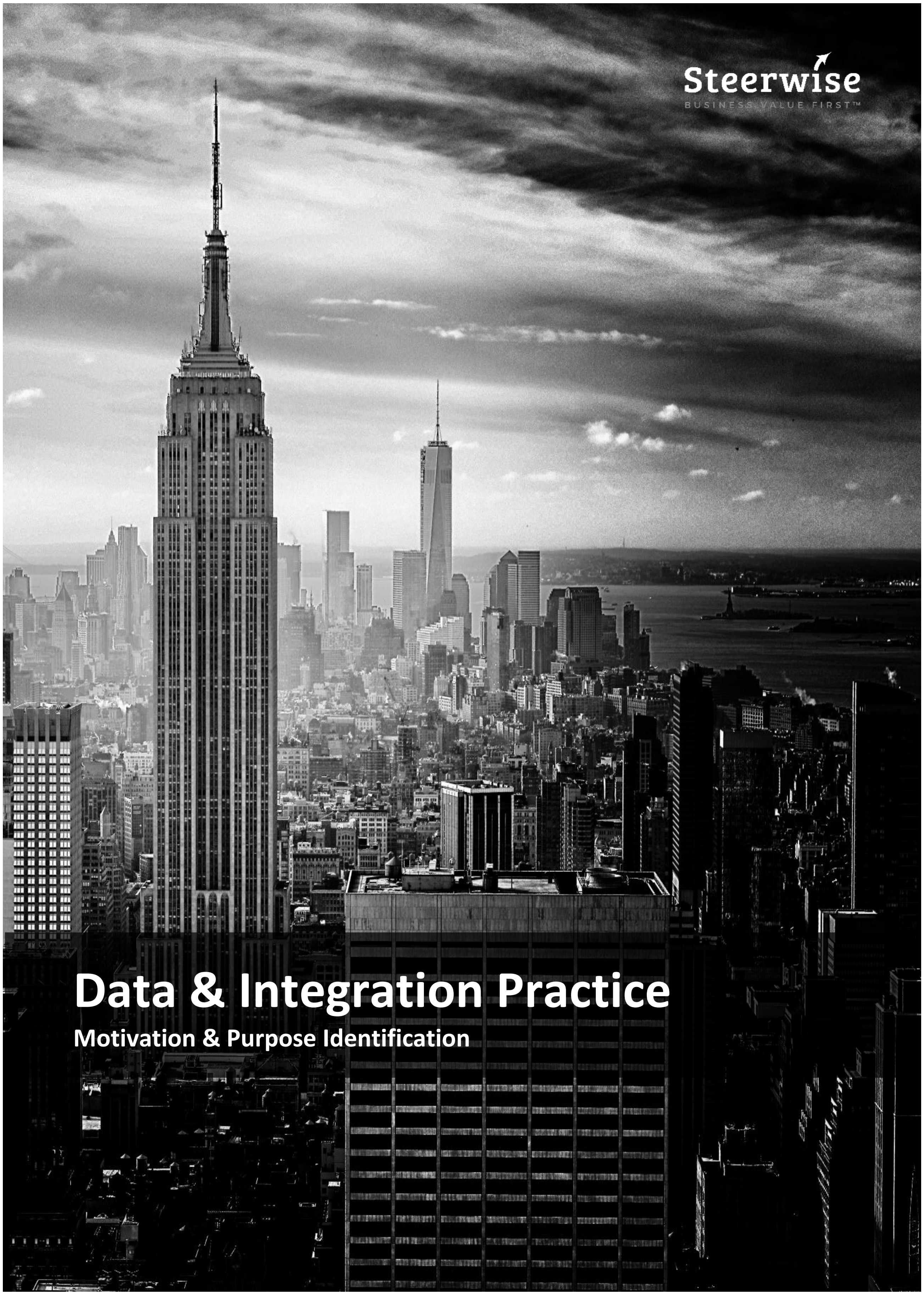

Steerwise

BUSINESS VALUE FIRST™

Data & Integration Practice

Motivation & Purpose Identification

Enterprise Data Platform

Toyota Financial Services (TFS) Drivers, Needs & Goals

Table of Contents

1.0 Introduction

1.1 Enterprise Data Platform (EDP) Background

1.1.1 Data & Integration Practice Challenges & Risks

Like most organizations, Toyota Financial Services (TFS) have multitude of application systems that are extremely useful to the functional areas that use them. These application systems are implemented over the years by various functional domain groups to meet their own specific needs, ranging from lightly configured to heavily customized vendor packaged solutions and from hybrid to completely purpose-built in-house solutions.

