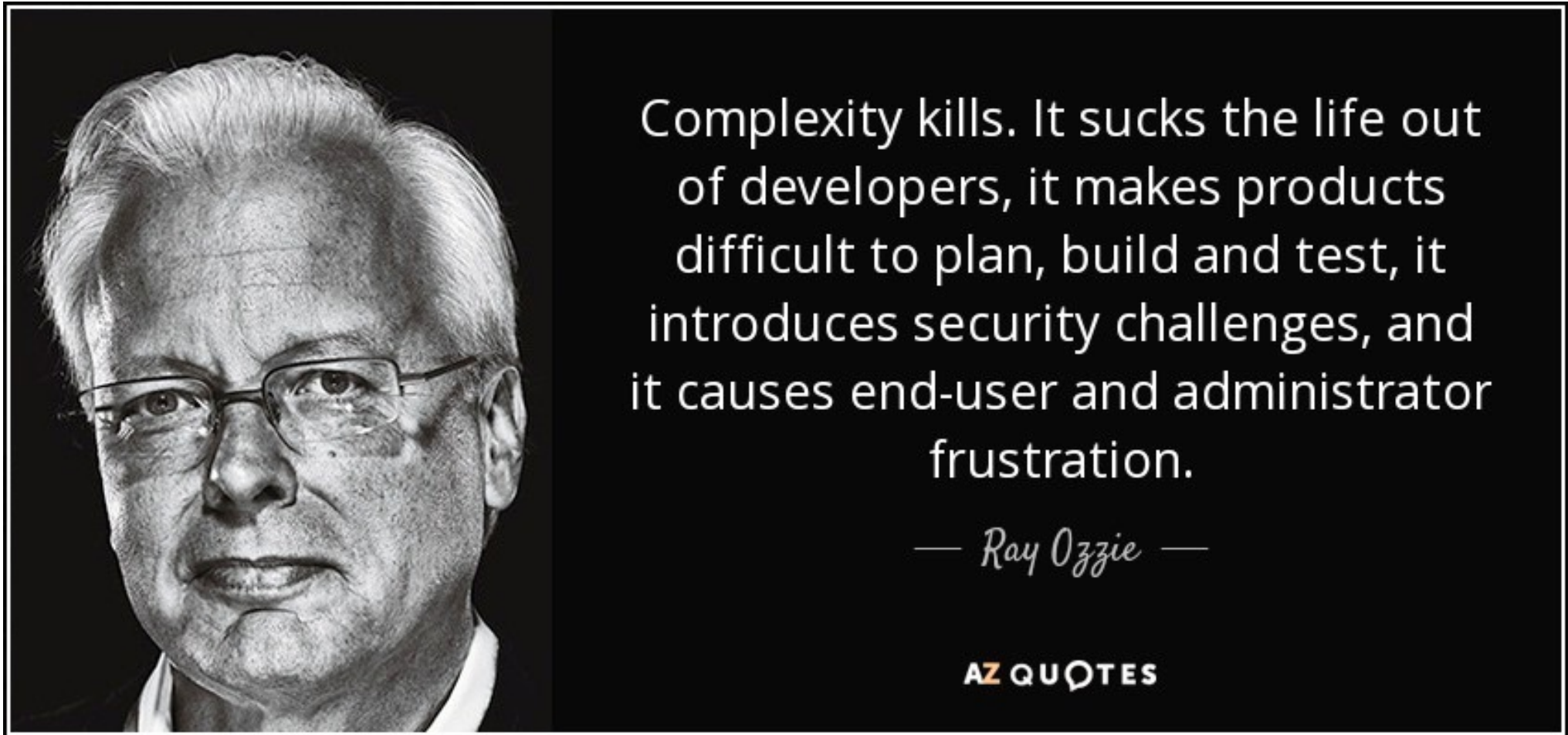




Why? Complexity Kills



Examples of Complexity In Data Teams

- Large teams, multiple functional roles
- 1000s of data sets
- Lots of team customers, each with their unique needs
- Each team member working on their compartmentalized tasks
- Long, interconnected project plans
- A convoluted release to the production process
- Multi-tool, multi-server, multi-tool data architectures
- Thousands of lines of code/configuration

Answers to Complexity in Software Engineering

Software Engineering

- DevOps
- Domain-Driven Design
- Software Products
- Test-driven Development
- Observability
- Event-Driven Architecture
- API
- Functional Programming

From Software Engineering to Data Analytics

Software Engineering

- DevOps
- Domain-Driven Design
- Software Products

Data & Analytics

- > DataOps
- > Data Mesh
- > Data Products

From Software Engineering to Data Analytics

Software Engineering

- DevOps
- Domain-Driven Design
- Software Products
- Test-driven Development
- Observability
- Event-Driven Architecture
- API
- Functional Programming

Data & Analytics

- DataOps
- Data Mesh
- Data Products
- Data Testing
- Data Observability
- Streaming; Kappa Architecture
- Data Contract
- Functional Data Engineering

Definitions

- Data Mesh
 - Organizes your data and analytic team's data work into **decentralized domains**.
 - Instead of boiling the data ocean, they focus on fewer data sets and fewer customers and deliver more insight.
- Data Products
 - All our work in data is part of an ongoing, ever-improving flow of **value** to customers.
 - Success is not defined by completing the task on your project plan; the actual usage and ever-expanding delivery of insight to a customer define it.

Data Mesh & Products

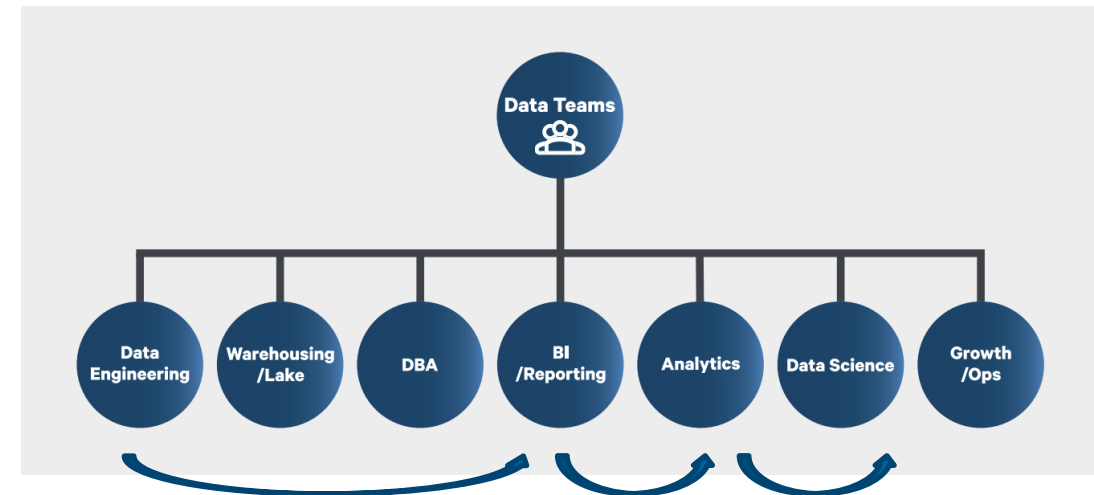


- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

The Human Side of a Data Mesh: Main Idea

Many companies are badly organized to deliver data & analytics effectively

- Many groups organize by data processing capabilities or work task type
- This actually **slows down** the delivery of data features
- The organizational structure builds walls & barriers to the changes
- When you make a change, you need to update each component & coordinate between several different teams



No, Data People Are Not Perfectly Fungible



Data Mesh = Organization Mesh

Assignment of full end-to-end ownership of a domain to one cross-functional team that gets the necessary support to fulfill that responsibility.

Data Organization Keys

Let the small team continually own the data set & not move from project to project is key.

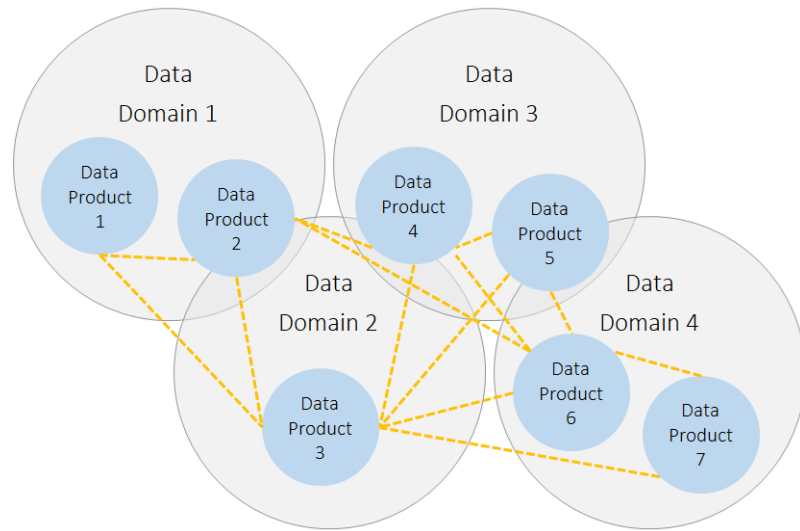
Data Engineers must live, work & understand a finite number of data sets to add value.

Data Mesh & Products



- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

What Data is in a Domain?



