Modeling the Mesh

Understanding data products and domains

Juha Korpela – Datakor Consulting

About me

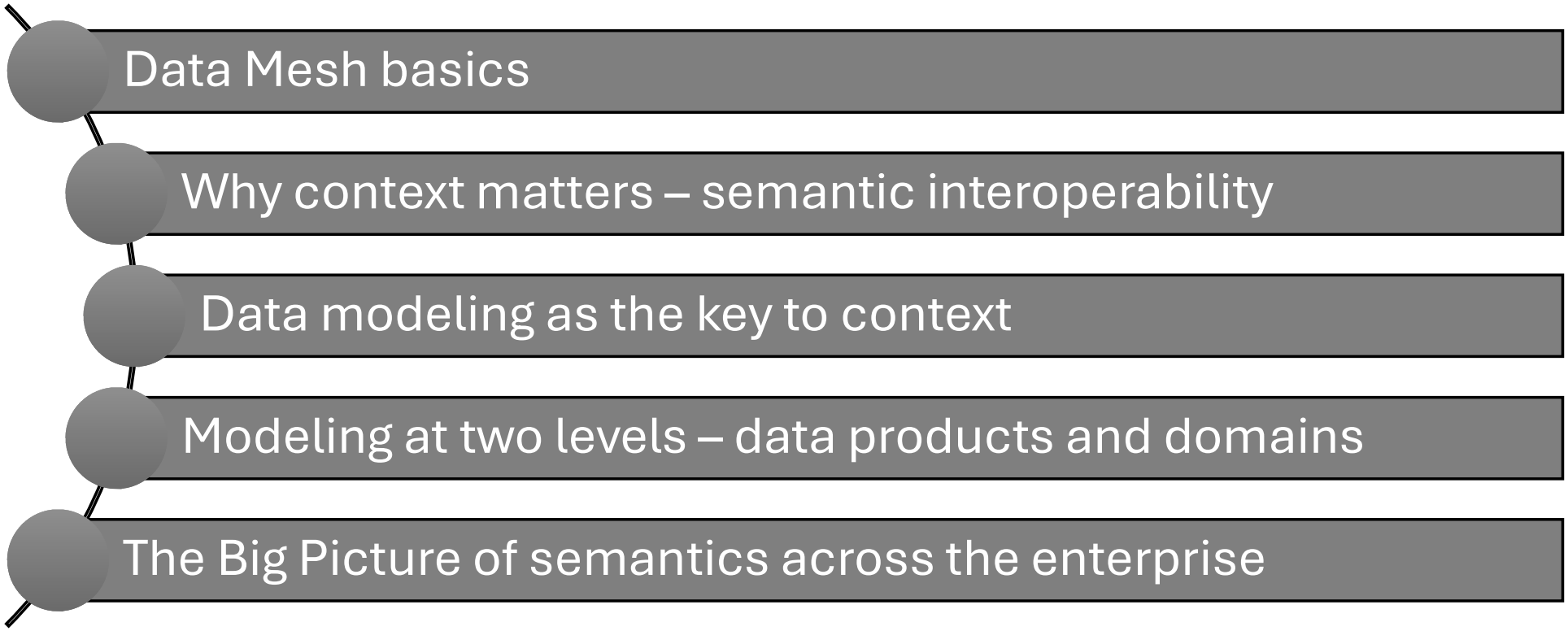
Juha Korpela

- Based in Helsinki, Finland
- 15+ years in Data Management in a variety of industries with diverse expertise:
  - Data Products
  - Data Governance
  - Data Modeling
  - Data Engineering
  - Data Strategy
- Entrepreneur & Consultant @ Datakor Consulting
- Founder & Chairman @ Helsinki Data Week
- Podcast host @ Helsinki Data Mafia
- Speaker: Data Day Texas, IRM UK Enterprise Data, Data Mesh Live...
- ex-CPO @ Ellie Technologies Inc.
- ex-Head of Data Platform @ UPM-Kymmene Oyj

DATAKOR CONSULTING

Follow me on LinkedIn:
linkedin.com/in/jkorpela

Agenda

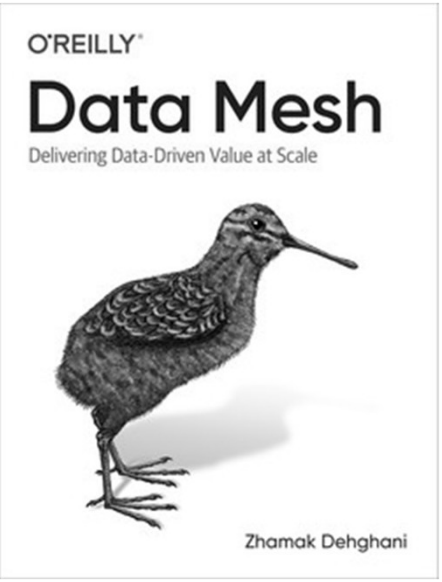

Data Mesh basics
Why context matters – semantic interoperability
Data modeling as the key to context
Modeling at two levels – data products and domains
The Big Picture of semantics across the enterprise

Data Mesh basics

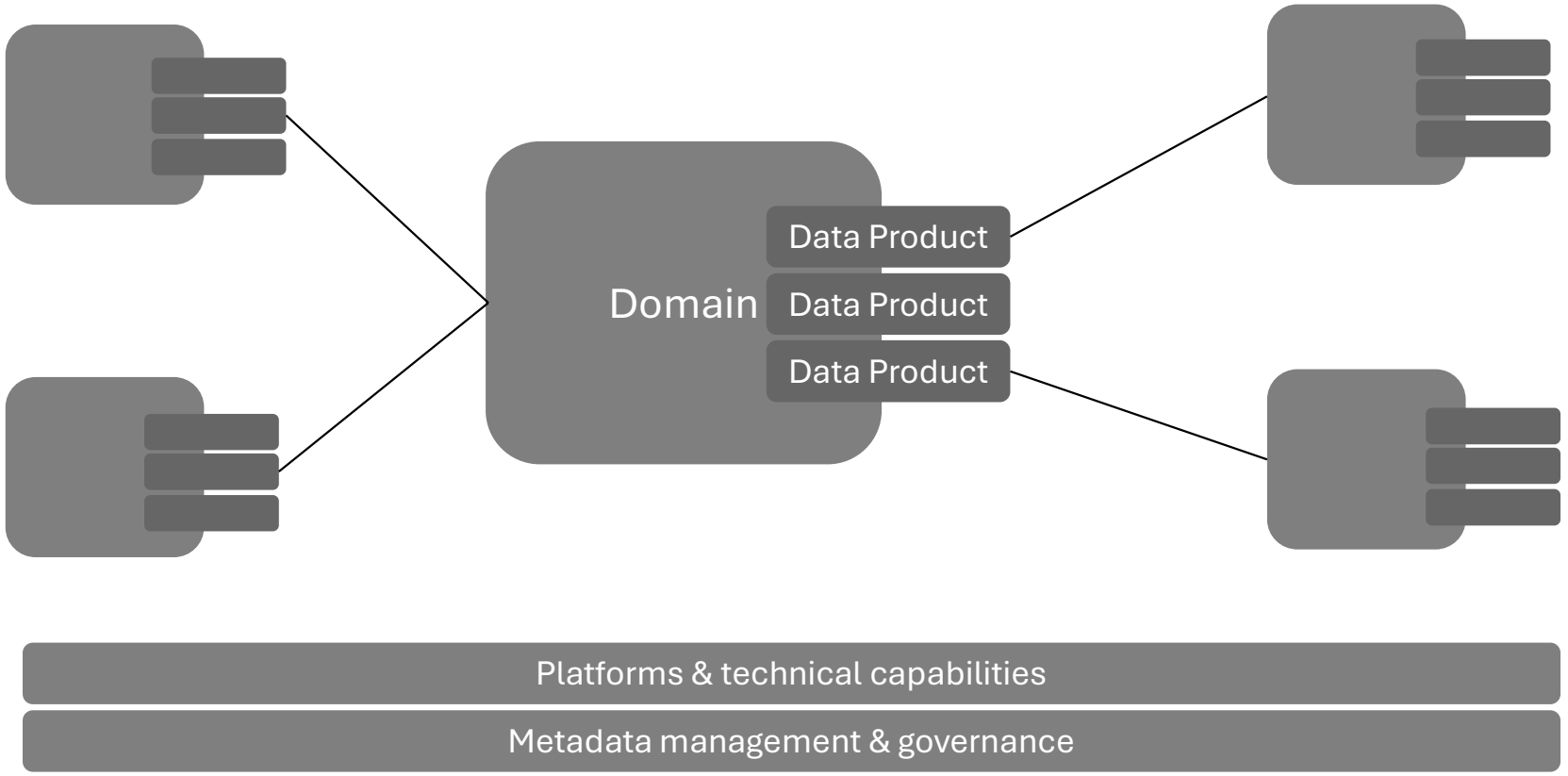
The paradigm, its principles, and how it works in reality

