Member-only story

# From Data Warehouses and Lakes to Data Mesh: A Guide to Enterprise Data Architecture

Understand how data works at large companies

Col Jung · Follow
Published in Towards Data Science · 18 min read · May 12, 2023



Image: Headway (Unsplash)

There's a disconnect between data science courses and the *reality* of working with data in the real world.





When I landed my first analytics job at one of Australia's 'Big Four' banks half a decade ago, I was confronted by a complex data landscape characterised by...

- Challenges in **finding**, **accessing** & **using** data;
- **Competing** business priorities pulling people in different directions;
- **Legacy** systems that are difficult to maintain & upgrade;
- A legacy **culture** resistant to data-driven insights;
- **Siloed** teams who didn't talk to each other.

For a while, I plodded on and resigned myself to the idea that perhaps this was just the way things were in the world of enterprise data. I held faith that while our tech stack evolved at a really fast pace, UX would eventually catch up...

I had trained myself in data science but actually getting to *do* data science wasn't straight-forward at all. Online courses don't prepare you for this.

But here's the kicker.

After some digging, I realised that my organisation wasn't alone in facing these data challenges — they were **pervasive across the industry**.

We're in a melting pot of technological innovation where things are moving at breakneck speed. Compute power is scaling fast, machine learning applications have become ubiquitous, cutting-edge generative AI and AI art is disrupting every industry with no signs of stopping, and consumer expectations are ever-changing.

Everyone involved in the analytics industry is just trying to find their footing. We're all stumbling forward together. Fail fast and fail forward.

That's why I penned this article.

I want to share my insights and help professionals like graduates, new business analysts and self-taught data scientists quickly understand the data

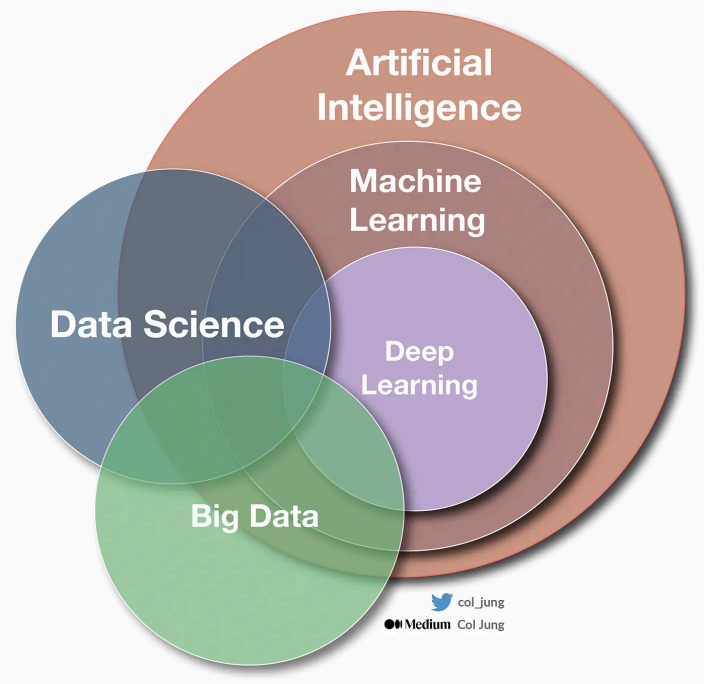
landscape at the enterprise level and shape expectations.

### 1. Data is the Lifeblood of Digital

Let's first align on the crucial role data plays in today's competitive fast-paced business environment.

Companies in every industry are moving towards data-driven decision-making.

At the same time, consumers are increasingly expecting **hyper-personalised digital products & services** that leverage powerful analytics like AI and machine learning that's trained on all the quality data the company can muster.




How the worlds of AI & machine learning intersect with enterprise analytics. Image by author





It's what enables you to watch personalised TV shows on demand (*entertainment*), order food and have it delivered within an hour (*groceries & shopping*), and get a pre-approved mortgage in minutes (*housing*).

This means a forward-thinking *data stack* is essential to survive and thrive, because **data is the lifeblood of digital.**

Or as British mathematician Clive Humby put it in 2006:

> *"Data is the new oil."*

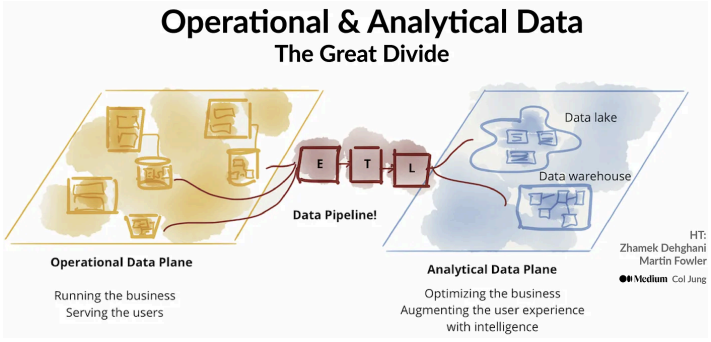
IT departments and data platforms are no longer basement-dwellers — they're now a **core part** of the enterprise strategy.

**Data is now a first-class citizen.**

Because data powers everything.

So without further ado, let's now dive into how data is organised, processed and stored at large companies.

Looking down from our helicopter, you'll see that the landscape is divided into **operational data** and **analytical data.**




30,000 feet view of the enterprise data landscape. Source: Z. Dehghani at MartinFowler.com with amendments by author

Read my *Explainer 101* on enterprise data strategy.

## 2. Operational (and Transactional) Data

**Operational data** often comes in the form of individual records that represent specific events, such as a sale, purchase, or a customer interaction, and is information a business relies on to run its day-to-day operations.

