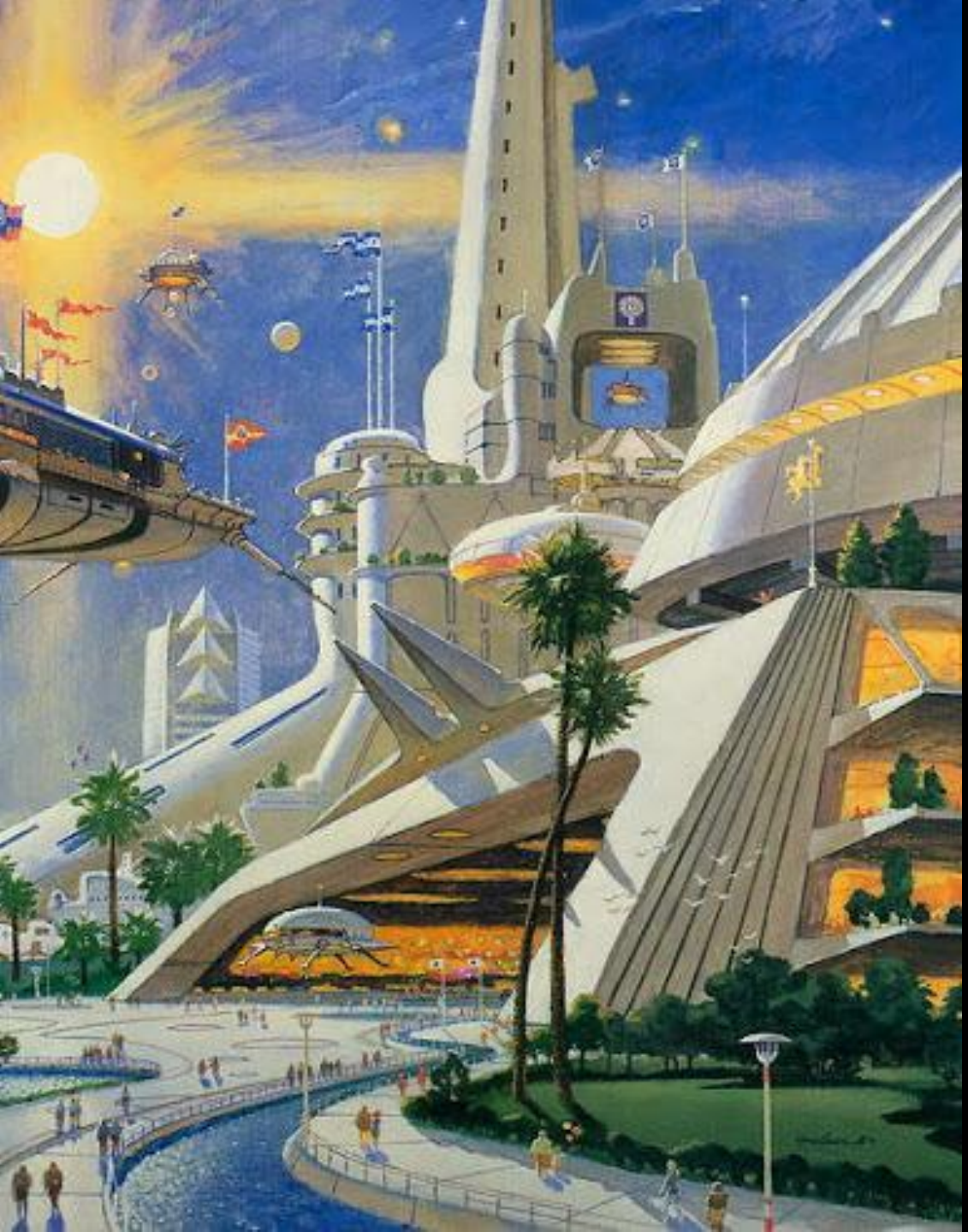


Architecting a Data Platform For Enterprise Use

September 2018

Mark Madsen
Todd Walter

Download the PDF at:

HTTPS://WWW.SLIDESHARE.NET/MRM0/ARCHITECTING-A-DATA-PLATFORM-FOR-ENTERPRISE-USE-STRATA-NY-2018

What do we hear?

