Abstract One of the most crucial operations in working with data is transporting the data from the source where it has been collected, to a system in which the data can be analyzed. To provide a solution that is scalable and prevents bottlenecks and data corruption, data pipelines provide a flexible solution for transforming and transporting data from static, as well as real time data streaming sources.

Source & Destination
The source describes where the data is collected from. This can be a Software Application, an API call, a webhooks which has been triggered or even just a relational database. One pipeline can have more than one Source, from which it can take data in.

The destination of a Pipeline is usually a Data Warehouse or a Data Lake, where the data waits to be analysed since it has been processed to the appropriate scheme. An output destination could also be a Software Application.

Transformation is a big part of data pipelining. It refers to operations that are being performed on the data. This happens in preparation, so that the data that is being dumped into the destination can be used right away.

5 main questions before implementing a data pipeline

- **WHAT to achieve**
- **WHERE does the data come from**
- **METRICS to measure outcome**
- **HOW to achieve the goals**
- **TEST the system**

Processing describes the way, how data is being processed. Although not being mutually exclusive, there are two systems that are used in praxis:

- **Batch processing** moves data in regular, scheduled intervals. This is of advantage when dealing with large datasets, which do not need to be moved in real-time.
- **Real-time processing** moves the data as it is being created. This can be useful when dealing with data that need immediate analysis.

Who needs data pipelining?
Since data collection, as well as data processment needs computing power it is more viable to run the tranformation on a different system, than the one, where data is being collected on to not impair their performance. This can also ensure security when access can be monitored more closely and only those who are supposed, are able to interact with the data. The flexibility of a data pipeline allows for multiple data streams to be processed in parallel, while also allowing the results of the pipeline to be sent to multiple destinations for different analyses.

Kay Erik Jenß
Bachelor Student
IT-Systems-Engineering
Hasso Plattner Institute, Potsdam, Germany
E-Mail: kay.jenss@student.hpi.de

Lecture Series on Practical Data Engineering
Prof. Dr. Felix Naumann
Prof. Dr. Tilmann Rabl

Based on the talk by Maximilian Jenders of
Lecture Series on Practical Data Engineering
Hasso Plattner Institute, Potsdam, Germany
E-Mail: kay.jenss@student.hpi.de

Hasso Plattner Institut