Data Architecture

1 Introduction

The essential components of Data Architecture:

- Data Architecture Outcomes: The Data Architecture Artefacts
- Data Architecture Activities: To fulfil Data Architecture intentions
- Data Architecture Behaviour: Collaboration among roles with an Enterprise view

Data Architecture Is fundamental to data Management. The vast data of an organisation must be represented at various levels of abstraction so that it can be understood for management to make decisions.

An organisation's Data Architecture consists of:

- Master design documents at different abstraction
- formal enterprise data model – containing data names, metadata definitions conceptual and logical entities and relationships and business rules.

Data Architecture enables consistent data standardisation and integration across the enterprise.

Data Architecture artefacts constitute metadata and should be stored and managed in and enterprise architecture artefact repository.

ISO/IEV 42010:2007 Definition:

The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

1.1 Business Drivers

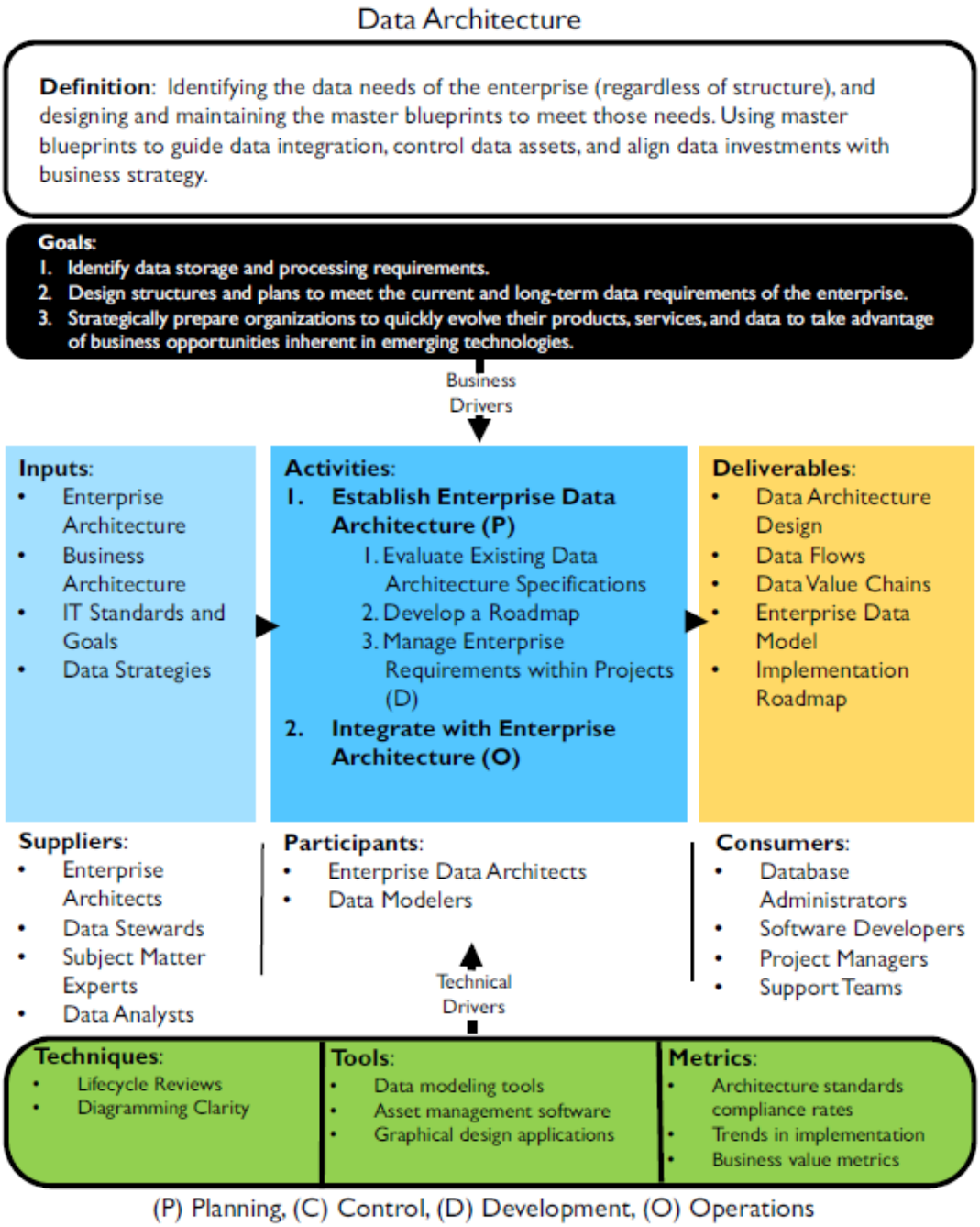
- Strategic preparation of evolution of products, services and data to take advantage of business opportunities in emerging technologies
- Translate business needs into data and system requirements
- Manage complex data delivery throughout the enterprise
- Facilitate alignment between business and IT
- Act as agents for transformation
- Influence measures of the value of data

