



Smart Payments

info@app55.com



White label, one-click payments from any device

App55 Technical Overview

A Seamless Payment Service

Evolving Online Payments

A new payment ecosystem is emerging which includes e-commerce, m-commerce and social commerce. Right now, your business wants to improve the conversion rates on its existing e-commerce capability and is considering adding these new sales channels, but is unsure of how best to move forward in this quickly evolving market.

What's more, the onus is on companies to make purchases easy and flexible regardless of the application or device serving up the front end to consumers who see computers, smartphones, tablets, etc. as interchangeable and expect the same buying experience across the board.

Your business needs to have a technical platform that seamlessly integrates all sales channels in a unified way and supports all payment models to give your customers a consistent, one-click, buying experience from within your brand.

Solution

App55 is an enterprise class solution to this payment problem that works across multiple devices. We take care of all security, banking and technical issues so your business can accept payments via your website, mobile website or app on any internet enabled device – giving you a global capability for a secure, quick and seamless checkout process.

Low Operational Impact

In addition to the technical solution, the design of App55's payment service minimises any changes needed to your existing back office operations, by letting you keep your merchant acquirer and existing interfaces and operations, such as reporting.

There is no need to install any software on your servers as our service is accessed through the web. In addition, you don't have to store your customers' confidential payment card details; we encrypt and hold them on your behalf on our PCI compliant servers.

Positive Integration

New platforms and services normally require significant integration into your corporate environment which represents overhead in terms of change, effort and risk. We see this as negative integration and our technical and operational design aim is to reduce that to zero. Positive integration is how we describe any additional work you may decide to perform over and above the payment platform integration that brings business benefit. An example could be removing redundant software and process from your operations backend. We will work closely with you to make this happen.

Partners

App55 has partnered with WorldPay who provide access to the card-processing network and have a global card processing capability. Our other major partner is Valtech, a multi-million Euro public company with global capability. This allows us to perform additional backend systems integration work, have an ongoing local operational support presence around the globe, create or modify your existing smartphone apps and more. Valtech has a heavyweight client list and is leading the digital agency charge. In summary, we will not leave you alone with an API definition and instruction book.

Architecture

The App55 solution is a true multi-tier system where each tier can run on separate servers on different network segments. This approach has led to a highly secure, resilient and scalable solution.

Multi-tier architecture coupled with functional partitioning

The Presentation Tier represents the front end of the system and handles page rendering and navigation of the external interfaces of the service. This tier has been built to allow functional partitioning meaning each service can run on its own web server.

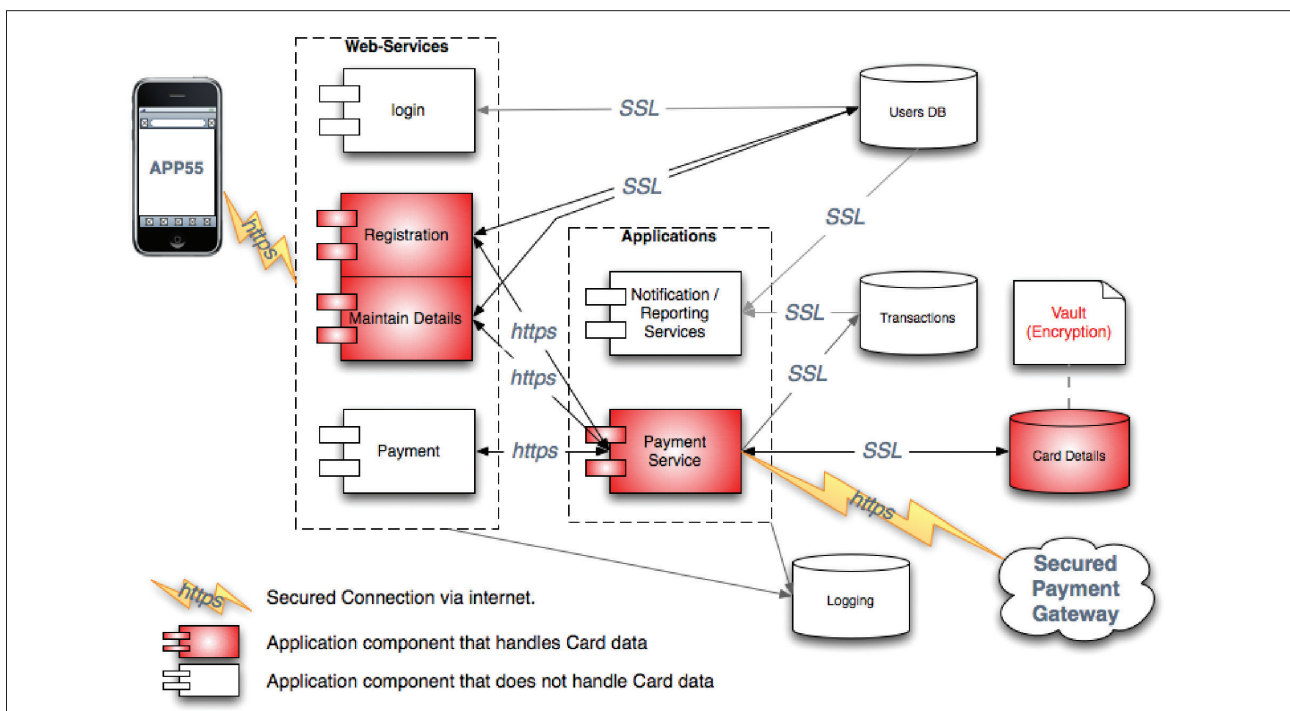
The Service Tier handles the payments, payment card data and notifications (email to end users, etc.). Again, each of these services can run on its own web server.

