

Solution Design Template

<Name of application/solution assessed>

MM/DD/YYYY

Prepared by Click and Edit Solution Architect

Table of Contents

[Confidentially Agreement 3](#_Toc192657684)

[Document Purpose 3](#_Toc192657685)

[Document Owner 3](#_Toc192657686)

[Project Summary 3](#_Toc192657687)

[Project/Solution Description 3](#_Toc192657688)

[Stakeholders 4](#_Toc192657689)

[Introduction 5](#_Toc192657690)

[1. Architectural Representation 5](#_Toc192657691)

[Definitions 5](#_Toc192657692)

[1.1 Current-State Reference 5](#_Toc192657693)

[1.2 Target-State Design – Note to be completed post move of workload into new Firewall Zones 6](#_Toc192657694)

[1.3 Architectural Goals and Constraints 6](#_Toc192657695)

[2.0 Logical View – **provide if you have it not critical for network segmentation.** 6](#_Toc192657696)

[3.0 Process View 8](#_Toc192657697)

[3.1 Non-Functional Requirements 9](#_Toc192657698)

[4.0 Data View 10](#_Toc192657699)

[4.1 HA Requirements – Need to know for Way2Zero 11](#_Toc192657700)

[5.0 Technical Infrastructure & Hosting 12](#_Toc192657701)

[6.0 Security View 14](#_Toc192657702)

[Authors/Reviewers 15](#_Toc192657703)

[Appendix 15](#_Toc192657704)

Document Details

## Confidentially Agreement

This document is classified as Regions Financial Corporation Confidential and is provided to the intended recipients for informational purposes only and may not be used, published or redistributed or disclosed without the prior consent of Regions Enterprise and Security Architecture. No part of it may be reproduced or disclosed without the prior consent of Regions Financial Corporation. Disclosure of this document is solely for review and design as is required for Regions Financial Corporation to meet pertinent architecture and engineering requirements and corporate policy compliance. No part of this document shall be taken to create or modify any contractually binding obligations upon Regions Financial Corporation.

## Document Purpose

This document is intended as a design and review for the ***Regions’ Enterprise and Security Architecture***, ***Cloud Center of Excellence***, ***Software & Performance Engineering, System Engineering, and Project Management teams*** to ensure the project describes the various aspects of the system design that are considered to be **architecturally significant**.

***Note to Author:***

*The following template shall be used when Solution Architecture is needed for completion of a Regions Technology Project as defined by the ITPMO Project Governance & Framework. The Enterprise/Solution Architect must approve any modifications to the overall structure of this document. This cover page is not a part of the final template and should be removed before publication.*

*Any green italicized text throughout this template is provided solely as background information to assist you in creating this document. Please delete this page and all green italicized text, as well as the instructions in each section, prior to submitting this document.*

* *This document template provides baseline information needed for an architectural design document.*
* ***The author may add additional content beyond what this document explicitly specifies.***

## Document Owner

Project team owns the Solution architecture document and is responsible for Gate updates. The template is owned by Regions’ Enterprise and Security Architecture.

Template = Outline of Solution architecture document

Solution Architecture Document = Filled in Template.

# Project Summary

## Project/Solution Description

|  |
| --- |
| Description |
| Please provide sufficient details to help us understand the business requirement and justification for this solution being reviewed including, but not limited to:   * the catalyst for this solution * the reason the application / solution was selected and/or why it is needed * when is implementation expected |

## Stakeholders

|  |  |  |
| --- | --- | --- |
| Role | Name | Phone Number / Email |
| Project Sponsor |  |  |
| Project Manager |  |  |
| Business Contact |  |  |
| Business IT Contact |  |  |
| External Vendor |  |  |

# Introduction

The Solution Architecture provides a comprehensive architectural overview of the ***ACBS System Architected*** . It presents a number of different architectural views to depict the different aspects of the system. This document will define goals of architecture, the use cases supported by the system, architectural styles and components that have been selected to enable the system. This document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.