Data Mesh: background

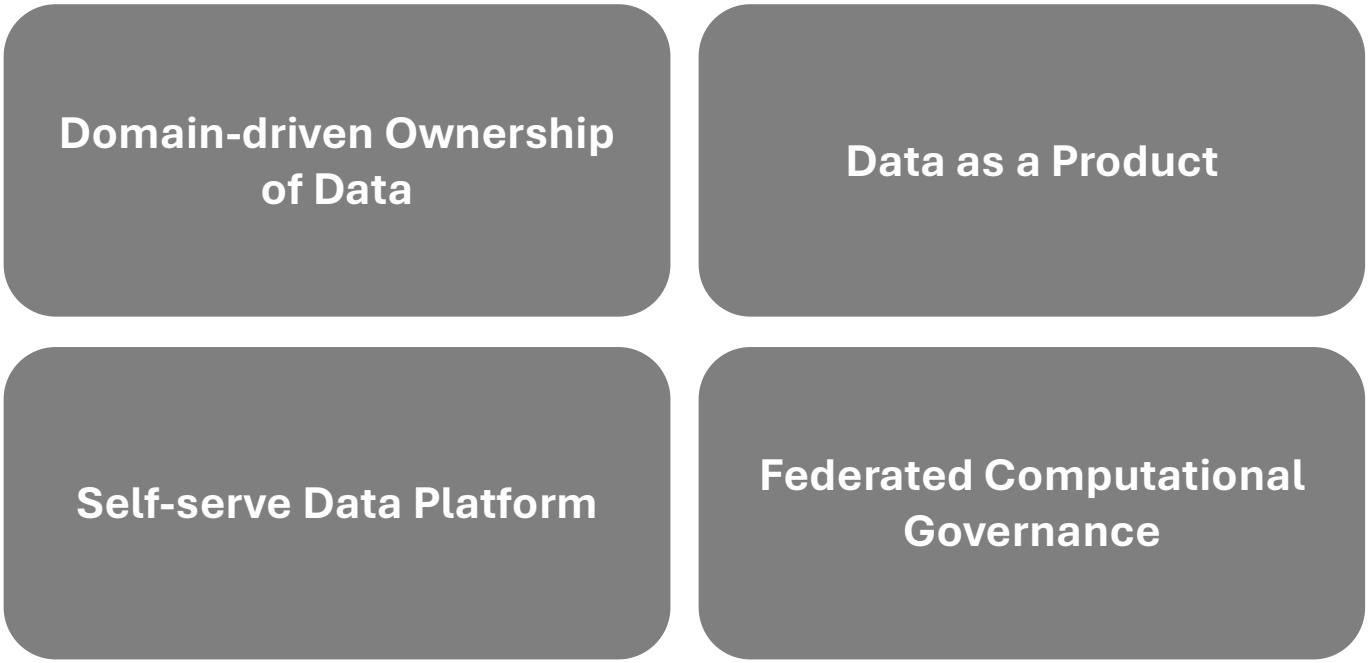
- A framework developed by **Zhamak Dehghani**
- **Domain-oriented**, decentralized model
- Domain teams are responsible for all the data within a domain & distributing it outside the domain
- All data distribution & consumption done through **data products**
- Domains and their data products form a **mesh**
- The Data Mesh is **supported by shared capabilities**: technology platforms + governance model



