



National Data Management Guidelines

Data Architecture and Modeling



Table of Contents

1. Data Architecture and Modeling Domain Introduction

- 1.1 Domain Overview
- 1.2 Target Audience
- 1.3 Abbreviations and Glossary
- 1.4 Readability Guide
- 1.5 Stakeholders Involved
- 1.6 Key Success Factors
- 1.7 Compliance with National Data Standards
- 1.8 Data Architecture and Modeling Process Map

2. Data Architecture & Modeling Assessment & Planning

- Overview
- 2.1 Data Architecture Plan
- 2.2 Data Modeling Guidelines & Methodology
- 2.3 Baseline Data Architecture Assessment
- 2.4 Data Architecture Gap Analysis
- 2.5 Data Architecture Roadmap

3. Data Architecture & Model Design

- Overview
- 3.1 Target Data Architecture
- 3.2 Analytical Data Model
- 3.3 Data Model Metadata

4. Data Architecture & Modeling Tools

- Overview
- 4.1 Data Architecture & Modeling Tools

5. Data Architecture & Modeling Performance & Change Management

- Overview
- 5.1 Data Architecture & Modeling KPIs
- 5.2 Data Architecture & Modeling Monitoring & Reporting
- 5.3 Data Architecture Change Management Process
- 5.4 QDTI

6. Data Architecture & Modeling Documentation

Overview

6.1 Data Architecture & Modeling Repository

7. Appendix

7.1 Templates and Frameworks

1. Data Architecture and Modeling Domain Introduction



1.1 Domain Overview

The Data Architecture & Modeling domain establishes a structured approach for designing, implementing, and maintaining a scalable data architecture and data models. It is fundamental to enabling consistent data definitions, seamless data integration, and efficient data flows across the organization. By aligning data structures with business processes and analytical needs, this domain ensures a coherent data environment that supports interoperability, governance, and strategic decision-making.

The primary objective of the Data Architecture & Modeling domain is to ensure that entities can design, implement, and maintain a robust data architecture that aligns with strategic goals. By following the below controls and respective activities, entities can create a unified view of data, streamline data architecture and modeling processes, and ensure alignment with the national data standards.

Key Components of the Data Architecture and Modeling Domain:

2. Data Architecture & Modeling Assessment & Planning

This control ensures that the Entity develops a structured, strategic approach to data architecture and modeling, ensuring alignment with business objectives, governance standards, and interoperability requirements. It involves defining modeling guidelines, assessing existing capabilities, identifying gaps, and establishing a roadmap for long-term improvements. This phase involves:

- **Data Architecture Plan:** This activity guides the Entity in initiating and structuring its Data Architecture Plan through a series of foundational steps. These include aligning architecture objectives with business and regulatory priorities, defining scope and architectural domains, identifying key stakeholders and responsibilities, and organizing a structured kick-off meeting. Together, these steps ensure clarity, accountability, and alignment across business and technical teams, while establishing a shared understanding of current and future architecture needs.
- **Data Modeling Guidelines & Methodology:** Leveraging the NPC data architecture & modeling guidelines, develop and formalize comprehensive Entity-specific data architecture & modeling guidelines that ensure consistency, compliance, and governance across all conceptual, logical, and physical data models. This includes establishing modeling standards, defining naming conventions, version control mechanisms, and ensuring compliance with NPC's prescribed methodologies (e.g., Star Schema).
- **Baseline Data Architecture Assessment:**
 - **Data Architecture Baseline Assessment:** This assessment focuses on identifying existing architecture artifacts and evaluating the Entity's current data architecture to assess system-level data movement. It also supports the Entity in ensuring that all architectural components including source systems, data platforms, integration points, architecture diagrams, and supporting metadata are comprehensively documented to provide a structured and holistic view of the overall data architecture and support future assessment and planning efforts.
 - **Baseline Analytical Data Model Assessment:** This assessment focuses on identifying and reviewing existing data modeling artifacts, extracting and evaluating the Entity's current analytical data models, documenting associated facts, dimensions, and data lineage, and developing a baseline physical data model diagram to inform future assessment and planning efforts.

➤ **Data Architecture Gap Analysis:**

- **Data Architecture Gap Analysis:** This activity guides the Entity in conducting a gaps analysis of the Entity's current data architecture against the Entity's data strategy, NPC's data architecture & modeling guidelines, and general best practices across data governance, scalability and performance, integration and interoperability, security and compliance.
- **Analytical Data Model Gap Analysis:** This activity guides the Entity in conducting a gap analysis of the Entity's current analytical data models against the Entity's data strategy, NPC's data architecture & modeling guidelines, data model integrity and consistency, and optimization and performance gaps.

➤ **Data Architecture Roadmap:** This activity guides the Entity in developing a structured data architecture and modeling design and implementation roadmap, outlining key initiatives, key activities, sub-activities, milestones, technologies, dependencies, assumptions, risks, stakeholders, budgets, and timelines to support planning and execution.

3. Data Architecture & Model Design

This control ensures the Entity adopts a structured, standardized approach to data architecture and modeling, aligned with its data strategy, business objectives, and national data standards. It provides guidance for defining the target state data architecture, analytical data models, and associated data lineage. In addition, it supports the establishment of governance mechanisms and the documentation of metadata for target architectures and analytical data models, enabling integration, scalability, and sustainable data management across the Entity. Specifications include:

- **Target Data Architecture:** This activity guides the Entity in designing its target data architecture to support analytical and reporting requirements, in alignment with NPC's data architecture and modeling guidelines. The design process considers the Entity's data strategy, business processes, integration and scalability needs, performance requirements, and previously identified architectural gaps, while ensuring adherence to established governance and security controls.
- **Analytical Data Model:** This activity guides the Entity in designing the target conceptual, logical, and physical analytical data models in alignment with NPC's data architecture & modeling guidelines, as well as the target data lineage. The design process considers the Entity's data strategy, business processes, and previously identified data modeling gaps whilst ensuring adherence to best practices for relational and analytical data modeling.
- **Data Model Metadata:** This activity guides the Entity in implementing comprehensive metadata management practices to ensure proper version control, traceability, and governance of analytical data models. This includes documenting metadata for conceptual, logical, and physical analytical data models, including fact and dimensions tables, and respective relationships and attributes, supporting the tracking of changes, discoverability, impact analysis and regulatory compliance.

4. Data Architecture & Modeling Tools

This section focuses on the selection, implementation, and use of tools that enable effective data architecture and modeling practices. These tools support key activities such as data model design, lineage tracking, metadata management, and the deployment of physical data models. Implementing the right tools ensures consistency, accuracy, and scalability in how data assets are architected, modeled, and governed across the Entity.

- **Data Architecture & Modeling Tool Selection:** This activity outlines the selection process of a tool that support data modeling, lineage tracking, and data model metadata governance. The tool should support the development of the conceptual, logical, and physical data models.
- **Data Architecture & Modeling Implementation:** This activity focuses on the implementation of the selected data modeling tool. Additionally, it covers configuring the modeling environment, deploying physical data models within the Entity's database platform, and ensuring alignment with metadata standards, governance requirements, and performance expectations.

5. Data Architecture & Modeling Performance & Change Management

This control emphasizes performance monitoring and effective change management for data architecture and modeling. It involves setting KPIs to track architectural effectiveness and managing updates in alignment with evolving business needs.

- **Data Architecture & Modeling KPIs:** Define KPIs to measure the effectiveness and progress of the data architecture, covering model completeness and compliance to national standards.
- **Data Architecture Monitoring & Reporting:** Continuously track KPIs and report on data architecture performance, enabling data-driven improvements and timely interventions.
- **Data Architecture Change Management:** Establish a structured process for handling changes to the data architecture, ensuring that updates are aligned with strategic goals and properly documented for consistency and traceability.
- **QDTI:** Evaluating the Entity's adherence to the National Data Architecture and Modeling Standards as part of the broader QDTI framework.

6. Data Architecture & Modeling Documentation

This control ensures the creation of a centralized repository for all documentation related to data architecture and modeling. It enhances accessibility, accuracy, and version control of the Entity's data architecture and modeling resources.

- **Data Architecture & Modeling Repository:** This activity guides the establishment of a centralized repository to store all data architecture and modeling documentation, including architectural diagrams, data model diagrams, and supporting metadata. This repository supports traceability, compliance, and accessibility, enabling stakeholders to access the latest architectural resources efficiently.

1.2 Target Audience

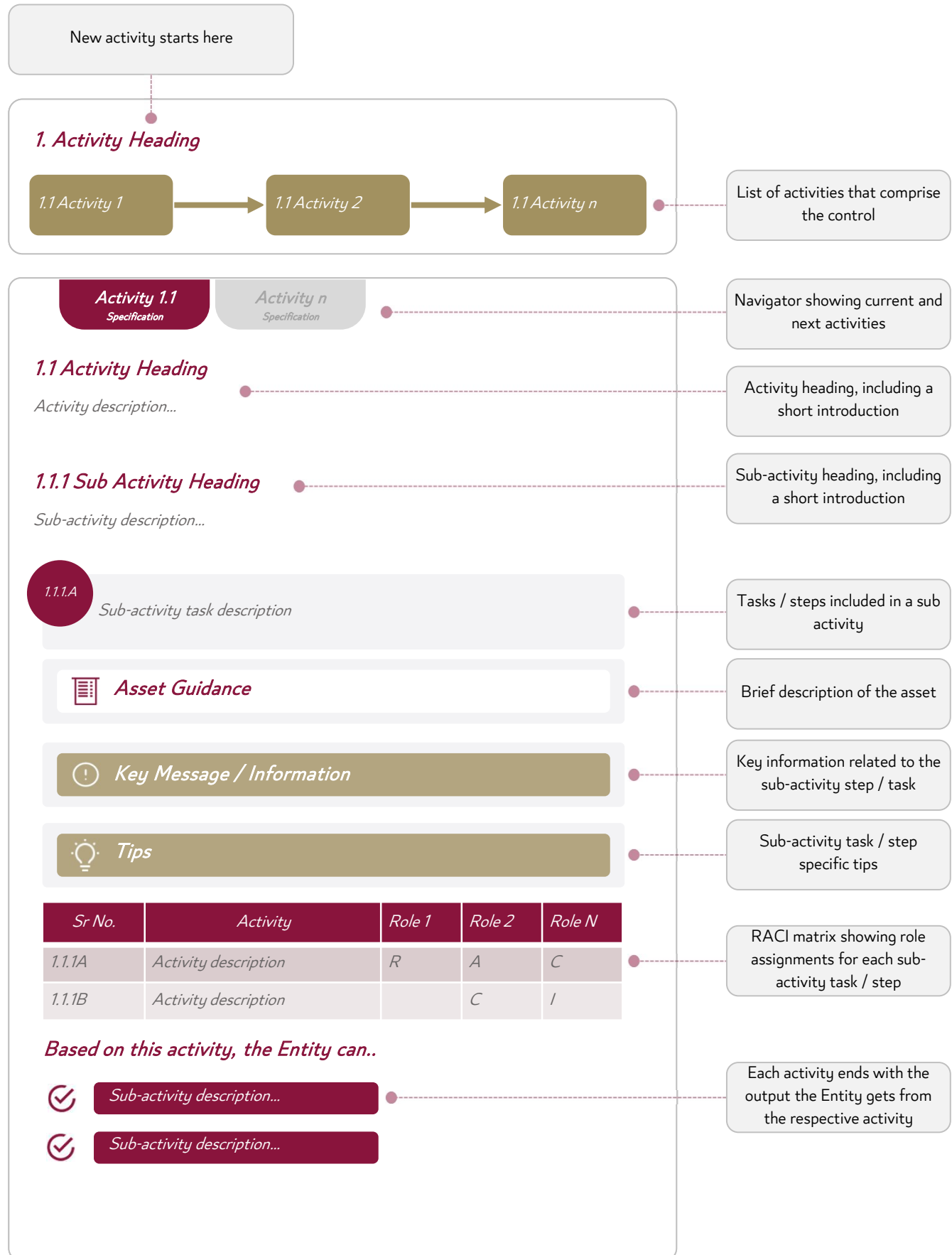
This handbook is intended for the Data Management and Security Director. It is also suitable for section heads and other individuals deemed responsible for managing data architecture and data models. The handbook serves as a resource, offering practical guidance to help establish the essential components of the Entity's data management program, setting the foundation for its data management journey.

1.3 Abbreviations and Glossary

Abbreviation/Term	Definition / Long Name
Bronze Layer	The raw data storage layer in a data warehouse where unprocessed data is ingested and retained for historical tracking.
Conceptual Data Model	A high-level data model that defines key business entities and their relationships without technical details.
Common Warehouse Metamodel (CWM)	A standard for metadata modeling and exchange, supporting interoperability between different data warehouse tools and systems.
DAM	Data Architecture and Modeling
DM	Data Management
DMBOK	Data Management Body of Knowledge
Data Warehouse (DWH)	A centralized system that stores, integrates, and processes structured data for analytics and reporting.
Analytical Data Model	A structured representation of the Entity's data assets that support analytical requirements, providing a unified view of the respective data across systems to support consistency, integration, and decision-making.
Extract, Transform, Load (ETL)	A process used to extract data from source systems, transform it into a suitable format, and load it into a target system, such as a data warehouse or analytics platform.
Gold Layer	The refined, final storage layer in a data warehouse, optimized for reporting and business intelligence.
Logical Data Model	A detailed representation of data that includes attributes, keys, and relationships but is independent of physical storage.
Master Dataset	Is the consistent and uniform set of identifiers and extended attributes that describes the core entities of the enterprise such as citizens, employees, suppliers
Metadata	Information about the data collected, this includes information about technical and business processes, data rules and constraints and logical and physical data structures
Physical Data Model	The technical representation of a database schema, including table structures, indexes, and data storage configurations.
QDKC	Qatar Data Knowledge Case
RAID Log	A tool for tracking Risks, Actions, Issues, and Dependencies in a project or initiative, ensuring visibility and proactive management of potential challenges.
Role-Based Access Control (RBAC)	A security model that restricts system access based on user roles and permissions.
Reference Dataset	Classifies or relates to other data or information; It describes the name and code of core entities of the business or enterprise, including customers, citizens, suppliers, sites, hierarchies, organization departments

Abbreviation/Term	Definition / Long Name
SME	Subject Matter Expert
Silver Layer	The intermediate storage layer in a data warehouse where data is cleaned, transformed, and prepared for analysis.
Statement of Architecture Work (SAW)	A formal document outlining approved data architecture changes, including their scope, rationale, and implementation plans.
QDTI	Qatar Data Trust Index
Strategic KPIs	These are the measures that need to be tracked for the strategic objectives to assess progress and indicate whether strategic objectives are being met or not
Strategic Objectives	These are the key goals that the Entity's data management program aims to achieve at a strategic level, aligning with broader business priorities and guiding the direction of data management efforts for long-term success.
Unified Modeling Language (UML)	A standardized visual modeling language used to represent system designs and data structures, commonly employed in software and data architecture design.

1.4 Readability Guide



Note: Should an activity only contain one sub activity, this sub activity will be in the format of "1.1.A" instead of "1.1.1.A"

1.5 Stakeholders Involved

Below are the stakeholders and actors who contribute to the different processes outlined in the upcoming chapters.

Stakeholder	Role Description
Data Governance Office (DGO)	A Data Governance Office (DGO) is the centralized national body in the National Planning Council responsible for facilitating and coordinating data governance efforts within all associated government entities. It serves as the principal body overseeing the management of data as a strategic asset, ensuring that data-related processes align with the national objectives and comply with relevant policies and regulations.
Data Management & Security Director	The Data Management and Security Director is responsible for leading and overseeing the development and implementation of the data agenda within the government entity, ensuring alignment with strategic objectives and compliance with security regulations. This role promotes best practices in data management while ensuring robust security measures are in place to safeguard the Entity's data assets. The director bridges business and IT to support data-driven decision-making, elevate data as a strategic asset, and ensure data is managed and secured across its lifecycle. Additionally, the director fosters a culture of data security and literacy within the entity, supervises the value realization agenda for data, and ensures alignment with national security policies and standards, as well as the vision of executive leadership.
Data Management and Governance Section Head	The Data Management and Governance Section Head is responsible for leading, managing, and overseeing both data management and governance initiatives within the government entity. This role focuses on establishing and implementing best practices to ensure data governance and management align with the Entity's business requirements, strategic objectives, and compliance with national data standards. The Section Head collaborates with various teams across the Entity to develop and operationalize frameworks, policies, and processes for data governance and management. This includes improving data quality, accessibility, and protection, while also ensuring data lifecycle management practices are in place. The Section Head fosters a data-driven culture and ensures that data is treated as a strategic asset to drive decision-making and operational excellence.

Stakeholder	Role Description
Data Architecture and Analytics Section Head	The Data Architecture and Analytics Section Head oversees the data architecture and analytics agenda/initiatives within the government entity, ensuring alignment with strategic objectives and national data management standards. This role is responsible for designing and maintaining a secure, scalable data infrastructure that supports data integration, management, and accessibility, as well as Overseeing data modeling, database design, and system integration, developing advanced analytics, data visualization, and reporting solutions to enable informed decision-making. The Section Head collaborates with departments to identify key performance indicators (KPIs), analyze trends, and provide actionable insights that optimize processes, enhance service delivery, and support policy-making. By combining expertise in data architecture and analytics, this role drives a culture of data-driven innovation and operational excellence across the Entity.
Data Governance Specialist	The Data Governance Specialist oversees the frameworks, policies, and procedures of governance and ensures their implementation to guarantee the effective management and utilization of data and resources within the Entity. This role includes developing and maintaining governance structures, leading compliance initiatives, and fostering a culture of accountability and transparency.
Metadata Specialist	The Metadata Specialist is responsible for managing the Entity's data catalog and metadata assets to enhance data visibility, accessibility, and governance. This role focuses on developing and maintaining metadata standards, ensuring metadata quality, and supporting integration with the data catalog. The specialist plays a key role in enabling data discovery, supporting data-driven initiatives, and ensuring alignment with the Entity's data governance and management strategies.
Data Architect	The Data Architect is responsible for designing and implementing the Entity's overall data architecture, including logical and physical data models. This involves creating secure, scalable data structures that align with business needs, security requirements, and national and international data management and governance standards. The Architect ensures proper data integration, quality, and accessibility across systems, working with teams to maintain consistency and accuracy, and support data-driven decision-making to optimize data as a strategic asset.

Stakeholder	Role Description
Data Strategy Expert	The Data Strategy Expert is tasked with developing and promoting a data management and analytics strategy to achieve broad acceptance across the government entity. The expert identifies innovative ways to utilize data to achieve strategic objectives and enhance effective decision-making. Additionally, they introduce frameworks and practices to deliver accurate, reliable, and actionable data insights.
Business Data Analyst	The Business Data Analyst is responsible for collecting and analyzing data from various sources to extract valuable insights that support strategic decision-making within the Entity. The specialist designs reports and dashboards to present analyses in clear and understandable ways and works closely with other teams to ensure data quality and alignment with business needs.
Data Visualization Officer	The Data Visualization Officer is responsible for building dashboards and visual reports based on data, which are used to provide insights to leadership for decision-making.
Data Owners	Data Owners are responsible for their department's (or sub-department's) data and its quality throughout its lifecycle across systems and ownership boundaries. They are accountable for supporting, guiding, reviewing, correcting, and/or approving the work of the data stewards managing their data to ensure their datasets are maintained to the highest possible standards.
Data Stewards	Data Stewards are responsible for identifying datasets related to relevant subdomains for inventory purposes and monitoring data quality metrics within and across entities. They are nominated as a business representative with a deep understanding of a specific data subdomain and can identify functional data requirements and needs.

1.6 Key Success Factors

Alignment with Business Objectives: Success in data architecture depends on ensuring that data design and structure align with the Entity's overarching goals. This alignment facilitates the seamless integration of data architecture with strategic initiatives, enhancing support for the Entity's analytical requirements.

Detailed Planning and Roadmap: A well-defined roadmap is essential for transitioning from current to target data architecture. This includes setting clear milestones, timelines, and resource allocations, enabling structured progress and effective management of dependencies across projects.

Standardized Modeling and Documentation Practices: Establishing consistent standards for data modeling and documentation promotes uniformity and scalability. By defining clear guidelines for data models and metadata, the Entity ensures that data is structured, documented, and understood universally.

Scalability and Flexibility in Design: Designing a flexible data architecture that can scale with changing data volumes, new technologies, and evolving business needs is crucial. A scalable architecture minimizes rework and supports future growth, enabling the Entity to adapt without compromising data integrity.

Continuous Performance Monitoring: Setting up KPIs and regular monitoring mechanisms enables the Entity to assess data architecture performance and make timely adjustments. This ensures that the architecture remains aligned with business objectives, maintains high standards, and evolves with operational demands.

Dedicated Resource Allocation: Allocating specialized roles, such as data architects and stewards, with the necessary resources ensures dedicated oversight and maintenance of the data architecture. These roles provide expertise in managing complex data landscapes and contribute to the architecture's resilience and reliability.

Comprehensive Change Management Process: A structured change management framework is essential to handle updates in data architecture systematically. This framework ensures that all changes are aligned with strategic objectives, properly documented, and communicated to stakeholders to prevent disruptions.

These success factors are instrumental in establishing a future-ready data architecture that supports efficient data management, fosters data-driven culture, and drives the Entity's long-term vision.

1.7 Compliance with National Data Standards

National Data Standards		QDKC
Control	Specification	Activity
DAM.1.1 Data Architecture & Modeling Assessment & Planning	DAM.1.1.1 Data Architecture Plan	2.1 Data Architecture Plan
	DAM.1.1.2 Data Modeling Guidelines & Methodology	2.2 Data Modeling Guidelines & Methodology
	DAM.1.1.3 Baseline Data Architecture Assessment	2.3 Baseline Data Architecture Assessment
	DAM.1.1.4 Target Data Architecture	3.1 Target Data Architecture
	DAM.1.1.5 Data Architecture Gap Analysis	2.4 Data Architecture Gap Analysis
	DAM.1.1.6 Data Architecture Roadmap	2.5 Data Architecture Roadmap
DAM.2.1 Data Architecture & Model Design	DAM.2.1.1 Analytical Data Model	3.2 Analytical Data Model
	DAM.2.1.1 Data Model Metadata	3.3 Data Model Metadata
DAM.2.2 Data Architecture & Modeling Tools	DAM.2.2.1 Data Architecture & Modeling Tools	4.1 Data Architecture & Modeling Tools
DAM.3.1 Data Architecture & Modeling Performance & Change Management	DAM.3.1.1 Data Architecture & Modeling KPIs	5.1 Data Architecture & Modeling KPIs
	DAM.3.1.2 Data Architecture & Modeling Monitoring & Reporting	5.2 Data Architecture & Modeling Monitoring & Reporting
	DAM.3.1.3 Data Architecture Change Management Process	5.3 Data Architecture Change Management Process
DAM.3.2 Data Architecture & Modeling Documentation	DAM.3.2.1 Data Architecture & Modeling Repository	6.1 Data Architecture & Modeling Repository

Legend: Capability Level



Foundational

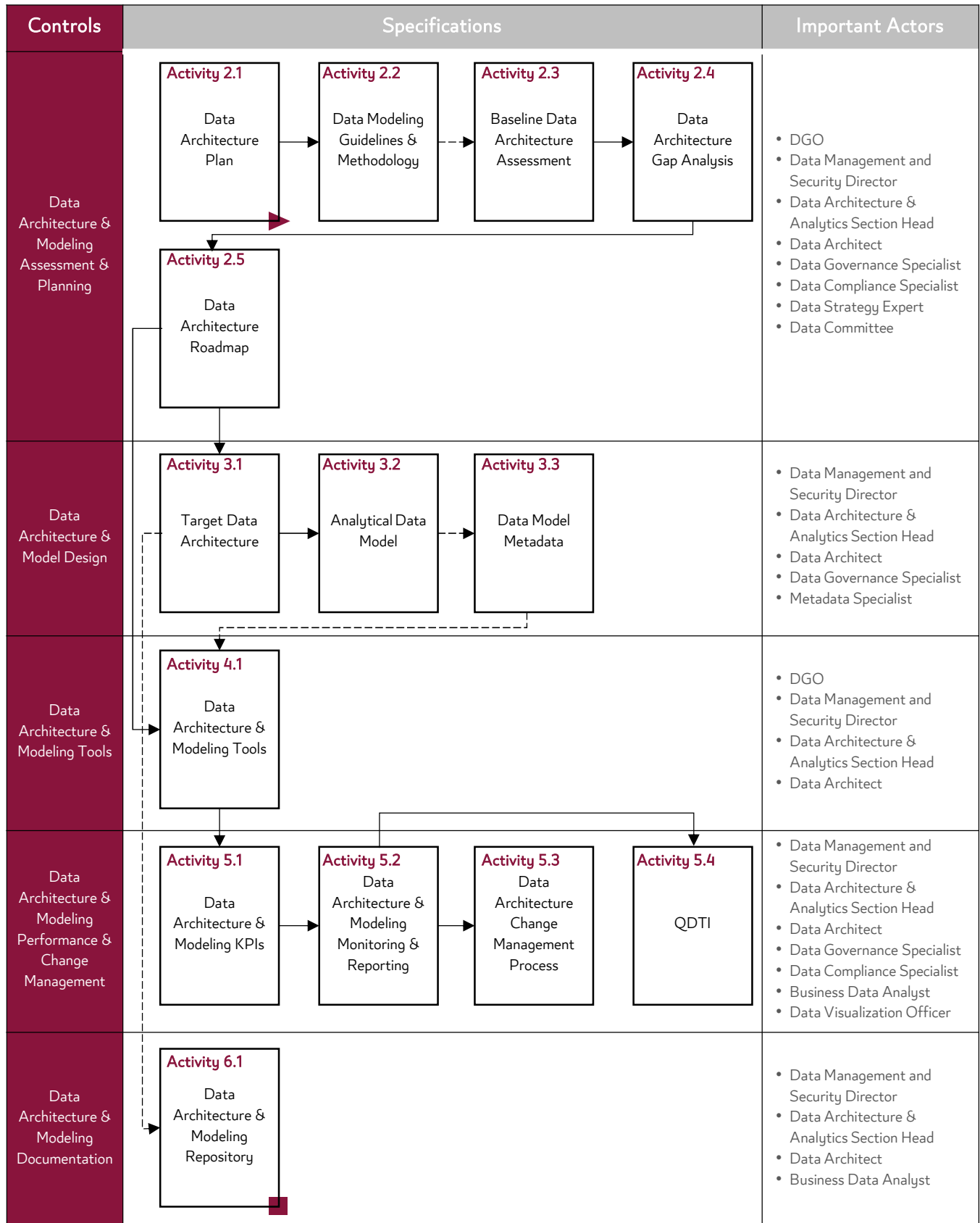


Advanced



Very Advanced

1.8 Domain Level Process Map



2. Data Architecture & Modeling Assessment & Planning



2. Data Architecture & Modeling Assessment & Planning

What is Data Architecture & Modeling Assessment & Planning and why does the Entity need it?

The Data Architecture & Modeling Assessment & Planning Control is essential for establishing a structured, scalable, and governed data environment. A well-defined data architecture provides a framework for managing data assets, ensuring alignment with national standards, business objectives, and interoperability requirements.

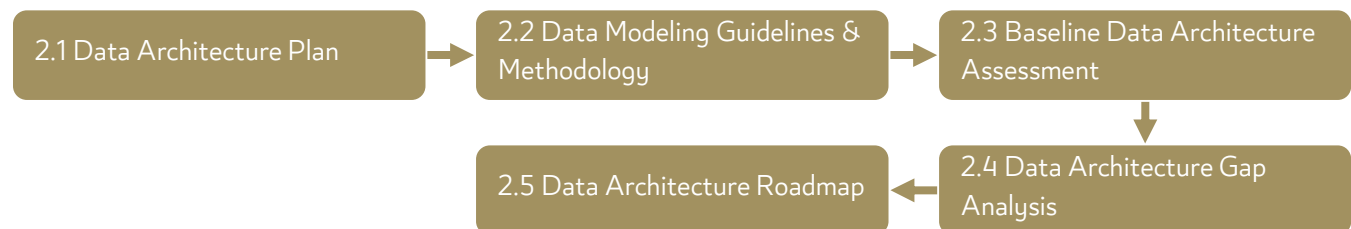
This control guides the Entity in developing a Data Architecture Plan, which establishes the scope, domains, governance roles, and stakeholder responsibilities. It also includes initiating a kick-off meeting to align expectations and introduce the upcoming baseline assessment and gap analysis.

Entities are required to adopt NPC's guidelines by developing their own Data Modeling Guidelines & Methodology, tailored to internal needs while ensuring alignment with the mandated Dimensional Star model and modeling best practices across conceptual, logical, and physical level.

The Baseline Data Architecture Assessment documents current systems, platforms, models, and data flows. This is followed by a Gap Analysis that identifies misalignments with the Entity's Data Strategy, NPC standards, and best practices, across both architecture and analytical data models.

Finally, the Data Architecture Roadmap outlines initiatives, timelines, and owners to address identified gaps. It prioritizes both short-term and strategic initiatives to strengthen governance, enhance data interoperability, and support analytics transformation

This section details the activities of the following process, guiding the Entity in executing the Data Architecture & Modeling Assessment & Planning:



Guiding Principles for Data Architecture & Modeling Assessment & Planning

- 1. Align Architecture with Business Objectives:** Ensure that data architecture plans align with the Entity's business objectives, enabling data structures to support analytical requirements.
- 2. Standardize Data Modeling Practices:** Define consistent standards and methodologies for data modeling across conceptual, logical, and physical layers to ensure uniformity, quality, and ease of integration.
- 3. Identify and Document Gaps:** Conduct a thorough gap analysis to identify areas where current capabilities fall short against best practices and the data architecture & modeling guidelines, creating a clear understanding of improvement needs.
- 4. Create a Clear Roadmap:** Develop a detailed roadmap with defined initiatives, key activities, milestones, timelines, technology requirements, and expected benefits to guide the design and implementation of the future-state data architecture.
- 5. Ensure Comprehensive Documentation:** Maintain detailed documentation of data architecture plans, models, and guidelines to support consistency, compliance, and future reference.

2.1 Data Architecture Plan

The Data Architecture Plan consists of two components: the maturity assessment and the structured planning activities to initiate the broader Data Architecture and Modeling effort.

2.1.1 Data Architecture and Modeling Maturity Assessment

To build an effective roadmap, the Entity must first assess its current state. This activity guides the Entity in retrieving or refreshing its maturity assessment for the Data Architecture and Modeling domain using the Data Management Current Maturity Assessment Framework and Data Management Maturity Model. The assessment identifies current strengths, gaps, and improvement priorities across organization, processes & policies, and tools, and serves as the foundation for roadmap planning.

Process – Data Architecture and Modeling Maturity Assessment

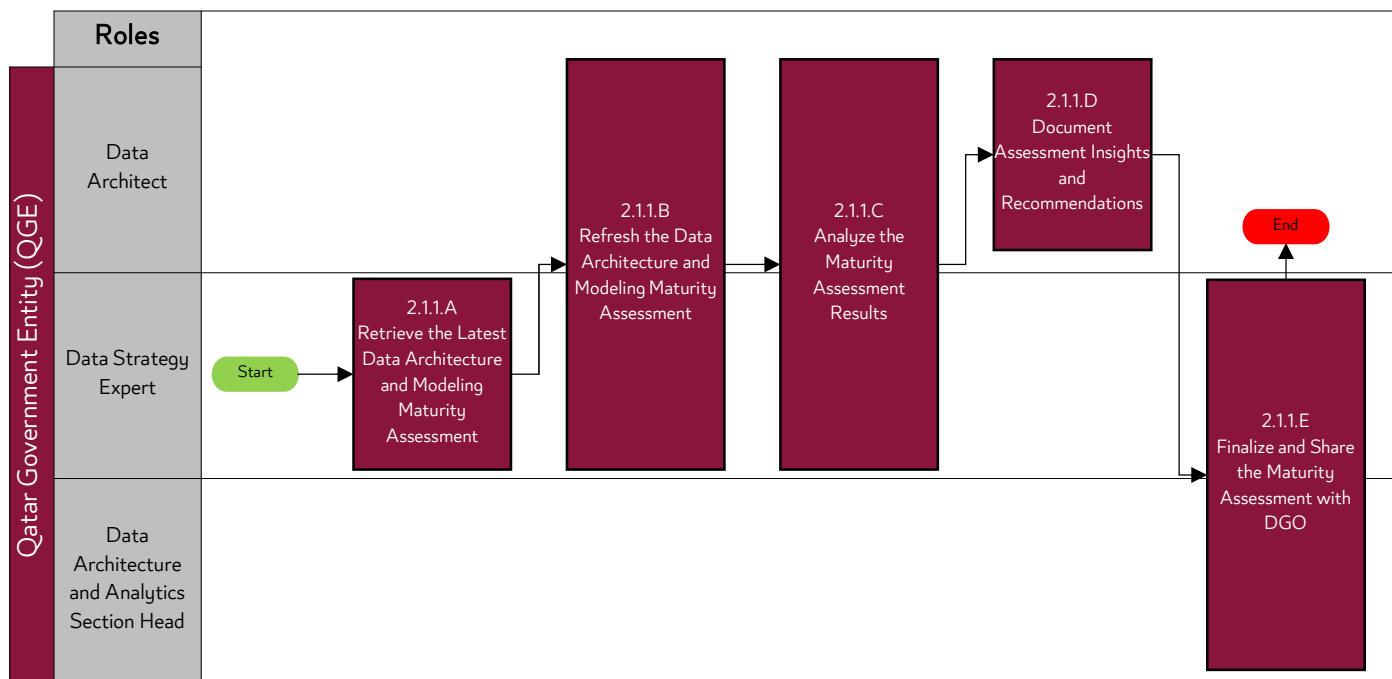


Figure 1: Data Architecture and Modeling Maturity Assessment Process Map

Legend: Process Step Decision Process Start Process End Document Off-Page Reference

2.1.1.A
(1/2)

Retrieve the Latest Data Architecture and Modeling Maturity Assessment. The Entity must retrieve its most recent maturity assessment to establish a baseline understanding of current capabilities across the Data Architecture and Modeling domain. This step is essential for determining whether an update or reassessment is required and provides the foundation for roadmap planning.

- i. **Request Existing Assessment Report:** The Data Architect collaborates with the Data Strategy Expert to locate and retrieve the latest data architecture and modeling maturity assessment.
 - Engage the Data Governance Specialist to obtain the most recent maturity assessment report.
 - Confirm that the report includes maturity ratings across all Data Architecture and Modeling dimensions.
 - Validate that the report was formally conducted and documented using the approved Data Management Current Maturity Assessment Framework.

2.1.1.A
Cont.
(2/2)

- ii. **Validate the Currency of the Assessment:** The Data Architect and Data Strategy Expert jointly evaluate whether the existing assessment is still valid and representative of current capabilities.
- Confirm that the assessment was completed within the past 12 months.
 - Assess whether any significant organizational, tooling, or process changes have occurred since the report.
 - If the report is outdated or misaligned with the Entity's current state, outline that a refresh will be required.
 - Note any open or unaddressed recommendations that may still be relevant and require integration into the new roadmap.



Key Message / Information

Ensure that the maturity assessment covers both business and technical requirements. If the assessment is fragmented or was conducted informally, treat it as insufficient and proceed to a formal refresh.

A current and validated maturity assessment is essential to building a grounded and targeted improvement roadmap. Without it, any planning activity risks being disconnected from the entity's operational reality and compliance posture.

2.1.1.B
(1/2)

Refresh the Data Architecture and Modeling Maturity Assessment (If Required): If the Entity's existing Data Architecture and Modeling maturity assessment is outdated, incomplete, or unavailable, a full reassessment must be conducted using the **Data Management Current Maturity Assessment Framework** and the supporting **Data Management Maturity Model** (referenced in Activity 2.2 within the Data Management Strategy and Governance domain guidelines). This ensures that the planning efforts in the roadmap are grounded in an accurate and current view of the Entity's data architecture and modeling capabilities.

- i. **Initiate the Refresh Process:** The Data Architecture and Analytics Section Head triggers the reassessment and mobilizes key stakeholders.
- Formally launch the maturity refresh process by assigning ownership to the Data Architect.
 - Confirm participation of relevant stakeholders including Data Stewards, Data Owners, IT, and business users.
 - Communicate the scope and purpose of the reassessment to all contributors.
- ii. **Apply the Data Management Current Maturity Assessment Framework:** The Entity must use the **Data Management Maturity Model** to evaluate Data Architecture and Modeling current-state capabilities.
- Review and respond to each of the assessment dimensions (Organization, Architecture Plan, Architecture Framework, etc.) using the 1–5 scale.
 - Ensure contributors refer to actual practices, tools, and governance documentation, and not perceived maturity.
 - Encourage team discussions and workshops to calibrate responses and promote honest evaluation.

2.1.1.B
Cont.
(2/2)

- iii. **Consolidate Input and Finalize Scores.** Leveraging the **Data Management Current Maturity Assessment Framework**, the Data Architect consolidates all responses, supporting evidence, and justifications.

- Validate consistency across responses and resolve conflicting input where necessary.
- Assign a final maturity level (1–5) for each dimension, including rationale.
- Prepare or update the maturity heatmap for the Data Architecture and Modeling domain, and document relevant observations.



Key Message / Information

Encourage transparency and use collaborative working sessions to resolve subjectivity and misalignment between technical and business contributors.

2.1.1.C
(1/2)

Analyze the Maturity Assessment Results: Once the Data Architecture and Modeling maturity assessment has been completed or refreshed, the Entity must analyze the results to understand current strengths, weaknesses, and capability gaps. This analysis forms the foundation for identifying high-priority improvement areas and preparing a focused roadmap.

- i. **Review Maturity Scores Across the Data Architecture and Modeling Dimensions:** The Data Architect and Data Strategy Expert jointly evaluate the final maturity scores to identify trends and performance patterns.
- Review assigned maturity levels for all Data Architecture and Modeling dimensions using the 1–5 scale.
 - Identify areas of high maturity that reflect institutionalized and standardized practices.
 - Identify low-performing areas (e.g., Organization, Architecture Plan, Architecture Standards) that could hinder effective governance or interoperability.
- ii. **Conduct Thematic Gap Analysis:** Group maturity results into logical clusters (e.g., Organization, processes & policies, tools) to derive broader insights.
- Categorize assessment results by functional theme:
 - Governance and ownership
 - Data Architecture
 - Data Modeling
 - Identify cross-cutting gaps that appear across multiple dimensions (e.g., lack of clearly defined roles and ownership affecting the data architecture plan and how often it is updated).
 - Highlight dependencies. For example, poor architecture planning (low score in “Architecture Plan”) may contribute to poor implementation of architecture standards (low score in “Architecture Standards”).
- iii. **Identify Quick Wins and Structural Gaps:** Segment findings based on the level of effort and impact to prioritize next steps.

2.1.1.C
Cont.
(2/2)

- **Quick wins:** improvements that can be implemented with minimal effort (e.g., assigning ownership, developing an architecture plan).
- **Structural gaps:** issues that require strategic investment or change management (e.g., tool implementation, entity-wide policy enforcement).
- Recommend areas that should be prioritized in the roadmap due to compliance implications or high strategic value (e.g., defining a process for managing data models and artifacts).



Key Message / Information

This analysis step is critical in converting maturity data into actionable insight. The quality of the roadmap depends on how well the Entity understands the underlying causes and impacts of its current maturity gaps.

2.1.1.D
(1/2)

Document Assessment Insights and Recommendations: The Entity must consolidate the results of the Data Architecture and Modeling maturity assessment into a structured executive-level slide. This summary is intended to present a high-level overview of the current maturity state, key findings, and prioritized opportunities for improvement. It will serve as a key input for roadmap planning and internal/external alignment.

- i. **Populate the Data Maturity Summary Slide:** The Data Architect prepares a summary slide (as referenced in the Data Management Current Maturity Assessment Framework in Activity 2.2 within the Data Management Strategy and Governance domain guidelines) that captures the summary maturity of the Data Architecture and Modeling domain.

- a. **Maturity Score (Gauge):**

- Calculate the overall average maturity score across all Data Architecture and Modeling dimensions.
- Use the 1–5 scale and visually represent the result in the dial/gauge format.
- Select the appropriate maturity level label (Initial, Limited, Evolving, etc.).

- b. **Organization, Processes & Policies, and Tools Ratings:**

- Assign indicative scores across Organization, Processes & Policies, and Tools based on maturity dimension grouping:
 - Organization: e.g., Organization
 - Processes & Policies: e.g., Architecture Plan, Architecture Framework
 - Tools: e.g., Enterprise Data Modeling Tool

- c. **Key Insights & Findings:**

- Summarize major pain points or maturity gaps discovered.
- Highlight inconsistencies in governance, lack of tooling, low coverage, or siloed practices.
- Use brief, high-impact bullet points appropriate for leadership-level review.

- ii. **Identify Key Improvement Opportunities:** This section translates findings into action-oriented strategic recommendations.

2.1.1.D
Cont.
(2/2)

- Define the (e.g., 2-4) highest-value opportunities for improvement.
- Ensure these align with major maturity gaps (e.g., introducing ownership, adopting a tool, formalizing change control).
- Frame opportunities in language suited for executive endorsement and roadmap alignment.

iii. **Validate with Key Stakeholders:** Before finalization, the summary slide must be reviewed by the Data Architecture and Analytics Section Head and Data Strategy Expert.

- Circulate the draft slide for review and feedback.
- Confirm that insights and opportunities are clearly and accurately captured.
- Refine language and formatting to ensure clarity, precision, and alignment with the rest of the Data Management Current Maturity Assessment Framework reporting pack.



Key Message / Information

Use concise, action-oriented phrasing; each bullet point should communicate a real insight or decision driver. Avoid vague recommendations and tie each suggestion to a specific weakness and pathway to resolution.

