Project MwM – Network Rail

Architecture Review Report

Rajashekar Srinivas,

Enterprise Architect

CTO Office, P&R

Cognizant

NetworkRail

Document Version

Version	Author	Date	Reviewed by
1.0	Rajashekar Srinivas	17 May 2023	

Glossary

Table of Contents

Document Objectives

Cognizant has built a first version of the software product called “Mobile Work Management (MwM)” which is aimed to replace an existing solution (FieldReach) with additional enhanced features which strengthens the product and its objectives. As the product is embarking in the journey into UAT phase, an architecture audit is conducted to validate the software product is architecturally strong and resilient, so that the organisation confidently rolls out to wider audience across. This document aims to capture various layers architectures and makes necessary recommendations which can strengthen the product architecturally, so that it can be successfully implemented.

Business Objectives & Personas

Brief on Application / Product

NetworkRail owns assets in entire UK rail network. Ellipse is the Asset management System hosted on-premise… This system holds around 5 million assets belonging to NetworkRail. Each asset will have to undergo maintenance activities based on the defined life either in Hours, Calendar days, months etc. Based on the schedule Work orders are generated (planned work schedules), so that the assets are maintained in serviceable condition. The RAMs (Route Asset Managers), Engineers / Planners generate the Workorders in the Ellipse System. The system also generates work orders for adhoc activities on assets.

Around 100,000 work orders are generated in a given month (approximately) i.e. system is expected to generate around 140 – 150 work orders in an hour. These work orders are to be distributed based on the Asset classification, its discipline which will have Engineers & Section managers geographically spread across UK. The Work orders generated by asset management system will be channelled into a web portal which can be access by Section managers. Each section manager will have multiple Mobile work groups (based on discipline), who executes Work orders on the field and update the status of the work order on Mobile device. Once the work order is updated by the Mobile work Group, Section manager will have to approve / reject, so that the relevant closure information flows back into asset management system via MuleSoft.

Mobile devices will use OS MAP for specific geo location of the asset. We have dump (NR data) of master maps, which is published through Geo Server. Asset & Work Order MAP services are published through Geo Server. Geo Server communicates with SQL Server and returns the Map Image for the work orders / Assets / Master Map. Master map packaging is done through manually using QGIS tool which is downloaded by Mobile device

Mobile work Group can also raise a WAIF (Work Arising Identification Form) on the mobile device along with photographs as evidence. Based on the approval of WAIFs by Section managers, A fresh work order will be generated from Ellipse.

Based on discipline, various custom forms (to gather specific information) can be built and published from portal into Mobile device. These forms are to be filled by Mobile Work Group and the relevant data flows into Asset management system.

Product Objectives (High level):

- To replace the current FieldReach Mobile solution, i.e., My Work app, Field Data Manager (FDM) desktop application and Script Builder with a modern and intuitive solution enhancing the user experience and improving data capture in the field.
- Decommission the existing FieldReach mobile solution, applications, and infrastructure

Business Processes:

- Work orders – Create, Update, Cancel
- Assets – Create (Ellipse), Modify
- Inspections – Create
- Time – Non Time On Tools
- Reports – View operational reports

Business Users / Actors:

- Mobile Work Teams (15K)
- Section Managers (1K)
- Engineers (250)
- RAMs, Planners, System Support Managers (1K)

