Data Mesh in practice

Our learnings from implementing Data Mesh at Roche, one of the world's largest healthcare companies

/thoughtworks

Disclaimer

The practical learnings explored herein have all come from our recent Data Mesh implementation engagement with Roche. However, the use cases and models shared have been simplified for the purposes of this eBook, and do not reflect the final artifacts delivered as part of that engagement.

Introduction

In May 2019, Thoughtworker Zhamak Dehghani introduced the world to Data Mesh — a distributed paradigm created to help bring leading practices such as self-service, Product Thinking, and domain-oriented design into the world of data architecture. Since then, chances are you've heard a lot more about Data Mesh. But most of the writing, discussion, and exploration of the topic that we've seen has been largely theoretical. Until now.

At Thoughtworks, Data Mesh has been a major area of focus for our global teams over the past two years. During that time, we've deepened and defined the architectural paradigm in granular detail and gained a wealth of practical experience from helping major organizations implement it themselves.

Data Mesh in Practice takes a detailed walk through an ongoing Data Mesh implementation project at Roche, one of the world's largest healthcare companies. We explore the decisions, processes, and diverse changes required to bring the full value of Data Mesh to life at scale.

From its inception, we've always known that Data Mesh is — and requires — far more than just technology and architectural transformation. With it comes huge changes in how data experts, domain teams, and even senior strategists work. *Data Mesh in Practice* lays out how to simultaneously manage the cultural, operational, process, product, and technology changes that come together to form the Data Mesh journey.

We'll explore:

- How to manage domain teams and evolve organizational structures in line with the Data Mesh paradigm
- Detailed walkthroughs of robust discovery processes that lead to the creation of high-value data products
- The new ways of thinking that should be embedded at every level of an organization throughout its Data Mesh journey
- Our own proven, 3-stream approach to Data Mesh inception, discovery, and implementation

This isn't theory. This is a practical blueprint for Data Mesh success that's already proven to deliver strong results for even the largest organizations in the most data-intensive industries.

It's the culmination of a lot of great work by numerous Thoughtworkers, and something that we're hugely excited to share with the world. This is where the Data Mesh paradigm truly begins.

Getting off to the right start

Since Thoughtworker Zhamak Dehghani published her first article **laying out the concept and principles of Data Mesh** in 2019, a huge amount has been written on the topic. Now, with a significant number of successful Data Mesh projects under our belt, we're able to enrich our understanding of the **practice** of Data Mesh.

In this eBook, we'll share practical learnings from our recent Data Mesh implementation engagement at Roche, and lay out what it really takes to successfully apply the principles of Data Mesh in a large corporate environment.

As Zhamak points out in her book **Data Mesh: Delivering Data-Driven Value at Scale**, Data Mesh isn't just another architectural approach — it's a **sociotechnical** paradigm that demands process, operating model, and technological transformation. "Sociotechnical" refers to both the layers and nuanced ways that human social systems interact with — and come to grips with — technology to produce improved outcomes.[1]

1. The term "sociotechnical" was introduced by researchers at the Tavistock Institute in London in the middle of the 20th century. The canonical paper explored how a change in sociotechnical arrangements of work resulted in improvements in team cohesiveness, work satisfaction, and reduced sickness and absenteeism. See Trist EL and Bamforth K (1951) Some Social and Psychological Consequences of the Longwall Method of Coal-Getting: An Examination of the Psychological Situation and Defences of a Work Group in relation to the Social Structure and Technological Content of the Work System. Human Relations 4(1): 3–38. DOI: 10.1177/001872675100400101

Principles guiding all building blocks of Data Mesh

Domain-oriented decentralized data ownership and architecture	Data as a product	Self-service data infrastructure as a platform	Federated computational governance

The four principles of Data Mesh — as defined by Zhamak — help us start to understand the range of sociotechnical requirements needed to build a strong Data Mesh.

The first two, 'domain ownership and architecture' and 'data as a product,' both demand significant changes to operating models. Whereas 'self-service platforms' and 'federated computational governance' demand deep engineering expertise and evolutionary architectures. These are two very different types of change, and if they aren't understood and approached in the right way, they can undermine the huge potential value of the Data Mesh approach.

We will take a deep dive into the organizational, process and operating model changes required to support Data Mesh, the practical demands of building data products, and the architectural requirements of the approach. We'll show you how we've balanced the diverse changes needed to bring Roche's vision for Data Mesh to life, and share some of our proven principles and practices to help you do the same.

The process and key practices at a glance

Before embarking on any data journey, especially a Data Mesh one, it's essential that you have a clearly articulated data strategy at the organizational level. To create that strategy, teams must answer value-oriented questions, such as:

- How are data, AI, and other leading capabilities going to deliver value for our business?
- Do we want data to help us increase revenues, cut costs, improve customer experiences, or create new revenue streams — or some combination of the four?
- What are our key strategic data use cases, and how can our data strategy support them?

Answering those questions enables organizations to make informed decisions about what they really need from their data architecture, and help shape how they ultimately measure the value delivered by their chosen architectural approach. In this eBook, we're looking at the process that follows that definition — exploring the journey that begins once an organization has clearly defined its data strategy and determined that Data Mesh is the right architectural choice to support and deliver it.

We focus on the practices, principles, and steps that Thoughtworks takes to plan and execute successful Data Mesh journeys. Each chapter explores a different aspect of those journeys, but here's a quick snapshot of how those actions come together across the journey as a whole:

Cross-domain planning and scoping

- The process begins with **cross-domain and leadership teams defining the 'why'** for their organization — their motivations for adopting a data-centric approach to their business. This helps them shape their data strategy, and begin with a clear view of the business value and benefits that data strategy can deliver for their team
- The team can then **determine if Data Mesh is a suitable architectural paradigm** to support that strategy, and set **measurable, value-based goals** that will be used to determine how effectively their decisions are moving them towards the benefits their strategy intends to deliver
- Leaders also **define the 'what'** — early hypotheses for how they're going to do that, mapping out domains to include early

Operating model: From project to product	Product thinking	Technology evolution
• **Operating model is defined** to ensure that teams across and within domains can work to create greatest value from the Data Mesh model with ease	• **Use principles of design thinking to** identify longlist potential data products	• **Approach and manage data products as atomic, functionally-cohesive units**
• **New roles and responsibilities are defined**, for clarity and empowerment of domain and data product actors	• **Map hypothesized products to a Lean Value Tree** to determine high-value options	• **Establish a streamlined developer experience** for creating and maintaining data products
• **Governance structures are defined**, building alignment on desired outcomes, balancing autonomy and interoperability, and ultimately ensuring consistency across the Data Mesh	• **Define the exact purpose and intended value of use cases and agree how to measure it**	• **Create a consistent definition of a data product across** the entire organization
	• Work the longlist down into a shortlist of high-value use cases, and **work back towards the data products needed to bring them to life**, in line with the lean value tree	• **Automate governance** and Access Control Policies
		• **Apply fitness functions** to guide the evolution of the mesh

Cross-domain maintenance and evolution

- Decisions made across all three areas are **measured against defined success metrics**, enabling continuous improvement, and constant iteration to improve and optimize the Data Mesh
- Guardrails ensure **high interoperability between data products**, and ultimately ensure the success and value of the Data Mesh

If you're looking for insight into a specific aspect of the Data Mesh journey, you can jump ahead to the relevant chapter using the links below, where you'll find expanded explanations of the practices and principles introduced above:

- **Planning your operational and organizational evolution**
- **Choosing and building your first data products**
- **Executing your technological and architectural transformation**

To start with, we'll look at a few of the consistent, high-level challenges faced when adopting Data Mesh, before introducing our design-thinking-based double diamond process that we use to kickstart successful, high-value Data Mesh journeys.

Challenge #1: Overcoming the dichotomy between rapid scaling and continuous learning
Following the principle of network effects[2], the value of Data Mesh increases with the number of interoperable use cases it serves, so scaling is a top priority for enterprises adopting the model. However, their eagerness to scale at speed has consistently created a challenge across all of the implementations we've seen.

From the social perspective of our sociotechnical paradigm, organizations adopting Data Mesh are on a learning journey. They have a lot of experimenting to do to determine what's going to work best for them. In the process, they'll discover powerful business and customer use cases that deliver unique value for their organization. Naturally, they want to apply those lessons across as many domains as possible.

2. Barabási A-L (2002) Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. London: Plume.

However, if you scale too fast, you won't have the opportunity to learn effectively or incorporate what you've learned. Going too quickly creates a dynamic where distributed domains are all doing their own learning, but not learning from one another or collaborating on collective data efforts. Teams falling prey to this challenge get stuck in the experimentation stage, searching for ways to solve their own issues without identifying any of the enterprise-wide challenges that the organization as a whole could overcome using Data Mesh.

To address this potential trade-off and overcome the challenges it creates across our Data Mesh projects, we've designed a set of practices to help organizations successfully balance both the learning and scaling demands of Data Mesh adoption. In keeping with the federated nature of the Data Mesh, our approach allows for parallel work by decentralized teams.

Challenge #2: Defining and empowering domains
Our long experience practicing **Domain Driven Design** has helped us understand the benefits of drawing clear boundaries around contexts to separate concerns and allow teams to focus on solving problems within the context they fully understand and can control. It is natural that many Data Mesh implementations try to define domains to which they can assign data product ownership. However, in our experience, it is not necessary to reorganize business units or departments for the sole purpose of designing data product teams and data products. In many cases, you can start assigning data product ownership within your existing structure, and evolve it as required, as your journey progresses. In determining who owns the data, we follow the business outcomes to business processes and existing decision making frameworks around it.

