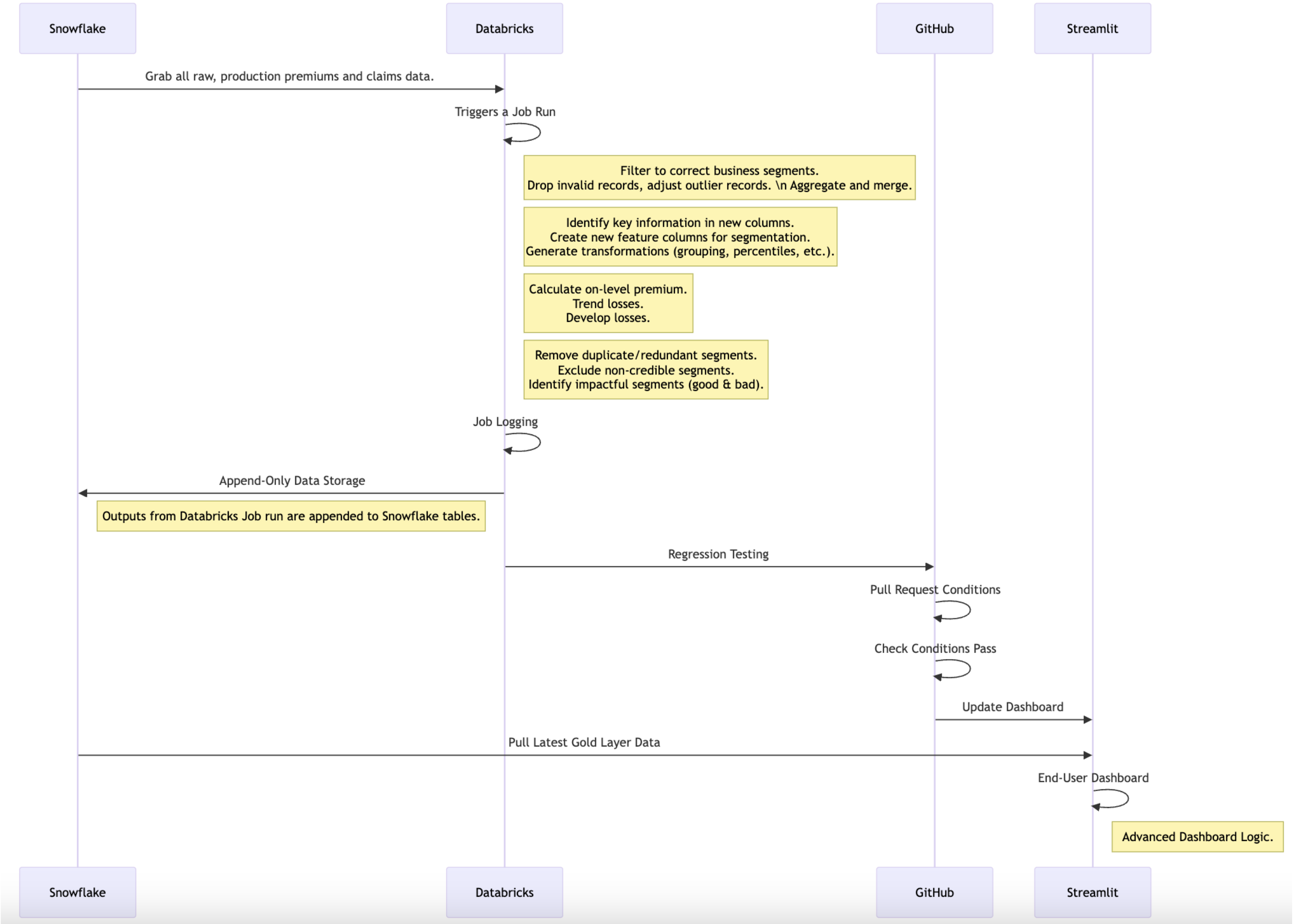
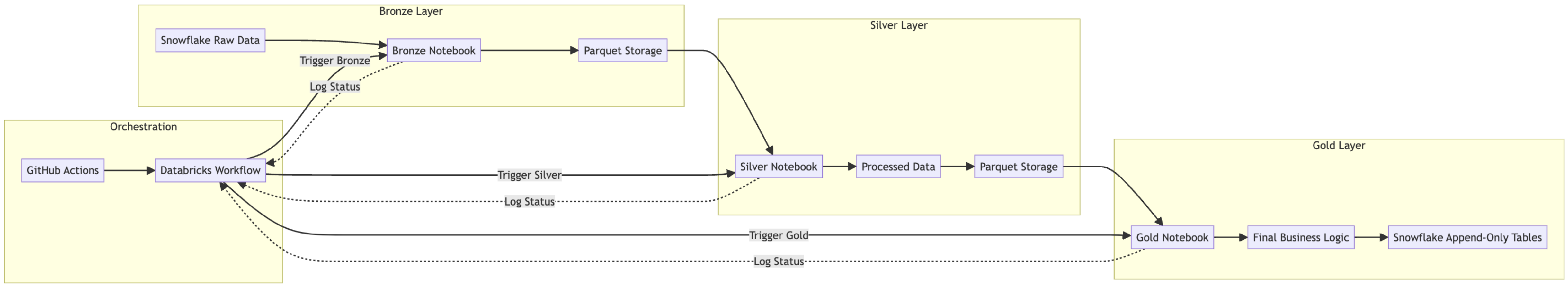
Sample Sequence Diagram of E2E Architecture