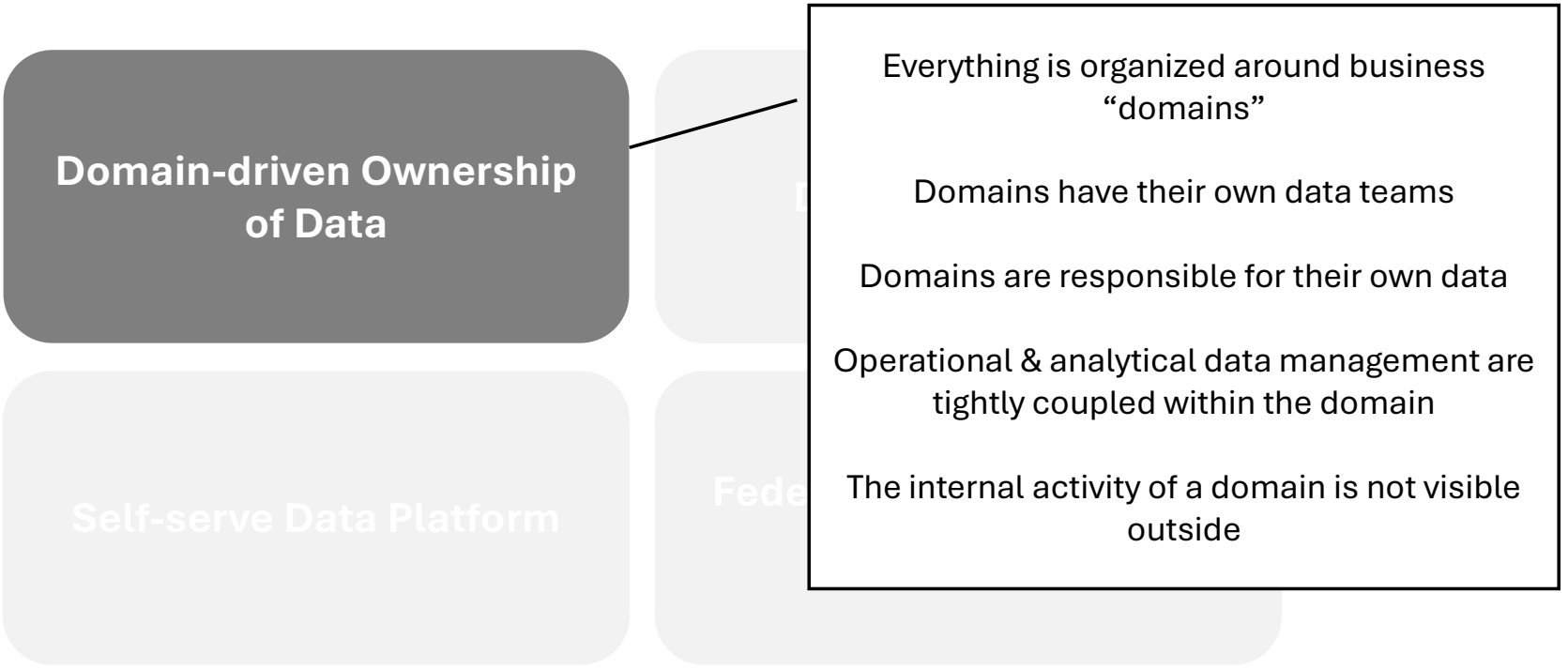
Data Mesh in a nutshell



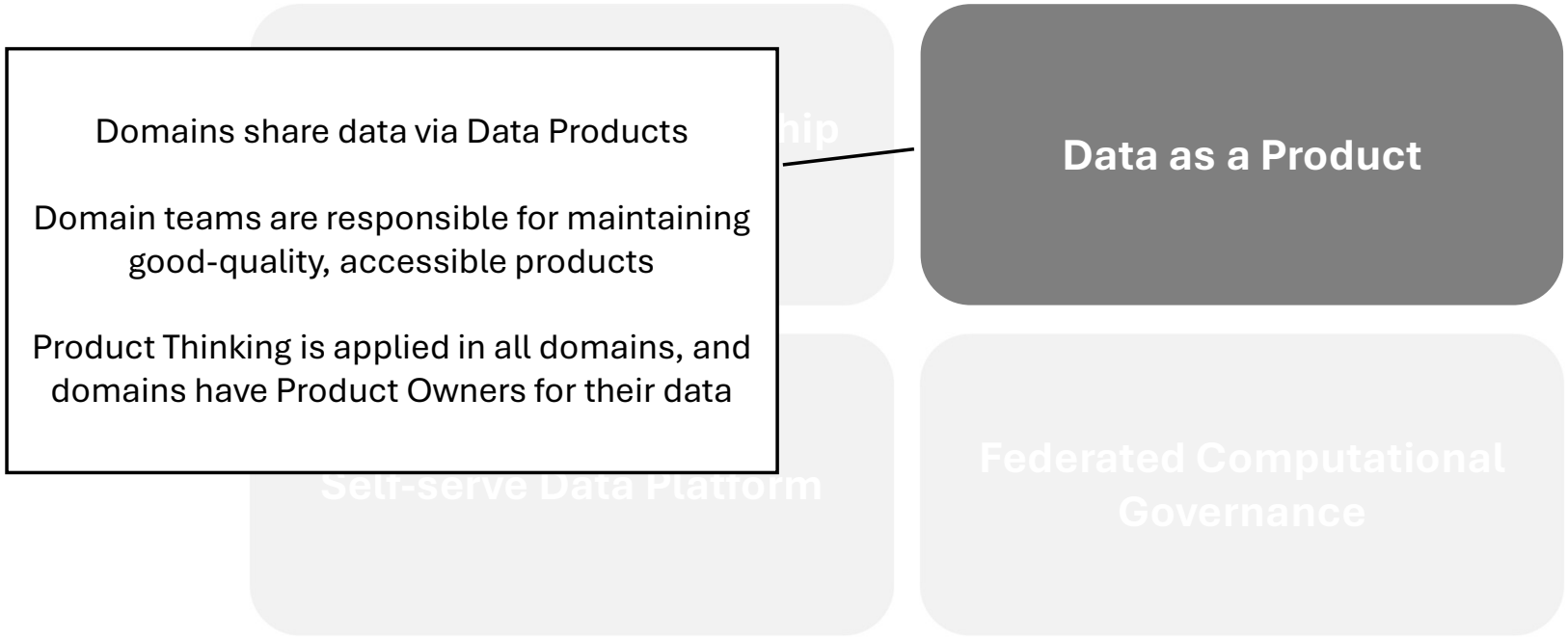
Four principles of Data Mesh



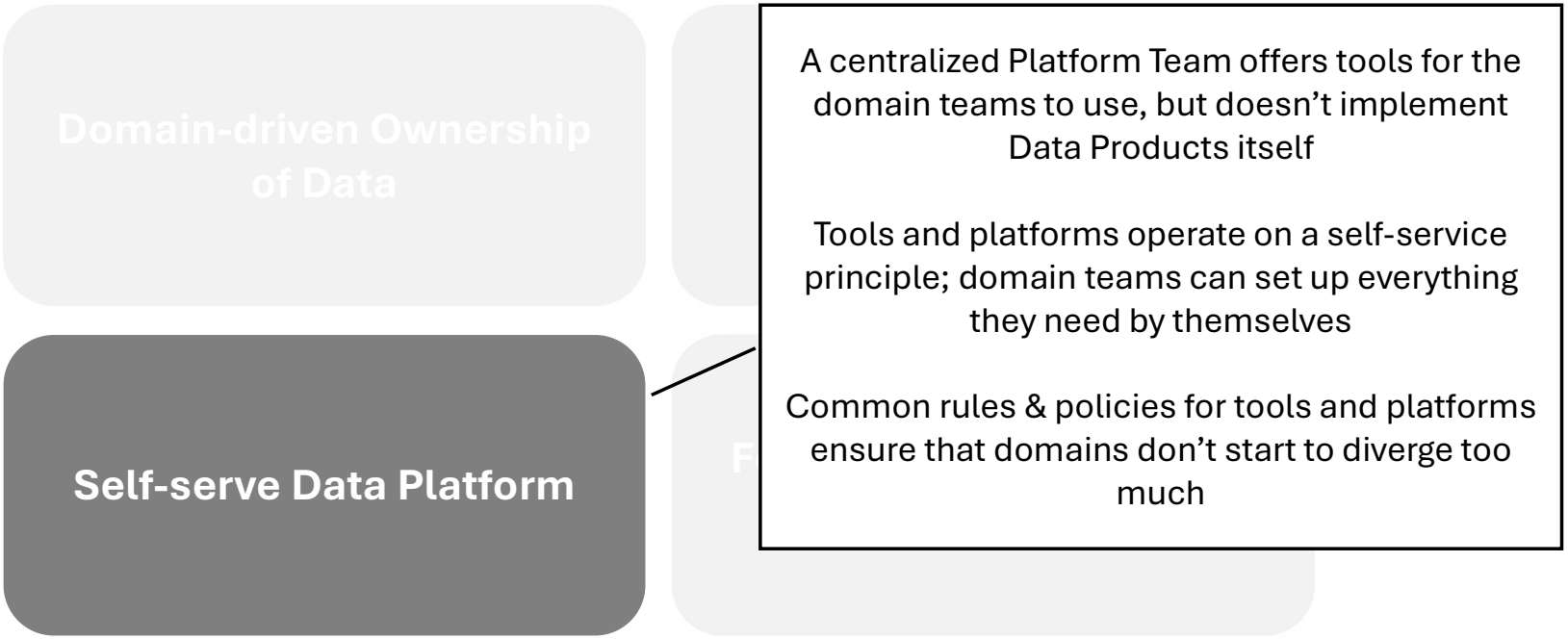
Four principles of Data Mesh



Four principles of Data Mesh



Four principles of Data Mesh

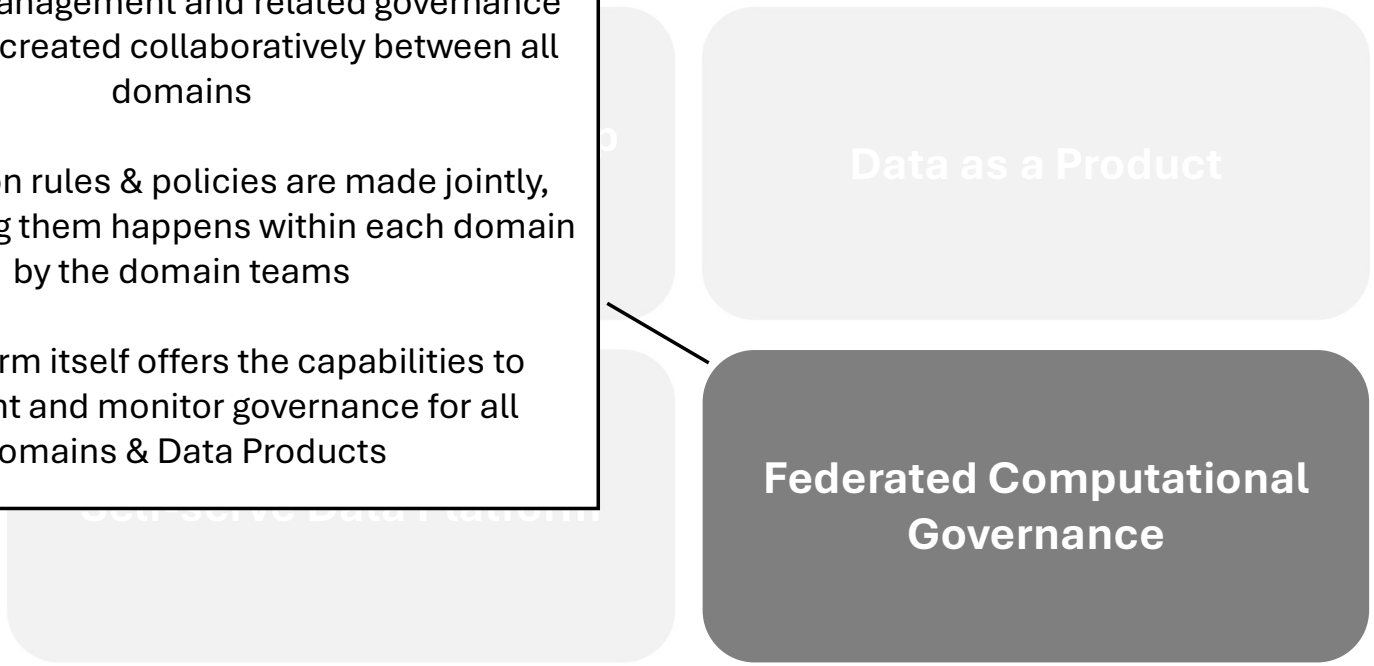


Four principles of Data Mesh

Metadata management and related governance policies are created collaboratively between all domains

Decisions on rules & policies are made jointly, implementing them happens within each domain by the domain teams

The platform itself offers the capabilities to implement and monitor governance for all domains & Data Products



Data Mesh in reality

- 100% pure Data Mesh implementation is **rare**
- Significant technology effort
- **Massive** organizational effort
- Benefits from **scale** - example organizations: PayPal, Spotify, Roche...

HOWEVER...

- Real-life benefits in **shifting data responsibilities closer to value-creation**
- **"Data as a Product" –thinking** valuable for everyone
- Separating **Platform teams** (building capabilities) and **Domain teams** (building solutions) clarifies roles and boundaries