Data Mesh empowers domains to make their own data-related decisions and build their own data products. That's a lot of autonomy — especially compared to traditional centralized data architecture approaches, where everything must go through a single IT or data team.

That level of freedom naturally raises a lot of important questions: Who needs to be part of goal definition conversations? How should outcomes be measured? Who keeps an eye on progress towards targets and offers management support to teams when they need it?

To help answer those questions, we drew upon Thoughtworks' EDGE operating model for inspiration on how to connect high-level business goals right the way down to a data product team's backlog items. We found that EDGE lends itself well to the context of domains adopting Data Mesh. Rather than giving teams prescriptive outputs that they need to work toward, they're aligned around specific goals to deliver customer value. Each domain has the autonomy to decide the best way to reach their domain-specific goals.

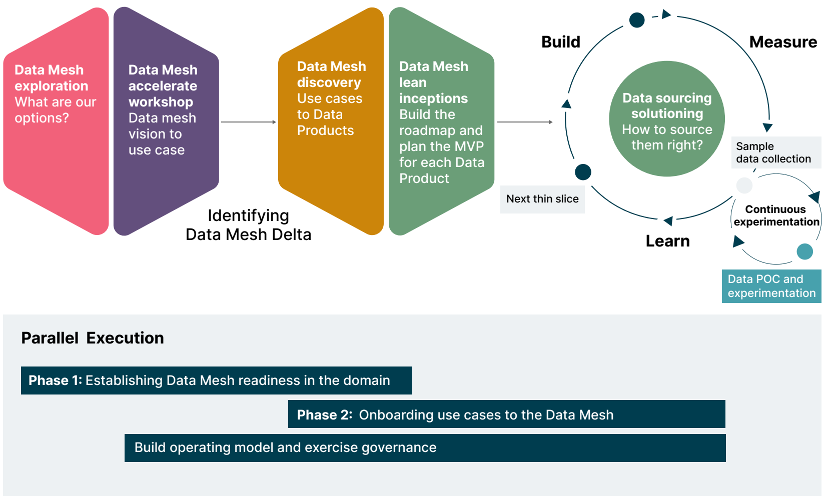
The result is a set of domains that are all aligned with organizational strategy and data strategy, but empowered to work toward strategic goals in ways that make sense based on their context, and apply their domain-specific knowledge to leverage data in creative ways, enabling them to deliver more customer value.

Challenge #3: Running before you can walk
As Data Mesh best practice is constantly evolving, there is no single 'best path' for teams to follow. They've determined that Data Mesh is right for them, defined some initial use cases, and

got to work on making it happen. Unfortunately, in some cases, that's led to ineffective projects that haven't made it past the POC phase.

Through our experience, we've applied the Design Council's 'double diamond' approach to onboarding a new domain to Data Mesh, as shown below.

Onboarding a business unit / domain to the Data Mesh process



The double-diamond ensures that domains focus on the business **'why'** and **'what'** before they move on to think about the engineering **'how'**. The first phase of the double diamond incorporates the vital discovery step providing the 'why' and 'what', while the second phase of the double diamond addresses the 'how'.

Key practice: start your journey with your existing organizational boundaries

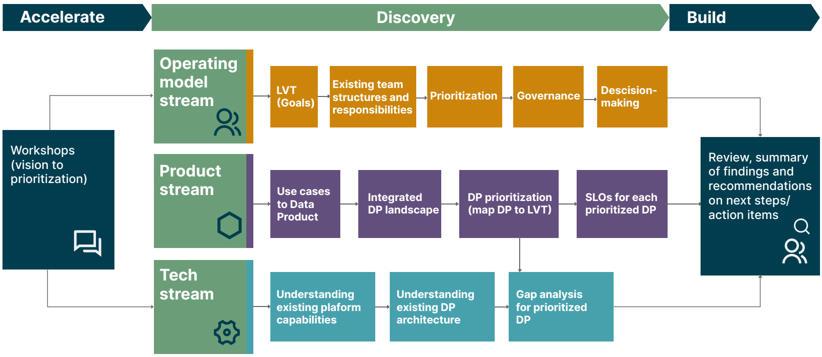
Defining domains is a key part of the Data Mesh journey, but redefining domain boundaries can introduce a lot of additional challenges, and isn't always a necessary prerequisite. We recommend starting with the organizational boundaries you already have, and only moving those boundaries to create new domains if you hit significant barriers along the way.

By going through a detailed discovery process, domain teams know exactly what they're getting into and can strategically align with business goals and customer value. They're able to clearly lay out what they want to achieve by joining the Data Mesh, how it can help them contribute to strategic goals, and what they'll need to get there. It starts the onboarding process, while also providing a framework for everything that needs to come after it.

One approach we have found very successful here is to define, as part of the discovery, the "Data Mesh delta" — the gap between where the domain currently is vs what they want to achieve with Data Mesh. To identify this delta, we break our discovery process down into three streams:

- The operating model stream
- The product stream
- The tech stream

In line with the principles of Data Mesh, those streams can run concurrently. The graphic below shows the three-stream discovery process that we're currently running across multiple domains at Roche:



The three streams identified and explored during the discovery process don't just help to scope and plan a domain's onboarding onto the Data Mesh. They provide a framework for how onboarding should be executed, and the simultaneous organizational, process, and technological changes that are needed to ensure that Data Mesh ultimately delivers its full value for each domain, and the organization as a whole.

Adopting Data Mesh requires more than just technology change

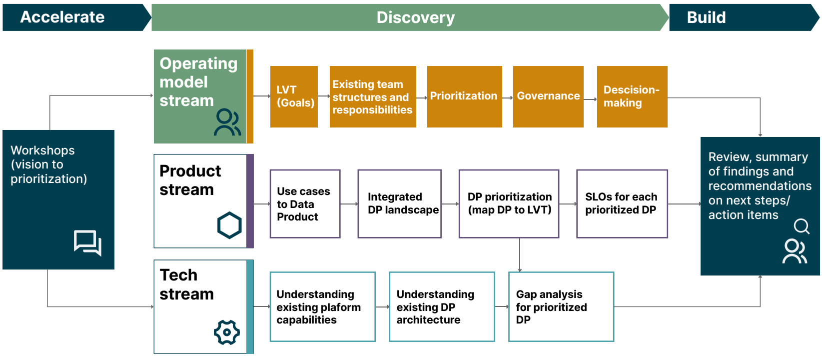
In the following chapters, we'll take a deep dive into the three different types and areas of change required to achieve Data Mesh success. Each will offer insights to those that work across the relevant domains, concluding with a high-level overview of the diverse changes and decisions required to successfully adopt, embed, and drive value from Data Mesh.

We'll share artifacts and insights from our recent Data Mesh implementation engagement with Roche throughout. Through the lens of their journey, we'll show you exactly how these different types of change come together to create a robust Data Mesh implementation and onboarding strategy.

In the next chapter, we'll explore the organizational and operating model decisions that teams need to make throughout their Data Mesh journeys, introducing principles and practices used across our projects to ensure success and maximize value creation.

Organizational operating model

We previously introduced a three-stream discovery process that we undertake at the beginning of an organization's or domain's Data Mesh journey to align business, product, and customer outcomes. Now we dive into the first of those streams, looking at the operating model changes required to support Data Mesh, and the discovery process that helps us identify and define them.



Identifying how operating models need to evolve to support Data Mesh

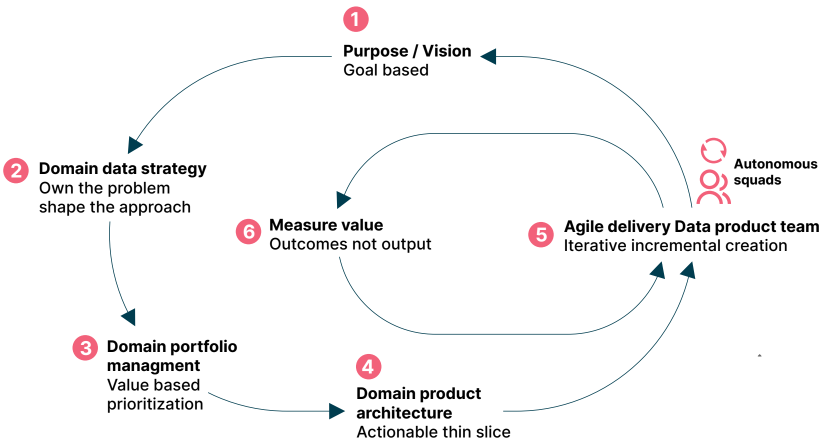
The operating model discovery stream starts with a clear view of the company's and domain's vision for Data Mesh. By starting with strategic goals and priorities, the domain can work back toward identifying the necessary steps, changes, and capabilities needed to bring that vision to life — rather than jumping in and working out what's needed along the way.

In the operating model stream, stakeholders define how they'll work once they're part of the Data Mesh, and how processes and practices will need to evolve to enable it. Through that process, they answer critical questions including:

- Does our current team structure align with Data Mesh principles, and will it need to change? If so, how?
- Who within the domain will take responsibility for our data products and become our internal data lead if we don't have one? Will they need a defined team to support them?
- How will we prioritize between the Data Mesh use cases that our domain wants to explore, and which of those are most valuable to us immediately?
- How will we prioritize use cases when they're relevant and valuable to multiple domains, and how can we incentivize data product sharing to ensure that each product delivers maximum business value?
- How will we manage governance internally, what guardrails will we need to put in place, and how can we ensure those guardrails don't undermine the autonomy and flexibility that Data Mesh enables?

