

Building your first data product with Fabric.

> one whatsapp chat. two sleep-deprived parents. one baby.

> end-to-end with **microsoft fabric**: lakehouse → notebooks → reports.

Christopher MANEU

data & cloud advocate @ microsoft

\$ whoami

> Christopher MANEU

> data & cloud advocate at microsoft.

> based in france. writes in fr/en.

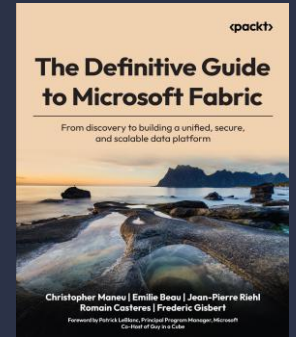
> builds demos, talks, books,

> and small tools.

BOOK · PACKT 2025

The Definitive Guide to Microsoft Fabric

from your first workspace to a secure,
scalable analytics estate.



FORTHCOMING 2027

Platform Engineering for Modern Data Platforms

governed, ai-native data platforms
with microsoft fabric.

WE NEED DATA PRODUCTS, EVEN ON MARS

```
cat Alexandra-holloway.txt
```

Some of the things that I wish we had now on [the Mars Science Laboratory] include a better understanding of where our power is going. I want to see how much power each component is drawing every minute, so that we could architect a software system that could balance loads better. [...]. **So if I were building a mission, I would have those discussions earlier and get operators into the room to say, "what do you want your data products to look like?"**



[The Ingenious Fixes Keeping the Curiosity Rover Rolling - IEEE Spectrum](#)

THE PROBLEM

> a newborn arrives.

> you suddenly need answers:

> how long did she sleep today?

> when was the last feed?

> is that fever trending up?

> did she sit up longer than yesterday?

> but all the data lives in...

> a whatsapp chat with my wife.

unstructured. bilingual. typo-rich. 100% real.



01. what is a data product?

> the data mesh definition, minus the buzzwords.

Data Product



A reusable, trusted, and discoverable data asset built for a specific business use case.



☆ A data product is not just data — it is data packaged for use. ☆

Fabric Notes

What is a data product?

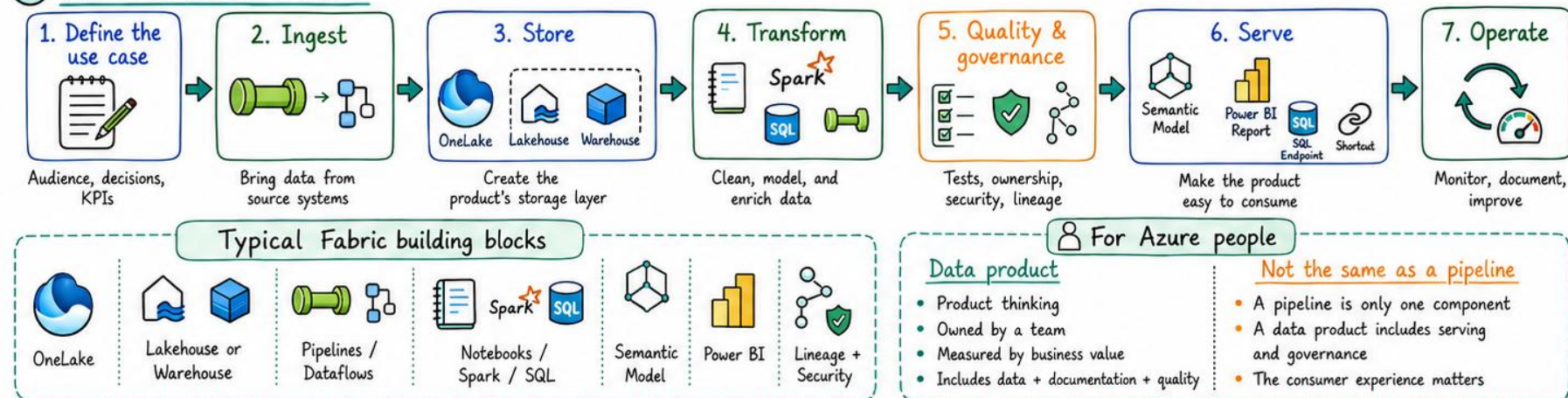
And how to build one in Microsoft Fabric

1 What is a data product?



☆ A data product is not just data — it is data packaged for use. ☆

2 How to build one in Fabric



☆ In short: a data product in Fabric combines data, ownership, quality, and consumption into something useful for the business.



