



Warwick
Business
School

Data Science & Generative AI

Dr Michael Mortenson
Associate Professor (Reader)

michael.mortenson@wbs.ac.uk



Session 2: Data Management for AI and Data Science

1.1 Why Data?



Image source (creative commons): https://morningstaronline.co.uk/sites/default/files/styles/article_full/public/zbynek-burival-GrmwVnVSSdU-unsplash.jpg?itok=EObuKnRY&c=ef17667d17ca75f0f9b22106b473877d

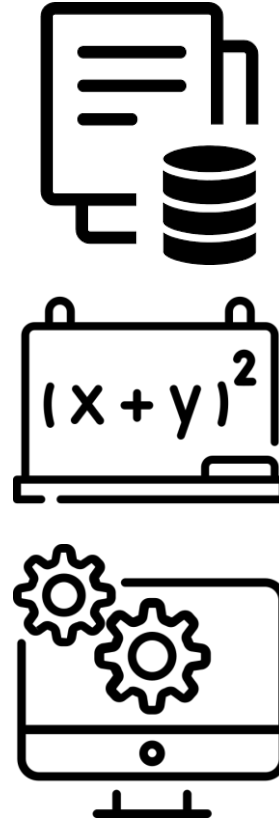
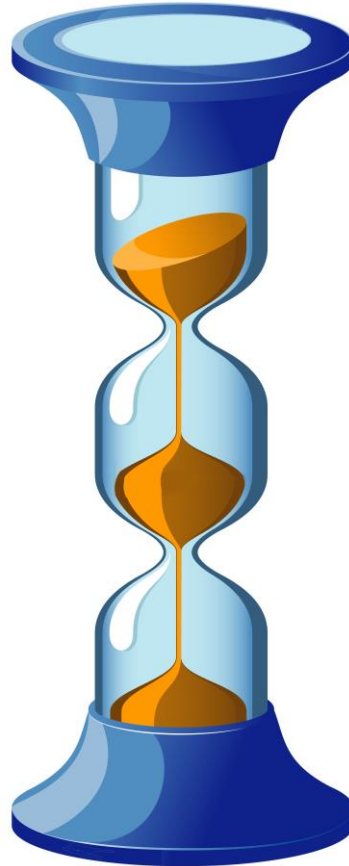
1.2 A Data Supply Chain



