BIPER - *Business Informatics Programme Reengineering*

Erasmus+ Call KA226, Duration 18 months (2021.03.01-2022.08.31)

Project Reference No	2020-1-HU01-KA226-HE-093987
Nature	Working document
Dissemination Level	Restricted
Last update	26/04/22
Status	Work-in-progress
Editor(s)	A. Gábor, M. Arru, C. Csáki, Z. Szabó, I. Szabó
Document Description	IO2: Architecture delivery - Version 3

List Abbreviations

Term / Abbreviation	Definition
ADM	Architecture Development Method
AI	Artificial intelligence
ASP	Application Software Provider
BI	Business Intelligence
BIS	Business Information Systems
BPM	Business Process Modelling
BPMN	Business Process Modelling Notation
CRM	Customer Relationship Management
CS	Case Study
CSA	Common Systems Architecture
DAQ	Data AcQuisition
DAS	Data Acquisition Systems
DB	Database
DBMS	Database Management System
DFD	Data Flow Diagram
EA	Enterprise Architecture
EAP	Enterprise Architecture Planning
EAM	Enterprise Architecture Modelling or Management
ebXML	Electronic Business Extensible Markup Language
ESCO	European Skills, Competences, Qualifications and Occupations
ETL	Extraction Transformation (and) Loading
ERD	Entity Relationship Diagram
ERP	Enterprise Resource Planning (System)
ESB	Enterprise Service Bus (of SOA)
FEA	Federal Enterprise Architecture
FEARM	Federal Enterprise Architecture Reference Model
HIPO	Hierarchical Input Process Output
HTTP	Hyper Text Transfer Protocol
IaaS	Infrastructure as a Service (cloud)
ICT	Information and Communication Technologies
ICOM	Input Control Output Mechanism
IO	Intellectual Output
IS	Information Systems
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
OAG	Open Applications Group
OAGIS	Open Applications Group Integration Standard
OMG	Object Management Group
OO	Object Oriented
PaaS	Platform as a Service (cloud)
REA	Resource-Event-Agent
SaaS	Software as a Service (cloud)
SCM	Supply Chain Management
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language (for DB)

STEP	STandard for Exchange of Product model data
TMF	Tele Management Forum (telecommunications industry framework)
TOGAF	The Open Group Architecture Framework
T&L	Teaching & Learning
UML	Unified Modelling Language
XML	Extensible Markup Language

List of Definitions

Term	Definition
Goal	A goal is a short statement of the desired outcome to be accomplished over a long-time frame, usually three to five years. It is a broad statement that focuses on the desired results and does not describe the methods used to get the intended outcome.
Aim	What you hope to get and you want to do this.
Objective	Objectives are specific, actionable targets that need to be achieved within a smaller time frame, such as a year or less, to reach a particular goal. Objectives describe the actions or activities involved in achieving a goal
Target	The exact result of what you want to get.
Purpose	… of something is the reason for which it is made or done

Document History

Date	Name	Description
20.05.2021	András Gábor	Document initiation (originally with IO1, integrated)
25.05.2021	Matteo	Revision, amendment of (old) Chapter 4
06.06.2021.	András Gábor	Document restructuring, adding Chapter 2,3 & 6
07.07.2021	András Gábor	Adding Output subsections to each ADM step
09.07.2021.	Ildikó Szabó	Adding competences to each step
26.08.2021	András Gábor	Case study metamodel (Ch5)
30.09.2021	András Gábor	Mapping knowledge structure to edoer.eu
30.09.2021	Csaba Csáki	Working on Introduction and Ch. 2 (steps)
04.10.2021	Dóra Őri	Working out Business architecture components
06.10.2021	Katalin Ternai	Adding data architecture elements
11.10.2021	András Gábor	Redefining structure of the Chapters
14.10.2021	Gábor Kismihók	eDoer Chapter with tool description
29.10.2021	Matteo Arru	Case study enrichment
01.11.2021	Ildikó Szabó Borbásné	Checking curriculum matching
04.11.2021	Dóra Őri	Adding further content to Business architecture
09.11.2021	Keshav Dahal	Data elements of the case study steps
16.11.2021	Mirjana Kljajić Borštnar	Comparing need to know to Maribor curriculum
17.11.2021	Gábor Kismihók	Adding BIPER content to the eDoer Chapter
30.11.2021	Katalin Ternai	Refining data architecture elements
14.12.2021	Csaba Csáki	Reverting integrated document into IO1/IO2 – open v2
15.12.2021	Csaba Csáki	Added (teaching) methodology related sections
17.12.2021	All	Review (marking parts to be worked out under IO3)
19.01.2022	All	Quality Review - closing first version
18.03.2022	András Gábor	Restructuring based on interim review feedback
21.03.2022	Csaba Csáki	Added 'Source' information to all images
06.04.2022	Keshav Dahal	Proof-reading and corrections
07.04.2022	C.Csáki, Z.Szabó, A.Gábor	QA Review of IO2 version 2 – open v3
22.04.2022	Csaba Csáki	Small corrections and refinements from all partners
25.04.2022	Marco Gilardi	Small corrections
26.04.2022	Zoltán Szabó	Final touches

Table of Contents

List of figures

1 Purpose of this Document

The BIPER project aims to develop a blended-learning compatible BIS curriculum, based on shared values, e-Learning experiences and sound pedagogical principles.

The academic field of Business Information Systems is a complex area bridging business and organisational topics with questions of applied information technology. Teaching such a multidisciplinary domain which assumes not only knowledge of theoretical concepts and technical skills to use tools but also a problem centred mindset and related problem-solving abilities is a challenge in itself. Fostering communication and information sharing in teaching BIS online could further the teaching of this field at the third and fourth level in light of a potentially longer battle with social distancing rules. BIS education in a classroom context may be characterised by what the literature calls 'active learning', which refers to pedagogies that increase and enhance student interaction. This is required by the project-oriented, and teamwork-based reality of developing, implementing and managing IT/IS solutions in an organisational context.

The BIPER project aims at allowing consortiums partners to share their knowledge and expertise concerning the above situation and develop a shared pool of shared resources benefiting both its members as well as the wider BIS education community.

This document describes Intellectual Output 2 by developing the details of the curriculum based on the results of IO1 (see appropriate document separately). Using the Enterprise Architecture elements as described in IO1 (and based on TOGAF) the required curriculum content is described through four steps for each element assuming a case study approach.

2 BIS PROGRAM DEVELOPMENT: TEACHING PROGRAM AND MAPPING TO CURRICULUM

The TOGAF based general curriculum framework laid out in IO1 (see separately) proposed a recommendation of how to structure required BIS knowledge and skills along a three dimensional space of capabilities. This is described as the "fragmentation" model (demonstrated in Figure 5 of page 19 of the IO1 document). In this document the first four of the six TOGAF elements are discussed regarding how to use them for teaching BIS: Architecture Context, Business Architecture, Information Systems Architecture, and Technology Architecture. For each of this architecture aspects teaching content will be discussed in four steps. In addition, these steps are introduced along appropriate sub-areas of the corresponding architecture element as follows:

- Architecture Context (Chapter 3) includes four areas: Introduction And Overview, Architecture Framework, Architecture Vision, and Target Architecture;
- Business Architecture (Chapter 4) covers two parts: Business Architecture Design and Business Architecture Modelling;
- Information Systems Architecture (Chapter 5) is split into three areas: Information Systems Architecture, Data Architecture, and Application Architecture;
- Technology Architecture (Chapter 6) has no sub-areas.

Notice, that Implementation and Governance as special aspects will be covered under IO3 – as a result of WP3 and WP4 respectively.

For each of the above 11 areas explanation of teaching (content and work) is done assuming four steps. These steps are 1) Enterprise Architecture Development; 2) Case study steps and/or output; 3) Need to know to perform the step; and 4) What section(s) of the BIS curriculum are covered. The general structure can be worked out in the following table format:

Enterprise Architecture Development	Case study steps and/or output	Need to know to perform the step	What section(s) of the BIS curriculum are covered
The architecture-related knowledge elements	How these architecture elements are related to the case study (or example) and what results (output) is expected to be produced – this is aimed at helping the facilitator (teacher)	Skills and knowledge required to execute these steps under the architecture progress (six levels)	In the context of the EQF (European Qualification Framework) or its national equivalent, what learning outcomes match the "need to know" items of BIPER

Within each step the depth of skills and knowledge is spread along six levels (see Section 3.5 of IO1).

3 THE ARCHITECTURE CONTEXT – VISION AND TARGET

3.1 THE ARCHITECTURE: INTRODUCTION AND OVERVIEW

3.1.1 Enterprise Architecture Development

1. Principles and framework for architectural design
2. Architectural frameworks (Zachmann and TOGAF)
3. TOGAF 9 ADM cycle
4. Architectural approaches to ERP systems

3.1.2 Case study steps and/or output

None

3.1.3 Need to know to perform the step

In the introduction, it is helpful to explain the relationship between the real and the information sphere, what approaches have been taken (historical dimension: to understand continuity and radical or less radical renewal, to prepare students for the correct evaluation of recent and current trends). The original concept of architecture and its metamorphosis (how an originally architectural but overall technical term became enterprise architecture). The essence, advantages and disadvantages of an architectural approach (Zachman). Concept of the enterprise, extended enterprise.

Here one can bring in the concepts of the enterprise learned in Business Economics and even the newer formations (e.g. platform) and interpret them in the context of EA.

3.1.4 What section(s) of the BIS curriculum are covered

Understand the principles of problem formulation and documenting context as well as the role of architecture within IT-based business enterprise, especially in the context of systems development or re-alignment. Convey and demonstrate understanding of the key models of the organization and operation of computer system organization and architecture.

3.2 ARCHITECTURE FRAMEWORK

3.2.1 Enterprise Architecture Development

1. Enterprise continuum
2. Tools for structuring content
3. The architecture of the repository
4. Architecture governance
5. Architecture framework and customisation
6. Granularity
7. Time horizon and time preference

3.2.2 Case study steps and/or output

Preparation and context description. Business Case presentation.

3.2.3 Need to know to perform the step

This block introduces students to the concept and toolset of EA, focusing on understanding the architecture repository because this is where the often-mentioned development requirements make sense, like reusability, sustainability, portability, interoperability, standardisation, etc. Of course, the other points are also important. By illustrating with concrete examples, it should be easy to understand. Still, the issues raised here should be discussed in detail later (e.g. when discussing the different architecture domains) because then the points made here will make sense.

3.2.4 What section(s) of the BIS curriculum are covered

Students will be competent in identifying problems in the BIS field and collecting, managing resources while solving them. Student will be familiar with the Zachmann framework and The Open Group Architecture Framework. They will be to utilize these frameworks to map problems to architectural concepts along these approaches.

3.3 ARCHITECTURE VISION

3.3.1 Enterprise Architecture Development

1. To define and organise an **architecture development cycle** within the overall context of the architecture framework, as established in the Preliminary phase
2. To validate the organisation's business principles, goals, and strategic business drivers and the enterprise architecture, **Key Performance Indicators** (KPIs) and **Objectives and Key Results** (OKR).
3. To define the **scope** of, and to **identify** and **prioritise** the components of the Baseline Architecture effort
4. To define the relevant **stakeholders**, and their concerns and objectives
5. To define the **key business requirements** to be addressed in this architecture effort and the constraints that must be dealt with.
6. To articulate an Architecture Vision and formalise the **value proposition** that demonstrates a response to those and constraints
7. To create a comprehensive plan that addresses **scheduling, resourcing, financing, communication, risks, constraints, assumptions, and dependencies**, in line with the project management frameworks adopted by the enterprise (including Scrum, PRINCE2 or PMBOK)
8. To understand the **impact** on, and of, other enterprise architecture development cycles ongoing in parallel
9. *Optional: To ensure ADM proper **recognition** and **endorsement** from the corporate management of the enterprise, and the **support** and **commitment** of the necessary line management*
10. *Optional: To secure **formal approval** to proceed*

3.3.2 Case study steps and/or output

Introducing the context of Case Study (Business Case), mainly emphasising the problems, context, history (rather in a storyteller mode). An example is illustrated below.

Urban development
The city council explored several options during the concept development and began to take the next idea seriously. The city would entrust rehabilitation to real estate developers operating under market conditions, selected according to competitive market rules (public procurement). The city management decided on the concept and started preparing the public procurement for real estate development companies. The requirements are summarised in a technical Annex. However, during the preparation of the technical annex, a background service company would be set up to coordinate the work of the selected companies and better articulate the aspects of city management. This service company will primarily help real estate developers through IT services.
Aspects to consider or can be considered
1) Who are the stakeholders?   a. What are the interests of each stakeholder group?   b. How conflicting are the particular interests?   c. What conflict management methods have reality? 2) What is the time horizon of the entire program? 3) How can development be phased or divided into smaller, feasible phases? 4) What does the extended enterprise model look like? 5) Target architecture   a. the means of communicating it,   b. the means and methods of acceptance, support and commitment with the hope of success 6) Architecture   a. designation of a development cycle,   b. what could be the key performance indicators 7) Scope   a. definition, in the context of the baseline architecture   b. prioritisation of each component 8) Architecture framework and business requirements   a. ensuring alignment,   b. a formal definition of value propagation 9) Granularity   a. provision of resources, scheduling,   b. risk management and dependency management What is the impact of the architecture (on what, and on what time horizon)?

The expected output is:

- Approved Statement of Architecture Work, including in particular:
    - Scope and constraints
    - Plan for the architectural work

    - Roles and responsibilities
    - Risks and mitigating activity
    - Work product performance assessments
    - Business case and KPI metrics
- Refined business principles, business goals, and business drivers
- Architecture principles
- Capability Assessment
- Tailored Architecture Framework (for the engagement), including:
    - Tailored architecture method
    - Tailored architecture content (deliverables and artefacts)
    - Configured and deployed tools.

3.3.3 Need to know to perform the step

They will be identifying business requirements as business scenarios: new business capabilities, architecture requirements. Use iteratively at different levels of detail. Input: scope, maturity, roles, responsibilities, constraints, reuse, budgetary, request for change requirements, governance and support strategy, populated architecture repository. Key stakeholders' engagement: vision, requirements, scope boundaries, concerns, culture. Business and technology capability assessment, readiness for change. Scope definition (breadth of coverage, level of details, period. High-level view (from business scenarios through business capabilities to architecture vision) from business, information system and technology viewpoints. Value proposition and KPIs: performance metrics, risk assessment. Approval and project planning.

3.3.4 What section(s) of the BIS curriculum are covered

Students will collaborate with business and IT partners to prepare solution versions of economic problems using the most efficient IT solutions to initiate and implement IT support and development.

Key concepts to know: Business case development and Project Management basics, feasibility and feasibility study, elementary organizational roles, architecture fundamentals, Stakeholder analysis or Soft Systems Methodology. Understand the need to use project management software in the planning, allocation and monitoring of resources.

Key skills to master: Team work, sharing opinion, argumentation.

3.4 TARGET ARCHITECTURE

3.4.1 Enterprise Architecture Development

1. The concept of the target architecture, its role
2. Awareness, acceptance, support and commitment of the target architecture
3. Define the architecture development cycle, define key performance indicators
4. Scope definition, prioritisation of components in the context of the baseline architecture
5. Ensure alignment of architecture framework with business requirements, formal definition of value generation
6. Granularity, resource provisioning, scheduling, risk management and dependency management

7. Impact and formal validation

3.4.2 Case study steps and/or output

Collect main business requirements, set project context, prepare first version of KPI.

The expected output is: main business requirements and corresponding KPIs.

3.4.3 Need to know to perform the step

Working with baseline and target architecture corresponds to the traditional system design As-Is and To-Be approach. The target architecture is in harmony with the formulation of strategic goals, as it describes where we want to go. In fact, the whole strategic cycle and all its components can be connected in this step (scope, business-IT requirements alignment, indicators, priorities, resource management, impact). This is also where *risk management* can be meaningfully introduced.

The target architecture is an implementation of the architecture vision, previously also called an ideal state. It can be derived from the strategy (a *top-down approach* that allows us to "dream big" by using technology foresight). Still, it is equivalent to a *bottom-up approach*: only a very sketchy vision is created, and the target state becomes more refined and detailed as the ADM steps are gradually implemented. This second approach has a danger: if we gradually refine the target architecture, including more and more detail, we will inadvertently overestimate the increasingly clear organisational-technological obstacles, and instead of a visionary target state, we will end up with "much ado about nothing". Ad absurdum, we end up with a worse solution than the one we currently have.

(This paradox may arise because, in the world of information systems (also), two structures coexist: formal and informal. The formal structure contains the processes that can be well - explicitly - described by rules. In contrast, the informal structure contains the rules of conduct, the procedural rules that can be derived from the organisational culture, including the handling of exceptions. Exploring these informal rules and attitudes is the terrain of *knowledge management* since it is about organisational culture, knowledge conversion, and trust, but different maturity models also lead here. In any case, it is an interesting topic.)

3.4.4 What section(s) of the BIS curriculum are covered

Students will be able to identify the operating conditions of applications under real business and organisational conditions. Display an appreciation for and understanding of the typical issues associated with managing technology in a modern business. They will be able to consider and communicate benefits, threats and risks with the help of knowing IT procedures and methods. Students will be able to explain why businesses adopt and make use of new approaches and new technologies. They will be able to see an integrated view IT and business in a corporate context.

4 BUSINESS ARCHITECTURE

4.1 BUSINESS ARCHITECTURE DESIGN

4.1.1 Enterprise Architecture Development

1. The concept and role of business architecture
2. Formulating the objectives of business architecture
    a. develop a product and/or service strategy
    b. organisational, functional processes related to information flow and processing, geo-location aspects
    c. identification of the business environment
    d. identification and definition of business principles, objectives and strategic drivers
3. Identify and analyse the differences between the baseline and target architecture
    a. a combination of top-down and bottom-up approaches
    b. information gathering, processing and validation or rejection of preliminary hypotheses
4. Selection of appropriate viewpoints, with particular attention to stakeholders' interests and motivations
    a. ensuring alignment of the architecture framework with business requirements, formal definition of value creation
    b. analyse risks, define residual risk
5. Defining the architecture development cycle, defining key performance indicators
6. Selection of the appropriate development toolkit and methodology in line with the perspectives
7. Scope definition, prioritisation of components in the context of the initial architecture
8. Granularity, resource provisioning, scheduling, risk management and dependency management
9. Impact and formal validation

4.1.2 Case study steps and/or output

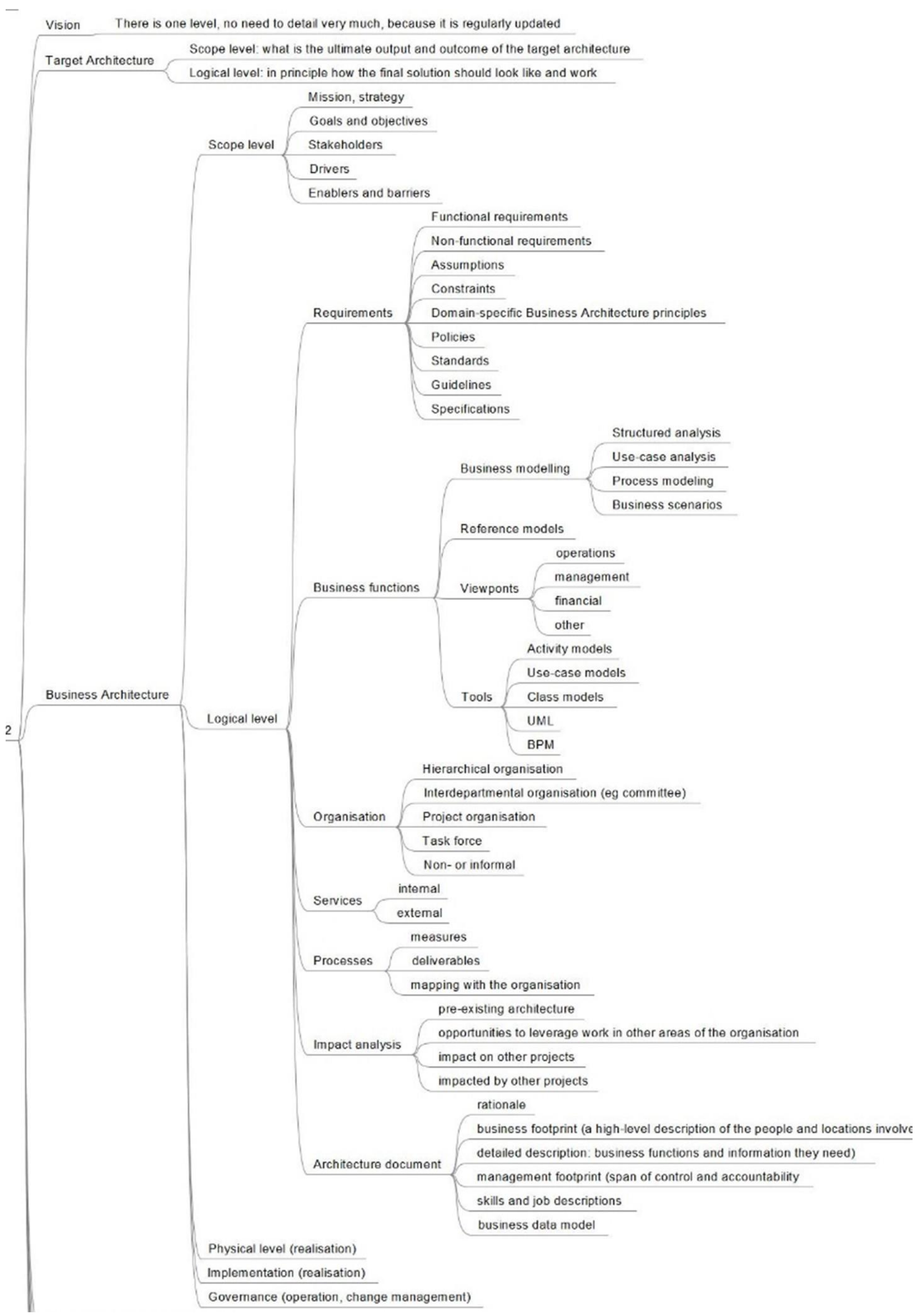
The expected output is:

- Draft Architecture Requirements Specification, including Business Architecture requirements:
    - Gap analysis results
    - Technical requirements — identifying, categorizing, and prioritizing the implications for work in the remaining architecture domains; for example, dependency/priority matrix (trade-off between speed of transaction processing and security); specific models that are expected to be produced (e.g. Zachman Framework)
    - Updated business requirements
- Business Architecture components of an Architecture Roadmap

4.1.3 Need to know to perform the step

Probably this is the most interesting part, where pure business logic meets IT opportunities. At this point, students need to be able to "dream" what they want the business to do or what business they want to go forward with (see, e.g. digital transformation). Scoping, which has already been emphasised in the previous strategy, can now be reinterpreted one level down and is worth emphasising. Depending on the case study(s) to be presented at the end of this stage, a business area can be highlighted (supply chain, production management, etc.). This is the "*differentia specifica"* of business informatics. It is also essential to highlight the perspectives (i.e. the different perspectives of people in different positions, with different qualifications, with different interests). Granularity and priorities can be raised here but will be discussed later.

