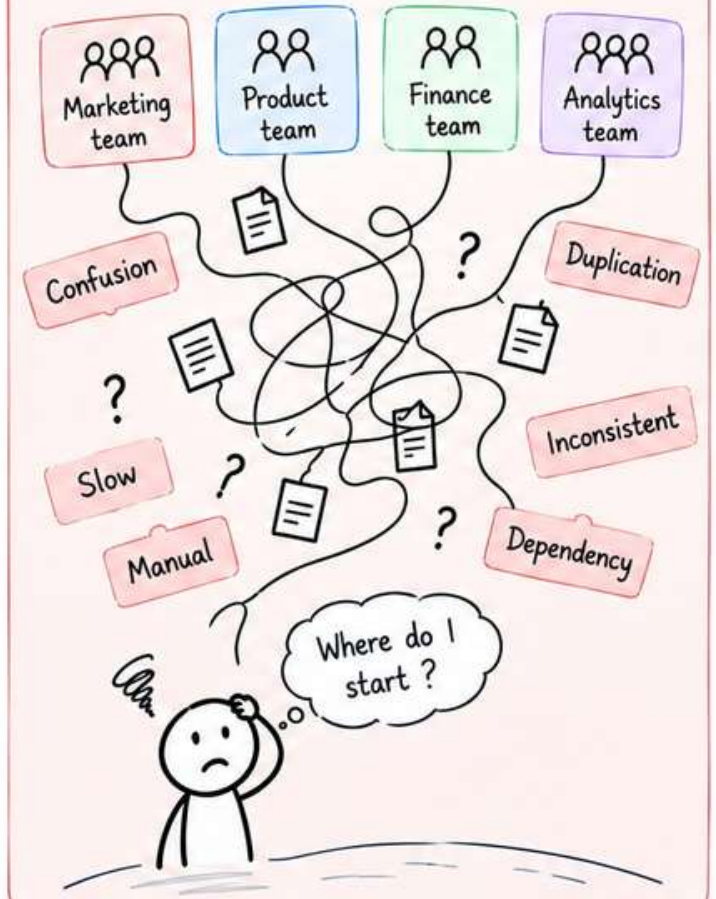

Hi, I'm
ISMAIL MEZZOUR
DATA PLATFORM ENGINEER / CLOUD ENGINEER
dbt
Lead on set up of dbt at CAC40 company
Airflow
Lead on set up of Airflow at CAC40 company
Snowflake
Data Warehouse
GCP
Cloud Platform
aws
AWS
Cloud Services
6 YEARS
OF EXPERIENCE
PEOPLE
PLATFORMS
IMPACT
DATA BUILDS WHAT MATTERS.
PODCAST
ARTICLES
IDEAS & SHARING

My Feedback on Self-Service Analytics

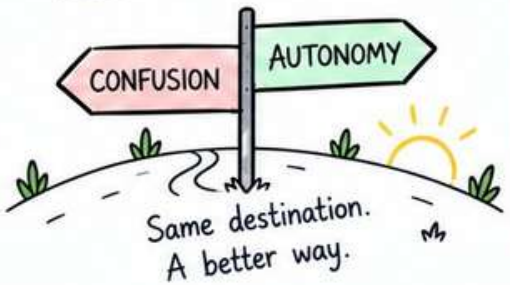
BEFORE

Different stacks, Different ways.
Lots of friction.



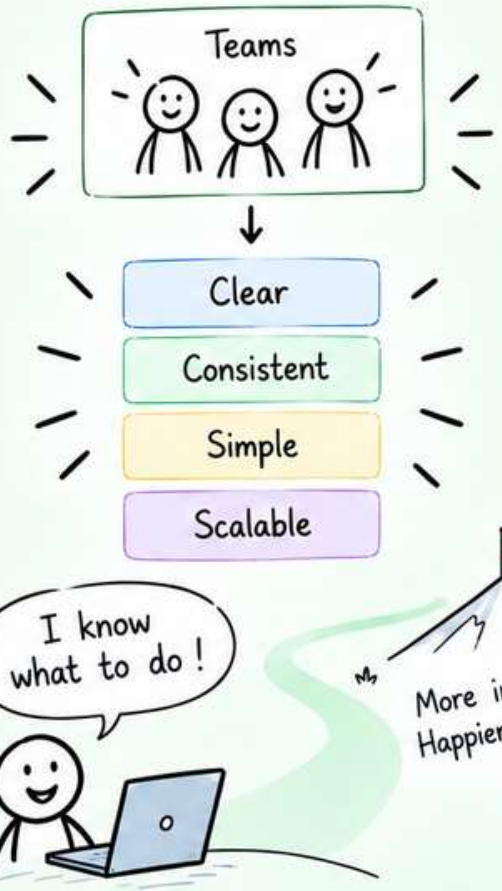


From guidance ...
... to standards



TODAY

One path, One way of working.
Teams are autonomous.



Freedom of choice everywhere = confusion

Freedom within standards = real autonomy

MEZZOUR ISMAIL

Who are the people we are going to follow ?

Three perspectives, one shared journey



Three Personas, One Platform Challenge

Different roles. Same problems.

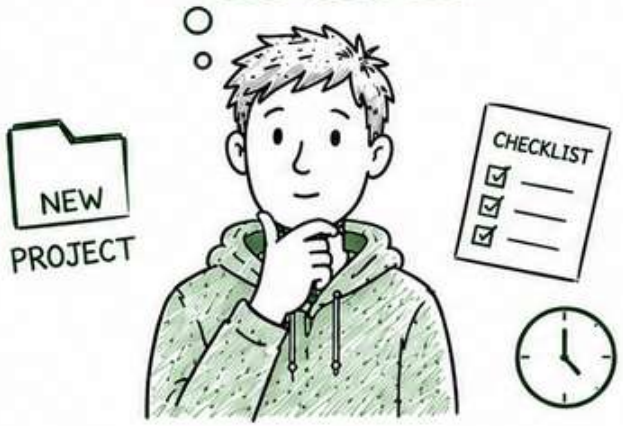
It's hard to find clear and consistent standards.



ALICE

Data analyst
in the revenue team

It's hard to build the right way and deliver on time.



THOMAS

Data engineer
in the marketing team

It's hard to keep up with the constant flow of questions.



ISMAIL

Data platform engineer

Different roles, same struggle: unclear guidance, inconsistency, and too much manual effort.

MEZZOUR
ISMAIL

Our Organization at a Glance

Here's a snapshot of our data organization.

1 Company Scale



We operate within a large-scale CAC 40 Group.

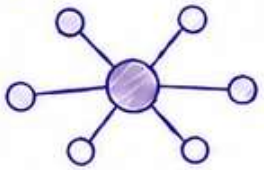
2 Team Size



100+

collaborators in our data teams.

3 Organizational Model



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

4 Diverse Profiles



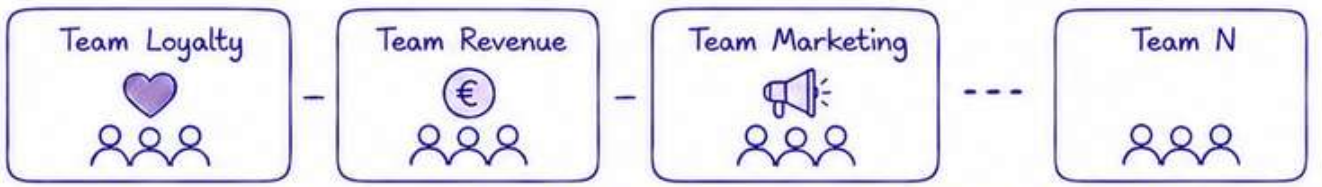
Various specialized roles working together.

5 Multiple Use Cases



Trusted data solutions across a wide range of use cases.

Our Teams (Each team specialized in its domain and owns its data)



Each team specialized in its domain and owns its data.

Who makes it happen?



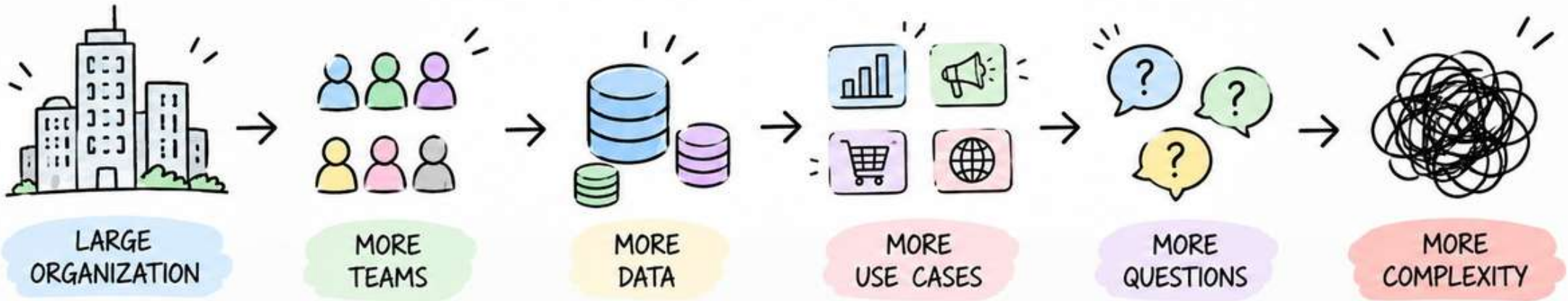
What we deliver?



Our solutions are a direct response to the practical realities of our specific environment.

MEZZOUR ISMAIL

Why did we need to change ?





TIME FOR A NEW APPROACH



FROM FRAGMENTATION TO AUTONOMY

How Platform Standards Transformed the Way We Work

A FEW YEARS AGO



→ Every new project, every question and every change went through the same experts with both technical expertise and business context.



→ Dependence on the central team due to a heterogeneous legacy stack.

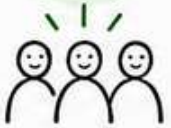


RESULT: DELAYS & BOTTLENECKS ☹

TODAY



→ Platform standards remain enforced.



→ Our goal is for teams to move forward independently.



→ Governance and autonomy coexist.



RESULT: STANDARDS & AUTONOMY ☺

IN THIS PRESENTATION



How platform standards enabled autonomy, reduced dependencies, and let teams focus on building, not operating.



MEZZOUR ISMAIL



EZZOUR ISMAIL

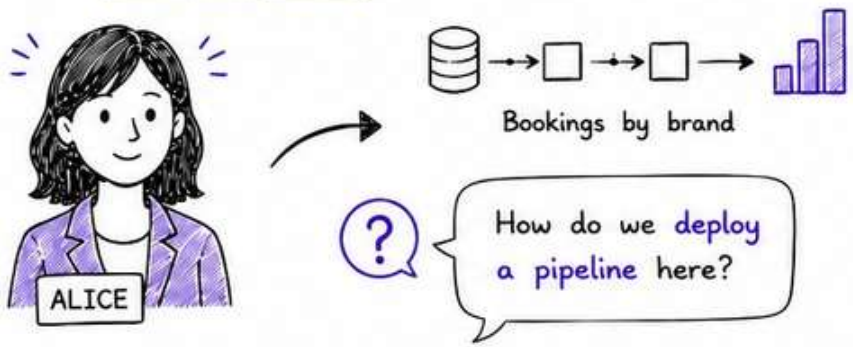


MEZZOUR ISMAIL

Alice's First Pipeline

1 New to the Revenue team

First mission, simple:
Build a pipeline that aggregates bookings by brand.



2 Five different answers.

We use stored procedures.

We use Talend for ETL.

talend

We use Dataiku on the platform.

data iku

Just write SQL.

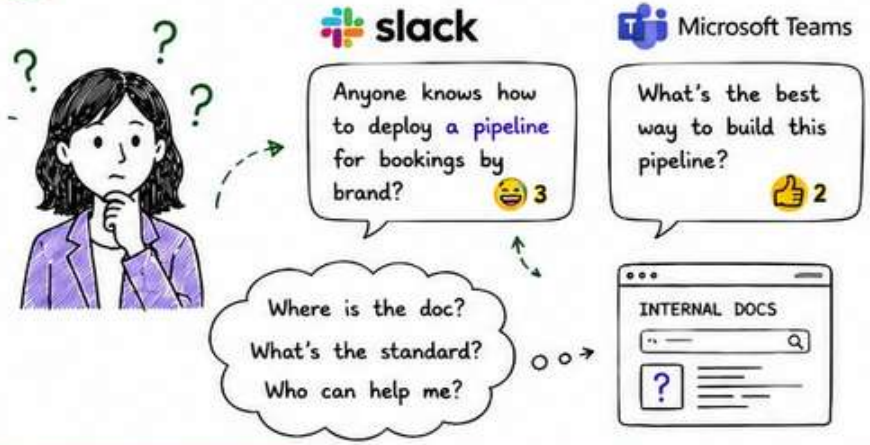
SQL

We provision everything with Terraform.



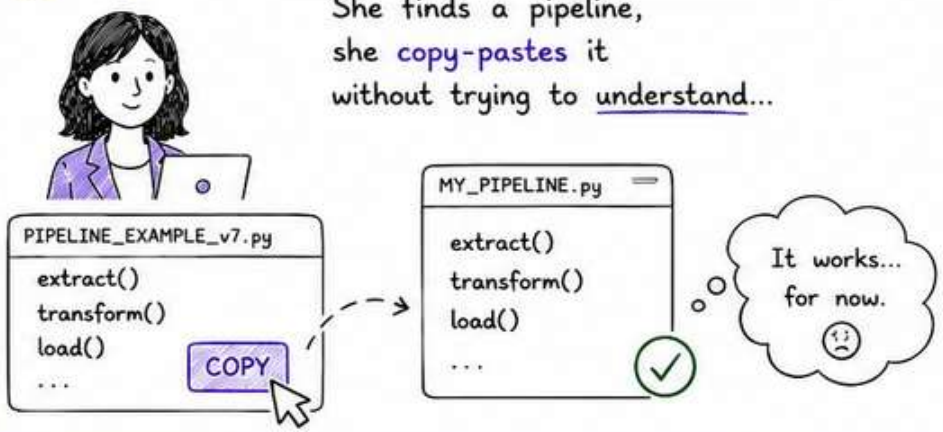
One week passes.
Zero lines of SQL written.

3 So Alice does what everyone does.



4 Key idea.

So Alice does what everyone does.
She finds a pipeline,
she copy-pastes it
without trying to understand...

