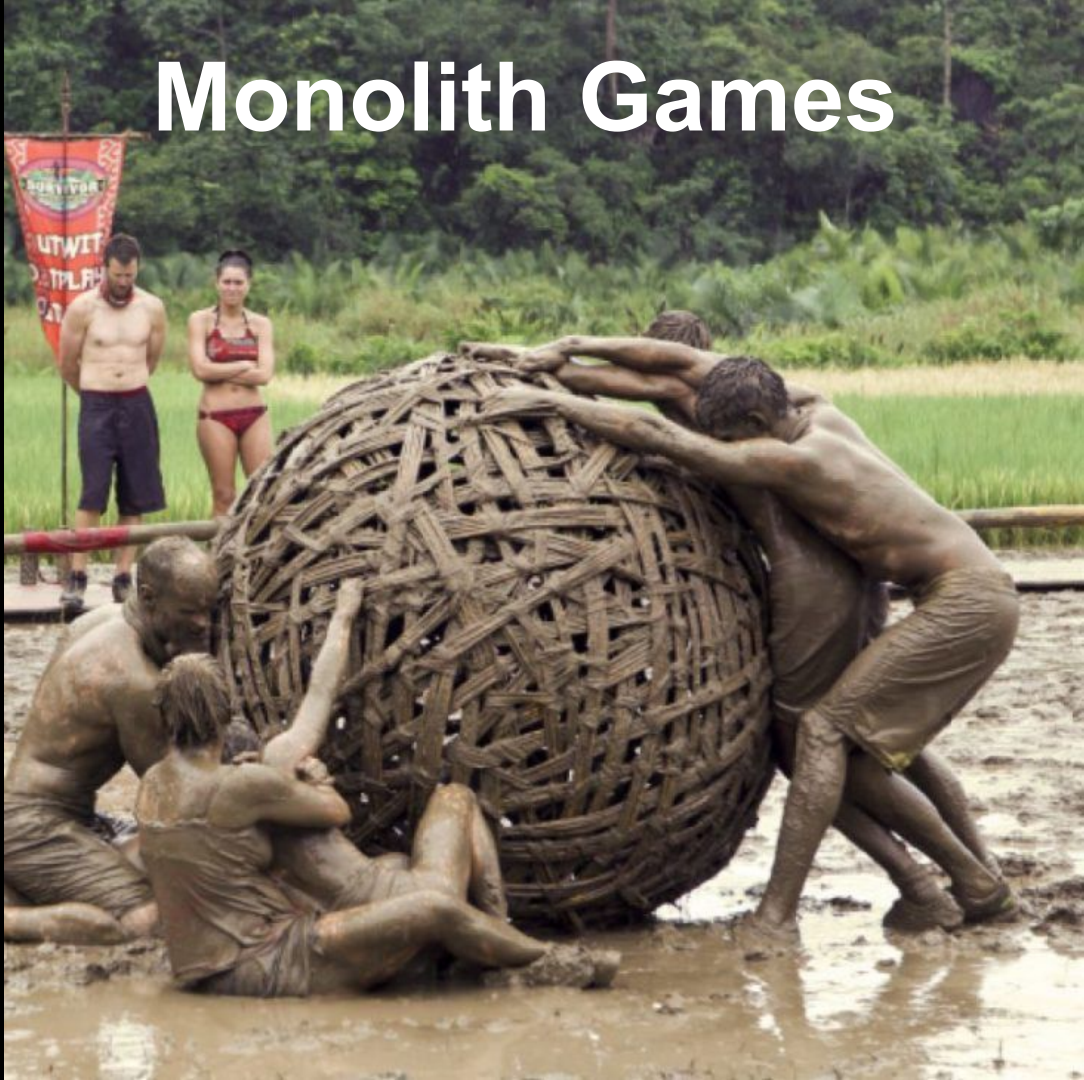
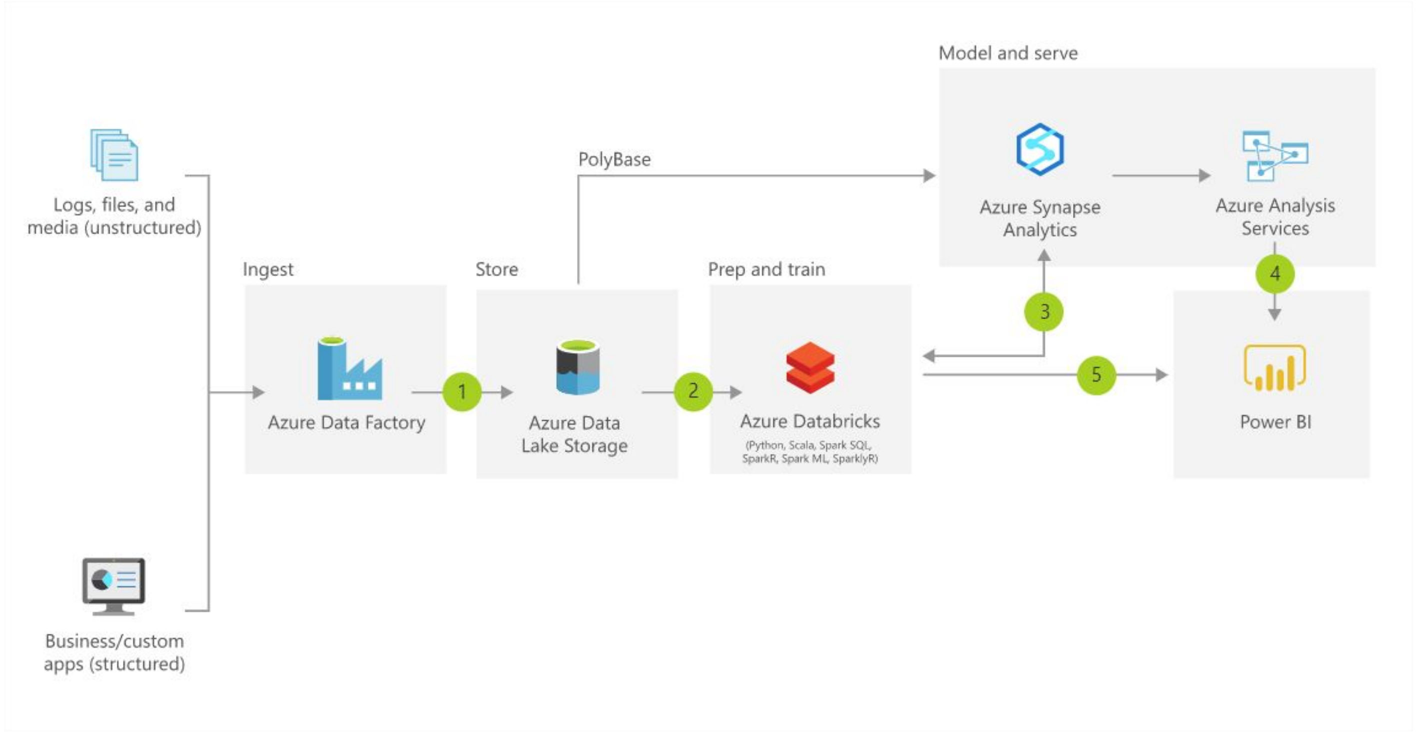
Monolith Games

