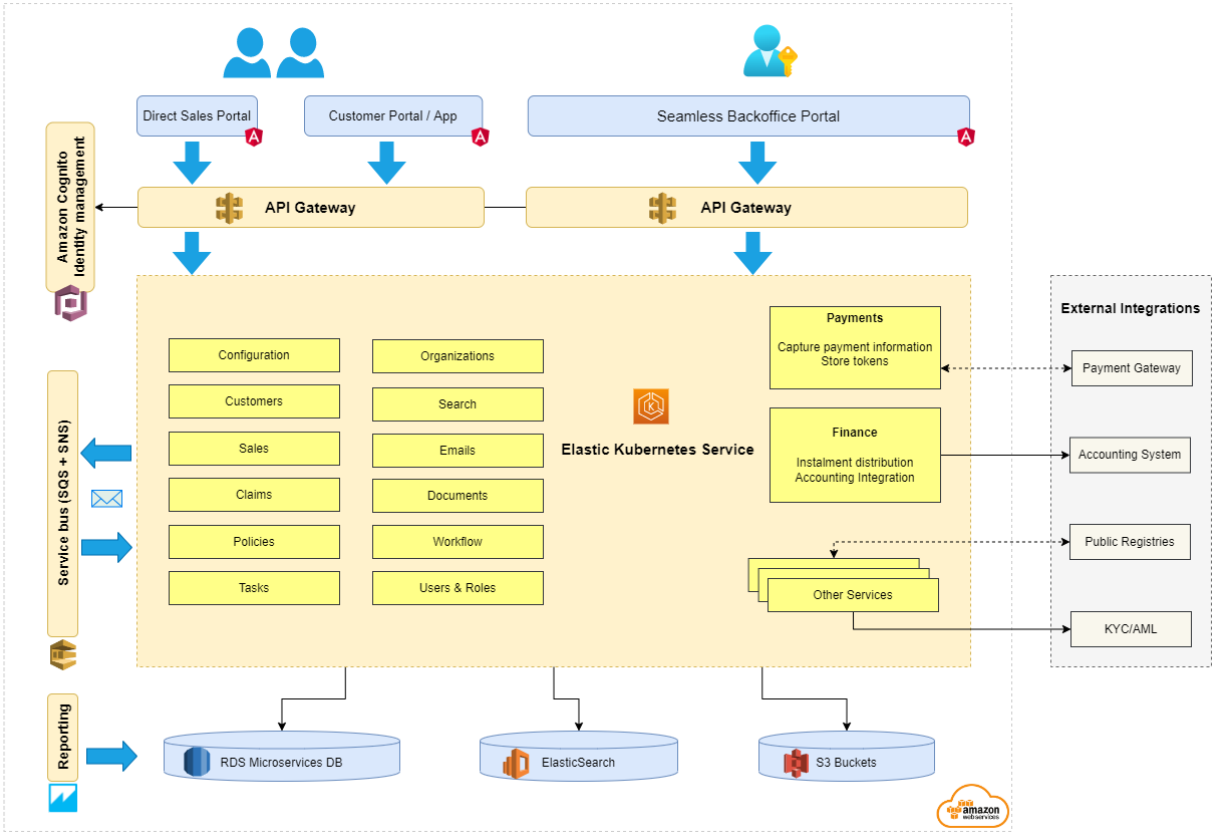
The application architecture is based on key design principles that there will not be a single point of failure and self-healing. Automation of devops functions ensure that the system is able to automatically recover from failures without any human intervention.

Below diagram depicts the 5 layered application architecture of Seamless platform which consist of:

- Presentation layer: Access via HTTPS and responds with API payloads and HTML content.
- NFR and Infra services layer: Handles authentication, authorization and all other aspects of the system including infrastructure.
- Core services layers: Perform most of business logic and client customization.
- Integration layer: Provides adapters that can be used to integrate with third-party systems for integration purposes.
- Data services layer: Data storage, Data warehouse and Business intelligence tools.