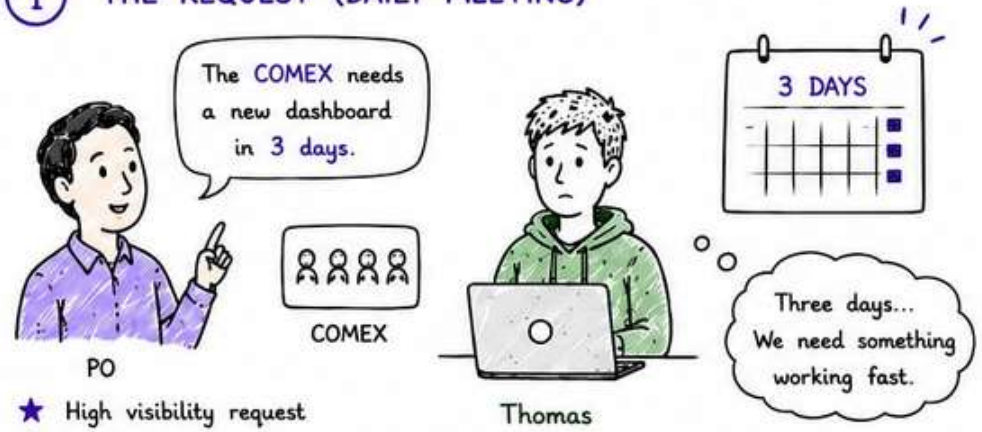


Knowledge lived in people, not in the platform.

MEZZOUR
ISMAIL

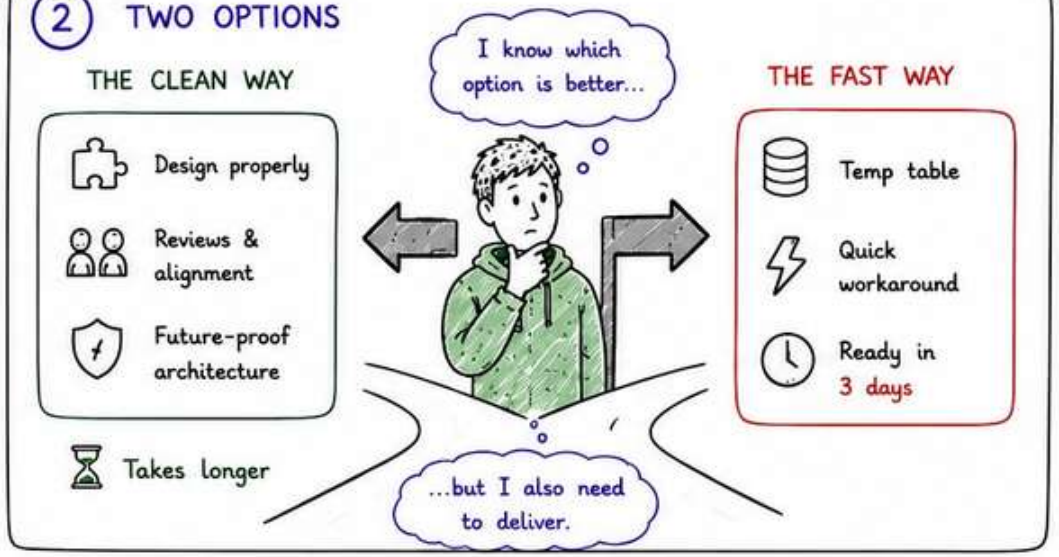
Thomas: Searching for the right path

1 THE REQUEST (DAILY MEETING)



- ★ High visibility request
- ★ Short deadline
- ★ Pressure is on

2 TWO OPTIONS



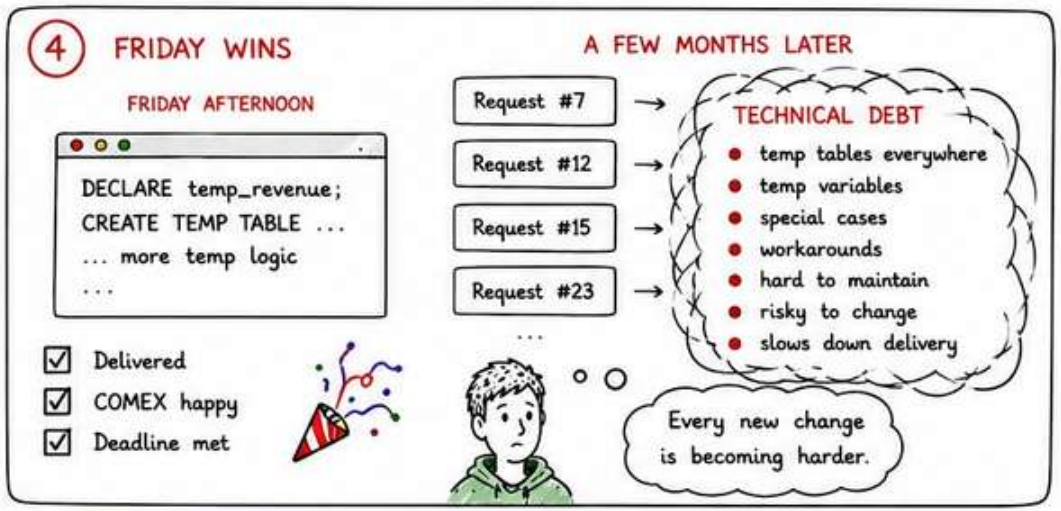
3 NO CLEAR STANDARD



★ No clear standard
→ Everyone interprets "good practice" differently.

Delivery pressure wins.

4 FRIDAY WINS



WITHOUT CLEAR PLATFORM STANDARDS,
LOCAL OPTIMIZATIONS BECOME TECHNICAL DEBT.

PIVOT

Imagine repeating this across 15 data mesh domains.
→ One shortcut becomes fifteen.
→ One workaround becomes a platform-wide maintenance problem.

MEZZOUR
ISMAIL

Ismail: Facing the scale effect

1 Ismail, the scale effect.



- 10 Platform people
- 100+ Engineers
- 15 Teams
- Across the organization

Lots of people. Lots of teams.
One platform.

2 Every Alice lost opens a ticket.



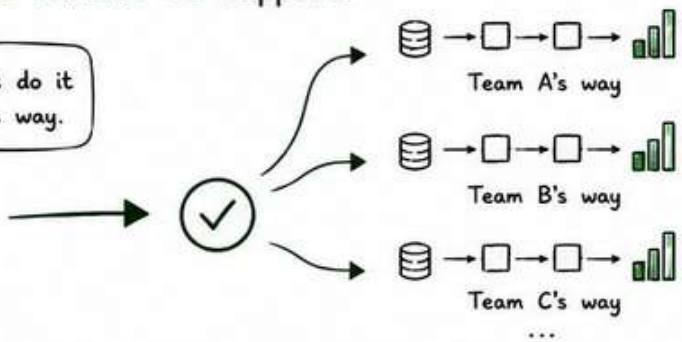
NEW TICKET
- Question
- No clear answer
- Waiting...

TICKET # 101 | TICKET # 102 | TICKET # 103 | ... | TICKET # 237

More confusion = More tickets

3 Every Thomas who decides creates one more variant to support.





More variants = More to maintain

4 The core team doesn't build anymore. They answer.



A DAY IN THE LIFE
- ☑ Answer questions
- ☑ Clarify
- ☑ Unblock
- ☑ Repeat...

All day. Every day.



TICKETS

We had become the human search engine of the data platform.

MEZZOUR ISMAIL

Three Personas, One Platform Problem

WHO	WHAT WE HAD	WHAT WE DID	WHAT HAPPENED (PROBLEMS)
 ALICE Analytics engineer	Too many tools, no clear starting point.	Searched everywhere for guidance and examples.	• Inconsistent practices • Lost productivity and time
 THOMAS Data engineer	High pressure and unclear engineering standards.	Prioritized delivery over long-term maintainability.	• Technical debt accumulated • Harder to maintain and evolve
 ISMAIL Data platform engineer	Repeated questions and growing support burden.	Provided guidance manually across multiple teams.	• Became a bottleneck • Less time for strategic work and improvement

MEZZOUR ISMAIL

What We Tried
What Didn't Work



MEZZOUR
ISMAIL

Three Personas, One Platform Problem

WHAT WE TRIED

DOCS

We wrote documentation for everyone.



VS.

- Docs are fragmented.
- Docs does not check if the rules are followed or not.





We have tried to hire people in the data platform team to better support people.



VS.

- Hard to hire for legacy ecosystems.
- Lengthy onboarding and ramp-up time.



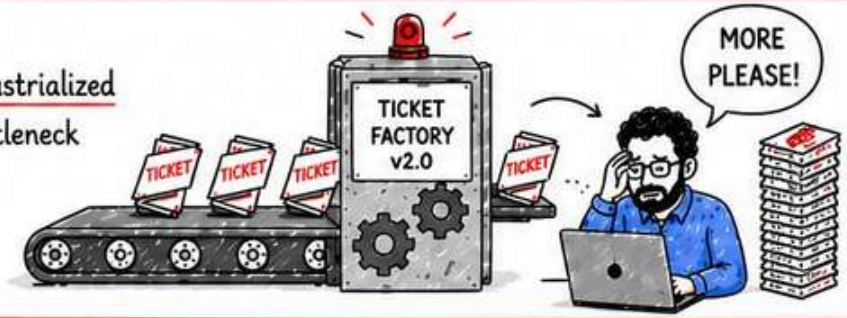
TICKETS

We wanted to reduce questions.



VS.

We industrialized the bottleneck instead.



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



MEZZOUR
ISMAIL



MEZZOUR ISMAIL

We stopped helping people choose.
We removed the choice.

BEFORE

Too many choices. Too many questions.

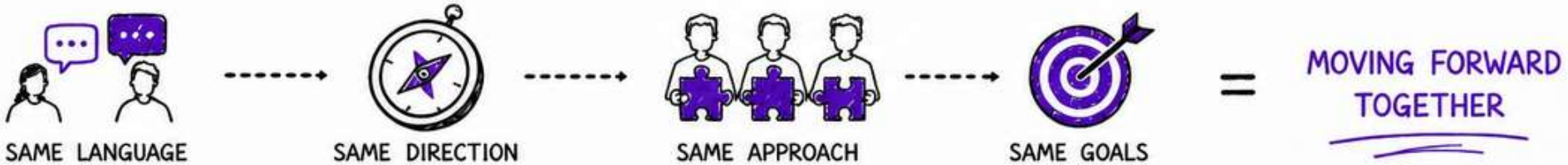


Result: slow decisions, inconsistency, and general fatigue.

AFTER

One clear path. Fewer questions. More impact.