**Data Processing,
Engineering & Storage**

**Data Analytics and
Modelling**

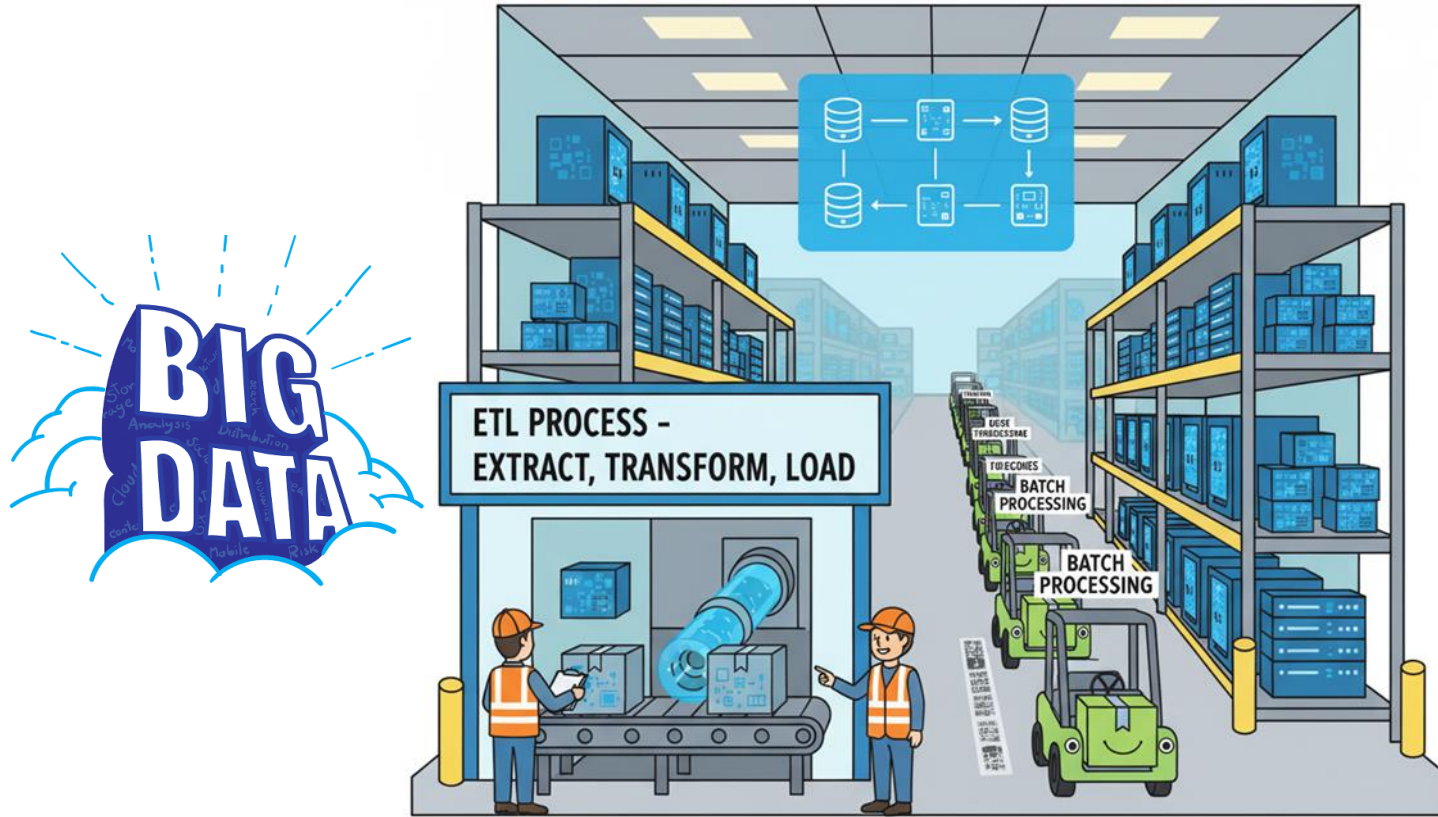
**Implementation /
Decision Making**



1.3 From Data to Features

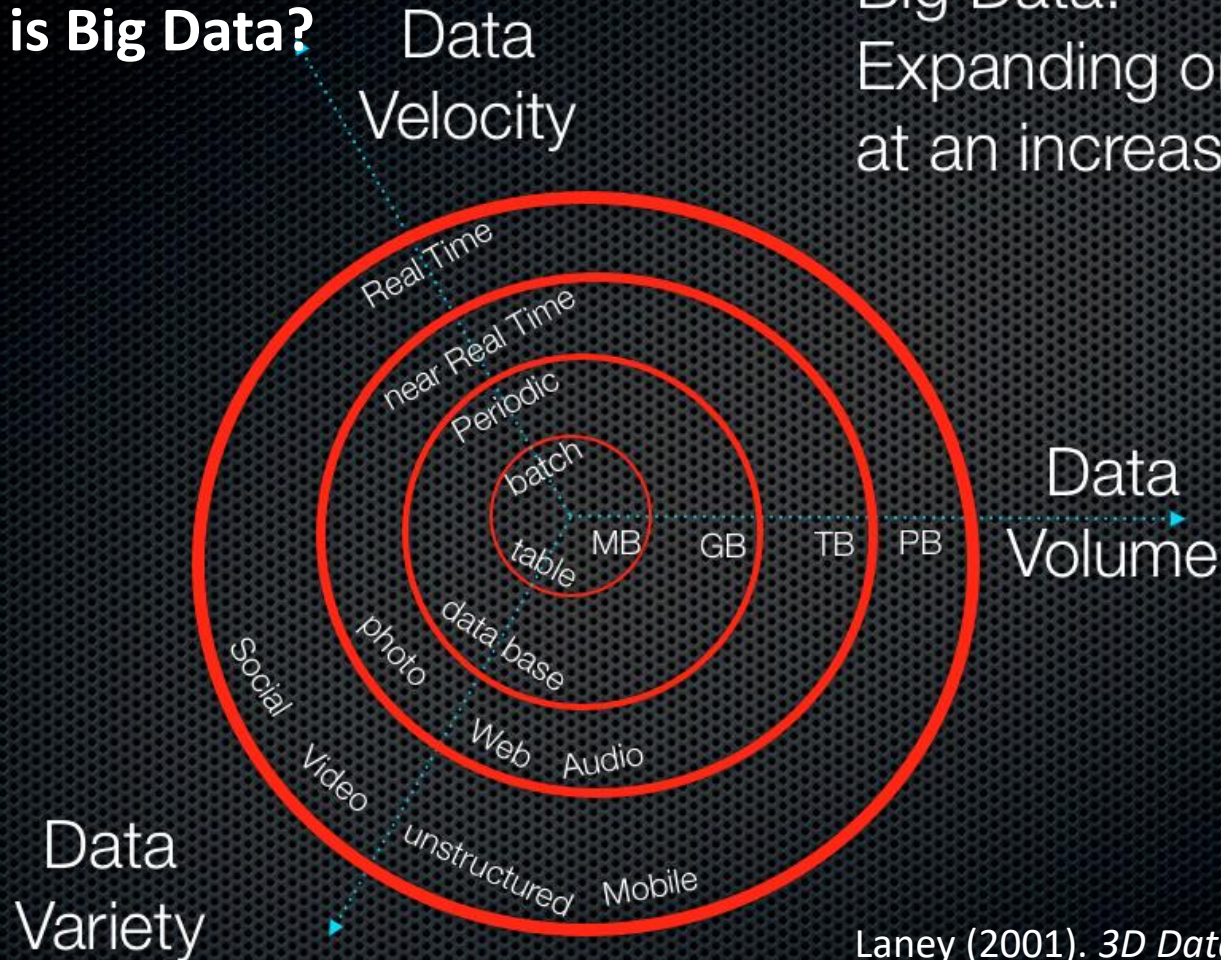
“Feature engineering is the process of transforming raw data into relevant information for use by machine learning models. [...] Because model performance largely rests on the quality of data used during training, feature engineering is a crucial preprocessing technique that requires selecting the most relevant aspects of raw training data for both the predictive task and model type under consideration”

1.4 Data Architecture: The challenge



1.5 What is Big Data?

Big Data:
Expanding on 3 fronts
at an increasing rate.



Laney (2001). 3D Data Management.

1.6 “Traditional” vs “Big Data”

“Traditional” Data

- *“Structured” data, typically quantitative data that can fit in spreadsheets and (relational) databases.*
- *Often collected specifically for the analysis task or by transactional systems.*



“Big Data”

- *Digital data, typically data that was too large, non-numeric and/or with real-time demands, making them ill suited for traditional data storage tools.*
- *Often collected as a side effect of digitalisation or via sensors and devices (Internet of Things)*