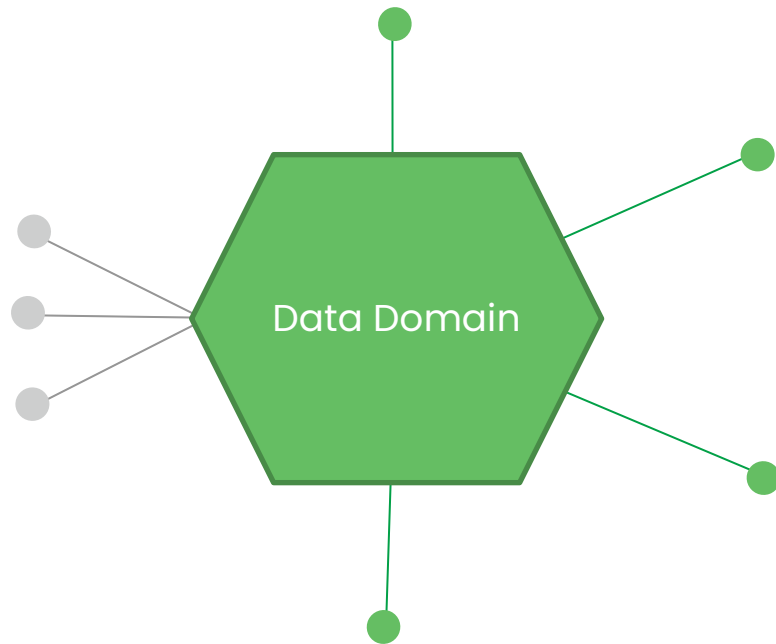
Domains Aligned with Sources / Types of Data

- 'Mastered' Data:
 - Entities of business/subject areas
 - Customers, products, etc.
- 'Sources' of Data:
 - Business reality: facts on the ground
 - Weblogs, user interaction history

Domains Aligned with Consumption of Data

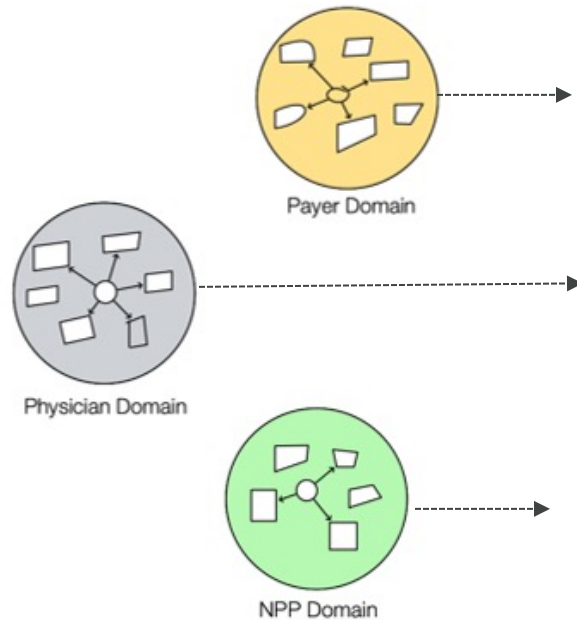
- Integrated Data / Ready for Consumption
 - Facts / Dimensions / Star Schemas
 - Aggregated Views

What are the Domain's Components?



1. **Data**
2. **Artifacts** created from that data: models, views, reports, dashboards, etc.
3. **Code** that acts upon that data: pipelines, toolchains, etc.
4. **The team** used to create/update/run that Domain
5. **Metadata**: catalogs, lineage, test results, processing history, etc.

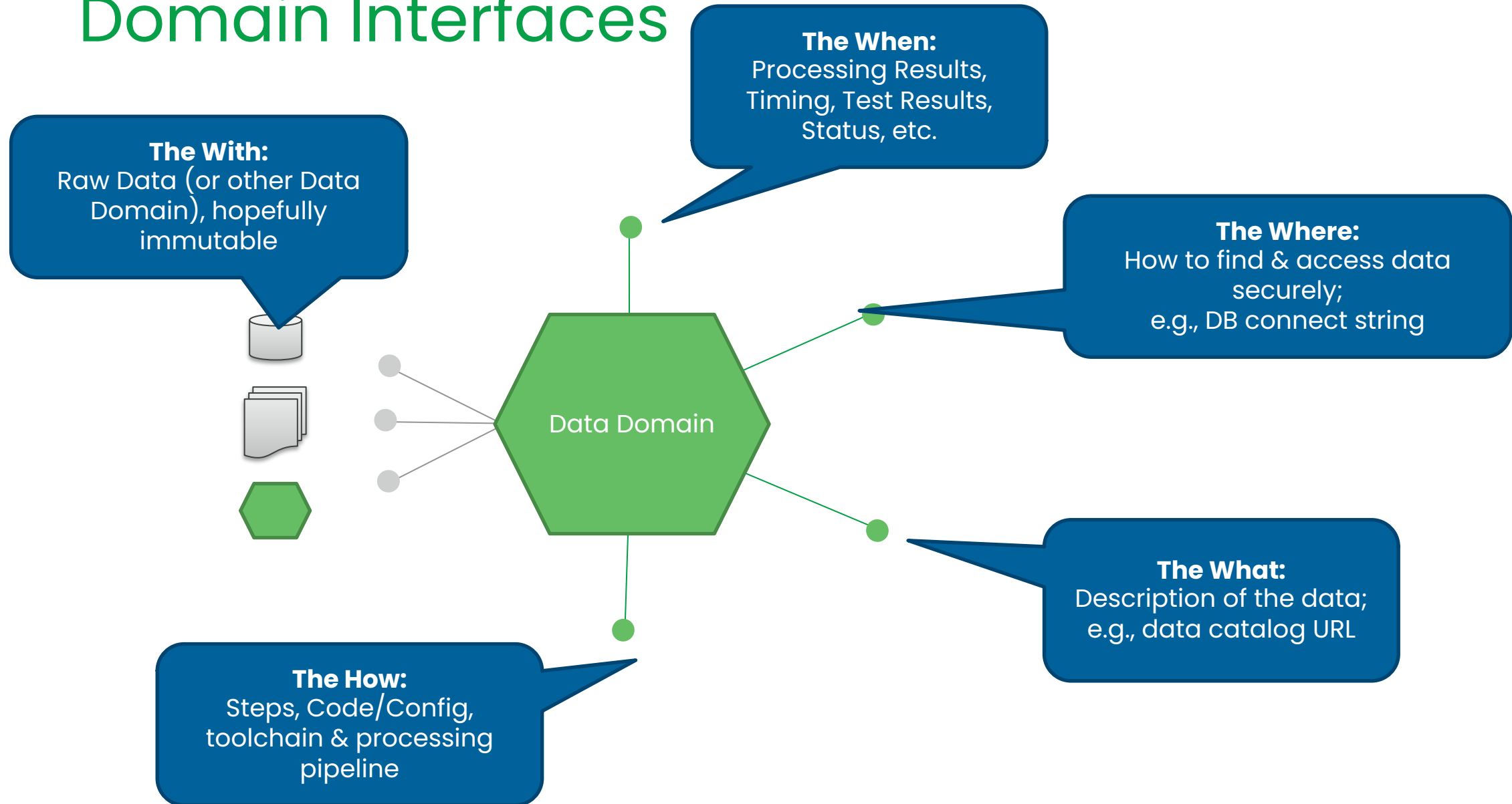
What do you want out of a Domain?



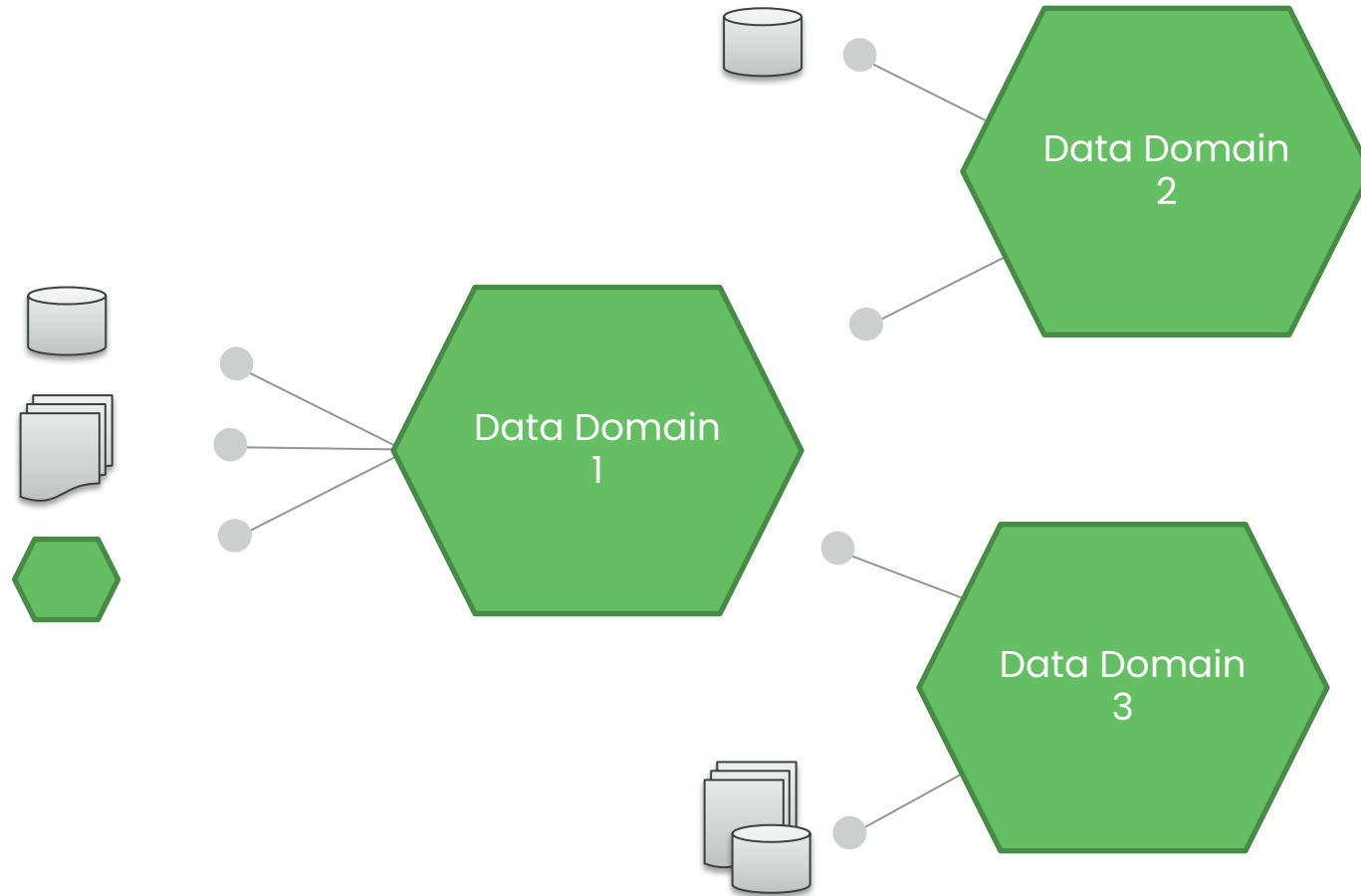
A series of independent domains of data that are:

1. Trusted
2. Usable by the teams' customer
3. Discoverable / Findable
4. Understandable & well-described
5. Secure & permissioned
6. URL/API Driven: & can inter-operate with other domains
7. Have 'single throat to choke' for the customer to easily:
 - Report problem & get updates on fixes
 - Ask for new insights / improvements & get them into production quickly

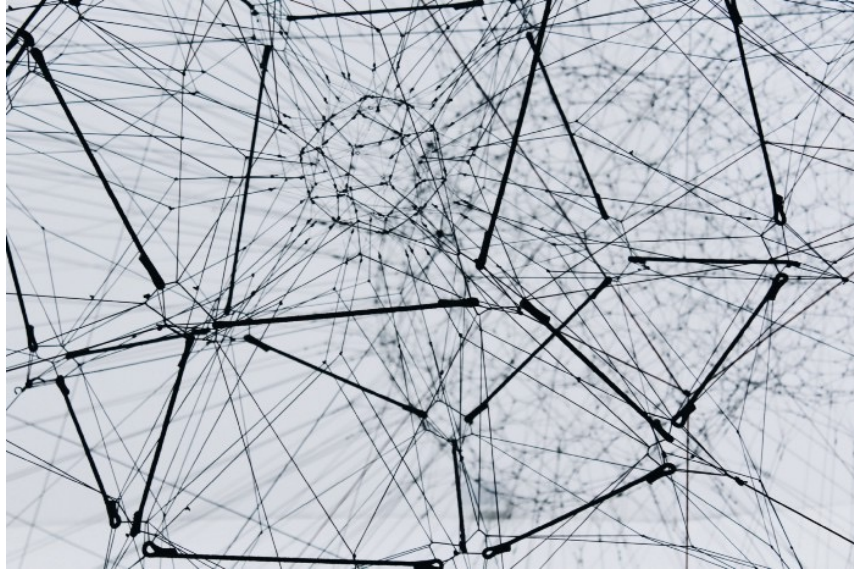
Domain Interfaces



Domains Must Be Composable & Observable



Data Mesh Team Change in Focus



1. **Domains** & the grouping of your work into small teams & partitions over 'one platform to rule them all'
2. What **services** that you are providing your customer, rather than what data you are loading
3. **Discovering** & using over extracting & loading
4. **Decentralization** & the freedom to innovate over central control
5. **Ecosystem** linked together

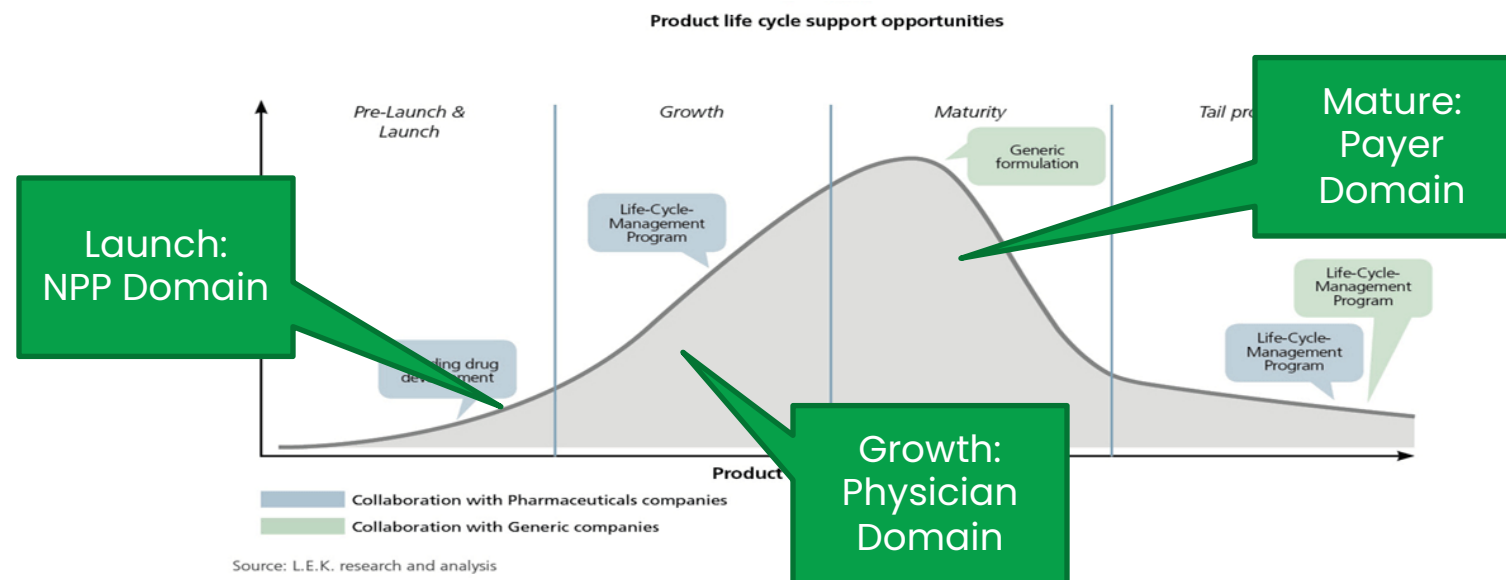
Data Mesh & Products



- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

An Example of Domains

Commercial Pharma Analytics



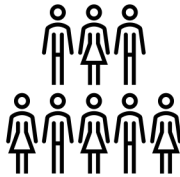
US Commercial Pharma Domains

- **NPP** (Non-Personal Promotion): emails, web site visits, even radio ads
- **Physician**: doctor (& other outlets) sales, claims data, anonymized patient data
- **Payer**: Payer/Plan, rebates, formulary

Pharma Sales & Marketing Teams



NPP Domain Marketing & Sales Team



Physician Domain Marketing & Sales Team



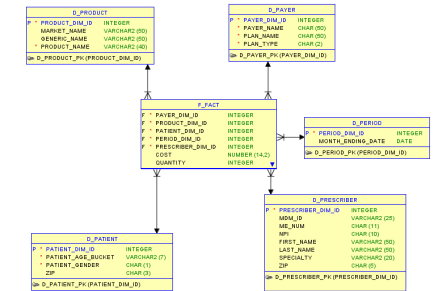
Payer Domain Marketing & Sales Team

What About the Data?

Sub-national Weekly data
Sub-national Payer Data
Sub-national Institutional (DDD) Data
National Prescription Audit Data
Sales Force Alignment Data
Longitudinal Patient Data
Sub-national Profit and Loss Data
Sub-national Claims and Co-pay Data
Payer and Plan Formulary Data
Census Data
Stocking Data
Source of Business
AMA Data
Retail OTC Data
Buy and Bill Data
Field Calls and Promotional Activity Data
Rep Expenses and Vacancy Data
Hotline Verification Data
Contract and Payer Rebates Data
Veeva CRM Data
ERP Data
NPP Data
Forecast Data
Primary Research Data

What about the data in each domain?

- Each domain has separate data sources.
- Overlapping entities (e.g., physicians) exist in each domain
- Each domain has different product cycle times (i.e., daily, weekly, hourly, etc.)
- Each data domain has its unique characteristics.



Domain Layers

Mastered Data Sets
(IT)

Integrated Data Sets
(Data Engineers)

Self Service Tools
(Analyst)

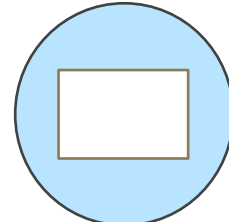
1. Mastering & small files foundation files are a domain layer
2. Of course, the main data warehouse is a domain layer
3. Self/Service & Data Science are a domain layers

Domain Layers

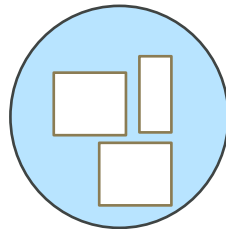
Raw, Sourced Data (Various)

Sub-national Weekly data
Sub-national Payer Data
Sub-national Institutional (DDD) Data
National Prescription Audit Data
Sales Force Alignment Data
Longitudinal Patient Data
Sub-national Profit and Loss Data
Sub-national Claims and Co-pay Data
Payer and Plan Formulary Data
Census Data
Stocking Data
Source of Business
AMA Data
Retail OTC Data
Buy and Bill Data
Field Calls and Promotional Activity Data
Rep Expenses and Vacancy Data
Hotline Verification Data
Contract and Payer Rebates Data
Veeva CRM Data
ERP Data
NPP Data
Forecast Data
Primary Research Data

Mastered Data Sets (IT)

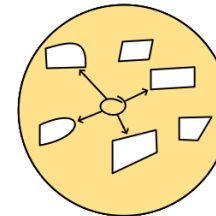


Mastering Domain: Physician MDM

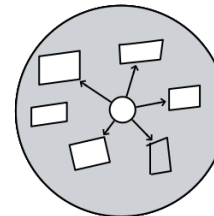


Mastering Domain: Target Lists, Product Market Baskets

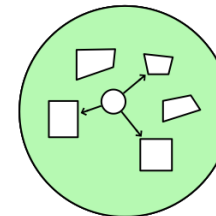
Integrated Data Sets (Data Engineers)



Payer Domain

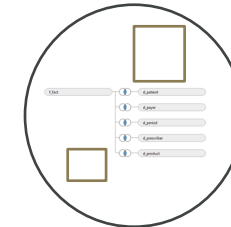


Physician Domain

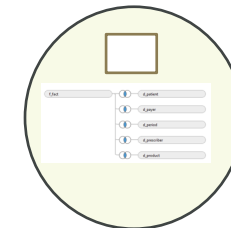


NPP Domain

Self Service Tools (Analyst)