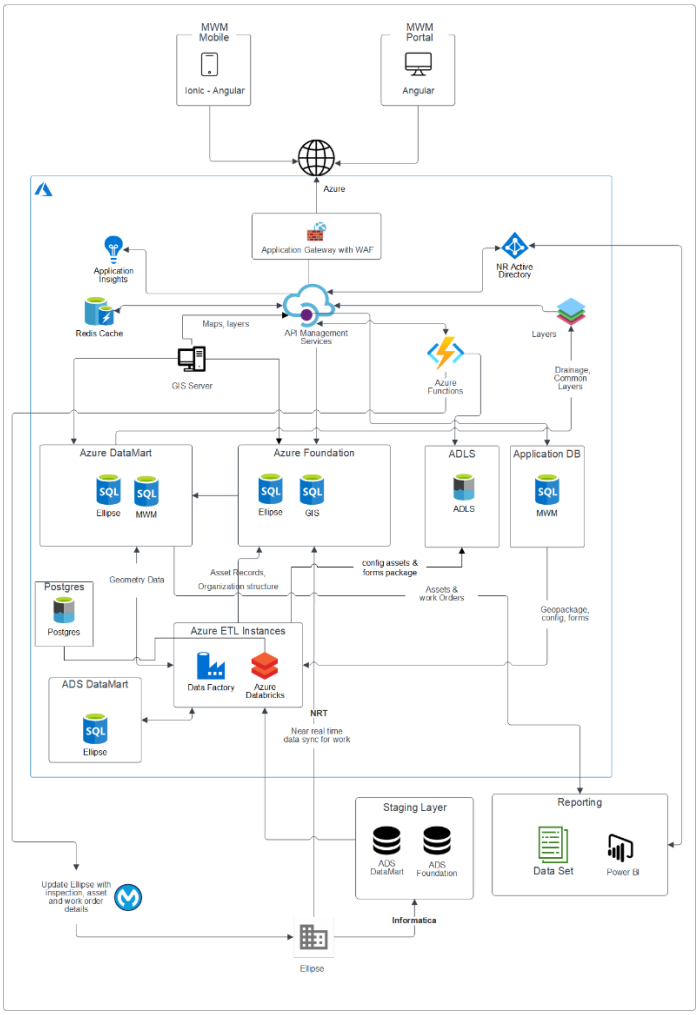
Scope:

- Replace the current AMT-SYBEX FieldReach mobile application (Work Manager) with a bespoke mobile solution on the NR Microsoft Azure platform
- Backoffice desktop application Field Data Manager (FDM) will be replaced by a bespoke web-based Work Review Release and Dispatch system (MWM Portal) on the NR Microsoft Azure platform
- Replace any existing and 'in use' scripts with new Azure forms for Field data collection
- Decommission the existing FieldReach applications & infrastructure

No. of User stories : Around 500

High level Architecture (End-to-End)

Below architecture diagram is submitted as part of DSD



List of architectural components

Component	Description
Ellipse	Enterprise Asset Management System
Informatica	Enterprise ETL tool used to extract data from Ellipse
ADF	Scheduling Databricks Job
Databricks	Logic to pull the data Azure Data foundation to Data mart pipeline, Geometry calculation, Data package Generation etc.
ADLS	Asset Package, Form Package, Config Package, OS Master Maps, Photos
Azure DW	No
RBAC	Authorisation & Role based access. Directory services for user authentication. Authorization of user against the organisation security standards.
Azure AD	Authentication (Federated AD from Network Rail AD)
Azure Keyvault	Any secrets like App registration secrets, SQL connection string securely stored in Key Vault and Retrieved at run time securely.
Azure Insights	Used for Monitoring and Debugging
Redis Cache	Used for URL caching
Azure API Manager	API Management / Routing
Azure Functions	APIs for (around 80), Micro services for integration between My Work portal, mobile to My Work DB, MuleSoft.
Azure SQL Server	Transaction Data, Configuration Data, Forms Data etc.
Azure Postgres	NetworkRail Data model, Tracks, Track Central line, Way points – This data is used for Geometry calculation
Azure App service	Used for Hosting Web Portal
Azure WAF	WAF – Firewall to regulate the traffic. Security layer for the application to protect from malicious attacks.
Geo Server	An opensource software hosted on WebApp called 'GeoServer' publish various types of GIS services. It is connected to DataMart & Foundation to publish the assets & network model data…. GIS Services published by GeoServer. Layer is a data service provided by GeoServer, which has data about Assets, Drainage assets… It generates pictorial images. This service is consumed by portal. Geo server exposes the data to the application in geospatial format (WMS/WFS etc.)
ADS	Asset Data Store – A replica copy of all assets from Ellipse
Power BI	Operational Reporting platform used along with Web Portal
MuleSoft	Integration layer between MwM with Ellipse (to complete Work Order)
NR Marketplace	Place where the latest version of the Mobile application is published for iOS only