Data Architecture is a bridge between business strategy and technology execution. Data architects create and maintain organisational knowledge about data and the systems through which it moves, which enables it to maintain data as an asset, and to increase the value it gets from data.

1.2 Data Architecture Outcomes and Practices.

Primary Data Architecture outcomes:

- Data storage and processing requirements
- Designs of structure and plans that meet current and long term data requirements of the enterprise



To reach goals Data Architects define and maintain specifications that:

- Define the current state of the organisation
- Provide standard business vocabulary for data and components
- Align Data Architecture with enterprise strategy and business architecture
- Express strategic data requirements
- Outline high level designs to meet these requirements
- Integrate with overall enterprise architecture roadmap

1.3 Essential Concepts

1.3.1 Enterprise Architecture Domains

Domain	Enterprise Business Architecture	Enterprise Data Architecture	Enterprise Applications Architecture	Enterprise Technology Architecture
Purpose	To identify how an enterprise creates value for customers and other stakeholders	To describe how data should be organized and managed	To describe the structure and functionality of applications in an enterprise	To describe the physical technology needed to enable systems to function and deliver value
Elements	Business models, processes, capabilities, services, events, strategies, vocabulary	Data models, data definitions, data mapping specifications, data flows, structured data APIs	Business systems, software packages, databases	Technical platforms, networks, security, integration tools
Dependencies	Establishes requirements for the other domains	Manages data created and required by business architecture	Acts on specified data according to business requirements	Hosts and executes the application architecture
Roles	Business architects and analysts, business data stewards	Data architects and modelers, data stewards	Applications architects	Infrastructure architects

1.3.2 Enterprise Architecture Frameworks

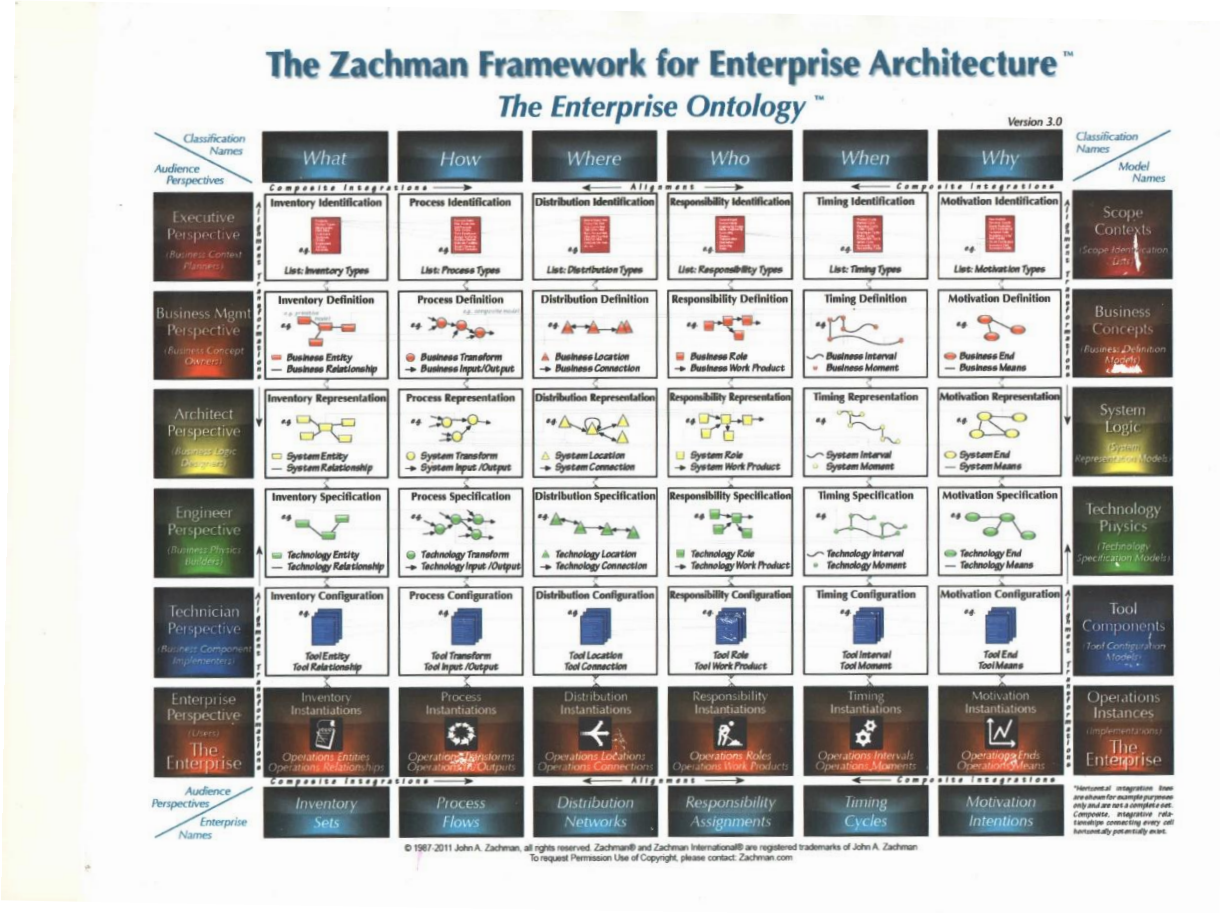
Provide a framework for thinking about and understanding architecture.

