TM Forum Introductory Guide

Digital Twin for Decision Intelligence (DT4DI) Reference Architecture

IG1310A

Maturity Level: **Alpha**	Team Approved Date: 16-Aug-2023
Release Status: Pre-production	Approval Status: Team Approved
Version 2.1.0	IPR Mode: RAND

Notice

Direct inquiries to the TM Forum office:

181 New Road, Suite 304
Parsippany, NJ 07054, USA
Tel No. +1 862 227 1648
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Figures

Executive Summary

'If you think good architecture is expensive, try bad architecture.' (Brian Foot and Joseph Yoder, [1])

Even worse, try to develop large, complex systems without architecture.

Your systems would probably resemble *Cidade de Deus* (City of God) or any other crowded *favela* in Rio de Janeiro (and any other shanty slums anywhere):

- A collection of makeshift items thrown together with no recognizable organization.
- Lack of essential capabilities.
- Low-quality materials.
- Poor infrastructure.
- Problems everywhere.
- Unsanitary conditions.
- Security and compliance issues.
- Narrow alleyways and unpaved pathways (we may say inadequate integration).
- Complex maintenance and unplanned expansion.
- Impoverishing the people who build them (in our analogy, engineers and developers) and live there (in our analogy, customers, end-users and process-owners).

Consequently, those informal housing structures can't compete in any sense with the beautiful and solid buildings in Ipanema and Leblon.

In software engineering, for conceptually the same reasons, to successfully build, deploy and operate complex and strategic systems (and DT4DI systems are part of this club), it is necessary to refer to and adopt reference architectural blueprints, design patterns and implementation guidelines during the design, development, deployment, and maintenance of those systems.

Introduction

When discussing architecture in software engineering, we enter an intricate realm that includes different architectural disciplines.

In the context of this paper, simplifying, we distinguish three different architectures:

- **Application Architecture** is the discipline that designs individual software applications, such as ticketing systems, charging & billing components, anti-fraud, provisioning, order management, churn prediction, up/cross-selling, campaign management systems, etc.
- **Domain Architecture** designs the structure of domain-specific systems involving multiple integrated and interrelated applications, such as BSS, OSS, ERP, BI & Analytics, etc.
- **Enterprise Architecture (EA)** focuses on designing and managing the overall structure and operations of an organization's systems, information, technology, and infrastructure, ensuring alignment with the business strategy and goals of the organization. Enterprise architecture includes the development of policies, standards, and guidelines for IT systems, as well as harmonizing and integrating all the domain-specific systems and applications within the organization.

The reality is more complex as there is also solution architecture, designing particular solutions combining different applications and interfaces, AI model architecture (CNN, RNN, autoencoders, generative models, LLM, etc.), infrastructure architecture, network architecture, etc.

However, for our purpose, let's simplify by considering the three architectural disciplines above.

These different architectural disciplines are all necessary, related, and synergetic, forming hierarchical and bidirectional relationships:

- **Enterprise Architecture (EA)** is the overarching and all-embracing framework, including all the domain-specific architectures.
- **Domain Architectures** address the needs of specific business areas, services, and processes, integrating multiple applications, in line and compliance with the Enterprise Architecture.
- **Application Architectures** define the solutions of the individual applications included in the Domain Architectures, in line with the Domain Architecture and the EA guidelines.

Following the above analogy with the construction industry, we may say that:

- Enterprise Architecture corresponds to city planning (also called urban planning), focusing on the design and planning of an entire city or metropolitan area.
- Domain Architecture corresponds to the design of specific residential, business, educational, institutional, and industrial areas or districts, reflecting the style of living and supporting particular valuable activities.
- Application Architecture corresponds to the design of individual buildings and houses that manage the underlying relation between structures and humans and address their expectations and functional and emotional experiences.

The DT4DI Reference Architecture is a **Domain Architecture** covering the Decision Intelligence (DI) application domain that, as described in the DT4DI Whitepaper [0], includes and absorbs the Business Intelligence & Analytics (BI&A) domain and addresses analytics- and decision-driven use cases capturing business opportunities and solving complex business problems.

As per good design practices, the DT4DI Reference Architecture shall be aligned with the organization's Enterprise Architecture and data-enabled design strategy.

Indeed, critical preconditions for successfully implementing DT4DI architectures include the availability of effective data-enabled design strategies, data governance, customer experience strategies, and business process management practices incorporating data-driven decision-making for customers', services', and products' lifecycles.

Digital Twin (DT) and AI/ML are key technological enablers of DT4DI domain architectures.

As shown in Figure 1 below, we may define DT4DI as the union of DI, BI & Analytics, DT, and AI.

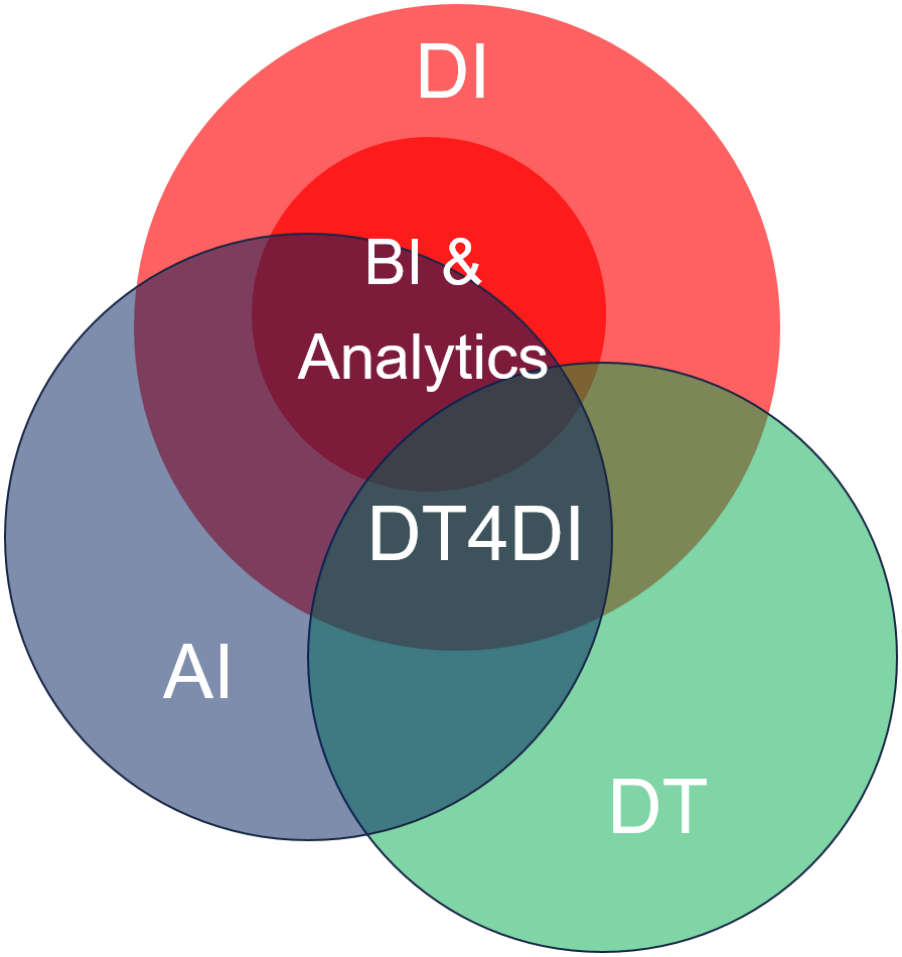

Figure 1. The DT4DI Domain, including BI & Analytics and using DT and AI as key enablers

1. Scope and Objectives

The purpose of this document is to define a reference architecture for the DT4DI system domain, including:

- DT4DI Level 0 Architecture (L0), the blueprint and visual description of the high-level (level 0) architecture and structure of the DT4DI system domain.
- DT4DI Level 1 Architecture (L1), the detailed structure of Level 0 architectural layers.

The DT4DI framework also includes engineering guidelines, a set of best practices, and recommendations for implementing DI processes (DIOps), DT models (DTOps), and AI/ML components (AIOps). These complementing and necessary engineering guidelines are shown in the DT4DI reference architecture of Figure 3 but are the object of separate and dedicated documents.

According to Uncle Bob (he calls himself this way, [1] [2]), the primary purpose of architecture is not (only) to provide the expected behavior of the system (this goal is implicit, and sometimes even *favela* architecture may work) but mainly to facilitate the development, deployment, daily operations, and maintenance of the system, including its future decommissioning.

A good domain architecture shall be:

- Easy to develop and expand with new or changed capabilities, addressing the constant flow of new functional and non-functional requirements efficiently and enabling the innovation of products and services.
- Easy to deploy, minimizing time to market, efforts, and risks, and streamlining the process of distributing software and updates between different environments via repeatable configuration, scheduling, and automation practices.
- Easy to operate daily in Production (and in any relevant environment), facilitating the integration with all the service management processes such as monitoring & event management, incident and problem management, configuration management, performance management, capacity management, dataops, security operations, etc.
- Easy to maintain (maintainability), supporting the evolution of the architecture, adding new features, incorporating new data, updating existing functionality, redesigning part of the system, fixing issues, etc.
- Easy to decommission, easing the removal of the system and the unnecessary related components effectively and efficiently.

In other ways, Steve Jobs said, "Design is not just what it looks and feels like. Design is how it works." In our context, we can replace "design" with "architecture" or consider them synonyms.

On the same line, the American architect Louis Sullivan, one of the fathers of modernism, stated that "Form ever follows function", meaning that the design of a building should be based on its intended function or purpose (use case) and that the form of the building should arise naturally from that function.

For our project, all these views are valid, necessary and complementary.

- The DT4DI Architecture is how the system shall work and behave.
- The form, including how it looks and feels, shall be harmonious and coherent with the DT4DI system's functions and use cases and design the perceptive qualities of interactive systems.
- The DT4DI Architecture shall support the end-to-end (E2E) lifecycle of systems and components effectively, flexibly, and efficiently.

Reshuffling these concepts, the DT4DI Architecture shall support the DT4DI use cases across the E2E software lifecycle, ensuring that the system is reliable, scalable, simultaneously robust and flexible, maintainable, and meet the needs and expectations of its end-users and stakeholders.

2. Architectural Patterns

"Divide et Impera" (Philip II of Macedon, Julius Caesar, Machiavelli, Napoleon...)

We can translate this popular refrain here as "divide and rule", which is the common leitmotif of practically all architectural and design patterns.

The architectural patterns aim to manage systems' complexity and magnitude by breaking down monoliths into smaller and more manageable components, making them easier to understand, design, develop, maintain, operate, control, reuse, share, decommission, etc.

Below, we list some popular architectural patterns, some classic, others more recent, without the pretension of being exhaustive.

This list serves as a menu to refer to when designing reference architectures.

And as in the kitchen or restaurant, we can combine different architectural patterns as we combine recipes to create our favorite dining experience.

Each layer/building block of the reference architecture may refer to specific patterns more suitable and compelling for that layer.

In other words, more than just a single architectural pattern is needed to define a domain reference architecture, including multiple applications that support an entire business domain.

We need to mix up multiple patterns balancing different and often opposite and conflicting requirements and expectations to design an architecture that is a champion of oxymoron:

- flexible robustness
- solid agility
- reliable speed
- dynamic stability
- elastic scalability
- loosely coupled
- safe velocity
- secure openness
- high-standard prototyping
- efficient maintainability
- versatile compliance
- etc.

The table below briefly describes each architectural/design pattern (listed in alphabetic order).

Please refer to the DT4DI & AIOps Ontology (IG1310 [3]) for a longer description and the sources/references used in this work.

Id	Design/Architecture Pattern	Short Description
1	**Component-based Architecture**	Components are developed independently of the rest of the system, reused across multiple applications, and designed to be easily integrated into other systems.
2	**Data Fabric**	Architectural pattern that facilitates organizations to seamlessly integrate and manage a unified abstraction layer over diverse data sources.
3	**Data Grid**	Distributed computing architecture processing and managing large volumes of data across multiple machines or nodes.
4	**Data Mesh**	Decentralized approach to data management where data are considered engineered products with their lifecycle and ownership within their data domains.
5	**Data Virtualization**	Accessing and integrating data from multiple on-premises and cloud sources into a single virtual view without physically copying or moving the data.
6	**Edge Computing Architecture**	Focused on processing data and performing analytics tasks closer to the source of the physical assets and systems. This architecture allows real-time data processing, reduces latency, and minimizes the dependence on centralized cloud resources, resulting in faster and more efficient decision-making processes.
7	**Event-Driven Architecture (EDA)**	Enabling loosely coupled, asynchronous communication between components by using events to trigger actions or processes.
8	**Feature-based Architecture**	Breaking down the application into distinct features or functional units, implemented as a separate module that provides a specific set of functions.
9	**Federated Architecture**	A decentralized approach that involves the collaboration and interaction of multiple distributed entities or components into a unified system and allows each entity to maintain a degree of autonomy and control over its operations.
10	**Layered Architecture**	Organizing systems into distinct layers, each having a specific responsibility and interacting with adjacent layers.
11	**Microservice Architecture**	Designing applications as a collection of small, independent services that communicate with each other through APIs.
12	**Model-View-Controller (MVC)**	Separating the application logic into three interconnected components, the Model (business logic), the View (user interface), and the Controller (broker).

Id	Design/Architecture Pattern	Short Description
13	**Object-Oriented Architecture**	Designing software systems and components using the principles of object-oriented programming (OOP), viewing the system as a collection of objects.
14	**Onion Architecture**	Based on the domain-driven design (DDD) approach, it consists of distinct concentric layers arranged like the layers of an onion, with the domain model at the center and the other layers surrounding it.
15	**Service Oriented Architecture (SOA)**	Based on self-contained services, which are loosely-coupled software components that can be accessed and reused across different applications and systems.

In designing the DT4DI Reference Architecture, we use or get inspired by most of the above design patterns.

3. DT4DI Use Cases

Following Uncle Bob, Steve Jobs, Louis Sullivan, and many other outstanding architects, architecture and use cases are closely related concepts in software engineering and, in our context, in the design of a system domain architecture.

Architecture defines how the system is structured and how its components work together cohesively to meet the domain business goals, supporting the specific tasks and activities that end-users or other applications will perform with the system.

The use cases define the system's functional requirements, what it will do, and how it will be used, and help ensure that the architecture meets the needs of its end-users and stakeholders.

3.1. DT4DI for Business Operations

The DT4DI Reference Architecture is a domain architecture that naturally supports applications currently covered by BI & Analytics systems, augmenting and extending their sharpness and effectiveness.

As illustrated in the DT4DI Whitepaper [0], DT4DI is an extension and evolution of the BI domain, providing business capabilities to orchestrate business analytics and operations.

In the ODA functional architecture, DT4DI is located in and overlays the Intelligence Management domain ([0]).

Therefore, the DT4DI typical use cases include (list not exhaustive):

- Marketing Operations: customer profiling and segmentation, customer behavior analysis, churn prediction, customer retention, loyalty programs, sentiment analysis, marketing performance, etc.
- Product Management: product design, pricing, product recommendations, performance management, etc.
- Sales Operations: cross-selling, up-selling, next best action/product, sales analytics and performance, sales process optimization, sales forecasting, etc.
- Customer Experience Management: customer experience analysis, customer journey optimization, loyalty improvement, and delivering the company's identity per the expected experience journey.
- Revenue Management, handling customer lifetime value, ARPU (Average Revenue Per User) development, revenue assurance, market trends, dynamic pricing, financial performance management, etc.
- Content Management, including defining content business objectives, target audience needs, content types, content recommendations, channels and formats, and the integration of DT4DI with Content Management Systems (CMS).
- Campaign Management, understanding customer preferences to suggest product features, rate plans, and offers yielding desired business outcomes such as improved customer reach, conversions, revenue realization, customer experience, etc.
- Hyper-Personalization of products & offers, predicting customers' behavior, understanding their evolving needs & engaging them with the right product-

channel-time combinations to improve experience and loyalty, thereby increasing NPS and reducing churn.

- Fraud Prevention, proactively deterring fraud in processes such as subscription, SIM swap, IRS, and other commercial operations, by predicting and pre-empting attacks, receiving actionable insights, and providing recommendations to improve business KPIs.
- Single point of view for multiple supply chain facets, understanding actual and future demand, enabling continuous S&OP (Sales & Operations) process with predictive insights, foreseeing inventory, creating dynamic replenishment logic, and pre-empting procurement & fulfillment process fallouts.
- Business analysis and strategy recommendations, monitoring, predicting, and optimizing critical CEM processes, marketing & sales, and financial KPIs to make faster evidence-based decisions.
- Event-driven marketing, triggered by predefined events, i.e., real-time location, user status or behaviors, customer preferences from content insight, etc.
- Complex events-driven recommendations in integrated and multi-domain scenarios.
- Strategy simulation, supporting a comprehensive marketing analysis, customer demand forecasting, and business planning for business development and revenue growth.
- And many others.

In DT4DI, these business use cases leverage Decision Intelligence (DI), Digital Twins (DT) and AI to enable better and faster decision-making in the corresponding business processes.

Business leaders and process stakeholders across the organization must be part of the journey as champions to fully unleash the potential benefits of Decision Intelligence using DT and AI.

A few of these key stakeholder groups are listed below:

- CEO, CFOs, and CSOs (Chief Strategy Officers) - enabling strategic intelligence and experimentation to make decisions on the organization's strategy and roadmaps, revenue, and cost opportunities, driven by insights on the market, competitors, and organizational performance data.
- CMO, Chief Customer Officer, Chief Experience Officer, Sales and Marketing leadership, and related stakeholders - using insights and exploration to track and improve customer lifetime value, customer journeys, and experience management, personalize products/services/offers, and reduce customer churn.
- Chief Product Officers, product owners, and other related stakeholders – leveraging insights and recommendations on product changes, suggestions for new products based on customer needs and behaviors, and better strategies to reach the products to customers.
- CIO, Chief Security Officer, Security Organization – fostering proactive insights and scenario-based experimentation to help identify and prevent fraud risks, data breaches, compliance issues, and security vulnerabilities.
- CIO and CTO - ensuring service quality, reliability, and performance.