# Architectural Representation

## Definitions

|  |
| --- |
| *This section contains important definitions of terms related to the solution. Application codes and the full application name that are part of this solution are included in this section for clarity.* |

|  |  |  |
| --- | --- | --- |
| Item | Term | Definition |
| DEF-001 | ACBS | Advanced Commercial Banking System |
| DEF-002 | **APIGW** | API Gateway |

## 1.1 Current-State Design (In Scope)

The current-state architecture spans all architecture view models (logical, and physical) and also relates to the operational solution intent which contains operational details. The full solution current-state is included in the solution’s blueprint, and can be found: <Additional Reference Material and Recommendation Link>

|  |
| --- |
| ***Related Artifacts:***   * *Include a diagram of the current-state application & technology architecture design for the technology being enhanced or updated through this project. If there is not a current state diagram because this is a new system, indicate as such in this section.*   ***Refer to RACI Chart for roles and responsibilities.*** |

A screenshot of a computer

AI-generated content may be incorrect.

**Figure:** Application Cooperation **Current-State Architecture**

## 1.2 Target-State Design – Note to be completed post move of workload into new Firewall Zones

The target-state architecture spans the (physical) model only for network segmentation.

|  |
| --- |
| ***Related Artifacts:***   * *Include a diagram of the target-state deployment model.*   ***Refer to RACI Chart for roles and responsibilities.*** |

**Please include Target State Deployment Diagram**

## 1.3 Architectural Goals and Constraints

List Architectural Goal & Constraints

*There are some key requirements and system constraints that have a significant bearing on the architecture. They are: (Define Examples Below)*

1. *The system is meant as a proof of concept for a more complete project prediction system to be built in the future. Therefore, one of the primary stakeholders in this document and the system are future architects and designers, not necessarily users as is normally the case. As a result, one goal of this document is to be useful to future architects and designers.*
2. *The system will be written using Microsoft .NET technologies but will use an open-source RDBMS system (MySQL) for data persistence and will be deployed to a Linux webserver running Mono. These special deployment requirements require additional consideration in the development of the architecture.*
3. *The system must communicate with multiple third-party APIs, Client and Predictive. Defining how the system interfaces with these third-party systems is a primary concern of the architecture.*
4. *Section 3.3 of the Software Requirements Specification outlines a number of anticipated changes that the application could face over time. One of the primary goals of the system architecture is to minimize the impact of these changes by minimizing the amount of code that would need to be modified to implement them. The architecture seeks to do this using modularization and information hiding to isolate components that are likely to change from the rest of the system.*

# 2.0 Logical View – **provided (not critical for network segmentation).**

|  |
| --- |
| ***Audience:*** *All Stakeholders.*  ***Area:*** *The logical view is concerned with the functionality that the system provides to end-users. Also describes the most important use-case realizations and business requirements of the system.*  ***Related Artifacts:***   * *Provide a logical component diagram* * *Link to Related Artifacts*   ***Refer to RACI Chart for roles and responsibilities.*** |

*Overview Example*

*The main goal of the logical view is to define the components that will make up the system and to define the interfaces through which they will communicate and interact with one another. The primary decision-making factor behind defining the system components is the need to isolate the components that are likely to change from the rest of the system. By clearly defining the interfaces of these components and hiding their internal implementations from the rest of the system, the impact of expected changes can be minimized.* ***Software Requirements Specification*** *outlines the changes that are likely to be made to the system. A summary of these changes and how the logical decomposition of the architecture addresses them is as follows:*

1. *Changes to the Client API*
   1. *The architecture addresses this by implementing the calls to the Client API in a Client component (see figure 5.1). The rest of the application will communicate with Client only through the interface exposed by this component. Therefore, any changes to the system to deal with changes in the Client API need only be made in the internal implementation of this component.*
2. *Changes to the Predictive API*
   1. *Similar to the above, this is addressed by implementing calls to the Predictive API in a Predictive Client component (see figure 5.1). Changes required to deal with changes to the Predictive API need only be made in the internal implementation of this component and not to the rest of the system.*
3. *Changes to the metrics used to construct the predictive model*
   1. *All business logic dealing with Predictive, including what metrics are sent to it to construct its model is isolated in a Prediction Service component (see figure 5.1). Changes to the metrics used to construct the model need only be made in this component without affecting the rest of the system.*
4. *Changes to the content of the prediction report generated for the user*
   1. *This report is generated in the Web Portal module (see figure 5.1). As such changes to its content need only be made in this module.*

A diagram of a computer process

AI-generated content may be incorrect.

