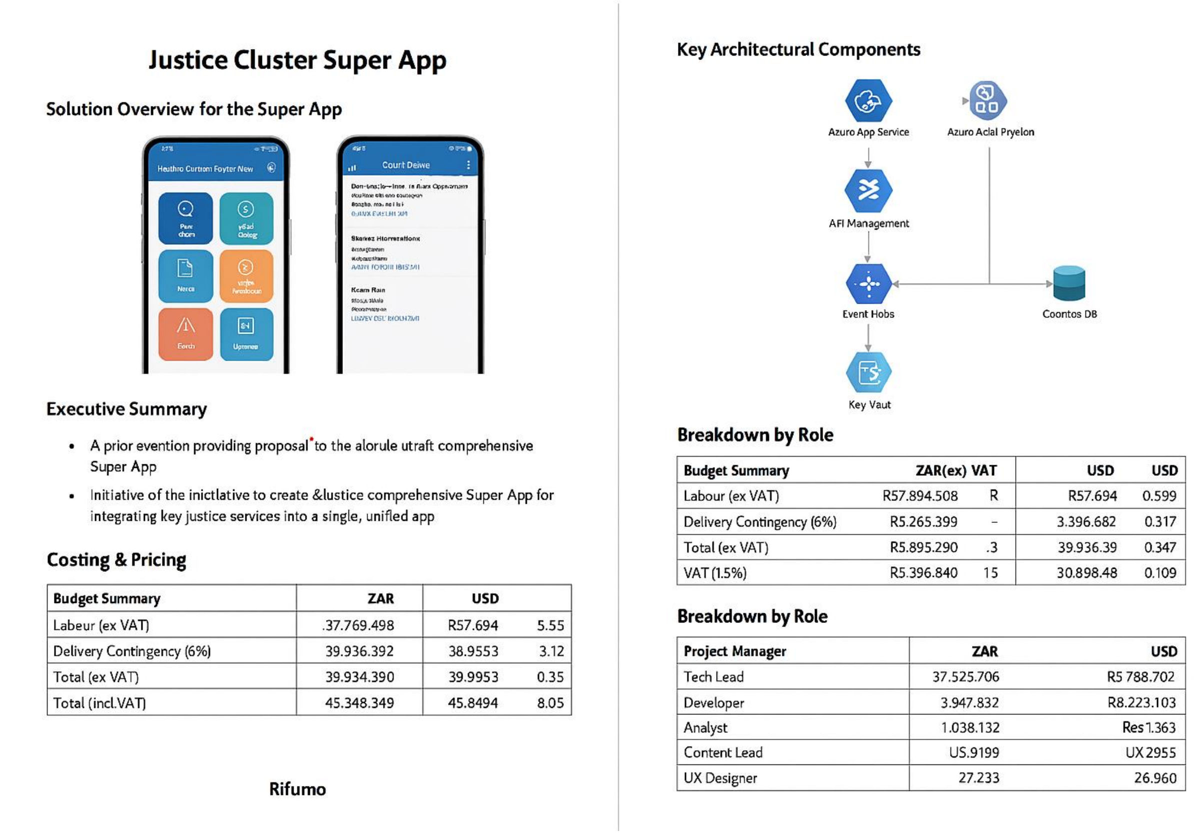
RIFUM
EMPOWERMENT HOLDINGS (PTY) Ltd

Justice Cluster Super App

Table of Contents

1 Executive Summary

City Super App and Enterprise Integration



Justice Cluster Super App

Solution Overview for the Super App

Executive Summary

- A prior evention providing proposal˚to the alorule utraft comprehensive Super App
- Initiative of the inictlative to create &lustice comprehensive Super App for integrating key justice services into a single, unifled app

Costing & Pricing

Budget Summary	ZAR	USD	
Labeur (ex VAT)	.37.769.498	R57.694	5.55
Delivery Contingency (6%)	39.936.392	38.9553	3.12
Total (ex VAT)	39.934.390	39.9953	0.35
Total (incl.VAT)	45.348.349	45.8494	8.05

Rifumo

Key Architectural Components

Breakdown by Role

Budget Summary	ZAR(ex) VAT		USD	USD
Labour (ex VAT)	R57.894.508	R	R57.694	0.599
Delivery Contingency (6%)	R5.265.399	–	3.396.682	0.317
Total (ex VAT)	R5.895.290	.3	39.936.39	0.347
VAT (1.5%)	R5.396.840	15	30.898.48	0.109

Breakdown by Role

Project Manager	ZAR	USD
Tech Lead	37.525.706	R5 788.702
Developer	3.947.832	R8.223.103
Analyst	1.038.132	Res 1.363
Content Lead	US.9199	UX 2955
UX Designer	27.233	26.960

Vision

To revolutionize urban living by creating a comprehensive, user-centric Super App that seamlessly integrates essential city services and private enterprise offerings. This innovative platform will streamline daily activities, enhance citizen experiences, and drive economic growth.

Key Objectives

1. Unified Platform: Develop a single, intuitive app that consolidates diverse city services, including transportation, payments, healthcare, education, and more.

2. Enhanced Citizen Experience: Provide a frictionless, personalized experience through AI-powered recommendations, real-time updates, and proactive notifications.
3. Economic Growth: Foster innovation and entrepreneurship by enabling businesses to integrate their services into the app, reaching a wider audience and boosting revenue.
4. Data-Driven Insights: Utilize advanced analytics to gain valuable insights into citizen behavior, optimize service delivery, and identify emerging trends.
5. Secure and Scalable Infrastructure: Implement robust security measures and a scalable architecture to ensure data privacy and system reliability.

Enterprise Integration Strategy

1. API Gateway: Establish a centralized API gateway to manage and secure external API traffic.
2. API Design and Development: Develop well-defined, standardized APIs that adhere to industry best practices (e.g., RESTful, GraphQL).
3. Security and Authentication: Implement robust authentication and authorization mechanisms to protect sensitive data and control access to APIs.
4. Monitoring and Logging: Monitor API performance, identify potential issues, and gather valuable usage data.
5. Collaboration and Partnerships: Foster strong partnerships with enterprises to facilitate seamless integration and co-create innovative solutions.

Expected Benefits

- Improved Citizen Satisfaction: Increased convenience, efficiency, and accessibility to city services.
- Economic Development: Stimulated economic growth through increased business opportunities and innovation.
- Data-Driven Decision Making: Enhanced ability to make informed decisions based on real-time data and analytics.
- Enhanced City Image: Position the city as a technologically advanced and citizen-centric metropolis.

By executing this strategic initiative, we will position our city at the forefront of urban innovation, empowering citizens and businesses alike to thrive in the digital age.

2 Super App and Enterprise Integration Model

2.1 Enterprise Integration Model in Azure

This document outlines a comprehensive enterprise integration model leveraging Azure services to facilitate seamless communication between internal systems and external applications. The model aims to expose APIs securely, enable API discovery, and ensure robust monitoring and security.

Key Components

2.1.1.1 API Management:

- Acts as a gateway for all API traffic, both internal and external.
- Provides features like rate limiting, authentication, and authorization to control access and protect APIs.
- Enables API discovery through a developer portal, allowing external applications to easily find and consume APIs.
- Can transform and enrich API responses to meet specific needs.

2.1.1.2 Stream Event Processing (Event Hubs/Kafka):

- Handles real-time data ingestion and processing.
- Captures events from various sources (e.g., IoT devices, system logs) and routes them to appropriate destinations.
- Enables near-real-time responses and triggers workflows based on event occurrences.