Our operating model foundation

As we answer those questions across our Data Mesh projects and engagements, we guide clients towards operating models based on the **EDGE operating model**, as shown in the graphic below:



1. **Purpose / vision:** Start with corporate level vision and goals and break those into domains' goals & value hypotheses. The value hypothesis articulates how analytics might support the business goals. Initiatives under them speak to how to validate the hypotheses. Each initiative has one or more measures of success.
2. **Domain data strategy:** Each domain decides how to leverage platform capabilities to build and maintain interoperable data products in their domain, in line with the company-wide data strategy.
3. **Portfolio management:** Value hypotheses are prioritized according to value. The corresponding analytical initiatives to build or change data products make up the portfolio of investments. These flow to product teams' backlogs. Work in progress is limited. Stop starting, start finishing!
4. **Product architecture:** Value is turned into actionable thin slices ready for delivery. Large programs of work are broken into small 'learning driven' chunks.
5. **Agile delivery:** An adaptive and incremental approach to delivery with fast feedback loops early and often
6. **Measure value:** Work is measured on value delivered, not work output.

The EDGE model is built on a lot of the same principles as Data Mesh. Both EDGE and Data Mesh:

- Emphasize autonomy across domain teams
- Empower teams to achieve their goals, their way, without prescriptive delivery requirements
- Advocate for developing multiple use cases 'bets', sometimes referred to as 'value hypotheses', simultaneously, so teams can easily pivot between them if one doesn't work out

- Challenge traditional centralized structures and propose new approaches to governance and the development and execution of strategy

That similarity makes the EDGE model a good fit for many organizations and domains that are adopting Data Mesh for the first time. By taking the EDGE operating model as your starting point, your journey can begin with a model that's tightly aligned with the principles of data mash. The model also ensures that every decision-maker has a consistent definition of what constitutes 'value' for the Data Mesh, and how to prioritize different value hypotheses across domains.

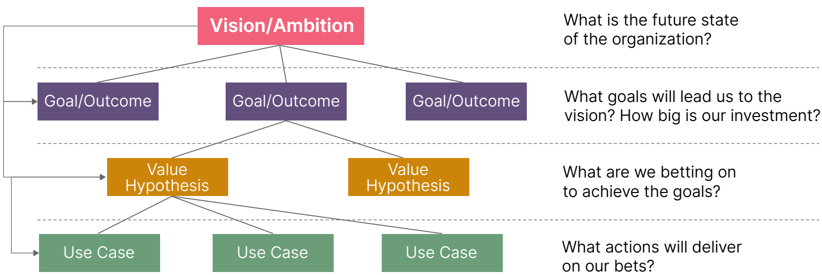
Key artifact: The Lean Value Tree

The EDGE model also helps us create one of the most valuable outputs of the operating model discovery process - alignment on priorities. The artifact which represents and communicates this alignment is a Lean Value Tree (LVT). It's worth noting that other methods call this cascade of outcomes differently; these can be just as valid if the principles described here are applied.

The LVT is broken down into three tiers:

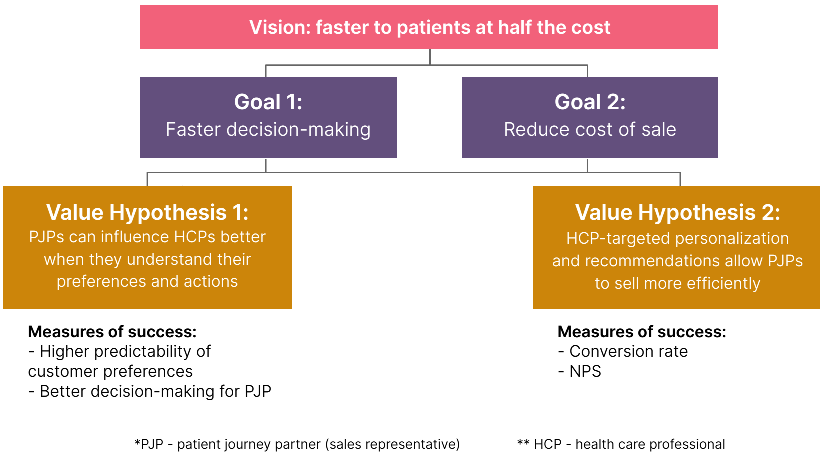
- **Our vision:** This describes at the broadest level, what the organization or domain wants to achieve. This is the starting point during the discovery process, as it defines the overarching goal that all Data Mesh efforts need to drive the domain and organization towards.
- **Contributing goals:** The tier below the vision defines a small number of specific goals that can come together to make the vision a reality.

- **Hypotheses or bets:** The third tier of the tree lays out multiple hypotheses for how Data Mesh and/or data products could help support each of those goals. Multiple hypotheses are created by applying design thinking, ensuring that if one doesn't deliver on its hypothesized value, the team can easily move along to the next hypothesis and maintain progress towards its goals. It's important to acknowledge that other types of efforts than data or Data Mesh will likely be needed to achieve the overarching goals. It's useful to capture those for transparency and improved communication between teams, even if the specific Data Mesh effort isn't charged with executing on them.
- **Measures of success:** While not a tier of its own, we list Measures of Success, or MoS, here. They define how we'll know if those hypotheses are delivering their intended value and are making progress towards our goals. This is an important step, as it helps us define granular success indicators, which will let us know which hypotheses to focus on, and when it's time to move on from one to the next.



Throughout this eBook, we'll follow a running example from our recent Data Mesh implementation project at Roche. The example LVT below was created through a discovery exercise with one of Roche's commercial business units — a team responsible for accelerating healthcare outcomes and bringing healthcare products to patients faster. This is done by PJP (Patient Journey Partners) engaging with their corresponding HCPs (Health care professionals), who in turn bring the medication to the patient population.

Here's how we mapped their vision, goals, value hypotheses, and measures of success into an LVT:



Key practice #1: Ensure your organization is ready for data decentralization

Data Mesh is a decentralized architectural paradigm. So, for the millions of businesses built around centralized structures and organizational designs, it represents a significant evolution, and requires well-planned change.

If you simply try to shoehorn Data Mesh into a centralized organizational structure, it's very unlikely to deliver the value you want it to. Within centralized structures, it's easy to create scenarios where central teams and leaders solely drive

Data Mesh adoption, and domain teams know very little about the initiative, or what it's designed to do. Naturally, that leads to low adoption and buy-in, and ultimately results in a Data Mesh implementation that has very little impact on the business.

Instead, organizations need to consider how well they're set up for Data Mesh success. You don't (Typically) need to rebuild your organization from the ground up, but it's certainly worth reevaluating your organizational model and structure, to see if there are any ways that you could better support and enable the collaborative and bottom-up input required to drive Data Mesh success.

That isn't a quick or insignificant step. Organizational evolution requires its own transformation program and careful change management. You're laying a foundation for decentralized data success and domain-driven innovation. Depending on how your organization operates today, aligning your structure and ways of working with that can be almost as significant as your Data Mesh implementation itself.

Key practice #2: Clearly allocate and define responsibilities
While the concept of Data Mesh is now well-known in the data and digital space, for many domain teams, it's still new territory. So, it's important that they understand what joining the Data Mesh means for them, and what their individual responsibilities within it will be.

Once a domain joins the Data Mesh, the people that make up that domain team become custodians of the data and data products they create. That's a type of responsibility many won't be used to, so we use the discovery process as an opportunity to educate teams about what it means for them, and what they'll need to do once everything is implemented.

Establishing accountability is also important at this stage. The whole team may be on board with the concept of Data Mesh and be eager to start building their own use cases and data products. What is the new role of leadership in this setup? Governance may be federated within the Data Mesh, but within each domain, who is ultimately accountable for data products and decisions made regarding them?

Answering those questions and allocating and defining any new roles required is a valuable part of the discovery exercises we've run so far. However, it's also important to put relevant incentives and support in place to ensure that the people nominated to take on new roles and responsibilities are supported and encouraged to do so.

As part of the process, domain owners — those individuals also accountable for the business outcomes of the domain — align defined objectives and priorities with their managers and peers in other domains to ensure cohesion in outcomes.

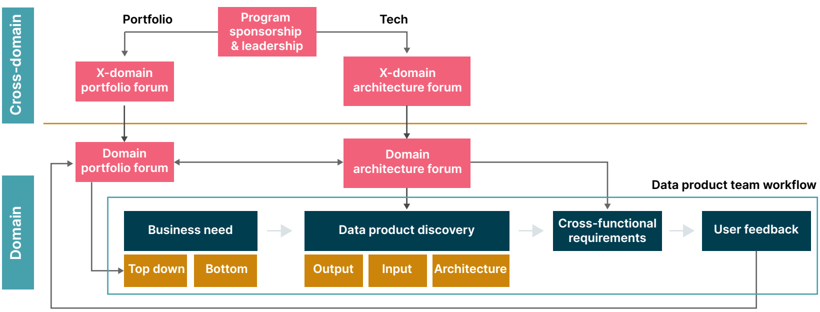
Data product teams then own their products collectively, just like software teams own their code collectively. Each data product needs a nominated owner who acts as the team's ambassador and key communicator to their stakeholders and other data product teams. They take charge of the product roadmap and lifecycle, communicating expectations and facilitating collaboration.

Finally, the data platform team provides services to the data product teams using the platform. Depending on the platform's scope and scale, a platform may have multiple teams working on providing those services, in which case they'll also scope their work along complete outcomes, just like the data product teams do.