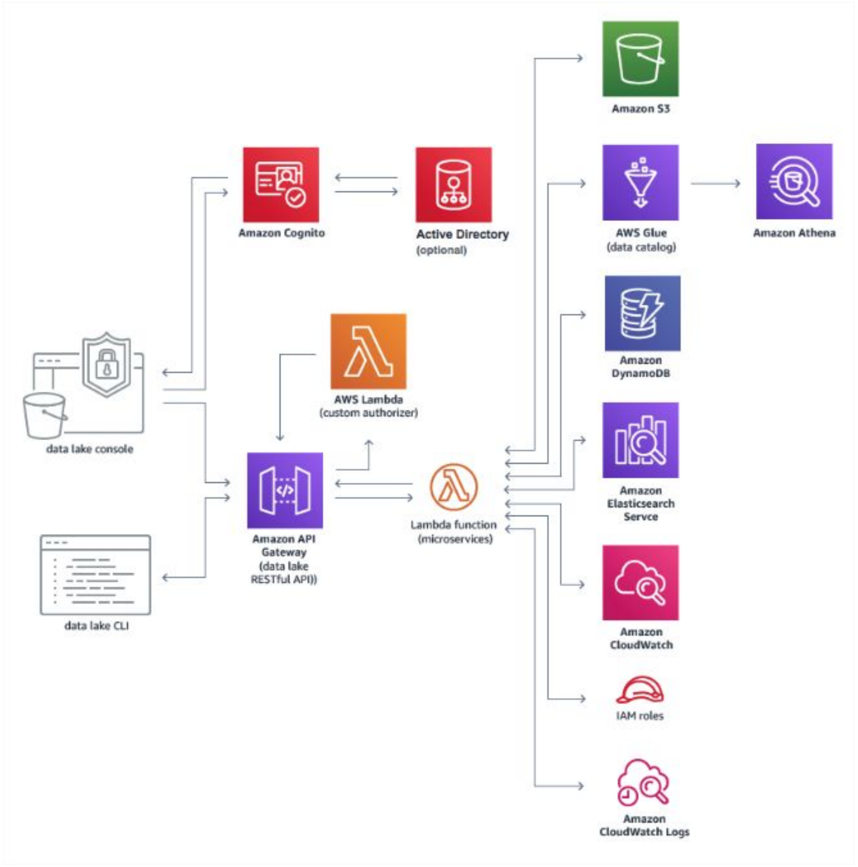


Azure Data Platform Reference Architecture



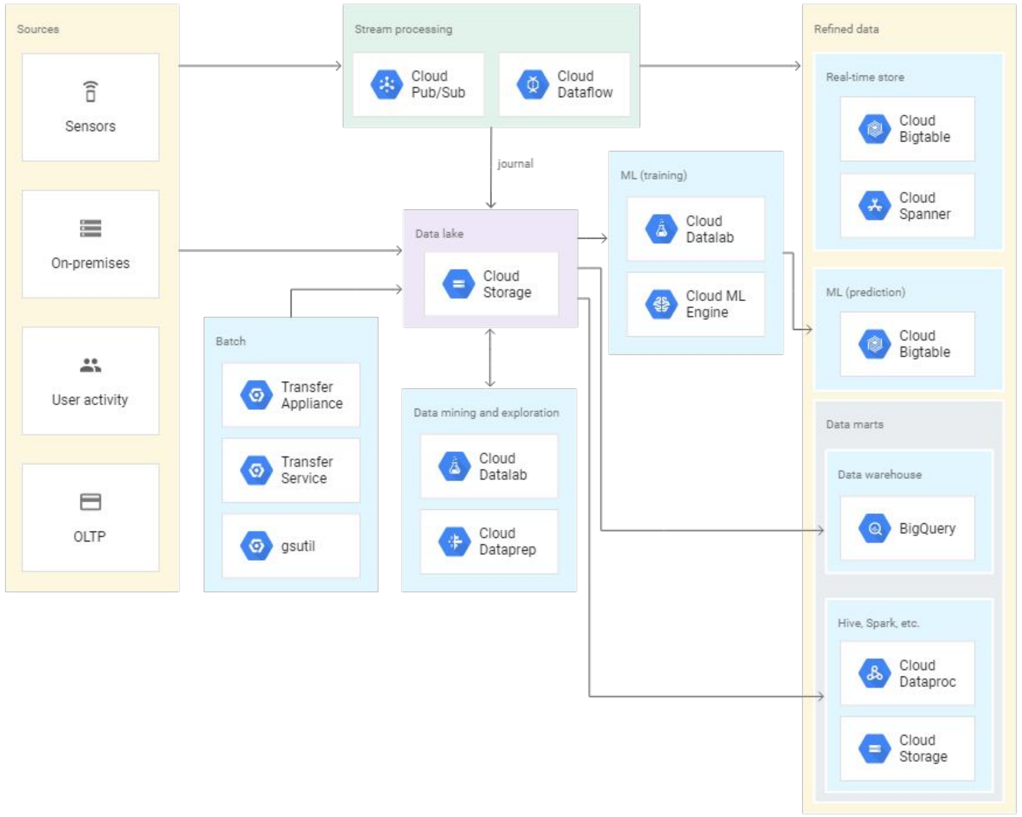
Source: https://azure.microsoft.com/en-in/solutions/architecture/modern-data-warehouse/

AWS Data Lake Reference Architecture


Amazon S3
Amazon Cognito
Active Directory (optional)
AWS Glue (data catalog)
Amazon Athena
AWS Lambda (custom authorizer)
Amazon DynamoDB
data lake console
Amazon Elasticsearch Servce
Amazon API Gateway (data lake RESTful API))
Lambda function (microservices)
data lake CLI
Amazon CloudWatch
IAM roles
Amazon CloudWatch Logs

Source: https://aws.amazon.com/solutions/data-lake-solution/

GCP Data Lake Reference Architecture



Source: https://cloud.google.com/solutions/build-a-data-lake-on-gcp

Why a monolithic data platform is bad?

