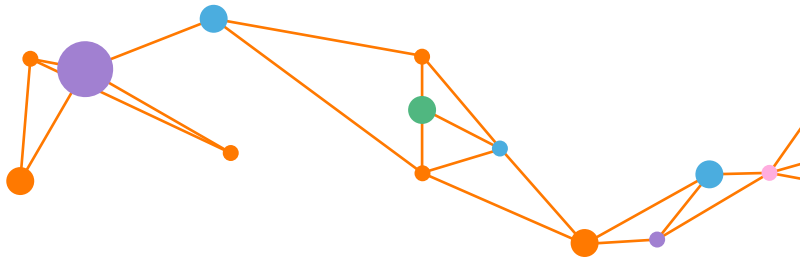
Introducing data mesh

A future-proof approach to managing company-wide data at scale

orange™ Business Services

A data-driven world requires cultural and technical change



Enterprises understand the importance of being a data-driven organization. The benefits of intelligence harvested from big data, include hyper-personalization, smart decision making, new business opportunities and faster innovation. But it isn't as easy as it sounds.

Many enterprises have invested in data platforms, especially large, centralized data lakes, to achieve the data-driven dream. Many, however, have been disappointed by the results. Data lakes don't scale well to meet changing organizational and process requirements. In addition, there is often a lack of alignment between the data lake creators and business teams, making it difficult to get any tangible value. Data lakes also hold data in a host of formats, which makes it a colossal task to make them available for usage, while keeping the quality at a necessary level.

It is also important to note that becoming a data-driven organization requires cultural change in addition to technological implementation. Shortcomings in organizational culture have been the main stumbling blocks to being successful in the digital age.

Data mesh: the next data platform

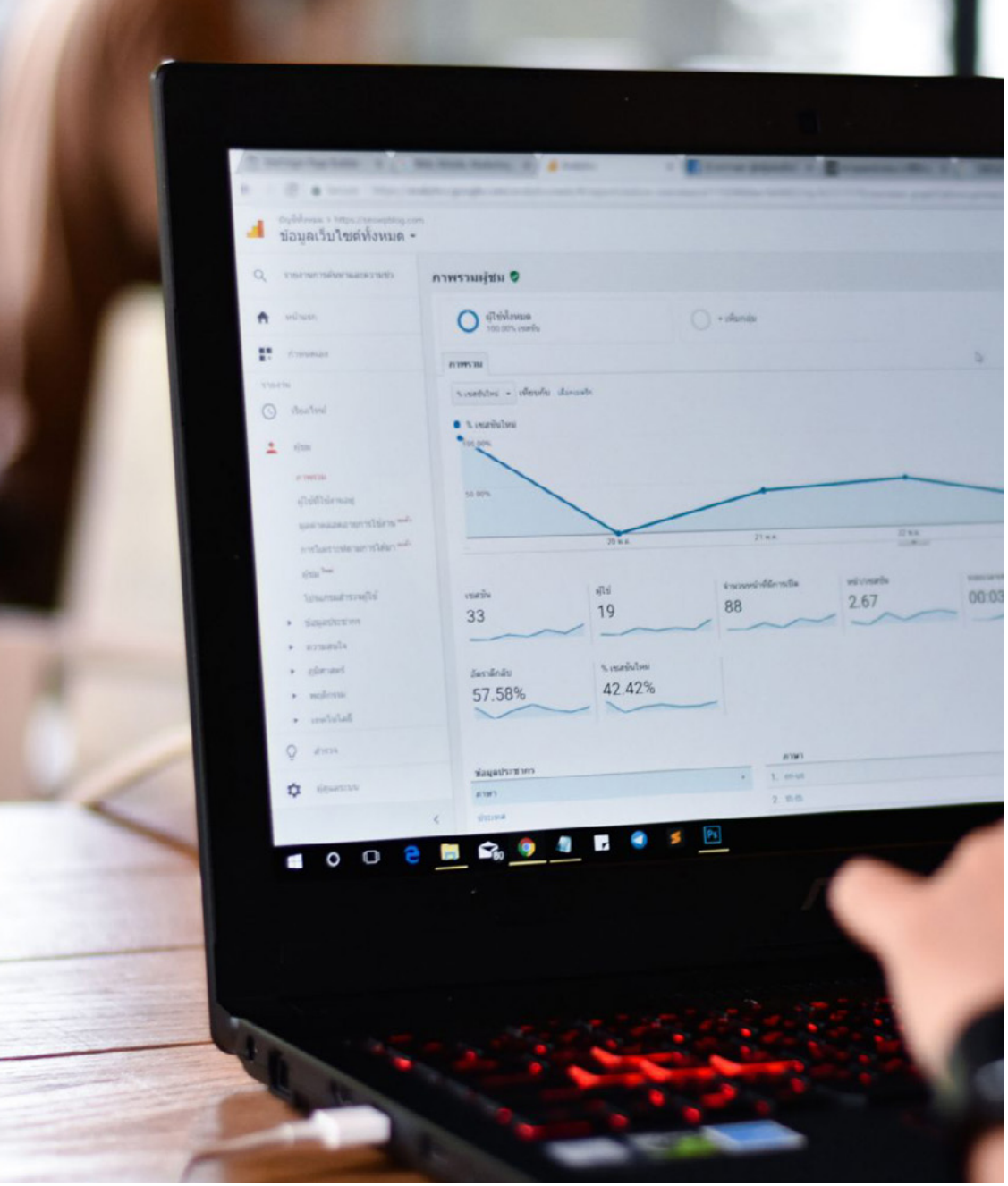
Data promises to help solve these issues. Instead of one large body of data, data mesh deconstructs it into distributed services built around a business node or domain capabilities. Because there is no centralized data function, data mesh supports decentralized ownership of domain-related data. Teams operate independently and autonomously as cross-functional units, much in the same way as DevOps. Data ownership and responsibility fall to these domains. They become the foundations of a mesh, resulting in a domain-driven distributed architecture.