The platform owner takes on product management tasks for the platform, working with the team to plan the platform's evolution, take on feedback from users and communicate any planned changes. For companies with multiple platforms, it's imperative that the various platform owners align on how they'll guarantee interoperability, but we'll explore that in more detail later.

Key practice #3: Define clear structures for governance
Another important output of the operating model stream are clear governance structures, like the example below.

Interaction between governing bodies and data product streams



The structure shows how governing bodies and data product teams should interact and helps communicate where different responsibilities lie. Crucially, it also helps demonstrate how governance within the Data Mesh transforms from a top-down approach where a single team acts as a gatekeeper, into a bottom-up approach where domain and cross-domain teams can make suggestions for how data products should be managed and connected.

Across the Data Mesh, we need to look at three different types of governance:

1. **Portfolio governance:** Portfolio governance is applied at the cross-domain level and is concerned with making sure the company-wide goals are being met and appropriate value hypotheses are articulated. Special attention is paid to achieving the cross-domain outcomes needed for the company-wide goals. Measures of Success from the LVT are continuously discussed to understand which bet has paid off and which needs adjusting or replacing. The domain owners and executive management representatives play an active role in this conversation.
2. **Domain governance:** At the domain level, product and domain owners decide on which data use cases to pursue, and make sure that the right data products are being created to support and help achieve overarching goals. Those teams break use cases down to identify the data product required to realize them. Large clients tend to run another round of portfolio governance within the domain to address the volume of work and federation of decision making.
3. **Technology governance:** Domain architects and technical leads agree on how to build data products and set standards for domain and product teams to follow. These guardrails ensure interoperability between data products, without limiting domain-driven innovation.

We'll explore more about technology governance in our final chapter, where we'll take a detailed look at computational and federated governance across the Data Mesh.

3 principles for successful operational evolution

- Start with a clear vision of what you want to achieve, and work down through the Lean Value Tree to identify specific hypotheses to help you get there
- Empower teams to take up and act on their new responsibilities, and ensure that they're able to move on from failed hypotheses quickly, in line with the EDGE operating model's feedback cycles.
- Consider governance early in the adoption and onboarding process, and build it into the product and technology decisions that will follow, rather than working to build structures around what you've created later

Establishing priorities and making a domain's first Data Mesh decisions

The operating model stream of the discovery process brings domain teams to the point where they've defined several high-value Data Mesh use cases, and prioritized the use cases they'd like to begin their journey with. This doesn't just help ensure their journey starts with fast, clear, high-value wins — it also helps establish repeatable processes for decision-making and prioritization on future Data Mesh projects and bets.

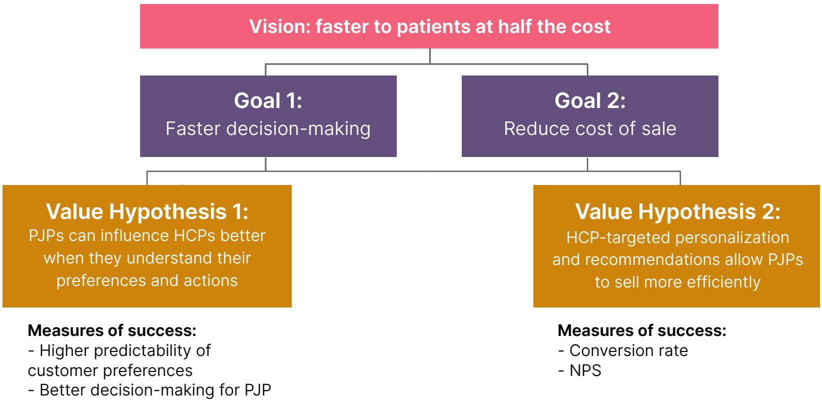
In deciding on the use case and the measure of success for the use case, the team has also already defined the very first consumer data product — the value measure by which they'll track and communicate their progress in adding business value from data.

Crucially, it does one more thing. Determining the use cases a domain wants to focus on provides valuable input for the product stream of the exercise, establishing what kind of data products might be needed to bring those use cases to life.

Next we'll explore the product stream and look at the processes we've built to prioritize data product creation and ensure data products are tightly aligned to domain and organizational strategy.

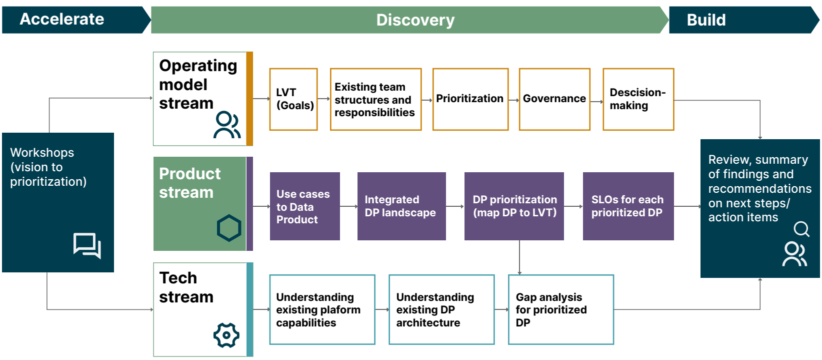
Product thinking and development

So far, we looked at the operating model changes required to support Data Mesh, and the discovery process we go through to define them using EDGE's Lean Value Tree (LVT) — ultimately leading to the creation of high value customer outcomes.



*PJP - patient journey partner (sales representative) ** HCP - health care professional

The principles of the LVT shape how we approach and think about data products and their creation. This serves as the starting point for the Product stream of the Data Mesh Discovery process.



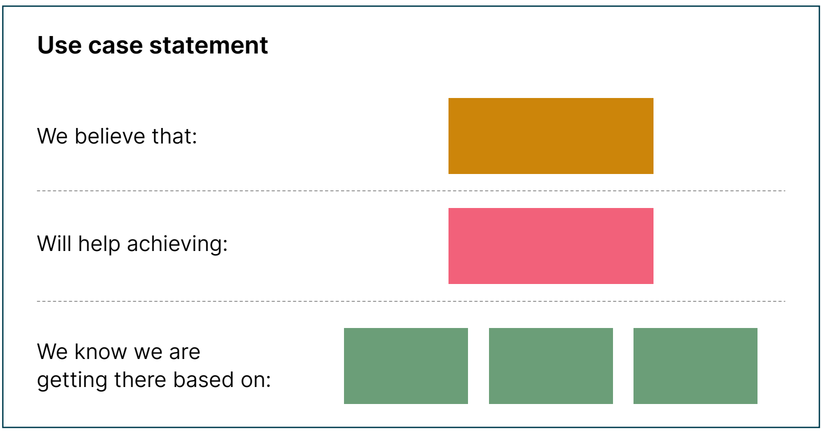
When teams want to start treating data as a product, we recommend working backwards from organizational goals to identifying high-value analytical use cases, and ultimately, which data products are needed to bring the use cases to life.

Throughout the process, relevant stakeholders prioritize their goals and hypothesized use cases, ultimately helping us make informed and value-oriented decisions about which Data Products should be built.

This approach ensures that domain teams and their organizations make intentional, considered choices about the data products they add to the mesh — guaranteeing that teams don't end up accidentally creating something similar to a data lake monster.

Key practice #1: Complete value-oriented templates for every identified use case

For each identified use case, we take a structured approach to ensure that it can easily be mapped back to the LVT and what we ultimately want to achieve. To help us do this, we use a hypothesis use case template:



It's a simple framework, but it helps ensure that every hypothesized use case for data products begins with a clear view of its intended value, and a clear definition of how that value will be measured and realized, as seen in the two examples from our work with Roche below:

Digital Engagement Dashboard

We believe that:	Understanding how HCPs react to PJP activities (example F2F meeting)
Will help achieving:	Better support brand customer interaction
We know we are getting there based on:	Prediction of customer preferences; Success of a campaign

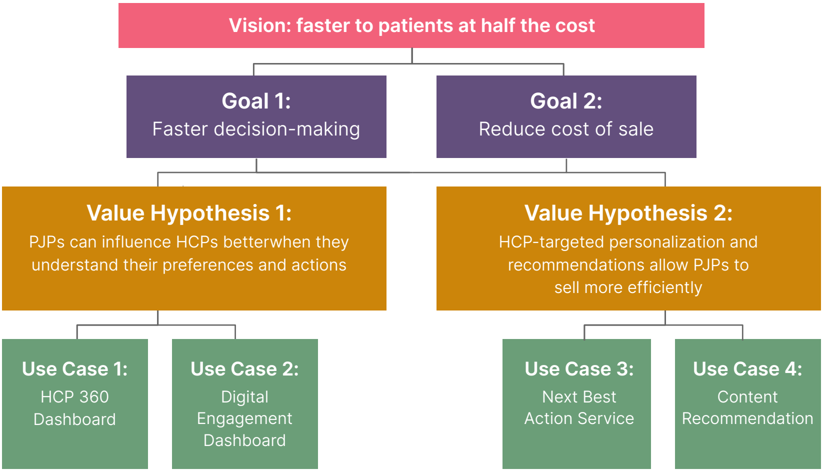
Example of its use in practice
As a Marketing Director, I want to evaluate the effectiveness of the last MS email campaign by comparing all relevant email metrics (e.g. opening / click / bounce rate) with previous MS campaigns.

HCP 360 Dashboard

We believe that:	Giving a PJP a 360 view of a HCP
Will help achieving:	A higher success rate in influencing a HCP
We know we are getting there based on:	Better decision making for PJP; Higher predictability of customer preferences

Example of its use in practice
As a Patient Journey Partner, I want to get an overview of the engagement history and previously discussed topics of an oncology HCP who I will meet for a F2F meeting tomorrow.