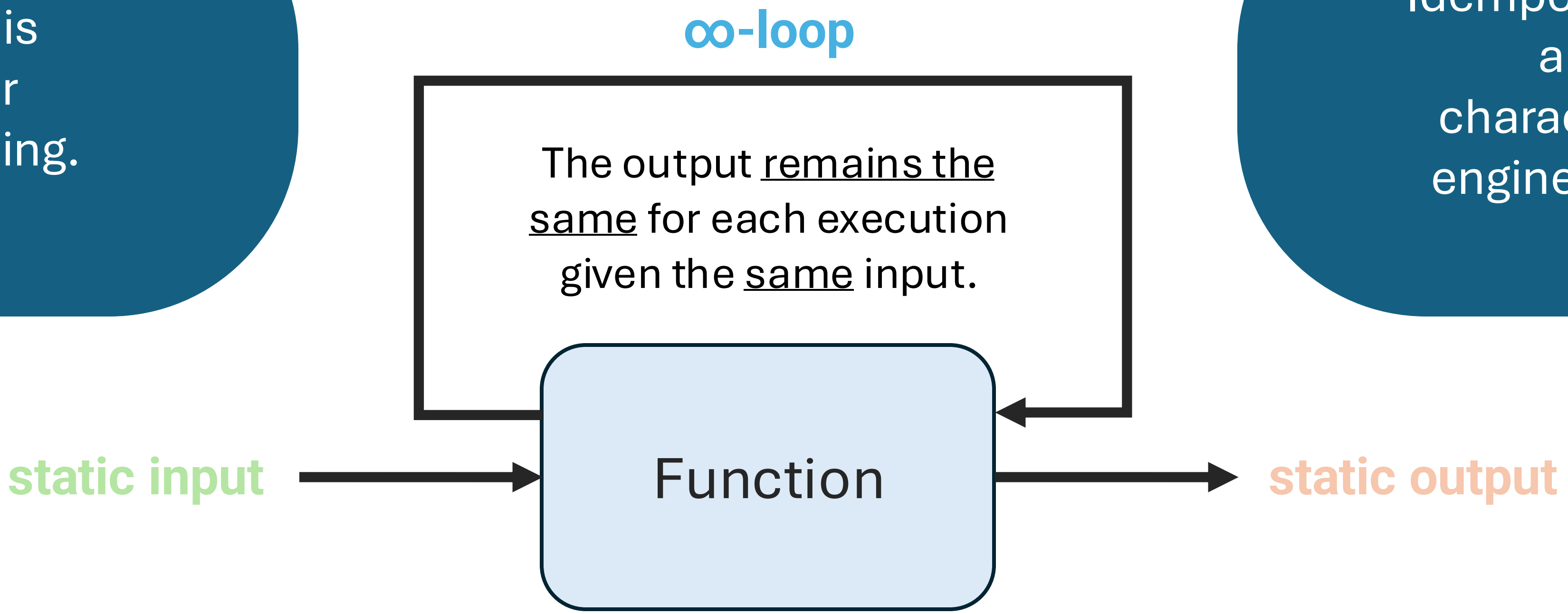
Sample Flow Chart of Medallion Architecture



Idempotency

Idempotency is necessary for application testing.

Idempotency is typically an essential characteristic of data engineering pipelines.



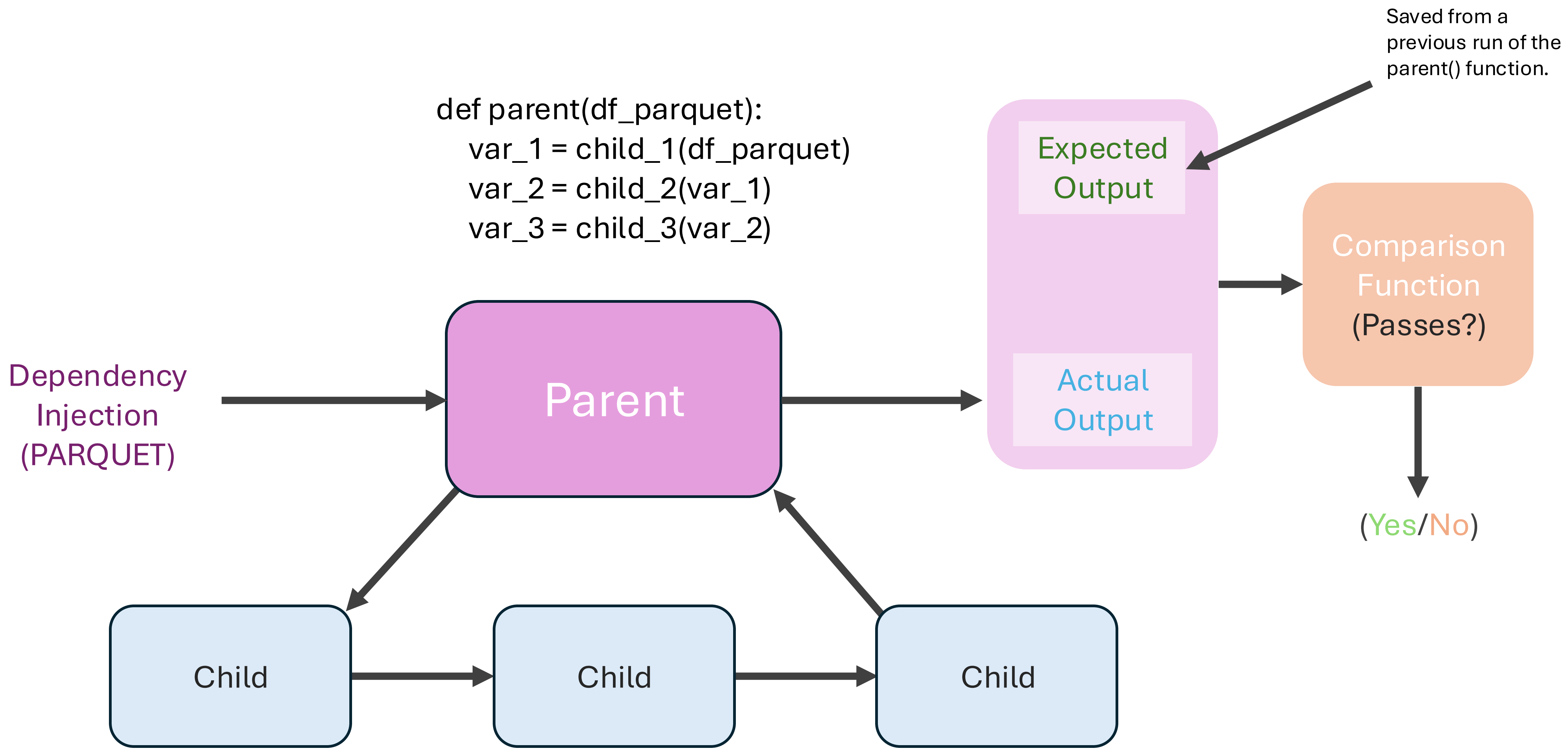
Non-Idempotent

```
def func(float_val):
    float_val += random_int()
    return float_val
```

