http://www.commuforce.com/wp-content/uploads/2018/01/400dpiLogoCropped-114-e1515956204576.png

***CommuServe***

**Architecture Document**

**Version 0.1**

Table of Contents

[Change History 4](#_Toc514408251)

[1 Introduction 5](#_Toc514408252)

[2 Logical Architecture – CommuServe Microservices 6](#_Toc514408253)

[3 Technical Architecture 7](#_Toc514408254)

[4 Technology Components/ Tools 8](#_Toc514408255)

[4.1 CommuServe Application Platform 8](#_Toc514408256)

[4.2 CommuServe Communication Platform 8](#_Toc514408257)

[4.3 Development Environment 8](#_Toc514408258)

[4.4 Deployment Environment 8](#_Toc514408259)

[5 Services Interaction 10](#_Toc514408260)

[5.1 Communication 10](#_Toc514408261)

[6 Detailed Design 10](#_Toc514408262)

[7 DevOps 10](#_Toc514408263)

[7.1 Tools 10](#_Toc514408264)

[7.2 Development Approach 10](#_Toc514408265)

[7.3 Coding Standards 10](#_Toc514408266)

[8 Non-Functional Requirements 10](#_Toc514408267)

[9 Product Testing Approach 10](#_Toc514408268)

[9.1 TDD 10](#_Toc514408269)

[9.2 NFR Testing 10](#_Toc514408270)

[10 Deployment Approach 10](#_Toc514408271)

[11 Licensing Checklist 10](#_Toc514408272)

[12 Risk 10](#_Toc514408273)

[13 Open Items 10](#_Toc514408274)

[14 Appendix 12](#_Toc514408275)

[14.1 Template Cross Platform Compatibility/Open Standard Study 12](#_Toc514408276)

[14.2 Technology Components/ Tools Decisions 13](#_Toc514408277)

[14.2.1 ESB Layer 13](#_Toc514408278)

[14.2.2 Reporting Components 13](#_Toc514408279)

[14.2.3 Workflow/ Rules Engine 13](#_Toc514408280)

[14.2.4 UI Framework 14](#_Toc514408281)

[14.3 Deployment Environment 14](#_Toc514408282)

**Change History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Date** | **Updates** | **Version** | **Author** |
| **1** | **23-May-2018** | **Initial Version** | **1.0** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Introduction**

**CommuServe is a Customer Communication Management (CCM) solution developed by CommuForce Inc.**

**CommuServe is an AI based end to end solution, handling all aspects of CCM from incoming communication to outgoing messages, both for real time applications and large batch processing.**

**The following are the objectives of the CommuServe technical architecture:**

1. **Portability – support multiple 3rd party tools**
2. **Easy to change - Microservices based**
3. **Pluggable – ability to add connectors to multiple input and output channels**
4. **Scalable**
5. **Secure**

**Other deployment related capabilities:**

1. **On-premise or cloud-based deployment supported**
2. **Ability to enable/ disable specific services**

**Logical Architecture – CommuServe Microservices**

**The *CommuServe* product is built based on microservices architecture. All the capabilities of the product are designed as independent services, as depicted in the logical architecture diagram below.**