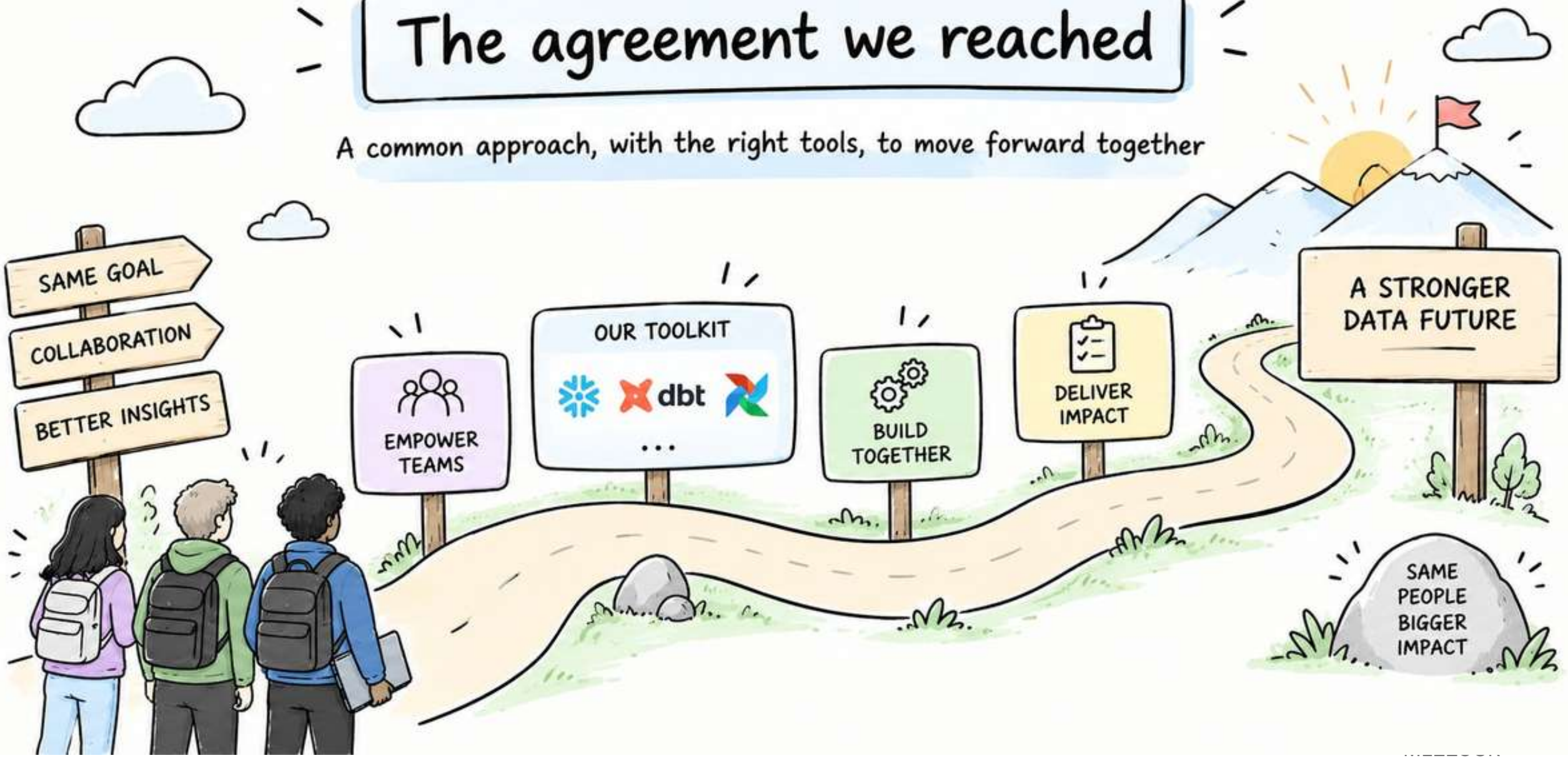




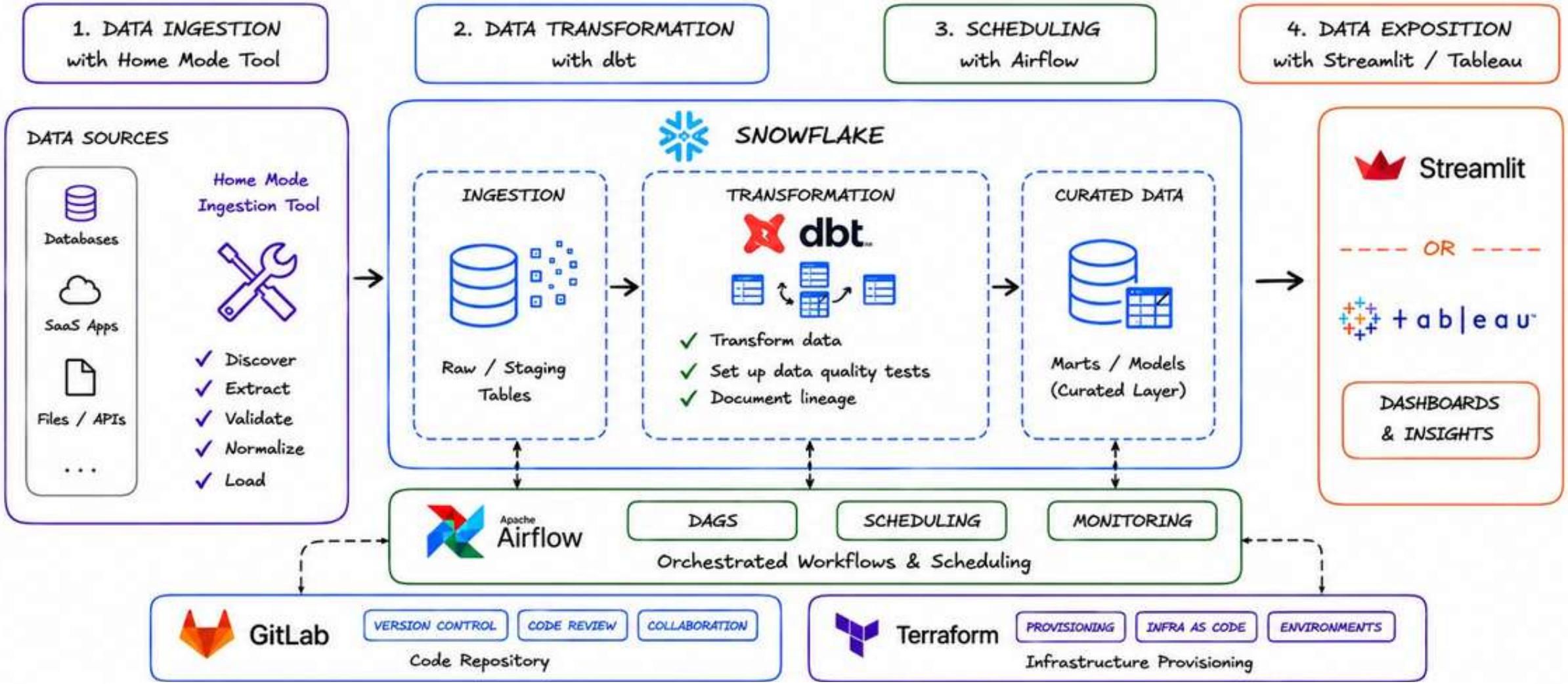
MEZZOUR ISMAIL

The agreement we reached

A common approach, with the right tools, to move forward together



The Agreement We Reached





MEZZOUR
ISMAIL

Target Technical Stack

HERE'S THE LANGUAGE EVERYONE SPEAKS NOW.





Where do we store our data ? → snowflake | STORES OUR DATA

How do we transform data ? → dbt | • Transform data • Ensure data quality • Document lineage

How do we orchestrate jobs ? → Airflow | ORCHESTRATES OUR WORKFLOWS

How do we version and deploy ? → GitLab | </> VERSIONS AND DEPLOYS

How do we provision infra ? → Terraform | PROVISIONS INFRASTRUCTURE

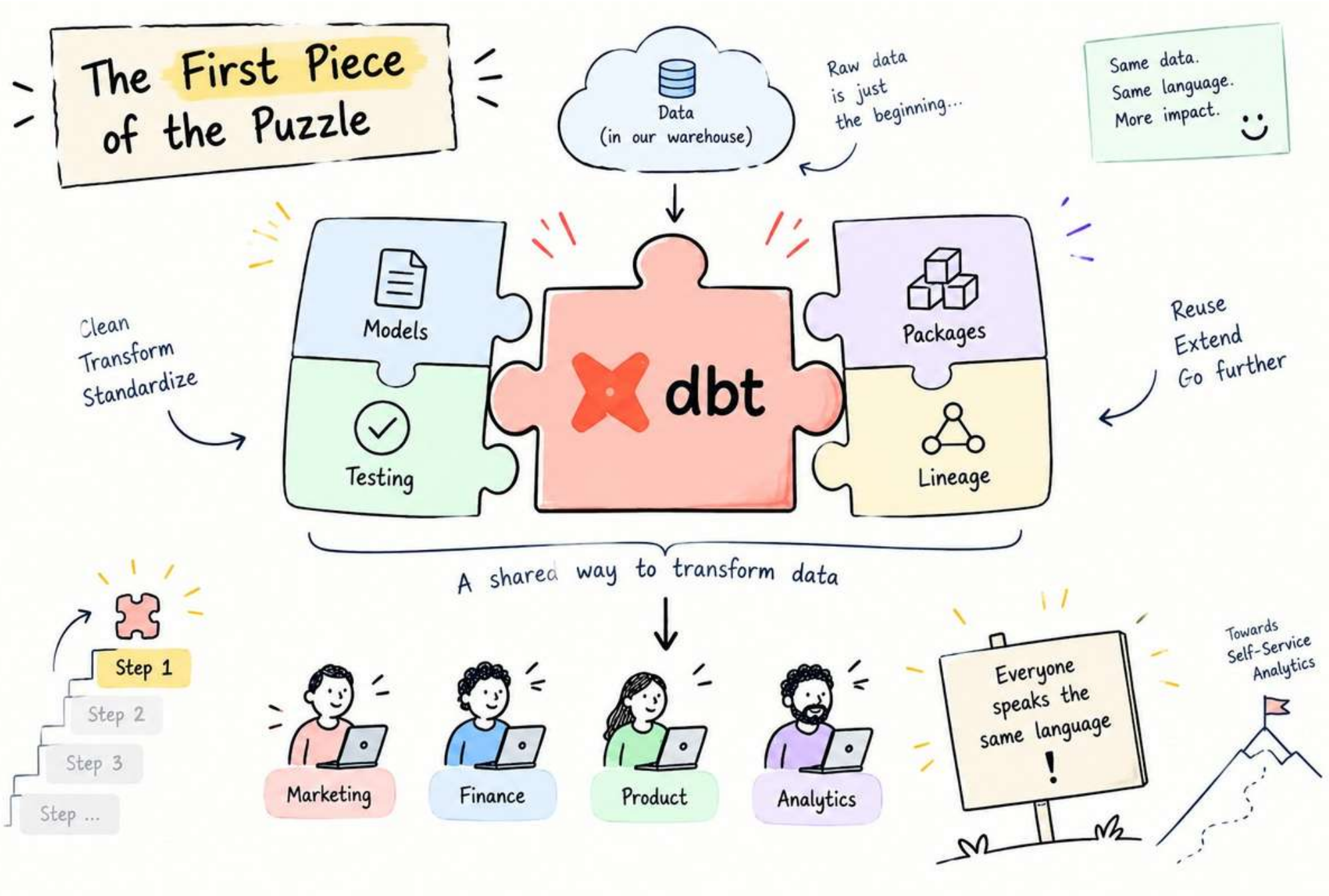


Each question has only ONE answer.

COMPOSITION OF TOOLS

Tools that work together to answer a need in a standard way and ensure consistency across the board.

MEZZOUR ISMAIL


The First Piece of the Puzzle
Data (in our warehouse)
Raw data is just the beginning...
Same data.
Same language.
More impact.
Clean
Transform
Standardize
Models
Testing
dbt
Packages
Lineage
Reuse
Extend
Go further
A shared way to transform data
Step 1
Step 2
Step 3
Step ...
Marketing
Finance
Product
Analytics
Everyone speaks the same language !
Towards Self-Service Analytics

Why DBT is our standard

From fragmented and manual... to standardized and automated.

BEFORE DBT

Many ways. Manual work. Inconsistent.

Data Transformations	Multiple ways to transform data...	Snowflake Procs, Python Scripts, Talend Jobs, ... and more
Lineage (manual)	We had to define manually what runs before and after...	TABLE A → TABLE B → TABLE C
Data Quality (inconsistent)	Each team had its own way of doing data quality...	TEAM A ✓ ✗ ✓ ... / TEAM B ✓ ✗ ✓ ... / TEAM C ✓ ✗ ...
Lineage (with Miro)	We used Miro to draw and maintain lineage...	miro
Testing (inconsistent)	Each team had its own way for non-regression tests and test coverage...	TEAM A: Custom Scripts + Manual Checks / TEAM B: PyTest + Custom Logic / TEAM C: Generic Tests + Spreadsheets ...

VS.

WITH DBT

One way. Automated. Consistent.

Data Transformations	One way to transform data:	dbt with SQL
Lineage (automatic)	Lineage is generated automatically.	TABLE A → TABLE B → TABLE C
Data Quality (standardized)	DBT provides a standard way for data quality tests.	not null, unique, relationships, accepted values ...
Lineage (automatic)	Built-in lineage is always up to date.	dbt docs
Testing (built-in)	DBT includes unit tests for stronger confidence and coverage.	dbt unit tests: select * from my_model where id = 1 -- expect ...





Great tool, real impact only with the right use

DBT is a super tool



It brings structure, visibility and reliability to our data.

But a tool misused remains ineffective...



It won't create value.
We'll go back to yesterday's problems.

Same problems, different tool.



Same pain.
Same waste.
No impact.

In our company, we speak the same language.

Shared definitions (one source of truth)

Aligned practices across teams



Clear direction, shared approach

Trust products with no technical debts and high confidence

Same language. Stronger collaboration. Better outcomes.

MEZZOUR ISMAIL

A tool without rules is difficult to manage

when you have 100+ people relying on it

and everyone wants to move fast and deliver.



MEZZOUR
ISMAIL

Same questions, same goal, same language

How do we make sure everyone uses dbt the same way and avoid the same issues as before?



Where do I put my models?

What SQL style should I use?



How should I name things?

What documentation should I write?

1. Where do I put my models?
2. How should I name them?
3. What SQL style should I use?
4. What documentation should I write?
5. What tests should I add?



Good questions!

Let's think together.

Different teams, different answers.

Team Revenue | Team Marketing | Team Loyalty

The problem is not the tool.

The problem is how we use it.

Key idea

dbt standardizes the language.
It does not standardize how teams use it.

⇒ Our next step: standardization.

MEZZOUR
ISMAIL

Same questions, same goal, same language

Let's agree on how we use dbt together!



– OUTPUT –

A SHARED PLAYBOOK

Clear rules

A way to evaluate projects

MEZZOUR ISMAIL

HOW WE SCORE EACH DBT PROJECT

Each project gets a score, calculated based on six dimensions

PROJECTS	
marketing_dbt	92
sales_dbt	76
finance_dbt	58
loyalty_dbt	34
operations_dbt	81
...	

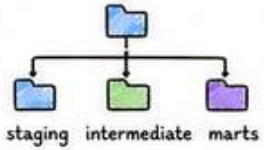
PROJECT SCORE: 92 /100 (0 — 50 — 100)

80–100	Excellent
60–79	Good
40–59	Needs improvement
0–39	Poor

This score helps us MEASURE and DRIVE the adoption of standards.

It also helps us VERIFY that the platform is used the right way.

1 FOLDER STRUCTURE



Models organized into the right layers (staging, intermediate, marts)

90/100

2 MODEL NAMING

- ✓ stg_customers.sql
- ✓ int_orders.sql
- ✓ fct_sales.sql
- ✓ dim_product.sql
- ✗ customers_final.sql

Consistent prefixes enforced (stg_, int_, fct_, dim_)

80/100

3 OWNERSHIP

stg_customers	@data-eng
int_orders	@analytics
fct_sales	@sales
...	

Every model has a declared owner

70/100

4 TEST COVERAGE

fct_sales

column	test
amount	✓
quantity	✓
order_id	✓
is_active	✗

At least one test on every non-boolean mart column

80/100

5 DOCUMENTATION

fct_sales

amount DECIMAL(10,2)

Total sales amount in USD after discounts and returns.

Typed columns, descriptions that reflect actual model logic

70/100

6 TAGGING

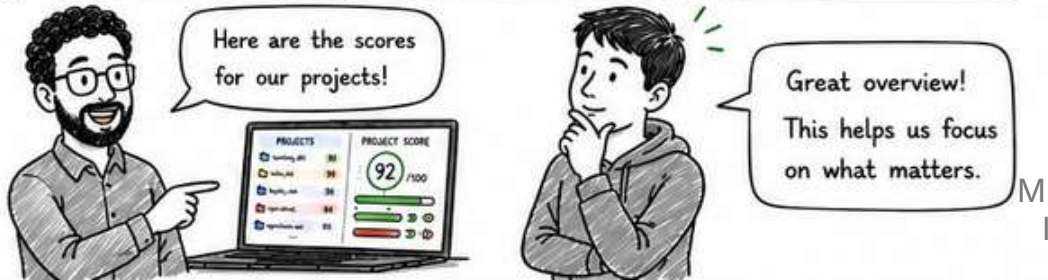
finance ✓ mart ✓ sales ✓

important ✗

Tags validated against an approved taxonomy, no freeform values

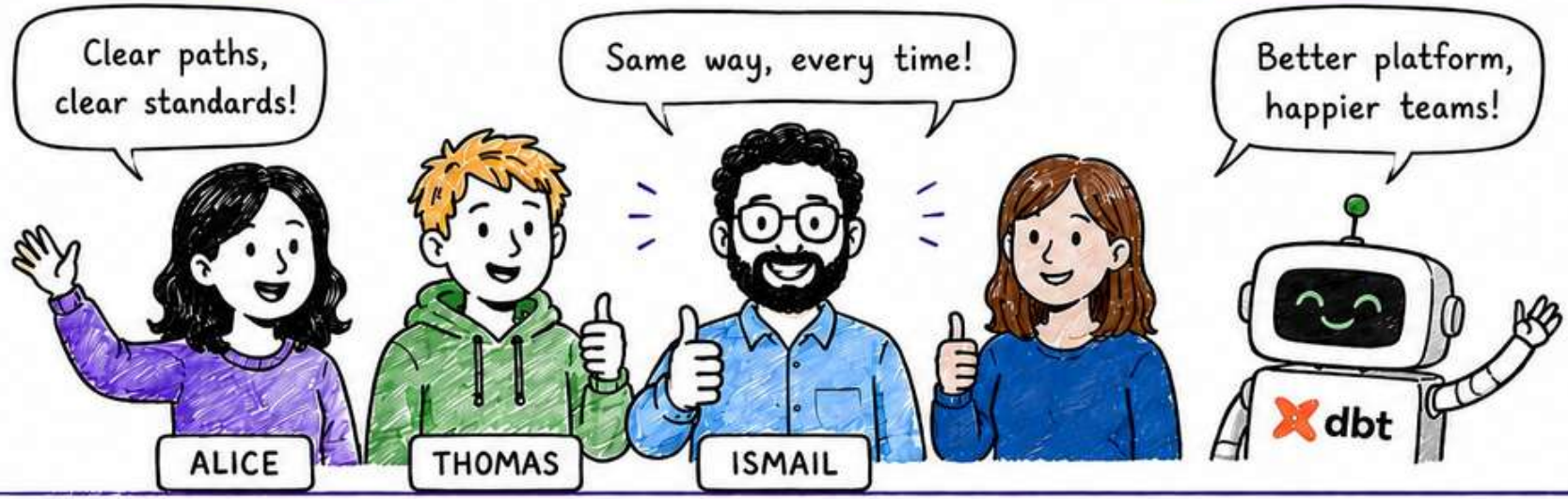
70/100

This is our homegrown STREAMLIT application. It gives us a simple and clear view of scores across all DBT projects.