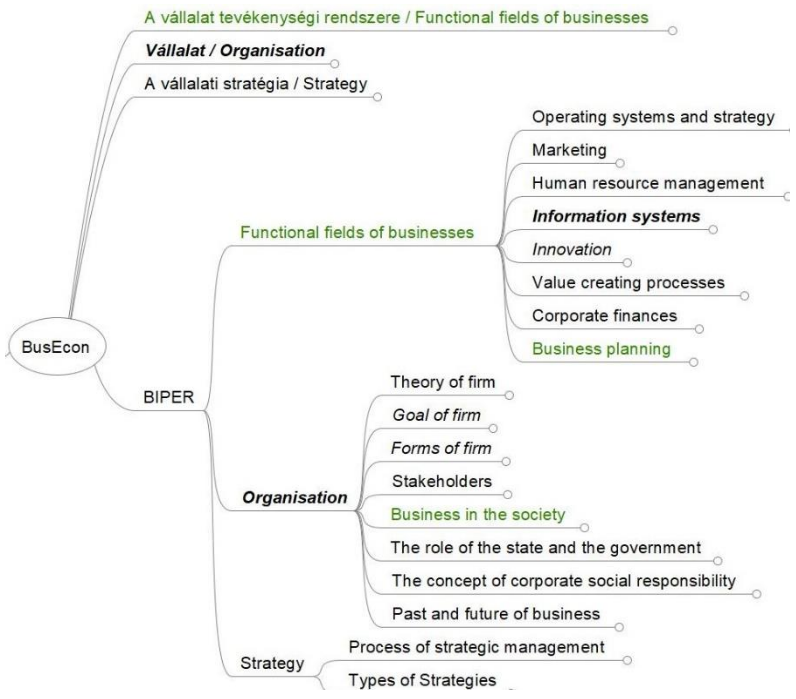
The case study that runs through the course should be introduced in more detail. It will be carried through evolutionarily, constantly enriched with new elements. For now, only the formulation and articulation of the business case can be practised, but students can continue with the modelling later. The first formulation of the target architecture is also a good choice at this point.



Vision
There is one level, no need to detail very much, because it is regularly updated
Target Architecture
Scope level: what is the ultimate output and outcome of the target architecture
Logical level: in principle how the final solution should look like and work
Mission, strategy
Goals and objectives
Scope level
Stakeholders
Drivers
Enablers and barriers
Functional requirements
Non-functional requirements
Assumptions
Constraints
Requirements
Domain-specific Business Architecture principles
Policies
Standards
Guidelines
Specifications
Structured analysis
Use-case analysis
Business modelling
Process modeling
Business scenarios
Reference models
operations
management
Business functions
Viewponts
financial
other
Activity models
Use-case models
Business Architecture
Tools
Class models
2
Logical level
UML
BPM
Hierarchical organisation
Interdepartmental organisation (eg committee)
Organisation
Project organisation
Task force
Non- or informal
internal
Services
external
measures
Processes
deliverables
mapping with the organisation
pre-existing architecture
opportunities to leverage work in other areas of the organisation
Impact analysis
impact on other projects
impacted by other projects
rationale
business footprint (a high-level description of the people and locations involve
detailed description: business functions and information they need)
Architecture document
management footprint (span of control and accountability
skills and job descriptions
business data model
Physical level (realisation)
Implementation (realisation)
Governance (operation, change management)

1. Figure Business architecture (source: Authors)

4.1.4 What section(s) of the BIS curriculum are covered



BusEcon
A vállalat tevékenységi rendszere / Functional fields of businesses
Vállalat / Organisation
A vállalati stratégia / Strategy
BIPER
Functional fields of businesses
Operating systems and strategy
Marketing
Human resource management
Information systems
Innovation
Value creating processes
Corporate finances
Business planning
Organisation
Theory of firm
Goal of firm
Forms of firm
Stakeholders
Business in the society
The role of the state and the government
The concept of corporate social responsibility
Past and future of business
Strategy
Process of strategic management
Types of Strategies

2. Figure Business Economics (source: Authors)

Students will be able explain the key concepts behind business enterprises. They will be able to analyse organisational cultures and structures with particular relevance to IS-induced changes. Understand the role of business analysis within IT-based business enterprise systems development or re-alignment. L1. Develop strategies for the planning, development and implementation of IT-based systems aligned with the business strategy. They will be able to exhibit the ability to critically evaluate the role of IT in a concrete setting. They will be able to apply their knowledge of operation management, the fundamentals and activities of the value chain and supply chain, the principles of process-oriented company operations, the functional organisation of the company, the concept and components of corporate strategy. Be able to convey complex ideas about the role and use of technology by businesses in a well-structured and coherent manner.

4.2 MODELLING BUSINESS ARCHITECTURE

4.2.1 Enterprise Architecture Development

1. Architectural business scenarios
2. Business modelling based on business scenarios
3. Business Process Modelling (BPM)
    a. functions, data/information flow
    b. modelling following the hierarchy (ICOM - input, controls, output, mechanisms/resources)
    c. annotation of activities, formulation of business rules
    d. ICOM relations
4. Business Process Modelling Notation (BPMN)
5. Use case models

    a. business events and actors (implementers, stakeholders)
    b. use-case diagrams and specifications
6. Class models (UML)
    a. a static model describing the relationships between information
    b. describes the behaviour of information (methods, operations) at the variable granularity
    c. business domain entities and implementation classes
    d. requires detailed specification (class models are not suitable)
7. Architecture repository
    a. generic business models (OMG - health, finance, transportation, TMF - telecommunications, Federal Enterprise Architecture Reference Model)
    b. Common Systems Architecture (REA, STEP, OAG, OAGIS, RosettaNet)

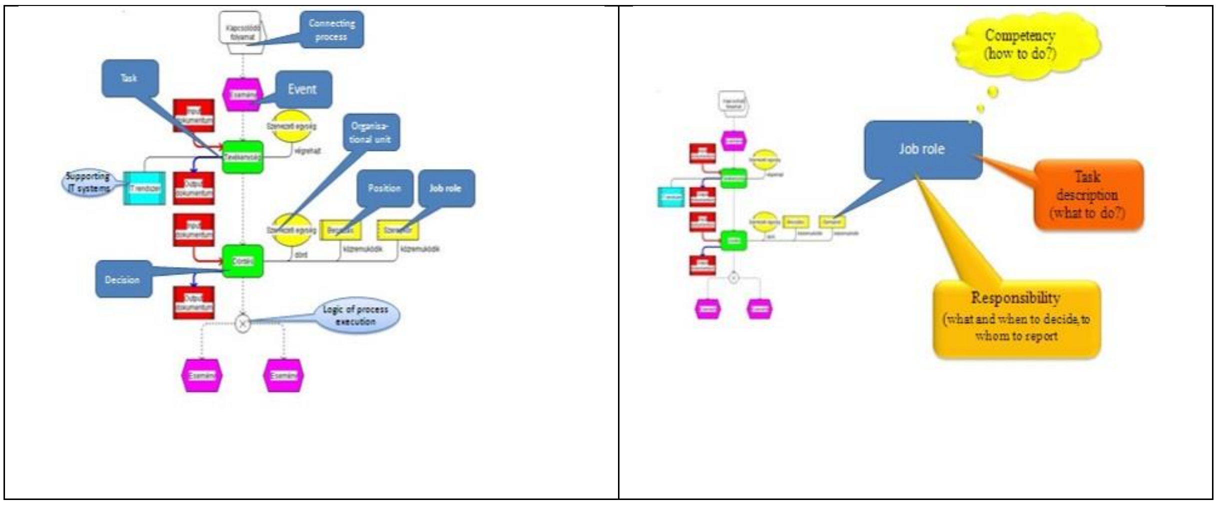
4.2.2 Case study steps and/or output

The expected output is:

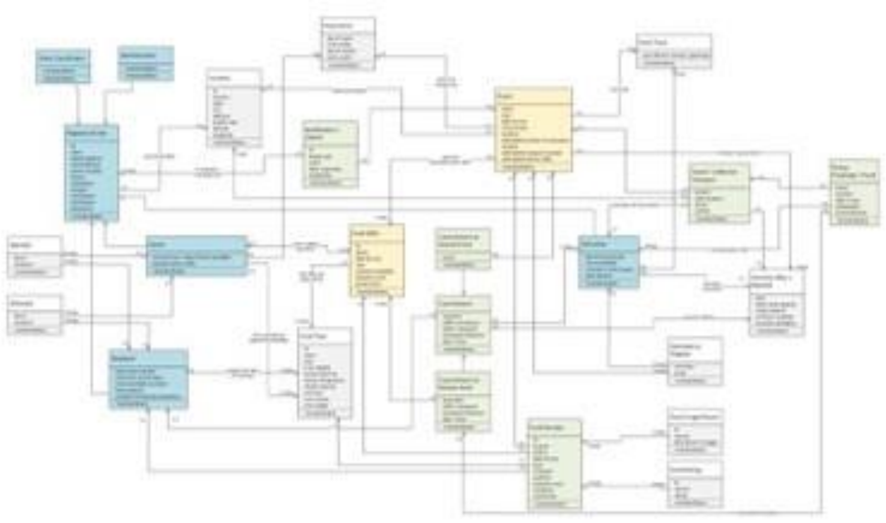
- Architectural business scenarios
- Business model using BPM
- Hierarchy of functions with information flow
- Business rules related to process steps
- Use case models using UML
- Class models using UML.

4.2.3 Need to know to perform the step

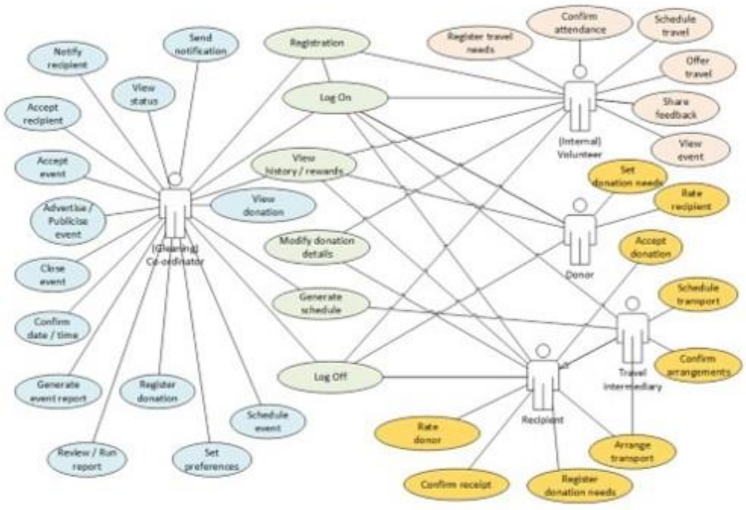
The case study can be continued with the *modelling tools* (HIPO, DFD, UML, BPM, BPMN ...) already known or introduced here. It should be pointed out that (1) use-cases are primarily a tool for *requirements analysis*, but also an indispensable tool for *testing*; (2) they are the primary means of *communication* between developer and user; (3) they (also) fill the *repository*. Process modelling (BPM) is presented; after the theoretical introduction, this phase provides an opportunity for practice (ARIS, ADONIS, Visio, …).



3. Figure Event-based Process Modelling (source: Authors)

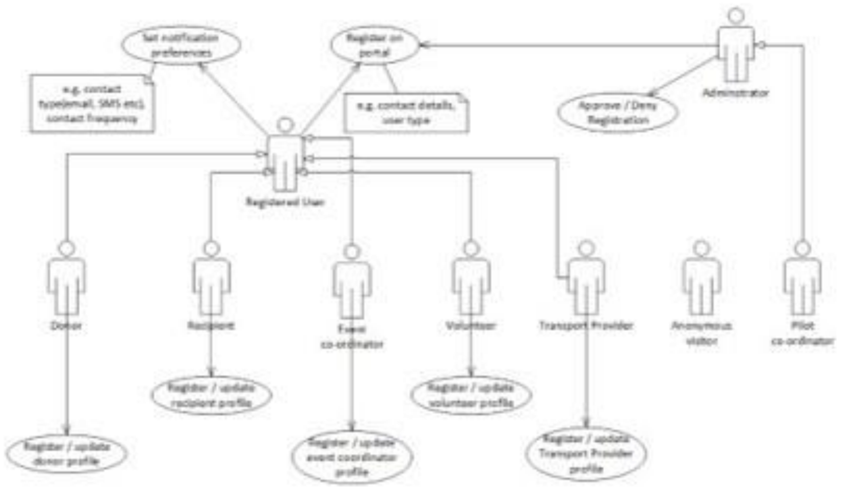


4. Figure Business Modelling option 2 (source: Authors)

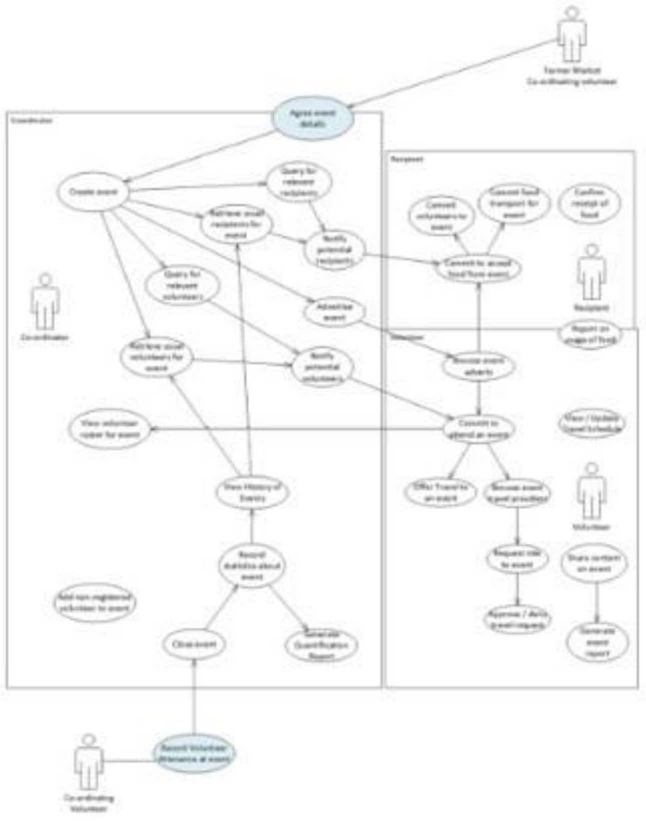


5. Figure Business Modelling option 3[1] (source: Savingfood)

[1] Savingfood 2.0 - *An innovative solution to tackle food waste through the collaborative power of ICT networks,* H2020 - Collective Awareness Platforms for Sustainability and Social Innovation" No. 688221


Set notification preferences
Register on portal
e.g. contact type(email, SMS etc), contact frequency
e.g. contact details, user type
Administrator
Approve / Deny Registration
Registered User
Donor
Recipient
Event co-ordinator
Volunteer
Transport Provider
Anonymous visitor
Pilot co-ordinator
Register / update recipient profile
Register / update volunteer profile
Register / update donor profile
Register / update event coordinator profile
Register / update Transport Provider profile

6. Figure Business Modelling option 4[2] (source: Savingfood)


Farmer Market Co-ordinating volunteer
Agree event details
Coordinator
Create event
Query for relevant recipients
Retrieve usual recipients for event
Notify potential recipients
Query for relevant volunteers
Advertise event
Retrieve usual volunteers for event
Notify potential volunteers
Co-ordinator
View volunteer roster for event
View History of Events
Record statistics about event
Add non registered volunteer to event
Close event
Generate Quantification Report
Record Volunteer Attendance at event
Co-ordinating Volunteer
Recipient
Cancel volunteers for event
Commit food transport for event
Confirm receipt of food
Commit to accept food from event
Recipient
Report on usage of food
Volunteer
Browse event adverts
Commit to attend an event
View / Update travel schedule
Offer Travel to an event
Browse event travel providers
Volunteer
Request ride to event
Share content on event
Approve / deny travel request
Generate event report

7. Figure Business Modelling option 5[3] (source: Savingfood)

[2] Savingfood 2.0 (idem)
[3] Savingfood 2.0 (idem)

4.2.4 What section(s) of the BIS curriculum are covered

Students will have the ability to understand and analyse business processes, prepare and perform requirements specification of software applications to support implementation, to perform simple programming tasks. Show a critical understanding of the process of business systems analysis and development and the role of the business analyst. Demonstrate a broad knowledge of the scope, defining features and main areas of the business systems provision process and environment. Display a broad knowledge of the general concepts associated with a typical business and the characteristics and importance of the processes that support that business. Students will understand the role of modelling and explain how models can be used to describe processes and requirements as well as design decisions. Students will be able to create based UML diagrams according to standard notations.

5 INFORMATION SYSTEMS ARCHITECTURE – APPLICATION AND DATA

5.1 INFORMATION SYSTEMS ARCHITECTURE VISION

5.1.1 Enterprise Architecture Development

1. Information system architecture focus: identification of the data and application architecture associated with the business architecture objectives
2. Approach: data-driven (EAP), application-driven (ERP) or a combination of both
3. Top-down and/or bottom-up implementation
    a. During the design phase: business - data (or application) - application (or data) - technology architecture
    b. In the implementation phase: reverse order
4. Architecture input
5. Reusable building blocks
6. Architecture output
    a. Revised architecture vision
    b. Architecture definition document
    c. Architecture requirements specifications (business, technical, constraints, gap, roadmap)

5.1.2 Case study steps and/or output

The expected output is:

- Refined and updated versions of the **Architecture Vision** phase deliverables, where applicable, including Statement of Architecture Work
- Draft **Architecture Definition Document**, including:
    - Baseline Data Architecture, Version 1.0
    - Target Data Architecture, Version 1.0
    - Baseline Application Architecture, Version 1.0
    - Target Application Architecture, Version 1.0
    - Data Architecture views (key stakeholder concerns)
    - Application Architecture views (key stakeholder concerns)
- **Draft Architecture Requirements Specification** including such Information Systems Architecture requirements as:
    - **Gap analysis** results
    - Relevant **technical** requirements that will apply to this evolution of the architecture development cycle
    - **Constraints** on the Technology Architecture about to be designed
    - Updated **business requirements**
    - Information systems components of an **Architecture Roadmap**

5.1.3 Need to know to perform the step

Geeks will not like this section, but it is a crucial step. First, the data-driven concepts (which are very trendy now) and application-driven architecture development need to be introduced, and the difference between the two is understood. Second, it needs to be made clear that all development has organisational limitations that can be managed smartly. The model of **maturity** and its use will be introduced[4]. A set of policies[5] should be formulated (application related, data management: GDPR is a trivial example. However, policies are intended to guide all subsequent steps (e.g., maintainability, reusability, etc.). The module ends with a revision of the target architecture.

Information system architecture modalities

5.1.3.1 Client/server

5.1.3.2 MVC[6]

MVC is an architectural pattern which means it rules the whole architecture of the applications. Even though often it is known as design pattern but we may be wrong if we refer it only as a design pattern because design patterns are used to solve a specific technical problem, whereas architecture pattern is used for solving architectural problems, so it affects the entire architecture of our application.

Model. It is known as the lowest level which means it is responsible for maintaining data. Handle data logically so it basically deals with data. The model is actually connected to the database so anything one does with data. Adding or retrieving data is done in the model component. It responds to the controller requests because the controller never talks to the database by itself. The model talks to the database back and forth and then it gives the needed data to the controller. Note: the model never communicated with the view directly.

View. Data representation is done by the view component. It actually generates UI or user interface for the user. So at web applications when students think of the view component just think the Html/CSS part. Views are created by the data which is collected by the model component but these data aren’t taken directly but through the controller, so the view only speaks to the controller.

Controller. It’s known as the main man because the controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn’t have to worry about handling data logic, it just tells the model what to do. After receiving data from the model it processes it and then it takes all that information it sends it to the view and explains how to represent to the user. Note: Views and models can not talk directly.

5.1.3.3 SoA[7]

SOA, or service-oriented architecture, defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications. This removes tasks from the application developer who previously redeveloped or duplicated existing functionality or had to know how to connect or provide interoperability with existing functions.

Each service in an SOA embodies the code and *data* required to execute a complete, discrete business function (e.g. checking a customer’s credit, calculating a monthly loan payment, or processing a mortgage application). The service interfaces provide loose coupling, meaning they can be called with little or no knowledge of how the *service* is implemented underneath, reducing the dependencies between applications.

4 These topics are also part of IT Governance, some of the details of which are also covered in the ADM steps.
5 Policy may mean many things, meta-standard, a choice of reference model, a standard, a guideline, a directive, etc.
6 https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1
7 https://www.ibm.com/cloud/learn/soa#toc-what-is-an-sqoa7ntn

This interface is a service contract between the service provider and service consumer. Applications behind the service interface can be written in Java, Microsoft .Net, Cobol or any other programming language, supplied as packaged software applications by a vendor (e.g., SAP), SaaS applications (e.g., Salesforce CRM), or obtained as open-source applications. Service interfaces are frequently defined using Web Service Definition Language (WSDL) which is a standard tag structure based on xml (extensible markup language).

The services are exposed using standard network protocols—such as SOAP (simple object access protocol)/HTTP or Restful HTTP (JSON/HTTP)—to send requests to read or change data. Service governance controls the lifecycle for development and at the appropriate stage the services are published in a *registry* that enables developers to quickly find them and reuse them to assemble new applications or business processes.

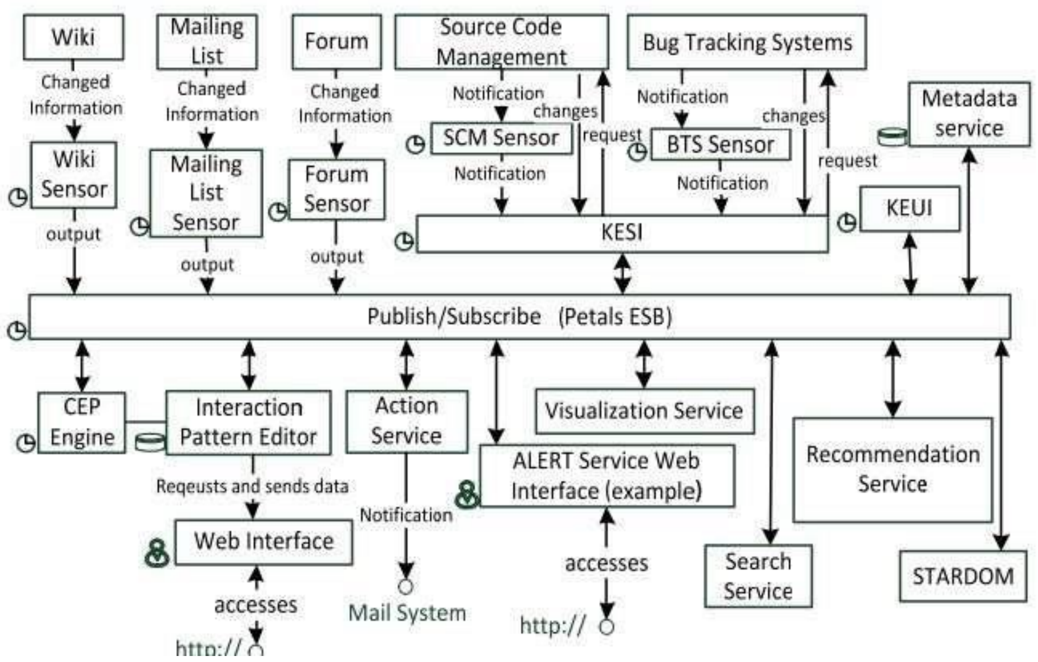
These services can be built from scratch but are often created by exposing functions from legacy systems of record as service interfaces.