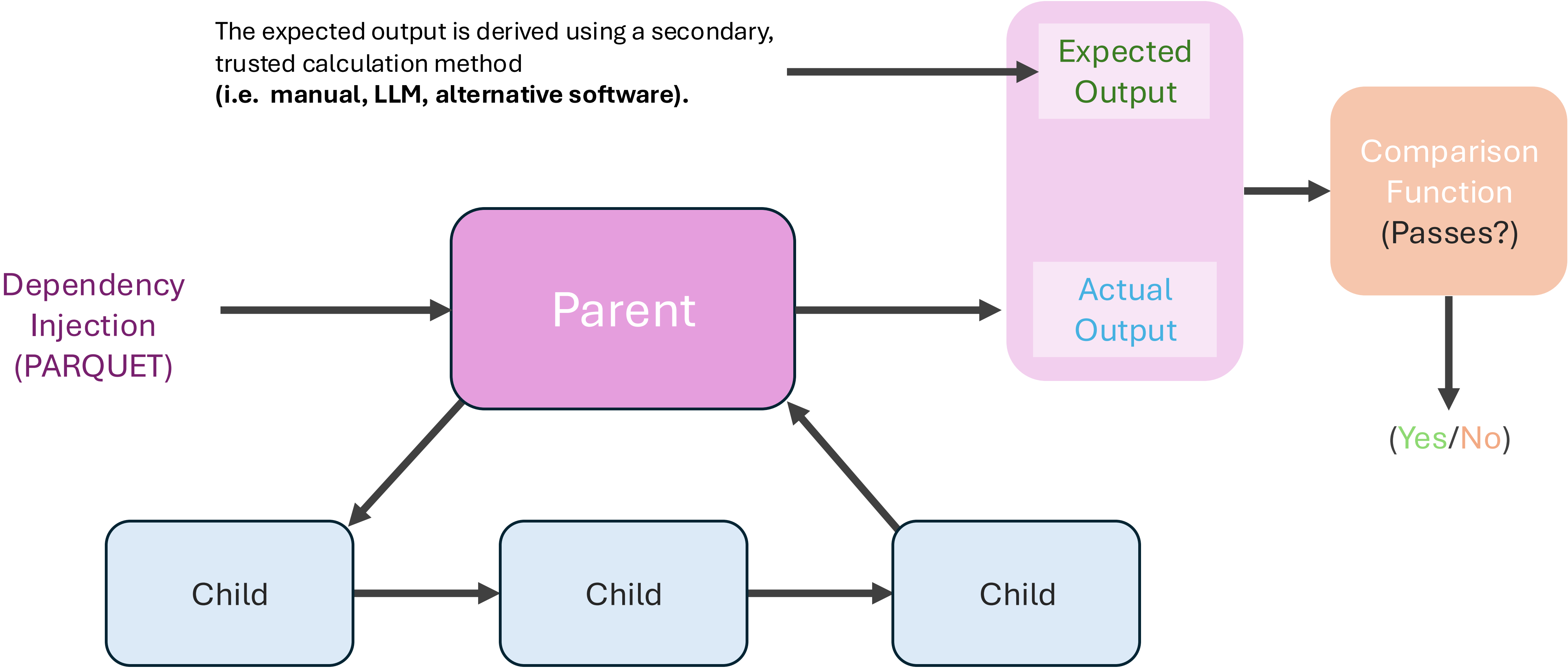
Idempotent

```
def func(float_val):
    return float_val
```

<u>Integration</u> Snapshot Testing (Analytics Engineering)



Integration Unit Testing



Alternative Testing

"Good software is well-tested software."

1. SMOKE TESTING

Does it run?

2. PERFORMANCE TESTING

Is it fast?

3. UI TESTING

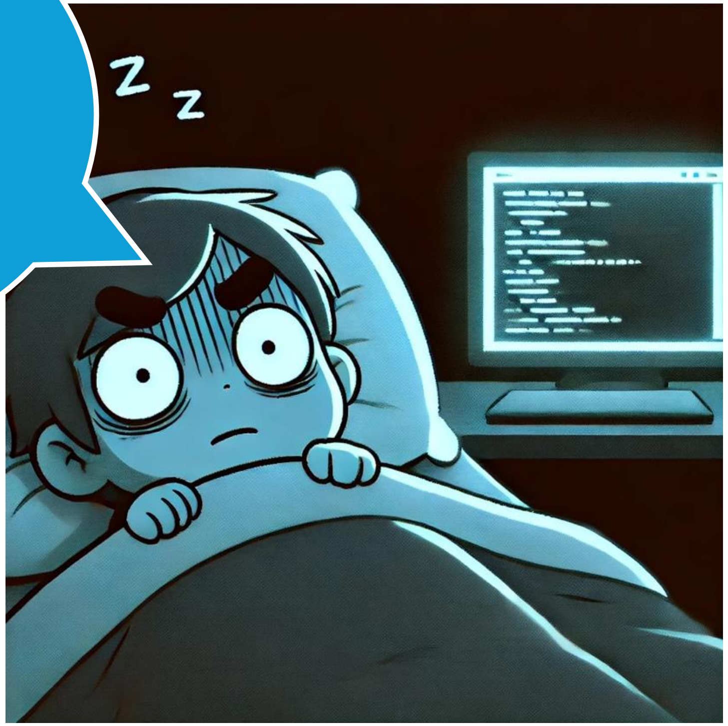
Does the UI operate as expected?

4. PROPERTY TESTING

Does it fit within the bounds of an expected output?

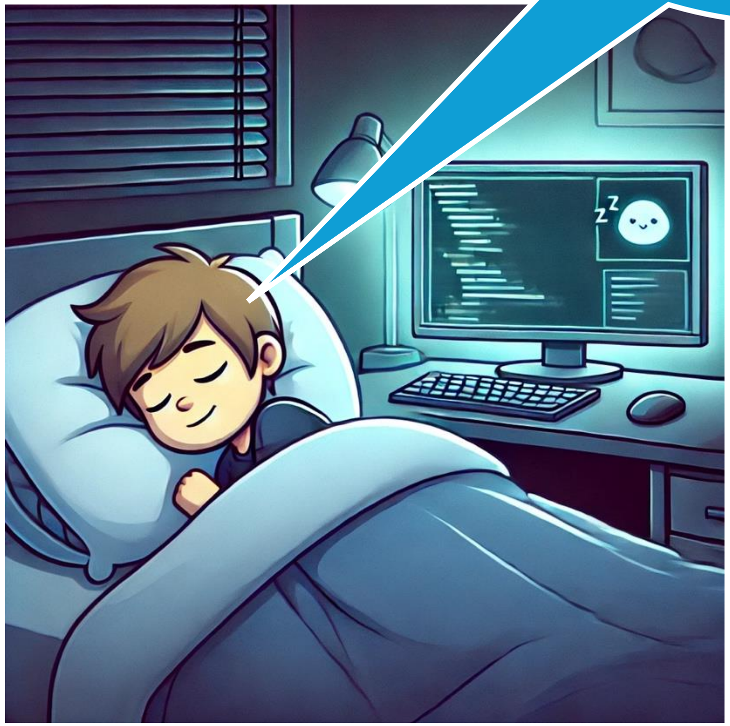
Snapshot Testing

- Probably not conceptually validated (using a previous out as a test case).
- **Fickle:** Needs to be changed often.
- Must be updated with each functional change.
- Verifies that only non-functional changes have occurred (i.e., refactors).
- Should only be updated with peer review.