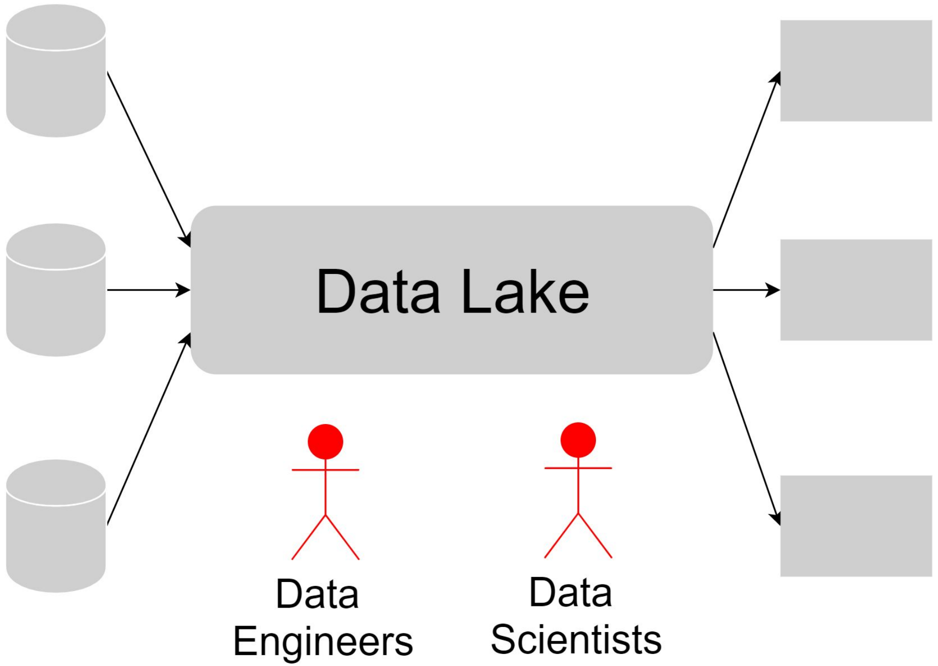
1. **Organization is not monolithic**

"organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations."

— Melvin Conway

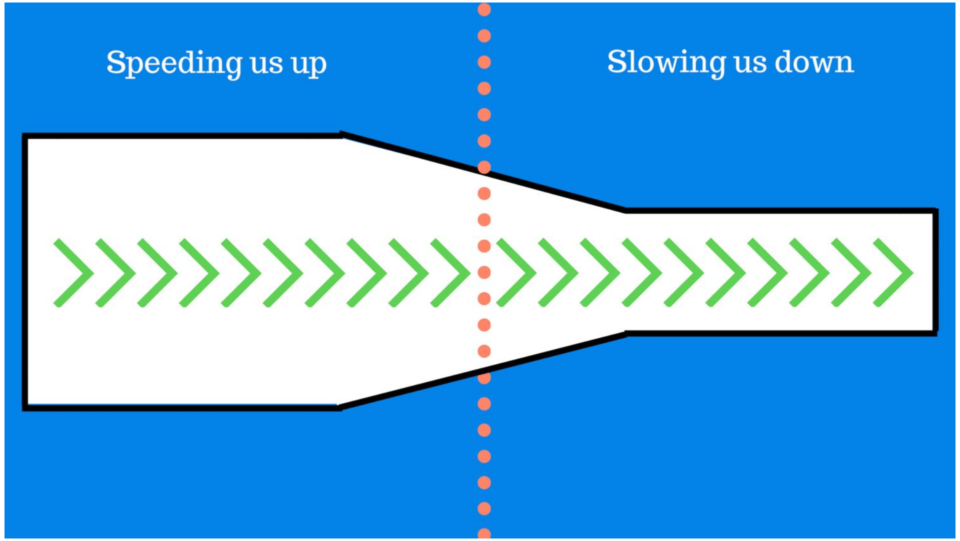
Why a monolithic data platform is bad?

1. Organization is not monolithic
2. **Skillset silos**



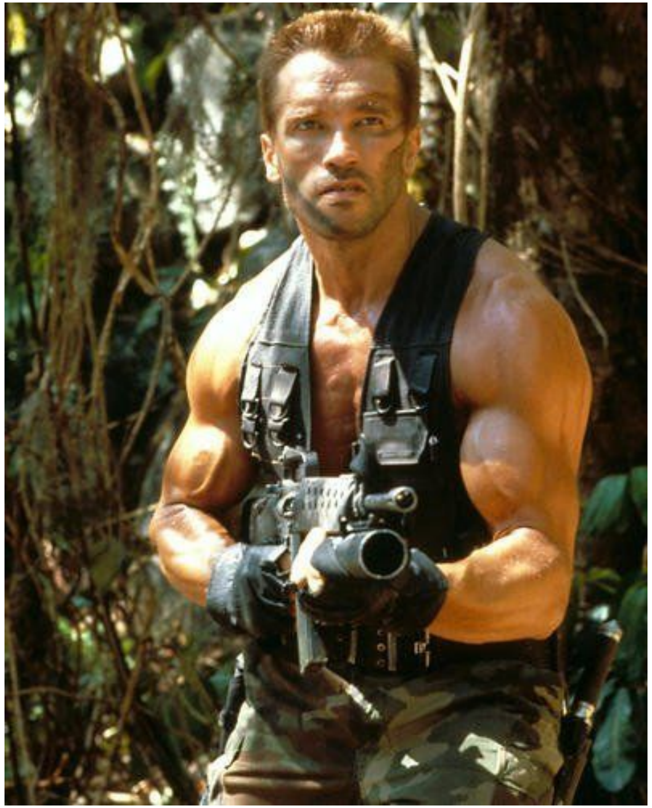
Why a monolithic data platform is bad?

1. Organization is not monolithic
2. Skillset silos
3. **Single backlog**



Why a monolithic data platform is bad?

