UMANG API Manager

Solution Design v 1.0

**Powered by Spice Digital Limited**

Document Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Change Type | Clause | Description | Date |
| 1.0 | First Draft |  | First Draft for Review | 8 Mar 2019 |

Contents

[1 Introduction 6](#_Toc12547545)

[1.1 Acronyms and Abbreviations 6](#_Toc12547546)

[1.2 References 6](#_Toc12547547)

[2 Technical Details 7](#_Toc12547548)

[2.1 Network Architecture 7](#_Toc12547549)

[2.1.1 Service Flow 8](#_Toc12547550)

[2.2 System Architecture 8](#_Toc12547551)

[2.2.1 System Components 8](#_Toc12547552)

[3 Design Considerations 9](#_Toc12547553)

[3.1.1 Scalability 9](#_Toc12547554)

[3.1.2 Security 9](#_Toc12547555)

[3.1.3 High Availability 9](#_Toc12547556)

[4 Software and Hardware 10](#_Toc12547557)

[4.1 Software 10](#_Toc12547558)

[4.2 Hardware 10](#_Toc12547559)

[5 Integration Details 10](#_Toc12547560)

[6 Exception Handling 11](#_Toc12547561)

[7 Assumptions 11](#_Toc12547562)

Figures

[Figure 1: Network Architecture 7](#_Toc12547563)

[Figure 2: System Architecture 8](#_Toc12547564)

Tables

[Table 1: Acronyms and Abbreviations 6](#_Toc12547566)

[Table 2: References 6](#_Toc12547567)

# Introduction

The UMANG API Manager is a tool that manages all APIs associated with UMANG. The following operations can be performed using UMANG API Manager Tool:

1. **Global Level API Blocking/Unblocking**: Admin can perform global level blocking/unblocking of an API.
2. **Global Level API Expiry Setting**: Admin can set the validity of an API.
3. **Global Level Pricing of APIs**: Admin can set the pricing of an API, which then helps in generating revenue reports.
4. **API Version Management**: Admin can manage versions of an API
5. **Vendor Management**: Admin can manage vendors (i.e. Add, Edit, Delete, Block, Unblock, Set Vendor-wise API Pricing)
6. **Vendor Level API Usage**: Admin/vendor can view the usage of API
7. **Vendor Invoicing**: Admin can generate Invoice and Vendor can view their Invoice.
8. **Notification Engine**: Vendor can raise a request through Notification Engine and Admin can take action (Approve/Reject) on the raised requests.

## Acronyms and Abbreviations

The following table lists the acronyms and abbreviations used in this document.

Table 1: Acronyms and Abbreviations

| Acronyms | Full Form |
| --- | --- |
| API | Application Program Interface |
| HTTP(S) | Hyper Text Transfer Protocol (Secure) |
| NeGD | National e-Governance Division |
| UMANG | Unified Mobile Application for New-age Governance |

## References

The following table lists the reference documents related to the UMANG API Manager portal.

Table 2: References

| S. No. | Document Name | Description |
| --- | --- | --- |
|  | UMANG RFP | This document contains detailed information on requirements of the UMANG platform. |
|  | Bid Clarification | This document contains detailed information on responses of NeGD on the queries related to the UMANG platform. |
|  | Corrigendum | This document contains high-level information on the UMANG platform. |

# Technical Details

This chapter explains the architecture and functionality of the system. The chapter is   
organized into the following sections:

* [Network Architecture](#_Overview)
* [System Architecture](#_System_Architecture_1)

## Network Architecture

The following figure depicts the Network Architecture of API Manager in detail.