\$ data-product --define

"the smallest unit of architecture that can be independently deployed and managed."

a node on the mesh ships three things:

code/

pipelines, parsing logic, tests, access policies

data+metadata/

delta tables, schema, lineage, quality metrics

infrastructure/

the storage & compute to build and serve it

diff dataset data-product

- a file someone dropped somewhere
- no owner, no docs, no promises
- you reverse-engineer the schema
- + input & output ports, documented
- + an owner accountable for quality
- + discoverable, with SLOs & contract

A DATA PRODUCT MUST BE...

[x] discoverable

consumers can find it in a catalog –
owner, lineage, samples.

[x] addressable

a stable, unique address.
programmatic access by convention.

[x] trustworthy

quality is measured & published. no
more "is this fresh?"

[x] self-describing

schema + semantics travel with the
data. no tribal knowledge.

[x] interoperable

open formats & global standards so
products compose.

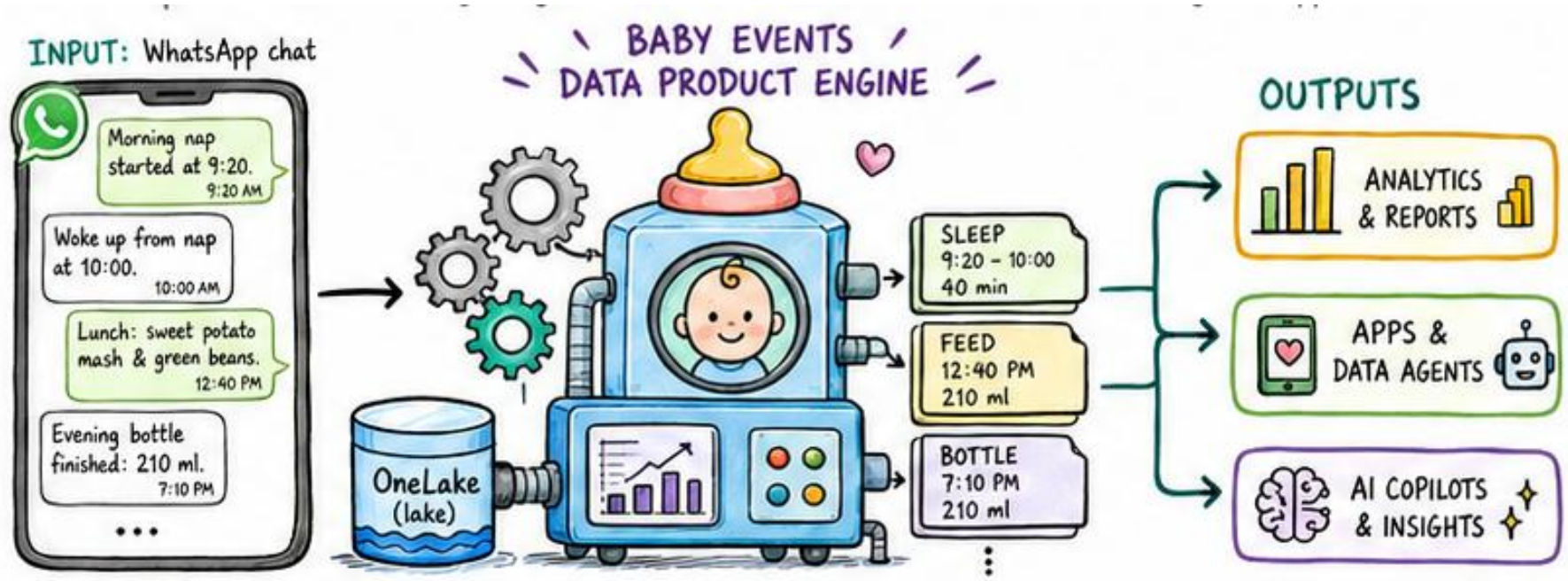
[x] secure

access governed by policy, enforced
automatically.

keep these six. we'll map each one to a fabric feature at the end.

