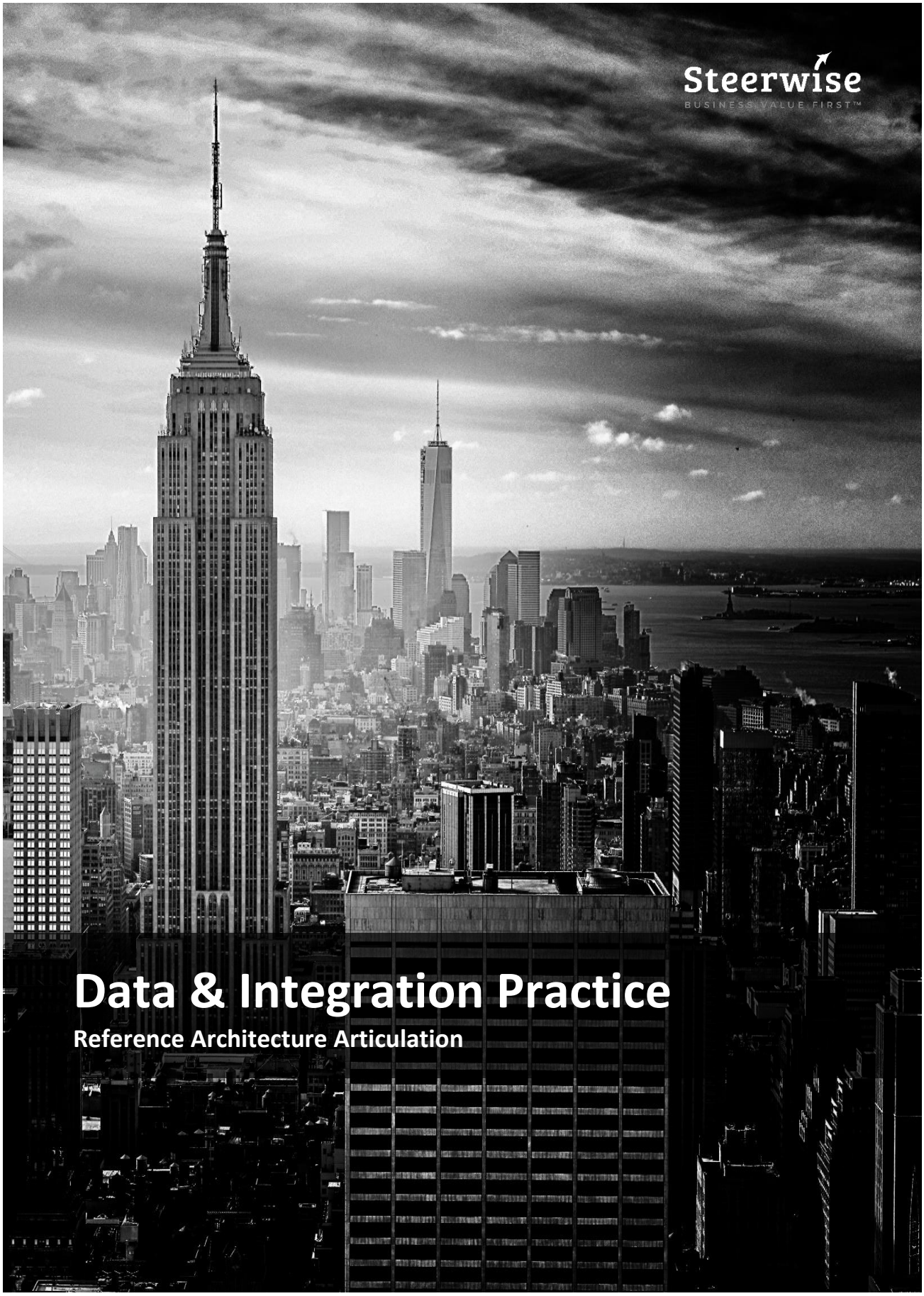

Steerwise
BUSINESS VALUE FIRST™
Data & Integration Practice
Reference Architecture Articulation

