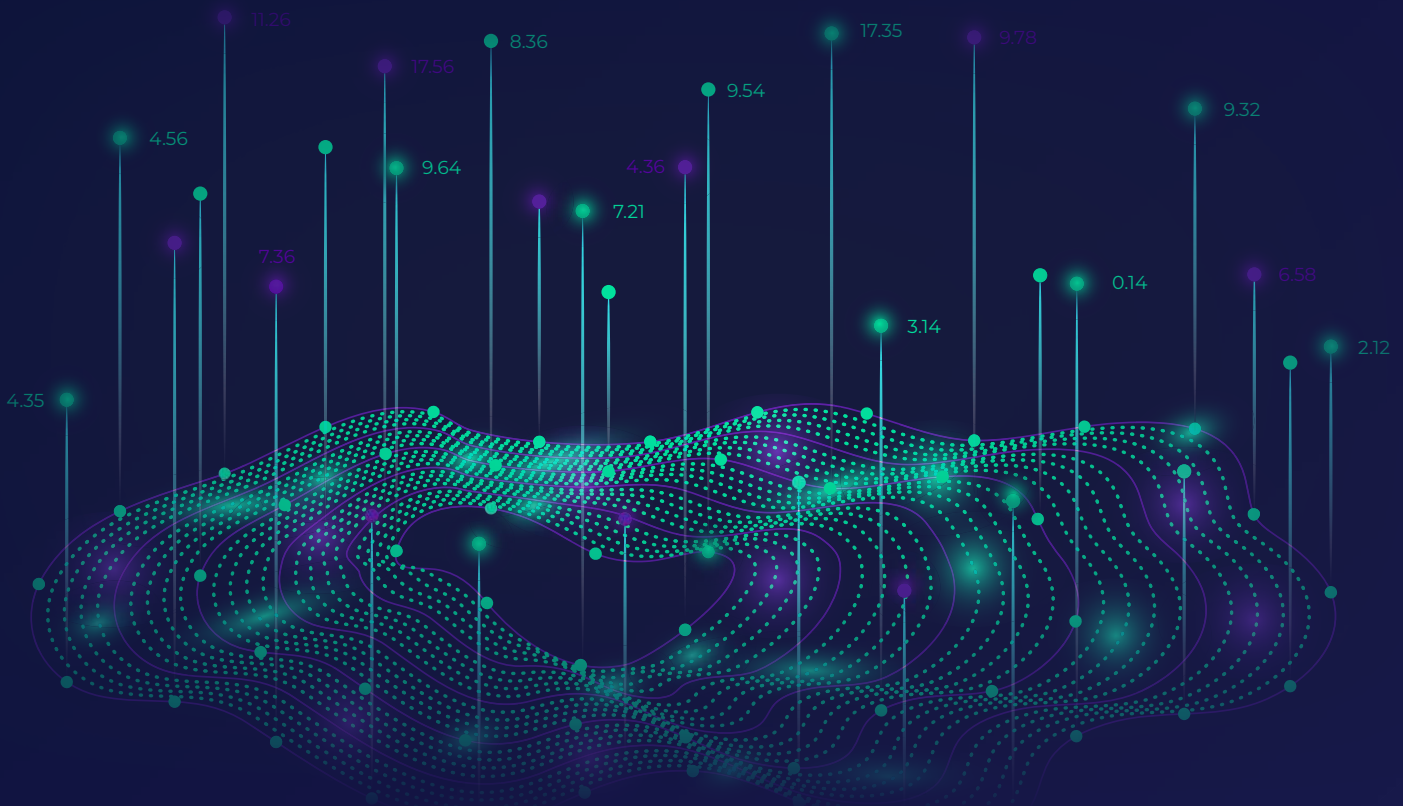




ER/STUDIO
An Idera, Inc. Company

DATA MESH AND DATA PRODUCT DELIVERY

Opportunities, Challenges and How to Succeed



By Hans Lux, the leading expert and innovative pioneer of enterprise data architecture, responsible for creating cutting-edge data solutions, designs and strategies.