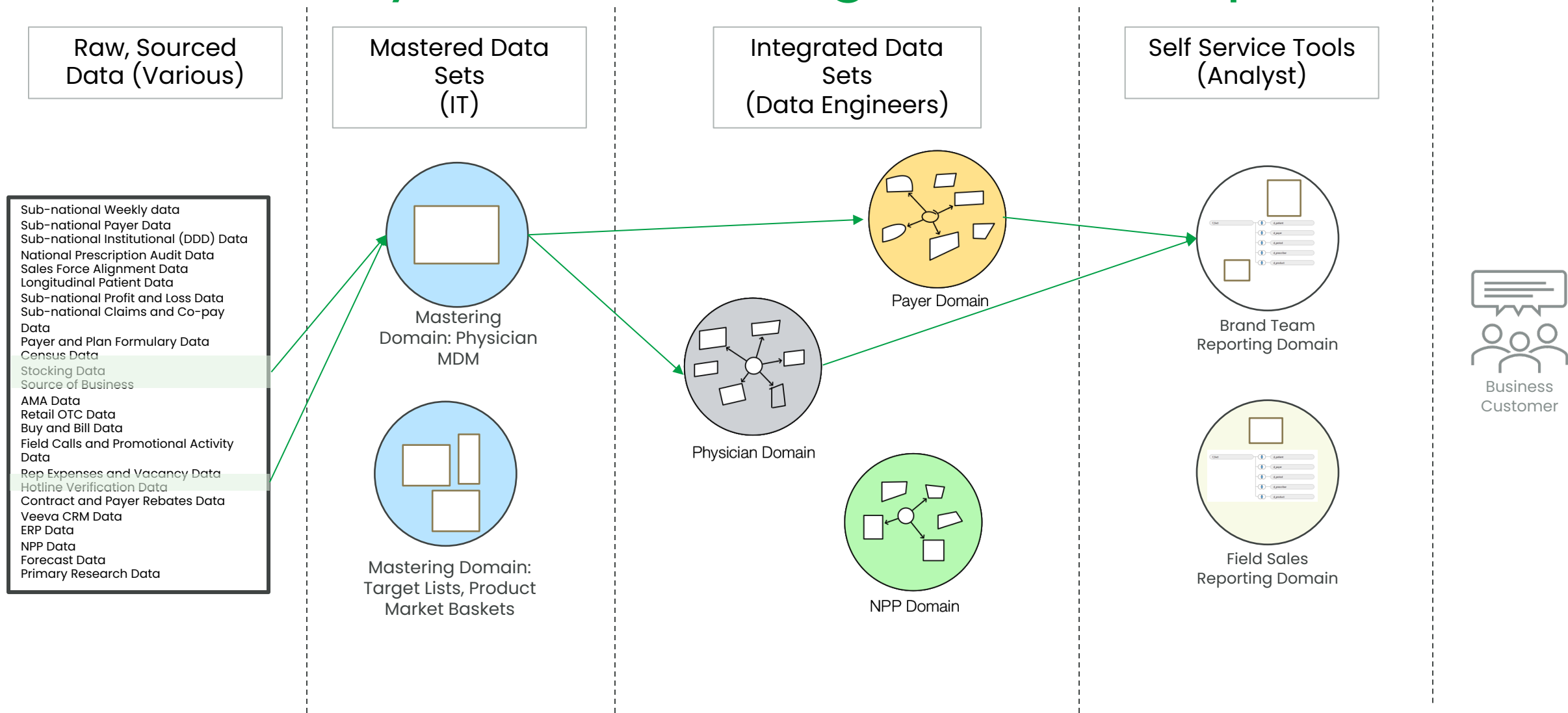
Brand Team Reporting Domain



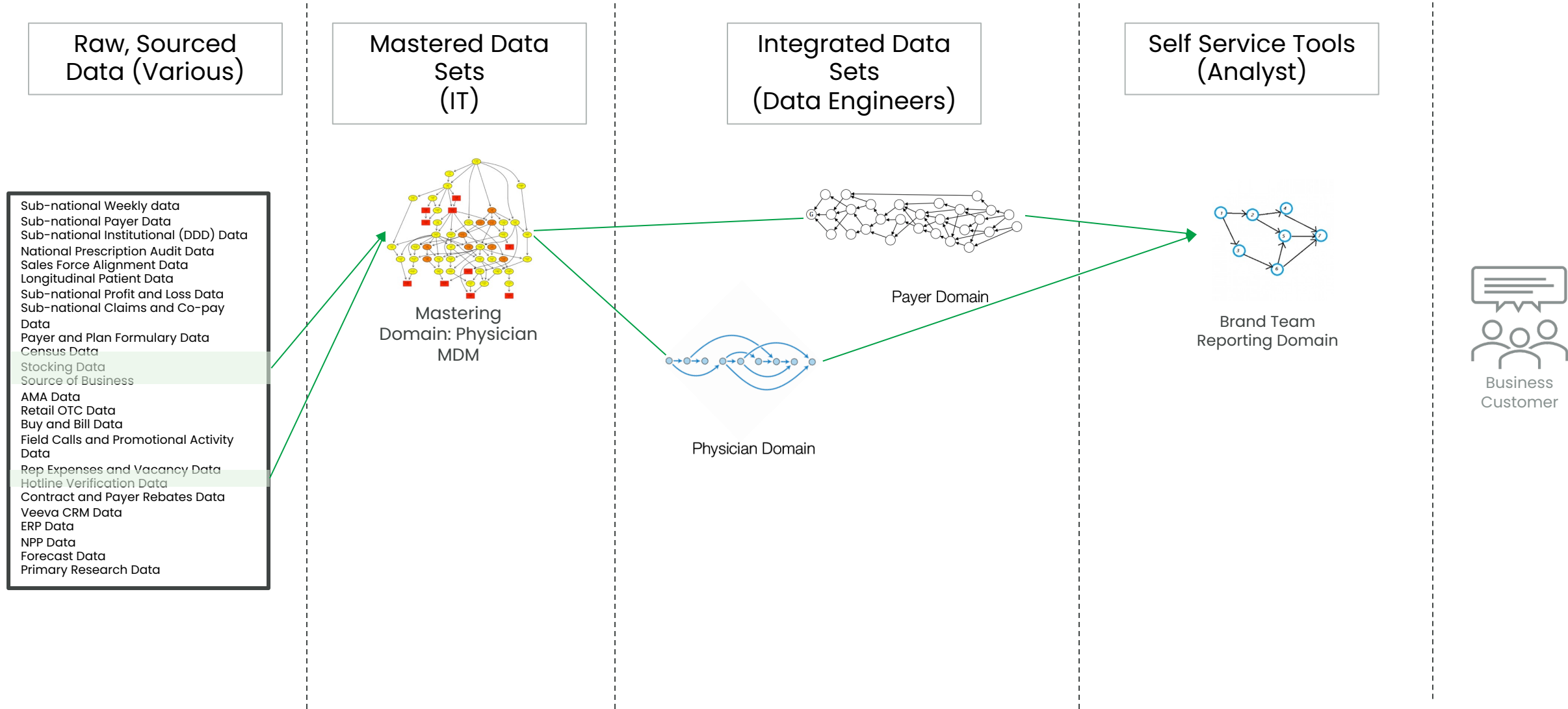
Field Sales Reporting Domain



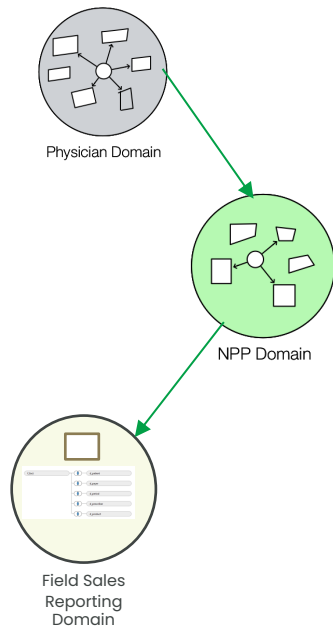
Domain Layers Processing Relationships



Domain Layers Processing Steps



Inter-Domain Communication Links



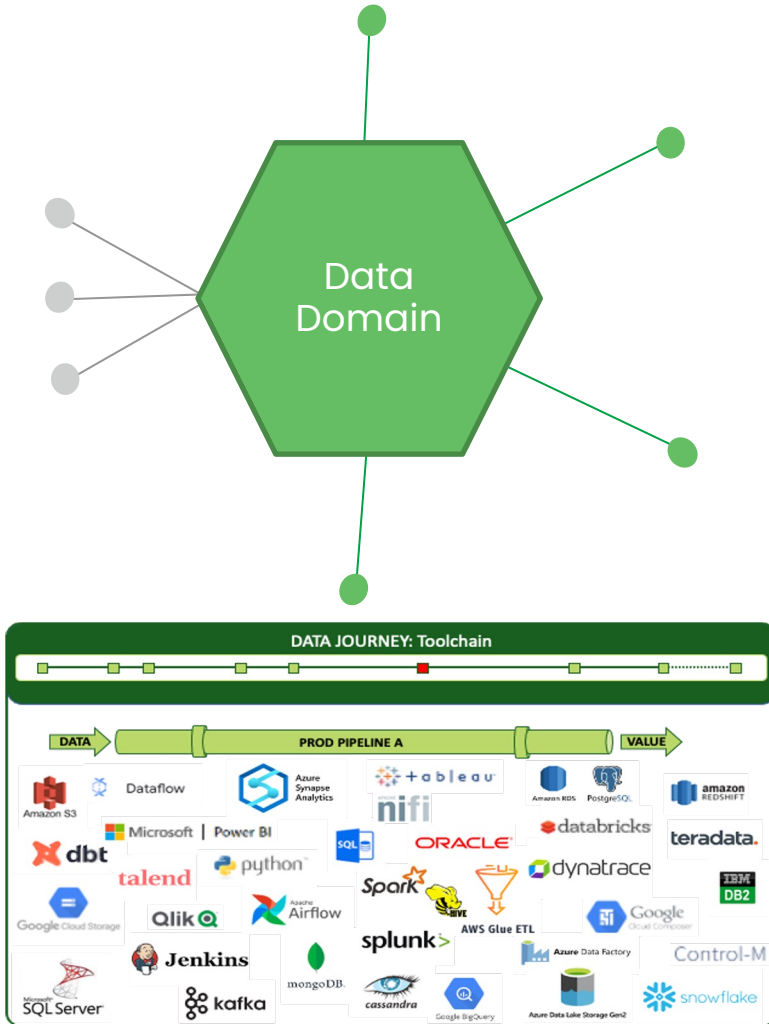
Inter-Domain Communication		Question / Steps Asked
	Domain Query	"When was the last time you were updated?" Successful or failure? Warnings?
	Domain Query	"Is the data or artifacts in your domain good?" Can you prove it with some test results?"
	Process Linkage	"Ok, you start. I am done."
	Process Linkage	"Ok, you start. I am done & here are a bunch of parameters you need to keep going."
	Event Linkage	"Here is an event: e.g., processing completed, error, warnings, etc."
	Data Linkage	"We share a common table (e.g., a dimension table) in our domain."
	Development Linkage	"Can I re-create your domain in development?" "Can I see the code you used to create it?" "Can I modify that code in development?" "Is there a path to production?"