Operational data is stored in databases and is accessed by *microservices*, which are small software programs that help manage the data. The data is constantly being updated and represents the current state of the business.

**Transactional data** is an important type of operational data. Examples of *transactions* in banking include:

- money moving between bank accounts;
- payments for goods and services;
- a customer interaction with one of our channels, e.g. branch or online.

Transactional data that's 'hot off the application' is called *source data*, or *System-of-Record (SOR)*. Source data is free of transformations and is the…

- preferred data format by *data scientists*;
- format of data ingested into *data lakes*;
- beginning of any *data lineage*.

More on these ideas later.

Transactional data processing systems, called **Online Transactional Processing (OLTP)** systems, must handle many transactions quite fast. They rely on databases that can quickly store and retrieve data, and ensure the data stays accurate by enforcing rules called *ACID semantics*:

- *Atomicity* — each transaction is treated as a single unit. We don't want to mix together two wire transfers or purchases!





- *Consistency* — transactions must pass or fail. My roast pork purchase either went through or it didn't!
- *Isolation* — multiple transactions can happen at the same time without interfering with each other. A core tenet of scalability.
- *Durability* — data changes are saved even if the system shuts down. Losing operational data will land your business in hot water.

OLTP systems are used for important business applications that need to work accurately, quickly and at scale.

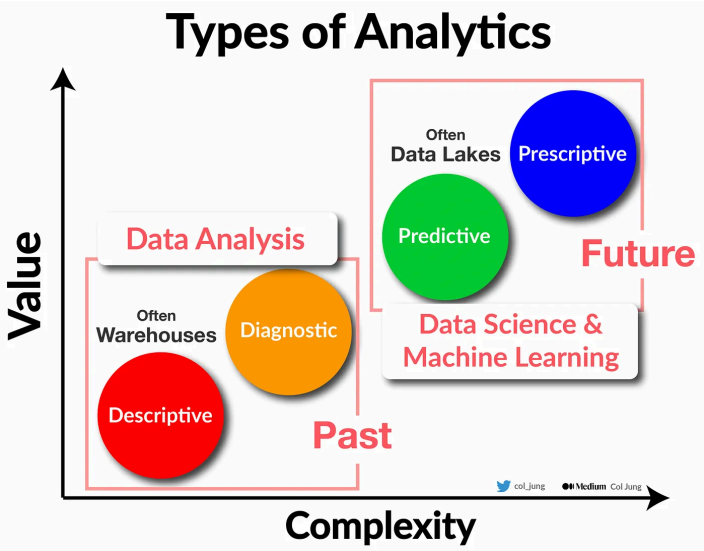
In banking, OLTP systems process deposits, withdrawals, transfers and balance enquiries, which include online banking apps, credit and debit authorisation systems, cheque processors and wire transfer systems that send money between banks.

As you can see, OLTP systems are typically the bread-and-butter interfaces between businesses and their customers.

## 3. Analytical Data

**Analytical data** is a temporal (time-based) and aggregated (consolidated) view of a company's operational or transactional data. This provides a summarised view of the *facts* of the organisation over time, designed to:

- gain insights into **past** business performance (*descriptive* and *diagnostic* analytics);
- make data-driven decisions for the **future** (*predictive* and *prescriptive* analytics).




From descriptive analysis to predictive modelling. Image by author

Analytical data is frequently used to create dashboards and reports (often built by **data analysts**) and train machine learning models (**data scientists**).

Check out my step-to-step guides on how to train regression and classification models with Python — bread and butter skills of contemporary data science.

Enterprises are moving towards increasingly powerful business intelligence (BI) tools and codeless machine learning platforms in an effort to ***democratise*** data and analytics capabilities.

The idea is many companies today possess only pockets or *silos* of advanced analytics skills. The game-changer lies in empowering 10,000 non-technical colleagues across the entire organisation with the right skillset and tools, providing a boost in *overall* productivity that surpasses the marginal benefits polishing up a specialist 20-member data science team. (Sorry to my data science pals!)

So keep your eyes on **data democratisation** — it's a huge thing right now.





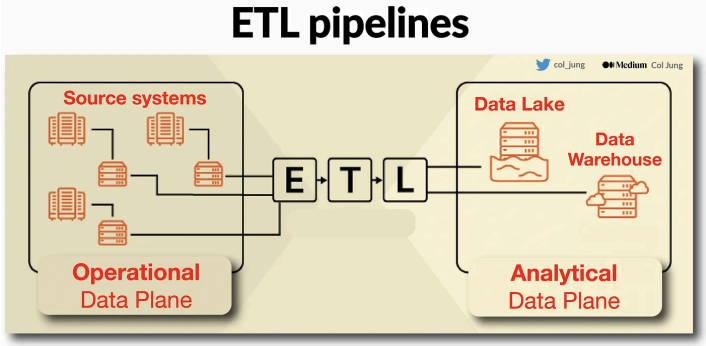
Overall, analytical *processing* is different from transactional processing because the former is focused on *analysing* data while the latter on *recording* specific events.

Analytical processing *systems* typically leverage read-only systems that store vast volumes of historical data or business metrics. Analytics can be performed on a snapshot of the data at a given point in time.

Now, let's connect the dots between operational and analytical data.

Operational data is transformed into analytical data through **data pipelines**, typically built by **data engineers**.

These 'pipelines' are typically **ETL pipelines** — which entails ***extracting*** the data from operational systems, ***transforming*** it for one's business needs, and ***loading*** it into a data warehouse or data lake, ready for analysis.