Executive Summary

Currently the most popular topic in analytic data topic, the Data Mesh ("DM") is a potential game changer. A successful DM

- Greatly improves
 - User experience and data quality
 - Support for AI, for highly distributed architectures and for agile delivery
- Alleviates bottlenecks due to monolithic architectures and centralization

The DM is loosely presented as a vision, as a "sociotechnical paradigm". For success, this vision needs to be interpreted and streamlined. The gaps need to be filled in. It needs to be combined with best practices. Pitfalls need to be avoided – some specific to the DM and some typical for large data projects in general.

This white paper summarizes the most important points to consider for a practical and successful DM implementation.

Introduction

The Data Mesh¹ ("DM") is currently the most popular topic and approach to the delivery of analytical data. It seeks to

- Greatly improve data quality by applying product thinking to data – creating "data products"² with measurable quality criteria, that are easy to understand, use and combine ("interoperate") with one another.
- Further improve quality and alleviate the bottlenecks of centralized teams by putting the people who know "their" data best, the business domains, in charge of analytical data³.
- Support the domain teams and shield them from responsibility for general-purpose functionality by means of a self-service data platform⁴.
- Keep the governance necessitated by this decentralization lightweight and mostly automated⁵.

All of these are sound ideas; few practitioners would take issue with them. Aimed at scale, the principles are nevertheless worthwhile to be considered for analytical data environments in general.

Clearly this is a potential game changer that may solve some of the most difficult analytic data problems and that could deliver recognisable business value

¹ The Data Mesh "manifesto" is Zhamak Dehghani's book Data Mesh: Delivering Data-Driven Value at Scale (O'Reilly, March 2022). She laid out the main points in 2019 <https://martinfowler.com/articles/data-monolith-to-mesh.html>

² Principle 2: Data as a Product: Chapter 3, 9 and 11

³ Principle 1: Domain-Ownership: Chapter 2

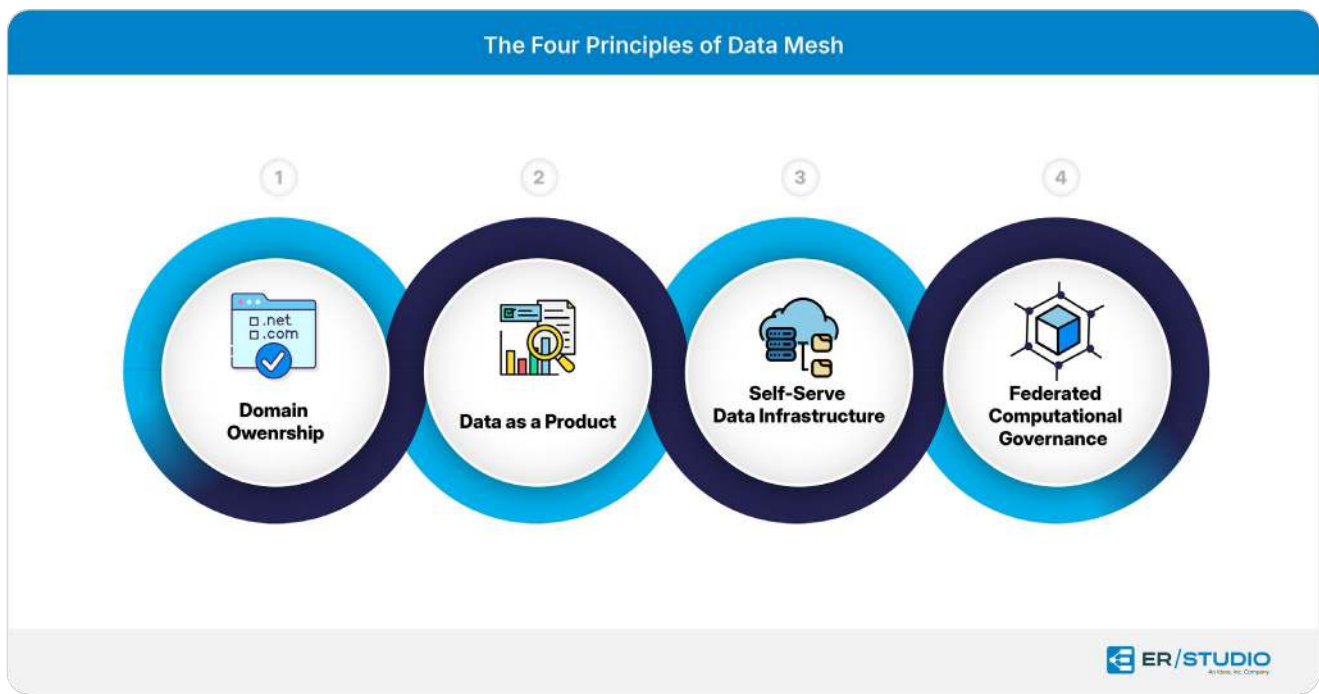
⁴ Principle 3: Self-Serve Data Platform: Chapters 4 and 10