MEZZOUR ISMAIL

WE'RE ALIGNED.
WE SHIP BETTER DATA.



CLEAR RULES + **A WAY TO EVALUATE PROJECTS**

BETTER DATA. TOGETHER.



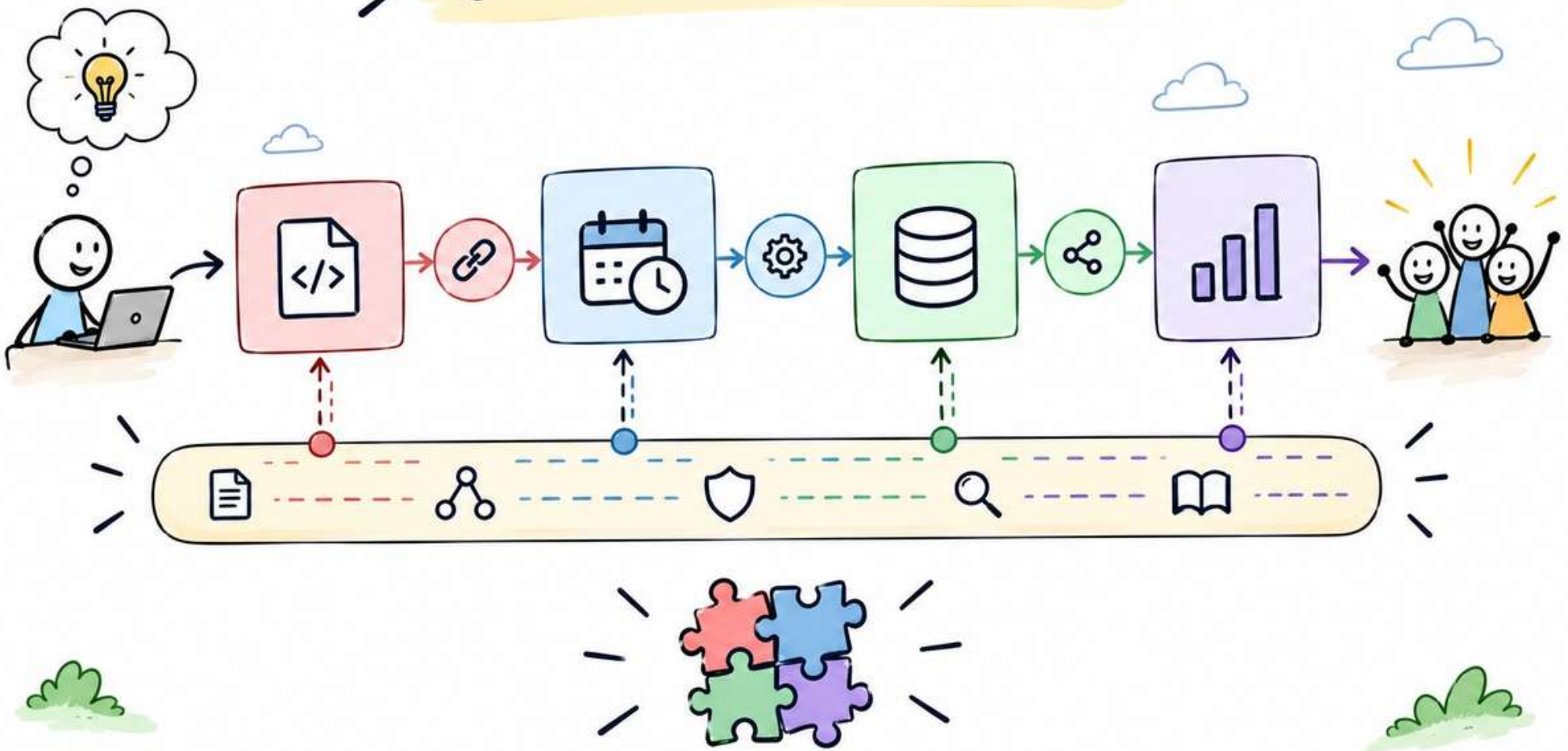
BUT **DBT** IS NOT ENOUGH.
IT'S THE **ENGINE**, BUT...

→ BUT HOW TO **COMPLETE** THE CAR NOW?



MEZZOUR
ISMAIL

But how do these tools talk to each other ?



dbt is an engine, but an engine alone can't move a car.

So we built the car around this engine.

dbt
The engine
Powerful for transformations (with SQL)





It's powerful, but it can't...

- We can't schedule runs automatically
- We can't deploy to production
- We can't provision or manage infra
- We can't provide a consistent experience for every team

Airflow
4-wheel drive for direction
Schedules, orchestrates and drives the execution where it needs to go.

GitLab
The maintenance logbook
Versioning, testing, and deployment history. Know what was done, by whom and why.

Data Platform Team
The pit crew
Defines standards, builds tools, ensures reliability and supports everyone.



Terraform
The connection and plumbing
Provisions and connects the infrastructure. Makes everything work together.



dbt gives us the power. Our platform gives us motion.

dbt gives us the power.

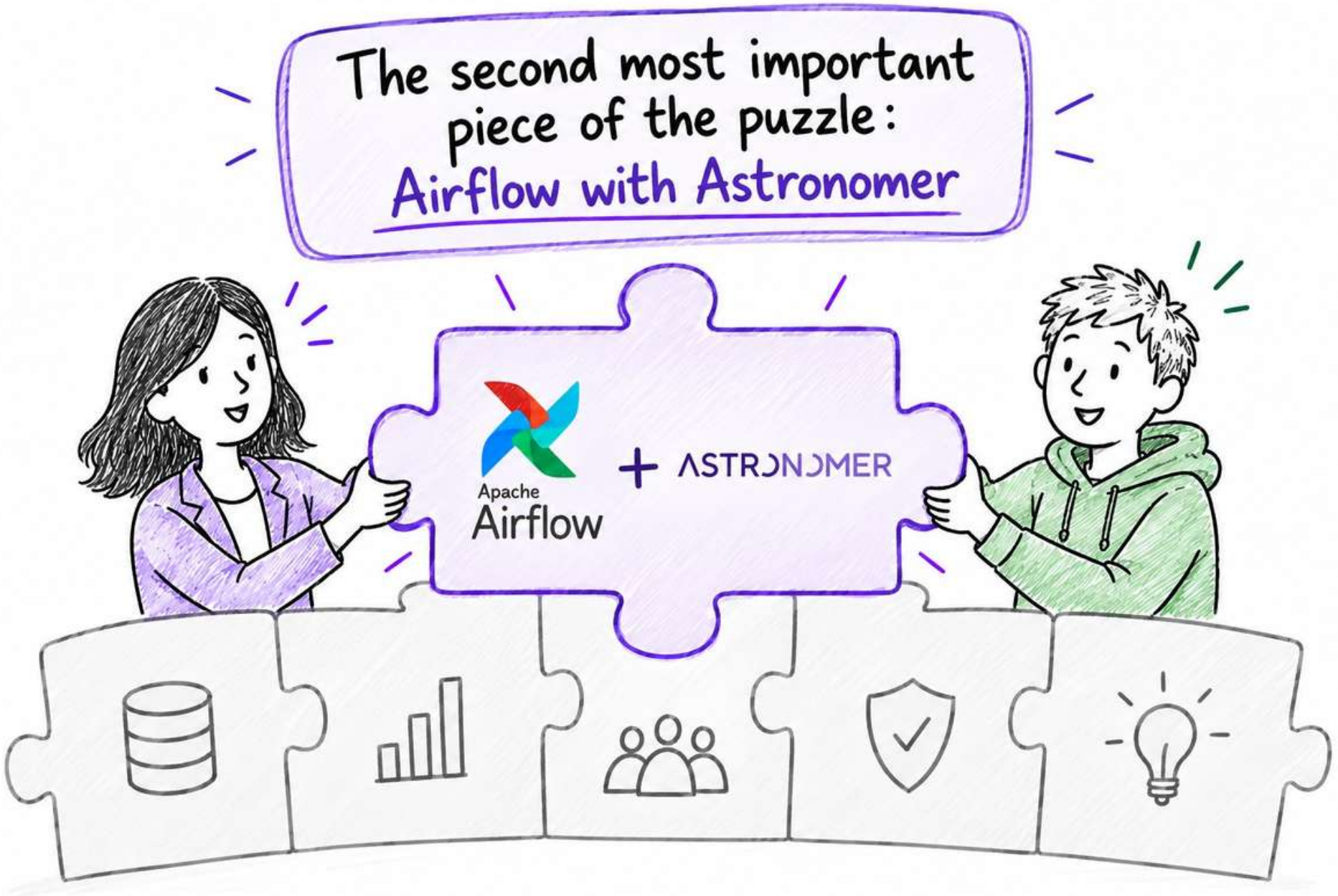
The platform around it gives us motion.

- Consistent way of working
- End-to-end automation
- Scalable and reliable data products

The right combination takes us further, faster.



MEZZOUR ISMAIL


The second most important piece of the puzzle:
Airflow with Astronomer
Apache Airflow
+
ASTRONOMER

AIRFLOW COMPLETES OUR TECH STACK

Airflow is our orchestrator and scheduler.
It connects our tools to the world.

BEFORE: ORCHESTRATING JOBS WAS HARD



- Manual triggers and cron jobs
- Scattered scripts and dependencies
- Hard to monitor and debug
- Difficult to adapt and scale

Time-consuming, error-prone, and hard to scale.

NOW: EASY, RELIABLE, AND SCALABLE



- Schedule everything
- Monitor & retry
- Dependencies managed
- Scalable & reliable

Airflow
ORCHESTRATOR & SCHEDULER

CONNECTED TO THE WORLD

- dbt — Data transformations
- snowflake — Data warehouse
- tableau — Analytics & visualization
- slack — Alerts & notifications
- aws — Cloud services (S3, EC2, RDS, ...)
- salesforce — Business applications
- Google Sheets — Spreadsheets
- apis (...) — Any API



Airflow brings clarity, automation and visibility to our data workflows.

We focus on building value, Airflow takes care of orchestrating everything!

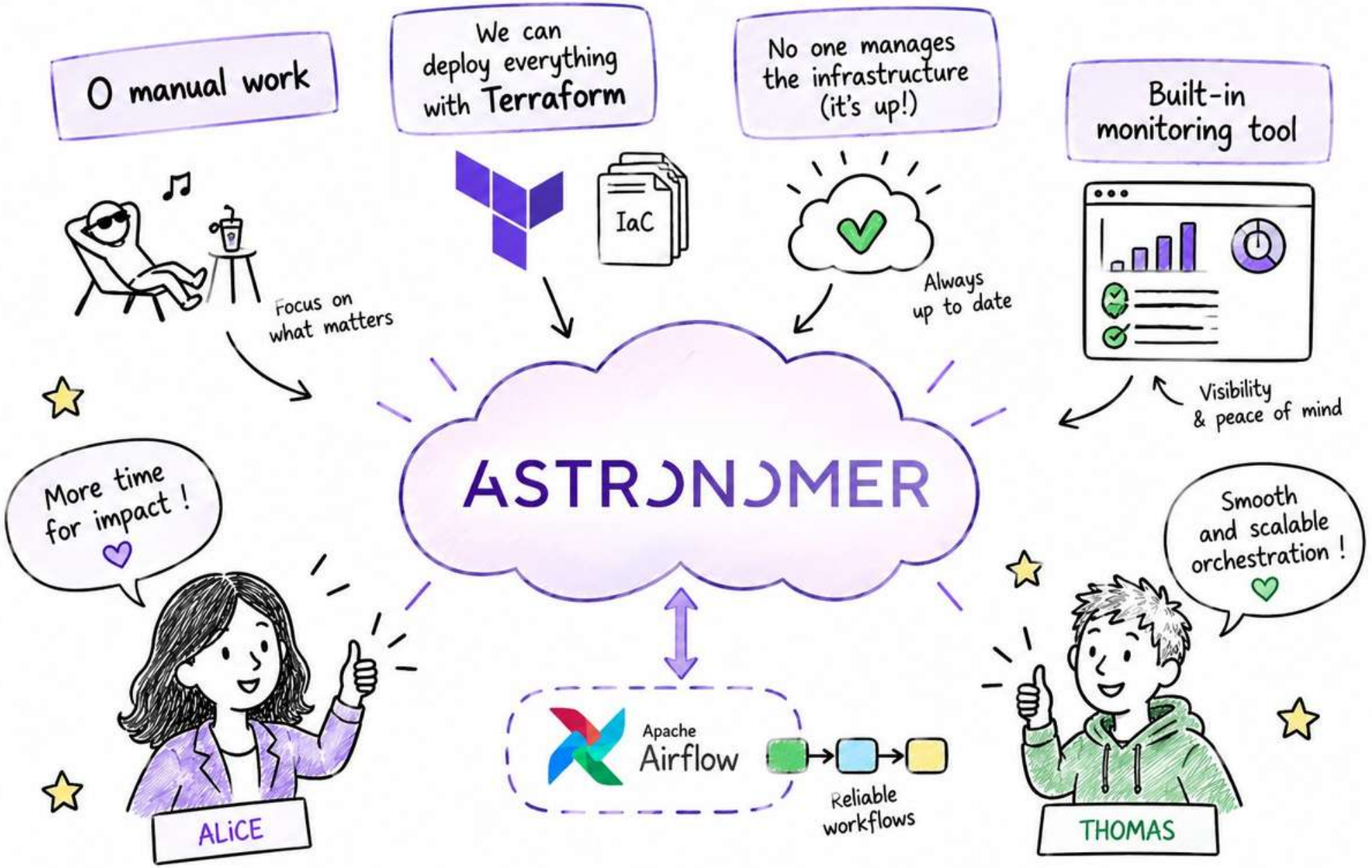


Airflow gives us the power.
The platform around it gives us motion.