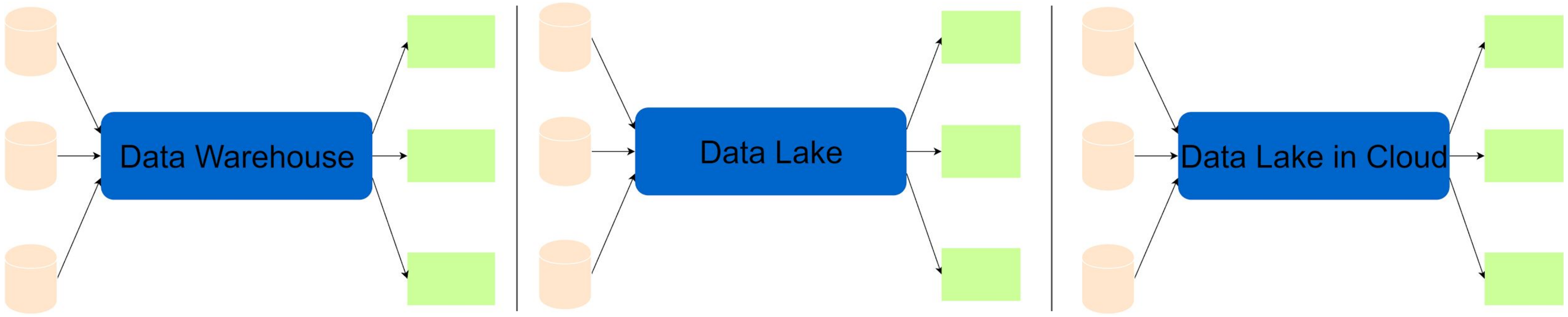
1. Organization is not monolithic
2. Skillset silos
3. Single backlog
4. **Data hunting**

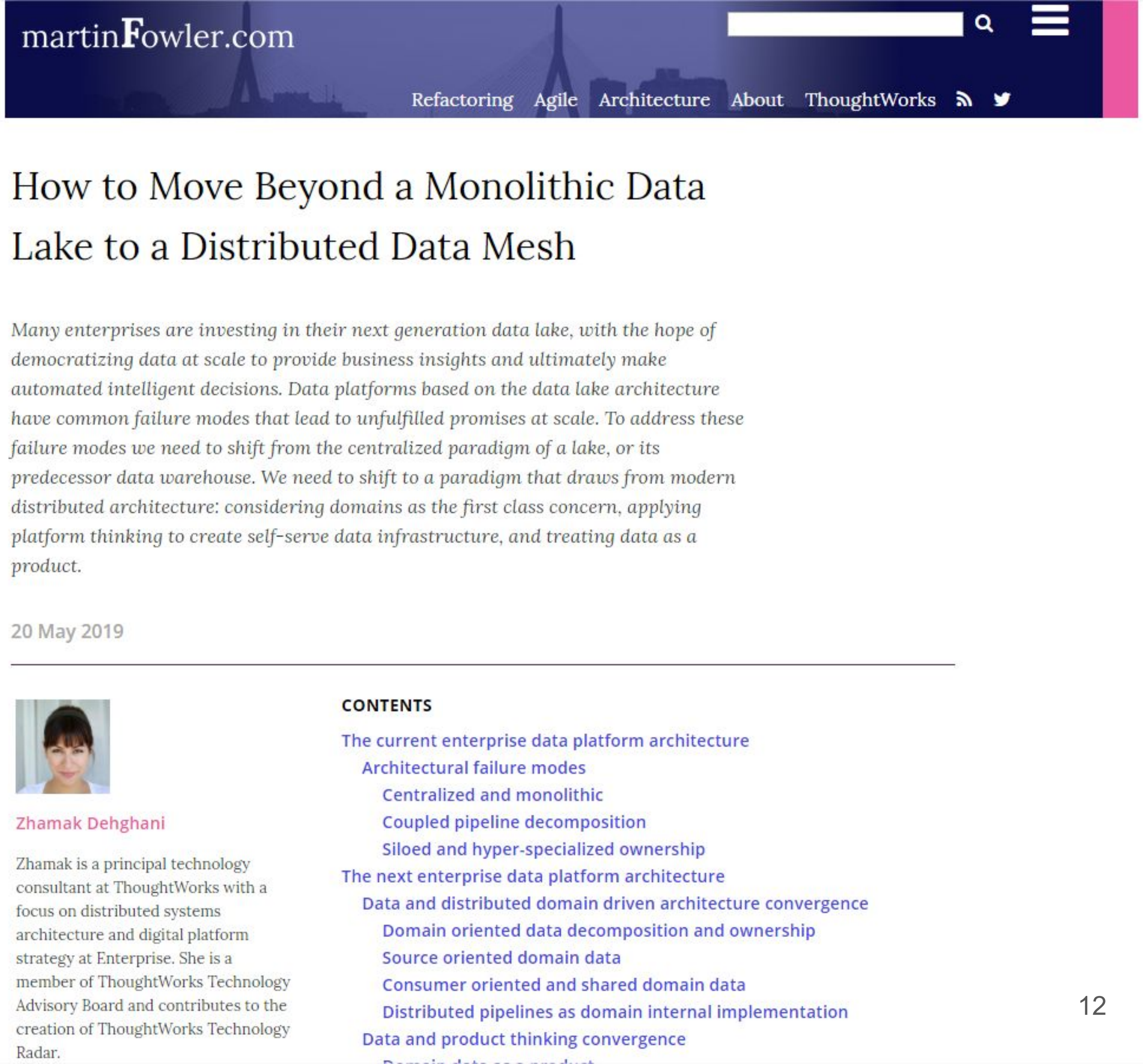


Why a monolithic data platform is bad?