Enterprise Data Platform

Reference Architecture

Table of Contents

1.0 Reference Architecture

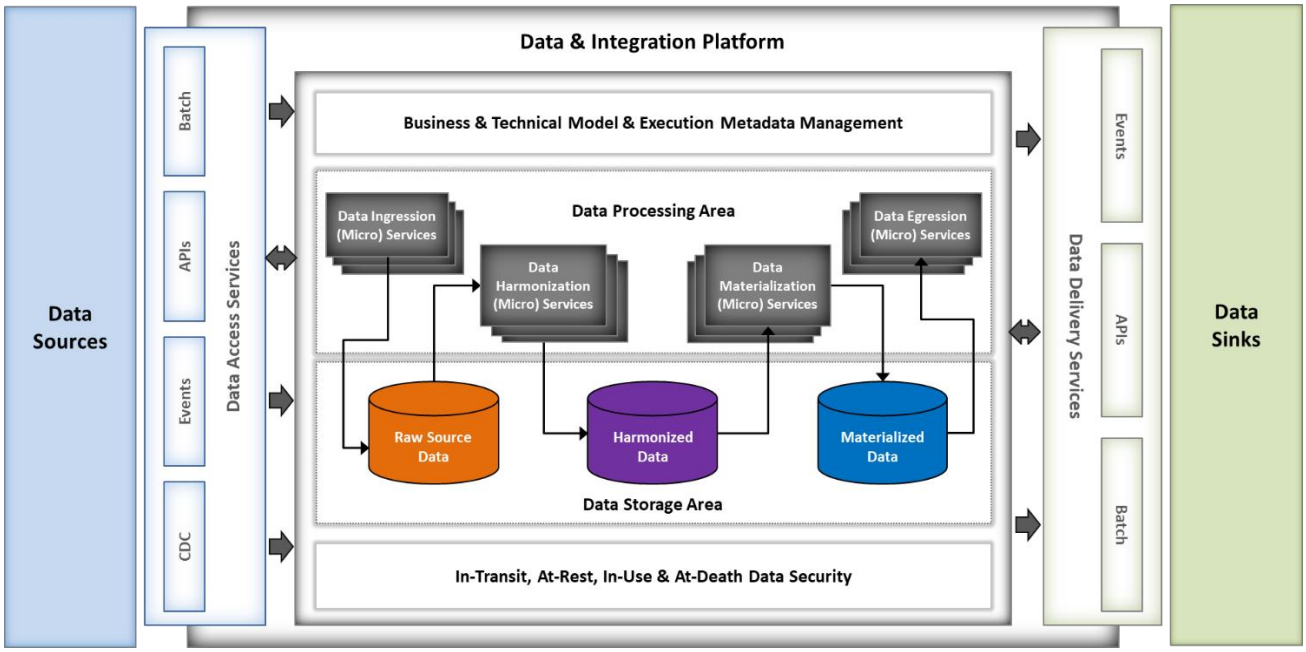
1.1 Enterprise Data Platform (EDP) Architecture Approach

1.1.1 Enterprise Data Platform (EDP) Reference Architecture

Enterprise Data Platform (EDP) Architecture incorporates the following **guiding principles** to meet the needs of the Enterprise through Core Receivables Program specific requirements:

- **Structurally-Flexible & Semantically-Cohesive Data Platform**
- **Highly Available, Self-Described, High Quality Data Assets**
- **Dumb-Pipes & Smart-Services Based Integration Platform**
- **Time-Machine Oriented Event-Driven Entity State Management**
- **Continuous Delivery of Modular & Incremental Functions**
- **Resilient, Low Latent & Highly Secured Enterprise Data Platform (EDP)**

The **Reference Architecture** of Enterprise Data Platform (EDP) is depicted below:



As shown in the reference view above, the platform mainly consists of the following key components viz., *Raw Source Data Storage Layer, Harmonized Data Storage Layer, Materialized Data Storage Layer, Data Processing Layer, Data Access Services Layer, Data Delivery Services Layer, Data Security & Metadata Management Layer.*

Data Access Services Layer provides inbound data movement capabilities through connectors & adapters supporting various interaction patterns & protocols:

- **Real-Time Acquisition:** Capture real-time events within data sources after key source-specific transactions or events (e.g. Change Data Capture (CDC))
- **Real-Time Reception:** Receive real-time events from data sources after key source-specific transactions or events (e.g. Messaging Bus)
- **On-Demand Acquisition:** Make a request-response call to data sources to access & acquire data (e.g. Web Services API)
- **On-Demand Reception:** Receive a request-response call from data sources for data ingression (e.g. Web Services API)
- **Bulk-Batch Acquisition:** Extract bulk data from data sources on a scheduled basis to load through ingression services (e.g. Extract-Transform-Load (ETL))
- **Bulk-Batch Reception:** Receive bulk data from data sources on a scheduled basis to load through ingression services (e.g. Managed File Transfer (MFT))

Data Delivery Services Layer provides outbound data movement capabilities through connectors & adapters supporting various interaction patterns & protocols:

- **Real-Time Propagation:** Send real-time events to data sinks after key transactions or events from within Enterprise Data Platform (EDP) (e.g. Messaging Bus)
- **On-Demand Propagation:** Make a request-response call to data sinks to notify or deliver data (e.g. Web Services API)
- **On-Demand Provision:** Receive a request-response call from data sinks for data egression (e.g. Web Services API)
- **Bulk-Batch Propagation:** Extract bulk data from Enterprise Data Platform (EDP) on a scheduled basis to deliver to data sinks through egression services (e.g. Extract-Transform-Load (ETL) & Managed File Transfer (MFT))
- **Bulk-Batch Provision:** Extract bulk data from Enterprise Data Platform (EDP) on a scheduled basis by data sinks (e.g. Extract-Transform-Load (ETL))

Data Access & Delivery Services standardizes the following:

- **Transport Mediums & Protocols: HTTP/S, REST, SOAP, OData, Messaging Server, File Server/System/FTP/S, Database & ODBC/JDBC**
- **Data Formats & Encodings: JSON, XML, Avro, Delimited, Fixed Length, UTF8, ASCII, EBCDIC**

Data Access & Delivery Services utilize API Gateway & Management infrastructure to manage & govern the lifecycle of all APIs including versioning, mediation, security, usage controls & monitoring.

Data Ingression Services:

- Utilize data access services to acquire or receive data from data sources and implement data ingestion functions
- Record source specific data in its original format in an immutable data store without updates & deletes
- Transform the original native data format to standard format such as CSV, JSON, Avro, etc. before publishing or loading the source data to the next stage of the data processing pipeline

Raw Data Layer:

- Stores data in its original native format without any modification (i.e. exact copy of source data) for auditing & compliance purposes
- Keeps a full versioned history of each data entity over time as the append-only nature of immutable data makes it easier to implement a system that can easily evolve over time as the needs change
- Supports rewind capabilities to view the state any point in time & allows re-computation of whole data set in case of errors or corruption or improved algorithms, and supports system evolution with changing needs
- Requires optimizing disk usage through storage tiering based on temporal usage patterns (e.g. hot, warm, cold & frozen) for performance, reliability & cost efficiency purposes

Data Harmonization Services:

- Implement structural level (i.e. structural data transformation) & content level (i.e. derivation, calculation, aggregation, summarization logic) semantic data integration & data quality functions from the shared enterprise perspective
- Are highly scalable modular functions that process data in parallel by entity types such as Contract, Account, Customer, Asset, Dealer, 3rd Party, Corporate Structure, etc. and in chained sequence by dependent processing functions
- Utilize flexible data model as data entities may arrive at different times in different sequences from varying data sources
- Process the formatted source data with harmonization logic using a set of microservices and stores the outputs into the harmonized data store
- Build a re-computation function by default to implement a robust & resilient system and where & when necessary builds an incremental computation function to reduce the write latency through efficient data processing

Harmonized Data Layer:

- Stores fine-grained well-linked dataset in a standardized format with high cohesion for enterprise use
- Utilizes flexible data modeling patterns for supporting system evolution over time
- Harmonizes data structures across data sources with unambiguous data types representing distinct business concepts

