DATAFROST
CONSULTING

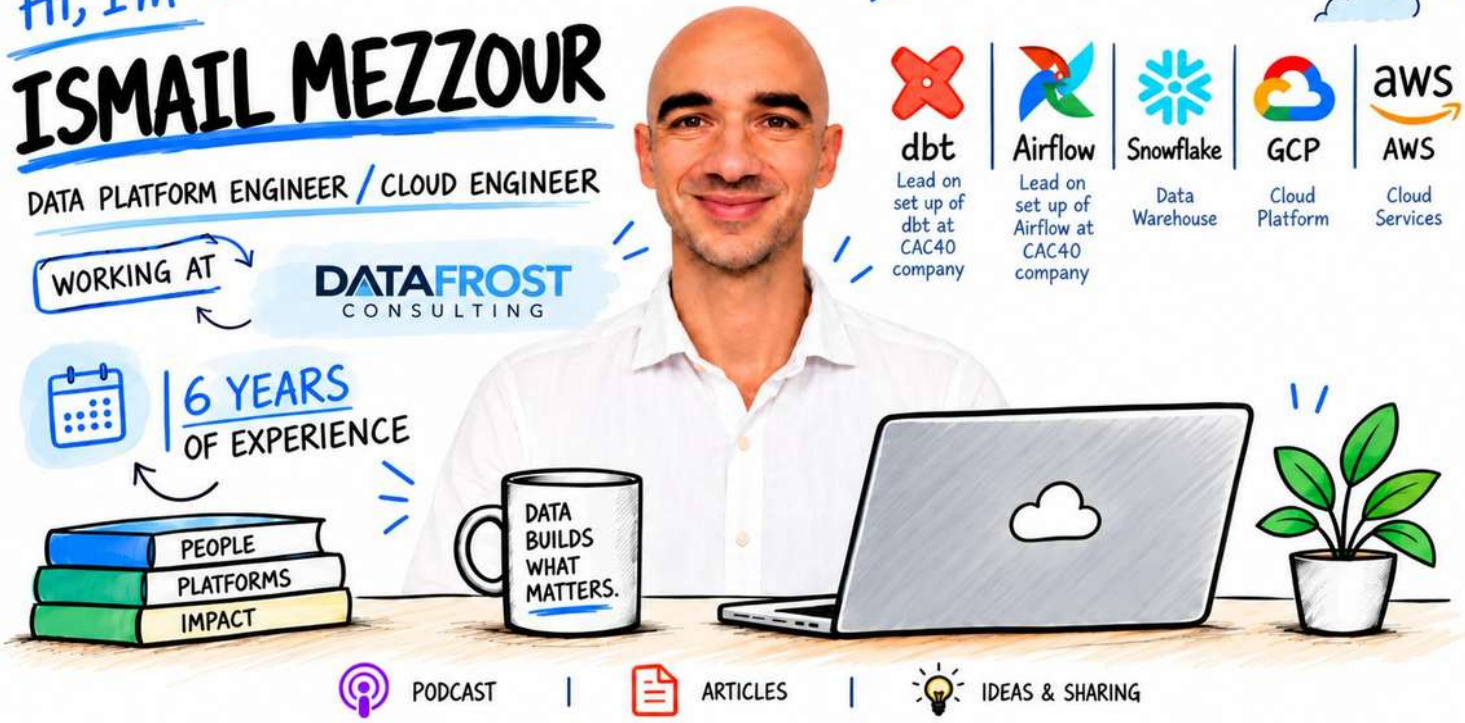
Hi, I'm

ISMAIL MEZZOUR

DATA PLATFORM ENGINEER / CLOUD ENGINEER

WORKING AT DATAFROST CONSULTING

6 YEARS OF EXPERIENCE



dbt — Lead on set up of dbt at CAC40 company

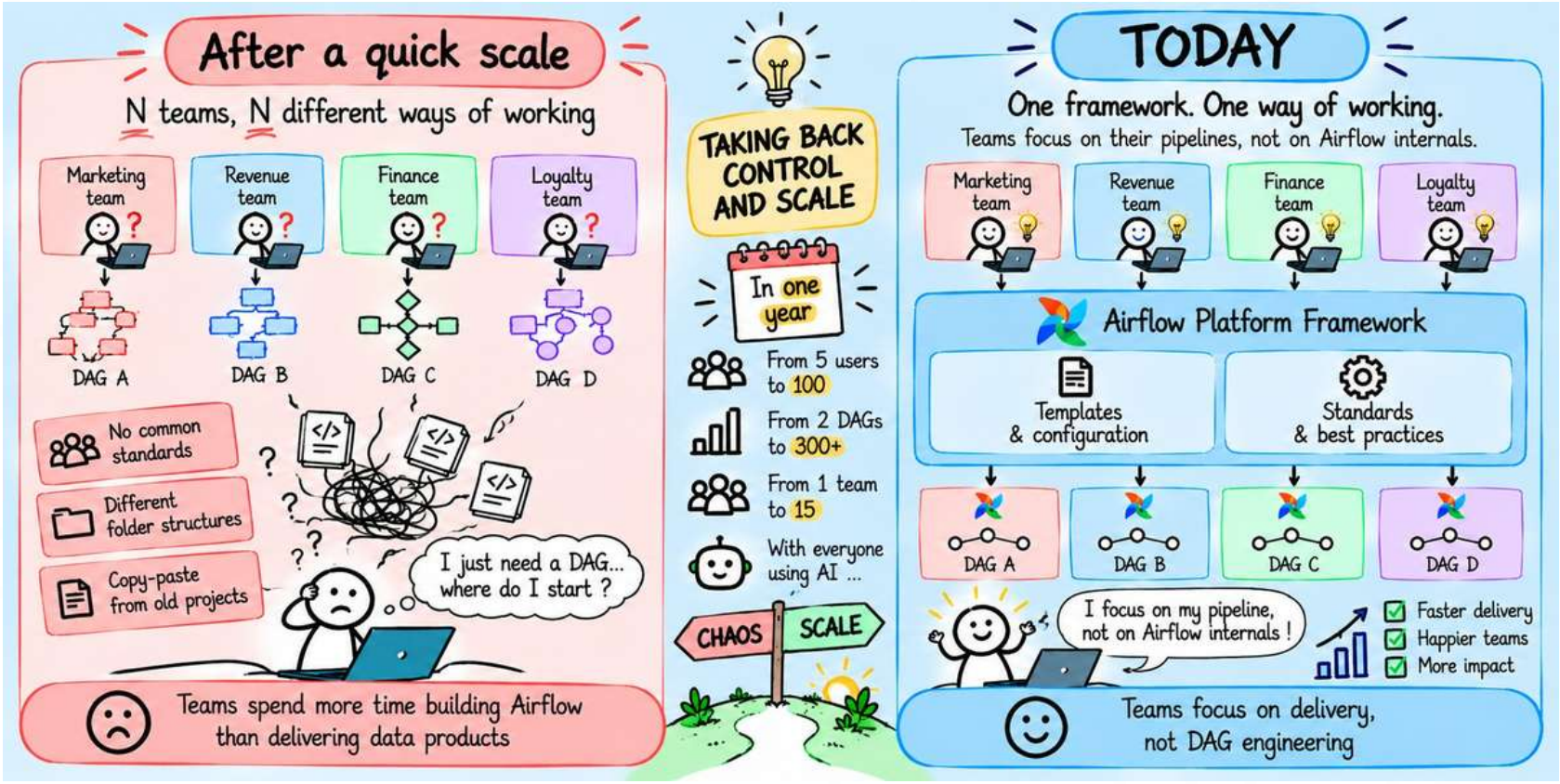
Airflow — Lead on set up of Airflow at CAC40 company

Snowflake — Data Warehouse

GCP — Cloud Platform

AWS — Cloud Services

PODCAST | ARTICLES | IDEAS & SHARING



DATAFROST CONSULTING

But firstly...

let's talk about the organization !



DATAFROST CONSULTING

Our Organization

A quick view before we dive into the details

Company Scale



We operate within a large-scale CAC 40 Group.

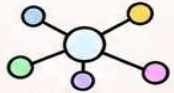
Team Size



100+

collaborators across data teams.

Organizational Model



We follow a Data Mesh approach, distributed across **~15 teams.**

Diverse Profiles



Engineers, analysts, scientists, product owners and more, working together.

Our Teams

Each team is specialized in its domain and owns its data.



Who makes it happen ?

- Data Engineers
- Analytics Engineers
- Data Analysts
- Data Scientists
- ML Engineers
- Data Product Owners

Our Organization

A quick view before we dive into the details

Company Scale

We operate within a large-scale CAC 40 Group.

Team Size

100+

collaborators across data teams.

Organizational Model

We follow a Data Mesh approach, distributed across **~15 teams.**

Diverse Profiles

Engineers, analysts, scientists, product owners and more, working together.

Multiple Use Cases

Trusted data solutions across a wide range of business use cases.

Our Teams

Each team is specialized in its domain and owns its data.