"I want to do analytics, beyond what I can do with BI tools"

"I need an analytics strategy"

"I need an analytics roadmap"

"I want to modernize my DW" aka "I want to speed up my DW"

"I have a specific analytics project of type..."

“We have a data lake. What should we do with it?”

“Technology Y is our replacement for technology X”

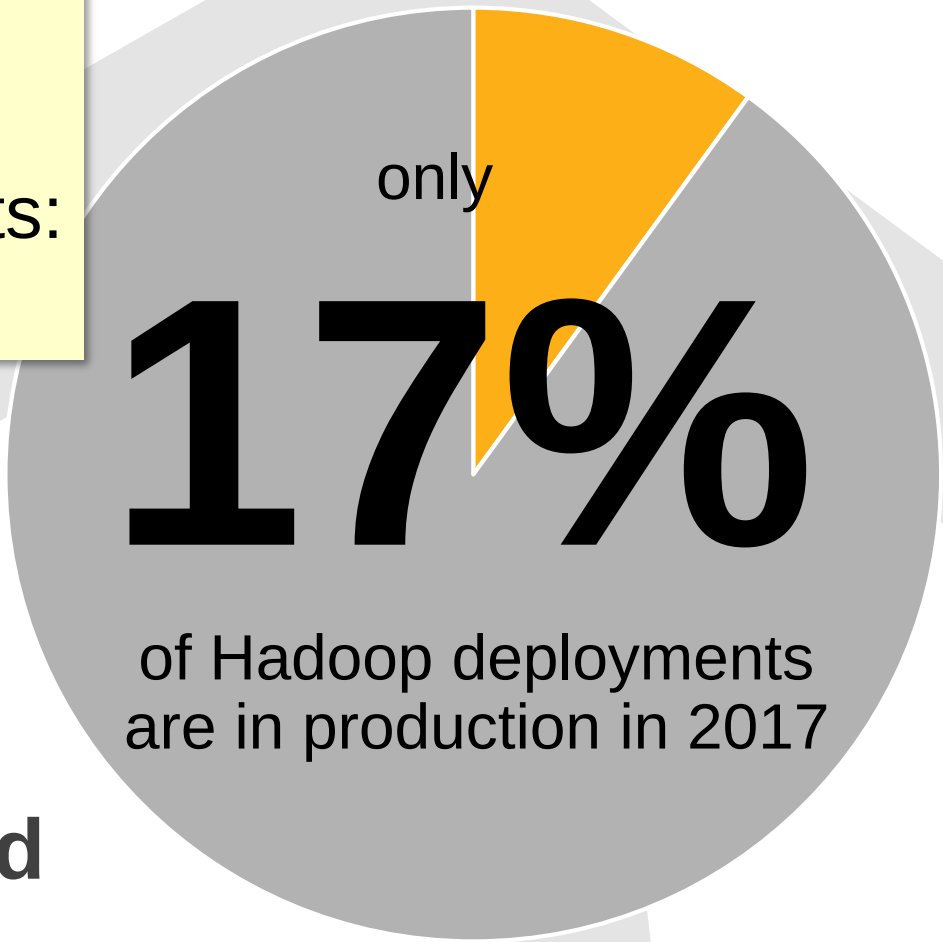
“Don’t need schema, curation, ETL, governance … anymore”

“How do I ensure the data is good enough for analytics?”