2.1.1.3 Workflow (Logic Apps):

- Orchestrates complex business processes and integrates disparate systems.
- Automates tasks, triggers actions based on events, and manages data flows.
- Can be used to implement API-driven workflows, where APIs are invoked to perform specific actions.

2.1.1.4 Messaging (Service Bus):

- Facilitates asynchronous communication between components.
- Enables reliable messaging and ensures message delivery even in case of failures.
- Can be used to decouple components and improve system scalability.

2.1.1.5 Data Factory:

- Orchestrates data integration and ETL processes.
- Extracts, transforms, and loads data from various sources (e.g., databases, files, APIs) into data stores.
- Can be used to populate data lakes and data warehouses for analysis.

2.1.1.5.1 Synapse Analytics:

- Provides a unified analytics platform for big data and real-time analytics.
- Combines data ingestion, data preparation, data warehousing, and machine learning capabilities.
- Can be used to analyze API usage data and generate insights.

2.1.1.5.2 Data Lake:

- Stores large volumes of raw data in a scalable and cost-effective manner.
- Enables data scientists and analysts to explore and analyze data.
- Can be used to store historical API usage data for long-term analysis.

2.1.1.5.3 FTP Server:

- Provides a secure file transfer mechanism for exchanging data with external systems.
- Can be used to transfer API specifications, data files, and other artifacts.
- Service Catalog:
- Manages and catalogs available services, including APIs.

2.2 Provides a central repository for service descriptions, documentation, and usage guidelines.

2.2.1 Service Registry:

- Stores metadata about services, including APIs, and their location.

2.2.2 Common Dictionary:

- Defines a shared vocabulary and data model for consistent data interpretation across systems.

- Ensures data consistency and interoperability.

2.3 External App Discovery and Consumption

- API Management Portal: External developers can discover available APIs through a user-friendly portal.

- API Documentation: Clear and concise documentation is provided for each API, including usage guidelines, request/response formats, and error handling.
- API Keys and Tokens: Secure authentication mechanisms are implemented to protect API access.
- Rate Limiting: Limits the number of API calls to prevent abuse and ensure fair usage.

2.4 Internal System Consumption

- Service Registry: Internal systems can query the service registry to locate and consume required APIs.
- API Gateway: Internal systems can access APIs through the API gateway, which enforces security policies and monitors usage.
- Client Libraries: SDKs or libraries can be provided to simplify API consumption from internal systems.

2.5 Security and Monitoring

- Authentication and Authorization: Implement strong authentication and authorization mechanisms to protect API access.
- Data Encryption: Encrypt sensitive data both in transit and at rest.
- Threat Protection: Utilize Azure Security Center to monitor for threats and vulnerabilities.
- Logging and Monitoring: Log API requests, responses, and errors for analysis and troubleshooting.
- Auditing: Track API usage and access to identify security incidents and compliance issues.

3 Solution Overview

3.1 Enterprise Integration Key Features

An **Enterprise Integration Framework** serves as the backbone for enabling seamless data flow and system interaction within a city's IT ecosystem. Built on **Azure's cloud-native tools**, the framework supports integration, data processing, and secure collaboration among internal city departments, external partners, and citizen-facing applications.

3.1.1 Cloud-Based API Management (Azure API Manager):

- Centralized API management for secure and scalable integration.
- Developer portal for API subscriptions and usage documentation.
- Support for both internal APIs (departmental use) and external APIs (partners and public developers).

3.1.2 Event Processing and Messaging:

- **Azure Event Hub and Kafka:** High-throughput event streaming for real-time data ingestion, e.g., IoT devices, traffic sensors.
- Messaging support for asynchronous communication between systems.

3.1.3 Data Orchestration and Storage:

- **Azure Data Factory:** Automated workflows for data movement and transformation.
- **Azure Synapse Analytics:** Centralized data warehouse for analytics and reporting.
- **Data Lake:** Unified storage for structured and unstructured data.

3.1.4 Workflow Automation:

- Low-code/no-code workflows for operational efficiency using Azure Logic Apps or similar tools.
- Automation of approvals, incident reporting, and citizen requests.

3.1.5 Common Dictionary and Data Standardization:

- Standardized data definitions and formats to improve interoperability.
- Ensures data consistency across systems and APIs.

3.1.6 Enterprise Connectors:

- Prebuilt integrations for popular enterprise applications (e.g., SAP, Dynamics, Salesforce).
- FTP for handling batch processes and legacy systems.

3.2 Benefits

3.2.1 Operational Efficiency:

- Centralized integration reduces silos, enabling faster communication across departments.
- Scalable and reusable APIs reduce redundant development efforts.

3.2.2 Enhanced Developer Experience:

- Developer portal provides a self-service interface for API discovery, testing, and subscription.
- Simplified onboarding for external partners and developers.

3.2.3 Real-Time Data Processing:

- Event-driven architecture supports real-time decision-making for traffic management, emergency responses, and citizen services.

3.2.4 Improved Data Insights:

- Centralized data aggregation enables advanced analytics, reporting, and AI-driven insights to optimize city operations.

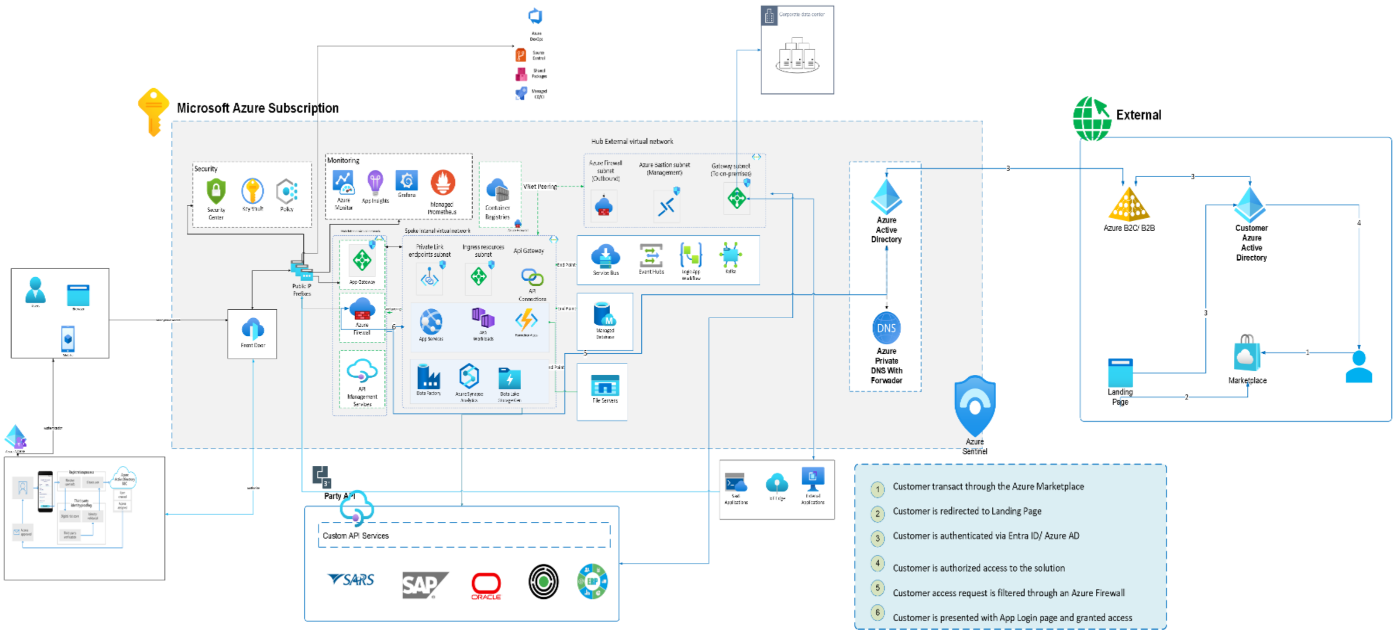
3.2.5 Flexibility and Scalability:

- Azure's serverless architecture ensures high availability and scalability to handle fluctuating workloads.