Application Architecture

Application is divided into 2 parts

Web Management Portal

Actors – Section Managers / Admin / Form User / Planners / Route Managers

Main Features

- Distribute / Channelise Work orders
- Approve Work completion submitted
- Approve Inspections
- Approve Asset creations and update
- Update back into Asset Management System

Web based - Yes

Development platform - Angular, CSS, HTML, JS

Hosted on – Azure WebApp (Scalable)

Session management – Using built-in Angular state management

Stateless – No

Logging enabled – Yes, partially

Exception Management – Handled <<Scenarios to be confirmed>>

Mobile application

Actors – Mobile Work Group

Main Features

- View Assigned work
- Complete assigned work
- Record Inspection results
- Request additional workorders
- Verify asset details
- Request creation of assets

OS – IoS Only <Supports Current -minus 1 Version>

Development platform - Ionic Angular

Additional plug-in required – Yes, Plug-ins (20-30 Plug-ins are used)

Application Hosted on – MS App Centre (Non-Prod), NR App store (UAT onwards)

Packages hosted on – Azure BLOB / ADLS

Stateless – Yes

Local storage required – Yes

Mobile Package store – SQLLite

File download on Mobile – while downloading data, if any interruptions like opening other application, receiving call, running in background etc. the download gets terminated,,,it need to download again…

FileTransfer Plugin - ??

High level process of packaging

- Configuration of Delivery Unit / Discipline in Admin portal – each DU & Discipline, Equipment Group Identifier, based on that the asset is assigned to discipline,
  - Admin sets the Frequency of package generation
  - These configurations are all stored in MwM DB (Application DB)
- Packing picks the data from
  - Datamart – Assets
  - Forms – MwM Database
  - Config - MwM Database
- Workorder data is fetched through API services directly by Devices (Mobile / WebPortal)
- Workorders when generated, they will be available on both places i.e. Mobile as well as Portal. This visibility will be based on RBAC. Based on workgroup assignments.

- Databricks Notebooks reads the configuration and generates the package
- Packages are stored in ADLS
- Mobile device request for Package through API call (Azure function)
- Data package downloaded into Mobile device

Package contains

- Base Map (one time generation & download) – This is one time manual generation – This is DU specific
- Asset package – Daily (incremental data), If any changes in data specific to DU / Discipline changes, then only we generate a fresh package
- Forms Package – This is Discipline specific
- Config Package – Generic package for all disciplines
- Packages are downloaded from Store based on changes. Mobile application probes for changes by calling an API, based on DU & Discipline.

What is downloaded – first time –

1. Base Maps (300 – 700 MB) – Master map dump from (One time download)
2. Assets (300 – 500 MB) – Based on changes
3. Config (70 – 150 MB)
4. Forms (70 – 500 MB)

Periodical download (based on a change in package) – Change in Assets, Configs, Form changes

Work order size – 5 – 10 MB

Daily downloads – 15 Mins (work orders, New created / edited forms)

Mode of Project Delivery – Agile

Data Architecture

MwM Application has 2 parts Web Poral & Mobile application which uses multi structured data i.e., Asset Data, Work Order Data, GIS data, Form data, Map Objects (Layers), Pictures etc. Following Azure services are being used for various purposes mentioned as below

Component			Description
ADLS		ADLS	Asset Package, Form Package, Config Package, OS Master Maps, Photos
Azure Datamart	Ellipse	SQL DB	This is processed data from Ellipse foundation with specific KPIs defined. All APIs use this Database to provide the Asset related data
Azure Datamart	MwM	SQL DB	This is a replicated data from MwM application Database for reporting purposes (replicating only form data)
Azure Foundation	Ellipse	SQL DB	Data from ADS, Work Orders (Raw data landing from Ellipse)
Azure Foundation	GIS	SQL DB	Separate Schema, GIS Data, Data coming from GeoRinim (On premise). One time Data Load, Geometry data which is generated through ETL
Application DB	MwM	SQL DB	SQL DB – Application data
Geometry		Postgre	NetworkRail Data model, Tracks, Track Central line, Way points – This data is used for Geometry calculation Layers are the services offered by Geo Server Different asset layers related to maps, -> from GIS data published using Geo server
Redis Cache		Redis	URL caching only
ADS			Asset Data Store (Replicated data from Ellipse)

Integrations

MwM application has multiple integrations which mainly deals with Data. List of integrations are as given below

Component	Description
ADF	Azure Data Factory is being used for scheduling Databrick jobs
Databricks	Databricks is used for Data transformation between foundation to Datamart, packaging, or any data related operations. (Scheduled jobs)
Azure functions	Azure functions are implemented as APIs for various activities within the application.
ADS	Asset Data Store is a replicated data from Ellipse on Azure

Hosting architecture

Hosting platform - Azure

MwM is designed & developed using Azure cloud native services and components. Majority of the components are either serverless or Azure managed services.

