# Perelyn

# Bring your DataOps to the Data Lakehouse

# Benefits & Challenges

#Conf42  #DevOps  #2025

Jorge Loaiciga-Rodriguez

*Senior Platform Engineer / Consultant @ Perelyn*

# Introduction

## Companies face recurring <u>challenges</u> in data projects

- Inconsistent data quality
- delayed insights
- poor alignment between stakeholder/developers
- etc.

!

## Applying <u>DataOps</u> and embracing <u>Data Lakehouse</u> architectures isn't just another fix — it's a <u>paradigm shift</u> that systematically reduces these challenges <u>while driving</u> insight delivery and trust in data.

?

# Learning Outcomes

## 1 Understanding

An understanding of <u>what</u> DataOps is, <u>why</u> it matters and <u>how</u> it approaches problems in modern data architectures.

## 2 Perspective

An appreciation for the huge <u>opportunity</u> that DataOps represents for your organisation or carrier.

# About Myself

**+6** **Years** of experience as data professional

**12** **Orgs** supported[1] in their IT transformation journey

**+24** **Months** in love with applied DataOps principles

Jorge Loaiciga-Rodriguez
Senior Platform Engineer / Consultant

jorge.loaiciga-rodriguez@perelyn.com

[1] Consultation & Implementation

# Agenda

| 1 | Case-Study | 2 | Key Concepts | 3 | Adoption Challenges | 4 | Conclusions |
|---|---|---|---|---|---|---|---|

FICSIT Incorporated

- The issues addressed by DataOps
- The grand vision behind DataOps

- Cost
- Performance
- Under-standability

- Technical
- Organisational

- Summary
- What comes next?

# Understanding the Problem
## The story of FICSIT Inc.

**FICSIT**

"FICSIT Inc. is cost-effective & efficient. We do not waste"

- **Headquarters on Earth[1]**

- **500k+ employees**

- **R&D, Aerospace Engineering & Mining**

- **Data-driven and cutting-edge**
  - 1st Chief Data Officer launched digital transformation program to drive growth
  - Central data department with ~500 engineers
  - Investment in Cloud-native compute/storage, and Martian BI tooling
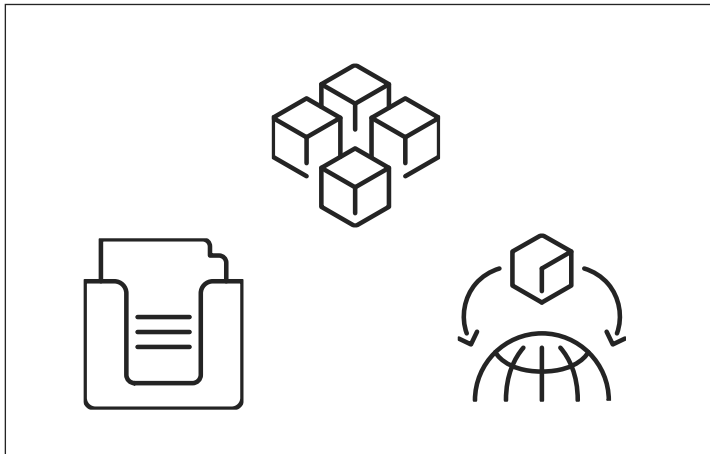
[1] Allegedly

# Current Tools

## IT Department



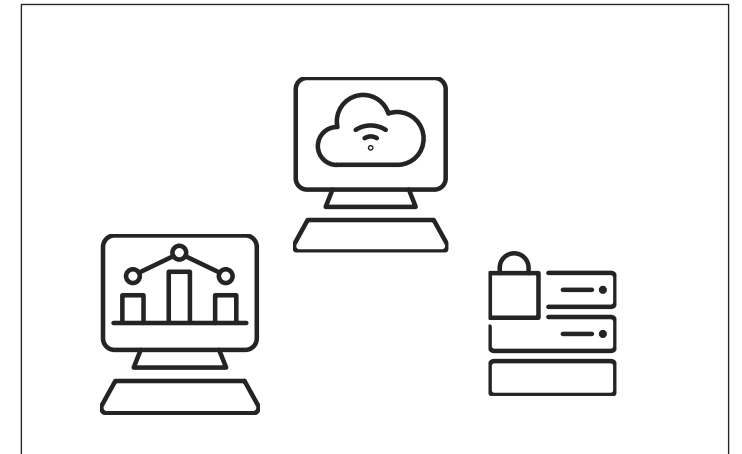## Data Engineering Department
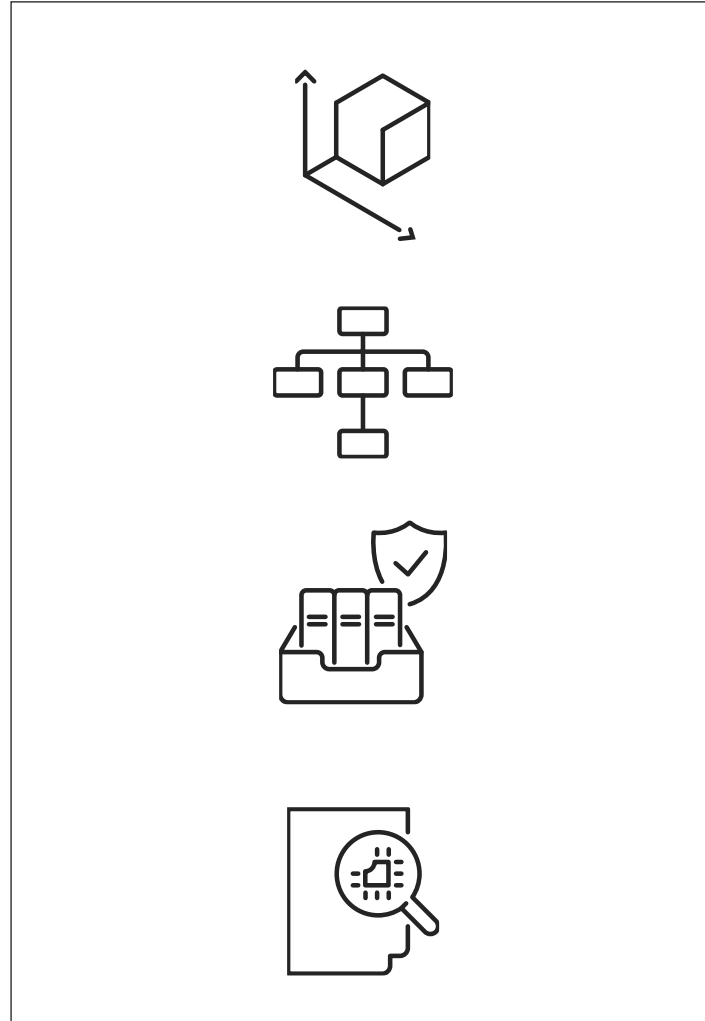


