High-Level Design Guidance

Enterprise Service Bus

Date: 9th December 2016 **Version: 1.1**

Registered office Bridge House, 1 Walnut Tree Close, Guildford GU1 4LZ
Highways England Company Limited registered in England and Wales number 09346363

Document Control

Document Title	Design Guidance | Enterprise Service Bus
Author	Dave Horton
Owner	Ivan Wells
Distribution	Highways England, 3rd parties as required
Document Status	First issue

Revision History

Version	Date	Description	Author
1.0	5 December 2016	First issue	Dave Horton
1.1	9 December 2016	Minor updates	Dave Horton

Note: The overview of the CHARM architecture is based on a separate project-based paper from the CHARM project written by from Dominic Paulo and Russell Mead.

Reviewer List

Name	Role
David Bradbrook	Operational Technology Strategy
Mark Pellowe	Operational Technology Architecture Team Leader

Approvals

Name	Title	Date of Issue	Version
Ivan Wells	Senior Enterprise Architect	9/12/2016	1.1

Table of Contents

Introduction

This document provides guidance on the application of an Enterprise Service Bus (ESB) capability within Highways England.

What is an ESB?

Enterprise Service Bus (ESB) is a term that describes a way of organising interfaces between distributed IT systems. It is a software architecture that can be built, but is often implemented using software products. An ESB is intended to reduce the complexity in organisations that have a large number of interfaces by acting as a go-between. It reliably connects systems so that they are interoperable.

An ESB removes direct dependencies and tight coupling between systems. A loose coupling between systems reduces the impact on one system when another system is changed. It helps to standardise the platform for integration and consolidate interfaces to promote their reusability.

An ESB is a piece of software infrastructure, middleware, that needs to be configured to provide the scalability, reliability and technical governance required. It is not a silver bullet solution, it needs to be designed and implemented with consideration to how information flows within an organisation.

Services

ESB interfaces are provided as services. This supports the implementation of a service-oriented architecture (SOA) to enables integration with a diverse set of products and technologies to support the exchange of business information. From a business perspective, the services will:

- Provide logical representations of business activities with predictable outcomes
- Hide the need for consumers of the service to know the low-level implementation details that delivers the business outcome that is needed.
- Be self-contained

Implementation

The ESB middleware will provide a number of technical capabilities. It is a toolbox of options, not all of which have to be implemented. A good ESB implementation will:

- Be stateless, there is no need to keep a record of previous interactions. It just facilitates the exchange of messages between systems through service adapters.
- Configure adapters that will connect the source and target system and be responsible for all data transformation that is required, applying any additional processing rules and communicating using the protocols that each system supports.

- Support multiple standard protocols such as HTTP, JMS, JDBC and in the case of Highways England the bespoke NMCS2 (National Motorways Control System 2) protocol.
- Be capable of combining multiple technical interfaces into a single service that represents a business service and/or transaction to simplify integration and promote reuse.

Our approach to integration

Although Highways England have a technical policy on integration using an ESB we are not taking an ESB-first and only approach to all integration scenarios. We are taking an architecture-first approach.

Data exchanged across business domains must be routed via a domain ESB. Data that is exchanged between different data models and standards should use an ESB.

Simple Web-based IT services that do not require data to be changed or reformatted can be referenced directly. Simple mediation logic within services is an acceptable approach to avoid the unnecessary application of middleware and additional overheads, for example using DNS and an API Gateway to resolve service interfaces or API endpoints.

Our IT strategy is cloud-first. With the increasing use of Software as a Service (SaaS) covering Social, Mobile, Analytics and Cloud (SMAC) solutions our integration approach needs to balance natively building on cloud-based APIs using Representational State Transfer (REST) and JavaScript Object Notation (JSON) and the control afforded by an ESB.

Fundamentally, integration is a business concern and should not be constrained by a technical solution. However, we are not yet at a sufficient level of IT operational maturity to support composing services, we still have a need for a traditional ‘connecting things’ approach. The application of an ESB to meet this need is especially relevant.

Our approach to deployment

A single/global ESB is not preferred for Highways England as there is no single canonical information model to support this. A directly connected set of ESB operating within business domains is preferred for business IT but not necessarily roadside IT where performance requires a lower latency solution. Directly connecting or federating ESBs across different business domains is an enduring architectural theme of Highways England from the vision for an Intelligent Highways Infrastructure and remains a technical policy with our Enterprise Architecture. The following diagram illustrates the preferred target operating model for an ESB with Highways England.