No. of environments – Dev, QA, SIT, UAT, PPD, Prod, Pen Test

Delivery methodology – CI / CD

Testing strategy (Specific to Load testing)

Unit Testing, Functional Testing, Integration testing, UAT, Load testing, Pen test / Security Testing (Third Party MDSEC)

Load testing scenarios - <<Team to detailed out>>

Some of the Load scenarios could be

(a) Login – Authentication & authorisation (Web portal)
(b) Packaging -> How many packages are made in a day. Which API is involved in packaging
(c) Mobile downloads – Initial download, Daily download (is it delta or full)
(d) Reports – How many reports are getting generated. Does it have impact on DB response time?
(e) No. of Work order data getting pulled into ADLS on hourly basis
(f) API performance – Identify specific APIs which is called for repetitive loads… This would require scaling horizontally.
(g) Identify functions which requires more processing time. (Long running functions which requires more processing power & time)
(h) Can MuleSoft scale for the load for completion of Work Orders. How many work orders are getting closed in a day.

Recommendations / Improvements

Team conducted a workshop to go through product architecture & functional demo session on 3rd & 4th May 2023. Discussions were also held post workshop who are part of the product development to gather information which is documented. Please find below recommendations / Improvements observed in order to make the product more resilient and fit for purpose.

Leverage Data Caching

Data Caching strategy is a very common methodology used to achieve better performance as well as lowering the load on Databases. In current state, The product is using Redis Cache service only to cache URLs. It is strongly recommended to Cache

(a) All Lookup data (plus tables)
(b) Master data which do not change very often
(c) Build indexes where search use cases in the product
(d) Any other dataset which can be cached which are access / used frequently

Some of the major advantages of caching would be

(a) Reduced load on Database
(b) Faster response time
(c) Low latency
(d) Improved scalability
(e) Better resource utilisation
(f) Improved user experience
(g) Better handling of surge in traffic

Note – Caching strategy to be carefully designed & executed.

Organising Azure functions

MwM Product (Web portal & Mobile) is extensively using APIs and Azure functions are being used as backbone for all APIs. Azure API Manager is being used as a front end tool to manage all the requests for APIs.

The challenge noticed is all the APIs are part of single Azure Function App, which is incorrect and this will lead to complexities.

It is highly recommended to group the APIs/ functions and host them in separate functions Apps. This achieve the following

- Isolation & Scalability
- Better resource Management
- Flexible deployments
- Security & Access control
- Trouble shooting & debug
- Better Code Management

Some useful links

Best practices for reliable Azure Functions

Improve the performance and reliability of Azure Functions

Using Views to access the data

Currently it is observed that data tables are directly exposed to APIs via stored procedures. Stored procedures are very helpful, but enforces tight coupling on to platform. Stored procedures builds the required dataset when API calls them (in real time) and this will increase the query execution time. Instead it is recommended to build consolidated views (Some can be Materialised) and APIs can query the views directly from DAL layer of the service… Some of the advantages would be

- Elimination of dependency on writing complex stored procedures. (which leads to tight coupling)
- Lowering the Query execution time leading to performance optimisation (mainly when dataset requires multiple joins)
- Abstracting the data sets leading to better security
- Simplified & bespoke data sets with complete control
- Reusability - Same views are reusable for multiple requirements

Conduct Load testing

MwM has 2 parts i.e. web management portal & Mobile application which is intended to be used by various actors in the system. The system is also expected to have various types of data (Multi structured data) flowing into Data systems on a near real time basis. End user experience and a strong reliance while using the product will be very crucial for the successful implementation. Keeping these factors in mind, it is strongly recommended the following