3.2.6 Security and Compliance:

- Built-in security for APIs, data access, and workflows ensures compliance with regulations.

3.3 Architecture Enterprise Integration



1. Customer transact through the Azure Marketplace
2. Customer is redirected to Landing Page
3. Customer is authenticated via Entra ID/ Azure AD
4. Customer is authorized access to the solution
5. Customer access request is filtered through an Azure Firewall
6. Customer is presented with App Login page and granted access

Here's a technical breakdown of the **enterprise integration architecture** you described, detailing each component's purpose and how they interact:

3.3.1.1 Hub-Spoke Network Design

- **Purpose:** Centralized network topology where the **Hub Virtual Network (VNet)** acts as the core, providing shared services and connectivity to various **Spoke VNets** (hosting different workloads and applications).
- **Uses:**
  - **Hub VNet:** Hosts shared services like firewalls, DNS forwarders, API gateways, and ExpressRoute gateway for on-premises connectivity.
  - **Spoke VNets:** Isolated environments for specific workloads like AKS, App Services, VMs, and PaaS services. Improves security and scalability.

3.3.1.2 Hub-Ext-VNet Components

These components provide the foundational services for external and internal connectivity.

3.3.1.2.1 Firewall

- **Purpose:** Acts as a centralized point of control for inspecting and securing network traffic.
- **Uses:**
  - Filters and monitors traffic between the internet, on-premises, and cloud environments.
  - Enforces inbound and outbound traffic rules for Spoke VNets.

3.3.1.2.2 Application Gateway

- **Purpose:** Provides HTTP-based load balancing and acts as a web application firewall (WAF).
- **Uses:**
  - Handles Layer 7 routing for web traffic.
  - Protects web applications using WAF features (e.g., against SQL injection, cross-site scripting).

3.3.1.2.3 Azure API Manager

- **Purpose:** Centralized API management for internal and external APIs.
- **Uses:**
  - Publishes APIs securely with usage policies.
  - Allows API discovery and subscription through a developer portal.
  - Enables analytics and monitoring of API traffic.

3.3.1.2.4 DNS Forwarder

- **Purpose:** Resolves DNS queries for on-premises, cloud, and external domains.
- **Uses:**
  - Provides consistent DNS resolution across VNets and ExpressRoute.
  - Ensures proper resolution of Private Link-enabled PaaS services.

3.3.1.3 Spoke VNet Components

Spoke VNets host workload-specific services while leveraging shared resources in the Hub VNet.

3.3.1.3.1 Azure Kubernetes Service (AKS)

- **Purpose:** Orchestrates and scales containerized applications.
- **Uses:**
  - Hosts microservices and applications.
  - Integrates with Private Link for secure connections to PaaS services.
  - Uses Azure Monitor with Prometheus and Grafana for observability.

3.3.1.3.2 Azure App Service

- **Purpose:** PaaS offering for hosting web and mobile applications.
- **Uses:**
  - Integrates with VNet using VNet Integration or Private Link.
  - Securely connects to databases and APIs in the Hub VNet.

3.3.1.3.3 Virtual Machines (VMs)

- **Purpose:** Hosts workloads requiring full OS-level control.
- **Uses:**
  - Runs legacy applications or specialized workloads.
  - Directly connects to on-premises systems via ExpressRoute.

3.3.1.3.4 Private Link Endpoints

- **Purpose:** Provides private access to Azure PaaS services.
- **Uses:**
  - Secures connections to services like Data Factory, Data Lake, Synapse, Service Bus, etc., without public exposure.
  - Eliminates data exposure on the public internet.

3.3.1.4 PaaS Services via Private Link

The architecture uses Azure PaaS services connected through Private Link to ensure secure, private communication.

3.3.1.4.1 Data Factory

- **Purpose:** Orchestrates ETL (Extract, Transform, Load) processes.
- **Uses:**
  - Moves and transforms data between on-premises and cloud environments.
  - Integrates with Data Lake, Synapse, and Databases for seamless workflows.

3.3.1.4.2 Data Lake

- **Purpose:** Provides scalable storage for structured and unstructured data.
- **Uses:**
  - Stores raw data for processing and analytics.
  - Connects to Synapse for querying and transformation.

3.3.1.4.3 Azure Synapse

- **Purpose:** Combines data warehousing and big data analytics.
- **Uses:**
  - Runs complex queries across large datasets.
  - Powers dashboards and reports for city operations.

3.3.1.4.4 Service Bus

- **Purpose:** Messaging service for decoupling application components.
- **Uses:**
  - Supports asynchronous messaging for workflows and system integration.

3.3.1.4.5 Kafka in HDInsight

- **Purpose:** High-throughput distributed messaging system for real-time streaming.
- **Uses:**
  - Handles IoT data ingestion and processing.
  - Integrates with Event Hubs for streaming pipelines.

3.3.1.4.6 Logic Apps

- **Purpose:** Automates workflows and integrations.
- **Uses:**
  - Connects SaaS and on-premises applications.
  - Orchestrates business processes like incident management or approvals.

3.3.1.4.7 Event Hubs

- **Purpose:** Real-time data ingestion for streaming scenarios.
- **Uses:**
  - Processes telemetry data from IoT devices.
  - Feeds data into Kafka, Synapse, or Event Grid.

3.3.1.5 Azure Container Registry

- **Purpose:** Stores and manages container images.
- **Uses:**
  - Hosts AKS container images securely.
  - Ensures version control and deployment consistency.

3.3.1.6 Monitoring and Observability

3.3.1.6.1 Azure Monitor

- **Purpose:** Centralized monitoring solution for performance and health.
- **Uses:**
  - Collects logs and metrics from all Azure resources.
  - Generates alerts for proactive issue resolution.

3.3.1.6.2 Prometheus and Grafana

- **Purpose:** Advanced monitoring and visualization tools.
- **Uses:**
  - Prometheus collects and queries metrics from AKS workloads.
  - Grafana visualizes performance metrics with custom dashboards.

3.3.1.7 Security and Authentication

3.3.1.7.1 Azure Active Directory (Azure AD B2B)

- **Purpose:** Identity provider for secure authentication and collaboration.
- **Uses:**
    - Enables authentication for external service providers.
    - Enforces role-based access control (RBAC) for internal resources.

3.3.1.7.2 Firewall and Private Link Security

- **Purpose:** Protects against unauthorized access and data exposure.
- **Uses:**
    - Private Link ensures all sensitive communications occur over private channels.
    - Firewalls enforce rules for inbound and outbound traffic.

3.3.1.8 Hub-Internal VNet with On-Premises Connectivity

3.3.1.8.1 ExpressRoute

- **Purpose:** Dedicated connection between on-premises data centers and Azure.
- **Uses:**
    - Ensures low-latency, high-bandwidth connectivity.
    - Transfers sensitive data securely without traversing the public internet.

3.3.1.8.2 Application Gateway

- **Purpose:** Load balancing and application security for internal web services.
- **Uses:**
    - Provides internal load balancing for on-premises applications.
    - Protects internal APIs and services with WAF features.

3.3.1.8.3 Firewall

- **Purpose:** Protects the internal network.
- **Uses:**
    - Controls traffic between on-premises systems and cloud resources.