ETL pipelines connect operational and analytical data stores. Image by author

Read my *Explainer 101* on data democratisation.

### 4. Data Warehouses & Data Lakes

The entire analytical data plane — where the enterprise stores its analytical data — has diverged into two main architectures and technology stacks:

- Data Warehouses;

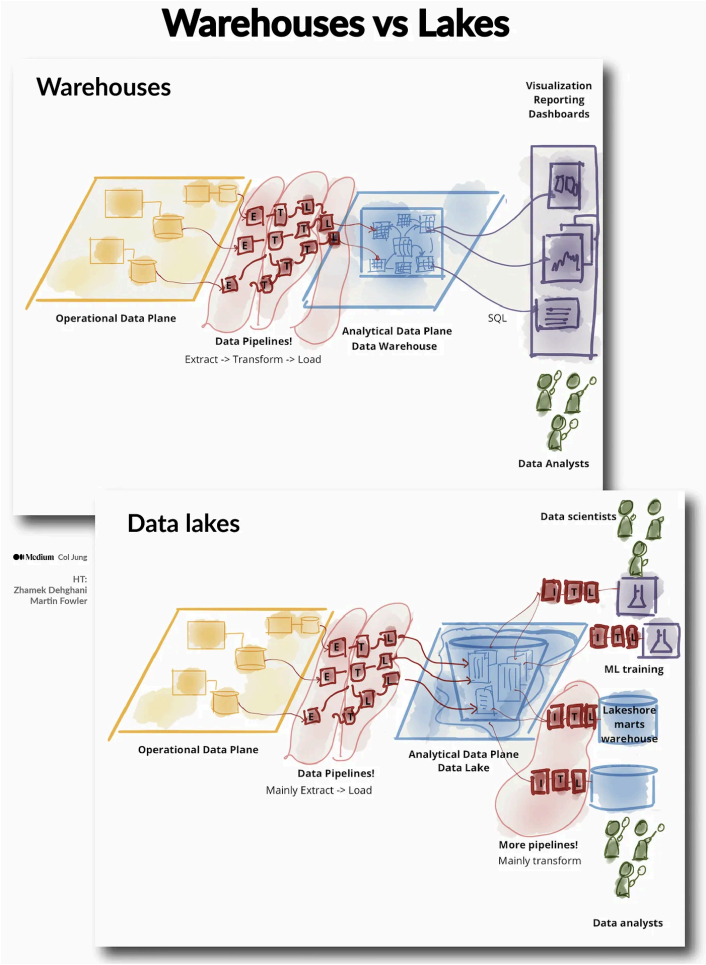
- Data Lakes.

Different users might perform data work at different stages within the enterprise architecture.

- **Data analysts** often query tables and aggregate data in the data warehouse to produce effective dashboards, reports and visualisations, which business users and decision-makers consume downstream.
- **Data scientists** often work in a data lake to explore data in ***shadow production environment***. This means prototyping their data wrangling and modelling in a *developer* (i.e. non-production) environment on *live* (i.e. production) data that's been meticulously prepared by **data engineers**. Once the business signs off on the value of the models, **ML engineers** operationalise them into production so the model can serve both internal and external customers at scale under the watch of an 24/7 operations team (MLOps).








Warehouses vs data lakes. Source: Z. Dehghani at MartinFowler.com with amendments by author

For those new to enterprise IT, there are two main types of *environments* you need to grasp:

- **Non-production**, where you build and try stuff. Change is cheap and breaking things don't bring down your business. Also known as a

**developer** environment. Projects are funded by the organisation's **capital expenditure (CapEx)**.

- **Production** where you deploy and serve your finalised and signed-off apps, data, pipelines and systems to real customers. Your work is now **live**. Make sure it's good, because change is expensive. *Prod* — as it's colloquially called — are highly-secure locked-down environments taken care of by an *operations* or *run team* that's funded by the org's **operational expenditure (OpEx)**. I wrote more on CapEx vs OpEx here.

In short, build stuff in non-prod, deploy it into prod. Gotcha!

Alright, let's now dive into some details of both data architectures.

### 4.1 Data Warehouses

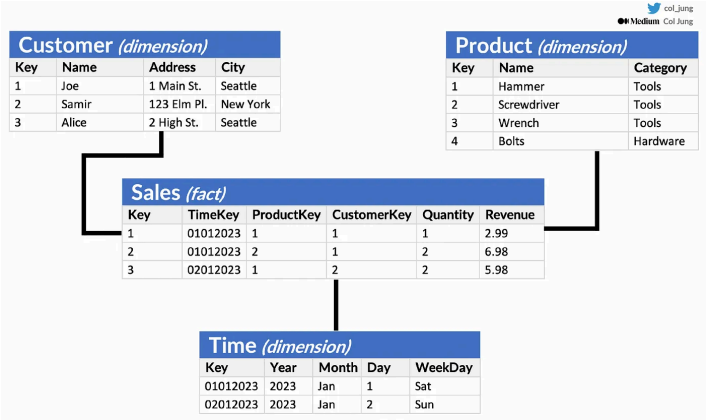
**Data Warehouses** are an established way to store **structured data** in a **relational schema** that's optimised for read operations — primarily SQL queries to support BI, reporting and visualisations.

Some features of warehouses:

- **Historical analysis:** Data warehouses have been the mainstay for descriptive analytics for decades, offering the ability to query and join large volumes of historical data quickly.
- **Schema-on-Write:** Data warehouses traditionally employ a *Schema-on-Write* approach, where the structure, or **schema**, of your tables are defined upfront.