***Figure 4.7:******Logical Component Diagram*** *Example*

***Element Responsibilities Example***

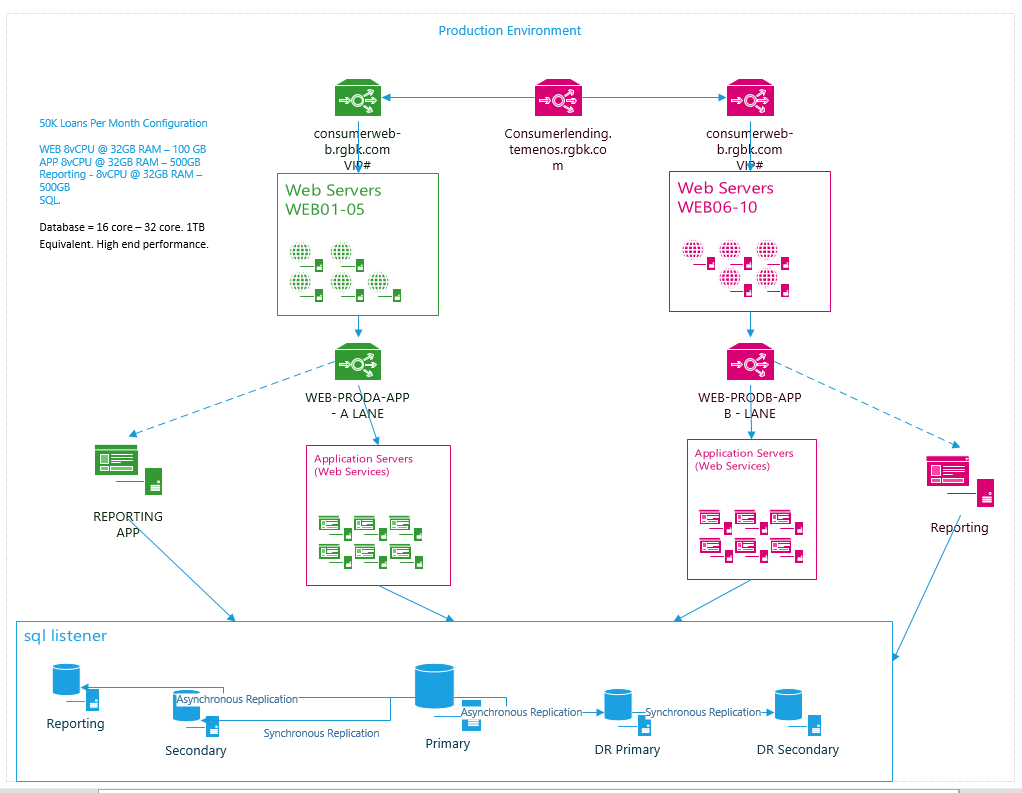
|  |  |
| --- | --- |
| ***Element Examples*** | ***Responsibilities*** |
| ***Example Client*** | * *Provide an interface for Client OAuth authentication.* * *Handle all communication with the Client REST API* * *Provide a native .NET interface for other components to use to access the Client API.* |
| ***Example Data Retrieval Service*** | * *Provide an interface for retrieving data from Client* * *Manage the saving of data retrieved from Client* |
| ***Client Data Store*** | * *Persist data to MySQL data store.* * *Provide a query interface to the MySQL data store.* |
| ***Web Portal*** | * *Present users with an HTML-based user interface accessible through a web browser.* * *Interact with other components in the system to allow users to authenticate with Client, choose a client project for analysis, and analyze the chosen project.* |
| ***Predictive Client*** | * *Handle all communication with the Predictive API.* * *Provide a native .NET interface for other components to use to access the Predictive API.* |
| ***Prediction Service*** | * *Provide an interface to get a prediction for a given Client project.* * *Provide an interface for providing training data to Predictive.* |
| ***Application Data Service*** | * *Provide an interface to save and retrieve application specific data, for example past prediction reports.* |
| ***Application Data Store*** | * *Persist application specific data such as past prediction reports to the MySQL data store.* * *Provide a query interface to the application specific MySQL data store* |