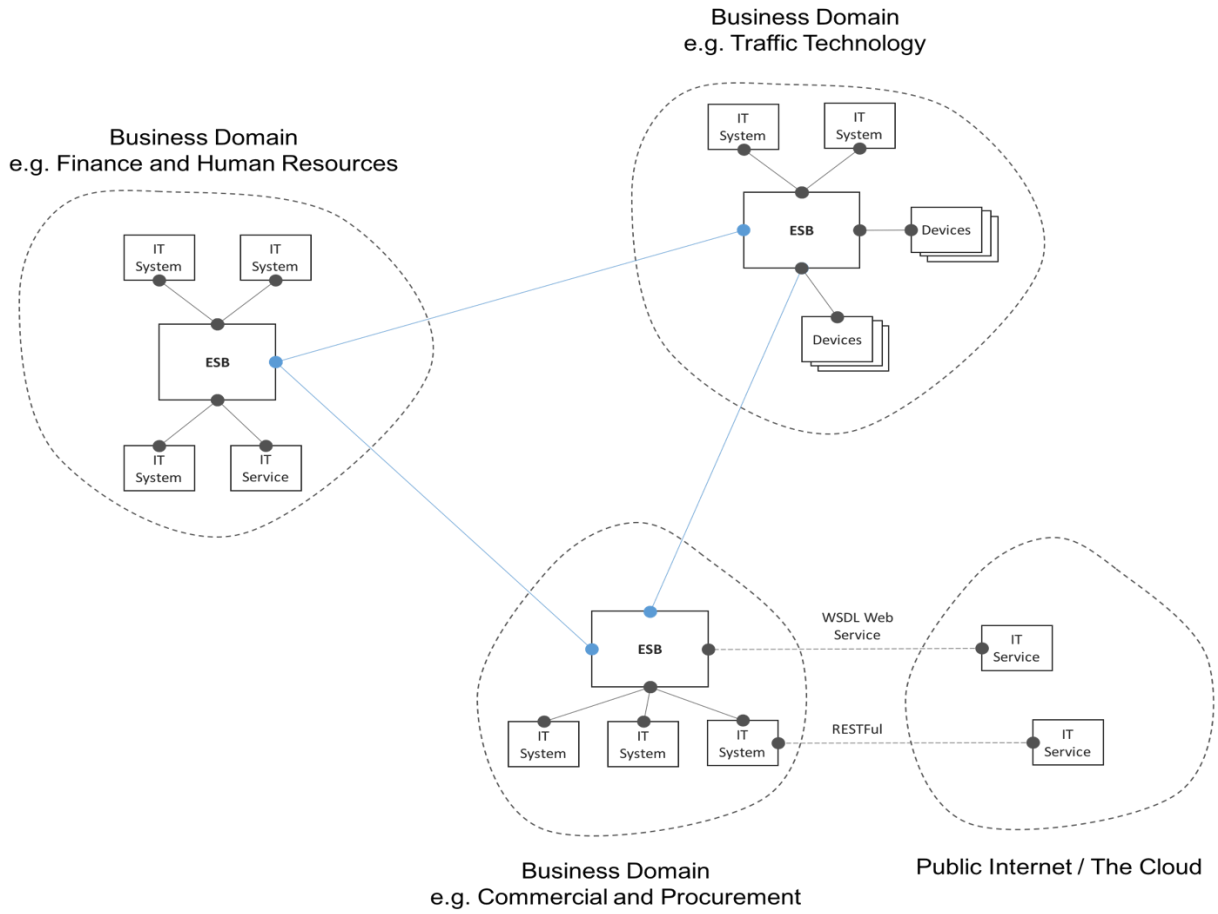

Figure 1 - Conceptual ESB Deployment Overview

This model allows each business domain a certain amount of autonomy over business interfaces within their domain to apply the required levels of security and regulate access. Services that are provided and managed within a business domain are made accessible through a single and resilient common registry of services.

The registry provides the governance for this deployment model.

Large Distributed Systems

Large distributed systems often have to deal with legacy technology and, our roadside estate illustrates this. We expect a long lifetime of operation for our roadside technology and varying ages of equipment are operationally deployed and must be interfaced with.

The ability to interface with our roadside information technology is an architecturally significant element of our infrastructure. It is the technical core of the multibillion pound Road Investment Strategy that is keeping roads safe and free flowing.

The following section describes the role that the ESB is fulfilling in the development of our next generation Advanced Traffic Management System (ATMS) solution that must work with new and legacy roadside technology.