Data Mesh & Products



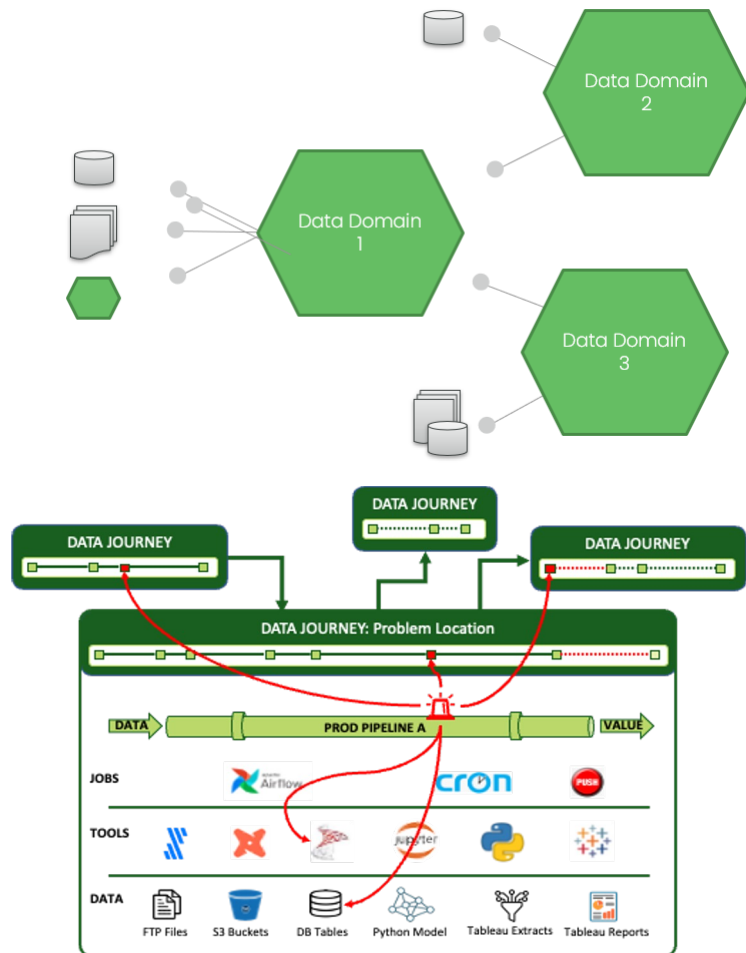
- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

Each Domain is a Data Journey



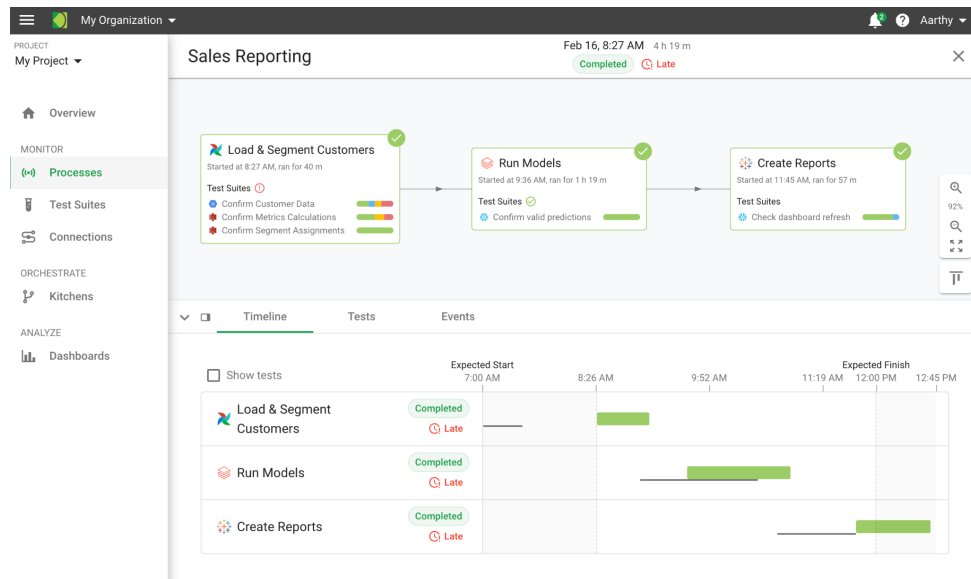
- Data Journeys track and monitor all levels of the data stack in a domain, from data to tools to code to tests across all critical dimensions.
- It supplies real-time statuses and alerts on start times, processing durations, test results, and infrastructure events, among other metrics.
- With this information, you can know if everything ran on time and without errors and immediately identify the parts that didn't.
- Journeys provide a context for the many complex elements of a domain.

Composition of Domains Need Data Journey observability



- Each domain can change independently, with a centralized data journey to ensure the entire system is correct.
- You must monitor, test, alert, and analyze your multi-domain data journeys in real time.
- Why? The first step to taming complexity is to monitor every tool, team, environment, and customer so that problems are detected, localized, and understood immediately.

DataKitchen Does Data Mesh Observability



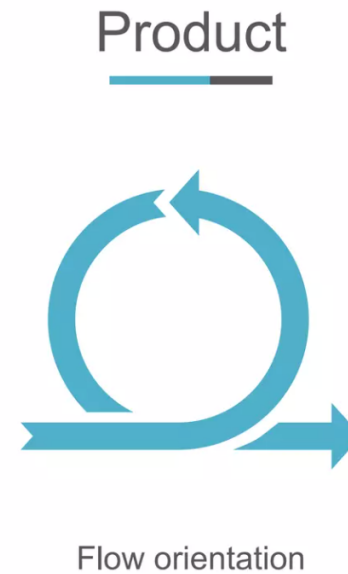
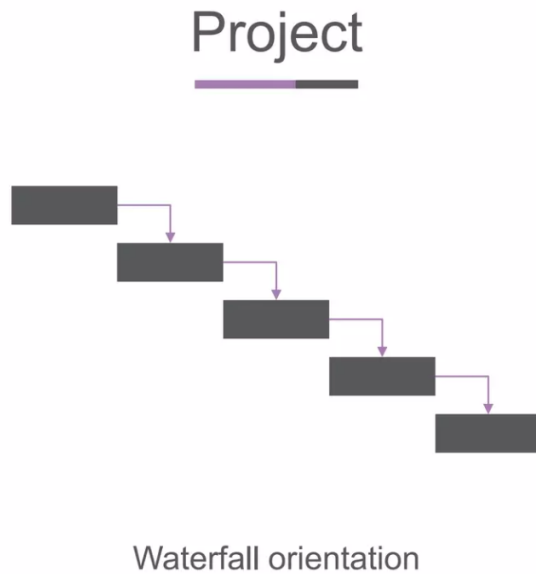
- Deliver DataOps value immediately
- Enterprise-wide views of 100's 1000's **data journeys**
- Monitoring, alerting – of the tools **and data**
- Across any existing toolchain
- In every Data Mesh domain