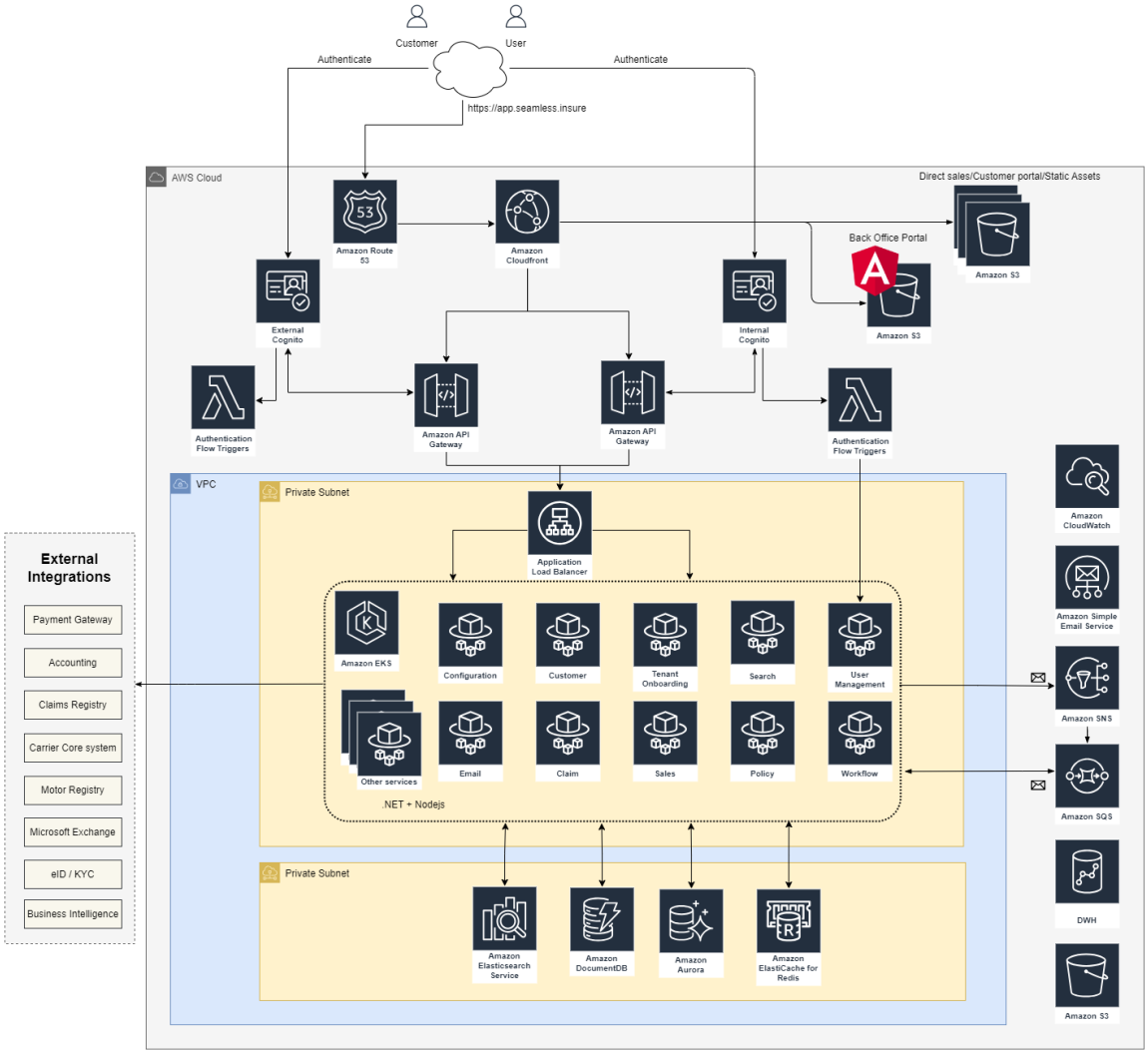


2. Container Diagram



3. Components Diagram

The seamless platform architecture is composed of various components and deployed on AWS cloud.



4. Core Technologies Used

Programming Language (backend)

Seamless microservices are written primarily in .NET and NodeJS. The popular Application development frameworks of .NET Core and NestJS are used extensively. .NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps. NodeJS is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for web applications and back-end API services, but was designed with real-time, push-based architectures in mind.

Programming Language (frontend)

TypeScript (Angular): Angular is a development platform, built on TypeScript. As a platform, Angular includes a component-based framework for building scalable web applications, a collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and much more.