## Customer Support Department



## Manufacturing Department



## Data Science Department

# Example Workflow:
## Customer Support Reviews

**FICSYT**

**1** Customer feels a feedback form for a technical support case that went badly

**2** Data is stored in an RDS, managed by the Customer Success Dep.

**3** Data Engineering Dep. built a data pipeline to copy data every hour to the data lake

**5** Data Science Dep. sends a BI report with findings to Manufacturing Dep.

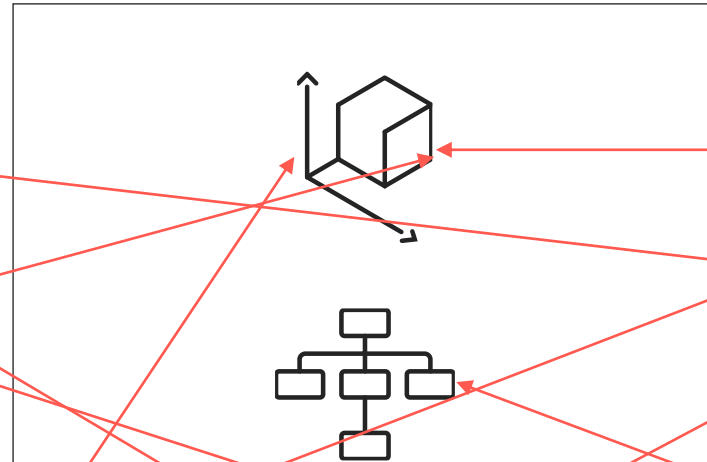**4** Data Science Dep. built model to predict the sentiment of reviews and export predictions
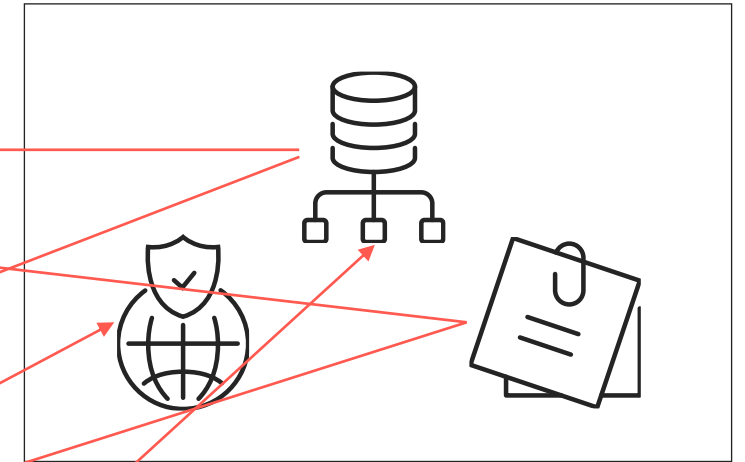
# Scaling up the Workflow Count
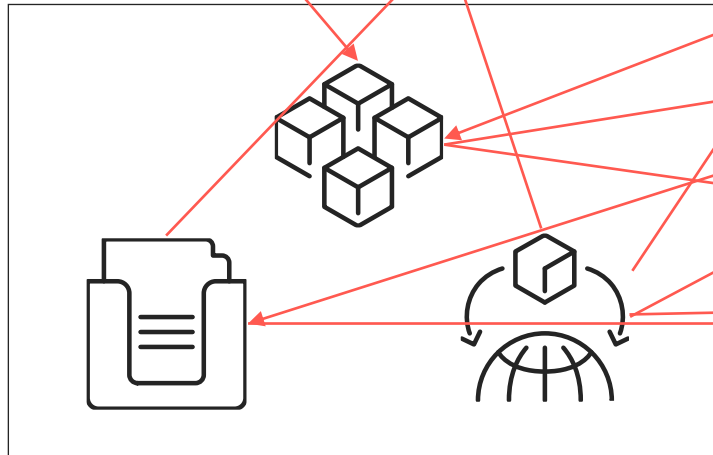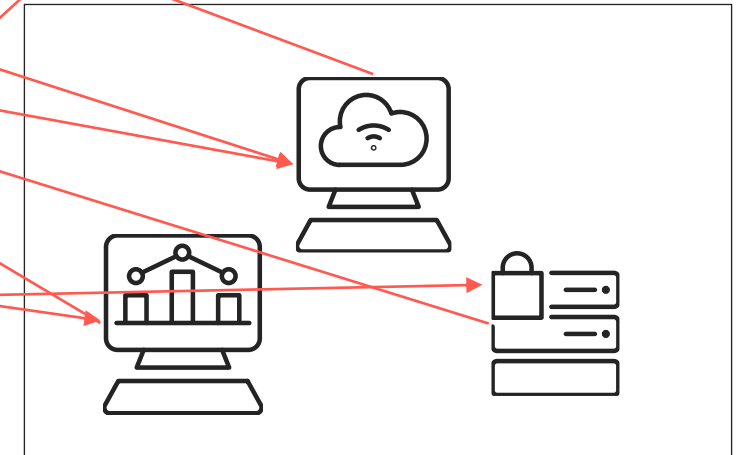