With each use case defined within the template, we can then flow them back into our original Lean Value Tree:



From there, the outlined use cases are prioritized based on what the business wants to achieve and pursue. We use the following data use case prioritization template for this, but any similar prioritization method is equally suited for the task:

		HCP 360 dashboard	Digital customer engagement dashboard	Next best action
What potential value or negative impact to our business? (value from 0 to 5)	Business value	5	5	4
How fast does business value decrease over time? Will users wait for us or find another option? (value from 0 to 5)	Time criticality	4	5	4
What is the risk of delaying this initiative for our business? Will this initiative open / facilitate new business opportunities? (value from 0 to 5)	Risk reduction	4	5	4
How many people will be impacted by this initiative? (Consider for same period) (value from 0 to 5)	Reach	3	5	4
How much will this impact each of these people? (value from 0 to 5)	Impact	4	5	4
How confident are you about these numbers? (value from 0 to 5)	Confidence	5	5	4
How does this work compared to others? (lowest effort 5, highest effort 0)	Effort	5	4	3
SCORE = Business value + Time criticality + Risk reduction + Reach + Impact + Confidence + Effort	SCORE	30	34	27

Whichever framework you choose to follow, it's critical that the right IT and business stakeholders are involved at every stage of this prioritization process. You want to begin your journey with a complete view of what's most important to the domain, and which route is best to get there, so gathering broad input is highly valuable.

Embracing product thinking

At this stage, it's worth noting the role that **product thinking** plays in the Data Mesh. Data products are named as such because that's exactly what they are — they're products, selected and valued by consumers.

To deliver its potential value, Data Mesh requires the domains building products to understand and apply the principles of product thinking. For some — especially those used to working closely with customers and responding to their needs — that may come naturally. For others, it may require enablement and upskilling.

In line with product thinking best practices, domain decision-makers joining the Data Mesh should understand principles including:

- **Knowing your customer** and understanding how they want to interact with your product — and by extension, how your product needs to be designed to align with how they operate.
- **How product and project mindsets differ** and how products will need to continuously evolve and improve throughout their lifecycle.
- **The value of cross-domain knowledge and data sharing** and the need to not only focus on your own use cases and hypothesized benefits, but those of other domains too.

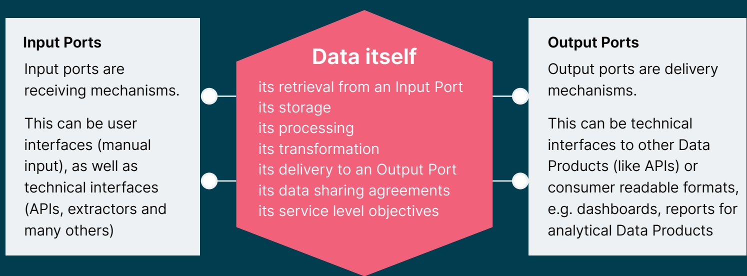
- **The value of diverse perspectives** and inputs, and where and when to create cross-functional teams to deliver specific product outputs and outcomes.

How does a data product differ from a data asset?

A data asset can be any entity that is composed of data — such as databases or application output files.

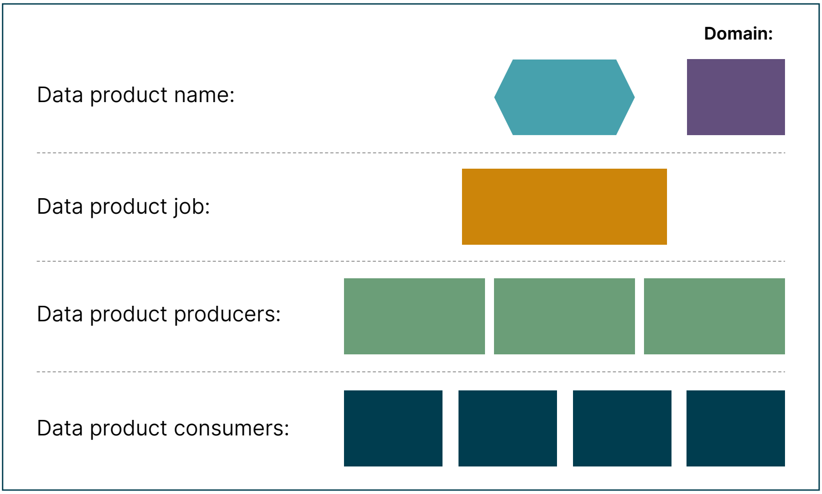
Data products however are:

- Created to serve a specific user-driven goal as identified in your Lean Value Tree
- Subject to clearly defined SLOs
- Owned by a single domain or stakeholder and maintained by a single data product team, who are responsible for their upkeep



Key practice #2: Completing the data product template
Once you've got a clearly prioritized list of use cases, it's time to start identifying the data products that are best suited to satisfy and enable them.

Here we introduce a simple data product template that helps articulate exactly what a data product needs to do, and how it will do it:



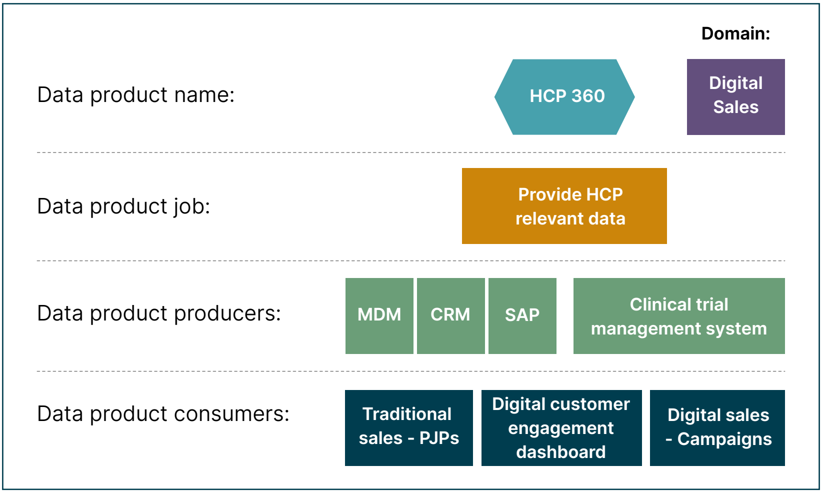
Six questions to shape your data product template

The following six questions have helped guide the product stream of our discovery process, allowing domains to determine exactly which products they need to create, and how those products should come together to deliver maximum value.

- Who will use the data product? And which stakeholders does it most directly serve?
- If we would expose this data product, would it be valuable for consumers? And are there any other stakeholders or domains that would be interested in this data product?
- How will they consume and engage with the data product? Which tasks or actions will they use the data product to support, and how can we meet their consumption requirements at that point?

- How would those consumers access or consume this data product?
- Which input data is required for the data product? Or what sources will need to be used to build and maintain the data product?

Together, the answers to those questions enable us to fill out our Data Product template as follows:

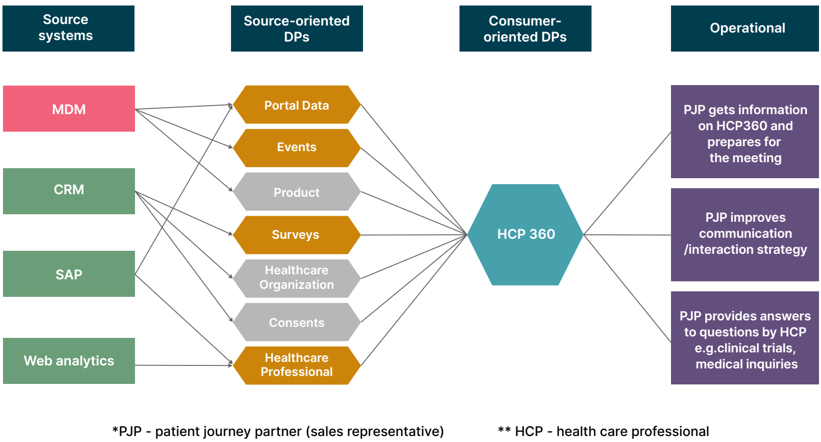


At Roche, we intentionally choose a 360-degree view data product because we encounter 360-degree data solutions very frequently. In this case, the HCP 360 Data product isn't just created by pulling together all data that's relevant to HCPs — every input is carefully considered and intentionally added to best meet the needs of our defined consumers.

For example, the first iteration of the product only included data on how HCPs responded to digital engagements, while the second iteration added vital information about recommendations and next steps. This iterative approach helped us build up a product that was extremely relevant and valuable for consumers, and served them with what they really needed.

Key artifact: The data product interaction map

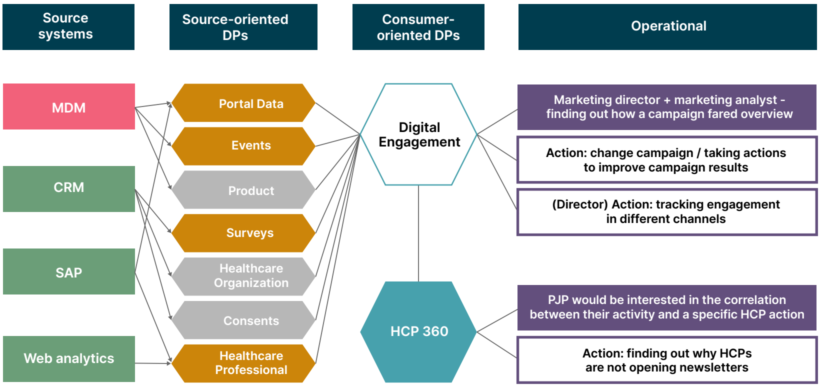
Once you've identified a collection of data products using the input templates, you can then start to draw out a data product interaction map, as shown below:



The data product interaction map clearly shows how data sources and integration sources feed into both source-oriented and consumer-oriented data products. But the most valuable aspect this map helps teams do is start to identify overlapping areas of data usage between prioritized data products.