➢ ***If you want to do decentralized data products, you will end up scaling to Data Mesh!***

Product thinking: from projects to products

Demand-driven value delivery

Build for identified use cases

Know your users

Explicit UX design

Product ownership

Governance for products

Lifecycle management

Prioritization by owner

Loosely coupled architecture

Clear boundaries

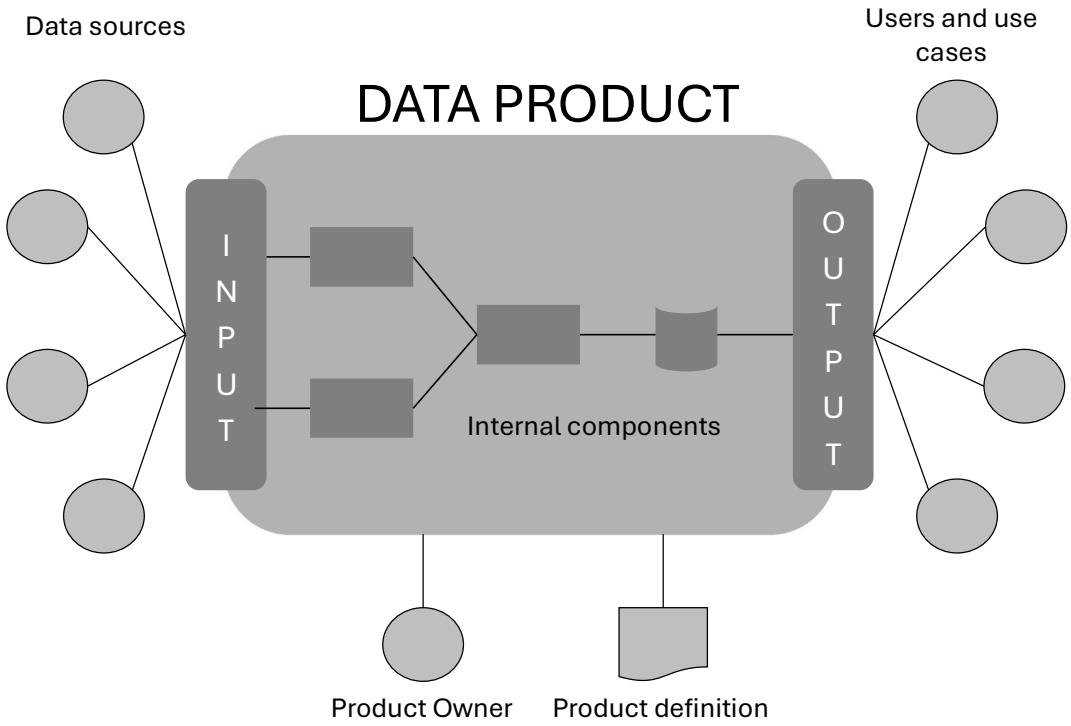
Shared platforms

Independent development

What is a “Data Product”? My definition!

A “packaged” set of data and components, that has:

- Clear boundaries
- One owner
- A lifecycle
- Identified users and benefits
- Input and output interfaces
- Understandable definition
- “Data and code”!



Why context matters

Semantic interoperability of Data Products & Domains

Data

”Productized” data

Product definition

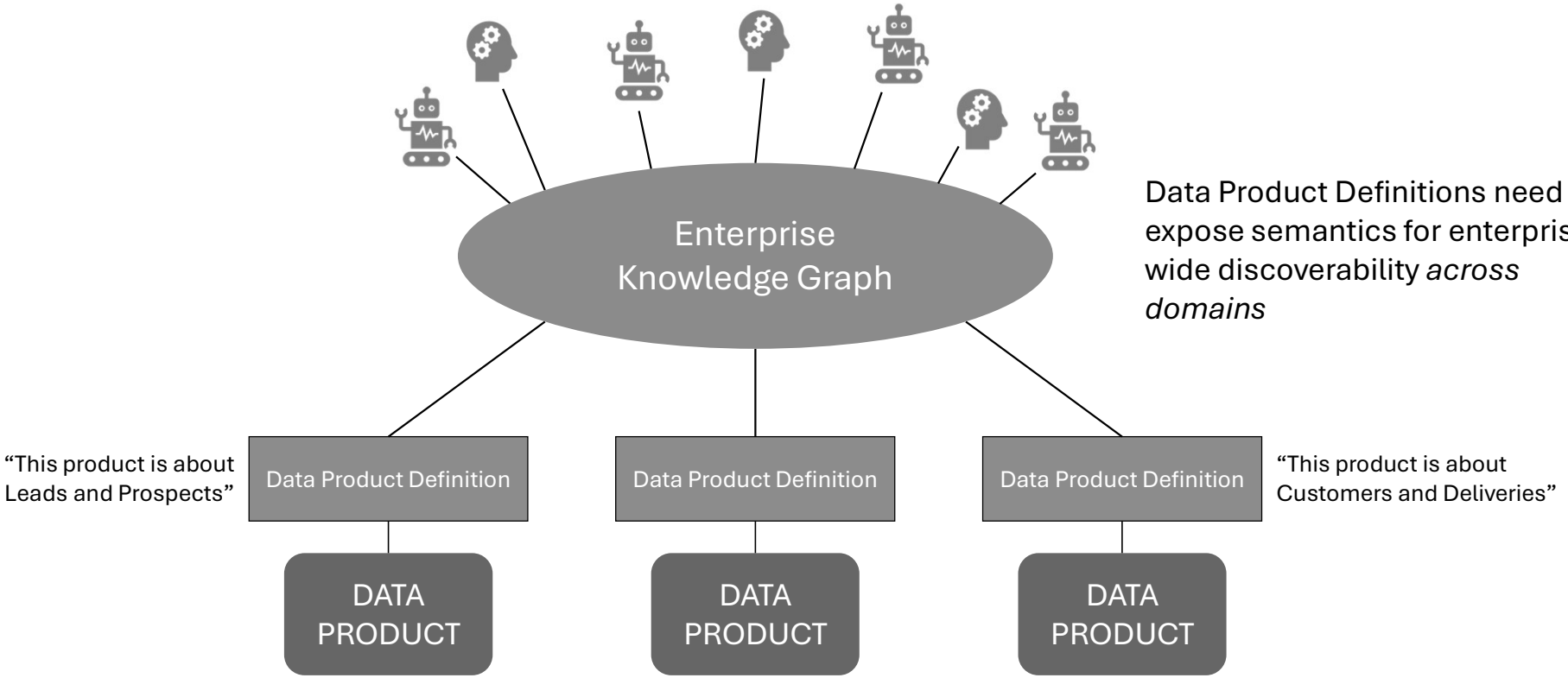
- Contents of the product: raw materials, allergens, production method…
- Instructions & policies: what can be done with the product and how (microwave & eat!)
- Warnings & limitations: how the product is NOT to be used
- Technical information: size, structure, packaging material, recycling…
- Producer and their contact details