Database

Amazon Aurora PostgreSQL-compatible: PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. Amazon Aurora is a highly scalable and fault tolerant managed service providing PostgreSQL syntax to application services.

Amazon DocumentDB: Managed MongoDB server built on a scale-out architecture that has become popular for developing scalable applications with evolving data schemas. As a document database, MongoDB makes it easy to store structured or unstructured data. It uses a JSON-like format to store documents.

Server OS

Debian: Debian is one of the oldest operating systems based on the Linux kernel. It is a highly stable and secure operating system used by the most trusted organizations around the world.

Web server

Nginx: NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability.

Kestrel: .NET Core microservices use a built-in cross-platform HTTP server implementation which provides great performance and memory utilization.

Containers

Kubernetes (EKS): Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS and on-premises. Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

File storage

Amazon S3: Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Caching

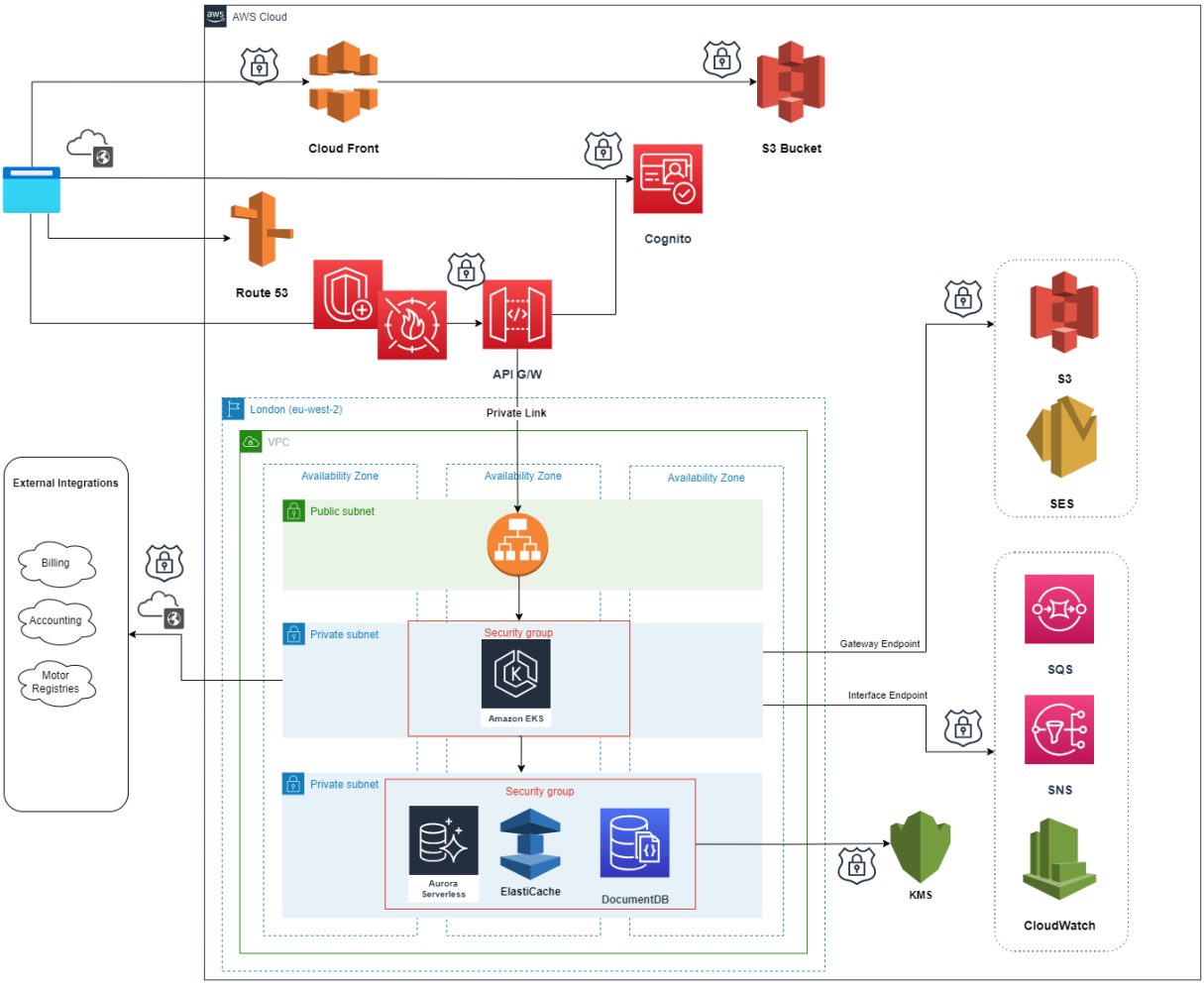
Redis: Redis offers purpose-built in-memory data structures and operators to manage real-time geospatial data at scale and speed.