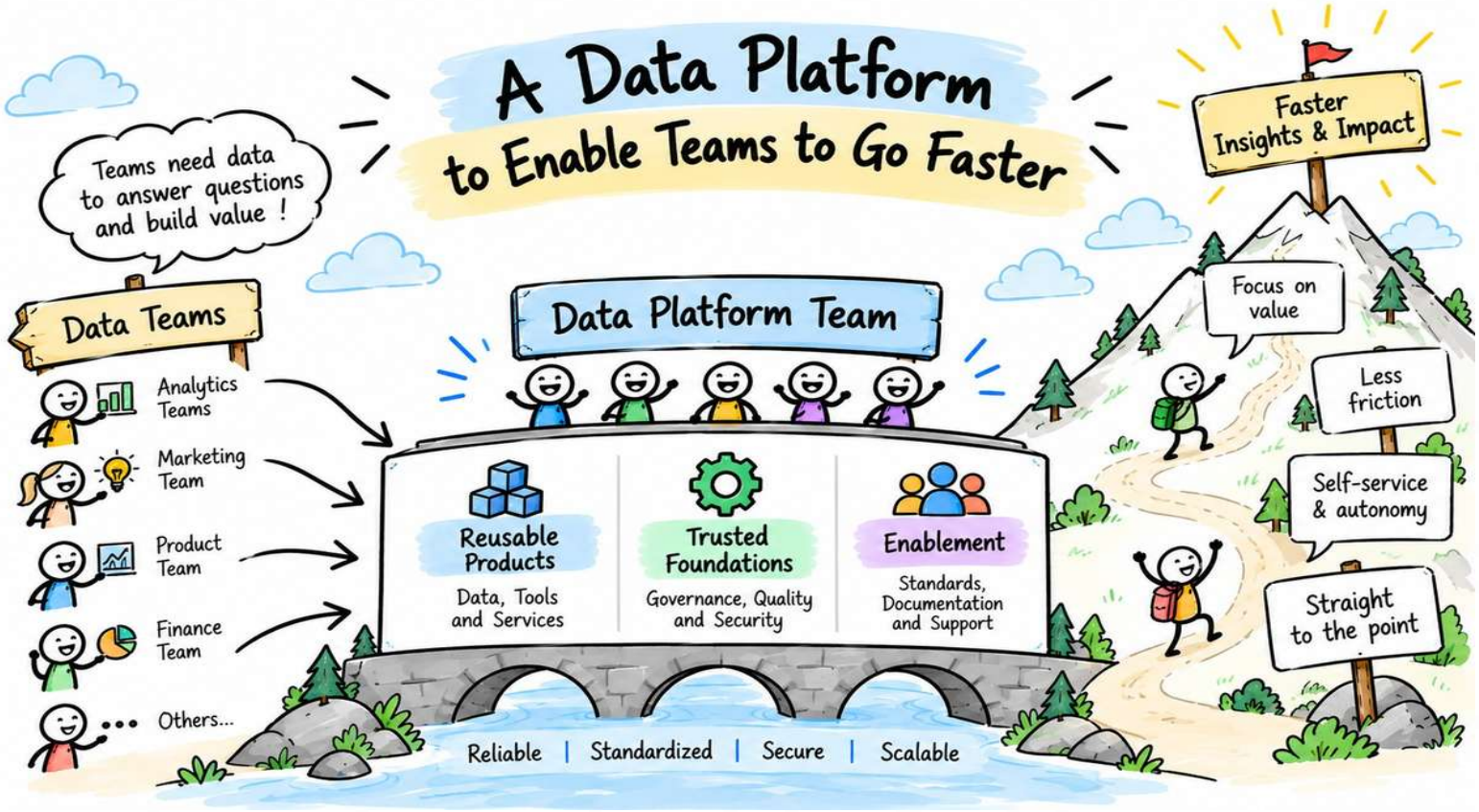
Who makes it happen ?

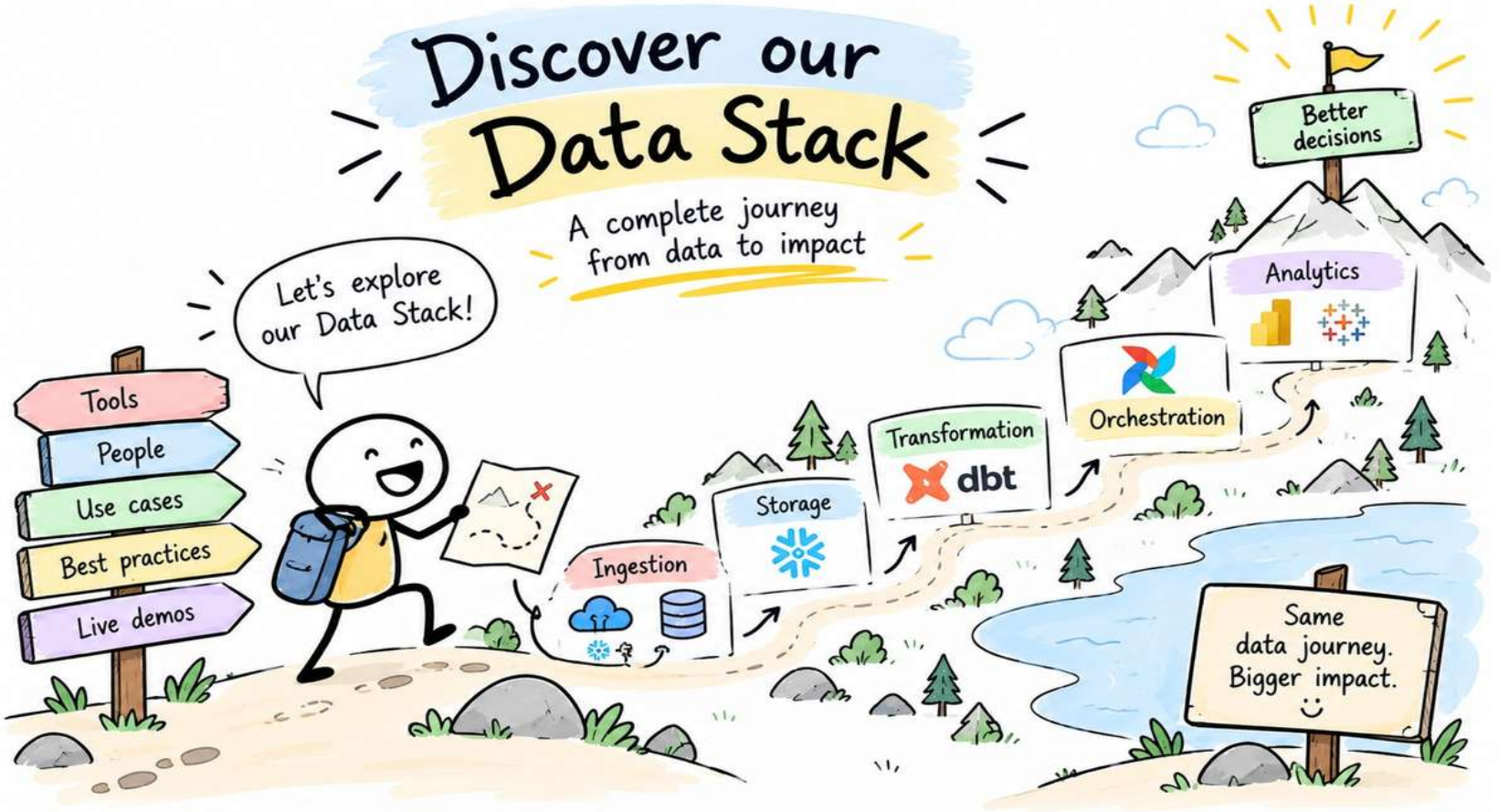
- Data Engineers
- Analytics Engineers
- Data Analysts
- Data Scientists
- ML Engineers
- Data Product Owners

What we deliver ?

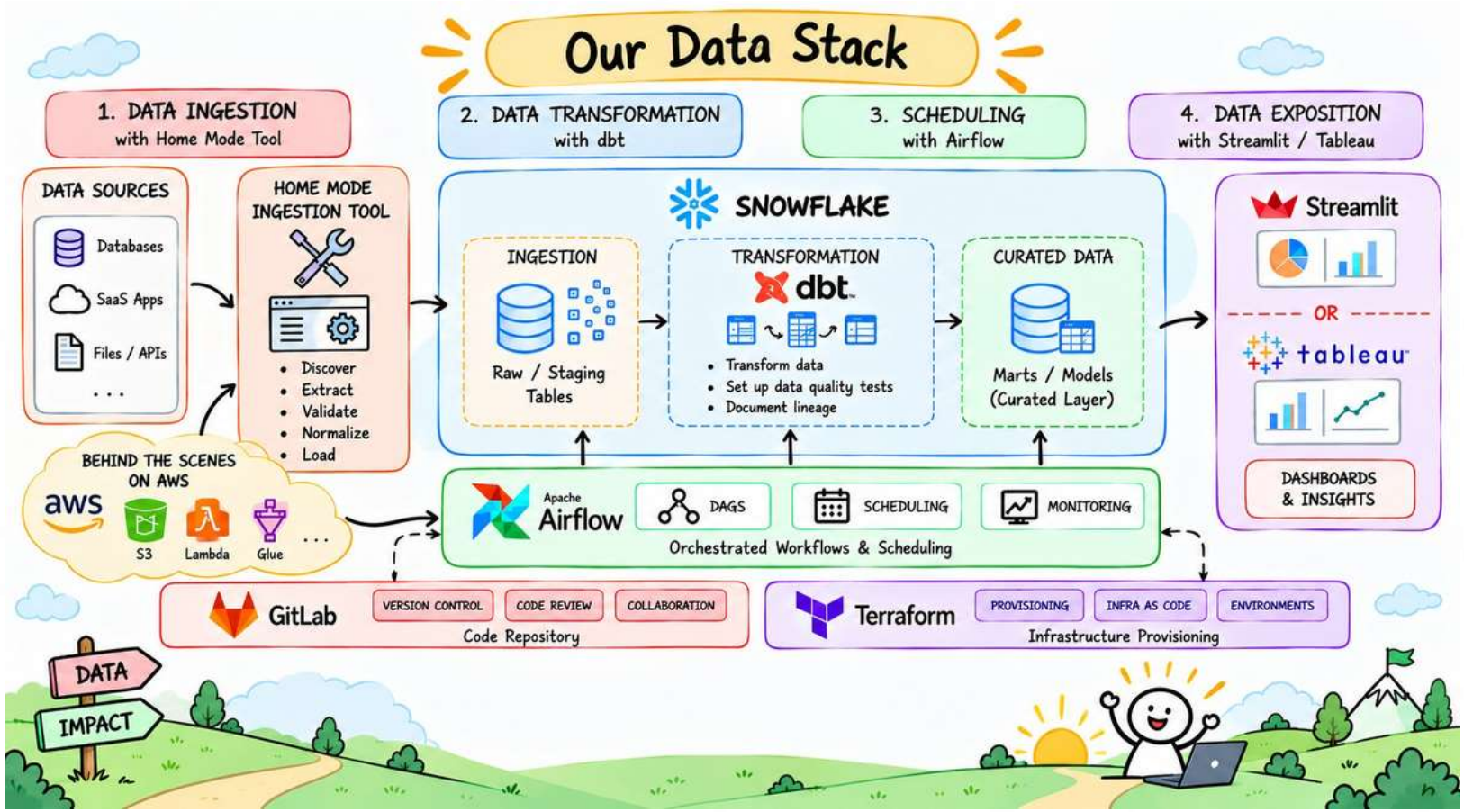
- Core data platform
- Data products and applications
- Reports and dashboards
- Analytics and ML solutions
- Business impact across the group

The Data Platform team provides the foundation to enable all teams to build, run and scale their data solutions.