The Persistence Tier comprises the databases. App55 has created several databases, each one storing a distinct data category: user data, transactions, payment card data and logs. Each database is clustered for resilience, scalability and performance.

Functional partitioning ensures high scalability and security. The services that fall within the scope of PCI (the services and databases that handle payment card data) are deployed on dedicated servers in highly secure, monitored and dedicated network segments.

Furthermore, the system has been designed to allow horizontal¹ partitioning of the data to guarantee further scalability.

High level view of the architecture



1. Horizontal partitioning: records can be stored in different tables or physical databases to handle high volumes.

Resilience

The architecture and implementation of App55's solution ensure that resilience is built into the design and that there is no single point of failure.

The multi-tier architecture with loosely coupled services/components allows the service to run in multiple web containers, multiple application servers and clustered databases, with hardware load balancers distributing the load effectively.

In addition, as an accredited PCI DSS service provider, rigorous backup policies and procedures have been put in place for audit purposes.

Performance

As an enterprise scale payment solution, App55 has been designed to process large numbers of concurrent transactions without adversely degrading the response times, so that the customer will not suffer from any lack of performance.

The architecture ensures that the load is spread over several web servers and databases. In addition, load balancers ensure that the load is distributed evenly across these multiple servers in order to get optimal resource utilization, maximise transaction throughput, minimise response time and avoid overload.

Our system is integrated with the WorldPay servers which already handle 49% of total UK card transactions.

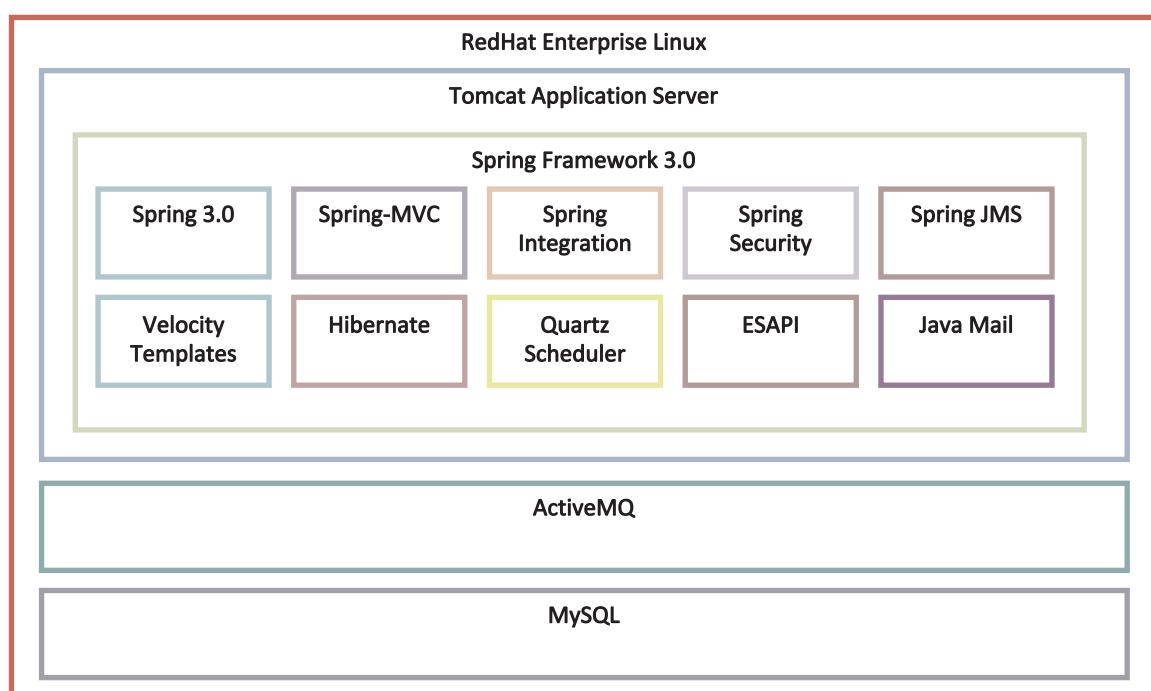
Scalability

Like any high-quality enterprise system that processes large numbers of transactions, App55 architecture is designed with the capability to scale. This was one of the main driving factors when designing the solution.