Data mesh also requires a shift in organizational culture. For many organizations this means a move from a centralized decision-making around governance to a federated model, for example, built on cross-organizational trust.

Reconsidering how data is distributed

Enterprises are increasingly pursuing data democratization – making trusted, quality data available to everyone in the organization for smart decision-making and, at the same time, increasing productivity and efficiencies to achieve business outcomes rapidly. Data mesh delivers this using several principles:

- Rethinking data as a product
- Leveraging a domain-oriented self-service design
- Supporting distributed domain-specific data consumers



Rethinking data as a product

Rethinking data as a product is about changing organizational, architectural, and technological concepts to get the most out of data, data teams and data consumers.

Often data is seen as an asset – something valuable an organization or part of an organization is not willing to part with. However, rethinking data as a product creates more value by enabling data sharing and data democratization. Often this approach makes a cultural change necessary. In the data mesh approach, product teams own, control, and are accountable for the data they create and share.

Data mesh creates an ecosystem of data products, as opposed to a large, centralized data lake. The teams responsible for the data include the producers, data scientists and engineers, business analysts, while other users are seen as the customers for the data. With this cross-functional composition, teams include business and domain knowledge along with engineering expertise to realize these data products.

Self-service approach

For teams to autonomously work and take ownership of their data products, they require a simple and efficient way of managing the lifecycle of data and its provisioning. This is where self-service infrastructure as a platform comes in. It supports domain autonomy and allows teams to create and disseminate valuable data by providing dedicated and highly standardized domain environments. These are ultimately the nodes of the data mesh. Again, this helps the domain's data ownership by underpinning it with secure and governed access.

Self-service simplifies data access, breaks down silos, and enables the scaled-up sharing of live data. The infrastructure as a platform provides dedicated, standardized domain environments with all necessary components (such as storage or compute resources) a domain needs to implement their use case. This ensures that domains can focus on their business problem by not managing and maintaining the underlying infrastructure. Domains are tasked with collecting, managing, and curating data so that business intelligence applications can use it, for example.