Data Mesh & Products



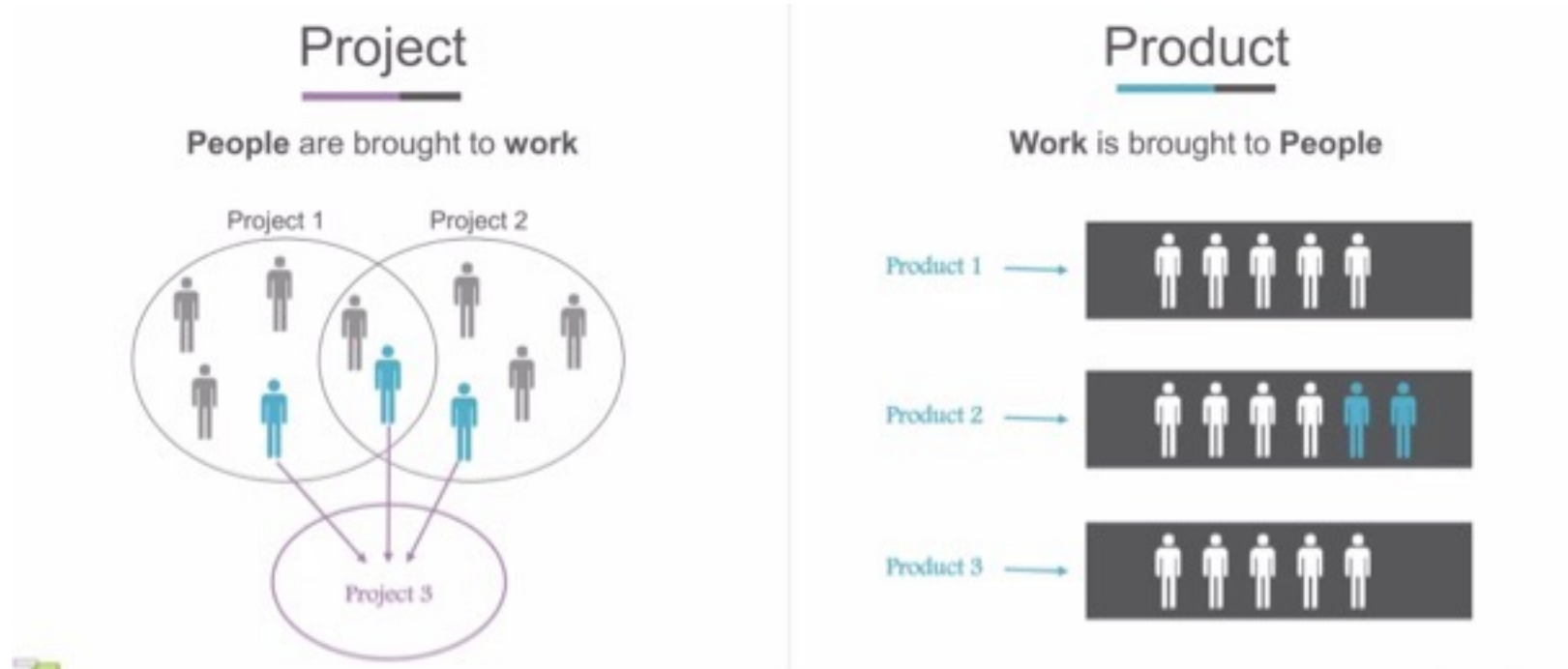
- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

Data Projects to Data Products



From Tasktop

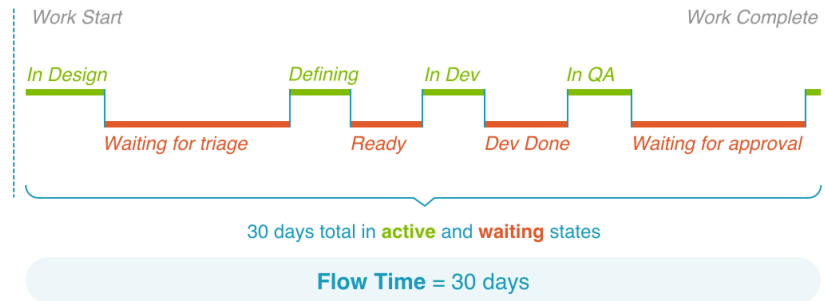
Data Projects to Data Products



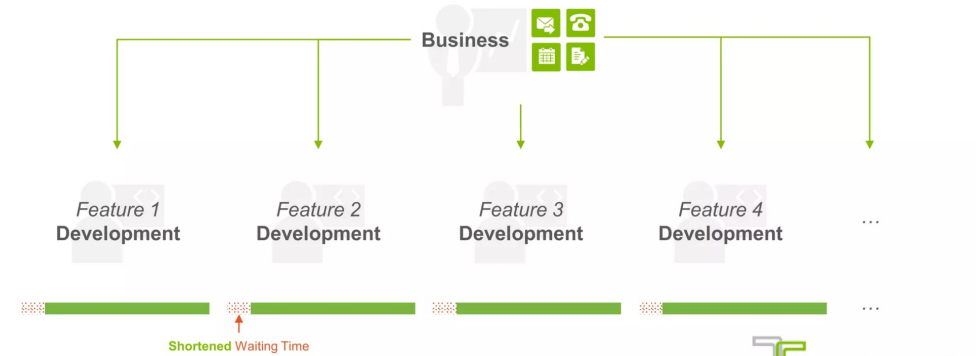
From Tasktop

Data Projects to Data Products

Project



Product



From Tasktop

Data Projects to Data Products

Attribute	Project	Product / Program
Funding	Annual	Incremental – frequent checkpoints to reallocate funding
Timeline	Defined start and end dates; managed via a detailed project schedule	Ongoing - lifecycle from idea to end of life; managed via a roadmap
People	Role-based, assigned to complete a set of tasks for the project	Cross-functional – Core team persists for the product lifecycle
Metrics	On-time / on-budget delivery; benefits realization	Efficiency; value-based – customer satisfaction, profit, etc.

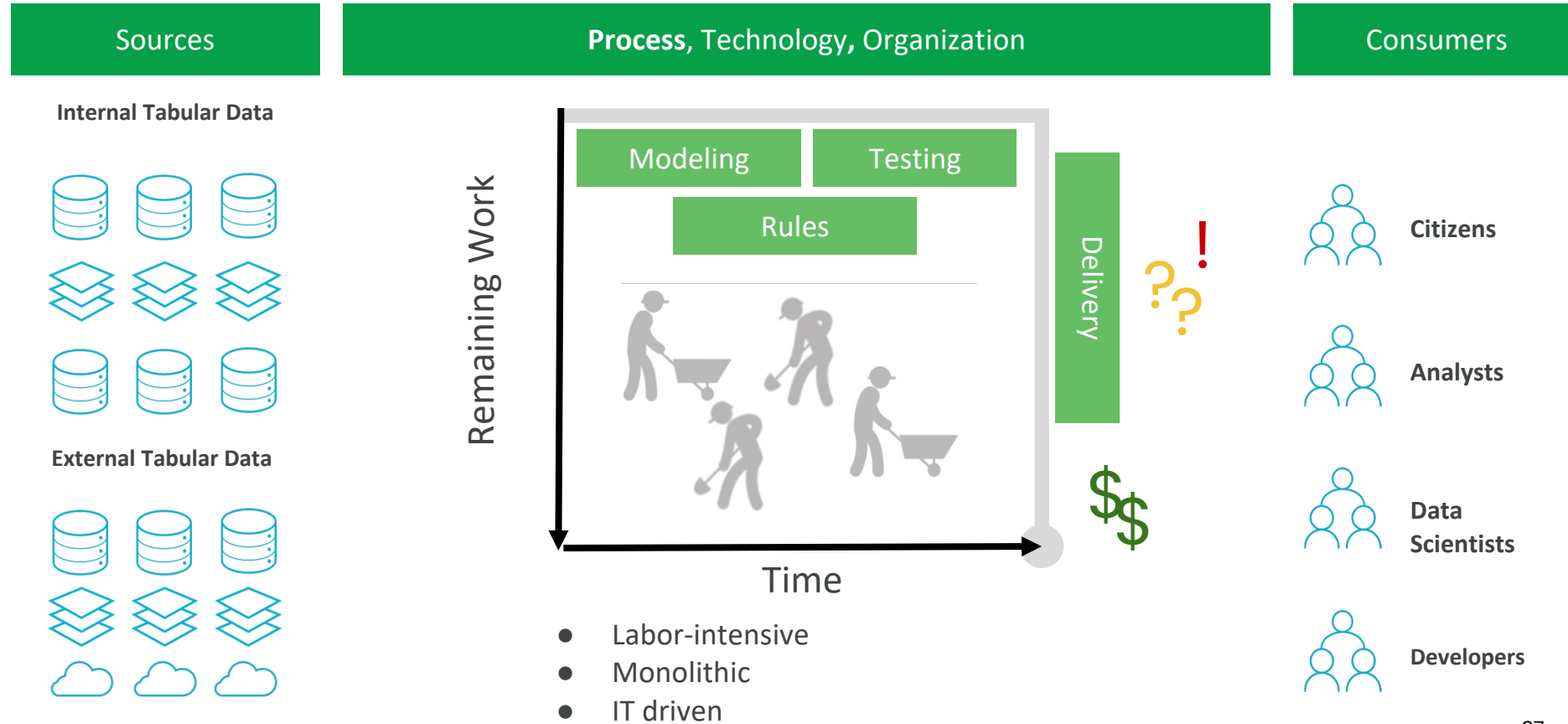
Attributes of Project vs Product/ Program Management

Data Mesh & Products



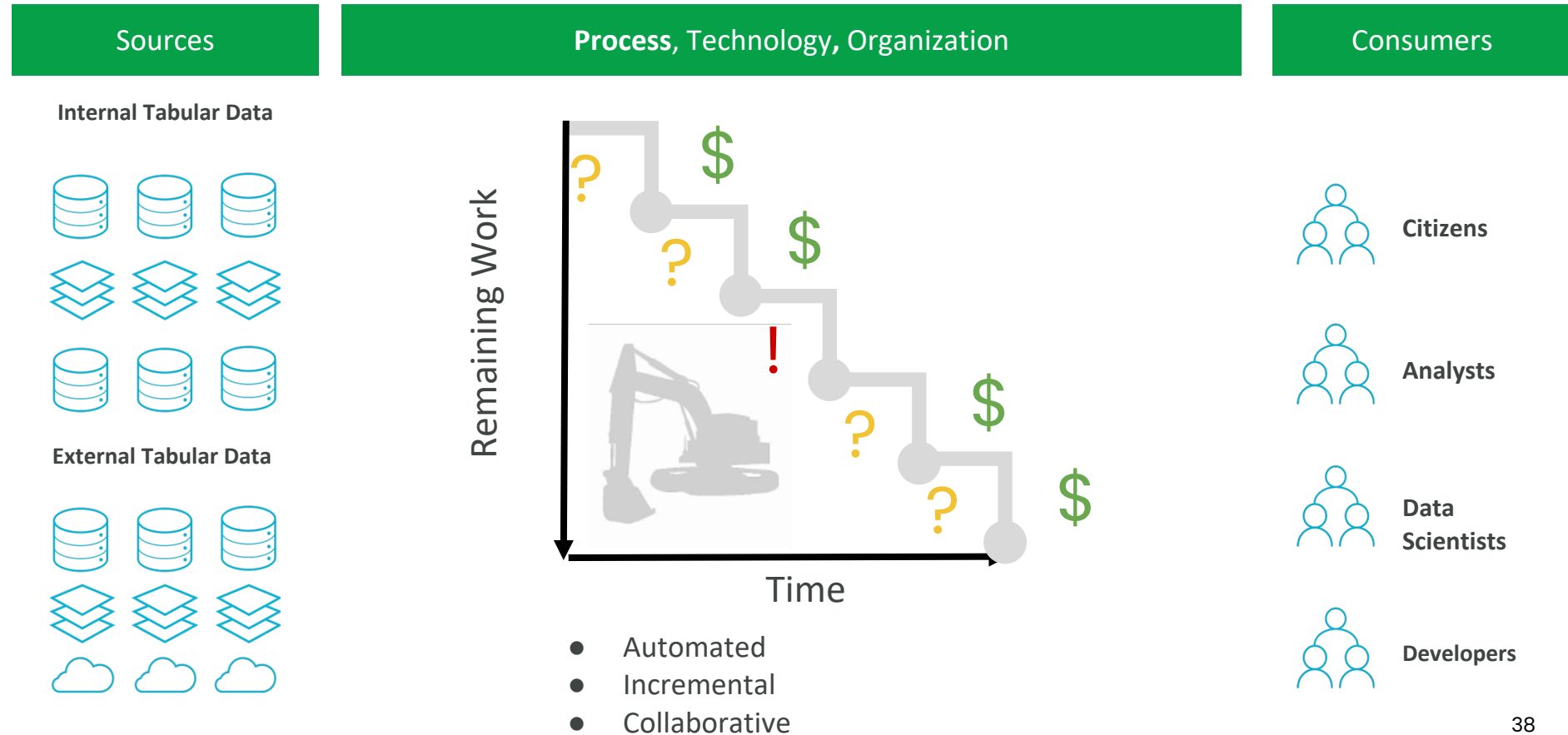
- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