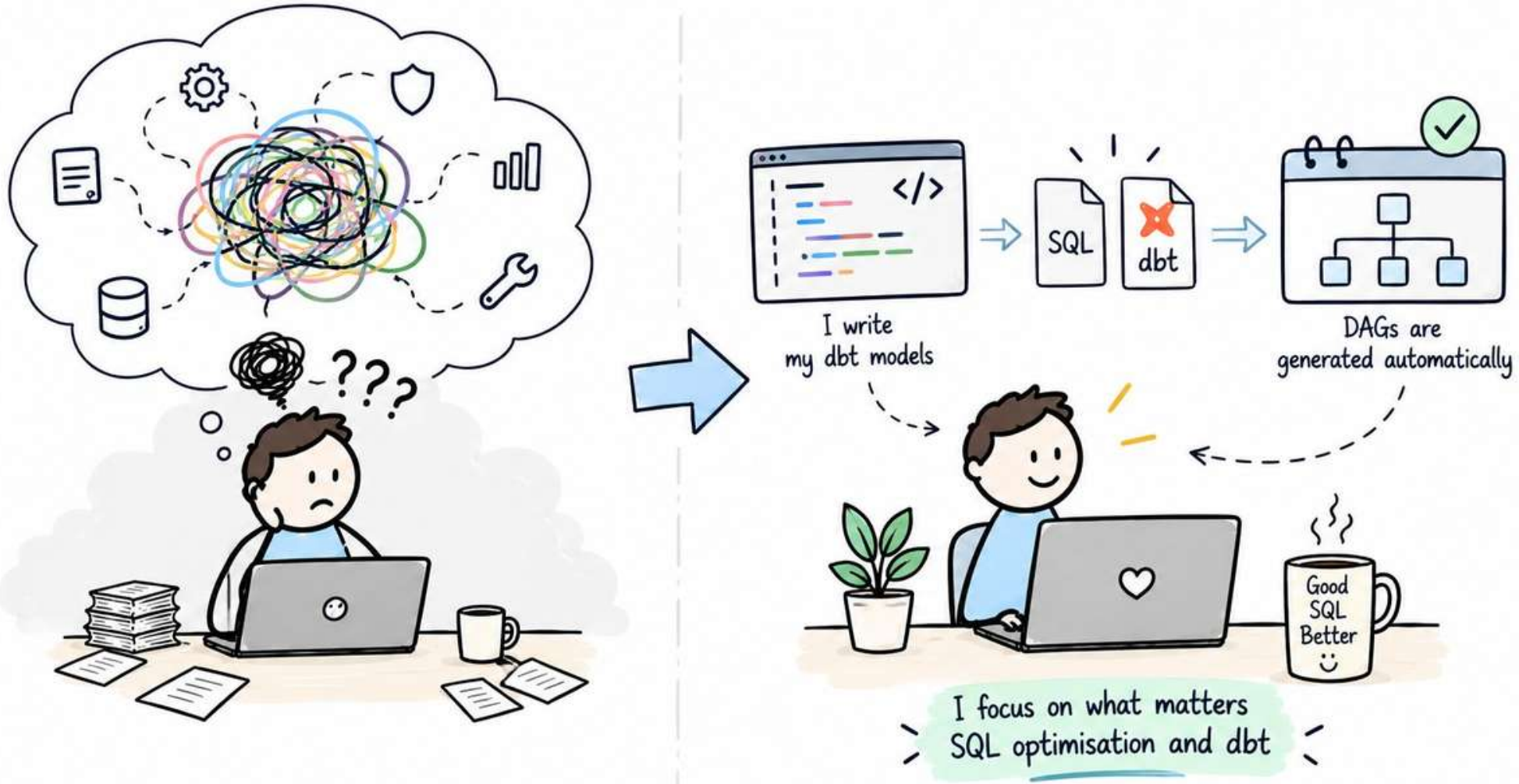
- Consistent way of working
- End-to-end automation
- Scalable and reliable data products



MEZZOUR ISMAIL


0 manual work
We can deploy everything with Terraform
No one manages the infrastructure (it's up!)
Built-in monitoring tool
IaC
Focus on what matters
Always up to date
Visibility & peace of mind
ASTRONOMER
More time for impact !
Smooth and scalable orchestration !
Apache Airflow
Reliable workflows
ALICE
THOMAS

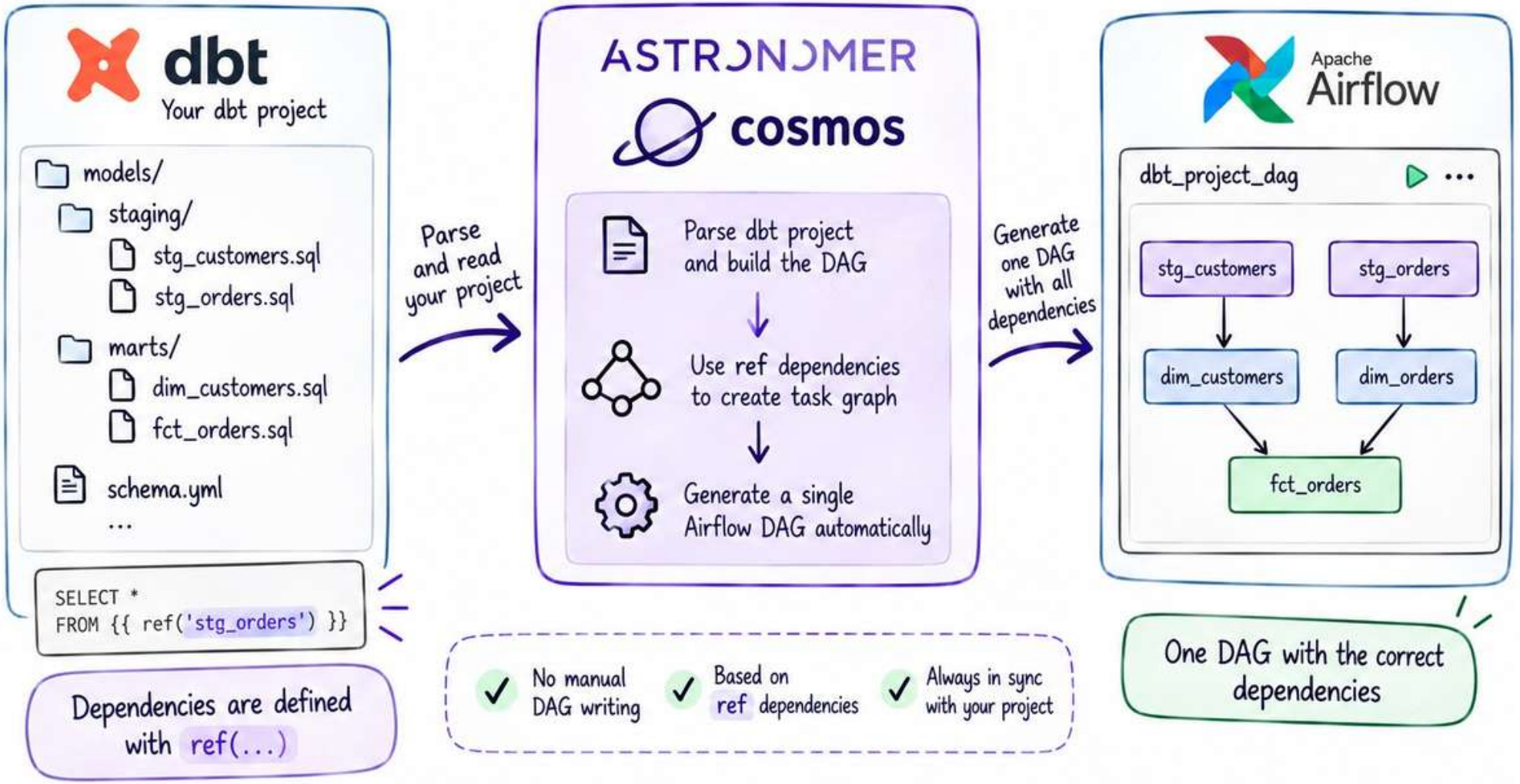
How do we allow Analytics Engineers to focus only on SQL optimisation and dbt ?



MEZZOUR
ISMAIL

From your dbt project to an Airflow DAG

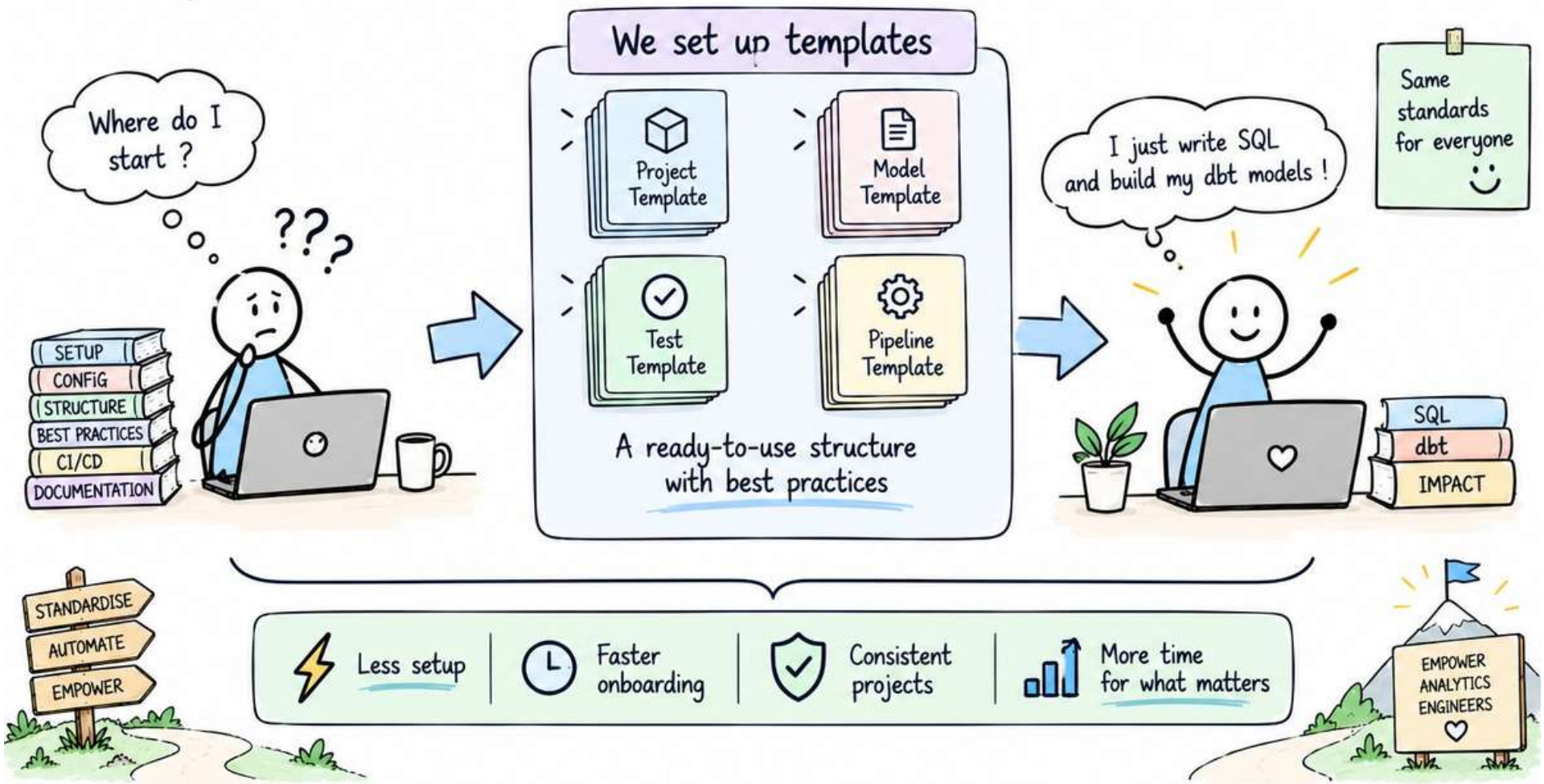
Automatically with dependencies (thanks to `ref`)



MEZZOUR
ISMAIL

We went even further

to make everything easy for our Analytics Engineers



FOCUS ON WRITING & OPTIMIZING SQL.
WE TAKE CARE OF WHAT RUNS, WHEN AND HOW.

DAG TEMPLATE

MANAGED BY
DATA PLATFORM TEAM

- Best practices
- Monitoring
- Retries & alerts
- Logging
- Dependencies
- ... and more

Standardized, secure and production-ready

Apache Airflow

Cosmos

AIRFLOW PACKAGE

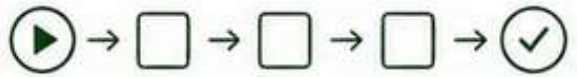
- Auto-generate DAGs from dbt
- Keep SQL & orchestration in sync

One package.
Zero orchestration complexity.

3 LINES OF CONFIGURATION
+ MORE OPTIONS

```
1 dbt_project: "marketing"
2 schedule: "0 2 * * *"
3 alerts_channel:
  "https://teams.microsoft.com/l/
  channel/19%3Aalerts-data"
# more options ...
```

GENERATED DAG



Cosmos generates the DAG.
You focus on what matters.

YOUR TEAM



FOCUS ON WRITING
& OPTIMIZING
SQL QUERIES.

NO NEED TO THINK ABOUT
WHAT RUNS BEFORE OR AFTER.

1. STANDARDIZED & RELIABLE

- Production-ready DAG template with best practices.

2. CONFIGURATION OVER CODE

- Just a few lines of config, Cosmos generates the DAG.

3. FOCUS ON WHAT MATTERS

- Teams write and optimize SQL, not orchestration.

MEZZOUR
ISMAIL


What about other operations not related to dbt ?
We use custom operators !

WE BUILD CUSTOM OPERATORS

A custom operator is:

Reusable building blocks for workflows

Apache Airflow

DATA PLATFORM TEAM



We set up operators.

- Tested
- Documented
- Reliable

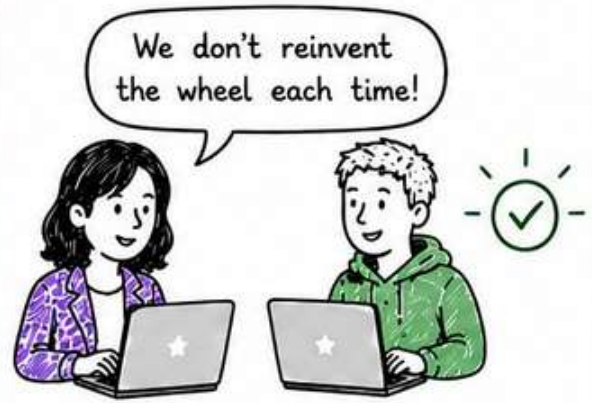
Snowflake cost monitoring operator	Astronomer setup operator	dbt & Airflow integration operator

OUR OPERATORS

dbt Operator	Run dbt transformations
Snowflake Operator	Execute Snowflake workloads
Tableau Operator	Interact with Tableau assets
More Operators	Extensible framework

Built once. Used everywhere.