While these individual application systems may provide robust functionality with purpose-driven data assets within its own domain, they are rarely integrated together from the enterprise perspective. Over the years, these application systems are connected together predominantly using legacy-dependent batch-oriented integration methods in an ad-hoc fashion either directly in a point-to-point fashion or through a locally-built point solutions of data integration bus or a hub system, resulting in highly complex & redundant data integration functions, systems & platforms. Due to the practice of legacy-oriented localized integration, data is redundantly stored & processed using project-specific semantics by both application & integration platforms resulting in structural, functional & content level coupling of such systems leading to rigidity.

On the other hand, due to the absence of an active & coordinated enterprise-wide data management & governance practice, knowledge & accountability around data definition and functional context of data acquisition, production, utilization, etc. live & die with project lifecycles & software implementations leading to 'data illiteracy & unaccountability' issues across functional domains over time. As more and more functionalities are getting implemented using existing or new systems through project-driven point solutions, data of varying semantic types & instances of such types that support varying business needs are growing within & across functional domains leading to multiple inconsistent versions of truth of data & its corresponding business functions.

This current situation of tightly-coupled, highly complex, rigid, redundant spaghetti of data integration functions, systems & platforms predominantly interacting through legacy-technology based batch interfaces and the existence of multiple inconsistent versions of truth of data & functions across application as well as integration systems along with data illiteracy across functional domains of the enterprise around structural & content meanings of data & data lineage is causing the negative effects of *delays in serving the right data at the* *right time to the right internal & external customers, inadvertent data quality issues, metadata related problems, data errors & compliance risks, failures in planning & executing data-oriented projects on-time, on-budget & on-objectives, higher maintenance & support costs of integration systems and the supporting infrastructure,* etc.

Summary of Challenges & Effects:

Present Situation

- Data Illiteracy & Unaccountability across Functional Domains around Structural & Content Semantics of Data & Data Lineage
- Multiple Inconsistent Versions of Truth of Data & its Corresponding Functions across Application & Integration Systems
- Predominance of Legacy-System Influenced Batch-Oriented Integration Methods & Techniques across Data & Systems
- Tightly-Coupled, Highly-Complex, Rigid & Redundant Spaghetti of Data & Integration Functions, Systems & Platforms

Negative Effects

- Failures in Planning & Executing Data-Oriented Projects On-Time, On-Budget & On-Objectives
- Inadvertent Data Quality Issues, Metadata related Problems, Data Influenced Errors & Compliance Risks
- Delays in Serving the Right Data at the Right Time to the Right Internal & External Customers
- Higher Maintenance & Support Costs of Integration Systems & the Supporting Infrastructure

1.1.2 Enterprise Data Platform (EDP) Overview & Scope

Enterprise Data Platform (EDP) aims to bring together the following critical capabilities to build the foundational components for enabling & maturing highly competent data-driven culture at TFS:

- **real-time & batch access & delivery of data**
- **structural & semantic processing & integration of data**
- **right quality & end-to-end security of data**
- **business & technical model & execution related metadata**
- **delivery of high quality data & integration functions at speed**
- **delivery of flexible, scalable & resilient data & integration components**

The functional scope of Enterprise Data Platform (EDP) is bound to:

- **accessing & acquiring data in real-time & batch from all data sources**
- **integrating data at structural & content level semantics to improve quality**
- **serving & delivering well-linked-high-quality data at speed to all consumers**

1.1.3 Enterprise Data Platform (EDP) Goal & Strategy

The **Enterprise Data Platform (EDP)** **envisions to create a desired situation of highly competent data & integration practices to effectuate positive outcomes at all levels viz. innovation, strategy, project execution & operations.**

Desired Situation

- Higher Data Literacy & Clear Accountability across Domains around Structural & Content Semantics of Data & Data Lineage
- Matured Implementation of Enterprise-Aligned Harmonized Data & Consumer-Aligned Materialized Data Practices
- Right-Time Oriented Integration of Data & Application Systems using Modern Methods & Techniques
- Highly Flexible, Simplified, Modular, Loosely-coupled, Scalable and Resilient Data & Integration Functions, Systems & Platforms

Positive Effects

- Serving the Right Data with the Desired Quality at the Right Time to the Right Internal & External Customers
- Improved Data Quality, Well-Defined Metadata, Reduced Data Influenced Errors & Reduced Compliance Risks
- Successful Planning & Execution of Data-Oriented Projects On-Time, On-Budget & On-Objectives
- Lower Maintenance & Reduced Support Costs of Integration Systems & the Supporting Infrastructure

The **Goal of Enterprise Data Platform (EDP)** is to build a **system of people, process, technology & knowledge around data & integration practice to provide well-linked-high-quality data at speed across internal & external customers & partners at TFS.**

The **Implementation & Realization Strategy of Enterprise Data Platform (EDP)** is to establish an architecture approach based on enterprise-level strategic drivers & needs and to incrementally design, develop & deploy the functional & technical capabilities as prescribed by the architecture approach for enterprise-use through key TFS initiative, program & project specific requirements, priorities & resources such as Core Receivables Program.

