Faster Analytics - Optimizing the Data Engineering Process

Jarid McKenzie

He/Him

Analytics Architect

Iteration Insights

Jarid

McKenzie

He/Him

Analytics Architect

Iteration Insights

- Lead Analytics Architect
- Post Secondary Instructor
- Nerd

SCAN ME

LinkedIn: jarid-mckenzie

Foundatum

Who I actually am…

Want to become a speaker or mentor?

NEW STARS OF DATA

newstarsofdata.com

5/16/2025

Call for Speakers & Helpers is open!

PASS24 Data Community Summit

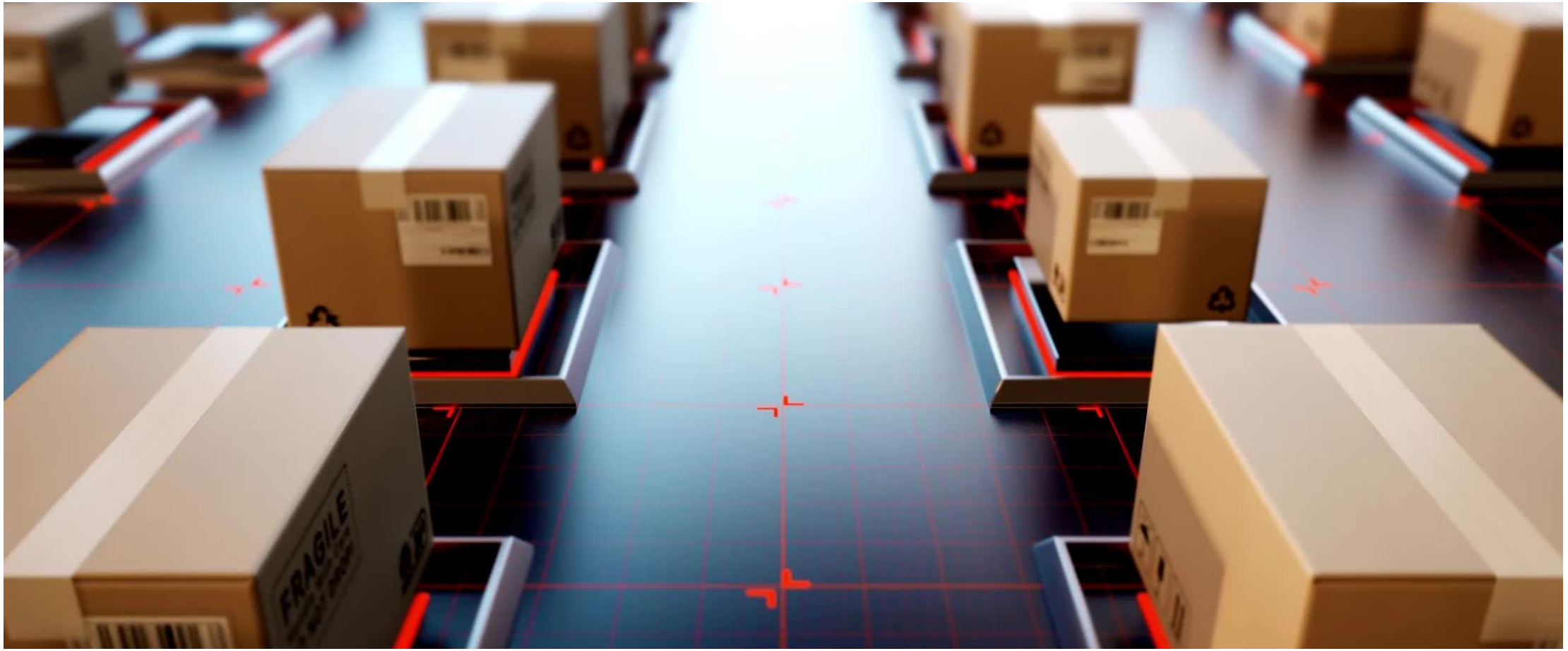
What are we going to talk about?

1. What sort of environment are we loading?
2. What are some of the tasks that need to be handled?
3. How do we manage the task dependencies?
4. Which tasks should we optimize?

What sort of environment are we loading?

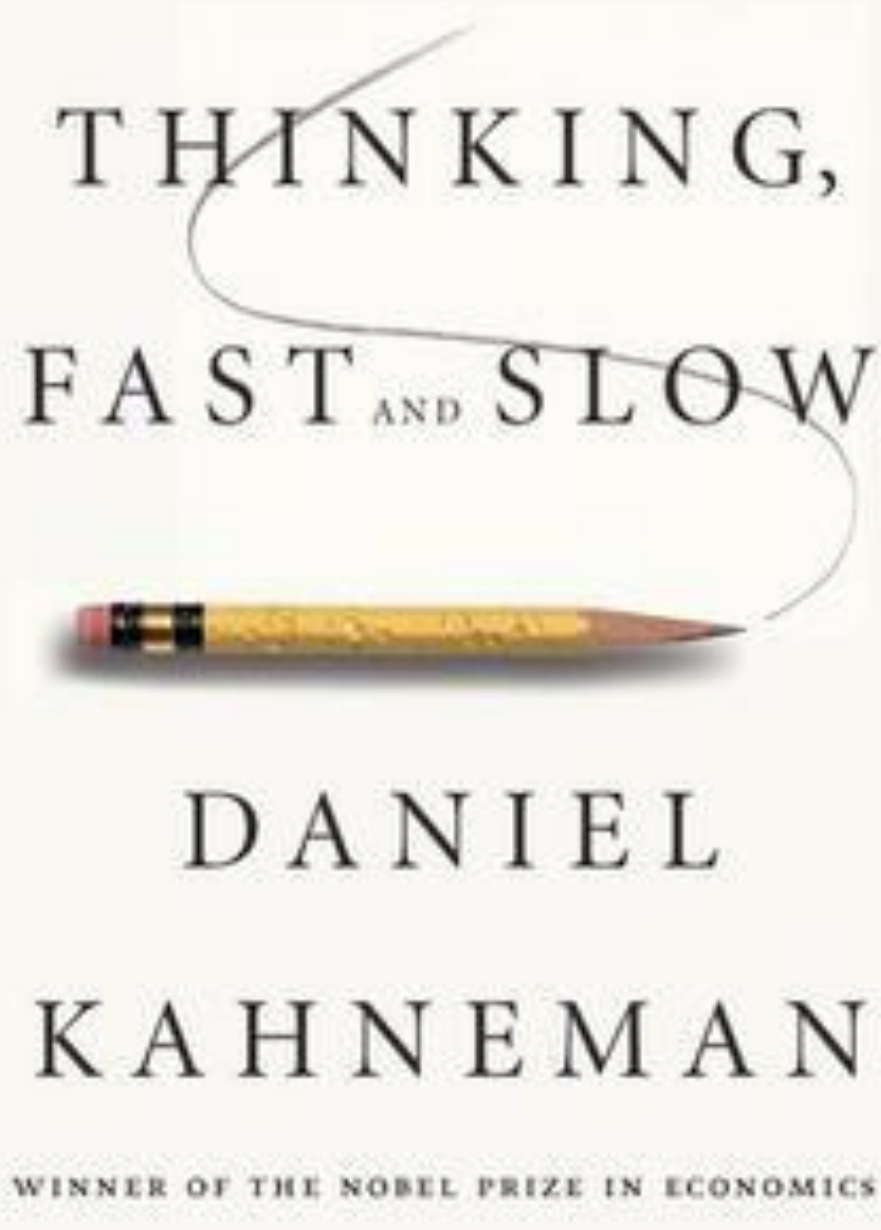
Data Lakehouse for Analytics

Loading Semantic Models

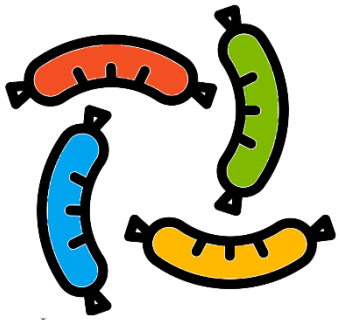


An Aside on Real-time Analytics

- System 1 – Fast, Instinctive, and Emotional
- System 2 – Slower, Deliberative, Logical