1. Organization is not monolithic
2. Skillset silos
3. Single backlog
4. Data hunting
5. **Exploration restrictions**

The flawed evolution of data platforms





martinFowler.com

Refactoring Agile Architecture About ThoughtWorks

How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh

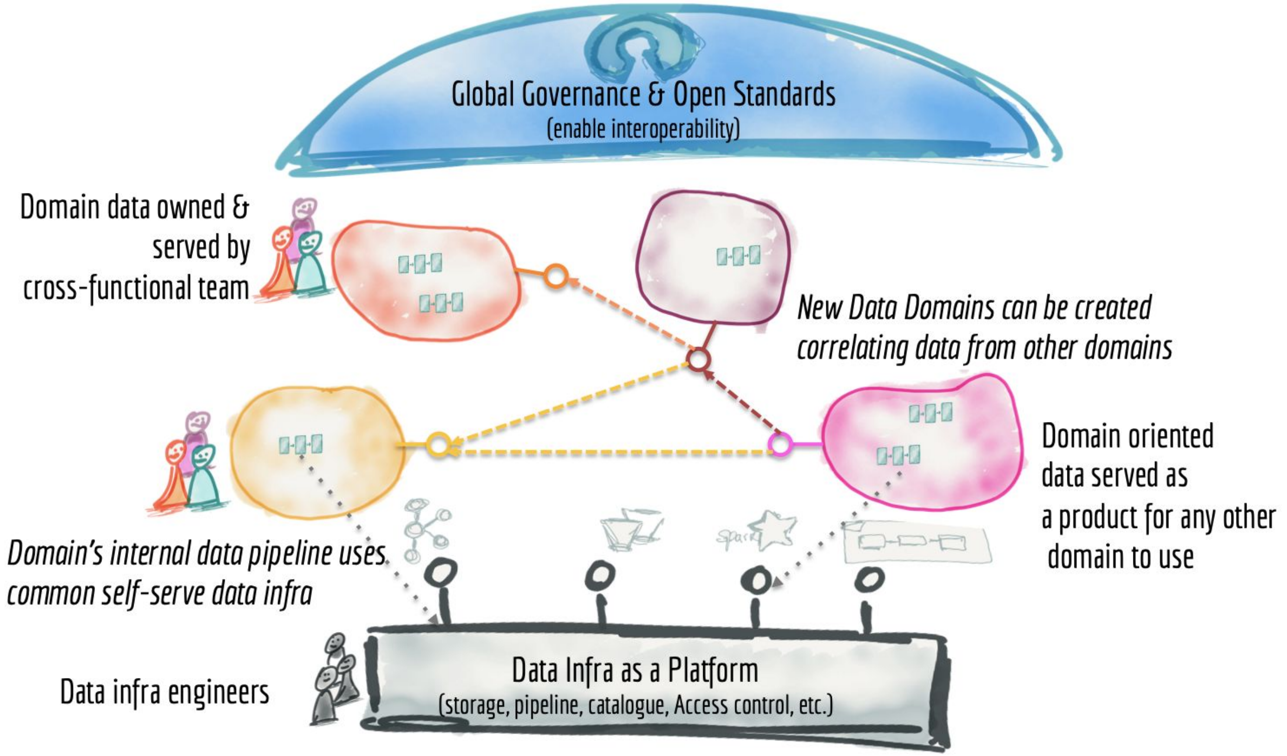
Many enterprises are investing in their next generation data lake, with the hope of democratizing data at scale to provide business insights and ultimately make automated intelligent decisions. Data platforms based on the data lake architecture have common failure modes that lead to unfulfilled promises at scale. To address these failure modes we need to shift from the centralized paradigm of a lake, or its predecessor data warehouse. We need to shift to a paradigm that draws from modern distributed architecture: considering domains as the first class concern, applying platform thinking to create self-serve data infrastructure, and treating data as a product.

20 May 2019

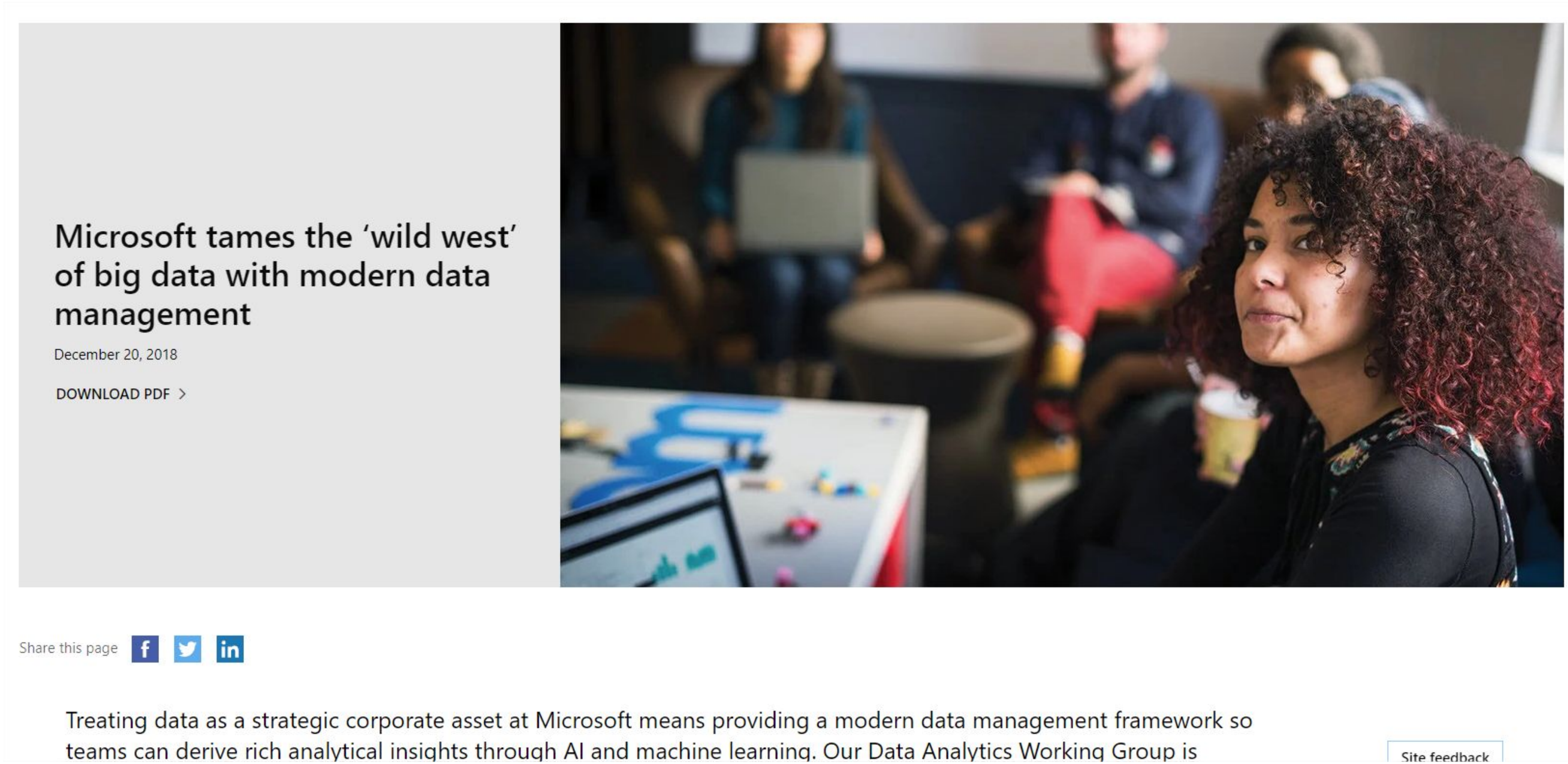
Zhamak Dehghani

Zhamak is a principal technology consultant at ThoughtWorks with a focus on distributed systems architecture and digital platform strategy at Enterprise. She is a member of ThoughtWorks Technology Advisory Board and contributes to the creation of ThoughtWorks Technology Radar.