The figure below shows a sparse and not exhaustive highlight of potential DT4DI Use Cases for business operations.

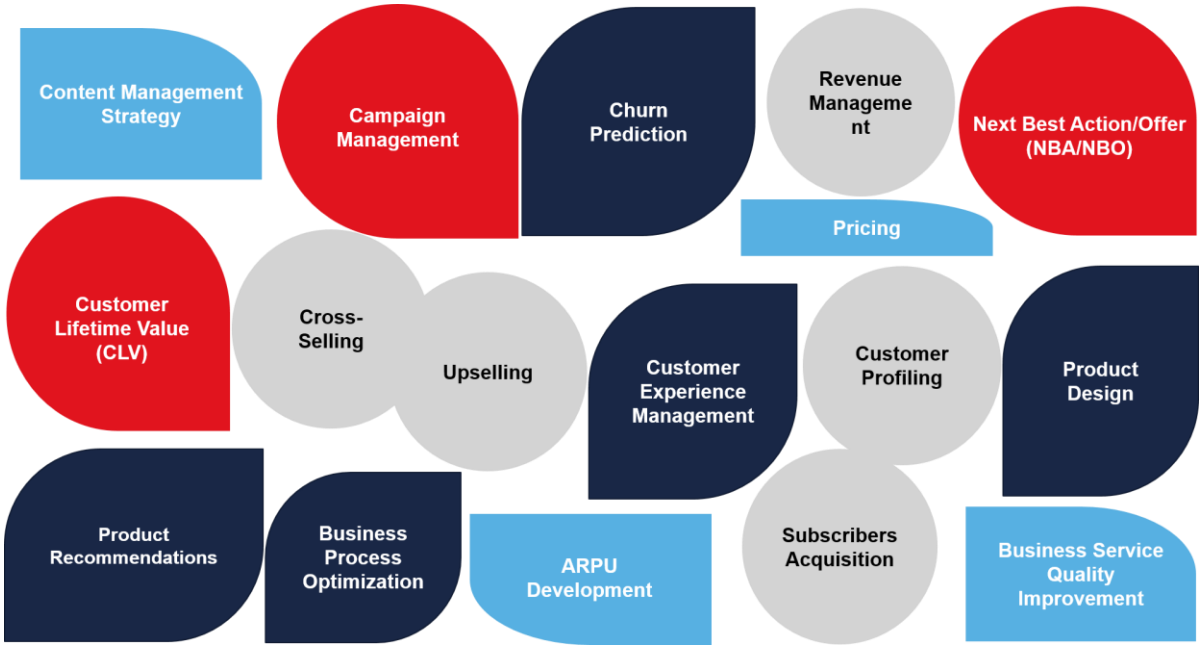

Figure 2. A sparse and not exhaustive selection of DT4DI Use Cases

The DT4DI Architecture shall support and implement the above use cases.

However, it's worth noting that the DT4DI Reference Architecture shall be designed to be flexible and adaptable, so it can be customized to meet particular business requirements of the domain.

3.2. DT4DI for Technical Operations (Network & IT)

Although we initially conceived DT4DI for business operations, a similar approach can be extended to any complex decision-making processes in Network & IT operations and other domains.

For this purpose, we shall consider the following guidelines for the use cases:

- Decisions in the use case are complex and require that humans, data, and machines are interconnected to design the best options and make the most appropriate decisions.
- The complexity of the challenge requires the implementation of data fabric, sophisticated modeling techniques (i.e., Digital Twin), and AI layers and capabilities.
- DT4DI doesn't replace or overlap existing OSS systems and capabilities.
- DT4DI doesn't explicitly address automation or autonomy (i.e., that's not the final goal).
- Multiple, distributed, diverse data sources are involved in the analysis and decision process.

- The use case architecture is compatible with the DT4DI Reference Architecture described in this document.

We are exploring extending the DT4DI solutions to technical operations in specific use cases.

3.3. DT4DI for Sustainability

Sustainability is the ability to maintain or support something over the long term, meeting the needs of the present without compromising the future.

Sustainability involves:

- Balancing economic, environmental, and social considerations.
- Combining individual actions and collective efforts.
- Adopting sophisticated technology to support the challenges.
- Changing culture and policies.

It consists of adopting sustainable practices in energy production and management, transportation, waste management, agriculture, urban planning, process reengineering, education, communication, etc.

The DT4DI approach can help organizations and governments determine and address sustainability goals, which are challenging and complex.

Here are examples of use cases that DT4DI can support:

- Energy Management, including energy optimization, demand forecasting, intelligent power feature management, resource utilization efficiency, and energy cost reduction using AI and IoT sensors.
- Environmental Sustainability, including carbon footprint monitoring, sustainability reporting, and sustainability analytics to improve sustainability performance.
- Energy savings in data centers (including cloud and edge data centers) by optimizing the resources and dynamically deploying/migrating the workloads. DT helps get a complete view of the assets and their status, and DI support dynamic and complex decision-making.

In future work, we will investigate the extension of the DT4DI solutions to sustainability challenges in specific use cases.

4. DT4DI High Level Architecture - Level 0

The level 0 (L0) architectural description overviews the general DT4DI solution, including critical components/layers, their functional scope, their capabilities, and how they interact.

It is the starting point for the detailed design work.

Figure 3 below shows the broader view of the DT4DI framework, which includes the following:

- DT4DI Level-0 Architecture (layered view).
- Vertical process methodologies enabling the implementation of DT4DI solutions:
- DIOps manage the lifecycle of Decision Intelligence processes.
- AIOps manage the lifecycle of AI components.
- DTOps manage the lifecycle of Digital Twins (DT) models.
- Platforms & Tools supporting each layer.

This document IG1310A covers the DT4DI Architecture.

TM Forum AIOps toolkit [4] covers the lifecycle of the AI components (AIOps).

We will cover DTOps and DIOps in future dedicated deliverables.

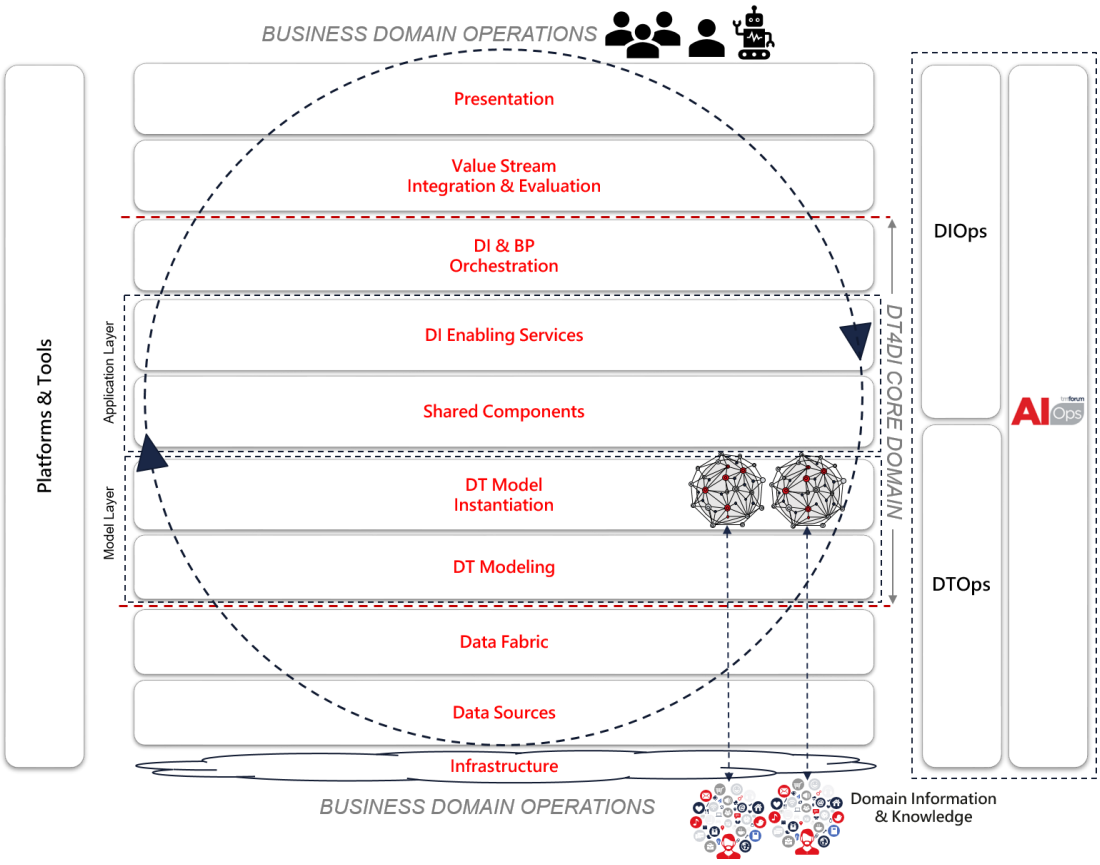

Figure 3. The DT4DI Framework including the Level-0 DT4DI Reference Architecture (layered view)

The essential layers of the DT4DI Reference Architecture are (from bottom to top):

- **Data Sources**: the various internal and external ways data can be obtained or collected.
- **Data Fabric**: the virtualized data layer that sits on top of multiple and disparate data sources, integrating and federating data, providing a unified view of the data and a holistic approach to data management within an organization. It also ensures the quality, consistency, security, and compliance of the data used in the DT4DI solution.
- **DT Modeling**: the layer builds virtual models of real-world objects, systems, processes, or services. It integrates data (provided by the Data Fabric) and the virtual models to represent the target objects comprehensively, using domain expertise, AI models, and simulation feedback.
- **DT Model Instantiation**: the layer holds the virtual models and provides an abstraction between the application and data storage layers. It manages the storage, mapping, updating, and retrieval of data from persistent data stores on behalf of the application layer. In addition to traditional data models, it includes faithful and purposive replicas of business domain entities such as Process Twin, Product Twin, Resource Twin, Customer Twin, etc., in a virtual environment.
- **Shared Components**: a logical collection of reusable software components that provide granular and modular functions to build applications and services.
- **DI Enabling Services**: the layer exposes through APIs independent services implementing business process and decision-making logic required to create the E2E DT4DI use cases. It uses capabilities, data models, and digital twins to deliver composable services such as data insights, patterns, predictions, simulations, product recommendations, next-best actions, knowledge graphs, lifecycle stage tracing, E2E single view, prioritization, fallout predictions, fraud scoring, etc.
- **DI & BP Orchestration:** the layer includes the workflows that design, integrate, automate, and optimize the business process (BP) steps and decision intelligence (DI) model, flows, and rules of the E2E DT4DI use cases. It includes the continuous improvement of the DT4DI solution by integrating feedback and insights from end-users, stakeholders, and other sources.
- **Value Stream Integration & Evaluation:** The layer includes the design and capabilities to integrate the DT4DI solution into the organization's current business value stream and operations and evaluate the effectiveness of the business and decision intelligence logic of the solutions to address the problem/opportunity statements. It also enables the continuous improvement of the DT4DI solution by interacting with the Presentation layer and integrating feedback and insights from end-users, stakeholders, and other sources (e.g., feedback mechanisms, A/B testing, experimentation, etc.).
- **Presentation:** the layer presents business data and functionality to the customers,end-users, stakeholders, and, generally, use case consumers. It provides intuitive, user-friendly multichannel interfaces that enable end-users and consumers to interact with the underlying applications and data and visualize outcomes, results, reports, and parameters (deviations, thresholds, simulation results, predictions, goals, constraints, etc.)

Figure 4 below represents the "Onion" view (a term coined by Jeffrey Palermo [55]) of the DT4DI Reference Architecture following the domain-driven design (DDD) approach ([44]), which drives the design process from the core business and decision logic outward to the technical details (from center to outer circles), not in the opposite direction.

DT4DI Core Domain Architecture is the grey area in Figure 4. The white external layers (Data Fabric, Data Sources, Infrastructure, Value Stream Integration & Evaluation, and Presentation) are not specific to DT4DI and are shared with other application domains.

The Onion architecture explicitly suggests a business/domain/use case-driven approach.

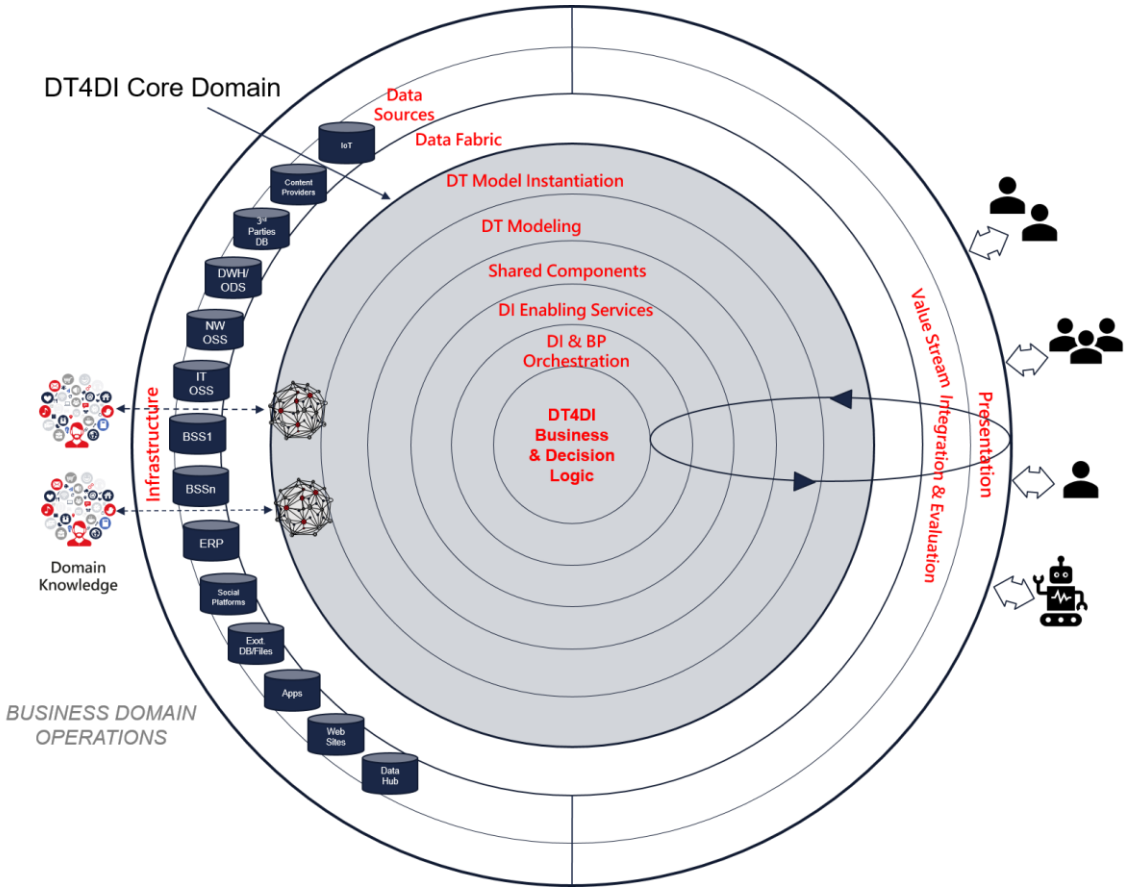

Figure 4. The DT4DI Onion Architecture with the DT4DI Business & Decision Logic at the center.

The concentric layers are the same as the layered architecture of Figure 3.

However, the domain or use case business and decision logic is at the center of the onion, and the other layers envelop it.

One of the fundamental principles of Onion Architecture is that the inner layers of the DT4DI core domain shall not depend on the outer layers, but the outer layers depend on the inner layers. In other words, all coupling is toward the center.

The center's core business and decision logic are independent and decoupled from the application, data models, integration, infrastructure, and presentation concerns.

The core business and decision logic represent the essential functions and rules of a DT4DI use case that directly relate to its primary purpose and unique value proposition.

The established primary purpose shall guide design decisions.

The inner and outer layers are coupled with the feedback and continuous improvement loop of the value interpretation managed through the Presentation layer.

In the DT4DI onion view, the logical sequence of the concentric layers of the DT4DI core domain is slightly different from the layered view (figure 3):

DT4DI Business & Decision Logic → DI & BP Orchestration → DI Enabling Services → Shared Components → DT Modeling → DT Model Instantiation

In the onion view, DT Model Instantiation is an implementation detail that depends on DT Modeling. DT Model Instantiation is the most external concentric layer of DT4DI Core Domain Architecture (the grey area in Figure 4).

We could also call the onion view "Solar System Architecture," where the system's center is the core business and decision logic (not the network connectivity, the data centers, or the databases).

The following chapter describes Level 1 of the DT4DI Architecture, going into more detail for each architectural layer.

5. DT4DI Detailed Architecture - Level 1

5.1. Data Sources

DT4DI solutions support use cases that require combining, ingesting, and elaborating multiple data sources.

Consequently, the DT4DI Reference Architecture sits on top of multiple, varied, and heterogeneous data sources.

The data sources are components exterior to the DT4DI Architecture and not part of it.

However, the architecture design shall:

- Identify all the relevant data sources used by core DT4DI systems and components.
- Manage the link to the data sources.
- Ingest new data elements.
- Add new data sources when required.

Data sources can be internal or external to the organization, such as data from third-party vendors or public datasets.

The internal data sources in a typical DT4DI architecture for telcos span all the categories of business and technical operations systems:

- BSS
- NW (Network) OSS
- IT OSS
- ERP
- Data warehouses (DWH)
- Operational Data Stores (ODS)
- Data lakes
- Web servers
- Corporate systems
- Corporate financial and assets data
- Network devices
- Company IoT sensors
- Company "experience object" (EO) sensors/sources
- Knowledge databases
- Any internally available files or databases.

