Architect’s Open-Source Guide for a Data Mesh Architecture

Lena Hall
Microsoft
Lena Hall

Dirctor at Microsoft Azure Engineering

✓ Architecture
✓ Cloud
✓ Data
✓ ML/AI

lenadroid
Entry Point

How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh
https://martinfowler.com/articles/data-monolith-to-mesh.html

Data Mesh Principles and Logical Architecture
https://martinfowler.com/articles/data-mesh-principles.html

Slack for Data-Mesh-Learning
https://launchpass.com/data-mesh-learning
Talk Snapshot

- What is Data Mesh
- When is Data Mesh a Good Idea
- Core Principles and Concepts
- Example: Drone Delivery Service
- Challenges
- OSS and Open Standards
When and Why
Data Mesh
Data Mesh is Not For Everyone
Challenges Indicating **Data Mesh** May Be Considered
Drone Delivery Service
WHYs

• Ambiguity in Ownership and Responsibility
• Slow Change due to Coupling to Monolithic System
• Data Engineering Resources Bottleneck
Ideas Composing *Data Mesh* Concept
Core Ideas

✓ Decentralized teams and data ownership
Core Ideas

- Decentralized teams and data ownership
- Data Products powered by Domain Driven Design
High-Level View of a Data Product

INPUT PORTS

OUTPUT PORTS

ACTION

SELF-SERVE DESCRIPTION AND METADATA

LOGS AND SUCCESS METRICS

IDENTITY OR SERVICE ACCOUNT

lenadroid
Core Ideas

✓ Decentralized teams and data ownership
✓ Data Products powered by Domain Driven Design
✓ Self-serve Shared Data Infrastructure
Core Ideas

✓ Decentralized teams and data ownership
✓ Data Products powered by Domain Driven Design
✓ Self-serve Shared Data Infrastructure
✓ Global Federated Governance
Drone Delivery Service Data Products
Core Principles for **Data Products**
Core Principles for Data Products

DISCOVERABLE
Core Principles for Data Products

DISCOVERABLE

SELF-DESCRIBING
Core Principles for Data Products

DISCOVERABLE

SELF-DESCRIBING

ADDRESSABLE
Core Principles for Data Products

DISCOVERABLE

SECURE

SELF-DESCRIBING

ADDRESSABLE
Core Principles for Data Products

DISCOVERABLE

SELF-DESCRIBING

ADDRESSABLE

SECURE

TRUSTWORTHY
Core Principles for Data Products

- Discoverable
- Self-describing
- Addressable
- Secure
- Trustworthy
- Interoperable
Cheat Sheet for Planning Data Products

Input Ports Questions
- **Data Source** - Where is the data coming from? External dataset or another data product?
- **Data Format** - What is the format of the source input?
- **Rate of Updates** - How frequently does the input need to be updated?

Output Ports Questions
- **End-consumers** - Who are the end-users of the data product?
- **Data purpose** - What are they planning to do with the data outputs?
- **Data access** - Who needs to have access? How do they prefer to access the data output?
- **Data address** - How do they prefer to access the data output?
- **Data Format** - What format of the data do they expect?

Identity and Permission Policies Questions
- Which resources can this data product be allowed to access?
- Which data products or users can read which output ports of this data product?
- Are all sensitive resources this data product offers protected according their required privacy standards (e.g. HIPAA, GDPR, PII, CCPA, etc.)
- Is the permissions policy stored and managed in the same package as the data product?

Data Product Action Questions
- What is the action that needs to happen to produce the outcomes for the end-users?
- What are the required adjustments, transformations, filters, updates, or quality improvements to the input data?

Operational Questions
- How can this data product be discovered and how should it be described to other data products that might want to consume it?
- Which metadata and information should it make available to the end-users?
- Where and how should data product versioning be managed during updates to ensure consistency with how the end-users consume it?
- Which SLAs or SLOs does the data product provide?
- Which product success metrics can this data product expose and keep track of? (adoption, usage, quality)
- Is the automation/resource orchestration logic stored in the same package?

Other Questions
- Is this product not tightly coupled to any other data source, data product, or any other resource that makes him not interoperable?
- Does this data product follow the defined global governance standards and practices defined by the organization?
- Does this data product have any implementation details that could interfere with its portability?
Self-Serve Shared Infrastructure
Types of Workloads Within a Data Product

REAL-TIME DATA

PROCESSING

INCOMING REQUEST

WEB SERVICE

PROCESSING

COLUMNAR STORAGE

OBJECT STORAGE
It can look like this

1. WEB SERVICE
2. Apache Kafka
3. Apache Parquet
4. Apache Spark
5. Azure Synapse Analytics
6. Azure Data Lake
7. Apache Airflow
Or, it can look like this
Self-Serve Shared Infrastructure

- Shared platform for streaming ingestion
- Shared platform for raw data storage
- Shared platform for columnar data storage
- Shared platform for container workloads
- Shared platform for continuous delivery
- Shared platform for observability
- Data catalogue