02. designing the baby tracker

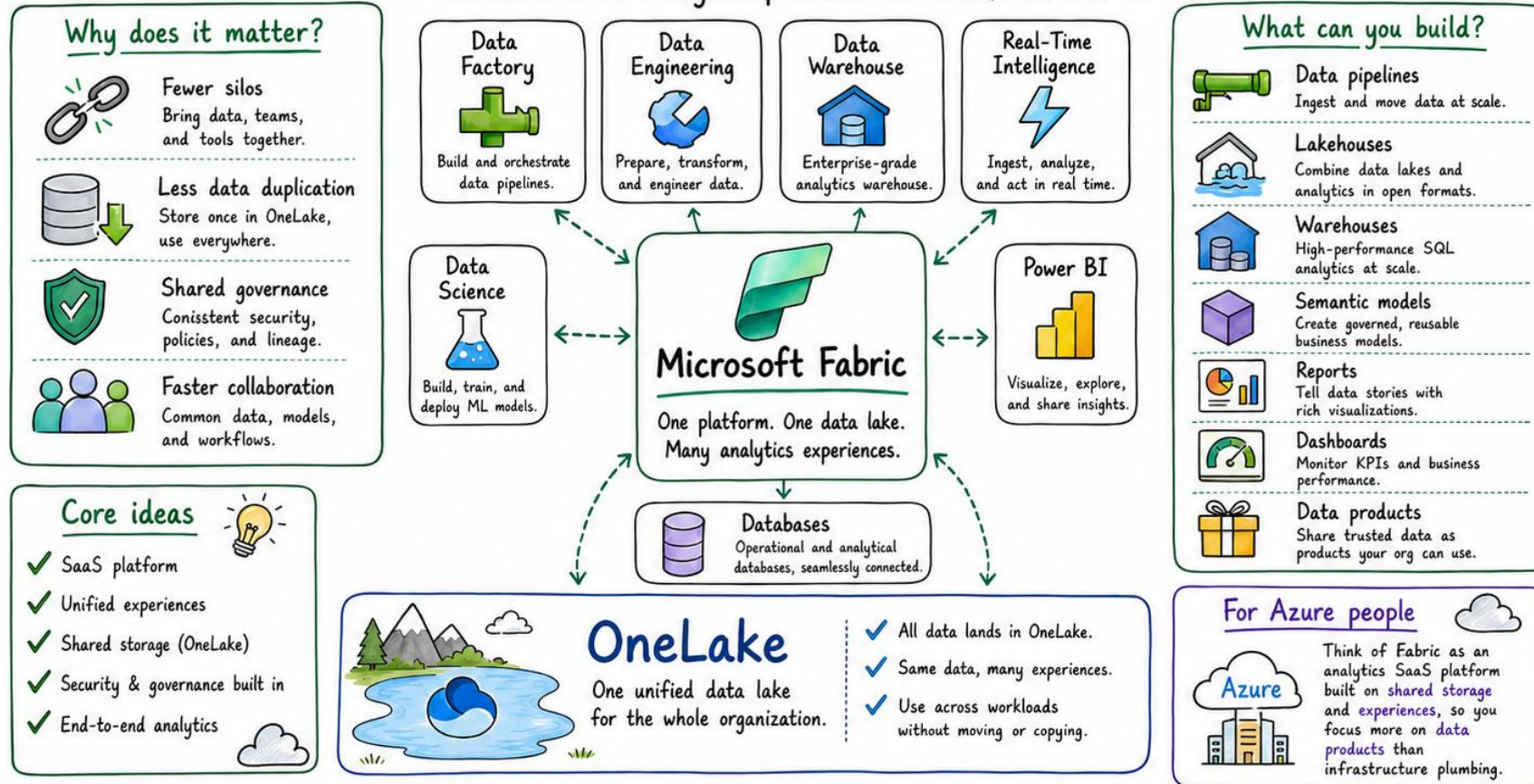
> applying product thinking to 3am feeds.



Fabric Notes

What is Microsoft Fabric?