IT Department
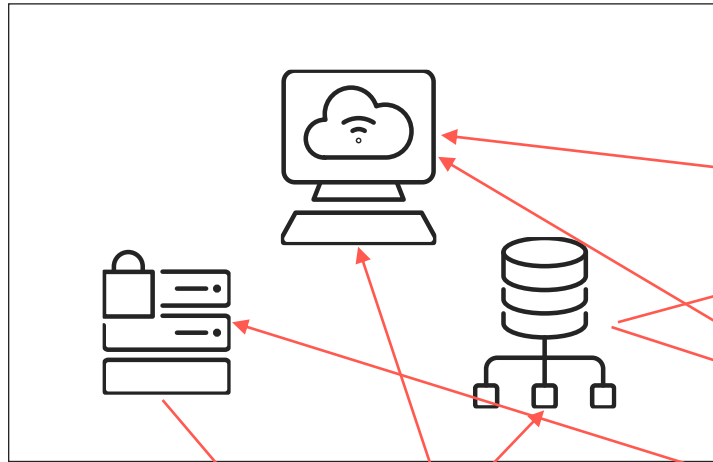
Data Engineering Department

Customer Support Department

Manufacturing Department

Data Science Department

# "The IT architecture grew organically, there are data silos everywhere"

— Chief Executive Officer

**"We maintain hundreds of pipelines, moving data between all departments, cyclically dependent even!"**

– Head of Data Engineering

"**Many obscure column names and transformations across data stores.
6 months in and onboarding is still going**"

— Data Science Newest Recruit

"**Too many systems, all with different permission schemes. The monthly bill is also unpredictable**"

— Head of IT

"Can I trust data in that dashboard. Last time I checked it looked funny. It also takes forever to load"

— Managers (Dashboard Users)

# Data Management Capability Assessment

| 1 Data Ingestion | 2 Data Storage | 3 Data Modeling | 4 Data Reporting |
|---|---|---|---|
| - No data silos | - 4 data silos | - 2 data silos | - 1 data silo |
| - 1 data copy | - 4 data copies | - 5 data copies | - 5 data copies |
| - 2 data formats | - 3 data formats | - 4 data formats | - 5 data formats |
| - 1 access control scheme | - 2 access control scheme | - 3 access control scheme | - 4 access control scheme |
| - No metadata management | - No metadata management | - 2 metadata management panels | - 3 metadata management panels |
| - 6 orchestration & monitoring panels | - 4 orchestration & monitoring panels | - 2 orchestration & monitoring panels | - 1 orchestration & monitoring panels |
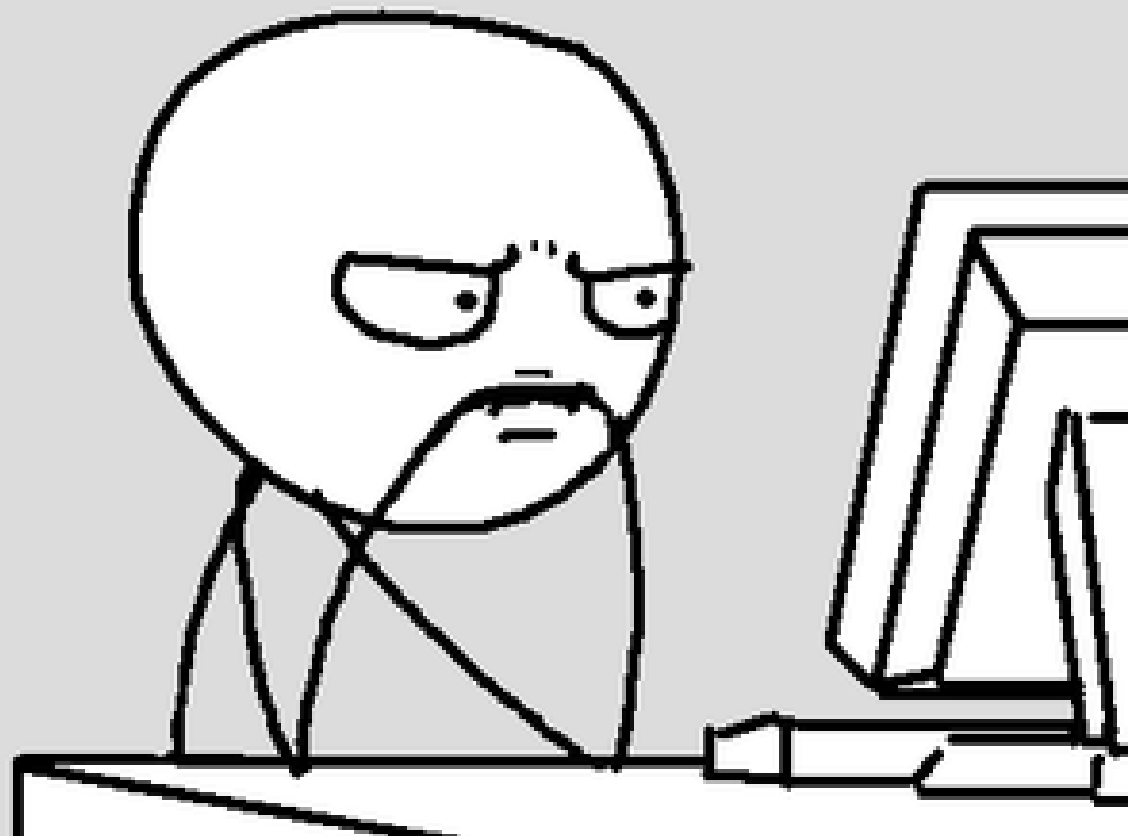
"What if we introduced <u>DataOps</u> and molded into a <u>Data Lakehouse</u> architecture?"