2.1.1.E

Finalize and Share the Maturity Assessment with DGO: If the Data Architecture and Modeling maturity assessment was refreshed, then the Entity must consolidate the findings and formally share the updated assessment (as part of the broader data management maturity assessment) with the DGO for alignment.

i. **Consolidate Data Architecture Final Materials for Submission:** The Data Strategy Expert prepares the formal assessment package for DGO review.

- Data Architecture and Modeling domain level maturity scores
- Data Architecture and Modeling maturity heatmap
- Data Architecture and Modeling assessment insights and recommendations

ii. **Submit Assessment Package to the DGO:** The Data Architecture and Analytics Section Head oversees submission and initiates engagement with the DGO.

- Send the materials to the official DGO point-of-contact through the designated submission channel.
- Request a formal review and optionally ask for advisory feedback or validation of scoring.

iii. **Incorporate DGO Feedback and Finalize Record:** Once the DGO has reviewed the submission, incorporate any clarifications or adjustments required.

- Address DGO comments or requested edits in the assessment summary slide or supporting documents.
- Finalize all materials and archive the submission internally.
- Ensure the validated version is reflected in the Entity's central data governance records.

RACI Matrix – Data Architecture and Modeling Maturity Assessment

Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Strategy Expert	Data Governance Specialist	Data Architect	Data Owner
2.1.1.A	Retrieve the Latest Data Architecture and Modeling Maturity Assessment		A	I	R	C	C	I
2.1.1.B	Refresh the Data Architecture and Modeling Maturity Assessment (If Required)		A	C	R	C	R	C
2.1.1.C	Analyze the Maturity Assessment Results		A	C	R	C	R	C
2.1.1.D	Document Assessment Insights and Recommendations		A	C	C		R	
2.1.1.E	Finalize and Share the Maturity Assessment with DGO	I/C	A	R	R	C	C	

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Refresh the Data Management Maturity Assessment



Analyze the Maturity Assessment Results

2.1.2 Data Architecture Plan

This activity guides the Entity in initiating and structuring its Data Architecture Plan through a series of foundational steps. These include aligning architecture objectives with business and regulatory priorities, defining scope and architectural domains, identifying key stakeholders and responsibilities, and organizing a structured kick-off meeting. Together, these steps ensure clarity, accountability, and alignment across business and technical teams, while establishing a shared understanding of current and future architecture needs. This sets the stage for conducting assessments, identifying gaps, and formulating a practical roadmap aligned with national data standards and enterprise data strategy.

2.1.2.A

Establish Architecture Objectives and Business Alignment: Define the objectives of the data architecture assessment and planning activities to ensure alignment with the entity's business and data strategy, regulatory requirements, and national data standards. Key activities include:

- **Business Needs Identification:** Engage with stakeholders (e.g., Data Management and Governance Section Head, Data Architecture and Analytics Section Head) to define how data architecture supports strategic and operational goals.
- **Regulatory Compliance Assessment:** Document the data governance, security, and interoperability standards that the data architecture must adhere to, including relevant national standards and international best practices.
- **Governance and Decision-Making Framework:** Establish governance mechanisms, including review cycles, approval processes, while identifying key roles involved in data architecture decisions.

2.1.2.B

Define Scope and Architectural Domains: Define the scope, coverage, and key components of the Data Architecture Plan to ensure clarity in boundaries and areas of focus.

- **Architectural Domains Identification:** Define core components, including but not limited to Data Models, Integration & Interoperability, Data Storage, Metadata Management, and Security.
- **Scope and Boundaries Definition:** Specify whether the Data Architecture applies to entity-wide systems, specific departments, or cross-functional initiatives.

2.1.2.C

Identify Key Stakeholders and Responsibilities: Building on the key roles identified in 2.1.A, identify any additional necessary roles and the corresponding stakeholders required in the assessment and planning phase. Clearly define their responsibilities to ensure accountability for the Data Architecture Plan. Key roles include but are not limited to:

- **Source System Owners:** Provide insights on existing systems and data models.
- **Data Stewards:** Ensure data integrity, governance, and compliance with policies.
- **IT:** Define technical feasibility.
- **Business Leaders:** Align architecture objectives with organizational strategy and analytics needs.

2.1.2.D

Conduct Data Architecture Kick-Off Meeting: Organize and facilitate a structured data architecture kick-off meeting to align stakeholders, set expectations, and review the key components of the Data Architecture Assessment and Planning activities. This meeting will serve as the foundation for architecture development, covering scope, business alignment, and stakeholder roles while introducing the approach for the Data Architecture Baseline Assessment and Gap Analysis. Key activities include:

Meeting Preparation:

- Define the agenda covering architecture objectives, scope, stakeholders, assessment methodology.
- Identify and invite key participants (e.g., Source System Owners, Data Stewards, IT, Business Leaders, and Governance Representatives).
- Prepare and circulate relevant pre-reading materials, such as existing architecture documentation (if exists), regulatory requirements, and the national data standards.

Key Discussion Areas:

- **Data Architecture Objectives:** Define the purpose and strategic goals of the data architecture, ensuring alignment with the entity's business and data strategy, regulatory requirements, and governance principles.
- **Preliminary Business and Technology Considerations:** Identify preliminary business drivers, technology enablers, and high-level interoperability needs that the business envisions. While this section does not define the target architecture, it helps establish a preliminary understanding of strategic considerations that should be kept in mind during the gap analysis.
- **Scope and Architectural Domains:** Define the coverage of the architecture, including but not limited to Data Models, Integration, Metadata Management, and Security.
- **Stakeholder Responsibilities:** Clarify roles and responsibilities to ensure accountability in architecture development and decision-making.
- **Baseline Assessment and Gap Analysis:** Discuss the scope and expected outcomes of the baseline assessment and gap analysis.
- **Next Steps & Roadmap:** Outline the subsequent activities (2.2–2.5) and assign action items for data modeling guidelines & methodology development, baseline assessment and gap analysis, and roadmap development.

RACI Matrix – Data Architecture Plan

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Stewards	Business SMEs
2.1.2.A	Establish Architecture Objectives and Business Alignment	I	C/A	R	C		
2.1.2.B	Define Scope and Architectural Domains	I	C/A	R			
2.1.2.C	Identify Key Stakeholders and Responsibilities	C/I	C/A	R	C	C	C
2.1.2.D	Conduct Data Architecture Kick-Off Meeting	I	C/A	R	I	I	I

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Outline the High-level Scope for DAM Assessment & Planning



Identify Key Stakeholders for Architecture Design

2.2 Data Modeling Guidelines & Methodology

NPC has developed the Data Architecture & Modeling Handbook, which serves as a foundational asset for government Entities. This handbook provides standardized guidelines and best practices for data architecture and modeling, ensuring consistency across conceptual, logical, and physical models while aligning with national data standards and data architecture principles.

Each government entity must leverage this handbook to develop its own data modeling guidelines and methodology, tailoring them to specific Entity requirements, while maintaining compliance with NPC's national data standards.

This activity outlines the process for entities to establish their own data modeling guidelines using the NPC-provided asset.

2.2.A
(1/4)

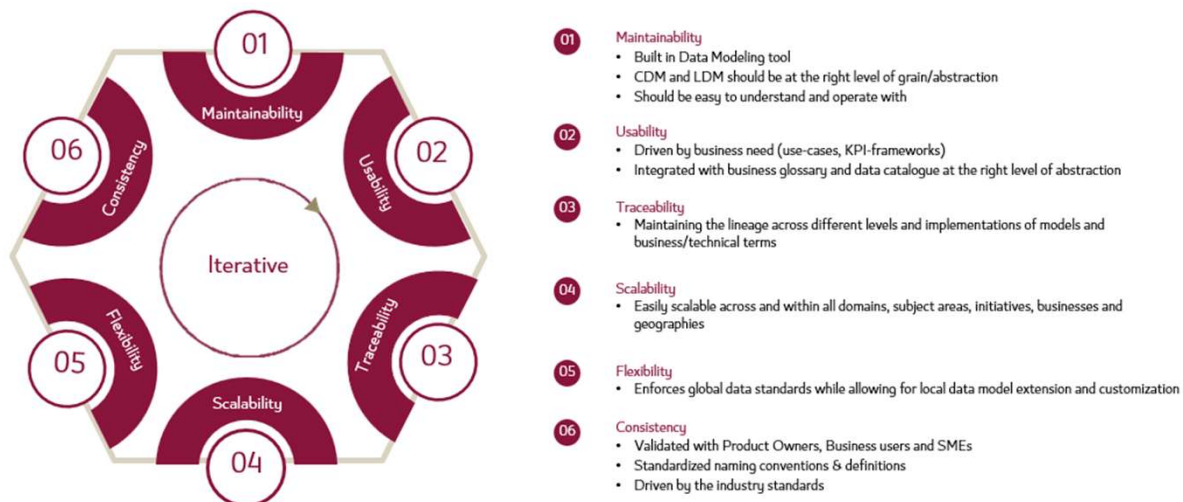
Obtain and Review NPC's Data Modeling Guidelines: The entity must first obtain the Data Architecture & Modeling Handbook provided by the NPC and conduct a detailed review to understand its key components. The review should focus on:

- i. **Data Modeling Design Principles:** Review the key design principles that ensure consistency, scalability, and usability of data models. These principles include Maintainability, Usability, Consistency, Traceability, Flexibility, and Scalability, all of which support standardized, structured, and high-quality data modeling practices. Entities must integrate these principles into their modeling approach to align with business and operational requirements.

Asset Guidance

The Entity can leverage the **Data Architecture & Data Modeling Handbook** to develop its own data modeling guidelines and methodology, tailored to its specific business and operational needs while ensuring compliance with NPC's National Data Standards. All assets referenced in this section are included within the handbook.

Illustrative



2.2.A
Cont.
(2/4)

- ii. **Data Modeling Methods Overview:** Review the data modeling methods overview, which establishes the Dimensional Star model as the mandated standard. While other approaches such as Relational, Object-Oriented, Fact-Based, Time-Based, and NoSQL are referenced, the guidelines focus on the Dimensional Star model as the required approach for ensuring analytical efficiency and compliance with NPC's national data standards. The purpose of this review is to ensure that entities develop their data models in alignment with the Dimensional Star model.

Asset Guidance

Illustrative

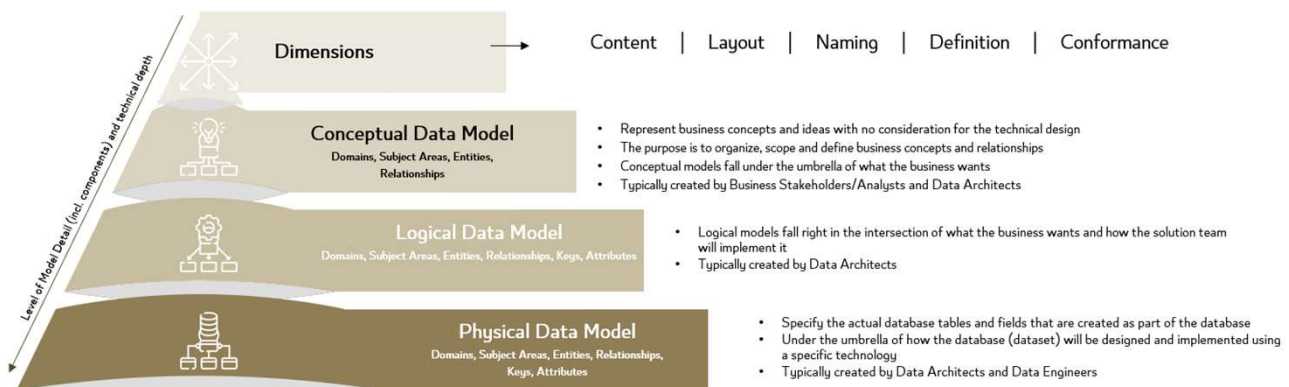
	Relational	Dimensional Star	Object-Oriented	Fact-Based	Time-Based	NoSQL
	Schema where the data is organized in multiple tables with relationships defined between them using keys	Schema where data is organized through a fact table in the middle with dimension tables surrounding it in a star-like shape	Schema where entities are represented as objects which are connected to other objects via relationships	Schema where data is organized around business processes rather than entities	Schema where the data is organized based on time, thus allowing information to be retrieved based on a specific date & time	NoSQL does not use a relational data modeling technique, and it emphasizes flexible design
Pros	• Suited for relational database management systems • Easy to articulate business requirements • Minimal data redundancy	• Suited for both relational and multidimensional database management systems • Optimized for BI and analytics use cases • Easy to form aggregates on different level of granularity	• Suited for relational database management systems • Widely used in Big Data implementations • Low number of objects to manage	• Suited for relational database management systems • Simplified data analysis	• Highly flexible technique • Allows for highly parallel data load • Allows fast retrieval of data based on time	• Allows flexibility in the design • Can handle large volumes of data
Cons	• PDM not suited for multidimensional database management systems • Not optimized for BI and analytics consumption • Low flexibility	• Hard to implement hierarchies and many2many relationships Current Approach for domains	• Not suited for multidimensional database systems • Requires high qualification and effort to implement • High data redundancy	• PDM not suited for multidimensional database management systems • Not suited to resolve multiple types of use cases • Limited flexibility • Data duplication can occur	• Not suited for multidimensional database systems; CDM & LDM not suited for relational database management system • Requires strong principle to build and follow	• Not suited for relational and multidimensional database systems • Data Model horizontally scalable only

- iii. **Data Model Framework:** Review the layers of data models.

- **Conceptual Data Model (CDM):** High-level business concepts.
- **Logical Data Model (LDM):** Aligns with business requirements.
- **Physical Data Model (PDM):** Defines database structures.

Asset Guidance

Illustrative



2.2.A
Cont.
(3/4)

- iv. **Data Model Components:** Review the essential building blocks of data models, including Domains, Subject Areas, Entities, Relationships, Keys, and Attributes. Each component plays a critical role in structuring and defining data relationships, ensuring consistency and standardization across conceptual, logical, and physical models. Entities must establish clear definitions and associations to support accurate data representation and interoperability.

**Asset Guidance***Illustrative*

COMPONENT	DEFINITION	EXAMPLE	CDM	LDM	PDM
Domain	• Sector's data model level, purpose, content & owner	human capital data model	✓	✓	✓
Subject Area	• Subject area of a domain, also referred as a 'sub-domain'	employee subject area	✓	✓	✓
Entity	• Entity about which information is to be collected, also referred as a 'table'	employee	✓	✓	✓
Relationship	• Type of association between two entities or between one and many others	one-to-many	✓	✓	✓
Logical Key	• Unique identifier of an entity, called a primary key/surrogate or foreign key, like the identification number of an employee	pk employee id		✓	
Attribute with logical data type	• One of the characteristics of an entity, also referred as a 'field' and its associated logical data type such as 'decimal'	total price (decimal)		✓	
Technical Key	• Like a logical key, a unique identifier of an entity written in technical terms as it would appear in a database	pk_employee_id			✓
Attribute with metadata	• One of the characteristics of an entity, also referred as a 'field' and its associated metadata such as 'decimal' as a data type and '10,2' as the precision, among other metadata information	total_price, DECIMAL (10, 2)			✓

- v. **Standardization Guidelines (Data Model Dimensions):** Review the standardization dimensions that guide the structure and consistency of data models. These dimensions - Content, Layout, Naming, Definition, and Conformance - ensure alignment across conceptual, logical, and physical data models. By adhering to these guidelines, entities can promote uniformity in data modeling practices, facilitating seamless integration and interoperability across systems.

**Asset Guidance***Illustrative*

	Conceptual	Logical	Physical
Content	Content of the data model including number & type of data components such as datasets, attributes, relationships.		
Layout	Layout of the data model as well as the structure and order of the components that make it.		
Naming	Naming convention of the models, subject areas, datasets, attributes, relationships and keys at any model level.		
Definition	Definition standards of models, subject areas, datasets, and attributes for any type of model.		
Conformance	Guidelines for ensuring models of all levels are conformed across and within domains.		

Dimensions

2.2.A
Cont.
(4/4)

- vi. **Data Model Dimensions Applicable to Data Model Components:** Review the mapping of data model levels, data model components, and dimensions to ensure the development of standardized and compliant data models. Each data model - Conceptual, Logical, and Physical - must adhere to key dimensions, including Content, Layout, Naming, Definition, and Conformance, to maintain consistency and interoperability across data management practices.

Asset Guidance *Illustrative*

		Dimensions				
		Content	Layout	Naming	Definition	Conformance
Data Model Components	Domain	What should a domain data model include? ✓ ✓ ✓	How should a domain data model look like? ✓ ✓ ✓	How should a domain data model be called? ✓ ✓ ✓	How should a domain data model be defined? ✓ ✓ ✓	How should a domain data model be to align with others? ✓ ✓ ✓
	Subject Areas	What should a subject area include? ✓ ✓ ✓	How should a subject area look like? ✓ ✓ ✓	How should a subject area be called? ✓ ✓ ✓	How should a subject area be defined? ✓ ✓ ✓	How should a subject area be to align with others? ✓ ✓ ✓
	Datasets	What should a dataset include? ✓ ✓ ✓	How should a dataset look like? ✓ ✓ ✓	How should an entity be called? ✓ ✓ ✓	How should a dataset be defined? ✓ ✓ ✓	How should a dataset be to align with all the others? ✓ ✓ ✓
	Relationships	What should a relationship include? ✓ ✓ ✓	How should a relationship look like? ✓ ✓ ✓	How should a relationship be called? ✓	N/A	How should a relationship be to align with all the others? ✓ ✓ ✓
	Keys	What should a key include? ✓ ✓	How should a key look like? ✓ ✓	How should a key be called? ✓ ✓	N/A	How should a key be to align with all the others? ✓ ✓
	Attributes	What should an attribute include? ✓ ✓	How should an attribute look like? ✓ ✓ ✓	How should an attribute be called? ✓ ✓	How should an attribute be defined? ✓ ✓	How should an attribute be to align with all the others? ✓ ✓

✓ CDM ✓ LDM ✓ PDM

- vii. **Best Practices & Methodologies:** Review the best practices and methodologies for structuring data models to ensure standardization, consistency, and compliance with NPC's data standards.

Asset Guidance *Illustrative*

Conceptual Data Model Guidelines (1/7)

The guidelines for each of the components' dimensions of the conceptual data model are identified below

1 DOMAIN DATA MODEL

A	Content	- Each domain includes a single primary conceptual data model. - The data model includes a clear, comprehensive, and specific definition. - A title block is present on the data model, detailing the model title, version, author, and creation/modification dates. - The data model encompasses multiple subject areas relevant to a specific domain. - The data model reflects a business-oriented view of datasets and their relationships within that domain. - The data model should not exceed 200 datasets.
B	Layout	- The data model is depicted in a well-organized diagram, using rectangles and dashed lines in a dimensional star schema. - The entire data model is fully visible when viewed at 100% zoom. - The data model is designed to be intuitive and understandable with minimal training required. - Font size in the data model is readable, and drawing objects follow a uniform style.
C	Naming	- The data model follows the CDM naming convention for domain schema names, formatted as '[Entity Initials] [Reference Number] [Version] Conceptual Data Model'. - The name of the data model is singular, with spaces between each word. - Each word in the data model's name begins with a capital letter. - The data model's name is free from numbers and special characters.
D	Definition	- The definition of the data model must comprehensively describe the domain in business terms and give an overview of the data included. - The definition should avoid stating the obvious. - The definition is written in singular form and is free of spelling errors. - The definition complies with standard practices for definitions.



Key Message / Information

The entity should ensure a thorough understanding of these aspects before proceeding to develop its customized guidelines. The Data Architecture and Modeling Handbook has guidelines for conceptual, logical, and physical data models.

2.2.B

Establish Data Modeling Guidelines & Methodology Scope: Before developing entity-specific data modeling guidelines, it is essential to define the objectives, scope, and constraints to ensure alignment with NPC's Data Architecture & Modeling Handbook. These guidelines must support the entity's business, analytical, and regulatory needs while maintaining compliance with national data standards and the mandated Dimensional Star model.

I. **Objectives:** Clearly define the purpose of the guidelines. Example objectives include:

- Ensuring consistent and standardized data modeling practices across the entity.
- Enforcing compliance with NPC's national data standards and the mandated Dimensional Star model.
- Improving data interoperability, integration, and governance within the entity.
- Aligning data models with business processes and analytical requirements.

II. **Scope:** Define the boundaries of the data modeling guidelines. Examples include:

- The types of data models covered (Conceptual, Logical, Physical).
- The applicable data domains (e.g., finance, human resources, operational data etc.).
- The systems and platforms where these models will be applied.
- Any external dependencies, such as integration with national databases or sector-wide data standards.

III. **Constraints:** Identify any limitations that must be considered. Examples include:

- Regulatory and compliance constraints
- Technical constraints (e.g., scalability issues, integration challenges)

2.2.C
(1/3)

Develop Entity-Specific Data Modeling Guidelines & Methodology: After defining the scope, the entity must develop its own data modeling guidelines & methodology based on NPC's Data Architecture & Modeling Handbook.

i. **Tailoring NPC's Guidelines to the Entity's Context:** Each entity must customize the NPC-provided guidelines to fit its specific business, analytical, and technical needs, while ensuring compliance. Key considerations include:

a. **Alignment with business and analytical requirements**

- Ensuring models reflect real-world business processes within the entity.
- Establishing rules for domain-specific data modeling requirements.

b. **Compliance with NPC's National Data Standards**

- Adopting the mandated Dimensional Star model across all data modeling efforts.
- Ensuring that all extensions or customizations remain within NPC compliance.

- ii. **Establishing Entity-Specific Data Modeling Standards:** To maintain consistency, quality, and interoperability, entities must define clear modeling standards within their guidelines, covering:
 - a. **Data Model Levels & Structure:** Entities must ensure a structured approach to data modeling by defining how each model level is applied.
 - **Conceptual Data Model:** Establishes high-level business concepts, including domains, subject areas, and entity relationships, without technical details.
 - **Logical Data Model:** Defines detailed attributes, relationships, and business rules while maintaining a technology-agnostic approach.
 - **Physical Data Model:** Governs database structures, including data types, technical keys, indexing, and storage considerations for implementation.
 - b. **Data Model Components & Governance Rules:** Entities must define how data model components are structured, ensuring alignment with the national data standards.
 - **Domains & Subject Areas:** Categorization of data based on business functions and processes.
 - **Entities & Relationships:** Standardized definitions and associations to maintain referential integrity.
 - **Keys & Constraints:** Clear conventions for primary keys, foreign keys, and surrogate keys to ensure data integrity.
 - **Data Type & Format Rules:** Standardized formats for numerical, textual, and temporal data to promote consistency.
 - c. **Documentation & Metadata Standards:** To ensure models are maintainable, traceable, and well-documented, entities must define the below:
 - **Naming Conventions:** Consistent naming structure for entities, attributes, relationships, and keys.
 - **Metadata Structure:** Required metadata elements such as definitions, data types, constraints, and lineage information.
 - **Version Control & Change Management:** Rules for model versioning, updates, and governance processes to maintain long-term consistency.
- iii. **Ensuring Compliance & Interoperability:** Entities must structure their guidelines in alignment with the below key considerations
 - Future interoperability with national and sector-wide data models for seamless data integration.
 - Consistency across internal systems to facilitate data-sharing and governance.
- iv. **Documentation & Repository Management:** Entities must create a formalized repository where the guidelines will be stored, maintained, and referenced:
 - A version-controlled document to track updates and refinements.
 - A repository for easy access and governance of modeling guidelines.
 - Defined roles and responsibilities for maintaining and enforcing standards.

2.2.C
Cont.
(3/3)**Tip**

Research and incorporate relevant international data modeling standards (such as ISO or DAMA-DMBOK) to ensure global alignment. Regularly consult industry publications to stay updated on best practices and periodically benchmark the Entity's modeling guidelines against these standards. This approach helps maintain high standards and encourages interoperability across global systems.

2.2.D
(1/2)

Review, Validate, and Finalize Data Modeling Guidelines: Once the entity-specific data modeling guidelines have been developed, they must undergo internal review, validation, and finalization to ensure compliance with NPC's national data standards and alignment with the entity's business, operational, and technical requirements. This process ensures that the guidelines are accurate, comprehensive, and effectively govern data modeling practices across the entity.

- i. **Conduct Internal Review of Data Modeling Guidelines:** Entities must conduct a structured review process involving key stakeholders, data architects, and governance teams to assess the quality, completeness, and alignment of the guidelines.

- a. **Stakeholders Involved in the Review Process:**

- **Data Committee:** Ensures compliance with the national data standards.
- **Enterprise Data Architects:** Validate alignment with data architecture and interoperability requirements.
- **Business Domain Experts:** Confirm that guidelines meet business and operational data needs.
- **IT and System Owners:** Assess the feasibility of implementation within existing and planned infrastructure.

- b. **Key Focus Areas for Review:**

- Adherence to NPC's mandated Dimensional Star model and national data standards.
- Consistency across Conceptual, Logical, and Physical models.
- Completeness and clarity of modeling components, including domains, subject areas, entities, relationships, attributes, and keys.
- Alignment with data governance and interoperability policies.

- ii. **Validate Alignment with National Data Standards:** Once the initial review is complete, entities must validate that their customized guidelines:

- Follow NPC's prescribed data modeling methodology without deviation.
- Ensure standardized documentation, metadata definitions, and versioning mechanisms are in place.
- Promote consistency across business units, operational functions, and integrated systems.
- Address interoperability requirements for integration with national or sector-wide data standards.

2.2.D
Cont.
(2/2)

- iii. **Incorporate Feedback and Refinements:** Based on stakeholder reviews and validation assessments, necessary refinements must be made before finalization.

- Gather and document stakeholder feedback systematically.
- Ensure all required corrections, clarifications, and enhancements are implemented.
- Validate updates with key reviewers before proceeding to the approval phase.
- Prepare a final draft version of the guidelines for official sign-off.

2.2.E
(1/2)

Approval, Publishing & Version Control of Data Modeling Guidelines: Once the data modeling guidelines have been reviewed, validated, and refined, they must be formally approved, published, and version-controlled to ensure structured adoption, accessibility, and ongoing maintenance. This step ensures that the guidelines are recognized as the official standard for data modeling across the entity and are kept up to date with evolving business and regulatory requirements.

- i. **Formal Approval and Endorsement:** To finalize the guidelines, entities must implement a structured approval process involving key governance stakeholders.

a. **Stakeholders Involved in Approval:**

- **Data Committee:** Provides final oversight and approval.
- **Executive Sponsors or Senior Management:** Endorse the guidelines for entity-wide adoption.

b. **Approval Process Steps:**

- Submit the final version of the guidelines to the Data Committee for formal review.
- Ensure compliance validation against NPC's mandated modeling standards.
- Obtain official sign-off from designated decision-makers.
- Communicate the approved status to all relevant teams.

- ii. **Publishing & Distribution of Guidelines:** Once approved, the guidelines must be published and made accessible across the entity to ensure adoption and adherence.

a. **Publishing Format:**

- Store in a centralized, version-controlled repository for accessibility.
- Provide structured documentation that is easy to navigate and apply.
- Ensure availability in both digital and printable formats.

b. **Distribution Strategy:**

- Notify all relevant teams, including data modelers, architects, governance teams, and business units.
- Integrate guidelines into data governance frameworks, system development lifecycles, and training materials.
- Conduct training or awareness sessions to ensure proper understanding and application.

2.2.E
Cont.
(2/2)

iii. **Implement Version Control & Change Management:** To ensure long-term effectiveness, entities must implement version control and change management processes for maintaining the guidelines.

a. **Version Control Mechanism:**

- Assign version numbers to track updates (e.g., v1.0, v1.1, v2.0).
- Maintain a change log documenting revisions, approvals, and effective dates.
- Ensure archiving of previous versions for historical reference.

b. **Review and Update Cycle:**

- Establish a periodic review cycle (e.g., annually or bi-annually).
- Define a governance process for requesting and approving changes.
- Ensure updates align with NPC's national data standards and evolving regulatory requirements.

2.2.F

Inform DGO: Once the Data Architecture & Data Modeling Handbook has been published in the centralized repository, inform the National Data Governance Office (DGO) of its completion and provide a copy for their review.

2.2.G
(1/2)

Governance and Compliance Mechanisms for Model Validation and Updates: To ensure long-term adherence to NPC's data modeling standards, entities must establish governance and compliance mechanisms for validating, maintaining, and updating their data modeling guidelines. This step defines roles, responsibilities, validation processes, compliance checkpoints, and change management procedures to keep data models aligned with national standards and evolving business needs.

i. **Establish Data Model Governance Processes:** Entities must define a governance structure to oversee the adoption, enforcement, and continuous improvement of data modeling standards.

a. **Governance Roles & Responsibilities:**

- **Data Governance Committee:** Provides oversight and ensures compliance with NPC's standards.
- **Enterprise Data Architects:** Monitor technical integrity and alignment with architectural frameworks.
- **Data Stewards & Model Owners:** Ensure day-to-day governance of modeling activities.
- **Business Stakeholders:** Validate that data models remain relevant to operational needs.

b. **Governance Responsibilities:**

- Enforce adherence to approved modeling standards.
- Resolve discrepancies or conflicts in data model implementation.
- Conduct periodic audits to assess model accuracy and compliance.
- Oversee the model validation, compliance, and update processes.

- ii. **Define the Model Validation Process:** To maintain accuracy, and business relevance, all data models must undergo a structured validation process before implementation.
 - a. **Validation Criteria:**
 - Ensures alignment with NPC's mandated Dimensional Star model.
 - Confirms consistency across Conceptual, Logical, and Physical models.
 - Verifies that business rules, constraints, and relationships are correctly implemented.
 - Assesses data integrity, referential integrity, and metadata accuracy.
 - b. **Validation Workflow:**
 - **Model Submission:** Data architects submit models for validation.
 - **Automated & Manual Checks:** Review for technical accuracy, compliance, and governance adherence.
 - **Stakeholder Review:** Business and governance teams confirm functional alignment.
 - **Approval & Certification:** Models are approved, logged, and published in a version-controlled repository.
- iii. **Implement Compliance Checkpoints in the Data Architecture & Modeling Lifecycle:** Entities must integrate compliance checkpoints to ensure that data models are validated at key stages of their lifecycle.
 - a. **Key Compliance Checkpoints:**
 - **During Model Development:** Ensure modeling standards are applied.
 - **Before Deployment:** Conduct pre-implementation validation to confirm adherence.
 - **Ongoing Monitoring:** Implement data quality & integrity checks at regular intervals.
 - **Annual Audits:** Perform structured reviews to assess long-term compliance.
- iv. **Outline the Change Management Process for Updating Models:** As business needs evolve, data models must be periodically updated while ensuring continued compliance. Entities must implement a formalized change management process to govern updates and modifications.
 - a. **Triggers for Model Updates:**
 - Changes in NPC's national data standards.
 - Introduction of new business processes or operational requirements.
 - Findings from compliance audits or quality assessments.
 - b. **Change Management Process:**
 - **Propose Change:** Data architects or business users submit a change request.
 - **Impact Assessment:** Evaluate technical, business, and compliance implications.
 - **Approval Workflow:** Governance team reviews and approves changes.
 - **Implementation & Testing:** Apply changes in a controlled environment.
 - **Version Update & Documentation:** Update the repository, metadata, and governance logs.

RACI Matrix – Data Modeling Guidelines & Methodology

Indicative

Sr No.	Activity	DGO	Data Committee	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Compliance Specialist	Data Owners
2.2.A	Obtain and Review NPC's Data Modeling Guidelines			I	A	R			
2.2.B	Establish Data Modeling Guidelines & Methodology Scope				A/C	R			
2.2.C	Develop Entity-Specific Data Modeling Guidelines & Methodology			I	A/C	R	C	C	
2.2.D	Review, Validate, and Finalize Data Modeling Guidelines		C		A/C	R	C	C	C
2.2.E	Approval, Publishing & Version Control of Data Modeling Guidelines		C	I	A/C	R			
2.2.F	Inform DGO	I		I	A	R			
2.2.G	Governance and Compliance Mechanisms for Model Validation and Updates			I	A	R	C	C	

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Create Data Modeling Guidelines Document



Implement a Version Control and Documentation Procedure

2.3 Baseline Data Architecture Assessment

The Baseline Data Architecture Assessment involves documenting the current state of the Entity's data architecture and data models. Having a clear view of the baseline architecture enables the Entity to identify gaps, dependencies, and opportunities for optimization as it progresses towards its target architecture.

2.3.1 Data Architecture Baseline Assessment

This section outlines the steps for gathering, documenting, and validating data architecture components, source systems, data platforms, and integration points. These steps ensure the baseline architecture is accurately captured to support future assessments and planning efforts.

2.3.1.A (1/2)

Gather & Consolidate Existing Documentation: Before conducting a manual assessment of the Entity's data architecture, the first step is to gather, review, and consolidate any existing documentation. This helps to leverage available resources, identify gaps, and minimize redundant efforts when documenting source systems, data platforms, integrations, and data flows.

- i. **Identify Available Documentation Sources:** The Entity must collect all existing architecture-related documentation across departments, IT teams, and governance bodies. Types of documentation include:
 - a. **Data Architecture Documentation:**
 - Previously developed architecture diagrams (if available).
 - Documentation of data flows, integration pipelines, and system dependencies.
 - Reports from previous assessments, audits, or IT infrastructure reviews.
 - b. **Data Governance & Metadata Repositories:**
 - Data dictionaries and metadata catalogs.
 - Business glossaries describing domains, subject areas, and key datasets.
 - Data governance policies defining ownership, stewardship, and access controls.
 - c. **Source System & Data Platform Documentation:**
 - ERDs and schema descriptions of databases.
 - Documentation of existing source systems and their data domains.
 - Platform-specific technical documentation (e.g., ETL workflows, cloud storage architectures).
 - d. **Data Lineage & Integration Reports:**
 - Existing lineage documentation mapping data sources to analytical platforms.
 - Reports on ETL/ELT pipelines and API-based integrations.
 - Data lineage tracking tools (if used).

Key sources to collect documentation from include:

- **Data Architecture Teams:** Architecture blueprints, data models, system landscapes.

2.3.1.A
Cont.
(2/2)

- Data Governance & Metadata Management Teams: Glossaries, dictionaries, lineage reports.
- IT & Database Administrators: Database schemas, cloud infrastructure documentation.
- BI & Analytics Teams: Data warehouse documentation, reporting structures.

ii. **Assess the Completeness & Accuracy of Existing Documentation:** Once documentation has been gathered, the Entity must evaluate its completeness and usability.

a. **Completeness Assessment:**

- Are all critical source systems, platforms, and integrations covered?
- Is there documentation available for both structured (RDBMS) and unstructured (Data Lakes, NoSQL) sources?
- Are the architecture diagrams and lineage reports up to date?

b. **Accuracy Assessment:**

- Briefly compare existing documentation with actual system configurations.
- Identify any inconsistencies or outdated details in data architecture reports.
- Validate metadata repositories for missing definitions, incorrect relationships, or incomplete descriptions.

c. **Gap Identification:**

- Identify areas where manual assessment will be required.
- Highlight any missing data repositories, undocumented integrations, or untracked data flows.

iii. **Organize & Store Documentation for Reference:** To ensure efficient access to collected materials, the Entity must:

- Centralize all documentation in a structured repository (e.g., SharePoint, metadata catalog).
- Apply version control to track changes and updates.
- Share documentation with relevant stakeholders involved in data architecture assessment.

This step enables the entity to establish a centralized and structured repository of existing documentation, ensuring easy reference for subsequent steps.

2.3.1.B
(1/5)

Identify & Document Source Systems: After consolidating existing documentation (2.3.1.A), the next step is to identify and document all source systems that provide data to the Entity's operational and analytical environments. This includes transactional databases, external sources, operational systems, and legacy platforms that contribute to the overall data architecture.

2.3.1.B
Cont.
(2/5)

- i. **Identify & Classify Source Systems:** The Entity must catalog all source systems that generate, store, or process structured and semi-structured data. Types of source systems to capture include but are not limited to:
 - a. **Transactional Databases (OLTP Systems):**
 - Systems that support business operations and capture real-time transactions.
 - **Examples:** CRM, ERP, Finance and HR systems.
 - b. **External Data Sources:**
 - APIs, third-party data feeds, industry data sources.
 - **Examples:** Third-party vendor data, social media APIs.
 - c. **Operational Systems:**
 - Applications that generate logs, monitoring data, IoT sensor data.
 - **Examples:** Cybersecurity Incident Monitoring System, Smart City Traffic Sensors.
 - d. **Legacy & Historical Systems:**
 - Older systems that still hold important historical data but may not be actively used.
 - **Examples:** Mainframe DB2, On-Prem Oracle Financials.
- ii. **Capture Key Metadata for Each Source System:** For each identified source system, the Entity must document the following metadata:



Asset Guidance

The **Current Data & Analytics Landscape Template** contains the **Source Systems** sheet that captures This sheet captures foundational metadata for each source system, including name, description, technology, ownership, stewardship, and data domain coverage. It defines who owns and manages the system and what domain(s) it supports. Use this sheet to establish visibility over system accountability and functional alignment.

Illustrative



Source Systems

This sheet is designed to capture key metadata about source systems, including their ownership, technology stack, data domain coverage, and the status of captured metadata and data quality checks.

Field	Description
Source System ID	A unique identifier assigned to each source system entry.
Source System Name	The official name or label for the data source.
Description	A brief summary of what the source system is and what it does.
Technology	The underlying platform or technology powering the source system.
Data Owner	The individual or department ultimately accountable for the data in this source system.
Data Steward	The person or group responsible for day-to-day data management and quality for the source system.
Data Domains/Sub-Domains	The specific data domain(s) or subdomain(s) covered by this source.

Source System ID	Source System Name	Description	Technology	Data Owner	Data Steward	Data Domains/Sub-Domains

2.3.1.B
Cont.
(3/5) **Asset Guidance**

The **Current Data & Analytics Landscape Template** contains the **Source System Details** sheet that captures operational metadata, key contacts, and access characteristics for each system, including business/IT SPOCs, volume, access methods, and system roadmap. It also captures critical flags such as business criticality, master source status, and internal/external classification. Use this sheet to assess system importance, contact structure, and future lifecycle plans.

Illustrative

	Source System Details
This sheet captures detailed ownership, support, and operational metadata about each source system, including departmental ownership, key technical contacts, business criticality, access mechanisms, and future roadmap. It supports system accountability, planning, and risk management.	

Field	Description
Id	Source System ID
Source System Name	The official name or label for the data source
Description	A small description about the source system
Department (Functional Owner of Source System)	Which department does the source system belong to
Business Spoc (Functional Owner of Source System)	Name of the Business Specific Point of Contact for this Source System
IT Source System Spoc	Name of the IT Specific Point of Contact for this Source System
System Administration Spoc	Name of the System Administrator for this Source System
DBA Spoc	Name of the DBA for this Source System
Data Volume (MB)	Total volume in MB
Data Accessibility (Direct/APIs/Proprietary)	Methodology used to access the data in the source system.
Business Critical (Y/N)	Yes or No if this source system is business critical
Key Business Processes (L1)	What are the Key Business Processes that this source system support
Future Roadmap(Upgrade/Replace/Maintain)	Any planned activity for this source system
Technical Metadata Captured (Y/N)	Indicates whether technical metadata (e.g., schemas, data types) has been documented.
Business Metadata Captured (Y/N)	Indicates whether business metadata (e.g., definitions, business rules) has been documented.
Master Source (Y/N)	Specifies if this source system is the "system of record" or master source for a particular data domain.
Data Quality Checks (Y/N)	Shows whether all required data quality checks have been performed on this source.
Internal/External	Identifies if the source system is managed internally or by an external party.

Id	Source System Name	Description	Department (Functional Owner of Source System)	Business Spoc (Functional Owner of Source System)	IT Source System Spoc

System Administration Spoc	DBA Spoc	Data Volume (MB)	Data Accessibility (Direct/APIs/Proprietary)	Business Critical (Y/N)	Key Business Processes (L1)

Future Roadmap (Upgrade/Replace/Maintain)	Technical Metadata Captured (Y/N)	Business Metadata Captured (Y/N)	Master Source (Y/N)	Data Quality Checks (Y/N)	Internal/External

2.3.1.B
Cont.
(4/5) **Asset Guidance**

The **Current Data & Analytics Landscape Template** contains the **Source Systems Environment** sheet that captures technical environment availability (production, test, dev) and test data readiness for each system. Use it to assess feasibility of profiling, integration, and QA activities. It helps identify technical limitations that may impact readiness.

Illustrative

 Source System Environments			
This sheet is designed to capture key metadata about source systems, including their ownership, technology stack, data domain coverage, and the status of captured metadata and data quality checks.			
Field	Description		
Source System ID	A unique identifier for the source system.		
Source System Name	The official name of the source system from which data is extracted or integrated.		
Production Environment (Y/N)	Indicates whether a production environment exists for the source system, used for live data and operational activities.		
Pre-Production Environment (Y/N)	Indicates whether a pre-production or staging environment exists. Typically used for final testing and validation before production deployment.		
Testing Environment (Y/N)	Indicates whether a dedicated test environment exists for development or QA activities, separate from pre-production.		
Development Environment (Y/N)	Indicates whether a development environment exists, used for configuration, prototyping, or model design before testing phases.		
Test Data Available in Preproduction: Y/N	Specifies whether anonymized or synthetic test data is available in the pre-production environment to support safe testing.		
Test Data Available in Testing Environment: Y/N	Specifies whether usable test data is available in the testing environment. This helps assess readiness for development and QA.		
Source System ID	Source System Name	Production Environment (Y/N)	Pre-Production Environment (Y/N)
Testing Environment (Y/N)	Development Environment (Y/N)	Test Data Available in Preproduction: Y/N	Test Data Available in Testing Environment: Y/N

 Asset Guidance

The **Current Data & Analytics Landscape Template** contains the **Key Datasets Inventory** sheet that inventories structured datasets per system, including schema, dataset name, volume, attribute and record counts. It also flags whether raw data has been ingested and whether the dataset contains reference or master entities. Use this sheet to identify key datasets, structure, and readiness for cataloging or quality review.