(a) Identify and detail out load generating scenarios within the system (both web & mobile)
(b) Define the expected concurrency / load which is expected in real time scenario.
(c) Document the load testing scenario for a specific load use case.
(d) Performance engineering is another prime activity to fix the flaws in the system.
(e) Performance tuning would be required at one or more layers i.e.
    a. Web layer
    b. API layer
    c. Data Layer
    d. Integration layer
    e. Infra / hosting layer
    f. Dependencies on External systems / devices
    g. Dependency on external networks

Enabling Logging, Monitoring & Alerting

Logging plays very important role in understanding how product is behaving at every stage / layer. It is very important to enable logging, collect the logs, enable a log dashboard to monitor the logs. The Logging platform can be configured to raise alerts based on the error severalty. Logs would help the tech support team to identify & resolve the bugs in a very optimised way.

It is strongly recommended to enable logging at each stage / component in deployment architecture i.e.

(a) Azure Service logs (ADF, Databricks, Functions, WebApp, WAF, API Manager etc.)
(b) Application logs
(c) API logs
(d) DB logs
(e) DB Transaction logs
(f) Access logs
(g) Geoserver logs

All the logs can be consolidated into an enterprise Log Management System (like Splunk / elastic) or use Azure Log analytics services for the purpose.

Enablement of Monitoring & alerting

MwM is built using multiple technology components on Azure as well as it has integration touch points. It is very important to enable Monitoring & reporting on the availability of services and to ensure service health. We can also evaluate the requirement of APM tools to monitor the application performance.

Azure Monitoring & Alerting services can be enabled to reap the benefit. Azure Insights can be enabled to have a detailed insights for Web application, Functions, Databricks & API Manager.

Build a Solid Exception handling framework (Identify exception scenarios)

MwM is built using multiple technology components on Azure as well as it has integration touch points. Any failures at any point with product architecture may lead to delay in further actions as well data discrepancies in the value chain. To make the product more resilient, it is recommended to build a strong Exception handling at every stage of the data flow between the components. It is recommended to document all possible exceptions and build how the system is designed to overcome the exception. The product support team should equipped with this document to ensure proper resolution.

Product Rollout Strategy

It is highly recommended to do a staggered / phased roll out of the product, rather than big bang approach. This can be detailed out post UAT and performance engineering of the product.

Product support strategy

Post production product support is key for successful product adoption. Initial 6 months of product adoption should be well orchestrated with a dedicated support team. Since this product used by field force, it is very important to hand hold them, support them for successful adoption. A dedicated support structure is strongly recommended for a minimum of 1 – 2 year with L2 & L3 team to resolve & fix the issues on top priority.

End note

Team has successfully demonstrated end-to-end functional flow as per the scope defined i.e.

(a) Creation of Work order in Ellipse system (choosing various options)
(b) Data flowing into foundation data store (
(c) Viewing of newly created Work orders on web Management portal.
(d) Downloading & Installation Mobile application on IoS device
(e) Downloading of packages
(f) Viewing of Work orders on Mobile device (Mobile work group)
(g) Closing of work order on Mobile device
(h) Approval of closed work order from Management Portal by Section Manager
(i) Closed work order data getting updated into Asset Management System via MuleSoft
(j) Creation of WAIF by mobile work Group
(k) Approval of WAIF from Manager
(l) Creation of custom forms using form builder module

Some of the minor functional observations were made during the workshop and the same was noted for corrections..

Even though the application is functionally working successfully, it is very important to strengthen the product for its resilience to sustain the field usage.

Successful adoption of this product is mainly rely on smooth End-user experience i.e. Mobile application & Flow of Work Order data. It is very prime importance to make the Mobile application

(a) Highly Reliable
(b) Intuitive -
(c) Light weight – Lighter data sets, Making packages lighter, so that it can be downloaded faster
(d) Easy to deploy & configure
(e) Less prone to crash / outages / hang / unresponsive
(f) Seamless upload / download of data

A deep dive study is recommended to analyse end-to-end Mobile architecture, its packaging process, various plug-ins used, Dependencies & challenges, what all can be optimised (both for initial download & recurring download) to make the end user adoption successful.