Identifying this can help teams adjust the boundaries of their data products and make sure effort isn't needlessly duplicated, or even devise ways to unite potential data products to serve multiple, closely-linked needs.

Over time, multiple interaction maps, that feed into one another, can be brought together to create a single integrated data product landscape for a domain, as pictured below:



*PJP - patient journey partner (sales representative) ** HCP - health care professional

Here the HCP 360 data product is used as an input for the Digital Engagement Data Product. Using the integrated view we identified three data products (in yellow), that serve both use cases and whose boundaries can be logically merged.

The integrated data product interaction map provides us with an overview of all the foundational data products within a given domain. The map will evolve as new use cases are prioritized and onboarded into the Data Mesh, continuously giving teams a clear view of their data product landscape that they can use to make informed decisions about data product development or evolution.

Key practice #3: Defining clear SLIs and SLOs

In our experience, one of the most common reasons behind low data reusability is data simply not being available in the format different teams or use cases need. When we treat data as a product, we make conscious decisions based on how the data product will be used for each use case it serves, enabling high interoperability and reusability.

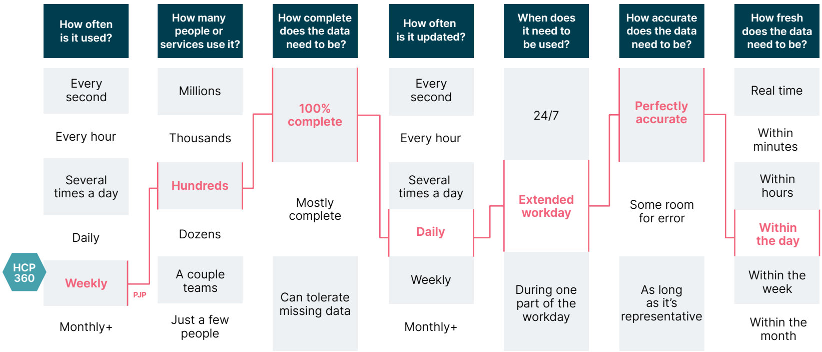
Before going any further, let's first break down a few key terms:

- Service-level objectives (SLOs) are the targeted levels of service, measured by SLIs. They are typically expressed as a percentage over a period of time. Eg. 99% availability over a three-month period
- Service-level indicators (SLIs) are the metrics used to measure the level of service provided to end users (e.g., availability, latency, accuracy)
- Error budgets are the acceptable levels of unreliability for a service before it falls out of compliance with an SLO

In Data Mesh, we use SLOs to make sure that individual data products work as expected. If outages or disruptions exceed the defined error budgets, that forces the product teams to check the backlog to improve the reliability or stability of the data product.

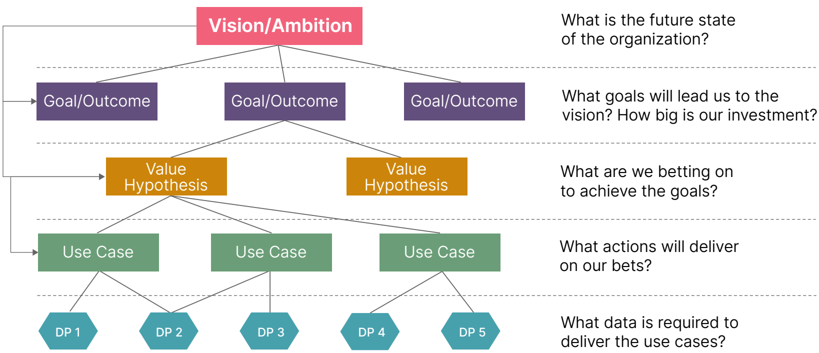
For example, an SLO of "99.5% of the transactions from previous day shall be processed before 9am every day" has an error budget of "0.5% of transactions missed to be processed per day" and this error budget can for example be set at, "2% of transaction missed per month". Should the error budget be exceeded or used up, it amounts to a violation of the SLO.

We use a discovery exercise called Product Usage Patterns to collectively brainstorm and understand how stakeholders wish to use a data product, and what their key expectations are for it. This enables us to determine the SLOs that need to be set for individual products.

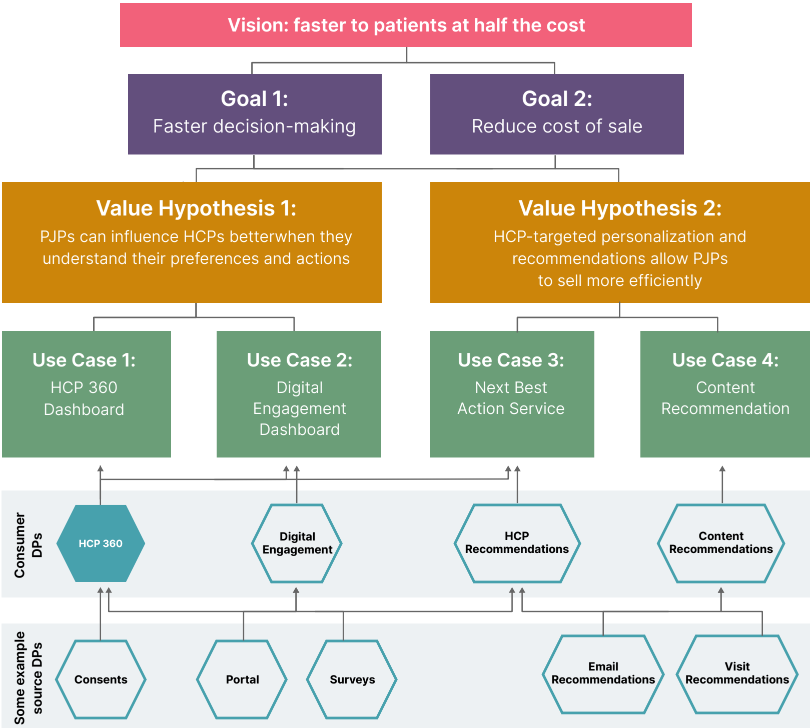


Key Practice #4 Mapping everything together - The Data Mesh LVT Extension

As a final step, we mapped Roche's identified data products to the domain LVT using the following Data Mesh Template:



For Roche, that looked like this:



*PJP - patient journey partner (sales representative)   ** HCP - health care professional

In terms of change management for Data Mesh, this is our most value creating and critical step. This tangible association of the data to one or more business goals, is what turns data into a data product - justifying its existence and clearly showing how it supports the domain and the wider business. Within this definition, it has measures of success, an owner, and a future roadmap. At this stage everyone is clear about the expected business outcome and their role in achieving it.

Depending on the goals of the domain, business decision makers can make a conscious decision about which goals they want to achieve, thereby helping them prioritize which data products to build.

The Data Product MVP Checklist
In addition to their hypothesized value and business purpose, for each data product, we clearly define the following criteria, to help everyone understand their purpose and value:

Minimum requirements for Data Products
Mandatory:

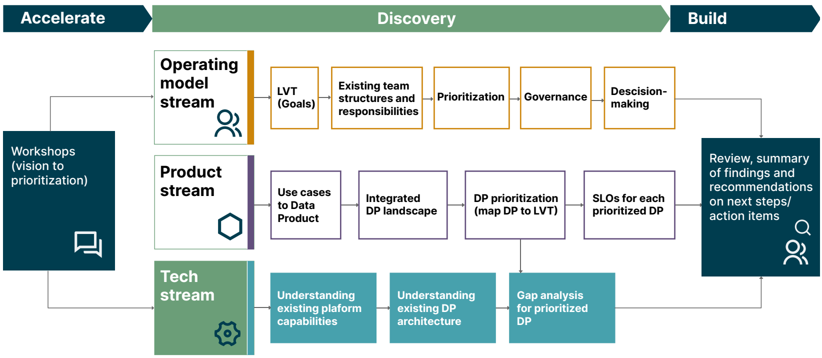
- ✓ **Owner / Steward** (first point of contact for the data product, approver of access)
- ✓ **Data Product name** (unique to the data domain)
- ✓ **Description** of the Data Product
- ✓ **Data sharing agreement** (published on a common marketplace catalog (e.g. Collibra))
    - ✓ “Open Access” or “Access Approval Required” (approval granted by DP Owner)
    - ✓ Published Data Access Policy: Define who is/isn’t allowed access to the data
    - ✓ Distribution rights: Whether modified (aggregated, filtered, merged) or unmodified data can be distributed to third parties by the consumer
    - ✓ SLO
- ✓ **Port** (a delivery mechanism for the Data Product)
- ✓ Data Product **type** (consumer-oriented or source-oriented)
- ✓ **Business Domain** (Business Function)
- ✓ Data Privacy, Classification and compliance (*mandatory only for regulated industries such as healthcare, banking etc.)

With the LVT, the Data Product Interaction map and the Data Product checklist all created, domains can move onto the final aspect of our discovery, and start making informed technology and architecture decisions.

In our final chapter, we'll look at those decisions, and walk you through some of the ways Data Mesh has helped organizations build a strong technical and architectural foundation for their Data Mesh operations.