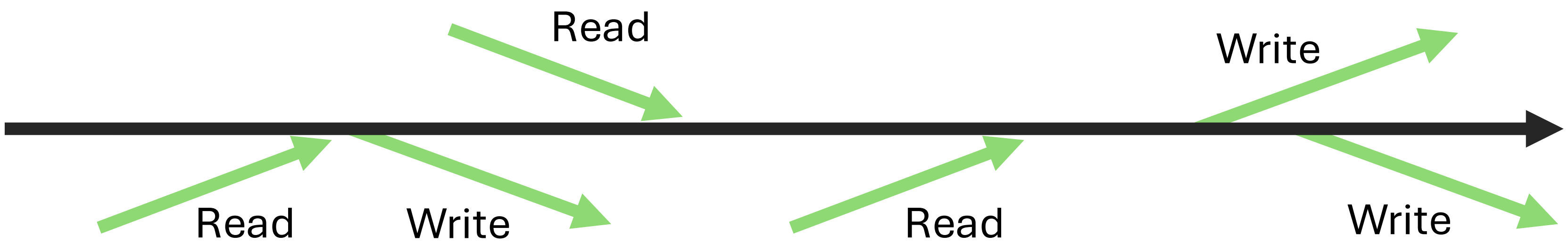


Unit Testing

- Conceptually validates a function is performing correctly.
- Ideally, these are written for critical building blocks of a software.
- Helps developers sleep at night.

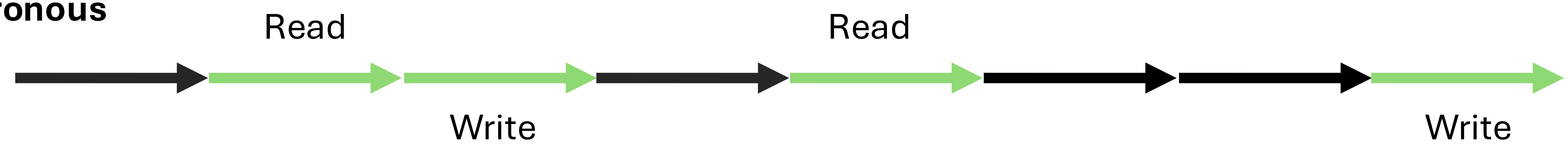


Asynchronous

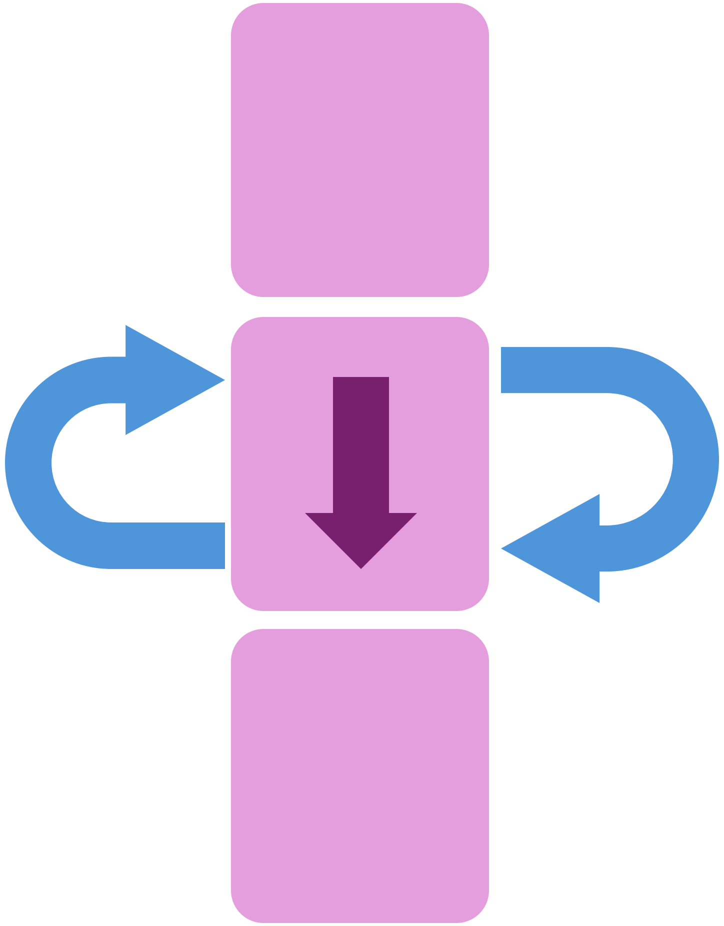


Asynchronous reads and writes run operations on separate threads decrease the runtime of the data pipeline.

Synchronous

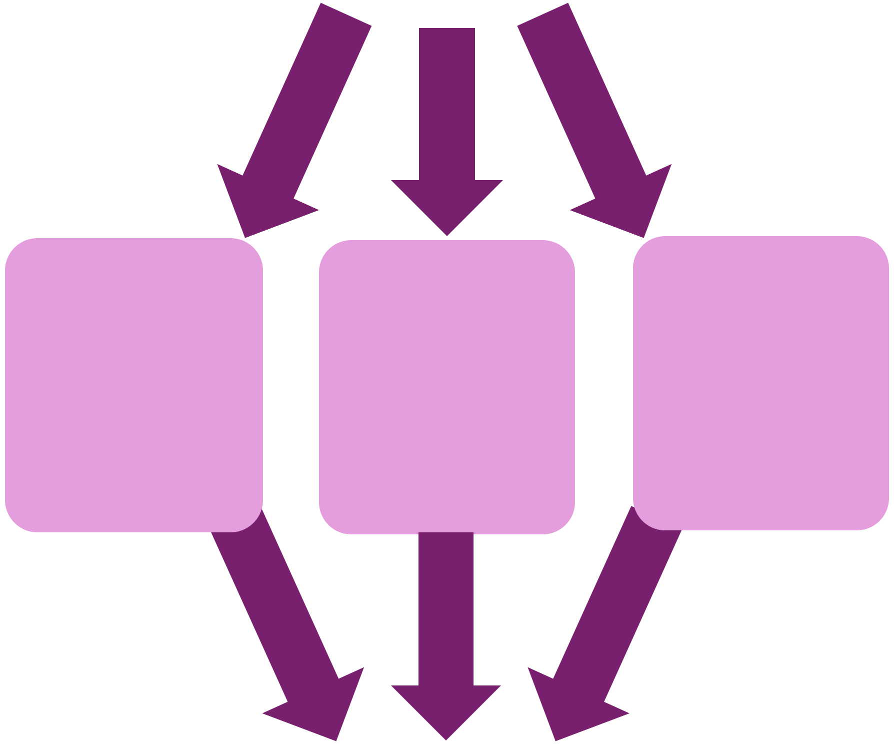


Chunking



Time Complexity: O(n)
Space Complexity: O(1)