Illustrative

 Key Dataset Inventory					
This sheet documents the structure and scale of key datasets within each source system. It captures metadata such as dataset names, schema, volume, record count, and whether the data is reference/master.					
Field	Description				
Source System ID	A unique identifier assigned to each source system entry.				
Source System Name	The official name or label for the data source where the dataset originates.				
Schema Name	The name of the schema or database namespace in which the dataset resides.				
Dataset ID	A unique identifier for the dataset within the source system (e.g., D_001).				
Dataset Name	The name of the dataset within the source system.				
Dataset Description	A brief explanation of the datasets purpose and what type of data it contains.				
Data Volume (MB)	The estimated or actual size of the data stored in the dataset, measured in megabytes (MB).				
Number of Attributes	The total number of columns or fields defined in the dataset schema.				
Number of Records	The total number of rows or entries currently stored in the dataset.				
Raw Data Ingested (Y/N)	Indicates whether raw (unprocessed) data from the dataset has been ingested into the target data platform.				
Reference/MasterEntity (Y/N)	Specifies whether the dataset contains reference or master data, such as controlled vocabularies or core domain entities used across systems.				
Source System ID	Source System Name	Schema Name	Dataset ID	Dataset Name	Dataset Description
Data Volume (MB)	Number of Attributes	Number of Records	Raw Data Ingested (Y/N)	Reference/Master Entity (Y/N)	

2.3.1.B
Cont.
(5/5)

iii. **Identify Legacy & Redundant Source Systems:** To optimize the Entity's data architecture, it is essential to identify outdated, redundant, or siloed data repositories that may need consolidation or decommissioning.

a. **Legacy Systems:**

- Identify databases and applications that run on outdated technologies or are planned for migration.
- **Example:** Mainframe-based DB2 system still hosting customer records but no longer updated.

b. **Duplicate or Overlapping Systems:**

- Identify cases where multiple systems store the same data without synchronization.
- **Example:** Customer data exists in both the CRM and a separate customer database, causing inconsistencies.

This step enables the entity to uncover a list of legacy systems and redundant repositories to support future consolidation efforts.

iv. **Ensure Documentation Completeness & Accuracy:** Before finalizing 2.3.1.B, the Entity must validate the accuracy and completeness of documented source systems.

a. **Confirm System Coverage:**

- Ensure that all critical operational and analytical data sources are included.
- Validate that structured, semi-structured, and external sources are captured.

b. **Conduct Stakeholder Review:**

- Engage DBAs, IT teams, and data governance teams to verify documentation.
- Ensure that ownership, stewardship, and metadata availability are correctly recorded.

2.3.1.C
(1/3)

Identify & Document Data Platforms: After identifying source systems (2.3.1.B), the next step is to document the data platforms where data is stored, processed, and integrated. These platforms support operational processes, analytics, reporting, and decision-making. This includes data warehouses, data lakes, integration platforms, and data hubs that serve as the backbone of the Entity's data architecture.

i. **Identify & Classify Data Platforms:** The Entity must document all platforms used for structured, semi-structured, and unstructured data storage and processing. Types of data platforms to capture include:

a. **Data Warehouses & Data Marts:**

- Stores structured, historical, and aggregated data for reporting and analytics.
- **Examples:** Enterprise DWH, Finance Data Mart, HR Analytics Platform.

b. **Data Lakes & Big Data Platforms:**

- Stores raw, semi-structured, and unstructured data for large-scale analytics.
- **Examples:** Data Lake on Hadoop, Cloud Data Lake (AWS S3, Azure Data Lake, Google Cloud Storage).

2.3.1.C
Cont.
(2/3)

c. Operational Data Stores (ODS)

- Provides near real-time operational reporting by integrating data from transactional systems.
- Examples: Customer 360 ODS, Citizen Service Request ODS.

d. Integration & Data Virtualization Platforms:

- Supports data movement, transformation, and on-demand access across multiple sources.
- Examples: ETL Pipelines (Informatica, Talend), Data Virtualization (Denodo)

ii. **Capture Key Metadata for Each Data Platform:** For each identified data platform, the Entity must document the following metadata:

Asset Guidance

The **Current Data & Analytics Landscape Template** contains a Data Platforms Sheet that the Entity can leverage to capture core information about its data platform(s).

Illustrative

 Data Platforms							
This template captures core information about an Entity's data platform(s), including the underlying technologies, integrated source systems, data layers,							
Field				Description			
Id				A unique identifier for each data platform entry.			
Data Platform Name				The official name or label of the data platform.			
Description				A brief summary of the platform's purpose or function within the organization.			
Technology				The primary technologies or frameworks used by the data platform.			
Source Systems Integrated				The data sources (internal or external) that feed into the platform, typically listed or counted.			
Data Layers: Count + Names + Descriptions				The various layers (staging, integration, analytics, etc.) within the platform, with a short description of each.			
Integration Interfaces (Count + Key Names + Descriptions + Complexity)				The interfaces or APIs that handle data ingestion and publishing, including how many there are and their key names.			
Data Volume (MB)				An estimate of the daily or total data volume stored or processed, measured in megabytes.			
Id	Data Platform Name	Description	Technology	Source Systems Integrated	Data Layers: Count + Names + Descriptions	Integration Interfaces (Count + Key Names + Descriptions + Complexity)	Data Volume (MB)

iii. **Identify Redundant Data Platforms (If Any):** As part of the data platform assessment, the Entity must document outdated or inefficient platforms that may need replacement or consolidation.

a. Legacy Data Platforms:

- Identify platforms running on outdated technologies.
- Example: On-prem Teradata warehouse that is planned for migration to Snowflake.

b. Data Platform Duplication:

- Identify multiple platforms performing similar functions without clear differentiation.
- Example: Two separate data warehouses storing the same financial data for reporting.

This step enables the entity to uncover a list of legacy platforms and redundant data repositories, supporting future modernization efforts.

2.3.1.C
Cont.
(3/3)

- iv. **Ensure Documentation Completeness & Accuracy:** Before finalizing this step, the Entity must review its findings to ensure data platforms are properly documented.

a. **Confirm Platform Coverage:**

- Ensure all major data lakes, warehouses, and processing environments are included.
- Validate ETL/ELT and integration workflows connecting these platforms.

b. **Conduct Stakeholder Review:**

- Engage data engineers, architects, and platform administrators to verify accuracy.
- Ensure that all integration dependencies and usage details are correctly recorded.

2.3.1.D
(1/2)

Identify & Document Data & Analytics Software Stack: To complete the baseline architecture assessment, the Entity must identify and document all software components that support data processing, analytics, and reporting. This includes both enterprise-wide tools and department-specific solutions used for tasks such as data integration, visualization, modeling, and orchestration.

- i. **Identify and Classify Software Components:** The Entity must compile a list of all software products used across its data and analytics environment. These tools should be grouped based on their functionality and deployment model.

a. **Types of Software to Capture:**

- **Data Integration / ETL Tools:** e.g., Informatica, Talend, SSIS
- **Business Intelligence / Visualization:** e.g., Tableau, Power BI, Qlik
- **AI/ML Development Platforms:** e.g., Python (scikit-learn), R, Azure ML Studio
- **Data Modeling Tools:** e.g., Erwin, ER/Studio
- **Data Orchestration / Pipeline Management:** e.g., Apache Airflow, Azure Data Factory
- **Metadata / Cataloging Tools:** e.g., Informatica EDC

b. **Classify Based on Deployment:**

- Cloud-based (SaaS)
- On-premise / Hosted in Entity's Data Center
- Hybrid Deployments

- ii. **Capture Key Metadata for Each Software Tool:** For each software product identified, the Entity must complete the Data & Analytics Software Stack Template with the metadata outlined in the respective asset.

2.3.1.D
Cont.
(2/2) **Asset Guidance**

The **Current Data & Analytics Landscape Template** contains the Data & Analytics SW Stack Template Sheet which the Entity can leverage to capture key metadata for each software tool.

Illustrative



Data & Analytics SW Stack

This template documents the key software components used in the Entity's data and analytics environment.

Field	Description
Id	A unique identifier for the software record.
Software Scope	A short explanation of the software's functional or operational coverage.
Product Name & Version/Release	The official product name along with the current version or release number in use.
Vendor Name	The name of the company or provider that develops and distributes the software.
Support Details	Information regarding support and maintenance arrangements (e.g., SLAs, contract).
Deployment Details	How and where the software is deployed (on-premise, cloud, hybrid) along with any key notes.
Departments & Users Served	Which departments use it, plus approximate user counts or key user roles.

Id	Software Scope	Product Name & Version/Release	Vendor Name	Support Details	Deployment Details	Departments & Users Served

iii. **Identify Redundant or Outdated Tools (If Any).** The Entity should assess the stack to identify:

a. **Legacy Tools:**

- Products running on outdated technologies or nearing end-of-life.
- Example: Legacy on-prem ETL tool no longer compatible with cloud platforms.

b. **Functional Overlap or Redundancy:**

- Tools with overlapping capabilities and unclear differentiation.
- Example: Two reporting tools used for the same purpose across departments.

This assessment will help inform future rationalization or modernization initiatives.

iv. **Ensure Documentation Completeness & Accuracy:**

a. **Confirm Software Coverage:**

- Ensure all critical tools across data ingestion, transformation, modeling, and visualization are included.
- Validate that any shadow IT or department-specific tools are not missed.

b. **Conduct Stakeholder Review:**

- Engage IT, system owners, and data engineers to validate the software list.
- Confirm version, usage, and deployment details are correctly captured.

2.3.1.E
(1/2)

Identify & Document Analytics, AI/ML & Generative AI Artifacts: To complete the baseline assessment of the Entity's analytics ecosystem, the Entity must identify and document all analytics, AI/ML, and generative AI artifacts currently in use. This includes reports, dashboards, traditional ML models, and generative AI solutions developed or deployed to support business intelligence, forecasting, process automation, and decision-making.

This step helps the Entity gain a clear understanding of how analytical solutions are consumed, supported, and governed, and which data sources, technologies, and users are involved.

- i. **Identify & Classify Analytics and AI Artifacts:** The Entity must gather an inventory of all active analytics and AI/ML solutions used across departments and business functions.
 - a. **Types of Artifacts to Capture:**
 - **Dashboards & Reports:** Business dashboards used for real-time monitoring, operational reporting, and strategic insights.
 - **AI/ML Models & Generative AI Solutions:** A broad class of models including predictive analytics, segmentation, anomaly detection etc., as well as generative AI solutions.
 - b. **Departments & Use Cases:**
 - Identify which departments use each artifact (e.g., Finance, HR, Operations).
 - Record the business purpose, target users, and decision-making impact.
- ii. **Capture Key Metadata for Each Report, Dashboard, Model, & Generative AI Solution:** Use the Analytics & AI/ML Documentation Template to record details about each artifact.

Asset Guidance

The **Current Data & Analytics Landscape Template** contains the Analytics, AI/ML Models, & Generative AI Solutions Template Sheet which the Entity can leverage to capture key metadata for each report, dashboard, model, and generative AI solution.

Illustrative



Analytics, AI/ML Models, & Generative AI Solutions

This template documents analytics and AI/ML artifacts such as reports, dashboards, and models, capturing their purpose, key stakeholders, and underlying data sources.

Field	Description
Id	A unique identifier for each analytic or AI/ML artifact.
Report/Dashboard/Model/Solution Name	The official or commonly used name of the analytics, AI/ML, or Generative AI solution.
Report/Dashboard/Model/Solution Description	A concise summary of the solution's purpose, key capabilities, and the type of insights or outputs it delivers.
Departments/Key Users Served	The departments and/or user groups that rely on the solution.
Data Platform	The data environment or platform that hosts or supports the artifact.
Source Systems	The data sources or inputs feeding into the model, dashboard, or report.
Technology & Frameworks	The tools, frameworks, or programming languages used to develop, train, deploy, or visualize the solution.

Id	Report / Dashboard / Model / Solution Name	Report / Dashboard / Model / Solution Description	Departments / Key Users Served	Data Platform	Source Systems	Technology

2.3.1.E
Cont.
(2/2)iii. **Validate and Finalize the Analytics & AI/ML Inventory:**a. **Confirm Artifact Coverage:**

- Ensure the inventory includes all major reports, dashboards, traditional AI/ML models, and generative AI use cases.
- Include both centralized enterprise solutions and department-level or experimental pilots.

b. **Conduct Stakeholder Review:**

- Engage report developers, analytics teams, AI/ML engineers, and innovation teams to validate the inventory.
- Confirm all metadata is accurate, current, and aligned with business use cases.

2.3.1.F
(1/3)

Document Current Data Integration Points: Once the source systems, data platforms, data and analytics software's, dashboards, reports, AI/ML models and generative AI solutions have been documented, the next step is to identify and document how systems are integrated. This includes ETL/ELT pipelines, API-based integrations, event-driven connections, and file-based transfers. The objective is to capture structured information on which systems exchange data, how frequently, and by what means.

i. **Identify and Categorize Integration Types:** The Entity must categorize system integration points according to the type and frequency of data exchange. Integration types include:a. **Batch Data Flows (ETL/ELT Pipelines):**

- Data is extracted, transformed, and loaded at scheduled intervals (e.g., daily).
- Commonly used for reporting, warehousing, and historical data synchronization.

b. **Real-Time Data Flows (API & Middleware-Based Integrations):**

- Data is requested and exchanged on demand using APIs, middleware, or web services.
- Common for citizen services, authentication systems, and online portals.

c. **Event-Driven Data Streams:**

- Data is published and subscribed to using event brokers (e.g., Kafka, Azure Event Hub).
- Used for monitoring systems, IoT sensor data, fraud detection, and security logs.

d. **Data Sharing & File-Based Transfers:**

- Data is exchanged using SFTP, shared folders, or government data hubs.
- Common for compliance submissions and inter-departmental reports.

d. Duplicate or Overlapping Systems:

- Identify cases where multiple systems store the same data without synchronization.
- Example: Customer data exists in both the CRM and a separate customer database, causing inconsistencies.

ii. **Document Integration Points Between Systems:** Once data flow types are identified, the Entity must document which systems are exchanging data and how frequently this occurs.

- a. **Capture System-to-System Integration Details.** Each integration point must be documented with source, destination, frequency, and transfer mechanism.

Asset Guidance

The Entity can leverage the **Integration Points Documentation Template** to capture key details about system integrations, data exchange mechanisms, and governance tracking *Illustrative*

Integration Points Documentation Template						
This template captures key details about system integrations, data exchange mechanisms, and governance tracking to ensure visibility, and operational efficiency.						
Field	Description					
Integration Name	The name of the integration process, describing the data exchange between source and destination systems.					
Source System	The system providing the data, responsible for originating the information being transferred.					
Destination System	The system receiving the data, where the transferred data will be processed, stored, or consumed.					
Integration Type	The type of integration approach used, specifying how data is exchanged between systems.					
Frequency	Defines how often the integration occurs, ensuring alignment with business requirements and data freshness.					
Data Transfer Mechanism	The protocol or method used for transferring data, ensuring secure and efficient data exchange.					
Data Sensitivity Level	Specifies whether the transferred data is Top Secret, Secret, Restricted, Internal, Public					
Owner/Responsible Party	Identifies the team or department managing the integration and responsible for monitoring, troubleshooting, and governance.					
Error Handling Mechanism	Defines how integration failures, errors, or exceptions are handled, ensuring resilience and automated recovery workflows.					
Data Volume	The approximate size of data transferred per integration cycle, helping plan for scalability and performance.					
Latency Expectation	Defines the expected time for data to be delivered and processed, ensuring performance meets business and operational needs.					
Logging & Monitoring	Specifies whether logs are maintained for tracking integration events, errors, and performance metrics.					
Security & Compliance Measures	Details encryption standards, authentication mechanisms, and other security protocols used to protect data during					
Last Updated	The date when the integration details were last reviewed or modified.					

Integration Name	Source System	Destination System	Integration Type	Frequency	Data Transfer Mechanism	Data Sensitivity Level

Owner/Responsible Party	Error Handling Mechanism	Data Volume	Latency Expectation	Logging & Monitoring	Security & Compliance Measures	Last Updated

b. Capture System Dependencies & Criticality:

- Identify which integrations are mission-critical (e.g., systems that impact government operations if disrupted).
- Document dependencies between systems (e.g., Budget Reporting requires tax transaction data before processing).

iii. **Ensure Documentation Completeness & Accuracy:** Before finalizing, the Entity must validate the data integration points with key stakeholders:

- **Confirm Coverage:** Ensure all critical data movements (batch, real-time, API, event-driven) are documented.
- **Stakeholder Review:** Engage IT teams, data engineers, and business users to verify accuracy of current state between source and target flow of data.

2.3.1.F
Cont.
(3/3)

iv. **Data Integration Documentation Assets:** To maintain structured documentation, the Entity must:

- **Maintain an Integration Inventory:** Store metadata on all system-to-system data exchanges.
- **Apply Version Control:** Ensure documentation updates are tracked and reviewed periodically.

By documenting integration points, the Entity gains a clear view of how data moves between systems, ensuring efficient tracking of batch, real-time, and event-driven processes. This serves as a foundation for documenting detailed data lineage in 2.3.2.D, where transformations and dependencies will be captured.

2.3.1.G
(1/2)

Develop the Current Data Architecture Diagram: With data sources (2.3.1.B), platforms (2.3.1.C), and integrations (2.3.1.D) documented, the Entity must now develop a high-level architectural diagram to visualize how data components interact within the current data ecosystem. This provides a structured view of source systems, integration layers, storage solutions, processing components, and consumption endpoints.

i. **Define the Scope of the Data Architecture Diagram:** The diagram should represent the overall structure of the data ecosystem without excessive detail.

a. **Key Components to Include in the Diagram:**

- **Source Systems:** Government databases, external feeds, transactional applications.
- **Data Ingestion & Integration Layer:** ETL/ELT pipelines, API gateways, event brokers.
- **Storage & Processing Layer:** Data warehouses, data lakes, operational data stores.
- **Analytics & Reporting Layer:** BI dashboards, regulatory reporting systems.
- **Governance & Security Components:** Metadata repositories, data catalogs, access controls.

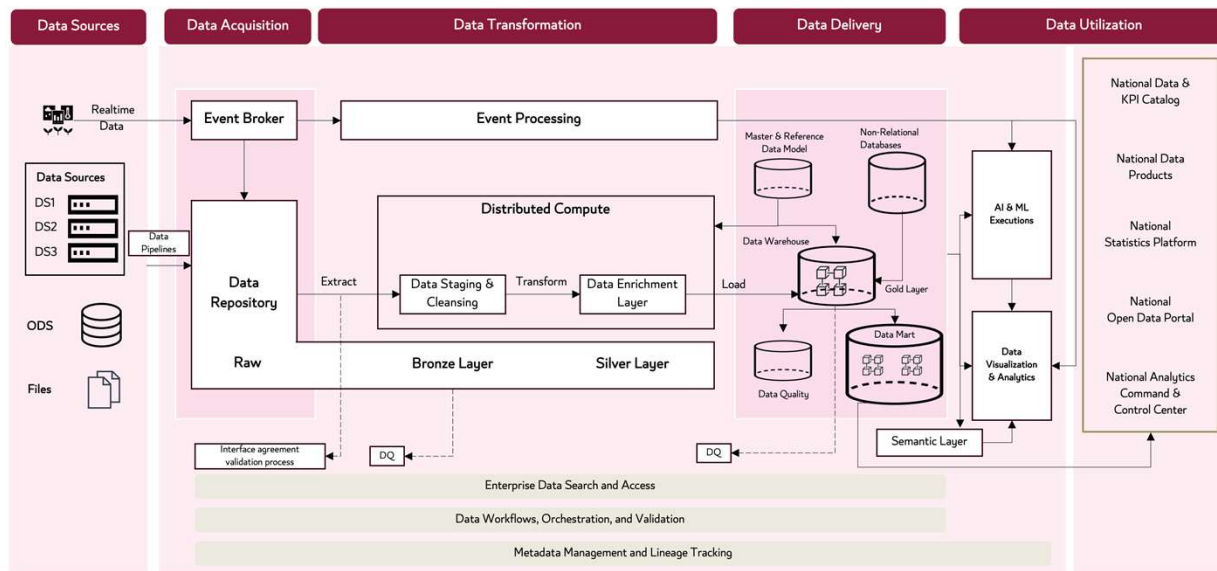
b. **Level of Detail to Capture:**

- Show how major systems are connected but avoid deep technical details.
- Focus on high-level interactions rather than individual datasets.
- Use modular sections to keep the diagram readable.

ii. **Design the Architecture Diagram:** The Entity must create a visual representation of its data architecture using standardized diagramming conventions. The suggested diagram structure is in alignment with the below reference architecture.

2.3.1.G
(2/2) **Asset Guidance**

The Entity can leverage the reference architecture in the **Data Architecture & Data Modeling Handbook** to design the architecture diagram.

Illustrative**Key Message / Information**

The entity should leverage tools for developing the data architecture diagram. Industry leading tools for manual diagrams include Visio, Lucidchart, and Draw.io.

iii. **Ensure Completeness & Accuracy of the Diagram:** Before finalizing, the Entity must validate the diagram to ensure it accurately reflects the current data architecture.

a. **Confirm All Key Components Are Represented:**

- Ensure all major data sources, storage solutions, and processing layers are included.
- Validate that critical integrations are captured.

b. **Conduct Stakeholder Review:**

- Engage enterprise architects, IT teams, and Data Governance Specialists to validate correctness.
- Ensure alignment with previously documented data flows and lineage records.

c. **Store in Centralized Repository:**

- Maintain the diagram in a version-controlled documentation platform.

The data architecture diagram offers a structured view of the Entity's data ecosystem, ensuring clarity in system interactions. This serves as a foundation for validating and finalizing the baseline assessment in 2.3.1.G.

2.3.1.H

Validate & Finalize the Baseline Data Architecture Assessment: With the data sources, platforms, flows, lineage, and architecture diagram documented, the Entity must now validate findings, consolidate documentation, and finalize the baseline assessment to support future analysis and planning.

i. Conduct Stakeholder Review & Validation:

- Verify completeness of the baseline assessment with IT, data architects, and governance teams.
- Ensure consistency between data architecture diagrams and captured documentation (source systems, data platforms, data and analytics software's, AI/ML & Generative AI solutions, and data integration documentation).
- Address any gaps or inaccuracies before finalizing documentation.

ii. Consolidate & Store the Assessment:

- Compile findings into a structured report and repository for easy reference.
- Store all diagrams, data lineage records, and integration details in a version-controlled platform.

RACI Matrix - Data Architecture Baseline Assessment

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Engineer	System Owners	Data Visualization Officer	IT Team
2.3.1.A	Gather & Consolidate Existing Documentation		A	R				
2.3.1.B	Identify & Document Source Systems		A	R	C	C		C
2.3.1.C	Identify & Document Data Platforms	I	A	R				
2.3.1.D	Identify & Document Data & Analytics Software Stack	I	A	R	C	C		C
2.3.1.E	Identify & Document Analytics, AI/ML & Generative AI Artifacts	I	A	R	C	C	C	C
2.3.1.F	Document Current Data Integration Points		A	R				
2.3.1.G	Develop the Current Data Architecture Diagram	I	A	R				
2.3.1.H	Validate & Finalize the Baseline Data Architecture Assessment	I	A/C	R				

RACI Matrix Legend: R: Responsible A: Accountable C: Consulted I: Informed

Based on this Activity, the Entity can



Develop the Current Data Architecture Diagram

2.3.2 Baseline Analytical Data Model Assessment

This step focuses on documenting the current state of the Entity's analytical data models, specifically datasets, tables, and attributes used for reporting and analytics. The goal is to gain a clear technical and business understanding of the analytical data model, ensuring it is well-structured, well-documented, and aligned with business needs.

2.3.2.A (1/2)

Identify Data Model Documentation and Metadata: Before extracting and analyzing the analytical data model, the Entity must identify, review, and assess existing data model documentation and metadata to determine the level of documentation available and its quality. This step ensures that any existing resources are leveraged before performing manual extraction and documentation.

i. **Identify Available Data Model Documentation:** Entities must determine whether any existing data model documentation is available within the organization. This includes:

a. **Entity-Relationship Diagrams (ERDs) / Data Model Diagrams:**

- Check if logical and physical data models have been documented.
- Identify any ERD diagrams for datasets and tables.
- Verify if relationship mappings between tables exist.

b. **Data Dictionary, Metadata Repository, and Schema Documentation:**

- Identify if a data dictionary exists with table and column definitions.
- Check metadata repositories for field descriptions, data types, constraints, and key relationships.
- Assess whether business rules, naming conventions, and transformation rules are documented.
- Review any database schema documentation or reports that describe fact and dimension tables.

ii. **Assess Documentation Completeness and Quality:** Once available documentation is identified, it must be assessed for completeness, accuracy, and relevance to the analytical data model. Key considerations include:

a. **Completeness Check:**

- Does the documentation cover all analytical tables?
- Are all attributes and data types clearly defined?
- Are business rules, hierarchies, and aggregations documented?

b. **Accuracy & Consistency Check:**

- Are there conflicting definitions for the same tables across different documents?
- Do relationships and cardinalities between tables align with actual system usage?

c. **Usability & Accessibility Check:**

- Is the documentation well-structured and easy to access?
- Does it follow naming conventions and NPC standards?
- Is it up-to-date, or does it require modifications?

2.3.2.A
Cont.
(2/2)

iii. **Validate Metadata Availability:** Metadata plays a critical role in ensuring data traceability, governance, and usability. The Entity must assess the availability and quality of metadata related to analytical models.

a. **Metadata Repositories & Lineage Tracking:**

- Identify if metadata is stored in dedicated metadata repositories or BI tools.
- Check if lineage tracking exists to show how data moves from source to analytical models.
- Assess if metadata includes data transformation rules, business logic, and historical changes.

b. **Key Metadata Elements to Validate:**

- Table & Column Definitions
- Data Types & Constraints
- Relationship Mappings
- Aggregation & Calculation Rules
- Business Glossary Alignment



Key Message / Information

The following steps are framed with the assumption that the Entity's data models includes fact and dimension tables. However, the guidance is still applicable to other table structures and data models or table structures and can be adapted accordingly based on the Entity's schema design.

2.3.2.B
(1/2)

Extract Analytical Data Model from Database/Data Warehouse: To gain a technical understanding of the current analytical data model, the Entity must extract table structures (e.g., fact and dimension tables) directly from the database or data warehouse. This ensures that the most up-to-date schema information is captured, regardless of whether previous documentation exists.

i. **Query Database / Data Warehouse to Extract Schema Information:** Since analytical data models (e.g., fact and dimension tables) are primarily stored in databases, data warehouses, or OLAP cubes, the Entity must retrieve schema details programmatically.

a. **Identify Relevant Data Sources:**

- Determine where fact and dimension tables reside (e.g., on-premise databases, cloud data warehouses, or federated storage).
- Verify if data is structured in star schema, snowflake schema, or hybrid models.

b. **Execute Queries to Extract Table & Column Metadata:**

- Retrieve fact and dimension tables using SQL queries / metadata extraction tools.
- Extract key metadata including:
 - Table names
 - Column names, data types, and constraints
 - Primary keys (PKs) and foreign keys (FKs)
 - Indexes and partitions

2.3.2.B
Cont.
(2/2)

- ii. **Analyze & Extract Fact Table Structures:** Fact tables store measurable business events and link to dimension tables via foreign keys. Extracting their structure helps understand granularity, aggregation rules, and measures.
 - a. **Identify All Fact Tables in the Data Warehouse:**
 - Verify fact tables based on naming conventions (e.g., fact_sales, fact_orders).
 - Retrieve all measures (metrics), calculations, and aggregations.
 - b. **Analyze & Extract Key Details for Each Fact Table.** Including but not limited to:
 - Fact table name
 - Attributes and their data types
 - Granularity level (e.g., daily, monthly, transactional)
 - List of measures (e.g., revenue, sales volume, cost)
 - Foreign keys linking to dimension tables
- iii. **Extract Dimension Table Structures:** Dimension tables provide descriptive attributes for facts and define hierarchies, aggregations, and reporting structures.
 - a. **Identify All Dimension Tables in the Data Warehouse:**
 - Verify dimensions based on naming conventions (e.g., dim_customer)
 - Extract all attributes and hierarchies.
 - b. **Extract Key Details for Each Dimension Table:**
 - Dimension table name
 - Attributes and their data types
 - Primary keys (natural or surrogate keys)
 - Relationships with fact tables (one-to-many relationships)
 - Hierarchy definitions (e.g., Year → Quarter → Month → Day for time dimensions)
- iv. **Capture Table Properties & Indexing Information:** Understanding how analytical tables are structured at the storage level is critical for performance tuning and optimization.
 - a. **Review Indexing Strategies:**
 - Identify primary and secondary indexes for fact and dimension tables.
 - Determine if indexing supports query performance optimization.
 - b. **Assess Partitioning & Clustering Strategies:**
 - Check if fact tables are partitioned (e.g., by date, region, or business unit).
 - Identify whether clustering keys are used to optimize analytical queries.
 - c. **Check Materialized Views or Summary Tables (if applicable):**
 - Determine if pre-aggregated tables exist to improve query performance.
 - Identify refresh frequency and dependency on raw fact/dimension data.

2.3.2.C
(1/4)

Document Tables and Attributes in a Structured Format: Once the fact and dimension tables have been extracted from the database or data warehouse, the Entity must document them in a structured format to improve data traceability, governance, and reporting clarity. This documentation ensures that data models are well-defined, consistently structured, and accessible for analysis and optimization.



Key Message / Information

If the entity does not utilize fact and dimension tables and instead employs alternative data models or table structures, they may use the designated alternative asset (**Datasets and Attributes Documentation Template**) to document their data tables and related assets accordingly.

Additionally, given that this is a baseline assessment, not all metadata attributes outlined in the templates may be available or previously defined within the Entity at this stage. The focus of this step is to gather key and essential baseline details to support future assessments and planning.

- i. **Populate the Baseline Fact & Dimension Table Inventory:** The first step is to compile an inventory of all tables, categorizing them based on business function or subject area.
 - a. **High-level Fact Table Details.** Capture metadata at the fact table level.
 - Group tables based on business domains (e.g., Sales, Finance, HR, Operations).
 - Identify active tables vs. deprecated or historical tables.



Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact Table Inventory Template which the Entity can leverage to capture metadata at the fact table level.

Illustrative

Fact Table Inventory							
The following table can be used to capture metadata at the Fact table level							
Field	Description						
Fact Table ID	ID number of the Fact Table.						
Fact Table Name	Name of the Fact table.						
Fact Table Description	Description of the respective Fact Table.						
Fact Table Business Domain	The business area or function the Fact table supports (e.g., Sales, Finance, HR).						
Primary Business Process	The business process the Fact table supports (e.g., Order Processing, Customer Transactions).						
Granularity	Defines the level of detail recorded in the Fact table (e.g., Per Transaction, Per Day).						
Data Model Name	Name of the data model where the Fact table resides.						
Source Systems	The original systems where the data is sourced from (e.g., CRM, ERP).						
Number of Measures	Count of key numeric values stored in the Fact table.						
Number of Dimensions	Count of Dimension tables related to this Fact table.						
Retention Policy	Defines how long data is retained before being archived or deleted.						
Security Classification	Sensitivity level of the data (e.g., Public, Confidential, Restricted).						
Owner	The department or individual responsible for maintaining the Fact table.						
Update Frequency	Defines how often the table is updated (e.g., Real-Time, Daily, Weekly).						
Date Created	Date when the table was created.						
Last Updated	Date of the most recent update.						

Fact Table ID	Fact Table Name	Fact Table Description	Fact Table Business Domain	Primary Business Process	Granularity	Data Model name	Source Systems

Number of Measures	Number of Dimensions	Retention Policy	Security Classification	Owner	Update Frequency	Date Created	Last Updated

- b. **High-level Dimension Table Details.** Capture dimension table level metadata, including:
 - Group tables based on business domains (e.g., Sales, Finance, HR, Operations).
 - Identify active tables vs. deprecated or historical tables.

2.3.2.C
(2/4) **Asset Guidance**

The **Fact & Dimension Tables Inventory** contains a Dimension Table Inventory Template which the Entity can leverage to capture metadata at the Dimension table level.

Illustrative

 Dim Table Inventory							
The following table can be used to capture metadata at the Dimension table level							
Field	Description						
Dimension Table ID	ID number of the Dimension Table.						
Dimension Table Name	Name of the Dimension table.						
Dimension Table Description	Description of the respective Dimension Table.						
Dimension Table Business Domain	The business area or function the Dimension table supports (e.g., Product, Customer, Employee).						
Dimension Type	Type of Dimension (e.g., Slowly Changing, Role-Playing, Degenerate).						
Hierarchy Available	Specifies if the Dimension has hierarchical levels (Yes/No).						
Hierarchy Levels	Lists the hierarchy levels if applicable (e.g., Country → Region → City).						
Data Model Name	Name of the data model where the Dimension table resides.						
Source Systems	The original systems where the Dimension data is sourced from (e.g., CRM, ERP, External Data).						
Number of Attributes	Count of attributes stored in the Dimension table.						
Retention Policy	Defines how long data is retained before being archived or deleted.						
Security Classification	Sensitivity level of the data (e.g., Public, Restricted).						
Owner	The department or individual responsible for maintaining the Dimension table.						
Update Frequency	Defines how often the table is updated (e.g., Weekly, Monthly).						
Date Created	Date when the table was created.						
Last Updated	Date of the most recent update.						

Dimension Table ID	Dimension Table Name	Dimension Table Description	Dimension Table Business Domain	Dimension Type	Hierarchy Available	Hierarchy Levels	Data Model Name

Source Systems	Number of Attributes	Retention Policy	Security Classification	Owner	Update Frequency	Date Created	Last Updated

- ii. **Document Fact Table Structures:** Fact tables store measurable business events and serve as the foundation for analytics. Each fact table must be documented with the following details:

 Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact Table Attribute Inventory Template which the Entity can leverage to document fact table structures.

Illustrative

 Fact Table Attribute Inventory							
The following table can be used to document Fact table details in a standardized format.							
Field	Description						
Fact Table ID	ID number of the Fact Table.						
Fact Table Name	The name of the Fact table where the attribute resides.						
Attribute Name	The name of the specific attribute within the respective Fact table.						
Attribute Description	A brief explanation of what the attribute represents and its business significance.						
Measure (Y/N)	Indicates whether the attribute is a measure (Y) or not (N). Measures are typically numeric values used for aggregations such as sums, averages, counts, or calculations in reports and dashboards.						
Calculation Methodology	If the attribute is a measure, this field describes how it is calculated. It should include the formula, business logic, or aggregation method used to derive the measure.						
Is Primary/Foreign Key	Specifies whether the attribute is a Primary Key (PK), Foreign Key (FK), or NA.						
Data Type	The data type of the attribute (e.g., Integer, Decimal, Date, Varchar).						
Length	Specifies the maximum number of characters or digits allowed for the attribute.						
Granularity	Defines the level of detail recorded in the Fact table (e.g., per transaction, per day).						
Source System	The original system where the data is sourced from (e.g., CRM, ERP).						
Owner	The department or individual responsible for maintaining the attribute.						
Update Frequency	Defines how often the attribute is updated (e.g., Real-Time, Daily, Weekly).						
Last Updated	The date when the attribute was last modified or refreshed.						

Fact Table ID	Fact Table Name	Attribute Name	Attribute Description	Measure (Y/N)	Calculation Methodology	Is Primary/Foreign Key

Data Type	Length	Granularity	Source System	Owner	Update Frequency	Last Updated

- iii. **Document Dimension Table Structures:** Dimension tables provide descriptive attributes for facts and support hierarchies, aggregations, and reporting structures. Each dimension table must be fully documented to ensure proper analytical usage.

2.3.2.C
Cont.
(3/4) **Asset Guidance**

The **Fact & Dimension Tables Inventory** contains a Dimension Table Attribute Inventory Template which the Entity can leverage to document dimension table structures. *Illustrative*


Dim Table Attribute Inventory

The following table can be used to document Dimension table details in a standardized format.

Field	Description
Dimension Table ID	ID number of the Dimension Table.
Dimension Table Name	The name of the Dimension table where the attribute resides.
Attribute Name	The name of the specific attribute within the respective Dimension table.
Attribute Description	A brief explanation of what the attribute represents and its business significance.
Is Primary/Foreign Key	Specifies whether the attribute is a Primary Key (PK), Foreign Key (FK), or NA.
Natural/Surrogate Key	Specifies whether the attribute is a natural key (derived from the source system and meaningful to the business) or a surrogate key (system-generated and used solely for integration and join operations). Common values include "Natural," "Surrogate," or "NA."
Slowly Changing Dimension Type	Indicates the SCD type used to manage historical changes for the attribute.
Data Type	The data type of the attribute (e.g., Integer, Decimal, Date, Varchar).
Length	Specifies the maximum number of characters or digits allowed for the attribute.
Hierarchies	Identifies whether the attribute is part of a hierarchical structure (e.g., Country → Region → City). If yes, specify the hierarchy level or grouping it belongs to (e.g., "Level 2 – Region").
Source System	A brief explanation of what the attribute represents and its business significance.
Owner	The department or individual responsible for maintaining the attribute.
Update Frequency	Defines how often the attribute is updated (e.g., Real-Time, Daily, Weekly).
Last Updated	The date when the attribute was last modified or refreshed.

Dimension Table ID	Dimension Table Name	Attribute Name	Attribute Description	Is Primary/Foreign Key	Natural/Surrogate Key	Slowly Changing Dimension Type

Data Type	Length	Hierarchies	Source System	Owner	Update Frequency	Last Updated

- iv. **Capture Data Model Relationships & Constraints:** To ensure data integrity and usability, the entity must map relationships between fact and dimension tables, as well as between dimension tables (in the case of a snowflake schema). The entity should also verify referential integrity by ensuring foreign keys properly correspond to their respective primary keys. Furthermore, the entity must identify any missing or orphaned records that could negatively affect analytics.

 Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact & Dimension Relationship Template which the Entity can leverage to document how fact and dimension tables are connected. *Illustrative*


Fact & Dimension Relationship

This Fact-Dimension Relationship Sheet provides a comprehensive view of how Fact and Dimension tables are connected.

Field	Description
S No.	A sequential identifier for each relationship entry in the table.
Relationship	A descriptive representation of the relationship between the two tables and their attributes (Syntax: "<from table>_<from column> -> <to table>_<to column>" e.g., DATE_DATEID -> EMPLOYEEHISTORY_JOININGDATEID).
From Fact Table ID	The unique identifier of the Fact table from which the relationship originates.
From Fact Table Name	The name of the Fact table that contains the measurable data (e.g., Sales Fact, Transactions Fact).
From Fact Table Attribute	The attribute in the Fact table that establishes the relationship with the Dimension table (typically a foreign key).
From Cardinality	The cardinality of the relationship from the Fact table (e.g., Many).
To Dimension Table ID	ID number of the Dimension Table.
To Dimension Table Name	The name of the related Dimension table (e.g., Customer Dimension, Product Dimension).
To Dimension Table Attribute	The name of the Dimension table that contains the primary key attribute.
To Cardinality	The cardinality of the relationship to the Dimension table (e.g., One).

S No.	Relationship	From Fact Table ID	From Fact Table Name	From Fact Table Attribute

From Cardinality	To Dimension Table ID	To Dimension Table Name	To Dimension Table Attribute	To Cardinality

2.3.2.C
Cont.
(4/4)

- v. **Compile Documentation in a Centralized Repository (as referenced in section 6):** Once fact and dimension tables are documented, the Entity must ensure proper version control, storage, and accessibility for future reference when conducting the gap analysis.

2.3.2.D
(1/2)

Document Baseline Data Lineage: The Entity must document how data moves, transforms, and is consumed across systems using the standardized Baseline Data Lineage Template. This ensures transparency into data origins, processing, and usage to support governance, compliance, and impact analysis.

- i. **Use the Baseline Data Lineage Template for Standardized Documentation:** To ensure consistency, the Entity must use the standardized Baseline Data Lineage Template to capture lineage at the attribute level across fact and dimension tables.

- a. **Capture Key Data Lineage Components.** Each documented data lineage entry must include:

- **Source Information:** Where the data originates (e.g., transactional systems).
- **Processing & Transformations:** Steps taken to clean, modify, or aggregate the data before storage.
- **Target System & Usage:** Where the processed data is stored and how it is used for analytics, reporting, or decision-making.

- b. **Baseline Data Lineage Template Structure:**

Asset Guidance

The Entity can leverage the **Baseline Data Lineage Template** to track the flow of data across systems.

Illustrative

Baseline Data Lineage Template							
The Baseline Data Lineage Template can be used to document and track the flow of data across systems.							
Field	Description						
Data Source	The name of the system, application, or database where the data originates (e.g., CRM system).						
Data Source Description	A brief description of the data source and the type of data it holds (e.g., transactional data, customer data).						
Source Dataset Name	Name of the Dataset in consideration (e.g., Entity_CRM_Customer).						
Source Attribute Name	Name of the Attribute in the source dataset (e.g., customer_ID).						
Data Transformation Description	Description of any transformation applied to the data, such as data cleansing, filtering, or aggregation (e.g., filter inactive customers).						
Data Transformation Logic	Specific logic, calculations, or rules applied to data during the transformation process (e.g., filter rows where status = 'inactive')						
Processing Step	Identify the step or system where data is processed (e.g., ETL tools, data pipelines).						
Target System	The destination where the data is stored or sent, such as a data warehouse, data lake, or BI system.						
Target System Description	A description of the target system and how the data is used or visualized (e.g., reports, analytics).						
Target Dataset Name	Name of the Dataset at the target destination (e.g., Fact_Customer).						
Target Attribute Name	Name of the Attribute in the target dataset (e.g., Customer_ID).						
Data Owner	The team or individual responsible for the data at each stage.						
Lineage Type	Whether the lineage is direct, derived, or aggregated.						
Compliance/Regulation	Any compliance or regulatory requirement that governs the data handling at that stage (e.g., GDPR).						
Change Impact	Notes on any possible impact of changes to this data lineage (e.g., schema changes, source changes).						
Comments	Any additional remarks or special instructions.						