CONTENTS


Global Governance & Open Standards
(enable interoperability)
Domain data owned & served by cross-functional team
New Data Domains can be created correlating data from other domains
Domain oriented data served as a product for any other domain to use
Domain's internal data pipeline uses common self-serve data infra
Data infra engineers
Data Infra as a Platform
(storage, pipeline, catalogue, Access control, etc.)

Source: https://martinfowler.com/articles/data-monolith-to-mesh.html



Microsoft tames the 'wild west' of big data with modern data management

December 20, 2018

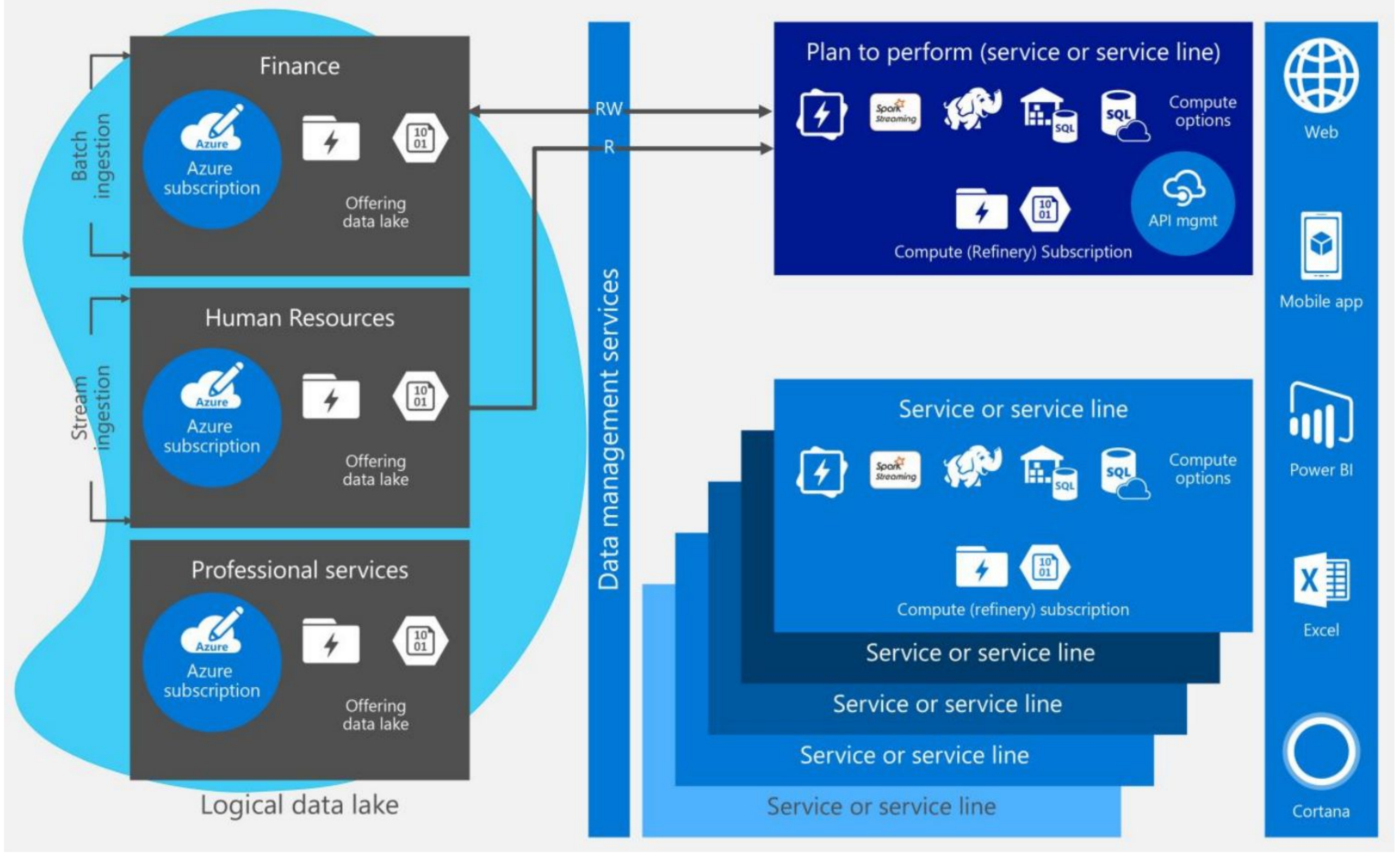
DOWNLOAD PDF >

Share this page

Treating data as a strategic corporate asset at Microsoft means providing a modern data management framework so teams can derive rich analytical insights through AI and machine learning. Our Data Analytics Working Group is

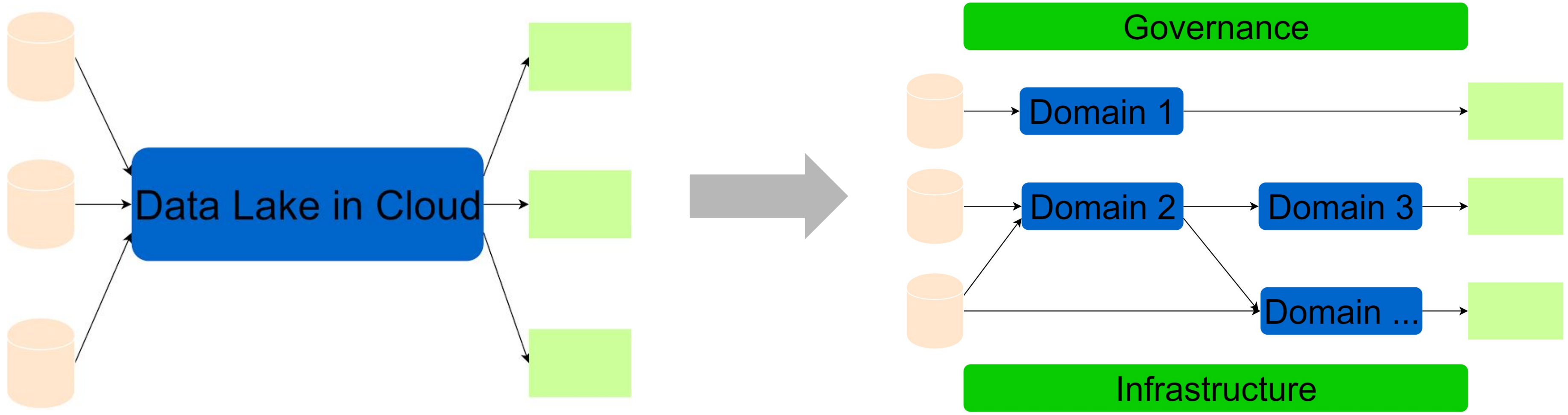
Site feedback

Source: https://www.microsoft.com/en-us/itshowcase/microsoft-tames-the-wild-west-of-big-data-with-modern-data-management