1.3.2.1 Zachman Framework for Enterprise Architecture

Developed by John A. Zachman in the 1980s. An ontology. The 6 x 6 matrix shows the models required to describe an enterprise, and the relationships between them. Each cell represents a unique type of design artefact defined by the intersection of row and column.

	What	How	Where	Who	When	Why	
Executive	Inventory Identification	Process Identification	Distribution Identification	Responsibility Identification	Timing Identification	Motivation Identification	Scope Context
Business Management	Inventory definition	Process Definition	Distribution Definition	Responsibility Definition	Timing Definition	Motivation Definition	Business Concepts
Architect	Inventory Representation	Process Representation	Distribution Representation	Responsibility Representation	Timing Representation	Motivation Representation	System Logic
Engineer	Inventory Specification	Process Specification	Distribution Specification	Responsibility Specification	Timing Specification	Motivation Specification	Technology Physics
Technician	Inventory Configuration	Process Configuration	Distribution Configuration	Responsibility Configuration	Timing Configuration	Motivation Configuration	Tool Components
Enterprise	Inventory Instantiations	Process Instantiations	Distribution Instantiations	Responsibility Instantiations	Timing Instantiations	Motivation Instantiations	Operational Instances
	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	

Figure 22 Simplified Zachman Framework



Columns: Communication interrogatives

- What (the inventory column): Entities used to build the architecture
- How (the process column: Activities performed
- Where (the distribution column): Business location and technology location
- Who (the responsibility column): Roles and organisations

- When (the timing column): Intervals, events, cycles and schedules
- Why (the motivation column): Goals, strategies and means

Rows: Reification transformations – the steps necessary to translate an abstract idea into a concrete instance from different perspectives (planner, owner, designer, builder, implementer and user). Each perspective has a different relation to the What column:

- The executive perspective – business context
- The business management perspective – business concepts
- The architect perspective – business logic
- The engineer perspective – business physics
- The technician perspective – component assemblies
- The user perspective – operations classes

1.3.3 Enterprise Data Architecture

Enterprise Data Architecture descriptions include:

- **Enterprise Data Model (EDM**): A holistic, enterprise-level, implementation-independent conceptual or logical data model providing a common consistent view of data across the enterprise.
- **Data flow design:** Defines requirements and master blueprint for storage and processing across databases, applications, platforms and networks (the components).