⁵ Principle 4: Federated Computational Governance: Chapters 5 and 14

- Much better data quality
 - More understandable thanks to much better documentation
 - SLA and clear responsibilities (“ownership”)
- Much improved user experience
 - Data are easy to find, use and combine (interoperate) with other data
 - Self-service work supported by platform
- Better support for AI
- Alleviation of bottlenecks that can occur with monolithic approaches and centralized specialty staff
- Better support of highly distributed architectures than monolithic Data Warehouses, Data Lakes and Lakehouses
- Support for agile delivery thanks to small and manageable data products

In short, a silver bullet!

But pulling a DM project off successfully is challenging: the DM is presented loosely as a vision, as a “sociotechnical paradigm”. While other DM resources fill in some gaps, most take this vision at face value - well-reasoned critical opinions are rare. There is little long-term experience as to what works sustainably and what doesn’t. It is up to a competent and diligent expert to interpret, augment and fill in gaps and – most importantly – avoid the numerous pitfalls specific to the DM on the one hand and typical for large data projects on the other.



Understanding and Methodology ("You need to know what you are doing")

This first set of challenges pertains to the DM itself, whose concepts must be understood, evaluated, in part fixed, augmented with best practices - in particular an effective approach to data modelling and data element management - to be successful.

1. Not understanding the DM

Problem: The Manifesto and DM resources are hard to understand due to confusing language and repetitive structure.

Only very few appear to have studied the manifesto back-to-back, working instead with shortcut interpretations based on the most salient points, but not addressing the problems of the original. Just for example, "Data Product" refers to a quality data resource in most cases, but also to the factory providing such resource. "Ubiquitous language" is not ubiquitous but, and a "data contract" is not a contract.

Impact: Without understanding what the DM is, how can its suitability be evaluated, let alone an implementation be considered?

Fix: Don't get hung-up on language, fix it. Resolve homonyms, revise misleading terminology and eliminate unnecessary wording. Restructure text and prune repetitive sections

- Can be done by oneself, assuming good reading comprehension and time
- Competent help may be engaged
- Consolidated resource may be licenced

Benefit: Understanding what the DM was supposed to be in the first place is a starting point for a suitability evaluation.

2. Implementing flawed concepts and ideas rather than fixing them beforehand

This section touches only on the most important flawed concepts, the incomplete yet overlapping Data Product ("DP") types, the "one-size-fits-all" approach to data sets and the "polyseme" based interoperability. Related are the issues of Sections 3 Ignoring or violating best practices and 4 Failure to implement an optimized modelling approach

Problem 1: The three Data Product types are incomplete yet overlapping

- The scope of "Source-aligned" DP is unclear
 - It could contain minimally processed source data that match structure, granularity and content of the source systems
 - It could be further processed for interoperability, temporality, standardization

It can't be both

- The "aggregate" – better called "derived" - DP type consists of several very dissimilar subtypes, one of which corresponds to the second meaning of "source-aligned"
- A DP may include facets of multiple DP types

Impact 1: It is not clear from the oversimplified and singular types, what the DP really is, and what rules should be applied. This results in confusion, endless yet fruitless philosophical discussions and many exceptions.