3.3.1.9

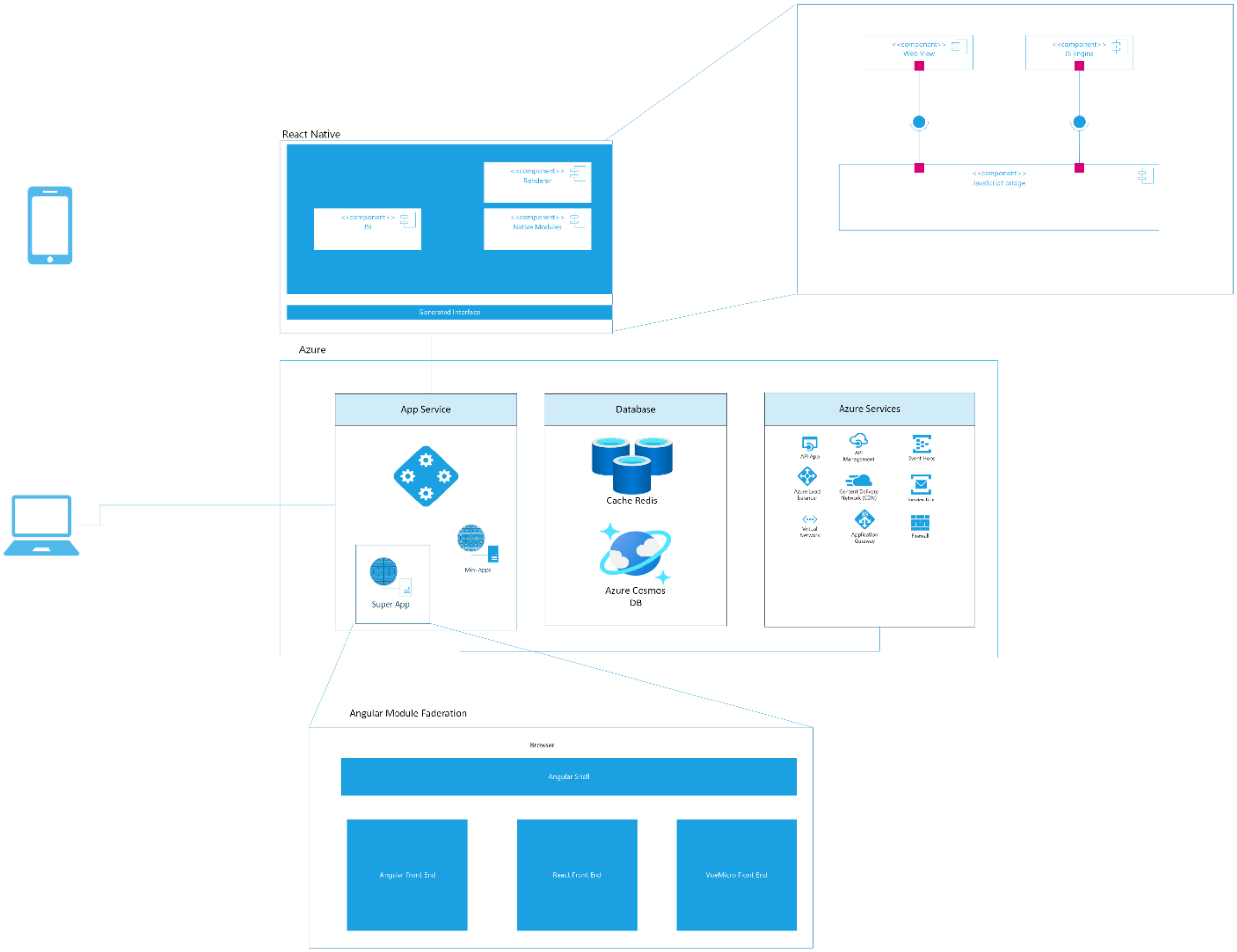
3.3.1.10 End-to-End Workflow

- **Data Ingestion:** IoT devices send telemetry data to Event Hubs or Kafka.
- **Processing:** Data flows through Data Factory for transformation or directly to Synapse/Data Lake for storage.

- **Application Workloads:** Apps in AKS or App Service consume processed data using APIs.
- **Communication:** Service Bus handles messages, while APIs provide data for external applications.
- **Insights:** Azure Monitor, Prometheus, and Grafana provide real-time dashboards for system health and metrics.

4.1. Solution Overview for the Super App

The proposed Super App leverages **Angular Module Federation** for building a dynamic, modular shell on the web and integrates with a **React Native** mobile application using WebView. This architecture supports micro frontends hosted across various environments, including Azure App Service, Azure Kubernetes Service (AKS), and on-premises infrastructure, ensuring flexibility, scalability, and seamless citizen engagement.



3.3.2 Key Architectural Components

3.3.3 Shell Application

The core of the Super App is built using Angular, with **Module Federation** enabling dynamic integration of micro frontends. The shell acts as the container that provides consistent navigation, shared services, and a unified user interface.

- **Responsibilities:**
  - Centralized routing to ensure users can navigate between services seamlessly.
  - Shared utilities, including authentication, theming, and common UI components.
  - Dynamic loading of micro frontends for performance optimization and modular updates.

3.3.4 Micro Frontends

The Super App is divided into independent, service-specific modules called **micro frontends**, each responsible for a distinct function like payments, incident reporting, or public transportation tracking. These micro frontends are developed and deployed independently, allowing updates without impacting the entire application.

- **Features:**
  - Built with Angular and integrated into the shell using Module Federation.
  - Hosted across different environments:
    - Azure App Service for straightforward, scalable deployments.
    - AKS for containerized workloads requiring advanced orchestration.
    - On-premises hosting for hybrid scenarios or compliance needs.
- **Examples of Micro Frontends:**
  - A citizen dashboard showing personalized notifications and upcoming actions.
  - Real-time updates for public transit schedules and maps.
  - Utility bill payment systems or permit application services.

Mobile Application

The mobile version of the Super App is built using React Native, with **WebView** as the mechanism to display web-based micro frontends. This approach ensures consistency between web and mobile platforms while allowing the native app to utilize device-specific features like GPS, camera access, and push notifications.

- **Key Features:**

    - React Native provides device-level functionality and app store deployment.
    - WebView loads micro frontends directly from their hosted environments, ensuring consistency with the web version.
    - Authentication and session management are unified across WebView and native layers.

3.3.5 Hosting and Deployment

The Super App and its micro frontends can be hosted on Azure's scalable infrastructure or hybrid on-premises systems to suit varying operational needs:

- **Azure App Service:**
    - Ideal for hosting lightweight micro frontends and the main shell.
    - Provides built-in scaling and high availability.
- **Azure Kubernetes Service (AKS):**
    - Recommended for micro frontends requiring advanced container orchestration.
    - Supports high-performance workloads and complex integrations.
- **On-Premises Hosting:**
    - For services requiring local data residency or compliance with regulatory requirements.
    - Connects to Azure using Private Link or VNet Integration for secure communication.

3.3.6 Core Features

3.3.6.1 Web Application

- Modular shell architecture with dynamic loading of micro frontends.
- Centralized user authentication and state management.
- Independent deployment and scaling of micro frontends.

3.3.6.2 Mobile Application

- Native-like experience with React Native while leveraging WebView for cross-platform consistency.
- Support for device-specific functionalities like offline capabilities and hardware integrations.

3.3.6.3 Micro Frontend Approach

- Service-specific micro frontends enable independent team development and deployment.

- Minimal coupling ensures that a failure or update in one service doesn't disrupt the overall system.
- Unified design system and shared libraries ensure a consistent user experience.