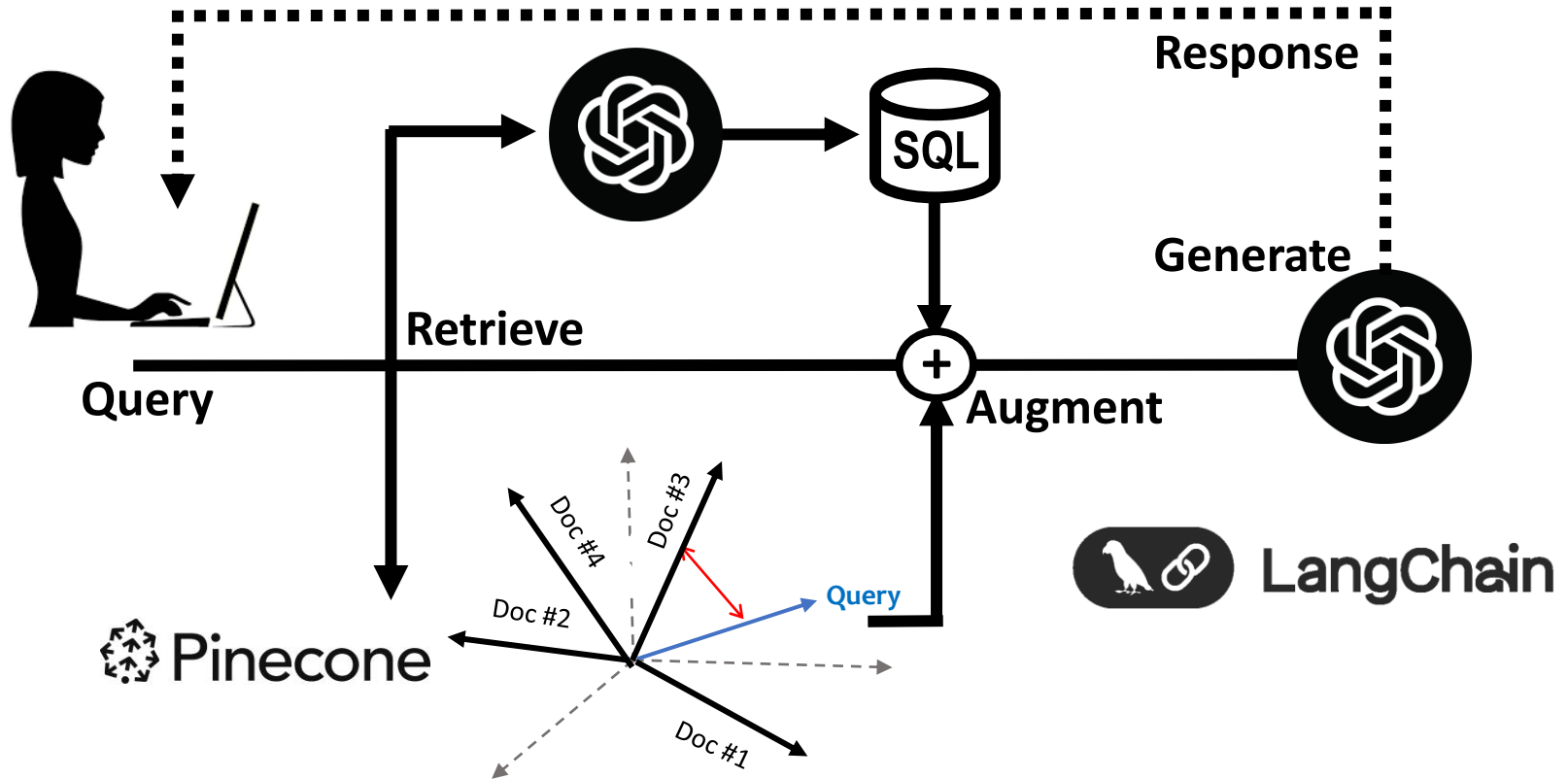
1.7 The Data Shadow



1.8 Data Science in a Post-Big Data World

- As such data has become more common place, really “Big Data” has become BAU (business as usual) for most large and medium-sized companies.
- The challenge, moreover, becomes how we run two systems (“traditional” and “big data” systems) in parallel – e.g. optimal data architecture.
- For the analyst / data scientist, of focus should be on maximising the value that can be created and identifying new opportunities to find predictive data sources.

1.9 However...



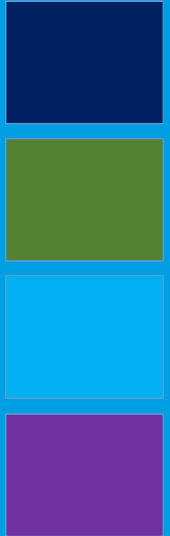
Session Structure

Introduction

Data Preparation and Feature Engineering

Organisation Data Architecture and Strategy

Asynchronous Tasks



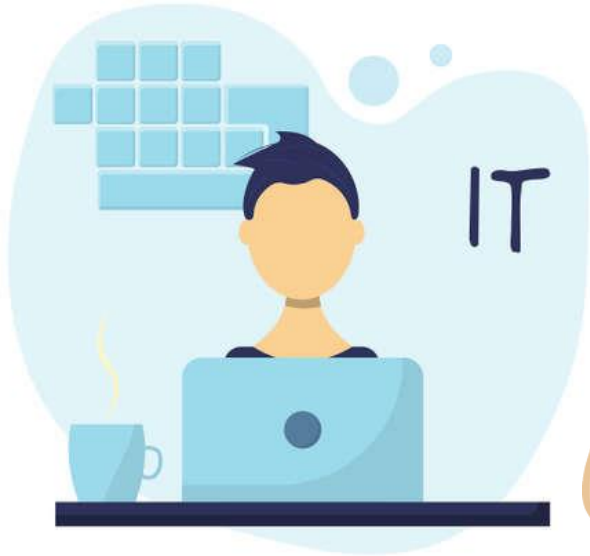
2.1 ANNOUNCEMENT!

- Unfortunately we cannot run next week's lecture here in person for operational reasons.
- I apologise unreservedly for this.
- Instead we will run the lecture as a synchronous online session (via Kaltura) on **Friday 24th at 09:00.**
- If this presents significant challenges for anyone I am more than happy to also record the lecture, and release it on Monday, as well as running the synchronous session Friday.



2.2 Data in Business

*Typically
undervalues context*



*Typically loses
control of data*



2.3 Data “W” Questions

- **WHERE** does the data reside?
- **WHO** knows about the data?
- **WHY** is the data collected?
- **WHAT** data is there
(and **WHAT** formats)?
- **HOW** is the data linked?



2.4 The Clean-Up



2.5 Features and Labels

- In data science and machine learning, we typically are trying to predict some label (available to us as a data item - Y), using a set of other features ($\mathbb{X} = \{x_1, x_2, \dots, x_k\}$). For instance, We may want to predict:
 - If a student will attend a class (Y), based on features (\mathbb{X}) such as age, subject area, year of study, etc.
 - Our daily sales total (Y), based on features (\mathbb{X}) such as marketing spend, temperature, time of year, etc.
- In the case of Y , the main goal is for this to be accurate and similar to the real values we may see in actual use.
- In the case of \mathbb{X} , we want to transform these values to be consistent, clean, and predictive. We call this process *feature engineering*.