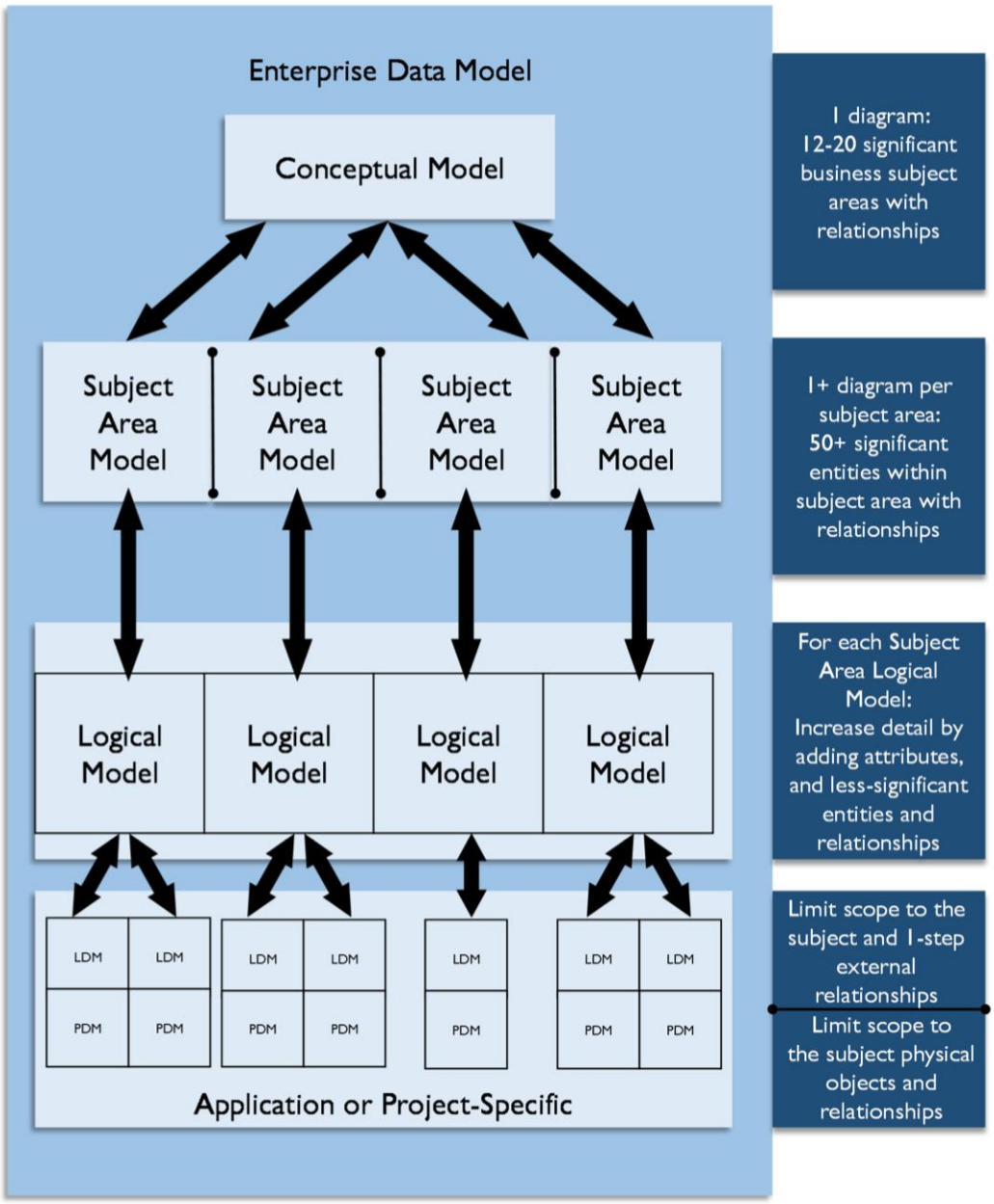
Both need to be reflected in current state (architecture perspective) and transition state (project perspective).

1.3.3.1 Enterprise Data Model

Can be a stand-alone artefact or may be composed of data models from different perspectives or levels of detail. An EDM includes both universal (Enterprise-wide Conceptual and Logical Models) and application or project specific data models, along with definitions, specifications, mappings and business rules.

Industry standard is a start but time and effort must also be invested to build and maintain the EDM.

- Conceptual overview over the enterprise's subject areas
- Views of entities and relationships for each subject area
- Detailed, partially attributed logical views for the same subject area
- Logical and physical models specific to an application or project



Different types of models are related vertically and horizontally.