3.3.6.4 Infrastructure and Security

- **Private Link:** Ensures secure access to APIs, databases, and other backend services.
- **Hybrid Deployment Support:** Enables flexibility to deploy in Azure or on-premises as required.
- **CI/CD Pipelines:** Automates build, testing, and deployment for each micro frontend.

3.3.7 Benefits

3.3.7.1 Scalable and Modular Design

- Micro frontends are self-contained, allowing for independent development and deployment. Teams can scale services individually based on demand.

3.3.7.2 Unified Experience Across Platforms

- The web and mobile applications deliver a consistent look and feel, improving usability. WebView in the mobile app ensures reuse of existing micro frontends for efficiency.

3.3.7.3 Efficient Resource Utilization

- Hosting flexibility allows services to be deployed in cost-effective environments while ensuring high performance.

3.3.7.4 Streamlined Updates

- Individual micro frontends can be updated without requiring changes to the shell or other modules, reducing downtime.

4.4.2.6.5. Future-Proof Architecture

- Built with modern frameworks like Angular Module Federation and React Native, the architecture is extensible and adaptable to new services or features.

4.4.2.6.6. Enhanced Citizen Services

- Provides a centralized platform for accessing multiple city services, improving convenience and user satisfaction.

3.3.8 Typical User Workflow

3.3.8.1 Web Access:

- The user accesses the Super App via a browser. The Angular shell dynamically loads the relevant micro frontends, such as payment services or traffic updates, based on the user's interactions.

4.4.2.2. Mobile Access:

- The React Native app is used for mobile interactions. Native modules handle notifications or device-specific actions, while the micro frontends load in WebView to maintain a consistent service offering.

4.4.2.3. Backend Integration:

- APIs hosted in Azure App Service, AKS, or on-premises provide data and services to the frontends. Backend integration is secured through Azure Private Link and identity management is handled by Azure AD B2C/B2B.

4.4.2.4. Monitoring and Maintenance:

- Azure Monitor, along with Grafana and Prometheus, ensures application health and provides insights into performance, enabling proactive resolution of issues.

Conclusion

The proposed Super App architecture creates a flexible and scalable platform for delivering citizen services. The combination of **Angular Module Federation**, **React Native**, and Azure's hosting options ensures seamless integration of web and mobile interfaces. This design empowers city administrations to adapt quickly to changing needs, deliver consistent and efficient services, and enhance citizen engagement.

1.1.1. On-going Support Services

Our support services packages will allow you to utilise Rifumo's expertise in the way that best suits Customer's business needs. Whether the support need for the Dynamics 365 Business Central ERP solution, or eCommerce applications assistance, Rifumo will fully provide support.

A Service Level Agreement can be used for any of the following:

- Break/fix Issues
- Solution Enhancements
- Customisations

- Development
- On-site or off-site Consultancy
- On-site User Training
- Telephone Support
- Advisory Service
- Remote Assistance
- Upgrades

There are also additional guidelines to the support and response that Rifumo follows to provide Customer with the best value for the SLA and from available support personnel. Standard solution deployment procedure requires any program error or new requirement to be;

- Detailed and documented.
- Error to be investigated and determined in test environment.
- Solution to be tested in test environment.
- Test environment solution to be signed off before deployment on production system after thorough testing.
- Solution to be deployed in production.
- Final user-acceptance conducted.
- Solution signed off.

5. Other Requirements.

Microsoft Dynamics 365 for Business Central offers role-based security functionality, which will separate key user access based on job roles within Customer as an organisation and within the ERP solution. An integral feature of the ERP solution is the integration of the entire system with the general ledger accounts to minimise the multiple posting of data and centralization of organizational information across the various and unique business functions within Customer.

5.4.2. Azure Data Residence Requirements.

Microsoft has South Africa as one of its Azure Regions, an Azure Data Region is a combination of multiple centres, and South Africa has 2 data centres (North - Gauteng and South – Western Cape) therefore if Customer should choose the cloud solution, Dynamics 365 Business Central will be hosted in one of these South African Data regions.

5.4.3. Information Security Requirements.

Each end-user and key-user will be assigned a user ID and password as the first level of security. In addition, the user IDs, roles will then be assigned to the Customer user groups based on the specific tasks each user must perform within the system. This second level of security will determine what each user views and has access to viewing, capturing, and posting.

Posting restriction can further be applied on open or closed accounting periods by user groups or by business unit. Further controls may be provided through the implementation of approval workflows on key processes within the ERP system, which will further restrict the unauthorised manipulation of the system's data. Each posted transaction will however be tagged with the user ID of the user responsible for posting the transaction, thus adding to the comprehensive audit trail capabilities of Dynamics 365 Business Central.

The use of automatically generated One-Time Pin codes using the user's mobile number can add strength to the standard security features offered by Business Central.

5.4.4. Disaster Recovery.

The Data Recovery plan in a Cloud based environment will be determined by engaging data centre owner, that is, Microsoft. However, all data will be stored securely on the Microsoft Data Centres.

7.3. Out of Scope or Exclusions

Any area or deliverable not explicitly included in the areas in scope section of this document is deemed out of scope for Rifumo's implementation of the Super App and Enterprise Intergration.

Specific items seen to be out of scope for this project are listed in the following table. Please note that the below exclusions are not exhaustive, and further discussions may be necessary to fully understand and document all exclusions once the project analysis phase has started.

Item	Description
Data governance policy implementation	While the scope of work includes an analysis of existing data governance, security and best-practice policies, the actual implementation of these policies or changes to them is outside the scope of the project.
Source system modifications	Any required changes or modifications to the source systems from which the data is drawn will be outside the scope of this project.

Data cleansing in source systems	While data quality checks and cleansing activities are included within the data warehouse, the cleansing of data at the source systems is not included.
Non-priority system data integration	The data integration for systems that are not designated as 'priority' in the initial stages of the project may not be included in the scope.
Custom analytics or reporting requests	The development of custom analytics or reports outside of the agreed-upon priority and standard analytics and reports.
End-User Hardware or Software	Provision or upgrade of end-user hardware or software tools to access the analytics and reporting platforms is not included.
Azure Consumption Costs	While the Azure platform will be used for data storage, any costs associated with the increase in data storage due to data ingestion, data storage overages, or related to specific Azure services are not covered in this SOW and will be for Customer's account.
Infrastructure	Hardware procurement and sourcing will be for the ownership of **Error! Unknown document property name.**.
Physical data security	While data security within the Azure platform is a part of the project, physical security of the servers or data centres (in the case of hybrid solutions) is not covered in this SOW.

Table 4: Out of Scope

6. Project Dependencies

The following project assumptions apply to the Rifumo's efforts and associated scope of work shared in this Smart E-Invoice solution implementation proposal:

Item	**Description**
Error! Unknown document property name. **Involvement**	▪ **Error! Unknown document property name.** will assign a project sponsor to the project. ▪ **Error! Unknown document property name.** will assign a project manager/lead to the project, with the responsibilities for the project manager to include the following: ○ Acting as a single point of contact into **Error! Unknown document property name.** for the project team. ○ Management of **Error! Unknown document property name.** assigned project escalations. ○ Arrange meetings with **Error! Unknown document property name.** stakeholders as required.