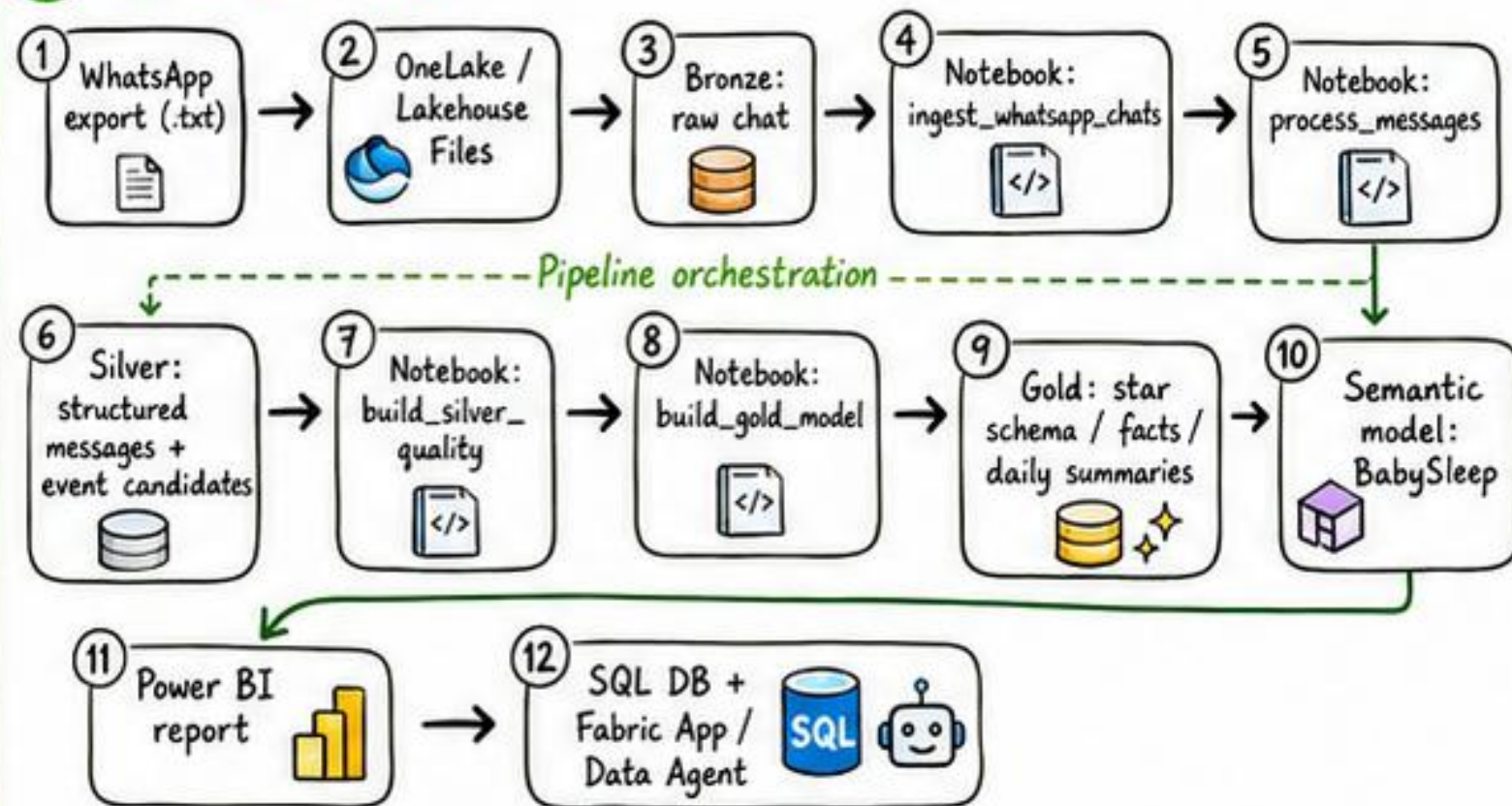
A unified SaaS analytics platform for data, AI, and BI



In short: Microsoft Fabric brings ingestion, engineering, warehousing, real-time analytics, AI, and BI together on **OneLake**.



3 How it works



Fabric Notes

OneLake: One data lake for everything in Fabric

OneLake is the unified, secure, and governed data lake for your entire organization.

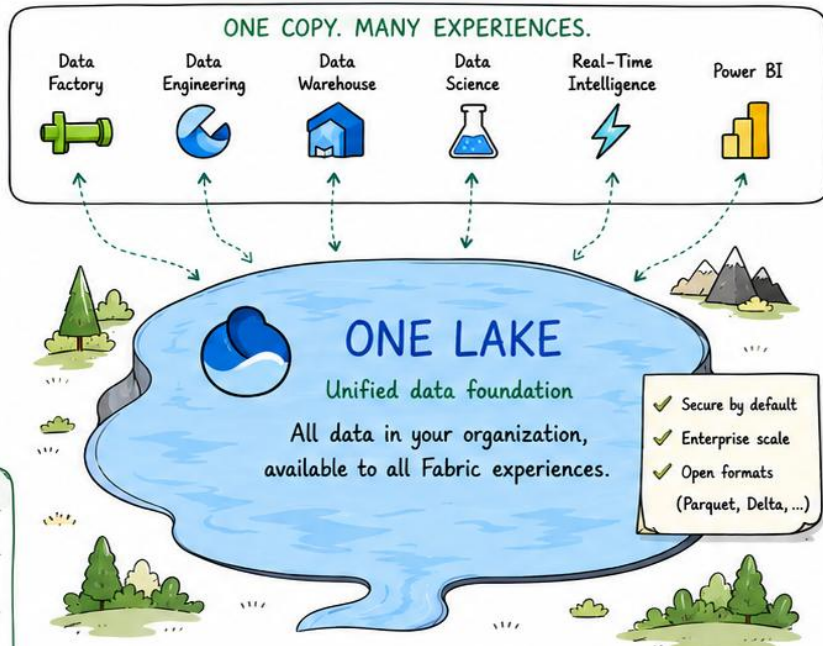
WHAT IS ONELAKE?