Parallelization

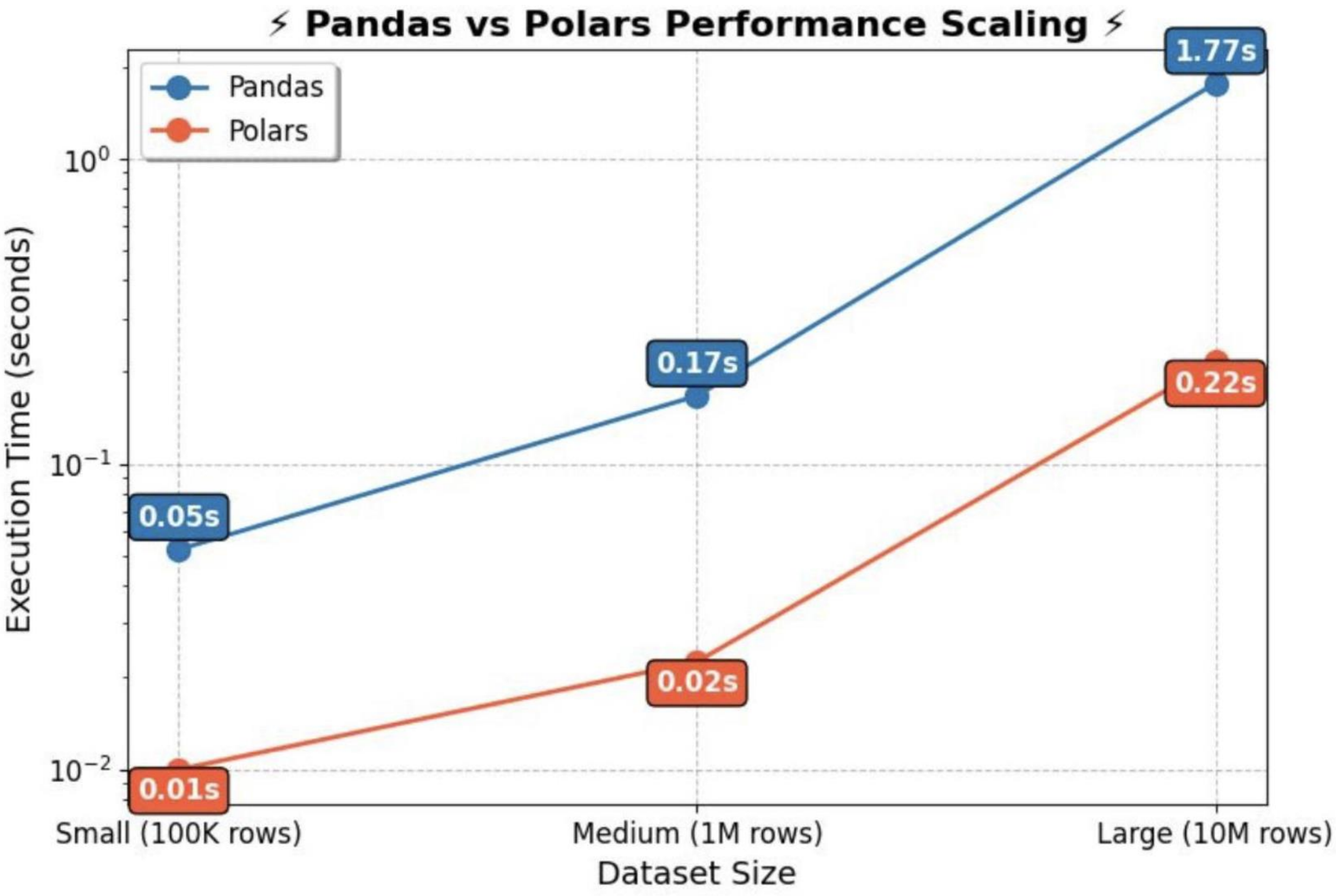


Time Complexity: O(1)
Space Complexity: O(n)