# 3.0 Process View - *(NA – Out of Scope for this Phase)*

|  |
| --- |
| ***Audience:*** *Architects, Integrators, Software Engineering, Performance Engineering, Subject Matter Experts*  ***Area:*** *The process architecture considers non-functional requirements, such as availability, scalability, interoperability, security, and performance. It addresses issues of concurrency and distribution, of system’s integrity, of fault-tolerance, and how the main abstractions from the logical view fit within the process architecture.*  ***Related Artifacts:***   * *Provide a process view diagram* * *Provide a high availability diagram* * *Link to Related Artifacts to support the process view*   ***Refer to RACI Chart for roles and responsibilities.*** |

The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. The process view addresses concurrency, distribution, integrator, performance, and scalability, etc.

*High Availability View (Example) This very much depends on whether we can find HA pairs in the logs that are sent to us. Typically, that would happen only when you are in HOT-HOT configuration. If not, then we may need access to configuration documents. (NA – Out of Scope for this Phase)*



*High Availability View Example*

## 3.1 Non-Functional Requirements - NA

Review list of NFRs for considerations that need to be noted: [NFR LIST](http://sharepoint/sites/acoe/Repository1/NFR/NFR%20Evaluation%20v1.9.docx). We need to understand this as routing through firewalls can add additional hops which can add some minor latency.

### 3.1.1 Performance Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Target** | **Description** |
| **ONLINE LOAD** | | | | |
| NFR-1 | Concurrent Users |  | System/Application should be able to support target concurrent users at peak load hour without degradation to performance.  **Example**  Able to support 3000 concurrent users during peak load hour. | |
| NFR-2 | Peak Request / Transactions per Second |  | System/Application should be able to support target peak transactions volume by TPS without degradation to performance.  **Example**  Able to support 200 peak transaction volume by total requests/transactions per second - TPS. | |
| NFR-3 | Average Response Times |  | System/Application should respond below target on average response times.  **Example**  Average transaction response times should be below 3 seconds. | |
| NFR-4 | 90th Percentile Response Times |  | 90% of system/application transactions should respond at or below a defined target response time.  **Example**  Transactions' 90th percentile response times should be at or below 5 seconds. | |
| NFR-5 | Maximum Response Times |  | System/Application should respond below target for all transactions. Target is the maximum allowable transaction response times.  **Example**  Transactions maximum response times should not be above 10 seconds. | |
| NFR-6 | Standard Deviation on Response Times |  | System/Application should have standard deviation below target for all transactions. The helps understand the application is responding within tolerable variance.  **Example**  Standard Deviation should not be above 2 seconds. | |
| NFR-7 | Error Count |  | The error count percentage during test should not exceed the target.  **Example**  Error count percentage should be less than 1% of total transactions. | |
| NFR-8 | System Resource Utilization |  | The system resources utilization (CPU/Memory) should not exceed target average and target maximum.  **Example**  Web CPU and Memory should be less than 60% average and less than 85% maximum.  App CPU and Memory should be less than 60% average and less than 85% maximum. | |
| NFR-9 | Reliability / Survivability |  | System/Application should pass all NFRs for long duration test, this helps validate longevity. Note NFR1 and NFR2 will be at 80% and not at peak load. This will simulate above average load for longer duration and mimic production.  **Example**  All above NFRs should pass for test duration 12 hours. | |
| NFR-10 | Failover |  | System/Application should recover system failure with existing resources available within time defined in target.  **Example**  Should recover from App layer 1 instance brought down during test in 15 minutes. | |
| NFR-11 | Scalability |  | System/Application should able to scale without impacting the performance as defined above NFRs.  **Example**  System should be able to scale 400% of peak hour load. | |
| **BATCH LOAD** | | | |
| NFR-12 | Concurrent Threads |  | System/Application should be able to support target concurrent threads for batch.  **Example**  Able to support 50 concurrent batch threads during peak load. | |
| NFR-13 | Peak Batch Size |  | System/Application should be able to support target peak batch size total requests.  **Example**  Able to support peak batch size of 10,000 total requests. | |
| NFR-14 | Batch Completion Time |  | System/Application should be able to complete batch execution within the target timeline allowed.  **Example**  Maximum allowed batch completion time is 2 hours and 30 minutes. | |
| NFR-15 | System Resource Utilization |  | The system resources utilization (CPU/Memory) should not exceed target average and target maximum.  **Example**  CPU and Memory should be less than 60% average and less than 85% maximum. | |