Data Materialization Services:

- Implement structural level (i.e. structural data transformation) & content level (i.e. derivation, calculation, aggregation, summarization logic) semantic data integration & data quality functions from the perspective of specific consumers
- Are highly scalable modular functions that process data in parallel by entity types such as Contract, Account, Customer, Asset, Dealer, 3rd Party, Corporate Structure, etc. and in chained sequence by dependent processing functions
- As functions are incrementally build and the systems may evolve, it utilizes a flexible data model to process new or changed data entities
- Process the harmonized data with materialization logic using a set of microservices and stores the outputs into the materialized data store
- Build a re-computation function by default to implement a robust & resilient system and where & when necessary builds an incremental computation function to reduce the write latency through efficient data processing

Materialized Data Layer:

- Stores pre-prepared well-linked dataset in a standardized format with high cohesion for consumer specific use
- Utilizes flexible data modeling patterns for supporting system evolution over time
- Harmonizes data structures across data consumers with unambiguous data types representing distinct business concepts

Data Egression Services:

- Utilize data delivery services to provision or transmit data to data sinks and implement data presentation functions
- Extract sink specific data in its standardized format such as CSV, JSON, Avro, etc. from the materialized data layer
- Transform the dataset to consumer specific format before provisioning and transmitting it to the consumer

Data Security Services:

- Provide authentication & authorization functions to control access to data & system components

- Protect sensitive data in any state: *at-rest in the data store, in-use during data processing, in-transit to other data systems, and at-death when deleted or erased;* and in any environment: *operations, system development, test, QA, and production*
- Provide format & data type agnostic encryption to prevent machines & humans from seeing sensitive data in the clear
- Provide centralized & stateless key management
- Perform encryption at big data scale with millions of transactions per second through horizontal scaling
- Collect, log & report access audit history

Metadata Management Services:

- Allow to discover, extract, harvest, link & catalog technical metadata
- Enable monitoring capabilities through business & technical execution metadata
- Provide data lineage describing where data came from, where it's going, how it's transformed, how it was derived, how it was updated over time, etc.

1.1.2 Enterprise Data Platform (EDP) Implementation Strategy

Enterprise Data Platform (EDP) will be implemented incrementally through a series of phases where foundational capabilities & advanced capabilities will be built along the way as prioritized by specific program or project.

The following are the implementation phases of Enterprise Data Platform (EDP):

Phase 1: Exploration Phase

- Identify blueprint (vendor agnostic & implementation specific) & footprint (vendor specific & deployment specific) architecture options
- Conduct proof-of-concept to compare & contrast architecture options
- Review & finalize Enterprise Data Platform (EDP) architecture

Phase 2: Foundation Phase

- Prioritize the foundational capabilities of Enterprise Data Platform (EDP)
- Implement the foundational capabilities of Enterprise Data Platform (EDP) as a Reference Implementation

Phase 3: Hubification Phase

- Route existing integration interfaces through Enterprise Data Platform (EDP)
- Build pass-through microservices across all layers & flow the data through the Enterprise Data Platform (EDP)
- Build reconciliation microservices to compare ingress data with egress data and to verify & validate the pass-through functions

- Continuously run for a period of time & harden the routed pass-through interfaces
- Migrate & Go live with first version of Enterprise Data Platform (EDP)

Phase 4: Modernization Phase

- Enable Change Data Capture (CDC) on legacy systems for scoped data entities & acquire data in near real-time
- Move transformation & business logic from integration interfaces to a set of microservices within Enterprise Data Platform (EDP)
- Rationalize & consolidate existing integration interfaces to reduce the data movement footprint where possible

Phase 5: Harmonization Phase

- Analyze & finalize the incremental functional scope by identifying a set of processes
- Identify upstream & downstream application systems and its corresponding interfaces
- Connect to upstream & downstream application systems that are scoped
- Build the harmonized data model for the functional scope
- Build data processing pipelines through a set of microservices across all layers of the Enterprise Data Platform (EDP) for each of the interfaces