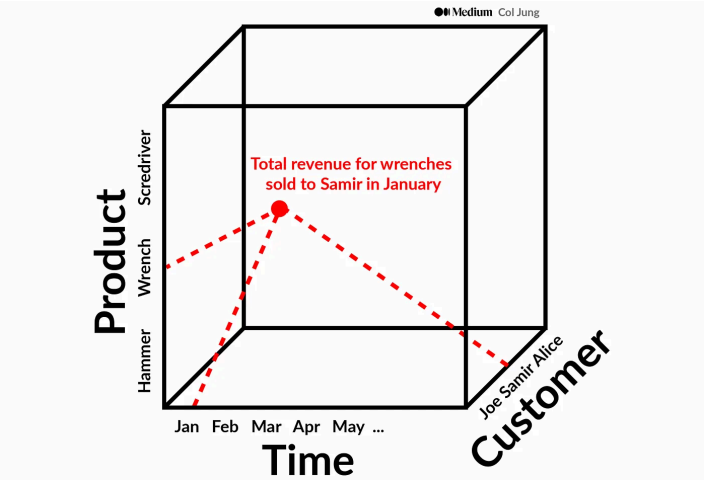







A common star schema. Image by author

- **Data Modelling:** While data analysts and data scientists can work with the data directly in the analytical data store, it's common to create **data models** that pre-aggregate the data to make it easier to produce reports, dashboards, and interactive visualisations. A common data model — called the **star schema** — is based on **fact** tables that contain numeric values you want to analyse (for example, some amount relating to ***Sales***), which are *related* to — hence called a *relational* database — **dimension** tables representing the entities (e.g. ***Customer*** or ***Product***) you want to measure.
- **Fast queries:** Data in warehouses may be aggregated and loaded into an **Online Analytical Processing (OLAP)** model, also known as the **cube**. Numeric values (**measures**) from fact tables are pre-aggregated across one or more dimensions — for example, total revenue (from the fact ***Sales*** table) by the dimensions ***Customer***, ***Product*** and ***Time***. Visually, this looks like the intersection of the 3 dimensions in a 3D cube. Benefit-wise, the OLAP/cube model captures relationships that support "drill-up/drill-down" analysis, and queries are fast because the data is pre-aggregated.




The "cube". Measures (e.g. sales) are aggregated by dimensions time, customer & product. Image by author

- **File types:** Structured data files include readable formats like *CSV* and *XLSX* (Excel), and optimised formats like *Avro*, *ORC* & *Parquet*. Relational databases can also store semi-structured data like *JSON* files.

Read my *Explainer 101* on data warehouses and data modelling.

### 4.2 Data Lakes

Data Lakes are the de facto industry approach to store a large volume of file-based data to support data science and large-scale analytical data processing scenarios.

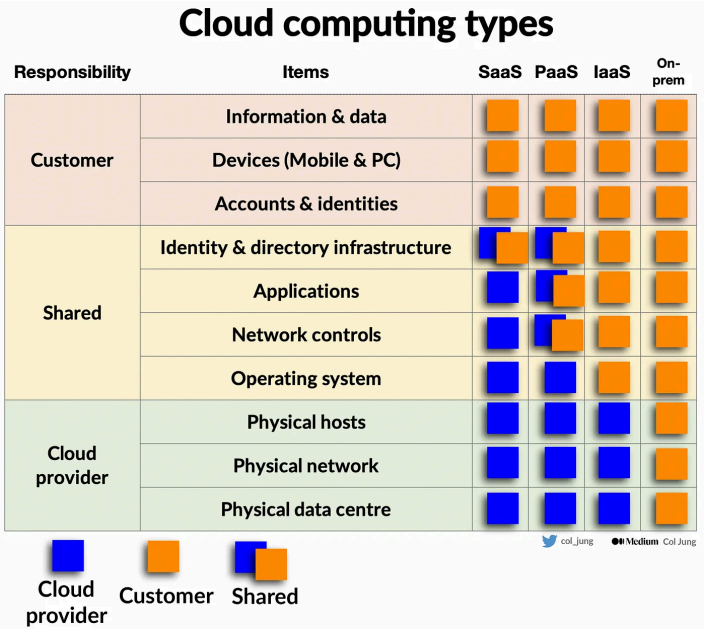
- **Distributed compute & storage:** Data lakes use distributed compute and distributed storage to process and store huge volumes of potentially **unstructured data**. This means the data is held and processed across potentially thousands of machines, known as a **big data cluster**. This technology took off in the 2010's, enabled by **Apache Hadoop**, a collection of open-source big data software that empowered organisations to distribute huge amounts of data across many machines (*HDFS* distributed storage) and run SQL-like queries on tables stored across them (*Hive* & *Spark* distributed compute). Companies like *Cloudera* and *Hortonworks*





later commercialised Apache software into packages that enabled easier onboarding and maintenance by organisations around the world.