# 4.0 Data View - (NA - Out of Scope - Phase 3)

|  |
| --- |
| ***Related Artifacts:***   * *Provide a data flow diagram / static data structure diagram* * *Provide event management requirements (if applicable). Are you connecting to MQ, Kafka etc.* * *Link to Related Artifacts* * *Data Flow – If data source is internal to include ci code in box. Also, indicate batch/streaming/real-time in data flow arrows.* |

Data Flow Diagram – Note this can be an external artifact or included here.

*Example*:

*This diagram illustrates how data will flow between external entities/ source system and the XXX application. Ovals represent external entities, tall boxes represent points where data is processed, arrows show the direction of data flow, and short boxes represent persistent data stores.*

A diagram of a project

Description automatically generated

Data Flow Diagram example

NA For this Phase We will create tags this Phase 1 and design for Way2Zero in Phase 2 along with rules for Segmentation and microsegmentation (Out of Scope NA Phase 3)

## 4.1 HA Requirements – Need to know for Way2Zero

### Database Administration

|  |  |  |
| --- | --- | --- |
| # | Question | Answer |
|  | Production Database High Availability (HA) – Specify if your app requires HA. (Yes/No)   * If you opt out of production database high availability, then the business unit understands and accepts that a manual restore of the database is required and could take several hours depending on the database size. No Sev-1 is generated if production database outage is related to opting out of database high availability. | *Yes* |

# 5.0 Technical Infrastructure & Hosting - (Mainframe apps out of scope)

Section details specific mainframe system hosting details. Delete section if solution is distributed, or cloud native.

|  |  |  |
| --- | --- | --- |
| # | Question | Answer |
|  | Is data sent or received by the mainframe? (Yes/No) | *No* |
|  | Protocols used by the application to communicate with the mainframe. | *No* |
|  | Estimated 1-year data growth | *No* |
|  | Estimated 3-year data growth | *No* |
|  | Where data is pulled from | *No* |
|  | Will the application run on zLinux? (Yes/No) | *No* |

### 5.1.1 On-premise Hosting

Section details specific distributed system hosting details. Do not fill out section if solution is mainframe, or cloud native.

|  |  |  |
| --- | --- | --- |
| # | Question | Answer |
|  | Web Service – If applicable provide **web service** and **version** the app is certified. (Apache, IHS, IIS, OHS) | *YES* |
|  | Middleware- If applicable provide **Middleware** and **version** the app is certified. (.Net, WebSphere, Weblogic, JBOSS, Other explain) | *YES* |
|  | Operating System - Provide **OS** and **version** the app is certified. (AIX, Windows, RHEL, zLinux, Other explain) | *YES – if data comes from ServiceNow* |
|  | Database Technology – Provide **database technology** and **version** app is certified. (DB2, SQL, Oracle, Other explain) | *Yes if data comes from ServiceNow* |
|  | Citrix Farm - Specify if you require your application to run on Regions Citrix farm. (Yes/No) | *Yes* |
|  | Any additional Server/Hardware info? | *Yes* |

#### Development Environment – Can be surmised from deployment diagram but this would be helpful.

Detail out the characteristics of the development environment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Type* | *Quantity* | *# CPUs* | *Memory (GBs)* | *Storage (GBs/TBs)* | *Notes* |
| *Web* | *2* |  |  |  |  |
| *App* | *2* |  |  |  |  |
| *Database* | *1* |  |  |  |  |
| *Other* |  |  |  |  |  |