Data Source	Data Source Description	Source Dataset Name	Source Attribute Name	Data Transformation Description	Data Transformation Logic	Processing Step	Target System

Target System Description	Target Dataset Name	Target Attribute Name	Data Owner	Lineage Type	Compliance/Regulation	Change Impact	Comments

2.3.2.D
(2/2)**c. Considerations for Documentation Quality:**

- Ensure all major datasets are documented, prioritizing those used in analytics, reporting, and compliance.
- Capture transformation logic at a high level, avoiding excessive technical details that belong in ETL documentation.

ii. Ensure Documentation Completeness & Accuracy: Before finalizing, the Entity must validate data lineage documentation with relevant stakeholders to ensure its accuracy and usability.

- **Review with Data Owners & Architects:**
 - Confirm all source, transformation, and target components are captured.
 - Ensure lineage entries reflect real-world use and ownership.
- **Check for Completeness & Gaps:**
 - Make sure all key data used for reporting and analytics is included.
 - Address any missing or inconsistent lineage records.
- **Store in a Central Repository:**
 - Maintain version-controlled documentation in a central repository for governance use, future reference, audits, and impact analysis.

Documenting data lineage ensures visibility into data movement, transformations, and usage, supporting governance, compliance, and impact analysis. This serves as a foundation for visualizing the overall data architecture.

2.3.2.E
(1/2)

Develop the Current Physical Analytical Data Model Diagrams: After documenting fact and dimension tables, the Entity must now visualize the current analytical data model by developing a Physical Data Model diagram. This step ensures that the structure, relationships, and dependencies of fact and dimension tables are clearly represented, providing a reference for governance, analysis, and future optimization.

I. Develop the Physical Data Model Diagram: The Physical Data Model diagram must be drawn to visually depict how fact and dimension tables are structured and interrelated.**a. Define Fact Tables & Their Relationships to Dimensions:**

- Position fact tables at the center with links to dimension tables.
- Ensure all primary key (PK) and foreign key (FK) relationships are reflected.
- Represent one-to-many relationships between facts and dimensions.

b. Include Table Attributes & Data Types:

- List columns within fact and dimension tables.
- Specify data types and primary/foreign key constraints.

c. Use Standard Notation & Diagramming Conventions:

- Follow Entity-Relationship Diagram notation.
- Ensure relationships are represented using crow's foot notation.
- Label relationships with cardinality rules (e.g., one-to-many).

2.3.2.E
Cont.
(2/2)

- II. **Ensure Completeness & Accuracy:** To ensure the Physical Data Model Diagram is accurate and comprehensive, the Entity must:
- a. **Cross-check against the documented fact and dimension inventory**
 - Ensure all fact and dimension tables are included.
 - Verify all primary and foreign key relationships.
 - b. **Validate Business Relevance:**
 - Ensure relationships reflect actual analytical reporting needs.
 - Confirm that all required measures and attributes are represented.
- III. **Store & Share the Physical Data Model Diagram:** Once completed, the Entity must store and share the Physical Data Model Diagram for governance and future reference.
- Save the diagram in a centralized repository (as referenced in section 6) for easy access.
 - Share with key stakeholders for validation (to be conducted in Step 2.3.2.E).
 - Maintain version control to track future updates.

2.3.2.F
(1/2)

Meet Stakeholders to Validate Domains, Subject Areas, and Data Model Diagrams: Once the Physical Data Model (PDM) diagram has been drawn, the Entity must engage key stakeholders to validate the accuracy of the documented fact tables, dimension tables, domains, subject areas, and data model diagrams. This step ensures that the baseline analytical data model correctly represents the current state as extracted and documented.

- I. **Identify & Engage Key Stakeholders:** To ensure that the documented analytical data model accurately reflects what exists in the database, the Entity must engage technical and functional stakeholders who have knowledge of the data sources.
- **Data Architects & Database Administrators:** Validate that the documented fact and dimension tables match the actual database schema.
 - **BI & Analytics Teams:** Verify that the relationships between fact and dimension tables are correctly captured.
 - **Data Governance & Stewardship Teams:** Confirm that domains and subject areas are correctly classified.
- II. **Conduct Validation Sessions for Data Model Accuracy:** The Entity must organize structured review sessions where stakeholders validate whether the documented analytical data model correctly reflects the database schema. Validation areas include:
- a. **Fact Tables, Attributes, & Measures:**
 - Confirm that the fact tables listed in the inventory match what exists in the database.
 - Verify that the measures and numeric attributes within fact tables are correctly documented.

2.3.2.F
Cont.
(2/2)**b. Dimension Tables & Attributes:**

- Validate that all dimension tables captured in the documentation exist in the database.
- Ensure that primary keys, attributes, and hierarchies are documented correctly.

c. Fact-Dimension Relationships:

- Confirm that foreign key relationships between fact and dimension tables match the actual database structure.
- Verify cardinality rules (one-to-many, many-to-one) in the data model diagram.

d. Domains & Subject Areas:

- Ensure that fact and dimension tables are grouped under the correct subject areas.

e. Physical Data Model Diagram:

- Validate that the diagram correctly represents the relationships, attributes, and constraints of the analytical data model.
- Ensure that the diagram aligns with what is physically implemented in the database.

III. Address Stakeholder Feedback & Finalize the Baseline Analytical Data Model: Based on stakeholder feedback, the Entity must incorporate any necessary adjustments before finalizing the baseline analytical data model.

a. Document Stakeholder Feedback Systematically:

- Track comments, discrepancies, and required corrections.
- Identify any missing tables, attributes, or relationships that were not captured.

b. Make Refinements to Fact & Dimension Tables and Model Diagrams (If Needed):

- Adjust table attributes, relationships, or hierarchies if any discrepancies are found.
- Ensure that all requested corrections are implemented before finalization.

c. Obtain Final Stakeholder Agreement:

- Conduct a final review session to confirm that all identified errors have been corrected.
- Secure formal sign-off from key technical stakeholders to validate that the documented baseline model correctly represents the database schema.

2.3.2.G

Analyze Analytical Data Model Storage & Indexing Efficiency: Once the Physical Data Model diagram has been validated with stakeholders, the Entity must assess how fact and dimension tables are physically stored and indexed to ensure optimized performance, efficient query execution, and scalable storage utilization.

- I. **Assess Fact Table Storage & Partitioning Strategies:** Fact tables store high-volume transactional or aggregated data, making storage and retrieval performance a key concern. The Entity must:
 - a. **Determine how large fact tables are partitioned. Examples include:**
 - Date (e.g., daily, monthly partitions)
 - Business unit (e.g., department-specific partitions)
 - b. **Evaluate whether partition pruning is effectively reducing query scan times:**
 - Check if analytical queries are hitting only relevant partitions instead of scanning full tables.
 - Ensure partitioning keys are aligned with common query filters for improved performance.
- II. **Review Indexing Strategies on Fact & Dimension Tables:** Indexing plays a critical role in query speed, table joins, and filtering operations. Identify whether primary keys (PKs) and foreign keys (FKs) are properly indexed:
 - Ensure fact table foreign keys are indexed for efficient joins with dimension tables.
 - Verify that dimension table primary keys are indexed to optimize lookups.

2.3.2.H

Finalize Baseline Analytical Data Model Assessment & Consolidate Findings: After evaluating storage and indexing efficiency, the Entity must finalize the baseline analytical data model assessment by consolidating all findings into a structured report. This step ensures that the current state of the analytical data model is fully documented, providing a foundation for future improvements, governance, and compliance assessments in the gap analysis (2.4).

RACI Matrix - Baseline Analytical Data Model Assessment

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Engineer	Data Steward	Data Governance Specialist
2.3.2.A	Identify Data Model Documentation and Metadata		A/C	R			
2.3.2.B	Extract Analytical Data Model from Database/Data Warehouse		A/C	R			
2.3.2.C	Document Tables and Attributes in a Structured Format	I	A/C	R			
2.3.2.D	Document Baseline Data Lineage		A	R			
2.3.2.E	Develop the Current Physical Analytical Data Model Diagrams	I	A/C	R			
2.3.2.F	Meet Stakeholders to Validate Domains, Subject Areas, and Data Model Diagrams		A/C	R	C	C	C
2.3.2.G	Analyze Analytical Data Model Storage & Indexing Efficiency		A/C	R			
2.3.2.H	Finalize Baseline Analytical Data Model Assessment & Consolidate Findings	I	A/C	R			

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Draw the Current Logical/Physical Analytical Data Model



Conduct Baseline Analytical Data Model Assessment

2.4 Data Architecture Gap Analysis

This section guides the Entity through a structured process to identify and address gaps in both the Data Architecture and Analytical Data Models. The gap analysis ensures that the Entity's current-state architecture and data models are aligned with the Entity's Data Strategy, the Entity's and NPC's Data Architecture & Modeling guidelines, national data standards, and best practices.

2.4.1 Data Architecture Gap Analysis

The Data Architecture Gap Analysis identifies gaps by assessing alignment with data strategy requirements, evaluating discrepancies against the Data Architecture & Modeling Handbook and general best practices, and consolidating and prioritizing identified gaps.

2.4.1.A
(1/2)

Assess Alignment with Data Strategy Requirements. The first step in the Data Architecture Gap Analysis is to evaluate how well the Entity's current data architecture aligns with the requirements set in its Data Strategy, as outlined in the Data Management Strategy & Governance domain. This ensures that the Entity's architecture supports its strategic objectives, data governance framework, and operational needs.

- i. **Assess Architectural Alignment with Data Strategy Objectives:** The Data Strategy defines the Entity's vision, goals, strategic objectives, and initiatives for managing data effectively. This assessment will identify gaps where the architecture does not fully support these strategic priorities.
 - a. **Data-Driven Decision-Making Enablement.** Key considerations include:
 - Does the current architecture enable data-driven decision-making across the Entity?
 - Are data storage, integration, and processing aligned with strategic objectives and initiatives?
 - b. **Governance & Data Management Framework Support.** Key considerations include:
 - Does the architecture reflect the governance structure outlined in the Data Strategy (e.g., data ownership, stewardship, compliance)?
 - Are data governance policies enforced through architecture components (e.g., security layers, metadata management, lineage tracking)?
 - Are governance workflows (approval processes, compliance checks, audit mechanisms) embedded into data operations?
 - c. **Data Integration & Interoperability.** Key considerations include:
 - Does the architecture enable data exchange across departments and external entities?
 - Are integration mechanisms (APIs, ETL/ELT pipelines, event streaming) aligned with strategic interoperability goals?
 - Is there a clear framework for managing cross-entity and / or public-private data collaborations?

2.4.1.A
Cont.
(2/2)d. **Completeness Check.** Key considerations include:

- Does the current architecture align with the Entity's long-term vision? (e.g., for cloud adoption, hybrid storage, modernization)
- Are legacy systems, data silos, and architectural bottlenecks limiting strategic execution?

ii. **Identify Gaps in Alignment with the Data Strategy:** After reviewing the strategic requirements, the Entity must identify misalignments between the current architecture and the Data Strategy.

Asset Guidance

The Entity can leverage the **Gap Analysis Template** to identify gaps between the current and future state of data architecture or data model.

Illustrative



Gap Analysis Template

The following table is a template for identifying gaps between current and future state of data architecture or data model.

Field	Description
Reference ID	Unique identifier for each gap item.
Gap Title	A brief title describing the gap.
Description	Detailed description of the gap.
Gap Source	Benchmark used to Determine the Gap (Data Strategy, Data Architecture & Modeling Guidelines, Best Practices, Other)
Gap Type	Categorize the gap (e.g., Process, Technology, Skill, Data, Governance).
Gap Category	Categorize the gap (e.g., Scalability & Performance, Integration & Interoperability, Architecture Documentation, Governance & Metatdata etc.)
Business Impact	Impact on business processes if the gap is not addressed.
Implementation Priority	Priority level for addressing the gap (e.g., High, Medium, Low).
Current State	Describe the current state that leads to the gap.
Target State	Define the desired future state to resolve the gap.
Proposed Solution	Describe the solution to address the gap.
Owner	Primary Owner of the gap.
Required Resources	List resources (e.g., tools, budget, personnel) required to resolve the gap.
Stakeholders Involved	Key stakeholders responsible for resolving or affected by the gap.
Dependencies	Dependencies that may affect gap resolution (e.g., other projects, systems).
Risk Level	Risk associated with resolving the gap (e.g., Low, Medium, High).
Status	Current status of gap resolution (e.g., Open, In Progress, Resolved).

Reference ID	Gap Title	Description	Gap Source	Gap Type	Gap Category	Business Impact	Implementation Priority	Current State

Target State	Proposed Solution	Owner	Required Resources	Stakeholders Involved	Dependencies	Risk Level	Status

iii. **Document Gaps:** The Entity should document all identified gaps to ensure alignment with the subsequent analysis of compliance with NPC's National Data Standards, Data Architecture & Modeling Handbook, and best practices

- Summarize misalignments with the Data Strategy.
- Categorize gaps by impact level (high, medium, low).
- Store findings in a structured format to support remediation planning in 2.5 (Data Architecture Roadmap).

2.4.1.B
(1/2)

Evaluate Gaps Against the NPC Data Architecture Guidelines. This step assesses how well the Entity's data architecture aligns with the NPC Data Architecture & Modeling Handbook, identifying gaps in architecture documentation, integration, governance, scalability, metadata management, and security.

- i. **Assess Compliance with NPC Data Architecture Guidelines:** The NPC Data Architecture & Modeling Handbook defines best practices for data architecture. The Entity must compare its current architecture against the following key areas:
 - a. **Data Architecture Documentation & Processes.** Key considerations include:
 - Has the Entity documented system-level data architectures, including data storage, processing, and integration flows?
 - Are data architecture diagrams and documentation maintained and regularly updated?
 - b. **Data Storage & Processing Design.** Key considerations include:
 - Does the Entity's architecture follow the recommended high-level data architecture layers (Data Acquisition, Transformation, Delivery, Utilization) as outlined in the Data Architecture & Modeling Guidelines reference architecture?
 - Are data repositories structured efficiently for scalability and performance?
 - Does the Entity use real-time, batch, and event-driven processing as needed?
 - c. **Integration & Interoperability Standards .** Key considerations include:
 - Does the architecture support seamless data exchange across internal and external systems?
 - Are ETL/ELT pipelines, APIs, and integration frameworks aligned with NPC's standards?
 - d. **Data Governance & Metadata Management.** Key considerations include:
 - Are data governance controls embedded into the architecture to ensure compliance with security and access policies?
 - Does the Entity have a centralized metadata repository and lineage tracking for architecture components?
 - e. **Scalability, Security, & Compliance.** Key considerations include:
 - Is the data architecture scalable across domains, subject areas, and business units?
 - Are security and compliance controls aligned with NPC's data security and privacy requirements?
- ii. **Document Gaps:** The Entity must document where its architecture does not align with NPC's best practices.

2.4.1.B
(2/2) **Asset Guidance**

The Entity can leverage the **Gap Analysis Template** to identify gaps between the current and preliminary future state of data architecture.

Illustrative

Field		Description
Reference ID		Unique identifier for each gap item.
Gap Title		A brief title describing the gap.
Description		Detailed description of the gap.
Gap Source		Benchmark used to Determine the Gap (Data Strategy, Data Architecture & Modeling Guidelines, Best Practices, Other)
Gap Type		Categorize the gap (e.g., Process, Technology, Skill, Data, Governance).
Gap Category		Categorize the gap (e.g., Scalability & Performance, Integration & Interoperability, Architecture Documentation, Governance & Metadata etc.)
Business Impact		Impact on business processes if the gap is not addressed.
Implementation Priority		Priority level for addressing the gap (e.g., High, Medium, Low).
Current State		Describe the current state that leads to the gap.
Target State		Define the desired future state to resolve the gap.
Proposed Solution		Describe the solution to address the gap.
Owner		Primary Owner of the gap.
Required Resources		List resources (e.g., tools, budget, personnel) required to resolve the gap.
Stakeholders Involved		Key stakeholders responsible for resolving or affected by the gap.
Dependencies		Dependencies that may affect gap resolution (e.g., other projects, systems).
Risk Level		Risk associated with resolving the gap (e.g., Low, Medium, High).
Status		Current status of gap resolution (e.g., Open, In Progress, Resolved).

Reference ID	Gap Title	Description	Gap Source	Gap Type	Gap Category	Business Impact	Implementation Priority	Current State

Target State	Proposed Solution	Owner	Required Resources	Stakeholders Involved	Dependencies	Risk Level	Status

iii. Finalize Findings & Store for Remediation Planning:

- Summarize key architecture gaps based on NPC's handbook.
- Categorize issues by impact level to guide prioritization.
- Store findings for future remediation planning in the Data Architecture Roadmap.

2.4.1.C
(1/2)

Evaluate Gaps Against General Best Practices: This step assesses how well the Entity's data architecture aligns with recognized industry best practices in scalability, integration, governance, metadata management, and security. The goal is to identify areas where improvements can enhance efficiency, compliance, and long-term sustainability.

i. Assess Compliance with Data Architecture Best Practices: The Entity must compare its current data architecture against widely accepted best practices. Key areas include but are not limited to:

a. Scalability & Performance:

- Does the architecture support scalability for future data growth and workload expansion?
- Are cloud, hybrid, and distributed storage solutions leveraged effectively?
- Is there high availability and disaster recovery planning for critical data stores?

b. Integration & Interoperability:

- Are API-driven data exchanges, event-driven architectures, and ETL/ELT pipelines optimized?
- Does the Entity use data virtualization or federation to enable seamless access to distributed data?
- Are cross-entity and intergovernmental data-sharing mechanisms standardized?

2.4.1.C
(2/2)c. **Data Governance & Metadata Management:**

- Is there a centralized metadata repository and automated lineage tracking?
- Are data catalogs, glossaries, and stewardship policies enforced across data platforms?
- Does the Entity have version control for architecture changes and data models?

d. **Security & Compliance:**

- Are role-based access controls, data masking, and encryption policies implemented consistently?
- Does the architecture support real-time monitoring, audit trails, and anomaly detection?
- Are data sovereignty, retention, and privacy policies aligned with regulatory requirements?

ii. **Identify & Document Gaps:** The Entity must document where its architecture does not align with best practices.

Asset Guidance

The Entity can leverage the **Gap Analysis Template** to identify gaps between the current and preliminary future state of data architecture.

Illustrative

	Gap Analysis Template
The following table is a template for identifying gaps between current and future state of data architecture or data model.	

Field	Description
Reference ID	Unique identifier for each gap item.
Gap Title	A brief title describing the gap.
Description	Detailed description of the gap.
Gap Source	Benchmark used to Determine the Gap (Data Strategy, Data Architecture & Modeling Guidelines, Best Practices, Other)
Gap Type	Categorize the gap (e.g., Process, Technology, Skill, Data, Governance).
Gap Category	Categorize the gap (e.g., Scalability & Performance, Integration & Interoperability, Architecture Documentation, Governance & Metadata etc.)
Business Impact	Impact on business processes if the gap is not addressed.
Implementation Priority	Priority level for addressing the gap (e.g., High, Medium, Low).
Current State	Describe the current state that leads to the gap.
Target State	Define the desired future state to resolve the gap.
Proposed Solution	Describe the solution to address the gap.
Owner	Primary Owner of the gap.
Required Resources	List resources (e.g., tools, budget, personnel) required to resolve the gap.
Stakeholders Involved	Key stakeholders responsible for resolving or affected by the gap.
Dependencies	Dependencies that may affect gap resolution (e.g., other projects, systems).
Risk Level	Risk associated with resolving the gap (e.g., Low, Medium, High).
Status	Current status of gap resolution (e.g., Open, In Progress, Resolved).

Reference ID	Gap Title	Description	Gap Source	Gap Type	Gap Category	Business Impact	Implementation Priority	Current State

Target State	Proposed Solution	Owner	Required Resources	Stakeholders Involved	Dependencies	Risk Level	Status

iii. **Finalize Findings & Store for Remediation Planning:**

- Summarize key architecture gaps based on best practices.
- Categorize issues by impact level to guide prioritization.
- Store findings for remediation planning in 2.5 (Data Architecture Roadmap).

2.4.1.D

Consolidate & Prioritize Data Architecture Gaps: After assessing data architecture gaps against the Data Strategy, NPC Data Architecture & Modeling Handbook, and general best practices, the Entity must now summarize key findings, categorize gaps by impact, and prepare insights for remediation planning in 2.5 (Data Architecture Roadmap).

- i. **Summarize Key Findings from the Gap Analysis:** The Entity must compile all identified gaps and structure them into clear categories and prioritize the gaps based on business impact and urgency for remediation.

Asset Guidance

The Entity can leverage the **Gap Analysis Key Findings Template** summarize all identified gaps and categorize them based on impact, and urgency for remediation.

Illustrative

Gap Analysis Key Findings Template Slide

Gap Title	Gap Category	Description of Gap	Gap Impact	Gap Priority
Limited Scalability & Lack of Standardized Data Interoperability	Best Practices	The current data architecture does not effectively support scalability for future data growth, and interoperability issues exist between internal and external systems due to inconsistent integration standards.	Scalability limitations may lead to performance bottlenecks, and increased storage costs, while lack of interoperability hinders seamless data exchange with external entities.	Medium
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	[Insert Description]	[Insert Impact]	[insert]

Priority High Medium Low

- ii. **Document Risks & Prepare for Remediation Planning:** For each high-impact gap, the Entity should document associated risks and dependencies to guide remediation planning in 2.5 (Data Architecture Roadmap).
- iii. **Store & Finalize Documentation for Roadmap Planning:**
 - Compile findings into a structured report for leadership and technical teams.
 - Store documentation in a centralized repository for reference during remediation planning.
 - Validate findings with key stakeholders before moving into 2.5 (Data Architecture Roadmap).

RACI Matrix – Data Architecture Gap Analysis

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Strategy Expert
2.4.1.A	Assess Alignment with Data Strategy Requirements		A	R		C
2.4.1.B	Evaluate Gaps Against the NPC Data Architecture Guidelines	I	A	R	C	
2.4.1.C	Evaluate Gaps Against General Best Practices	I	A	R	C	
2.4.1.D	Consolidate & Prioritize Data Architecture Gaps:	I	A/C	R		

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Identify Gaps in the Data Architecture

2.4.2 Analytical Data Model Gap Analysis

The Analytical Data Model Gap Analysis examines fact and dimension tables to identify missing attributes, modeling inconsistencies, and performance inefficiencies. It ensures data models follow best practices for structure, metadata, and optimization.

2.4.2.A
(1/2)

Assess Alignment with Data Strategy Requirements. The first step in the Analytical Data Model Gap Analysis is to assess whether the Entity's analytical data models (fact and dimension tables) support the requirements outlined in the Data Strategy. This ensures that the models enable business intelligence, reporting, and decision-making as envisioned in the strategy.

i. **Evaluate Analytical Data Model Alignment with the Data Strategy:** The Data Strategy defines analytical priorities, KPIs, and business intelligence use cases. The Entity must assess whether its fact and dimension models provide the necessary data to meet these needs.

a. **Assess Completeness of Analytical Models for Use Cases.** Key considerations include:

- Do the current fact and dimension tables contain all necessary attributes to support the Entity's key analytics and reporting needs?
- Are all required business measures (KPIs, calculations, aggregations) available in the data model?
- Are the dimension tables structured to support slicing and dicing data across required attributes?

b. **Validate Business Context & Relevance of Analytical Models.** Key considerations:

- Do the fact-dimension relationships correctly reflect Entity business processes?
- Are there redundant or missing tables that impact analytics efficiency?
- Is the level of granularity in fact tables appropriate for business decision-making?

ii. **Identify & Document Gaps:** The Entity must document where Data Strategy requirements are not met by analytical data models.

Asset Guidance

The **Analytical Data Model Gap Assessment Template** contains the Data Strategy Gaps Sheet

Illustrative

 Data Strategy Gaps				
Captures gaps between the analytical data model and the analytics needs outlined in the Data Strategy.				
Field	Description			
Gap ID	Unique gap reference.			
Gap Description	A concise statement of the issue or deficiency identified.			
Gap Category	Classification of the gap based on what it affects (e.g., Initiative, Use Case, or KPI).			
Business Requirement	Description of the business requirement outlined in the data strategy (e.g., details supporting an Initiative, Use Case, or KPI).			
Impact on Analytics/Strategy	How the gap limits analytics, KPI tracking, strategic execution, or decision-making.			
Suggested Remediation	High-level recommendation to resolve the issue (e.g., add fields, enrich metadata, create relationship).			
Priority	Severity or urgency of the gap: High / Medium / Low.			
Owner / Reviewer	Role or team responsible for remediation or validation.			
Status	Lifecycle status of the gap: Open / In Progress / Resolved.			
Date Logged	Date when the gap was recorded.			

Gap ID	Gap Description	Gap Category	Business Requirement	Impact on Analytics/Strategy

Suggested Remediation	Priority	Owner / Reviewer	Status	Date Logged

2.4.2.A
(2/2)

iii. Finalize Findings & Prepare for Further Analysis:

- Summarize misalignments with the Data Strategy and categorize them by impact level.
- Store findings for integration with further assessments (2.4.2.B, 2.4.2.C, etc.).
- Ensure documentation is ready for remediation planning in 2.5 (Data Architecture Roadmap).

2.4.2.B
(1/2)

Evaluate Gaps Against the NPC Data Architecture & Modeling Handbook. This step assesses how well the Entity's analytical data models (fact and dimension tables) align with the NPC Data Architecture & Modeling Handbook. The evaluation focuses on compliance with NPC's prescribed modeling methodologies, best practices for conceptual, logical, and physical data models, and recommended structuring of analytical models.

- Assess Compliance with NPC Data Modeling Guidelines:** The NPC Data Architecture & Modeling Handbook outlines best practices for conceptual, logical, and physical modeling. The Entity must compare its current analytical data models against the following guidelines:
 - Conceptual Data Model Guidelines
 - Logical Data Model Guidelines
 - Physical Data Model Guidelines



Key Message / Information

The gap assessment needs to be conducted across all data model components and dimensions for the respective data models (conceptual, logical, physical), as seen below.



Asset Guidance

The Entity can leverage the **Data Architecture & Data Modeling Handbook** to evaluate gaps in the Conceptual, Logical, and Physical Data Models.

Illustrative

		Dimensions				
		Content	Layout	Naming	Definition	Conformance
Data Model Components	Domain	What should a domain data model include? ✓ ✓ ✓	How should a domain data model look like? ✓ ✓ ✓	How should a domain data model be called? ✓ ✓ ✓	How should a domain data model be defined? ✓ ✓ ✓	How should a domain data model be to align with others? ✓ ✓ ✓
	Subject Areas	What should a subject area include? ✓ ✓ ✓	How should a subject area look like? ✓ ✓ ✓	How should a subject area be called? ✓ ✓ ✓	How should a subject area be defined? ✓ ✓ ✓	How should a subject area be to align with others? ✓ ✓ ✓
	Datasets	What should a dataset include? ✓ ✓ ✓	How should a dataset look like? ✓ ✓ ✓	How should an entity be called? ✓ ✓ ✓	How should a dataset be defined? ✓ ✓ ✓	How should a dataset be to align with all the others? ✓ ✓ ✓
	Relationships	What should a relationship include? ✓ ✓ ✓	How should a relationship look like? ✓ ✓ ✓	How should a relationship be called? ✓	N/A	How should a relationship be to align with all the others? ✓ ✓ ✓
	Keys	What should a key include? ✓ ✓	How should a key look like? ✓ ✓	How should a key be called? ✓ ✓	N/A	How should a key be to align with all the others? ✓ ✓
	Attributes	What should an attribute include? ✓ ✓	How should an attribute look like? ✓ ✓	How should an attribute be called? ✓ ✓	How should an attribute be defined? ✓ ✓	How should an attribute be to align with all the others? ✓ ✓

✓ CDM ✓ LDM ✓ PDM

2.4.2.B
(2/2) **Asset Guidance***Illustrative***Conceptual Data Model Guidelines (1/7)**

The guidelines for each of the components' dimensions of the conceptual data model are identified below

1 DOMAIN DATA MODEL

A	Content	- Each domain includes a single primary conceptual data model. - The data model includes a clear, comprehensive, and specific definition. - A title block is present on the data model, detailing the model title, version, author, and creation/modification dates. - The data model encompasses multiple subject areas relevant to a specific domain. - The data model reflects a business-oriented view of datasets and their relationships within that domain. - The data model should not exceed 200 datasets.
B	Layout	- The data model is depicted in a well-organized diagram, using rectangles and dashed lines in a dimensional star schema. - The entire data model is fully visible when viewed at 100% zoom. - The data model is designed to be intuitive and understandable with minimal training required. - Font size in the data model is readable, and drawing objects follow a uniform style.
C	Naming	- The data model follows the CDM naming convention for domain schema names, formatted as '[Entity Initials] [Reference Number] [Version] Conceptual Data Model'. - The name of the data model is singular, with spaces between each word. - Each word in the data model's name begins with a capital letter. - The data model's name is free from numbers and special characters.
D	Definition	- The definition of the data model must comprehensively describe the domain in business terms and give an overview of the data included. - The definition should avoid stating the obvious. - The definition is written in singular form and is free of spelling errors. - The definition complies with standard practices for definitions.

- ii. **Identify & Document Gaps.** The Entity must compare its analytical models against NPC guidelines and document any non-compliance.

 Asset Guidance

The **Analytical Data Model Gap Assessment Template** contains the Data Architecture & Modeling Guidelines Compliance Sheet.

Illustrative

 Data Architecture & Modeling Guidelines Compliance							
Assesses alignment of analytical models with NPC Data Architecture & Modeling Handbook (Conceptual, Logical, Physical).							
Field	Description						
Data Model ID	Unique identifier assigned to the data model where the gap was identified						
Data Model Type	The model layer where the gap occurs: Conceptual / Logical / Physical						
Data Model Name	Name of the data model						
Data Model Description	Brief description of the purpose or content of the data model						
Component Type	What is being evaluated (e.g., Domain, Dataset, Attribute, Relationship, Key)						
Dimension	Which guideline dimension is breached (e.g., Domain, Subject Areas, Datasets, Relationships, Keys, Attributes)						
Table Name (if Applicable)	The specific table affected by the gap, if relevant.						
Attribute (if Applicable)	The specific attribute where the gap is observed, if applicable.						
Guideline Reference	Reference or excerpt from the NPC Handbook describing the expected standard						
Gap ID	Unique reference for the identified modeling gap						
Gap Description	Explanation of how the actual implementation deviates from the guideline						
Impact	How the non-compliance affects usability, maintainability, integration, or consistency (High / Medium / Low)						
Suggested Remediation	Compliance action needed to resolve the issue						
Priority	High / Medium / Low						
Status	Open / In Progress / Resolved						
Date Logged	Date when the gap was recorded						

Data Model ID	Data Model Type	Data Model Name	Data Model Description	Component Type	Dimension	Table Name (if Applicable)	Attribute (if Applicable)

Guideline Reference	Gap ID	Gap Description	Impact	Suggested Remediation	Priority	Status	Date Logged

- iii. **Finalize Findings & Prepare for Next Steps:**

- Summarize key modeling gaps and categorize them by impact level.
- Store findings for integration with further assessments.
- Ensure documentation is ready for remediation planning (DAM Roadmap).

2.4.2.C
(1/2)

Evaluate Gaps in Data Model Integrity & Consistency. This step assesses the technical integrity of fact and dimension tables, identifying missing relationships, incomplete records, referential integrity issues, and inconsistencies in stored data. Unlike 2.4.2.B, which checks compliance with NPC modeling guidelines, this step focuses on structural and integrity problems that impact analytics and reporting accuracy.

- i. **Assess Data Model Integrity & Consistency in Analytical Models:** The Entity must review fact and dimension tables to identify issues that could lead to inaccurate reports, failed joins, or inconsistent aggregations. This includes but is not limited to:

- a. **Referential Integrity & Relationship Consistency.** Key considerations include:

- Are foreign keys properly enforced, ensuring all fact records have matching dimension entries?
- Do fact-dimension relationships maintain referential integrity, or are there orphaned records?
- Are duplicate or redundant records present, impacting accuracy in aggregations?

Example Gap:

- **Issue:** Some transactions in fact_sales do not have matching entries in dim_customer due to missing foreign key constraints.
- **Impact:** Sales reports show unmatched revenue figures.

- b. **Attribute Completeness & Standardization.** Key considerations include:

- Are all required attributes populated, or are there excessive null values?
- Are fact tables strictly numerical and dimension tables strictly descriptive?
- Are data types consistent across all tables (e.g., date formats, numeric precision, currency standardization)?

Example Gap:

- **Issue:** The "Revenue" column in fact_financials is sometimes recorded in USD and sometimes in local currency without a conversion flag.
- **Impact:** Analysts cannot accurately compare revenue across different financial reports.

- c. **Time & Date Consistency for Historical Reporting.** Key considerations include:

- Do fact tables correctly link to a time dimension to allow historical trend analysis?
- Are timestamp formats consistent, avoiding issues with time zone adjustments?
- Are data refresh cycles properly aligned across all analytical models?

Example Gap:

- **Issue:** The fact_transactions table has a mix of foreign and local timestamps, leading to misalignment in time-based aggregations.
- **Impact:** BI dashboards show incorrect daily revenue totals due to duplicate time entries.

2.4.2.C
(2/2)

- ii. **Identify & Document Gaps.** The Entity must document technical inconsistencies that impact data quality and integrity.

**Asset Guidance***Illustrative*

The **Analytical Data Model Gap Assessment Template** contains the Data Model Integrity & Consistency Sheet.

**Data Model Integrity & Consistency Gaps**

Identifies structural and integrity issues that may affect reporting accuracy, such as referential integrity, completeness, and timestamp consistency.

Field	Description
Data Model ID	Unique identifier assigned to the data model where the gap was identified.
Data Model Type	The model layer where the gap occurs: Conceptual / Logical / Physical.
Data Model Name	Name of the data model.
Data Model Description	Brief description of the purpose or content of the data model.
Gap ID	Unique reference for the identified gap (e.g., INT-GAP-001).
Gap Issue Type	Type of integrity issue (e.g., Foreign Key Violation, Orphan Record, Null Values, Timestamp Inconsistency).
Gap Description	Explanation of the structural inconsistency or integrity issue.
Table Affected	Name of the fact or dimension table where the issue exists.
Impact on Analytics	How the issue affects joins, aggregations, reporting accuracy, or trend analysis.
Example/Evidence	Specific example or observation demonstrating the issue.
Suggested Remediation	Recommended technical fix (e.g., enforce FK, standardize timestamp, deduplicate data).
Priority	High / Medium / Low.
Status	Open / In Progress / Resolved.
Date Logged	Date when the issue was recorded.

Data Model ID	Data Model Type	Data Model Name	Data Model Description	Gap ID	Gap Issue Type	Gap Description

Table Affected	Impact on Analytics	Example/Evidence	Suggested Remediation	Priority	Status	Date Logged

- iii. **Finalize Findings & Prepare for Next Steps:**

- Summarize key data quality gaps and categorize them by impact level.
- Store findings for integration with further assessments.
- Ensure documentation is ready for remediation planning (DAM Roadmap)

2.4.2.D
(1/3)

Analyze Optimization & Performance Gaps: This step assesses whether the Entity's analytical data models (fact and dimension tables) are optimized for performance, scalability, and efficient querying. The focus is on identifying bottlenecks, inefficient indexing, poor partitioning strategies, and query execution issues that impact analytical workloads, reporting speed, and overall system efficiency.

- i. **Assess Performance & Optimization of Analytical Models:** The Entity must evaluate how well fact and dimension tables are structured to support fast, efficient, and scalable analytics.

- a. **Indexing & Query Performance Optimization.** Key considerations include:

- Are fact tables indexed efficiently to optimize join operations and aggregations?
- Are dimension tables indexed on primary and foreign keys to speed up lookups?
- Are frequently queried columns optimized with composite or bitmap indexes?

Example Gap:

- **Issue:** The fact_sales table lacks indexing on key attributes (date, product_id, store_id).
- **Impact:** Query performance is slow when filtering by multiple dimensions, increasing report generation time.

2.4.2.D
(2/3)**b. Partitioning & Data Distribution.** Key considerations include:

- Are fact tables partitioned effectively (e.g., by date, region, department) to reduce query scan times?
- Is partition pruning working correctly, ensuring queries scan only relevant partitions?
- Are large tables distributed across storage systems efficiently to handle high volumes of analytical queries?

Example Gap:

- **Issue:** The fact_financials table is not partitioned, leading to full table scans for each query.
- **Impact:** Increases query execution time, affecting performance of reports.

c. Storage Efficiency & Data Redundancy. Key considerations include:

- Are fact tables storing only necessary historical data, or is excessive data retention slowing queries?
- Are duplicate or redundant tables increasing storage costs and query complexity?
- Does the Entity use columnar storage for analytical workloads where applicable?

Example Gap:

- **Issue:** The fact_transactions table retains over 10 years of raw transaction data, but only last 3 years are actively used.
- **Impact:** Unoptimized queries take longer to run.

ii. **Identify & Document Gaps:** The Entity must document performance and optimization inefficiencies in its analytical data models.

Asset Guidance*Illustrative*

The **Analytical Data Model Gap Assessment Template** contains the Optimization & Performance Assessment Sheet.

**Optimization & Performance Assessment**

Evaluates whether data models are optimized for query performance, storage efficiency, and scalability.

Field	Description
Data Model ID	Unique identifier assigned to the data model where the gap was identified.
Data Model Type	The model layer where the gap occurs: Conceptual / Logical / Physical.
Data Model Name	Name of the data model.
Data Model Description	Brief description of the purpose or content of the data model.
Gap ID	Unique gap reference (e.g., PERF-GAP-001).
Performance Issue Type	Type of performance gap (e.g., Indexing, Partitioning, Query Efficiency, Storage Redundancy).
Gap Description	Detailed description of the performance or scalability issue.
Table(s) Affected	Name of the fact or dimension table(s) experiencing performance or efficiency issues.
Impact on Analytics	How the issue affects query speed, reporting time, scalability, or system efficiency.
Example/Evidence	Observed behavior, query logs, or technical evidence demonstrating the issue.
Suggested Optimization	Recommended action to resolve the issue.
Priority	High / Medium / Low.
Status	Open / In Progress / Resolved.
Date Logged	Date when the performance issue was identified.

Data Model ID	Data Model Type	Data Model Name	Data Model Description	Gap ID	Performance Issue Type	Gap Description

Table(s) Affected	Impact on Analytics	Example/Evidence	Suggested Optimization	Priority	Status	Date Logged

2.4.2.D
(3/3)

iii. Finalize Findings & Prepare for Next Steps:

- Summarize key performance gaps and categorize them by impact level.
- Store findings for integration with further assessments (2.4.2.E).
- Ensure documentation is ready for remediation planning in 2.5 (Data Architecture Roadmap).

2.4.2.E
(1/2)

Consolidate & Prioritize Analytical Data Model Gaps: After assessing analytical data model gaps in alignment with the Data Strategy, NPC's Modeling Handbook, Data Quality & Consistency, and Performance Optimization, the Entity must now summarize key findings, categorize gaps by impact, and prepare insights for remediation planning.

i. **Summarize Key Findings from the Analytical Data Model Gap Analysis.** The assessment identified gaps in the following areas:

- Alignment with Data Strategy
- Compliance with NPC Modeling Standards
- Structural Integrity & Consistency Issues
- Optimization & Performance Gaps

ii. **Prioritize Gaps for Remediation:**

- High-priority gaps directly affect decision-making, reporting, and data accuracy and require immediate action.
- Medium-priority gaps impact governance, model standardization, and efficiency but do not disrupt core analytics.
- Low-priority gaps are long-term improvements that enhance future scalability and optimization.

**Asset Guidance***Illustrative*

The Entity can leverage the **Gap Analysis Key Findings Template** summarize all identified gaps and categorize them based on impact, and urgency.

Gap Analysis Key Findings Template Slide*Illustrative*

Gap Title	Gap Category	Description of Gap	Gap Impact	Gap Priority
Limited Scalability & Lack of Standardized Data Interoperability	Best Practices	• The current data architecture does not effectively support scalability for future data growth, and interoperability issues exist between internal and external systems due to inconsistent integration standards.	• Scalability limitations may lead to performance bottlenecks, and increased storage costs, while lack of interoperability hinders seamless data exchange with external entities.	Medium
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]
[Insert Gap Title]	[Insert Category]	• [Insert Description]	• [Insert Impact]	[insert]

Priority High Medium Low

2.4.2.E
(2/2)

ii. Finalize Documentation & Prepare for Remediation Planning:

- Findings are documented and validated with key stakeholders.
- Gaps are structured into categories for easier remediation.
- The final prioritized list of analytical data model gaps will feed into 2.5 (Data Architecture Roadmap) for action planning.

RACI Matrix – Analytical Data Model Gap Analysis

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Strategy Expert	Data Engineer
2.4.2.A	Assess Alignment with Data Strategy Requirements		A	R		C	
2.4.2.B	Evaluate Gaps Against the NPC Data Architecture & Modeling Handbook	I	A	R	C		
2.4.2.C	Evaluate Gaps in Data Model Integrity & Consistency	I	A	R	C		
2.4.2.D	Analyze Optimization & Performance Gaps	I	A	R			C
2.4.2.E	Consolidate & Prioritize Analytical Data Model Gaps	I	A/C	R			

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Identify Gaps in the Current Analytical Data Model

2.5 Data Architecture Roadmap

The Data Architecture Roadmap provides a structured plan for addressing identified gaps and implementing both long-term initiatives and quick wins to enhance the Entity's data architecture and modeling capabilities. The Entity must first develop a Data Architecture & Modeling Plan, which will then serve as the foundation for creating the Data Architecture Roadmap.