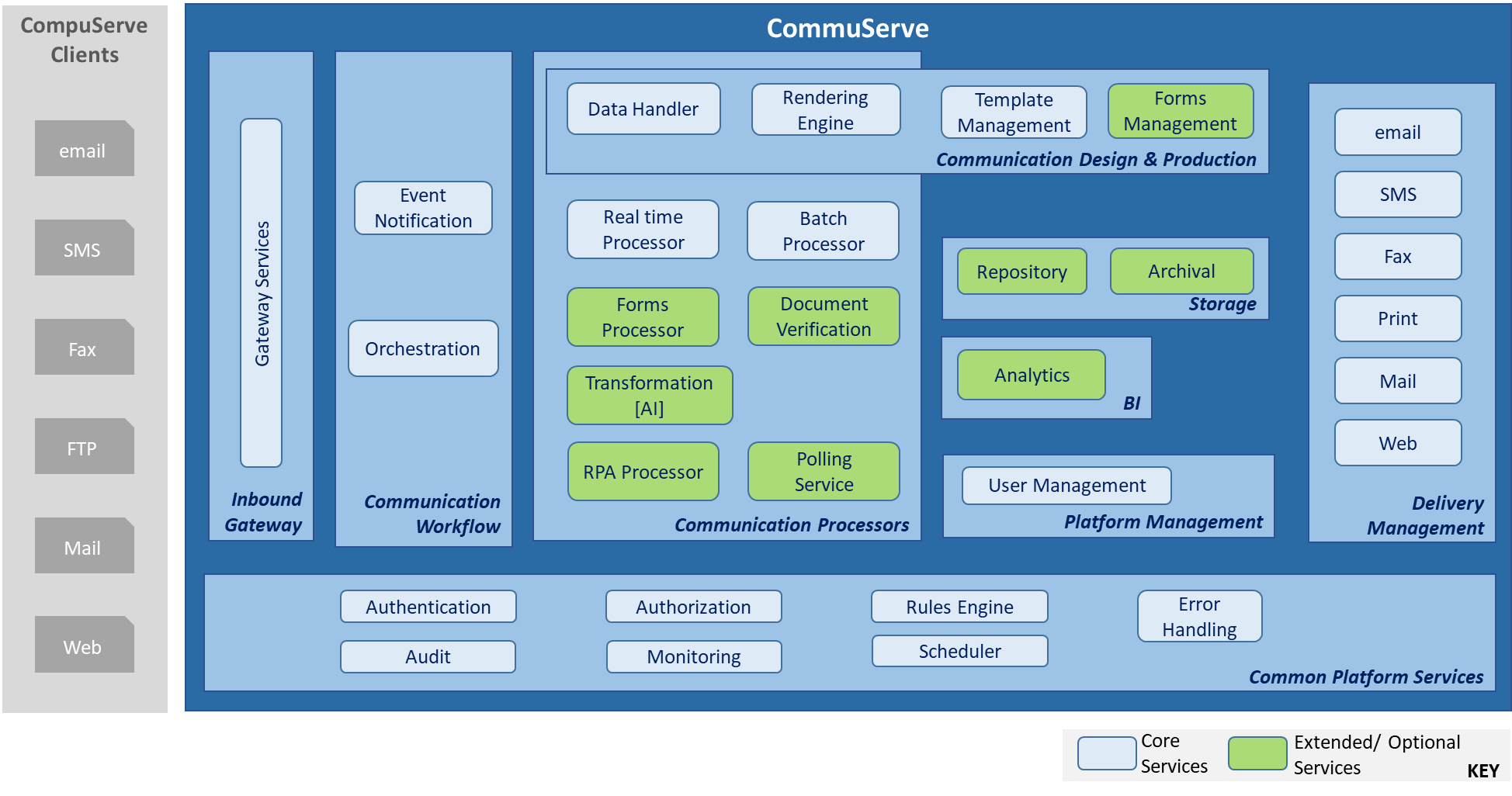


Figure 1 - CompuServe Microservices

**Technical Architecture**

**Technology Components/ Tools**

CommuServe Application Platform

|  |  |  |
| --- | --- | --- |
| **SNo** | **Software** |  |
| **1** | **Microservices Framework** | **Spring Boot** |
| **2** | **UI Framework** | **Backbone/ NodeJS/ AngularJS** |
| **3** | **Database** | **MySQL** |

CommuServe Communication (ESB) Platform

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Software** | **AWS** | **Azure** | **On Premise** |
| **1** | **JBoss Fuse** | **Supported** | **Supported\*** |  |
| **2** | **Mulesoft** | **Supported** | **Supported** |  |
| **3** | **WS02** | **Supported** | **Supported** |  |
| **4** | **Workflow/ Events Database** | **NoSQL** | **NoSQL** | **NoSQL** |

Development Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Software** | **AWS** | **Azure** | **On Premise** |
| **1** | **Eclipse IDE** |  |  |  |
| **2** | **Source Code Repository** | **Github**  **Integrates with**  **AWS Code Commit** | **Github** | **github** |
| **3** | **Continuous Integration** | **Jenkins** |  |  |