#### Test Environment

Detail out the characteristics of the Test environment being set up

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Type* | *Quantity* | *# CPUs* | *Memory (GBs)* | *Storage (GBs/TBs)* | *Notes* |
| *Web* | *1* |  |  |  |  |
| *App* | *1* |  |  |  |  |
| *Database* | *1* |  |  |  |  |
| *Other* |  |  |  |  |  |

#### Production Environment

Detail out the characteristics of the production environment being set up

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Type* | *Quantity* | *# CPUs* | *Memory (GBs)* | *Storage (GBs/TBs)* | *Notes* |
| *Web* | *2* |  |  |  |  |
| *App* | *2* |  |  |  |  |
| *Database* | *2* |  |  |  |  |
|  |  |  |  |  |  |
| *Other Convergence Diagram* |  |  |  |  |  |

**Network Data Flow and Convergence View along with Top Talkers - Enabling Segmentation – Defines application importance**

A network of colorful dots and lines

AI-generated content may be incorrect.

***Archetypes****:*

*-* ***description****:*

*standard enterprise structure with separate UI, API, and database components.*

***expected\_hosts:***

*- App Server*

*- DB Server*

*- Web Server*

Recommendations  
  
The 7 Rs of cloud migration represent a set of strategies for moving applications and data to the cloud. These strategies, which include Rehost, Replatform, Refactor, Repurchase, Retire, Retain, and Relocate, offer varying degrees of complexity and potential benefits. Choosing the right approach depends on an organization's specific needs, infrastructure, and goals.

* [**Replatform (Lift, Tinker, and Shift)**](https://www.google.com/search?rlz=1C1ONGR_enUS1160US1160&cs=0&sca_esv=b3f6a4e4ace9f962&sxsrf=AE3TifNpRKQinP1h6pBOBqoDDGYzHFDvSQ%3A1752849083871&q=Replatform+%28Lift%2C+Tinker%2C+and+Shift%29&sa=X&ved=2ahUKEwia2qXXz8aOAxVMl-4BHX_PIzYQxccNegQIJBAB&mstk=AUtExfC0wf8oDnt_Ksu-RzVUif9Qz9bZFqYhnB07mw3E4gB7qupkJdxrV55PZ26m7o2E0749y8mSwDFvHk8hlXQtnjy_nwiXu8-K80aF2AJRlnhTzN00txhPNhqvoOfenrbF5Yw2uzEj2HgnMsO4EPIqw7EnOuM6OZkqYsGjZM8I3SNzcNE&csui=3)**:**

Making minor modifications to an application during migration to optimize it for the cloud environment. This offers a balance between speed and cloud optimization.

**Key Benefits of Cloud Migration:**

Cloud migration offers numerous potential benefits, including:

* **Cost Reduction:**

Reduced infrastructure costs, potentially lower operational expenses, and pay-as-you-go pricing.

* **Increased Scalability and Flexibility:**

Easily scale resources up or down based on demand, adapting to changing business needs.

* **Improved Performance and Reliability:**

Leveraging cloud providers' infrastructure for better performance and higher availability.

* **Enhanced Agility and Innovation:**

Faster development cycles and the ability to experiment with new technologies.

* **Better Security:**

Cloud providers often offer robust security features and compliance certifications.

* **Simplified Management:**

Cloud management tools can streamline IT operations.