- **Schema-on-Read:** Data lakes use a *Schema-on-Read* paradigm where a schema is only created when the data is read. This means data can be dumped in the lake en-masse without the costly need to define schemas immediately, while allowing for schemas to be created for specific use cases down the line — precisely the kind of flexibility that data scientists require for modelling.
- **File types:** Data lakes are the home of **unstructured data** — this include text files like *txt* & *doc*, audio files like *MP3* & *WAV*, images like *JPEG* & *PNG*, videos like *MP4*, and even entire PDFs, social media posts, emails, webpages and sensor data. Data lakes (and NoSQL databases) also allow you to store your **semi-structured data**, like *JSON* and *XML* files, as-is.
- **Cloud computing:** Data lakes are increasingly being hosted on public cloud providers like *Amazon Web Services*, *Microsoft Azure* and *Google Cloud*. This *elastic* and *scalable* infrastructure enables the organisation to automatically and quickly adjust to changing demands in resources in both compute and storage while maintaining performance and paying only for exactly what you use. There are three common types of cloud computing, with different divisions of shared responsibility between the cloud provider and client. The most flexible **Infrastructure-as-a-Service (IaaS)** allows you to essentially rent empty space in the data centre. The cloud provider maintains the physical infrastructure and access to the internet. In contrast, the **Software-as-a-Service (SaaS)** model has the client renting a fully-developed software solution ran through the internet (think *Microsoft Office*). For enterprise data, the most cloud popular model is the middle ground **Platform-as-a-Service (PaaS)** where the provider chooses the OS with the client able to build its data architecture and enterprise applications on top.




Cloud computing types & shared responsibility model. Image by author

- **Real-time analytics & streaming:** Technologies like *Apache Kafka* has enabled data to be processed near real-time as a perpetual *stream* of data, enabling the ability for companies to unlock *instant* insights and trends or take *immediate* responsive action to events as they occur. Take the seamless delivery of timely product and service recommendations straight to a customer's mobile phone, *precisely* when and where they're most likely to benefit from them — like my bank offering me travel insurance when I reach the airport. Or consider the ability to proactively alert customers in real-time when they're about to transfer money to scammers, thereby safeguarding their financial well-being. This almighty fusion of **big data, fast & low-latency compute** and **machine learning** has been the key enabler of hyper-personalised services in every industry worldwide.

Read my *Explainer 101* articles on cloud computing and enterprise generative AI. Also see my article on Azure Synapse & Fabric — Microsoft's attempt to bring data warehouses and big data under one roof.





Ps…enjoying this story? Get an email when I post similar articles.

## 5. Data Mesh & Data Products

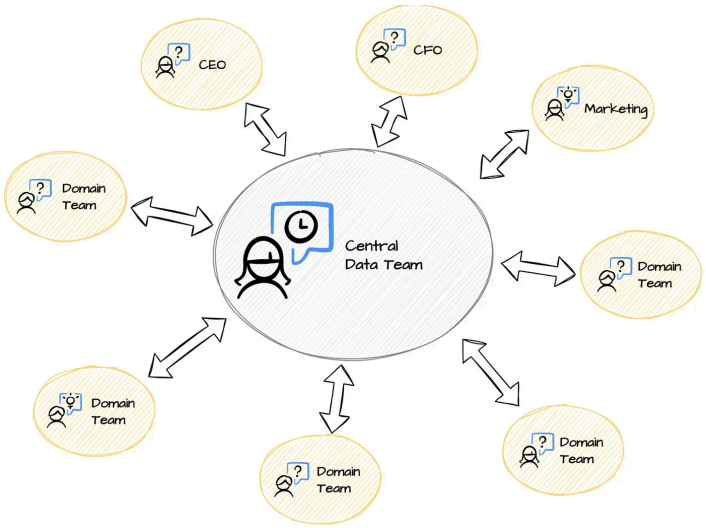
Architect Zhamek Dehghani condensed the evolution — challenges, progress and failings— of the enterprise data landscape across three generations:

> ***First generation:*** proprietary enterprise data warehouse and business intelligence *platforms; solutions with large price tags that have left companies with equally* ***large amounts of technical debt [in the form of] thousands of unmaintainable ETL jobs****, and tables and reports that only a small group of specialised people understand, resulting in an under-realized positive impact on the business.*
>
> ***Second generation:*** big data ecosystem with a data lake as a silver bullet*; complex big data ecosystem and long running batch jobs operated by a central team of hyper-specialised data engineers have created* ***data lake monsters*** *that at best has enabled pockets of R&D analytics; over-promised and under-realised.*
>
> ***Third (and current generation) data platforms:*** *more or less similar to the previous generation, with a modern twist towards streaming for real-time data availability with architectures, unifying the batch and stream processing for data transformation, as well as fully embracing cloud-based managed services for storage, data pipeline execution engines and machine learning platforms.*

The current data lake architecture can be summarised as:

- **Centralised.** All analytical data is stored in one place, managed by a central data engineering team that don't have domain knowledge on the data, making it difficult to unlock its full potential or fix data quality issues stemming from source. Opposite of a *decentralised* architecture that federates data ingestion to teams across the business.
- **Domain-agnostic.** Architecture that strive to serve everyone without specifically catering for anyone. A jack-of-all-trades platform. Opposite of a *domain-driven* architecture whereby data is owned by the different business domains.

- **Monolithic**. The data platform is built as one big piece that's hard to change and upgrade. Opposite of a *modular* architecture allowing individual parts or micro-services to be tweaked and modified.




A central data team manages a monolithic domain-agnostic data lake (or is it data monster?). Source: Data Mesh Architecture (with permission)

The problems are clear and so appears to be some of the solutions.

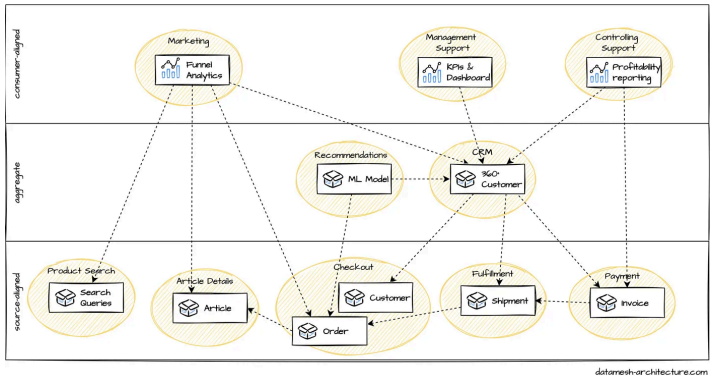
Enter **data mesh**.