In this way, SOA represents an important stage in the evolution of application development and integration over the last few decades. Before SOA emerged in the late 1990s, connecting an application to data or functionality housed in another system required complex point-to-point integration—integration that developers had to recreate, in part or whole, for each new development project. Exposing those functions through SOA *services allowed the developer to simply reuse the existing capability and connect through the SOA ESB architecture* (see below).

Note that although SOA, and the more recent microservices architecture, share many words in common (i.e. "service" and "architecture"), they are only loosely related and, in fact, operate at different scopes, as discussed later in this article.

An ESB, or **enterprise service bus**, is an architectural pattern whereby a centralized software component performs integrations between applications. It performs transformations of data models, handles connectivity/messaging, performs routing, converts communication protocols and potentially manages the composition of multiple requests. The ESB can make these integrations and transformations available as a service interface for reuse by new applications. The ESB pattern is typically implemented using a specially designed integration runtime and toolset that ensures the best possible productivity.

It is possible to implement an SOA without an ESB, but this would be equivalent to just having a bunch of services. Each application owner would need to directly connect to any service it needs and perform the necessary data transformations to meet each of the service interfaces. This is a lot of work (even if the interfaces are reusable) and creates a significant maintenance challenges in the future as each connection is point to point. In fact, ESBs were, eventually, considered such a de facto element of any SOA implementation that the two terms are sometimes used as synonyms, creating confusion.



8. Figure Publish/Subscribe[8] (Source: ALERT)

Publish/subscribe messaging, or pub/sub messaging, is a form of asynchronous service-to-service communication used in serverless and microservices architectures. In a pub/sub model, any message published to a topic is immediately received by all of the subscribers to the topic. Pub/sub messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability.[9] In modern cloud architecture, applications are decoupled into smaller, independent building blocks that are easier to develop, deploy and maintain. Publish/Subscribe (Pub/Sub) messaging provides instant event notifications for these distributed applications.

The Publish/Subscribe model allows messages to be broadcast to different parts of a system asynchronously. A sibling to a message queue, a message topic provides a lightweight mechanism to broadcast asynchronous event notifications, and endpoints that allow software components to connect to the topic in order to send and receive those messages. To broadcast a message, a component called a publisher simply pushes a message to the topic. Unlike message queues, which batch messages until they are retrieved, message topics transfer messages with no or very little queuing and push them out immediately to all subscribers. All components that subscribe to the topic will receive every message that is broadcast, unless a message filtering policy is set by the subscriber. The subscribers to the message topic often perform different functions and can each do something different with the message in parallel. The publisher doesn’t need to know who is using the information that it is broadcasting, and the subscribers don’t need to know who the message comes from.

[8] ALERT - *Active support and reaL-time coordination based on Event pRocessing in open source software development*, FP7-ICT, No. 258098

[9] https://aws.amazon.com/pub-sub-messaging/

5.1.4 What section(s) of the BIS curriculum are covered

Students will be able to apply their knowledge of information systems, understand the principles of architectural design, and interpret the components of computer and information architecture in context. Understand the role and impact of technology services on business performance. Understand the role of technology in business process-based value creation. Show detailed knowledge in service level management. Understand modern technologies and appropriate applications in explaining the key activities involved in the process of digitalizing a business and its processes. Demonstrate a broad knowledge of information systems that support business processes and activities (ERP, BI, CRM, SCM, …). Understand classification of IT applications based on the main business processes, sub-domains and strategic layers.

5.2 APPLICATION ARCHITECTURE

5.2.1 Enterprise Architecture Development

1. Application architecture design subordinate to business functions (Not equal to systems design, logical grouping of capabilities and technology-independent accounting)
2. Architecture repository - what is available (generic business models, ebXML, UML, integrated information system reference model, etc.)
3. Inputs: organisational model, framework, application protocols, target architecture, reusable building blocks, reference models, definition documents, requirements specification, data architecture and roadmap; maturity survey, communication plan
4. Implementation
    a. Reference model selection, perspectives and tools
    b. Modelling process (enumeration of applications, application components, logical and physical model, implementation plan for migration, development, operation, choice of decomposition rigour and granularity)
    c. Catalogue of building blocks (applications, interfaces) following the meta-model hierarchy
    d. Comparison with business functions (the data model can be used to validate the business architecture), iterations expected
    e. Create necessary diagrams (per perspective: communication, application-user, operational, process perspective, migration, software design, maintenance
    f. Requirement analysis (functional and non-functional requirements, assumptions and constraints, domain-specific data architecture principles, policies, standards, recommendations, specifications)
5. Application architecture design subordinate to business functions (Not equal to systems design, logical grouping of capabilities and technology-independent accounting)
    a. Baseline application architecture description (in terms of target architecture)
    b. Description of the target architecture
    c. Gap analysis (reused and/or new building blocks, checking model accuracy, consistency)
    d. Define roadmap elements (project plan, milestones)
    e. Impact assessment
    f. Formal stakeholder analysis (motivations, business practices, data architecture), identification of change needs
6. Architecture output

5.2.2 Case study steps and/or output

The expected output is:

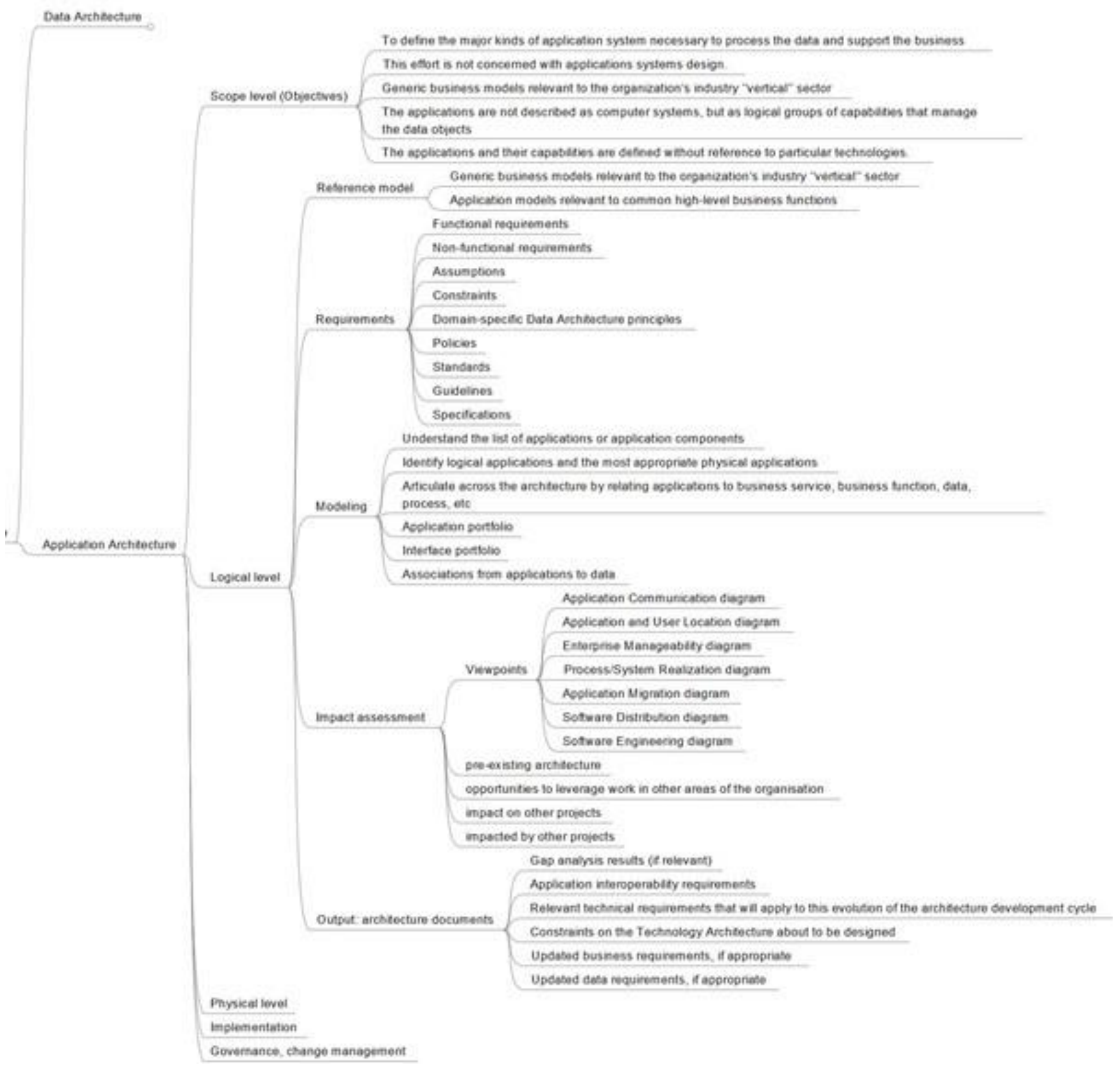
- Refined and updated versions of the Architecture Vision phase deliverables, where applicable:
    - Statement of Architecture Work, updated if necessary
    - Validated **application principles** or new application principles (if generated here)
- Draft Architecture Definition Document, including:
    - Baseline Application Architecture, Version 1.0, if appropriate
    - Target Application Architecture, Version 1.0
        - Process systems model
        - Place systems model
        - Time systems model
        - People systems model
- Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Draft Architecture Requirements Specification, including such Data Architecture requirements as:
    - Gap analysis results
    - Application interoperability requirements
    - Relevant technical requirements that will apply to this evolution of the architecture development cycle
    - Constraints on the Technology Architecture about to be designed
    - Updated business requirements, if appropriate
    - Updated data requirements, if appropriate
- Application Architecture components of an Architecture Roadmap

5.2.3 Need to know to perform the step

The training logic is similar to that described for data architecture. It is now possible to go further and include applications in the design by using the design software. At this point, project planning is enriched, the roadmap, priorities can change (it is good if they do), and stakeholders can have a meaningful say at this point.

Waterfall model vs **agile** development paradigms can be introduced, analysing the impact on the organisation, or conversely, identifying the causes and drivers of organisational resistance.

At this point, one can also dive into **business intelligence** as a specific application layer. On the other hand, the world of IoT, machine learning, automatic control (robots, AI), in short, intelligent systems, can also come into the picture here, of course, consistent with the data architecture.


Data Architecture
Application Architecture
Scope level (Objectives)
To define the major kinds of application system necessary to process the data and support the business
This effort is not concerned with applications systems design
Generic business models relevant to the organization's industry "vertical" sector
The applications are not described as computer systems, but as logical groups of capabilities that manage the data objects
The applications and their capabilities are defined without reference to particular technologies
Logical level
Reference model
Generic business models relevant to the organization's industry "vertical" sector
Application models relevant to common high-level business functions
Requirements
Functional requirements
Non-functional requirements
Assumptions
Constraints
Domain-specific Data Architecture principles
Policies
Standards
Guidelines
Specifications
Modeling
Understand the list of applications or application components
Identify logical applications and the most appropriate physical applications
Articulate across the architecture by relating applications to business service, business function, data, process, etc
Application portfolio
Interface portfolio
Associations from applications to data
Impact assessment
Viewpoints
Application Communication diagram
Application and User Location diagram
Enterprise Manageability diagram
Process/System Realization diagram
Application Migration diagram
Software Distribution diagram
Software Engineering diagram
pre-existing architecture
opportunities to leverage work in other areas of the organisation
impact on other projects
impacted by other projects
Output: architecture documents
Gap analysis results (if relevant)
Application interoperability requirements
Relevant technical requirements that will apply to this evolution of the architecture development cycle
Constraints on the Technology Architecture about to be designed
Updated business requirements, if appropriate
Updated data requirements, if appropriate
Physical level
Implementation
Governance, change management

9. Figure Application architecture (Source: Authors)

5.2.4 What section(s) of the BIS curriculum are covered

Students will:

- have the ability to apply the principles and tools of system development methodologies, implement business applications, and initiate organisational changes required for the implementation;
- be able to plan and manage small development projects;
- have basic knowledge of all areas of information management, including IT strategy, process management, systems development, knowledge management, IT service management, project management, risk management, performance management, IT asset management, IT security and IT audit;
- demonstrate an understanding of the concepts and techniques by applying object-oriented modelling techniques in analysis and design of an object-oriented system;

- demonstrate a systematic understanding of a range of current technologies and methods and the role of these in the effective design, implementation and usability of computer-based systems;
- be able to systematically apply software development methods in the production of a software system and evaluate the system's fitness for purpose.

5.3 DATA ARCHITECTURE

5.3.1 Enterprise Architecture Development

1. Data architecture design: to design data content and entities that are appropriate for the business architecture, to explore the relationships with existing databases
2. Data management
    a. Which applications produce or manage the master data?
    b. All applications must relate to the data in a consistent and coherent way
    c. Explicitly define relationships between data and business functions
    d. Where data is generated, how it is stored, transmitted, data security
    e. Software applications to support data integration (e.g. ETL)
3. Data migration
4. Architecture input (organisational model, architecture framework, data management principles, policies
5. Architecture repository: generic data models
6. Data architecture design: choice of level of detail, resolution
7. Reference models, perspectives and tools (E-R models, class diagrams)
    a. Define the data modelling process (viewpoints, stakeholder perspectives, data collection, rationalisation, review, finalisation)
    b. Identification and definition of the catalogue of data building blocks (data set, hierarchy, data model preparation, key data entities, traceability, finalisation)
    c. Assessment of data relationships and their impact, data set rationalisation, comparison with application architecture
    d. Preparation of data relationship diagrams (enterprise and local, class diagrams, lifecycle, security, migration, hierarchy diagrams)
    e. Requirement's identification (functional and non-functional, assumptions, constraints, domain data architecture principles, policies, standards, recommendations, specifications
8. Data architecture design objective: design data content and entities appropriate to the business architecture, explore relationships with existing databases
9. Description of baseline and target data architecture
    a. the level of detail is determined by the target architecture and target business architecture
    b. rely as much as possible on building blocks and repository
    c. stakeholder considerations should be validated (security, redundancy)
10. Perform a gap analysis
    a. check consistency, completeness and accuracy
    b. identify obsolete (to be deleted) and new building blocks
    c. identification of building blocks to be upgraded and procured
11. Identification of roadmap components
    a. preparation of the roadmap

    b. prioritisation
12. Exploring the impacts on the architectural landscape
    a. how the new architecture will affect the existing ones
    b. whether the data architecture has any impact on architectures in other areas of the organisation
    c. whether the data architecture has any impact on other projects in the organisation or their impact on the data architecture
13. Formal stakeholder interviews and analysis
14. Finalisation of the data architecture
15. Creation of architecture document
16. Architecture output

5.3.2 Case study steps and/or output

The expected output is:

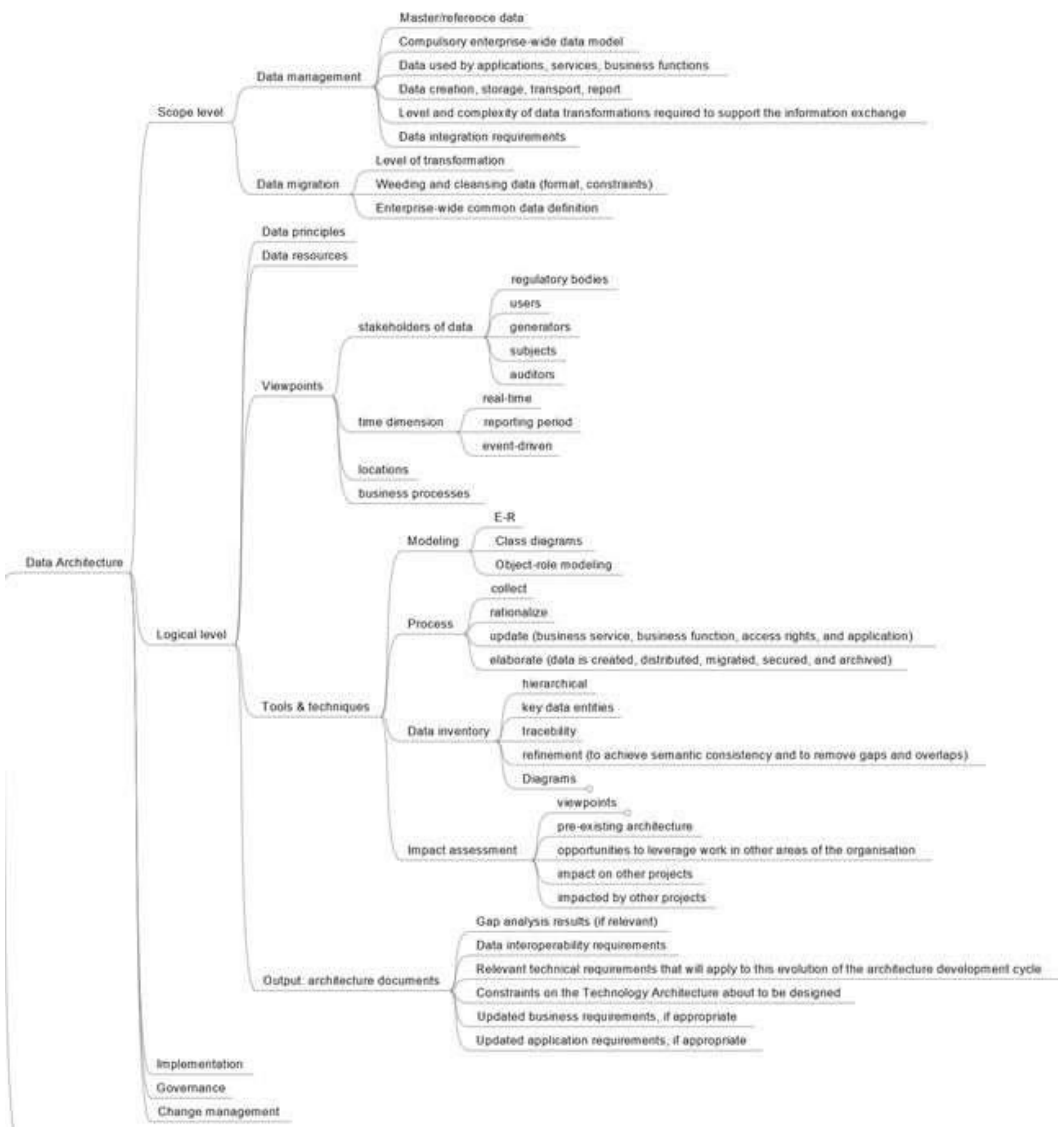
- Refined and updated versions of the Architecture Vision phase deliverables, where applicable:
  - Statement of Architecture Work, updated if necessary
  - Validated **data principles** or new data principles (if generated here)
- Draft Architecture Definition Document, including:
  - Baseline Data Architecture, Version 1.0, if appropriate
  - Target Data Architecture, Version 1.0 (
    - Business data model
    - Logical data model
    - Data management process models
    - Data Entity/Business Function matrix
  - Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Draft Architecture Requirements Specification, including such Data Architecture requirements as:
  - Gap analysis results
  - Data interoperability requirements
  - Relevant technical requirements that will apply to this evolution of the architecture development cycle
  - Constraints on the Technology Architecture about to be designed
  - Updated business requirements, if appropriate
  - Updated application requirements, if appropriate
- Data Architecture components of an Architecture Roadmap

5.3.3 Need to know to perform the step

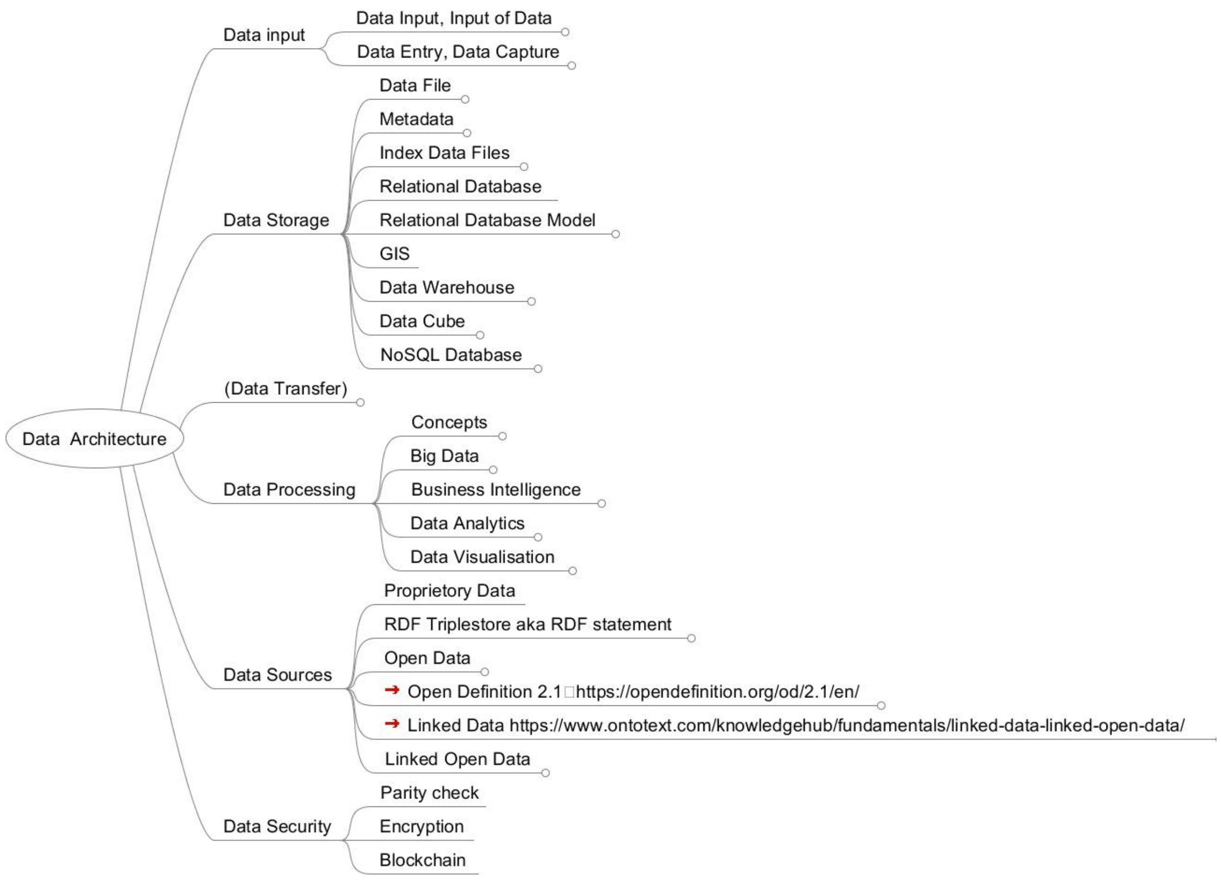
At this point, the architecture design software and practice can be introduced because it can be used to implement what has been said for demonstration purposes. Obviously, at this point, students can focus on *data structures*. Still, everything from *transactional data processing to OLAP* and *data marketplace to cloud computing* can and should be brought forward or referenced to what has already been learned. The first part focuses mainly on the *data asset management* aspects, while the second focuses on implementation. The emphasis on *security, integrity and repository* is significant.

Data modelling continues with *gap analysis*. This is didactically necessary because it highlights the benefits of using a repository, the importance of road mapping (see also **project management**), prioritisation, and coordination with other ongoing projects.