Data Products – The Wrong Way



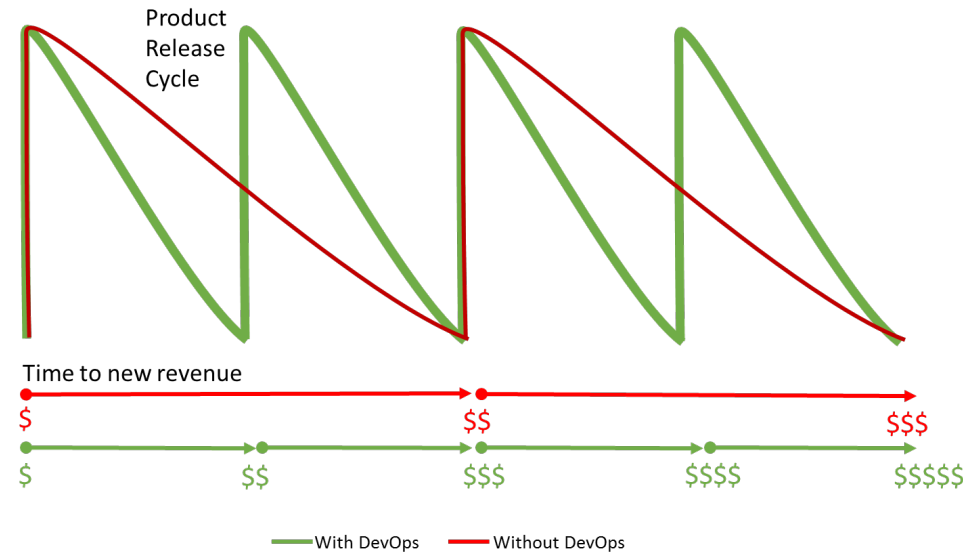
37

Data Products Are Continuously Improved



38

Data Products Are Never Done



- Faster releases of analytics products
- Increased learning of your team
- Better insight
- Faster and more value for your data analytics

Data Product Maximize Flow



- Improving and measuring the ‘flow’ of value to your customer is the #1 goal of Data Products
- What is the ‘flow’ of value to your customer?
 - New data sets
 - Improved models/visualizations/data
 - New code/configuration
- Key measurements
 - Cycle time of deployment (‘flow time’)
 - Success Rates of Deployment (‘flow efficiency’)
 - Work done (‘flow velocity’)
- Flow: The value stream, continuous delivery, and product-centric thinking are emphasized over traditional waterfall project management.

Data Mesh & Products

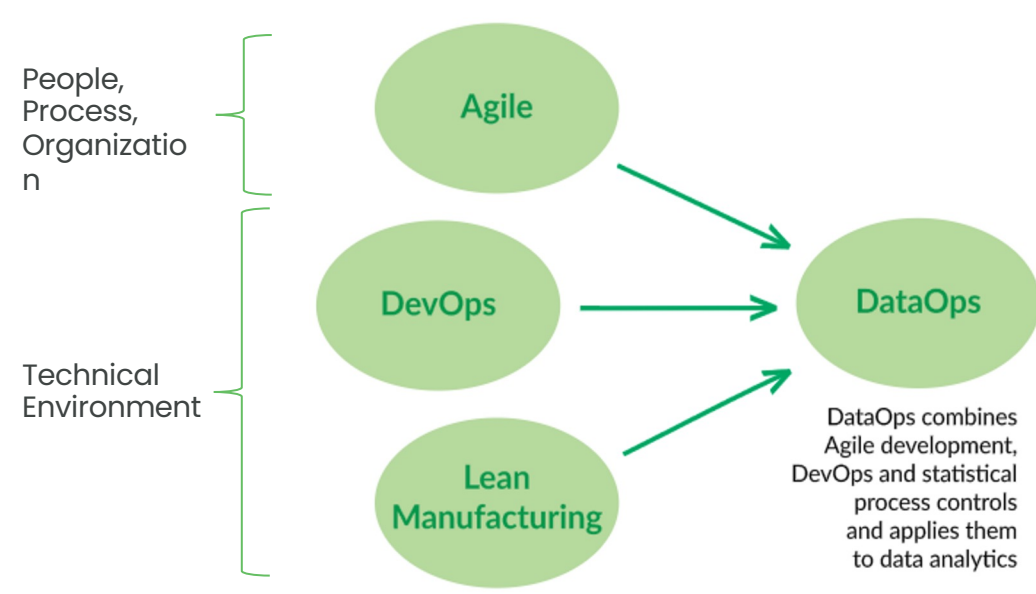


- Data Mesh
 - The People Side: Team Structure
 - The Tech Side: Domains
 - The Inside: Extended Example
 - The Outside: Observability
- Data Products
 - The People Side: Team Goals
 - The Tech Side: Iterations
 - The Inside: DataOps
 - The Outside: Observability
- Concluding Thoughts

DataOps – Enabling Data Mesh & Products

DataOps is a set of technical practices, cultural norms, and architecture that enable:

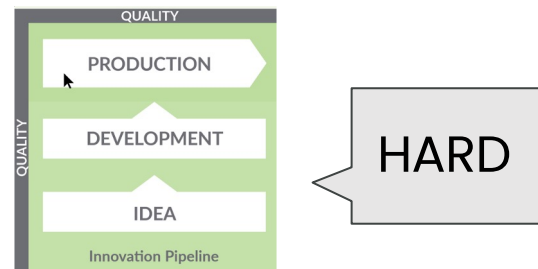
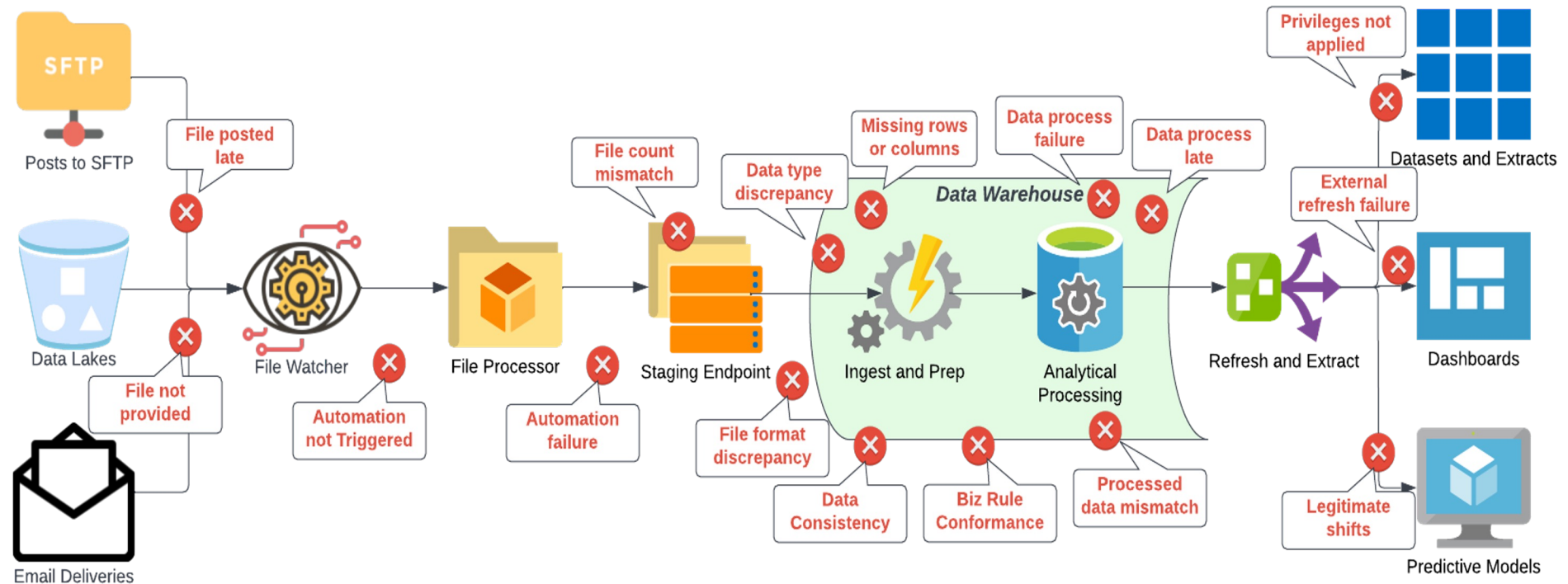
- **Rapid experimentation** and innovation for the fastest delivery of new insights to our customers
- Low production **error rates**
- **Automation** and collaboration across complex sets of people, technology, and environments
- Clear **measurement and observability** of results