The architecture was designed to provide a loosely coupled but highly cohesive tier structure. Furthermore, any one of the tiers can be scaled vertically or horizontally.

Software Stack

Below is an illustration detailing the main software components of the system.



RedHat Linux OS

The use of Red Hat Enterprise Linux supports a long-term IT platform, it is deployed on all leading hardware architectures and has a comprehensive support model with 24x7 unlimited incident resolution.

Scalability with RedHat Enterprise Linux can be achieved vertically by increasing the hardware specification of the servers and scaling out horizontally via Red Hat Clustering.

Tomcat Application Server

Tomcat 6 is an open source and best-of-breed application server running on the Java platform and is the probably the most popular Java application server in the world. With support for Java Servlets 2.5 and Java Server Pages 2.1, Tomcat provides a robust environment for developing and deploying dynamic web applications, together with a small memory footprint. The selected technologies for this solution only rely on the Java and Java Servlet standards and the solution is application server vendor agnostic.

Spring 3.0 Framework

Spring is most commonly recognised as being an Inversion of Control (or IoC) container. Spring enables the key benefits of JEE, while minimising the complexity encountered by application code. Spring also provides enterprise services to Plain Old Java Objects (POJOs) in this JEE environment.

Spring MVC

Spring MVC provides a clear separation between presentation, business and navigation logic that enables the construction of a thin and clean presentation tier.

Spring Security

Spring Security provides comprehensive security services for J2EE-based enterprise software applications. Primarily, Spring Security facilitates Authentication and Authorisation. Authentication being the process of establishing who a user is, and Authorisation referring to the process of deciding whether the user has access to a resource.

In terms of Authorisation, Spring Security supports a wide range of authentication models and App55 currently uses the following:

- Form-based authentication (for simple user interface needs)
- Transparent authentication context propagation for Remote Method Invocation (RMI) and HttpInvoker (a Spring remoting protocol)

Spring Integration

Java Message Service (JMS) is used as the Message Oriented Middleware (MOM) for Spring Integration and is used for sending messages between filters. JMS is a messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It allows communication between different components of a distributed application to be loosely coupled, reliable and asynchronous.

Spring Integration provides an asynchronous service bus for processing orders. Spring Integration uses the abstract “pipes-and-filters” model. The “filters” represent any component that is capable of producing and/or consuming messages, and the “pipes” transport the messages between filters so that the components themselves remain loosely coupled.

Hibernate

Hibernate provides a scalable, high performing and transparent persistence layer and provides the ability to develop persistent classes, supports lazy initialisation and works on application server cluster to deliver a highly scalable architecture.

Quartz Scheduler

Quartz is a full-featured, open source job scheduling service that can be integrated with, or used along side virtually any Java EE or Java SE application.

Email Services

Email is required in the application to send notifications, for example account creation and payment transactions.

ActiveMQ Broker

ActiveMQ is an open source message broker that conforms to and implements Java Message Service 1.1. It provides features such as clustering, multiple message stores, and JDBC Backed Persistence.

Velocity Template Engine

Velocity is a Java-based template engine that separates Java code from the web pages or any other document type. The template engine is responsible for merging a template and content to create a document. Velocity is ideally suited to producing dynamic content such as notifications that include mainly static content but have aspects of dynamic data contained within.

MySQL

The MySQL database is the most popular open source database in the world. This is largely due to its high performance, high reliability and ease of use. Many of the world's largest and fastest-growing organisations including Facebook, Google, Adobe, etc. rely on MySQL to power their high-volume web sites, business-critical systems and packaged software.

Client Branded User Interface (CBUI)

A key benefit of the App55 payment solution is its ability to maintain the look and feel (brand) of your website or app across all the payment and registration pages. This is achieved by providing hosted pages that mimic the mobile application/e-commerce web site through use of CSS/html. The consumer only sees your corporate design, logo and content. App55 is a complete white label service unlike other 3rd party payment companies.