Advantages of virtualization

Separating and abstracting the software from the underlying hardware creates many possibilities, but two are especially important.

The first is that commodity equipment using more open technology can replace the expensive proprietary hardware that these products used in the past. The software then runs on that hardware in a virtualized environment. That makes the costly, inflexible hardware component cheaper and easier to support, leaving the real value in the software.

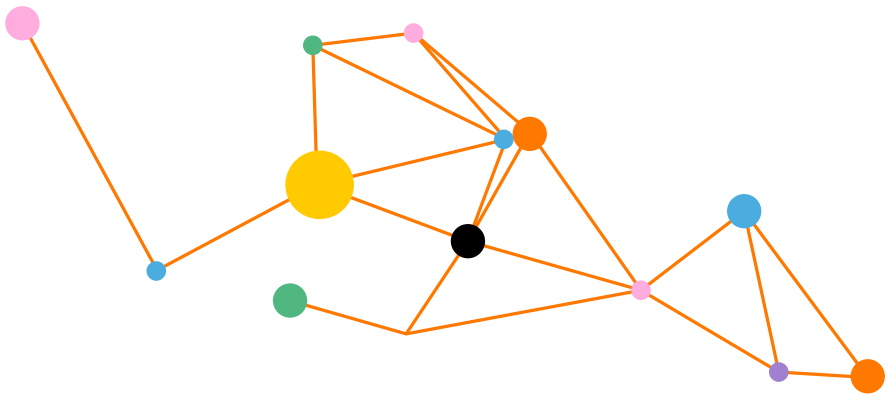
The second possibility is the management of that software. Administrators can control that virtualized software centrally from a dashboard, bringing the same centralized configuration and control capabilities to the entire infrastructure. This also makes it a lot simpler to enable automation and orchestration capabilities with proven IT efficiency and cost optimization benefits.

Interoperability, standardization, and governance

Maintaining data standards are imperative to data quality and trust. Every domain provides standardized interfaces to access their data which allows effective collaboration. The data output created can be helpful to more than one domain. Interoperability, standardization, and governance allows for efficient cross-domain collaboration at all levels, providing more significant innovation potential. It also allows trusted data to be offered as products across the enterprise.

It is essential to understand that this model requires both energy and commitment. With a decentralized model, it is crucial to establish governance and common standards that ensure data products are trusted and interoperable going forward. Moving from a monolith to a microservices model requires cultural change. It involves reorganization and changes in how teams and users work together. Without these changes, you can end up with a fragmented data system that doesn't work.

The effort, however, is worth it. The deconstructed data model brings with it greater business agility, scalability, and accelerated time to market. It also eliminates process complexities.



Data mesh in depth

Data mesh is an architectural paradigm that opens up analytical data at scale. It provides an organizational view of how to structure data, data platforms, and decentralized teams.

Instead of having a central data lake and central engineering teams, a data mesh consists of many data nodes (domains) that interact with each other, but operate independently. It describes a distributed domain-driven and self-service platform approach where data is treated as a product.

The concept is built around domain-driven data decomposition, where domains have full ownership of their data. A team includes both deep business knowledge, such as product managers or domain experts, and technical expertise, such as data engineers and data scientists. They are responsible for managing a domain together. This enables the team to consume, process, and serve data that closely matches the consumer requirements.

Domains are no longer dependent on engineering teams implementing their requirements. Instead, they can produce and consume data sets by themselves, while loosely coupled to other instances within the organization by following governance standards. In addition, domains can benefit from each other by consuming the data sets created as required.

Taking responsibility for business cases

Each domain is responsible for domain-related use cases or is involved in solving a specific business problem. This approach is designed to ensure high data quality as the data processing will be done by the team that has

the most knowledge about a use case. In contrast, data lakes, built on a centralized approach, often exhibit issues because:

1. The producers have the capabilities but are not motivated to fulfill requirements as their output doesn't relate to any particular use case.
2. The consumers are motivated but are dependent on the output of the centralized engineering team for data and data quality.
3. The engineering team is responsible for every implementation but has no specific domain knowledge.