Process – Data Architecture Plan & Roadmap

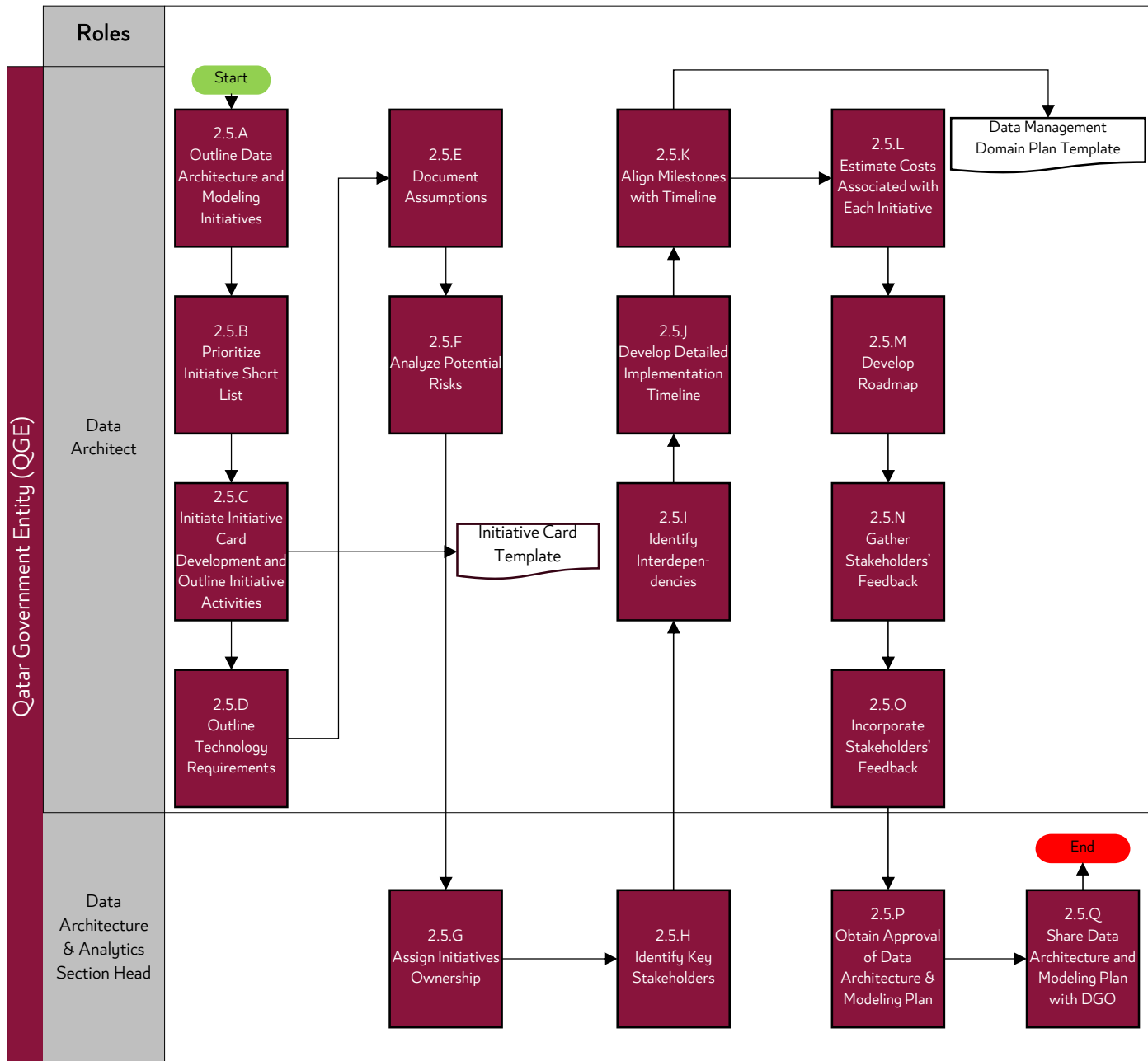


Figure 2: Data Architecture Plan & Roadmap Process Map

2.5.A

Outline Data Architecture & Modeling Initiatives: Identify and document a long list of data architecture and modeling initiatives, covering both short-term and long-term objectives, while ensuring that key gaps identified during the gap analysis are addressed. Initiatives may include improving data governance frameworks, standardizing data integration methods, and enhancing metadata management, as well as larger-scale projects such as establishing a scalable data architecture, optimizing analytical data models, and implementing advanced data lineage tracking.

Additionally, include initiatives specifically aimed at closing the identified gaps, ensuring that missing components, governance structures, and performance issues are resolved. This list should also include quick-win initiatives that can be implemented rapidly with minimal resources, such as optimizing existing data flows, refining data modeling documentation, or automating metadata tracking.

Ensure these initiatives are aligned with the Entity's broader data strategy to support efficient resource allocation and address critical data architecture and modeling needs.

2.5.B

Prioritize Initiative Short List: Use the long list of data architecture and modeling initiatives to identify and prioritize a short list based on potential impact and feasibility. Focus on high-impact areas where enhancements to data architecture, governance, and modeling standards will most significantly improve data accessibility, quality, and interoperability, supporting strategic initiatives and enterprise analytics goals.

Additionally, prioritize quick-win initiatives—projects that can be implemented quickly with minimal resources—to provide immediate benefits and demonstrate the value of structured data architecture improvements. Prioritization should consider business-critical gaps, long-term scalability, and alignment with governance frameworks, ensuring that initiatives are sequenced effectively for sustainable impact.

2.5.C

Initiate Initiative Card Development and Outline Initiative Activities: Develop initiative cards for each prioritized initiative outlined in the short list, capturing essential activities and steps required to guide implementation. For each initiative, outline the key activities by breaking them down into specific steps and sub-activities that need to be completed in sequence (where applicable). Include descriptions of each step to provide clear guidance on execution, along with any necessary details on tools, techniques, or methodologies that may be required.

Asset Guidance

An **Initiative Card Template** provides a structured format to document key information related to prioritized initiatives, such as initiative overview, key activities, sponsor, owner, stakeholders, timeline, key milestones, budget, technology requirements, dependencies, assumptions, risks, and milestones. The initiative cards provide a structured overview of proposed projects, allowing stakeholders to quickly understand each initiative's purpose, expected impact, and resource requirements.

Illustrative

OVERVIEW		SPONSOR	OWNER	STAKEHOLDERS
xxx		xxx	xxx	xxx
KEY ACTIVITIES				TIMELINE
1. Key Activity I. Sub-activity X - Detailed Steps II. Sub-activity Y - Detailed Steps 2. Key Activity I. Sub-activity X - Detailed Steps II. Sub-activity Y - Detailed Steps				BUDGET (IN QAR)
				xxx QAR xxx
				EXPECTED BENEFITS
				• xxx
				KEY MILESTONES
				A. xxx B. xxx C. xxx
				TECHNOLOGY REQUIREMENTS
				• xxx

OVERVIEW		SPONSOR	OWNER	STAKEHOLDERS																																																																																																									
xxx		xxx	xxxx	xxx																																																																																																									
KEY ACTIVITIES		TIMELINE																																																																																																											
<table border="1"><thead><tr><th>#</th><th>Milestones & Key Activities</th></tr></thead><tbody><tr><td>A.</td><td>Milestone</td></tr><tr><td>1.</td><td>Key Activity</td></tr><tr><td>B.</td><td>Milestone</td></tr><tr><td>2.</td><td>Key Activity</td></tr><tr><td>C.</td><td>Milestone</td></tr><tr><td>3.</td><td>Key Activity</td></tr></tbody></table>		#	Milestones & Key Activities	A.	Milestone	1.	Key Activity	B.	Milestone	2.	Key Activity	C.	Milestone	3.	Key Activity	<table border="1"><thead><tr><th>#</th><th>M1</th><th>M2</th><th>M3</th><th>M4</th><th>M5</th><th>M6</th><th>M7</th><th>M8</th><th>M9</th><th>M10</th><th>M11</th><th>M12</th></tr></thead><tbody><tr><td>A.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>B.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>C.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>3.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>			#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	A.													1.													B.													2.													C.													3.												
#	Milestones & Key Activities																																																																																																												
A.	Milestone																																																																																																												
1.	Key Activity																																																																																																												
B.	Milestone																																																																																																												
2.	Key Activity																																																																																																												
C.	Milestone																																																																																																												
3.	Key Activity																																																																																																												
#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12																																																																																																	
A.																																																																																																													
1.																																																																																																													
B.																																																																																																													
2.																																																																																																													
C.																																																																																																													
3.																																																																																																													
DEPENDENCIES		ASSUMPTIONS & RISKS																																																																																																											
• xxx		• xxx																																																																																																											

2.5.D

Outline Technology Requirements: Identify and document the technology requirements necessary to support each prioritized data architecture and modeling initiative. This includes outlining the required tools, software, and systems needed to ensure successful implementation and adoption across the Entity.

2.5.E

Document Assumptions: Record assumptions regarding resource availability, stakeholder engagement, and initiative complexity to ensure transparency and guide decision-making.

2.5.F

Analyze Potential Risks: Clearly identify and assess the risks associated with data architecture and modeling initiatives, ensuring that potential challenges are proactively addressed. This includes:

- Technical risks, such as compatibility issues between data architecture components, integration failures, or scalability limitations in data processing frameworks.
- Operational risks, such as lack of adoption of new data modeling standards, resistance to governance policies, or inadequate documentation leading to inconsistencies.
- Financial risks, such as budget overruns due to unforeseen complexity in system migrations, higher-than-expected costs for tool implementations, or extended project timelines.

2.5.G

Assign Initiatives Ownership: The Data Architecture & Analytics Section Head will be the owner of all data architecture and modeling initiatives. This designation ensures that the initiatives have a dedicated leader who can oversee progress and resolve roadblocks that may arise.

2.5.H

Identify Key Stakeholders: Recognize individuals and groups that have a vested interest in data architecture and modeling, including data architects, system owners, data engineers, and governance teams, facilitating collaboration and support. Understanding who needs to be engaged helps streamline communication and decision-making, improving the overall effectiveness of data architecture and modeling initiatives.

2.5.I

Identify Interdependencies: Analyze how data architecture and modeling initiatives may impact or be influenced by ongoing or planned initiatives to enhance resource allocation and coordination. This step involves:

- Reviewing how improvements in data architecture and modeling could affect or be affected by other enterprise data management efforts.
- Identifying resource overlaps, ensuring that the same teams, tools, or data sources are not overburdened or duplicated across multiple initiatives.
- Ensuring proper coordination between departments, preventing conflicts in timelines, dependencies, and implementation efforts.

2.5.J

Develop Detailed Implementation Timeline: Break down each activity in an initiative into actionable steps with start and end dates, ensuring the timeline is realistic and achievable. Each step should have a clear description of the required task, dependencies on other activities, and allocated resources. Ensure that the timeline accounts for potential delays or risks identified earlier, so adjustments can be made without derailing the project.

2.5.K

Align Milestones with Timeline: Set specific checkpoints or milestones throughout the implementation to assess progress. These milestones allow the evaluation of whether the initiative is proceeding as planned and provide opportunities for course corrections.

Asset Guidance

The **Data Management Domain Plan Template** helps the Entity capture an overview of each identified initiative, detail its sub-activities, owners, stakeholders, and activity timeframe.

Illustrative

Data Management Initiative Plan Template									
The following table is a template capturing a brief overview of the domain relevant initiatives									
Field	Description								
Initiative ID	A unique identifier for each initiative to facilitate tracking and reference								
Initiative Title	the name of the initiative								
Initiative Description	A short description of the initiative, outlining its objectives and expected outcomes								
Cost	Estimated financial resources required for the initiative, including both direct and indirect costs								
Start Date	The date when the initiative is scheduled to begin								
End Date	The date by which the initiative is expected to be completed								
Interdependencies	Any other identified initiatives or projects that this initiative depends on for successful execution								
Assumptions	Conditions assumed to be true for the initiative's success (e.g., resource availability)								
Risks	Potential risks that could affect the initiative's success								

Initiative ID	Initiative Title	Initiative Description	Cost	Start Date	End Date	Interdependencies	Assumptions	Risks

Initiative ID	Initiative Title	Sub-activities	Owner	Stakeholder(s)	Status	Activity Timeframes		Roadmap			
						Start Date	Due Date	0	1	2	3
								1 Jan	8 Jan	15 Jan	###
DMSG-001					In Progress	1 Jul	30 Aug				
DMSG-002					Completed	1 Jul	31 Jan				
DMSG-003					Not Started	30 Mar	31 Jul				

2.5.L

Estimate Costs Associated with Each Initiative: Consider expenses related to tools, technology, personnel, and training required for effective implementation. Break down costs based on the needs of each task, such as acquiring software, hiring additional staff, or providing necessary training for existing teams. This step ensures the budget is well-planned, realistic, and includes all critical aspects to enable smooth implementation of the initiatives.

2.5.M

Develop Roadmap: Create an overall roadmap that outlines timelines, phases, and key milestones for implementing the prioritized data architecture and modeling initiatives. Sequence the initiatives based on dependencies, resource availability, and alignment with organizational goals. The roadmap should also include projected timelines for each initiative, allowing stakeholders to monitor progress and allocate resources effectively over time.

2.5.N

Gather Stakeholders' Feedback: Present the draft plan to all relevant parties, inviting input and suggestions to enhance its effectiveness. This feedback process helps surface any potential issues or overlooked areas, improving the plan's effectiveness and ensuring all voices are heard. Engaging stakeholders early also builds buy-in and commitment to the plan's success.

2.5.O

Incorporate Stakeholders' Feedback: Actively gather and incorporate constructive feedback from key stakeholders to refine the plan and ensure it aligns with their needs and expectations. This collaborative approach helps to address concerns early, fine-tune the details, and strengthen the overall effectiveness of the plan.

2.5.P

Obtain Approval of Data Architecture and Modeling Plan: After incorporating all feedback, the Data Management and Security Director will be presented with the finalized Data Architecture and Modeling Plan for review and approval. This ensures alignment with the Entity's priorities and secures the necessary support for successful implementation. Once approved, the plan can be formally adopted and published.

2.5.Q

Share Data Architecture and Modeling Plan with DGO: Submit the finalized Data Architecture and Modeling Plan to the National Data Governance Office (DGO) for review. The DGO will review the plan and provide recommendations, allowing the Entity to ensure their plan aligns with the broader national data program agenda and best practices.

RACI Matrix - Data Architecture Plan & Roadmap

Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Architecture & Analytics Section Head	Data Governance Specialist	Data Architect	Data Stewards	Data Strategy Expert
2.5.A	Outline Data Architecture & Modeling Initiatives		A	C	C	R	C	C
2.5.B	Prioritize Initiative Short List		A	C	C	R	C	C
2.5.C	Initiate Initiative Card Development and Outline Initiative Activities			A	C	R		C
2.5.D	Outline Technology Requirements			A	C	R	C	C
2.5.E	Document Assumptions			A	C	R	C	C
2.5.F	Analyze Potential Risks			A	C	R		C
2.5.G	Assign Initiatives Ownership		A	R				
2.5.H	Identify Key Stakeholders		A	R	C	C	C	
2.5.I	Identify Interdependencies			A	C	R		C
2.5.J	Develop Detailed Implementation Timeline			A	C	R		C
2.5.K	Align Milestones with Timeline			A	C	R		C
2.5.L	Estimate Costs Associated with Each Initiative			A	C	R	C	C
2.5.M	Develop Roadmap			A	C	R		C
2.5.N	Gather Stakeholders' Feedback			A	C	R		
2.5.O	Incorporate Stakeholders' Feedback			A		R		C
2.5.P	Obtain Approval of Data Architecture and Modeling Plan		A	R				
2.5.Q	Share Data Architecture and Modeling Plan with DGO	C/I	A	R				

RACI Matrix Legend: R: Responsible A: Accountable C: Consulted I: Informed

Based on this Activity, the Entity can



Outline Data Architecture & Modeling Initiatives



Develop Data Architecture & Modeling Plan and Roadmap

3. Data Architecture & Model Design



3. Data Architecture & Model Design

What is Data Architecture & Model Design and why does the Entity need it?

Data Architecture & Model Design is a structured approach to defining the Target Data Architecture and developing an Analytical Data Model. Together, these components form the foundation for a secure, interoperable, and scalable data ecosystem that aligns with the national data standards, business objectives, and analytical requirements.

The Target Data Architecture establishes the future-state enterprise data ecosystem, outlining data sources and ingestion, storage and processing, integration and interoperability, governance and security, and data utilization and analytics. It also addresses performance optimization strategies to support scalable, high-performing data management. It ensures that data platforms, pipelines, and governance structures enable efficient data processing and seamless interoperability.

Following this, the Analytical Data Model provides a layered modeling structure - conceptual, logical, and physical - to organize data assets consistently across domains. The Analytical Data Model ensures consistent data representation, well-defined relationships between datasets, and compliance with data modeling best practices.

Finally, Data Model Metadata specifications are developed to manage model ownership, version control, and lifecycle governance. Metadata is documented and maintained in a centralized repository to ensure model traceability, compliance, and continuous evolution.

The remaining sections outline the step-by-step process to support execution of Data Architecture & Model Design:



Guiding Principles

Guiding Principles for Data Architecture & Model Design

1. **Leverage NPC's Reference Architecture:** Leverage NPC's reference architecture to guide the development of the Entity's data architecture.
2. **Design for Interoperability & Integration:** Structure the architecture to support seamless data exchange across systems, using standardized integration patterns.
3. **Adhere to Data Modeling Standards:** Ensure all data models comply with data architecture & modeling policies, industry best practices, and the Entity's and NPC data architecture & modeling guidelines to maintain consistency, interoperability, and scalability across the Entity's data ecosystem.
4. **Ensure Consistency Across Model Layers:** Align conceptual, logical, and physical data models to maintain a coherent and integrated structure. Consistency across model layers supports seamless data integration and clear data lineage.
5. **Design for Scalability & Performance:** Structure data models to support future growth, ensuring they can accommodate increasing data volumes, and performance demands for analytics and reporting.
6. **Ensure Comprehensive Metadata Management:** Manage metadata for all data models to enhance traceability, support version control, and enable effective data governance.

3.1 Target Data Architecture

The Entity must design a scalable, and interoperable Target Enterprise Analytics Data Architecture, ensuring alignment with business needs, technology advancements, and regulatory standards. This section outlines the key components, integration patterns, and governance structures required to establish a future-ready data architecture.

3.1.A
(1/2)

Define Target Data Architecture Principles & Objectives: The Target Data Architecture must be designed to support business objectives, address gaps identified in 2.4.1, and align with NPC's Reference Architecture and best practices. This step defines the guiding principles and objectives for the future-state architecture, ensuring it meets strategic, operational, and governance requirements.

- i. **Establish Core Principles for the Target Data Architecture:** The Entity must define key architectural principles that will guide the design, implementation, and management of the Target Data Architecture.
 - a. **Alignment with the Entity's Data Strategy:**
 - The architecture must support business intelligence, reporting, and advanced analytics use cases outlined in the Entity's Data Strategy.
 - It must enable efficient data sharing, integration, and governance across departments and respective external entities where and if required.



Key Message / Information

The design should accommodate newly identified data architecture requirements (if any) that emerged after the Data Strategy was developed.

- b. **Alignment with Key Existing Business Processes (Not Outlined in the Data Strategy):**
 - The Entity must determine which core business processes the Target Data Architecture must support and optimize, even if they were not explicitly outlined in the Data Strategy. These should include:
 - Mission-critical processes that depend on reliable data exchange.
 - Operational workflows that require optimization through better data integration and accessibility.
 - Processes that are expanding due to evolving regulatory, technological, or organizational needs.
- c. **Addressing Gaps Identified in 2.4.1 (Data Architecture Gap Analysis):**
 - The new architecture must resolve scalability, integration, governance, security, and metadata management issues.
 - Identified deficiencies in interoperability, performance optimization, and lineage tracking must be corrected.
- d. **Compliance with NPC's Reference Architecture (Data Architecture & Modeling Handbook):**
 - The Target Data Architecture must conform to NPC's recommended standards for data storage, processing, and governance.

3.1.A
(2/2)

- It must align with NPC-defined architectural layers (e.g., data ingestion, processing, storage, analytics, governance).

e. Scalability & Future-Readiness:

- The architecture must be modular, scalable, and adaptable to support future data growth, AI/ML adoption, and cloud transformation.
- It should leverage cloud-native, hybrid, and on-premise solutions as required.

ii. Define Target Data Architecture Objectives: The Entity must set clear objectives for its future data architecture to ensure it is efficient, secure, and aligned with national standards.

a. Data Integration & Interoperability:

- Standardize integration frameworks (ETL/ELT, APIs, event-driven data flows) to enable seamless data movement.
- Ensure cross-agency data sharing mechanisms are in place.



Key Message / Information

The data integration & interoperability aspects will be further elaborated in the Data Sharing, Integration, & Interoperability domain.

b. Security, Privacy & Compliance:

- Implement robust access control, encryption, and compliance tracking based on NPC guidelines.
- Ensure data classification, retention, and governance policies are embedded in the architecture.

c. Performance Optimization:

- Adopt efficient indexing, partitioning, and query acceleration techniques to enhance analytical workloads.

d. Data Governance & Metadata Management:

- Establish a centralized metadata repository to track data lineage, definitions, and quality metrics.
- Ensure data stewardship models and governance policies are integrated into architectural workflows.

iii. Finalize & Document Target Data Architecture Principles & Objectives:

- Summarize the key principles and objectives that will drive the new architecture design.
- Ensure alignment with NPC standards, business needs, and scalability requirements.
- Store documentation for reference in subsequent steps (3.1.B–3.1.F).

3.1.B
(1/3)

Describe Target Data Architecture Layers & Components: The Target Data Architecture must be structured into logical layers that define how data is ingested, processed, stored, governed, and consumed. This ensures a modular, scalable, and interoperable architecture in alignment with NPC's Reference Architecture.

- i. **Define the Layers of the Target Data Architecture:** The Target Data Architecture should follow the layered approach outlined in NPC's Reference Architecture, ensuring each layer has a clear function and responsibility.

a. **Data Sources & Ingestion Layer (Data Acquisition):**

- **Sources:** Includes transactional systems, external APIs, IoT data streams, operational databases, and legacy systems.
- **Ingestion Methods:** Supports batch ETL/ELT, API-based real-time ingestion, and event-driven streaming via an Event Broker.
- **Key Considerations:**
 - Enable structured, semi-structured, and unstructured data ingestion.
 - Support metadata validation and schema enforcement at ingestion.

b. **Data Storage & Processing Layer (Transformation Layer):**

- **Storage Components:**
 - Raw Data Repository (Landing zone for all incoming data).
 - Bronze Layer (Initial data staging and cleansing).
 - Silver Layer (Data enrichment and transformation).
 - Gold Layer (Curated, optimized data for analytics).
 - Master & Reference Data Models (Standardized data definitions).
 - Data Quality Controls (Ensuring consistency before storage).
- **Processing Frameworks:**
 - Supports batch processing, real-time event processing, and distributed compute workloads (if required).
- **Key Considerations:**
 - Ensure data standardization & harmonization across layers.
 - Implement automated data quality checks at each stage.

c. **Data Integration & Interoperability Layer (Data Delivery & Exchange Mechanisms):**

- **Integration Mechanisms:**
 - APIs and Microservices for interdepartmental and cross-entity data exchange.
 - Streaming and Message Queues (Event Brokers, Kafka) for event-driven architectures.
 - ETL/ELT Pipelines for structured batch data movement.
- **Key Considerations:**
 - Standardize data exchange mechanisms and interoperability protocols.
 - Enable low-latency data sharing for real-time analytics.

3.1.B
(2/3)

d. **Data Governance & Security Layer (Metadata, Orchestration, and Control):**

- **Governance Components:**
 - Metadata Repository for lineage, definitions, and quality tracking.
 - Master Data Management to standardize entity definitions.
- **Security Measures:**
 - Role-based access control (RBAC), encryption, and logging mechanisms.
- **Key Considerations:**
 - Ensure compliance with NPC data security policies.
 - Implement automated governance controls for lineage tracking and data workflows.

d. **Data Utilization & Analytics Layer:**

- **Consumption Methods:**
 - BI Dashboards, AI/ML Models, Reports, National Data Portals.
- **Consumption Methods:**
 - Ensure data availability for real-time analytics and machine learning.
 - Implement a semantic layer to standardize business definitions across analytical platforms.

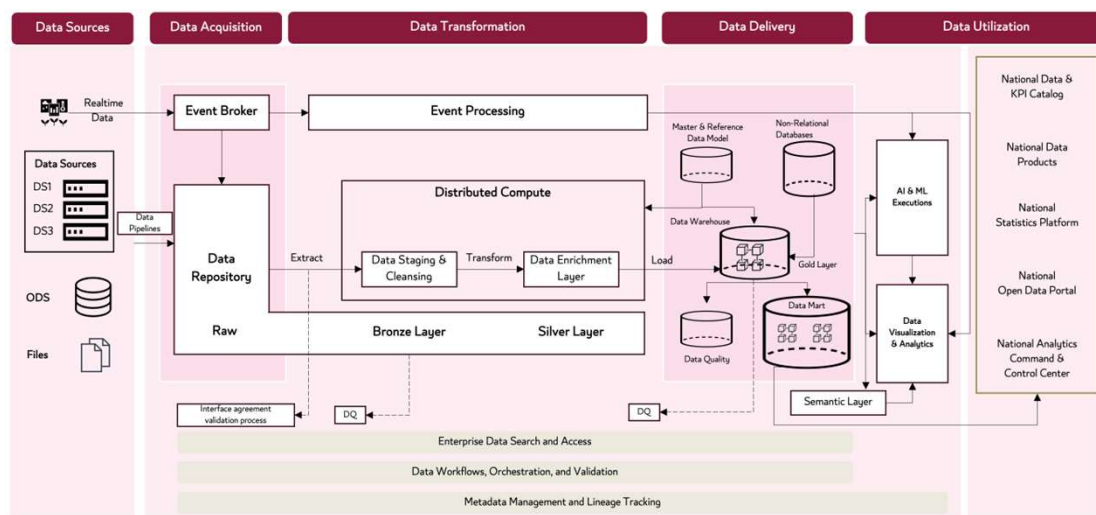
ii. **Ensure Alignment with NPC's Reference Architecture:** The Target Data Architecture must align with NPC's recommended layered model, including:

- Structured data flow from Data Acquisition to Utilization.
- Tiered processing model (Bronze → Silver → Gold layers).
- Event-driven & real-time processing where applicable.

 Asset Guidance

NPC's Reference Architecture can be found in the **Data Architecture & Data Modeling Handbook**.

Illustrative



3.1.B
(3/3)

iii. Finalize Documentation & Store for Next Steps:

- Document the defined layers and components.

Asset Guidance

The Entity can leverage the Architecture Components Template to document the defined layers and components *Illustrative*

Architecture Components			
The following table can be used to document each architecture component			
Field	Description		
Component ID	A unique identifier assigned to the architecture component (e.g., ARCH-001).		
Architecture Component	The high-level classification of the component (e.g., Data Warehouse, ETL Pipeline, API Gateway, Data Lake, Reporting Layer).		
Description	A brief overview of the component's purpose, functionality, and its role in the Enterprise Analytics Data Architecture.		
Dependency/ Flow of information	Specifies how the component interacts with other components, detailing input sources, data transformations, and output destinations (e.g., Data Warehouse receives structured data from ETL, serves reports via BI tools).		
Component ID	Architecture Components	Description	Dependency/ Flow of information

- Ensure that all architectural elements are aligned with business needs and governance policies.

3.1.C
(1/2)

Define Target Integration Approach & Document Integration Points: The Target Data Architecture must ensure seamless integration between source systems, data platforms, and analytical tools. This step defines the target integration approach and documents integration points using the Integration Points Documentation Template.

i. Define the Target Integration Approach: The Entity must establish how data will be integrated and exchanged across systems in the target architecture.

a. Integration Mechanisms:

- **Batch Processing (ETL/ELT):** For structured data movements at scheduled intervals.
- **API-Based Integrations:** For real-time, request-response data exchange.
- **Event-Driven & Streaming Integrations:** Using event brokers (e.g., Kafka) for near real-time data movement.

b. Standardization of Data Exchange:

- Define standard data formats (JSON, XML, Parquet, CSV) for interoperability.
- Establish governance for API versioning, data contracts, and metadata sharing.
- Ensure integration follows NPC-recommended interoperability guidelines (as outlined in the Data Sharing, Integration, and Interoperability domain).

c. Security & Compliance in Integration:

- Enforce authentication & authorization (e.g., API Keys, RBAC).
- Implement encryption & secure transport mechanisms (e.g., private endpoints).
- Align integration workflows with the required data security & governance frameworks.

3.1.C
(2/2)

ii. Document Integration Points:

Asset Guidance

The Entity can leverage the **Integration Points Documentation Template** to capture key details about system integrations, data exchange mechanisms, and governance tracking

Illustrative

Integration Points Documentation Template						
This template captures key details about system integrations, data exchange mechanisms, and governance tracking to ensure visibility, and operational efficiency.						
Field	Description					
Integration Name	The name of the integration process, describing the data exchange between source and destination systems.					
Source System	The system providing the data, responsible for originating the information being transferred.					
Destination System	The system receiving the data, where the transferred data will be processed, stored, or consumed.					
Integration Type	The type of integration approach used, specifying how data is exchanged between systems.					
Frequency	Defines how often the integration occurs, ensuring alignment with business requirements and data freshness.					
Data Transfer Mechanism	The protocol or method used for transferring data, ensuring secure and efficient data exchange.					
Data Sensitivity Level	Specifies whether the transferred data is Top Secret, Secret, Restricted, Internal, Public					
Owner/Responsible Party	Identifies the team or department managing the integration and responsible for monitoring, troubleshooting, and governance.					
Error Handling Mechanism	Defines how integration failures, errors, or exceptions are handled, ensuring resilience and automated recovery workflows.					
Data Volume	The approximate size of data transferred per integration cycle, helping plan for scalability and performance.					
Latency Expectation	Defines the expected time for data to be delivered and processed, ensuring performance meets business and operational needs.					
Logging & Monitoring	Specifies whether logs are maintained for tracking integration events, errors, and performance metrics.					
Security & Compliance Measures	Details encryption standards, authentication mechanisms, and other security protocols used to protect data during					
Last Updated	The date when the integration details were last reviewed or modified.					

Integration Name	Source System	Destination System	Integration Type	Frequency	Data Transfer Mechanism	Data Sensitivity Level

Owner/Responsible Party	Error Handling Mechanism	Data Volume	Latency Expectation	Logging & Monitoring	Security & Compliance Measures	Last Updated

iii. Finalize Documentation & Prepare for Implementation:

- Document all integration points in a structured format.
- Store integration metadata in a centralized repository for tracking and governance.
- Ensure integration workflows are aligned with architectural components defined in 3.1.B (Target Data Architecture Layers & Components).

3.1.D

Adhere to Target Data Governance & Security Controls: The Target Data Architecture must adhere to governance and security controls to ensure proper access management, data protection, compliance, and monitoring. At a minimum, the Entity should consider the below key governance and security considerations to be embedded into the future-state architecture.

i. **Data Access & Usage Controls:**

- Role-based access control (RBAC) to regulate user permissions across data platforms.
- Restricted access to sensitive data based on classification levels (e.g., Internal, Restricted, Public).
- Data usage policies that govern who can query, modify, or share datasets.

ii. **Data Security & Protection Measures:**

- Data encryption for both data at rest and in transit.
- Data masking and anonymization techniques for handling personally identifiable information (PII) and classified data.
- Secure data transfer mechanisms are used across integrations.

iii. **Audit Logging & Monitoring Requirements**

- Logging of all data access, modifications, and transfers to track user activity.
- Real-time monitoring to detect anomalies in data usage.
- Logs are centrally stored and regularly reviewed for security audits.

iv. **Compliance with NPC & Regulatory Standards**

- The Entity's data security, privacy, and compliance requirements and the National Data Standards.
- Adherence to national data protection laws, privacy frameworks, and governance policies.
- Requirements for data classification, retention, and regulatory reporting.

3.1.E
(1/2)

Define Scalability & Performance Enhancements: The Target Data Architecture must be designed to scale efficiently and optimize performance, ensuring that it can handle increasing data volumes, evolving analytical workloads, and future business needs. This step defines the scalability and performance enhancements that will be incorporated into the future-state architecture. Sample considerations include but are not limited to:

i. **Scalability Considerations for the Target Data Architecture:** The Entity should ensure the architecture is modular, elastic, and capable of supporting future growth while maintaining high performance and availability.

a. **Data Storage & Processing Scalability:**

- Implement a tiered storage approach (hot, warm, cold storage) to optimize cost and retrieval speed.

3.1.E
(2/2)

- Ensure data partitioning strategies (e.g., by date, region, business unit) to distribute large datasets effectively.
- Design the architecture to support horizontal scaling (adding nodes to clusters) and vertical scaling (upgrading compute power).

b. Compute & Query Scalability:

- Utilize distributed computing frameworks (e.g., Spark) for large-scale data processing.
- Implement query optimization techniques such as materialized views, caching, and parallel processing.
- Ensure real-time and batch processing can run efficiently without resource contention.

ii. **Performance Enhancements for Optimized Workloads:** The Entity should implement performance optimizations to ensure that analytical workloads, data ingestion, and retrieval processes are fast and reliable.

a. Indexing & Partitioning for Query Optimization:

- Define indexing strategies for frequently queried columns in fact and dimension tables.
- Implement partition pruning to reduce the number of records scanned in analytical queries.
- Use columnar storage formats (e.g., Parquet) for optimized analytics workloads.

b. Efficient Data Ingestion & Transformation:

- Optimize data pipelines to ensure efficient data movement and transformation.
- Use incremental data loading instead of full refreshes to improve processing times.
- Implement streaming ingestion frameworks (Kafka, Kinesis) for real-time processing needs.

c. Query & Report Performance Optimization:

- Use semantic layers to pre-aggregate data and reduce query complexity.
- Implement BI tool optimizations such as in-memory processing and materialized views.
- Optimize dashboard load times by precomputing frequently used metrics.

iii. Document Enhancements & Prepare for Next Steps:

- Document the scalability and performance enhancements planned for the Target Data Architecture.
- Ensure alignment with integration, security, and governance requirements.

**Key Message / Information**

Since scalability and performance enhancements cover multiple aspects of the Target Data Architecture, the Entity must document these requirements in a format best suited to their needs.

3.1.F
(1/2)

Develop Target Data Architecture Diagram: The Target Data Architecture Diagram provides a visual representation of the Entity's future-state data ecosystem, ensuring alignment with NPC's Reference Architecture. This diagram must incorporate all key architectural layers, data movement processes, and governance components to enable scalable, efficient, and secure data management.

i. **Define the Structure of the Target Data Architecture Diagram.** The Target Data Architecture must follow NPC's High-Level Enterprise Analytics Data Architecture framework, ensuring that data flows seamlessly from acquisition to utilization. Key components and considerations include:

a. **Data Sources & Ingestion Layer (Data Acquisition):**

- **Sources:** Includes transactional databases, operational systems, IoT streams, external APIs, and file-based data.
- **Ingestion Mechanisms:**
 - Event Broker for streaming data (Kafka, RabbitMQ).
 - Data Pipelines for batch ingestion (ETL/ELT processes).
 - ODS (Operational Data Store) for near real-time access.

b. **Data Storage & Processing Layer (Transformation Layer):**

- **Storage Tiers:**
 - **Raw Data Repository (Landing zone for all incoming data).**
 - **Bronze Layer:** Initial data staging and cleansing.
 - **Silver Layer:** Data enrichment and transformation.
 - **Gold Layer:** Curated, optimized data for analytics.
 - **Master & Reference Data Models:** Golden records.
- **Processing Frameworks:**
 - Distributed Compute Engines for large-scale transformations.
 - Data Staging & Cleansing Pipelines to ensure data quality before enrichment.
 - Data Enrichment Layer for contextualization and aggregation.

c. **Data Governance & Metadata Layer (Metadata Management & Lineage Tracking):**

- Metadata Repository to track schema evolution, definitions, and business rules.
- Data Quality Controls embedded within transformation and storage layers.
- Enterprise Data Search & Access Mechanisms for efficient data discovery and cataloging.

d. **Data utilization & Analytics Layer:**

- **Consumption Methods:**
 - BI Dashboards, AI/ML Execution Engines, Reporting, National Data Portals etc.
- **Semantic Layer:**
 - Standardizes business logic and KPI definitions across analytics platforms.
- **Governance & Security:**
 - Enforces data access policies, security classifications, and RBAC.

3.1.F
(2/2)

- ii. **Develop the Target Data Architecture Diagram.** The Entity must create a high-level Target Data Architecture Diagram using standard visualization tools (e.g., Visio, Draw.io) while ensuring:
- Alignment with NPC's Reference Architecture and its data flow structure.
 - Clear representation of layers, integration points, and governance frameworks.
 - Consistency with enhancements defined in previous steps (3.1.A–3.1.E).

3.1.G

Conduct Internal Review & Finalize Target Data Architecture: Once the Target Data Architecture Diagram is developed, the Entity must internally review and validate it to ensure alignment with business needs, technical feasibility, and governance requirements before engaging external stakeholders.

i. **Internal Review & Stakeholder Alignment:**

- Engage IT leadership, governance teams, and key business units to review the architecture.
- **Ensure it aligns with:**
 - Entity data strategy and operational goals.
 - Security, compliance, and integration requirements.
 - Scalability and performance objectives.

ii. **Address Internal Feedback & Finalize Architecture:**

- Gather input from internal stakeholders and adjust the architecture as needed.
- Ensure documentation is clear, version-controlled, and accessible.

3.1.H

Share with DGO & Store Final Architecture: After internal approval, the Entity must submit the finalized Target Data Architecture to NPC's National Data Governance Office (DGO) for review and alignment with national standards.

i. **Submit to DGO for Review & Feedback:**

- Provide the architecture diagram, integration details, and governance considerations.
- If applicable, incorporate DGO recommendations.

ii. **Store & Maintain the Final Architecture:**

- Save the validated architecture in a centralized repository for future reference.
- Ensure it is version-controlled.

RACI Matrix – Target Data Architecture

Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Engineer	Data Steward	Data Owner
3.1.A	Define Target Data Architecture Principles & Objectives		I	A/C	R	C			
3.1.B	Describe Target Data Architecture Layers & Components			A	R		C		
3.1.C	Define Target Integration Approach & Document Integration Points		I	A	R		C		
3.1.D	Adhere to Target Data Governance & Security Controls			A	R				
3.1.E	Define Scalability & Performance Enhancements		I	A/C	R				
3.1.F	Develop Target Data Architecture Diagram			A/C	R				
3.1.G	Conduct Internal Review & Finalize Target Data Architecture			A	R	C	C	C	C
3.1.H	Share with DGO & Store Final Architecture	C/I	I	A	R				

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Develop Target Data Architecture



Define Key System Components of the Target Data Architecture

3.2 Analytical Data Model

The Entity must develop a structured, and standardized Analytical Data Model to ensure consistency, integration, and reusability of data assets. This section outlines the design of conceptual, logical, and physical data models, ensuring alignment with the Entity's data strategy, business processes, analytics and reporting requirements, and the data architecture & modeling handbook.



Key Message / Information

Use a Data Modeling Tool to design the target data model and data lineage. If a tool is not yet in place, follow the steps below to manually design and develop the target data model and data lineage. Refer to activity 4.1 for guidance on selecting and setting up the data modeling tool.

3.2.A

Define Target Data Modeling Approach & Compliance Requirements: The Target Data Model establishes the future-state structure for analytical data, ensuring alignment with business needs, compliance with NPC guidelines, and resolution of previously identified gaps (2.4.2).

- i. **Ensure Alignment with Key Standards & Requirements:** The Target Data Model must follow NPC's prescribed modeling design principles while also addressing specific needs of the Entity.

- a. **Alignment with the Entity's Data Strategy:**

- The target model must support business intelligence, reporting, and use cases.

- b. **Alignment with Key Existing Business Processes (Not Outlined in Data Strategy):**

- Identify critical business processes that the data model must support, even if not explicitly covered in the Data Strategy.
- Ensure fact-dimension structures align with operational workflows and domain-specific reporting needs.

- c. **Addressing Gaps Identified in 2.4.2 (Analytical Data Model Gap Analysis):**

- The target model must resolve issues related to Integrity & Consistency (2.4.2.C) and Optimization and Performance (2.4.2.D).

- d. **Compliance with NPC's Data Architecture & Modeling Guidelines:**

- The target model must conform to NPC's data modeling guidelines and standards.

- ii. **Target Data Model Framework:** The Target Data Model must be structured across:

- a. **Conceptual Data Model:**

- Defines domains, subject areas, and high-level entities.
- Ensures alignment with business processes and analytical use cases.

- b. **Logical Data Model:**

- Includes entities, attributes, relationships, and primary/foreign keys.
- Establishes fact-dimension structures and hierarchies for analytics.

- c. **Physical Data Model:**

- Builds upon the logical data model and specifies tables, columns and data types.