Technology and the architecture

So far, we've looked at the operating model and product streams of our Data Mesh discovery process. Now it's time to turn our attention toward the technical stream, and look at the architectural decisions that organizations need to make along their journey to Data Mesh success.



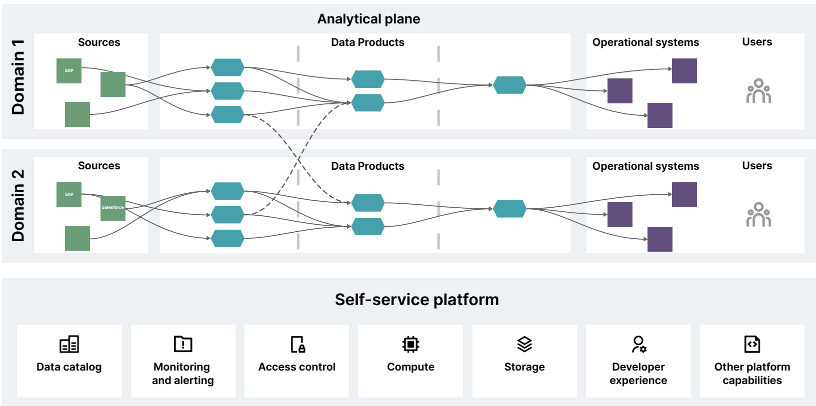
Key artifact: The Data Mesh Logical Architecture

Much like the operating model and product streams, the technical stream also has a very important output asset — a Data Mesh logical architecture, as shown above. This logical architecture maps out each domain's data products across the Data Mesh, and provides a clear overview of:

- Which domain owns and is responsible for which data products
- Which use cases are served by data products, including how different data products come together to support further use cases
- The users of each data product, showing how they consume data products using polyglot output ports

- How data products are consumed and what value-creating actions are taken based on the insights from consumer data products
- The operational systems that are the sources or system of records for data ingested into the data products
- The capabilities that make up the self-service platform that provides the foundation for the Data Mesh model.

Data Mesh logical architecture



Across the technical discovery stream, we work to define those points and build up that logical architecture in greater detail. In the technology stream of our discovery process, data engineers engage with the domain that's being onboarded to understand their existing platform capabilities and the scope of any data products they already have in place. That helps them identify the Data Mesh delta that will need to be bridged with new technology and architecture, and what new data products should look like from a technical perspective.

Throughout the discovery process undertaken with Roche, we took steps to align the team and our planned actions with a set of architectural practices and principles that we use to help create a consistent ecosystem of interoperable data products and build a strong foundation to help the Data Mesh evolve within the organization.

Key practice #1: Approaching data products as an architectural quantum

Data products are the fundamental units that make up the Data Mesh. Each one has its own lifecycle, and can be deployed and maintained independently. During our engagements, we've created an individual git repository for every data product, containing:

- Code for ingestion, transformation and publishing to output ports
- Sample data, unit tests, and data quality tests
- Infrastructure as code to provision data pipelines, CI/CD pipelines and other platform capabilities like storage, compute, monitoring configuration etc.
- Access policies as code that specify who can access the data products and how

Each data product is an atomic and functionally cohesive unit which, in our case, exposes a single denormalized data set via one or more output ports. It may have additional intermediate tables as an implementation detail of their pipeline, but ultimately publish one data set through its output ports.

One may wonder if this rule is too stringent to be applied to all consumer-oriented data products, many of which have to read from multiple data sets to meet their objectives. However, our experience shows otherwise. If we find a need to expose multiple data sets via output ports, this is a good indication that we should create a new data product instead.

Building the mesh with data products as its **architectural quantum** — the smallest unit of the mesh that can be deployed independently, with high cohesion and includes all the structural elements required for its function — is what makes the Data Mesh so robust.

Any given data product can easily be replaced or removed without affecting the system as a whole. It also makes it easy to reassign ownership of the data products to a new team as required, helping the mesh scale horizontally and evolve organically.

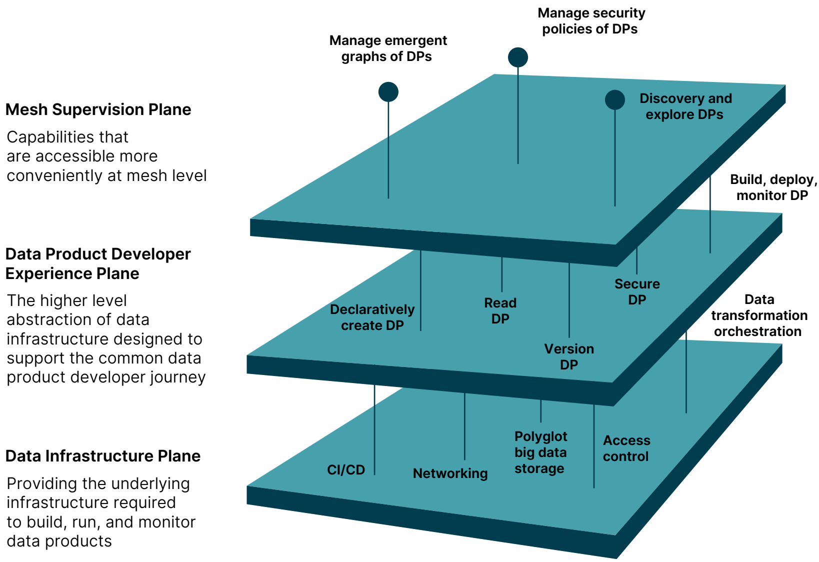
Key principle: Data products as atomic, functionally-cohesive units

Data products are the architectural quantum of the Data Mesh. They should be designed as the smallest functionally cohesive unit of the mesh, each with an independent life cycle. This is a foundational principle of Data Mesh architecture.

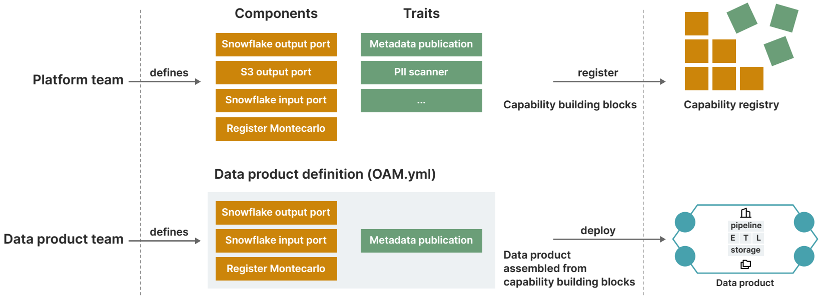
Key practice #2: Self-service data platform design
Within the Data Mesh, the data platform has multiple planes. One common mistake that many organizations make is only focusing on the data infrastructure plane when devising and constructing a platform. But for a Data Mesh implementation

to be successful, teams need to carefully assess and make the right decisions at two further levels: the data product developer experience level and the mesh supervision level.

The diagram below, taken from **Zhamak's original article** shows the components that form each layer.



Key practice #3: Create streamlined developer experiences
Removing friction around the creation and maintenance of data products is key to the success of Data Mesh. For the model to work, domains must be able to easily create their own data products. So, one of the top priorities when constructing platform architecture and defining how Data Mesh will be implemented is ensuring smooth and intuitive developer experiences.



To help, we've:

- Developed an OAM-inspired specification language that the product teams use to declaratively to specify their data products
- Enabled domains to build their own products using this domain-specific language. The platform provides the framework and tools which can read the data product specification and take appropriate actions via CI/CD pipelines.
- Developed and maintained a registry of capabilities to help everyone see what's available to them

Done right, the platform cuts lead time to create new data products, empowering developers across domains to focus on creation and maintenance of data products to add business value, rather than solving the same data engineering problems again and again. It also helps to codify blueprints and patterns for implementing standard functionality, making data products more consistent and interoperable.

Key practice #4: Establish consistent metamodels for data products

To ensure all data products are easily searchable and tracked so that they can be adequately maintained, we have also established a cataloging process for them. All data products are published to a common catalog (Collibra in our case) that's accessible across the organization. We created a **consistent metamodel** of a data product so that it means the same thing no matter which domain they are owned by. This is key for interoperability between data products.

The metamodel enforced certain mandatory attributes for data products like

- Name (unique within the data domain)
- Description of the data product
- Owner/ Steward (first point of contact for the data product, approver of access)
- Data Sharing Agreement
- "Open Access" or "Access Approval Required" (approval granted by DP owner)
- Published Data Access Policy: Define who is/isn't allowed access to the data,
- Data classification (public, internal, secret etc)
- Distribution rights: Whether modified (aggregated, filtered, merged) or unmodified data can be distributed to third parties by the consumer.
- SLOs and SLIs
- Port (a delivery mechanism for the Data Product)
- Data product type (Source/Consumer oriented)
- Linked to Business Domain (Business Function)

To further improve the developer experience, we developed and **innersourced** client libraries which could publish the data products using a REST interface that implements the above definition. The data product teams could use these platform capabilities via a declarative DSL as shown above to publish data products with minimal effort via their CI/CD pipelines.

Key principle: Create a consistent definition of a data product across the entire organization

Across a Data Mesh, teams are empowered to build and manage their own data products, in ways that best serve their needs. However, there needs to be clear guardrails and consistent definition of a data product (meta model) to ensure interoperability across the entire organization.

Key practice #5: Automate governance and Access Control Policies

As part of the developer experience, data product teams should be able to programmatically specify both human and machine user access policy rules. They should be able to employ both role based access control or attribute based access control techniques to achieve this.