#### Disaster Recovery Environment – We need to understand DR connectivity

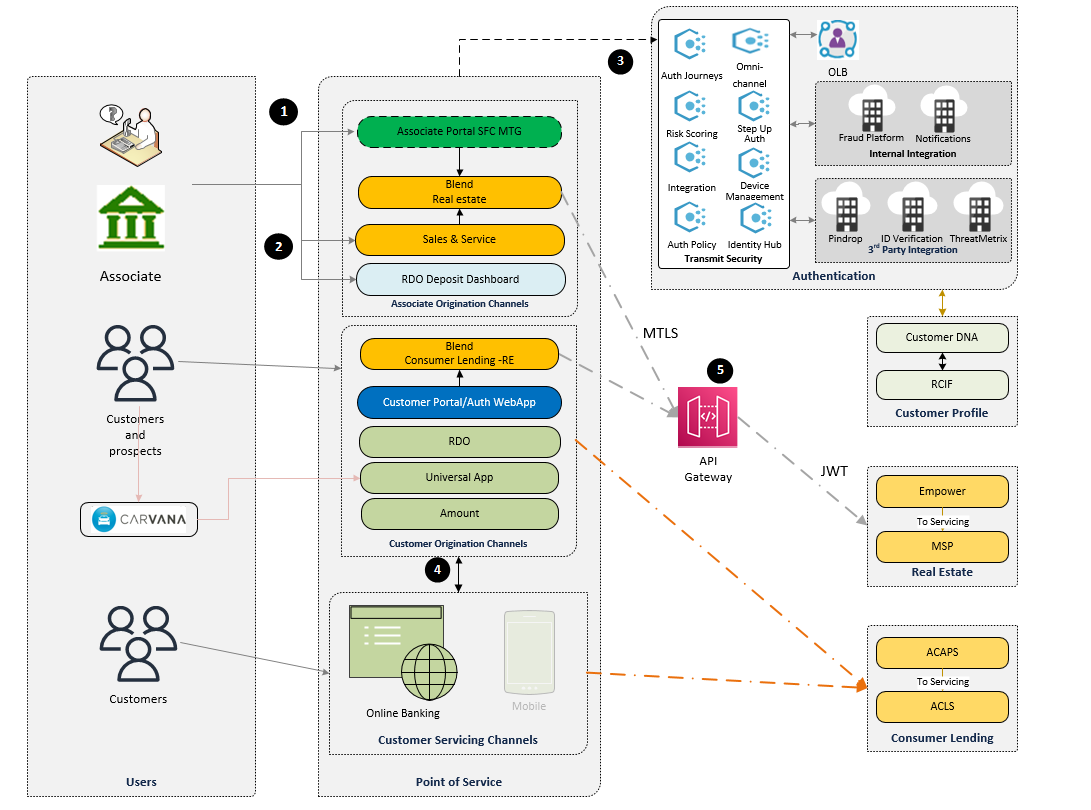
Detail out the characteristics of the Disaster Recovery environment being set up

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Type* | *Quantity* | *# CPUs* | *Memory (GBs)* | *Storage (GBs/TBs)* | *Notes* |
| *Web* | *2* |  |  |  |  |
| *App* | *2* |  |  |  |  |
| *Database* | *2* |  |  |  |  |
| *Other – LB, APIGW, Kafka* | *1* |  |  |  |  |

# 6.0 Security View – NA for this phase

### Network Security Diagram – Can be deployment diagram

Include Any Networking Security Diagrams for Review with Security Architecture



Application Security View Example

A screenshot of a computer

Description automatically generated

Network Security Example

### Network Security

Detail all network security requirements in this section. Contact [FWIS@regions.com](mailto:FWIS@regions.com) for additional security requirements.

|  |  |  |
| --- | --- | --- |
| # | Question | Answer |
|  | Primary Accessibility – (Internal or External) |  |
|  | Is the network deployed in the external DMZ, Internal network, or externally hosted(Internal, externally hosted or DMZ) |  |
|  | Application layer(s) requiring load balancing (N/A, Web, App, Both) |  |
|  | Load Balancing type required by application (N/A, DMZ/Internet Facing, Internal Network, None) |  |
|  | Does the application require SSL sessions? If yes, where? (N/A, Load Balancer, Web Server, Both) |  |
|  | If externally hosted is network connectivity over VPN, private circuit, or public Internet? |  |
|  | Any additional Firewall info. |  |

# Authors/Reviewers

|  |  |  |
| --- | --- | --- |
| Role | Completed By | Date |
| Network Security Architecture | Zachary Dewey | 07/18/2025 |
| Network Infrastructure |  |  |

# Appendix

Add any relevant data to project in appendix that may be needed. Links to larger documents/ vendor material/ references etc.