Fix1:

- Don't stick slavishly to the DP types as proposed – it won't work
- Split source-aligned DT into two types
 - "Source" – corresponding to "landing" in traditional environments
 - Accept that operations teams will deliver what can be easily provided from source applications
 - Treat such data as an interim resource not intended for general use with technical quality guarantees only
 - "Harmonized" – corresponding to ODS or Data Vault content
 - Normalized, non-redundant, with temporality
 - High data quality guarantees, well documented
 - Most likely created and maintained by the domain's DM team
- Split the "aggregate" (better named "derived") DP type into subtypes
- Use a multidimensional approach, e.g., "this is a consumer-aligned DP, which also has a source and a derived component."

Benefit 1: DP can now be accurately defined and rules applied without exceptions and endless discussions. Staff is optimally deployed and responsibilities are clear.

Problem 2: The market paradigm is forced on data resources, for which it is not suitable.

Impact 2:

- The DP paradigm works best where there is a true market for data with multiple vendors competing on scope, quality and price, e.g.,
 - Market data feeds
 - Weather
 - News events
- There is no market for structured data from single in-house sources, which must be provided to those who need them – usually for free – and denied to those who don't. For example, there ought to be only one legitimate source for HR data, provided for free when needed and denied otherwise. But there could be multiple data marts with overlapping coverage, where a market approach might be considered.

Just as market theory does not apply to public goods, market and product metaphors don't quite apply to these DP resulting in poor decisions and allocations.

Fix 2: “Horses for Courses” - enhance with a more differentiated approach and apply the market idea only where suitable.

Benefit 2: By taking advantage of market concepts where it is beneficial and avoiding them, where it is not the best results are achieved

Problem 3: Free-standing DP are most suited for largely independent resources such as sensor-data, textual resources, data available on markets, non-structured data as well as data marts and well-defined, standardized data feeds.

Impact 3: DM data are typically highly structured, come from single in-house sources. They are at best components but not products – effectively replacing data feeds into an ODS or Data Vault. They are usually consumed as a whole like a car or a symphony performance and not as individual parts like a steering wheel, engine, or the individual part played by a musician. The product metaphor causes endless discussion and focus on irrelevant activities.

Fix 3: Again: “Horses for Courses” - enhance the approach with highly structured internal single-source data → harmonized data product.

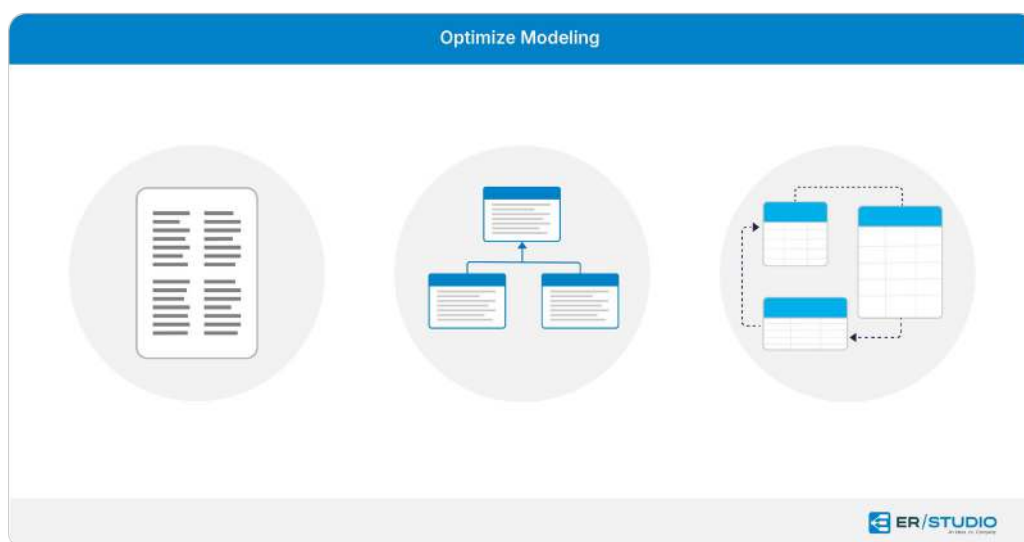
Benefit 3: By providing free-standing DP where it is beneficial and avoiding them, where it is not the best results are achieved