# Ideal Scenario

| 1 | Data Ingestion | 2 | Data Storage | 3 | Data Modeling | 4 | Data Reporting |
|---|---|---|---|---|---|---|---|

1 lake

1 data copy[1]

1 central panel for metadata management

1 data format (open-source)

1 central orchestration & monitoring panel

1 central access control scheme

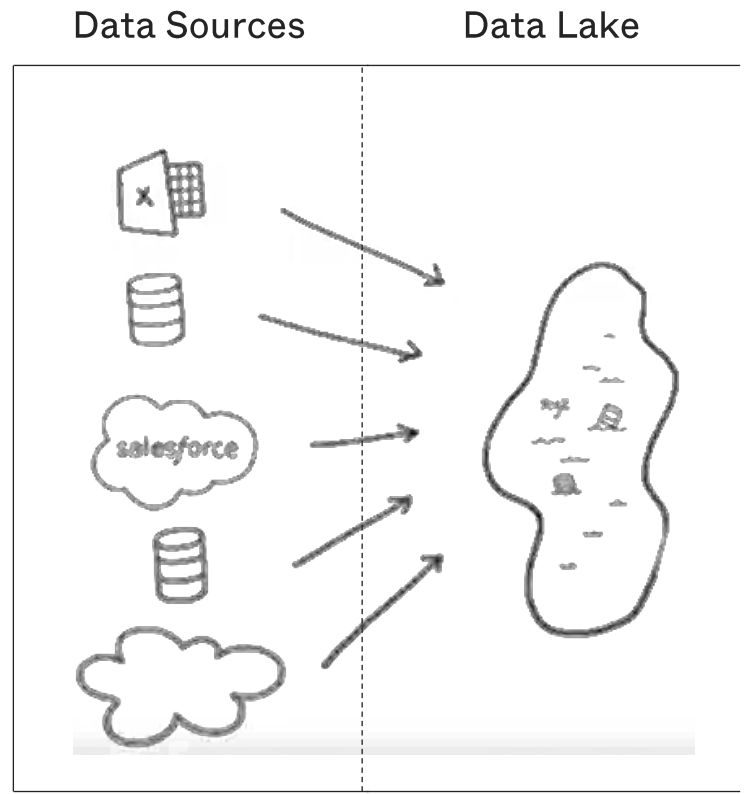[1] Not counting backups for disaster recovery

# How are the following affected?

1. Costs

2. Performance

3. Understandability

## Transactional

Data Sources          Data Lake

Source: chartio.com/blog

# Storage & Syncing

- System integration tends to produce far <u>more unstructured/semi-structured data than structure data</u>.

- In the pre-cloud era, storage was expensive vs. In modern data platforms, <u>storage is cheap</u>.

- <u>Open-source data formats</u> like Iceberg and Delta Lake are now widely supported. Free high-performance read and write (columnar, binary, ACID-support).