3.2.B.I
(1/2)

Develop the Target Conceptual Data Model: The Target Conceptual Data Model provides a high-level representation of business entities, relationships, and subject areas without technical implementation details. It ensures that the data model aligns with analytical requirements, reporting needs, and governance standards, while remaining consistent with the modeling guidelines established in 2.2 and NPC's Data Architecture & Modeling Handbook.

i. Define the Structure of the Target Conceptual Data Model:

The Conceptual Data Model must:

- Represent business concepts and relationships without considering physical implementation.
- Define domains, subject areas, entities, and high-level relationships.
- Incorporate newly identified analytical requirements that were not outlined in the Data Strategy.

a. Identify Key Data Domains & Subject Areas:

- Group data into high-level domains (e.g., Citizen Services, Finance, HR).
- Define subject areas within each domain (e.g., Payroll under HR, Transactions under Finance).
- Ensure the model supports cross-functional analytics and enterprise reporting needs.

b. Define Core Business Entities & Relationships:

- Identify key business entities that are critical for analytical reporting.
- Establish high-level relationships between entities (e.g., Customers place Orders, Employees belong to Departments).
- Ensure the model captures key analytical requirements, even if they were not explicitly defined in the Data Strategy.

c. Incorporate Business & Analytical Requirements:

- The model must support business intelligence, reporting, and analytical use cases.
- Ensure the structure reflects real-world relationships required for multi-dimensional analytics and decision-making.
- Align with the Entity's Data Strategy, previously identified gaps (2.4.2), key analytical processes, and governance policies.

ii. Develop Conceptual Data Model:

a. Create a Visual Representation of the Conceptual Data Model:

- Develop an Entity-Relationship Diagram showing domains, entities, and relationships.
- Ensure alignment with NPC's conceptual modeling best practices

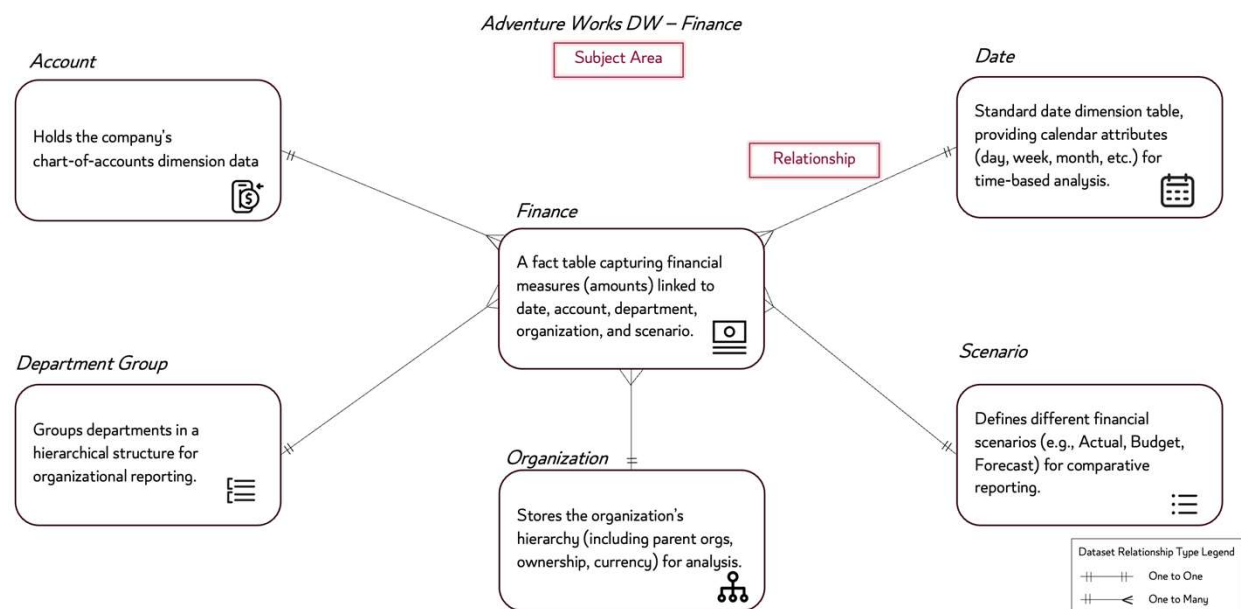
3.2.B.1
(2/2)

Asset Guidance

The **Data Architecture and Modeling Handbook** can be leveraged by entities to explore the conceptual data model, including guidelines across its components (Domain, Subject Areas, Datasets, Relationships) and dimensions (Content, Layout, Naming, Definition, Conformance). In addition to these guidelines, the handbook also provides a sample visual representation of a conceptual data model.

Illustrative

		Dimensions				
		Content	Layout	Naming	Definition	Conformance
Conceptual Data Model Components	Domain	What should a domain data model include?	How should a domain data model look like?	How should a domain data model be called?	How should a domain data model be defined?	How should a domain data model be to align with others?
	Subject Areas	What should a subject area include?	How should a subject area look like?	How should a subject area be called?	How should a subject area be defined?	How should a subject area be to align with others?
	Datasets	What should a dataset include?	How should a dataset look like?	How should an entity be called?	How should a dataset be defined?	How should a dataset be to align with all the others?
	Relationships	What should a relationship include?	How should a relationship look like?	How should a relationship be called?	N/A	How should a relationship be to align with all the others?



b. Document Entity Definitions & Business Relationships:

- Provide clear descriptions of each entity, subject area, and domain.
- Define relationships between key business entities in alignment with analytical use cases.

c. Ensure Standardization & Compliance:

- Apply naming conventions, definitions, and model conformance rules as per NPC's guidelines.
- Maintain consistency across subject areas to ensure reusability and governance.

3.2.B.II

Conduct Internal Review & Finalize Conceptual Data Model:**i. Internal Review & Stakeholder Alignment:**

- Engage business analysts, data architects, and governance teams to review the model.
- Ensure all required domains, entities, and relationships are covered.

ii. Address Feedback (If Any) & Finalize the Conceptual Data Model:

- Incorporate stakeholder feedback and validate business alignment.
- Ensure the model is consistent with and adheres to the Entity's and NPC's data architecture and modeling guidelines.

3.2.B.III

Share Conceptual Data Model Diagram and Supporting Documentation with DGO & Store in Data Model Repository:**i. Submit to DGO for Review & Feedback:**

- Share the finalized Conceptual Data Model and any supporting documentation with the Data Governance Office (DGO).
- Engage with DGO representatives to discuss the model's alignment with the national data standards, best practices, and NPC's Data Modeling Guidelines.
- Incorporate any feedback or recommendations from the DGO to ensure consistency and compliance with national regulations.

ii. Store & Maintain the Conceptual Data Model:

- Save the validated model in a centralized data model repository (as referenced in section 6).
- Ensure version control is applied to track changes, maintain governance oversight, and support future updates.
- Establish a structured model maintenance process to periodically review, refine, and update the model in line with evolving data governance requirements.
- Maintain a repository of supporting documentation, ensuring entities can easily access reference materials for data integration, analytics, and reporting needs.

3.2.C.I
(1/3)

Develop the Target Logical Data Model: The Target Logical Data Model provides a detailed representation of data entities, attributes, relationships, and constraints, ensuring consistency and analytical usability. It defines fact-dimension structures, hierarchies, and relationships while remaining independent of physical implementation details.

3.2.C.I
(2/3)

i. Define the Structure of the Target Logical Data Model:

The Logical Data Model must:

- Provide a structured view of analytical data including entities, attributes, primary/foreign keys, and relationships.
- Ensure fact-dimension relationships are well-defined to support analytics and reporting.
- Align with the Entity's and NPC's modeling guidelines and best practices.

a. Define Entities, Attributes & Keys:

- Identify fact tables and dimension tables.
- Define primary keys (PK) and foreign keys (FK) to enforce referential integrity.
- Ensure each entity contains the attributes required for reporting and analytical processing.

b. Establish Fact-Dimension Relationships:

- Define one-to-many or one-to-one relationships between fact and dimension tables.
- Introduce bridge tables for complex many-to-many relationships where required.
- Ensure relationships align with analytical queries and reporting structures.

c. Ensure Standardization & Compliance:

- Apply naming conventions, definitions, and modeling conformance rules as per the Entity's and NPC's guidelines.
- Maintain consistency across subject areas and fact-dimension structures for governance and usability.

ii. Develop the Logical Data Model:

a. Create a Visual Representation of the Logical Data Model:

- Develop an Entity-Relationship Diagram capturing entities, attributes, relationships, and keys.
- Ensure alignment with the NPC's logical modeling best practices.

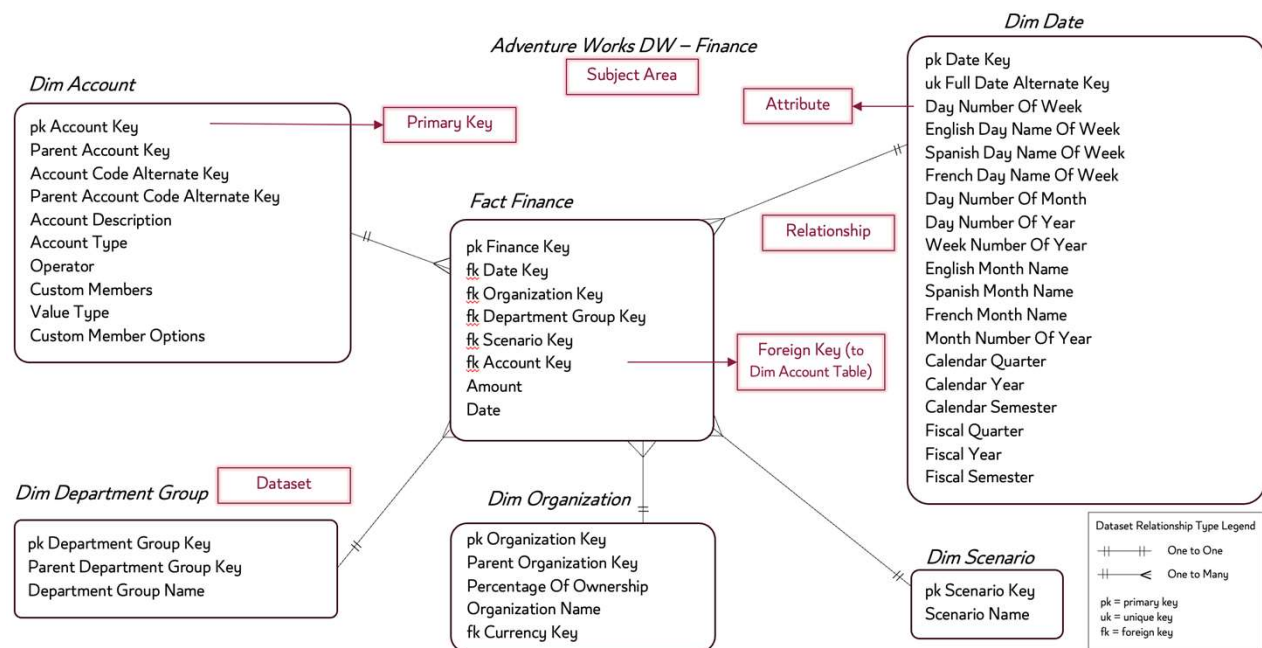
3.2.C.I
(3/3)

Asset Guidance

The Data Architecture and Modeling Handbook can be leveraged by entities to explore the logical data model, including guidelines across its components (Domain, Subject Areas, Datasets, Relationships, Keys, and Attributes) and dimensions (Content, Layout, Naming, Definition, Conformance). In addition to these guidelines, the handbook also provides a sample visual representation of a logical data model.

Illustrative

		Dimensions				
		Content	Layout	Naming	Definition	Conformance
Logical Data Model Components	Domain	What should a domain data model include?	How should a domain data model look like?	How should a domain data model be called?	How should a domain data model be defined?	How should a domain data model be to align with others?
	Subject Areas	What should a subject area include?	How should a subject area look like?	How should a subject area be called?	How should a subject area be defined?	How should a subject area be to align with others?
	Datasets	What should a dataset include?	How should a dataset look like?	How should an entity be called?	How should a dataset be defined?	How should a dataset be to align with all the others?
	Relationships	What should a relationship include?	How should a relationship look like?	N/A	N/A	How should a relationship be to align with all the others?
	Keys	What should a key include?	How should a key look like?	How should a key be called?	N/A	How should a key be to align with all the others?
	Attributes	What should an attribute include?	How should an attribute look like?	How should an attribute be called?	How should an attribute be defined?	How should an attribute be to align with all the others?



b. Document Entity Definitions & Business Relationships:

- Provide clear definitions for all entities and attributes within fact and dimension tables.
- Define relationships between key entities in alignment with analytical use cases and other reporting requirements.

3.2.C.II

*Conduct Internal Review & Finalize Logical Data Model:***i. Internal Review & Stakeholder Alignment:**

- Engage business data analysts, governance teams, and the data committee to review the model.
- Ensure all required fact tables, dimensions, and relationships are captured and validated.
- Conduct structured validation sessions with Business Domain Experts to confirm model completeness.
- Verify compliance with National Data Standards, ensuring that dimensional modeling methodologies (e.g., Star Schema) are correctly implemented.

ii. Address Feedback & Finalize the Model:

- Incorporate stakeholder feedback and validate logical integrity.
- Ensure the model meets NPC's guidelines and the Entity's analytical and reporting requirements.

3.2.C.III

*Share the Logical Data Model Diagram and Supporting Documentation with DGO & Store in Data Model Repository:***i. Submit to DGO for Review & Feedback:**

- Share the finalized Logical Data Model and any supporting documentation with the Data Governance Office (DGO).
- Engage with DGO representatives to discuss the model's alignment with the national data standards, best practices, and NPC's Data Modeling Guidelines.
- Incorporate any feedback or recommendations from the DGO to ensure consistency and compliance with national regulations.

ii. Store & Maintain the Logical Data Model:

- Save the validated model in a centralized data model repository (as referenced in section 6).
- Ensure version control is applied to track changes, maintain governance oversight, and support future updates.
- Establish a structured model maintenance process to periodically review, refine, and update the model in line with evolving data governance requirements.
- Maintain a repository of supporting documentation, ensuring entities can easily access reference materials for data integration, analytics, and reporting needs.

3.2.D.I
(1/2)

Develop the Target Physical Data Model: The Target Physical Data Model represents the structured design of the database schema, specifying tables, columns, data types, indexing strategies, and partitioning approaches while remaining independent of actual implementation. This step ensures the data model is optimized for performance, scalability, and compliance with the Entity's and NPC's modeling guidelines while aligning with the Logical Data Model (3.2.C).

i. **Define the Structure of the Target Physical Data Model:**

The Physical Data Model must:

- Translate the Logical Data Model into a detailed database design while remaining technology-agnostic.
- Specify tables, columns, data types, indexing strategies, and partitioning methods.
- Align with NPC's modeling guidelines and best practices for performance and governance.

a. **Define Tables, Columns & Data Types:**

- Establish the final table structures, including fact tables, dimension tables, and bridge tables.
- Assign data types for each column to ensure consistency and efficiency (e.g., VARCHAR(255), DECIMAL(10,2), DATETIME).

b. **Specify Indexing & Partitioning Strategies:**

- Define primary, foreign, and secondary indexes for query optimization.
- Identify partitioning strategies (e.g., range partitioning by date, hash partitioning for load balancing).

ii. **Develop the Physical Data Model Diagram:**

- Develop a visual representation of the database schema, including:
 - Tables with columns and data types.
 - Primary keys (PK) and foreign keys (FK) relationships.
 - Indexes and partitioning annotations where applicable.

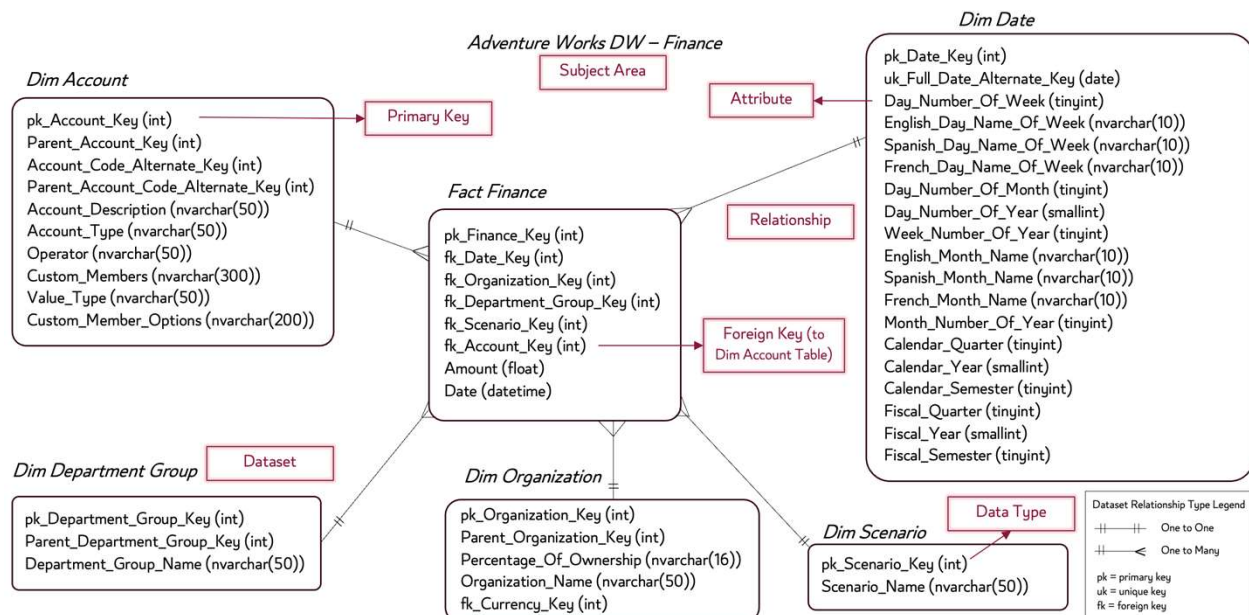
3.2.D.I
Cont.
(2/2) **Asset Guidance**

The Data Architecture and Modeling Handbook can be leveraged by entities to explore the physical data model, including guidelines across its components (Domain, Subject Areas, Datasets, Relationships, Keys, and Attributes) and dimensions (Content, Layout, Naming, Definition, Conformance).

Illustrative

Dimensions					
	Content	Layout	Naming	Definition	Conformance
Domain	What should a domain data model include?	How should a domain data model look like?	How should a domain data model be called?	How should a domain data model be defined?	How should a domain data model be to align with others?
Subject Areas	What should a subject area include?	How should a subject area look like?	How should a subject area be called?	How should a subject area be defined?	How should a subject area be to align with others?
Datasets	What should a dataset include?	How should a dataset look like?	How should an entity be called?	How should a dataset be defined?	How should a dataset be to align with all the others?
Relationships	What should a relationship include?	How should a relationship look like?	N/A	N/A	How should a relationship be to align with all the others?
Keys	What should a key include?	How should a key look like?	How should a key be called?	N/A	How should a key be to align with all the others?
Attributes	What should an attribute include?	How should an attribute look like?	How should an attribute be called?	How should an attribute be defined?	How should an attribute be to align with all the others?

In addition to these guidelines, the handbook also provides a sample visual representation of a physical data model.



3.2.D.II

Conduct Internal Review & Finalize Physical Data Model:

i. Internal Review & Stakeholder Alignment:

- Engage database architects, infrastructure teams, and governance leads to review the model and supporting documentation.
- Ensure tables, relationships, and performance considerations are fully captured.

ii. Address Feedback (If Any) & Finalize the Conceptual Data Model:

- Incorporate stakeholder feedback and validate database schema integrity.
- Ensure the model is consistent with the Entity's and NPC's guidelines and optimization requirements.

**Key Message / Information**

Regularly review and refine the conceptual, logical, and physical data models, adapting to evolving business needs, regulatory changes, and technological advancements

3.2.D.III

Share the Physical Data Model Diagram & Supporting Documentation with DGO & Store in Data Model Repository:

i. Submit to DGO for Review & Feedback:

- Share the finalized Physical Data Model Diagram and supporting documentation to the DGO.
- Include a summary of key details, such as fact and dimension structures, and compliance with NPC's Data Modeling Guidelines & Methodology.
- Engage with DGO representatives to discuss the model's alignment with the national data standards, best practices, and NPC's Data Modeling Guidelines.
- Incorporate DGO recommendations (if applicable) to ensure compliance.

ii. Store & Maintain the Physical Data Model:

- Save the validated Physical Data Model in a centralized Data Model Repository (as referenced in section 6) for future reference.
- Ensure version control is applied to track changes, maintain governance oversight, and support future updates.
- Establish a structured model maintenance process to periodically review, refine, and update the model in line with evolving data governance requirements.
- Maintain a repository of supporting documentation, ensuring entities can easily access reference materials for data integration, analytics, and reporting needs.

3.2.E.I
(1/3)

Define Target Data Lineage: The Target Data Lineage outlines how data attributes move, transform, and are consumed across the analytical landscape. The Entity will document lineage using the Data Lineage Template to ensure full traceability from source to report.

i. Define the Scope of Data Lineage Tracking:

The Target Data Lineage must:

- Map the complete flow of data attributes from source systems -> raw layer -> bronze layer -> silver layer -> gold layer -> to reports.
- Document transformations, aggregations, and dependencies for fact and dimension attributes.

a. Capture Lineage for Fact & Dimension Tables:

- Track source attributes feeding into fact and dimension tables.
- Document calculated fields, aggregations, and derived measures.
- Ensure alignment with business definitions in the Entity's data catalog, as outlined in the Data Catalog and Metadata Management domain.

b. Fill Out the Data Lineage Template to track:

- **Data Sources & Descriptions:** The system(s) providing the data and type it holds.
- **Source & Target Datasets / Tables and Attributes:** Specific source and target datasets / tables and attributes undergoing transformation.
- **Data Transformation Logic:** Any calculations, filtering, or processing rules applied.
- **Layer:** The stage (Raw, Bronze, Silver, or Gold) where the transformations occur.

 Asset Guidance

Illustrative

The Entity can leverage the **Target Data Lineage Template** to track the flow of data from the Source Systems to the Raw Layer.

Raw - Target Data Lineage	
The following table outlines the Data Lineage from the Source Systems to the Raw Layer.	

Field	Description	Example
Source System	The name of the source system from which the data originates.	HR_Core_System
Source System Description	A short description of the source system, its purpose, or business domain.	HR_Core_System
Source System Dataset Name	The table or dataset name from the source system.	employee_registry
Source System Attribute Name	The specific attribute or field being extracted from the source dataset.	emp_status_code
Data Type	The data type of the source attribute (e.g., Integer, String, Date).	String
Transformation Logic (If Any)	Description of the transformation applied to the source attribute before loading it into the raw layer.	Combine yearly files into one file
Target Dataset Name	The name of the dataset in the raw layer where the data is loaded.	raw_employee_profile
Target Attribute Name	The attribute in the raw dataset that stores the data.	raw_emp_status_code
Data Type	The target data type for the attribute after ingestion.	String
Comments	Any additional context or notes (e.g., exceptions, known issues, assumptions).	Include both active/inactive employees

Source System					
Source System	Source System Description	Source System Dataset Name	Source System Attribute Name	Data Type	Transformation Logic

Raw Layer			
Target Dataset Name	Target Attribute Name	Data Type	Comments

 Asset Guidance*Illustrative*

The Entity can leverage the **Target Data Lineage Template** to track the flow of data from the Raw Layer to the Bronze Layer.

Bronze - Target Data Lineage				
The following table outlines the Data Lineage from the Raw Layer to the Bronze Layer.				
Field	Description			Example
Dataset Name	The name of the raw layer dataset where the original attribute resides.			raw_employee_profile
Attribute Name	The specific attribute in the raw dataset being transformed or used in further processing.			raw_emp_birthdate
Data Type	The data type of the attribute in the raw dataset (e.g., String, Date, Integer).			Date
Transformation Logic	Description of the transformation applied to the source attribute before loading it into the bronze layer. This field may alternatively reference the object that performs the transformation (e.g., Python script, stored procedure, or database view).			Convert to ISO format (YYYY-MM-DD)
Target Dataset Name	The name of the dataset in the bronze layer where the transformed data is loaded.			bronze_employee_validated
Target Attribute Name	The attribute in the bronze dataset that stores the transformed data.			bronze_emp_birthdate
Data Type	The final data type after transformation, stored in the bronze layer.			Date
Lineage Type	Whether the lineage is direct, derived, or aggregated.			Aggregated
Comments	Any additional context or notes (e.g., exceptions, known issues, assumptions).			PII data is already masked.

Raw Layer			
Dataset Name	Attribute Name	Data Type	Transformation Logic

Bronze Layer				
Target Dataset Name	Target Attribute Name	Data Type	Lineage Type	Comments

 Asset Guidance*Illustrative*

The Entity can leverage the **Target Data Lineage Template** to track the flow of data from the Bronze Layer to the Silver Layer.

Silver - Target Data Lineage				
The following table outlines the Data Lineage from the Bronze Layer to the Silver Layer.				
Field	Description			Example
Dataset Name	The name of the bronze layer dataset where the original attribute resides.			bronze_employee_profile
Attribute Name	The specific attribute in the bronze dataset being transformed or used in further processing.			bronze_emp_birthdate
Data Type	The data type of the attribute in the bronze dataset (e.g., String, Date, Integer).			Date
Transformation Logic	Description of the transformation or enrichment applied to the attribute before moving to silver. This field may alternatively reference the object that performs the transformation (e.g., Python script, stored procedure, or database view).			Join with dept table; cast to VARCHAR(15)
Target Dataset Name	The name of the silver dataset where the result of the transformation is loaded.			silver_employee_validated
Target Attribute Name	The corresponding attribute in the silver dataset receiving the transformed data.			silver_emp_birthdate
Data Type	The final data type after transformation, stored in the silver layer.			Date
Lineage Type	Whether the lineage is direct, derived, or aggregated.			Direct
Comments	Any additional context or notes (e.g., exceptions, known issues, assumptions).			PII data is already masked.

Bronze Layer			
Dataset Name	Attribute Name	Data Type	Transformation Logic

Silver Layer				
Target Dataset Name	Target Attribute Name	Data Type	Lineage Type	Comments

3.2.E.I
Cont.
(3/3) Asset Guidance*Illustrative*

The Entity can leverage the **Target Data Lineage Template** to track the flow of data from the Silver Layer to the Gold Layer.

Gold - Target Data Lineage		
The following table outlines the Data Lineage from the Silver Layer to the Gold Layer.		

Field	Description	Example
Dataset Name	The name of the silver dataset where the attribute originates.	silver_employee_validated
Attribute Name	The specific attribute from the silver layer used as input in the gold layer.	silver_emp_birthdate
Data Type	The data type of the attribute in the silver dataset (e.g., String, Date, Integer).	Date
Transformation Logic	Description of final transformations, aggregations, or validations applied before storing in the gold layer. This field may alternatively reference the object that performs the transformation (e.g., Python script, stored procedure, or database view).	Derived age from birthdate
Target Dataset Name	The name of the gold layer dataset where the processed data is stored for consumption.	Dim_Employee
Target Attribute Name	The name of the attribute in the gold dataset receiving the transformed data.	Age
Data Type	The final data type after transformation, stored in the gold layer.	Int
Lineage Type	Whether the lineage is direct, derived, or aggregated.	Derived
Comments	Any additional context or notes (e.g., exceptions, known issues, assumptions).	Employees before 19XX don't have captured birthdate in the source system

Silver Layer			
Dataset Name	Attribute Name	Data Type	Transformation Logic

Gold Layer				
Target Dataset Name	Target Attribute Name	Data Type	Lineage Type	Comments

 Asset Guidance*Illustrative*

The Entity can leverage the **Target Data Lineage Template** to track the flow of data from the Gold Layer to the Reports.

Report Lineage		
The following table outlines the Data Lineage from the Gold Layer to the Reports.		

Field	Description	Example
Report Name	The title of the report or dashboard as seen by end users.	HR Dashboard
File Name	The file name of the report, typically including the file extension (e.g., .pbix, .rdl).	HR_Dashboard_2025.pbix
Transformation Logic	Any specific transformations or logic applied to data for the report (e.g., calculations, concatenations, or formatting rules).	Concatenated quarter name with department name
Datamart Name	The name of the datamart or logical grouping where the dataset resides.	HR
Dataset Name	The name of the dataset used within the report.	Dim_Employee
Attribute Name	The specific attribute or field from the dataset used in the report.	Age
Data Type	The final data type.	Int

Report Name	File Name	Transformation Logic	Dataset Name	Attribute Name	Data Type

3.2.E.II

*Conduct Internal Review & Finalize Target Data Lineage:***i. Internal Review & Stakeholder Alignment:**

- Engage data governance, analytics, ETL, and modeling teams to review the lineage documentation.
- Ensure all key transformations, aggregations, and dependencies are correctly captured.
- Validate that lineage supports impact analysis, data governance, and auditability needs.

ii. Address Feedback & Finalize the Data Lineage Documentation:

- Incorporate stakeholder feedback and validate attribute-level tracking.
- Save the validated lineage documentation in a centralized data model repository (as referenced in section 6).
- Ensure version control is applied for governance and future updates.
- Maintain the lineage documentation as part of ongoing governance processes for tracking data transformations over time.

RACI Matrix – Analytical Data Model

Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architect	Data Governance Specialist	Data Engineer	Data Steward	Data Compliance Specialist
3.2.A	Define Target Data Modeling Approach & Compliance Requirements			A	R		C		C
3.2.B.I	Develop the Target Conceptual Data Model			A	R				
3.2.B.II	Conduct Internal Review & Finalize Conceptual Data Model			A	R	C	C	C	
3.2.B.III	Share Conceptual Data Model Diagram and Supporting Documentation with DGO & Store in Data Model Repository	I	I	A	R				
3.2.C.I	Develop the Target Logical Data Model			A/C	R				
3.2.C.II	Conduct Internal Review & Finalize Logical Data Model			A	R	C	C	C	C
3.2.C.III	Share the Logical Data Model Diagram and Supporting Documentation with DGO & Store in Data Model Repository	I	I	A	R				
3.2.D.I	Develop the Target Physical Data Model:			A/C	R				
3.2.D.II	Conduct Internal Review & Finalize Physical Data Model			A	R	C	C	C	C
3.2.D.III	Share the Physical Data Model Diagram & Documentation with DGO & Store in Data Model Repository	I	I	A	R				
3.2.E.I	Define Target Data Lineage			A	R				
3.2.E.II	Conduct Internal Review & Finalize Target Data Lineage			A	R	C		C	

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can

-  Define the Target Conceptual Data Model
-  Define the Target Logical Data Model
-  Define the Target Physical Data Model
-  Document the Target Data Lineage Documentation

3.3 Analytical Data Model Metadata

The Analytical Data Model Metadata specification involves defining and maintaining comprehensive metadata for the analytical data model (Gold Layer) to ensure proper management, traceability, and version control. This includes unique identification, role assignments, status tracking, change history, and storage with version control.

Process – Data Model Metadata

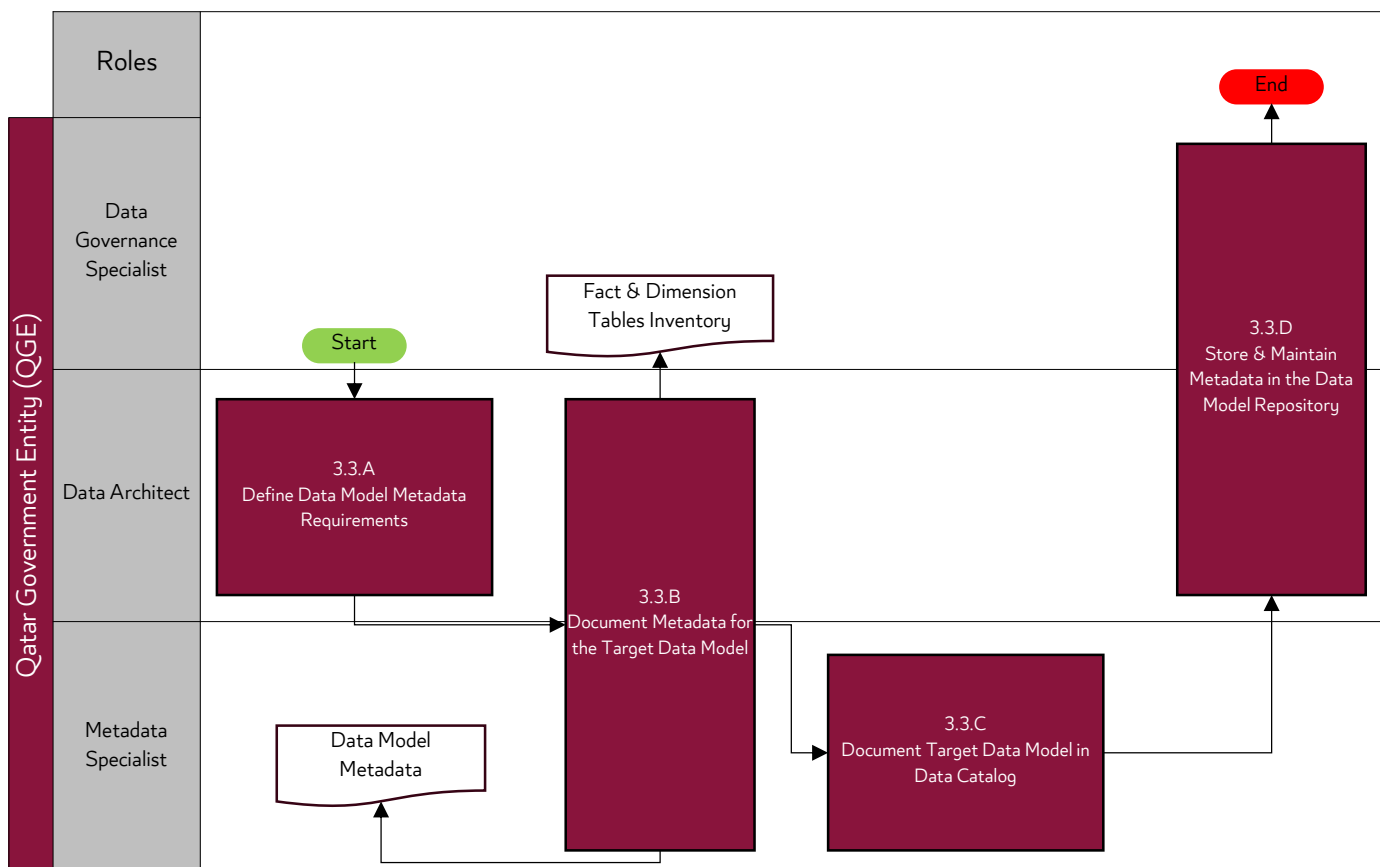


Figure 3: Process Map for Data Model Metadata

Legend: Process Step Decision Process Start Process End Document Off-Page Reference

3.3.A (1/2)

Define Data Model Metadata Requirements: To ensure proper management, traceability, and version control, the Entity must maintain comprehensive metadata for all data models. The metadata must align with NPC's National Data Standards and the Entity's data modeling guidelines (2.2) to facilitate governance, compliance, and seamless data integration across systems.

- i. **Establish Core Metadata Requirements:** The Entity must define standardized metadata attributes that apply to all conceptual, logical, and physical data models to ensure.
 - Consistent identification of data models across the organization.
 - Governance roles and responsibilities for model ownership and maintenance.
 - Version control to track model updates, changes, and approvals.

3.3.A
(2/2)

a. Unique Identification & Naming Convention:

- Each data model must have a unique identifier in the format:
 - [Entity Initials]-[Reference Number]-[Version] (e.g., NPC-123-V1.0).
- This ensures standardized tracking, prevents duplication, and enables easy retrieval.
- Naming conventions should be clear, descriptive, and aligned with organizational standards.

b. RACI Matrix for Model Governance:

- Define clear roles and responsibilities using the Responsible, Accountable, Consulted, and Informed (RACI) framework.
- Assign ownership for managing updates, approvals, and compliance tracking.
- Ensure stakeholders understand their involvement in data model maintenance.

c. Versioning & Model Status Tracking:

- Implement a version control system to track changes, updates, and approvals.
- Define status indicators to classify models as:
 - **Draft:** In development, subject to change.
 - **Published:** Approved and used in production.
 - **Retired:** No longer in use, but stored for reference.

d. Data Model Repository & Governance Integration:

- All models must be stored in a centralized repository (as referenced in section 6).
- Repository options (per NPC standards):
 - Integrated version control within modeling tools (preferred).
 - External repositories or document management systems.
- Ensure metadata is governed within the broader enterprise data governance framework.



Key Message / Information

The Data Architecture & Modeling Repository (section 6) establishes a centralized platform for storing, managing, and maintaining all documentation related to data architecture and modeling.

ii. Finalize Metadata Requirements for Governance:

- Align metadata requirements with NPC's metadata management policies.
- Document and approve the standardized metadata structure before moving forward.
- Store metadata definitions in a centralized metadata repository to support governance and future updates.

3.3.B
(1/5)

Document Metadata for the Target Data Model: Once metadata requirements have been defined, the Entity must document metadata for the Target Data Model, ensuring consistency, governance, and compliance with NPC's National Data Standards. This step involves capturing key metadata attributes for conceptual, logical, and physical data models using a structured approach to support traceability, version control, and governance.

- i. **Capture High-level Metadata for Conceptual, Logical, and Physical Data Models:** The Entity must document metadata for all three layers of the Target Data Model using a structured metadata template that ensures alignment across business, analytical, and governance needs.

- a. **Metadata for Conceptual Data Models:**

Asset Guidance

The **Data Model Metadata** Template can be leveraged by the Entity to capture high-level metadata for the conceptual, logical, and physical data models. It supports the Entity in maintaining accurate, consistent, and traceable metadata.

Illustrative

 Conceptual Data Model Metadata							
The following table can be used to document conceptual data model metadata.							
Field	Description						
Data Model ID	Unique identifier for the conceptual data model.						
Data Model Name	Name of the conceptual data model.						
Data Model Description	A brief description of the conceptual data model and the respective business entities it contains.						
Data Model Business Domain	Business area or function the conceptual data model represents (e.g., Sales, HR, Finance).						
Data Sources	Source systems that contribute to the conceptual model.						
Purpose & Scope	Why this model exists and its intended use (e.g., cross-departmental integration, reporting, analytics, a specific use case).						
Status	Status of the conceptual data model (e.g., Draft, Published, Retired).						
Owner	The business unit head or executive responsible for the conceptual data model. Ensures the model aligns with business goals.						
Responsible to Maintain	The team/person who ensures the conceptual data model stays up-to-date, accurate, and compliant with data policies. (e.g., Data Steward, Data Architect).						
Responsible to Update	The person/team responsible for making updates to the conceptual data model (e.g., Data Modeler, Data Architect).						
Responsible to Approve	Final decision-maker who authorizes changes. Should be separate from the updater to ensure governance.						
Date Created	Date when the conceptual data model was created.						
Last Updated	Date of the most recent update.						
Version	Version number of the conceptual data model for tracking changes.						
Assumptions & Constraints	Key design assumptions and limitations that impact the conceptual data model (e.g., only internal data sources, no real-time integration).						
Comments	Any comments or key notes on the respective conceptual data model.						

Data Model ID	Data Model Name	Data Model Description	Data Model Business Domain	Data Sources	Purpose & Scope	Status	Owner

Responsible to Maintain	Responsible to Update	Responsible to Approve	Date Created	Last Updated	Version	Assumptions & Constraints	Comments

3.3.B
(2/5)

b. Metadata for Logical Data Model:

Asset Guidance

Illustrative



Logical Data Model Metadata

The following table can be used to document logical data model metadata.

Field	Description
Data Model ID	Unique identifier for the logical data model.
Data Model Name	Name of the logical data model.
Data Model Description	A brief description of the logical data model and the business context it represents.
Data Model Business Domain	Business area or function the logical data model represents (e.g., Sales, HR, Finance).
Data Sources	Source systems that contribute to the logical data model.
Purpose & Scope	Why this logical data model exists and its intended use (e.g., cross-departmental integration, reporting, analytics, a specific use case).
Status	Status of the logical model (e.g., Draft, Published, Retired).
Owner	The business unit head or executive responsible for the logical data model. Ensures the model aligns with business goals.
Responsible to Maintain	The team/person who ensures the logical data model stays up-to-date, accurate, and compliant with data policies. (e.g., Data Steward, Data Architect).
Responsible to Update	The person/team responsible for making updates to the logical data model (e.g., Data Modeler, Data Architect).
Responsible to Approve	Final decision-maker who authorizes changes. Should be separate from the updater to ensure governance.
Date Created	Date when the logical data model was created.
Last Updated	Date of the most recent update.
Version	Version number of the logical data model for tracking changes.
Assumptions & Constraints	Key design assumptions and limitations that impact the logical data model (e.g., only internal data sources, no real-time integration).
Comments	Any comments or key notes on the respective logical data model.

Data Model ID	Data Model Name	Data Model Description	Data Model Business Domain	Data Sources	Purpose & Scope	Status	Owner

Responsible to Maintain	Responsible to Update	Responsible to Approve	Date Created	Last Updated	Version	Assumptions & Constraints	Comments

c. Metadata for Physical Data Model:

Asset Guidance

Illustrative



Physical Data Model Metadata

The following table can be used to document physical data model metadata.

Field	Description
Data Model ID	Unique identifier for the physical data model.
Data Model Name	Name of the physical data model.
Data Model Description	A brief description of the physical data model and the business context it represents.
Data Model Business Domain	Business area or function the physical data model represents (e.g., Sales, HR, Finance).
Database Platform	The target database system where the model is implemented (e.g., Oracle, MySQL, SQL Server, PostgreSQL).
Schema Name	The database schema where the model resides.
Data Sources	Source systems that contribute to the physical data model.
Purpose & Scope	Why this physical data model exists and its intended use (e.g., cross-departmental integration, reporting, analytics, a specific use case).
Status	Status of the physical model (e.g., Draft, Published, Retired).
Owner	The business unit head or executive responsible for the physical data model. Ensures the model aligns with business goals.
Responsible to Maintain	The team/person who ensures the physical data model stays up-to-date, accurate, and compliant with data policies. (e.g., Data Steward, Data Architect).
Responsible to Update	The person/team responsible for making updates to the physical data model (e.g., Data Modeler, Data Architect).
Responsible to Approve	Final decision-maker who authorizes changes. Should be separate from the updater to ensure governance.
Date Created	Date when the physical data model was created.
Last Updated	Date of the most recent update.
Version	Version number of the physical data model for tracking changes.
Assumptions & Constraints	Key design assumptions and limitations that impact the physical data model (e.g., only internal data sources, no real-time integration).
Comments	Any comments or key notes on the respective physical data model.