	▪ **Error! Unknown document property name.** resource availability is the responsibility of assigned **Error! Unknown document property name.** project manager.
Error! Unknown document property name. **Resources**	▪ **Error! Unknown document property name.** to ensure that Rifumo has access to all relevant resources for the duration of this engagement. o All business unit or process owners that will be impacted by the solution. o A single individual from all operational areas who will be using the solution.
Error! Unknown document property name. **Facilities**	▪ **Error! Unknown document property name.** will provide the required facilities, namely: o A board room or training room capable for hosting workshops/meetings. o A workspace for Rifumo resources for the periods of the engagement. o A whiteboard with markers.
Data Availability and Quality	The project depends on the availability and quality of data from the source systems. Poor data quality may require additional effort in data cleansing that may not be included in this project scope.
Access to existing systems	The team requires sufficient access to existing data, analytics, and reporting systems to conduct thorough analyses.
Timely Approvals	Gaining timely approvals for deployments, changes, and system access is critical to avoid project delays.
Technical infrastructure	The existing technical infrastructure needs to be robust enough to support the new data platforms and systems.
Source System stability	The stability and availability of source systems from which data is being ingested are vital to maintain the planned project schedule.
Security compliance	Compliance with data security policies is critical, and any adjustments or changes to meet these requirements are dependent on the internal security teams.
Software and tools	The project is dependent on the availability and performance of Microsoft Azure, PowerBI and Business Central SaaS. Any outages, updates, or changes to these tools could impact the project timeline and outcomes.
DevOps	▪ Rifumo will utilise their Azure DevOps instance for management of the activities to ensure that the deliverables are met. o **Error! Unknown document property name.** is liable for any licensing and setup required for this.

	○ License costs may be deferred if **Error! Unknown document property name.** has applicable licensing that can be utilised. ○ Alternatively, Rifumo will align with any such tooling that **Error! Unknown document property name.** uses and will train the RIfumo team. ○ Training costs are for **Error! Unknown document property name.**’s expense. ▪ Azure DevOps license pricing is subject to the USD / ZAR exchange rate fluctuations. **Information:** The Azure DevOps features, and pricing are as per the following link: https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/ ▪ Should **Error! Unknown document property name.** opt to utilise an on-premise TFS environment, the time put aside for setting up the Azure DevOps environment will instead be utilised to plan how this will be done and will factor into account what aspects will need to be configured and the current setup. ○ An output of this will be an estimate to get the same functionality setup with TFS.
Travel	▪ Travel & Accommodation is excluded from the price quoted whereby **Error! Unknown document property name.** will be invoiced monthly for such expenses. ○ Where so required the Rifumo team members allocated to this project will be working from the premises of **Error! Unknown document property name.**. ○ Travel is calculated as the extended distance beyond the resource’s travel to the nearest Rifumo office, in a single direction only. ○ The cost will be based on the official AA rates.

Table 5: Project Dependencies

7. Technical Approach and Methodology

1.1. Methodology

Our implementation methodology for Customer will be the trusted and proven Microsoft SureStep Agile/Waterfall Methodology. SureStep methodology is an industry best-practices methodology developed by Microsoft to facilitate the on-time and on-budget completion of projects, to reduce project risks for the partner and customer and to ensure that the solution meets the customer's highest.

A waterfall methodology is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks.

The diagram below is graphical display of the phases in the SureStep Methodology.

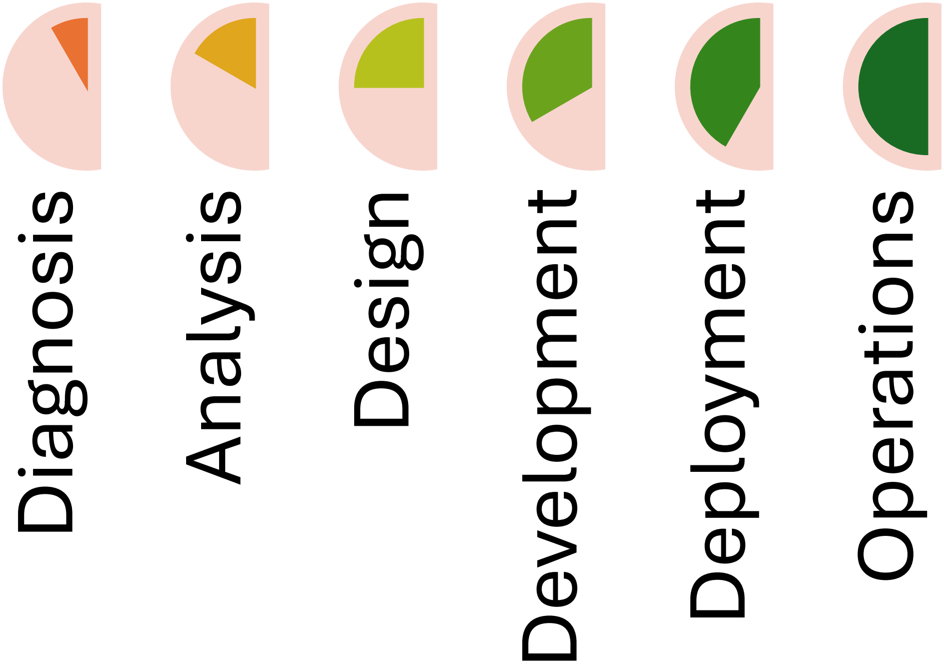


FIGURE 8: SURESTEP PROJECT LIFECYCLE PHASES

The benefits of the Microsoft SureStep method for our customers are:

- Allows you to keep control of project efficiency and success
- Supports close collaboration, communication and coordination in ongoing projects
- Ensures that the process is fully documented in a standardized, integrated manner
- It can be scaled to any size of business
- It can be adapted to any project type (local, global, central or decentralized)

1.1.1. Diagnostic Phase

The initial phase in our Methodology will be the Diagnostic Phase, where we will facilitate sessions with all stakeholders and decision makers to re-affirm and finalise requirements as stated in the tender document.

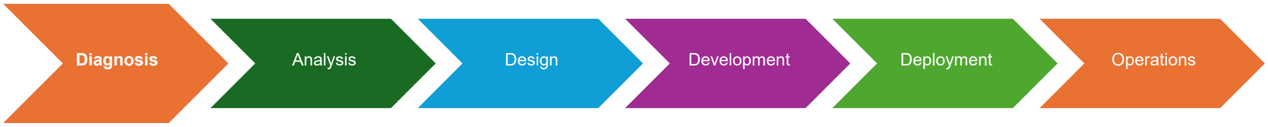


FIGURE 9: SURE STEP DIAGNOSIS PHASE

Key Activities/ Deliverables

The goal of the Diagnostic phase is to gather enough information to define the high-level project scope and make a confident proposal for the remaining phases of the implementation. The major deliverable is the implementation proposal and contract.

1.1.2. Analysis Phase

The Analysis phase represents the official start of the implementation project. The overall goal is to define and agree upon all the business requirements related to the new Microsoft Dynamics™ solution. This includes the business and data migration requirements, as well as integration requirements. The major deliverable is the comprehensive functional requirements document.