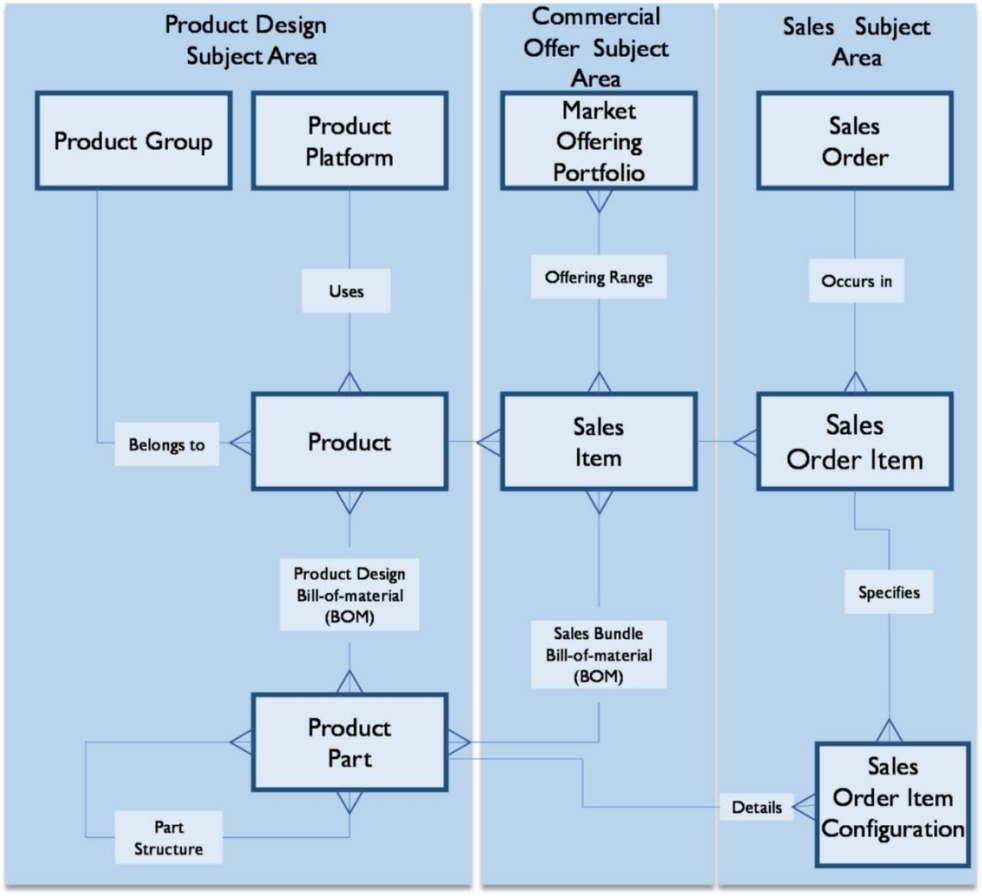
The Enterprise Data Model is built up by the combination of Subject Area Models:

- **Top Down approach**: Form Subject Areas then populate them with models
- **Bottom up Approach:** Subject Area structure is based on existing data models

The Subject Area Discriminator (the way subject areas are formed) must be consistent throughout the enterprise data model:

- **Funding:** Systems portfolios
- **Organisational**: Data governance structure and data ownership
- **Business value chains**: Top-level processes
- **Using business capabilities**