Let's look at a relatively simple example



Request

Deliver

Pay

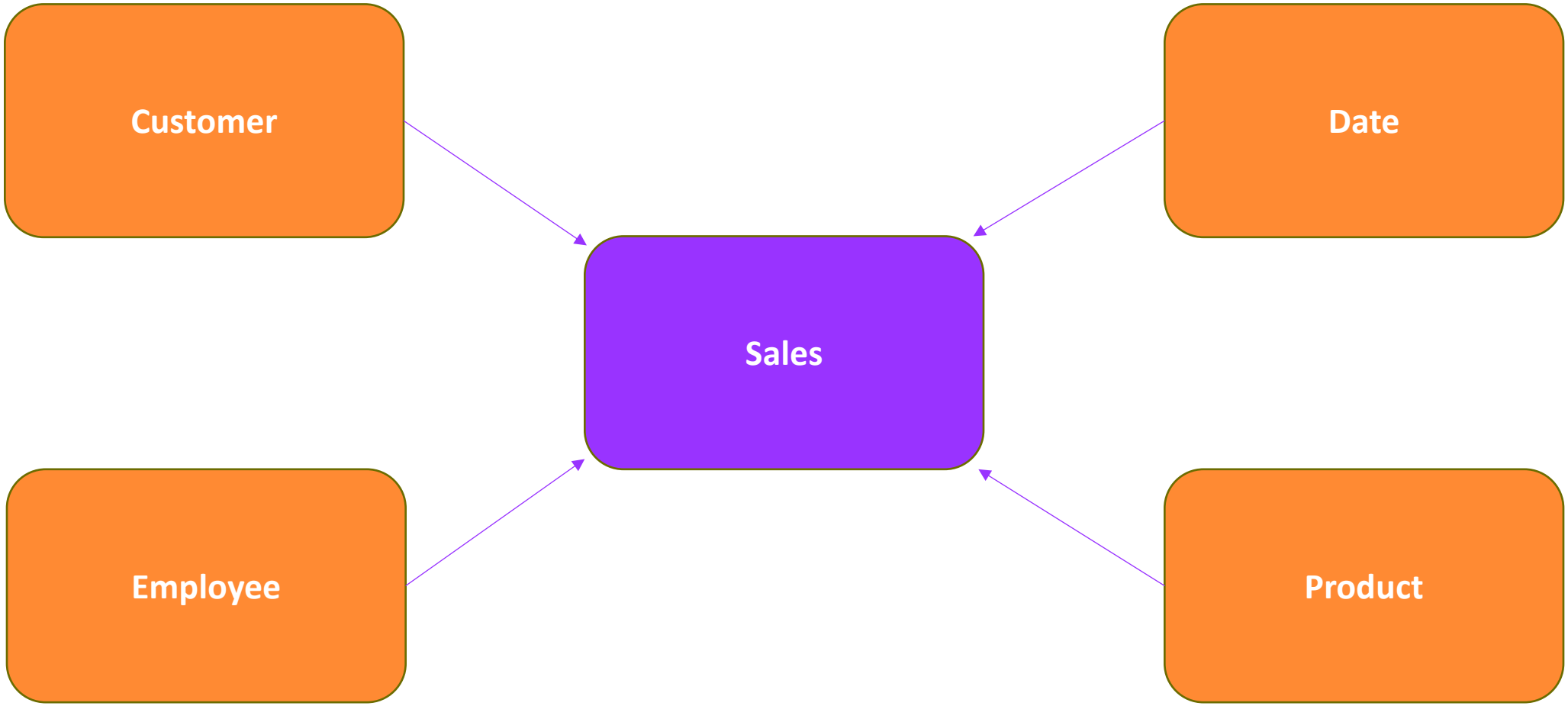
Request

Deliver

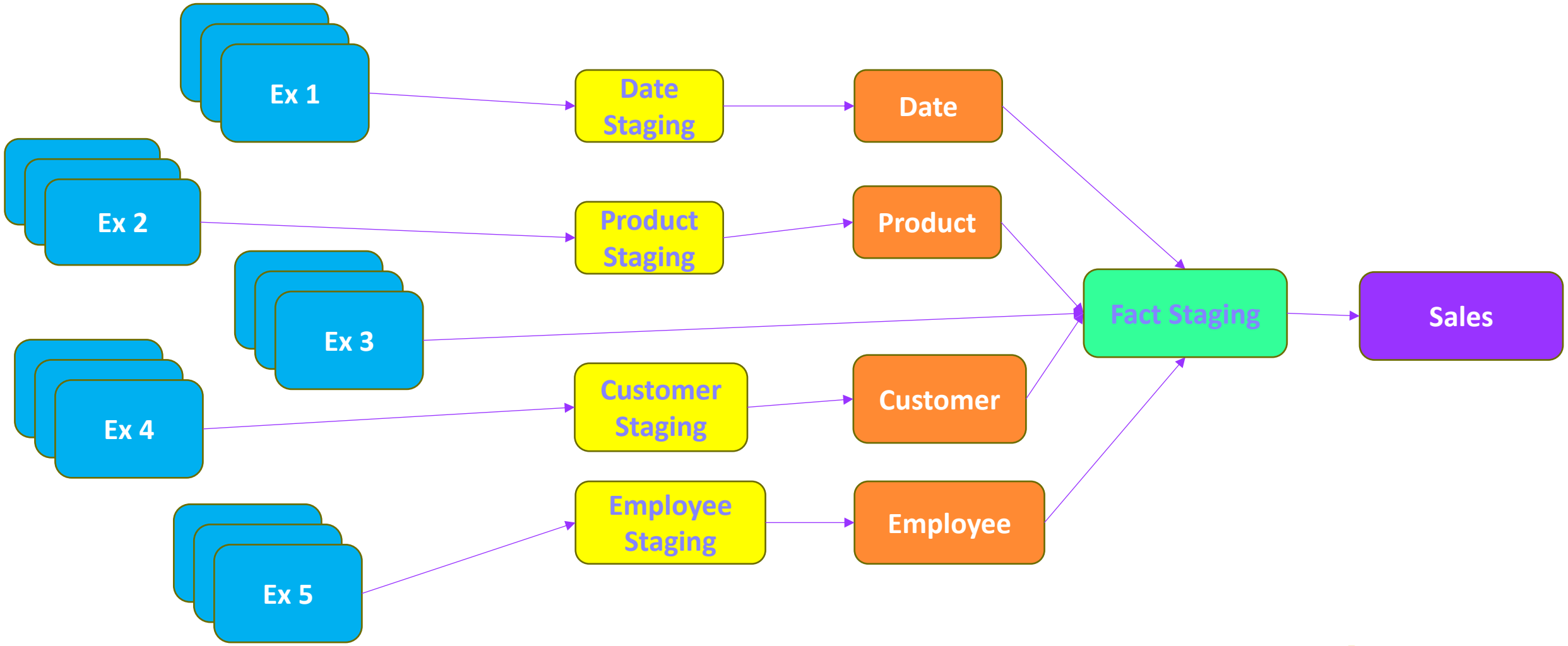
Pay

Ben Weissman

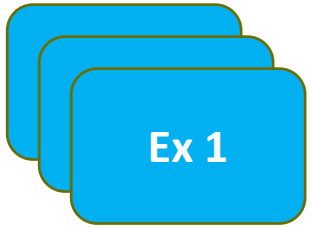
Let’s look at a relatively simple example