FEATURE TEAMS



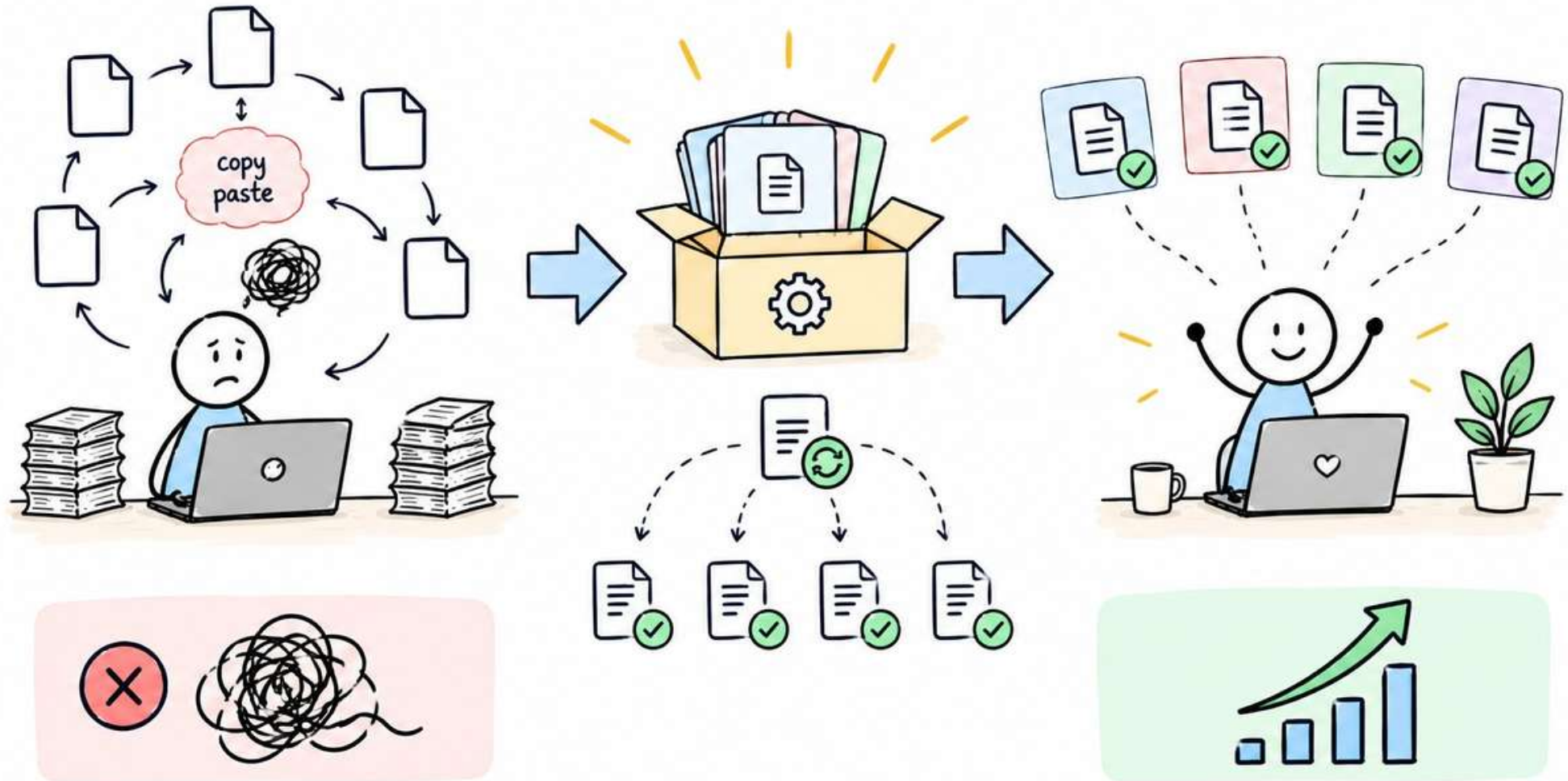
Monitor costs	Reliable scheduling	Build & run transformations

We focus on building features, not plumbing.

Build once. → Empower everyone. → Deliver something great.

MEZZOUR ISMAIL

Scale without losing control



MEZZOUR
ISMAIL

GitLab
Code. Review. Deploy.

Terraform
Infrastructure as Code.

BUILD ONCE. MAINTAIN FOREVER.

PREPARE EVERY CAR. READY FOR THE ROAD. READY FOR MAINTENANCE.

We prepare every car the same way. Ready for the road. Ready for maintenance.



TEMPLATE (CAR BLUEPRINT)

- Standard components
- Safety checks
- Maintenance ready

Built by the platform team

ALL CARS USE THE SAME BLUEPRINT

- Car_1 — Pre-configured, Ready to drive
- Car_2 — Pre-configured, Ready to drive
- Car_N — Pre-configured, Ready to drive

SERVICE STATION (CI/CD PIPELINE)

- Shared maintenance tools
- Common inspections
- Automated servicing

Keeps every car healthy and road-ready

MANAGED BY DATA PLATFORM TEAM



Ismail

BUILD A NEW CAR

FROM THE BLUEPRINT
- ✓ Standard components
- ✓ Safety checks
- ✓ Maintenance ready



READY FOR THE ROAD!

GitLab — Source code and CI/CD

Terraform — Provision and manage infrastructure

Pre-configured. Safe. Reliable. Ready to drive.

MAINTAIN EXISTING CARS

APPLY MAINTENANCE TO ALL CARS
- ✓ Updates applied
- ✓ Inspections run
- ✓ Automated servicing

Car_1 ✓ Car_2 ✓ ... Car_N ✓

All cars serviced

✓ Same maintenance for every car. No manual action, no drift.

MEZZOUR ISMAIL

CONCLUSION

Let's take a step back...



BUSINESS TEAMS

Focus on business needs
Understand priorities and ask the right questions.

Deliver value
Turn data into insights that drive better decisions.

Create impact
Use data to solve problems and support the success of the business.





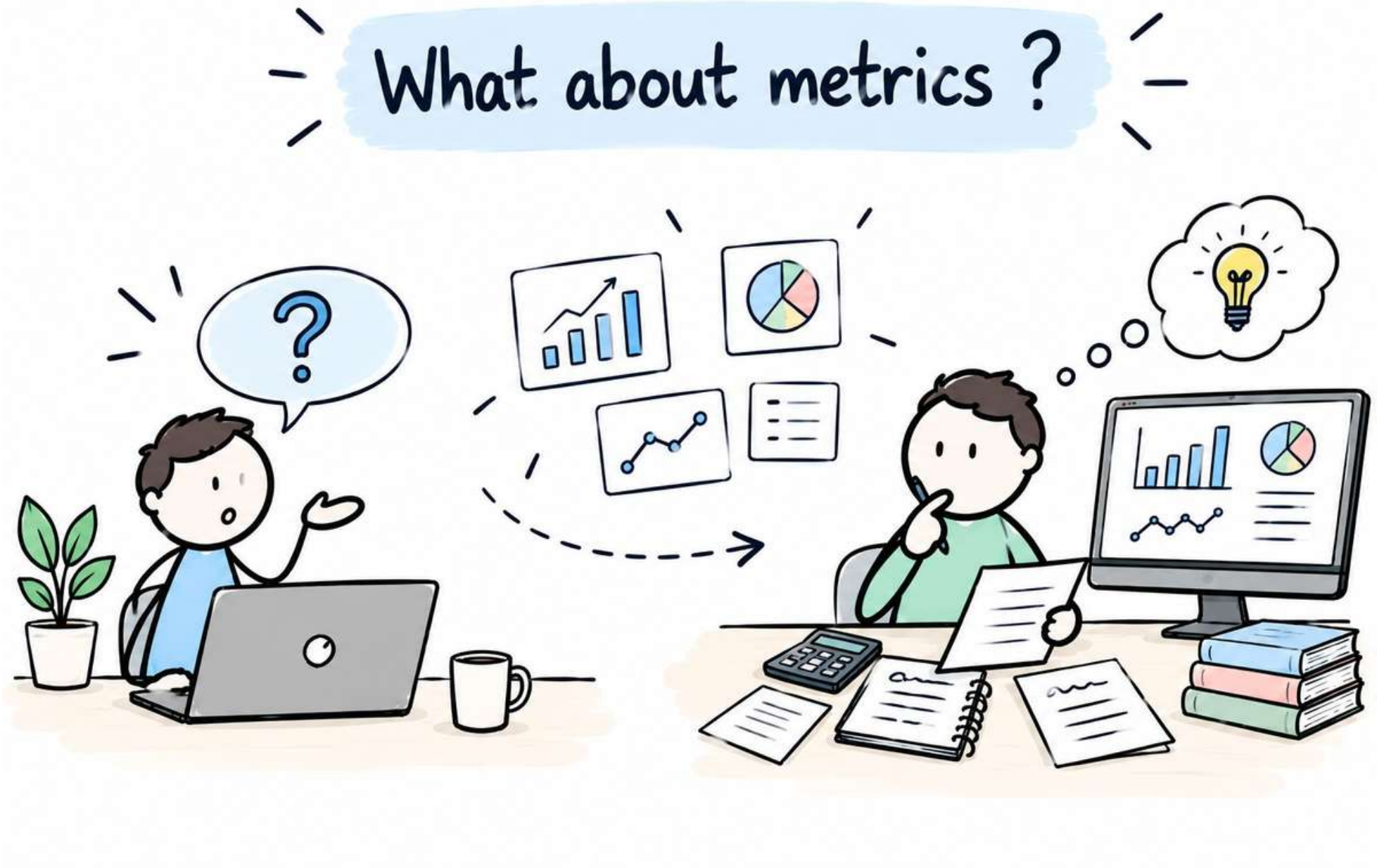
DATA PLATFORM TEAM

Continuously improves processes
Ensures a stable and efficient platform without constant interruptions.

Focuses on delivery and technology watch
Delivers reliably while staying informed about the best technologies.

Focuses on new needs and the future
Explores and enables new capabilities (semantic layer, talk to your data, agents, ML...) to prepare for what's next.

MEZZOUR
ISMAIL


What about metrics ?

The Impact: Before vs After Self-Service Data Platform

Metric	Before	After
Engineer Onboarding	2 Weeks	1 Day
Data Product Change	2 Sprints	1 Week
New Project Setup	2-3 Days	1 Minute
Time on Orchestration	30%	0%
Quality Validation	Days	Instant
Feature Testing	High Risk	Safe Sandbox
Support Response	Hours / Days	Seconds

MEZZOUR
ISMAIL

Human investment

2 data platform engineers

to design and implement the airflow and dbt platform and to support **15 teams** in their first use case development

Collaboration with tech leads

to establish **common practices** across domains

Tools / products of our data stack

snowflake — Licensed platform

dbt — Open-source (free)

Apache Airflow — Managed platform (Astronomer)

Cost

snowflake

4%

reduction compared to Q2 last year over the same period on Snowflake.

Apache Airflow

30%

cost reduction between the time we decommissioned other orchestrators in Q2.

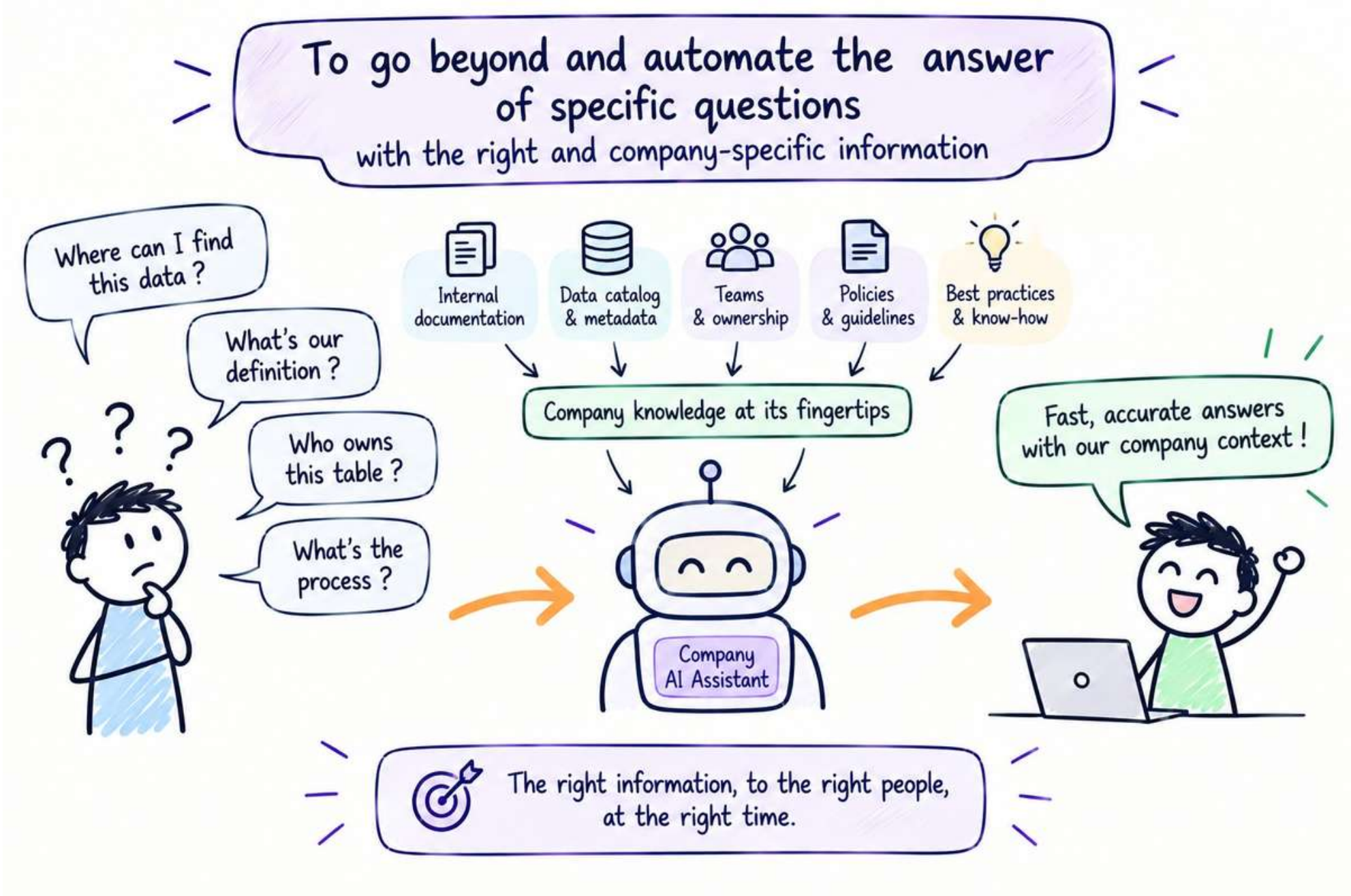
Maintenance cost reduced:

5 → 1

from 5 orchestrators to 1.