Subject Area Model Example

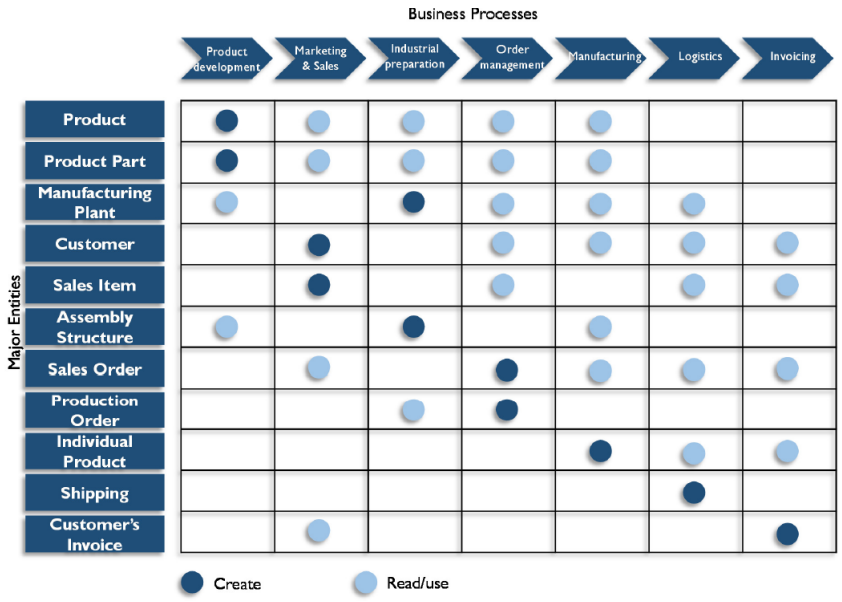


1.3.3.2 Data Flow Design

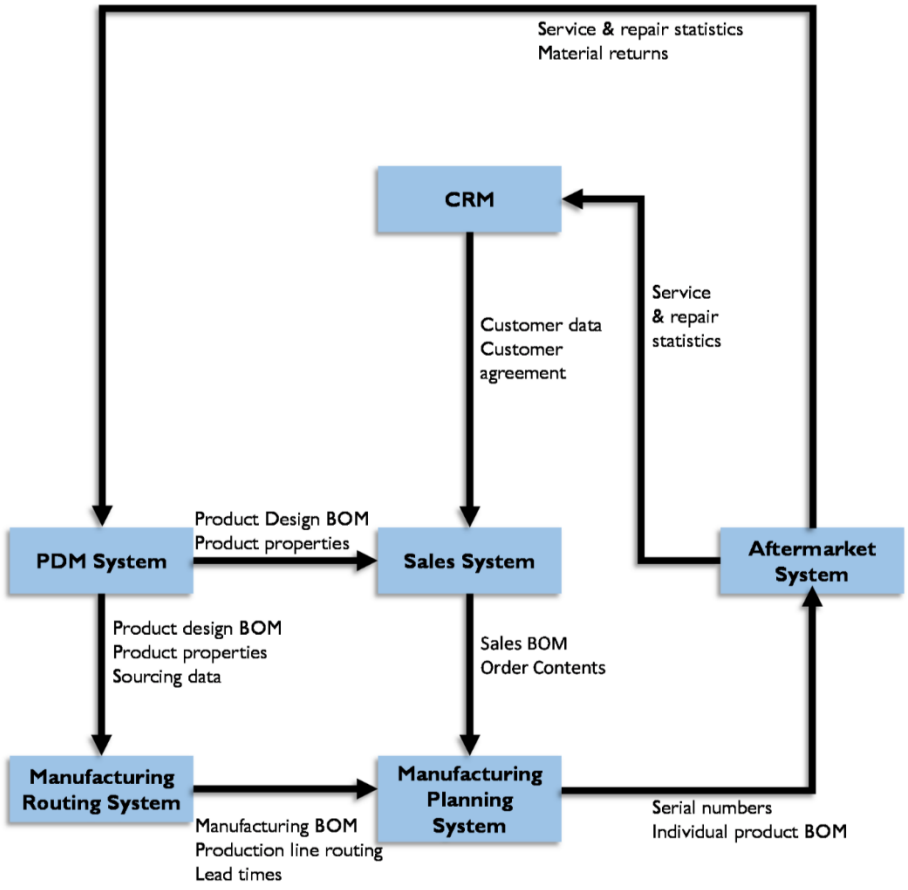
Data flows are data lineage documentation that depicts how data moves end-to-end through business processes and systems, where it originated, is stored and used and how it transforms.

Data flows can be documented at different levels of detail: Subject Area, Business entity or attribute level.

Data Flow Matrix example:



High level data flow diagram:



2 Activities

Two approaches:

- **Quality-oriented**: Focus on execution within business and IT structures. Unless architecture is managed it will deteriorate. Architectural improvements are incremental.

- **Innovation-oriented:** Focus on transforming business and IT to address new expectations and opportunities. Requires interaction with business development representatives and designers

2.1 Establish Data Architecture Practice

A Data Architecture practice includes the following work streams:

- **Strategy:** Select frameworks, state approaches, develop roadmap
- **Acceptance and culture:** Inform and motivate changes in behaviour
- **Organisation:** Assign Data Architecture accountabilities and responsibilities
- **Working methods:** Define best practices and perform Data Architecture work within development projects, in coordination with Enterprise Architecture
- **Results:** Produce Data Architecture artefacts within an overall roadmap

Enterprise Architecture influences scope boundaries of projects/system releases:

- Defining project data requirements
- Reviewing project data designs
- Determining data lineage impact
- Data replication control
- Enforcing Data Architecture standards
- Guide data technology and renewal decisions