A Data Platform
to Enable Teams to Go Faster
Teams need data to answer questions and build value !
Data Teams
Analytics Teams
Marketing Team
Product Team
Finance Team
Others...
Data Platform Team
Reusable Products
Data, Tools and Services
Trusted Foundations
Governance, Quality and Security
Enablement
Standards, Documentation and Support
Reliable | Standardized | Secure | Scalable
Faster Insights & Impact
Focus on value
Less friction
Self-service & autonomy
Straight to the point



DATAFROST
CONSULTING


Our Data Stack
1. DATA INGESTION
with Home Mode Tool
2. DATA TRANSFORMATION
with dbt
3. SCHEDULING
with Airflow
4. DATA EXPOSITION
with Streamlit / Tableau
DATA SOURCES
Databases
SaaS Apps
Files / APIs
. . .
HOME MODE INGESTION TOOL
Discover
Extract
Validate
Normalize
Load
SNOWFLAKE
INGESTION
Raw / Staging Tables
TRANSFORMATION
dbt
Transform data
Set up data quality tests
Document lineage
CURATED DATA
Marts / Models (Curated Layer)
Streamlit
OR
tableau
DASHBOARDS & INSIGHTS
BEHIND THE SCENES ON AWS
aws
S3
Lambda
Glue
. . .
Apache Airflow
DAGS
SCHEDULING
MONITORING
Orchestrated Workflows & Scheduling
GitLab
VERSION CONTROL
CODE REVIEW
COLLABORATION
Code Repository
Terraform
PROVISIONING
INFRA AS CODE
ENVIRONMENTS
Infrastructure Provisioning
DATA
IMPACT

But firstly... where we started ?



We decided to use Airflow...

But how do we use it ?

Thomas
Data Engineer - Marketing team



I need to ingest data from AWS to Snowflake. How do I schedule it ?

aws → Snowflake

Alice
Analytics engineer - Marketing team



I have dbt models in Snowflake. How do I run them and manage dependencies ?

SQL → dbt → Data models

Emma
Business analyst - Marketing team



I need to refresh my dashboard regularly. How do I orchestrate the data and notify in case of failure ?

Tableau → Dashboards & insights

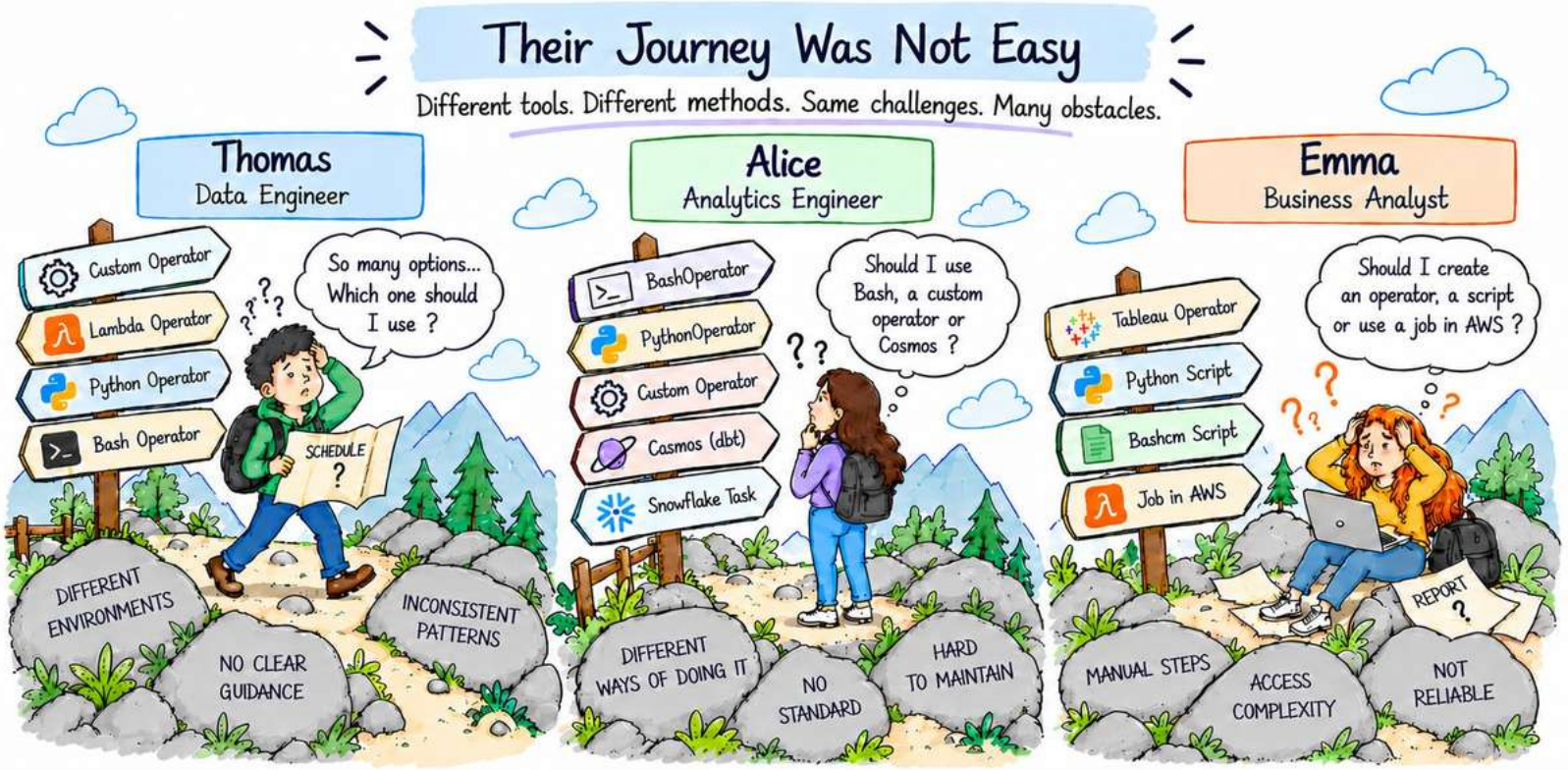
Airflow

- One platform
- One way to schedule
- Multiple use cases

How do we use it ? ?

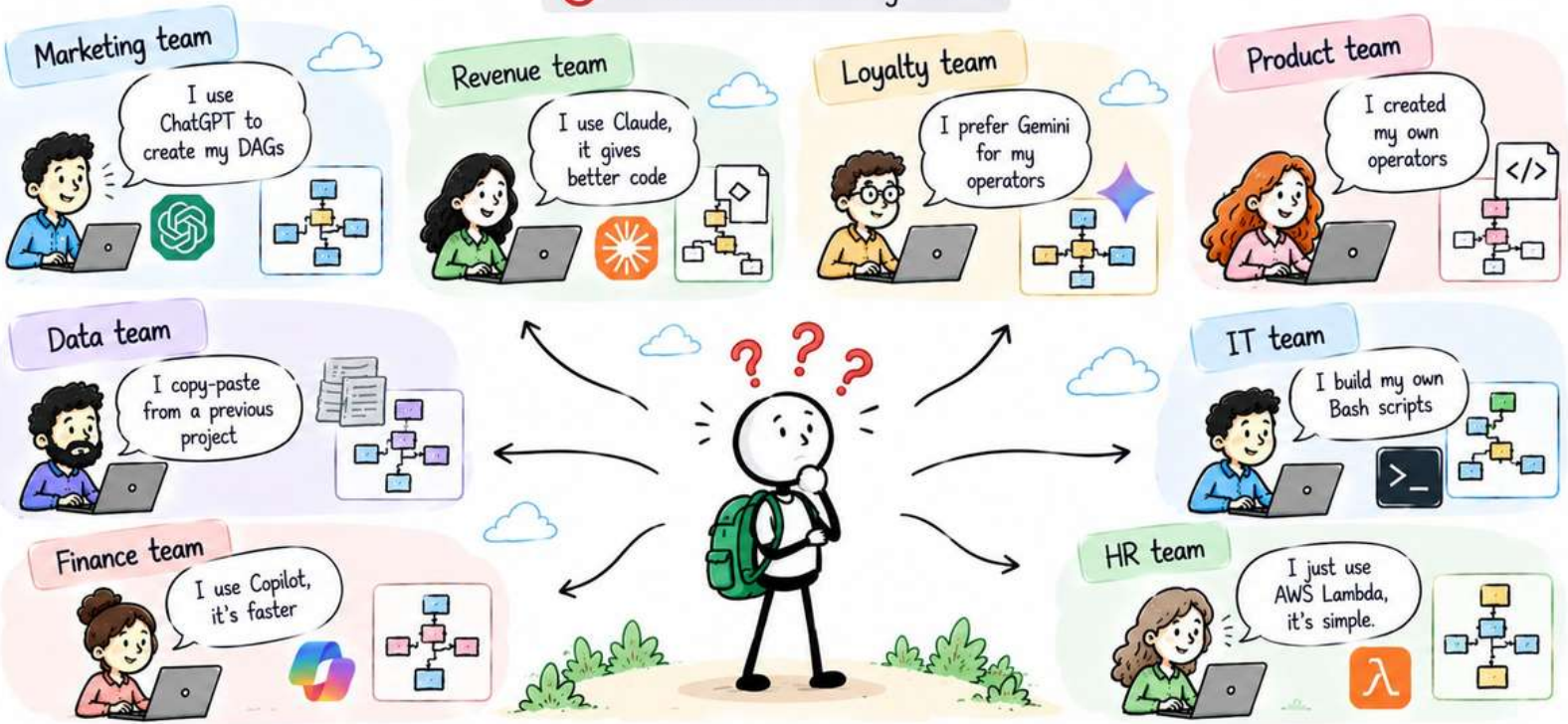
- Project structure and organisation
- DAG design and best practices
- Environment, CI/CD and deployment
- Monitoring and alerting


But how to schedule things in Airflow ?
There are many options and many ways to do it ?
SQL Operator
Docker Operator
Python Operator
Lambda Operator
BashOperator
Cosmos (dbt)
Sensors
HTTP Operator
Task Groups
Scheduling & Timetables