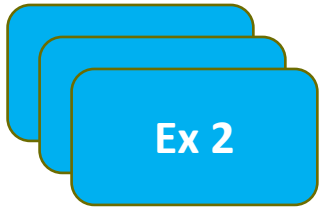
Let's look at a relatively simple example



Let's look at a relatively simple example



Extracts – Full/Incremental



Integrate into ODS

Dimension Staging

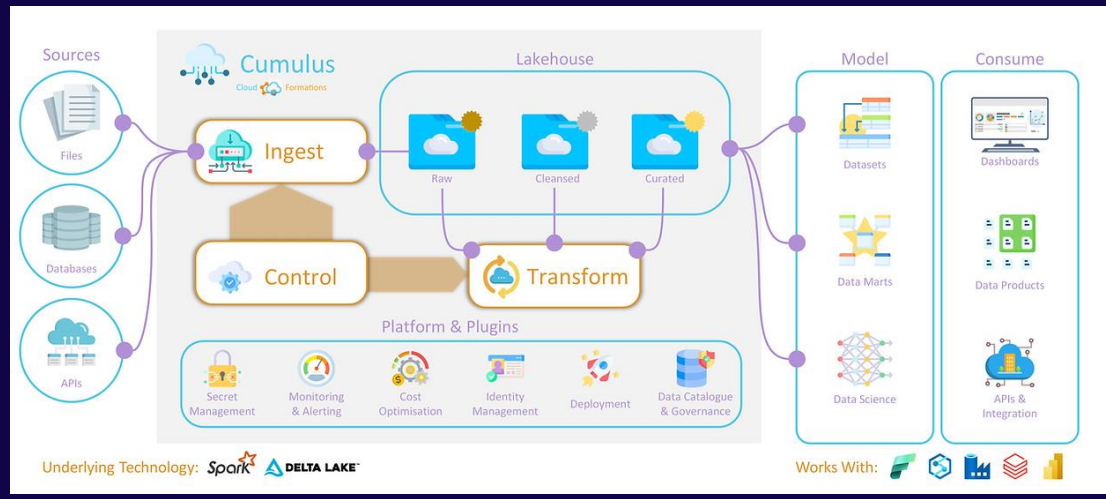
Replace Surrogate Keys

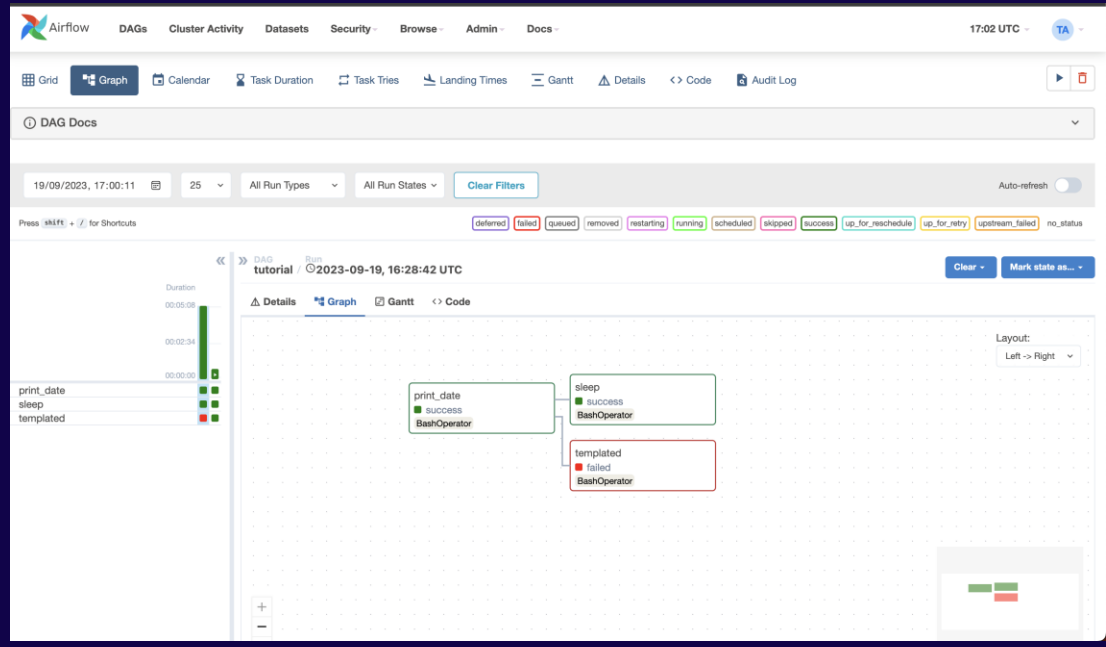
Dimension Loads (SCD 1 & 2)

Incremental Fact Load

How on earth do we mange all these tasks?

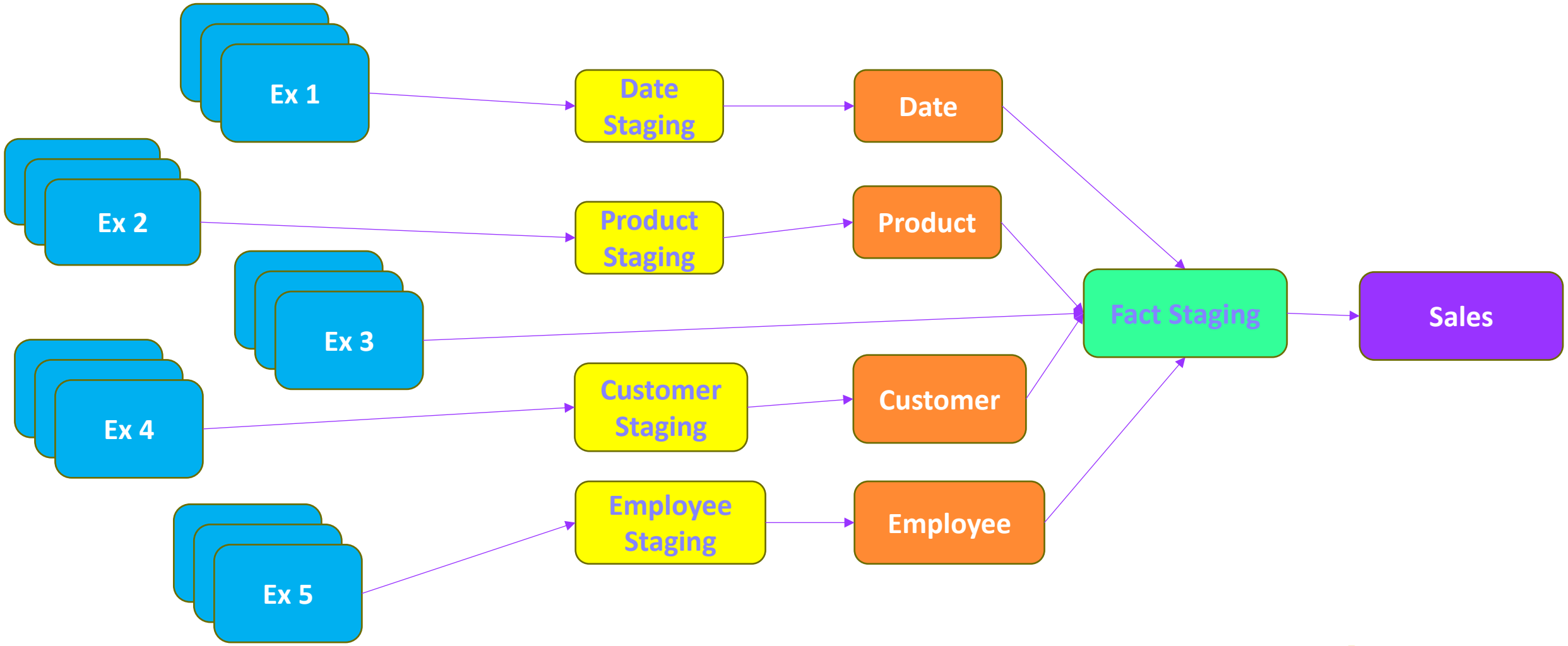
- Handwriting process dependencies within their own pipeline
- Find some open-source solution (CF.Cumulus)
- Apache Airflow (need to deploy a container or service)