The external data sources may include:

- Social media
- Web/internet platforms
- Third-party suppliers' systems and databases
- Business partners' databases and datasets
- IoT platforms and databases
- Publicly available databases
- Publicly available files
- External sales and marketing data
- External customer data
- External financial data
- Any externally available files or databases.

The data sources layer may include structured, semi-structured, and unstructured data.

The data sources may reside on-premises, in the organization's private cloud, in partners' and suppliers' premises or private clouds, in the public cloud. In a word, everywhere.

The crucial requirement is to be internally certified and approved as safe and reliable. This includes ensuring that data is collected, stored, and processed in compliance with relevant regulations, such as GDPR or HIPAA, and that appropriate measures are in place to protect sensitive data from unauthorized access, breaches, or cyberattacks.

5.2. Data Fabric

Data Fabric provides a unified and integrated view of data across multiple sources, formats, interfaces, and locations, including on-premises data centers, cloud environments, and edge devices.

It creates a cohesive, seamless, safe, and consistent view of data enabling the downstream layers (DT Modeling and DT Model Instantiation) to access and analyze data to design and implement data models effectively and efficiently, ignoring details about the data sources, locations, provenance, formatting, duplication, security concerns, etc.

Indeed, Data Fabric is a data service layer providing data products, objects, and pipelines to the downstream layers and eventually to the application components processing those data sets.

Data Fabric uses distributed and decentralized data architecture approaches adopting data virtualization, data integration, data ingestion, data orchestration, metadata management, data cataloging, ontolology, knowledge graph technologies and techniques.

Data Fabric also provides specific governance and security capabilities for managing and governing data policies, metadata, and access controls across different data sources and platforms.

A Data Fabric layer is a permanent block of the DT4DI architecture and can be reused and shared by multiple application domains.

Figure 5 below shows the building blocks of the Level 1 Data Fabric layer.

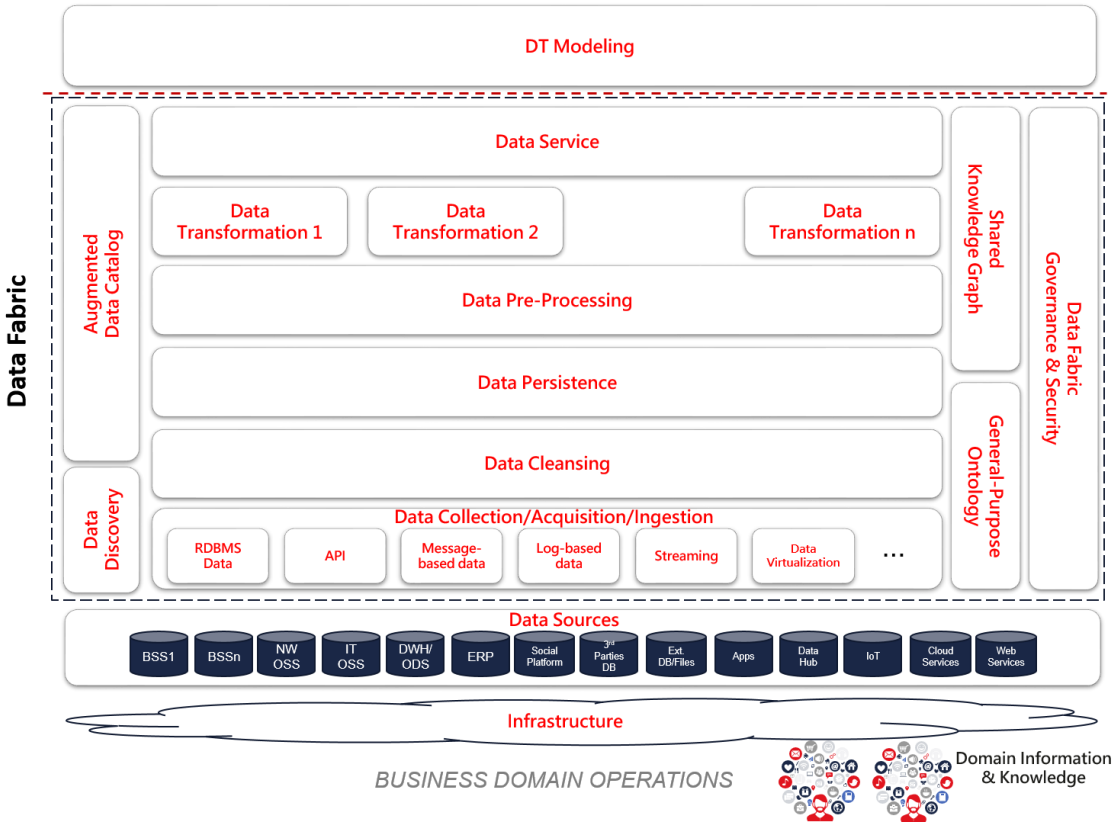

Figure 5. The Data Fabric Layer with examples of Data Sources

The essential building blocks of the Data Fabric layer are:

- **Data Acquisition or Collection or Ingestion** (which we consider synonyms) is responsible for collecting data from various data sources and importing them, physically or virtually, into the DT4DI system.
- **Data Discovery** identifies, explores, and understands new internal and external data assets. It involves discovering relevant data sets, understanding their structure, content, and relationships, and cataloging them in the Augmented Data Catalog.
- **Data Cleansing** is responsible for detecting, correcting, or removing corrupt, inconsistent, duplicated, or inaccurate records from the data collection process.
- **Data Persistence** retains the data after the ingestion for further processing.
- **Data Preprocessing** centralizes and shares preprocessing activities common to target services and components requiring the same data (e.g., data formatting, data enrichment, data augmentation, etc.)
- **Data Transformation** ensures that the target DT models, AI/ML components, and any components get the most accurate, complete, reliable, and suitable input datasets according to their specifications, design, and configuration baseline.

- **Augmented Data Catalog** (i.e., AI-powered) provides a comprehensive, centralized, and automated metadata inventory describing the organization's data assets.
  It enhances the traditional data catalog with additional AI-driven capabilities like automated data discovery, data profiling, data lineage, data quality assessment, data classification, natural language processing, semantic search, access controls, etc.
- **General-Purpose Ontology** aims to represent knowledge across multiple domains broadly. It captures the fundamental concepts and relationships that apply to a wide range of domains. It provides a common foundation for representing knowledge in the organization.
- **Shared Knowledge Graph** stores and organizes fundamental and foundational data in a way that represents the relationships between entities. It uses a schema to define entities, attributes, and relationships between them and employs semantic technologies to add meaning to the data. A semantic knowledge graph consists of the following:
  - Nodes: represent entities or concepts, such as people, places, and events.
  - Edges: represent relationships between nodes, such as "is-a," "part-of," and "related-to."
  - Attributes: describe properties of nodes, such as name, description, and date.
  - Ontology: defines the vocabulary and rules for representing the domain-specific knowledge in the graph.

In the Data Fabric, the Shared Knowledge Graph describes entities foundational for the in-scope domains (i.e., marketing, sales, customer experience management, revenue management, financial management, etc.) and common and shared across the organization. These foundational definitions are part of the General-Purpose Ontology of the organizations that include key concepts and classes and their relationships (e.g., service, product, device, consumer customer, business customer, corporate customer, prospect, company's signature experiences, etc.). It is a general-purpose knowledge graph.

- **Data Fabric Governance & Security** provides the capabilities for protecting an organization's data assets, managing and governing data policies, metadata, and access controls across different data sources and platforms, assessing and mitigating risks associated with the data.
- **Data Service** interfaces the Data Fabric users and consumers. It manages the data access, retrieval, and transformation functions the data consumers require. The Data Service layer acts as an orchestration layer to coordinate multiple data services and ensure data sets are delivered consistently and securely.

5.3. DT Modeling

The DT Modeling layer creates structured representations of data and their relationships to develop logical models enabling the instantiation of data repositories and physical data models to support business and decision-making processes and use cases.

The DT Modeling layer defines the semantics of the data from the application and business perspective, enabling the knowledge and exploration of the underlying business information and hidden patterns.

Modeling involves comprehensively understanding the business domain and data entities fundamental to specific DT4DI business and decision-making processes and use cases.

The DT Modeling contains the capabilities to model different data models that co-habit in DT4DI architectures, in particular:

- Digital Twins (DT).
- Traditional data models (e.g., relational, hierarchical, etc.).

DT Modeling creates a digital replica of real-world systems, processes, or assets to better understand their characteristics and behaviors.

DT Modeling also includes establishing relationships between dimensions and interactions between various aspects of real-world objects.

Various methods exist to create models that simulate real-world systems or phenomena (list not exhaustive):

- Mathematical models: mathematical representations of a system, process, or phenomenon described using equations and algorithms.
- Statistical models: models that use statistical methods to analyze data and make predictions.
- Simulation models: models that simulate the behavior of a system, process, or phenomenon over time using essential features and parameters of the physical entity.
- Machine Learning (ML) models: models that use ML algorithms to learn from data and create patterns based on that learning.
- Agent-based modeling is a computational technique that simulates individual agents' actions and interactions within a system.
- Physical models: physical representations of a system, process, or phenomenon used for testing and experimentation.
- Network modeling uses Graph theory and network analysis techniques to represent and analyze the relationships and interactions between entities, such as social networks, carrier networks, or computer networks.
- Etc.

Figure 6 below shows the Level 1 architecture of the DT Modeling layer with the fundamental building blocks.

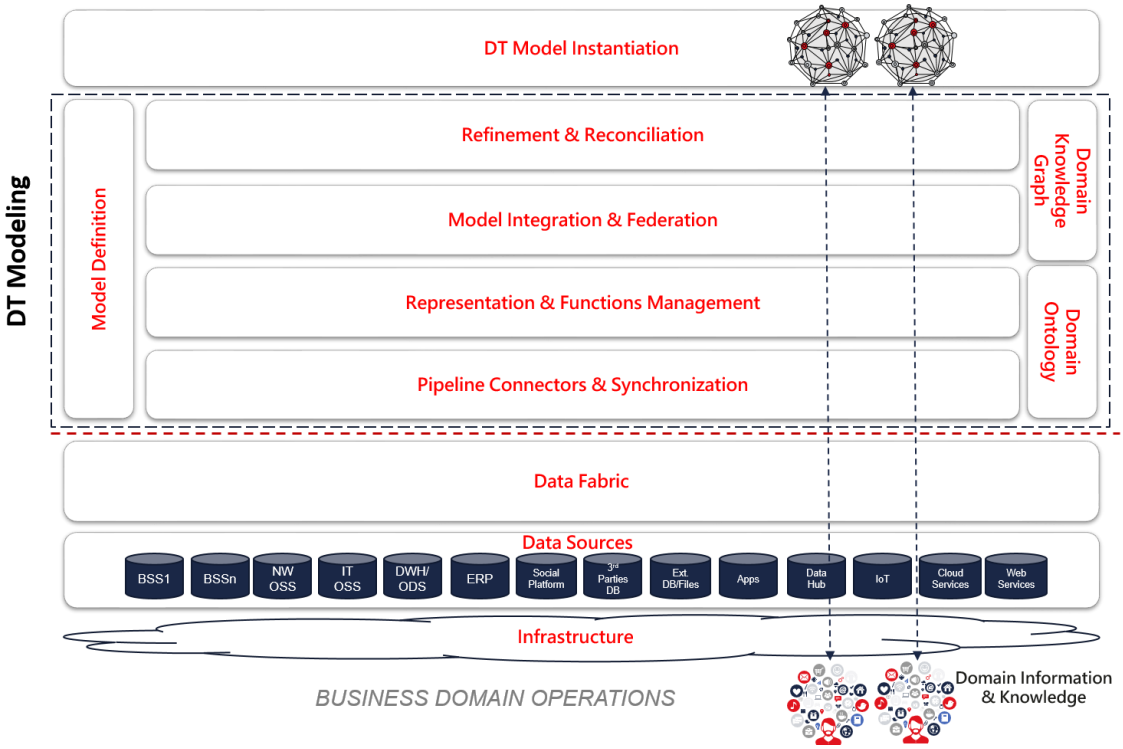

Figure 6. The DT Modeling Layer

The essential building blocks of the DT Modeling layer are:

- **Model Definition** specifies the model's objectives, scope, and boundaries. It selects the appropriate model or combination of different modeling approaches that best represent and simulate the behavior and interactions of real-world objects and systems tailored to the requirements of the particular use case and the characteristics of the mirrored real-world scenarios.
- **Domain Ontology** defines core concepts and entities that constitute or affect the specific domain and use cases, together with the relations describing how these entities relate. It provides a formal and explicit knowledge schema to understand, analyze, reuse, and share the domain knowledge of the in-scope use cases.
- **Domain Knowledge Graphs** rely on an underlying Domain Ontology and integrate domain knowledge and data as graph data models to populate with real data.
- **Pipeline Connectors & Synchronization** connect the DT4DI core system to the Data Fabric to get the necessary data sets to represent the physical objects accurately. It defines the synchronization strategy (real-time, near-real-time, batch, hybrid, etc.) and controls the data exchange between the real-world and virtual twins, keeping them synchronized.
- **Representation & Functions Management** includes the capabilities to select, map, develop, control, and enhance the features to create an accurate and effective virtual representation of real-world objects or systems.

- **Model Integration & Federation** combines multiple models to comprehensively and accurately represent and simulate real-world systems' behavior and performance and capture the interactions and dependencies between different aspects of the mirrored systems.
- **Refinement & Reconciliation** iteratively compares the virtual and real-world twins' behavior and refines the models to ensure their adherence and improve accuracy.

The DT Modeling layer provides and maintains the logical models for the DT Model Instantiation layer.

The resulting logical models widely depend on the modeling strategy and techniques selected to replicate the real-world objects in scope.

5.4. DT Model Instantiation

The DT Model Instantiation layer defines the physical structure of the data and specifies how data is deployed, stored, organized, formatted, accessed, and explored.

A physical model is created after the logical model has been designed in the DT Modeling layer and provides a blueprint for implementing the databases and data files used by the application components.

This layer includes digital twin and traditional model instances that co-habit and contribute to building holistic representations of composite and articulated real-world objects.

It specifies the data types, tables, columns, graph structures, indexes, attributes, values, and other data and file objects needed to implement databases and store the data.

The DT Model Instantiation layer also assesses the model's implementation and operations requirements and behavior, such as performance, availability, consistency, security, scalability, and frequency of the updates (real-time, near real-time, or batch mode).

Figure 7 below shows the Level 1 architecture of the DT Model Instantiation layer with its fundamental building blocks.

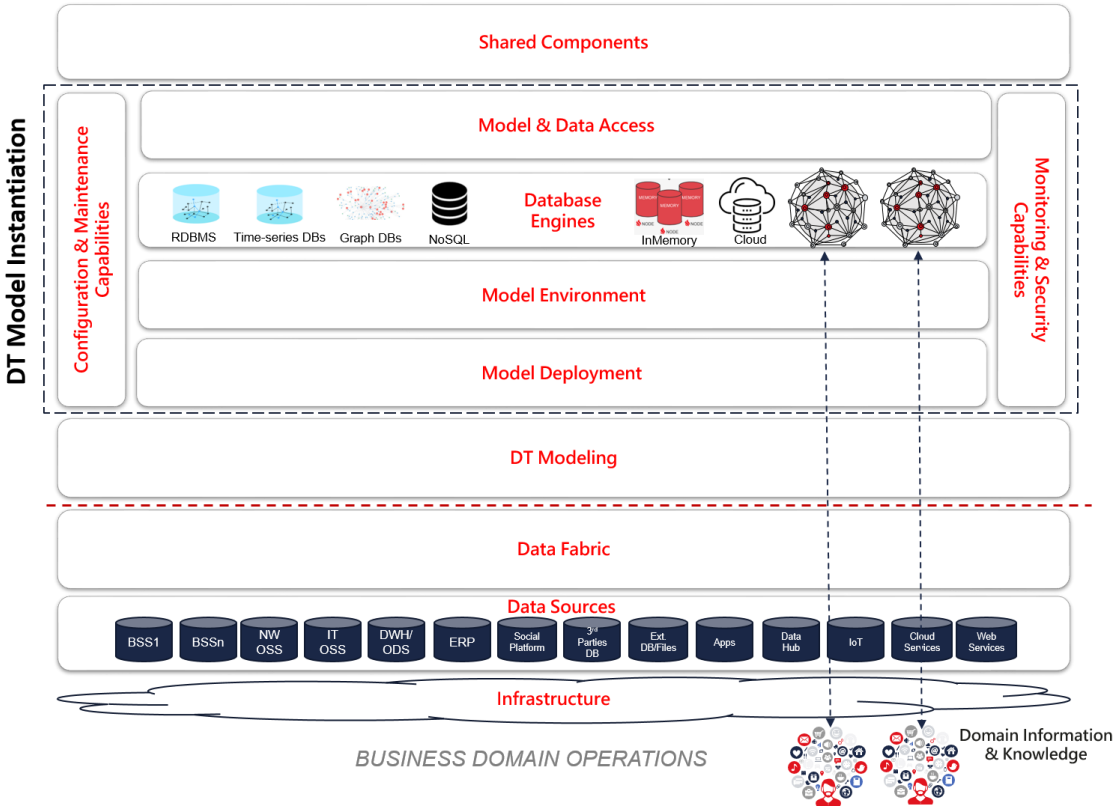

Figure 7. The DT Model Instantiation Layer

The building blocks of the DT Model Instantiation layer are:

- **Model Deployment** defines the deployment strategy, process, and techniques to deploy and operationalize the data models, the infrastructure strategy and architecture to host, connect, and integrate the models, the containerization and virtualization strategy, and other required hardware and software resources. It considers requirements such as capacity, scalability, reliability, and security.
- **Model Environment** selects and sets up the infrastructure, software, and system solutions to create, operate, and maintain digital twins and any model effectively.
- **Database Engines**' architectural block selects and provides the tools and functionality to efficiently store, manage, and retrieve the models' data. Common database engines include Relational Database (RDBMS), NoSQL, Columnar DB, Graph Database, Time Series Database, and NewSQL Database Management Systems. Domain and use cases' requirements drive and influence the choice of the database engine (data model, data types, volume, quality, scalability, performance, consistency, security, costs, etc.).
- **Model & Data Access** provides interfaces (e.g., APIs) for retrieving, inserting, manipulating, and deleting objects, attributes, and relationships of the models stored in the databases. It acts as an intermediary between the application layer and the underlying database engines, abstracting the complexities of data access and providing a consistent and standardized interface to interact with the data.
- **Configuration & Maintenance Capability** includes capabilities crucial for instantiating and configuring the DTs and any models, tailoring the modeled

systems' behavior, functionality, and performance in line with expectations, and continuously improving and fine-tuning them.