Data Model ID	Data Model Name	Data Model Description	Data Model Business Domain	Database Platform	Schema Name	Data Sources	Purpose & Scope	Status

Owner	Responsible to Maintain	Responsible to Update	Responsible to Approve	Date Created	Last Updated	Version	Assumptions & Constraints	Comments

3.3.B
(3/5)

d. **Storage & Accessibility of Metadata:**

- Metadata must be stored in a centralized repository with version control.
- Ensure metadata is accessible for governance teams, architects, and analysts.
- Align with NPC's metadata management framework to support traceability and compliance.

ii. **Capture Attribute-Level Metadata for the Physical Data Model:** For the Physical Data Model, attribute-level metadata must be documented using a five-sheet metadata template, ensuring completeness in fact and dimension tables.

a. **High-level Fact Table Details.** Capture metadata at the fact table level, including:

 Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact Table Inventory Template which the Entity can leverage to capture metadata at the fact table level.

Illustrative


Fact Table Inventory

The following table can be used to capture metadata at the Fact table level

Field	Description
Fact Table ID	ID number of the Fact Table.
Fact Table Name	Name of the Fact table.
Fact Table Description	Description of the respective Fact Table.
Fact Table Business Domain	The business area or function the Fact table supports (e.g., Sales, Finance, HR).
Primary Business Process	The business process the Fact table supports (e.g., Order Processing, Customer Transactions).
Granularity	Defines the level of detail recorded in the Fact table (e.g., Per Transaction, Per Day).
Data Model Name	Name of the data model where the Fact table resides.
Source Systems	The original systems where the data is sourced from (e.g., CRM, ERP).
Number of Measures	Count of key numeric values stored in the Fact table.
Number of Dimensions	Count of Dimension tables related to this Fact table.
Retention Policy	Defines how long data is retained before being archived or deleted.
Security Classification	Sensitivity level of the data (e.g., Public, Confidential, Restricted).
Owner	The department or individual responsible for maintaining the Fact table.
Update Frequency	Defines how often the table is updated (e.g., Real-Time, Daily, Weekly).
Date Created	Date when the table was created.
Last Updated	Date of the most recent update.

Fact Table ID	Fact Table Name	Fact Table Description	Fact Table Business Domain	Primary Business Process	Granularity	Data Model name	Source Systems

Number of Measures	Number of Dimensions	Retention Policy	Security Classification	Owner	Update Frequency	Date Created	Last Updated

3.3.B
(4/5)

b. High-level Dimension Table Details. Capture dimension table level metadata, including:

 Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Dimension Table Inventory Template which the Entity can leverage to capture metadata at the Dimension table level.

Illustrative

 Dim Table Inventory							
The following table can be used to capture metadata at the Dimension table level							
Field	Description						
Dimension Table ID	ID number of the Dimension Table.						
Dimension Table Name	Name of the Dimension table.						
Dimension Table Description	Description of the respective Dimension Table.						
Dimension Table Business Domain	The business area or function the Dimension table supports (e.g., Product, Customer, Employee).						
Dimension Type	Type of Dimension (e.g., Slowly Changing, Role-Playing, Degenerate).						
Hierarchy Available	Specifies if the Dimension has hierarchical levels (Yes/No).						
Hierarchy Levels	Lists the hierarchy levels if applicable (e.g., Country → Region → City).						
Data Model Name	Name of the data model where the Dimension table resides.						
Source Systems	The original systems where the Dimension data is sourced from (e.g., CRM, ERP, External Data).						
Number of Attributes	Count of attributes stored in the Dimension table.						
Retention Policy	Defines how long data is retained before being archived or deleted.						
Security Classification	Sensitivity level of the data (e.g., Public, Restricted).						
Owner	The department or individual responsible for maintaining the Dimension table.						
Update Frequency	Defines how often the table is updated (e.g., Weekly, Monthly).						
Date Created	Date when the table was created.						
Last Updated	Date of the most recent update.						

Dimension Table ID	Dimension Table Name	Dimension Table Description	Dimension Table Business Domain	Dimension Type	Hierarchy Available	Hierarchy Levels	Data Model Name

Source Systems	Number of Attributes	Retention Policy	Security Classification	Owner	Update Frequency	Date Created	Last Updated

c. Detailed Fact Table Attributes. Capture fact table attribute level metadata, including:

 Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact Table Attribute Inventory Template which the Entity can leverage to document fact table structures.

Illustrative

 Fact Table Attribute Inventory						
The following table can be used to document Fact table details in a standardized format.						
Field	Description					
Fact Table ID	ID number of the Fact Table.					
Fact Table Name	The name of the Fact table where the attribute resides.					
Attribute Name	The name of the specific attribute within the respective Fact table.					
Attribute Description	A brief explanation of what the attribute represents and its business significance.					
Measure (Y/N)	Indicates whether the attribute is a measure (Y) or not (N). Measures are typically numeric values used for aggregations such as sums, averages, counts, or calculations in reports and dashboards.					
Calculation Methodology	If the attribute is a measure, this field describes how it is calculated. It should include the formula, business logic, or aggregation method used to derive the measure.					
Is Primary/Foreign Key	Specifies whether the attribute is a Primary Key (PK), Foreign Key (FK), or NA.					
Data Type	The data type of the attribute (e.g., Integer, Decimal, Date, Varchar).					
Length	Specifies the maximum number of characters or digits allowed for the attribute.					
Granularity	Defines the level of detail recorded in the Fact table (e.g., per transaction, per day).					
Source System	The original system where the data is sourced from (e.g., CRM, ERP).					
Owner	The department or individual responsible for maintaining the attribute.					
Update Frequency	Defines how often the attribute is updated (e.g., Real-Time, Daily, Weekly).					
Last Updated	The date when the attribute was last modified or refreshed.					

Fact Table ID	Fact Table Name	Attribute Name	Attribute Description	Measure (Y/N)	Calculation Methodology	Is Primary/Foreign Key

Data Type	Length	Granularity	Source System	Owner	Update Frequency	Last Updated

3.3.B
(5/5)

- d. **Detailed Dimension Table Attributes.** Capture metadata at the attribute level for dimension tables, including:

Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Dimension Table Attribute Inventory Template which the Entity can leverage to document dimension table structures.

Illustrative

 Dim Table Attribute Inventory						
The following table can be used to document Dimension table details in a standardized format.						
Field	Description					
Dimension Table ID	ID number of the Dimension Table.					
Dimension Table Name	The name of the Dimension table where the attribute resides.					
Attribute Name	The name of the specific attribute within the respective Dimension table.					
Attribute Description	A brief explanation of what the attribute represents and its business significance.					
Is Primary/Foreign Key	Specifies whether the attribute is a Primary Key (PK), Foreign Key (FK), or NA.					
Natural/Surrogate Key	Specifies whether the attribute is a natural key (derived from the source system and meaningful to the business) or a surrogate key (system-generated and used solely for integration and join operations). Common values include "Natural," "Surrogate," or "NA."					
Slowly Changing Dimension Type	Indicates the SCD type used to manage historical changes for the attribute.					
Data Type	The data type of the attribute (e.g., Integer, Decimal, Date, Varchar).					
Length	Specifies the maximum number of characters or digits allowed for the attribute.					
Hierarchies	Identifies whether the attribute is part of a hierarchical structure (e.g., Country → Region → City). If yes, specify the hierarchy level or grouping it belongs to (e.g., "Level 2 – Region").					
Source System	A brief explanation of what the attribute represents and its business significance.					
Owner	The department or individual responsible for maintaining the attribute.					
Update Frequency	Defines how often the attribute is updated (e.g., Real-Time, Daily, Weekly).					
Last Updated	The date when the attribute was last modified or refreshed.					
Dimension Table ID	Dimension Table Name	Attribute Name	Attribute Description	Is Primary/Foreign Key	Natural/Surrogate Key	Slowly Changing Dimension Type
Data Type	Length	Hierarchies	Source System	Owner	Update Frequency	Last Updated

- e. **Document Relationships & Constraints.** Document relationships and referential integrity constraints between fact-dimension and dimension-dimension tables:

Asset Guidance

The **Fact & Dimension Tables Inventory** contains a Fact-Dimension Relationships and Dimension-Dimension Relationships Template which the Entity can leverage to document how the respective tables are connected. For illustrative purposes, only the Fact-Dimension Relationships Template is shown below.

Illustrative

 Fact & Dimension Relationship						
This Fact-Dimension Relationship Sheet provides a comprehensive view of how Fact and Dimension tables are connected.						
Field	Description					
Fact Table ID	ID number of the Fact Table.					
Fact Table Name	The name of the Fact table that contains the measurable data (e.g., Sales Fact, Transactions Fact).					
Dimension Table ID	ID number of the Dimension Table.					
Dimension Table Name	The name of the related Dimension table (e.g., Customer Dimension, Product Dimension).					
Fact Table Key (FK)	The Foreign Key in the Fact table that establishes the relationship with the Dimension table.					
Dimension Table Key (PK)	The Primary Key in the Dimension table that links to the Fact table's foreign key.					
Relationship Type	Defines the relationship type (e.g., One-to-Many, Many-to-One).					
Fact Table ID	Fact Table Name	Dimension Table ID	Dimension Table Name	Fact Table Key (FK)	Dimension Table Key (PK)	Relationship Type

3.3.C

Document Target Data Model in the Data Catalog: Once the Target Data Model (facts and dimensions) has been designed and documented, the Entity must enter the relevant metadata into the entity Data Catalog (as outlined in the Data Catalog & Metadata Management Domain). This enables discoverability, promotes reuse, and supports governance by ensuring that datasets and their associated metadata are structured, consistent, and centrally accessible.

i. **Register Metadata in the Data Catalog:**

- Enter metadata for all fact tables, dimension tables, and key attributes into the designated fields within the Data Catalog.
- Link relevant business glossary terms to ensure consistent business understanding across the Entity.

ii. **Align Metadata with Defined Standards:**

- Ensure that metadata fields adhere to NPC's metadata standards and controlled vocabularies.
- Use standardized naming conventions and ensure consistency in terminology and formatting across all entries.

iii. **Enable Discoverability & Governance Integration:**

- Ensure that metadata is searchable and accessible to authorized users via the Data Catalog.
- Tag metadata appropriately (e.g., by domain, sensitivity, or department) to facilitate filtering and discovery.
- Ensure integration of data catalog metadata with the Data Architecture & Modeling Repository (Section 6) to maintain alignment across documentation sources.

3.3.D
(1/2)

Store & Maintain Metadata in the Data Model Repository: Once metadata for the Target Data Model has been documented, the Entity must store and maintain it in a centralized repository to ensure traceability, accountability, and governance oversight. Metadata must be properly versioned and updated in alignment with the standards defined in 3.3.A and NPC's National Data Standards (DAM.2.1.2).

i. **Store Metadata in the Central Repository:**

- Store metadata for all conceptual, logical, and physical data models in the designated repository.
- Ensure that metadata is kept up to date as the data models evolve over time.
- Limit metadata storage to finalized and approved versions only (i.e., do not include incomplete drafts unless required for audit purposes).

3.3.D
(2/2)

- ii. **Maintain Version Control & Change History:** In alignment with Section 6 (Data Architecture & Modeling Repository), to support transparency and governance, metadata must be version-controlled and auditable.

a. **Version Tracking:**

- Apply a standardized versioning format (e.g., NPC-001-V1.0).
- For each version, record:
 - Model status (e.g., Draft, Published, Retired)
 - Version number and release date
 - Author and approver details

b. **Change History & Audit Trail:**

- Maintain a record of modifications made to metadata over time, including:
 - What was changed and why
 - Who made the change and when
- Ensure that metadata history is accessible for review and governance audits.

RACI Matrix - Data Model Metadata

Indicative

Sr No.	Activity	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Management and Governance Section Head	Data Architect	Metadata Specialist	Data Governance Specialist
3.3.A	Define Data Model Metadata Requirements		A		R	C	
3.3.B	Document Metadata for the Target Data Model	I	A		R	R	
3.3.C	Document Target Data Model in Data Catalog	I	I	A	C	R	
3.3.D	Store & Maintain Metadata in the Data Model Repository	I	A		R	C	R

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Document Data Model Metadata



Document Target Data Model in Data Catalog



Store & Maintain Metadata in the Data Architecture & Modeling Repository

4. Data Architecture & Modeling Tools



4. Data Architecture & Modeling Tools

What is a Data Architecture & Modeling Tools and why does the Entity need it?

Data Architecture & Modeling Tools refer to specialized software used to design, document, and manage data models across conceptual, logical, and physical layers. These tools support structured data modeling, metadata governance, and model deployment in alignment with the national data standards. With capabilities such as model validation, lineage tracking, version control, and DDL generation, the selected tool serves as the technical foundation for data modeling activities across the Entity.

Data Architecture & Modeling Tools are essential for enabling consistent model design, enforcing governance policies, and supporting implementation across data platforms. These tools streamline collaboration between architects, data stewards, and engineering teams, ensuring that models are accurate, compliant, and ready for deployment.

The remaining sections of this chapter will detail the activities of the below process that will guide the Entity on selecting and implementing a Data Architecture & Modeling Tool.

4.1 Data Architecture & Modeling Tools

Guiding Principles

Guiding Principles for Data Architecture & Modeling Tools

1. **Enable Standardized Data Modeling Practices:** Select tools that support the design and documentation of conceptual, logical, and physical data models in alignment with the National Data Standards. Ensure the tool enforces relationship structures, and version control to promote consistency and traceability.
2. **Integrate with Metadata and Governance Frameworks:** Choose tools that capture and manage metadata across model layers, including data definitions, ownership, lineage, and transformation logic. Ensure integration with data catalogs and governance workflows to support compliance and impact analysis.
3. **Support Deployment and Scalability:** Implement tools that generate platform-specific DDL and support large-scale, production-ready data environments. Ensure modeling outputs can be deployed efficiently and optimized for indexing, partitioning, and analytical performance..
4. **Enable Collaboration and Lifecycle Management:** Adopt tools that allow multi-user access, change tracking, and collaborative design processes. Ensure roles such as architects, engineers, and stewards can review, approve, and maintain models as a shared asset throughout the modeling lifecycle.

4.1 Data Architecture & Modeling Tools

The objective of this activity is to guide the Entity in selecting and implementing a data architecture and modeling tool and developing the bronze, silver, and gold layers of the data architecture.

4.1.1 Data Architecture and Modeling Tool Selection and Implementation Plan

The objective of this process is to guide the Entity in selecting the most suitable Data Architecture & Modeling tool.

Process – Data Architecture and Modeling Tool Selection and Implementation Plan

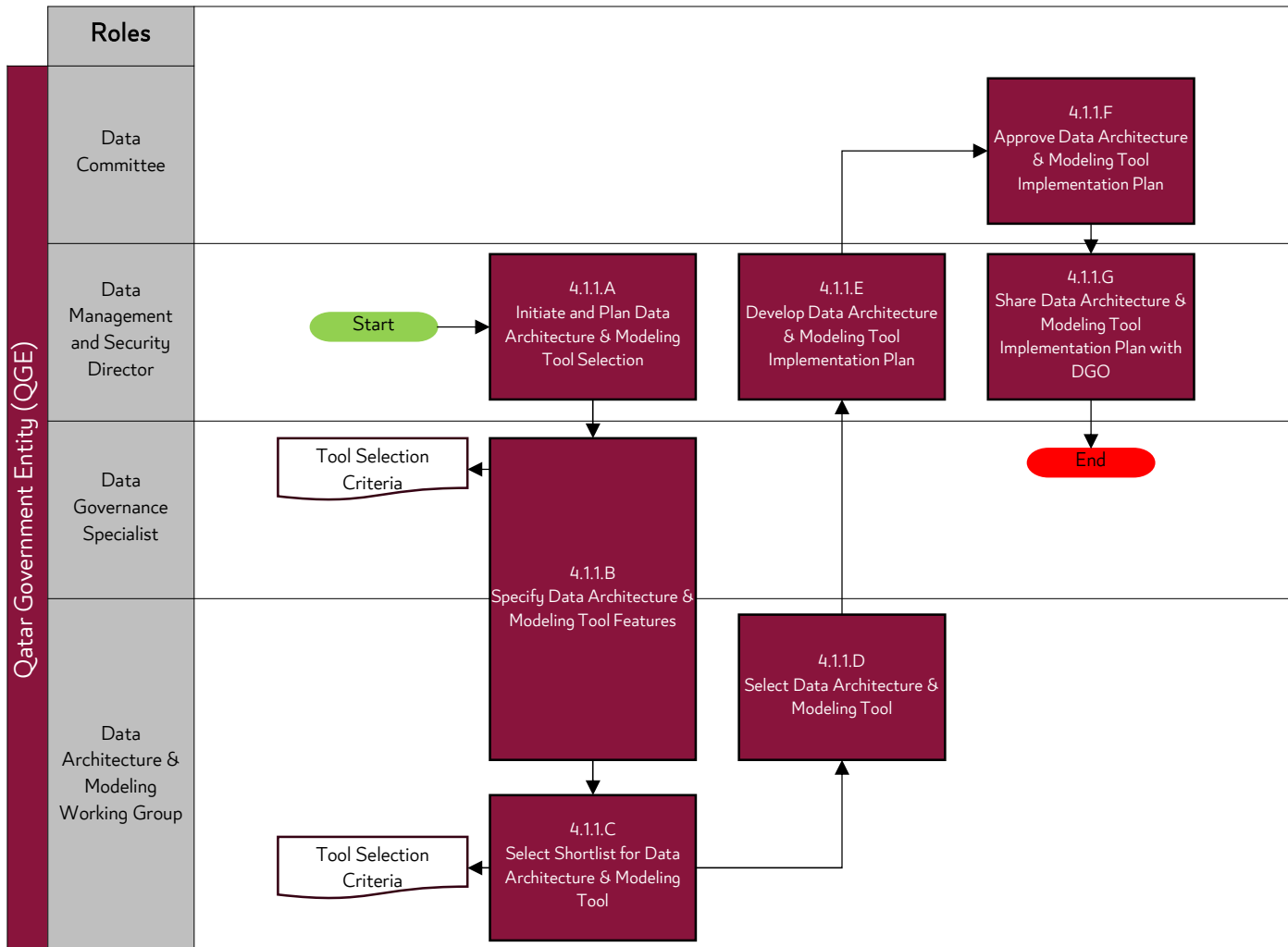


Figure 4 – Data Architecture and Modeling Tool Selection and Implementation Plan Process Map

Legend: Process Step Decision Process Start Process End Document Off-Page Reference

4.1.1.A

Initiate and Plan Data Architecture & Modeling Tool Selection: Define the objectives and intended use cases for the Data Architecture & Modeling Tool, ensuring alignment with the Entity's data strategy and architectural requirements as outlined in the National Data Standards. Form a Data Architecture & Modeling Working Group comprising stakeholders from data architecture, IT, compliance, metadata management, and analytics, including data modelers and data stewards. Engage this group throughout the process to provide technical insight, ensure compliance, and support informed decision-making.

4.1.1.B
(1/2)

Specify Data Architecture & Modeling Tool Features: Identify and document the core functional and non-functional requirements for the tool based on the Entity's Target architectural maturity. At a minimum, the Data Architecture & Modeling Tool must support the following:

i. **Foundational Features:**

- **Data Modeling (Conceptual, Logical, Physical):** Support visual modeling of data structures with entity relationships, normalization.
- **Data Lineage Tracking:** Provide automated or semi-automated lineage capabilities to visualize data movement from source to consumption.
- **Metadata Management:** Support for defining and managing business, technical, and operational metadata across data assets.
- **Collaboration & Review:** Allow multiple users to collaborate on model design, review, and approvals through workflows or commenting.
- **Export & Integration:** Export models to common formats (e.g., XMI, JSON) and integrate with metadata catalogs or data warehouse platforms.
- **Version Control & Audit Trails:** Maintain model history with rollback, version comparison, and change tracking for governance and impact analysis.
- **Role-Based Access Control:** Support secure access to models and artifacts based on user roles (e.g., viewer, editor, approver).
- **Model-to-DDL Generation:** Generate physical DDL scripts to deploy database structures in supported platforms (e.g., Snowflake, SQL Server, PostgreSQL).

Once requirements are identified, populate the Data Architecture & Modeling Tool Selection Criteria Template, specifying the tool features, feature weightings, and cost weightings to guide the selection process.

4.1.1.B
Cont.
(2/2) **Asset Guidance**

The **Data Architecture & Modeling Tool Selection Criteria Template** helps the Entity document mandatory and additional Data Architecture and Modeling tool features, list various tool options, specify feature and cost weightings based on importance, and guide the Entity through the decision-making process.

Illustrative

Data Architecture and Modeling Tool Selection Criteria Template						
The following tables provides a structured template to guide the evaluation and selection of a data architecture and modeling tool.						
Field	Description					
Tool feature	The specific tool feature required based on the domain's standards (e.g., version control).					
Tool feature weight	The weighted percentage assigned to the tool feature based on importance to the respective entity.					
Feature level of Advancement [Tool name 1,2,3,X]	The level of advancement of the feature as per the standards (Foundational, Advanced, or Very Advanced). Whether the suggested tool has the feature described (Yes/No).					
Features Scoring	Description					
1 - Low	Limited functionality, insufficient to meet key requirements					
2 - Medium	Basic functionality, meets most requirements but with some limitations					
3 - High	Strong functionality, meets all requirements effectively with minor gaps					
4 - Very High	Comprehensive functionality, meets all requirements with advanced features and flexibility					
Pricing Scoring	Description					
1 - Low	High cost solution					
2 - Medium	Moderate cost solution					
3 - High	Budget friendly solution					
4 - Very High	Low cost solution					
Tool Feature	Tool Feature Weight	Feature level of Advancement	[Tool name 1]	[Tool name 2]	[Tool name 3]	[Tool name X]
Cost	30%	Foundational	4			
Data Modeling	20%	Foundational	3			
Data Lineage Tracking	10%	Foundational	2			
Metadata Management	10%	Foundational	2			
Collaboration & Review	10%	Foundational	2			
Export & Integration	5%	Foundational	1			
Version Control & Audit Trails	5%	Foundational	2			
Role Based Access Control	5%	Foundational	4			
Model-to-DDL Generation	5%	Foundational	3			
Total:			73%	0%	0%	0%

4.1.1.C

Select Shortlist for Data Architecture & Modeling Tool: Conduct market research and evaluate available Data Architecture & Modeling Tools. Shortlist those that meet the required features and align with the Entity's technology ecosystem. Populate the **Data Architecture & Modeling Tool Selection Criteria Template** with shortlisted tools and assign scores based on the defined criteria.

4.1.1.D

Select Data Architecture & Modeling Tool: Review scores for feature support, integration capability, usability, and cost. Finalize the selection of the Data Architecture & Modeling tool that offers the best balance between functionality, compliance with the National Data Standards, and alignment with the Entity's data architecture vision.

4.1.1.E

Develop Data Architecture & Modeling Tool Implementation Plan: Create a detailed implementation plan for the selected Data Architecture & Modeling Tool, covering:

- Tool installation and environment setup
- Configuration and integration with existing metadata repositories, catalogs, and warehouses
- Migration of existing models (if applicable)
- User training, enablement sessions, and governance onboarding
- Timelines, roles and responsibilities, budget, and resource requirements

4.1.1.F

Approve Data Architecture & Modeling Tool Implementation Plan: Present the finalized implementation plan and budget for internal approval. Ensure it is aligned with data architecture and modeling standards, data governance priorities, and available funding.

4.1.1.G

Share Data Architecture & Modeling Tool Implementation Plan with DGO: Submit the approved Data Architecture & Modeling Tool Implementation Plan to the National Data Governance Office (DGO) for review and visibility. This ensures alignment with national priorities and transparency around tool adoption, usage, and integration across government architecture.

RACI Matrix – Data Architecture and Modeling Tool Selection and Implementation Plan Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Management and Governance Section Head	DAM Working Group	Data Governance Specialist	Data Committee
4.1.1.A	Initiate and Plan Data Architecture & Modeling Tool Selection		A/R	I		C	C
4.1.1.B	Specify Data Architecture & Modeling Tool Features		I	I	R	A/R	
4.1.1.C	Select Shortlist for Data Architecture & Modeling Tool		I	A	R	I	
4.1.1.D	Select Data Architecture & Modeling Tool		C	A	R	C	
4.1.1.E	Develop Data Architecture & Modeling Tool Implementation Plan		A/R	C	C	C	
4.1.1.F	Approve Data Architecture & Modeling Tool Implementation Plan		C	I	I	C	A/R
4.1.1.G	Share Data Architecture & Modeling Tool Implementation Plan with DGO	I	A/R				

RACI Matrix Legend: R: Responsible A: Accountable C: Consulted I: Informed

Based on this Activity, the Entity can



Specify Tool Features and Select Tool



Develop Tool Implementation Plan

4.1.2 Implement Data Architecture & Modeling Tool

The objective of this process is to guide the Entity in implementing the selected Data Architecture & Modeling Tool.

Process – Implement Data Architecture & Modeling Tool

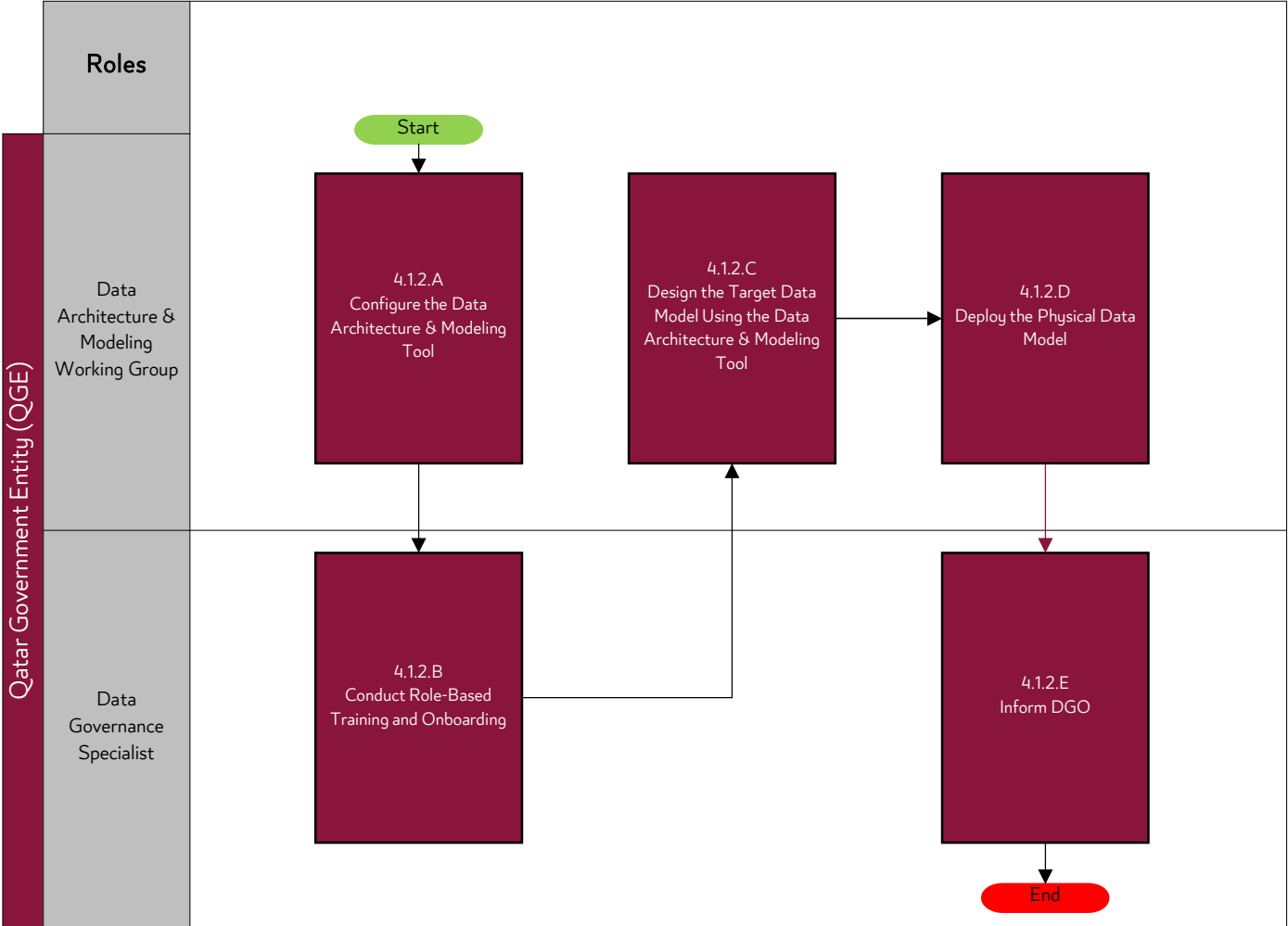


Figure 5 – Process Map for Implementing Data Architecture & Modeling Tool

Legend: Process Step Decision Process Start Process End Document Off-Page Reference

4.1.2.A

Configure the Data Architecture & Modeling Tool: Set up the Data Architecture & Modeling Tool to support modeling workflows, governance standards, and integration needs:

- Install and configure the Data Architecture & Modeling Tool in alignment with national data architecture and modeling standards.
- Set up repositories and libraries for managing models (conceptual, logical, physical).
- Define modeling standards (e.g., naming conventions, key structures, versioning) within the tool’s configuration in alignment with the data architecture and modeling handbook.
- Integrate the tool with external systems such as the data catalog, and data platforms (e.g., Snowflake, SQL Server).
- Enable governance features such as change tracking, model validation, approval workflows, and role-based access controls.

4.1.2.B

Conduct Role-Based Training and Onboarding: Before modeling activities begin, provide tailored training sessions for all relevant roles as well as user guides, naming convention documents, and walk-throughs of bronze/silver/gold modeling examples.

- Data Architects : How to design, document, and validate data models using the tool.
- Data Stewards and Metadata Specialists: How to review, annotate, and approve models and metadata.
- Data Engineers: How to interpret generated DDL scripts and maintain synchronization between model and implementation.

4.1.2.C

Design the Target Data Model Using the Data Architecture & Modeling Tool: The Entity shall use the configured Data Architecture & Modeling Tool to design its Conceptual, Logical, and Physical Data Models, as detailed in Section 3.2 of this guide. The tool serves as the central platform for modeling, documentation, governance, and future maintenance. The Entity shall:

- Execute all data modeling activities described in Section 3.2 using the Data Architecture & Modeling Tool.
- Ensure all metadata, constraints, and governance rules are captured within the tool throughout the design process.
- Utilize built-in features such as model validation, versioning, and lineage tracking to maintain model quality and compliance with the national data standards.
- Store all models in a repository (as detailed in section 6 of this guide).

4.1.2.D

Deploy the Physical Data Model. Once the physical model is finalized:

- Generate DDL scripts from the Data Architecture & Modeling Tool for the target database platform (e.g., Snowflake, SQL Server, Oracle).
- Collaborate with data engineering teams to execute the scripts and deploy the model into the staging or production environment.
- Validate the deployment to ensure schema integrity, referential integrity, and alignment with the approved model.
- Store the deployed model version within the tool for future synchronization, maintenance, and impact analysis.

4.1.2.E

Inform DGO: After the Data Architecture & Modeling Tool is fully implemented and operational, the Data Governance Specialist shall inform the National Data Governance Office (DGO). This communication should include a summary of the implementation, key configuration and validation outcomes, and confirmation that the tool aligns with the National Data Standards and supports the Entity's data modeling and governance requirements.

RACI Matrix - Implement Data Architecture & Modeling Tool

Indicative

Sr No.	Activity	DGO	Data Management and Security Director	Data Architecture and Analytics Section Head	Data Architecture & Modeling Working Group	Data Governance Specialist
4.1.2.A	Configure the Data Architecture & Modeling Tool		I	A	R	
4.1.2.B	Conduct Role-Based Training and Onboarding			A	I	R
4.1.2.C	Design the Target Data Model Using the Data Architecture & Modeling Tool		I	A	R	
4.1.2.D	Deploy the Physical Data Model		I	A	R	
4.1.2.E	Inform DGO	I	I	A	I	R

RACI Matrix Legend:

R: Responsible

A: Accountable

C: Consulted

I: Informed

Based on this Activity, the Entity can



Configure Data Architecture & Modeling Tool



Deploy the Physical Data Model

A city skyline at sunset, featuring a prominent, modern building with a sharp, triangular peak in the center. The sky is a mix of orange, pink, and purple hues, and the city lights are visible in the foreground. The text "5. Data Architecture & Modeling Performance & Change Management" is overlaid in white.

5. Data Architecture & Modeling Performance & Change Management

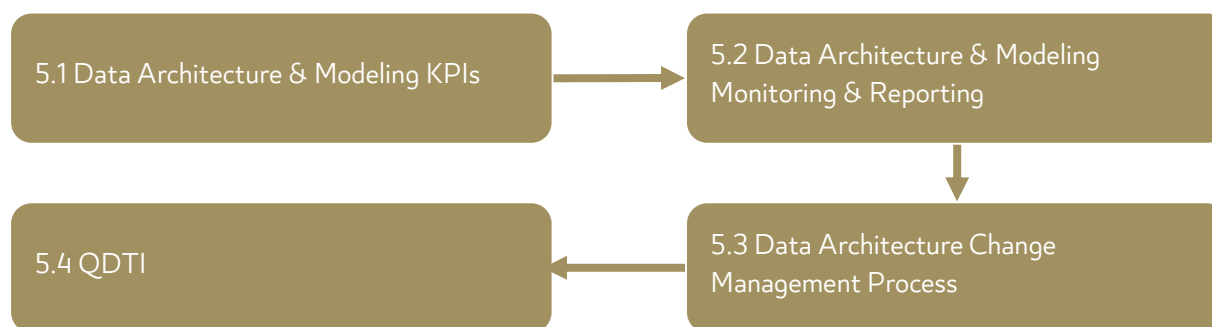
5. Data Architecture & Modeling Performance & Change Management

What is a Data Architecture & Modeling Performance & Change Management and why does the Entity need it?

Data Architecture & Modeling Performance & Change Management is a control designed to measure, monitor, and manage the effectiveness of data architecture and modeling initiatives. This control suggests the use of key performance indicators (KPIs) to track progress and compliance, routine reporting for continuous improvement, and a robust change management process to align modifications with strategic goals.

The Entity needs Data Architecture & Modeling Performance & Change Management to ensure that the data architecture initiatives are on track, measurable, and adaptable to changing requirements. By implementing KPIs, entities can measure the maturity of their data architecture, identify areas of non-compliance, and address performance gaps proactively. The change management aspect ensures that all architectural modifications are systematically evaluated, approved, and aligned with the Entity's objectives, minimizing risks and disruptions while fostering accountability and transparency.

The remaining sections of this chapter will detail the activities of the below process that will guide the Entity on executing Data Architecture & Modeling Performance & Change Management:



Guiding Principles

Guiding Principles for Data Architecture & Modeling Performance & Change Management

- 1. Establish Clear and Relevant KPIs:** Define measurable KPIs that align with strategic objectives, focusing on compliance, progress toward the target state, data model completeness, and adherence to standards. KPIs should provide actionable insights into the effectiveness of data architecture and modeling initiatives.
- 2. Conduct Regular Reporting and Reviews:** Implement a structured reporting cadence (e.g., quarterly or monthly) to analyze KPI trends, highlight achievements, and address performance gaps. Use tailored reports for different stakeholder levels, from executive summaries for leadership to detailed operational analyses for technical teams.
- 3. Maintain Comprehensive Change Records:** Document all changes, including rationale, approvals, and implementation details, in a centralized repository. Comprehensive records ensure traceability, support audits, and facilitate knowledge transfer across teams.

5.1 Data Architecture and Modeling KPIs

As part of the Master & Reference Data Management KPIs specification, the entity shall establish key performance indicators (KPIs) to measure the state and effectiveness of its Data Architecture and Modeling capabilities. These KPIs shall include, at a minimum:

- Metrics to track progress in transforming current state data architecture towards a target state data architecture
- Data Architecture Compliance (i.e.: percentage of compliant target state data architecture layers)
- Percentage completeness of analytical data model
- Adherence to Data Modeling Standards (i.e.: number and percentage of compliant data models)

5.1.A

Establish KPIs for Data Architecture & Modeling: Define and establish KPIs to measure and track the effectiveness of data architecture and modeling initiatives, ensuring alignment with best practices and governance standards. These KPIs should include:

- **Progress Toward Target Data Architecture:** Measure improvements in the Entity's data architecture transformation, tracking progress in migrating from baseline to target state.
- **Data Architecture Compliance:** Track the percentage of percentage of compliant target state data architecture layers.
- **Adherence to Data Modeling Standards:** Measure compliance with defined modeling guidelines, including naming conventions, relationship definitions, and data type standardization.
- **Data Model Completeness:** Assess the percentage completeness of the Analytical Data Models.
- **Data Lineage & Traceability Metrics:** Evaluate the percentage of datasets with documented data lineage, ensuring transparency in data movement and transformations.

Asset Guidance

The **Data Management KPIs Register Template** includes a table that provides a brief overview of identified KPIs. The Entity can use this template to define each KPI, along with its calculation methodology, target value, and status.

Illustrative

 Data Management KPIs Register Template								
The following table is a template containing what is needed to provide a brief overview of the performance of data management initiatives								
Field	Description							
KPI ID	A unique identifier assigned to each Key Performance Indicator (KPI).							
KPI Name	A short, descriptive name that summarizes what the KPI measures.							
KPI Type	Operational or Strategic KPI (if applicable).							
Calculation Methodology	The formula / approach (if any) used to calculate the KPI value.							
Target Value	The predefined value or benchmark to achieve for the KPI.							
Last Updated Date	The date when the KPI's current value was last recorded or updated.							
Status	On Track/Off Track.							
Comments	Additional notes or explanations regarding the KPI's performance.							
Owner	Outlines whether the business or IT is the owner of the KPI.							
KPI ID	KPI Name	KPI Type (if applicable)	Calculation Methodology	Target Value	Last Updated Date	Status	Comments	Owner

5.1.A
Cont.**Key Message / Information**

The Entity must establish a clear process for reviewing and updating KPIs for Data Architecture and Modeling to ensure they remain relevant and aligned with evolving business needs and data quality standards. This process should include setting regular intervals for KPI evaluation, incorporating stakeholder feedback, and assessing the effectiveness of each KPI in measuring data quality across dimensions such as accuracy, completeness, and consistency. By periodically revisiting these metrics, the Entity can refine its approach to address emerging challenges, ensure compliance with data quality standards, and leverage advancements in data management practices to drive improvements in data architecture and modeling.

RACI Matrix – Data Architecture and Modeling KPIs*Indicative*

Sr No.	Activity	Data Management and Security Director	Data Management and Governance Section Head	Data Governance Specialist	Data Architect
5.1.A	Establish KPIs for Data Architecture & Modeling	I	A	R	R

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Establish Data Architecture and Modeling KPIs

5.2 Data Architecture and Modeling Monitoring and Reporting

This activity will guide the Entity through the steps to establish monitoring and reporting mechanisms for Data Architecture & Modeling, ensuring compliance with governance standards and performance objectives. The Entity will implement continuous monitoring mechanisms to track key performance indicators (KPIs) on a quarterly basis, enabling consistent oversight of data architecture and modeling effectiveness.

Process – Data Architecture and Modeling Monitoring and Reporting

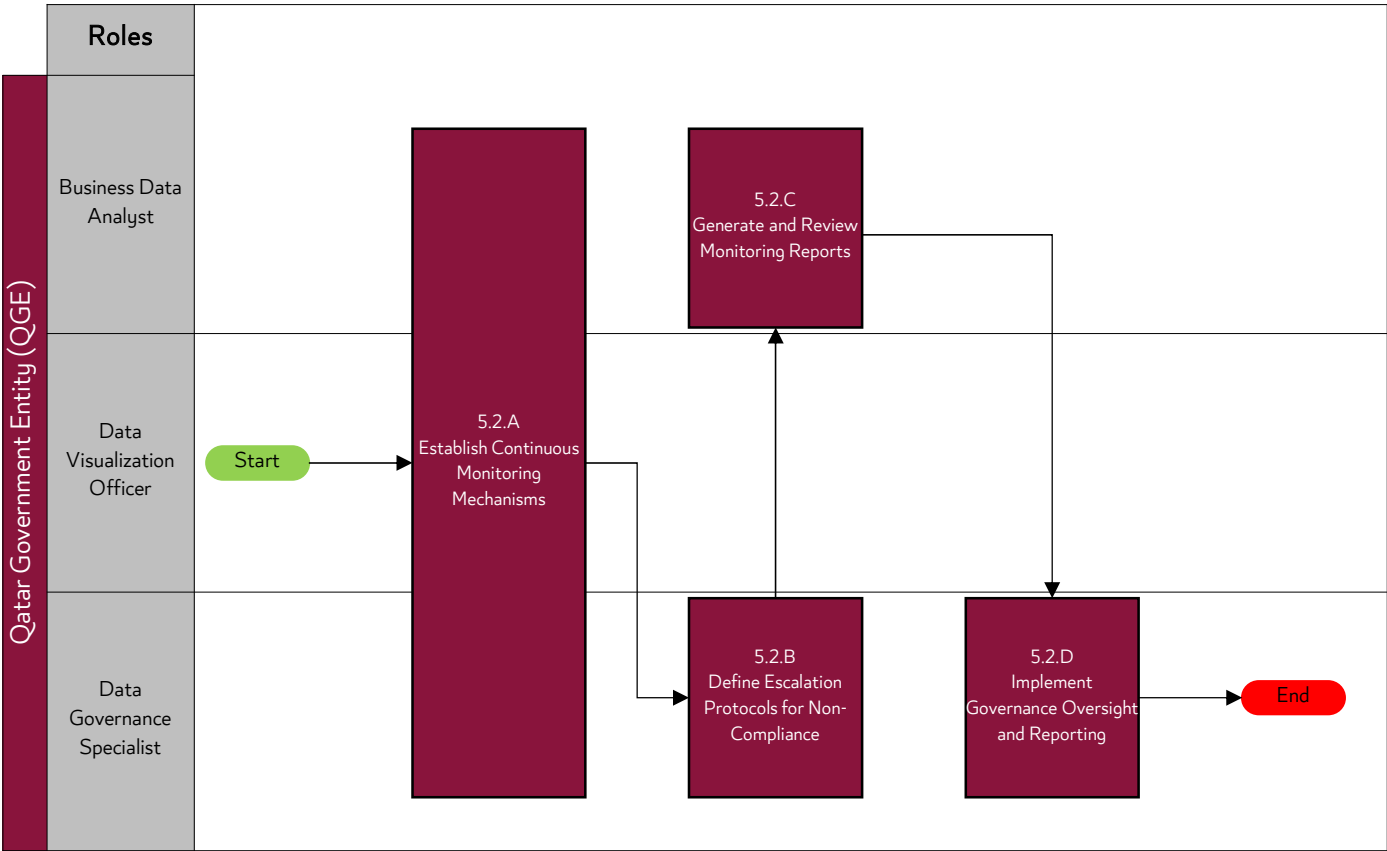


Figure 6: Data Architecture and Modeling Monitoring and Reporting Process Map

Legend: Process Step Decision Process Start Process End Document Off-Page Reference

5.2.A

Establish Continuous Monitoring Mechanisms: The Entity must implement systematic monitoring processes to track the performance and compliance of data architecture and modeling activities. The Entity should:

- Establish a quarterly monitoring schedule to review data architecture and modeling KPIs, such as model currency, metadata completeness, alignment with approved standards, and architecture reuse.
- Leverage dashboards and reports to visualize architecture adoption trends, identify deviations from defined modeling standards, and monitor progress against the data architecture roadmap.
- Ensure that monitoring mechanisms are integrated with the Entity's broader governance framework, supporting proactive decision-making, risk mitigation, and compliance with national data standards.