Data mesh is the next-generation data architecture that moves away from a single centralised data team towards a *decentralised* design where data is owned and managed by teams across the organisation that understands it the most, known as **domain-driven ownership**.

Importantly, each business unit or domain aims to infuse *product thinking* to create quality and reusable **data products** — a self-contained and accessible data set treated as a product by the data's *producers* — which can then published and shared across the mesh to *consumers* in other domains and business units — called **nodes** on the mesh.








Data Mesh: Individual business units share finely crafted data built to a 'product standard'. Source: Data Mesh Architecture (with permission)

Data mesh enables teams to work independently with greater autonomy and agility, while still ensuring that data is consistent, reliable and well-governed.

Here's an example from my job.

Right now, data for our customers along with their transactions, products, income and liabilities are sitting in our centralised data lake. (And across our data warehouses too.)

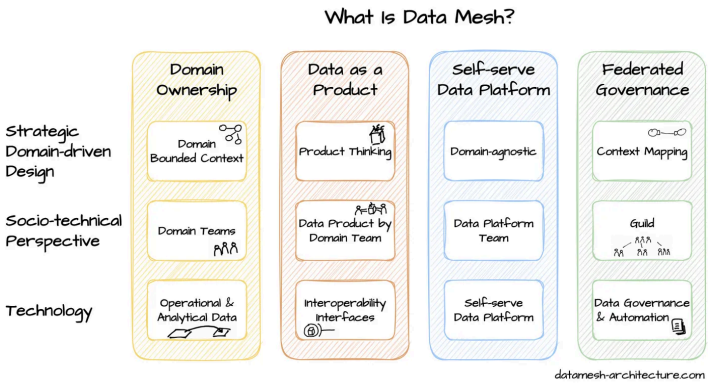
In the future, as we federate our capabilities and ownership across the bank, the ***credit risk*** domain's own data engineers can independently create and manage their data pipelines, without relying on a centralised ingestion team far removed from the business and lacking in credit expertise.

This credit team will take **pride** in building and refining high-quality, strategic, and reusable data products that can be shared to different nodes (business domains) across the mesh, providing the ***mortgages*** team with reliable credit information to make better decisions about approving home loans.

These same data products can also be utilised by the ***consumer credit*** domain to develop machine learning models to better understand the behaviours of

our credit card customers so that we may offer them better services and identify those at risk.

These are examples of leveraging the strategic value of data products within the mesh.




Four principles of mesh: domain-ownership, data as a product, self-serve platform, federated governance.
Source: Data Mesh Architecture (with permission)

Data mesh fosters a culture of data ownership and collaboration where data is treated as a **first-class citizen** that's furthermore **productised** and seamlessly shared across teams and departments, rather than languishing in a entangled web of often-duplicated ETL pipelines crafted by siloed teams for specific ad hoc tasks.

Data mesh pushes organisations away from a costly and inefficient **project-based mindset** towards a scalable and forward-thinking **product-based mindset**.

Read my *Explainer 101* articles on data products and data mesh.

## 6. Data Governance

Data governance is like a big game of *Who's the Boss*, but for data. Just like the show, there are a lot of complicated relationships to navigate.

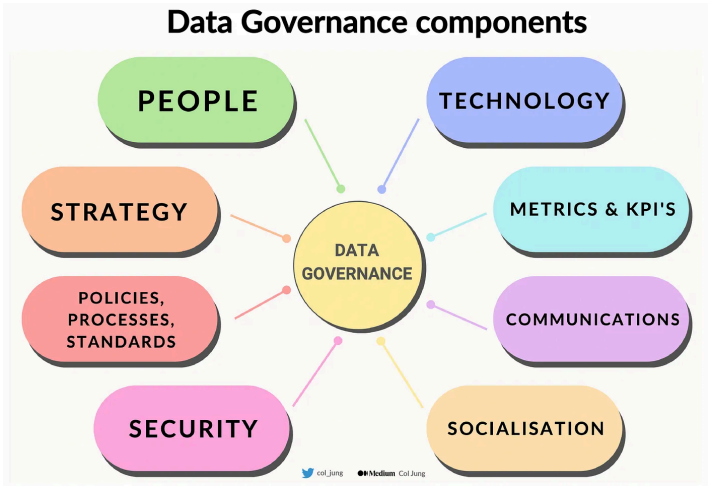




It's about figuring out who's in charge of what data, who gets to access it, who needs to protect it and what controls and monitoring is in place to ensure things don't go wrong.

With my workplace boasting 40,000 employees, tons of processes, and competing priorities, it can feel like a real challenge to maintain order and ensure everyone is on the same page.

To data analysts, data scientists and developers, data governance can feel like that annoying friend who always wants to know what you're up to. But they're absolutely necessary for organisations, especially well-regulated ones. Otherwise, it would be like a circus without a ringmaster — chaotic, impossible to manage, and **very risky**.




Data governance components. Image by author

Some core considerations of data governance are:

**Data privacy.** It's like trying to keep your embarrassing childhood photos hidden from the world. But for businesses, it's a lot more serious than just dodgy haircuts. Let's say a bank accidentally reveals all of its customer's

financial info. That's gonna cost them a ton of cash, and even more importantly, trust.

**Data security**. You want to make sure that your customer's data is safe from both external threats (like hackers) and internal threats (like rogue employees). That means robust authentication systems, fault-tolerant firewalls, ironclad encryption technologies and vigilant 24/7 cybersecurity. Nobody wants their data ending up on the dark web auctioned to criminals.