Additionally, we are reducing costs by **10%** in May.

MEZZOUR ISMAIL

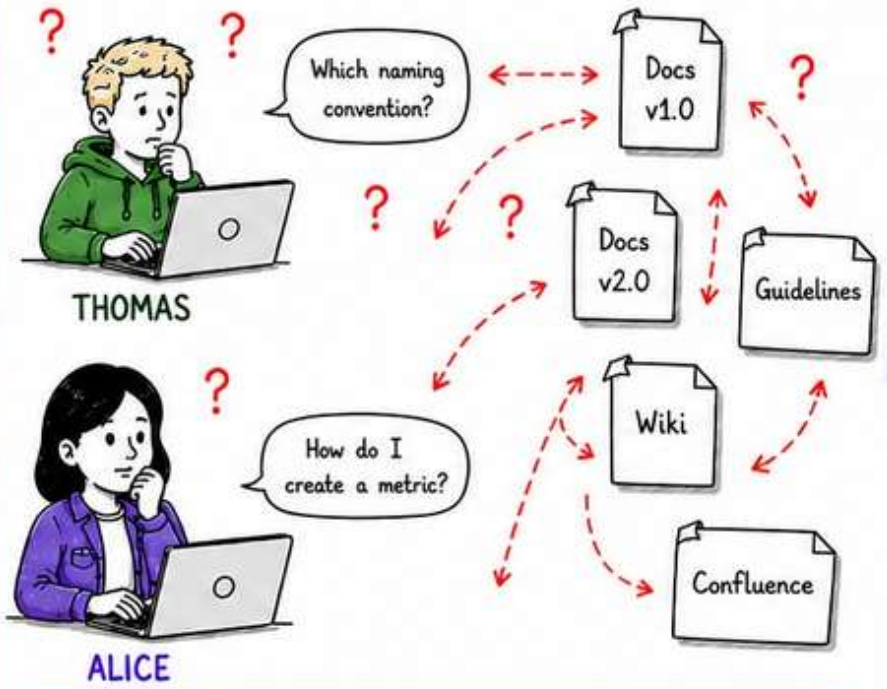


MEZZOUR
ISMAIL

EXAMPLE OF CONTINUOUS PROCESS IMPROVEMENT

Standardization + AI-powered knowledge access = Less friction, more impact

THE CHALLENGE THAT REMAINED

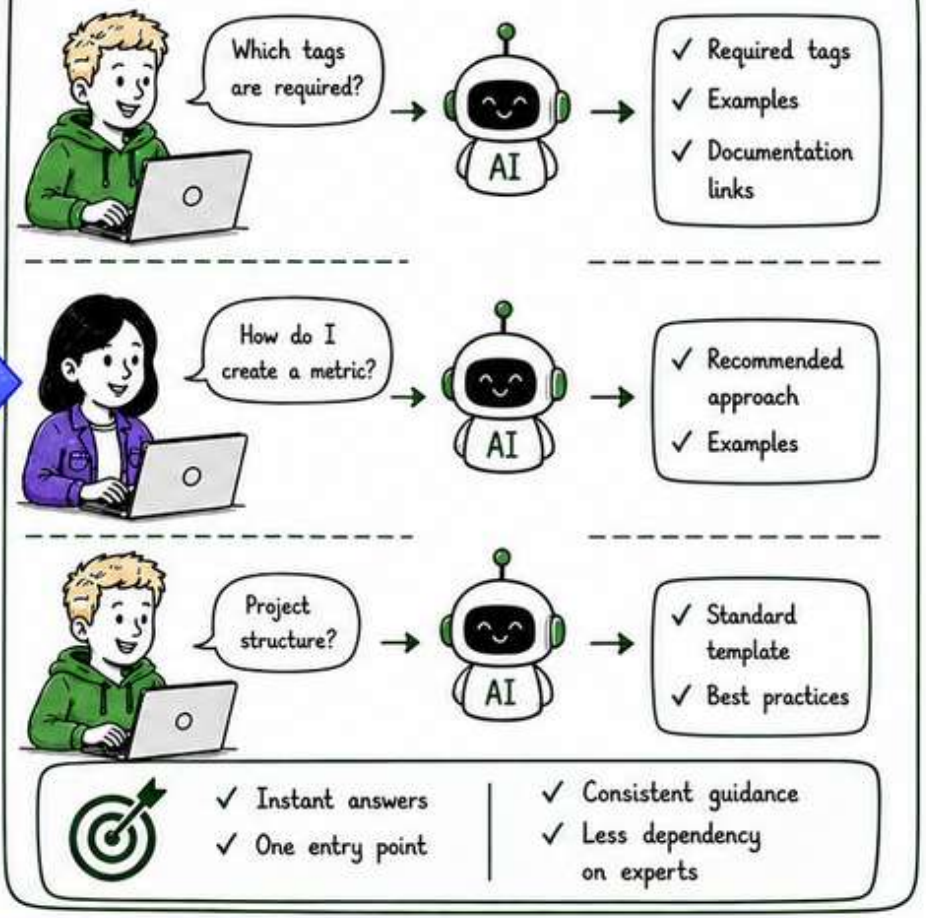


- ✕ Searching information
- ✕ Repeated questions
- ✕ Multiple sources
- ✕ Expert dependency

PLATFORM STANDARDS IN PLACE

- Templates ✓
- Custom Operators ✓
- Naming Conventions ✓
- Governance Rules ✓
- Best Practices ✓

AI-POWERED KNOWLEDGE ACCESS



- ✓ Instant answers
- ✓ One entry point
- ✓ Consistent guidance
- ✓ Less dependency on experts

Standards solved how to build.
AI solved how to find.

THIS HAS FREED UP TIME TO WORK ON OTHER THINGS!

MEZZOUR ISMAIL

Podcasts, Articles & References



Links:

- ❑ Podcast
- ❑ How we set up dbt and airflow
  - ❑ article 1
  - ❑ article 2
- ❑ How we turned data teams into a streamlit factory: article
- ❑ Escaping the Data Platform Maze: How Airflow Helped Us Take Back Control article
- ❑ Building a Semantic Layer in Snowflake: article

Same content.
More details.
For the curious ones.

Same journey,
more details !



Annexe

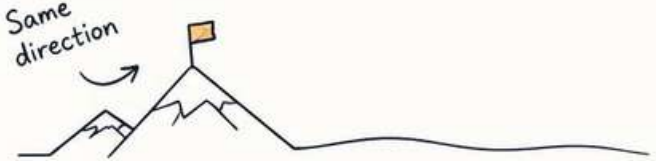
Go deeper
when you're ready

- [x] Extra examples
- [x] Useful queries
- [x] Configs
- [x] References
- [x] And more...

DEEP
DIVE
MODE

For the
curious minds

Same
direction



MEZZOUR
ISMAIL

NAMING & TAGGING RULES = CONSISTENCY, CLARITY & CONTROL

DAG NAMING STRUCTURE

team_domain_type

team domain type

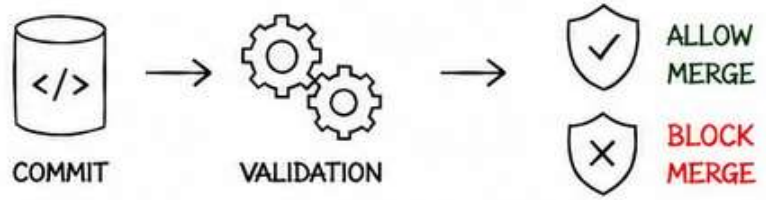
REQUIRED DAG TAGS

Owner

Domain

Criticality

ENFORCED BY CI/CD



EXAMPLE 1: NON-CONFORMING (MERGE BLOCKED)

DAG NAME	DAG TAGS
daily_sales_update	compliance_core_business
	compliance_sales
	compliance_medium

✗ Does not follow team_domain_type

EXAMPLE 2: COMPLIANT (MERGE APPROVED)

DAG NAME	DAG TAGS
compliance_marketing_growth_ingestion	compliance_growth
	compliance_marketing
	compliance_high

✓ Follows team_domain_type

DAG	Owner (Team)	Runs	Schedule	Next Run	Actions
sales__kpi_daily_aggregation Aggregate and publish daily sales KPIs sales kpi reporting	Sales	2	@daily	2024-05-22, 00:00 In 02:48:57	
marketing__leads_ingestion Ingest marketing leads from source systems marketing leads ingestion	Marketing	1	@hourly	2024-05-21, 21:00 In 00:48:57	
revenue__360_build Build and refresh revenue 360 mart revenue mart transformation	Revenue	1	@daily	2024-05-22, 01:00 In 04:48:57	
loyalty__data_quality_checks Run data quality checks on critical datasets loyalty dq validation	Loyalty	3	@daily	2024-05-21, 24:00 In 02:48:57	

MEZZOUR ISMAIL

THOMAS & ALICE WORKFLOW

1 USE CONFIG FILE OR EXISTING OPERATORS

config.py

existing operators



2 TWO WAYS TO BUILD YOUR PIPELINE

A. BUILD WITH dbt (TRANSFORMATIONS)

dbt

- Use template
- Focus on SQL
- Dependencies generated automatically

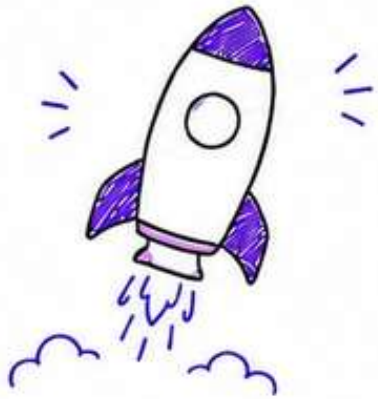
B. BUILD SOMETHING ELSE (REUSE OPERATORS)

Use operators available

3 VISUALIZE & VALIDATE YOUR DAG

- Check structure
- Check dependencies
- Ensure everything works

4 DEPLOY TO ENVIRONMENTS



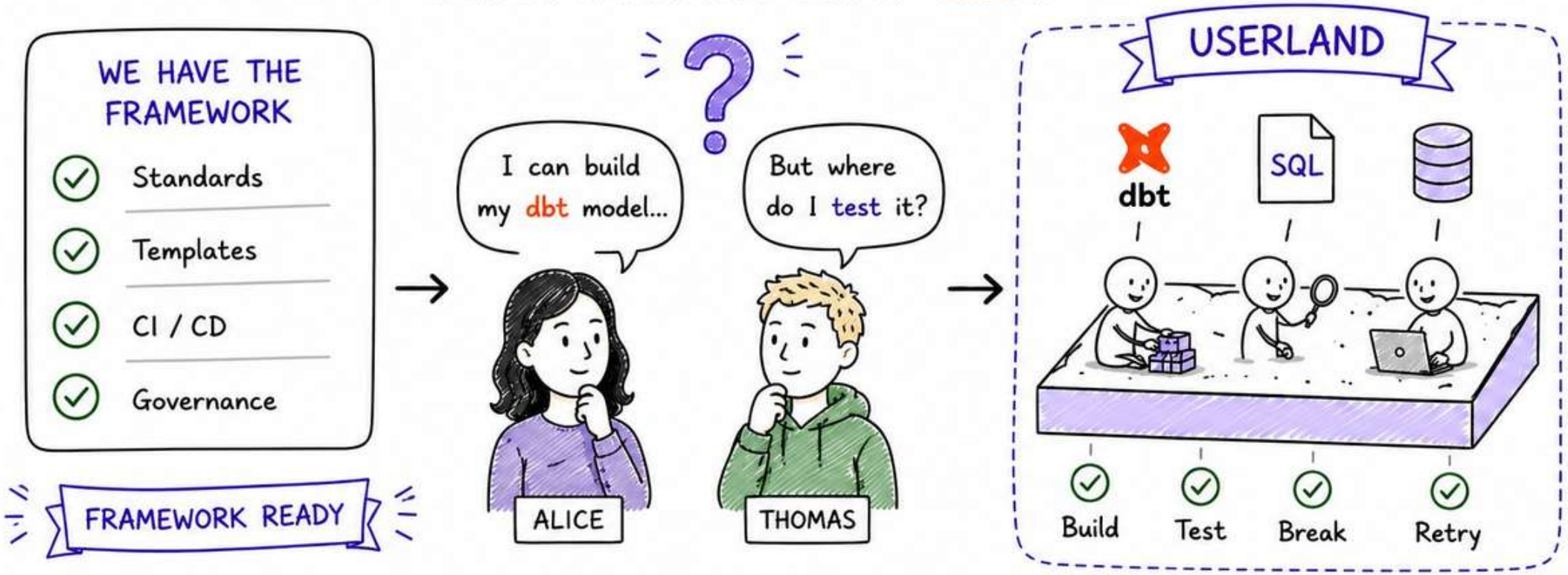
- Dev
- Staging
- Production

Build once → Use everywhere → Deliver something great

MEZZOUR ISMAIL

NOW WE HAVE THE FRAMEWORK FOR SELF DATA ANALYTICS

NOW HOW TO ALLOW PEOPLE TO TEST THEIR TRANSFORMATION USING dbt ?