The platform should support a data sharing workflow and automated execution of these policies with seamless integration between the corporate identity management system (system of records for roles) and the target data storage solution to grant appropriate permissions to the schema and tables.

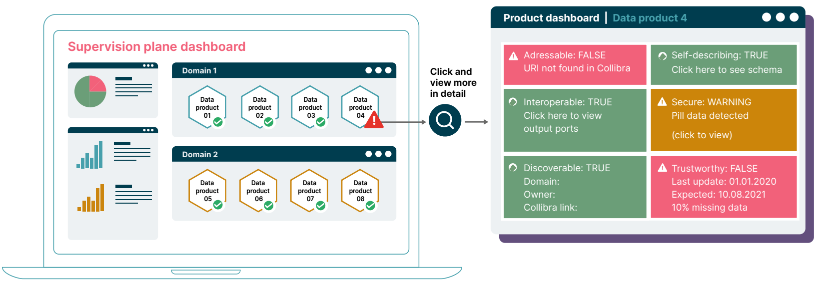
Several commercial tools exist which promise this functionality for a polyglot set of data storages. We are currently experimenting with a few; however, we haven't yet found one that's a perfect fit. There doesn't seem to be an out of the box solution available yet — commercial or otherwise — that meets the demands of programmatic policy authoring, federated ownership and polyglot storage of data products in a Data Mesh.

Most of the commercial tools we've seen seem to provide programmatic access (APIs) as an afterthought. It's an area that appears ripe for innovation, and one we'll be focusing on in the near future. [Extending Open Policy Agent](), with its Rego DSL to specify policies programmatically that supports common big data storage solutions, seems like the most promising direction forward that's in the spirit of Data Mesh.

This problem becomes a lot easier to deal with if you don't have to deal with polyglot storage across your organization. As an example, if your organization relies solely on AWS-native services, have a look at this [detailed solution architecture]() to find out how this can be achieved using AWS lake formation.

Key practice #6: Apply fitness functions to guide the evolution of the mesh

The supervision plane dashboard - monitors the six characteristics of the data products



Within a Data Mesh, every team is empowered to build its own data products. But, with teams across domains all working on their own use cases, what can we do to guide the evolution of the Data Mesh, and ensure that as it grows, the products within it remain interoperable and valuable?

To help answer that question, we lean on the idea of architectural fitness functions. We define automated tests for six characteristics of a data product that could be run centrally against all data products in the data catalog. These tests ensured that the data products that were created by autonomous domain data product teams are up to the organization's required standards:

- **Self-describable:** Automated check for mandatory publication in the data catalog, well described semantics, product description and syntax of data, ideally accompanied with example datasets
- **Addressable:** Check for a unique URI that represents the location of data set owned by the data product
- **Discoverable:** Automated checks to ensure the data product is published and searchable in the catalog and the marketplace where discovery happens.
- **Secure:** Check that access to data products is blocked by default. Checks to ensure PII has been sanitized.
- **Trustworthy:** Check that the SLOs and SLIs are published in the catalog. Check for adherence of certain SLOs for. eg. refresh rate.
- **Interoperable:** Automated checks to ensure that standard output ports and standard file formats are supported

These tests weren't designed to be comprehensive, but rather a starting point for making these architectural characteristics visible and incentivizing teams to follow the required governance principles. The results of these checks were made available in an easily accessible organization-wide dashboard. This served as an important incentive for data product teams to play by the rules as no team likes to see their data products showing up as red.

Key practice #7: Provide clear guidance (or patterns) for data sharing

The federated architecture of Data Mesh, and the polyglot storage used across it, makes enabling data sharing between teams one of the tricker and more nuanced challenges of building a high-value Data Mesh. Although data virtualization options are improving by the day, the technology isn't quite there yet, with three significant problems persisting:

1. Virtual tables are leaky abstractions. In our experience, for most non trivial use cases, you still can't get away without having to worry about source representation of the data.
2. They're still generally very slow, with query performance bottlenecked to the speed of the slowest source.
3. They aren't really built for programmatic usage. Workflows for creating, altering, governing the virtual tables remain heavily UI dependent, making them harder to test and enable continuous integration and continuous delivery with.

With that said, data virtualization is largely good enough for most reporting needs. However, If **data locality** is important to you — eg. If you're training a machine learning model over a massive data set — virtualization is not going to work.

Across our engagements, we apply the following guidance and patterns to help us define how data sharing is executed across an organization:

1. Data products can expose Virtual DB as an additional output port. For most simple reporting type use cases, this is sufficient and no further capabilities are needed.
2. For more advanced use cases, and when the producer and consumers are on similar storage platforms, always use the underlying native mechanism of the storage platform for

sharing data. For example, **Redshift data sharing** or the native **data sharing capability in snowflake**.
3. If you need data locality and your producer and consumers are on different storage platforms, you probably can't get away without copying over data. This is the least desirable option and should be avoided wherever possible. If you do need to do this however, consumers should exercise extra caution to ensure that governance and access control policies are preserved throughout.

Although data virtualization holds a lot of promise and there's much to look forward to in this space, there is a dangerous tendency to equate data virtualization to Data Mesh. That's at least partly due to some intense marketing from the data virtualization platforms who want to cash in while hype for Data Mesh is high.

Data virtualization is an interesting solution to a specific problem that arises in a federated architecture. The technology is still maturing, and we believe the increasing adoption of Data Mesh is going to expedite advances in this technology. However, there is still some way to go before it can be recommended as a default solution.

Key principle: Defining and building your own path

One important thing to note about the technology part of adopting Data Mesh is that many of the tools required to build federated data architectures don't yet exist. As the adoption of Data Mesh grows, new tools and frameworks will emerge. But until then, adopting Data Mesh will take you into new territory, creating opportunities to define your own path forwards and demonstrate innovation and thought leadership.

Conclusion

Organizational change, product thinking, and technology — the three pillars of Data Mesh success

Around Thoughtworks, you'll often hear Thoughtworkers saying, **"Data Mesh is not all about the technology"**. When the model was first gaining traction, that statement served an important purpose — it helped prevent Data Mesh being seen as just another data platform or architecture.

Today, with lots of practical implementation experience, our thinking has evolved a little. Data Mesh is about technology — but it also needs to involve a lot more. To successfully bring your vision for Data Mesh to life, you need to lead organizational change, embrace product thinking, make the right technology decisions, and ensure all three evolve in harmony.

For any Data Mesh implementation to be successful and deliver its intended value, it's paramount that organizations begin by clearly defining their 'Why' and 'What' — the things they're trying to achieve, and what they want to build to help achieve it. But defining those things alone isn't enough. Teams also need to find ways to measure how effectively their efforts and hypotheses are achieving them — enabling the team to course correct on a regular basis, and experiment to find the best route towards their 'how'.

By taking an iterative, value-based approach to the entire initiative — one that applies the EDGE operating model — teams can work back from their vision to the technology and architecture required. This approach ensures that whatever an organization's Data Mesh and its underlying architecture end up looking like, what they deploy will deliver exactly what they want, and drive value in meaningful ways for multiple domains.

That's the approach we take at Thoughtworks, and it's been fundamental to our leading Data Mesh success stories.

It's a challenging undertaking, but the rewards for organizations that get it right are huge. Implemented correctly, Data Mesh has the potential to empower domains, improve data utilization, support future growth, and enable organizations to get new value from data.

If that's something you're interested in, and you'd like some expert help to bring your Data Mesh vision to life, **talk to us today**.

About the authors

Ammara Gafoor - Principal Business Analyst

Ammara joined Thoughtworks as a Business Analyst in 2018. In this role she primarily holds leadership positions in the healthcare digital transformation industry tapping into her decade long experience in software engineering, biotechnology and business management. Since 2020, she has the added responsibility of being a Delivery Principal for some of our client accounts.

Before Thoughtworks, Ammara ran her own company (analytics for HR professionals) and worked in leadership roles in the healthcare IT sector.

She is passionate about breaking stereotypes and increasing the representation of Hijabi women in the technology industry. She is currently an Advisor for the Qatar Women's Startup and Incubator Program. She also volunteers for the "girls - let's dream big initiative" and supports the educational journey of several young girls.

Ian Murdoch - Principal Advisory Consultant

Over the past 20+ years Ian has been working on moving technology closer to the core of business. As CIO and CTO in the manufacturing and automotive industries, he has had the opportunity to shape organizations, products and technology in strategy and execution.

At Thoughtworks Ian focuses his efforts on responsive operating models and helping our client's leadership teams become brilliant digital executives.

Ian loves IoT and analytics and has been fortunate to work on the topics in every role he has held in the past 10 years.

Kiran Prakash - Principal Engineer

A Principal Engineer with over 17 years of experience in helping our clients build custom software, Kiran is an avid extreme programming practitioner with expertise in TDD, refactoring, CI/CD and pair programming techniques.

Although he has a full stack background with experience in many different technologies, he's been focused on data engineering in the recent past. He helps our large, strategic clients leverage data for their business.

He enjoys helping businesses with their digital transformation and likes the challenge of scaling systems while keeping the complexity under check.

About Thoughtworks

Thoughtworks is a global technology consultancy and community of passionate purpose-led individuals, 11,000+ people strong across 49 offices in 17 countries. Over 25+ years, we have helped our clients solve complex business problems by integrating strategy, design and engineering to drive digital innovation.

For more information visit: **thoughtworks.com**

Get in touch with us
contact-de@thoughtworks.com
Thoughtworks Deutschland GmbH
Caffamacherreihe 7
20355 Hamburg
Germany

+49 (0)40 300 95 880
thoughtworks.com

/thoughtworks