In a data mesh, these data and data quality drivers are all placed within each domain.

Data mesh is a decentralized data platform which makes it easier for organizations to create new use cases and enables faster delivery of new features. This is made possible because it allows domain teams to act independently by utilizing the self-service platforms, with better understanding of the use case requirements.

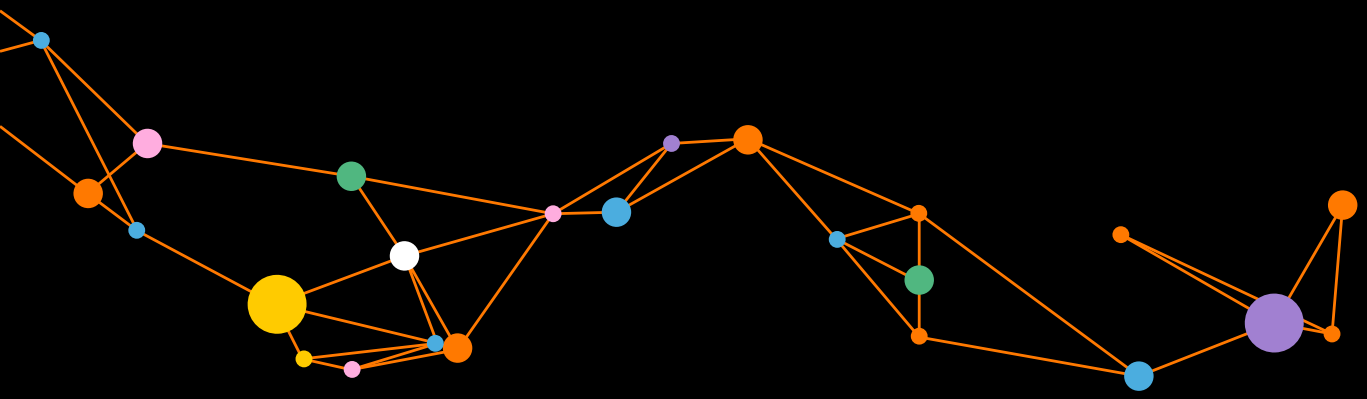
Data governance and infrastructure

In addition, data mesh has central components for data governance and infrastructure. Both components act as self-service platforms to efficiently support product owner workflows and eradicate friction when connecting to different parts of the infrastructure.

The governance component ensures that domain data is consumable across the organization. The data catalog holding and providing meta information about each domain will also need to be publicly accessible. Global standards are key to ensure interoperability between domains. Domain API specifications, schemas, member, permissions, and so forth will need to be provided in a standardized format.

Furthermore, domains can deploy a standard set of compute and storage resources on a self-service basis from a central infrastructure component. This reduces engineering overheads, while allowing the domain to focus on the actual data processing. The standard set of resources should enable domains to implement batch or streaming use cases and connect internal or external data sources. Note that a very high degree of automation is required here to create pre-configured and ready-to-use domain environments.

Both self-service components combined enable domains to act independently and thus more efficiently. In addition, the generation of new domains and use cases should also be frictionless.



A governance framework

In the context of treating data as a product, the governance of data in a decentralized data architecture is crucial. The key focus areas of data governance include availability, usability, consistency, data integrity, and data security.

The fact that a data mesh is a distributed domain-driven architecture and has a self-service platform design makes the data governance even more critical.

We have designed a data governance layer that provides the functionalities needed for all key focus areas of governing data mentioned before, including a data catalog backend (data discovery API) and a service (domain information service) covering the whole domain schema evolution and lifecycle. This includes five steps as follows:

1 **Registration**
Whenever a domain joins the data mesh, it needs to be initially registered. In this registration process, all basic information about the domain is stored: data format, data schema, processed data in terms of data lineage, processing steps, and other data quality indicators provided by the domain.