-  A single logical data lake built on Azure storage.
-  Data is stored once, used by many Fabric experiences.
-  One security model, consistent governance, end-to-end.
-  Accessible from any Fabric workspace in your tenant.

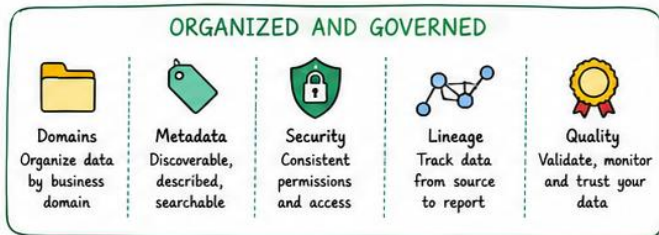
SHORTCUT vs MIRRORING

Feature	Shortcut	Mirroring
Data location	Stays in source	Copied to OneLake
Data movement	None	One-time + continuous
Cost	Only source storage	Source + OneLake storage
Performance	Depends on source	Optimized in OneLake
Use case	Access in place, data virtualization	Analytics, reporting, AI/ML, reliability

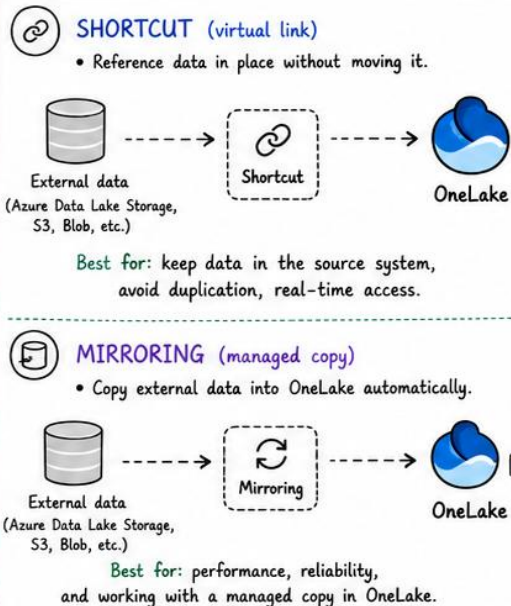
ONE COPY. MANY EXPERIENCES.



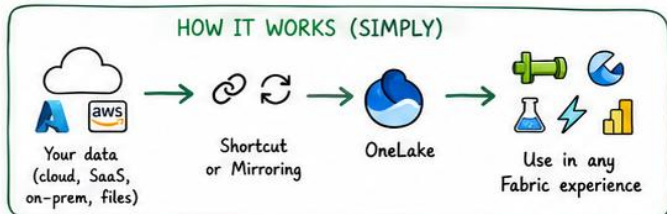
ORGANIZED AND GOVERNED



TWO WAYS TO USE EXISTING DATA



HOW IT WORKS (SIMPLY)



Key idea

OneLake eliminates data silos and copies. Access data where it lives with **Shortcuts**, or bring it into OneLake with **Mirroring**. Then build, analyze and share with any Fabric experience.