Problem 4: The proposed approach to interoperability is haphazard relying on “polysemes”, where apparently a name-match indicates the existence of an implicit foreign key.

Impact 4: This is a big step back from the status-quo in data warehousing - a bunch of independent data sets loosely connected via “polysemes” is not an analytical environment. Instead of having a guaranteed way to join data, the analyst wastes time on guessing and testing possible joins from polysemes.

Fix 4: Put most efforts on creating a family of harmonized data products with guaranteed interoperability by means of defined foreign key – primary key pairs. From this resource, everything else will be provided (see model kernel below).

Benefit 4: No time wasted for guessing and testing, avoiding errors, avoiding costly erroneous decisions due to poor analyses.



3. Ignoring or violating best practices

Problem: Asserting a crisis in data and an inflection point with a serious impact on the success and survival of organizations, the Data Mesh is presented as a radical paradigm shift. A good portion of best practices is ignored or rejected outright. Data problems have been with us for a long time, however, and will likely continue to do so. “T’was always thus”; neither is there a need for a paradigm shift nor is it productive to assume so.

Impact: Throwing out proven best practices and starting over will potentially result in disaster in the form of massive cost-overruns and greatly reduced benefit.

Fix 1: Take advantage of the past, don’t reject it willy-nilly. Work with and within and not against the well-honed analytic data journey:

- Well-defined boundaries
 - Separation between analytical and operational environments
 - Stable and efficient pipelines/interfaces⁶ between them
- Landing area, e.g. lake for
 - Raw structured data with minimum processing – effectively source product as they are typically implemented. Will need to be processed further to be useful.
 - Textual and video, image, signal data. Useful, as is; usually no need for further processing, just accessed from there (lake-portion of lakehouse)
- Harmonized layer with temporalization
 - Simplified data vault approach (effectively enhanced house-portion of lakehouse)
 - Normalized, non-redundant, integrated, very high quality (consistent with Inmon’s operational data store)
- Data marts, etc.

Fix 2: Understand where the DM improves on best practices – either augment the DM with best practices or incorporate DM improvements into the current estate.

Benefit: Working with, rather than against, best practices yields by far the highest likelihood of delivering a successful and sustainable analytic data project. The result may

- Be a fully-fledged Data Mesh that delivers on its promise
- Improve a more traditional estate by incorporating Data Mesh principles such as product thinking or federating to domains

4. Failure to implement an optimized modelling approach

An **optimized data** model is essential for effective data work, for understanding semantics in context and for data governance. An intentional *enterprise* model is difficult to pull off due to size, the large number of people potentially involved and the ever-dwindling number of competent modelers.

⁶Check the term to keep it DM compatible

DM Principle 1, consistent with Domain Driven Design (“DDD”) ⁷, on which is based, pushes modelling from the enterprise to the domains. While this breaks the task into manageable chunks, further challenges need to be dealt with.

Problem 1: What “good” means in regard to modelling is hardly understood let alone consistently practiced; the DM offers no help. This is especially critical for the DM because domain-driven modelling puts staff in charge of modelling who lack modelling skills and experience.

Impact 1: Poor models don’t create much benefit because they cannot be used to drive and automate subsequent work. The domain models must be of high-quality and be very consistent for the DM, in particular the self-service data platform, to deliver on its potential.