# DataOps on Data Lakehouse

Transactional   Operational

Data Sources   Data Lake   Staging Schemas
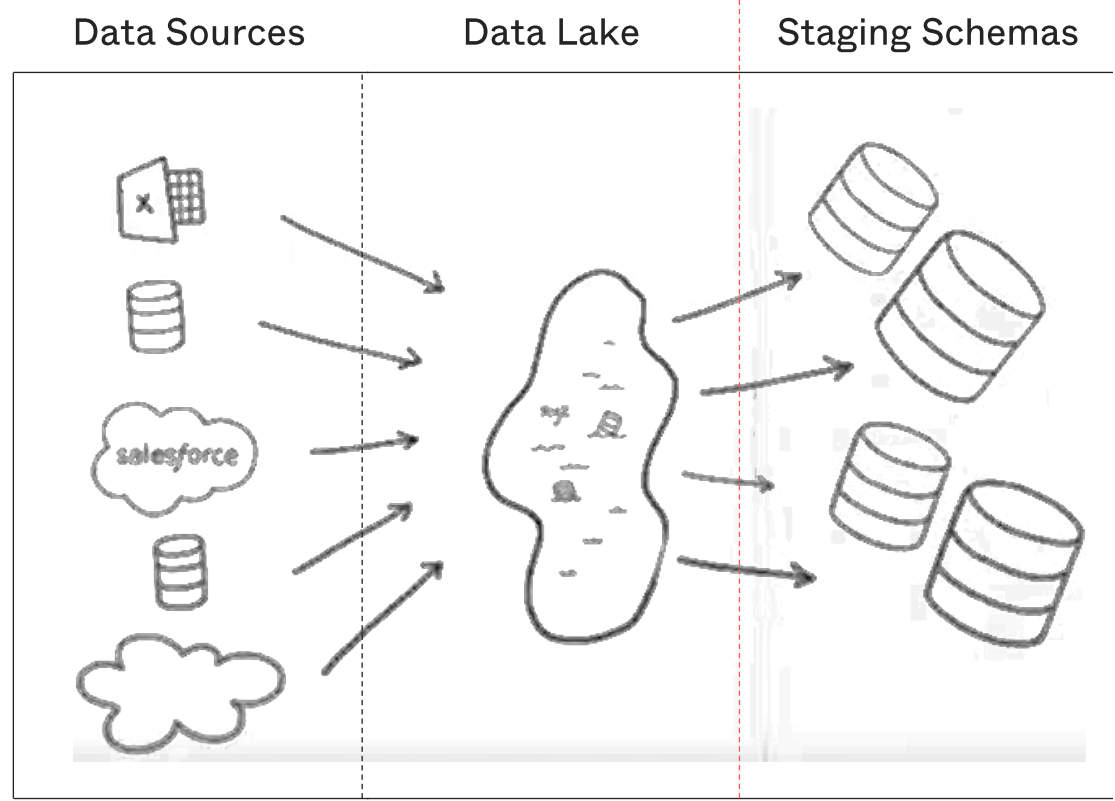


Source: chartio.com/blog

# Staging Schemas

- ■ **The strength of the system is introduced at this step**

- ■ **Well-define stages for each of the common data refining operations**

  1. Language translation
  2. Column renaming
  3. Column type formatting
  4. Storage standard format
  5. Storage partitioning & compression
  6. Model quality testing and freshness testing
  7. Business logic introduction
  8. Inter-dependency orchestration[1]

- ■ **Easy to build, troubleshoot and <span style="color:#FF5A4D">version</span>**

[1] With tools such as dbt
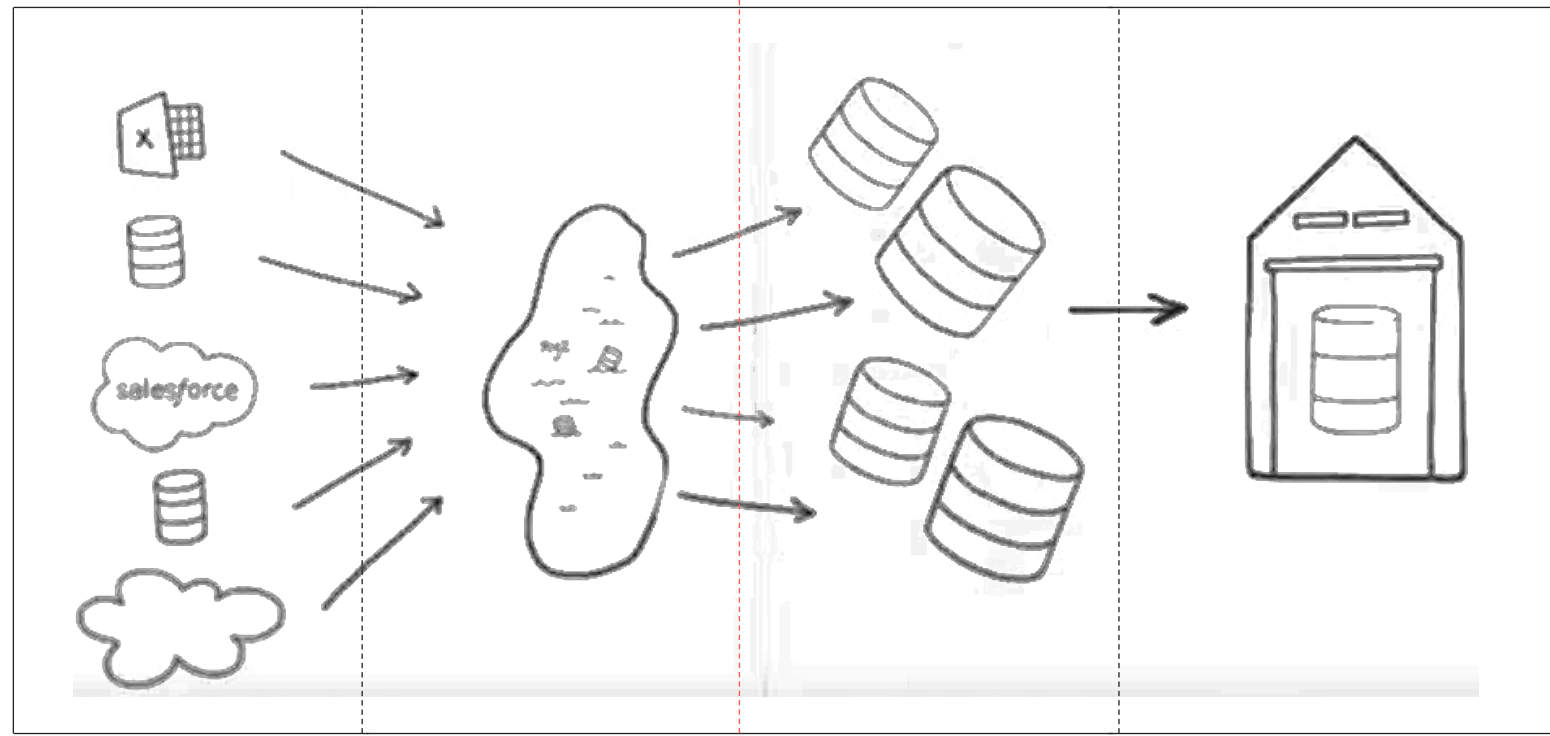
# DataOps on Data Lakehouse