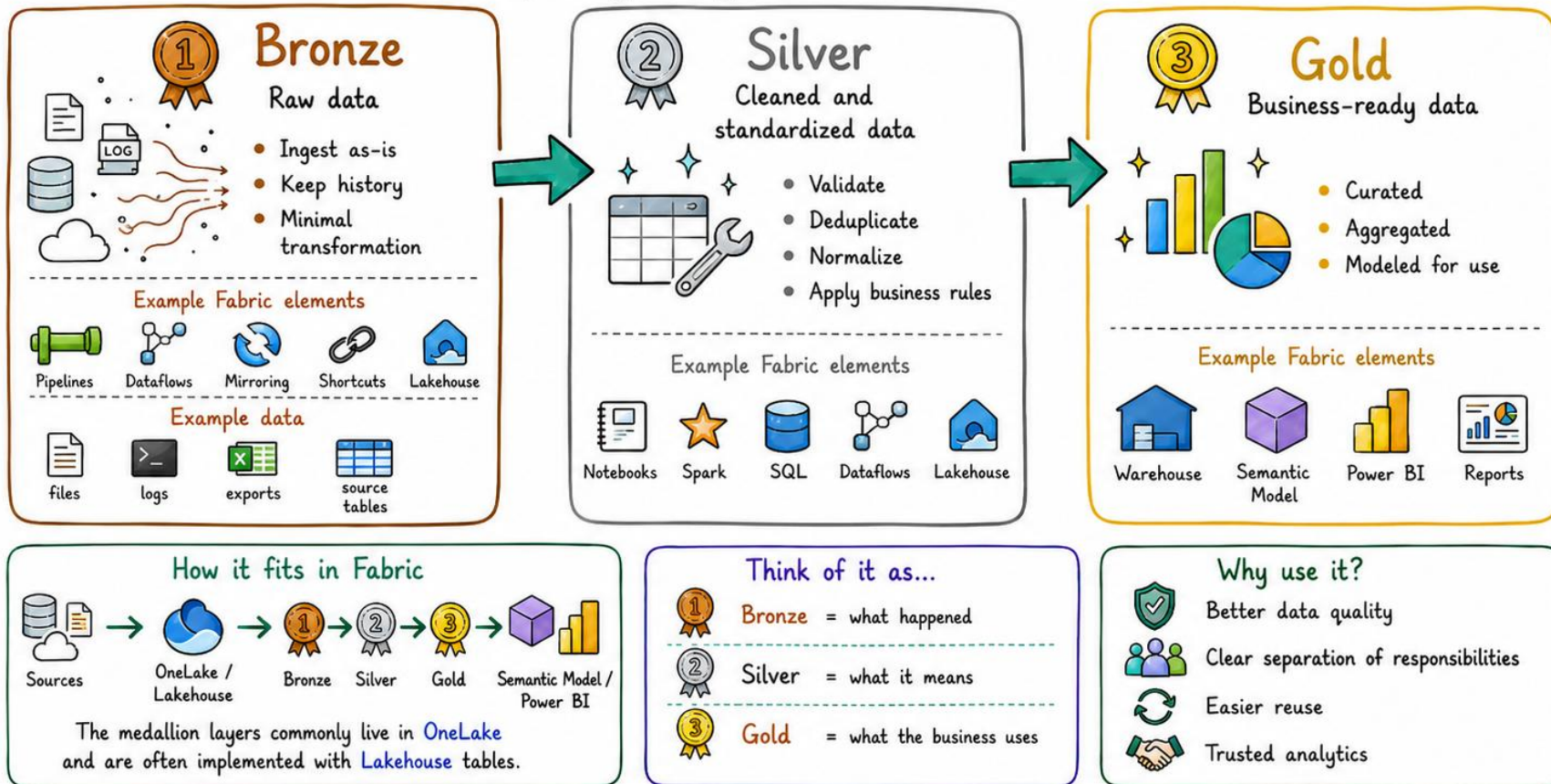
In short

OneLake is the foundation that makes your data simple to access, govern, and trust.



What is the medallion architecture?

A simple way to organize data in Microsoft Fabric



In short: medallion architecture organizes data from raw to refined so teams can build trusted analytics products in **Fabric**.



How do you turn a WhatsApp baby chat into a data product with Microsoft Fabric?

A data product turns messy daily events into trusted, reusable data for analytics, apps, and AI.

1 What it is

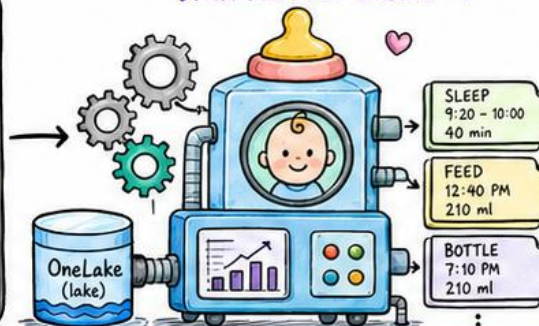
- A domain-owned baby routine data product
- Source: shared WhatsApp export between parents
- Main product = trusted baby events timeline
- Not just a dashboard: it is reusable data + logic + quality + access
- Common event types: sleep, wake, feeding, temperature, sitting