Fix 1:

- Develop a simple, strong and effective modelling approach
- Adopt strong and effective modelling standards
- Keep models simple and tight through optimized abstraction
- Provide reusable model components and artefacts
- Incorporate into the self-service data platform the best data modelling tools that you can get your hands on
- Provide effective modelling training and support on an ongoing basis.

Benefit 1: DM on steroids – consistent high-quality domain models drive DP delivery via the platform fast, at high quality and at great performance. Happy and productive domain teams, whose members can be rotated between domain teams for flexibility and capacity management.

Problem 2: DM proposes point-to-point mappings – i.e., translations - between domain models to keep them interoperable.

Impact 2: Point-to-point cross-model mappings work just as well as point-to-point data feeds, which is... not well at all.

- Translation-based interoperability is brittle and error-prone
- Many mappings are necessary, which creates a new bottleneck and is costly to maintain.

Fix 2: Use the most suitable option proposed by Evans – the shared kernel

- Keep most modelling in the domains
- Use a small hub-and-spoke jointly-maintained kernel for shared classes, concepts and the most essential⁸ data
- Guarantee reliable joining across DP through foreign-primary key pairs in the kernel
The enterprise-wide model emerges from the kernel, which is the unifying core together with the

⁷ Eric Evans’ 2003 Domain-Driven Design: Tackling Complexity in the Heart of Software (“DDD”)

⁸ Typically, no more than 1000 DE total

domain models that line up to the kernel and provide the detail.

Benefit 2: Most modelling remains in the domain, where it belongs. Only the comparably small kernel requires coordination and cross-domain agreement. The kernel also provides for rock-solid interoperability (quality criterion from Principle 2) by defining and guaranteeing foreign-primary key pairs, through which data from different products can be reliably joined.⁹

Problem 3: DM is rather derisive regarding canonical¹⁰ models and a single source of truth^{11, 12, 13}.

Impact 3: Not dealing with these concepts effectively hurts successful DM delivery:

- Principle 2 puts a major emphasis on understandable data both in terms of semantics and syntax. Without a good model, whether conceptual or ontology-based, concepts aren't grounded and understandable in the context of other concepts. Semantics are limited to free-standing language descriptions which only succeed with an iron will and meticulous use of language both in short supply in modern organizations.
- Principle 4 – regardless of whether it is computational or traditional – data governance requires a clear definition of the governed meta objects; the data classes and the data elements. This requires an enterprise-wide context.

Whether this is intended or not, in both cases the model need not be huge, but it will end up canonical and limited to a “single source of truth”.

Fix 3: Don't fight the enterprise-wide model, with an effective modelling approach and proper data governance, for the harmonized core data, a canonical model emerges from domain-models and the kernel anyways. Enjoy the freebie – there is no need for intentional enterprise modelling:

- The kernel is the unifying core of the enterprise model
- The domain models (or the portions that the domains mark as essential for the enterprise, respectively) line up to the kernel
- Together they form the enterprise model, and as long as there is effective data governance,
- this will also be a canonical model.

Benefit 3: This is the best of all worlds, it provides the benefits of domain-driven models, which are close to the business and fast to adapt with the benefits of a small canonical model of high quality.