And more... depending on the organization.
Data Mesh

DISCOVERABLE

SELF-DESCRIBING

ADDRESSABLE

SECURE

TRUSTWORTHY

INTEROPERABLE

SHARED PLATFORM FOR CONTAINER WORKLOADS

SHARED PLATFORM FOR CONTINUOUS DELIVERY

SHARED PLATFORM FOR OBSERVABILITY

SHARED PLATFORM FOR STREAMING INGESTION

SHARED PLATFORM FOR RAW DATA STORAGE

SHARED PLATFORM FOR COLUMNAR DATA STORAGE

SHARED PLATFORM FOR CONTINUOUS DELIVERY

SHARE PLATFORM FOR OBSERVABILITY

DATA CATALOGUE
Wait, What About the OSS Tools for Data Mesh??
Challenges with Data Mesh
Challenges

- Cost questions
- Lack of end-to-end examples
- Efforts to shift from centralized architecture to decentralization-friendly techniques
- Automation required for enabling creating data products
- Underestimating the importance organizational aspects
Considerations for Technology Choices
Considerations for Technology Choices

- Workload sharing and multi-tenancy
- No-copy data and compute mobility support
- Granularity of access-control
- Richness of automation and extensibility capabilities
- Flexibility and elasticity
- Provider-agnostic/multi-cloud operations support
- Variety of limitations (quotas, data volume, resource count, etc.)

- Open Standards, Open Protocols, Open-Source Integrations
Examples of Data Mesh-friendly Technologies
<table>
<thead>
<tr>
<th>Data Catalogue, Data Lineage, Data Governance</th>
<th>OSS Data Analytics, Data Processing, Data Querying</th>
<th>Data Ingestion, Streaming</th>
<th>Products for Data Analytics and Processing</th>
<th>Data Visualization and BI Tools</th>
<th>Cross-Platform Concepts and Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSS Data Ingestion, Streaming, Data Orchestration, Workflows</td>
<td>OSS Storage</td>
<td>Infrastructure Automation</td>
<td>Data Experimentation</td>
<td>Multi and Hybrid Cloud Tools</td>
<td></td>
</tr>
<tr>
<td>ICEBERG, Parquet, Orc, Avro, HDFS</td>
<td>Amazon S3, Azure Data Lake, Google Storage</td>
<td>Ceph, Pulumi, Terraform, Datafy, Jupyter</td>
<td>Anthos, Azure Arc, Crossplane</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Governance Systems

- Metadata
- Data lineage
- Data schemas
- Data relationships
- Data classification
- Data security
- Data catalog
Open Formats

- Open standard
- Atomic updates, serializable isolation, transactions
- Concurrent operations
- Versioning, rollbacks, time-travel
- Schema Evolution
- Scale, Efficiency, Data Volumes
- Compatibility with existing data stores and languages
Data Platforms (Cloud or OSS)

- Separation of storage and compute
- Support for no-copy data sharing
- Bringing compute to data
- Fine-tuned granularity of permissions for access
- Support for automation and resource management
- Open standards and interoperability with other platforms and tools for governance, visualization, analytics, etc.
Multi-Cloud Infrastructure Management

• **Terraform**
  Open-source infrastructure as code software tool that enables you to safely and predictably create, change, and improve infrastructure.

• **Pulumi**
  Open-source infrastructure as code SDK that enables you to create, deploy, and manage infrastructure on any cloud, using your favorite languages.

• **Crossplane**
  Assemble infrastructure from multiple vendors, and expose higher level self-service APIs for application teams to consume, without having to write any code.
Multi-Cloud Workload Portability

- **Azure Arc**
  Build cloud-native apps anywhere, at scale. Run Azure services in any Kubernetes environment, whether it’s on-premises, multi-cloud, or at the edge

- **Google Athnos**
  A modern application management platform that provides a consistent development and operations experience for **cloud** and on-premises environments
Kubernetes Open-Standard Technologies

NOT AN EXHAUSTIVE LIST

- **Open Application Model**
  An open standard for defining cloud native apps.
  KubeVella - [https://kubevela.io/docs/concepts](https://kubevela.io/docs/concepts)

- **Open Policy Agent**
  [https://www.openpolicyagent.org/docs/latest](https://www.openpolicyagent.org/docs/latest)

- **Service Catalog**
  Provision managed services and make them available within a Kubernetes cluster.
  [https://kubernetes.io/docs/concepts/extend-kubernetes/service-catalog/](https://kubernetes.io/docs/concepts/extend-kubernetes/service-catalog/)
Benefits Brought by Data Mesh

• Data Quality
• Tailored resource and focus allocation
• Organizational cohesion while allowing flexibility
• Reducing complexity
• Democratizing creating value
• Better understanding of value and innovation opportunities
• Empowering a more consistent and fast change
Important Focus Areas for Technology Providers

- Open Standards, Open Protocols, Open-Source Integrations
- Workload sharing and multi-tenancy
- No-copy data and compute mobility support
- Granularity of access-control
- Richness of automation and extensibility capabilities
- Flexibility and elasticity
- Provider-agnostic/multi-cloud operations support
- Variety of limitations (quotas, data volume, resource count, etc.)
Data Mesh will drive better Interoperability, Open Standards, and Data Quality in the Industry
Thank you!

Follow lenadroid for more insights