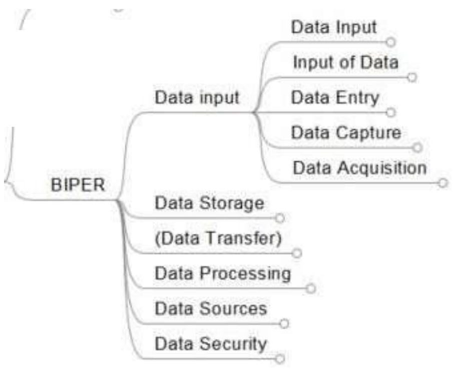
This is partly a matter of taste, but it can also be a given that data architecture precedes or follows the development of the application architecture; in fact, the two go hand in hand and should be seen as an iterative process. From an educational point of view, it is helpful to have many examples.


Data Architecture
Scope level
Data management
Master/reference data
Compulsory enterprise-wide data model
Data used by applications, services, business functions
Data creation, storage, transport, report
Level and complexity of data transformations required to support the information exchange
Data integration requirements
Data migration
Level of transformation
Weeding and cleansing data (format, constraints)
Enterprise-wide common data definition
Logical level
Data principles
Data resources
Viewpoints
stakeholders of data
regulatory bodies
users
generators
subjects
auditors
time dimension
real-time
reporting period
event-driven
locations
business processes
Tools & techniques
Modeling
E-R
Class diagrams
Object-role modeling
Process
collect
rationalize
update (business service, business function, access rights, and application)
elaborate (data is created, distributed, migrated, secured, and archived)
Data inventory
hierarchical
key data entities
tracebility
refinement (to achieve semantic consistency and to remove gaps and overlaps)
Diagrams
Impact assessment
viewpoints
pre-existing architecture
opportunities to leverage work in other areas of the organisation
impact on other projects
impacted by other projects
Output: architecture documents
Gap analysis results (if relevant)
Data interoperability requirements
Relevant technical requirements that will apply to this evolution of the architecture development cycle
Constraints on the Technology Architecture about to be designed
Updated business requirements, if appropriate
Updated application requirements, if appropriate
Implementation
Governance
Change management

1. Figure Data architecture (Source: Authors)



10. Figure Data Processing Overview (Source: Authors)

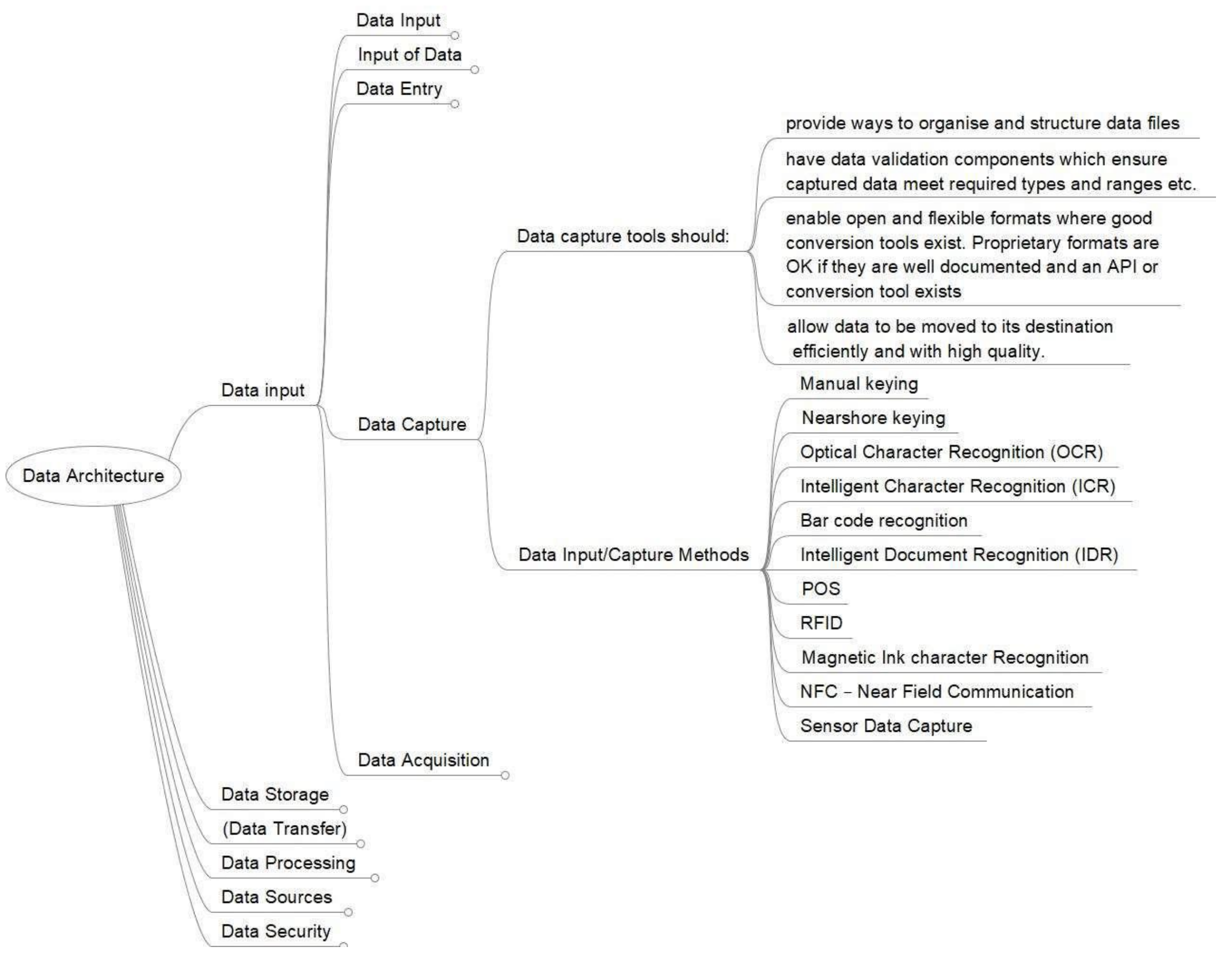


11. Figure Data Input (Source: Authors)

Data input can be performed in several ways, including by data entry. In data entry, data is placed in chosen fields of a database by a human agent using a device such as a mouse, keypad, keyboard, touch screen, or stylus, or alternatively, with speech recognition software. Data capture is a kind of data input in which there is no data entry. Instead, data is collected in conjunction with a separate activity.

Input of data, not as a direct result of data entry but instead as a result of performing a different but related activity. E.g- Barcode reader equipped supermarket checkout counters, for example, capture inventory related data while recording a sale. See also data collection and data logging.

In **data entry**, data is placed in chosen fields of a database by a human agent using a device such as a mouse, keypad, keyboard, touch screen, or stylus, or alternatively, with speech recognition software. Data capture is a kind of data input in which there is no data entry.

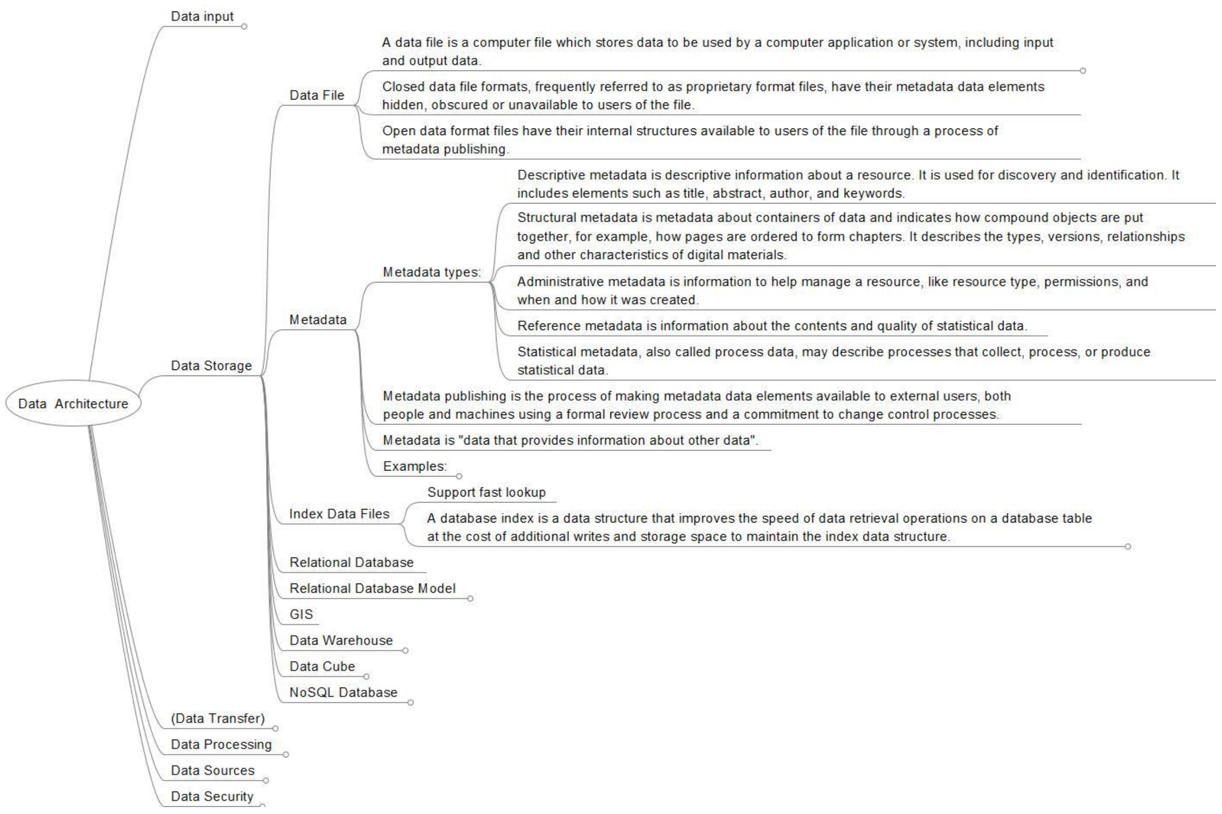


Data Architecture
Data input
Data Input
Input of Data
Data Entry
Data Capture
Data capture tools should:
provide ways to organise and structure data files
have data validation components which ensure captured data meet required types and ranges etc.
enable open and flexible formats where good conversion tools exist. Proprietary formats are OK if they are well documented and an API or conversion tool exists
allow data to be moved to its destination efficiently and with high quality.
Data Input/Capture Methods
Manual keying
Nearshore keying
Optical Character Recognition (OCR)
Intelligent Character Recognition (ICR)
Bar code recognition
Intelligent Document Recognition (IDR)
POS
RFID
Magnetic Ink character Recognition
NFC – Near Field Communication
Sensor Data Capture
Data Acquisition
Data Storage
(Data Transfer)
Data Processing
Data Sources
Data Security

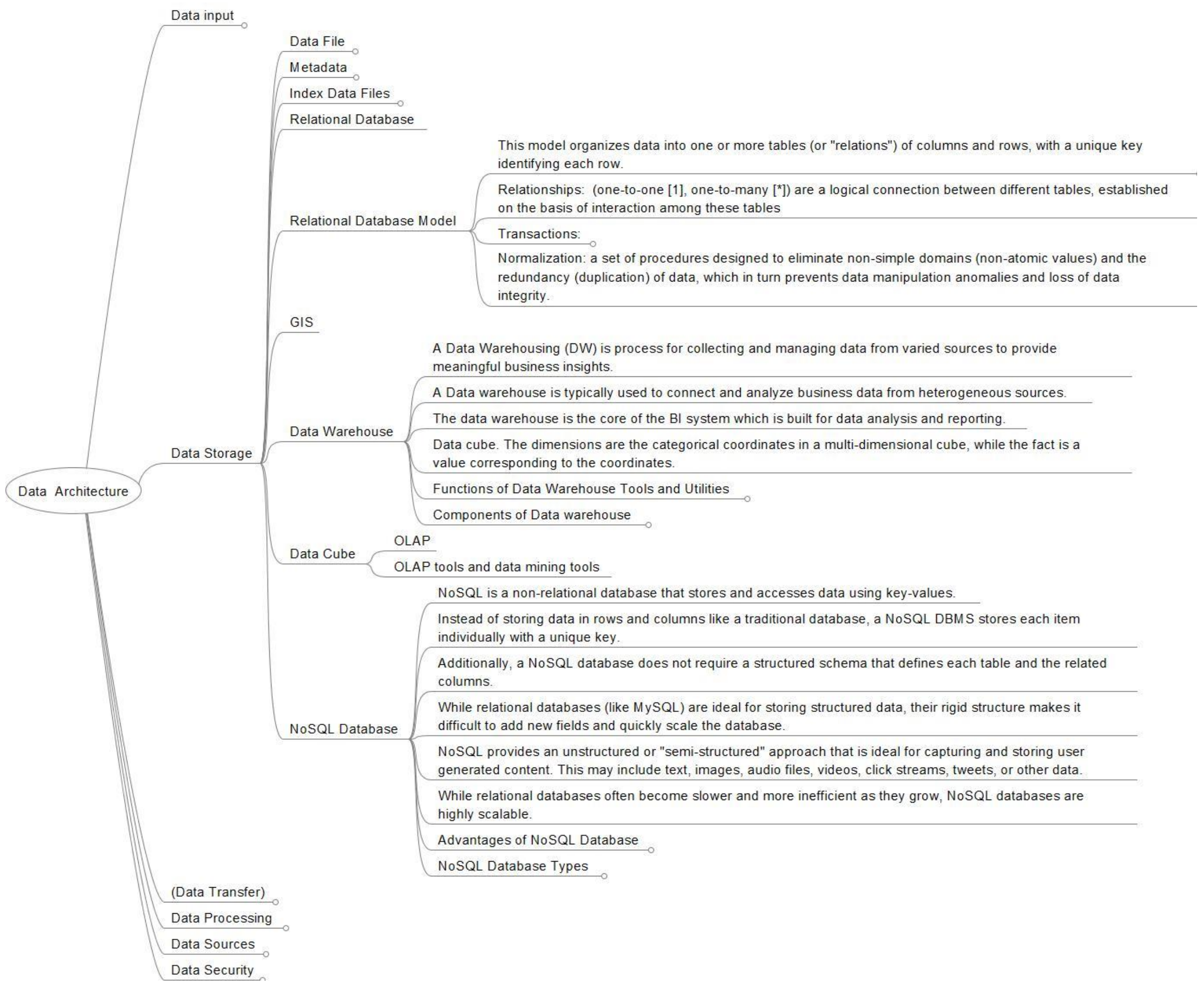
12. **Figure Data Capture** (Source: Authors)

Data capture is the process of collecting data which will be processed and used later to fulfil certain purposes. Ways of capturing data can range from high end technologies (e.g. Synchrotron, sensor networks and computer simulation models) to low end paper instruments used in the field. Data with good metadata attached at the point of capture can expediate data sharing, publishing and citation.

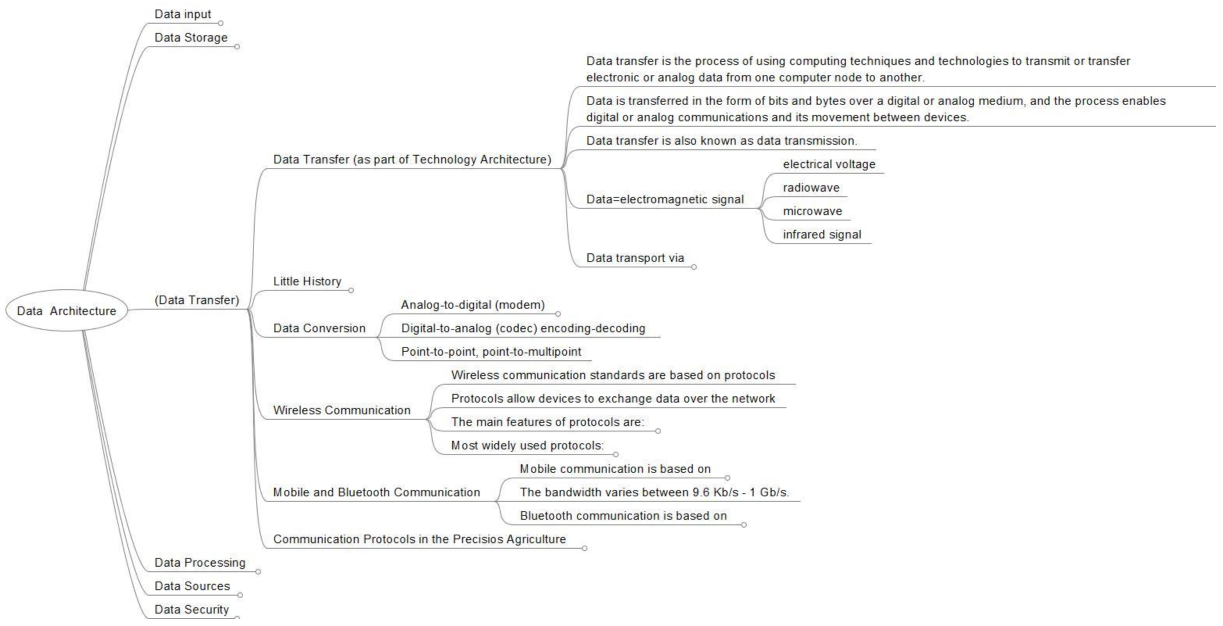
Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems, abbreviated by the acronyms DAS or DAQ, typically convert analogue waveforms into digital values for processing.


Data Architecture
Data input
Data Storage
Data File
A data file is a computer file which stores data to be used by a computer application or system, including input and output data.
Closed data file formats, frequently referred to as proprietary format files, have their metadata data elements hidden, obscured or unavailable to users of the file.
Open data format files have their internal structures available to users of the file through a process of metadata publishing.
Metadata
Metadata types:
Descriptive metadata is descriptive information about a resource. It is used for discovery and identification. It includes elements such as title, abstract, author, and keywords.
Structural metadata is metadata about containers of data and indicates how compound objects are put together, for example, how pages are ordered to form chapters. It describes the types, versions, relationships and other characteristics of digital materials.
Administrative metadata is information to help manage a resource, like resource type, permissions, and when and how it was created.
Reference metadata is information about the contents and quality of statistical data.
Statistical metadata, also called process data, may describe processes that collect, process, or produce statistical data.
Metadata publishing is the process of making metadata data elements available to external users, both people and machines using a formal review process and a commitment to change control processes.
Metadata is "data that provides information about other data".
Examples:
Index Data Files
Support fast lookup
A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure.
Relational Database
Relational Database Model
GIS
Data Warehouse
Data Cube
NoSQL Database
(Data Transfer)
Data Processing
Data Sources
Data Security

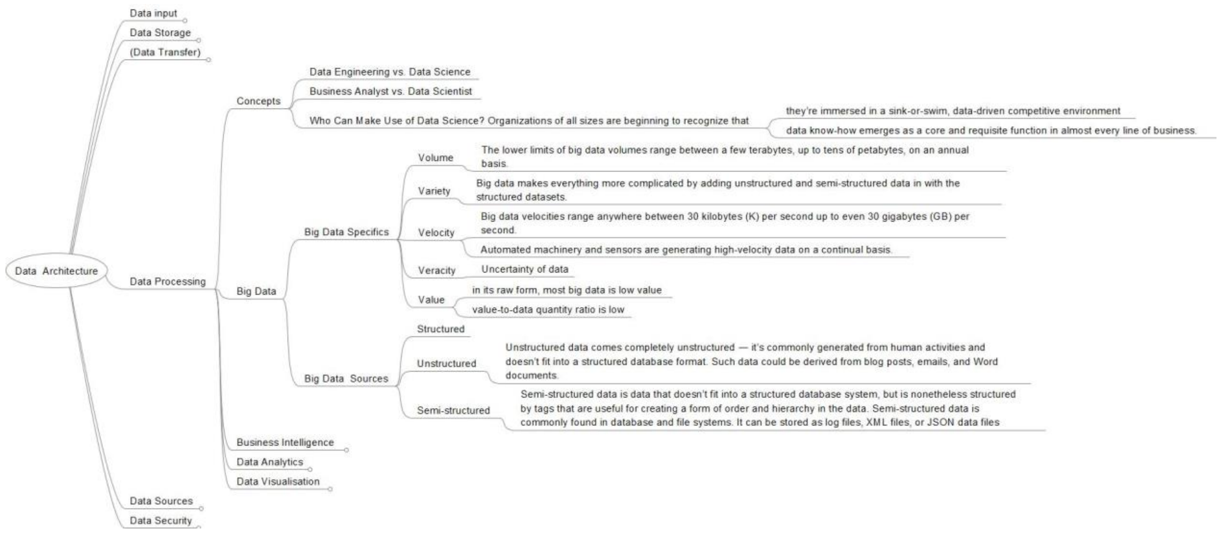
13. Figure Data Storage 1 (Source: Authors)


Data Architecture
Data input
Data Storage
Data File
Metadata
Index Data Files
Relational Database
Relational Database Model
This model organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row.
Relationships: (one-to-one [1], one-to-many [*]) are a logical connection between different tables, established on the basis of interaction among these tables
Transactions:
Normalization: a set of procedures designed to eliminate non-simple domains (non-atomic values) and the redundancy (duplication) of data, which in turn prevents data manipulation anomalies and loss of data integrity.
GIS
Data Warehouse
A Data Warehousing (DW) is process for collecting and managing data from varied sources to provide meaningful business insights.
A Data warehouse is typically used to connect and analyze business data from heterogeneous sources.
The data warehouse is the core of the BI system which is built for data analysis and reporting.
Data cube. The dimensions are the categorical coordinates in a multi-dimensional cube, while the fact is a value corresponding to the coordinates.
Functions of Data Warehouse Tools and Utilities
Components of Data warehouse
Data Cube
OLAP
OLAP tools and data mining tools
NoSQL Database
NoSQL is a non-relational database that stores and accesses data using key-values.
Instead of storing data in rows and columns like a traditional database, a NoSQL DBMS stores each item individually with a unique key.
Additionally, a NoSQL database does not require a structured schema that defines each table and the related columns.
While relational databases (like MySQL) are ideal for storing structured data, their rigid structure makes it difficult to add new fields and quickly scale the database.
NoSQL provides an unstructured or "semi-structured" approach that is ideal for capturing and storing user generated content. This may include text, images, audio files, videos, click streams, tweets, or other data.
While relational databases often become slower and more inefficient as they grow, NoSQL databases are highly scalable.
Advantages of NoSQL Database
NoSQL Database Types
(Data Transfer)
Data Processing
Data Sources
Data Security