Data Product metadata

What information should we have in the “wrapper” for a Data Product?

- **Contents**: **semantic description of the data**
- **Instructions & policies**: what are the intended use cases, how to get access
- **Warnings & limitations**: infosec, legal limitations (e.g. GDPR), deployment status (in production or just for testing?)
- **Technical information**: database objects, files, pipelines, authentication methods
- **Producer**: who owns the Data Product, who to ask for help, who is responsible for operations

The need for semantics

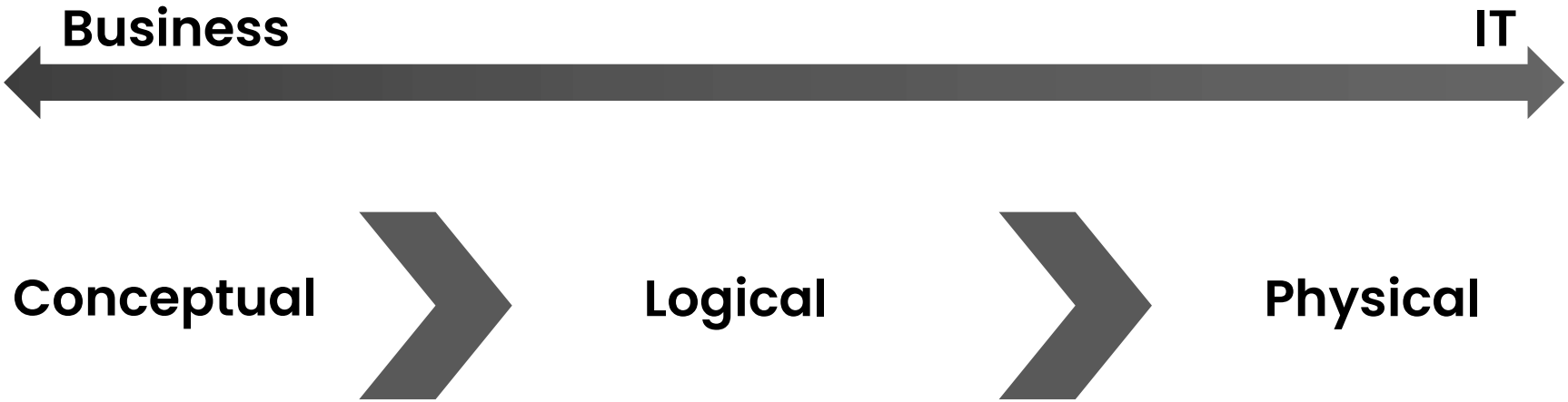


Data Product Definitions need to expose semantics for enterprise-wide discoverability *across domains*