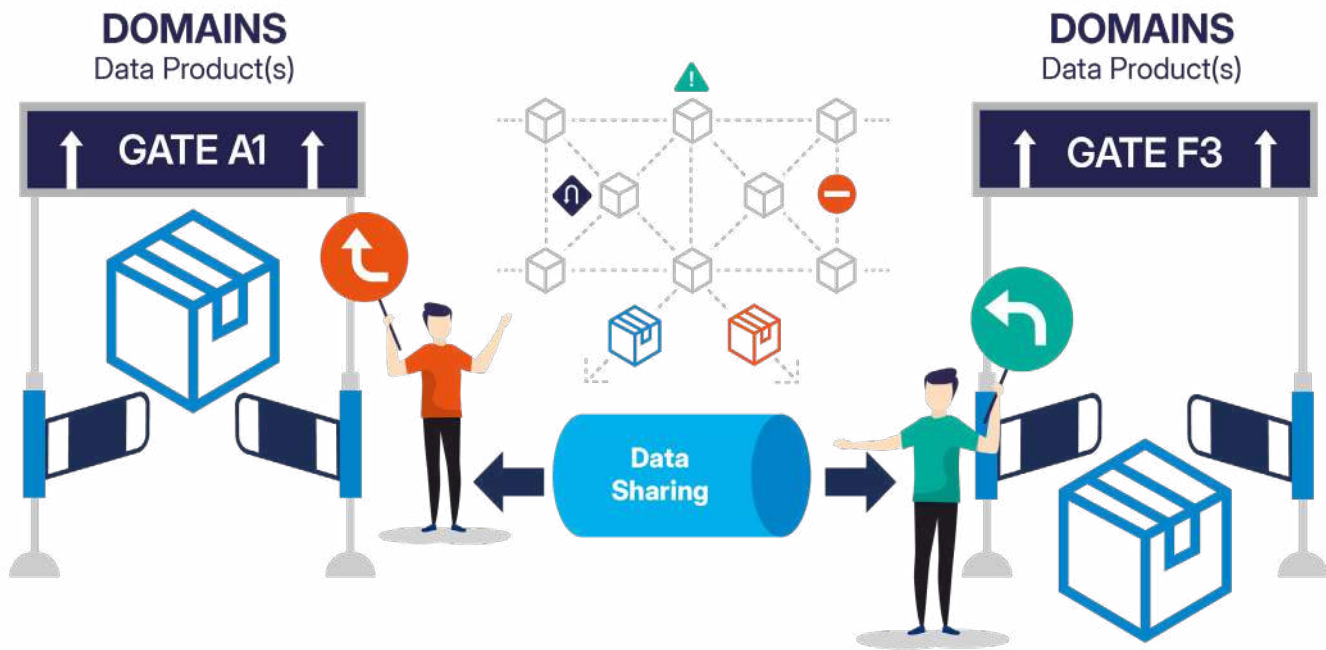
⁹ Depending on how many non-key data elements are maintained in the kernel, it also supports vertical interoperability where data from different sources can be combined via unions.

¹⁰ “one canonical model holy grail”, ironically termed “it's a wonderful idea.” pg. 23,

¹¹ Also ironically: “This is a wonderful idea” pg. 26, “the ideology of a single source of truth” pg 26

¹² “data myths such as a single source of truth, or one tightly controlled canonical data model” pg 105

¹³ “the concept of the mythical single source of truth.” Pg 216



5. Failure to Focus on Quality

Quality and trustworthy data that are easy to find, understand – semantically and syntactically –, use and combine are the rationale for applying product thinking to data in the first place (DM Principle 2).

A long chapter (3) describes, in great detail, the non-negotiable baseline usability characteristics¹⁴ of data. Most of these result from effective design and concern quality metadata or depend on them.

Problem: Quality metadata takes effort and competence to produce and maintain; the temptation for cutting corners is often overwhelming under project pressure. Not surprisingly, the Data Contract – typically a YAML file for expressing and sharing metadata is limited to the bare minimum and primarily technical parameters.

Impact: The provided information is insufficient for understanding the contents and working with it efficiently and effectively. The “savings” of this minimalist approach are lost manifold as analysts keep muddling through data.

Fix:

- Address Pitfall 4: Failure to implement an optimized modelling approach

¹⁴ In the original order, these are: discoverability, addressability, understandability, trustworthiness, native accessibility, interoperability, independently valuable, security.

The nouns get in the way, though, and the respective importance is far from equal. For efficient and effective work with data the analyst must

1. Find data in the first place
2. Understand what data mean in terms of semantics and syntax
3. Trust their quality, such as completeness, accuracy, etc.
4. Be able to combine data with other data

Not quality criteria, but obviously indispensable, are security, being able to access the data and proper addresses. “independently valuable” is relevant only in some cases and pointless in others.