14. Figure Data Storage 2 (Source: Authors)

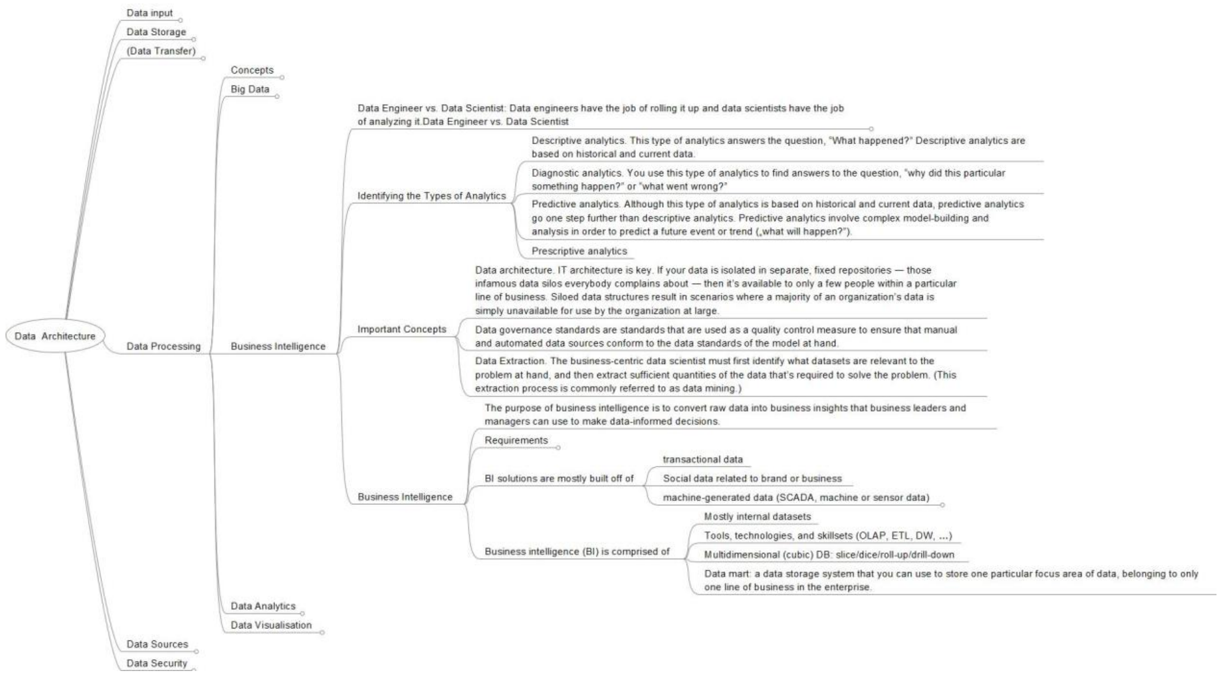

Data Architecture
Data input
Data Storage
(Data Transfer)
Data Transfer (as part of Technology Architecture)
Data transfer is the process of using computing techniques and technologies to transmit or transfer electronic or analog data from one computer node to another.
Data is transferred in the form of bits and bytes over a digital or analog medium, and the process enables digital or analog communications and its movement between devices.
Data transfer is also known as data transmission.
Data=electromagnetic signal
electrical voltage
radiowave
microwave
infrared signal
Data transport via
Little History
Data Conversion
Analog-to-digital (modem)
Digital-to-analog (codec) encoding-decoding
Point-to-point, point-to-multipoint
Wireless Communication
Wireless communication standards are based on protocols
Protocols allow devices to exchange data over the network
The main features of protocols are:
Most widely used protocols:
Mobile and Bluetooth Communication
Mobile communication is based on
The bandwidth varies between 9.6 Kb/s - 1 Gb/s.
Bluetooth communication is based on
Communication Protocols in the Precisios Agriculture
Data Processing
Data Sources
Data Security

15. Figure Data Transfer (Source: Authors)



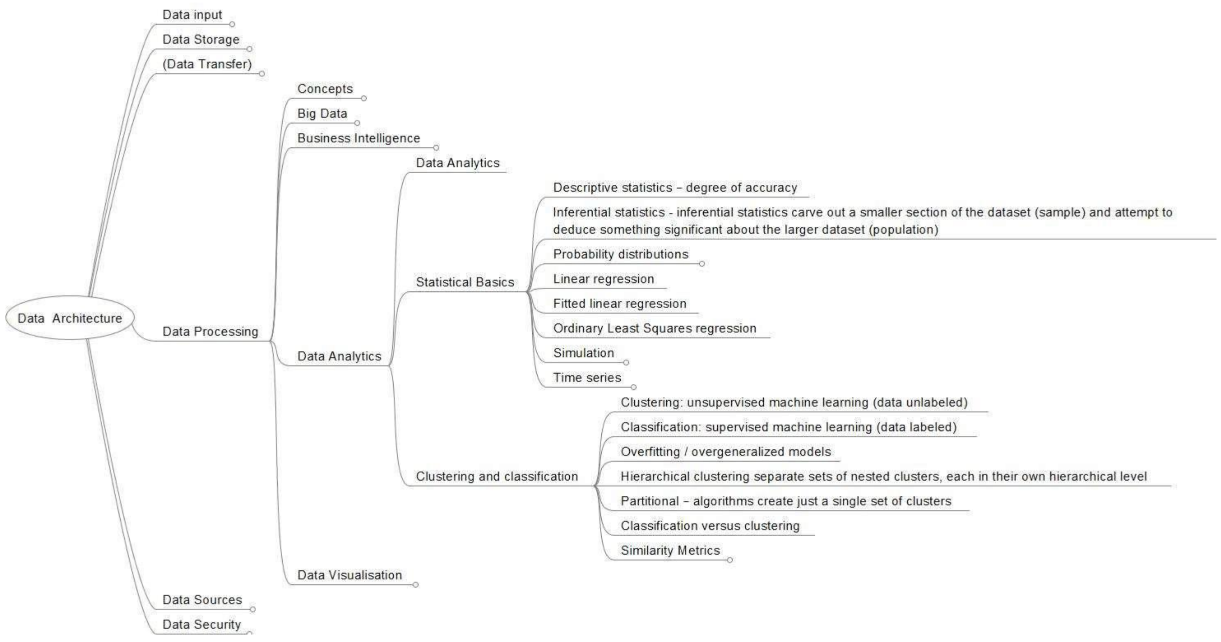
- Data Architecture
  - Data input
  - Data Storage
  - (Data Transfer)
  - Data Processing
    - Concepts
      - Data Engineering vs. Data Science
      - Business Analyst vs. Data Scientist
      - Who Can Make Use of Data Science? Organizations of all sizes are beginning to recognize that
        - they're immersed in a sink-or-swim, data-driven competitive environment
        - data know-how emerges as a core and requisite function in almost every line of business.
    - Big Data
      - Big Data Specifics
        - Volume
          - The lower limits of big data volumes range between a few terabytes, up to tens of petabytes, on an annual basis.
        - Variety
          - Big data makes everything more complicated by adding unstructured and semi-structured data in with the structured datasets.
        - Velocity
          - Big data velocities range anywhere between 30 kilobytes (K) per second up to even 30 gigabytes (GB) per second.
          - Automated machinery and sensors are generating high-velocity data on a continual basis.
        - Veracity
          - Uncertainty of data
        - Value
          - in its raw form, most big data is low value
          - value-to-data quantity ratio is low
      - Big Data Sources
        - Structured
        - Unstructured
          - Unstructured data comes completely unstructured — it's commonly generated from human activities and doesn't fit into a structured database format. Such data could be derived from blog posts, emails, and Word documents.
        - Semi-structured
          - Semi-structured data is data that doesn't fit into a structured database system, but is nonetheless structured by tags that are useful for creating a form of order and hierarchy in the data. Semi-structured data is commonly found in database and file systems. It can be stored as log files, XML files, or JSON data files
    - Business Intelligence
    - Data Analytics
    - Data Visualisation
  - Data Sources
  - Data Security

16. Figure Big Data (Source: Authors)



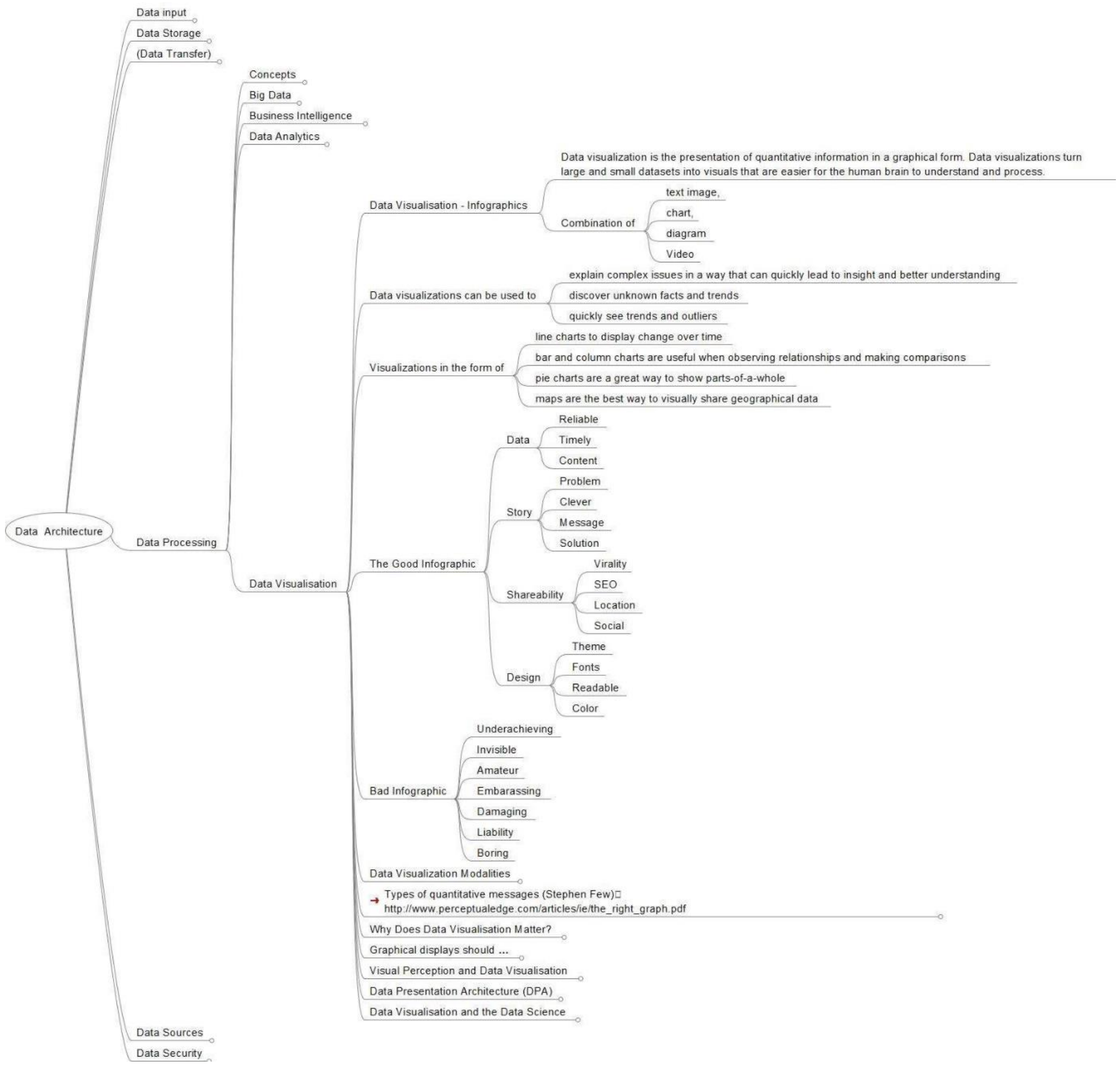
- Data Architecture
  - Data input
  - Data Storage
  - (Data Transfer)
  - Data Processing
    - Concepts
    - Big Data
    - Business Intelligence
      - Data Engineer vs. Data Scientist: Data engineers have the job of rolling it up and data scientists have the job of analyzing it.Data Engineer vs. Data Scientist
      - Identifying the Types of Analytics
        - Descriptive analytics. This type of analytics answers the question, "What happened?" Descriptive analytics are based on historical and current data.
        - Diagnostic analytics. You use this type of analytics to find answers to the question, "why did this particular something happen?" or "what went wrong?"
        - Predictive analytics. Although this type of analytics is based on historical and current data, predictive analytics go one step further than descriptive analytics. Predictive analytics involve complex model-building and analysis in order to predict a future event or trend („what will happen?").
        - Prescriptive analytics
      - Important Concepts
        - Data architecture. IT architecture is key. If your data is isolated in separate, fixed repositories — those infamous data silos everybody complains about — then it's available to only a few people within a particular line of business. Siloed data structures result in scenarios where a majority of an organization's data is simply unavailable for use by the organization at large.
        - Data governance standards are standards that are used as a quality control measure to ensure that manual and automated data sources conform to the data standards of the model at hand.
        - Data Extraction. The business-centric data scientist must first identify what datasets are relevant to the problem at hand, and then extract sufficient quantities of the data that's required to solve the problem. (This extraction process is commonly referred to as data mining.)
      - Business Intelligence
        - The purpose of business intelligence is to convert raw data into business insights that business leaders and managers can use to make data-informed decisions.
        - Requirements
        - BI solutions are mostly built off of
          - transactional data
          - Social data related to brand or business
          - machine-generated data (SCADA, machine or sensor data)
        - Business intelligence (BI) is comprised of
          - Mostly internal datasets
          - Tools, technologies, and skillsets (OLAP, ETL, DW, ...)
          - Multidimensional (cubic) DB: slice/dice/roll-up/drill-down
          - Data mart: a data storage system that you can use to store one particular focus area of data, belonging to only one line of business in the enterprise.
    - Data Analytics
    - Data Visualisation
  - Data Sources
  - Data Security

17. Figure Business Intelligence (Source: Authors)

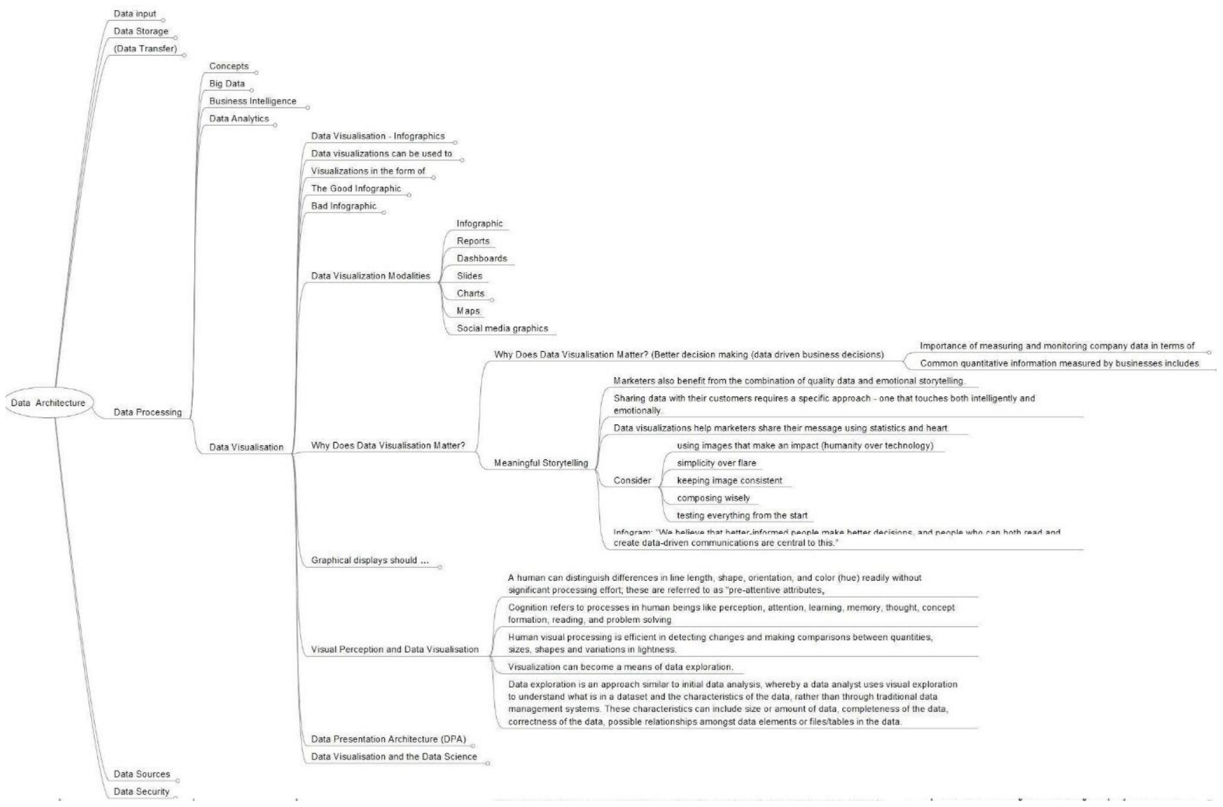


Data Architecture
Data input
Data Storage
(Data Transfer)
Data Processing
Concepts
Big Data
Business Intelligence
Data Analytics
Data Analytics
Statistical Basics
Descriptive statistics – degree of accuracy
Inferential statistics - inferential statistics carve out a smaller section of the dataset (sample) and attempt to deduce something significant about the larger dataset (population)
Probability distributions
Linear regression
Fitted linear regression
Ordinary Least Squares regression
Simulation
Time series
Clustering and classification
Clustering: unsupervised machine learning (data unlabeled)
Classification: supervised machine learning (data labeled)
Overfitting / overgeneralized models
Hierarchical clustering separate sets of nested clusters, each in their own hierarchical level
Partitional – algorithms create just a single set of clusters
Classification versus clustering
Similarity Metrics
Data Visualisation
Data Sources
Data Security

18. Figure Data Analytics (Source: Authors)

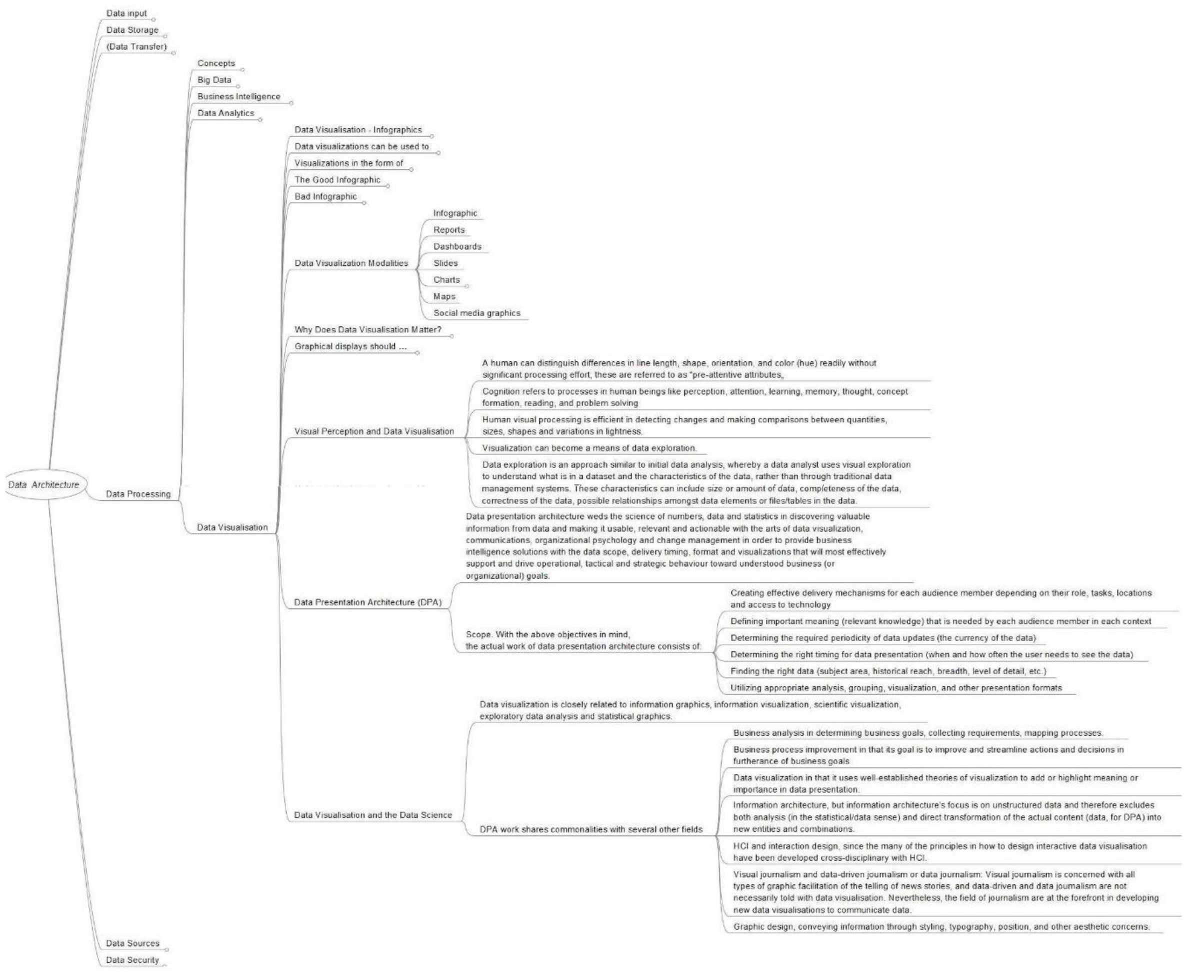

Data Architecture
Data input
Data Storage
(Data Transfer)
Data Processing
Concepts
Big Data
Business Intelligence
Data Analytics
Data Visualisation
Data Visualisation - Infographics
Data visualization is the presentation of quantitative information in a graphical form. Data visualizations turn large and small datasets into visuals that are easier for the human brain to understand and process.
Combination of
text image,
chart,
diagram
Video
Data visualizations can be used to
explain complex issues in a way that can quickly lead to insight and better understanding
discover unknown facts and trends
quickly see trends and outliers
Visualizations in the form of
line charts to display change over time
bar and column charts are useful when observing relationships and making comparisons
pie charts are a great way to show parts-of-a-whole
maps are the best way to visually share geographical data
The Good Infographic
Data
Reliable
Timely
Content
Story
Problem
Clever
Message
Solution
Shareability
Virality
SEO
Location
Social
Design
Theme
Fonts
Readable
Color
Bad Infographic
Underachieving
Invisible
Amateur
Embarassing
Damaging
Liability
Boring
Data Visualization Modalities
Types of quantitative messages (Stephen Few)
http://www.perceptualedge.com/articles/ie/the_right_graph.pdf
Why Does Data Visualisation Matter?
Graphical displays should ...
Visual Perception and Data Visualisation
Data Presentation Architecture (DPA)
Data Visualisation and the Data Science
Data Sources
Data Security

19. Figure Data Visualisation 1 (Source: Authors)



Data Architecture
Data input
Data Storage
(Data Transfer)
Data Processing
Concepts
Big Data
Business Intelligence
Data Analytics
Data Visualisation
Data Visualisation - Infographics
Data visualizations can be used to
Visualizations in the form of
The Good Infographic
Bad Infographic
Data Visualization Modalities
Infographic
Reports
Dashboards
Slides
Charts
Maps
Social media graphics
Why Does Data Visualisation Matter?
Why Does Data Visualisation Matter? (Better decision making (data driven business decisions)
Importance of measuring and monitoring company data in terms of
Common quantitative information measured by businesses includes
Meaningful Storytelling
Marketers also benefit from the combination of quality data and emotional storytelling.
Sharing data with their customers requires a specific approach - one that touches both intelligently and emotionally.
Data visualizations help marketers share their message using statistics and heart.
Consider
using images that make an impact (humanity over technology)
simplicity over flare
keeping image consistent
composing wisely
testing everything from the start
Infogram: "We believe that better-informed people make better decisions, and people who can both read and create data-driven communications are central to this."
Graphical displays should ...
Visual Perception and Data Visualisation
A human can distinguish differences in line length, shape, orientation, and color (hue) readily without significant processing effort; these are referred to as "pre-attentive attributes,
Cognition refers to processes in human beings like perception, attention, learning, memory, thought, concept formation, reading, and problem solving
Human visual processing is efficient in detecting changes and making comparisons between quantities, sizes, shapes and variations in lightness.
Visualization can become a means of data exploration.
Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems. These characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data.
Data Presentation Architecture (DPA)
Data Visualisation and the Data Science
Data Sources
Data Security