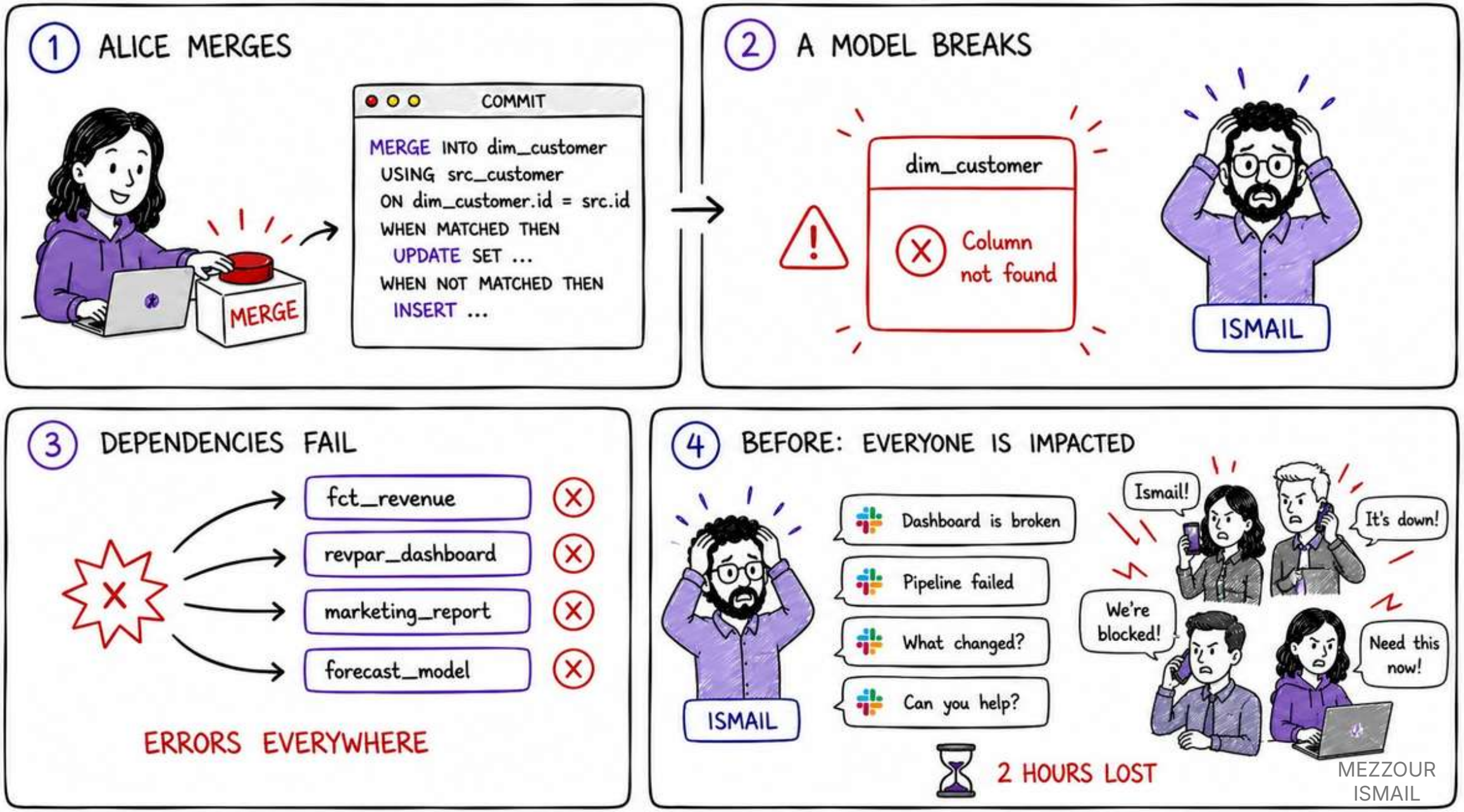


We had the framework for Self Data Analytics.

→ Now we need a safe place for people to test their transformations using dbt.

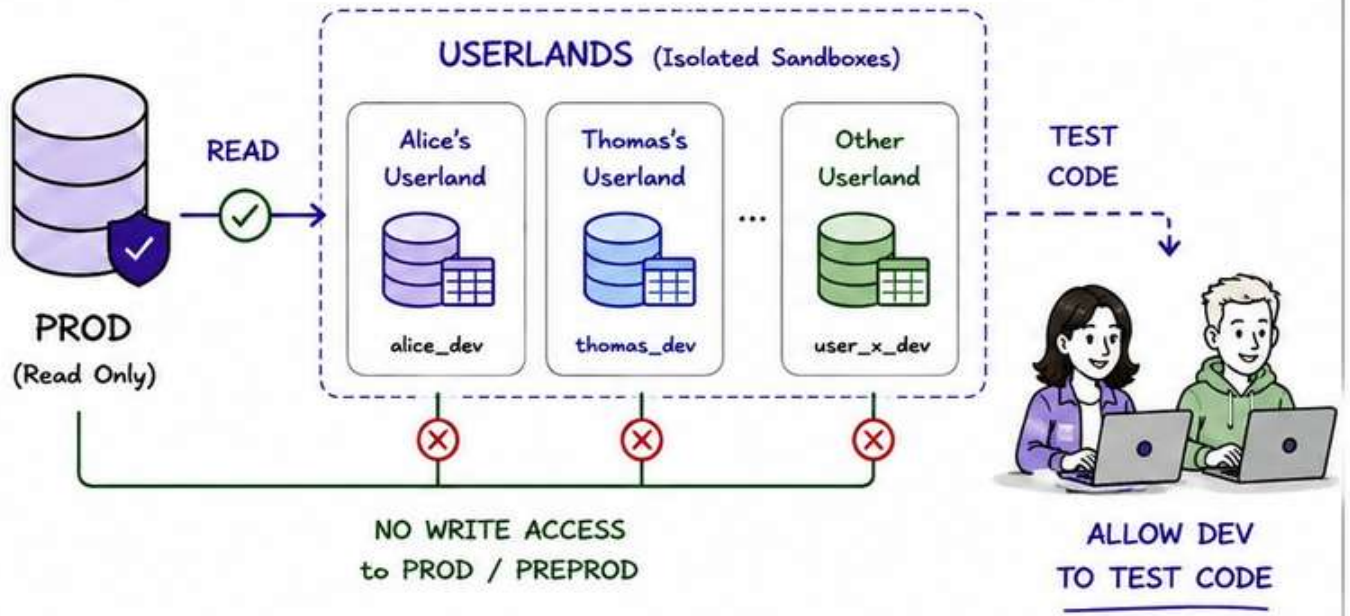
MEZZOUR
ISMAIL

Situation when everyone works on the same in dev



USERLANDS

Your personal, isolated sandbox



USERLAND RULES

READ from PROD
Use real, live production data

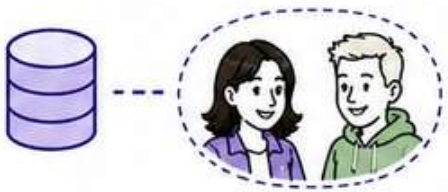
WRITE ONLY to your Userland
Write to your dedicated schema space

NO WRITE ACCESS
to Production or Pre-production

ALLOW DEV TO TEST CODE
Build, run and test safely in your Userland

Safe · Fast · Build, break, test with confidence

1 ISOLATED SANDBOX



Your own space.

Safe to build, break and test.

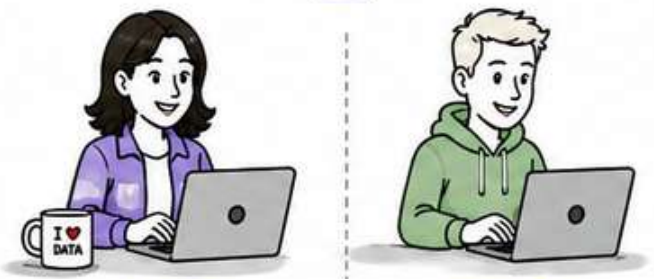
2 REAL DATA, ZERO RISK



Use live data.
No write access to PROD / PREPROD.

Allow dev to test the code.

3 CONFIDENCE & AUTONOMY



ALICE

Focus on creating business value.

THOMAS

Experiment freely. Learn fast. Ship faster.

MEZZOUR ISMAIL

OUR JOURNEY TO SELF-SERVICE DATA PLATFORM

BEFORE: NO GUIDELINES

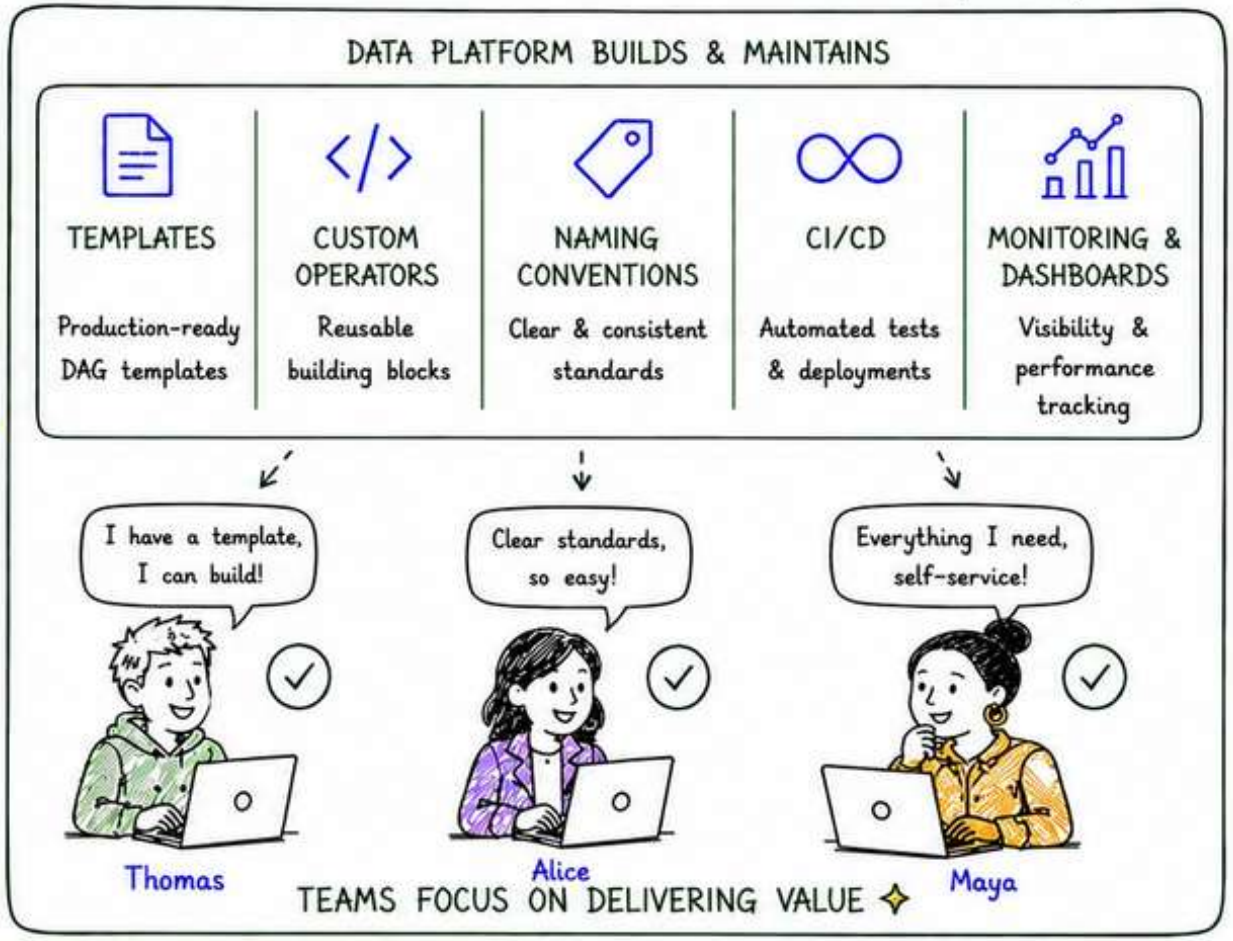
We needed Data Platform for everything



- ✕ No standards
- ✕ Reinventing the wheel
- ✕ Inconsistent solutions
- ✕ Slow delivery

NOW: SELF-SERVICE DATA PLATFORM

Data Platform sets the foundation, teams move independently



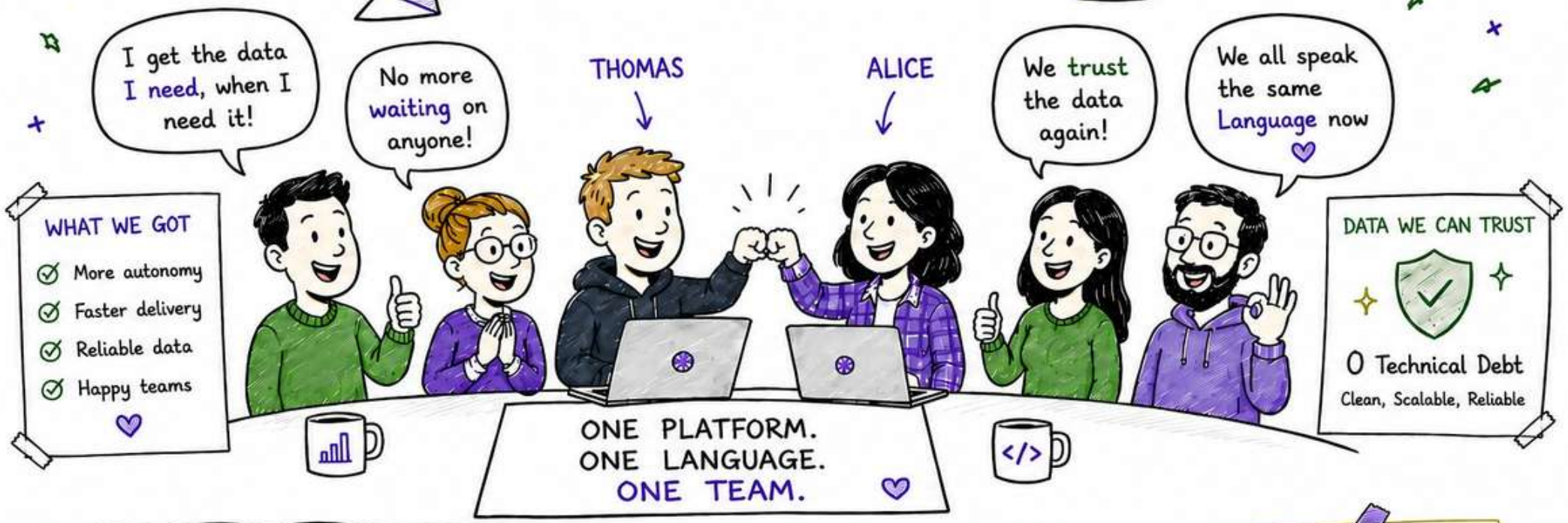
1 FOUNDATION SET
Platform provides building blocks and standards.

2 SELF-SERVICE ENABLED
Teams build independently with confidence.

3 FASTER VALUE DELIVERY
Less friction, better quality, more impact.

MEZZOUR ISMAIL

THOMAS & ALICE ARE EMPOWERED AND AUTONOMOUS!



We built so much.
What do we build **next?**



ML PATTERN

Build, train and serve ML models directly where the data lives.

TALK TO YOUR DATA

Ask questions. Get answers. Instantly.



THE BEST PART?

- More autonomy for teams
- Faster delivery
- Reliable and trusted data
- 0 technical debt
- Clean, scalable and reliable
- One platform, one language
- Happy teams



MEZZOUR
ISMAIL