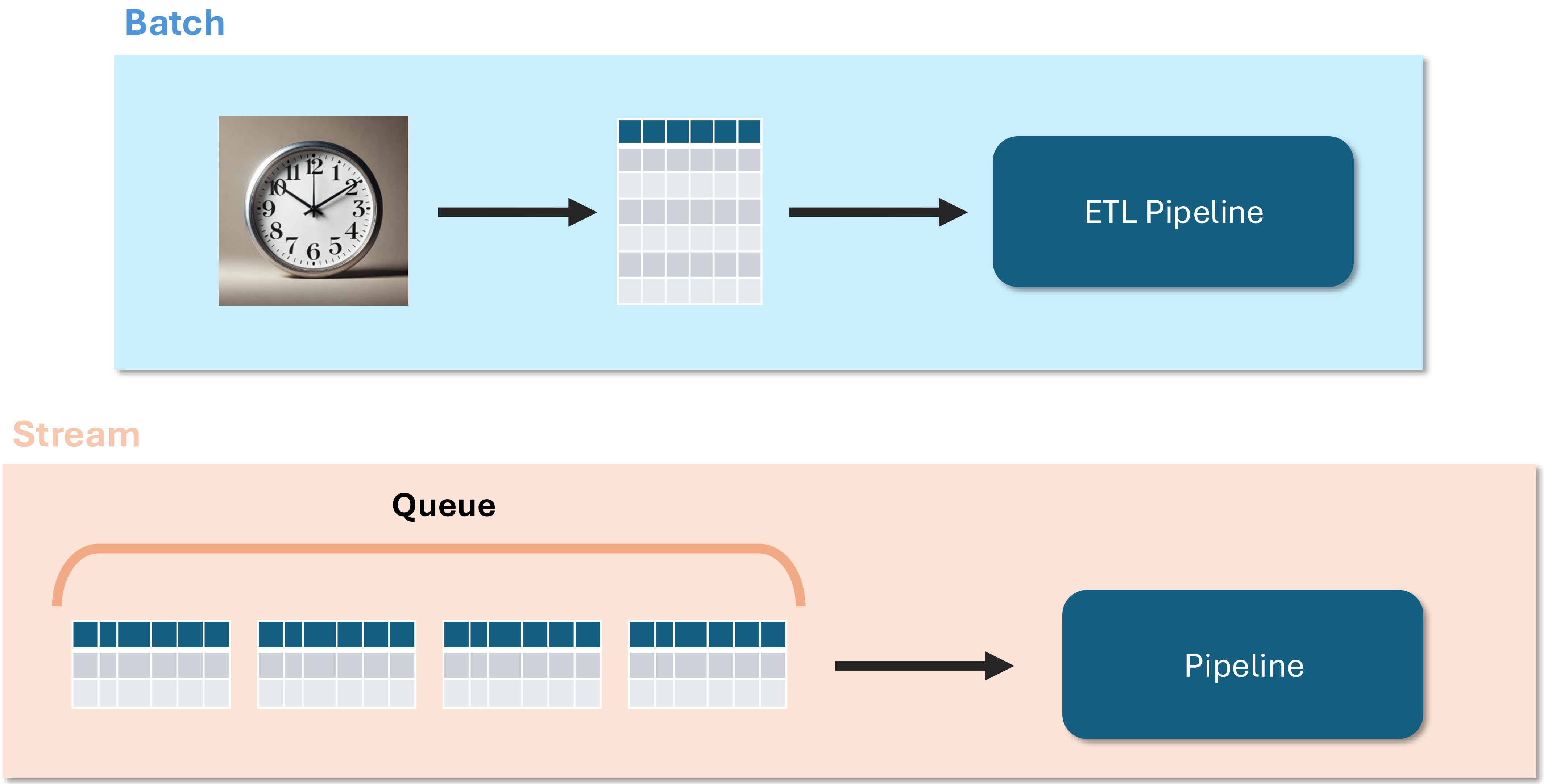
Programming Language Trade-Offs

Use Case	Pandas	Polars	PySpark	SQL
Small datasets (<10M rows)	✅ Best	✅ Faster	❌ Overkill	✅ Good for querying
Large datasets (>10M rows, fits in RAM)	❌ Memory issues	✅ Efficient	✅ Good	✅ Good
Big Data (TB-scale, distributed)	❌ Impossible	❌ Limited	✅ Best	✅ Best
Parallel processing	❌ Single-threaded	✅ Multi-threaded	✅ Distributed	✅ Query optimizations
Complex ETL	✅ Simple	✅ Efficient	✅ Distributed pipelines	✅ SQL transformations
ML/Statistical modeling	✅ Best for ML	✅ Works	❌ Spark ML (limited)	❌ Not ideal
Cloud-based processing	❌ Local only	❌ Local only	✅ Cloud (Databricks, EMR)	✅ Cloud-native (BigQuery, Snowflake)

Pandas VS. Polars



Batch or Stream Processing



Vectorization

PyArrow
Numpy
Pandas

CI/CD

INTERNET COMPUTER

Cloud Provider as State Machine Replication for Byzantine Fault Tolerance

INTERNET COMPUTER

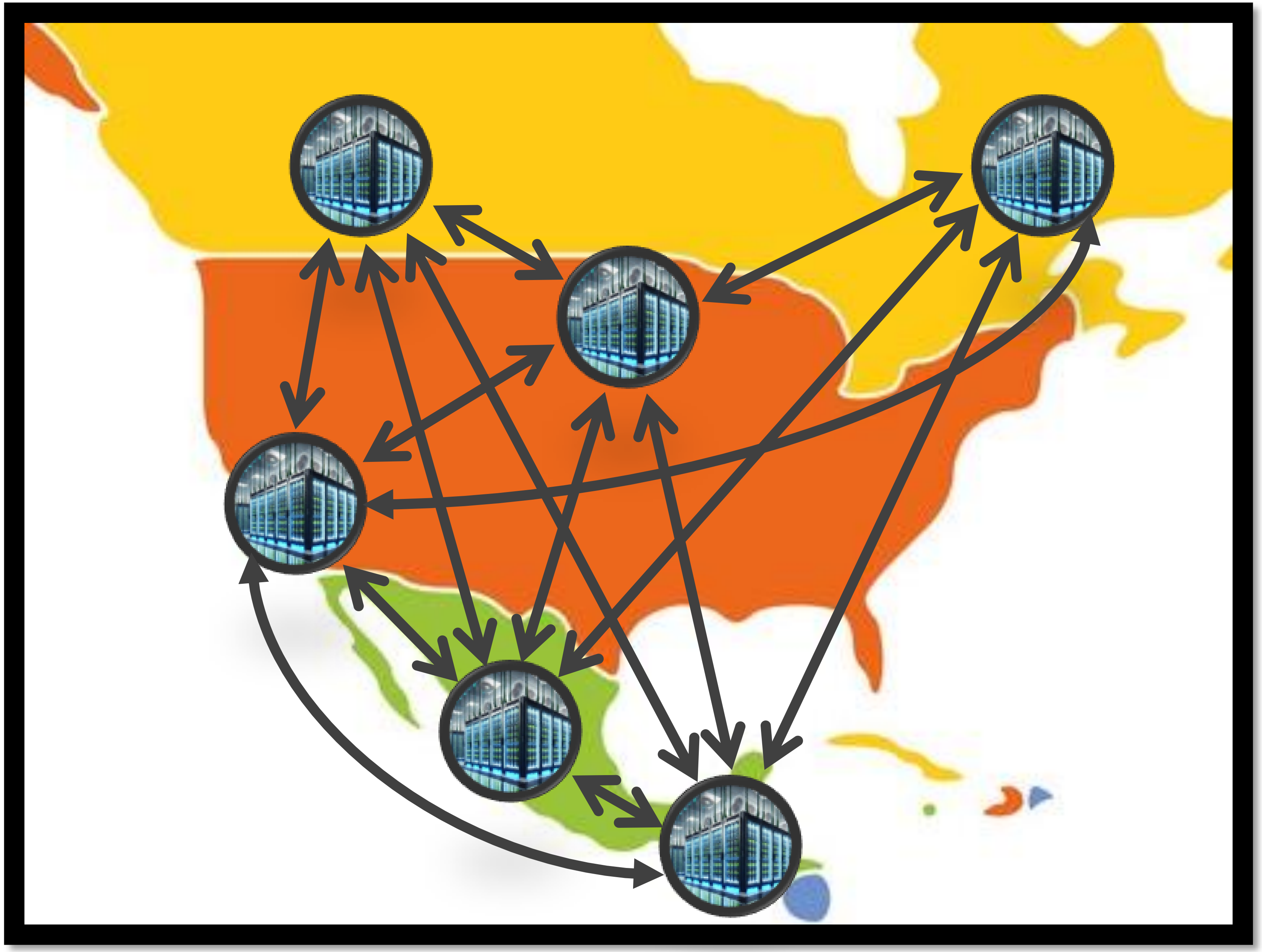
Subnets consist of 13 or more nodes. This allows for variable decentralization.