2.6 Labels are Hell

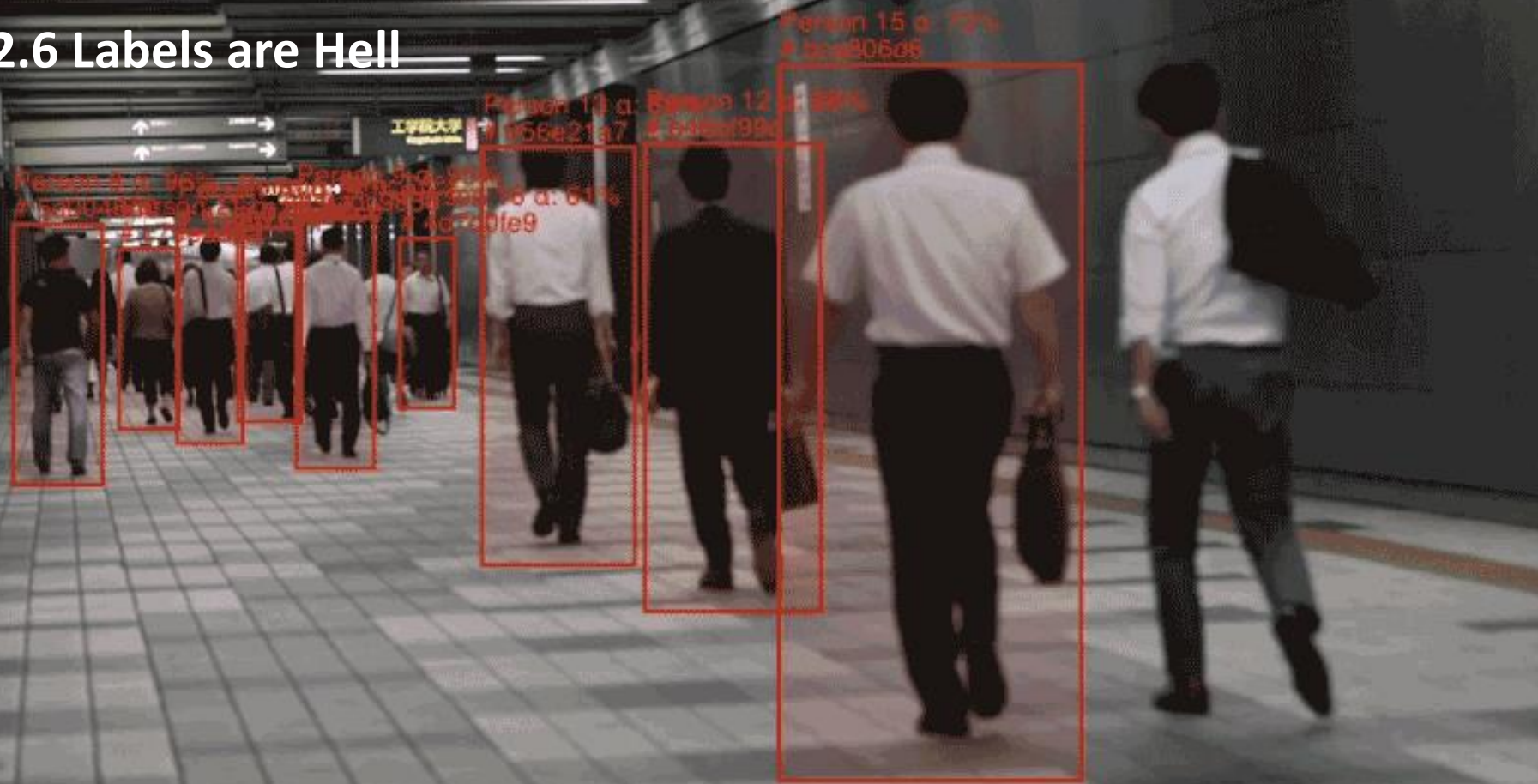


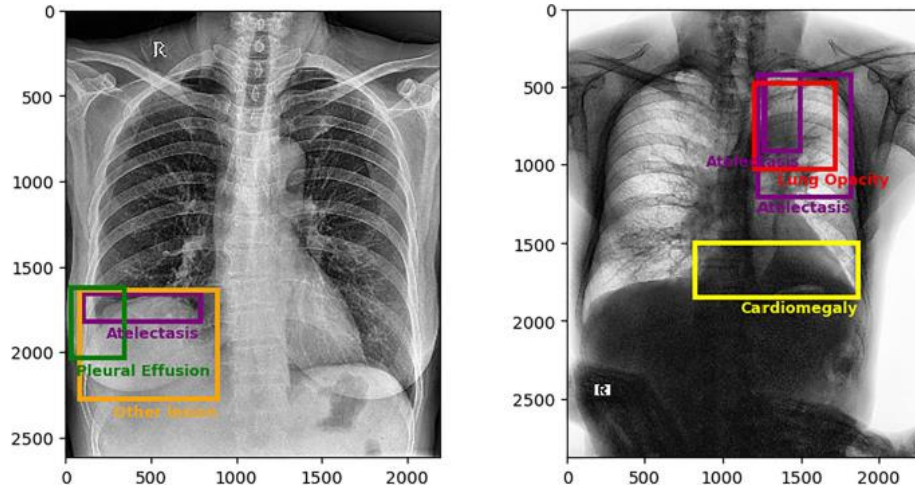
Image Credit: <https://encord.com/blog/automated-data-annotation-guide/>

2.6 Labels are Hell

- There are various solutions to the label/data annotation problem:
 - Paying humans small amounts of money to annotate your data;
 - Using AI-based tools to automate the annotation of your data;
 - Using AI-based tools to *semi*-automate and then paying humans small amounts of money to complete the annotation of your data.
- All of them are relatively arduous and time consuming, prone to errors and often expensive to organise.
- Will not be an option in all cases ...