1.2 Core Receivables Program Background

1.2.1 Core Receivables Program Overview

In order to propel Toyota Financial Services (TFS) to a new level of innovation and market competitiveness, **Core Receivables Program** was initiated to transform the retail, lease and asset inventory tracking functions into an enterprise-wide solution.

The Core Receivables Program aims to excel in customer-experience (CX) & innovation through well-integrated data, self-service, expanded retail & lease product offerings, etc. and to improve user-experience (UX) & operational excellence through integration & automation.

The implementation strategy of Core Receivables Program is to replace the existing legacy application systems viz. Shaw (Retail), LeMans (Lease) & Inventory Tracking System (Asset) using a vendor implemented commercial-off-the-shelf (COTS) application system with minimal impact to upstream and downstream application systems of TFS.

1.2.2 Core Receivables Data & Integration Drivers & Needs

To enable a successful delivery of Core Receivables Program and to provide a sustainable business value, the **Enterprise Data Platform (EDP)** around which all current and future data sources, consumers, services, and processes interact, is essential and a pre-requisite.

Currently, in order to avoid business risks, the strategy is to migrate to new COTS system incrementally from the existing legacy application systems instead of a big bang approach of deploying business functions and migrating production data all at once. The implication is that both legacy & new COTS application systems will coexist for a period of time in production till all functions & data are migrated. The Enterprise Data Platform (EDP) must support incremental implementation & migration of application systems capabilities in the form of immutable data stores, re-computation of datasets, flexible data model, microservices based integration functions, etc.

The coexistence scenario of new COTS & legacy application systems, unsynchronized arrival of data from multiple sources, unavailability of pre-computed datasets at certain source systems, pre-existing data needs of dependent application system functions & integration interfaces, etc. drive the need for heavy data processing such as end-of-day batch processing on the legacy systems, COTS system and the Enterprise Data Platform (EDP) in certain sequence. This may cause delays in serving the required datasets to the consuming systems on time. The Enterprise Data Platform (EDP) must support low latency ingression, processing & egression of large datasets to avoid service-level-agreement (SLA) violations.

The proliferation of redundant integration systems, integration interfaces, integration logic, datasets, etc. and the manual development, testing & deployment of integration systems slows the overall delivery of production ready application systems. The Enterprise Data Platform (EDP) must enable rationalization & refactoring of existing integration assets and automate Software System Development Lifecycle (SDLC) activities through DevOps practices. In addition, the Enterprise Data Platform (EDP) must provide foundational capabilities to build robust & simplified data & integration functions at speed.

Summary of Drivers & Needs:

Drivers	Needs
Incremental Implementation	Raw Data Layer Storing Data Source Specific Datasets without Modification
	Harmonized & Materialized Data Layer with Rapidly-Evolving Data Model
	Incrementally Deployable Data Integration Functions
	Incrementally Deployable Data Quality Functions
	Fact-Based Immutable Datasets Stored Forever (To Enable Re-computation)
Low Latency Data Processing	Low Latency Data Ingression of Varying Interface Patterns & Data Characteristics
	Low Latency Data Processing of Varying Workload Characteristics
	Low Latency Replay/Re-Computation Capabilities
	Low Latency Query & Search Capabilities
	Low Latency Data Egression of Varying Interface Patterns & Data Characteristics
Operational Efficiency	Existing Operational Data Hubs (ODS, IR, Focus Host, etc.) Rationalization
	Integration Interfaces Rationalization
	Integration Interfaces as Dumb Pipes Refactoring (i.e. Data Movement)
	Linearly Scalable Infrastructure with Long-Term Economic Viability
	Automation & DevOps Capabilities
Foundational Capabilities	Fine-Grained Harmonized Data & Coarse-Grained Materialized Data
	Near Real-Time Data (CDC & ESB) & Batch Data (MFT & ETL) Integration
	Metadata Management (e.g. Data Definitions, Data Lineage, etc.)
	Application & Data Security (e.g. Auth, Encryption, etc.)
	Highly Available, Fault-Tolerant & Resilient System