INPUT: WhatsApp chat



BABY EVENTS
DATA PRODUCT ENGINE



OUTPUTS



ANALYTICS
& REPORTS



APPS &
DATA AGENTS

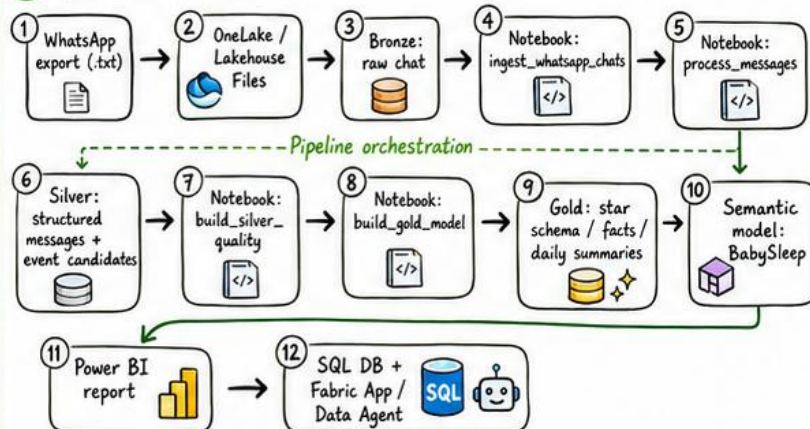


AI COPILOTS
& INSIGHTS

2 Why it matters

- 🎯 Turns messy text into structured insight
- ❤️ Tracks routines, health signals, and milestones
- 📊 Reusable by reports, apps, and copilots
- 👤 Fits data mesh thinking: domain ownership + data as a product

3 How it works



4 Key Fabric components

- OneLake – one data foundation
- Lakehouse (babylakehouse) – bronze/silver/gold storage
- Notebooks – parsing, classification, quality, modeling
- Pipelines – orchestrate the steps
- Semantic model (BabySleep) – business-ready analytics layer
- Power BI – reports and dashboards
- SQL Database – write-back / app scenarios
- Fabric App / Data Agent – operational & conversational experiences
- Git integration – version the project in GitHub
- Governance & lineage – know where data came from

5 Common use cases

- 🌙 Daily sleep analysis
- 🍼 Feeding frequency and bottle/breast trends
- 🌡️ Temperature monitoring
- ★ Milestone tracking: sitting, wake patterns
- 📊 Parent-facing dashboard
- ✍️ Data app for corrections and annotations
- 🤖 Copilot / agent answers: "How much did the baby sleep yesterday?"

6 Things to remember

- ✓ Start simple: deterministic parsing first
- ✓ Keep the raw export in Bronze
- ✓ Use Silver for structured events and quality checks
- ✓ Gold is business-ready and reusable
- ✓ Document ownership, freshness, and trust
- ✓ Protect personal and health-related data

For beginners

- 📊 Dashboard = one way to view data
- ↓
- 🗄️ Data product = trusted dataset + logic + quality + access
- 📱 App consumes the data product

Common mistake

- ⚠️ The dashboard is NOT the data product.
- ✗ The real product is the trusted baby events dataset that many experiences can reuse.

For Azure people

- Think of it like: source + platform + contract + consumers
- 🧠 Closest mental model: a domain-owned analytics service

Security

- 🛡️ Protect the data & control access

Lineage

- 📊 Track data flow end to end

Source control

- 🌐 Version everything in GitHub

Medallion architecture

- 🔄 Bronze → Silver → Gold
- 🔄 Raw → Structured → Trusted



In short:

A data product is a trusted, reusable baby events dataset that powers reports, apps and AI experiences.



\$ python parse_chat.py

```
notebook · cell 3

pattern = r"\[(\d+/\d+), (\d+:\d+)\]
          (.*?) : (.*?) "

EVENTS = {
    "tétée|biberon": "feed",
    "dodo":          "sleep",
    "réveil":        "wake",
    "°c":            "temperature",
    "assise":        "sitting" }
```

bronze/

raw chat lines, untouched. replayable forever.



silver/

parsed events: ts · author · type · value · unit



gold/

daily aggregates: sleep hrs, feeds, temp max

> rules first. when rules run out: **ai functions (ai.classify)** on the weird messages.

> emoji-only messages ("❤️❤️❤️") → dropped. sentiment is out of scope. for now.