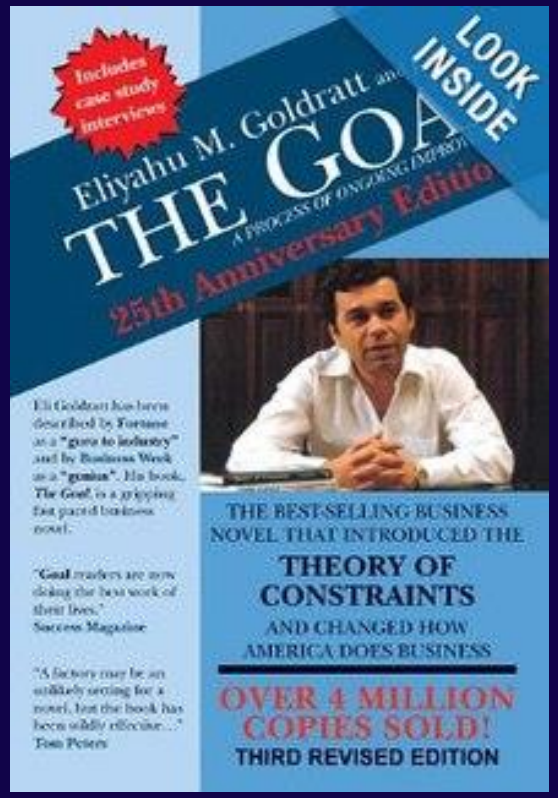


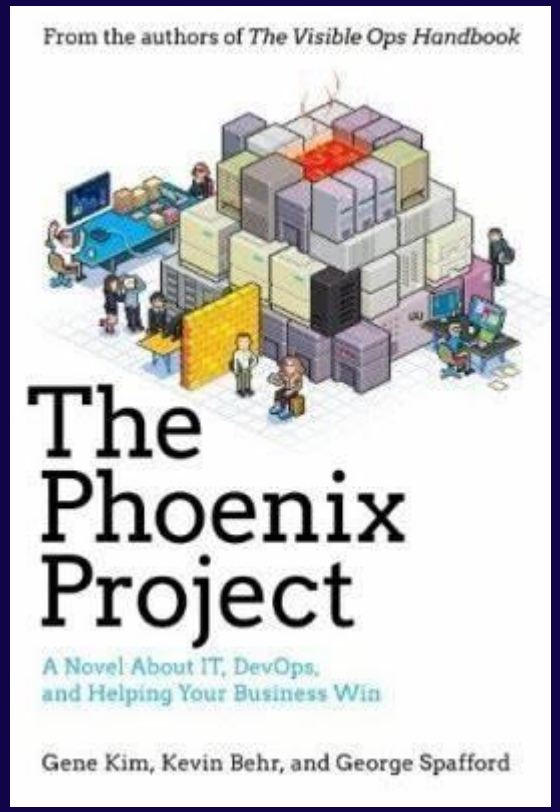
What do Frameworks Look Like?