- Make sure everyone understands, throughout the project, why a metadata-driven approach is essential for success.
- Stay on message in every meeting, in every communication: Quality data are the reason for the DM in the first place and quality metadata and models are essential for driving it
- Don't assume that initial agreements hold – work to keep them alive
- Stay on top of politics

Benefit:

- The optimized modelling approach will provide the quality metadata needed or driving the DM
- The emerging enterprise model will produce
 - the context, within which data classes and data elements can be easily understood
 - the kernel, which will guarantee that data can be easily and reliably combined (interoperability)
- Staying on message makes it least likely that the project goes under in unmanaged chaos
- Staying on top of politics helps avoid the sinking of a disciplined approach by an unexpected sniper attack

Focus and Priorities

6. Primarily technical implementation

Problem: Many DM projects are IT and technology-driven with the primary focus on the delivery platform, often with a move to new technology and to the cloud. Much less consideration is given to what data are really needed, rather than wanted, by the business and to ensuring how data and metadata should be organized and processed to assure quality, ease of use and to limit faults and cost. The social issues due to decentralizing work (to people who already have a full-time job) are considered at a late stage.

Impact: At best, much money is spent on an often brilliant but also costly technology foundation, which may end up far beyond needs. More likely are late-state surprises:

- The platform may fail to address critical needs, which require costly and sub-optimal emergency fixes
- The domains refuse to take on the additional work unless further improvements are made
- Modules that have been built ad-hoc, “just to demonstrate” the approach are near impossible to improve because other modules are already dependent on them

Fix:

- Optimize the conceptual data architecture first
- Test by designing and creating a handful of core data products end-to-end
 - Collaborate closely with the domains and as soon as possible driven by the domains

- Iteratively improve the design until all DM quality requirements are met
- Optimize
- Standardize
- Ensure extensibility
 - Implement using existing technology
 - Iteratively improve the modules until they work satisfactorily
- Evaluate
- Improve design
- Rebuild
- Restart the cycle
- Plan out all harmonized data products for the first two build-out phases including estimated volume and usage metrics
- Delay technology decisions to this point because only now can technology needs be properly assessed
- Evaluate existing as well as the shiny new technologies - using what is paid for and known may be a good stepping stone or perhaps even a cheap permanent solution

Benefit:

Greatly reduces unnecessary cost and risk because

- The implementation stays clear of both over and under-engineering
- Unneeded or unused components won't be built
- Late-state surprises, emergency fixes and refits can be avoided

Very often, the Data Mesh can be implemented at a much cheaper platform or even at existing technology.

7. DM may be poor match for needs of the organization

Problem: Organizations engage in DM implementations without ascertaining first that it is even appropriate and beneficial.

Impact: wasted time and resources for no benefit

Fix:

- Identify needs and pain-points
- See what portions of the DM make sense
- Take care of prerequisites first:
 - Pitfall 1: Not understanding the DM
 - Pitfall 2: Implementing flawed concepts and ideas rather than fixing them beforehand

Benefit:

- Avoids the waste of implementing a full-blown DM for organizations that won't benefit from one
- Identifies DM facets, from which the organization will likely benefit

8. Biting off too much and unfocussed implementation

Problem: Many DM projects planned or underway try to bite off far too much at one time

- Multiple topologies
- Multiple data journey types
- Too large a data scope
- Overly complex technical approaches

Impact:

- Big, lengthy and costly projects
- Potentially poor delivery or project failure

Fix:

Tightly limit scope and do what you do well.

- Design, plan and implement
 - Once
 - In one way
 - In one place
- Focus on essential core data, bring additional data in only after a successful initial delivery
- Fight feature and scope creep

Benefit: predictable, focussed and successful implementation much more likely

WHAT ARE THE NEXT STEPS?

To explore how ER/Studio creates an accurate, efficient data architecture using the data mesh approach, visit [erstudio.com](https://www.erstudio.com) to schedule a personalized product demonstration.

Talk To An Expert →