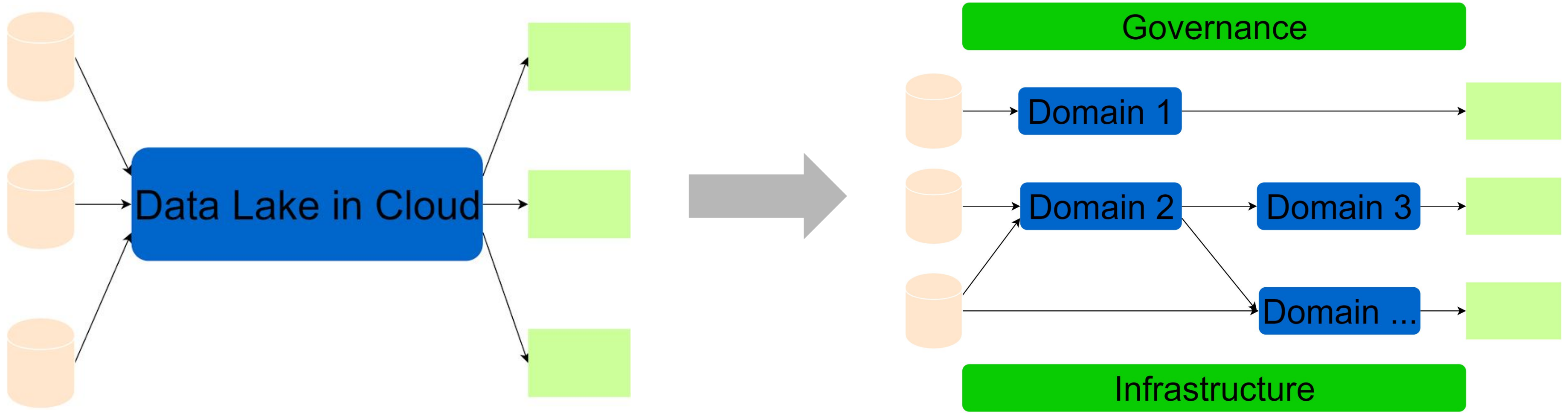


Source: https://www.microsoft.com/en-us/itshowcase/microsoft-tames-the-wild-west-of-big-data-with-modern-data-management

Instead, there might be a distributed approach



Instead, there might be a distributed approach



What does it mean?

- Multiple storages instead of one instance? Silos?
- What are “Infrastructure” and “Governance” boxes?
- Who puts things in production?
- Is it the same as a single Data Lake but with restricted access?

Data Mesh

“The data mesh platform is an intentionally designed distributed data architecture, under centralized governance and standardization for interoperability, enabled by a shared and harmonized self-serve data infrastructure.

[...] it is far from a landscape of fragmented silos of inaccessible data.”

— Zhamak Dehghani

Data Mesh

Domain Driven Design

Autonomous teams with clear bounded context building and running products independently

Product Thinking

Produce datasets, both raw and transformed. Ensure data lifecycle, versioning, SLA, documentation, quality

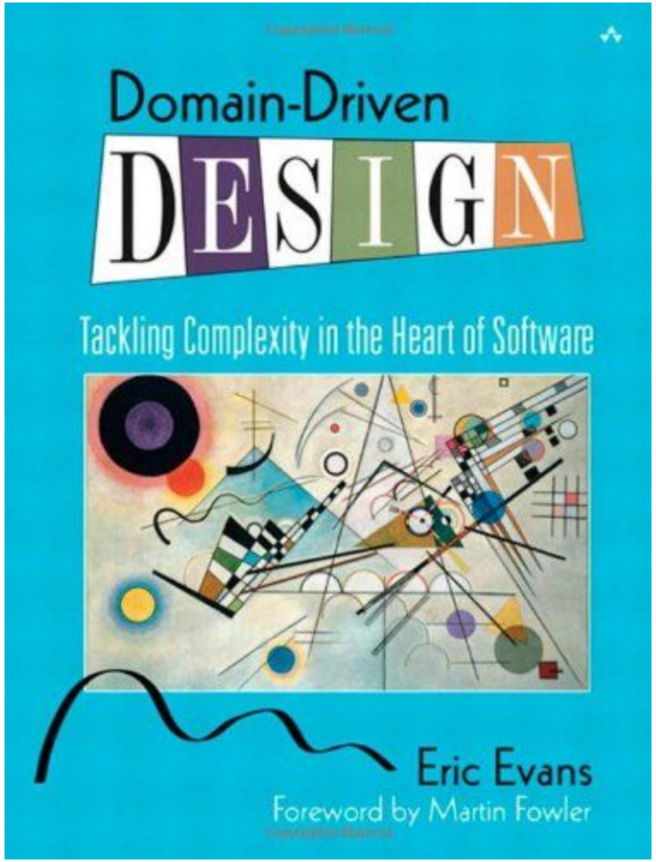
Platform Thinking

Create self-service platform for storage, computation, access rights, pipelines, etc.