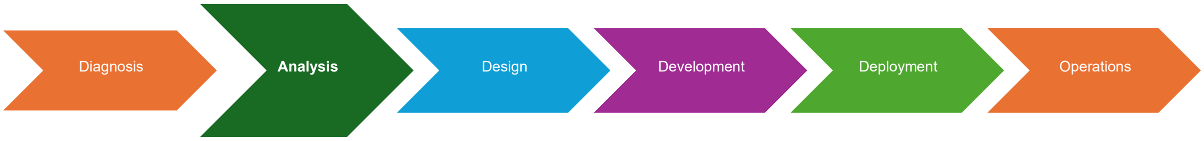


FIGURE 10: SURESTEP ANALYSIS PHASE

Key Activities/ Deliverables

- Key User Training
- Detailed Business Process Analysis
- Gap/Fit Analysis
- Gap Resolutions
- Description of Interfaces
- Data Migration Plan
- Project Plan
- Functional Requirements Document

1.1.3. Design Phase

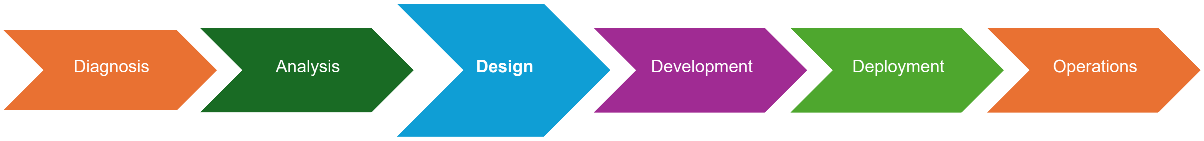


FIGURE 11: SURESTEP DESIGN PHASE

Key Activities

The goal of the Design phase is to define how the business requirements will be implemented. The phase includes design of both the overall Microsoft Dynamics™ solution configuration and the design of specific customizations and integrations needed to satisfy business requirements identified during the Analysis phase. The customizations can range from simple user interface or report modifications to complex functionality additions or modifications. The phase also includes mapping and designing processes for data migration, as well as an analysis of current hardware and infrastructure environment to ensure optimum system performance, (on premise or in the cloud).

Deliverables

- Design Specifications
- High level design specifications
- Technical design specifications
- Data Migration Design and Mapping
- Test Cases, Scenarios and Plan
- Technical architecture and infrastructure support

1.1.4. Development Phase

Diagnosis
Analysis
Design
Development
Deployment
Operations

FIGURE 12: SURESTEP DEVELOPMENT PHASE

Key Activities

The goal of the Development phase is to develop the customizations, integrations and data migration processes that are defined and approved in the design specifications. The major deliverables are the completed feature customizations, reports, integrations and data migration processes. Each component is tested and verified to be functioning according to specifications.

Deliverables

- Feature customizations coded and tested, including integrations
- Data migration processes and Conversion coded and tested as per the requirements in the RFP

Data Conversion Architecture

The Data Import/Export Framework creates a staging table for each entity in the Microsoft D365 database where the target table resides. Data that is being migrated is first moved to the staging table. There, you can verify the data, and perform any cleanup or conversion that is required. You can then move the data to the target table or export it.

Systems Integration - The Import/Export Process

The following diagram shows the steps that are required to import or export data in Microsoft Dynamics 365 Business Central.

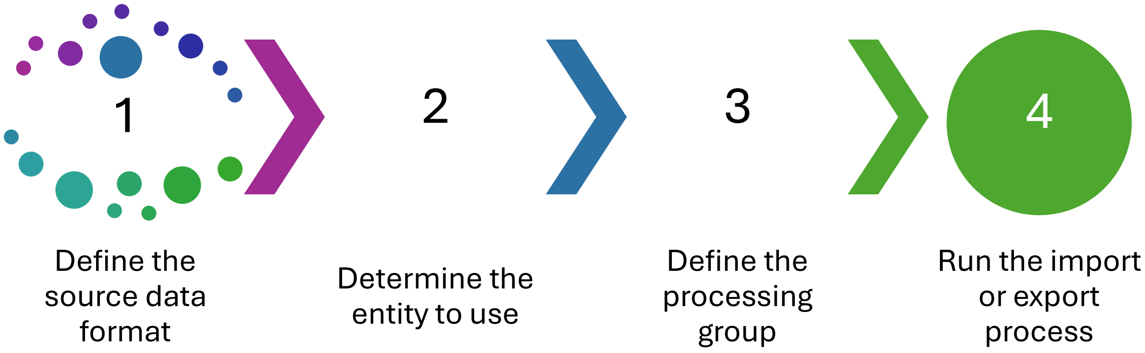

1
2
3
4
Define the source data format
Determine the entity to use
Define the processing group
Run the import or export process

FIGURE 13: DATA EXPORT/IMPORT PROCESS

Determine the source of the data to export or import and create a source data format for the data.

1. Determine which entity to associate with the data. This entity is either the source of the export data or the target for the import data. You can use an existing entity or create a custom entity.
2. Determine which entities should be imported or exported together and put all these entities in a processing group. A processing group is a set of entities that must be processed in a sequence, or that can logically be grouped together. The entities in a processing group are exported together, or they are imported together from source to staging and then from staging to target. In a processing group, you also associate each entity with a source data format.
3. Use the processing group options to either import or export data. For import, you first import the data to a staging table, where you can clean or transform the data as you require. You should validate that the data appears accurate, and that the reference data is mapped correctly. You then migrate the data from the staging table to the target table. You should validate that the entity appears accurate in the target table.

   For export, you also move the data from the source to a staging table, where you can clean or transform the data as you require. You then export the data to Microsoft Dynamics 365 Business Central.

1.1.5. Deployment Phase

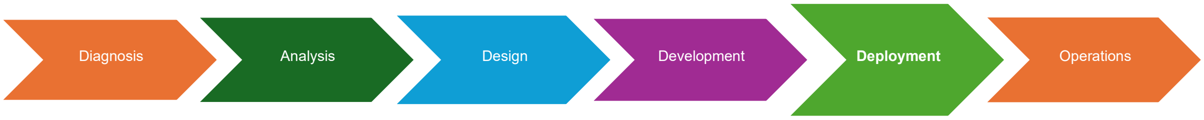


FIGURE 14: SURESTEP DEPLOYMENT PHASE

Key Activities

The Deployment phase is where all the efforts of the project team come together for a successful transition to the new Microsoft Dynamics 365 Business Central solution. There are several important activities that must be completed to reach the end goal. The phase includes all the activities related to final system and load testing, training End Users, and the actual cut-over to the new production environment.

Deliverables

- Go Live Plan and Checklist
- System (User Acceptance) Test Plan, as per Testing and Acceptance requirements in RFP
- End User Training Plan and Training Documentation, (user, training manuals etc.)
- Functioning Live (Production) System

1.2. Operation Phase

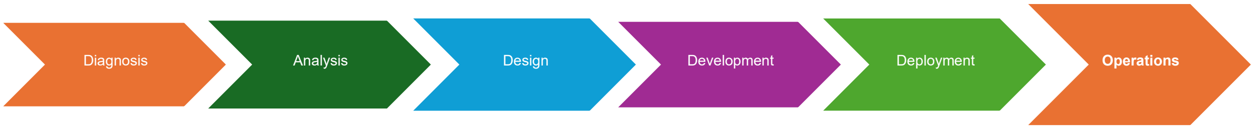


FIGURE 15: SURESTEP OPERATION PHASE

Key Activities

The goal of the Operation phase is to transition the customer from the implementation project into on-going support following a successful go live. The deliverables for this phase include final project and software related documentation, project review documentation, and on-going support contracts. At the end of this phase, the project will be transitioned to the customer and on-going support.

Deliverables

- System Acceptance Sign-off
- Facilitating User adoption through proper training plan
- Project Review Documentation

- Post Live Support Agreement, (System Stabilization Support)
- SLA Schedule of Services Doc. (Provided with RFP response, please note scope needs to be confirmed with Customer).

1.3. Communication Plan

A formal process will be employed to facilitate communication during the delivery of the scoped services. There will be two key vehicles for providing this communication: a weekly status report and a weekly status meeting.

- The Rifumo Project Manager, working in conjunction with City of Johannesburg Project Manager, will compile status reports for distribution to both City of Johannesburg and Rifumo Management.
- Meetings will be held to review overall status, the project schedule and open issues noted in the status report.
- Additionally, a Steering Committee will be constituted, and a meeting will be conducted on a monthly basis, after which a status report will be prepared and circulated.