- **Monitoring & Security Capability** includes capabilities to monitor the system's performance, allow for early detection and resolution of issues, and ensure the system's reliability, security, compliance, and stability. As the DT Model Instantiation layer physically stores data, security capabilities are crucial to protect and safeguard them.

5.5. Shared Components

The Shared Component layer contains pre-built software components encapsulating specific capabilities and functionalities.

The components are designed and developed to be modular, independent, and easily reusable across different parts of an application or across multiple applications and services.

Shared Components can include components and capabilities developed internally, sourced from third-party vendors, or open-source repositories.

Figure 8 below shows the Level 1 architecture of the Shared Components layer with the fundamental building blocks.

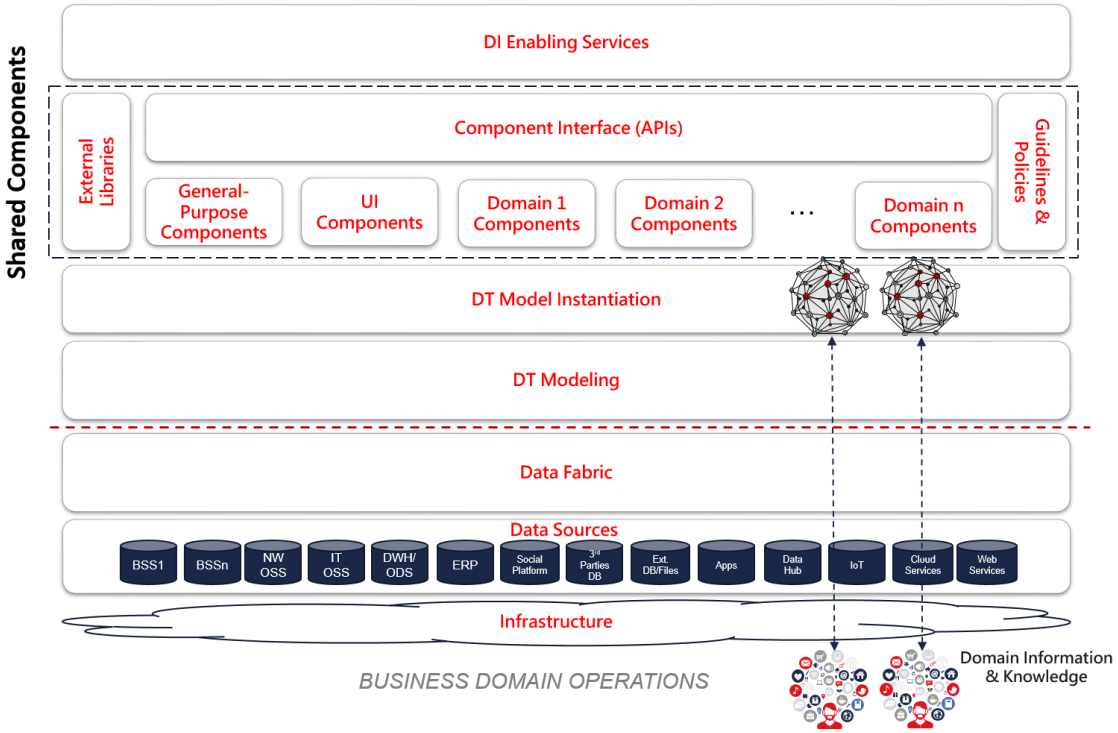

Figure 8. The Shared Components Layer

The building blocks of the Shared Components layer are:

- **Component Interface (APIs)** provides a standardized interface for accessing and utilizing the available functionalities.
- **Homogeneous subsets of components** focus on specific aspects of application development, ranging from general-purpose components, User Interface (UI) components, backend integration, and domain-specific components (e.g., marketing operations components, sales operations components, financial components, etc.).
- **Guidelines & Policies** provide generally applicable guidelines and rules all shared components shall follow.
- **External Libraries** import pre-built reliable libraries and components for standard functionality so that developers focus on building the specific aspects of the DT4DI applications and services.

Organizing shared capabilities into a dedicated layer is a recommended best practice that enables modularity, code reusability, maintainability, and separation of concerns within the architecture, reduces development effort, aligns the architecture design and development standards, and improves overall application quality and maintainability.

DT4DI developers can import pre-built reliable libraries and components for standard functionality and focus on building the specific aspects of the DT4DI applications and services and the related business and decision-making logic.

Shared Components are iteratively enriched with newly developed or imported components.

5.6. DI Enabling Services

The DI Enabling Services layer's primary purpose is to provide well-defined services that clients (the above layers) can invoke and orchestrate.

It implements the APIs supporting concrete business applications and use cases.

The DI Enabling Services layer abstracts the underlying complexities of data access, transformation, and technical operations, allowing users to interact with the DI services in a standardized and consistent manner.

Figure 9 below shows the Level 1 architecture of the DI Enabling Services layer.

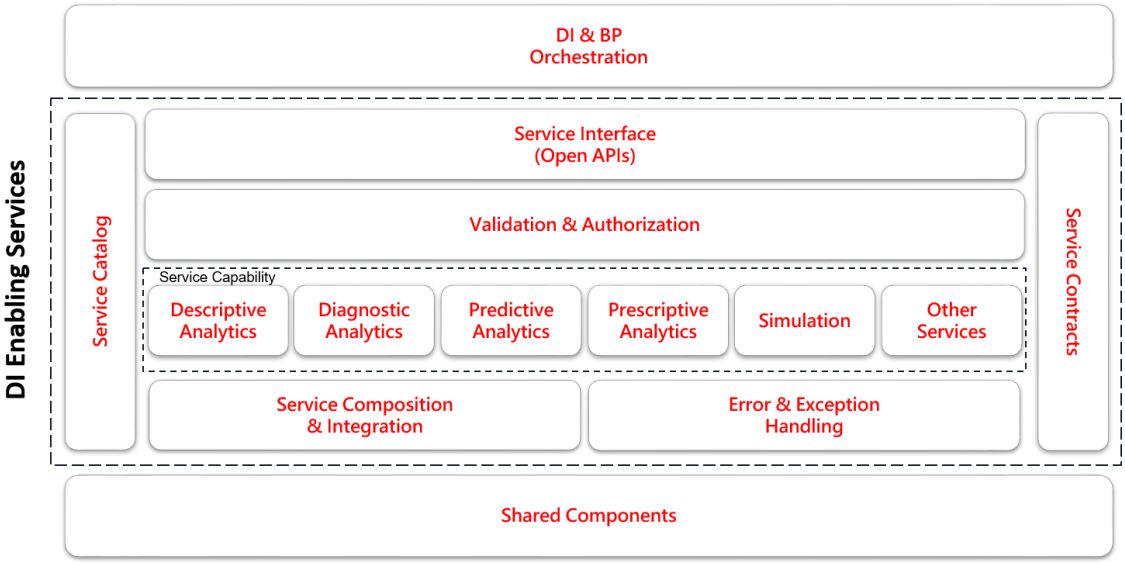

Figure 9. The DI Enabling Services Layer

The DI Enabling Services layer consists of the following main components that work together to provide access to the necessary services to the upper layers:

- **Service Interfaces (Open APIs)** are the entry points for clients to interact with the service layer, predominately based on Open APIs.
- **Validation and Authorization** ensure that the requested operations comply with business rules, security, and compliance policies. They perform validation checks on input parameters, verify user permissions, and enforce access control rules before executing the requested services.
- **Service Catalog** is a centralized repository that provides a comprehensive list of services offered by the DT4DI system. It is a reference guide to understand the available services, their descriptions, features, dependencies, service-level agreements (SLAs), and associated information.
- **Service Contracts** specify the expectations and obligations of the service layer and its clients, including non-functional aspects like security, performance, compliance, explainability, reliability, etc.
- **Service Capability** contains the business logic and functionality of the services. It encapsulates the services' rules, algorithms, and operations, executes the necessary tasks, and interacts with the Shared Components layer and the Model & Data Access capability of the DT Model Instantiation layer. Essential DI services are descriptive, diagnostic, predictive, prescriptive, and simulation capabilities.
- **Error & Exception Handling** catches and manages exceptions, providing clients with appropriate error messages or status codes. They also log error information for troubleshooting and monitoring purposes.
- **Service Composition & Integration** combines and integrates multiple components and services to create more complex services. It also enables communication and interaction with external systems or services.

The DI Enabling Services APIs support different stages of the E2E DT4DI business use cases, as depicted in Figure 10:

- What happened (Descriptive analytics)
- Why it happened (Diagnostic analytics)
- What will happen next (Predictive analytics)
- What should I do in the present situation (Prescriptive analytics)

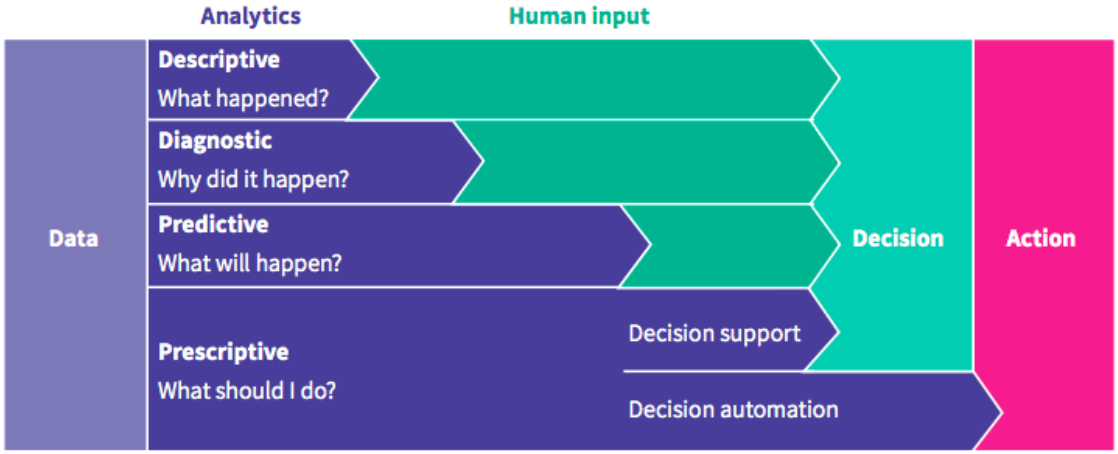

Figure 10. Essential DI Services

Descriptive analytics focuses on understanding "what happened" or "what is happening" by summarizing and describing the characteristics of a dataset (for example, an increase in sales of a product following a promotion). Descriptive analytics uses descriptive statistics, correlation analysis, machine learning, and visualizations to identify patterns and trends within the data, providing insights into the underlying structure and behavior of the data. Results are typically presented in reports, dashboards, bar charts, and other easily understood visualizations. Descriptive analytics is a foundational starting point to inform or prepare data for further analysis.

Diagnostic analytics identifies the cause of a specific event or outcome, answering the question, "Why did it happen?". Diagnostic analytics aims to provide insight into the root cause of an event or outcome, helping organizations solve and prevent problems. Diagnostic analytics involves various techniques, including drill-down, data discovery, data mining, correlations, data visualization, statistical analysis, and machine learning. Diagnostic analytics enables companies to make more informed decisions about remediating problems and driving continuous improvements.

Predictive analytics uses transactional and historical data and various statistical techniques, from data mining to machine learning, to predict future events, outcomes, or otherwise unknown events, answering questions such as "What is likely to happen?". Predictive analytics is used to identify patterns, relationships, and trends in data and predict future events based on those findings. Predictive analytics aims to gain insights to help organizations make more informed decisions and take proactive measures to improve future outcomes. Business stakeholders can use predictive analytics to forecast customer behavior, purchasing patterns, sales trends, supply chain throughput, inventory demands, perform product recommendations, churn prediction, fraud prediction, etc.

Prescriptive analytics makes recommendations for future actions or decisions, answering questions such as "What should be done?" or "What can we do to make (or not to make) _____ happen?". Prescriptive analytics builds on the insights generated from descriptive, diagnostic, and predictive analytics. Prescriptive analytics uses decision intelligence, advanced mathematical techniques such as optimization, machine learning, and simulation to identify the best actions to achieve specific goals. It also identifies trade-offs and constraints and evaluates the potential impacts of different decisions. Prescriptive analytics suggests courses of action or execution strategies.

5.7. DI & BP Orchestration

The DI & BP Orchestration layer designs and integrates the workflows of DT4DI Use Cases, combining and coordinating the following two different but complementary processes:

- Business Process (BP) provides the process representation and composition of the process steps, data objects, and decision-making points as a sequencing process aligned with business goals.
- Decision Intelligence (DI) process diagrams the flow that links inputs and outcomes, their relationships, and the consequences of decisions. It handles and solves the critical decision points in the related Business Process.

This layer defines and optimizes the business and decision-making processes, data, and rules required to execute the E2E DT4DI use cases. It creates workflows representing the sequence of tasks to be performed, the conditions under which tasks should be completed, and the rules governing decision-making.

The DI & BP Orchestration layer enables aggregating loosely-coupled services and related decision-making points like interlocking LEGO bricks as a cohesive process aligned with business goals and decision-making strategies.

The bricks are the functionalities, data objects, interfaces, and DI capabilities the below layers provide.

The Business Process (BP) identifies, locates, and classifies the decision-making points.

For those decision-making points that need a DI process (typically the complex decisions), BP triggers the corresponding DI process whose outcome (the decision) is input back to BP.

BP and DI are cross-linked, automated, integrated, and optimized with the business and decision-making rules and policies to create the end-to-end DT4DI business use cases.

The DT4DI use cases are cross-domain and combine analytics, executive, and operations steps to turn data into actions and concrete business outcomes.

Consequently, the DT4DI use cases link, assemble, and steer analytical, decisional, and operational process steps:

- Problem, experience, and opportunity definition
- Data sources and data points definition
- Decision-making points definition
- Data collection step
- Data preparation step
- Analysis & Diagnosis steps
- Decision strategy definition
- Decision options development
- Options evaluation
- Simulation and what-if analysis
- Option selection (final decisions)
- Decision implementation triggering operational capabilities
- Outcome monitoring and evaluation
- Feedback loops enabling process and decision-making adjustments and improvements according to the effectiveness of outcomes and actions, new information, or changing conditions.

Figure 11 below shows the Level 1 architecture of the DI & BP Orchestration layer with the essential building blocks.

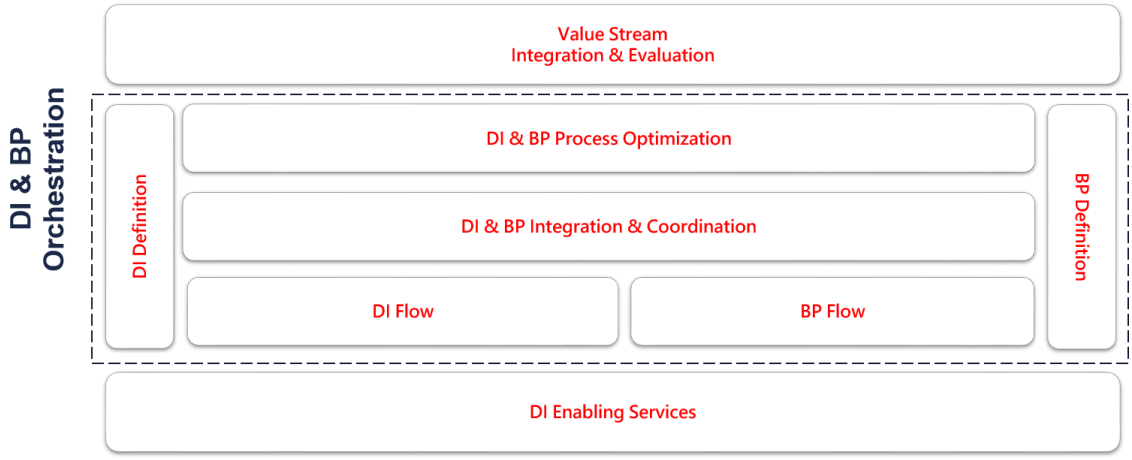

Figure 11. The DI & BP Orchestration Layer

The DI & BP Orchestration layer consists of the following primary elements that work together to build up the DI & BP business and decision-making process:

- **BP Definition** provides a comprehensive definition of As-is (when applicable) and To-be business processes and the corresponding formal specification used as starting point to model the business process flows.
- **BP Flow** streamlines and formalizes the sequence of steps and their relationships to execute a business process in the selected formal notation. It involves the identification of inputs, activities, data, decision-making points and their classification (simple, complex, automated, human-in-the-loop, etc.), dependencies, goals, outputs, and the roles and responsibilities of people interested in the process. In DT4DI, business processes shall externalize and explicitly describe critical and complex decision-making points, as shown in Figure 12 below.
- **DI Definition** involves the specification of the Decision Meta-Model. It specifies the decision-making points in the business use cases and processes and the related decision-making processes, the goals, inputs, criteria, decision-makers, roles & responsibilities, expected outcomes, risks, and all relevant aspects involved.
- **DI Flow** captures actions, causal links, and outcomes in the selected formal language (e.g., CDD, DMN, Decision Trees, etc.), enabling the mapping of structured decision-making process flows to automate through AI (and non-AI) components provided by the DI Enabling Services layer.
- **DI & BP Integration & Coordination** interconnect business processes (BP) and decision intelligence (DI) flows, identify the interfaces that trigger the DI process, and how decisions are reverted into the business processes.
- **DI & BP Process Optimization** establishes feedback loops between decision intelligence and business processes, evaluates the impact of decisions on the processes, monitors key performance indicators to assess effectiveness, and refines decision-making criteria, guidelines, and the execution of business processes, optimizing business processes and decision-making practices. This layer ensures that processes and decisions evolve with changing business dynamics and objectives.