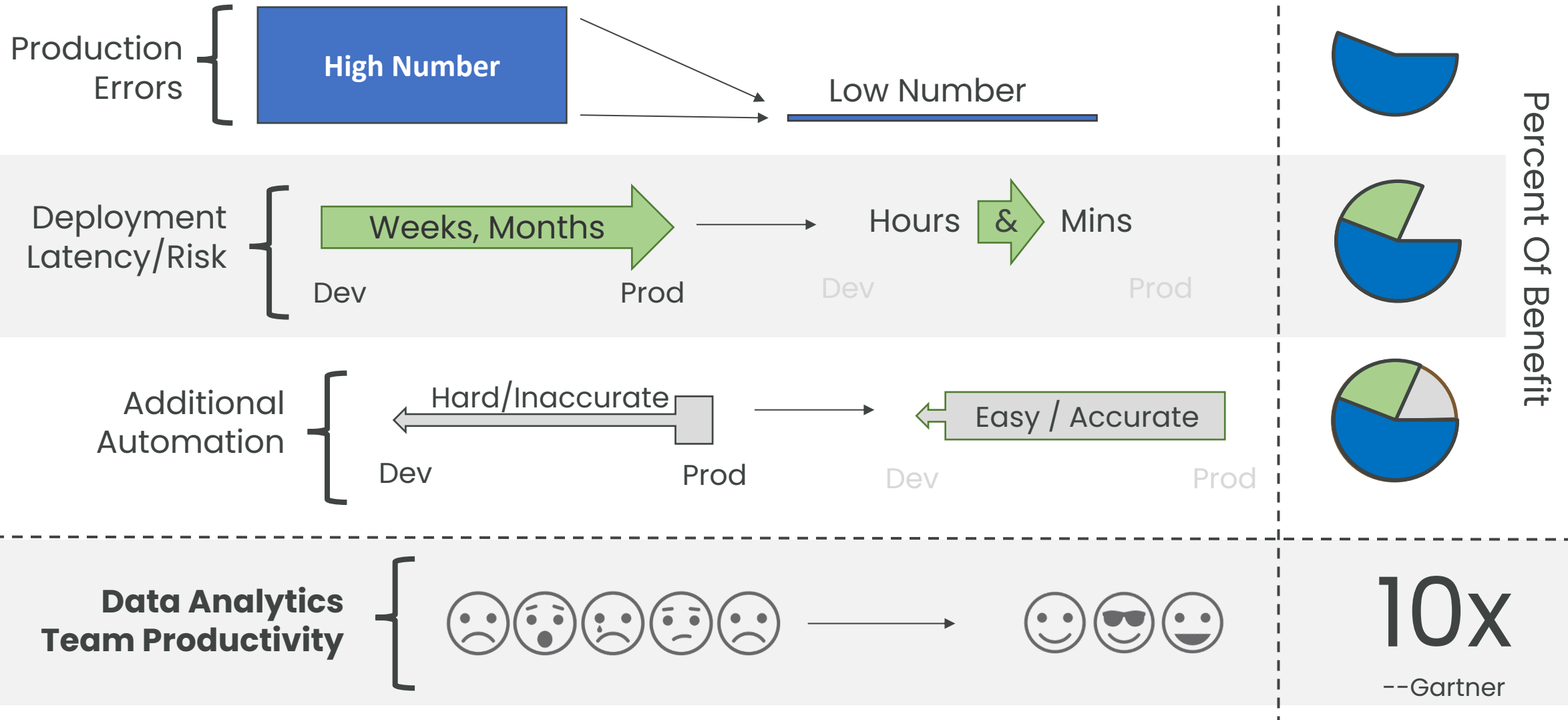
*Gartner Strategic Planning Assumption: “By 2025, data engineering teams guided by DataOps practices and tools will be **ten times more productive** than teams that do not use DataOps.”*

Source: Gartner, 2023

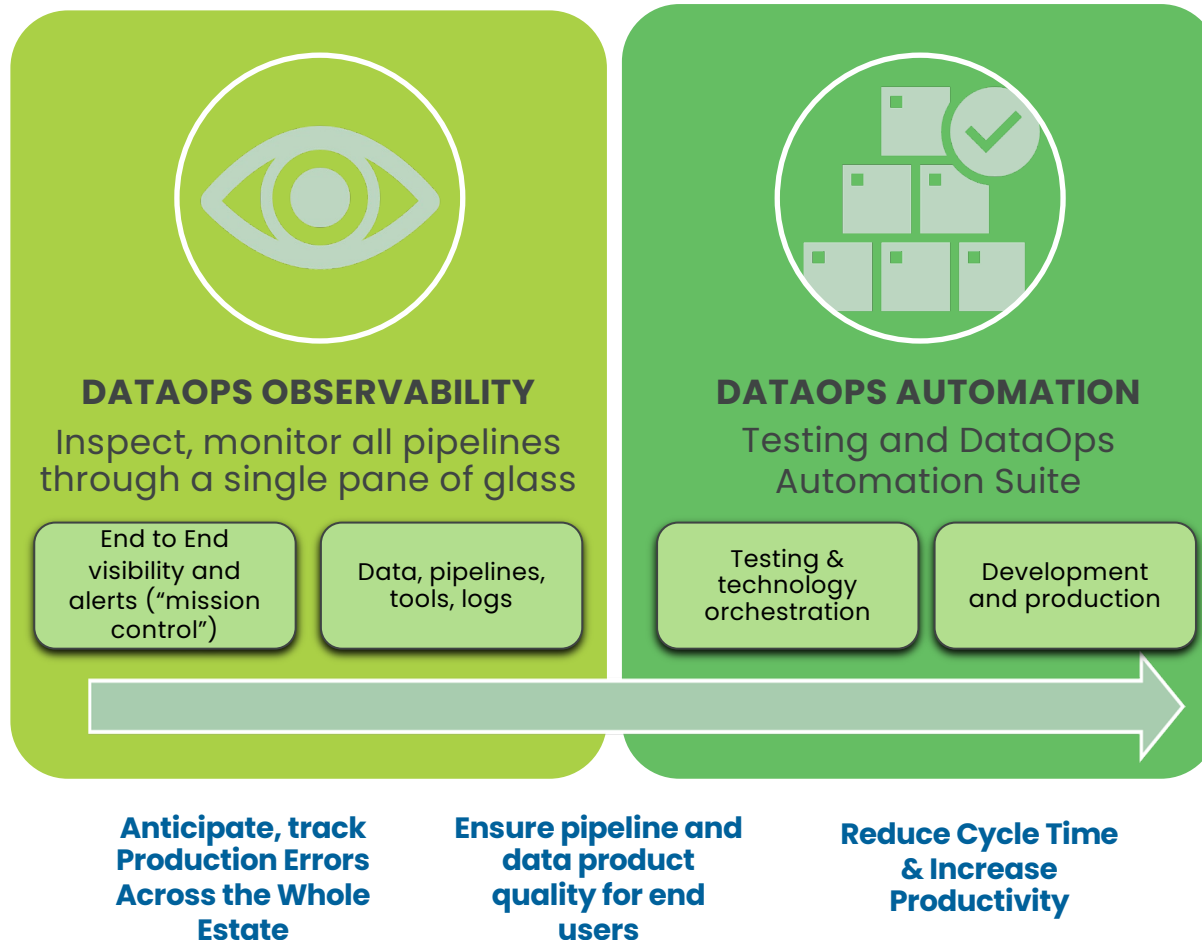
Problems During The Data Journey



DataKitchen: Faster, Better & Happier



DataKitchen Products: DataOps Observability and DataOps Automation



What: Observability-led DataOps software for every data journey from source to customer value, from any development environment into production, across every tool, team, and data consumer.

Benefit: So that data and analytic teams' *problems are detected, understood, and automated away rapidly.*

Conclusion

Data Mesh and Data Products are an exciting new paradigm that aims to:

- Complexity Reduction
- Maximize Flow

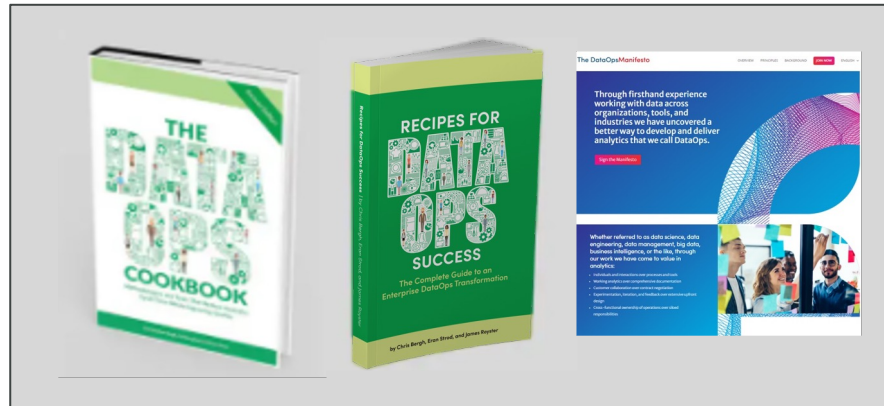
Data Mesh and Data Products are not:

- Data Virtualization, Data Federation
- Data as a product

Remember, the DataOps part of the mesh is equally important!

- Observability
- Automation

Learn More About DataOps and DataKitchen



Sign The DataOps Manifesto:

<http://dataopsmanifesto.org>

Free DataOps Cookbook:

<https://datakitchen.io/the-dataops-cookbook/>

Free DataOps Transformation Book

<https://datakitchen.io/recipes-for-dataops-success-guide-to-dataops-transformation/>

DataOps Observability and Automation
Software

<https://datakitchen.io/platform/>