**Data quality**. Think of making a sandwich — put in rotten ingredients and you'll end up with a gross meal. If your data quality stinks, you'll end up with unreliable insights that nobody wants to bite into. And if you're in a regulated industry, you better make sure your sandwich is made with fresh ingredients, or your data might not pass your compliance obligations.

Maintaining reliable information on how data flows through the enterprise — a.k.a **data lineage** — is crucial for ensuring data quality and troubleshooting when things go wrong.

Having weak data privacy, security and/or quality means more **data risk**.

This is where **data ownership** comes in. Who gets to call the shots and make decisions about the data? Who *owns the risk* of things going wrong?

In practice, it's a bit like a game of hot potato where nobody really wants to hold onto the potato for too long. But somebody has to take responsibility for it, so we can avoid data mishaps and keep our potato hot, fresh and secure.

The move towards towards mesh aims to:

- enhance data quality across the board (via reusable data products);
- optimise data ownership (have appropriate domains own their data);
- simplify data lineage (byebye distant ETLs into a centralised data lake).

See my *Explainer 101* article on data governance.





## 7. Final Words

The realm of enterprise-level data can often be a perplexing one, marked by the accumulation of technical debt resulting from a cycle of experimentation followed by overcorrection, not unlike the fluctuations of the stock market.

While the tales of large companies are each unique, they share a few common threads. One such thread is the organic expansion towards an unwieldy and daunting enterprise data warehouse, subsequently succeeded by the eager embrace of a *centralised* data lake aimed at saving costs, concentrating expertise, and magnifying the value of data.

This approach brought forth an entirely new set of issues. So back we all go — this time a dramatic swing towards *decentralising* data stacks and federating data management to the teams that best understand their own data.

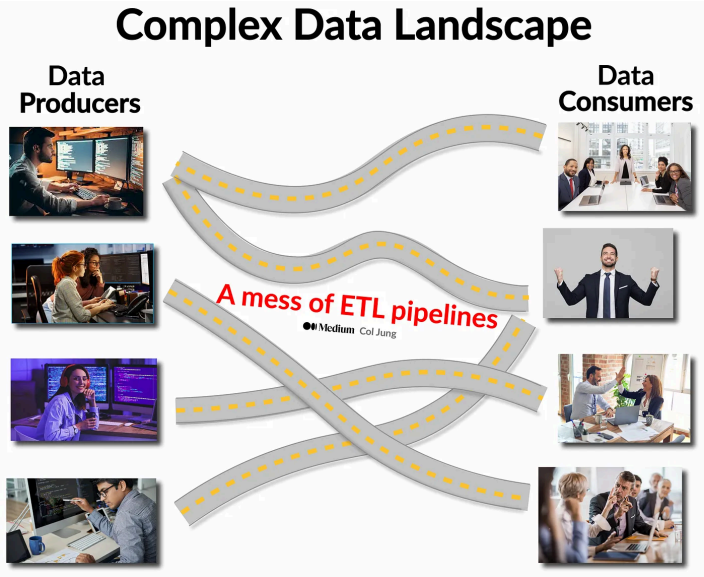
Phew! Like a colony of penguins shuffling along a constantly-shifting expanse of ice.

To give some personal background, the bank I work for has navigated through all the data architecture eras I described in this article.

We spent decades on data warehouses. We then embarked on a now-7-year journey to prop up a strategic data lake intended to become the cornerstone of our data infrastructure.

Long story short, our data warehouses and data lake are still around today, living together in a bit of an awkward marriage of sorts. (It's a work-in-progress...)

We've started our own journey to decentralise this data lake towards mesh. We're busting the spaghetti-like complexity of our data landscape by leveraging the power of reusable data products.




Large companies are presently focusing on busting decades of tech debt in their data landscape. Image by author

And I'm proud to say that among the *Big 4 Banks* of Australia, we're apparently leading the way. This is absolutely delightful because large blue-chip organisations aren't typically at the forefront of technology innovation.

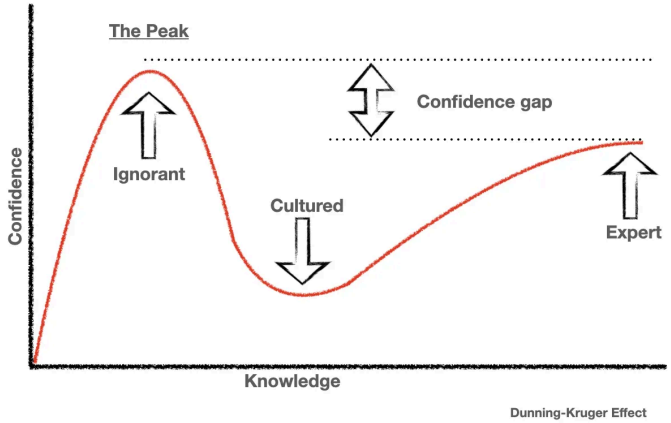
Like many companies, our challenges are big, as all this technical debt is the by-product of hundreds of projects, steered by thousands of colleagues over the years who have come and gone.

My online data science courses — kindly sponsored by my company — taught me how to wrangle data and train logistic regression models and gradient boosted trees, but ill-prepared me for the realities of working with data at large organisations.

On my first day, I thought I'd be handed some nice juicy data on a platter and dive straight into training models.








Hope I'm far along the Dunning Kruger curve?! Source: Wikipedia

Like Forrest Gump discovered, life ain't that simple.