One of the pillars of DI, and consequently one of the foundational concepts of DT4DI, is that decisions meeting some requisites (complex, critical, business-relevant, etc.) shall be addressed and solved with DI practices and the applicable enabling technologies (e.g., AI, DT).

The in-scope business process shall then trigger the Decision Intelligence process, which "implements" the decision-making of relevant decisions.

Figure 12 below shows an illustrative example of this core concept.

We borrow the DI diagrams from Lorient Pratt [36] and her company Quantellia [53], pioneers and evangelists of DI.

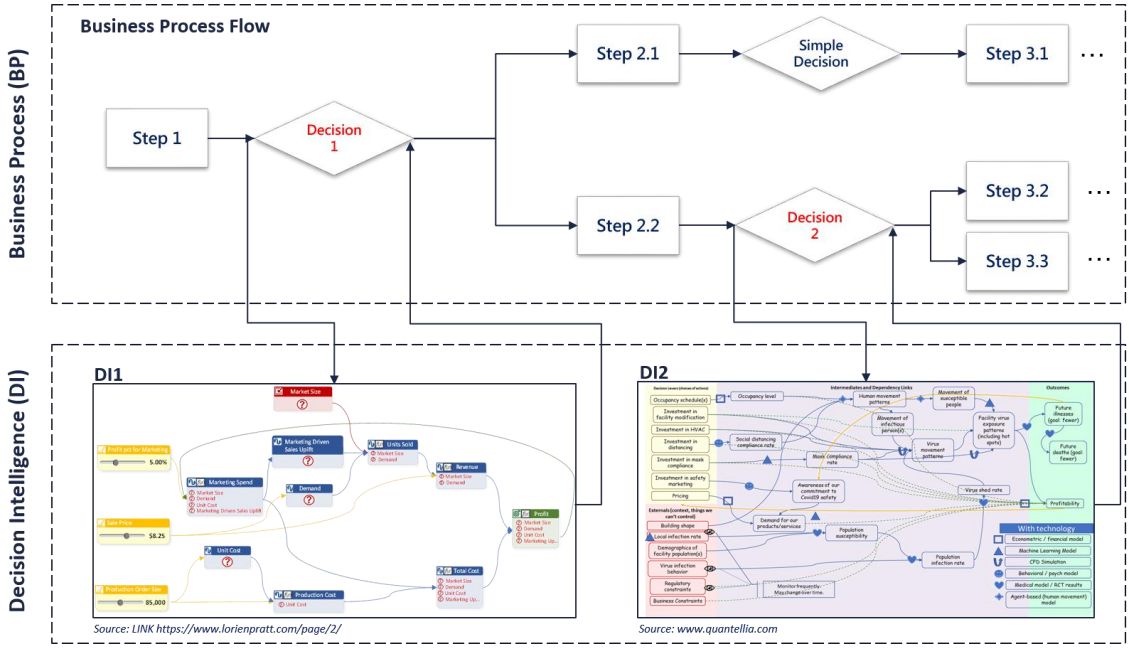

Figure 12. DI & BP Orchestration - Example

DI & BP Orchestration layer incorporates feedback loops and continuous improvement mechanisms in the DI & BP Process Optimization block.

This involves monitoring and evaluating the outcomes of the executed DI & BP workflows and using this information to refine and optimize the processes over time.

By continuously analyzing and adjusting the workflows, businesses can ensure that they adapt to changing conditions and improve their performance and outcomes.

The DI & BP Orchestration layer interacts closely with the DI Enabling Services layer to use the proper capabilities (insights, predictions, recommendations, simulation, etc.) and data objects at the right time to design and implement the business and decision-making processes.

5.8. Value Stream Integration & Evaluation

While the DI & BP Orchestration layer provides the instructions for building specific DT4DI processes and systems, this layer defines all the necessary aspects to deploy and operationalize the DT4DI solutions to cooperate with other processes and systems outside the DT4DI domain for achieving the expected business objectives.

In this context, a Value Stream refers to the complete sequence of processes and activities required to create value for customers and stakeholders. It encompasses all the steps to create, deliver, and support a target product, service, or outcome.

Value Stream Integration involves assessing the organization's current business and technical operations and designing and implementing plans to address the following imperatives (list not exhaustive):

- Integration of the DT4DI solution with enterprise business rules and policy management.
- Integration of the solution into current operations management processes.
- Integration with supply chain management, inventory management, and logistics when physical objects are involved (e.g., devices, smartphones, tablets, etc.)
- Distribution and delivery management.
- IT Infrastructure (hardware, base software, platforms) and cloud deployment strategy.
- Definition of the operations schedules in Production.
- Systems settings and configuration management.
- Integration with Knowledge Management.
- Integration with the Monitoring process and platforms.
- Integration with Identity Management and Security Management processes and platforms.
- Integration with sustainability business management (when applicable).
- Etc.

Value Stream Evaluation is responsible for providing quality assurance and performance management capabilities to end-users and stakeholders to:

- Validate the implemented business and decision logic.
- Evaluate outcomes generated by the DT4DI Use Cases regarding quantity, quality, effectiveness, the success rate of the recommended/selected activities, sustainability, revenue results, costs, and customer experience impacts.
- Assess the overall value of the Value Stream and the related BP and DI processes.
- Identify weaknesses and use them for subsequent improvements.
- Collect feedback and inputs from users and stakeholders to improve the BP and DI processes.
- Ensure compliance with regulations, ethical, legal, and security policies.

Figure 13 below shows an example of a generic product go-to-market value stream supported by the DT4DI core domain architecture.

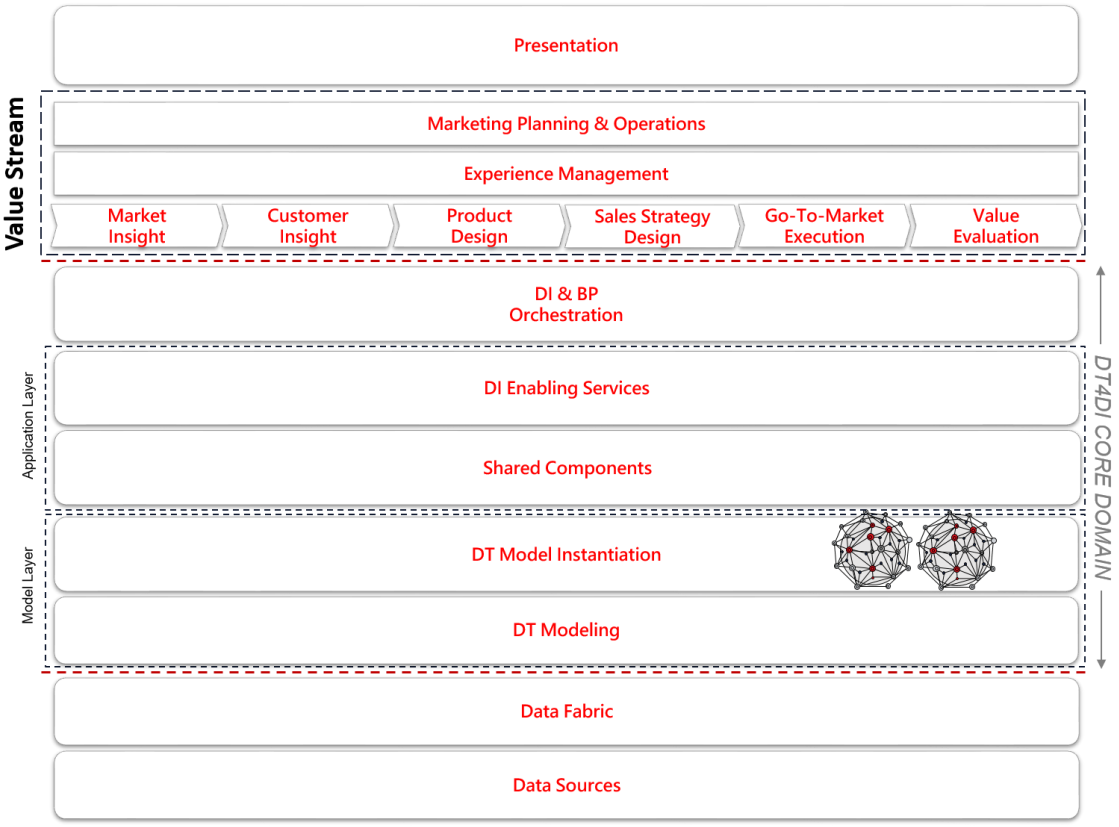

Figure 13. Example of DT4DI-enabled Value Stream

In marketing operations, value streams encompass several essential processes and activities:

- Market Insight: This involves understanding various market aspects, including competition, trends, consumer behavior, and other factors influencing product or service demand.
- Customer Insight: Deeply exploring customers' characteristics and preferences and segmenting customers to define target groups.
- Product Design: Designing products or portfolios tailored to the identified target groups or markets.
- Sales Strategy Design: Determining campaign parameters, such as target group, product selection, and channels to be used.
- Go-to-Market Execution: Implementing marketing plans, such as delivering targeted messages to the identified target groups and promoting recommended products or services.
- Value Evaluation: Thoroughly assessing campaign feedback and analyzing results to identify areas for improvement and achieve better outcomes.
- Experience Management: Facilitating integrated collaboration between marketing and customer experience teams, emphasizing strategic alignment

and coordinated execution. It includes post-use evaluation to collect user feedback and provide exceptional care capabilities.

5.9. Presentation

The presentation layer delivers the information and outcomes to the end-users and stakeholders across multiple channels, devices, and user interfaces.

The multichannel layer provides users with a seamless and consistent experience across all channels considering each device's unique capabilities and constraints (e.g., small, medium, and big screen, audio quality, etc.).

The presentation layer includes user interfaces (UIs) that enable users to interact with the application, including menus, buttons, forms, tables, charts, and other visual elements. It may also include dashboards, reports, notifications, alerts, and chat boxes that give users real-time information on key business metrics and performance indicators.

Figure 14 below emphasizes the Presentation layer in the onion view of the DT4DI Architecture.

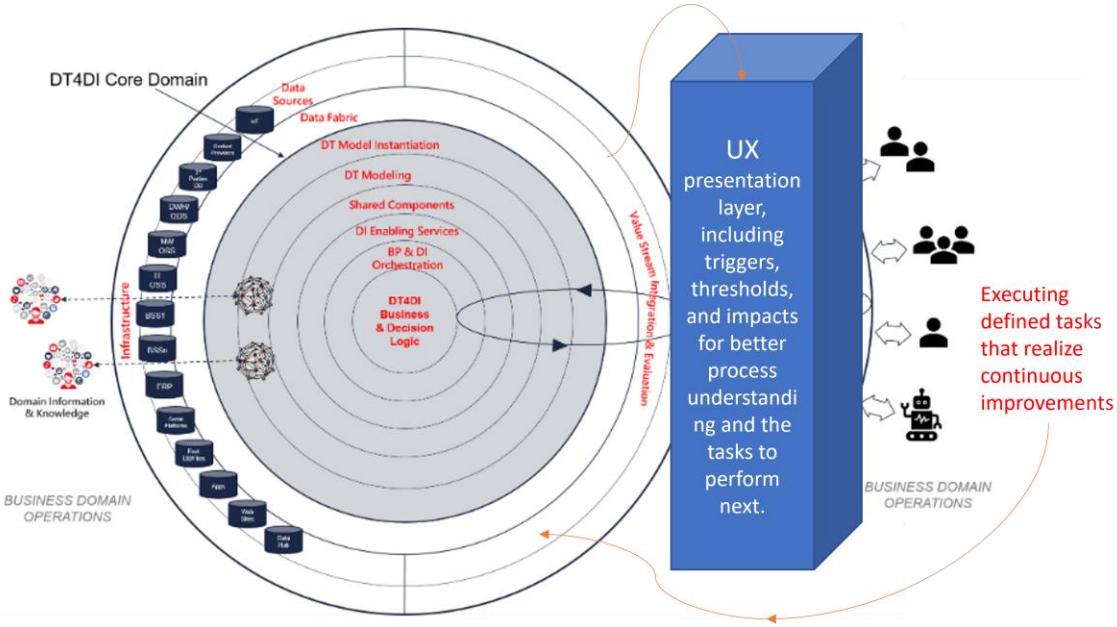

Figure 14. DT4DI Onion Architecture highlighting the Presentation layer.

The Presentation layer design shall consider the following:

- Support continuous improvement tasks (feedback, adjustments, inputs, etc.).
- Usability and accessibility to ensure all users can interact effectively with the application.
- Personas, task flows, and user feedback to create an intuitive and efficient user experience.
- Scalability to support massive deployments.
- Adaptability, ensuring it can accommodate changes in user needs, emerging technologies (AI, AR/VR, etc.), and evolving business requirements over time.

The Presentation layer allows users to interact with the DT4DI solution and the whole value stream, monitor the value stream's performance and related processes' performance, identify trends, evaluate outcomes, detect deviations, provide feedback, and continuously adjust processes and decisions to improve outcomes and results.

5.10. DT4DI Level-1 Architectural View (L1)

Figure 15 below shows the holistic view of Level-1 (L1) DT4DI Reference Architecture, assembling all the architectural components as bricks like a LEGO construction.

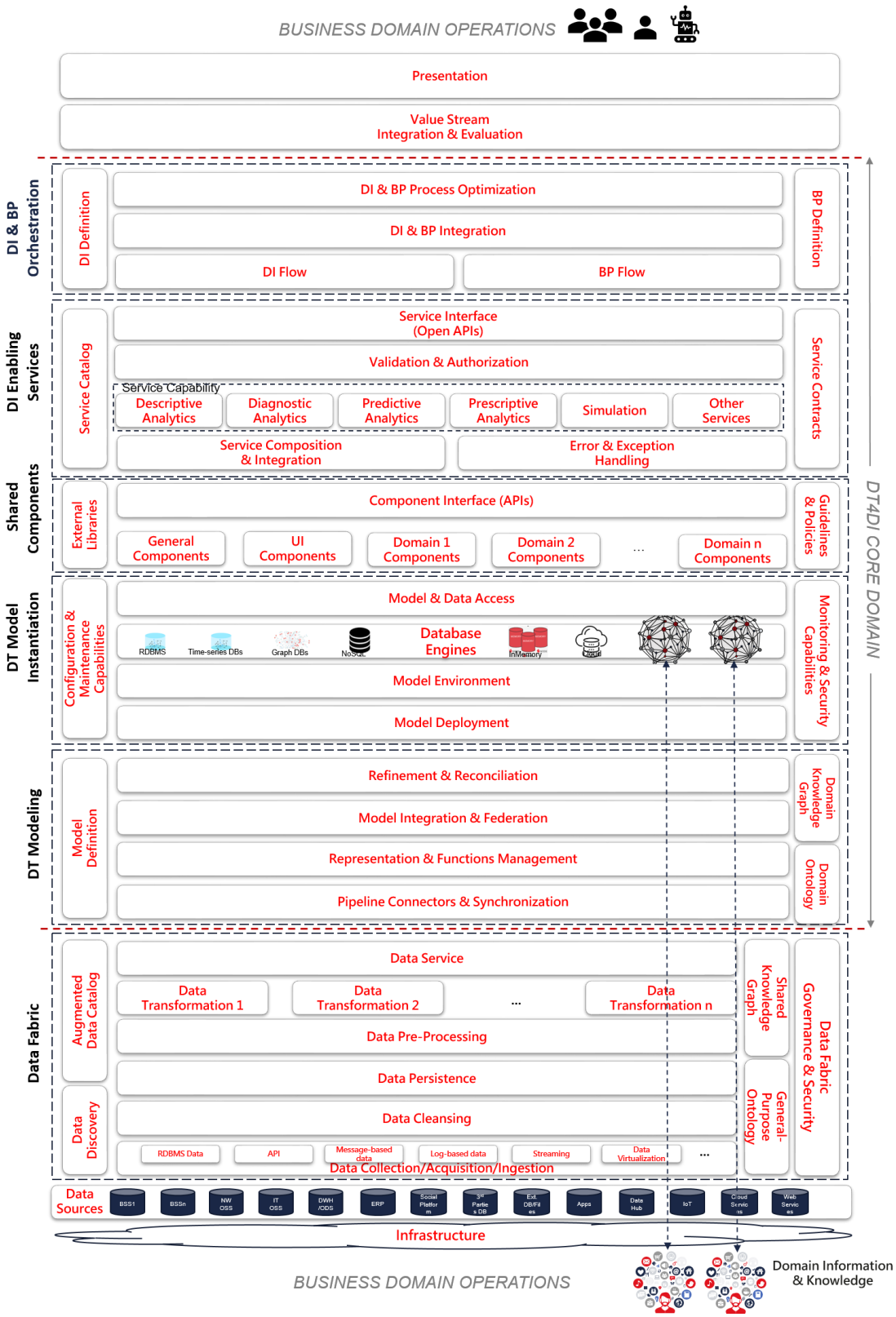

Figure 15. DT4DI Reference Architecture - L1

The overall view of the Level-1 Architecture illustrates how these components interact with each other and the data flows between them.