1.4. Issue Management Procedure

A formal process will be employed to track and resolve Service issues. The following general procedure will be used to identify and manage project issues:

1. Identify.
2. Document.
3. Assign Responsibility.
4. Monitor and Report Progress.
5. Communicate Issue Resolution.

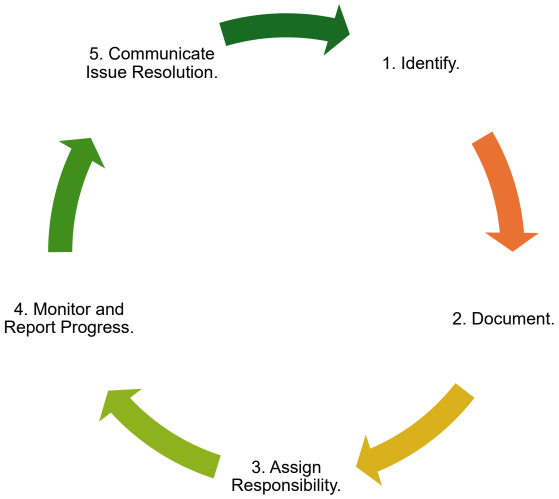


FIGURE 16: ISSUE MANAGEMENT PROCESS

1.5. Change Control

During the delivery of the scoped services, items might arise that are outside the scope of work contemplated by either party. To address such items so that they are given proper consideration and to increase the probability that project scope is carefully managed, and costs are contained, Rifumo will employ a formal change control process.

The proposed change control process that Rifumo will employ is as follows:

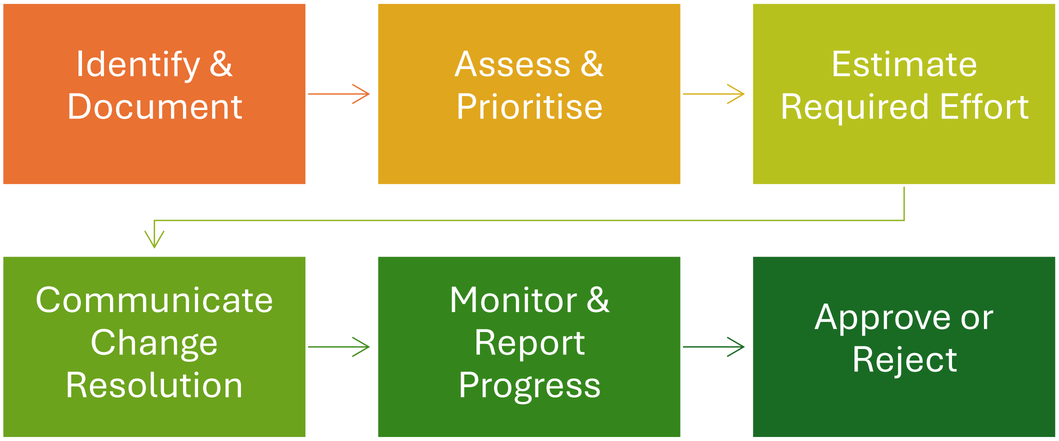


FIGURE 17: CHANGE CONTROL PROCESS

4 Project Roles and Responsibilities

4.1 Rifumo Roles and Responsibilities

Role	Responsibilities
Functional Consultant	▪ Responsible for configuring and customizing the Microsoft Dynamics application to achieve the customer's business requirements.
Solution Developer	▪ Responsible for the design and development of modifications to the Microsoft Dynamics standard application and integrations to the customer's existing systems.
Technical Team Lead	▪ Serve as primary technical point of contact for the team responsible for technical architecture and code deliverables
Power BI Report Developers	▪ Develops data models for priority reports/analytics. ▪ Develops all business logic, measures, and calculations. ▪ Develops priority report front ends including static reports and dashboards.
Project Manager	▪ Manage and coordinate Rifumo project delivery. ▪ Take responsibility for problem and risk management, change management, project priorities, status communications, and status meetings. ▪ Coordinate Rifumo and subcontractor resources but not the **Error! Unknown document property name.** resources. ▪ Serve as a single point of contact for escalations, billing problems, personnel matters, and contract extensions.
Azure Infrastructure Engineer	▪ Assists with the deployment and configuration of Azure services. ▪ Ensures the security of the data platform. ▪ Validates the security of integrations and system changes. ▪ Manages system maintenance, backups, and recovery plans.
Solution Architect	▪ Define the architecture and technical requirements for the solution

	▪ Design and develop the framework and tools best suited for the Customer environment.

Table 6: Rifumo Project team Roles and Responsibilities

4.2 8.2. Customers Roles and Responsibilities

Role	Responsibilities
Project Sponsor	▪ Provides strategic guidance and makes key decisions. ▪ Resolves any high-level issues or risks. ▪ Approves critical project elements and changes.
Project Manager	▪ Provide the estimated project commitment: part-time. ▪ Serve as primary point of contact for the project team. ▪ Manage and coordinate the overall project. ▪ Take responsibility for resource allocation, risk management, project priorities, and communication with executive management. ▪ Manage day-to-day project activities. ▪ Coordinate team activities to complete deliverables according to the project schedule.
Product Lead	▪ Identifies and include appropriate stakeholder representatives as needed. ▪ Participate in the workshops. ▪ Determine a remediation strategy that is based on the findings document and remediation and rationalization guidance. ▪ Provide additional information and answer questions regarding the current implementation, if relevant.

Table 7: Customer Project Team Roles and Responsibilities

5 Costing & Pricing

5.1 Role-based Cost Breakdown (ex VAT)

Role	Hours	Rate (ZAR/hr)	Cost (ZAR)	Cost (USD)
Project Manager	408.67	R1,900.00	R776,473.00	$44,026.02
Project Coordinator	93.63	R800.00	R74,904.00	$4,247.06
Solution Architect	630.55	R2,200.00	R1,387,210.00	$78,654.81
UI/UX Dev	2,257.03	R1,300.00	R2,934,139.00	$166,365.68
Tech Lead	3,749.26	R1,900.00	R7,123,594.00	$403,907.78
Senior Dev	2,703.98	R1,500.00	R4,055,970.00	$229,973.50
DevOps Engineer	791.12	R1,900.00	R1,503,128.00	$85,227.36
Senior Test Analyst	3,029.95	R1,200.00	R3,635,940.00	$206,157.80
Mid Dev	2,640.54	R1,200.00	R3,168,648.00	$179,662.34
Junior Dev 1	2,275.26	R800.00	R1,820,208.00	$103,205.79
Junior Dev 2	2,275.26	R800.00	R1,820,208.00	$103,205.79
Senior BA	2,598.54	R1,500.00	R3,897,810.00	$221,005.83
Junior BA	2,298.60	R900.00	R2,068,740.00	$117,297.56
Junior Test Analyst	150.43	R800.00	R120,344.00	$6,823.50
Data Architect	632.40	R2,100.00	R1,328,040.00	$75,299.87
Integration Architect	549.82	R2,100.00	R1,154,622.00	$65,467.07
Integration Dev	549.82	R1,500.00	R824,730.00	$46,762.19

5.2 Costing Summary

Metric	USD (indicative)
Labour (ex VAT)	$2,137,289.94
Delivery contingency 6% (ex VAT)	$128,237.40
Total (ex VAT)	$2,265,527.34
VAT 15%	$339,829.10
Total (incl VAT)	$2,605,356.44