Through trial and failure — I learnt first-hand that there are so many skills that determine your impact as a data scientist beyond what courses offer you — from business engagement to data storytelling to navigating politics and all the nuances of a flawed yet constantly-evolving enterprise data landscape.

By writing this article, I hope I can spare you some of my own stumbles.

Let me know if you relate to these experiences in your own journey!

Find me on Linkedin, Twitter & YouTube.

## My Popular AI, ML & Data Science articles

- AI & Machine Learning: A Fast-Paced Introduction — here
- Machine Learning versus Mechanistic Modelling — here
- Generative AI: How Big Companies are Scrambling for Adoption — here
- ChatGPT & GPT-4: How OpenAI Won the NLU War — here
- GenAI Art: DALL-E, Midjourney & Stable Diffusion Explained — here

- Beyond ChatGPT: Search for a Truly Intelligence Machine — here
- Modern Enterprise Data Strategy Explained — here
- From Data Warehouses & Data Lakes to Data Mesh — here
- From Data Lakes to Data Mesh: A Guide to Latest Architecture — here
- Azure Synapse Analytics in Action: 7 Use Cases Explained — here
- Cloud Computing 101: Harness Cloud for Your Business — here
- Data Warehouses & Data Modelling — a Quick Crash Course — here
- Data Products: Building a Strong Foundation for Analytics — here
- Data Democratisation: 5 'Data For All' Strategies — here
- Data Governance: 5 Common Pain Points for Analysts — here
- Power of Data Storytelling — Sell Stories, Not Data — here
- Intro to Data Analysis: The Google Method — here
- Power BI — From Data Modelling to Stunning Reports — here
- Regression: Predict House Prices using Python — here
- Classification: Predict Employee Churn using Python — here
- Python Jupyter Notebooks versus Dataiku DSS — here
- Popular Machine Learning Performance Metrics Explained — here
- Building GenAI on AWS — My First Experience — here
- Future of Work: Is Your Career Safe in Age of AI — here

Data Science | Machine Learning | Data Architecture | Business | Technology





## Written by Col Jung

Follow

2.3K Followers · Writer for Towards Data Science

Engineer writing about AI, Web3 & Science. https://twitter.com/col_jung https://youtube.com/c/CryptoFilmmaker https://linktr.ee/col_jung

### More from Col Jung and Towards Data Science

Col Jung

**Dismembered in 0.1 Seconds—Byford Dolphin Accident Explained**

Moment of death dissected

9 min read · Jul 22, 2023

1.4K 10

Tim Sumner in Towards Data Science

**A New Coefficient of Correlation**

What if you were told there exists a new way to measure the relationship between two...

10 min read · Mar 31, 2024

2.7K 35

Cristian Leo in Towards Data Science

**The Math Behind Neural Networks**

Dive into Neural Networks, the backbone of modern AI, understand its mathematics,...

28 min read · Mar 28, 2024

Col Jung

**DNA Fried in a Nanosecond—Tōkaimura Nuclear Accident &...**

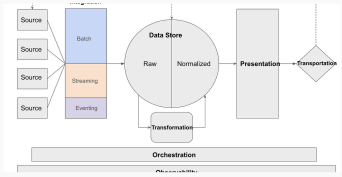
How a worker received 20,000 fatal doses of radiation

14 min read · Jul 31, 2023

2.8K 19 1.3K 10

See all from Col Jung See all from Towards Data Science

## Recommended from Medium




Dave Melillo in Towards Data Science

### Building a Data Platform in 2024

How to build a modern, scalable data platform to power your analytics and data science...

9 min read · Feb 5, 2024

2.7K 38

Mirko Pet... in Mirko Peters—Data & Analytics ...

### Exploring the Foundations of Data Mesh Architecture and...

The evolution of data management in corporate entities has been rapid and...

13 min read · Feb 29, 2024

6

### Lists

**Predictive Modeling w/ Python**
20 stories · 1111 saves

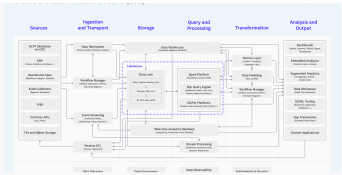
**Practical Guides to Machine Learning**
10 stories · 1326 saves

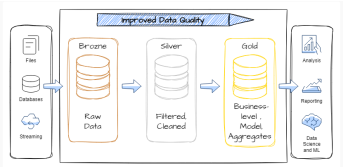
**Natural Language Processing**
1380 stories · 874 saves

**data science and AI**
40 stories · 130 saves



4/19/24, 9:41 PM From Data Warehouses and Lakes to Data Mesh: A Guide to Enterprise Data Architecture | by Col Jung | Towards Data Science






Prem Vishnoi(cloudvala)

### Modern Architecture for merging Data Infrastructures

The growth of the data infrastructure industry has continued unabated since in late 2020.

11 min read · Dec 15, 2023

303 2

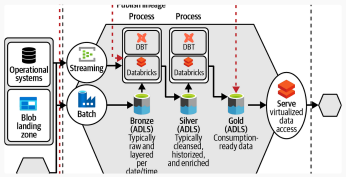
Mariusz Kujawski

### Exploring the Medallion Architecture in Microsoft Fabric

The Medallion architecture stands out as one of the most popular frameworks for...

9 min read · Feb 5, 2024

184 2





Piethein Strengholt

### Data Management at Scale

19 min read · Jul 11, 2023

424 3

RK Iyer

### Striking Gold: Mastering Medallion Architecture for Bronze, Silver, an...

Co-Author—Sen Sayantani

7 min read · Dec 17, 2023

148

See more recommendations