Data modeling as the key to context

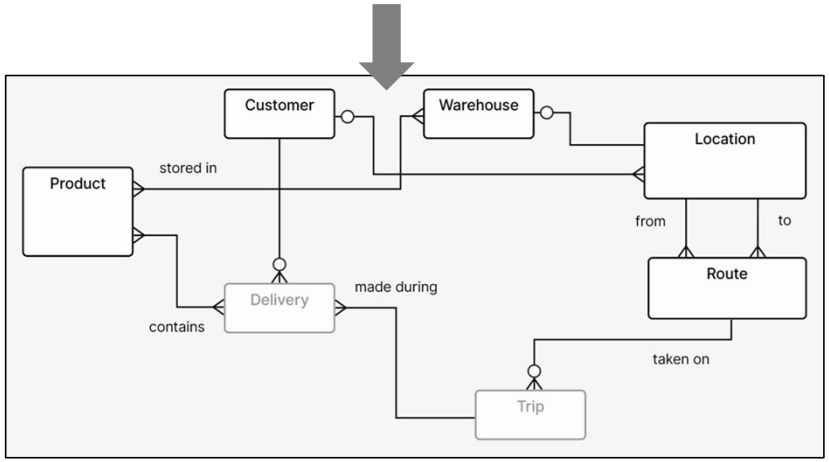
Capturing semantics & structure

Three levels of data modeling



Capturing context with data models

Business entities and relationships of the domain

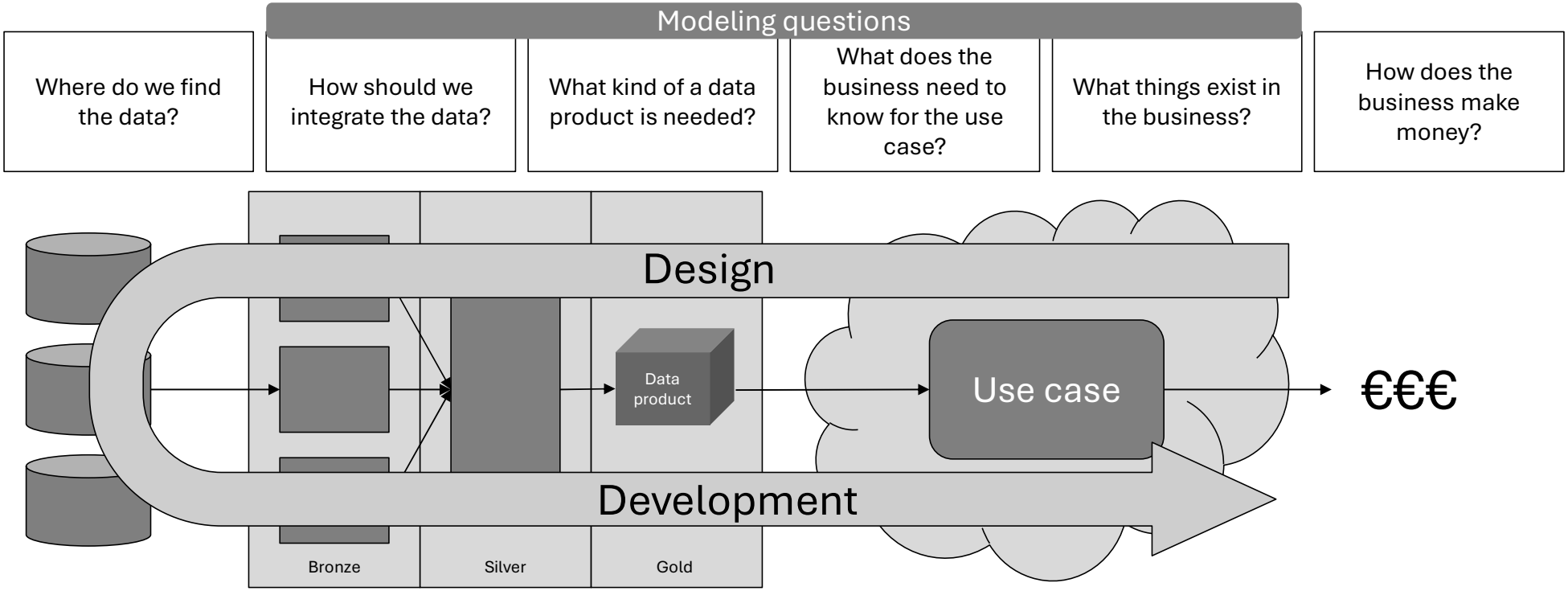


Derived design of a Data Product

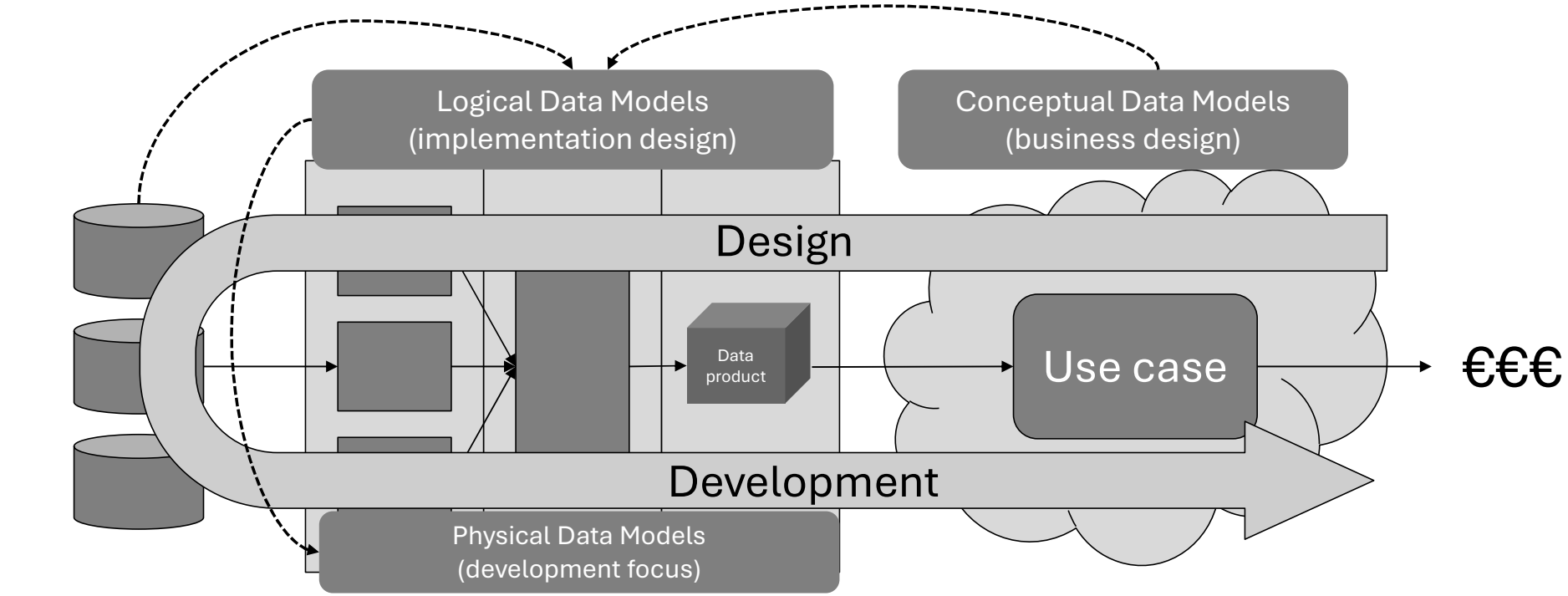
Conceptual model: semantics

Logical model: use case-specific structure

Business-driven data design & development



How modeling happens during the process



Modeling Domains & Data Products

Working on two levels

Two problems to solve at the same time

“SMALL PROBLEM”

How does an **individual Data Product** deliver as valuable results as possible, as efficiently as possible?

“BIG PROBLEM”

How does the **whole organization** ensure that ALL Data Products can CONSISTENTLY deliver valuable results efficiently?

Different approaches needed

“SMALL PROBLEM”

Focus:

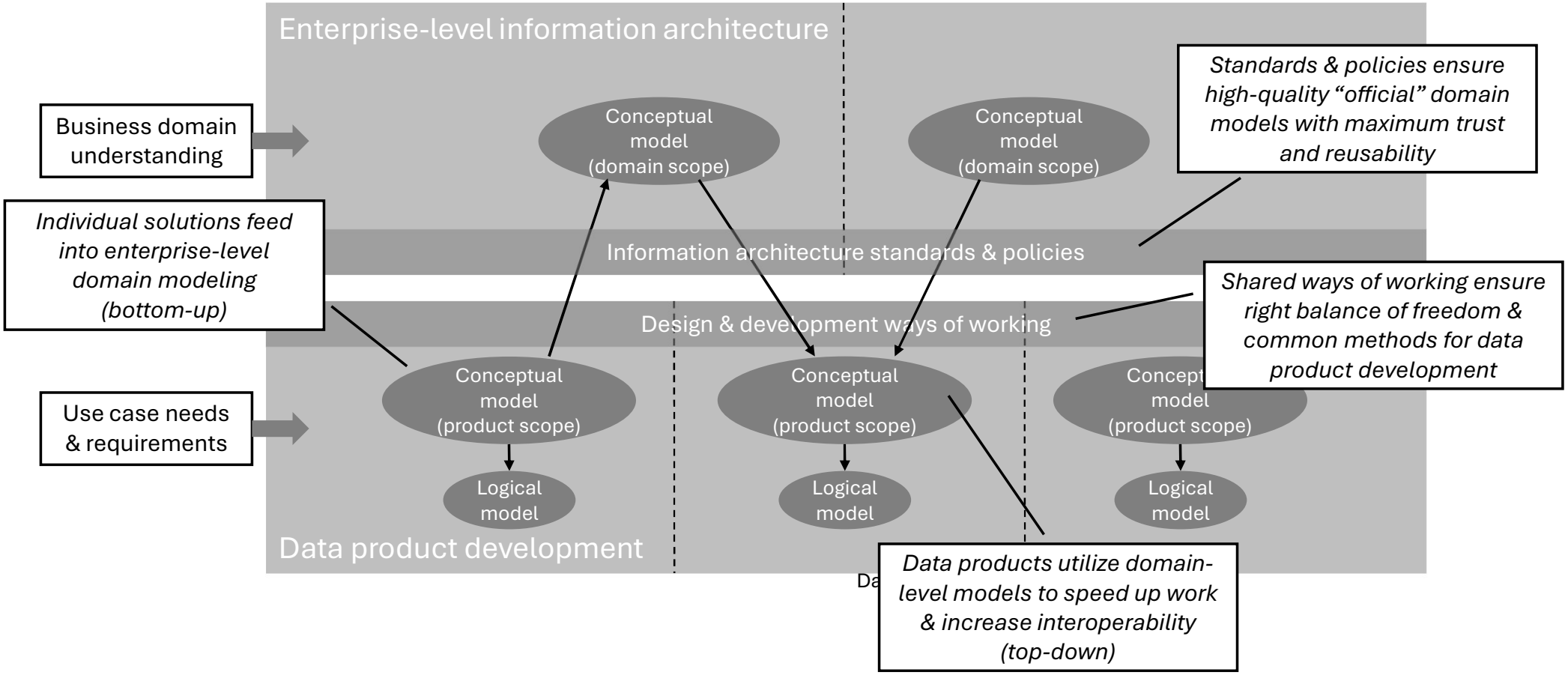
- Understanding the use case
- Development velocity
- Technical optimization
- Minimal documentation overhead

“BIG PROBLEM”