Deployment Environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Software** | **AWS** | **Azure** | **On Premise** |
| **1** | **Container** | **Docker** | **Docker** | **Docker** |
| **1** | **Microservices management** | **Amazon EKS (Kubernetes)** |  |  |
| **2** | **Load Balancer & API Gateway** |  |  |  |
| **3** | **Security** |  |  |  |
| **4** | **Logs Management** |  |  |  |
| **5** | **Monitoring** |  |  |  |

**Services Interaction**

Communication

**Detailed Design**

**Refer to the individual service detailed design documents.**

**DevOps**

Tools

Development Approach

Coding Standards

**Non-Functional Requirements**

**Product Testing Approach**

TDD

NFR Testing

**Deployment Approach**

**Licensing Checklist**

**Risk**

**Open Items**

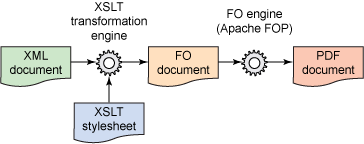
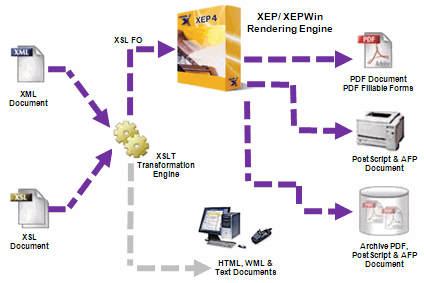
**Appendix**

Cloud Platform

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **AWS** | **Azure** |
| **1** | **Pros** | * **Integrated with wide range of 3rd party tools** | * **Better at Hybrid cloud solutions** * **Better compliance (according to their website)** * **Advantage if already on Microsoft platform, better integration with developer tools** |
| **2** |  |  |  |

**Template Cross Platform Compatibility/Open Standard Study**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNo** | **Software** | **Pros** | **Cons** | **Notes** |
| **1** | **Ecrion Design Studio, Data Modeller** |  |  |  |
| **2** | **JasperReports from Jaspersoft** | * **Popular open source report designer & rendering engine** | * **Report designer, not robust, and may not be appropriate for business users** |  |
| **3** | **BIRT from Eclipse Foundation** |  |  | **Project sponsored by ‘OpenText’** |
| **4** | **Apache FOP** | * **Java XSLT transformer can be leveraged** * **Ecrion is XSL-FO compliant** | * **Outdated, not as popular as CSS for web pages** | Based on w3c XSL-FO standard.  XSL-FO is part of XSL (Extensible Stylesheet Language), a set of W3C technologies designed for the transformation and formatting of XML data. |

**Technology Components/ Tools Decisions**

ESB Layer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SNo** | **Software** | **Pros** | **Cons** | **Notes** | **Selected** | **Date** |
| *****ESB***** | | | | | | |
| **1** | **MuleESB from MuleSoft** |  |  |  |  |  |
| **2** | **JBoss Jfuse** |  |  |  |  |  |
| **3** | **Spring with Apache ActiveMQ** |  |  |  |  |  |
| **4** | **WSO2** |  |  |  |  |  |

Reporting Components

Workflow/ Rules Engine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SNo** | **Software** | **Pros** | **Cons** | **Notes** | **Selected** | **Date** |
| **1** | **JBPM** |  |  |  |  |  |
| **2** | **Drools** |  |  |  |  |  |

UI Framework

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SNo** | **Software** | **Pros** | **Cons** | **Notes** | **Selected** | **Date** |
| **1** | **Backbone** |  |  |  |  |  |
| **2** | **Node JS/ ext js** |  |  |  |  |  |

Deployment Environment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SNo** | **Software** | **Pros** | **Cons** | **Notes** | **Selected** | **Date** |
| *****Container***** | | | | | | |
| **1** | **Docker** | * **Popular tool** | * **Community version does not support enterprise level OS such Red Hat Linux** * **Security features available only in licensed version** |  |  |  |

**Open Items**