The Data Fabric provides the data to the DT Modeling layer, which creates logical models implemented into the DT Instantiation layer and used by the Shared Components and DI Enabling Services to implement the open APIs supporting the business use cases.

The DI & BP Orchestration layer defines and interlinks DI and BP processes and uses the exposed APIs to assemble the workflows for each use case.

The Value Stream Integration & Evaluation layer incorporates the DT4DI capabilities into the organization's E2E value stream. It validates the implemented business and decision logic, evaluates outcomes, and assesses the overall benefits of the DT4DI-enabled values stream.

The Presentation layer interacts with end-users/consumers across multiple channels and devices, delivering information and outcomes and collecting feedback and inputs.

6. Business-Driven Design Approach

You've got to start with the customer experience and work backward to the technology. You can't start with the technology then try to figure out where to sell it."

Steve Jobs, 1997

While the DT4DI layered architecture (Figure 3) defined in the previous section is agnostic to any specific design approach (it can be read top-down and bottom-up), the DT4DI Onion Architecture (Figure 4) explicitly recommends a business-driven design approach.

Business-driven design starts from a deep understanding of the business domain or problem being addressed and aligns the solution design with the business and decision-making logic, which are at the core of the DT4DI Onion architecture.

Business-driven design should also be able to quickly accommodate changes without needing to revisit the entire DT4DI system.

A Business-driven design approach emphasizes the expressiveness of the business problem, the business context, and the identification of internal/external dependencies contributing to the business outcomes.

In Decision Intelligence, this is relevant because it allows for assessing the business goals and objectives and business impacts of decisions, simulating different decision paths (what-if simulation), optimizing the decision-making process, and identifying opportunities for improvement.

There can be two perspectives to the business-driven design approach:

1. Scoping perspective: defining the scope of Decision Intelligence in a manner applicable across a wide range of business decision-making scenarios.
2. Use case perspective: translating business decision-making scenarios into atomic decision problems and their components.

We explain more details of these two perspectives in the following paragraphs.

6.1. Business Driven Design Approach – Next Best Experience (NBX)

Let's visualize the business-driven design approach in Decision Intelligence through an example of the Next Best Experience (NBX) referring to the TM Forum ODA Business architecture (IG1277, [56]).

NBX is a new approach that evolved from the other "Next-Best" paradigms, such as Next-Best-Action (NBA), Next-Best-Offer (NBO), and Next-Best-Product (NBP).

NBX differs from existing Next-Best paradigms in that the operational actions are tailored more towards enhancing customer experience than influencing their behavior based on sales or marketing targets.

In a typical decision-making scenario related to NBX, a stakeholder needs to determine which actions may influence the experience positively or enhance a particular experience metric and what action needs to be initiated.

It is not possible to make such decisions on an individual customer basis, and that is where the need for an automated, repeatable, and consistent Decision Intelligence mechanism arises.

Identifying what capabilities are required to support the value delivery for various organizational stakeholders is equally important.

The following diagram shows the contextual view of Decision Intelligence in the NBX scenario.

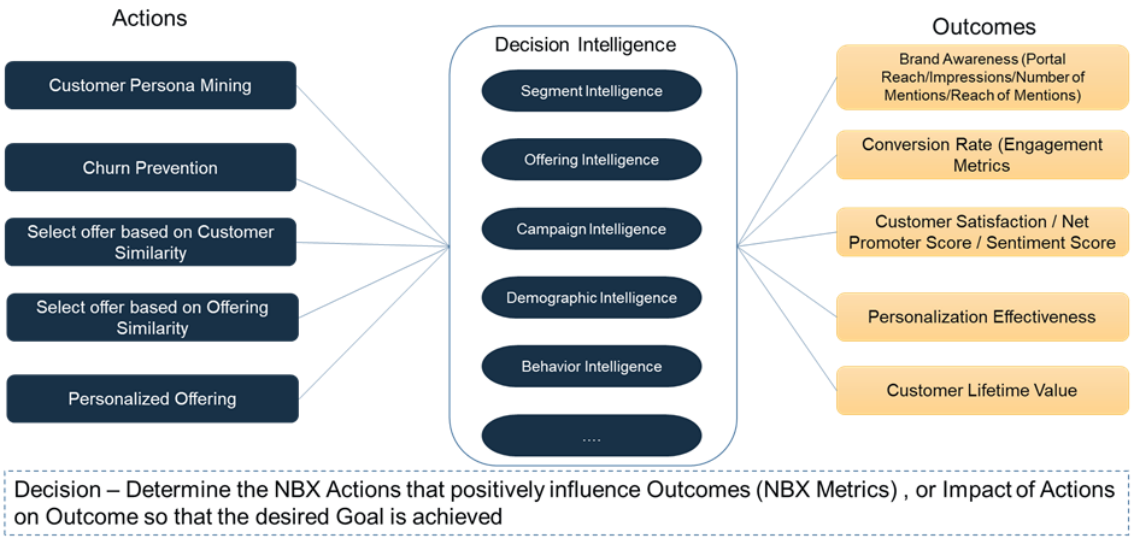

Figure 16. Choice of Actions, Decisions, Outcomes

ODA Business architecture (IG1277) defines a capability model and set of business value streams organized as domains and vertical aspects.

According to this model, a simple value stream of a Shopper and a Marketer is shown in the diagram below (Figure 17).

The delivered value for a shopper is an enhanced shopping experience; for the marketer, the value is offering recommendation that leads to customer engagement and sales.

Each stage of the Shopper is supported by the Marketer's identification and collection of insights related to the Shopper's needs, wishes, and context (business scenario, market condition, timing, demographics, etc.).

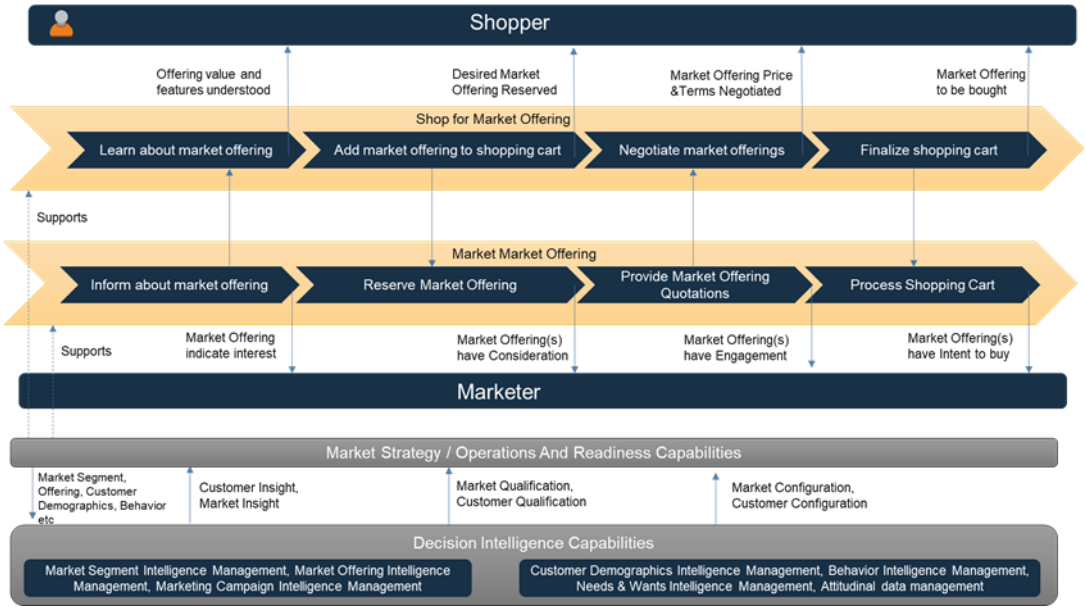

Figure 17. Shopper-Marketer Value Stream Example

A set of capabilities supports the two value streams (refer to the IG1277 capability model): marketing strategy to readiness capabilities, marketing operations capabilities, etc.

Among these capabilities, the key insights and decision recommendations are derived through a more granular set of capabilities belonging to each domain/vertical such as:

- Marketing domain: Market Segment Intelligence Management, Market Offering Intelligence Management, Marketing Campaign Intelligence Management.
- Customer domain: Customer Demographics Intelligence Management, Behavior Intelligence Management, Needs & Wants Intelligence Management, and Attitudinal data management.

These capabilities will typically enable automated decision-making regarding NBX actions that will enhance the shopper experience, resulting in better NBX performance. This could include providing personalized product recommendations based on the customer's past purchases, recognizing shoppers when they enter a store, and delivering personalized offers and promotions.

In the Shopper value stream, an NBX decision can be made on behalf of the Marketer on the personalized marketing offerings that can yield positive engagement during the shopping process. For example, a group of offerings can be shortlisted and presented based on customer demographics, needs & wishes, and customer similarity.

In the Marketer value stream, an NBX decision can trigger potential experience-enhancing action and associated offering mechanisms. In this case, it can be a decision on churn prevention that will result in including a discount on monthly payments or a waiver of the shipping charges associated with a product.

Thus, based on the specific business value stream, decision-making may require different insights supported by data and operational capabilities. The business stakeholder can identify potential requirements to enable automated Decision Intelligence processes and, when coupled with the decision operational flow, can determine the optimum data required, thus reducing time spent extracting and curating unwanted data.

The business-driven design approach leveraging value streams and capabilities also helps to define the decision requirements clearly.

The above perspective focuses on a methodology for identifying the scope of Decision Intelligence by looking at specific business value streams and supporting decision-making capabilities.

Another perspective is how seamlessly translate stakeholder goals into a business-driven design approach.

The illustration in Figure 18 below shows the decomposition of the business goals into the decision goals and constituent intelligence components in the case of NBX.

Figure 18. Decomposition of business goals into decision goals

The NBX goal may originate from a business vision statement from the CXO level with high-level expectations, for example:

- Provide customers with experiences that are relevant, timely, and helpful by understanding customer behavior and preferences, and then using that information to make real-time decisions, create experiences that are more likely to keep customers engaged and satisfied.

After the requirement analysis, we formulate the goal as follows:

- As a Marketer, determine the best offer to Shopper A matching the demography pattern of GenX so that Customer Lifetime Value growth is > 15 %

From the Marketer objective, the critical decision requirements are:

- **Decision 1:** Given the customer's existing orders, predicted CLV growth, customer loyalty based on churn propensity, and account health, determine the discount/reward that can be applied to different classes of offers so that predicted engagement metrics have positive growth.
- **Decision 2:** Given the build schedule and readiness of services associated with various offers and service health history, determine which offer may suit the customer so that the predicted health of services related to each offer is green.

- **Decision 3:** Given product eligibility, customer profile, current needs, and predicted needs, determine the set of offers from Decision 2 that are suitable for the customer so that there is an 80% match between current and predicted needs.
- **Decision 4**: Given the Shopper profile, demography = GenX, marketing offers from Decision 3, predicted propensity to buy, historical offer uptake pattern, average CLV growth for Shoppers per offer, and predicted CLV growth, determine the best offer so that MAX (Avg CLV growth, Predicted CLV growth) > 70%

The above decision requirements form the basis for the decision model, representing the interrelation between these decisions, internal/external dependency, the input/output requirements, and the sequencing of logic/rules to influence a decision outcome.

The decision model also forms the basis for the intelligence requirements listed below, which an AI/ML or Analytics models address.

Decision Intelligence Requirements	Decision Intelligence approach
Identify if the shopper has churn propensity given credit score/loyalty points, demography (gender, age, tenure), balance, activity (transaction count, transaction age), and financial status.	Potential binary classification problem, where the goal is to predict whether a shopper will churn. The features that predict churn propensity include credit score/loyalty points, demography, balance, activity, and financial status.
Prediction of customer lifetime value (CLV) based on labeled shopper profile, demography parameters, and current products.	Potential regression problem, where the goal is to predict a shopper's CLV. The features that can predict CLV include labeled shopper profiles, demography parameters, and current products.
Prediction of engagement metrics and preferred channels based on historical pattern and profile/needs similarity.	Potential classification problem, where the goal is to predict a shopper's engagement metrics and preferred channels. The features that predict engagement metrics and select channels include historical patterns and profile/needs similarity.
Expected positive CLV based on the shopper profile and churn propensity (if Predicted CLV less than target)	Potential regression problem, where the goal is to predict the expected positive CLV for a shopper given their profile and churn propensity. The features that predict expected positive CLV include shopper profile and churn propensity.
Identify the product needs of the shopper based on current product mix , demography, churn propensity , product similarity , product wishlist.	Potential content-based filtering problem, where the goal is to recommend products to a shopper based on their current product mix, demography, churn propensity, product similarity, and product wishlist.
Identify and group offers (current + existing) based on the popularity mix (users who bought X also liked Y) , profile similarity, churn propensity , product needs.	Potential collaborative filtering/association rule problem, where the goal is to identify and group offers based on the popularity mix (users who bought X also liked Y), profile similarity, churn propensity, and product needs

Decision Intelligence Requirements	Decision Intelligence approach
Predict health of services associated with the short-listed offers.	Potential time series prediction problem, where the goal is to predict the health of services associated with the short-listed offers. The features that predict the health of services include historical data on service usage, service performance, and service availability.
Given service constraints such as traffic growth, number of users, and geography, calculate the risk of associating a high revenue potential service for the shopper as part of the offer.	Potential constraint-optimization problem, where the goal is to calculate the risk of associating a high revenue service for the shopper as part of an offer given a set of service constraints. The constraints include traffic growth, the number of users, and geography.
Calculate the discount that can be offered to shopper based on churn propensity, offer mix, credit score/loyalty points, predicted CLV.	Potential gamification problem, where the goal is to calculate the discount to offer to shoppers based on their churn propensity, offer mix, credit score/loyalty points, and predicted CLV. The discount should be high enough to incentivize the shoppers to stay with the company but low enough that the company does not lose earnings.

The Decision Intelligence (DI) requirements form the basis for identifying the features and data requirements to design the DI Enabling Services.

6.2. Decision Modeling & Decision Meta-Model

In the business-driven NBX DI scenario, decision requirements identified based on the stakeholder value streams help suggest the required data insights.

The following two challenges remain:

1. Represent decisions with external and internal influencers, dependencies, policies, rules, and analytical models, so stakeholders may validate them and ensure explainability, transparency, repeatability, and completeness.
2. Ensure consistency when representing data in decisions and insights.

These issues point to the need for two additional capabilities to operationalize Decision Intelligence:

- Decision Modeling.
- Decision Meta-Model.

Decision Modeling is a visual and machine-readable representation of decisions.

In the DT4DI Reference Architecture, the Decision Modeling is the outcome of the 'DI Flow' component of the 'DI & BO Orchestration' layer shown in Figure 11.

Decision Modeling helps to:

- Identify decision needs and gaps.
- Identify the impact of changes from decision dependencies.
- Provide standardized structure to decisions.

- Enable interaction between various stakeholders with different technical backgrounds.
- Identify input and output, policies, and know-how required for decisions.
- Enhance explainability of decisions.
- Make decisions data-driven in collaboration with humans when in the loop.

Examples of Decision Modeling techniques available today are:

- Object Management Group Decision Modeling Notation (DMN) ([27])
- Causal Decision Design promoted by DI evangelist Lorien Pratt ([36])
- Decision Trees.

Decision Meta-Model is a model that provides a structured representation of decisions and their associated elements within a specific context. It helps standardize the data entities used across the decision model. With well-defined Decision Meta-Models, stakeholders can represent the decision structure without waiting for the data to be curated and described in a conceivable form.

The specification of the Decision Meta-Model is part of the 'DI Definition' architectural block of the 'DI & BP Orchestration' Layer shown in Figure 11.

The meta-model helps the data engineering team identify the data needs, such as the data sources, transformation needs, interdependencies, etc.

DI can leverage and extend TM Forum ABDR, Customer Experience Metrics (IG1303 [51]), and Business Metrics to define the meta-model for related domains and use cases.

Decision Meta-Model is the basis for all the subsequent DI flow specification, decision execution, and data management activities required in Decision Intelligence.

After validation by stakeholders, a modeled decision can be treated with historic or synthetic data. This allows micro-experiments and simulations to help optimize decisions in relation to actions and outcomes.

6.3. Continuous Optimization – Balancing Actions and Outcomes

In the NBX example, the stakeholder expectation is to tune the actions to achieve targeted outcomes or select the steps for adjusting particular outcomes.

Continuous optimization and adaptation of decisions are needed to deliver the desired results.

For example, if churn control actions are not good enough to improve customer lifetime value, a personalized offering with attractive pricing may likely influence the customer lifetime value.

Different techniques exist to implement continuous optimization in the context of Decision Intelligence.

Simulation is one technique that can be leveraged by monitoring the outcome against the micro-experiments being carried out. This technique allows for testing different hypotheses, helping to identify the optimal approach.

Another technique leverages the incorporation of the closed-loop control mechanism. With the closed loop, the actions-to-outcomes decision flows are controlled and orchestrated based on the business rules, policies, and predicted measurements following an OODA (Observe, Orient, Define, and Act) pattern. Closed-loop techniques can control any observed variations in the outcome through a calibrated selection of actions.

An essential requirement for supporting continuous optimization is a feedback mechanism for feeding the results of actions back into the decision intelligence system so that the best approach can be determined for tuning the actions.

Consider the case of NBX, where the operational actions are decided based on their potential influence on the customer experience.

The following figure illustrates a high-level decision logic where, at the initial decision-making phase, multiple sub-branches of decision are possible for a customer having different CLV ranges, such as top 50% and bottom 50%.

For the top 50% of CLV customers, a sub-decision is made based on the Persona randomly. The choice of offering is either based on the Persona similarity (offerings determined by similar Persona), based on a popular OTT bundle, or based on plan similarity.