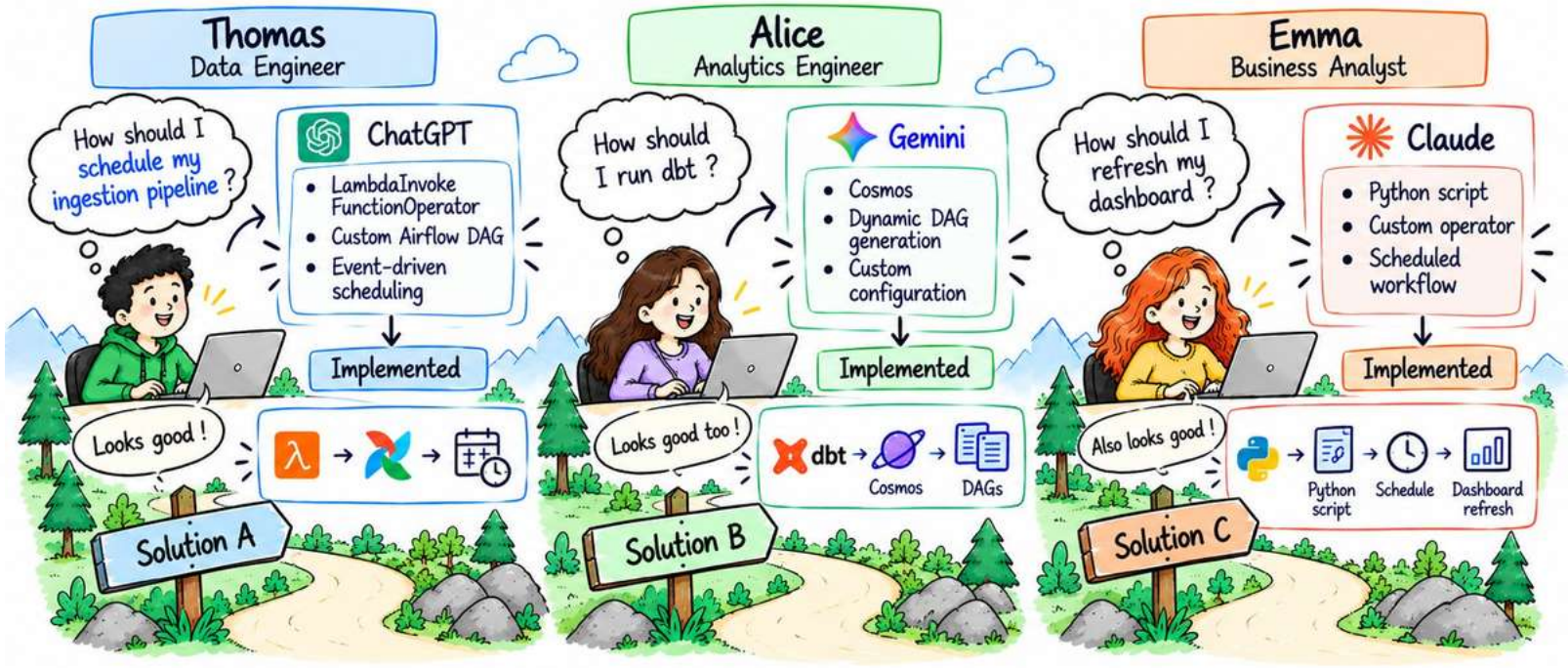

Their Journey Was Not Easy
Different tools. Different methods. Same challenges. Many obstacles.
Thomas
Data Engineer
Custom Operator
Lambda Operator
Python Operator
Bash Operator
So many options... Which one should I use ?
SCHEDULE
?
DIFFERENT ENVIRONMENTS
NO CLEAR GUIDANCE
INCONSISTENT PATTERNS
Alice
Analytics Engineer
BashOperator
PythonOperator
Custom Operator
Casmos (dbt)
Snowflake Task
Should I use Bash, a custom operator or Cosmos ?
DIFFERENT WAYS OF DOING IT
NO STANDARD
HARD TO MAINTAIN
Emma
Business Analyst
Tableau Operator
Python Script
Bashcm Script
Job in AWS
Should I create an operator, a script or use a job in AWS ?
REPORT
?
MANUAL STEPS
ACCESS COMPLEXITY
NOT RELIABLE

But wait... Who to follow ?

⊗ No clear standards or guidelines

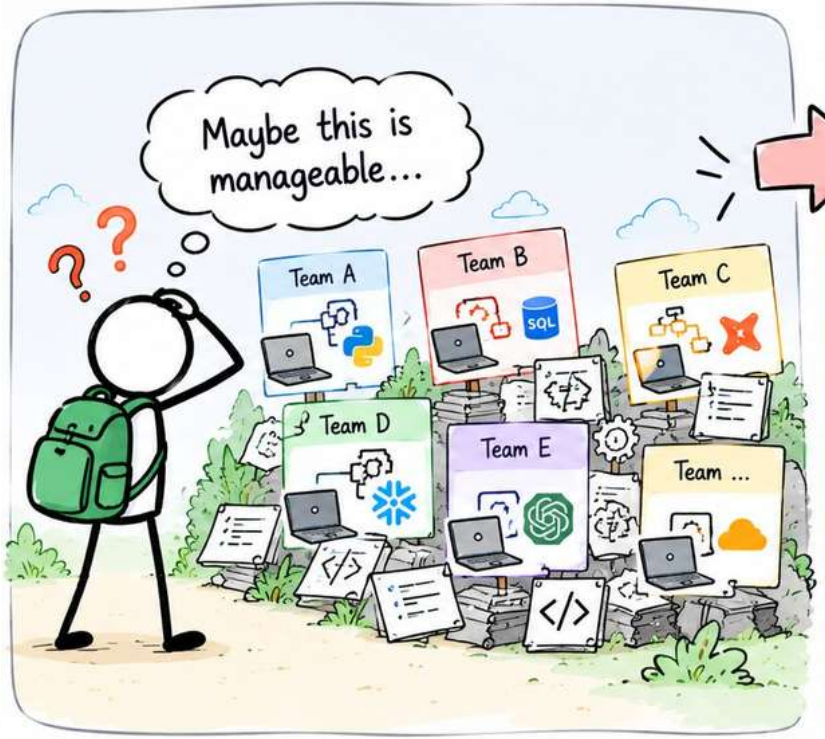


No Standard,
So Everyone Uses Their Own AI Tool



BUT WAIT...

How many teams are we actually supporting ?



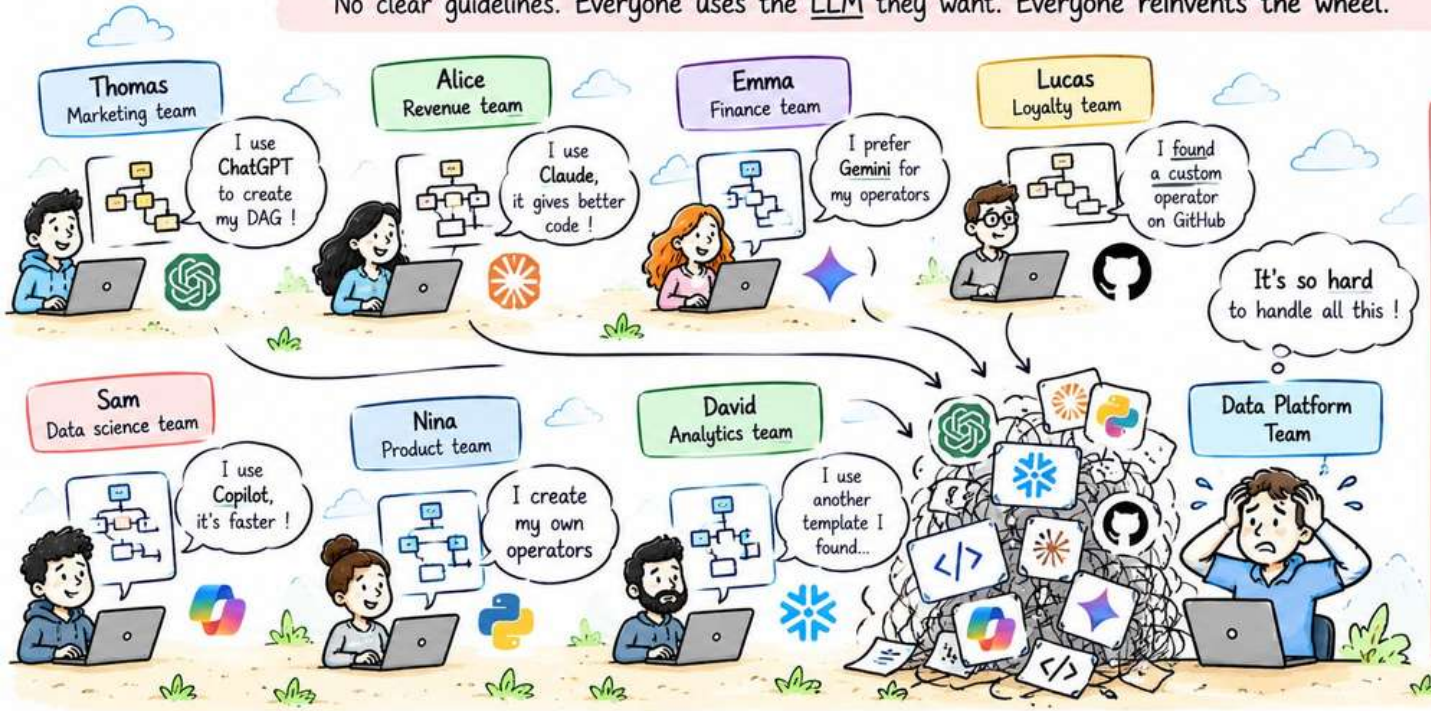
15 TEAMS

Team 1 Marketing	Team 2 Revenue	Team 3 Loyalty	Team 4 Digital	Team 5 Product
Team 6 Finance	Team 7 HR	Team 8 Data	Team 9 IT	Team 10 Legal
Team 11 Operations	Team 12 Customer	Team 13 Sales	Team 14 Partnership	Team 15 Analytics