Latency is < 2 seconds.

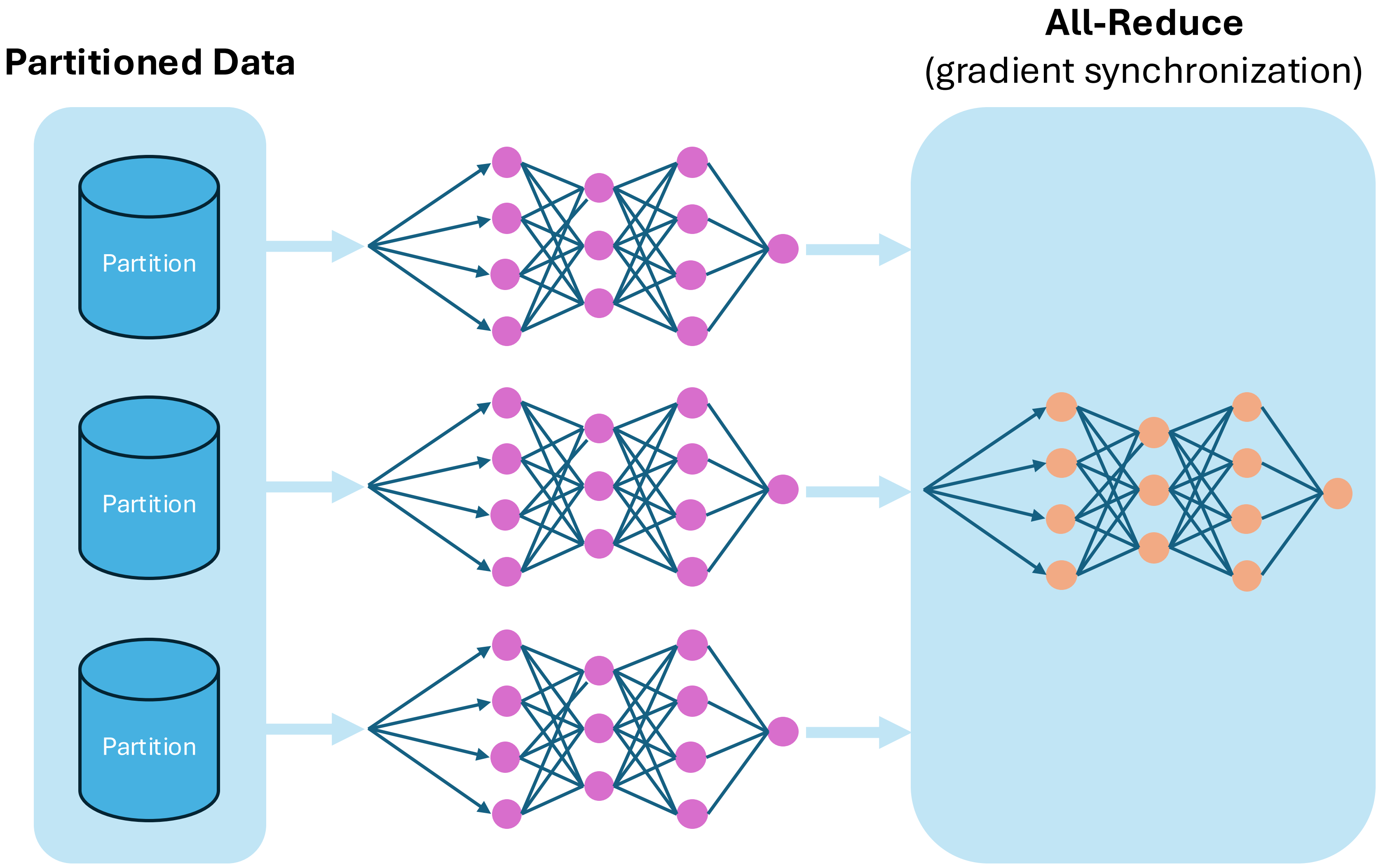
Egress (query calls) doesn't need to go through consensus.

Reverse gas: Developers pay for gas fees, and end-users can use the blockchain without needing to pay in ICP (like a normal website).