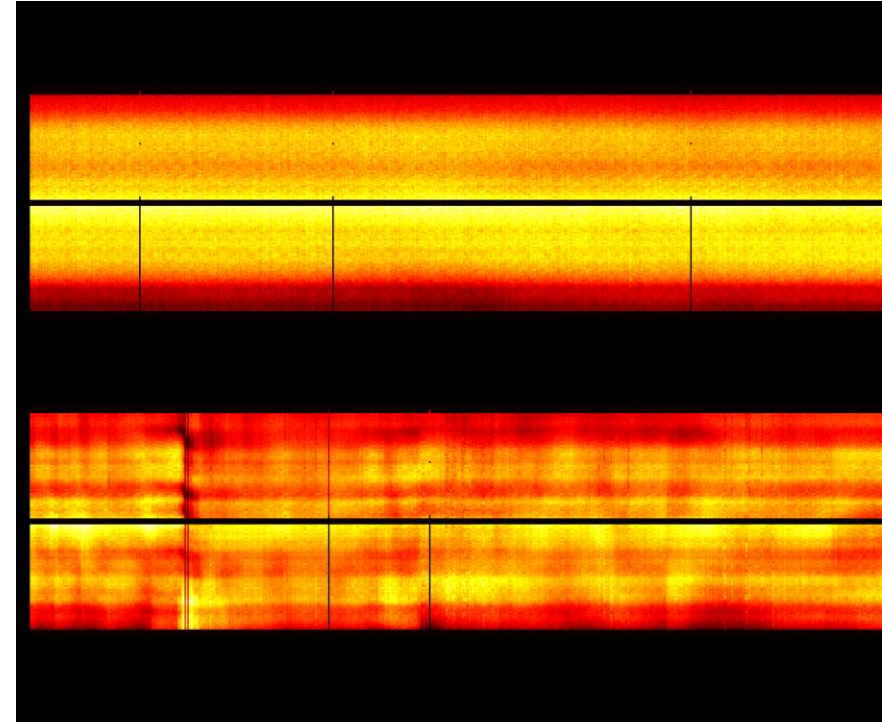
2.6 Labels are Hell

- Will not be an option in all cases ...



You can't pay people small amounts of money to label complex, specialist data. We need SMEs who will be time limited and/or expensive.

Other data will be largely uninterpretable by humans or by machines.



2.6 Labels are Hell

- No labels = no (supervised) learning.
- Errors in labelling = model biases, misclassifications and other bad things.
- Labelling is not always obvious ...



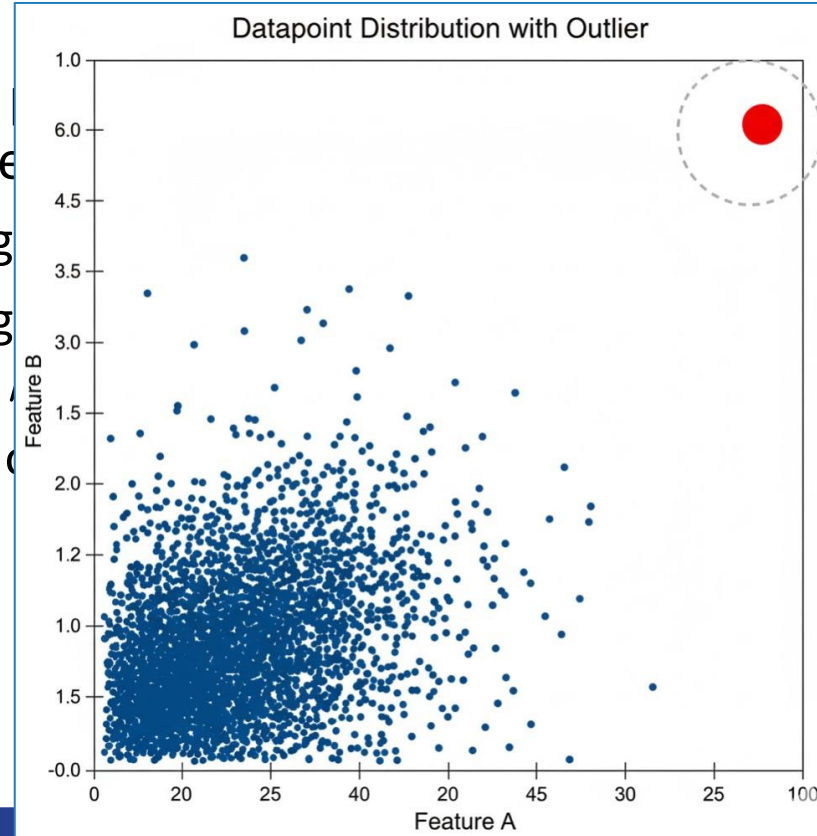
2.7 Data-driven Feature Engineering

- In data-driven approaches to feature engineering, we use statistical and programming methods to transform data based upon the values within it. For example:
 - Converting Boolean values such as "DEAD", "ALIVE" to 0 and 1.
 - Converting ordered lists to a dictionary of values. E.g. ["High School", "UG", "PG", "PHD"] to [0, 1, 2, 3].
 - Removing or modifying outliers in the data ...

2.7 Data-driven Feature Engineering

- In data-driven statistical and machine learning, we use statistical and machine learning upon the value of features.

- Converting categorical features to numerical
- Converting numerical features to categorical
- Removing outliers



, we use
machine learning based

“LIVE” to 0 and 1.

E.g. [“High
1, 2, 3”].

2.7 Data-driven Feature Engineering

- In data-driven approaches to feature engineering, we use statistical and programming methods to transform data based upon the values within it. For example:
 - Converting Boolean values such as "DEAD", "ALIVE" to 0 and 1.
 - Converting ordered lists to a dictionary of values. E.g. ["High School", "UG", "PG", "PHD"] to [0, 1, 2, 3].
 - Removing or modifying outliers in the data.
 - Dealing with missing values in the data...

2.7 Data-driven Feature Engineering

- In data-
statistic
upon the
- Cor
- Cor
- Re
- De

Sales Dataframe with Missing Values

Order_ID	Product	Quantity	Price	Customer_City
1001	Laptop	1	1200.00	New York
1002	Keyboard	1	75.50	New York
1003	Monitor	2	2	Los Angeles
1004	Monitor	NAN	300.00	London
1004	NAN	NAN	3	Paris
1005	Mouse	5	25.00	Paris
NAN	Webcam	1	50	Sydney

e
based

o 0 and 1.
High
3].