15 different domains
15 different priorities
15 different ways of working

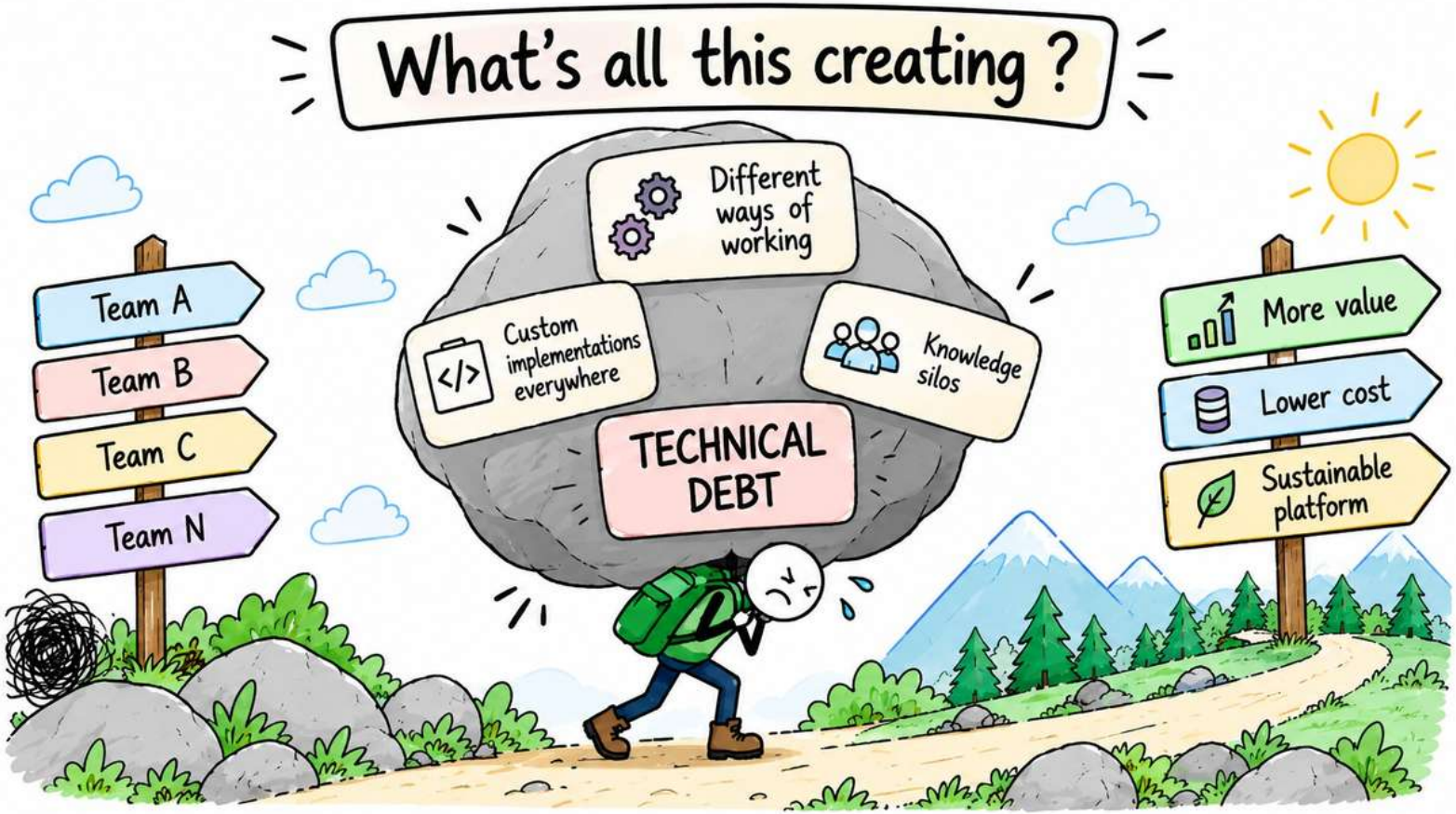
Everyone did things their own way

No clear guidelines. Everyone uses the LLM they want. Everyone reinvents the wheel.



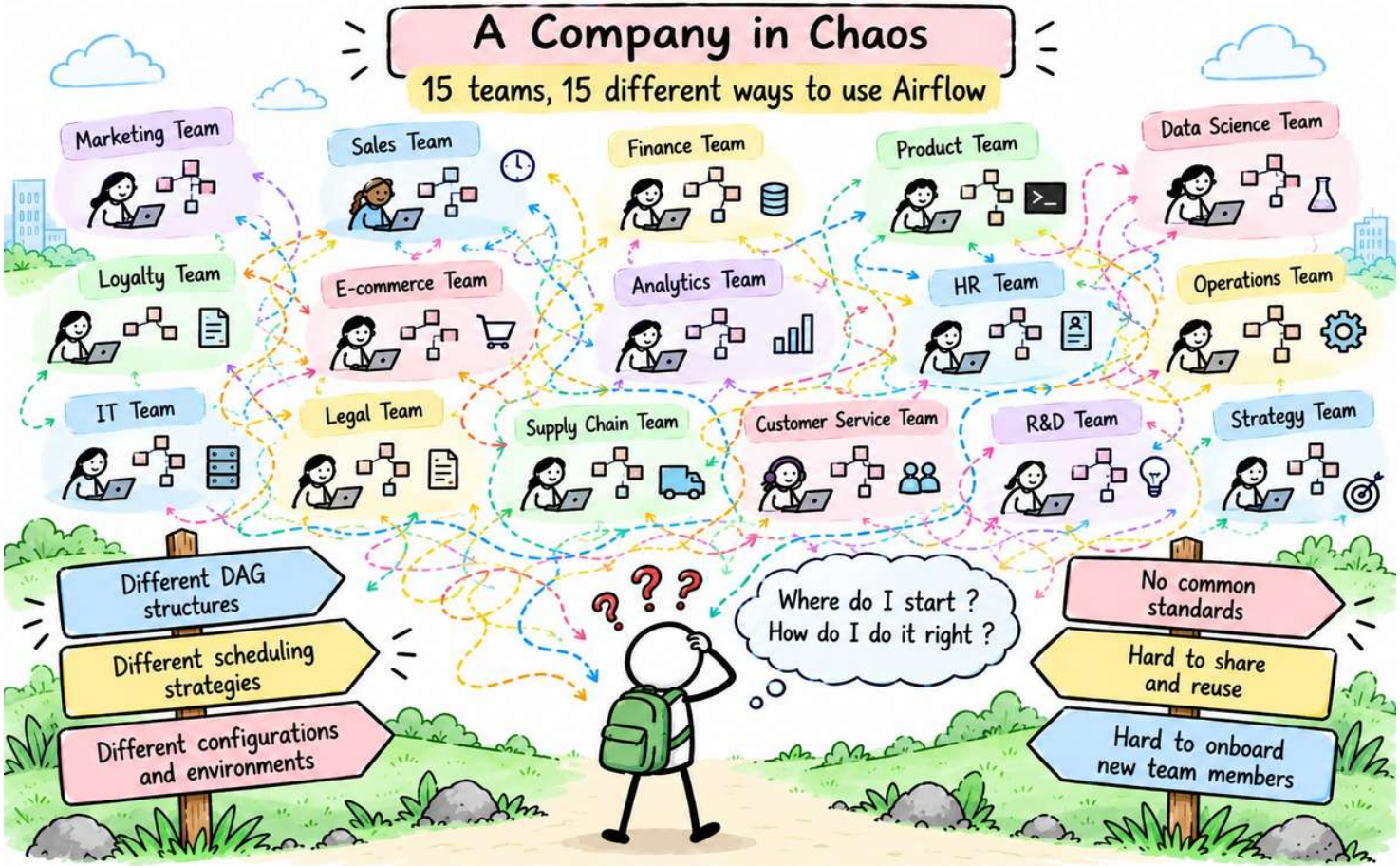
HARD TO MANAGE

- N different implementations
- No clear guidelines and standards
- Different LLMs, different approaches
- Duplicated code and custom operators
- Inconsistent quality and monitoring
- High support load for the platform team


What's all this creating ?
Team A
Team B
Team C
Team N
Different ways of working
Custom implementations everywhere
Knowledge silos
TECHNICAL DEBT
More value
Lower cost
Sustainable platform

The Consequences of No Guidance

Team	Person	What happens without guidance ?
Data Engineering		 Too many options, inconsistent patterns → Longer delivery times
Business Intelligence		 Refreshes dashboards manually (not automated) →  →   Loss of trust in the data
Data Platform		 Growing support requests and context switching →  Increasing technical debt


A Company in Chaos
15 teams, 15 different ways to use Airflow
Marketing Team
Sales Team
Finance Team
Product Team
Data Science Team
Loyalty Team
E-commerce Team
Analytics Team
HR Team
Operations Team
IT Team
Legal Team
Supply Chain Team
Customer Service Team
R&D Team
Strategy Team
Different DAG structures
Different scheduling strategies
Different configurations and environments
Where do I start ?
How do I do it right ?
No common standards
Hard to share and reuse
Hard to onboard new team members

What We Tried
What Didn't Work



We weren't missing tools.
We were missing a scalable way of working.

Three Attempts, One Platform Problem

1

We tried documentation



VS.

But it didn't work



But the documentation **tests nothing.**

2

We tried templates



VS.

But it didn't work



Everyone creates their own version **with LLM.** No consistency.

3

We tried platform support



VS.