Data Mesh - Domain Driven Design

Key features:

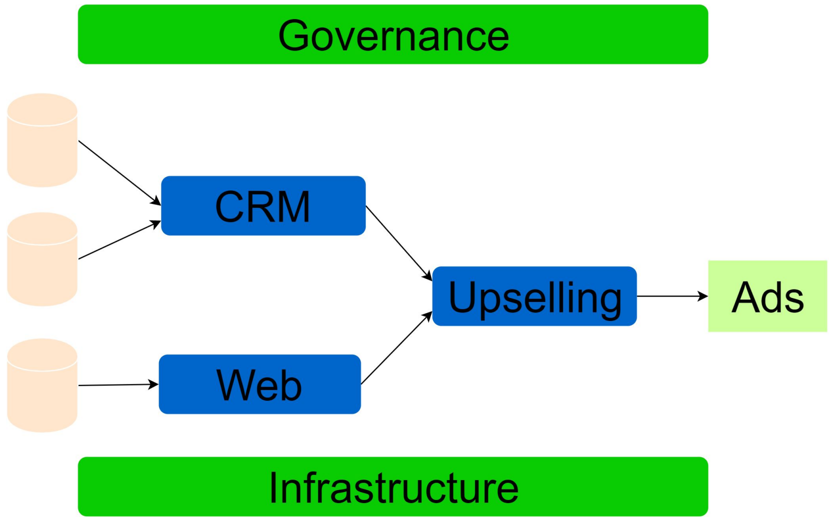
- Bounded context of a domain
- Model to describe a domain
- Unified language
- Actors


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Data Mesh - Domain Driven Design

Domain team definition:

Independent cross-functional teams who have embedded data engineers and data product owners, using common data infrastructure as a platform to host, prep and serve their data assets



- Three domains (CRM, Web, Upselling)
- Each has responsibility to prepare and share their models
- Unified language (e.g. what is a customer)
- Actors

Data Mesh - Product Thinking

Key features:

- Raw or transformed datasets
- Domain team is responsible for its lifecycle, SLA
- Discoverable, addressable, trustworthy, self-describing, interoperable, secure
- Each producer is responsible of sharing data products to organization

Data Mesh - Product Thinking

Key features:

- Raw or transformed datasets
- Domain team is responsible for its lifecycle, SLA
- Discoverable, addressable, trustworthy, self-describing, interoperable, secure
- Each producer is responsible of sharing data products to organization

Example properties:

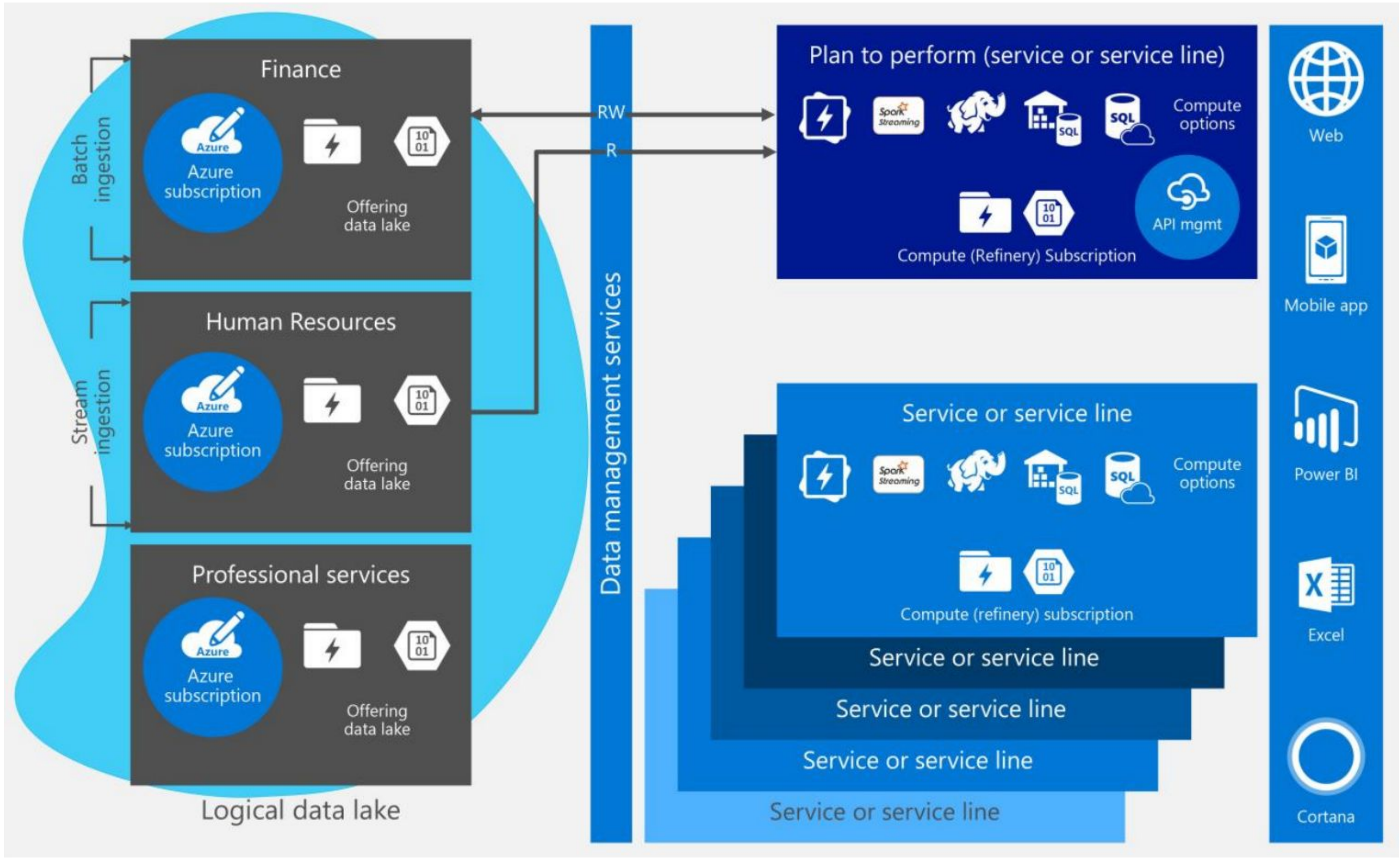
Unique_id: *<guid>*
Owner: *<domain_name>*
Schema: *<path_to_schema>*
Refresh_rate: *<minute/hour/day>*
Sensitive_columns: *<list_of_sensitive_columns>*
Description: *<text>*
History: *<up to X years>*
Storage: *<no sql/data lake/sql>*
Access_group: *<ad group X>*
Dependencies: *<dataset a, b, x>*
….

Data Mesh - Platform Thinking

Key features:

- Infrastructure for self-service
- Polyglot storages, tools, compute
- Administer data platform (monitoring, alerting, log)
- Data catalogue & governance
- Ensure security (encryption, firewalls, etc.)
- Data access control and logging
- Domain agnostic

Data Mesh - Platform Thinking



Data Mesh - Platform Thinking

Considerations:

- What tools enable support such distributed architecture?
- Are cloud services mature enough for it?
- What is the best tool for orchestration?
- What data catalogue is preferred?

What do you think?

-

Let’s forget cloud specifics

Let’s not focus on solving specific product limitations, but instead focus on a bigger picture

“There really is no magic, when you have done certain task enough times, you started to see patterns that can be automated.” Maxime Beauchemin

Distributed Data Lake / Data Mesh