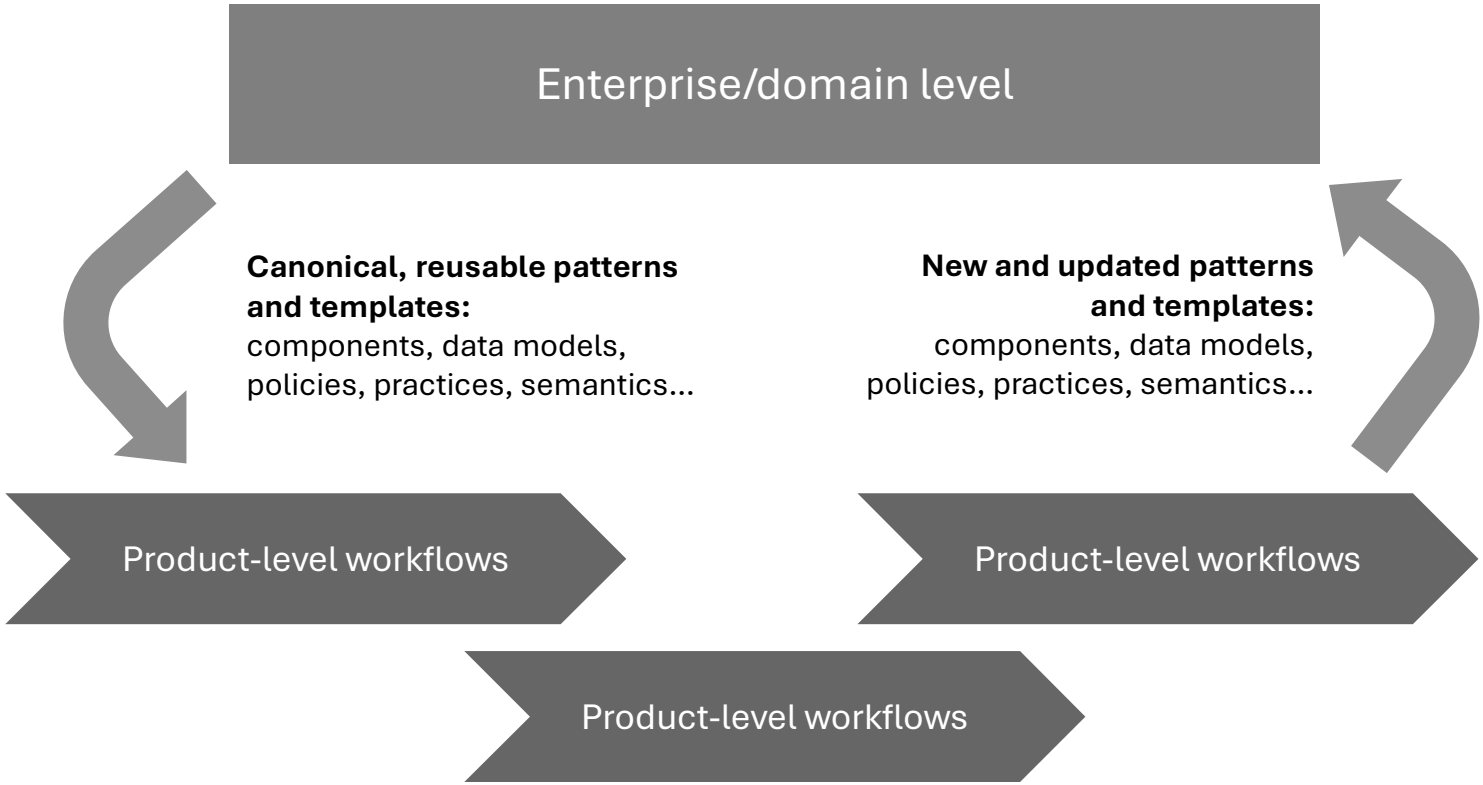
Focus:

- Reusable components & templates
- Shared ways of working
- Avoiding reinventing the wheel
- Shared semantics for interoperability