For the bottom 50%, a sub-decision is made based on the customer care tickets they raised in the past. Randomly, 50% of offerings are selected based on the popular bundle discounts, and the rest are selected based on data promotion packages.

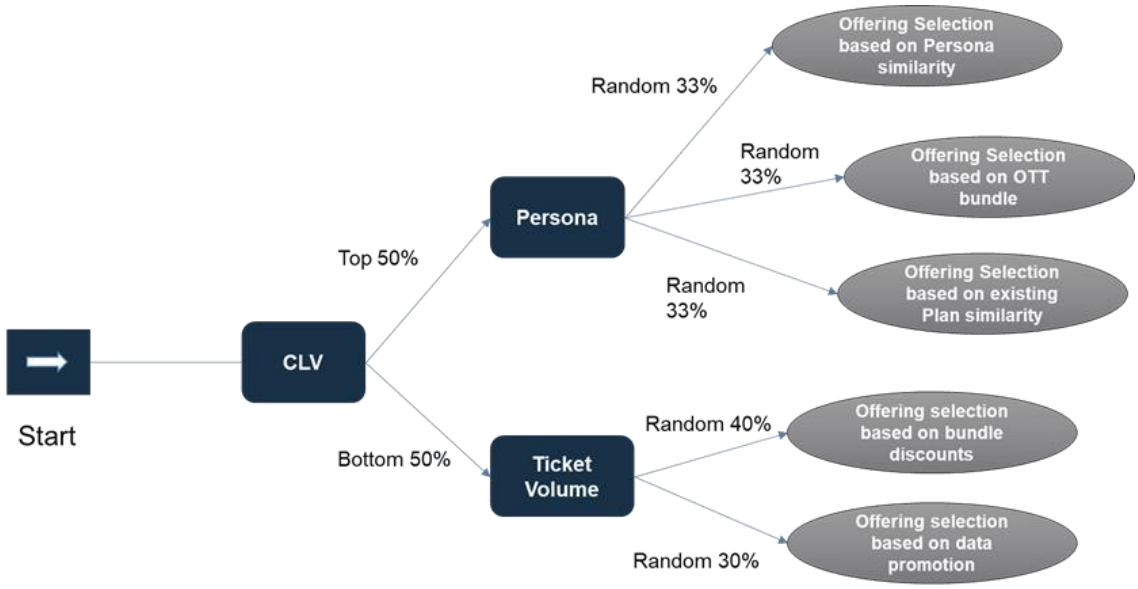

Figure 19. Initial Decision Model for NBX Scenario with random weights

After gathering sufficient feedback from customer channels and order management systems regarding product selection and quote acceptance, the random choices made during the initial decision-making phase can be further optimized.

Alternatively, an AI/ML model can analyze the collected feedback and recommend an optimal distribution of weights.

The following example demonstrates feedback collected from a customer 360-degree system to refine the weightage distribution across decision layers.

Customer	Offering Variant Recommendation	Offering Feedback	CLV	Churn Propensity	Persona	Ticket Volume
A	V1	Positive	100	Low	Business Professional	Medium
B	V2	Negative	150	Low	Gamer	High
C	V3	Negative	200	Low	Content Creator	Low
A	V4	Negative	100	Low	Business Professional	Medium
D	V2	Positive	50	High	Student	Low

Statistical sampling of decision feedback

Figure 20. Statistical samples collected based on the Decision outcome feedback (Offering recommendation feedback)

The above example shows a set of feedback collected based on the customer acceptance of the NBX-based offerings.

In addition to the positive or negative feedback (i.e., acceptance of the offerings selected for the particular customer channel), the feedback also captures the input values such as CLV, Churn propensity, Persona, Customer Care ticket volume, etc.

Once enough samples are available, statistical analysis or a more advanced algorithmic analysis incorporating AI/ML is carried out to determine the recommended offering that has a positive customer response.

This information is fed back into the Decision Intelligence system to revise the weights, adjusting random weights in the initial phase with more optimum weights, as shown in the diagram below.

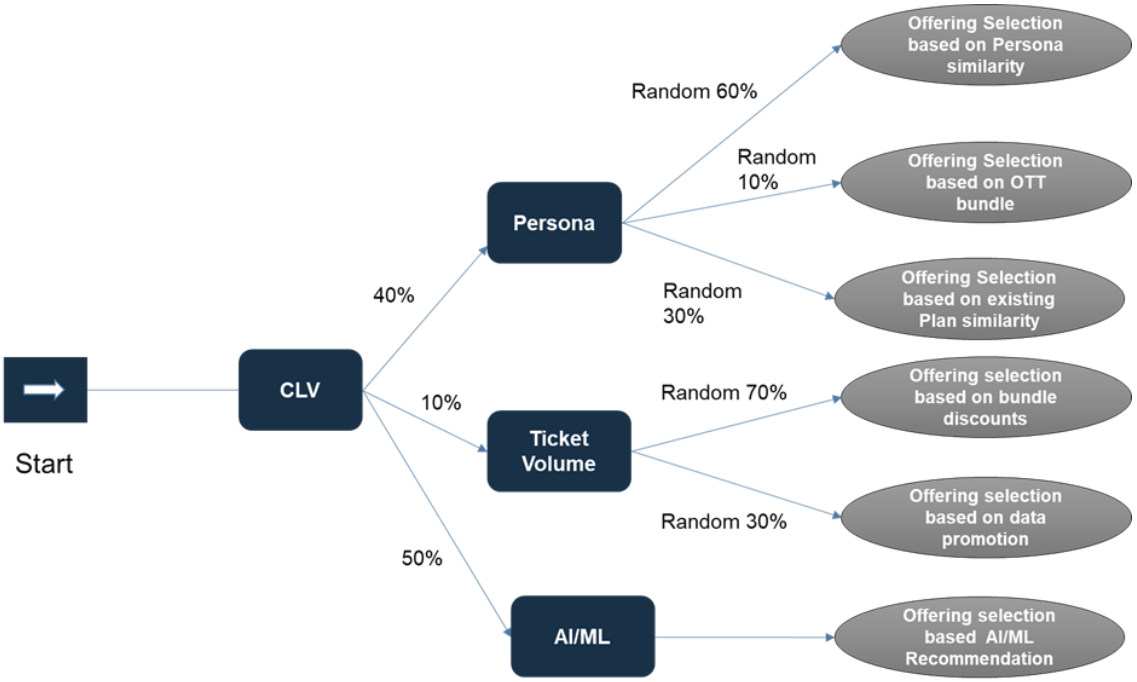

Figure 21. Revised Decision weights based on the statistical analysis of feedback collected

In the next phase, if the feedback is unfavorable, the decision model can be revised to incorporate additional branches with AI/ML models or rebalance the weight distribution across the sub-decisions.

The following figure puts the above example into the DT4DI context and illustrates how a feedback mechanism functions in the case of decision intelligence for NBX.

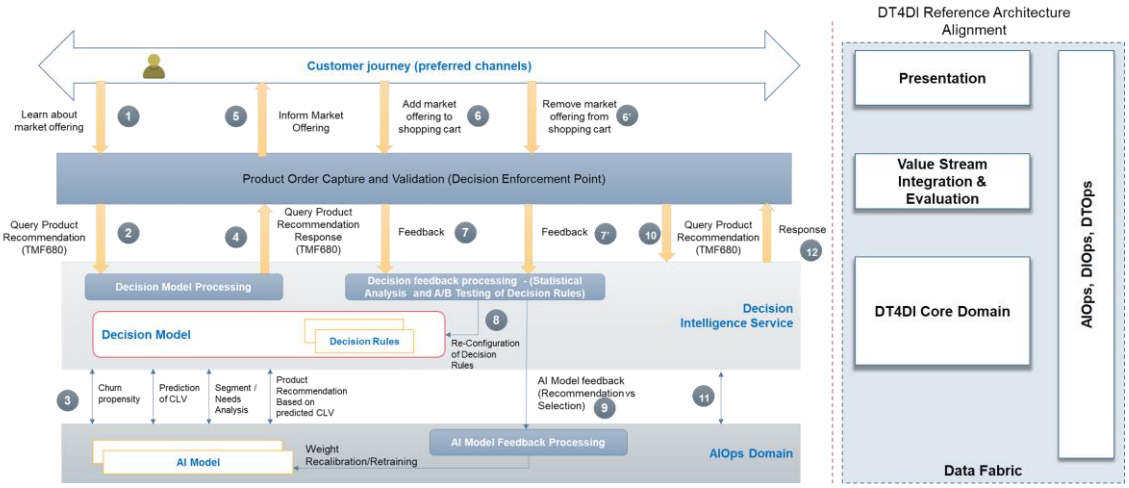

Figure 22. Continuous Optimization with Decision Intelligence for NBX Scenario

For the customer shopping journey depicted above, Product Order Capture and Validation are crucial to interfacing with the end customer through various preferred channels.

The Decision Intelligence Service implements the DT4DI reference architecture and supports POCV in preparing product offer recommendations following the NBX objectives of different stakeholders.

The decision Intelligence service uses the decision model prepared in consultation with business stakeholders to support the NBX decision requirements.

The decision model leverages decision rules and AI/ML Models to derive insights for decision-making.

Two capabilities of the Decision Intelligence Service are emphasized: decision model processing and decision feedback processing.

These two capabilities are shown separately because each requires different statistical analysis and validation mechanisms.

Following is a sequence of operations highlighted in the diagram.

1. Customers approach the CSP marketplace through their preferred channel based on a campaign, self-needs, or word of mouth. Product Order Capture and Validation receives customer intent to learn more about market offerings.
2. Through TMF680 Recommendation API, POCV places a request to receive a list of offering suiting the customer.
3. Decision Intelligence Service uses the internal capability for Decision model processing to identify the best offer that aligns with NBX objectives at this stage - influencing the progression of the customer lifetime value to predefined levels. For this, multiple AI/ML Models and Decision Rules may be leveraged depending on the overall structure of the decision model.
4. The Decision Intelligence service responds with the shortlisted offers matching the NBX objectives.
5. The offers are presented to end users through their preferred channels.
6. The end user selects preferred product offers and adds/removes/updates the shopping cart from the list of offers presented.
7. The end user's selection is consolidated as a user's preference and sent to the Decision Intelligence Service as feedback on the NBX decision.
8. The Decision Intelligence layer accumulates samples through feedback collected from different end users. Once sufficient samples are available, the Decision Intelligence layer evaluates the most and least preferred recommendations based on statistical analysis. Such evaluations, called A/B tests, help identify the weightage of future recommendations. Based on this, Decision rules as part of the Decision Model are reconfigured with appropriate weight for specific NBX criteria – that means for future Decision requests, the reconfigured weights and constraints are used to determine the best actions that lead to expected NBX outcome. In the NBX scenario, a group of customers may be offered variant X of the product offering. In contrast, another group of customers may be offered variant Y depending on their NBX evaluation. When sufficient samples of such customers have been collected, an A/B test determines if the offering variant X is found to be more popular among customers and leads to positive NBX metrics (e.g., 60% more customer offer acceptance and improved customer lifetime value for variant X compared to

Variant Y). The Decision rules are then configured to give a higher weightage for decision actions that select offering X for similar customers.

9. The reconfiguration of the NBX Decision rules may not be sufficient as this may lead to inconsistent behavior at two intelligence planes – i.e., at the AIOps and Decision planes. To avoid this inconsistency, the feedback from end users on the recommendation is forwarded to the AIOps plane. Here the feedback is deconstructed, and discrete inferences, which led to the decision, and associated feedback is determined. It is to be noted that the incorporation of the explainable AI principles is essential here to determine the reason for particular inferences made by specific AI Models. Following the evaluation of the explainability of the AI Model, it is recalibrated or retrained to ensure that the results of the subsequent requests are in line with the end user's preferences.

In the NBX example highlighted in #9 above, if the product offering variant X has the highest weightage based on the statistical analysis, the decision components that resulted in the selection of variant X are analyzed.

This includes analysis of individual AI/ML Model inferences with the set of inputs, hyperparameter values, and inference results.

It is recommended to incorporate appropriate Explainable AI (XAI) techniques to identify the reasoning for producing particular inference outcomes such as - feature importance (a feature that strongly influences the choice of offer recommendations), counterfactual explanation (possible outcome in case a different feature value had been chosen), LIME (Local Interpretable Model agnostic Explanation), SHAP (Shapley Additive Explanations) used to generate explanations for individual predictions, or to create global explanations that show how different features contribute to the overall prediction) etc.

For example, the product recommendation AI model XAI technique can highlight which feature (Predicted CLV, churn propensity, engagement metrics, etc.) influenced the selection of a particular choice of offers.

If the final selection made by the end customer does not match the recommendations, weights can be recalibrated, and models can be retrained to match closely with the real-life scenario.

6.4. DT Modeling - Marketing Operations

Decision Intelligence (DI) requires deep domain expertise and tremendous data and insights exploration capabilities in real or near-real time.

To tackle this challenge, DT4DI combines modeling techniques based on Digital Twin (DT), knowledge graphs, and AI/ML to engineer decision-making and integrate DI into the business process (BP), as shown in Figure 12 above.

DT and knowledge graphs-based modeling techniques enable to:

1. Explore the explicit and implicit characteristics of objects of interest and relationships and interactions among them.
2. Identify and explain the most influential factors to understand the behavior pattern of the object or predict the trajectory or trends of metrics.

In marketing operations, one of the crucial decision-making processes relates to the go-to-market (GTM) of products and services, which may be daily-based and impose significant challenges on data engineering, exploration, analysis, and decisions.

Adopting digital twin modeling techniques in this domain brings new and incredible opportunities.

The digital twin models in the marketing domain include the following aspects (we can call them 5W & 2H):

- Who: indicates the CSP's service object, for example, users or user groups.
- What: content, products, and services directly or indirectly accessed through telco services, such as apps, packages, marketing activities, consultation, and complaints.
- What: devices used to access products or services through CSP services.
- When: the time a user uses a product or service, which can be divided by hour, day, or month based on the data granularity in the OSS and BSS domains.
- Where: indicates the place or location where a user uses a product or service, for example, the base station or cell where the user is located.
- How: indicates how users use products or services and how they experience them.
- How much: indicates the usage of a product or service, for example, the total traffic used by an app, the total duration used by an app, and the number of package payments.

The figure below shows the primary influential objects digital twins need to model.

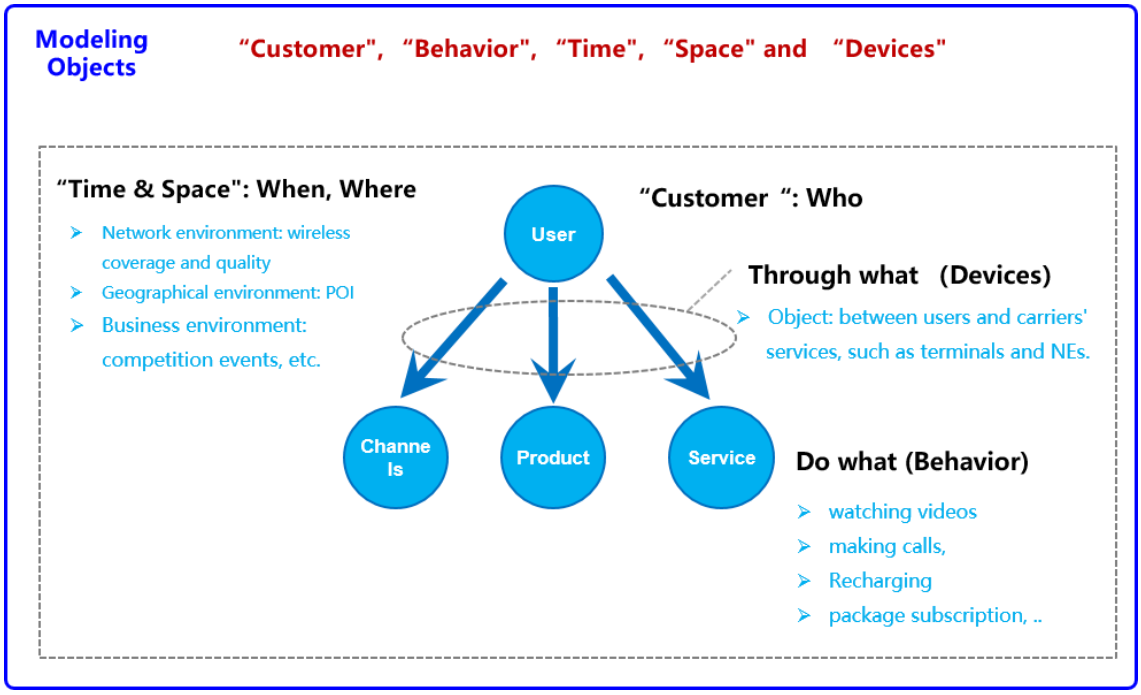

Figure 23. Objects to model in a Marketing Domain Digital Twin.

The figure below shows the DT Modeling steps mapping with the DT4DI Layered Architecture of Figure 3.