2.7 Data-driven Feature Engineering

- In data-driven approaches to feature engineering, we use statistical and programming methods to transform data based upon the values within it. For example:
 - Converting Boolean values such as "DEAD", "ALIVE" to 0 and 1.
 - Converting ordered lists to a dictionary of values. E.g. ["High School", "UG", "PG", "PHD"] to [0, 1, 2, 3].
 - Removing or modifying outliers in the data.
 - Dealing with missing values in the data.
 - Algorithmic methods such as dimension reduction (e.g. principal component analysis) or clustering (e.g. *K*-means).

2.8 Theory-driven Feature Engineering

Theory-driven feature engineering involves injecting subject-matter expertise into raw data. This is usually achieved through:

- ✓ Facilitations/brainstorming workshops;
- ✓ Literature reviews;
- ✓ Contextualising data items.

Question: What data items may be predictive of whether a student will pass a module?

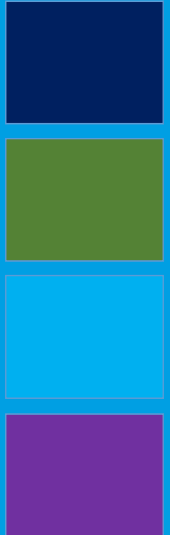
Session Structure

Introduction

Data Preparation and Feature Engineering

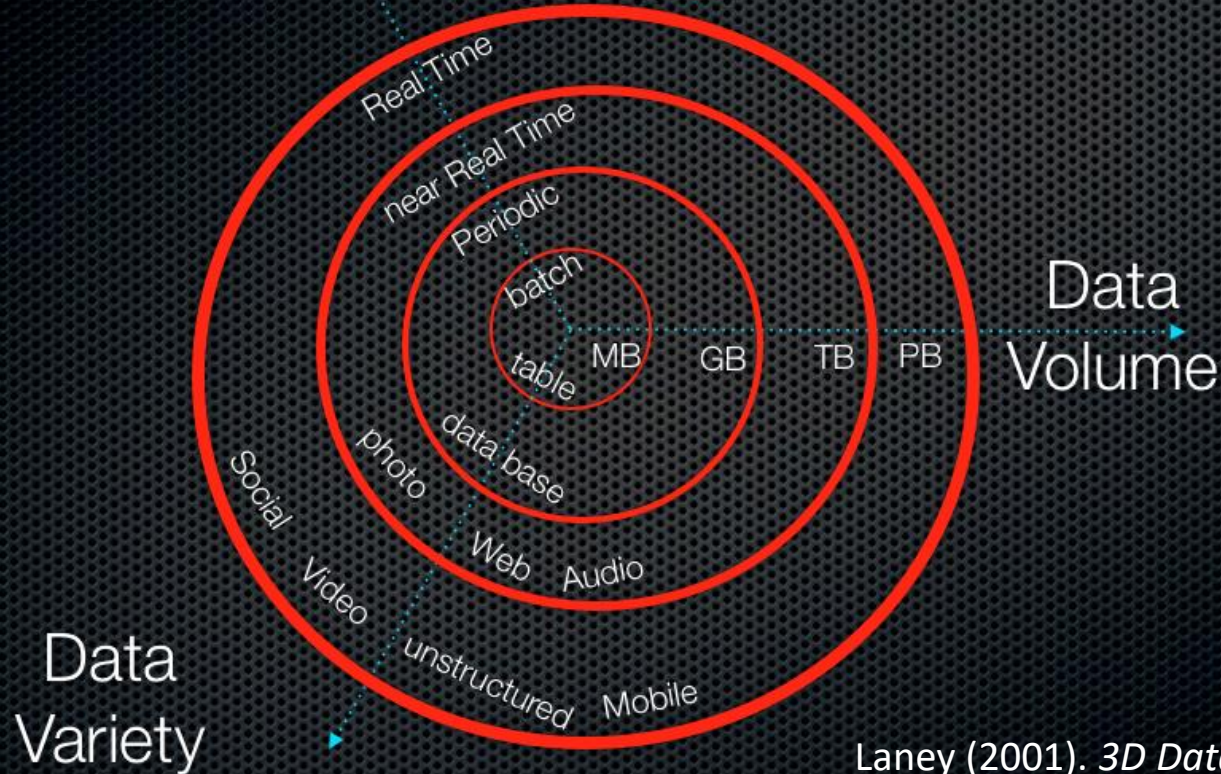
Organisation Data Architecture and Strategy

Asynchronous Tasks



3.1 Big Data Challenges

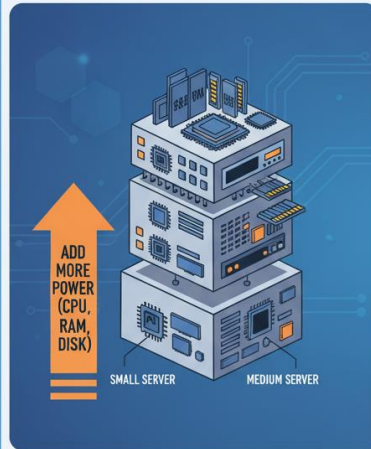
Big Data:
Expanding on 3 fronts
at an increasing rate.



Laney (2001). 3D Data Management.

3.2 Volume (Distributed Computing)

COMPUTING SCALING STRATEGIES



SCALE UP (Vertical Scaling)

Increase resources of a
single server.



SCALE OUT (Horizontal Scaling)

Add *more servers* to a
distributed system.