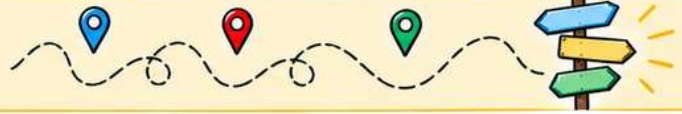
But it didn't work



The platform team becomes the **bottleneck.**

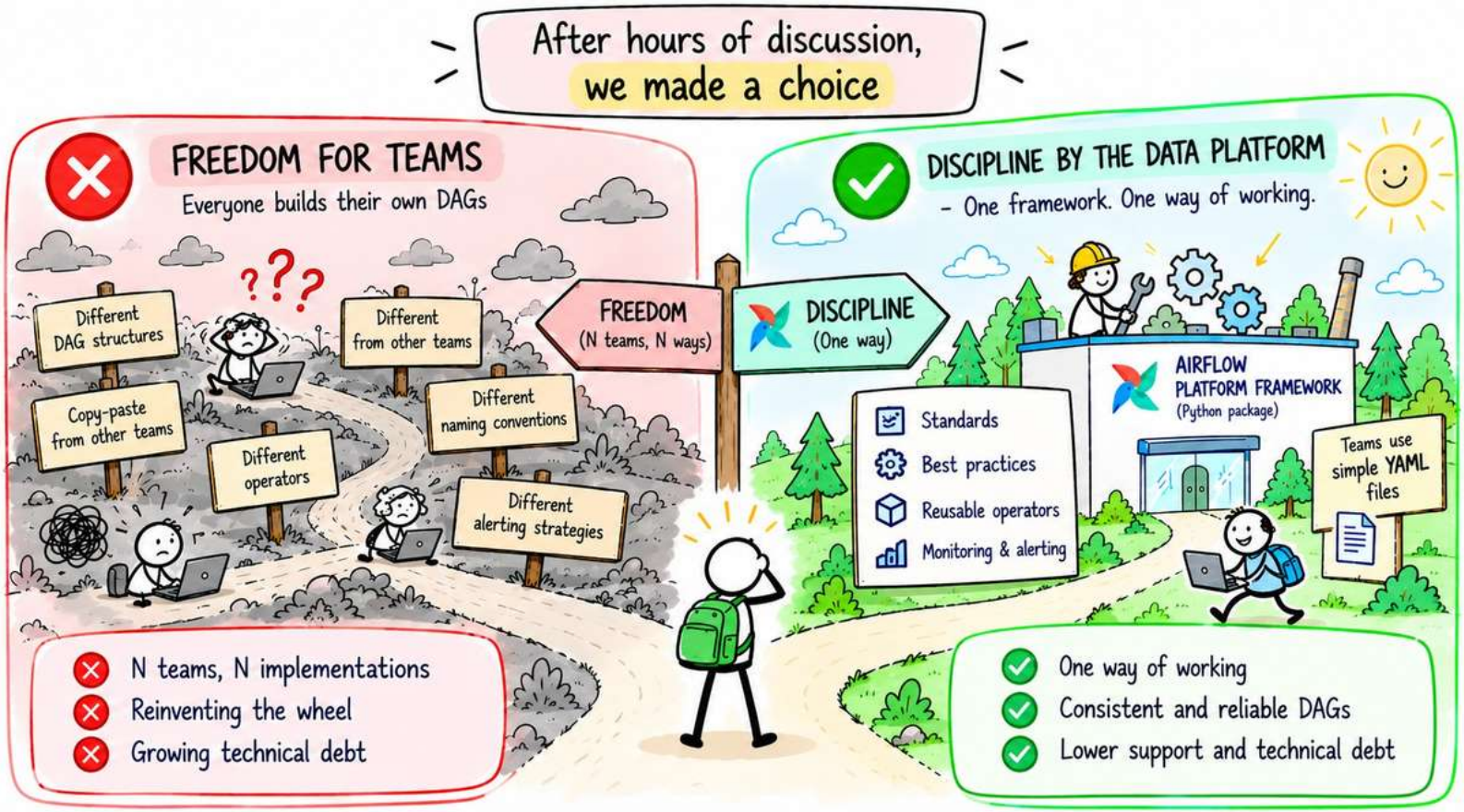


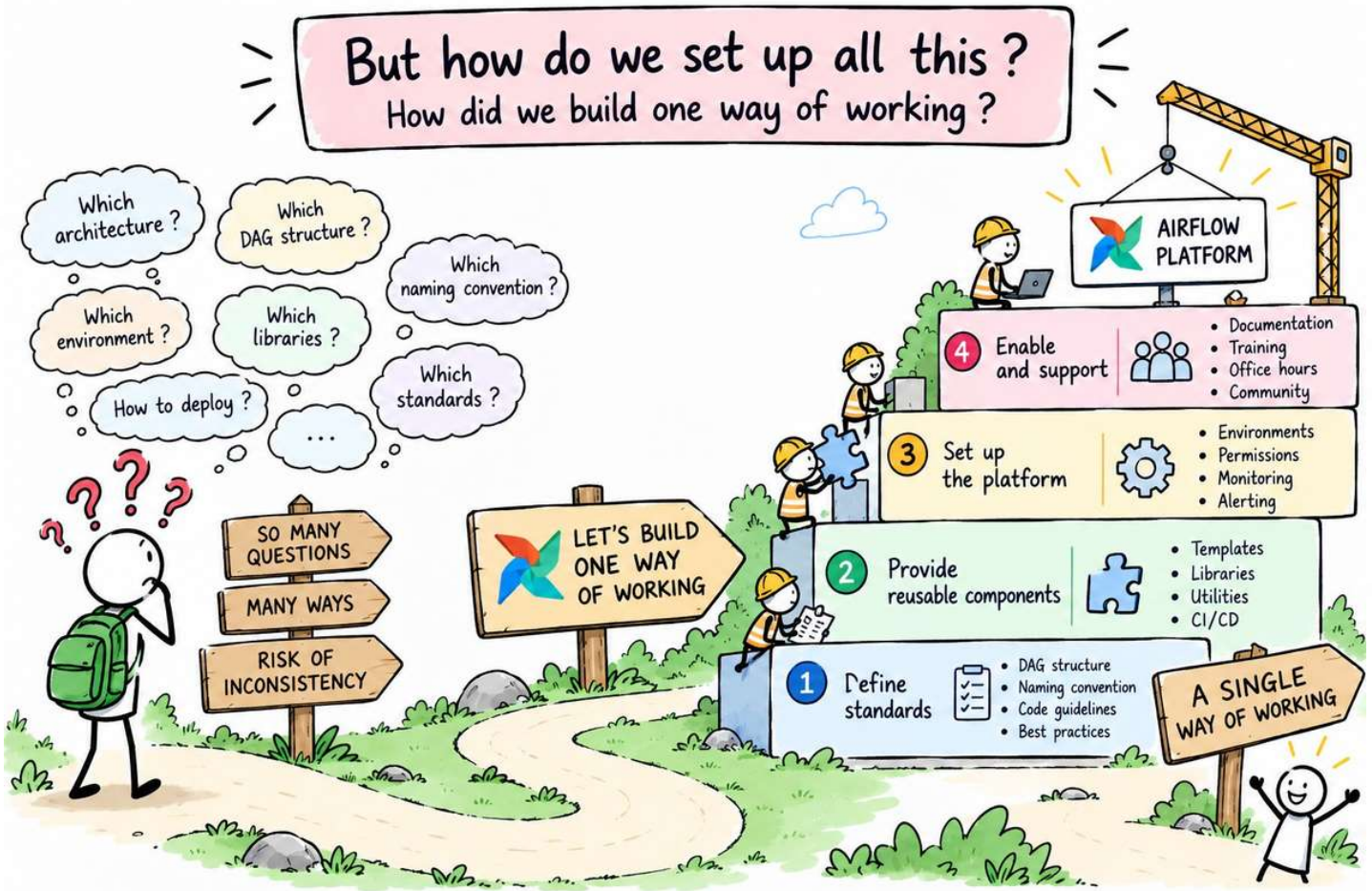
The problem wasn't a lack of help.
It was the number of possible paths.



DATAFROST CONSULTING


After trying all this, we have two ways now :
Option 1
Option 2


After hours of discussion,
we made a choice
FREEDOM FOR TEAMS
Everyone builds their own DAGs
Different DAG structures
Different from other teams
Copy-paste from other teams
Different naming conventions
Different operators
Different alerting strategies
FREEDOM
(N teams, N ways)
DISCIPLINE
(One way)
DISCIPLINE BY THE DATA PLATFORM
– One framework. One way of working.
AIRFLOW
PLATFORM FRAMEWORK
(Python package)
Standards
Best practices
Reusable operators
Monitoring & alerting
Teams use simple YAML files
N teams, N implementations
Reinventing the wheel
Growing technical debt
One way of working
Consistent and reliable DAGs
Lower support and technical debt


But how do we set up all this ?
How did we build one way of working ?
Which architecture ?
Which DAG structure ?
Which naming convention ?
Which environment ?
Which libraries ?
How to deploy ?
...
Which standards ?
SO MANY QUESTIONS
MANY WAYS
RISK OF INCONSISTENCY
LET'S BUILD ONE WAY OF WORKING
AIRFLOW PLATFORM
4 Enable and support
• Documentation
• Training
• Office hours
• Community
3 Set up the platform
• Environments
• Permissions
• Monitoring
• Alerting
2 Provide reusable components
• Templates
• Libraries
• Utilities
• CI/CD
1 Define standards
• DAG structure
• Naming convention
• Code guidelines
• Best practices
A SINGLE WAY OF WORKING

Reusable template through a Python package

Data Platform Team

Build and maintain the reusable template



Datafrost package

`pip install datafrost`

- Generate DAGs from YML
- Built-in best practices
- Standards and defaults
- Continuous improvements

Domain Teams

Use the template, not reinvent the wheel

Thomas	Alice	Emma
Data Engineer	Analytics Engineer	Business Analyst



Pre-built Airflow Templates (in YML)

Only requirement: datafrost v1.4.0)

DATAFROST CONSULTING

Domain Teams

Use the template, not reinvent the wheel

Thomas
Data Engineer



Alice
Analytics Engineer



Emma
Business Analyst



ingestion.yml

```
type: ingestion
source: aws
target: snowflake
...
```

model.yml

```
type: dbt
models: my_models
dependencies: true
...
```

dashboard.yml

```
type: dashboard
tool: tableau
refresh: daily
...
```

Pre-built Airflow Templates (in YAML)

Only requirement: datafrost v1.4.0)