2 **Domain schema**
A domain can change over time regarding the data format or schema, its behavior, or the internally used data sources. This means that domain information in the data catalog will need to be kept up to date. For this reason, each domain must provide an endpoint where this information can be retrieved (domain schema API). In our architectural design, the domain information service (DIS) pulls this data by using the provisioned endpoint of a specific domain

to store schema evolution information, etc. The DIS implements the logic for registering, updating, and querying schemas within the data catalog. It represents the service layer for the data discovery API and executes every request to the data catalog.

3 Data discovery

Domain data is made discoverable by the data discovery API, which makes use of the DIS. The data discovery API is technically a backend that exposes and manages the DIS endpoints and all domain APIs. Here, domain APIs access is controlled and restricted to secure its data from unauthorized access.

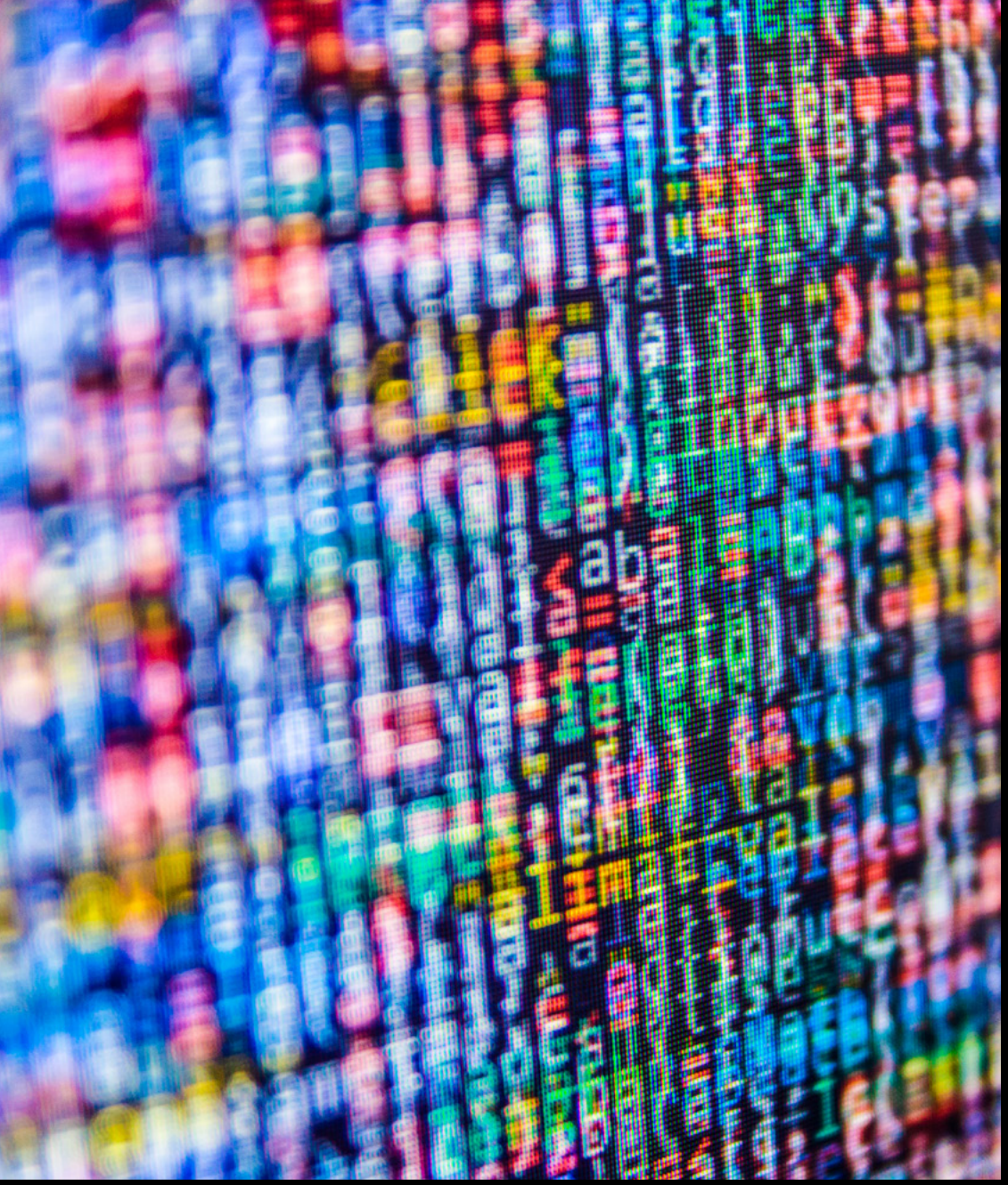
The addressability of the data products can be achieved by following global standards for access to data via endpoints and data schema descriptions.