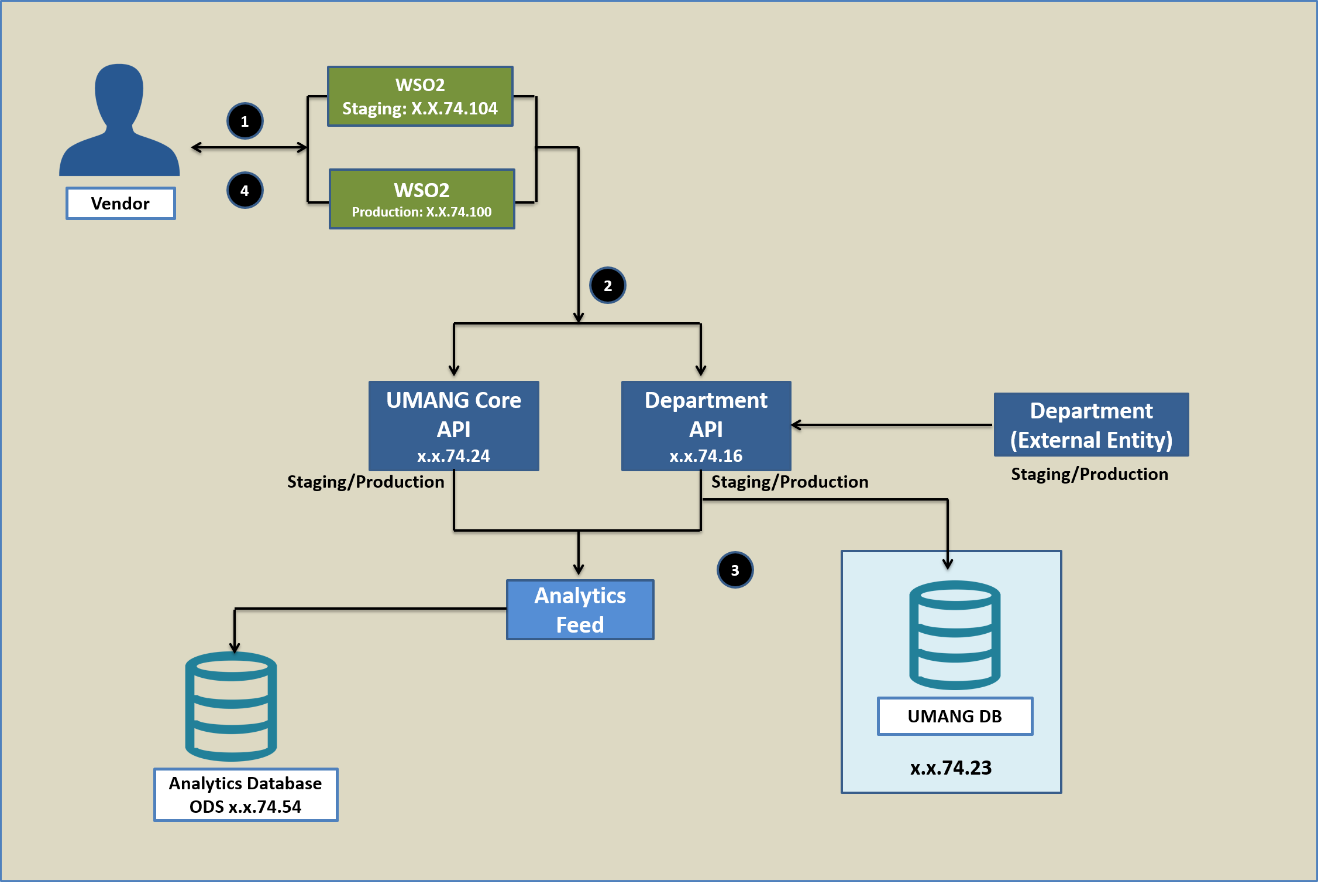


Figure 1: Network Architecture

### Service Flow

A typical flow for a service available to the user will be as follows:

1. Vendor sends a request for API Management (Registration, Issue Raising for API and Request for Production API) through API Manager Interface using Client ID through vendor interface.
2. Depending upon the type of request (Staging or Production), UMANG Core Platform hits the WSO2 & database.

* **Note**: Admin can Approve or Reject the request.

1. Data is fetched from the WSO2 and sends back to Vendor in the form of response.
2. This response is shared with the Vendor through UMANG API Manager Interface in notification engine.

## System Architecture

The following figure depicts the System Architecture of the API Manager.