2.1.1 Evaluate Existing Data Architecture Specifications

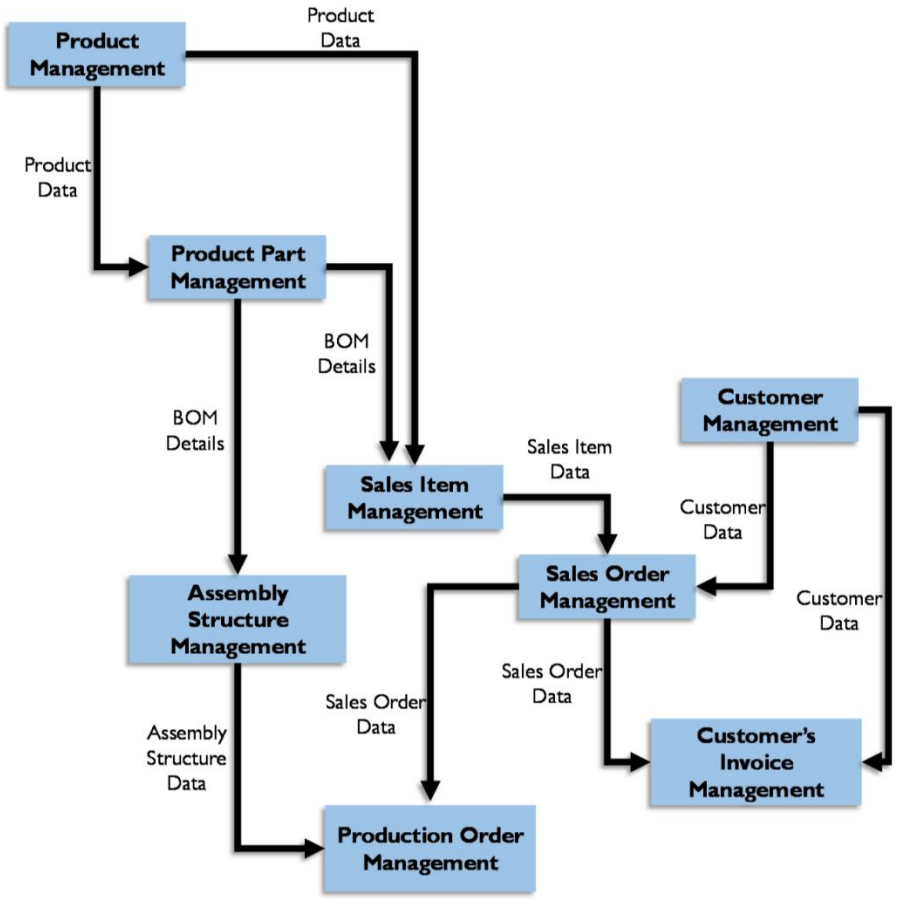
Evaluate existing documentation for accuracy, completeness and level of detail. Update if necessary.

2.1.2 Develop a Roadmap

Describes the architecture's 3 to 5-year development path, with business requirements, consideration of actual conditions and technical assessments. Must be integrated into the enterprise architecture roadmap. Include milestones, resources needed, cost estimations, and divided into work streams.

The Enterprise Data Architecture can be formed by resolving input and output data flows in the chain of dependencies between business capabilities.

Start with the most independent business capabilities and end with those most dependent on other activities. The following diagram has the lowest dependency on the top.



2.1.3 Manage Enterprise Requirements within Projects

Enterprise Data Architecture project-related activities include:

- **Define scope:** Ensure the scope and interface are aligned with Enterprise Data Model. Identify components that can be reused, and down-stream dependencies.
- **Understand business requirements**
- **Design:** Form detailed target specifications. Look for shareable constructs in the enterprise logical data model. Review and use technology standards.
- **Implement:**
  - **When buying:** Reverse engineer purchased applications and map against data structure
  - **When reusing data:** Map application data models against common data structures and new and existing processes to understand CRUD operations. Enforce use of authoritative data.
  - **When building:** Data storage according to data structure.

The role of Enterprise Data Architects and the process of building architectural activities into projects depends on the development methodologies:

- **Waterfall methods:** Construct systems in sequential phases as part of an overall design.
- **Incremental methods:** Learn and construct in gradual steps. creates prototypes based on vague overall requirements.
- **Agile, iterative methods:** Learn, construct and test in discrete delivery packages (Sprints).

2.2 Integrate with Enterprise Architecture

Integrate Enterprise Data Architecture matters with project portfolio management as funded projects drive architecture priorities and Data Architecture can influence the scope of projects.