20. Figure Data Visualisation 2 (Source: Authors)



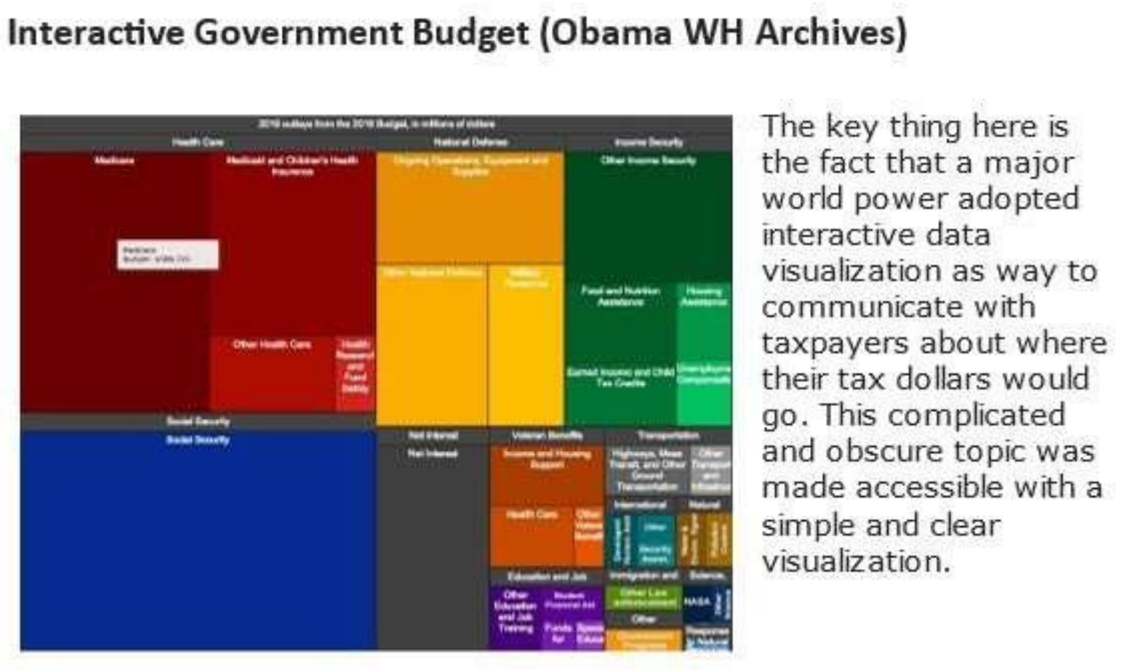
Data Architecture

- Data input
- Data Storage
- (Data Transfer)
- Data Processing
  - Concepts
  - Big Data
  - Business Intelligence
  - Data Analytics
  - Data Visualisation
    - Data Visualisation - Infographics
    - Data visualizations can be used to
    - Visualizations in the form of
    - The Good Infographic
    - Bad Infographic
    - Data Visualization Modalities
      - Infographic
      - Reports
      - Dashboards
      - Slides
      - Charts
      - Maps
      - Social media graphics
    - Why Does Data Visualisation Matter?
    - Graphical displays should ...
    - Visual Perception and Data Visualisation
      - A human can distinguish differences in line length, shape, orientation, and color (hue) readily without significant processing effort; these are referred to as "pre-attentive attributes„
      - Cognition refers to processes in human beings like perception, attention, learning, memory, thought, concept formation, reading, and problem solving
      - Human visual processing is efficient in detecting changes and making comparisons between quantities, sizes, shapes and variations in lightness.
      - Visualization can become a means of data exploration.
      - Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems. These characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data.
    - Data Presentation Architecture (DPA)
      - Data presentation architecture weds the science of numbers, data and statistics in discovering valuable information from data and making it usable, relevant and actionable with the arts of data visualization, communications, organizational psychology and change management in order to provide business intelligence solutions with the data scope, delivery timing, format and visualizations that will most effectively support and drive operational, tactical and strategic behaviour toward understood business (or organizational) goals.
      - Scope. With the above objectives in mind, the actual work of data presentation architecture consists of:
        - Creating effective delivery mechanisms for each audience member depending on their role, tasks, locations and access to technology
        - Defining important meaning (relevant knowledge) that is needed by each audience member in each context
        - Determining the required periodicity of data updates (the currency of the data)
        - Determining the right timing for data presentation (when and how often the user needs to see the data)
        - Finding the right data (subject area, historical reach, breadth, level of detail, etc.)
        - Utilizing appropriate analysis, grouping, visualization, and other presentation formats
    - Data Visualisation and the Data Science
      - Data visualization is closely related to information graphics, information visualization, scientific visualization, exploratory data analysis and statistical graphics.
      - DPA work shares commonalities with several other fields
        - Business analysis in determining business goals, collecting requirements, mapping processes.
        - Business process improvement in that its goal is to improve and streamline actions and decisions in furtherance of business goals
        - Data visualization in that it uses well-established theories of visualization to add or highlight meaning or importance in data presentation.
        - Information architecture, but information architecture's focus is on unstructured data and therefore excludes both analysis (in the statistical/data sense) and direct transformation of the actual content (data, for DPA) into new entities and combinations.
        - HCI and interaction design, since the many of the principles in how to design interactive data visualisation have been developed cross-disciplinary with HCI.
        - Visual journalism and data-driven journalism or data journalism: Visual journalism is concerned with all types of graphic facilitation of the telling of news stories, and data-driven and data journalism are not necessarily told with data visualisation. Nevertheless, the field of journalism are at the forefront in developing new data visualisations to communicate data.
        - Graphic design, conveying information through styling, typography, position, and other aesthetic concerns.
- Data Sources
- Data Security

21. Figure Data Visualisation 3 (Source: Authors)

Harlesden
Willesden Junction
Kilburn
Brondesbury Park
West Hampstead
Belsize Park
Chalk Farm
Camden Town
Camden Road
Caledonian Road
Highbury & Islington
Dalston Kingsland
Clapton
Leyton
Stratford International
Kensal Rise
Brondesbury
Finchley Road
Swiss Cottage
St. John's Wood
Caledonian Road & Barnsbury
Canonbury
Hackney Downs
Hackney Central
Hackney Wick
Maryland
Kensal Green
Queen's Park
Kilburn High Road
South Hampstead
Mornington Crescent
King's Cross St. Pancras
Dalston Junction
Homerton
Stratford
Kilburn Park
Maida Vale
Warwick Avenue
Royal Oak
Westbourne Park
Paddington
Edgware Road
Marylebone
Baker Street
Great Portland Street
Euston
Haggerston
London Fields
Cambridge Heath
Hoxton
Bethnal Green
Stratford High Street
Angel
Warren Street
Euston Square
Edgware Road
Regent's Park
Farringdon
Old Street
Mile End
Pudding Mill Lane
Abbey Road
Ladbroke Grove
Latimer Road
Bayswater
Russell Square
Barbican
Shoreditch High Street
Bow Church
Bow Road
Bromley-by-Bow
West Ham
East Acton
White City
Shepherd's Bush
Holland Park
Notting Hill Gate
Lancaster Gate
Bond Street
Oxford Circus
Goodge Street
Holborn
Chancery Lane
Moorgate
Liverpool Street
Stepney Green
Devons Road
Star Lane
Wood Lane
Queensway
Marble Arch
Tottenham Court Road
St. Paul's
Aldgate East
Whitechapel
Langdon Park
High Street Kensington
Covent Garden
Bank
Aldgate
All Saints
Canning Town
Shepherd's Bush Market
Kensington (Olympia)
Green Park
Leicester Square
Mansion House
Cannon Street
Shadwell
Westferry
Poplar
Hyde Park Corner
Piccadilly Circus
Monument
Tower Hill
Tower Gateway
Limehouse
Blackwall
East India
Goldhawk Road
Knightsbridge
Charing Cross
Blackfriars
Wapping
West India Quay
Barons Court
Gloucester Road
Sloane Square
St. James's Park
Temple
River Thames
Hammersmith
Rotherhithe
Canary Wharf
Ravenscourt Park
West Kensington
Earl's Court
South Kensington
Victoria
Westminster
Embankment
London Bridge
Bermondsey
Canada Water
North Greenwich
Emirates Greenwich Peninsula
West Brompton
Heron Quays
South Quay
Waterloo
Pimlico
Southwark
Borough
Step-free access from street to train
Step-free access from street to platform
Bakerloo
Central
Circle
District
Hammersmith & City
Jubilee
Metropolitan
Northern
Piccadilly
Victoria
Waterloo & City
DLR
Emirates Air Line
Overground
TfL Rail
MAYOR OF LONDON
tfl.gov.uk
0343 222 1234
tfl.gov.uk/emailupdates
@TfLTravelAlerts
UNDERGROUND
TRANSPORT FOR LONDON
EVERY JOURNEY MATTERS
© Transport for London
Improvement works may affect your journey, please check before you travel.
Correct at time of going to print

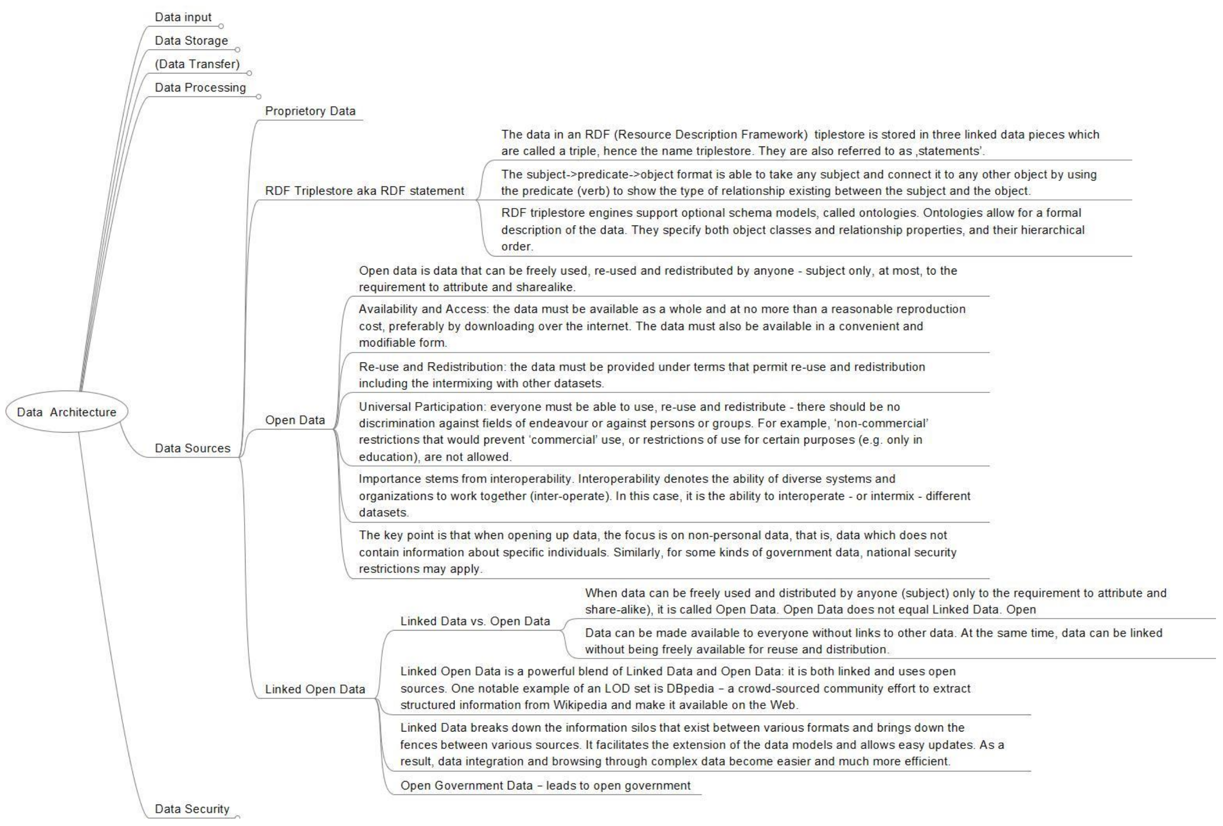
22. Figure Data visualisation example: London Transport (Source: Authors)



Interactive Government Budget (Obama WH Archives)

The key thing here is the fact that a major world power adopted interactive data visualization as way to communicate with taxpayers about where their tax dollars would go. This complicated and obscure topic was made accessible with a simple and clear visualization.

23. Figure Data visualisation example: Interactive Government Budget (Source: Authors)



- Data Architecture
  - Data input
  - Data Storage
  - (Data Transfer)
  - Data Processing
  - Data Sources
    - Proprietary Data
    - RDF Triplestore aka RDF statement
      - The data in an RDF (Resource Description Framework) tiplestore is stored in three linked data pieces which are called a triple, hence the name triplestore. They are also referred to as ‚statements'.
      - The subject->predicate->object format is able to take any subject and connect it to any other object by using the predicate (verb) to show the type of relationship existing between the subject and the object.
      - RDF triplestore engines support optional schema models, called ontologies. Ontologies allow for a formal description of the data. They specify both object classes and relationship properties, and their hierarchical order.
    - Open Data
      - Open data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike.
      - Availability and Access: the data must be available as a whole and at no more than a reasonable reproduction cost, preferably by downloading over the internet. The data must also be available in a convenient and modifiable form.
      - Re-use and Redistribution: the data must be provided under terms that permit re-use and redistribution including the intermixing with other datasets.
      - Universal Participation: everyone must be able to use, re-use and redistribute - there should be no discrimination against fields of endeavour or against persons or groups. For example, 'non-commercial' restrictions that would prevent 'commercial' use, or restrictions of use for certain purposes (e.g. only in education), are not allowed.
      - Importance stems from interoperability. Interoperability denotes the ability of diverse systems and organizations to work together (inter-operate). In this case, it is the ability to interoperate - or intermix - different datasets.
      - The key point is that when opening up data, the focus is on non-personal data, that is, data which does not contain information about specific individuals. Similarly, for some kinds of government data, national security restrictions may apply.
    - Linked Open Data
      - Linked Data vs. Open Data
        - When data can be freely used and distributed by anyone (subject) only to the requirement to attribute and share-alike), it is called Open Data. Open Data does not equal Linked Data. Open
        - Data can be made available to everyone without links to other data. At the same time, data can be linked without being freely available for reuse and distribution.
      - Linked Open Data is a powerful blend of Linked Data and Open Data: it is both linked and uses open sources. One notable example of an LOD set is DBpedia – a crowd-sourced community effort to extract structured information from Wikipedia and make it available on the Web.
      - Linked Data breaks down the information silos that exist between various formats and brings down the fences between various sources. It facilitates the extension of the data models and allows easy updates. As a result, data integration and browsing through complex data become easier and much more efficient.
      - Open Government Data – leads to open government
  - Data Security

24. Figure Data Sources (Source: Authors)

5.3.4 What section(s) of the BIS curriculum are covered

Knowledge and understanding of the factors that have led to 'data explosion' and the consequences for the management of data. A broad knowledge and understanding of the key attributes and value of data and the scope and services associated with managing data in an enterprise.

Students will be able to design and develop a database application using relational modelling (ERD) and SQL database systems. Specifically:

- demonstrate a broad knowledge of the how database management system (DBMS) software can be used to organise and secure a valuable business asset – namely data;
- demonstrate a broad knowledge of the stages of the database system development lifecycle which enables the creation of a database system to meet the requirements of users;
- demonstrate understanding of the core theories and principles associated with the relational data model;
- use a range of routine skills and techniques to produce a conceptual and logical design for a database;
- use of a range of standard functions and services provided by a relational database management system (DBMS) to implement a prototype database system.

They will also be capable of performing basic activities of database management as well as simple tasks of data migration.

6 TECHNOLOGY ARCHITECTURE

6.1 Enterprise Architecture Development

1. Technology architecture starts with the integration of existing or to be acquired hardware and software components with the applications outlined in the application architecture
    a. the physical implementation of the architecture
    b. a close link with implementation and migration planning
    c. baseline and target architecture, roadmap
    d. identification of work packages and costing
2. Input from the architecture repository
    a. IT services documentation, catalogue
    b. TOGAF and generic or sector-specific reference model
    c. Product information
    d. Scope, maturity assessment, gap analysis and resolution, constraints, budget, governance and support
    e. Tailored architecture development methodology, products and building blocks deployed services and tools
    f. Non-architectural input: capability assessment, communication plan
3. Process
    a. Detailing of building elements according to scope, objectives, identification of new elements, possible redefinition of existing elements
    b. Selection of reference model and perspectives
        i. clarification of technology application principle
        ii. resource allocation (business considerations, stakeholder interests)
        iii. define perspectives (how business objectives, stakeholder perspectives are validated) - model selection, taxonomy, localisation
        iv. business requirements, application of fit-for-purpose, configuration and impact analysis
        v. service specification and boundaries, performance, sustainability, localisation and latency, availability, product selection and reusability
        vi. deviations from standards or risks identified in the impact analysis should be addressed in the options and solutions phase
        vii. choice of tools and techniques (size, cost, capacity)
4. Technology architecture: models, perspectives and tools
    a. Technology catalogue: product list (portfolio), consideration of standards, structure follows meta-model attributes
    b. Diagram-based visualisation
        i. explicit exploration of relationships between model entities, elements are accompanied by development plan (diagram) and impact analysis, diagrams represent different perspectives
        ii. alignment of platform and hosting requirements
        iii. stack diagrams: hardware, operating system, software infrastructure, application packages
        iv. logic diagrams: hardware and software environment, communication links, capacity data

            v. environment dependencies, platform decomposition, network and communication
    c. Data-driven requirements specification (assuming target architecture)
    d. Engagement includes: functional and non-functional requirements, assumptions, constraints, technology principles, policies, recommendations, standards
    e. Services from a catalogue, conflict-free

5. Baseline architecture description - granularity proportional to the scope
6. Description of target architecture - granularity proportional to scope and conceptual description of building blocks (functionality is the determinant)
7. Gap analysis after analysis of consistency and accuracy of the architecture model, qualification and classification of gaps, definition of new elements, decision on development and/or procurement
8. Assignment and prioritisation of roadmap components
9. Impact analysis: what works and for what? (Whether it affects the existing architecture, whether another area of the organisation or a project running elsewhere has an impact on the architecture, and vice versa.)
10. Stakeholder analysis (motivation, fit-for-purpose), clarification if necessary
11. Finalisation: documentation, cross-checks, traceability, comparison with architecture repository (reuse), publication.
12. Technology architecture development outputs

6.2 Case study steps and/or output

The expected output is:

- Refined and updated versions of the Architecture Vision phase deliverables, where applicable, Statement of Architecture Work, updated if necessary
- Validated technology **principles**, or new technology principles (if generated here)

Draft Architecture Definition Document

- Target Technology Architecture, Version 1.0 (detailed), including:
    - Technology **Components** and their relationships to information systems
    - Technology **platforms** and their decomposition show the technology combinations required to realize a particular technology ''stack.''
    - **Environments and locations** — a grouping of the required technology into computing environments (e.g., development, production)
    - Expected **processing load** and distribution of load across technology components
    - Physical (network) **communications**
    - Hardware and network **specifications**
- Baseline Technology Architecture, Version 1.0 (detailed), if appropriate
- Views corresponding to the **selected viewpoints** addressing key stakeholder concerns

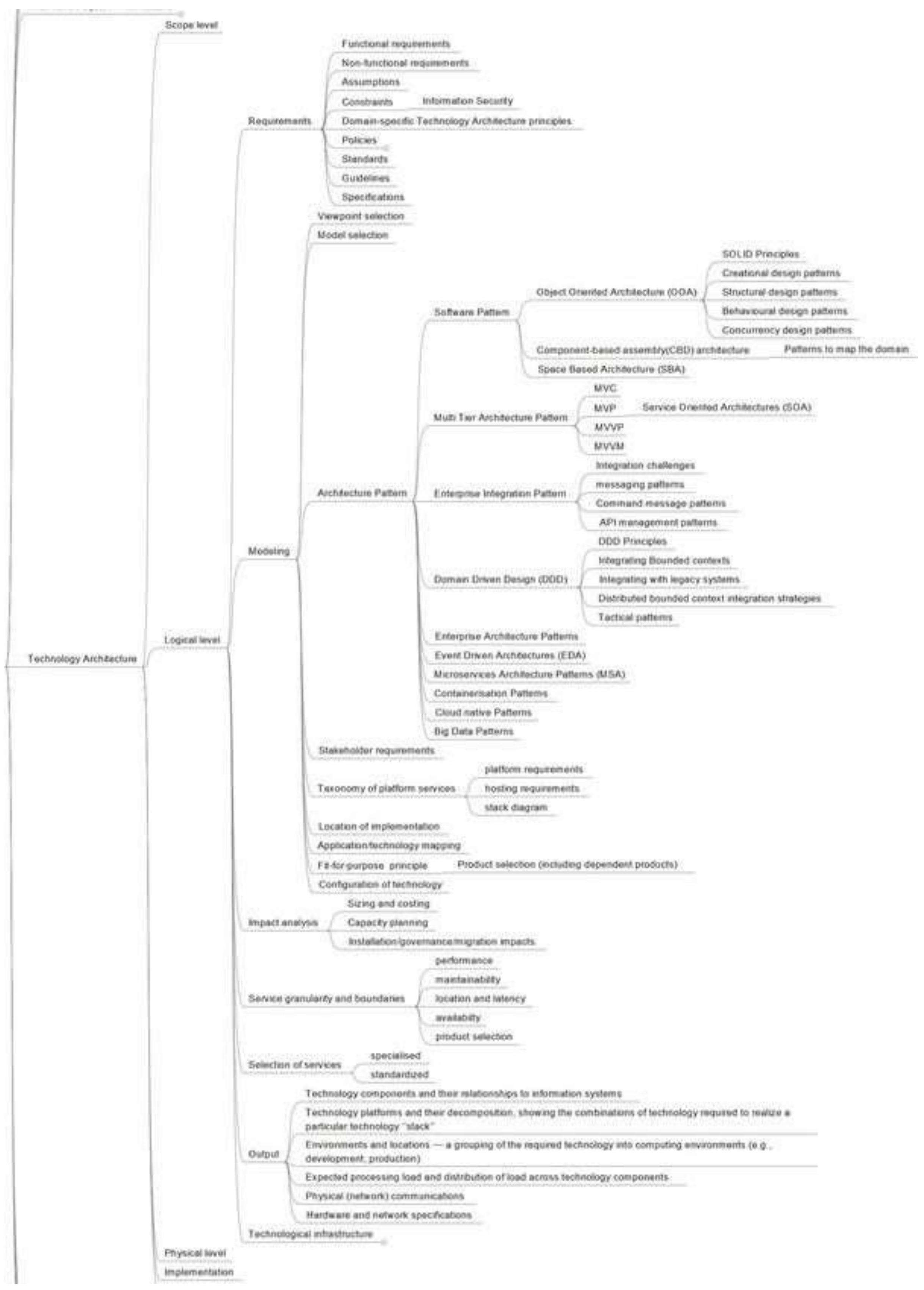
Draft Architecture Requirements Specification