5. Cloud Service Provider

Seamless is ready for deployment with different cloud providers in order for us to provide cloud based services to our clients in different Regions. It is currently based on Amazon AWS and built using AWS services to leverage maximum scalability and efficiency. And can be made available in multiple regions around the world.

Service type	AWS Service
Compute	• EC2 • EKS • Lambda
Storage	• S3 • Elastic Block Store
Network & Delivery	• VPC • CloudFront • Route 53 • Elastic Load Balancing
Developer Tools	• CodeBuild • CodePipeline • CodeDeploy • CLI
Management Tools	• CloudWatch • CloudFormation
Security, Identity & Compliance	• IAM • Certificate Manager • KMS • Shield • WAF
Messaging	• SES • SNS

6. Network & Infrastructure



Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) lets us provision a private, isolated section of the AWS cloud where we can launch AWS resources in a virtual network that is defined specifically for the client instance. We have complete control over the virtual networking environment, including selection of our own IP address range, creation of subnets, and configuration of route tables and network gateways

DNS Management

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It gives developers and businesses a reliable, cost-effective way to route users to Internet applications. Amazon Route 53 includes a number of global load-balancing capabilities (which can be effective when you are dealing with DR scenarios such as DNS endpoint health checks) and the ability to failover between multiple endpoints and even static websites hosted in Amazon S3.

Multi-AZ Architectures to Achieve High Availability

High Availability is achieved by deploying across multiple Availability Zones in a single AWS region. Redundant instances for each tier (e.g., web, application, services and database) of the application are placed in distinct Availability Zones thereby creating a multi-site solution. The desired goal is to have an independent copy of each application stack in two or more Availability Zones. In order to achieve more fault tolerance with less manual intervention, Elastic Load Balancer is placed before the compute instances, as it can automatically balance traffic across multiple instances and multiple Availability Zones and ensure that only healthy Amazon EC2 instances receive traffic.

Elastic Load Balancer

Load Balancing is an AWS product that distributes incoming traffic to your application across several instances of application services. ELB detects unhealthy instances within its pool of services and automatically reroutes traffic to healthy instances, until the unhealthy instances have been restored. It also handles HTTPS traffic, thereby reducing load on downstream application servers. Application services are hosted on EC2 nodes managed by EKS which can automatically scale in and out available Kubernetes nodes as well as individual microservices using Horizontal Pod Autoscaler.

API gateway

API gateway is an API management tool that sits between a client and a collection of backend services. It also acts as a reverse proxy to accept all application programming interface calls, aggregate to various services required to fulfill them and return the appropriate response.

Web Application Firewall

Seamless uses Web Application Firewall to route all traffic coming from the public internet to Seamless applications. The firewall detects all inbound web traffic and blocks SQL injections, Cross-Site Scripting, malware uploads, volumetric & application DDoS, or any other attacks against the web applications. It also inspects the HTTP responses from the configured back-end servers for Data Loss Prevention (DLP).

Documents

All documents (agreements, proofs, etc) uploaded by users and files from external systems are hosted in an S3 bucket. Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web. It is designed to deliver 99.999999999% durability, and scale past trillions of objects worldwide. All access to S3 objects is access controlled and documents uploaded by customers are encrypted and stored. Behind the scenes, Amazon S3 stores objects redundantly on multiple devices across multiple facilities in an Amazon S3 Region – so even in the case of a failure in an Amazon Web Service data center, you will still have access to your data. Amazon S3's Versioning feature allows you to retain prior versions of objects stored in S3 and also protects against accidental deletions initiated by a misbehaving application

Static File Store

Static files are used to host the web site and other components of dynamic content (such as html, css, js and images). Amazon S3 provides support for copying this content over to its edge servers worldwide, this minimizes network latency and improves load times.