Advanced Traffic Management System (ATMS)

The CHARM[1] project will deliver the next generation of day-to-day traffic operations technology for us. A key part of this technology is the Advanced Traffic Management System (ATMS), delivered using the DYNAC product from Kapsch TrafficCom.

DYNAC is based around the National Transportation Communications for ITS Protocol (NTCIP). This is a family of open standards that helps with the interoperability and remote control of roadside and transportation management devices. NTCIP is a standard adopted in North America. The Urban Traffic Management and Control (UTMC) is a standard widely adopted by local authorities in the UK. There is a separate standard in Europe called DATEX[2] with which UTMS is becoming more aligned with. Finally, most message signs, signals and MIDAS[3] deployed on the English motorway network use a protocol known as NMCS2 which is bespoke to Highways England.

The ATMS system needs to interface with a wide range of technologies and systems including:

- Roadside technology such as traffic counting (loops / radar), Signs (all VMS types), Signals (all indictors types) and fog detectors
- CCTV cameras and video recording
- Severe Weather Information System
- Local Authority Traffic Control Systems

Other operational technology systems have a need to exchange data and services between legacy and future systems, for example:

- Future asset and service management systems via web services.
- Fault and performance management systems using open protocols such as SNMP
- Other standards for device configuration (netconf), network configuration (DHCP) and remote maintenance (SSH2)

A solution is required to enable the standards supported by DYNAC and our technologies to work together. A ESB fulfils this role by using a canonical data model to communicate between the different data formats and standards. It loosely couples the DYNAC product with legacy devices and other systems that require data to be exchanged with our traffic management systems.

[1] https://www.kapsch.net/ktc/downloads/reference/references/KTC-BR-CHARM-Advanced_Traffic_Management_System-EN?lang=en-US Note, that this document shows a Mule ESB whereas a JBoss Fuse ESB is being implemented.
2 DATEX currently deals with centre to centre communications rather than centre to roadside
3 MIDAS – Motorway Incident Detection and Automatic Signalling

The CHARM ESB

The CHARM ESB will be the solution for controlling the exchange of traffic management data with all other systems. It will be reused to ensure that the definition and policies relating to interfaces are controlled and enforced.

The integration platform is based on Red Hat OpenShift Enterprise. This is used to support the deployment, scaling and lifecycle management services within our own data centres. OpenShift is an extension of the JBoss Fuse integration platform to provide a Platform-as-a-Service.

JBoss Fuse is a lightweight and modern Enterprise Service Bus (ESB) platform built upon proven open source technologies. It provides simplified management and monitoring capabilities. It implements the most commonly used integration patterns and provides a large library of connectors ready to use. The CHARM service bus will be operated from a Infrastructure as a Service (IaaS) supplier.

The platform is operated, configured and managed from a central location known as the fabric. The fabric simplifies integration and distribution of processing workloads on physical and virtual machines and in the public and private Cloud.

A key feature of the platform is containers. These are self-contained applications with all of their dependencies bundled together. A container can be hosted locally, remotely and even in the Cloud.

A simple overview of the core platform and features of the core ESB functions and supported operating platforms is show below.