- Step 1: data are collected from multiple sources and cleansed, then aggregated in different periods, such as hourly, daily, weekly, and monthly, according to the context.
- Step 2: it performs data statistics, feature calculation, and feature mining based on ontological table fields.
- Step 3: Digital Twin Graph is created by defining entities and determining the relationships and attributes between entities, enriching entities, relationships, and attributes by graph calculation and mining, which is an iterative process, as Figure 21 depicts

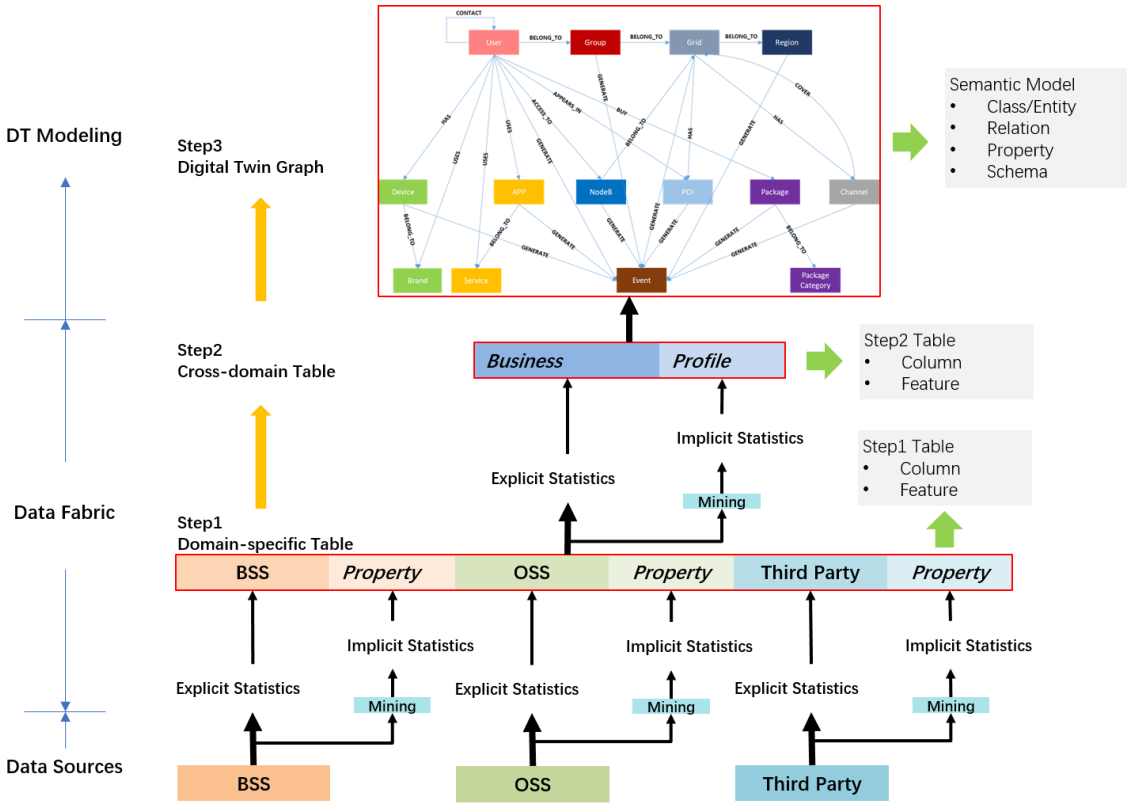

Figure 24. DT Modeling Steps

The figure below shows the corresponding DT Modeling implementation pipeline.

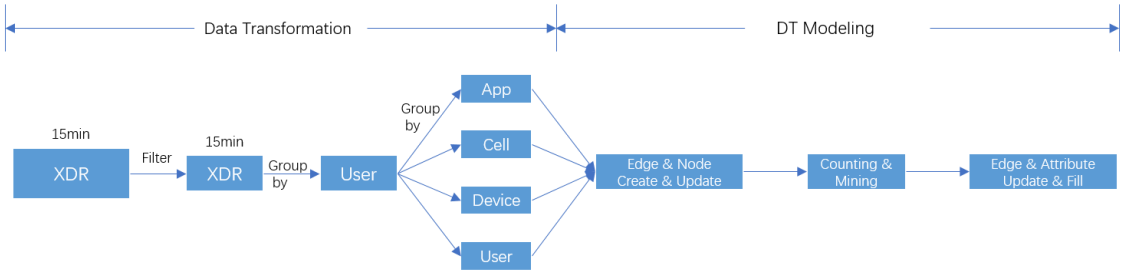

Figure 25. Implementation Pipeline of DT Modeling

7. DIOps, DTOps and AIOps

The DT4DI Reference Architecture designs the architecture of the DT4DI domain and is a reference for DT4DI solutions.

It shows how to combine and interlock software and data components, their capabilities, data models, interfaces, and how they interact.

Domain and solution architectures are fundamental and necessary inputs for the software lifecycle's development, testing, and deployment stages.

However, even the best architecture would only work if it is backed by well-defined and engineered processes supporting the lifecycle of critical architectural elements.

In DT4DI, the vital elements and enablers are:

- Decision Intelligence (DI) is the decision-making and management discipline.
- AI/ML is the enabling technology for Descriptive, Diagnostic, Predictive, and Prescriptive analytics.
- Digital Twin (DT) enables virtual representations of customers/personas, objects, systems, and processes and simulates their behavior.

These disciplines require comprehensive frameworks to handle DI systems and processes based on AI and DT across the lifecycle of their operations, scale them, manage those systems' development, deployment, production, and maintenance, and integrate those DI processes into existing or new business and decision-making processes.

This will include managing tens or hundreds of AI-based DT4DI component instances that may be deployed and run simultaneously in Production environments.

For these reasons, we include in the DT4DI Reference Architecture the following frameworks (shown as vertical blocks of the DT4DI Layered Architecture in Figure 3):

- AIOps (AI Operations) Framework [4]: the TM Forum's set of principles, practices, and processes to design, develop, deploy, execute, and maintain AI components effectively and safely.
- DTOps (Digital Twin Operations): A methodology to design, deploy, synchronize, and maintain Digital Twin models.
- DIOps (Decision Intelligence Operations): A methodology to design and manage E2E Decision Intelligence processes and their lifecycle.

For the AI software, we will refer to and use the available (and continuously evolving) TM Forum AIOps framework [4].

About DTOps, we may refer to other frameworks or develop/tailor one.

DIOps is in the roadmap of the TM Forum DT4DI project.

8. Conclusion

The DT4DI Reference Architecture is a domain architecture framing the Decision Intelligence (DI) and the Business Intelligence & Analytics (BI&A) application domains.

DT4DI combines Decision Intelligence (DI), Business Process Management, Analytics, AI, AIOps, Digital Twin (DT), and Data Fabric to revolutionize the management of challenging decision-making in complex business processes, improve business outcomes, and create data-driven and results-oriented organizations.

The DT4DI Architecture primarily supports business operations use cases, including marketing & sales operations, customer management, experience management, product design, recommendations, revenue management, strategy simulation, etc.

However, this approach can be extended to any complex business process involving challenging decisions, particularly technical operations (Network, IT) and environmental sustainability.

This document describes level-0 and level-1 of the DT4DI Reference Architecture, showing the fundamental building blocks of DT4DI solutions with different levels of detail.

The design, implementation, and proper adoption of DT4DI architectures and solutions will enable organizations to attain knowledge-driven, decision-centric, and result-oriented operations.

"A building is not just a place to be, but a way to be"

Frank Lloyd Wright

9. References

[0] IG1307 - Digital Twin for Decision Intelligence (DT4DI) - Whitepaper - TM Forum - Dec 2022 - https://www.tmforum.org/resources/how-to-guide/ig1307-digital-twin-for-decision-intelligence-dt4di-whitepaper-v1-0-0/

[1] Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) - Robert Martin - https://www.amazon.com/Clean-Architecture-Craftsmans-Software-Structure/dp/0134494164

[2] The Clean Code Blog by Robert C. Martin (Uncle Bob) - Aug 2012 - https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

[3] IG1310 - DT4DI & AIOps Ontology TM Forum - Feb 2023 - https://www.tmforum.org/resources/standard/ig1310-dt4di-aiops-ontology-v1-0-0/

[4] AI Operations (AIOps) Toolkit - https://www.tmforum.org/resources/toolkit/ai-operations-toolkit

[5] ChatGPT - https://chat.openai.com/chat

[6] Evolution and Trends of Business Intelligence Systems: A Systematic Mapping Study - Master's Thesis - Pekka Marjamäki - University of Oulu Information Processing Sciences - Apr 2017

[7] Vargas, J. Business Intelligence. Preprints 2021, 2021030579 (doi: 10.20944/preprints202103.0579.v1).

[8] An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management - SRM Institute of Science and Technology, Chennai - https://www.mdpi.com/2504-2289/6/4/132

[9] What is service-oriented architecture (SOA)? - Red Hat - Jul 2020 - https://www.redhat.com/en/topics/cloud-native-apps/what-is-service-oriented-architecture

[10] Component Based Architecture - Omar Elgabry - Apr 2019 - https://medium.com/omarelgabrys-blog/component-based-architecture-3c3c23c7e348#:~:text=In%20a%20component%2Dbased%20architecture,is%20rather%20an%20implementation%20detail.

[11] Layered Architecture - H Graca - Aug 2017 - https://herbertograca.com/2017/08/03/layered-architecture/

[12] Feature oriented architecture for web applications - NAte Wang - Jul 2017 - https://medium.com/react-weekly/feature-oriented-architecture-for-web-applications-2b48e358afb0

[13] The TOGAF® Standard, 10th Edition - https://www.opengroup.org/togaf

[14] Pattern: Microservice Architecture - Chris Richardson - https://microservices.io/patterns/microservices.html

[15] What is virtualization - https://aws.amazon.com/what-is/virtualization/

[16] Cloud Computing Basics https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angriffszielen/Cloud-Computing/Grundlagen/grundlagen_node.html

[17] Event-Driven Architecture - https://ebrary.net/85093/computer_science/event_driven_architecture

[18] The Engineers Guide to Event-Driven Architectures: Benefits and Challenges - Oscar uit de Bos - Nov 2020 - https://medium.com/swlh/the-engineers-guide-to-event-driven-architectures-benefits-and-challenges-3e96ded8568b

[19] EDGE computing https://www.ericsson.com/en/edge-computing

[20] What is Data Fabric? - Talend - https://www.talend.com/resources/what-is-data-fabric/

[21] Data Fabric Architecture - Gartner - Mar 2021 - https://www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration

[22] DATA MESH ARCHITECTURE - https://www.datamesh-architecture.com/

[23] How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh - Zhamak Dehghani - May 2019 - https://martinfowler.com/articles/data-monolith-to-mesh.html

[24] GraphQL - https://graphql.org/

[25] In-Memory Data Grid - Apache Ignit - https://ignite.apache.org/use-cases/in-memory-data-grid.html

[26] Use Cases and Architectures for Apache Kafka across Industries - Kai Waehner - Oct 2020 - https://www.kai-waehner.de/blog/2020/10/20/apache-kafka-event-streaming-use-cases-architectures-examples-real-world-across-industries/

[27] Decision Modeling Notation Specification - Object Management Group - https://www.omg.org/dmn/

[28] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

[29] A systematic review: machine learning based recommendation systems for e-learning - Shristi Shakya Khanal, P.W.C. Prasad, Abeer Alsadoon, Angelika Maag - December 2019 - https://researchoutput.csu.edu.au/ws/portalfiles/portal/56945122/37059927_Published_article.pdf

[30] Digital Twin Definition Language - https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v3/DTDL.v3.md

[31] Market Guide for Service Orchestration and Automation Platforms - Gartner - Aug 2021 - https://www.gartner.com/en/documents/4004556

[32] GB1022 ODA Functional Architecture Guidebook - TM Forum - https://www.tmforum.org/resources/reference/gb1022-oda-functional-architecture-guidebook-v1-1-0/

[33] Catalyst Project C22.0.384 - Outcome-based decision intelligence for business growth - DTW 2022 - https://www.tmforum.org/labs/

[34] Building Industrial Digital Twins - Shiam Varan Nath, Pieter van Schalkwyk - 2021 - Packt Publishing

[35] Groshev, M., Guimaraes, C., Martin-Perez, J. & de la Oliva, A. (2021). Toward Intelligent Cyber-Physical Systems: Digital Twin Meets Artificial Intelligence. IEEE Communications Magazine, 59(8), 14–20. https://e-archivo.uc3m.es/bitstream/handle/10016/33912/Towards_CM_2021_ps.pdf?sequence=1

[36] Link, How Decision Intelligence Connects Data, Actions, and Outcomes for a Better World - Lorien Pratt - 2019 - Emerald Publishing

[37] How knowledge graphs create data-driven cultures - Brett Hurt - CIO - Apr 2019 - https://www.cio.com/article/219971/how-knowledge-graphs-create-data-driven-cultures.html

[38] Design, Modeling and Implementation of Digital Twins - Mariana Segovia and Joaquin Garcia-Alfaro - Sensors 2022, 22, 5396. https://doi.org/10.3390/s22145396

[39] Advancing digital twin implementation: a toolbox for modelling and simulation - Soumya Singha, Max Weebera, Kai-Peter Birke - 2021 - Advancing digital twin implementation: a toolbox for modelling and simulation - ScienceDirect

[40] Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model - Aheleroff S., Xu X., Zhong R.Y., Lu Y - Adv. Eng. Inform. Jan 2021 - Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model - ScienceDirect

[41] 10 Software Architecture Patterns in Enterprise Software Development - Jin - Medium - Nov 2021 - https://blog.devgenius.io/10-software-architecture-patterns-in-enterprise-software-development-fabacb5ed0c8

[42] What are Design Systems? How are they different from Style Guides and Pattern Libraries? - Preran - Adobe Support Community - Dec 2019 - https://community.adobe.com/t5/adobe-xd-discussions/what-are-design-systems-how-are-they-different-from-style-guides-and-pattern-libraries/m-p/10800661

[43] Serverless Architecture Layers - Serverless Advocate - Apr 2022 - Level Up Coding - https://levelup.gitconnected.com/serverless-architecture-layers-a9dc50e9b342

[44] Domain-Driven Design: Tackling Complexity in the Heart of Software - Eric Evans - Sep 2003 - Addison-Wesley Professional

[45] How to Integrate Process Models with Decision Intelligence - William O'Shea · May 2022 - https://www.lorienpratt.com/guest-post-how-to-integrate-process-models-with-decision-intelligence/

[46] SOA Reference Architecture - The Open Group - https://www.opengroup.org/soa/source-book/soa_refarch/index.htm

[47] How to easily build your own domain specific knowledge graph - Henri Egle Sorotos - Medium - Dec 2022 - https://henri-egle-sorotos.medium.com/how-to-easily-build-your-own-domain-specific-knowledge-graph-aeb70735631e

[48] Why your data catalog needs to be powered by a knowledge graph - Juan Sequeda - data.world - Feb 2022 - https://data.world/blog/3-ways-to-confirm-your-data-catalog-is-really-powered-by-a-knowledge-graph/

[49] Decision making - Eisenfuhr, F. - 2011 - Springer.

[50] IG1203 Capabilities for CEM Concepts and Processes v3

[51] IG1303 Customer Experience KPI Metrics

[52] A Framework for Embedding Decision Intelligence into your Organization - Erik Balodis - May 2022 - https://towardsdatascience.com/a-framework-for-embedding-decision-intelligence-into-your-organization-f104947651ae

[53] Quantellia: Futures Powered By Intelligence - https://quantellia.com/

[54] Introduction to Decision Intelligence - Cassie Kozyrkov - Medium - Aug 2019 - https://towardsdatascience.com/introduction-to-decision-intelligence-5d147ddab767

[55] The Onion Architecture: part 1 - Jeffrey Palermo - Jul 2008 - https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/

[56] IG1277 Business Architecture: Capability Driven Transformation - TM Forum - https://www.tmforum.org/resources/standard/ig1277-business-architecture-capability-driven-transformation-v3-1-0/

10.Administrative Appendix

10.1. Document History

10.1.1. Version History

Version Number	Date Modified	Modified by:	Description of changes
1.0.0		Alan Pope	Final edits prior to publication
2.0.0	17-Jun-2023	Stuart Dunn	Final edits prior to publication
2.1.0	18-Aug-2023	Stuart Dunn	Final edits prior to publication

10.1.2. Release History

Release Status	Date Modified	Modified by:	Description of changes
Pre-production		Alan Pope	Initial Release
Pre Production	17-Jun-2023	Stuart Dunn	v2.0.0 Release
Pre Production	18-Aug-2023	Stuart Dunn	v2.1.0 Release

10.2. Acknowledgements

The members of the TM Forum DT4DI project listed below prepared this document:

Member	Company	Role
Luca Franco Varvello	Huawei	Project Chair
Luca Franco Varvello	Huawei	Author
Simin Huang	Huawei	Editor
Sreeja Renganath	TCS	Editor
Lekshmy Sasidharan	TCS	Editor
Zhang Jian	Huawei	Editor
Pengbinqi	Huawei	Editor
Shaoyi Liang	Huawei	Editor
Mehmet Beyaz	TTG Int.	Editor
Meng Xu	China Mobile IT	Editor
Rui Hu	China Mobile IT	Editor
Xin Chen	China Mobile Guangdong	Editor
Manoj K Nair	Netcracker	Editor
Arnold Buddenberg	Orange	Editor

Member	Company	Role
Sarah Haojing	Huawei	Key Contributor
Feng Chen	Huawei	Key Contributor
Christian Maitre	Orange	Key Contributor
Tao Tao	China Mobile IT	Key Contributor
Xin Jing	China Mobile IT	Key Contributor
Rakesh Ashok	NBN Australia	Key Contributor
Jiyong Sun	China Mobile Guangdong	Key Contributor
Derek Chen	HKT	Key Contributor
Delia Deng	Huawei	Key Contributor
Mohammed Alzaaidi	STC	Key Contributor
Lei Tian	China Mobile Tianjin	Key Contributor
Tao Zhang	China Mobile Tianjin	Key Contributor
Lingyan Zhang	China Mobile Guangxi	Key Contributor
Khalid Attia	stc	Key Contributor
GuoJian Wei	Huawei	Key Contributor
Jiamu Wang	China Mobile AnHui	Key Contributor
Wanghua Chen	China Mobile AnHui	Key Contributor
Fan Li	Huawei	Key Contributor
Sunny Sunchao	Huawei	Key Contributor
Pramath Jha	TCS	Key Contributor
Kevin McDonnell	Huawei	Key Contributor
Herve Guesdon	Ubiqube	Key Contributor