System Management, Monitoring and Alerts

System events are constantly monitored with a combination of Amazon CloudWatch, Grafana and Prometheus services.

7. Integration approach

Seamless is built with integration in mind, it currently supports the following methods and integration patterns:

- Rest API / Web services-based interface. All services are Open API with Swagger available.
- Message queueing for asynchronous communication or Publisher/subscriber pattern.
- Secure FTP file-based interface.

8. System Management, Monitoring and Alerts

Seamless use metrics and logs to monitor application and infrastructure performance. This is an important aspect of ensuring a good user experience, service level agreement (SLA) requirements. Our monitor system notifies real-time analysis to help Staff to be more proactive and take right actions on the right time.

Service type	Services
Infra and application monitoring	Icinga is used for the following tasks: • Monitoring infrastructure and application status, sending notification emails. • Realtime SLA report. • Endpoint monitoring • Support alarms and alerts.
System & process monitoring	Grafana and Prometheus are used for: • Application performance • Support alarms and alerts.
Governance, compliance, operational and risk auditing	• WS CloudTrail • Cloudwatch

9. Maintenance and Extension

Seamless principles strive to ensure that any changes to the core system are done in such a way that the impact on the rest of the system is limited. General best practices, guidelines and design principles include:

- The microservices architecture is used to create a more modular and partitioned system. This ensures that a small team of developers can manage and maintain each microservice and have full knowledge of the same service. Each micro service is independent of the other and therefore the impact of change is well managed from the outset.

- Base product is developed with the flexibility to perform client specific requirements with minimal impact.
  - Abstraction of features into their own reusable and configurable modules and libraries.
  - Use of feature config settings within tenant configuration to introduce customer specific changes without impact to base product or other client functionality.
  - Developing and maintaining a library of reusable UI components.
  - Focus on API's first, to ensure that clients can use Seamless functionality independent of the user interface.
- The Seamless delivery management process ensures that all impact is identified, and risks are mitigated in a timely manner.
- Quality Assurance Testing, including both automated and manual tests are performed on all features to ensure that impact is fully understood.

No-code/Low-code Customization

Seamless supports many parameters driven capabilities that do not require code changes, including:

- Dynamic entities, attributes.

- Dynamic documents and templates.
- Dynamic workflow and process.
- Dynamic product and product builder

Extensions

Seamless architecture supports extensions to the platform through plugins and event integration.

10. Security and Resilience

Contemi's Seamless multi-tenant SAAS platform is a secure managed and hosted service available to insurance agents & insurance brokers.

- The system is hosted in Amazon Web Services (AWS UK) data centres.
- The following technical controls are in place for identity and access management:
  - For local users at the firm access will be via a direct connection and a web browser (for some modules) and by supplying a username and password.
  - For Contemi staff, only selected staff are given access and the access is logged at the following levels:
    - IAM Accounts for AWS
    - Active directory on to the Contemi corporate network
    - Separate Active directory on to the AWS Support platform (Bastion (Terminal Server) to access the system for support)
- The following controls are in place for application security:
  - The services and underlying virtual machines are accessible only via a secure site-to-site VPN or direct connection into Contemi's AWS infrastructure, from both the client and Contemi, with no remote access for external parties.
  - The VPC itself will be set up with appropriate AWS Security Groups such that only required services and ports are exposed outside of each VM.
  - The application components are distributed in Docker containers which likewise expose only required application ports and limit manual intrusion.
  - Inbound data will be transport encrypted using a secure protocol and encrypted at-rest after arrival; any agreed extract would be subject to the same security restrictions.
  - The database will also be encrypted at-rest. User access will be managed as described in the previous answer.
  - The Seamless database is multi-tenant – tenant data isolated within application, utilising middleware user identification & row-level security.

- The following vendor technologies are used for perimeter security:
  - Juniper
  - F5 devices for AWS
- Contemi have carried out penetration testing on AWS and a copy of the report can be supplied if required.
- Data will be supplied to Seamless via receipt of data files over a secure encrypted connection from the firm.
- Once a vulnerability has been reported, Contemi will aim to address it within the following timescales.