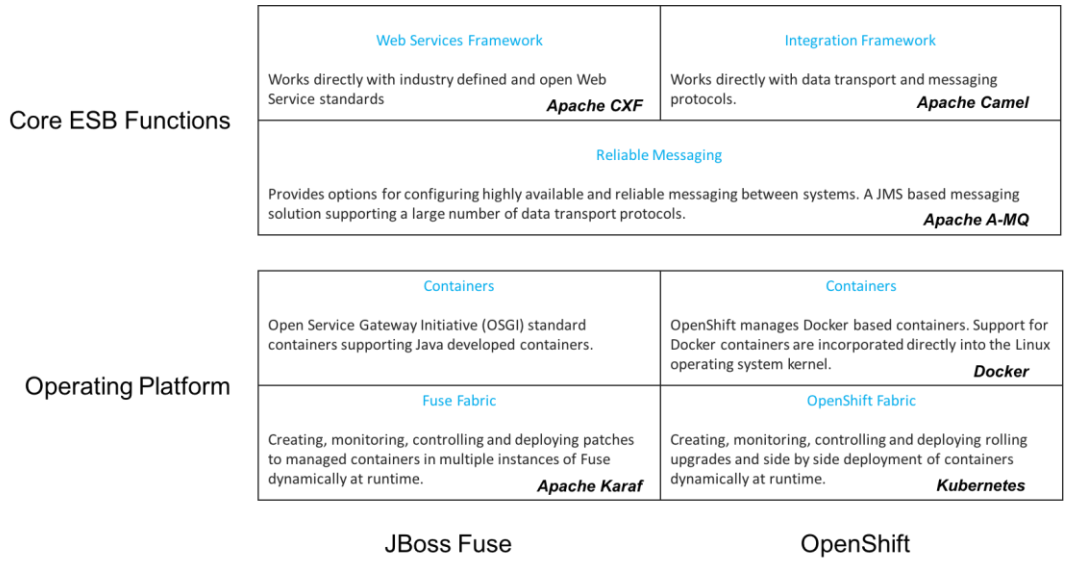

Figure 2 - High Level Overview of Fuse ESB

CHARM ESB Architecture

There is a requirement for the ATMS to interface with IT devices installed at the roadside and non-roadside IT systems. The logical architecture for this is supported by three ESB configurations called Core, Edge and Service.

The Core ESB is the key component of the ATMS and acts as the single point of control for all integration. It is designed to support medium to high complexity interfaces with the ATMS application. It uses reliable messaging to ensure correct distribution and guaranteed delivery of messages between systems and to and from the EDGE ESB.

The Edge ESB is the component that interfaces with all roadside devices. It is designed to handle the low-latency and high volume messages to and from devices, however the interfaces are intentionally less complex. It contains the interfaces that connect with several different device types operating on the National Motorway Communications System (NMCS2) including signs, signals and Motorway Incident Detections and Automatic Signalling (MIDAS).

The Service ESB interacts with external services where low-latency is important but less critical that interfacing with roadside devices.

The following diagram shows the design overview which is aligned to our preferred federated deployment approach.