Security and PCI compliance

Overview

App55 is a secure and trusted payment solution and is PCI DSS (*Payment Card Industry Data Security Standard*) Compliant. Security considerations are central to the design of App55's solution:

- External and internal encrypted communications (https, TLS)
- Strong encryption of vital data (256 bits TripleDES and SHA-2)
- Security Information and Event Management (SIEM) system that monitors and analyses continuously any activity on the servers
- Intrusion Detection System (IDS)
- Web Application Firewalls (WAF) that protect the front end servers against known attacks: code injection, cross-site scripting (XSS), Cross site request forgery (CSRF), probing, bad user agents...
- Hardware Firewalls with strict rules
- Network segmentation: the payment card details are processed and stored in dedicated network segments with enhanced security
- Monthly vulnerability scan of the servers and network
- Intrusion Prevention System (IPS)
- Hardware Firewall
- Drastic internal security policy; the data never leaves the servers.

Compliance with PCI DSS requires continuous verification, auditing and policy adherence. App55 takes on this responsibility on behalf of your business.

Encrypted Communications

All communications to the production environment are encrypted with HTTPS/TLS (128 bits or above). This rule applies not only to the external connections (WorldPay gateway, mobile applications, etc.) but also to the internal connections (between application tiers). Certificates are used on all servers to ensure mutual authentication of the servers.

The notification service is the one exception to this rule since the emails do not contain sensitive data (e.g. card expiry date).

Strong Data Encryption

The passwords, cardholder data (PAN), responses to the secret question and the encrypting keys are strongly encrypted (TripleDES and SHA-2 256 bits) before being stored in a database or a key store.

The PAN (payment card number) is encrypted using a strong Password Based Encryption (PBE) algorithm. The password, or data-encrypting key, is chosen randomly from one of the thousands of long keys stored in the card database. A random salt is added and one thousand encryption iterations are applied to ensure a strong encryption.

The encrypting keys to encrypt the cardholder data are also encrypted and stored. Again, a strong Password Based Encryption (PBE) algorithm is used, with a random salt and one thousand iterations. The password or key-encrypting key used to encrypt the data-encrypting keys is read from a secure key store that is kept on a separate machine.

In addition, all encrypting keys are modified and all data re-encrypted regularly.

Authentication/Authorisation Roles

A user must authenticate with a username and password to be able to use any application. A role-based access mechanism is employed to provide authorisation of services, with every user being assigned at least one role.

Protection against Attacks

The servers are protected by two dedicated hardware firewalls. The first tier servers which host the web services also host the Web Application Firewalls (WAFs). These analyse all requests sent to the servers for known attacks.

Furthermore, all requests sent to the services are checked against the top 10² attacks listed by the OWASP (Open Web Application Security Project³) project before being processed.

App55's code includes best of breed components such as Java ESAPI (Enterprise Security API⁴) library and Spring Security⁵ framework to ensure that security is an integral feature of the solution.

The PCI compliant environment includes an Intrusion Detection System (IDS) and a Security Information and Event Management (SIEM) that continuously monitor the servers.

2. http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

3. <http://www.owasp.org>

4. http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

5. <http://static.springsource.org/spring-security/site/features.html>

App55 is a secure and trusted payment solution and is fully PCI DSS (Payment Card Industry Data Security Standard) Compliant. Compliance with PCI DSS requires continuous verification, auditing, and policy adherence. App55 manages the responsibility on your behalf and provides a PCI compliant channel for accepting payment for goods or services.

Development process tuned to Quality and Security

Development procedures rigorously adhere to OWASP's development and test guidelines. The process includes daily automated analysis⁶ of the code and publication of internal reports on its quality: security, portability, reliability, maintainability, efficiency and usability. All code is also reviewed manually at regular intervals for more in-depth checks in accordance with OWASP code review guidelines. Automated functional and performance tests are also performed regularly.

Code and Properties Files Protection

Since Java classes can be decompiled, the source code is obfuscated as an additional security measure before the application is deployed. Even if the rigorous security constraints in the hosted environment were compromised all source code is rendered unreadable.

App55 application servers rely on property files for their configuration. Property files are clear text files that are encrypted before being deployed. Only the application and essential application support staff know the password used to encrypt the files. Further, these property files do not contain the keys used to encrypt the data and do not contain the passwords used to access the key stores.

Hosting Partner

Our hosting partner is Datapipe. Datapipe are a leading global provider of managed services and data centre infrastructure for IT and cloud computing. They deliver managed services to some of the largest software, banking, insurance, telecommunications and pharmaceutical companies in the world.

DataPipe's supply managed services to App55 which include hardware provisioning, service management and monitoring, network and security management, server and operating system management, business continuity and disaster recovery, network connectivity, remote infrastructure management, and storage management.

Also DataPipe supply a comprehensive range of security services related to PCI such as proactive firewall, IPS, VPN, and intrusion detection management; and vulnerability scanning and assessment, and patch management.

6. Tools used: CheckStyle, PMD, FindBugs, Squid and Sonar.