Severity	Schedule for system patch/workaround	Schedule for software update
5 (Critical)	3 working days	
4 (High)	7 working days	
3	30 days	Next software release cycle
2	90 days	Next software release cycle
1	180 days	Next software release cycle

- Amazon has provided the following statement in relation to Amazon's key management service:

  "AWS KMS is designed so that no one, including AWS employees, can retrieve your plaintext keys from the service. The service uses FIPS 140-2 validated hardware security modules (HSMs) to protect the confidentiality and integrity of your keys regardless of whether you request AWS KMS to create keys on your behalf or you import them into the service. Your plaintext keys are never written to disk and only ever used in volatile memory of the HSMs for the time needed to perform your requested cryptographic operation. AWS KMS keys are never transmitted outside of the AWS regions in which they were created. Updates to software on the service hosts and to the AWS KMS HSM firmware is controlled by multi-party access control that is audited and reviewed by an independent group within Amazon."

  For further information, see:
  https://aws.amazon.com/kms/faqs/#security
  https://d1.awsstatic.com/whitepapers/KMS-Cryptographic-Details.pdf

- Amazon has provided the following information on Amazons RDS database service:

  "Amazon RDS strives to keep your database instance up to date by providing you newer versions of the supported database engines. After a new version of a database engine is released by the vendor or development organization, it is thoroughly tested by our database engineering team before it is made available in Amazon RDS.
  We recommend that you keep your database instance upgraded to the most current minor version as it will contain the latest security and functionality fixes. Unlike major version upgrades, minor version upgrades only include database changes that are backward-compatible with previous minor versions (of the same major version) of the database engine. If a new minor version does not contain fixes that would benefit RDS customers, we may choose not to make it available in RDS. Soon after a new minor version is available in RDS, we will set it to be the preferred minor version for new DB instances."

11. Application Security

Seamless is using Cognito with a standard OAuth 2.0 for authorization and authentication. For integration with our API, Seamless will provide each client with credentials as follows:

- Client Name
- Authorized redirect URL post-authentication.
- Client support email

Post-client set up the following information will be provided to the client:

- Client id
- Client secret key

Detailed information about the security of Seamless on the infrastructure level can be found in our Security Document.

12. Security Architecture

Seamless is a cloud-based platform that uses AWS infrastructure as they are a cloud service provider with a good reputation and strong leadership position in cloud-based services and security.

Seamless provides a robust architecture and controls on top of watch AWS cloud infrastructure to ensure that the system operates in a safe and secure manner ensuring that our customer's data is protected, and operation environment and systems are highly available and trustworthy.

- Several measures have been put in place to ensure Seamless is a highly secure application. Please read the Security and Infrastructure document for more details.
- The system is frequently audited by experts to analyze known and identifiable vulnerabilities.
- Security experts are hired from renowned firms to perform penetration testing on the application to uncover potential security holes.

Data protection

Seamless ensure that customer's data is protected.

- All data are encrypted at Rest.
- All data are encrypted in Transit.
- Sensitive data are encrypted.
- Verbose error messages, banners and hard coded data scrutinized for information leakage.

Identity and access management

Seamless provides a number of security controls to ensure that access to the platform and application are secure. This includes:

- Segregation of duties.
    - Provide role-based access control matrix.
    - Strict separation of access to production, development and test environment.
    - Restricted use of super admin accounts
    - Separate administrator and end-user accounts.
- Secure access to infrastructure, we follow AWS best practices as follows
    - Use of AWS Root account is controlled.
    - IAM Accounts using principle of least-privilege.
    - MFA is strictly enforced.
    - Role based access control.
- Secure access to applications
    - Role based access.
    - Password policies are added and notifications to detect suspicious actions.
    - 2FA option.

System configuration

System configuration is set in place to ensure the deployed infrastructure is secure.

- Securing the network
    - Setup AWS Virtual Private Cloud (VPC)
    - Create and configure private subnets
    - Use Web application firewall (WAF)
        - Application DDoS protection
        - Complete OWASP vulnerabilities protection
    - Setup security groups as virtual firewalls to control inbound and outbound traffic at various layers.

13. Localization & Language support

Seamless is a multiple tenant system therefore it also supports multiple languages globally. It has been developed the following software internationalization best practices:

- Use of resource files for static and reference text.
- Use of UTF-8 character encoding.
- No concatenation of text to derive other display text.
- Derive date, time and currency formats are all supported for localization.
- Style guides (style, terminology and convention) are defined from the beginning and provide uniformity in style and formatting throughout development and documentation