3.2 Volume (Distributed Computing)

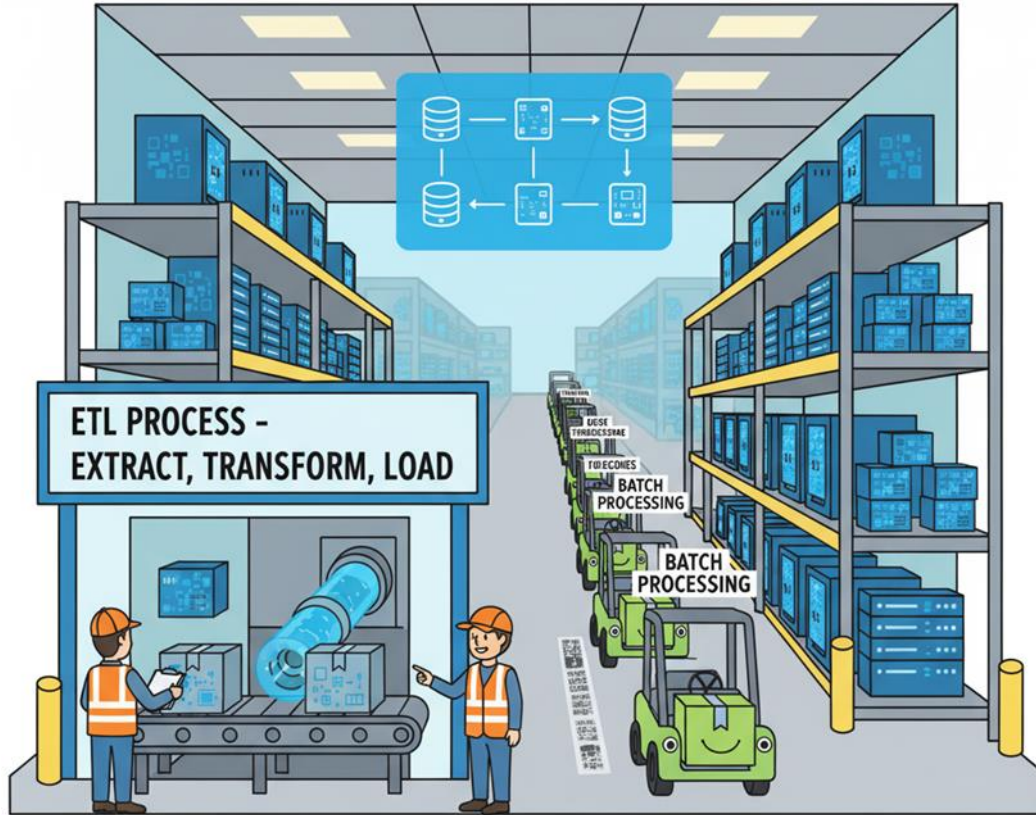
- For transactional workloads – e.g. value look-ups, data joining, writing new data – we prefer to distribute data by rows. I.e. each new data record is added to a single machine.
- For analytical workloads – e.g. calculating the average of a field or aggregating a value by another value – we prefer to distribute by columns. I.e. each new record is stored on different machines based on the columns in the data.

Transaction	Cust name	Prod ID	Quantity	Unit price	City	Delivery
1234	Leonardo	XYZ	1	£1,234	Coventry	15/10/2025
1235	Donatello	ABC	2	£1,245	Birmingham	16/10/2025
1236	Raphael	DEF	3	£1,236	Coventry	01/11/2025
1237	Michaelangelo	WBS	4	£1	Warwick	15/10/2025

3.3 Variety (Heterogeneous Databases)

- As more organisations have embraced working with *wide variety* data types, we have seen a wide variety of databases/datastores introduced. For example:
 - For text data we may use a *Document DB* or a *Vector DB* (more on these later in the module);
 - For a set of transaction we may use a *Ledger DB* (e.g. Blockchain);
 - For a social network or geospatial data we may use a *Graph DB*;
 - For features we use to build machine learning models we may use a *Feature Store*;
 - And many, many, many other types. Many projects themselves will have multiple DBs. **Fit the DB to the data not the data to the DB.**

3.4 Velocity (From Warehouses to Lakes)



3.4 Velocity (From Warehouses to Lakes)

"Pay attention to data flows as opposed to stocks"

3.4 Velocity (From Warehouses to Lakes)



3.5 Master Data Management (MDM)

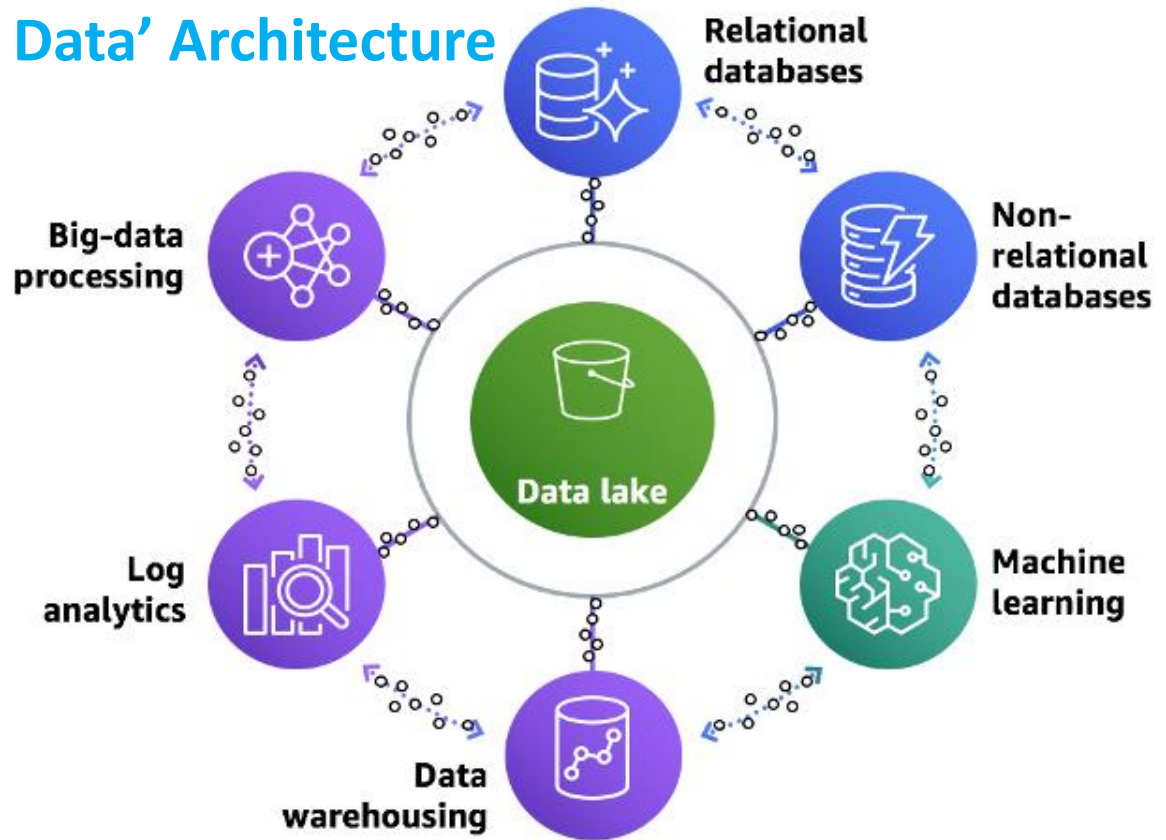
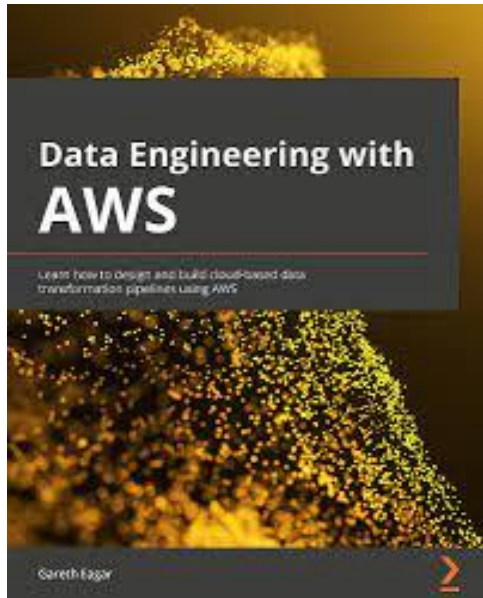