4 Interoperability

Interoperability and standardization of communications are one of the fundamental pillars for building distributed systems. It is vital to establish global standards regarding data fields, and the metadata of the domain data, such as data provenance and data lineage. This increases the quality of the service level objective around the truthfulness of the data. This information is governed globally by the DIS and stored in the data catalog.

5 Security

Secure access to the data mesh and its individual domains is a mandatory standard in every data architecture. In our architectural design, we assume that the data is accessible via REST endpoints. The access to these endpoints is managed and secured by using API management services.



Rapid deployment with infrastructure-as-a-platform

The primary purpose of the infrastructure platform is to enable domains to immediately start working on their use cases by utilizing predefined automated infrastructure deployments.

The infrastructure platform consists of two components. One is the provisioning service handling requests for new domain environments. The other is code repositories containing all the automation code (IaC) for core components and domain environments, and several CI/CD pipelines for automated deployments. The IaC code for domain environments is designed to be suitable for every domain.

As a self-service platform, the provisioning service can be used by domains to request new environments. The following resource types should be automatically deployed into a domain's target environment:

- Secret/key management tool
- Workflow management tools
- Compute resources for large-scale data processing (such as Spark)
- Runnable container/serverless application code
- Persistence backend, such as blob storage, SQL, or NoSQL solutions
- Stream processing tools
- Monitoring and alerting solutions

Where domains fit in

A domain is responsible for the data of a clearly definable problem area. In doing so, it consumes data from one or more other domains or external sources, processes it, and provides output data, which again is consumable by other domains. The domain offers the schema for its output data. One team should be in charge of both a domain and its data quality.