Organizing data modeling on two levels



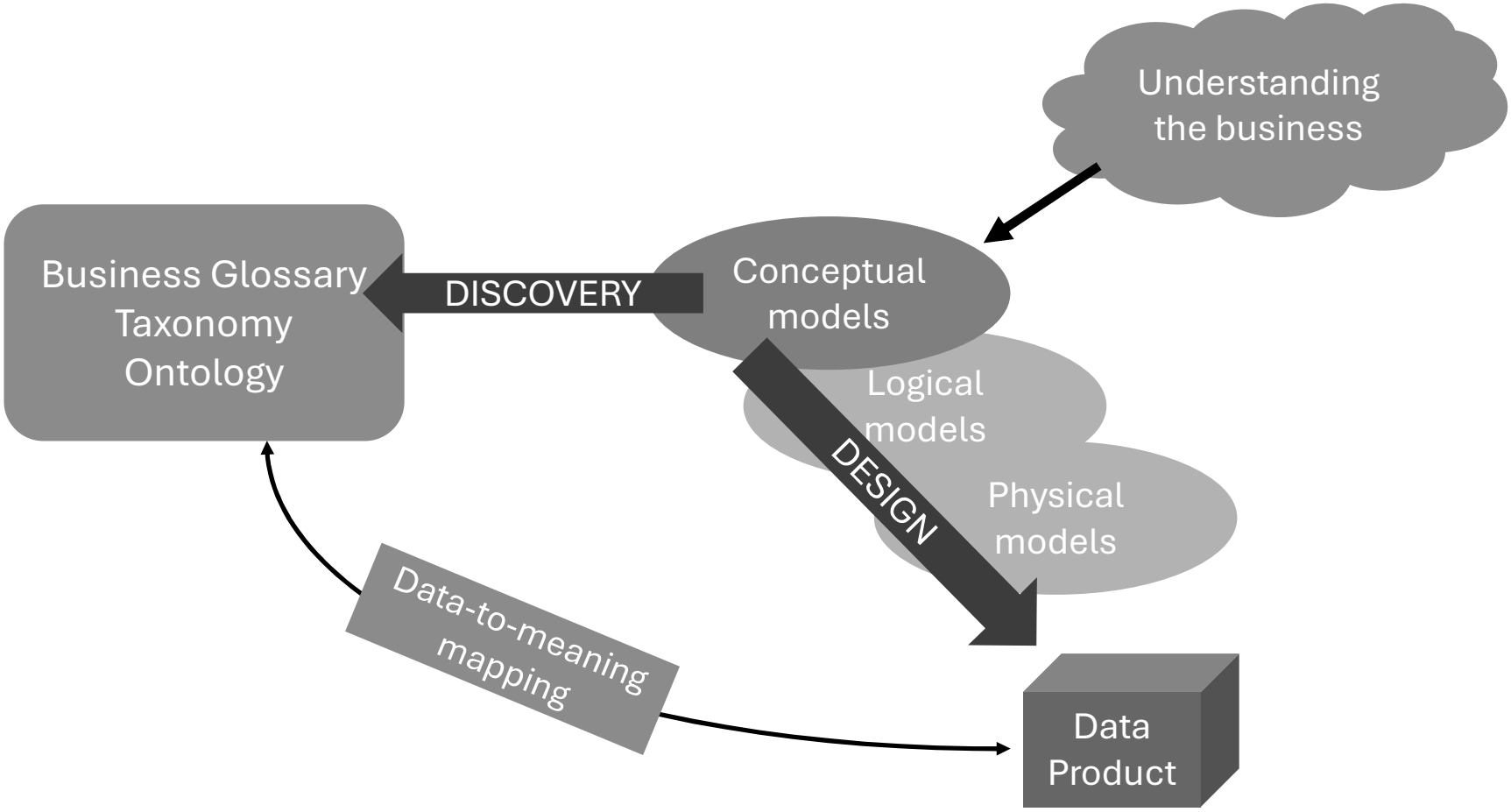
Key success factor: feedback loops



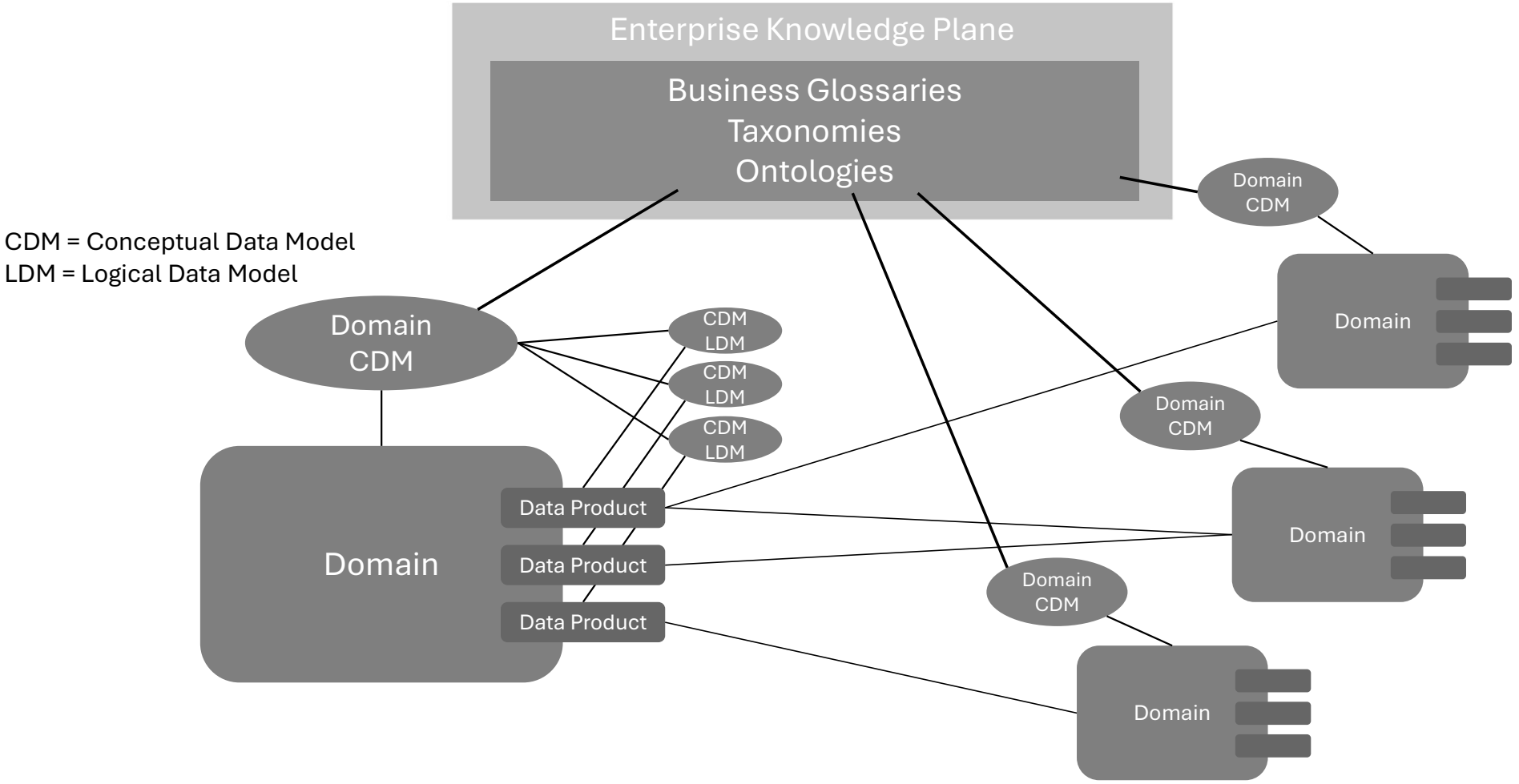
The Big Picture

Managing semantics across the enterprise

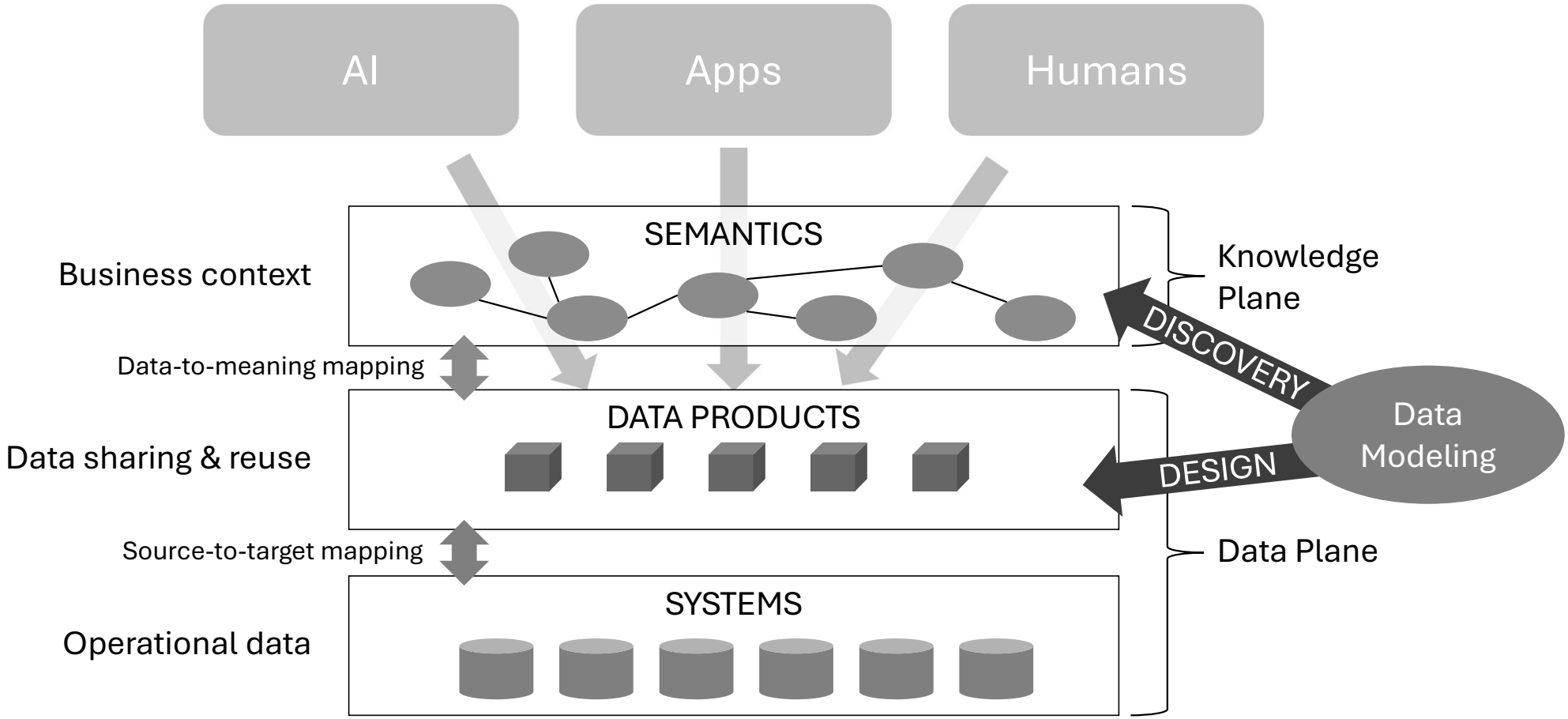
Design & discovery paths



Cross-domain interoperability



Goal: context-aware data utilization



Conclusions

Build interoperability with shared semantics

- Federated model requires common rules & shared understanding
- Data Products must be self-describing – metadata “wrapper”
- Capture semantics with Conceptual models, design product structures with Logical models
- Solve for both product-level delivery and enterprise-level consistency at the same time
- Enable context-aware data use by building a Knowledge Plane of shared semantics

More thoughts on Substack!