3.5 Master Data Management (MDM)

Master Data Management / Data Warehouses

- ✓ Data is carefully managed and maintained.
- ✓ Data items are transformed **prior to storage** to ensure they are consistent, clean and well understood.
- ✓ The goal is **single version of the truth** and high levels of confidence in the *veracity* of the data.
- ✗ Slow to create, slow to load data, slow to adapt to new data.
- ✗ Requires significant subject matter expertise to setup.

3.6 Data Lakes / 'Big Data' Architecture

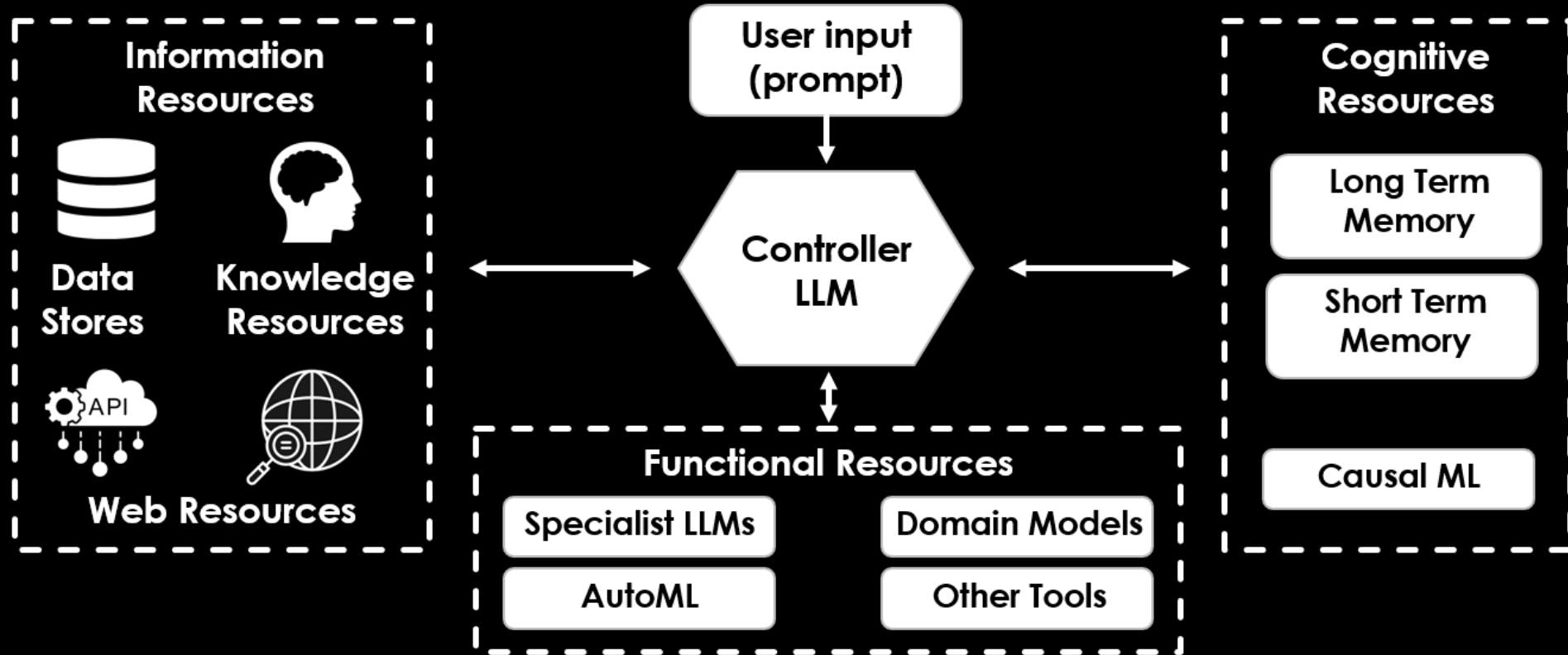


3.6 Data Lakes / 'Big Data' Architecture

Data Lakes / 'Big Data' Architecture

- ✓ Storage is highly parallelised and scalable.
- ✓ Data items are transformed **based on the specific requirements of the task in hand** (potentially many times). Typically both raw and transformed data are stored.
- ✓ The goal is **fast, scalable and flexible storage** where pipelines quickly process streams of **heterogeneous data** based on different use cases.
- ✗ Often there are many, many version of the truth.
- ✗ Typically heterogeneous data means **heterogenous databases**.

3.7 Data Architecture in the AI Age



Session Structure

Introduction

Data Preparation and Feature Engineering

Organisation Data Architecture and Strategy

Asynchronous Tasks