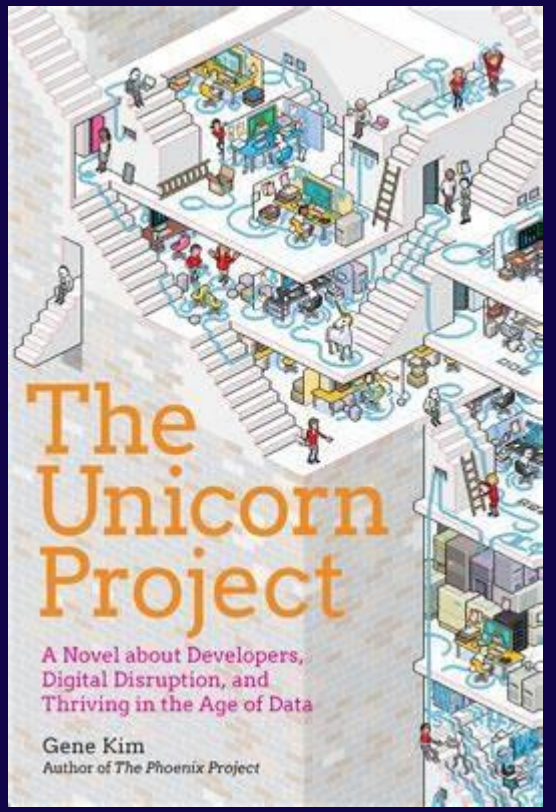
Using Stages



Books that motivated this approach

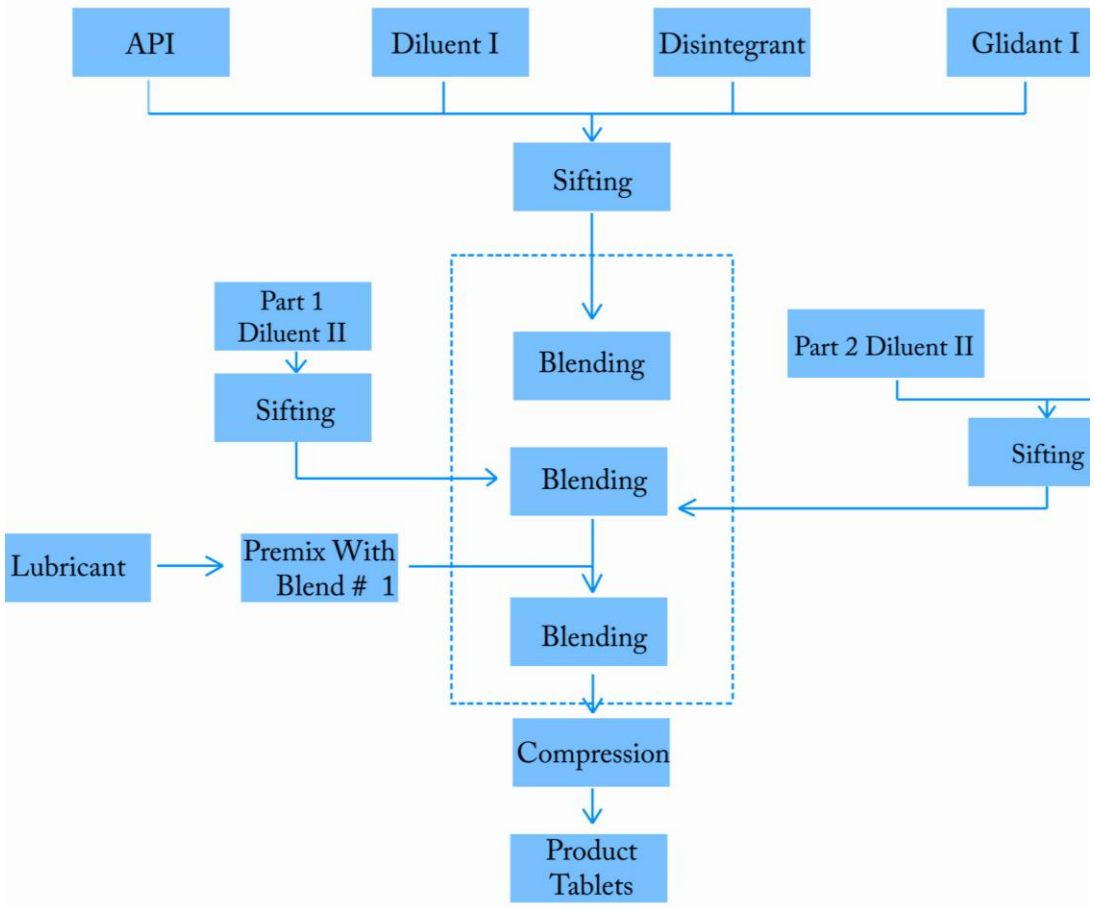




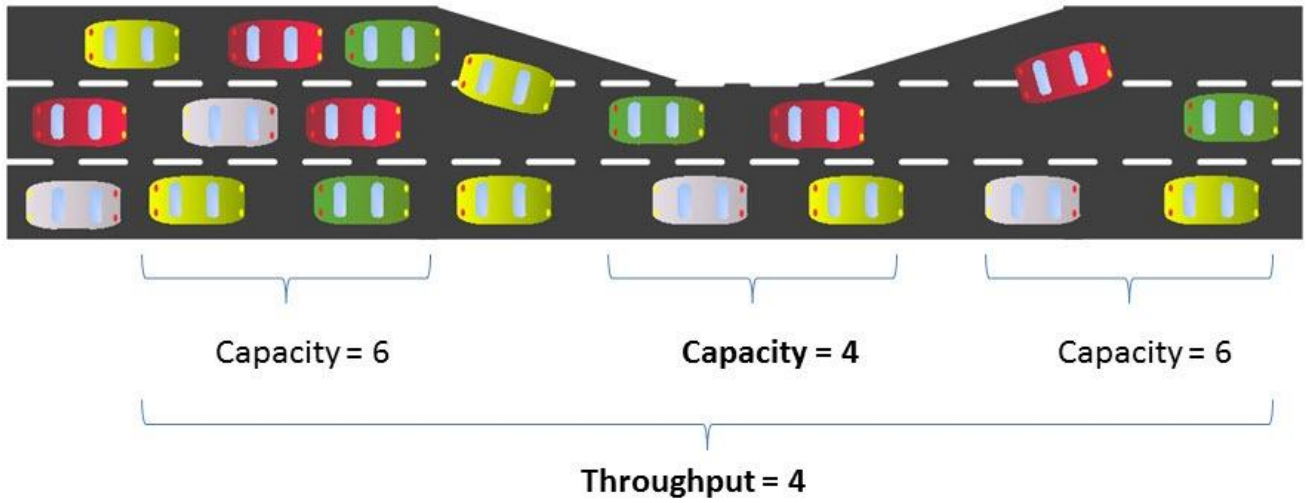


Analogy from Manufacturing

- In Manufacturing, there are many steps that need to be taken to arrive at a finished product.
- Some can be done in parallel, some in series. Most need to be performed using **separate, specialized** equipment.
- The key takeaway is to **Identify the Bottlenecks**.



Bottlenecks Data Engineering

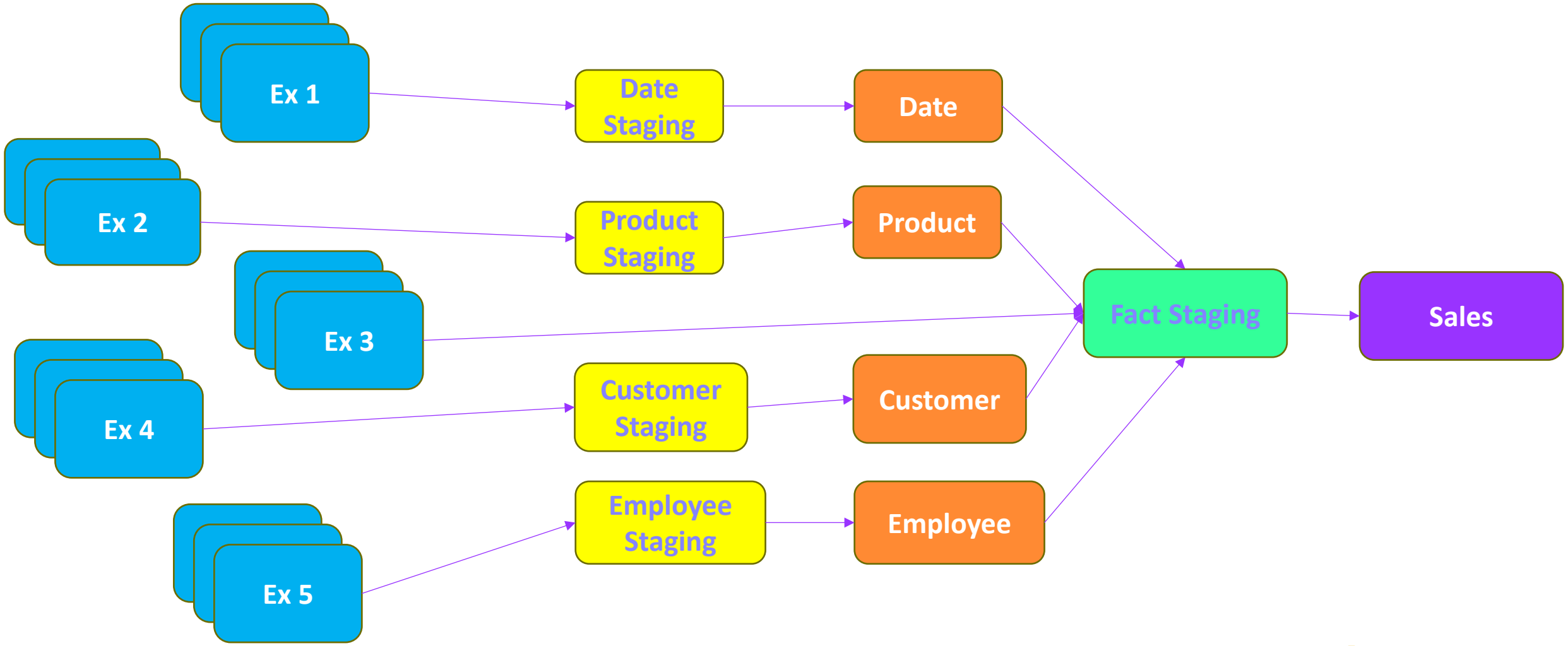


Identify the Bottlenecks:

- Self-hosted Integration Runtimes
- Servers that we're pulling data from
- Rate limited APIs
- Spark Pools

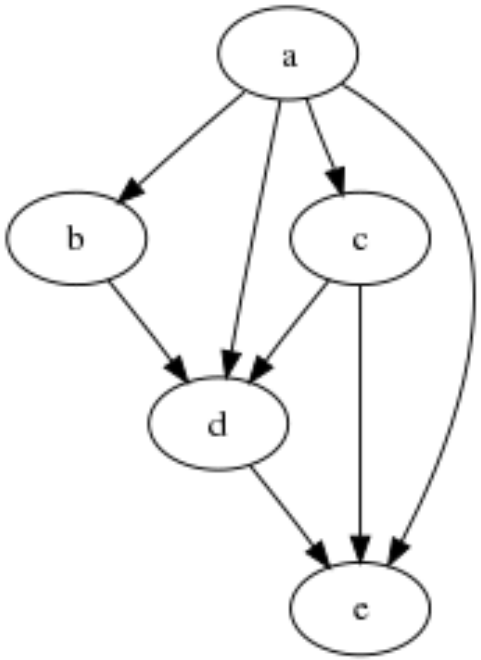
Remember that this is cloud processing. We can run significantly more operations in parallel than a physical manufacturing plant.