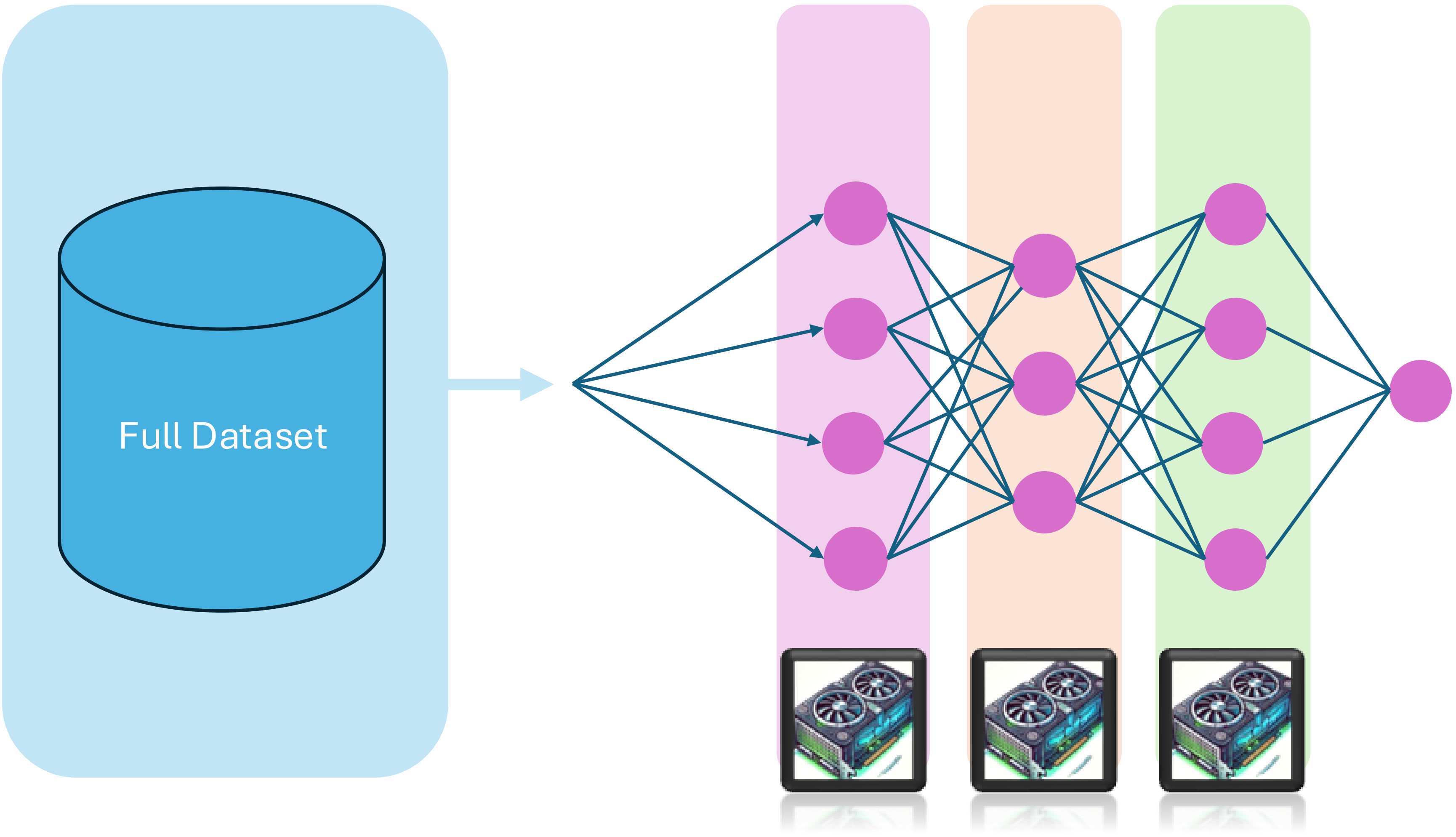
Software becomes immune to cyber hacks or datacenter outages due to any cause of failure.



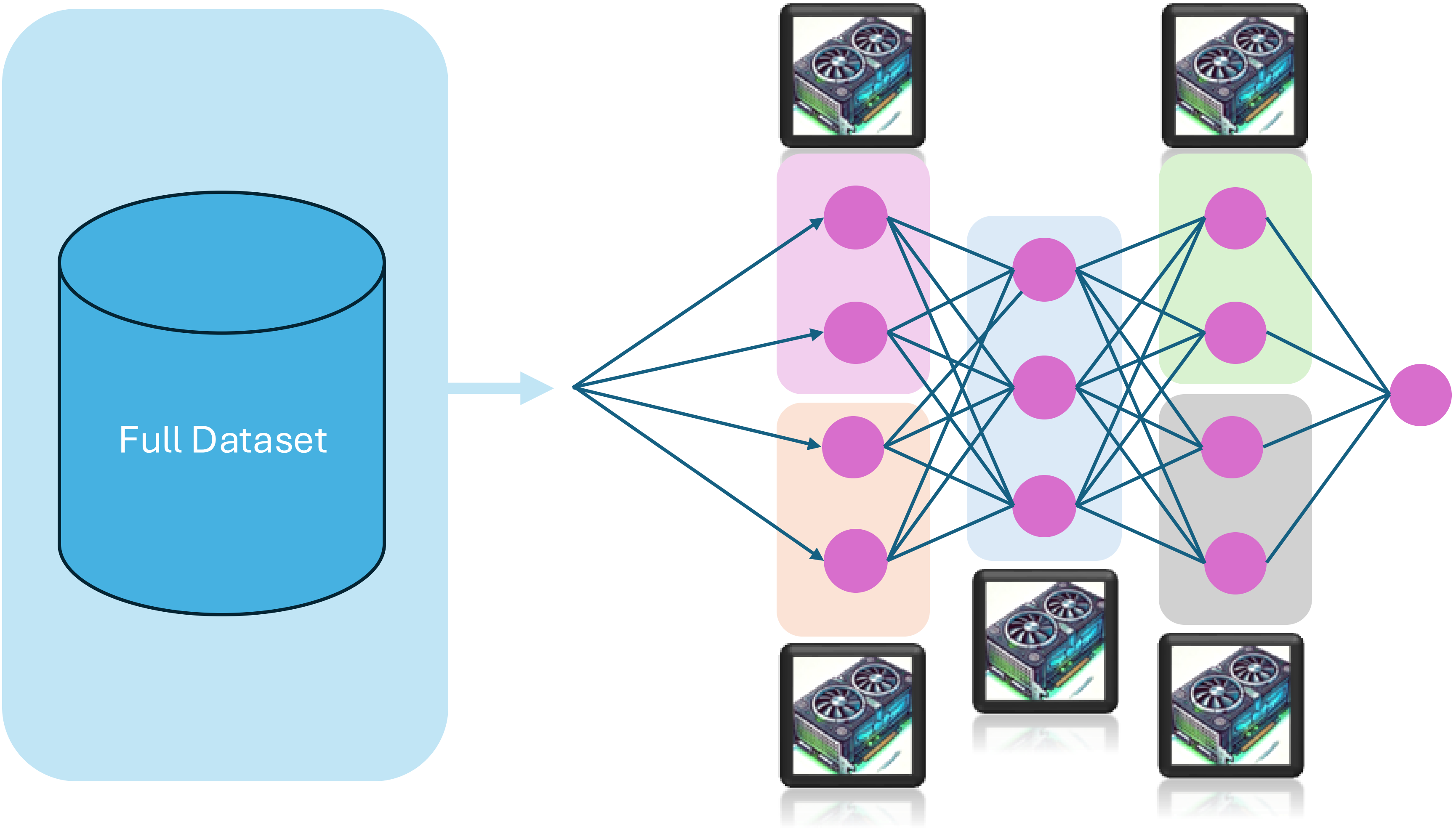
Distributed Data Parallel Training



Model Parallel Training



Tensor Parallel Training



RAG: Retrieval Augmented Generation

Example DL Project