- Technology Architecture requirements as gap analysis results, requirements output from Phases B and C and updated technology requirements
- Technology Architecture components of an Architecture Roadmap

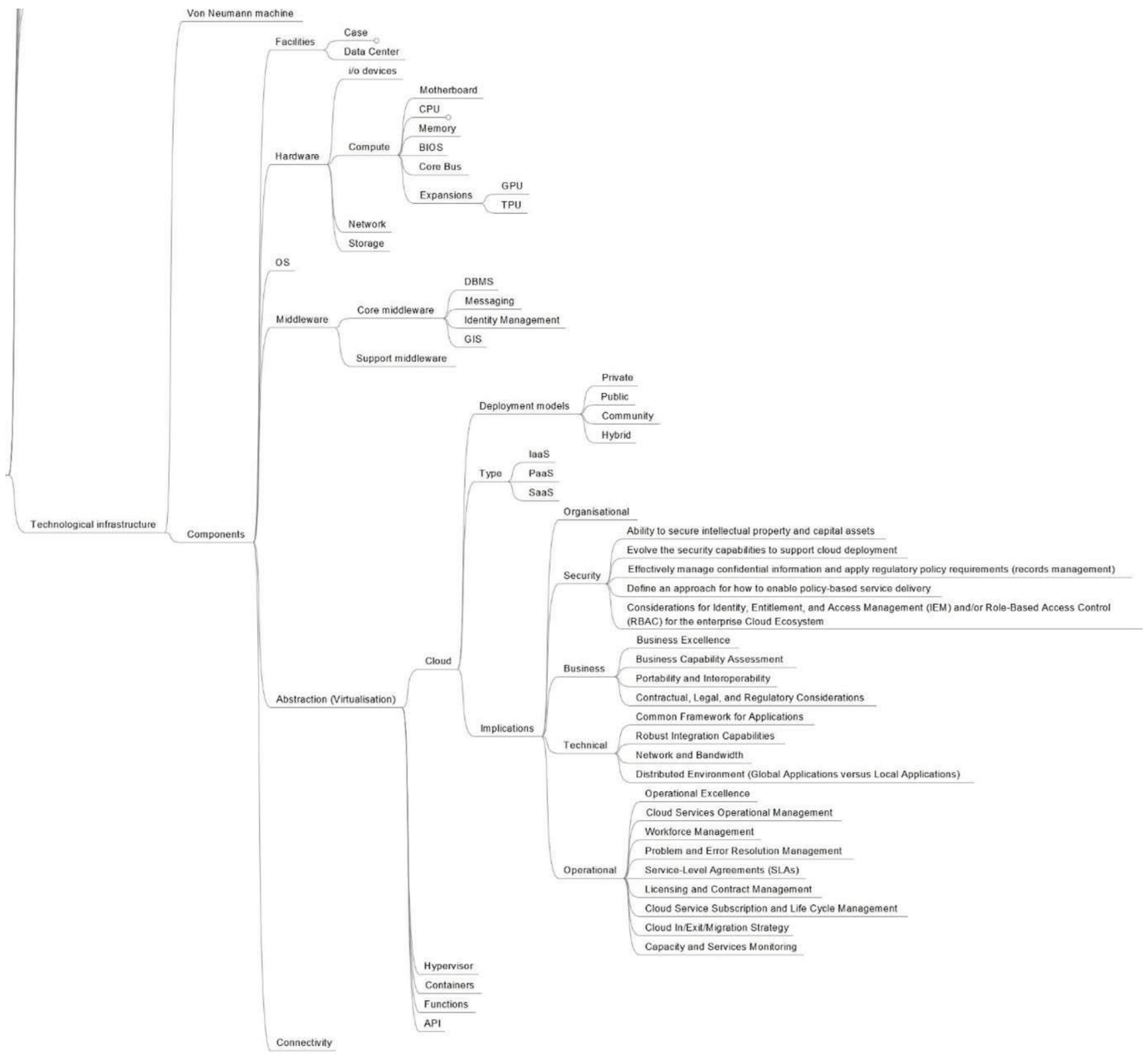
6.3 Need to know to perform the step

This is a highly technical part, but it is worth mentioning the different service models (in-house, cloud, client/server, ASP, etc.). However, C/B analysis, security aspects, dependency analysis should be covered here. It is also worth showing industry reference models, not just the advantages and disadvantages of platforms.

Topics to highlight (avoid excessive technical detail): *engagement* - this should be included in the strategy section but can also be filled with concrete content; *gap analysis* and *granularity* - mainly approached from the perspective of transit architectures but also based on maturity models, mainly due to cloud computing, can be well discussed in combination with *scalability*; *roadmap* and *project management* (mainly highlighting resource allocation and scheduling aspects) and in combination with *impact analysis*, the latter making sense together with *stakeholder analysis*.


Technology Architecture
Scope level
Logical level
Requirements
Functional requirements
Non-functional requirements
Assumptions
Constraints
Information Security
Domain-specific Technology Architecture principles
Policies
Standards
Guidelines
Specifications
Modeling
Viewpoint selection
Model selection
Architecture Pattern
Software Pattern
Object Oriented Architecture (OOA)
SOLID Principles
Creational design patterns
Structural design patterns
Behavioural design patterns
Concurrency design patterns
Component-based assembly(CBD) architecture
Patterns to map the domain
Space Based Architecture (SBA)
Multi Tier Architecture Pattern
MVC
MVP
Service Oriented Architectures (SOA)
MVVP
MVVM
Enterprise Integration Pattern
Integration challenges
messaging patterns
Command message patterns
API management patterns
Domain Driven Design (DDD)
DDD Principles
Integrating Bounded contexts
Integrating with legacy systems
Distributed bounded context integration strategies
Tactical patterns
Enterprise Architecture Patterns
Event Driven Architectures (EDA)
Microservices Architecture Patterns (MSA)
Containerisation Patterns
Cloud native Patterns
Big Data Patterns
Stakeholder requirements
Taxonomy of platform services
platform requirements
hosting requirements
stack diagram
Location of implementation
Application/technology mapping
Fit-for-purpose principle
Product selection (including dependent products)
Configuration of technology
Impact analysis
Sizing and costing
Capacity planning
Installation/governance/migration impacts
Service granularity and boundaries
performance
maintainability
location and latency
availability
product selection
Selection of services
specialised
standardized
Output
Technology components and their relationships to information systems
Technology platforms and their decomposition, showing the combinations of technology required to realize a particular technology "stack"
Environments and locations — a grouping of the required technology into computing environments (e.g., development, production)
Expected processing load and distribution of load across technology components
Physical (network) communications
Hardware and network specifications
Technological infrastructure
Physical level
Implementation

25. Figure Technology Architecture (Source: Authors)


Technological infrastructure
Von Neumann machine
Components
Facilities
Case
Data Center
Hardware
i/o devices
Compute
Motherboard
CPU
Memory
BIOS
Core Bus
Expansions
GPU
TPU
Network
Storage
OS
Middleware
Core middleware
DBMS
Messaging
Identity Management
GIS
Support middleware
Abstraction (Virtualisation)
Cloud
Deployment models
Private
Public
Community
Hybrid
Type
IaaS
PaaS
SaaS
Implications
Organisational
Security
Ability to secure intellectual property and capital assets
Evolve the security capabilities to support cloud deployment
Effectively manage confidential information and apply regulatory policy requirements (records management)
Define an approach for how to enable policy-based service delivery
Considerations for Identity, Entitlement, and Access Management (IEM) and/or Role-Based Access Control (RBAC) for the enterprise Cloud Ecosystem
Business
Business Excellence
Business Capability Assessment
Portability and Interoperability
Contractual, Legal, and Regulatory Considerations
Technical
Common Framework for Applications
Robust Integration Capabilities
Network and Bandwidth
Distributed Environment (Global Applications versus Local Applications)
Operational
Operational Excellence
Cloud Services Operational Management
Workforce Management
Problem and Error Resolution Management
Service-Level Agreements (SLAs)
Licensing and Contract Management
Cloud Service Subscription and Life Cycle Management
Cloud In/Exit/Migration Strategy
Capacity and Services Monitoring
Hypervisor
Containers
Functions
API
Connectivity

26. Figure Technology infrastructure (Source: Authors)

6.4 What section(s) of the BIS curriculum are covered

Students will be able to understand the technology architecture and its resources including hardware, software, and middleware. They will be able to explain the differences including advantages and issues of the different types of technology architecture such as stand alone, server-based, N-gears, Virtualized, Cloud-based, and Edge computing. Understand the types of Clouds – a) Private, Public, and Global; b) based on level of service (IaaS, PaaS, SaaS, etc.).

Understand requirements for facilities, computer hardware, connectivity and computation delivery, hardware abstraction, operating systems, middleware, and API. Be able to explain how data is stored.

Students will be familiar with concepts of computation services, Computer Networking, Encryption, Storage services, Virtual Machines, Kubernetes, DBMS, Messaging and Queueing, Firewall, as well as

Security principles, an presentation types. Knowledge and understanding of the main theories and principles associated with the relational data model and language (i.e. SQL). Knowledge and understanding of concepts associated with database design techniques (e.g. entity-relationship (ER) modelling and normalisation).

7 EDOER.EU

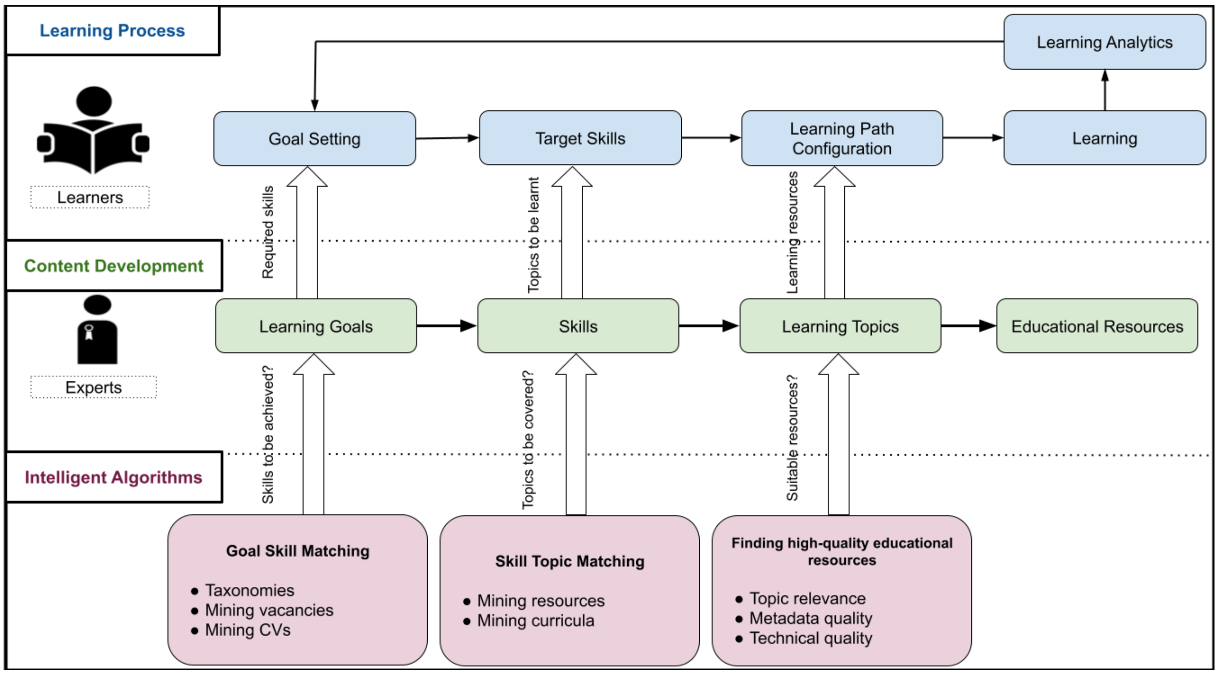
EDOER is A *research project* run by Technische Informationsbibliothek (TIB) and University of Amsterdam (UvA).

This research project departs from the recent, dramatic changes in global societies, forcing many citizens to re-skill themselves to (re)gain employment. Therefore, learners need to be informed about their skills to achieve in their current/future jobs. Subsequently, high-quality, personalized educational content and services are also essential to serve this high demand for learning. Free and Open Educational Resources can play a key role in this regard, as they are available in a wide range of learning and occupational contexts globally. However, their applicability so far has been limited due to low metadata quality and complex quality control. These issues resulted in a lack of personalised functions, like recommendation and search.

7.1 eDoer General Description

eDoer is aiming at empowering learners through open, personalised learning and curriculum recommendations, on the basis of up to date labour market information and freely available online Open Educational Resources (OERs). In order to meet this objective, eDoer utilizes cutting edge AI technology (see section on eDoer methodology). Learners in eDoer interact with a learning content recommender through a dashboard, in which they can set their own learning objectives, display the list of required skills and learning topics, set their learning context, and receive relevant OER recommendations accordingly. During their learning process, learners can also rate their satisfaction with recommendations, and update their learning preferences. This strategy detects changes in learner profiles, and fine tune the precision of recommendations.

eDoer is strongly being built on open science principles, and it is one of the few new generation learning environments, which 1) empower learners to take control and responsibility for their own skill development, 2) improve skills on the basis of labour market information and personalised OER recommendations, and 3) foster the uptake and reuse of OERs through a systematic, hybrid (human-AI) quality control mechanism. eDoer is a community based application, meaning that it is available for a very wide range of learners globally (intelligent functions need a free registration), and both the technical and curriculum development sides are available for any professionals and professional communities. The Open Science basis also means that the source code and the algorithms are available as open source software. eDoer is hosted by TIB, which is part of the official German national research infrastructure.

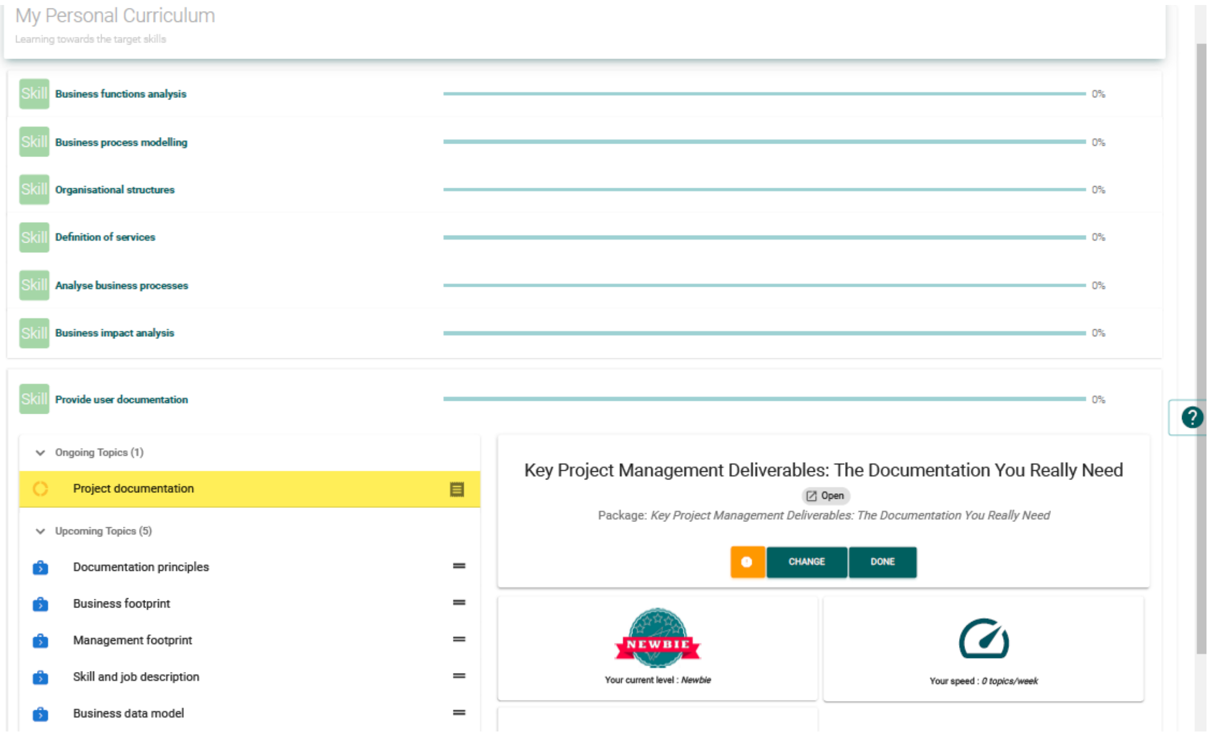


27. Figure The eDoer recommendation concept (Source: eDoer)

7.2 eDoer Methodology

eDoer is based on a community based open software, on openly available education and knowledge resources and AI driven recommendation algorithms, aiding both learners and educational content curators. In general, human experts together with AI are co-curating knowledge (educational content), organizing this content in a four level (learning goal, skill, learning topic and educational resources) curricula, and making these curricula available for learners. This logic is depicted in Figure 27 and works as follows:

On the learner side, learners need to set their learning goals. These goals are translated to individual skill or learning topic targets, and build up individual learning pathways. Furthermore, based on the user profile, which contains contextual information about the learning (e.g. geographical location, or learning content related personal preferences), information on previous learning activities on the eDoer platform, and the user's behavioural data (e.g. what content the user selects) eDoer's AI algorithm recommends learning content for the user. On the learner dashboard, the user can follow its progress towards learning goals, and fine tune the personal curricula based on the feedback . In case learners face problems they can't resolve themselves, they can request a (human) mentor to discuss their problems. On Figure 28 learners can navigate on the left among their target skills (green) and underlying topics (blue). On the right, a topic relevant open learning content recommendation is visible.


My Personal Curriculum
Learning towards the target skills

Skill	Business functions analysis	0%
Skill	Business process modelling	0%
Skill	Organisational structures	0%
Skill	Definition of services	0%
Skill	Analyse business processes	0%
Skill	Business impact analysis	0%
Skill	Provide user documentation	0%

Ongoing Topics (1)
- Project documentation

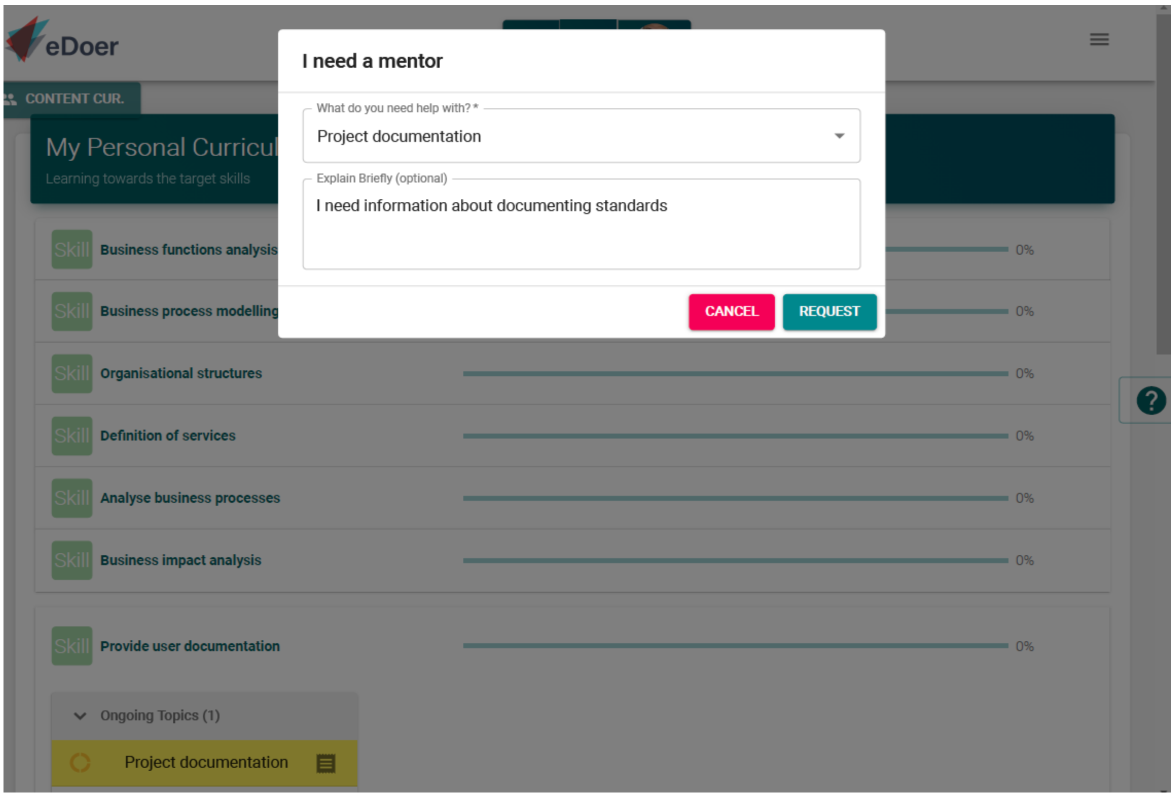
Upcoming Topics (5)
- Documentation principles
- Business footprint
- Management footprint
- Skill and job description
- Business data model

Key Project Management Deliverables: The Documentation You Really Need
Open
Package: *Key Project Management Deliverables: The Documentation You Really Need*
CHANGE DONE

Your current level : *Newbie*
Your speed : *0 topics/week*

28. Figure The eDoer learner interface (Source: eDoer)

Then, if needed, they can ask for mentoring help (see Figure 29): in case learners need the help of a human expert, they can request help in relation to a specific skill or a specific topic (here: project documentation).


I need a mentor

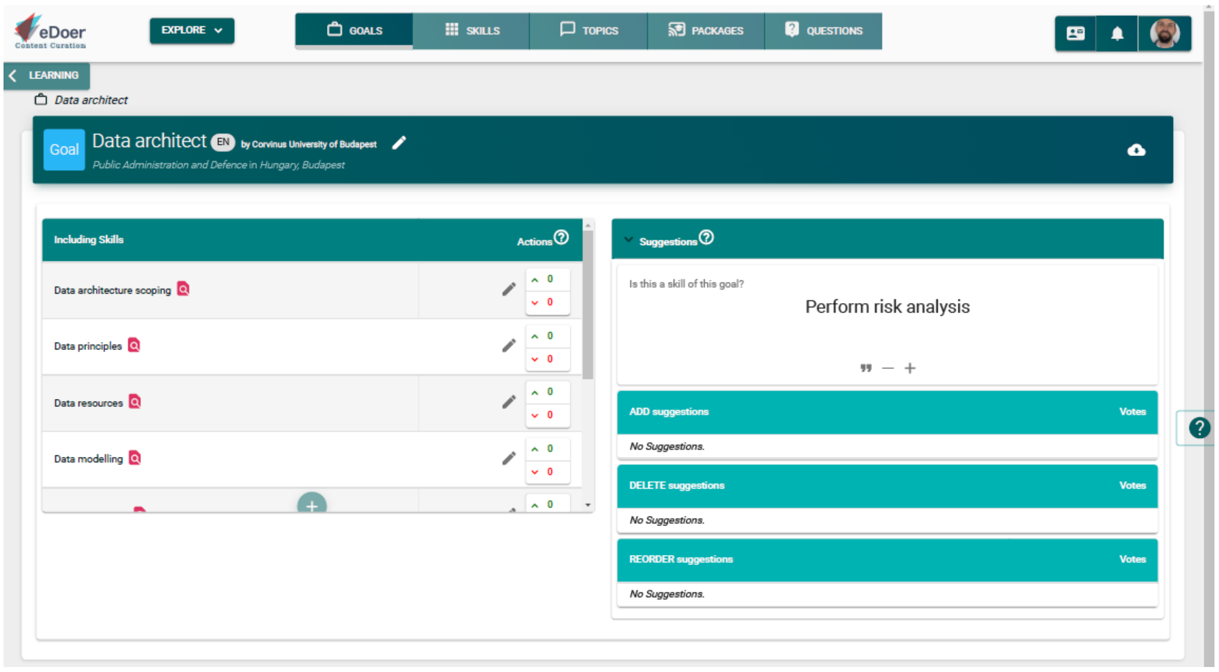
What do you need help with? *: Project documentation

Explain Briefly (optional): I need information about documenting standards

CANCEL REQUEST

29. Figure Mentoring request in eDoer. (Source: eDoer)

On the content curator side the AI is working under the hands of domain experts. When creating a curriculum according to the four levels (learning goal, skill, learning topic and educational resources), the AI constantly recommends elements for the curriculum. These recommendations are based on information coming from labour market related databases and taxonomies (e.g. ESCO or vacancy announcements), text mining of openly available educational curricula and educational content. When it comes to educational content the AI also performs an initial quality assessment in terms of technical quality (e.g. video or sound quality of audiovisual content), metadata quality and relevance to learning topics. Human experts are confronted with content, which have passed this quality check, to decide what content or concept they include in the curriculum. Content creator recommendations demonstrated in Figures 30-32.