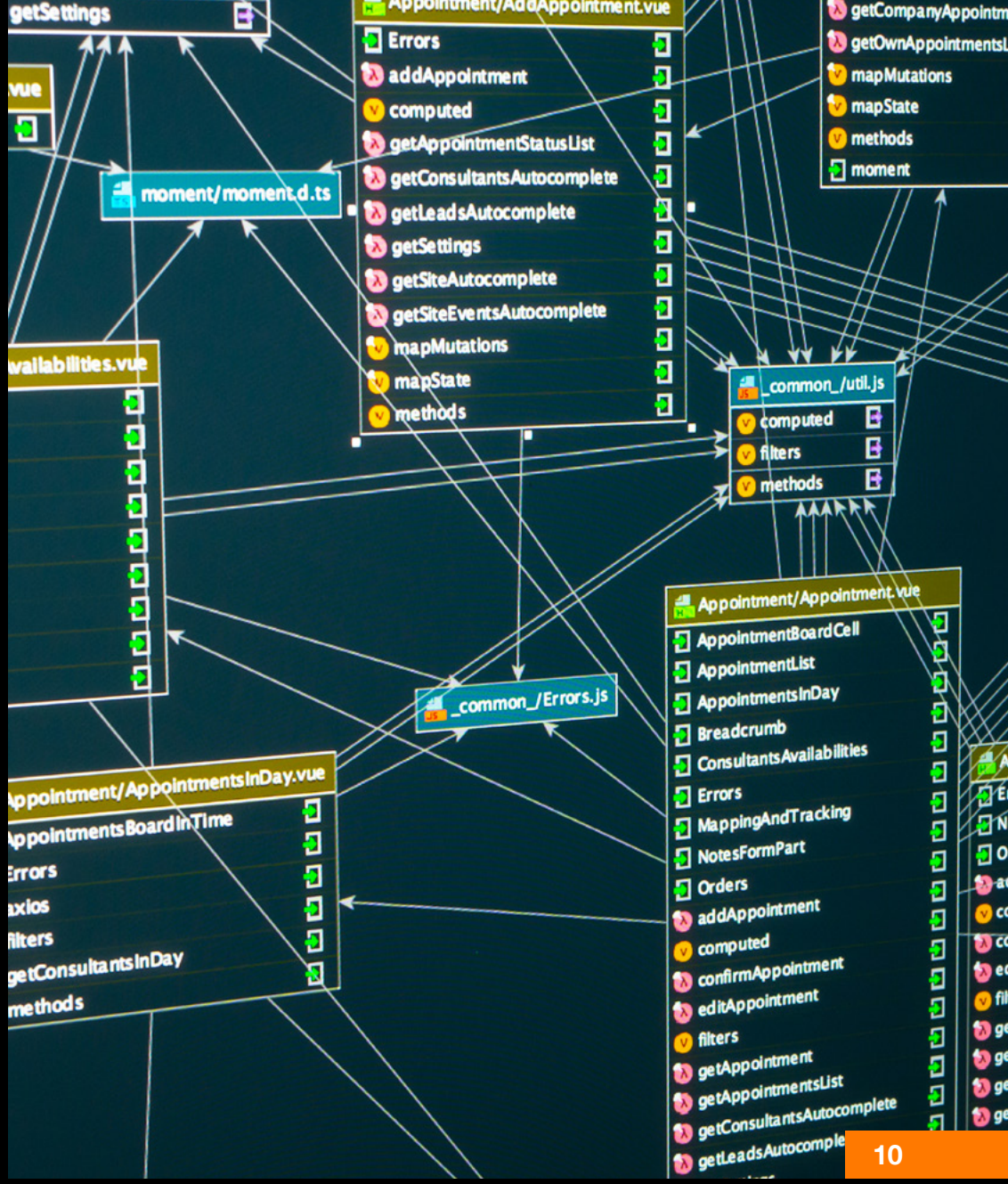
Furthermore, a well-defined toolset belongs to every domain. With the help of this, the domain can carry out all necessary work steps. This toolset is provided by the infrastructure platform and can be requested by every domain independently.

To avoid every domain data set being accessed differently, an abstraction of the backend technology will be applied. Therefore, every domain must implement a REST API – the domain API – to provide the requested data. Once implemented, domains can register their API in the data discovery API. The registration triggers a process that:

1. Stores the domain schema within the data catalog
2. Exposes the domain's API endpoints centrally in the data discovery API

This makes the domain schema discoverable and accessible to other domains.

Other domains must utilize the API in question – otherwise, direct access to the domain data will not be possible. This makes the backend technology interchangeable and insignificant to other domains. The domain API provides endpoints to retrieve the data (e.g. by date) and the current schema. Create, update and delete endpoints should not necessarily be provided. Cases where the schema changes over time are covered by the DIS frequently querying the schema to update changes in the data catalog.



Open vs strict model

Two different approaches to managing domains are possible in a data mesh: the open and strict models.

In brief, the open model gives domain teams as much freedom as possible. The strict model supports domain teams in highly-regulated environments that cannot be changed. Both approaches have pros and cons, and of course, hybrid solutions are feasible too. We discuss them in depth below.

Open model

In the open model, domains have no limitations in choosing their tools for data processing and storing. In addition to the standard toolset deployed by the infrastructure platform, further resources of every type can be added by the team by customizing the infrastructure code.