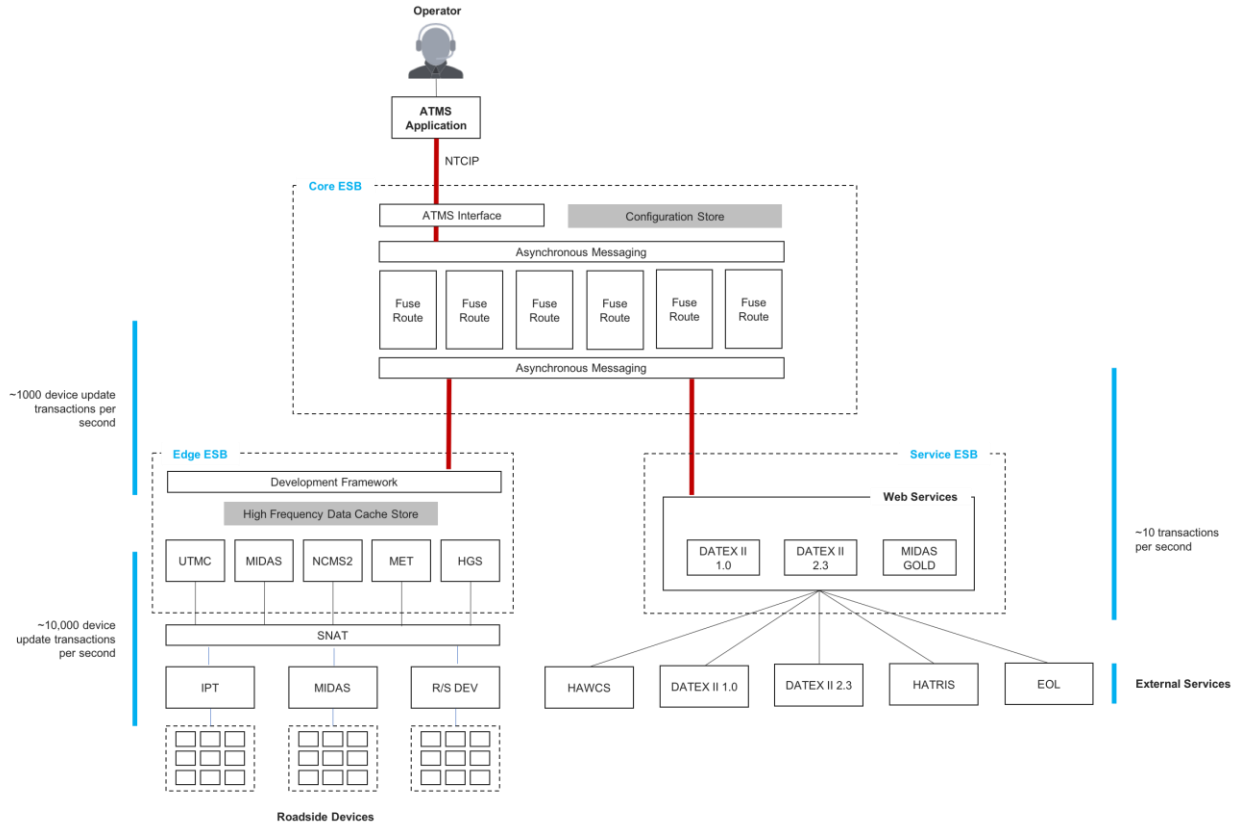

Figure 3 - Overview of the CHARM ESB Architecture

Controlling Technical Diversity

In a federated architecture model as shown in Figure 1 it would be possible to procure and implement different technical platforms however there is an architectural preference for all ESBs that will be under our direct control or supported for us by a service provider to operate on a single technology platform. This will prevent duplicating IT capabilities from various vendor products and reduce the complexity and operational costs of our IT environment.

Appendix A | Glossary

Term	Description
Adaptor	An adaptor represents the method of communication with devices. Adapters will be used to provide the communication with roadside devices. The will hold the specific communication settings and configuration needed to communicate with a specific type of physical device.
API	An Application Programming Interface prescribes the expected use and behaviour of a specific part of a system. It provides a specification for implementing interfaces to the services that an application provides.
API Gateway	A software architecture that makes it easier for developers to build an Application Programming Interface to access application services. An API Gateway is focused on agility and variability.
Cloud	Cloud is the term used to describe shared computer processing and data resources that are provided on demand. This can include entire applications, operating platforms and specific infrastructure resources that are typically delivered by service providers using their environments but can also be architected to be delivered on-premises.
Container	A container is a self-contained application with all of its dependencies bundled together so that it will run reliably when moved from one platform to another for example from a development to production environment. Containers run on a shared platform and are easier to change without having to reboot of the underlying operating system.
DATEX	DATEX is the European standard developed to facilitate the exchange of information between traffic management centres and service providers. DATEX II is the latest version of the standard that extends the standard to support a broader set of traffic and travel information exchanges. DATEX II is the reference for all applications requiring access to dynamic traffic and travel information in Europe. http://www.datex2.eu/content/datex-background
DYNAC	DYNAC is an advanced traffic management application developed by Kapsch TrafficCom that collects traffic and facilities-related data from highways, managed lanes, bridges and tunnels.
ESB	Enterprise Service Bus. A software architecture that describes a way of organising interfaces between distributed IT systems. Commonly implemented as a middleware product. An ESB is focused on providing stability, systems of record and reuse within an organisation.
JDBC	Java Database Connectivity. An API for the Java programming language that supports access to a database.
JMS	Java Message Service. An API for the Java Programming Language that

Term	Description
	supports sending messages between two or more clients.
JSON	JavaScript Object Notation. An open-standard and programming language independent format for specifying data as a collection of named (description) and attribute (value) pairs. http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf
NAT	Network Address Translation. A method of mapping one Internet Protocol (IP) address into another.
NMCS2	National Motorway Communications System Mark 2 refers to computer-controlled network of roadside devices on all motorways and trunk roads, the strategic road network, that Highways England is responsible for.
NTCIP	National Transportation Communications for Intelligent Transportation System Protocol. This is a family of open standards that helps with the interoperability and remote control of roadside and transportation management devices. Developed to support transportation systems in the North American marketplace.
OpenShift	OpenShift is a Platform-as-a-Service (PaaS) from RedHat that allows developers to build, host and scale container-based applications in a Cloud architected environment available online or on-premise.
OSGi	Open Services Gateway initiative. An open standard that specified a modular platform for the Java programming language. Application or components are represented as bundles (containers) that can be remotely installed, started, stopped, updated and uninstalled without requiring a reboot of the operating system.
Registry	A Registry holds information on what interface services exist and the details necessary to use them. It allows applications to dynamically retrieve the endpoint of the service at runtime and avoid point-to-point connections.
REST	Representational State Transfer is a style of architecture that builds on Internet standards to deliver a highly scalable architecture for web-based services.
SMAC	Social, Mobile, Analytics and Cloud. The term describes four technology categories that are driving business change. When integrated they offer business benefits through better association and understanding of business data.
SNAT	Source Network Address Translation or Source NAT. Used to ensure that roadside devices will always be presented with an acceptable connection. It allows multiple devices to share a single adaptor and for the redistribution of devices to a different adaptor to protect against failure scenarios.
SOA	Service-Oriented Architecture. A style of software design for dealing with business processes distributed over a large landscape of systems that are under the control of different owners.