▶ demo 02 — transform

● ● ● fabric demo --step transform

> run the parsing notebook → silver & gold delta tables

> inspect tables in the lakehouse explorer + sql endpoint

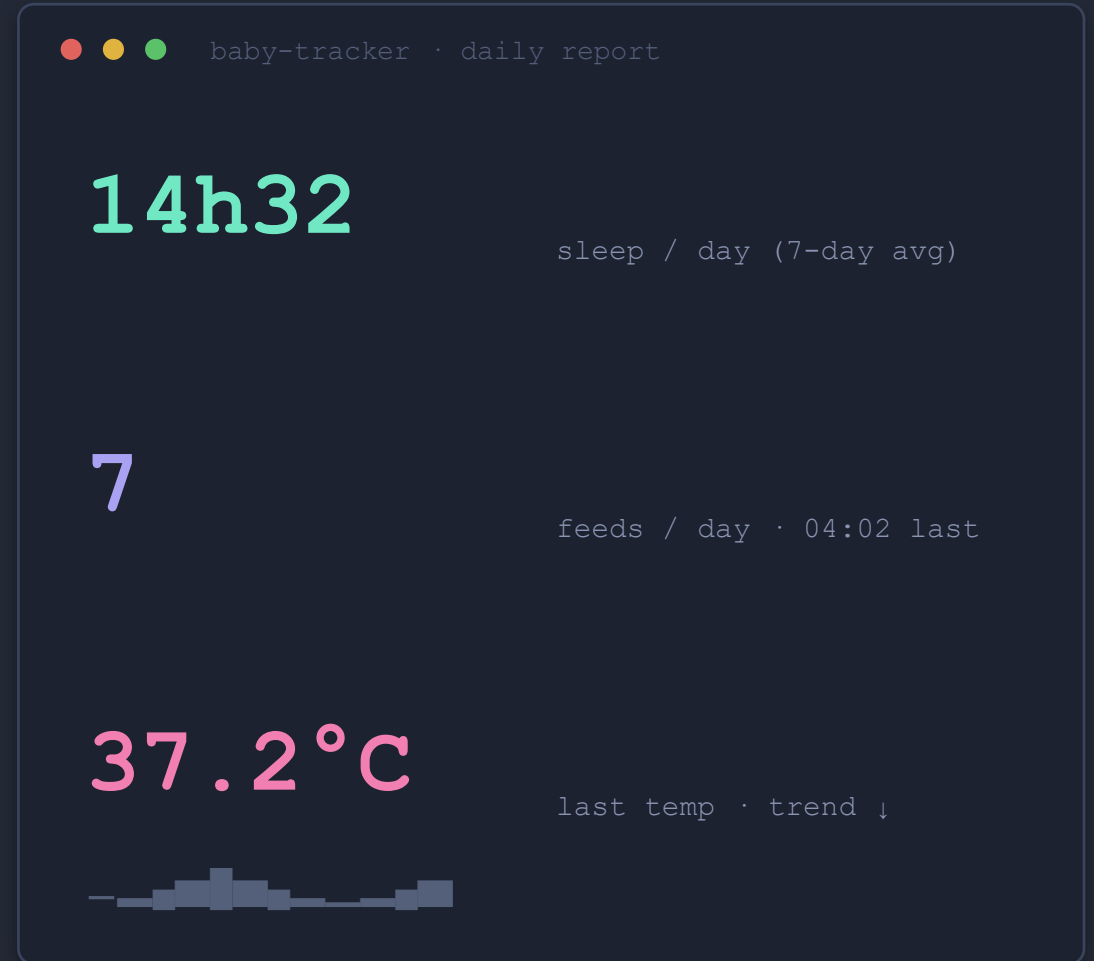
> wire a data pipeline: upload triggers parse, on schedule



[switching to fabric ui — the deck will wait.]

SERVE — FROM TABLES TO ANSWERS

- > gold tables → **semantic model**
- > direct lake: reads delta straight
- > from onelake. no import, no copy.
- > one **power bi report** for both parents:
- > sleep heatmap · feed cadence ·
- > temperature timeline · milestones
- > *the pediatrician gets a link, not a csv.*



SHIPPING IT AS A *PRODUCT*

the six characteristics, mapped to fabric:

[x] discoverable	onelake catalog + item descriptions + endorsement (promoted/certified)
[x] addressable	stable workspace & item names · abfss:// paths · sql endpoint
[x] trustworthy	quality checks in the notebook · refresh slo · activator alerts on failure
[x] self-describing	semantic model metadata · data contract in the repo · column descriptions
[x] interoperable	delta + parquet everywhere · shortcuts instead of copies
[x] secure	workspace roles · item permissions · sensitivity labels (it's my kid's data)

same checklist scales from one baby to one enterprise. that's the point.

\$ exit 0

> a data product is a promise, not a dataset. owner, ports, slos.

> fabric gives every port one home: onelake. one copy, open delta.

> start embarrassingly small. a whatsapp chat is a real domain.

links

blog + slides → cmaneu.net

the book → [definitive guide to microsoft fabric](#)

code → github.com/cmaneu/fabric-baby-tracker

merci.

questions? →