Identify the primary constraint of each task



What on Earth is a DAG?

What on Earth is a DAG?

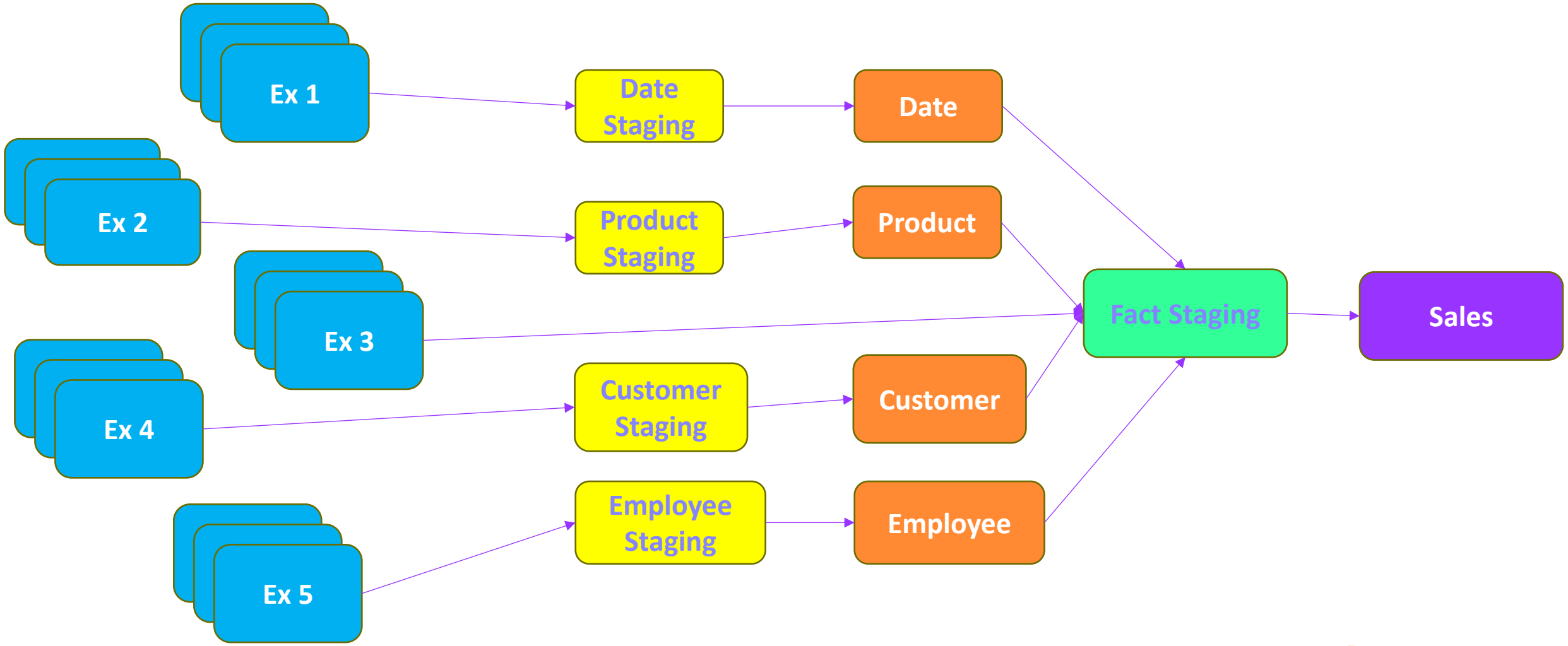


D **irected – the edges within a graph have direction from one vertex to another**

A **cyclic – the graph contains no cycles. Once a vertex has been visited, there is no way to 'walk' back to that vertex**

G **raph – A set of vertices (objects) and edges (relationships). An edge joins two vertices**

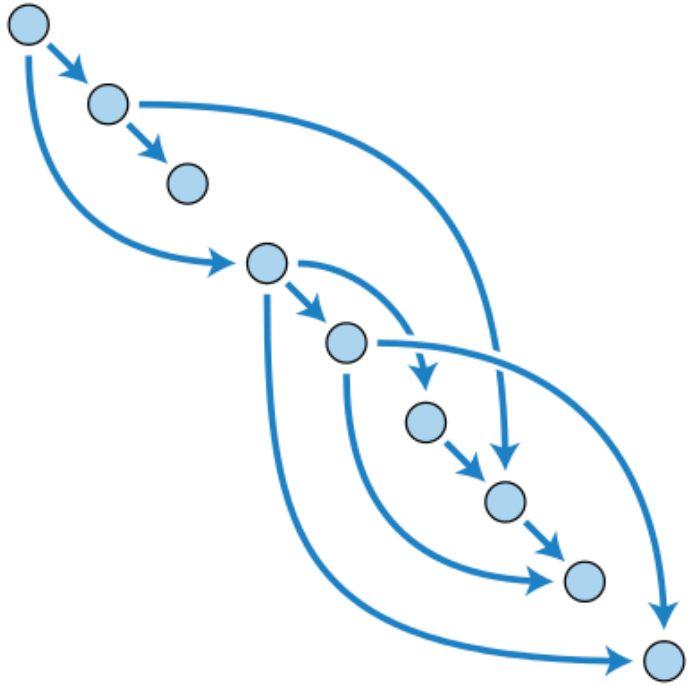
Semantic Model DAG



DAGs Help Us Schedule

Topological Sorting:

Arranging the nodes of the graph in such a way that we can complete them one after the other.



DAG Scheduling Demo

Some Easy Queue Ordering Strategies

1. Longest Waiting (FIFO)
2. Shortest Average Runtime
3. Longest Average Runtime

Longest Waiting (FIFO)

Include the timestamp of when the task is added to the queue.

Shortest and Longest Average Runtime

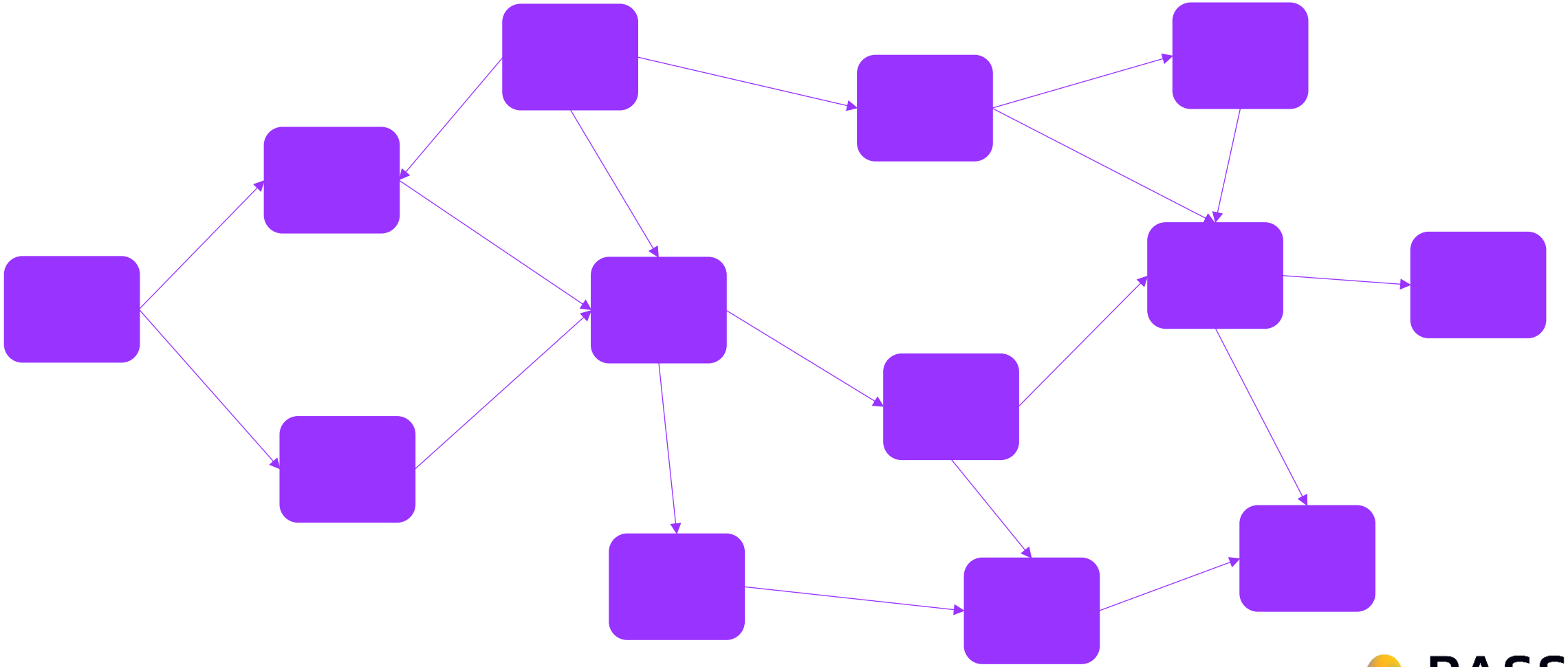
1. Keep a record of when the task starts and ends
2. Take a recent sample of runs (10ish)
3. When calculating the DAG, include the average