3 Tools

- **Data Modelling Tools:** Include lineage and relation tracking to manage linkages between models
- **Asset Management Software:** Used to inventory systems, describe their content and track the relationships between them
- **Graphical Design Applications:** To create architectural design diagrams and other architectural artefacts

4 Techniques

4.1 Lifecycle Projections

Architecture designs can be:

- Aspirational and future-looking
- Implemented and active
- Plans for retirement

What architectural plans represent should be clearly documented:

- **Current:** Products supported and used
- **Deployment period:** Products deployed for use in 1-2 years
- **Strategic period:** Products available in the next 2+ years
- **Retirement:** Retired products, or retirement within 1 year
- **Preferred:** Products preferred for use by most applications
- **Containment:** Limited for use by certain applications
- **Emerging:** Researched and piloted for possible future deployment
- **Reviewed:** Evaluated products and their evaluation results

4.2 Diagramming clarity

Models and diagrams must conform to an established set of visual conventions:

- **Clear and consistent legend:** Identify all objects and lines and placed in the same spot in all diagrams.
- **Match between all diagram objects and the legend:** Not all legend objects need appear on diagram
- **Clear and consistent line direction:** Usually left to right. Backward lines must be clear
- **Consistent object attributes:** Differences in size, line thickness and colour should signify something
- **Linear symmetry:** Line up at least half of the objects to improve readability

5 Implementation Guidelines

As Data Architecture is about artefacts, activities and behaviour, Enterprise Data Architecture is about:

- Organising Enterprise Data Architecture teams and forums

- Producing initial versions of Data Architecture artefacts such as enterprise data model, enterprise wide data flow and road maps.
- Forming and establishing a data architectural way of working in development projects.
- Creating organisation wide awareness of the value of Data Architecture efforts.

A Data Architecture implementation should include at least 2 of the above.

Data models and other Data Architecture artefacts are captured within development projects and are then standardised and maintained by architects. There will be more architectural work in early projects which may need special architectural funding.

Enterprise Data Architecture evolves incrementally in a solution-oriented culture using agile development.

Enterprise Data Architecture starts with Master Data areas in need of improvement in planned development projects, and expands to include business and other data.

5.1 Readiness / Risk Assessment

More risks than other projects, especially during an organisation's first attempt:

- **Lack of management support**
- **No proven record of accomplishment**
- **Apprehensive sponsor**
- **Counter-productive executive decisions**
- **Culture shock**
- **Inexperienced project leader**
- **Dominance of a one-dimensional view**

5.2 Organisation and Cultural change

The ability of an organisation to adopt Data Architecture practices depends on several factors:

- Cultural receptivity to architectural approach
- Organisation recognises data as a business asset, not just an IT concern
- Ability to let go of a local perspective and adopt an enterprise perspective on data
- Ability to integrate architectural deliverables into project methodology
- Level of acceptance of formal data governance
- Ability to look holistically at the enterprise

6 Data Architecture Governance

Enterprise Data Architecture and the Data Governance organisation must be well aligned. A data steward and a data architect should be assigned to each subject area, even to each entity within, as Data Architecture activities support the alignment and control of data. Business event subject areas should be aligned with business processes governance as each event entity usually corresponds to a business process.

Data Architecture governance activities include:

- **Overseeing projects:** Projects comply with required Data Architecture activities, use architectural assets and are implemented according to data architectural standards.
- **Managing architectural designs, lifecycle and tools:** Designs must be defined, evaluated and maintained

- **Defining standards**
- **Creating data-related artefacts**

6.1 Metrics

Data Architecture metrics may be monitored annually for business customer satisfaction:

- **Architecture standard compliance rate:** How far projects comply with established data architectures.
- **Implementation trends:** The degree to which enterprise architecture has improved the organisation's ability to implement projects along at least two lines:
    - **Use/reuse/replace/retire measurements:** Proportion of new architectural artefacts to reused, replaced or retired artefacts.
    - **Project execution efficiency measurements:** Measure lead times for projects and their resource costs for delivery improvements with reusable artefacts and guiding artefacts.
- **Business value measurements:** Track progress towards expected business benefits
    - **Business agility improvements:** Account for the benefits of lifecycle improvements or the cost of delay
    - **Business quality:** Measure whether business cases are fulfilled and projects deliver changes leading to business improvements
    - **Business operation quality:** Measure of improved efficiency and accuracy
    - **Business environment improvements**