But more importantly, storing and publishing output data is fully managed by the domain itself. There is no central instance for storing the output data in a predefined backend technology. Instead, the domain can decide to use a blob container, SQL database, document store, etc. They can choose between structure and naming conventions and only need to make sure to expose their domain API. They have full ownership and responsibility to ensure consistency between the exposed API and the actual implementation.

This approach requires reliable and responsible domain teams to avoid inconsistencies and data quality. It gives domains more freedom, reduces implementation and automation effort within the platform team, and only works in organizations with senior-level domain teams. Because of the flexible approach, it is suited to business users adopting big data and DataOps.

Strict model

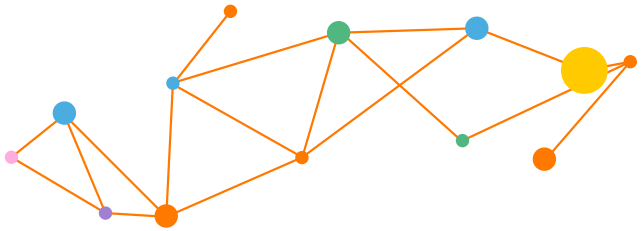
The strict model predefines the whole domain environment without any possibility of changing it. Domains have no access to their infrastructure code, so they must stick to the standard set of resources. Furthermore, their persistence layer is under central management.

With this approach, there is an area for each domain with strict regulations and policies on where and how to store the data. Also, their exposed domain API is regulated and controlled by a central validation process. This ensures that domain API and implementation will always be congruent.

This strict model requires a lot of implementation and automation effort within the platform team and presupposes a very sophisticated data mesh platform. On the other hand, it ensures high data quality and consistency by design. This highly developed model is targeted towards research institutions and advanced big data and analytics users.

Going down the data mesh route

Data is increasingly distributed in all enterprises. Now is a good time for any enterprise that has moved to the cloud and is deploying microservices to think data mesh. The concept allows for easier, more efficient, small domain name components that enhance the user experience and are key to a data-driven organization.



Data mesh results speak for themselves

Data mesh is a burgeoning paradigm in data architecture that enables enterprises to take control of large data and improve business outcomes.

By 2025, IDC maintains global data will grow 61% to 175 zettabytes 2025.[1] The collection, integration, and governance of this data to gain valuable business insight is increasingly complex.

For enterprises that require flexible access to their data to accelerate time to market, data mesh's democratized approach to data management provides an ideal solution. The direct benefits for enterprises adopting this model include:

- Establishing global data governance guidelines that encourage teams to produce and deliver high-quality data in a standardized and reliable format
- Eliminating the challenges of data availability, making discoverability and accessibility easier in a secure and interoperable environment
- Increasing agility with decentralized data operations and a self-service infrastructure
- Allowing teams to operate in a more agile and independent way to reduce time-to-market and deliver new data products faster

Why Orange?

Ready to adopt a new way of managing data?

For enterprises working with large, diverse, and often dynamic data sets and wanting to sell up rapidly or are worried their current data infrastructure is slowing down innovation, data mesh is the ideal model. It provides a sustainable way for you to harvest business value from your growing amounts of data.

Here at Orange Business Services, we can help you establish, run and manage your data mesh solutions if required.

- We provide a business strategy evaluation to assess your maturity and suitability for data mesh.
- We can design, architect and develop a data mesh based on your unique business needs. We can approach this via a co-innovation model, enabling you to be self-sufficient in development in the future.
- We provide a managed service to run and manage enterprise data mesh solutions.
- Adopting data mesh requires cultural change. We can help you drive cultural change in the workplace, and help you reach your goal of being a data-driven company.

To find out how Orange Business Services can help you to empower your users and create a decentralized data mesh architecture that fits your specific business needs, contact us at:

Orange Business Services
Philipp Ringgenberg
Head Innovation & Business Consulting
philipp.ringgenberg@orange.com

Orange Business Consulting & Innovation
www.orange-business.com/en/products/consulting-services-translate-business-benefits-digital-technologies

orange™ Business Services

Sources:
1. IDC Data Age 2025 whitepaper