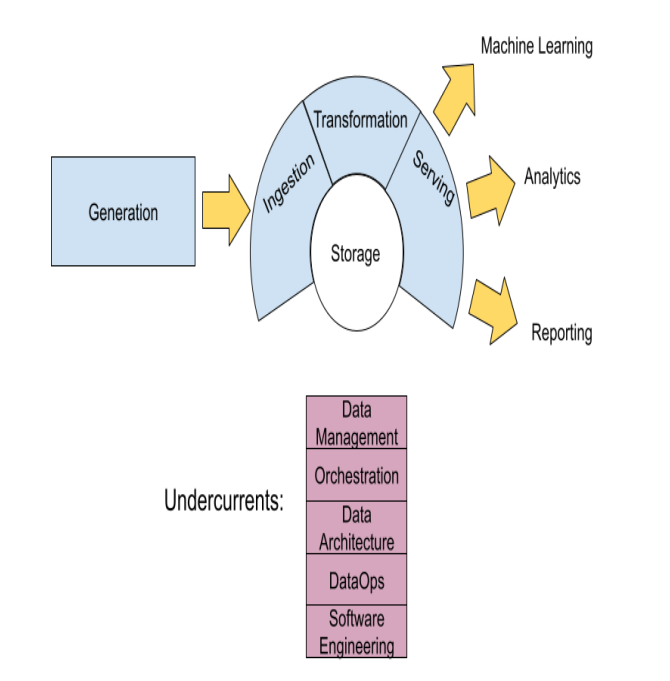
**Fundamentals of Data Engineering**

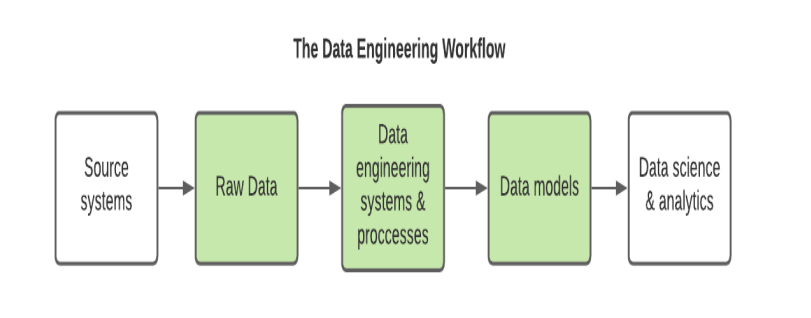
Chapter 1: Defining a Data Engineer

Vocab:

* **Data engineers** set up, operate, maintain, and update a company's data pipelines and infrastructure for it to be further analyzed by data scientists and other consumers of said data. In short data engineering is about movement, manipulation and management of data
  + **SQL-Focused** - primarily works in SQL and utilizes separate ETL tool to handle the storage and maintenance of the database where the end users would fetch their data
  + **Big Data focused** - processing and handling of big data in Hadoop, Cassandra or other distributed system for data streaming and computation at scale
  + Data engineers produce reliable data that serves the business with predictable quality and meaning
  + A data engineer does NOT build ML models, create reports or dashboards, analysis, build KPIs or develop software applications
  + **External engineer -** normally aligns with users of external applications such as social media apps, IOT, e-commerce applications and build their architecture around these sources
  + **Internal engineer -** focuses on business and internal stakeholders such as creation and maintenance of data pipelines, data warehouses for BI dashboards, reports and data science
* **Software Engineers -** build the applications and systems that run the business and usually largely responsible for generating the internal data that data engineers will consume
* **Big Data** - 3 Vs is velocity (how frequently new data is coming in), variety (data types), and volume (quantity)

Concepts:

* Data engineering began in the 1980s when business required dedicated systems and professionals to house and store the data used for modeling business logic and further analysis and BI
* Early 2006s open sourced, flexible storage began with Yahoo developing Hadoop followed by Amazon Web Services, DynamoDB, Kinesis for streaming data
  + All open source, storage flexible and pay as you go rather than hardware purchasing
* Common Python libraries - Pandas, NumPy, Airflow, SKLearn, Tensorflow, Pytorch, PySpark, BIG FOR APIs
* Command line interfaces - Bash for LINUX, Powershell for Windows
* FOCUS ON THE FUNDAMENTALS TO UNDERSTAND WHAT'S NOT GOING TO CHANGE AND PAY ATTENTION TO ONGOING DEVELOPMENTS TO SEE WHERE THE FIELD IS GOING
* 

* 

Chapter 2: The Data Engineering Lifecycle

* **Data Engineering Lifecycle** - series of stages that turn raw ingredients into a useful end product ready for consumption by analysts and data scientists
* Difference between data life cycle and data engineer lifecycle is that DE cycle is a subset of the overall data lifecycle and is only what the DE can control - usually from generation, ingestion, transformation and serving
* **5 Main Parts of DE Lifecycle**
  + Generation - DEs must understand the source systems that initially generate the data we use. Examples can be a CRM system to database path.
    - Useful considerations: rate of data generation, consistency, error handling, duplicate handling, RT or ST or delta, updates with additions or subtractions of features, how does the database schema look where these data are stored
  + Ingestion - data flow from source system to storage
    - Batch ingestion is a way of processing data in chunks rather than all at once for performance purposes
    - Streaming is processing data in real time, continuous, or as it happens
    - Push model is source system sending data to storage and pull is database retrieves the data from the source
  + Storage - can occur in multiple places with overlapping stages
  + Transformation - changing the form of the raw data to fit the required need
    - correcting data types, standardizing formats, duplicates etc.
    - transformation can be done in batch or in in flight while streaming
  + Serving Data - providing data for practical purposes that produce value to its end user
* Served for Analytics - report building, dashboards and ad hoc analysis
  + **Operational** - focus on present operating details that can be acted on ASAP like inventory or website performance
  + **BI** - describe the past and current state of business by processing data using business logic. Most mature for is self-serve which is the business to do their own analysis without IT intervention
  + **Customer-facing** - BI but it is meant for the customer. Where BI is done by the business or the business, customer-facing is done for the customer’s business and not done internally
  + **ML** - data engineers should be familiar with how the data will be used. ML is a later stage in data maturity that involves using past data events to create prediction models and dynamic targets
  + **Reverse ETL -**  process of taking data from the output side of the lifecycle and feeding it back into the source system such as taking processed data from storage and calculating quota using sales data and feeding it back into the CRM

**The Major Undercurrents Across the Data Engineering Lifecycle**