Transactional

Operational

Data Sources

Data Lake

Staging Schemas

Data Warehouse

Source: chartio.com/blog

# Operational Data Lakehouse

- If staging schemas are perfect for data exploration, the warehouse is perfect for data exploitation.

- Offers the computation power needed to meet even the time-critical use-cases.

- Can introduce fancy materialization strategies (incremental loading of records), or Machine Learning-based transformations.

- <span style="color:red">Data</span> it reads and writes <span style="color:red">stay withing the one-lake</span>.

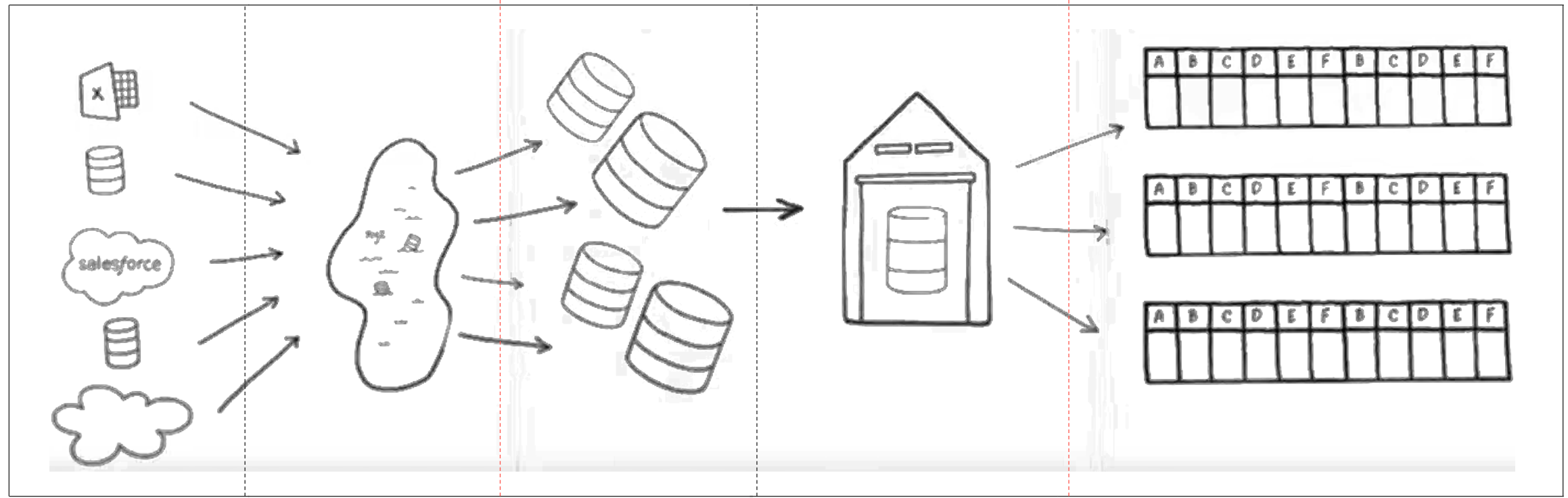# DataOps on Data Lakehouse

**Transactional**　　　　　**Operational**　　　　　**Analytical**

Data Sources　　　Data Lake　　　Staging Schemas　　　Data Warehouse　　　Wide Tables



Source: chartio.com/blog

# Data Consumption from Wide Tables

- 25-50% more performant that running joins[1]

- Extremely intuitive for analysts and business users

- BI tools tend to favor wide tables as input. Setup and maintainability of dashboards suffer otherwise.