Asset Guidance

A Data Architecture Monitoring Dashboard can provide insights into the state and effectiveness of data architecture and modeling practices. It may include metrics such as:

- Percentage of systems aligned with the target architecture
- Completeness and consistency of data models and metadata
- Frequency of updates to architectural documentation
- Number of deviations from modeling standards or governance requirements

This illustrative example can serve as a reference for Entities when designing dashboards that effectively monitor data architecture performance, compliance, and maturity over time.

Illustrative



5.2.B

Define Escalation Protocols for Non-Compliance: To maintain governance and accountability, the Entity must develop structured escalation procedures to address performance deviations and non-compliance issues. The Entity should:

- Define clear thresholds for acceptable performance, ensuring that deviations from data modeling or architecture standards trigger corrective actions.
- Establish a tiered escalation framework, specifying who is responsible for reviewing, investigating, and resolving issues at each level.
- Assign the Data Committee as the oversight body, ensuring that all escalations related to data architecture and modeling compliance are reviewed and addressed efficiently.
- Integrate escalation workflows into governance processes, ensuring that corrective actions are documented and monitored.

5.2.C

Generate and Review Monitoring Reports: To provide insights into data architecture and modeling performance, the Entity must generate structured monitoring reports that summarize KPIs, compliance status, and governance adherence. The Entity should:

- Compile monitoring reports on a quarterly basis, summarizing KPI performance.
- Include trend analysis to assess performance improvements or risks, ensuring continuous optimization.
- Highlight compliance violations, unresolved gaps, or areas requiring immediate intervention, enabling proactive decision-making.
- Ensure transparency in reporting, making reports accessible to data architects, governance teams, and IT leadership.

5.2.D

Implement Governance Oversight and Reporting: To ensure accountability and alignment with strategic goals, the Entity must establish a governance framework for reviewing monitoring reports and ensuring compliance. The Entity should:

- Schedule periodic reviews with the Data Committee, ensuring that monitoring reports are analyzed and corrective actions are tracked.
- Use monitoring insights to refine data modeling practices, ensuring alignment with business needs.
- Document resolutions and adjustments made based on monitoring insights, ensuring that improvements are continuously implemented.

RACI Matrix - Data Architecture and Modeling Monitoring and Reporting

Indicative

Sr No.	Activity	Data Committee	Data Management and Security Director	Data Management and Governance Section Head	Data Architect	Business Data Analyst	Data Visualization Officer	Data Governance Specialist
5.2.A	Establish Continuous Monitoring Mechanisms	I	I	A	C	R	R	R
5.2.B	Define Escalation Protocols for Non-Compliance	I	I	A	C			R
5.2.C	Generate and Review Monitoring Reports		I	A	C	R		C
5.2.D	Implement Governance Oversight and Reporting	C	I	A	C			R

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Generate KPI Dashboards



Generate Monitoring Reports

5.3 Data Architecture Change Management Process

The Data Architecture Change Management Process ensures a structured approach for reviewing, approving, and implementing changes to both current and target state data architectures. This process aligns architectural modifications with the Entity's strategic goals while maintaining consistency, governance, and traceability.

Process – Data Architecture Change Management Process

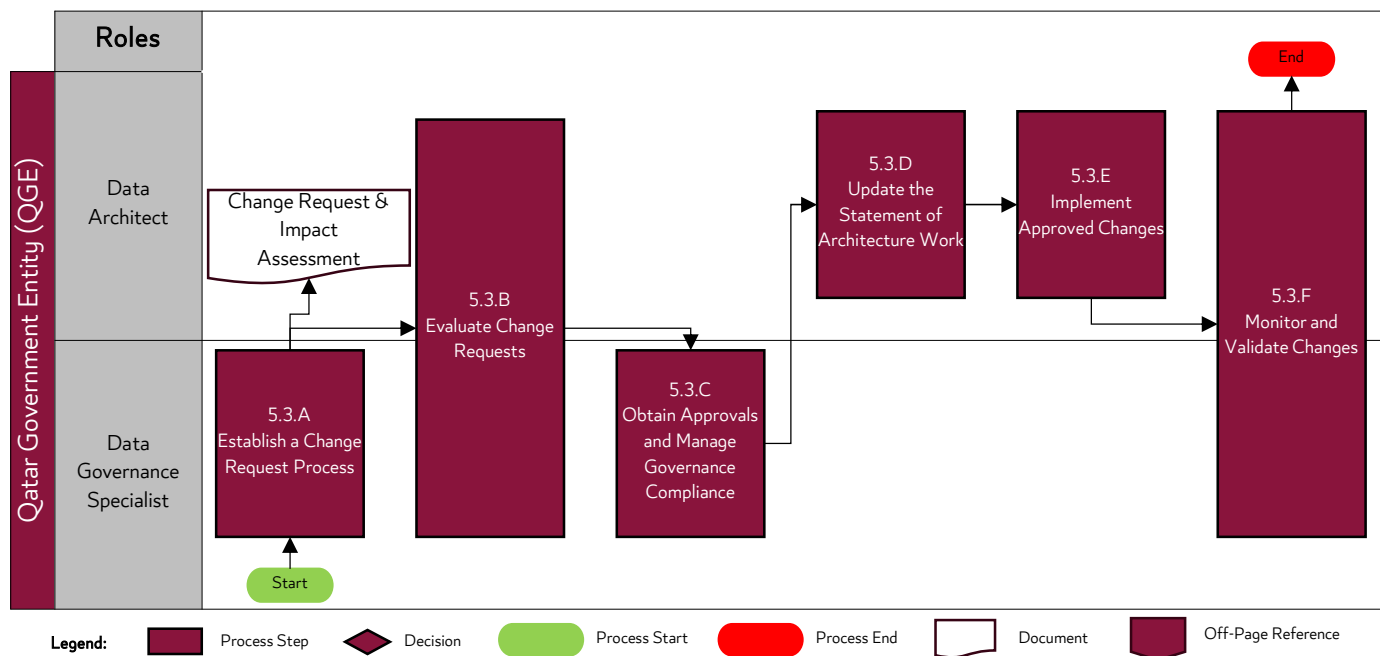


Figure 7: Data Architecture Change Management Process Map

5.3.A
(1/2)

Establish a Change Request Process: To ensure a structured and governed approach to handling data architecture and modeling change requests, the Entity must define a formalized process for submitting, evaluating, and tracking changes. The Entity should:

- Develop a standardized approach for submitting change requests, capturing details such as the nature of the change, justification, and affected areas (e.g., processes, systems, data models, or roles).
- Classify change requests based on priority and impact level, ensuring that changes undergoing review are assessed for technical feasibility, business alignment, and governance compliance.
- Identify key stakeholders affected by the change, including departments, teams, and individuals who may be impacted by modifications to data architecture and models.
- Assess potential risks associated with the requested change, including system downtime, data integrity concerns, or resistance to adoption.
- Define mitigation approaches to address identified risks, such as training, technical support, or phased rollouts to ensure smooth implementation.
- Establish an approval process, outlining the qualifications for review, assessment procedures, and criteria for approval or rejection.
- Track and monitor the status of change requests, ensuring that all changes progress through appropriate stages, from submission to approval and implementation.

A

5.3.A
Cont.
(2/2) **Asset Guidance**

The Data Management Change Management Template includes a Change Request & Impact Assessment template that the Entity can use for documenting changes related to data management and their impact

Illustrative

Change Request & Impact Assessment						
The following table serves as a template for documenting changes related to data management and their impact						
Field	Description					
Change ID	Unique identifier for the change request or project					
Change Title	A short, descriptive title summarizing the change request or project					
Change Description	Detailed description of the requested change, including what is being modified or introduced and the rationale for the change					
Request Date	The date when the change request was submitted					
Requested By	Name and department of the individual or team requesting the change					
Change Owner	Person or department responsible for managing and overseeing the change implementation					
Priority Level	The urgency of the change request (e.g., High, Medium, Low)					
Impact Level/Change Severity	Estimated impact or severity of the change (High, Medium, Low) on departments, systems, or processes					
Impacted Departments/Teams	List of departments, teams, or individuals impacted by the change					
Change Type	Type of change (Process Change, Technology Change, Role Change)					
Impacted Areas	Specific areas affected by the change (Systems, Processes, Data, Roles)					
Stakeholders Affected	Key stakeholders who will be impacted by the change					
Potential Risks	Risks associated with the change (e.g., Resistance to Change, System Downtime)					
Mitigation Approach	Strategies to mitigate risks (e.g., training, early communication, technical support)					
Requested Implementation Date	Proposed date by which the change should be implemented					
Change Qualification Status	Classification indicating whether the change request meets the necessary criteria for further review and assessment					
Change Assessment Date (If Applicable)	Date when the change impact assessment was conducted					
Assessment Outcome (If Applicable)	Outcome of the assessment (e.g., Change approved, Change Rejected, Further Review Needed)					
Approval Status	The status of approval for the change (Pending, Approved, Rejected)					
Change Status	The current status of the change request or project (In Progress, Completed, Deferred)					
Comments	Any additional notes or comments regarding the request or project					

Change ID	Change Title	Change Description	Request Date	Requested By	Change Owner	Priority Level

Impact Level/Change Severity	Impacted Departments/Teams	Change Type	Impacted Areas	Stakeholders Affected	Potential Risks	Mitigation Approach

Requested Implementation Date	Change Qualification Status	Change Assessment Date (If Applicable)	Assessment Outcome (If Applicable)	Approval Status	Change Status	Comments

5.3.B

Evaluate Change Requests: To ensure systematic and informed decision-making, the Entity must evaluate each change request, assessing its feasibility, risks, and impact on data architecture and modeling .The Entity should:

- Analyze the impact of the proposed change on data models, governance policies, technical infrastructure, and business processes, ensuring alignment with strategic objectives.
- Evaluate change severity and risk factors, identifying potential disruptions, system dependencies, and compliance considerations.
- Assess mitigation approaches, determining necessary support mechanisms such as phased rollouts, technical validations, or stakeholder training.
- Document findings, capturing impact analysis results, assessment outcomes, and approval recommendations.
- Ensure transparency in decision-making, making evaluations accessible to key stakeholders and governance bodies for final approval.

**Key Message / Information**

The Change Request & Impact Assessment template contains columns for the Entity to document its impact assessment findings

A

5.3.C

Obtain Approvals and Manage Governance Compliance: Route the evaluated change requests to the designated approvers for final decision-making. Ensure that the proposed changes comply with governance policies, standards, and security protocols.

A

5.3.D

Update the Statement of Architecture Work: Revise the Statement of Architecture Work (SAW) to reflect approved changes. Include details such as the scope, rationale, and implementation plan for each modification. Ensure that updates to the SAW are version-controlled and shared with relevant stakeholders.

A

5.3.E

Implement Approved Changes: Collaborate with the Data Architect for implementation of the approved changes in alignment with the SAW. Use project management tools to track tasks, milestones, and resource allocations. Conduct regular status updates to ensure progress aligns with the approved plan.

A

5.3.F

Monitor and Validate Changes: After implementing changes, monitor the value addition to validate intended objectives of the change request. Validate the changes against predefined KPIs or success criteria, such as performance improvements or compliance with standards. Schedule a review session to assess the overall success of the change implementation. Gather feedback from stakeholders to identify lessons learned and opportunities for process improvement.

A

RACI Matrix - Data Architecture Change Management Process

Indicative

Sr No.	Activity	Data Management & Security Director	Data Architecture and Analytics Section Head	Data Management and Governance Section Head	Data Governance Specialist	Data Architect
5.3.A	Establish a Change Request Process	I	C/A		R	C
5.3.B	Evaluate Change Requests	I	C/A	C	R	R
5.3.C	Obtain Approvals and Manage Governance Compliance	I	C/A		R	C
5.3.D	Update the Statement of Architecture Work	I	C/A			R
5.3.E	Implement Approved Changes	I	C/A		C	R
5.3.F	Monitor and Validate Changes	I	C/A			R

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Maintain an updated Statement of Architecture Work (SAW)

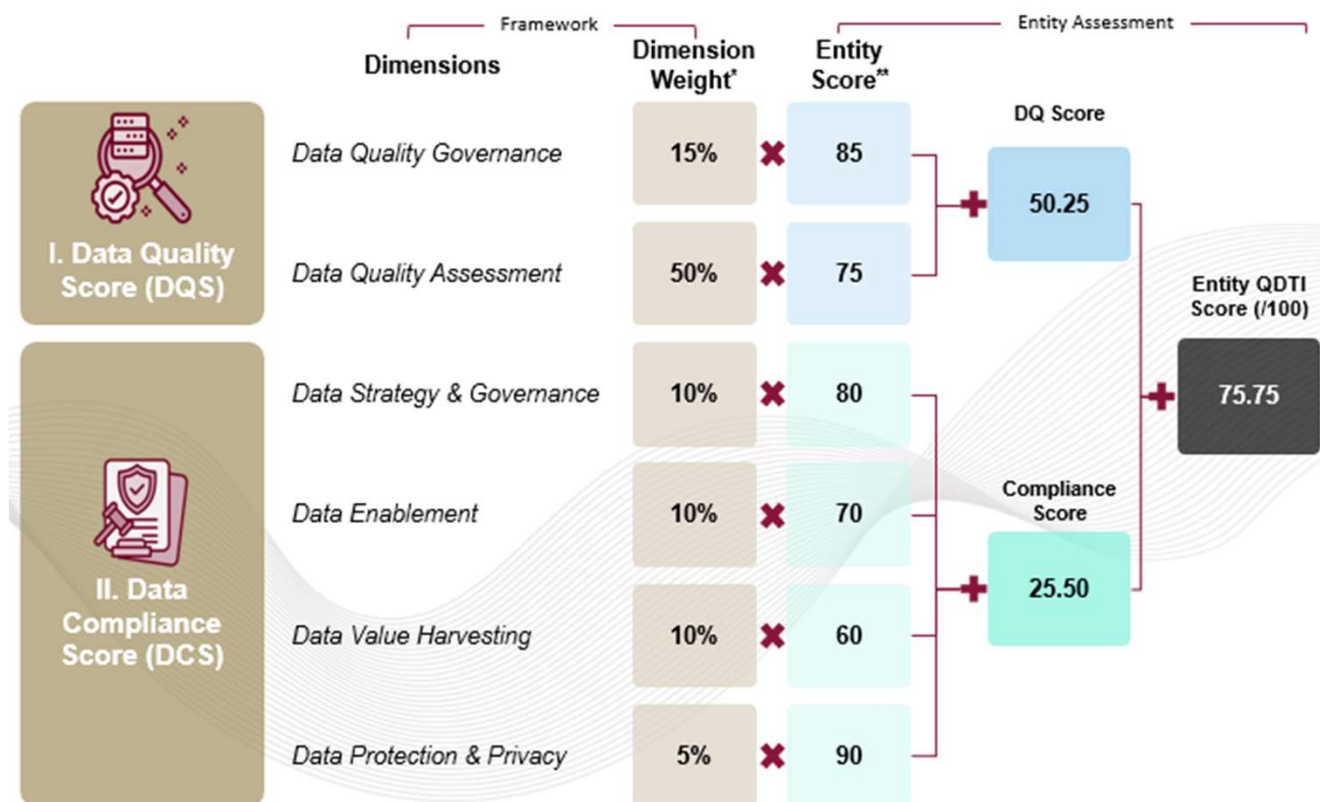


Execute Approved Change Implementations

5.4 QDTI

The Qatar Data Trust Index (QDTI) is designed to measure and enhance data integrity across national entities by promoting adherence to governance standards and ensuring the accuracy, reliability, and responsible management of data. This initiative aims to build trust and support a robust digital economy. The overall QDTI score is derived from two dimensions: the Data Quality Score (DQS) and the Data Compliance Score (DCS), which together combine to determine the Entity's QDTI score.

This section outlines the steps needed to gather the necessary information for calculating the DCS.



***Note:** The sum of the weights across all dimensions should add up to 100%

****Note:** This score is calculated by computing the weighted average score of all the sub-dimensions under each respective Dimension (scores are on a scale of 0 to 100)

Figure 8: Illustrative QDTI Scoring Methodology for an Entity

5.4.A

Initiate Data Architecture and Modeling Compliance Assessment: The Data Management and Governance Section Head assigns the Data Compliance Specialist to oversee the assessment process. The specialist is responsible for managing the evaluation of the Entity's adherence to the National Data Architecture and Modeling Standards.



Key Message / Information

The Data Architecture and Modeling Domain falls under the Data Enablement Dimension in the National Data Management Framework and, consequently, within the QDTI Dimension.

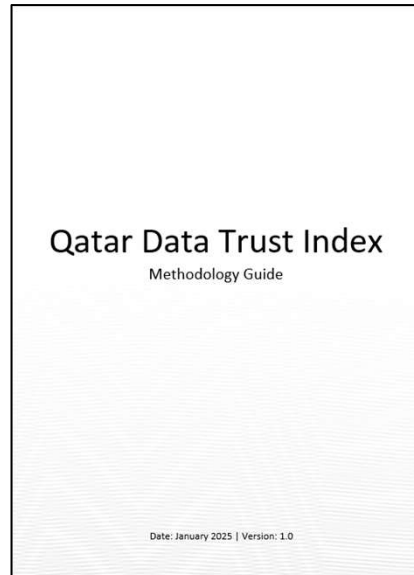
5.4.B

Conduct Data Architecture and Modeling Assessment: Initiate the self-assessment to evaluate adherence to the National Data Architecture and Modeling Domain Standards. This assessment will ensure compliance and accountability while identifying areas for improvement.

Asset Guidance

The **Qatar Data Trust Index Methodology Guide** can be leveraged to guide the Entity through the steps required to conduct the self-assessment.

Illustrative



Asset Guidance

The **QDTI Assessment Template** can be used to document compliance with the National Data Standards. This template is intended for use prior to the DGO launching the self-assessment on its platform to calculate the overall QDTI Score.

Illustrative

Domain	Control	Control Description
DQM	DQM.1.1.1 Data Quality Policy	As part of the Data Quality Policy control, the Entity shall create and maintain an enterprise-wide Data Quality Management Policy...
DQM	DQM.1.1.2 Data Quality Policy	As part of the Data Quality Policy control, the Entity shall publish and distribute a Data Quality Management Policy...
DQM	DQM.1.1.3 Data Quality Planning	As part of the Data Quality Planning control, the Entity shall perform initial data quality profiling for the identified Critical Data Elements (CDEs)...
DQM	DQM.1.1.4 Data Quality Planning	As part of the Data Quality Planning control, the Entity shall develop a data quality plan aligned with its data strategy and profiling results...

Specification	Control Specification	Capability Level	Compliance
DQM.1.1.1.1 Data Quality Policy Creation	The Entity shall create a data quality policy in alignment with the NDQAF published by the NPC...	Foundational	Compliant
DQM.1.1.2.1 Data Quality Policy Publication	The Entity shall publish a data quality policy....	Foundational	Compliant
DQM.1.1.3.1 Initial Data Quality Profiling	The Entity shall identify CDEs and perform initial data quality profiling for the identified CDEs....	Foundational	Compliant
DQM.1.1.4.1 Data Quality Plan Design	Based on the initial data profiling results, the Entity shall develop a data quality plan in alignment with the Entity's data strategy...	Foundational	Compliant



Key Message / Information

The DGO will launch the self-assessment on its platform and will provide toolkits and host workshops to guide entities through the self-assessment process during the first year.

5.4.C

Submit National Data Architecture and Modeling Score to DGO: Submit the National Data Architecture and Modeling Compliance Score to the DGO for evaluation



Key Message / Information

The DGO will calculate the Entity's overall Qatar Data Trust Index (QDTI) score only after the Entity has completed both the Data Quality Score (DQS) assessment and the required self-assessments for all other data management domains to determine the Data Compliance Score (DCS). Once all assessments are completed, the DGO will aggregate the compliance scores across domains to finalize the DCS, which will then be combined with the DQS to determine the Entity's overall QDTI score.

RACI Matrix - QDTI

Indicative

Sr No.	Activity	DGO	Data Management & Security Director	Data Management and Governance Section Head	Data Compliance Specialist	Data Governance Specialist
5.4.A	Initiate Data Architecture and Modeling Compliance Assessment		I	A	R	
5.4.B	Conduct Data Architecture and Modeling Assessment		I	A	R	C
5.4.C	Submit Data Architecture and Modeling Score to DGO	I	I	A	R	I

RACI Matrix Legend: **R:** Responsible **A:** Accountable **C:** Consulted **I:** Informed

Based on this Activity, the Entity can



Calculate Data Architecture & Modeling Compliance Score

6. Data Architecture & Modeling Documentation



6. Data Architecture & Modeling Repository

What is a Data Architecture & Modeling Repository and why does the Entity need it?

The Data Architecture & Modeling Repository establishes a centralized platform for storing, managing, and maintaining all documentation related to data architecture and modeling, including architectural diagrams, data model diagrams, supporting metadata, reference materials and so forth. Additionally, it ensure that these documents are properly versioned, governed, and accessible to authorized users.

This section will detail the activity below to ensure that the repository is purpose-built, technically configured, and governed in line with the national data standards, enabling traceability, consistency, and collaboration across data initiatives.

6.1 Data Architecture & Modeling Repository

Guiding Principles

The guiding principles for Data Architecture & Modeling Documentation are:

1. **Centralize Documentation for Accessibility:** Establish a centralized repository for all data architecture and modeling artifacts, ensuring that stakeholders have easy access to the latest documentation.
2. **Ensure Comprehensive and Standardized Content:** Maintain detailed and standardized documentation, including diagrams (UML, IDEF1X, Entity-Relationship Diagrams) and reference materials, to support a consistent understanding of data models across teams.
3. **Enable Robust Version Control:** Implement version control mechanisms to track changes, maintain historical records, and ensure traceability. This ensures accuracy and simplifies auditing processes.
4. **Capture Relationships Between Systems:** Document master-slave or federated relationships between datasets, providing clarity on data flow, ownership, and dependencies across systems.
5. **Integrate Structured and Unstructured Data Models:** Ensure documentation covers both structured and unstructured data to provide a comprehensive view of the Entity's data landscape, supporting diverse analytical and operational needs.

6.1 Data Architecture & Modeling Repository

The Data Architecture & Modeling Repository establishes a centralized platform for storing, managing, and maintaining all documentation related to data architecture and modeling.

Process – Data Architecture & Modeling Repository

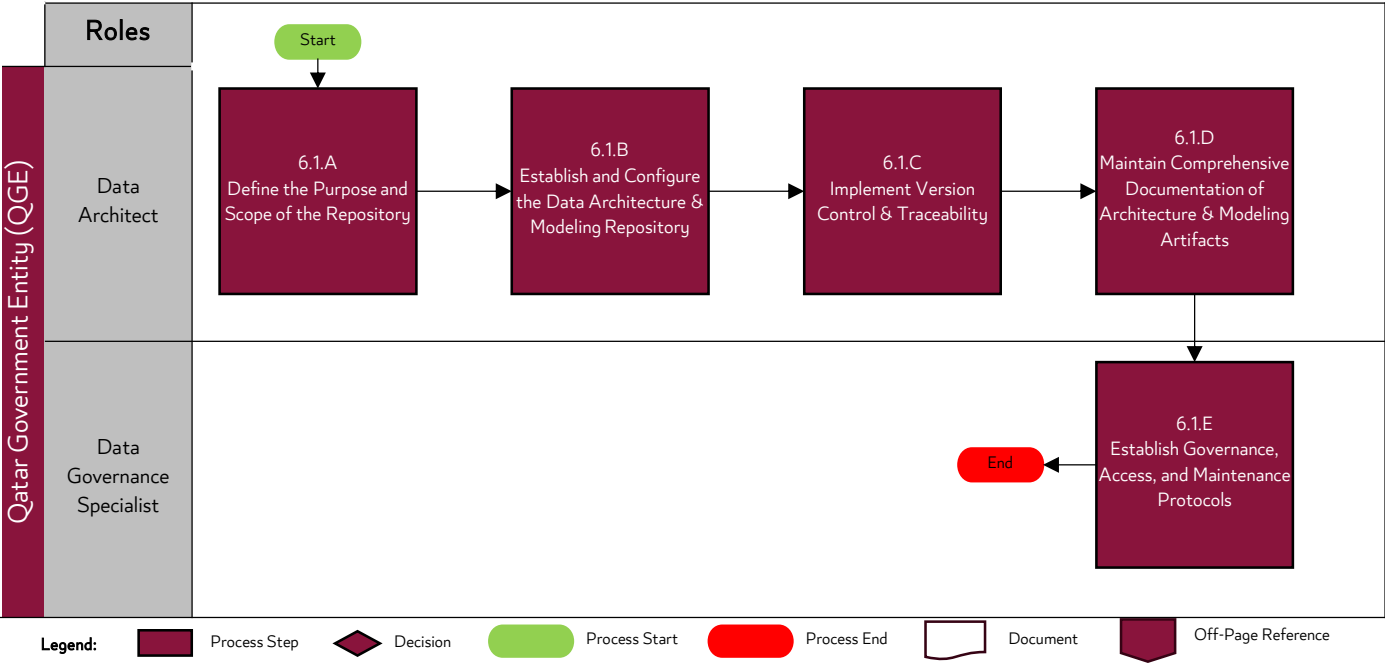


Figure 9: Data Architecture & Modeling Repository Process Map

6.1.A
(1/2)

Define the Purpose and Scope of the Repository: The Data Architecture & Modeling Repository is a centralized, version-controlled environment used to store, manage, and govern all documentation and artifacts related to the Entity’s data architecture and modeling activities. It ensures that models, diagrams, lineage records, and metadata are consistently maintained, traceable, and aligned with the National Data Standards.

This repository acts as the single source of truth for architecture and modeling assets, enabling stakeholders to access accurate, approved, and up-to-date documentation for analytics, governance, integration, and compliance purposes.

i. Purpose of the Repository:

- Support governance by enforcing traceability, accountability, and standardized documentation.
- Enable reuse and consistency of data architecture and modeling assets across projects and domains.
- Provide a single, authoritative location for approved conceptual, logical, and physical data models.
- Facilitate collaboration between data architects, modelers, analysts, and governance teams.
- Ensure readiness for audit and compliance through versioned records and clear ownership.

6.1.A
Cont.
(2/2)

- ii. **Scope of Repository Contents:** The repository must store all official and approved documentation developed as part of data architecture and modeling activities. This includes, but is not limited to:
- a. **Data Architecture Documentation:**
 - Target Data Architecture Diagrams
 - Integration Points Documentation
 - Architecture Review & Approval Records
 - b. **Data Models Documentation:**
 - Conceptual, Logical, and Physical Data Model Diagrams
 - Data Model Metadata Documentation
 - Target Fact & Dimension Table Documentation
 - Fact-Dimension Relationship Documentation
 - c. **Data Lineage Documentation:**
 - Attribute-level Data Lineage for fact and dimension tables.
 - Includes transformation logic, processing steps, target system usage, and lineage type.
 - d. **Model Governance Artifacts:**
 - Version control records including model ID, status (draft/published/retired), and change history.
 - RACI assignments for all models and lineage assets.
 - Approval logs from internal stakeholders and the DGO.
 - e. **Reference Assets & Standards:**
 - Templates used throughout the data architecture & modeling exercise.
 - Entity and NPC Data Architecture & Modeling Guidelines and Definitions for consistency and alignment
 - Any approved standards or formats used by the Entity to support model design.

6.1.B
(1/2)

Establish and Configure the Data Architecture & Modeling Repository: Once the purpose and scope of the repository have been defined, the Entity must establish and configure the repository environment that will serve as the centralized platform for storing all data architecture and modeling documentation.

This step involves setting up the technical infrastructure, folder structure, metadata templates, and user access framework needed to operate the repository effectively.

- i. **Select and Configure the Repository Platform:** The Entity must choose a repository platform that supports version control, structured documentation, and secure access.

6.1.B
Cont.
(2/2)

a. Platform Options / Considerations:

- Modeling tools with built-in repositories (e.g., ER/Studio, PowerDesigner, Erwin)
- Document or metadata management systems (e.g., SharePoint)
- External version control tools (e.g., Git-based platforms)
- Structured file systems with access control (least preferred; interim only)

b. Repository Setup Considerations:

- Ensure the platform supports version tagging, audit logging, and user permissions.
- Verify compatibility with standard documentation formats (e.g., PDF, PNG, XML, XLSX, modeling exports).

ii. Design the Folder & Classification Structure: To enable clarity, searchability, and traceability, the repository must follow a standardized structure.

- a. Data Architecture Documentation:
- b. Group by Domain, Department, or Project

iii. Perform Initial Testing & Access Setup:

a. Assign Repository Admin Roles:

- Responsible for configuring and maintaining the platform.

b. Grant Role-Based Access:

- Provide edit access to model owners, governance leads
- Assign read-only access for consumers and reviewers

6.1.C
(1/2)

Implement Version Control & Traceability: To ensure that data architecture and modeling documentation is reliable, auditable, and consistently maintained, the Entity must implement version control and traceability mechanisms within the Data Architecture & Modeling Repository. This ensures that all model updates, revisions, and approvals are properly tracked, in alignment with the National Data Standards.

i. Establish Version Control for Architecture & Modeling Assets: All data architecture and conceptual, logical, and physical data models, as well as supporting documentation, must be version-controlled to track their evolution, ensure accountability, and maintain historical context.

a. Define a Standard Versioning Format:

- Apply a consistent format to all models and documents:
- [Entity Initials]-[Reference Number]-[Version] (e.g., NPC-001-V1.0)
- Maintain major and minor version increments (e.g., V1.0, V1.1, V2.0) to distinguish between significant and minor updates.

- b. **Track Model Status.** Every version of a data model must include a status field to indicate lifecycle stage:
 - **Draft:** In development, not yet validated.
 - **Published:** Approved and actively used.
 - **Retired:** No longer in use but retained for audit purposes.
 - c. **Maintain Version Metadata.** Metadata for each version must include:
 - Date of creation and last modification.
 - Model owner and approver(s).
 - Summary of changes or updates made.
- ii. **Ensure Traceability of Changes:** The Entity must ensure that all changes to data models and architectural artifacts are traceable across time and stakeholders.
 - a. **Maintain a Change Log.** For each versioned model, maintain a change log that includes:
 - Description of changes (e.g., added attribute, modified relationship).
 - Date of change.
 - Name and role of the person making the change.
 - Reason for change (if applicable).
 - b. **Link Changes to Source Documentation.** Metadata for each version must include:
 - Ensure that changes are traceable to the original model or related document (e.g., lineage template)
 - Cross-reference changes with relevant approval records or governance decisions.
- iii. **Integrate Version Control into Governance Workflows:** Version control must be integrated with the Entity's broader governance processes to ensure transparency, approval tracking, and accountability.
 - a. **Approval Workflow Integration:**
 - Require that model updates go through formal review and approval before publishing.
 - Maintain review logs and sign-off records from architects, analysts, and data governance teams.
 - b. **Version Access Management:**
 - Limit editing access to designated model owners or stewards.
 - Provide read-only access to all approved/published versions for other stakeholders

6.1.D

Maintain Comprehensive Documentation of Architecture & Modeling Artifacts: To ensure transparency, traceability, and reusability, the Entity must maintain complete and well-structured documentation of all data architecture and modeling activities within the repository. This includes diagrams, templates, lineage records, and metadata, and so forth.

This documentation supports Entity-wide collaboration, enables consistent implementation practices, and facilitates ongoing governance and audit readiness.

- i. **Store Previously Defined Architecture & Modeling Diagrams:** In alignment with the repository scope defined in 6.1.A, the Entity must ensure that all major architectural and data modeling diagrams and documentation are stored and maintained in the repository.

- a. **Required Diagrams and Documentation:**

- Data Architecture Documentation.
- Data Model Documentation.
- Data Lineage Documentation.
- Model Governance Artifacts.
- Reference Assets and Standards.

- b. **Format & Versioning:**

- Diagrams should be maintained in standardized formats (e.g., PDF, PNG, XML, modeling tool exports).
- Each diagram must follow the same version control and metadata tagging rules defined in 6.B.

- c. **Organizational Structure:**

- Documentation must be logically organized by domain, project, or model layer (e.g., Conceptual, Logical, Physical).
- File naming and folder structures should ensure easy retrieval and version tracking.

- ii. **Apply Consistent Documentation Practices.** To maintain high quality and consistency, the Entity must:

- Apply standard naming conventions and design notations as per the Entity's and NPC's Data Architecture & Modeling guidelines.
- Ensure completeness and clarity in all documentation before publishing to the repository.
- Require that all finalized models and diagrams are accompanied by supporting metadata and lineage.

6.1.E
(1/2)

Establish Governance, Access, and Maintenance Protocols: To ensure that the Data Architecture & Modeling Repository remains secure, up-to-date, and aligned with governance objectives, the Entity must implement clear protocols for its ongoing management, user access, and operational oversight. These protocols ensure that the repository supports collaborative usage, role-based access, and lifecycle maintenance in alignment with the Entity's broader data governance framework and NPC's National Data Standards.

- i. **Define Governance Roles & Responsibilities:** The Entity must define clear ownership and accountability for managing the repository and its contents.

- a. **Assign Repository Roles:**

- **Repository Administrator:** Oversees structure, access, and system-level configuration.
- **Data Architects / Model Owners:** Responsible for uploading and maintaining architecture/modeling artifacts.
- **Data Stewards / Governance Teams:** Review and validate models, metadata, and lineage.
- **General Users (Read-Only):** Business analysts, developers, auditors, and others who require access to published models.

- b. **Establish RACI for Key Activities.** Define a Responsible, Accountable, Consulted, Informed (RACI) matrix for:

- Version control and change approvals
- Metadata updates and review
- Model publishing and archival
- Audit and compliance reporting

- ii. **Manage User Access & Permissions.** The repository must be secured with RBAC to ensure that only authorized users can create, modify, or publish content.

- a. **Assign Repository Roles:**

- **Edit Access:** Limited to model owners, data architects, and governance leads.
- **Review/Approve Access:** Reserved for designated reviewers or approvers.
- **Read-Only Access:** For broader consumption by analysts, business users, and IT staff.

- b. **Periodic Access Reviews:**

- Conduct regular access audits to ensure permissions remain valid.
- Revoke access for inactive or unauthorized users as part of governance housekeeping.

- ii. **Define Repository Maintenance & Update Protocols:** The Entity must ensure that the repository remains current, usable, and well-organized over time.

- a. Refresh & Review Schedule:
- Conduct scheduled reviews of models, metadata, and lineage (e.g., quarterly or semi-annually).
 - Archive outdated or retired assets using clear retention policies.
- b. Repository Clean-Up & Documentation Standards:
- Apply naming conventions, tagging, and folder structures consistently across all files.
 - Remove duplicate or obsolete files to avoid confusion and maintain integrity.
 - Maintain a change log or audit report summarizing key updates to the repository.

RACI Matrix - Data Architecture & Modeling Repository

Indicative

Sr No.	Activity	Data Management and Security Director	Data Management and Governance Section Head	Data Architect	Metadata Specialist	Data Governance Specialist
6.1.A	Define the Purpose and Scope of the Repository	I	A	R		c
6.1.B	Establish and Configure the Data Architecture & Modeling Repository		A	R	C	C
6.1.C	Implement Version Control & Traceability		A	R		
6.1.D	Maintain Comprehensive Documentation of Architecture & Modeling Artifacts		A	R	C	C
6.1.E	Establish Governance, Access, and Maintenance Protocols	I	A	C		R

RACI Matrix Legend:

R: Responsible

A: Accountable

C: Consulted

I: Informed

Based on this Activity, the Entity can



Develop a Centralized Repository Structure



Implement a Version Control System

7. Appendix



7.1 Templates and Frameworks

Asset No.	Asset Name	Description	File Name
1	Data Architecture & Data Modeling Handbook	Handbook provides guidance on the components of a typical Data Architecture and Data Models (Conceptual, Logical and Physical). It also provides guidelines for all types of Data Models.	NPC_DGO_QDKC_DAM_Data Architecture & Data Modeling Handbook
2	Current Data & Analytics Landscape Template	Workbook provides a single, standardized repository for documenting Entity wide data assets and solutions such as source systems, data platforms, analytics software, reports, dashboards, AI/ML models and generative AI solutions, ensuring visibility and streamlined governance.	NPC_DGO_QDKC_DAM_Current Data & Analytics Landscape Template
3	Integration Points Documentation Template	Template can be leveraged to capture key details about system integrations, data exchange mechanisms, and governance tracking.	NPC_DGO_QDKC_DAM_Integration Points Documentation Template
4	Fact & Dimension Tables Inventory	Template is designed to support entities in structuring, managing, and governing analytical data models by providing a standardized format for documenting fact and dimension tables, their attributes, relationships, and metadata.	NPC_DGO_QDKC_DAM_Data Architecture & Modeling Repository
5	Datasets and Attributes Documentation Template	Template is designed to support entities in documenting tables, attributes, and relationships .	NPC_DGO_QDKC_DAM_Datasets and Attributes Documentation Template
6	Baseline Data Lineage Template	Template serves as a reference for tracking and managing data movement, transformation, and dependencies across the Entity's data ecosystem. It provides a structured approach to documenting how data flows from its source to its final destination, ensuring transparency, consistency, and compliance with data governance policies.	NPC_DGO_QDKC_DAM_Baseline Data Lineage Template
7	Gap Analysis Template	Template Identify areas where improvements or changes are needed and how to bridge the gap by comparing the original data architecture & data models (conceptual, logical, and physical) with the target data models.	NPC_DGO_QDKC_DAM_Gap Analysis Template

Asset No.	Asset Name	Description	File Name
8	DM Gap Analysis Key Findings Template	Slide is designed to summarize key findings from a gap analysis exercise across a specific domain. It provides a structured view of identified gaps, their categories, descriptions, business impact, and priority level.	NPC_DGO_QDKC_DAM_DM Gap Analysis Key Findings Template
9	Analytical Data Model Gap Assessment Template	Workbook is to provide a structured, comprehensive framework for identifying, documenting, and prioritizing gaps in the analytical data models across an entity's data ecosystem.	NPC_DGO_QDKC_DAM_Analytical Data Model Gap Assessment Template
10	Initiative Card Template	Template used to outline key details of a specific initiative.	NPC_DGO_QDKC_General_Initiative Card Template
11	Data Management Domain Plan Template	Template that helps the Entity capture an overview of identified initiatives, detail their sub-activities, owners, stakeholders, and activity timeframe.	NPC_DGO_QDKC_General_Data Management Domain Plan Template
12	Architecture Components Template	Template provides a structured sheet for documenting key architecture components within the Enterprise Analytics Data Architecture.	NPC_DGO_QDKC_DAM_Architecture Components Template
13	Target Data Lineage	Template provides a structured, traceable approach for documenting data lineage across the Entity's data processing layers—raw, bronze, silver, and gold layers, to reports.	NPC_DGO_QDKC_DAM_Target Data Lineage Template
14	Data Model Metadata	Template provides a standardized structure for documenting metadata across the Conceptual, Logical, and Physical Data Models.	NPC_DGO_QDKC_DAM_Data Model Metadata
15	Tool Selection Criteria Template	Template provides a structured approach to selecting an appropriate data architecture and modeling tool to meet the organizational needs. It defines key features and evaluation criteria for assessing different tools.	NPC_DGO_QDKC_DAM_Tool Selection Criteria Template
16	Data Management KPIs Register Template	Template is designed to provide a standardized approach for documenting, tracking, and managing Key Performance Indicators (KPIs) related to data management activities.	NPC_DGO_QDKC_General_Data Management KPIs Register Template

Asset No.	Asset Name	Description	File Name
17	Data Management Change Management Template	Template is designed to support entities in planning and managing organizational changes related to data management initiatives.	NPC_DGO_QDKC_General_Data Management Change Management Template
18	Qatar Data Trust Index Methodology Guide	The Qatar Data Trust Index Methodology Guide can be leveraged to guide the Entity through the compliance self-assessment.	22012025_NPC_QDTI Methodology Guide_V1.0
19	QDTI Assessment Template	The QDTI Assessment Template can be used to document compliance with the National Data Standards.	23012025_NPC_QDTI Assessment Template_V1.0



المجلس الوطني للتخطيط
National Planning Council

دولة قطر • State of Qatar