Data Platform Team

Build and maintain templates and other platform projects

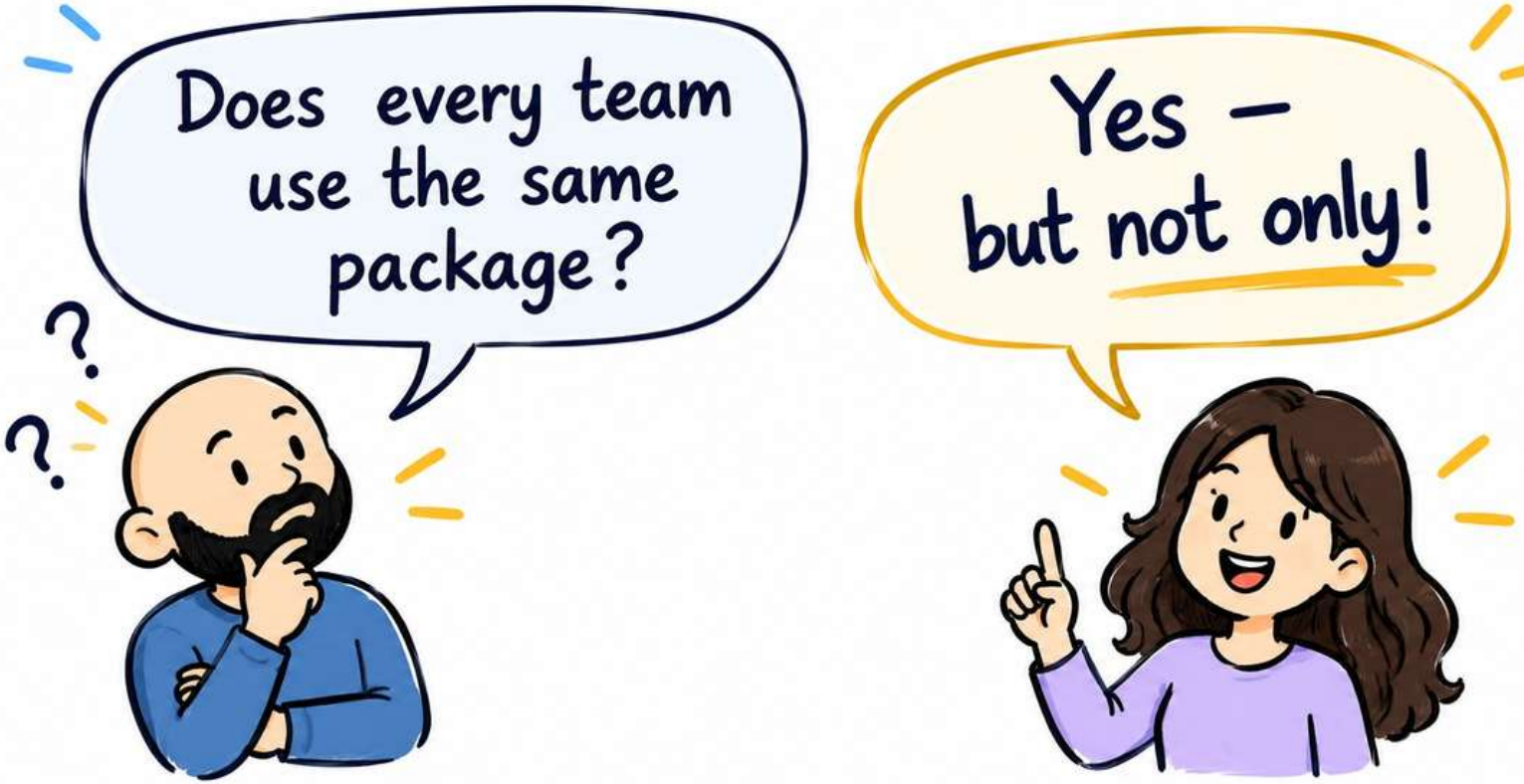


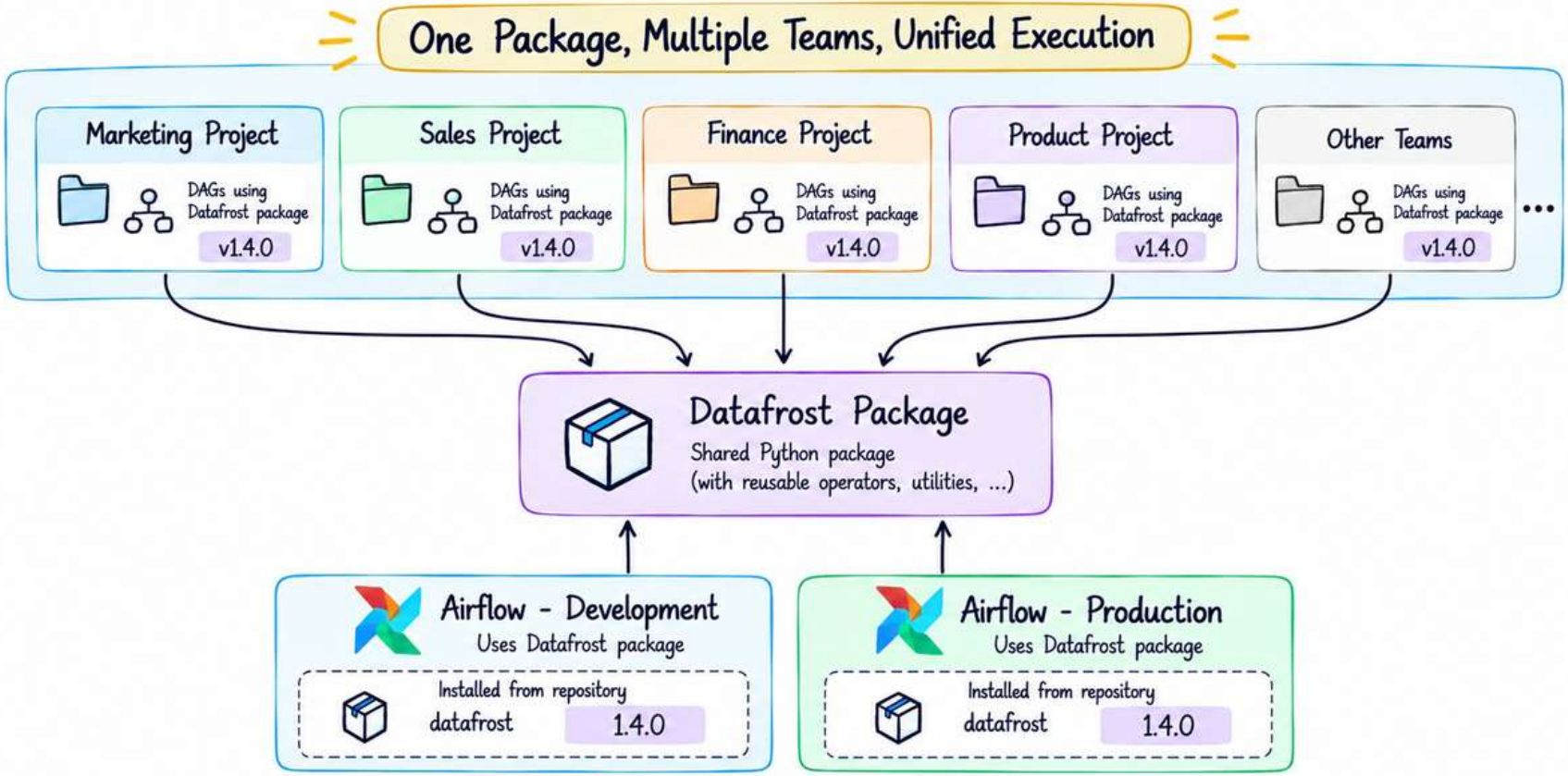
Airflow Templates (YAML)

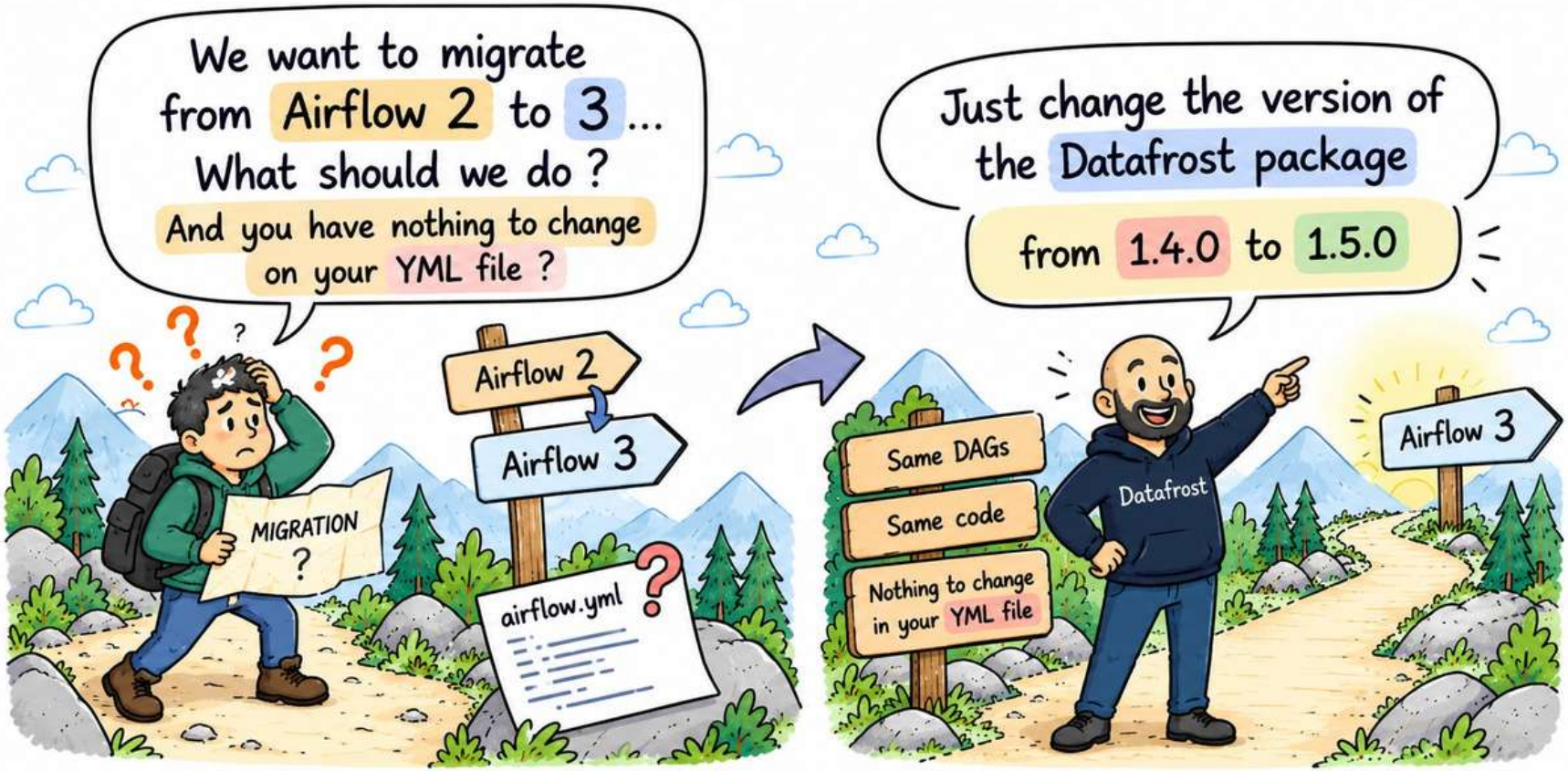
- Build and maintain templates
- Add new features
- Ensure best practices