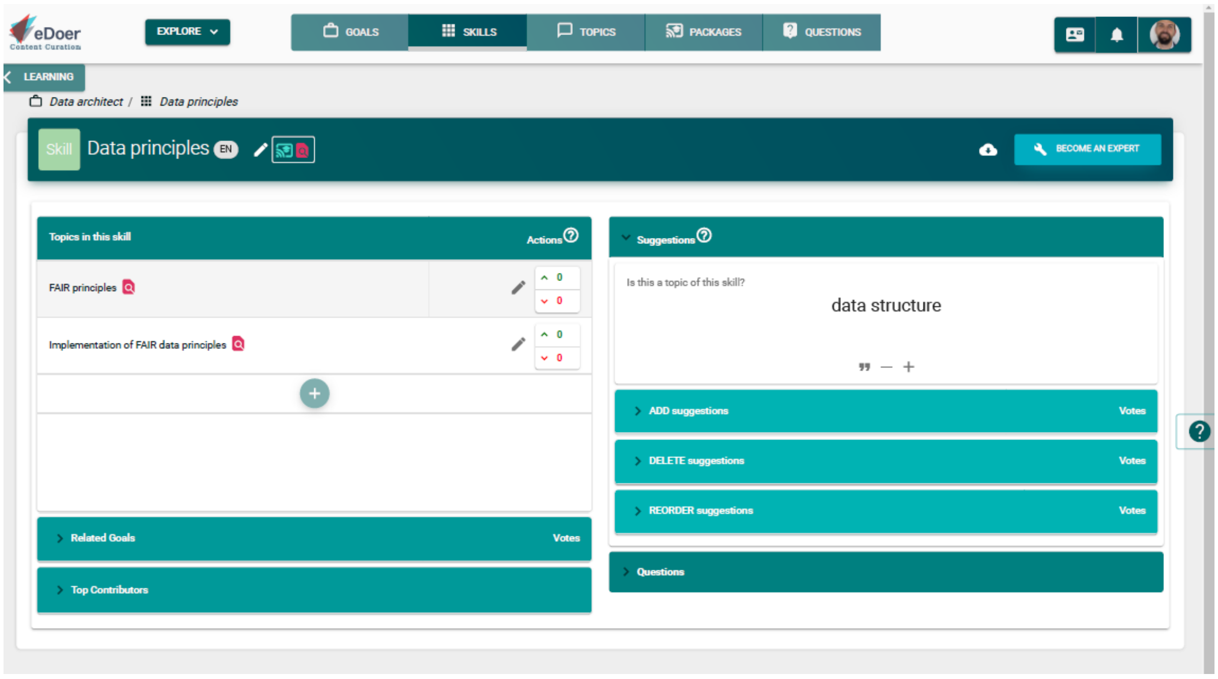
Data architect by Corvinus University of Budapest
Public Administration and Defence in Hungary, Budapest

Including Skills	Actions
Data architecture scoping	0 0
Data principles	0 0
Data resources	0 0
Data modelling	0 0

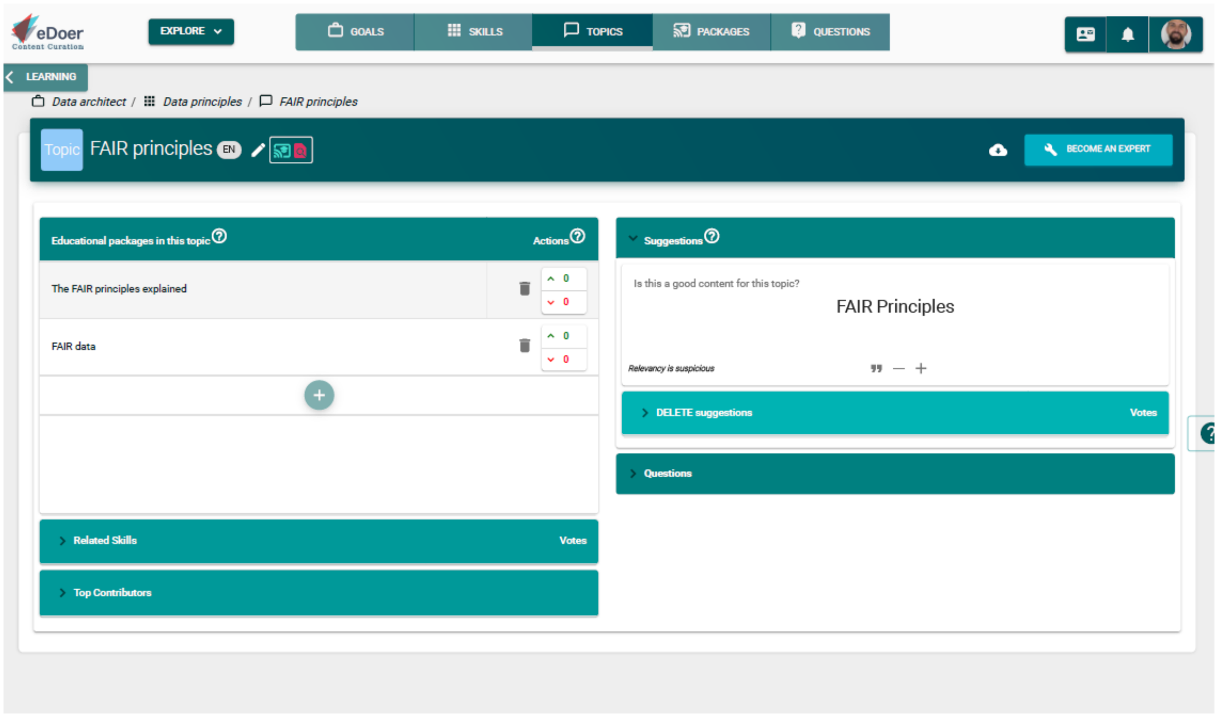
Is this a skill of this goal?
Perform risk analysis

ADD suggestions
No Suggestions.
DELETE suggestions
No Suggestions.
REORDER suggestions
No Suggestions.

30. Figure eDoer Skill recommendations for learning goals on the curriculum designer interface (Source: eDoer)

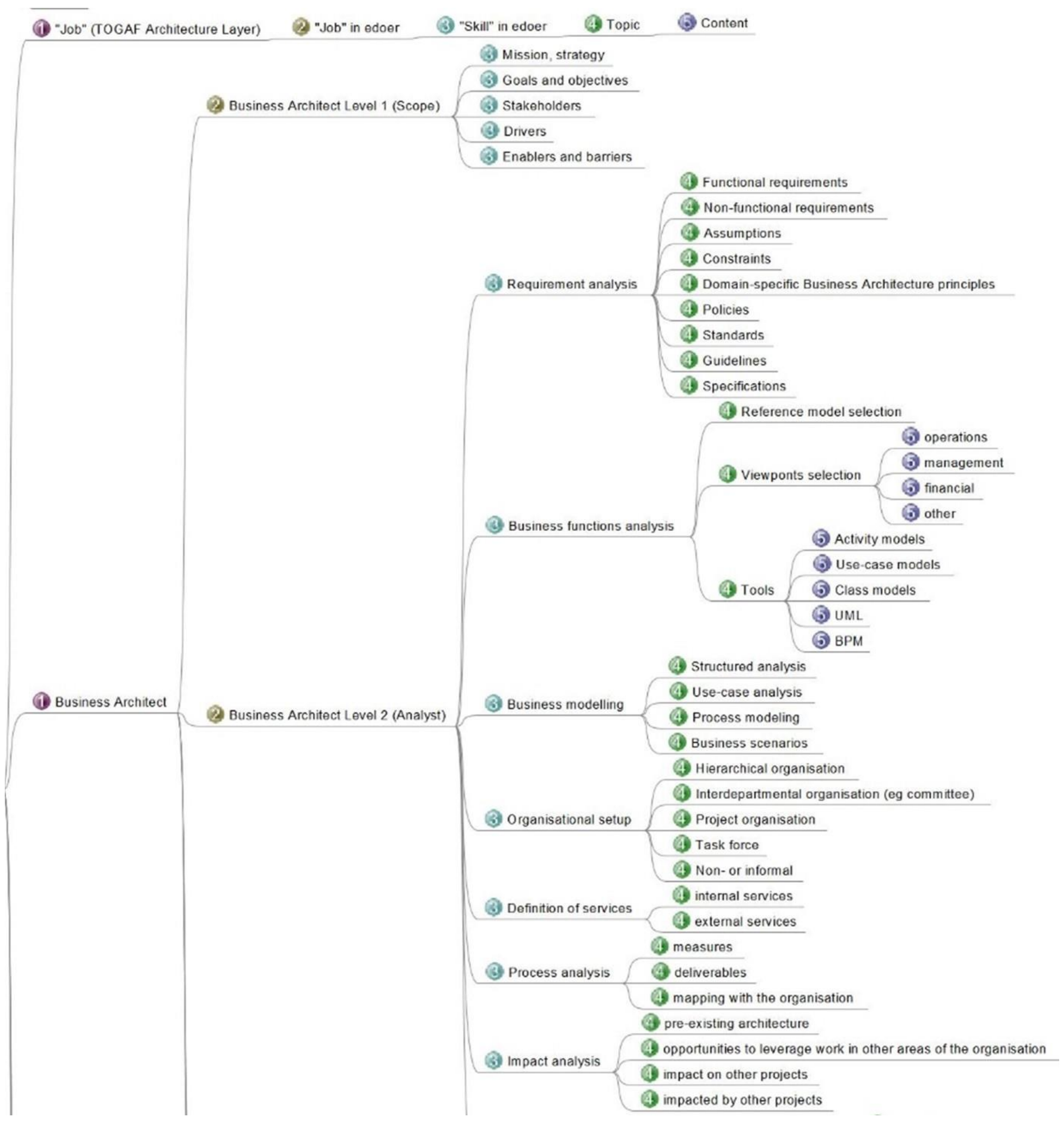


31. Figure eDoer topic recommendations for skills on the curriculum designer interface (Source: eDoer)

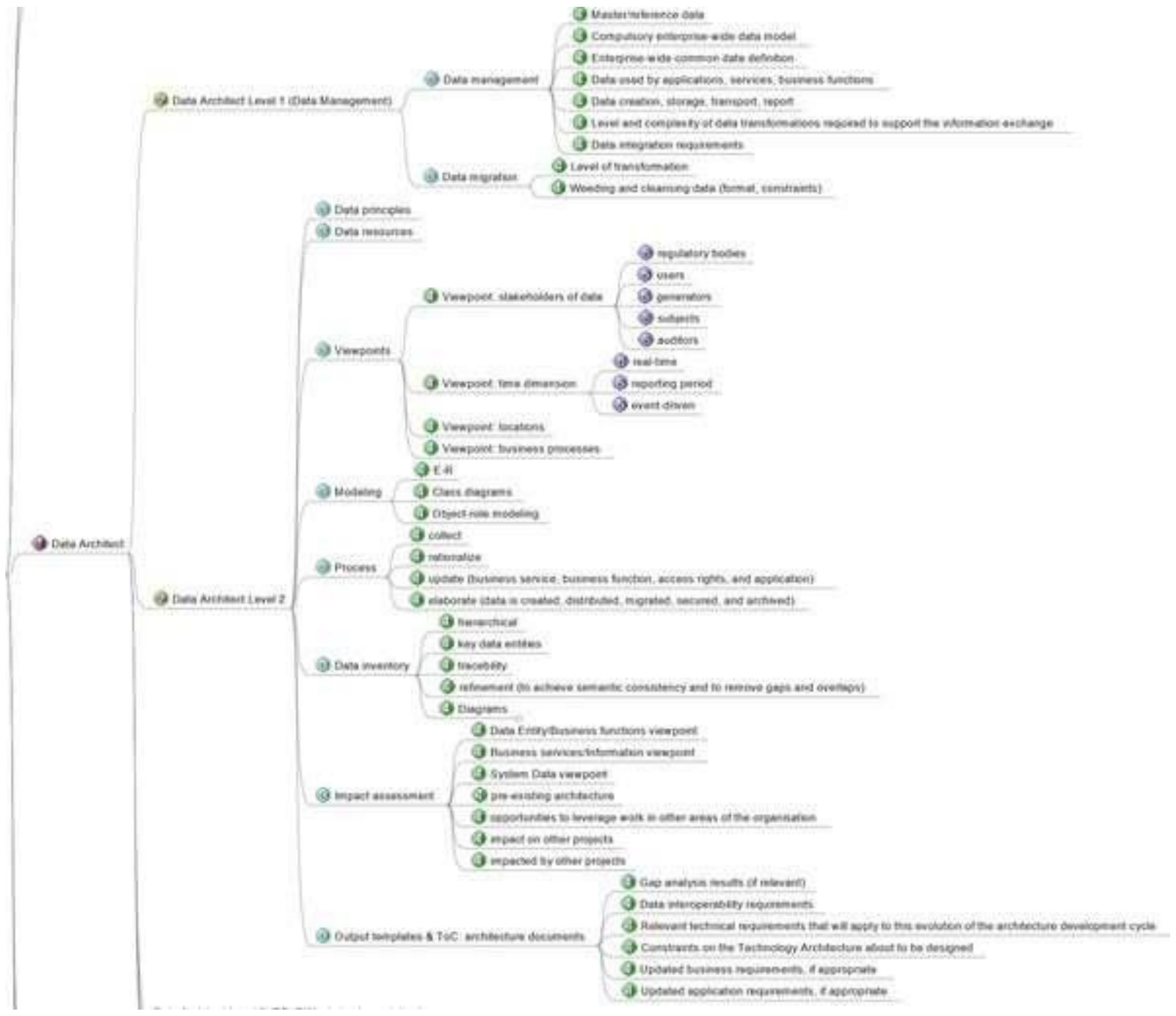


32. Figure eDoer learning content recommendations for topics on the curriculum designer interface (Source: eDoer)

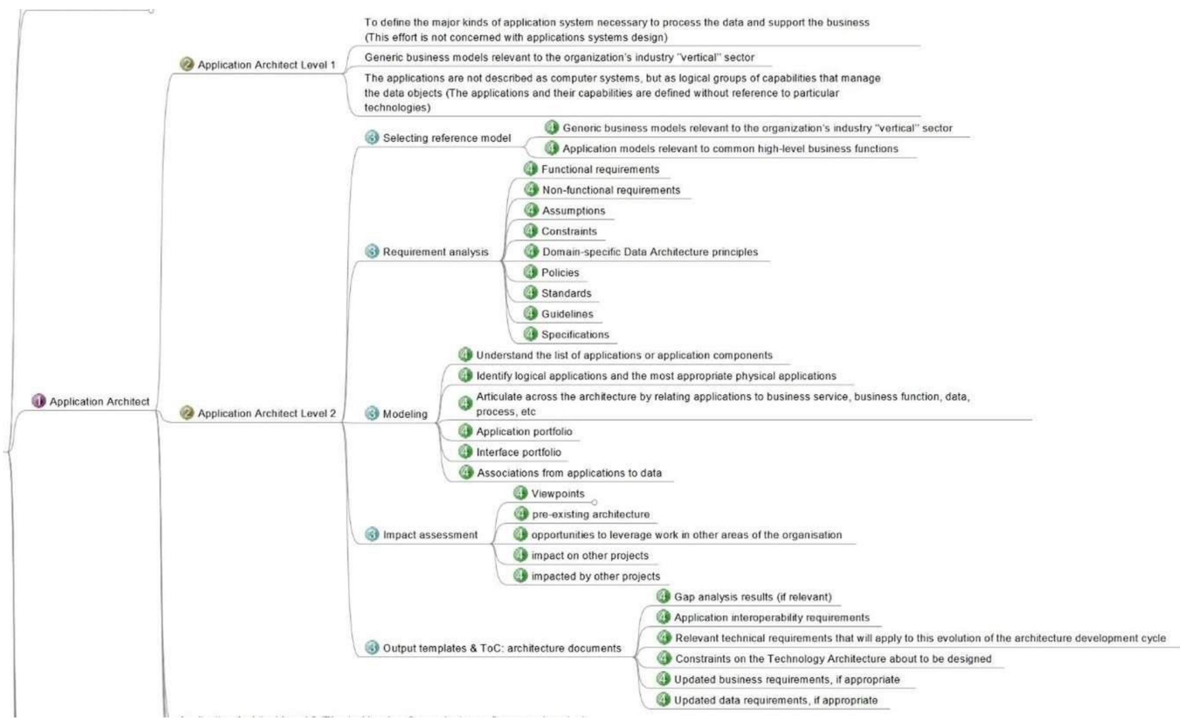
Figures 33 to 36 show actual BIPER content uploaded for roles specifically added for BIPER.



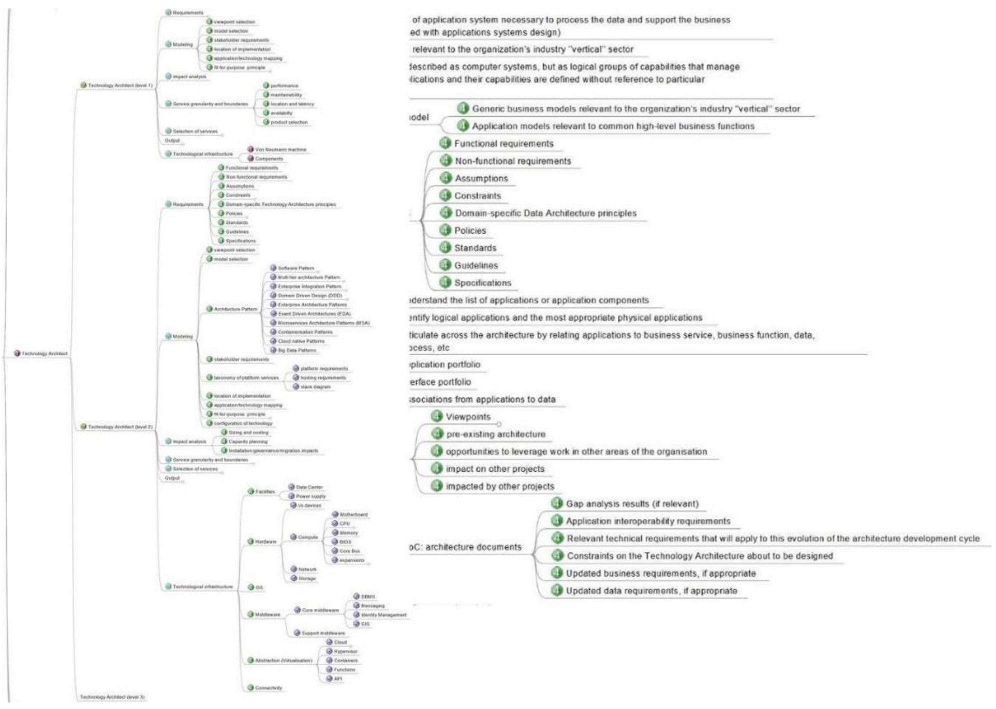
33. Figure Business architect (Source: Authors)


Data Architect
Data Architect Level 1 (Data Management)
Data management
Master/reference data
Compulsory enterprise-wide data model
Enterprise-wide common date definition
Data used by applications, services, business functions
Data creation, storage, transport, report
Level and complexity of data transformations required to support the information exchange
Data integration requirements
Data migration
Level of transformation
Weeding and cleansing data (format, constraints)
Data Architect Level 2
Data principles
Data resources
Viewpoints
Viewpoint: stakeholders of data
regulatory bodies
users
generators
subjects
auditors
Viewpoint: time dimension
real-time
reporting period
event-driven
Viewpoint: locations
Viewpoint: business processes
Modeling
E-R
Class diagrams
Object-role modeling
Process
collect
rationalize
update (business service, business function, access rights, and application)
elaborate (data is created, distributed, migrated, secured, and archived)
Data inventory
hierarchical
key data entities
traceability
refinement (to achieve semantic consistency and to remove gaps and overlaps)
Diagrams
Impact assessment
Data Entity/Business functions viewpoint
Business services/Information viewpoint
System Data viewpoint
pre-existing architecture
opportunities to leverage work in other areas of the organisation
impact on other projects
impacted by other projects
Output templates & ToC: architecture documents
Gap analysis results (if relevant)
Data interoperability requirements
Relevant technical requirements that will apply to this evolution of the architecture development cycle
Constraints on the Technology Architecture about to be designed
Updated business requirements, if appropriate
Updated application requirements, if appropriate

34. Figure Data architect (Source**:** Authors)


Application Architect
Application Architect Level 1
To define the major kinds of application system necessary to process the data and support the business (This effort is not concerned with applications systems design)
Generic business models relevant to the organization's industry "vertical" sector
The applications are not described as computer systems, but as logical groups of capabilities that manage the data objects (The applications and their capabilities are defined without reference to particular technologies)
Application Architect Level 2
Selecting reference model
Generic business models relevant to the organization's industry "vertical" sector
Application models relevant to common high-level business functions
Requirement analysis
Functional requirements
Non-functional requirements
Assumptions
Constraints
Domain-specific Data Architecture principles
Policies
Standards
Guidelines
Specifications
Modeling
Understand the list of applications or application components
Identify logical applications and the most appropriate physical applications
Articulate across the architecture by relating applications to business service, business function, data, process, etc
Application portfolio
Interface portfolio
Associations from applications to data
Impact assessment
Viewpoints
pre-existing architecture
opportunities to leverage work in other areas of the organisation
impact on other projects
impacted by other projects
Output templates & ToC: architecture documents
Gap analysis results (if relevant)
Application interoperability requirements
Relevant technical requirements that will apply to this evolution of the architecture development cycle
Constraints on the Technology Architecture about to be designed
Updated business requirements, if appropriate
Updated data requirements, if appropriate

35. Figure Application architect (Source: Authors)



36. Figure Technology architect (Source: Authors)