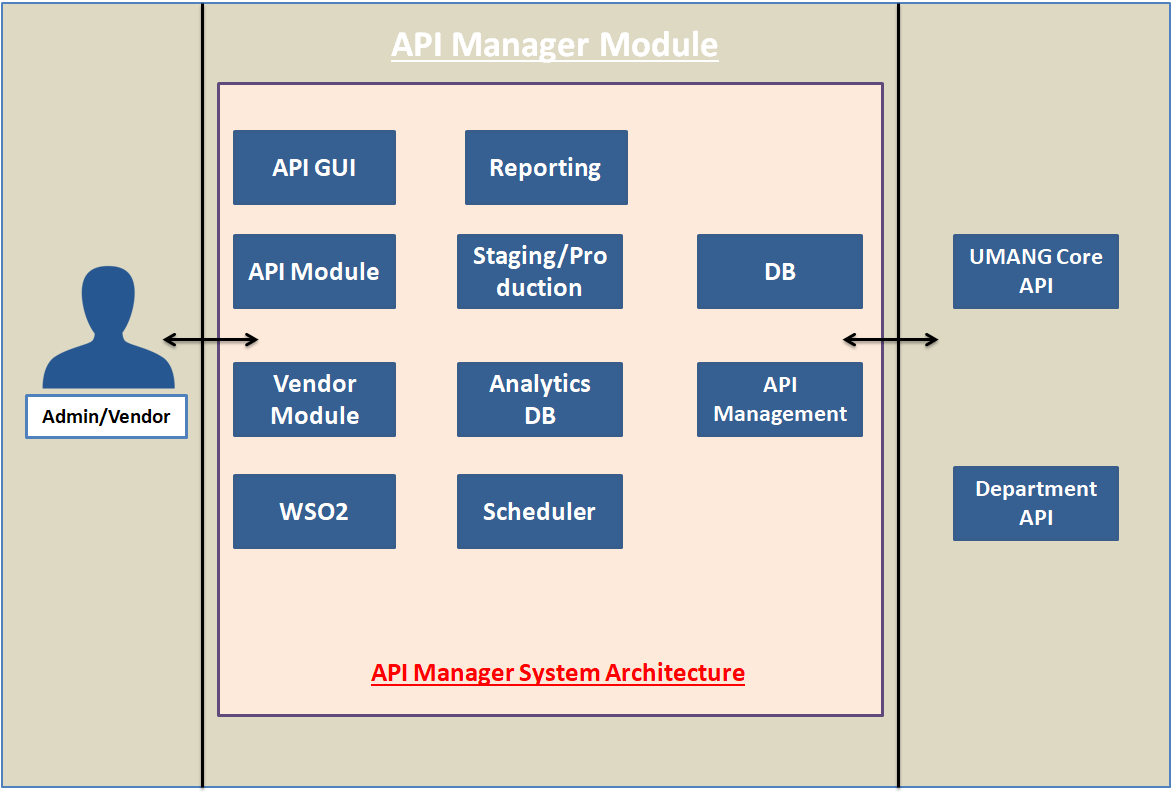


Figure 2: System Architecture

### System Components

The following table lists the details of the Self Care system components.

|  |  |
| --- | --- |
| Component Name | Description |
| **API GUI** | It is a developerportal, where vendor can register and request Admin (NeGD) for accessing the UMANG API. |
| **Reporting** | There are various kind of reporting in API Manager i.e. Vendor Invoice, API Usage Report etc. This allows admin to check the consumption of API vendor-wise. |
| **API Module** | It is a primary component of API Manager which interacts with WSO2 and updates the API access information etc. |
| **Staging** | This is responsible for getting API access on staging environment for vendor (Staging WSO2). |
| **Production** | This is responsible for getting API access on production environment for vendor (Production WSO2). |
| **Vendor Module** | This component allows Admin to check Vendor Reports, Delete and Block the Vendor. Admin can also set vendor API pricing. |
| **WSO2** | WSO2 is API Manager that can be accessed through WSO2 API. API Manager is the base of WSO2 API Manager. |
| **Scheduler** | Scheduler runs on a set time and updates all the documents on WSO2. These documents are then available on API Manager through scheduler. WSO2 provides input and update API Manager. |
| **UMANG Core API** | UMANG Core API is responsible for UMANG Registration, MPIN, and Change Password etc. which provides UMANG Core functionalities. |
| **Department API** | This is a third party API e.g. EPFO, Kisan Suvidha etc. and vendor can access these APIs through API Manager as per department assigned by the Admin. |

# Design Considerations

The subsequent sections provide information on the design considerations of the UMANG API Manager.

### Scalability

* All communications are stateless and there is no need to manage the session.
* APIs hosting will be performed on highly scalable platform through load balancer system. More systems can be added at run time to cater more requests.

### Security

* All APIs integration should be done on HTTPS only. JSON communications simplifies authentication efforts.
* User authentication will be performed using token system.

### High Availability

A clustered environment ensures high availability of the system as API hosting is done through load balancer system in Active-Passive mode.

# Software and Hardware

This section provides information on the software and hardware used in the UMANG API Manager implementation.

## Software

The software used will be:

|  |  |
| --- | --- |
| Software | Usage |
| Postgres DB | For maintaining entire functionality |
| Appium | For API |
| Linux | Deployment OS |
| Nginx | For Clustered Environment |
| Angular 5 | For Frontend |
| Spring Hibernate | For Backend |

## Hardware

Hardware required will be:

|  |  |  |
| --- | --- | --- |
| Hardware | Configuration | Quantity |
| Servers | 16 GB RAM | 2 |
| Hard Disk Drive | 200GB | 1 |
| Processor | Octa Core | 1 |

# Integration Details

UMANG API Manager Module will be integrated with WSO2. Entire interaction will happen through WSO2 provided APIs and WSO2 will further interact with UMANG core as well as Department system.

# Exception Handling

# Case 1: Document is not Available

Handling: Check the scheduler and restart. Also, check the exception in WSO2 console.

# Case 2: API is not reflecting

Handling: Check WOS2 console error.

# Case 3: Department Image is not visible on portal

Handling: Upload corresponding department image in PNG format.

# Assumptions

* The UMANG API Manager platform will be hosted on the infrastructure provided on the NIC cloud.
* Integration with the UMANG API Manager will be done using HTTP(S) protocol.