Simulation in Python

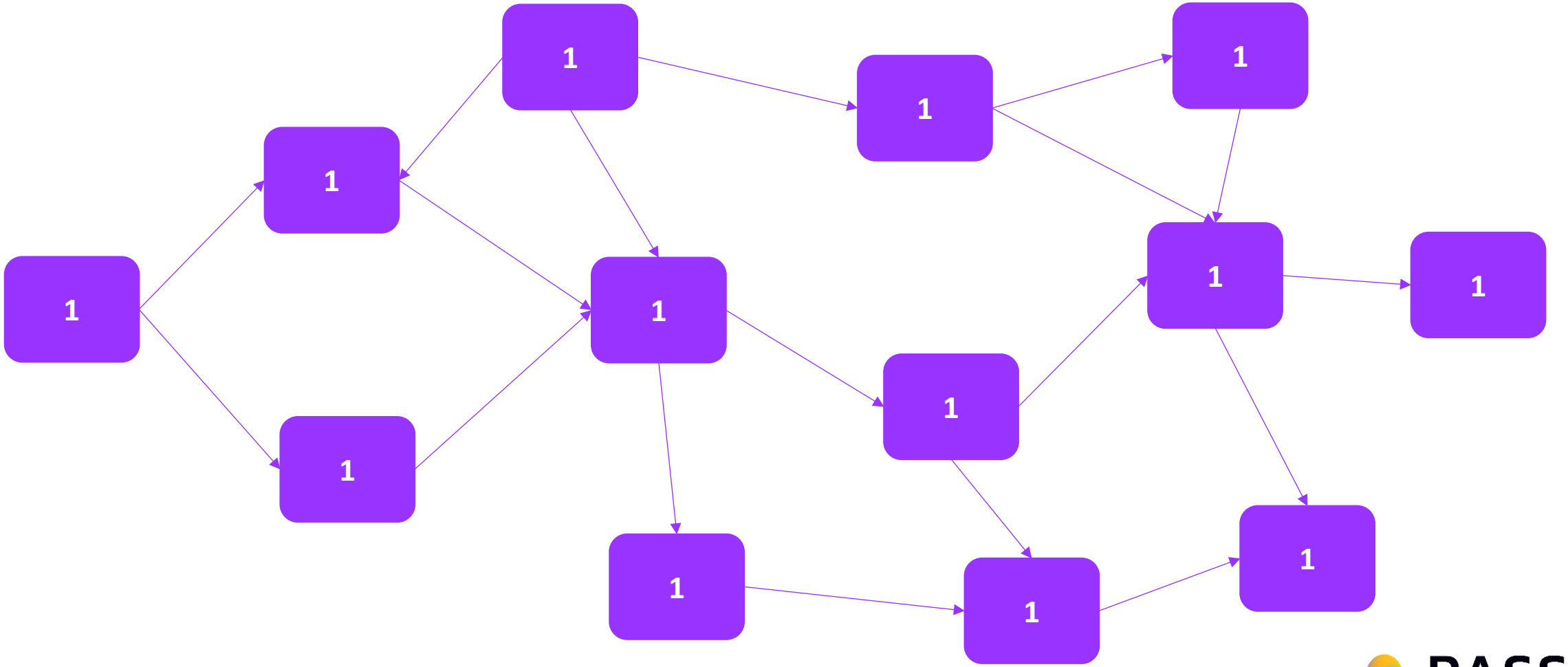
Hard Queue Ordering Strategies

1. Most Dependent Tasks
2. Longest Cumulative Dependent Tasks

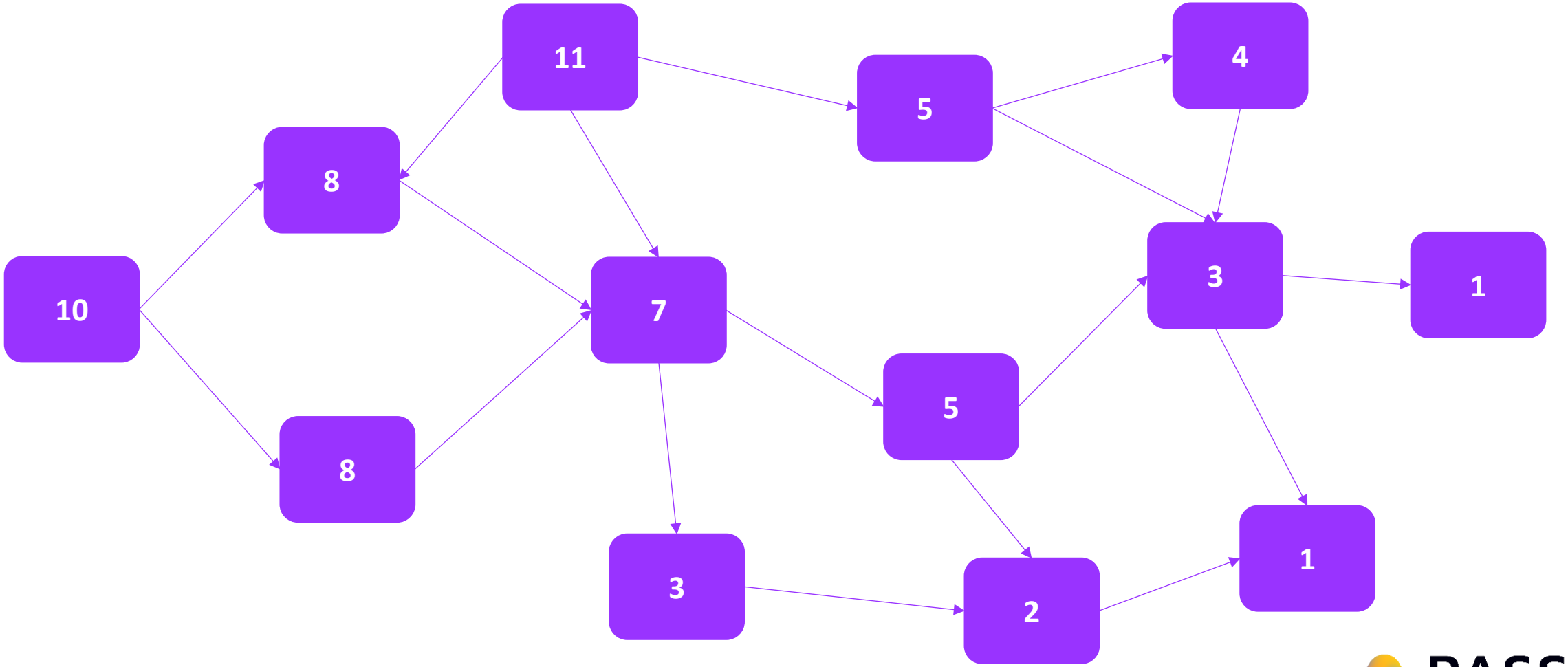
Dependent Tasks



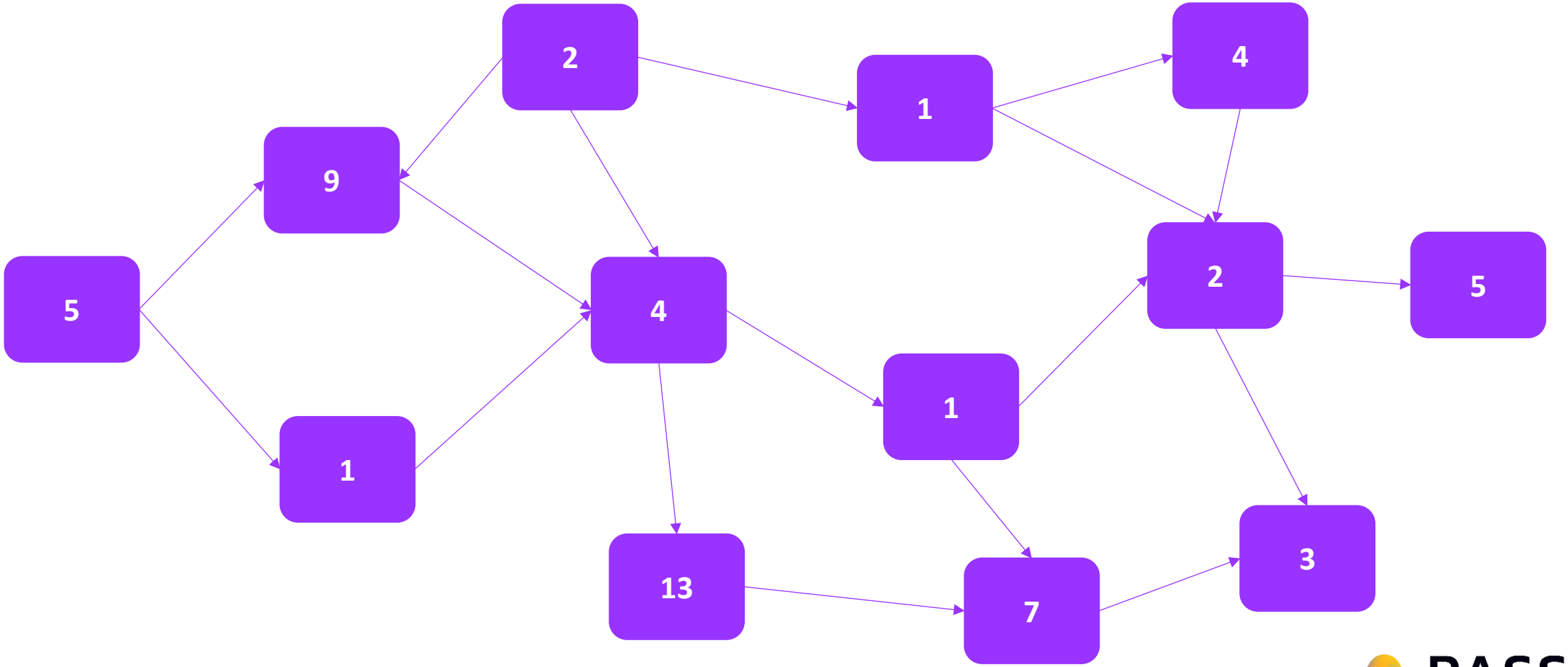
Dependent Task Calculation



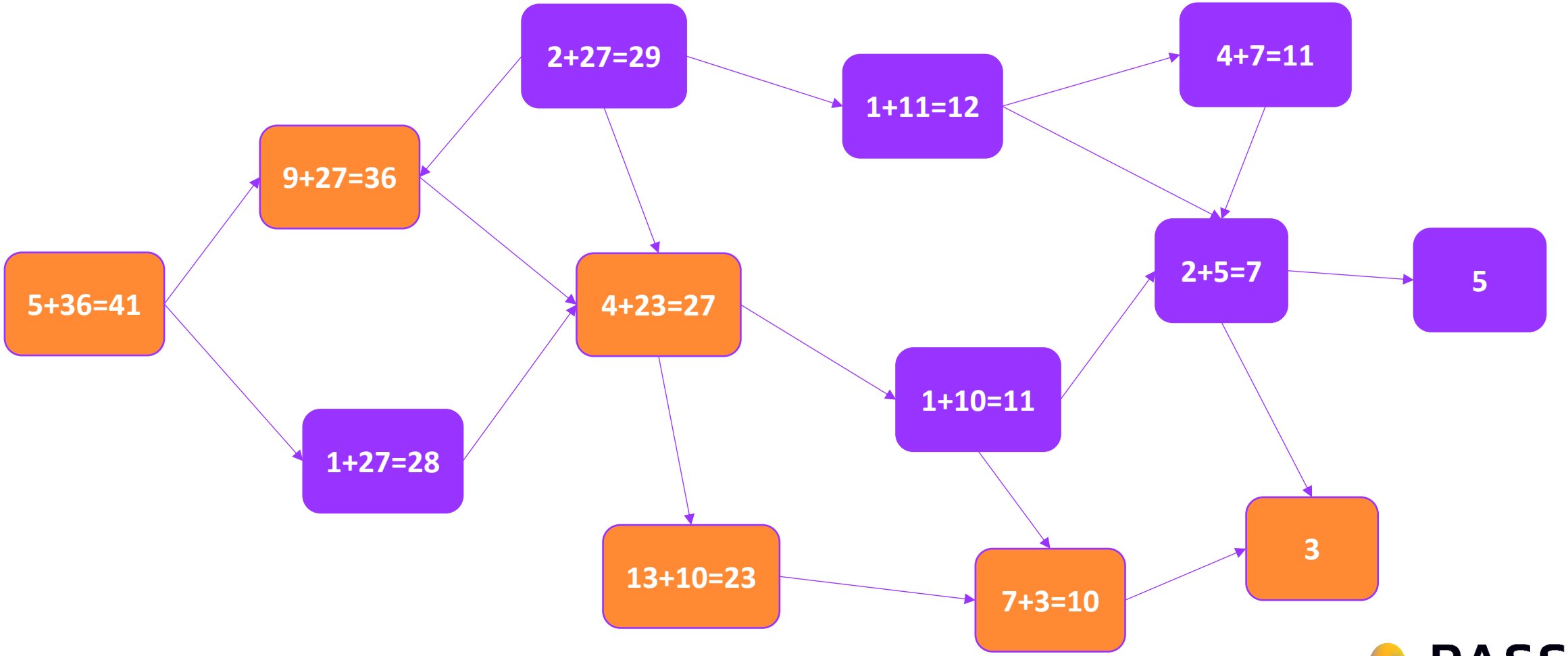
Dependent Task Calculation - BFS



Longest Dependency – Finding the Critical Path



Longest Dependency – Finding the Critical Path



Thank you

Let’s go faster together!

Name goes here

 GitHub (Foundatum)

 My Website (foundatum.ca)

 LinkedIn (jarid-mckenzie)

Your feedback is important to us

Evaluate this session at:

www.PASSDataCommunitySummit.com/evaluation