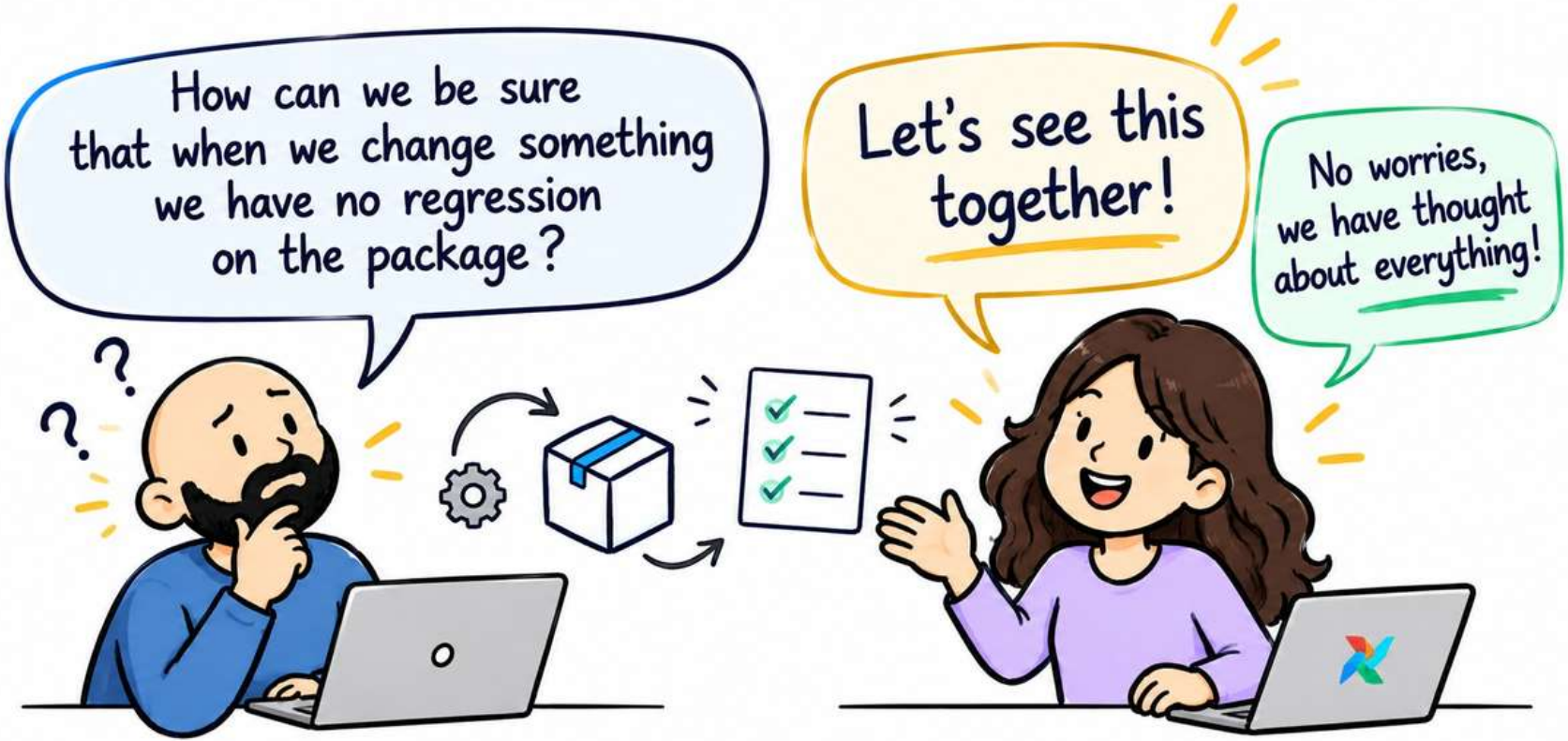
- Data Quality project
- Monitoring & Observability
- Self-service tools
- Other platform projects

DATAFROST CONSULTING


Does every team use the same package?
Yes – but not only!


One Package, Multiple Teams, Unified Execution
Marketing Project
DAGs using Datafrost package
v1.4.0
Sales Project
DAGs using Datafrost package
v1.4.0
Finance Project
DAGs using Datafrost package
v1.4.0
Product Project
DAGs using Datafrost package
v1.4.0
Other Teams
DAGs using Datafrost package
v1.4.0
Datafrost Package
Shared Python package
(with reusable operators, utilities, ...)
Airflow - Development
Uses Datafrost package
Installed from repository
datafrost
1.4.0
Airflow - Production
Uses Datafrost package
Installed from repository
datafrost
1.4.0


We want to migrate from Airflow 2 to 3 ...
What should we do ?
And you have nothing to change on your YML file ?
MIGRATION ?
Airflow 2
Airflow 3
airflow.yml
Just change the version of the Datafrost package
from 1.4.0 to 1.5.0
Same DAGs
Same code
Nothing to change in your YML file
Datafrost
Airflow 3


How can we be sure that when we change something we have no regression on the package?
Let's see this together!
No worries, we have thought about everything!

How we ensure no regression when we change the package ?

1 Datafrost Package Project

Code changes
New features or fixes

Unit tests
Test each function and component

Integration tests
Test real connections with AWS, Snowflake and Tableau

CI pipeline
Run all tests automatically on every change

2 Ephemeral test environment

Automatically created for each run to test DAGs and connections

Airflow
Run DAGs

- Snowflake ✓
- AWS ✓
- Tableau ✓

3 Dev environment

Last tests in dev with all the projects

Airflow
Run DAGs
with real projects

- Snowflake ✓
- AWS ✓
- Tableau ✓

4 Release

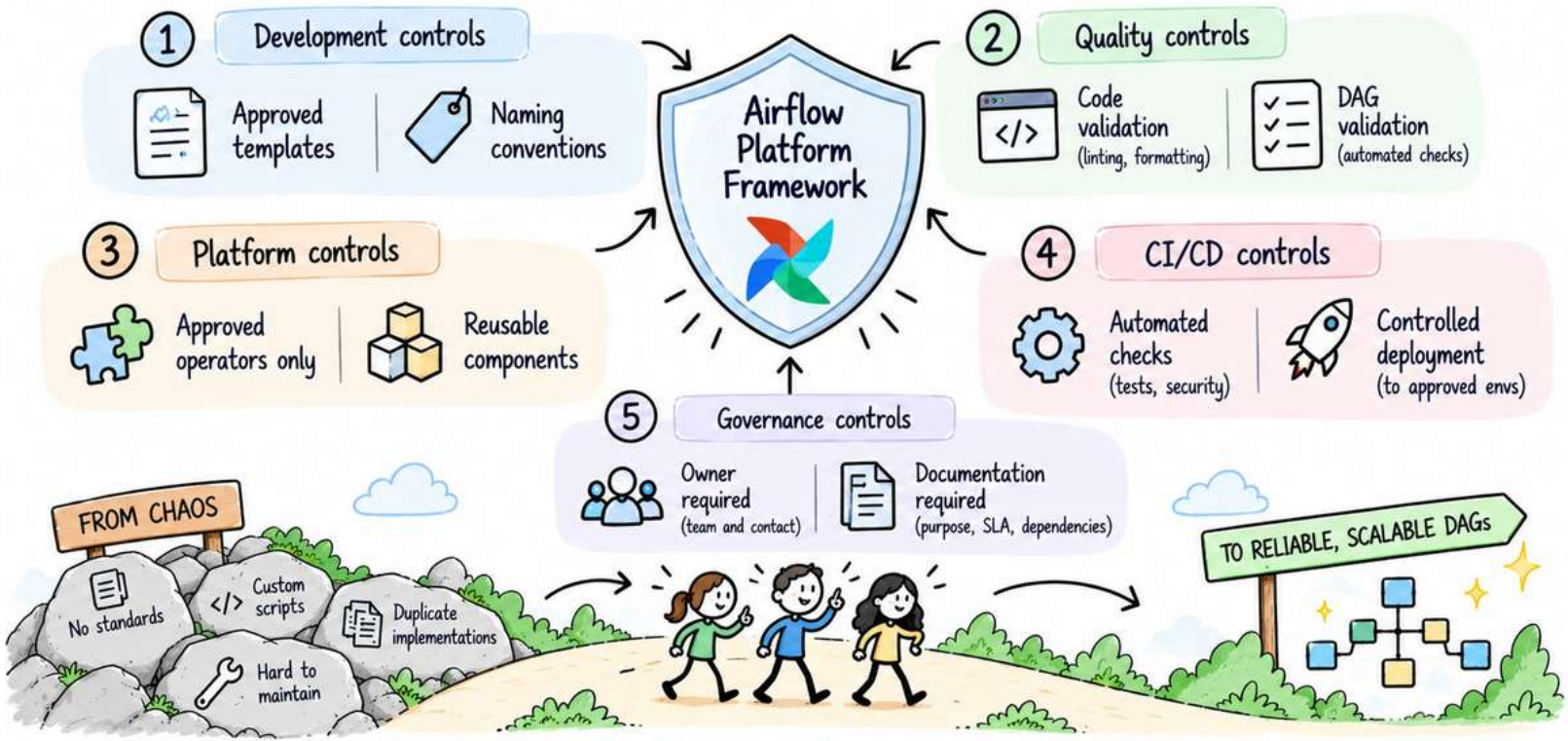
Publish new version of the package

datafrost
v1.4.0



DATAFROST
CONSULTING

What controls did we put in place ?



We also decide to evaluate the standard



How We Keep Control at Scale

Monitoring hundreds of workflows from one place.

Data Platform Team

Scan, monitor and ensure standards



DATAFROST CONSULTING | Platform Governance

- Overview
- Projects
- DAGs
- Standards
- Issues
- Settings

15 Teams · 312 DAGs · 96% Compliant · 8 Issues

Team	Project	DAGs	Compliance	Issues
Marketing	marketing-data	42	100%	-
Revenue	revenue-analytics	38	95%	2
Loyalty	loyalty-data	27	100%	-
Finance	finance-data	35	91%	6
Product	product-analytics	28	100%	-
...	...	...	...	...

Standard checks (via DataFrost)

- Uses DataFrost package
- Valid YAML configuration
- Required tags and owner
- Approved operators
- DAG structure and naming
- Monitoring and alerts configured

Scan all team repositories (daily)

DATAFROST CONSULTING scan engine

- Read project configuration
- Check YAML files
- Verify DataFrost usage
- Validate standards
- Collect metadata (DAGs, tags, ...)

Visibility and insights for the platform team

Domain Teams

Each team has their own project using DataFrost

Marketing Team



marketing-data
- requirements.txt
- marketing_ingestion.yml
- marketing_dbt.yml
- ...

Uses DataFrost

Revenue Team



revenue-analytics
- requirements.txt
- revenue_ingestion.yml
- revenue_dbt.yml
- ...

Uses DataFrost

Loyalty Team



loyalty-data
- requirements.txt
- loyalty_ingestion.yml
- loyalty_dbt.yml
- ...

Uses DataFrost

Finance Team



finance-data
- requirements.txt
- finance_ingestion.yml
- finance_dbt.yml
- ...

Uses DataFrost

...

What this approach brings ?



Now We Are Aligned

One package. One way of working for everybody.

Thomas
Data Engineer (Marketing)



ingestion.yml
- type: ingestion
- source: salesforce
- schedule: daily
- ...

Alice
Analytics Engineer (Revenue)



dbt.yml
- type: dbt
- project: revenue
- schedule: hourly
- ...

Emma
Business Analyst (Finance)



dashboard.yml
- type: dashboard_refresh
- dashboard: finance
- schedule: daily
- ...



Everyone Aligned



- Same tools
- Same standards
- Same best practices
- One way of working for everyone

More Control



- Visibility across all projects
- Verify standards
- Detect issues early
- Governance at scale

Easy to Migrate



- Teams just update the package version
- No code to rewrite
- Seamless migration
- Less maintenance

Easy to Evolve



- New features for all teams
- Bug fixes automatically applied
- Continuous improvements
- Everyone benefits

DATAFROST
CONSULTING



DATAFROST
CONSULTING

Mezzour Ismail
Founder

Articles and insights | Podcast episodes | Ideas and use cases | Updates and more

Let's stay in touch!

Scan the QR code to **get in touch** and **discover new updates!**