※ You might end up with duplicated columns with different versions of the data values

[1] fivetran.com/blog

# DataOps on Data Lakehouse

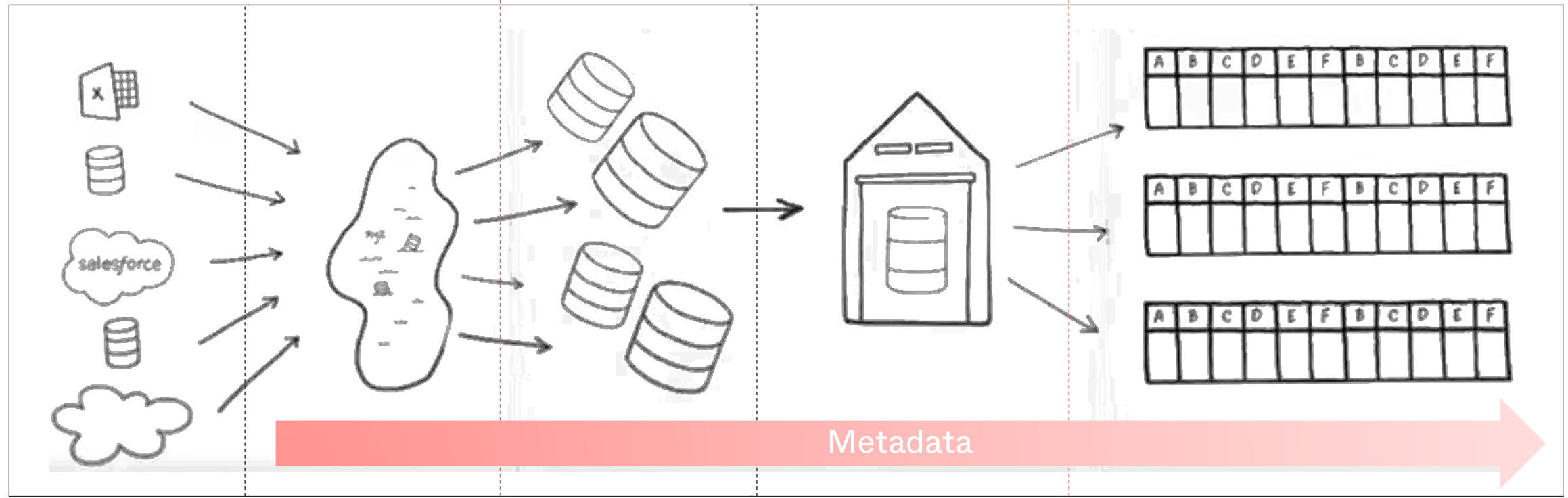Transactional          Operational          Analytical

Data Sources      Data Lake      Staging Schemas      Data Warehouse      Wide Tables



Metadata

Source: chartio.com/blog

# Typical Challenges of DataOps adoption

| Technical | Organizational |
|---|---|
| <ul><li>Existing long-running licences for enterprise tools.</li><li>Too much trivial testing (i.e testing only for not null and unique values)</li><li>Model version deprecation without proper communication to downstream users</li><li>Fail to create a mockup of new use-cases in coordination with stakeholders</li></ul> | <ul><li>The conversions report you built does not match with the one from that other team</li><li>Cool, but I am not a technical person…, why don't you talk to the engineers?</li><li>We've have worked with tool X for years. Now you want me change into Y?</li></ul> |

"**The only thing constant is change**"

— **Heraclitus of Ephesus**

# DataOps embraces change

- Business requirements change

- KPIs change

- Data changes

- Sources change

- Tools change

- People change

- Best-practices change
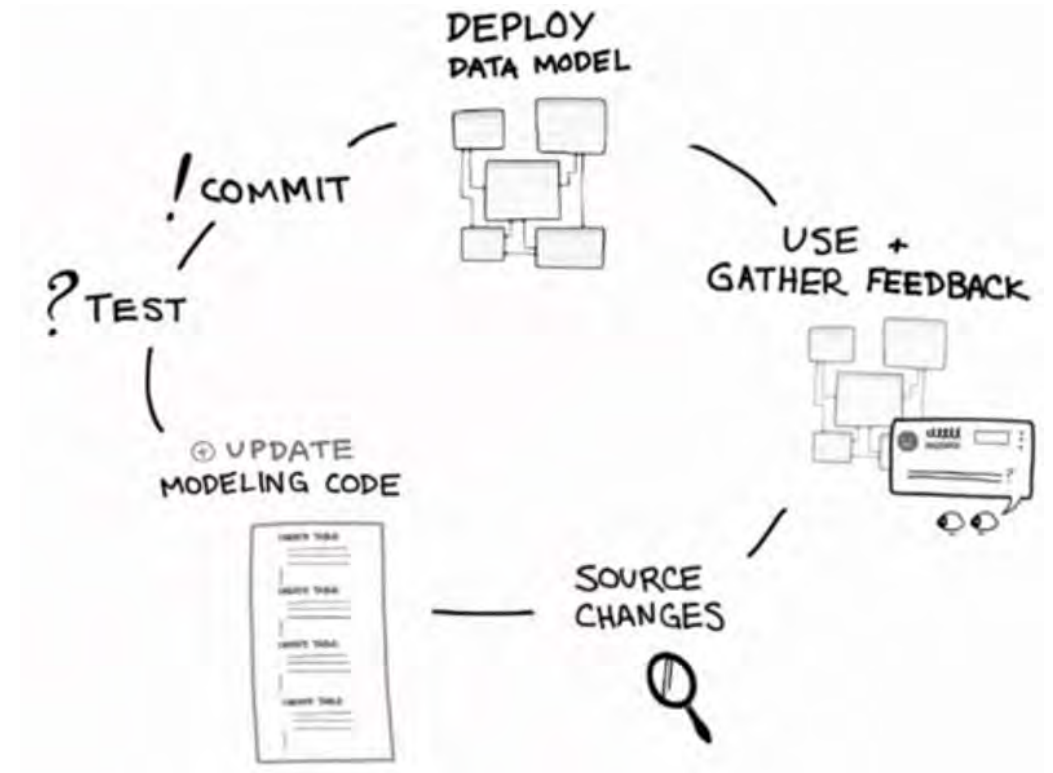


Source:

# Putting it all together

**Data Lakehouse & open-source storage formats** are a response to system integration and cloud billing-models.

**Cost-effective** thanks to cheap storage.

**Wide table(s)** are intuitive, performant, and even a requirement for data consumption tools.

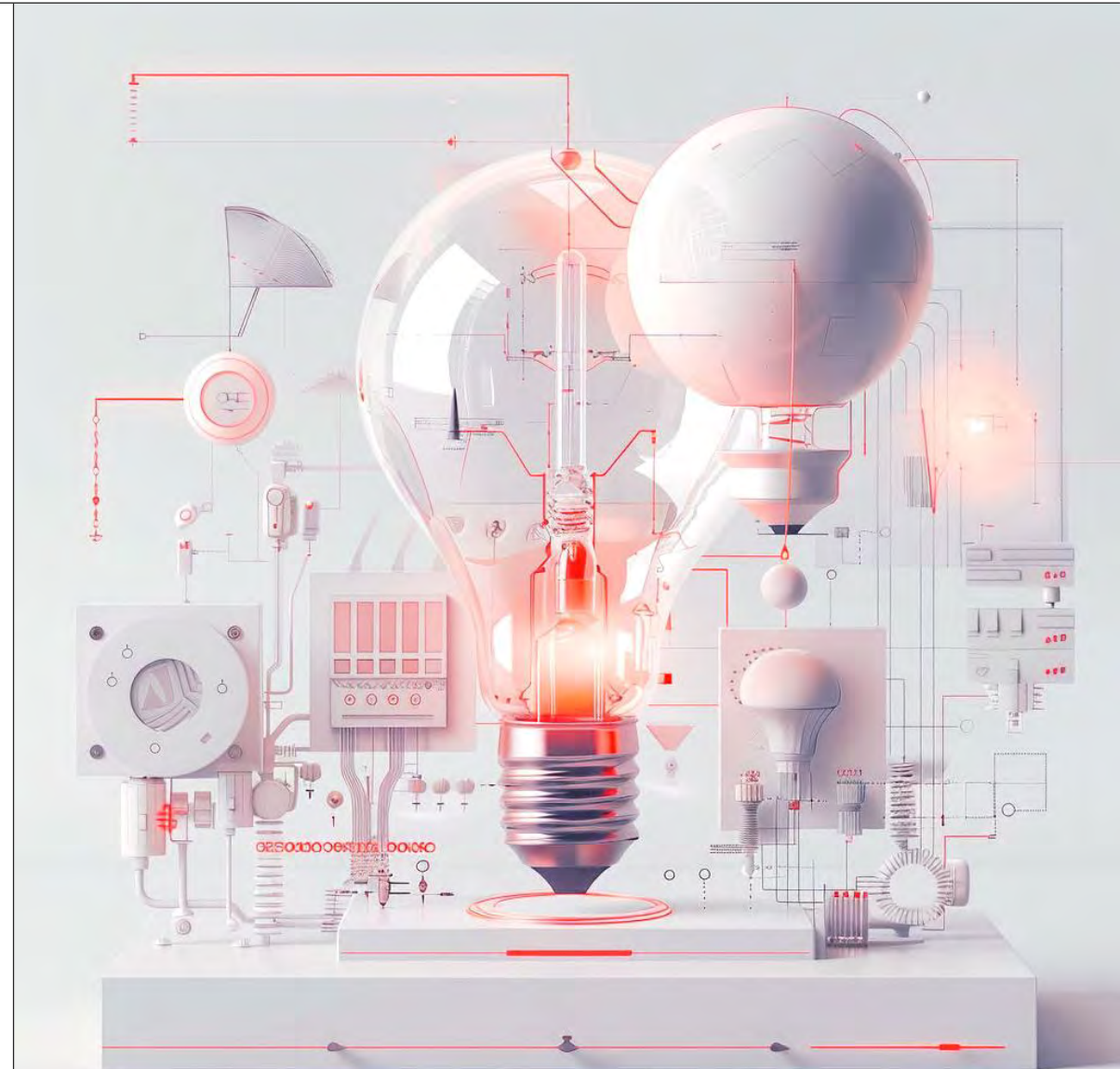Its downside is outweighed by its benefits and **performance**.

**Metadata** is key to removing silos and tracking, monitoring and managing data as it moves toward the end products.

**Understand-ability** from start to end!

**DataOps** is a paradigm shift. The way we treat data, metadata, and adapt to change itself.

# How to get started?

1. Review the 18 principles of the DataOps manifest @ dataopsmanifesto.org

2. Read the best-practice and reasoning provided by the dbt developers @ docs.getdbt.com

3. Check out "Cloud Data Management — 4 stages for informed companies", by Dave Fowler et.al.

4. Join the dbt Slack community @ getdbt.com/community