Good judgement is the result of experience.
Experience is the result of bad judgement.
—Fred Brooks

A McKinsey survey this year asked executives if their company had achieved a positive ROI with their big data projects:
7% answered "yes"

Gartner Finding:
in 2017

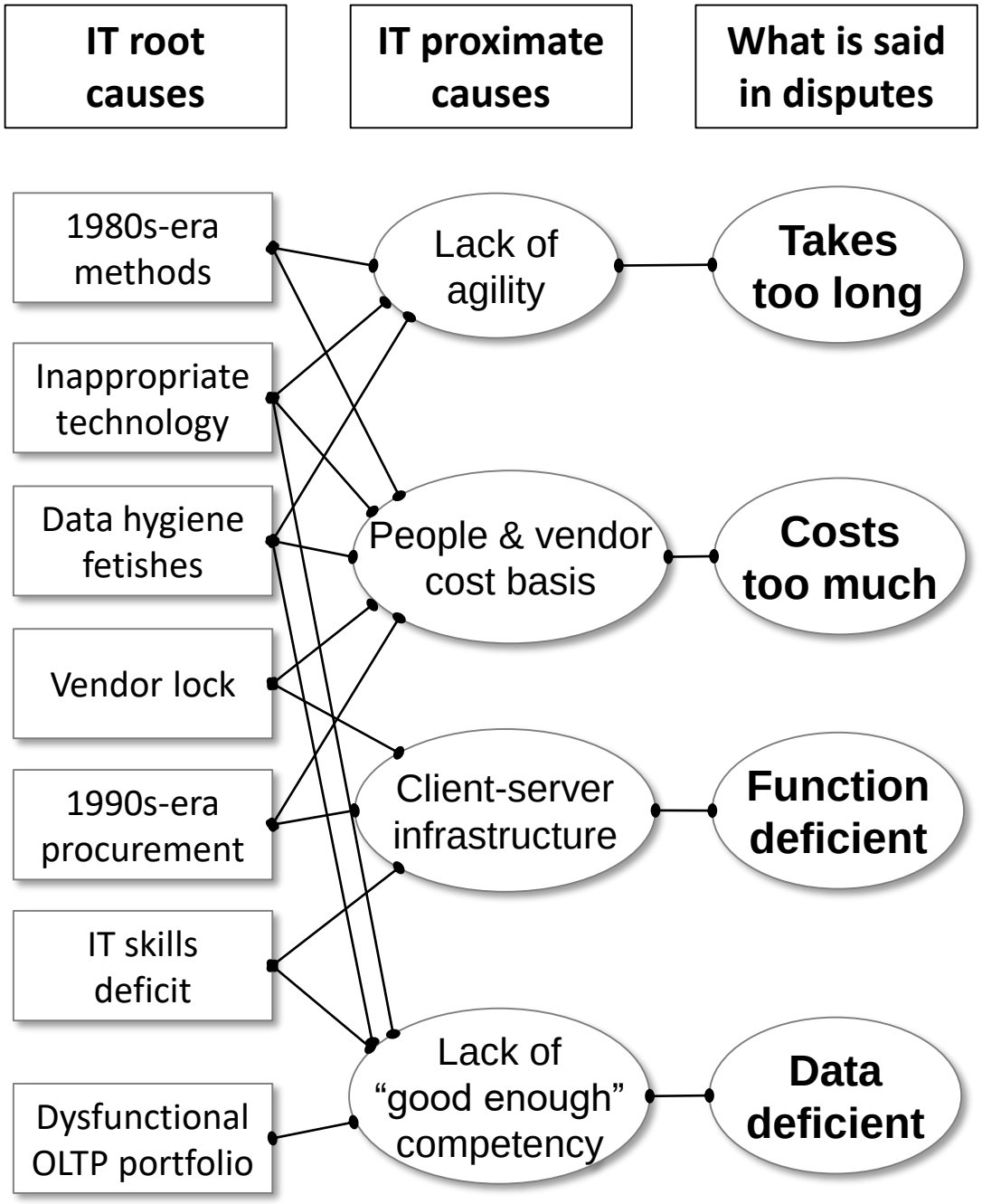


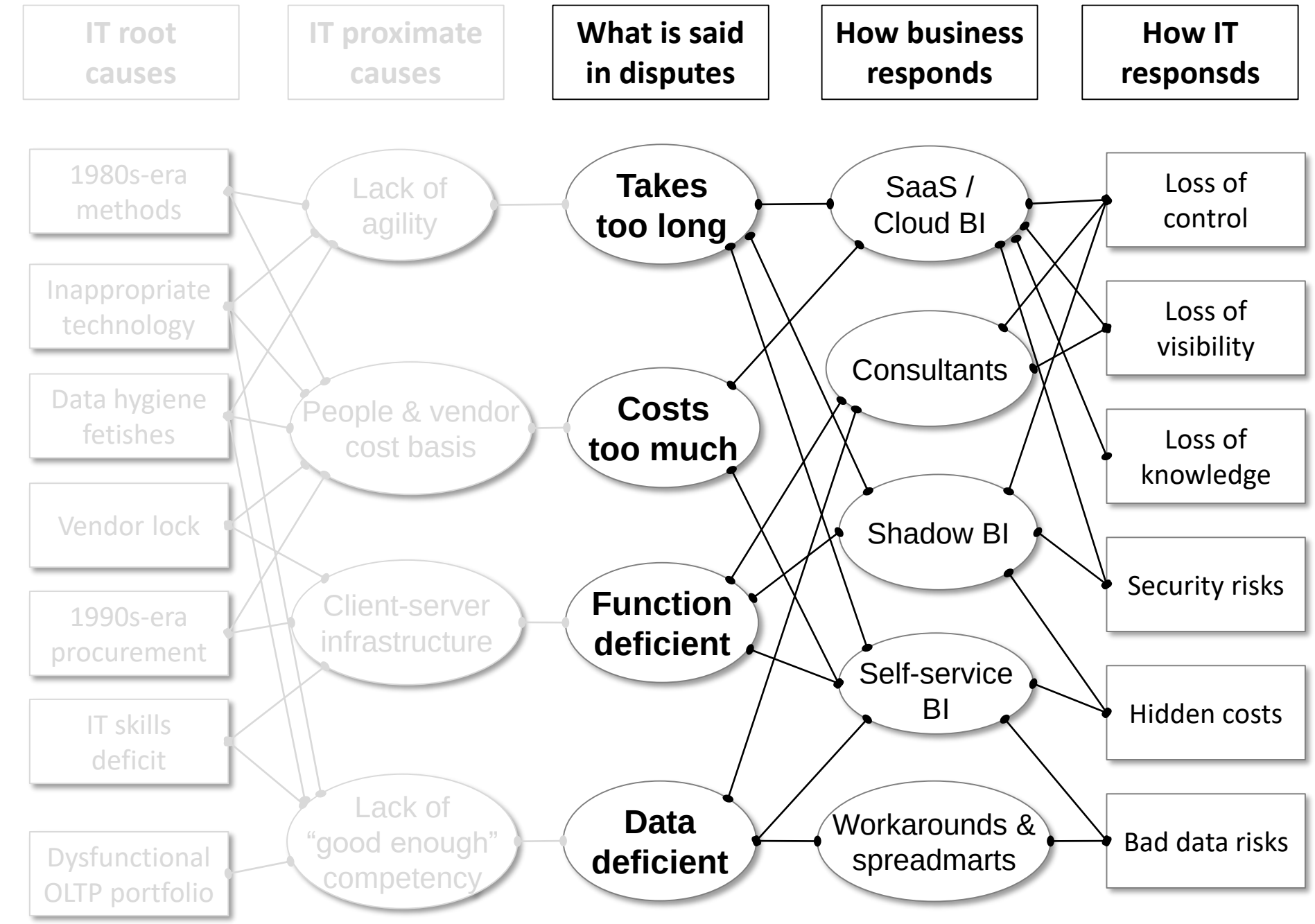
Gartner statement in 2018:
only 15% are reported to be successful

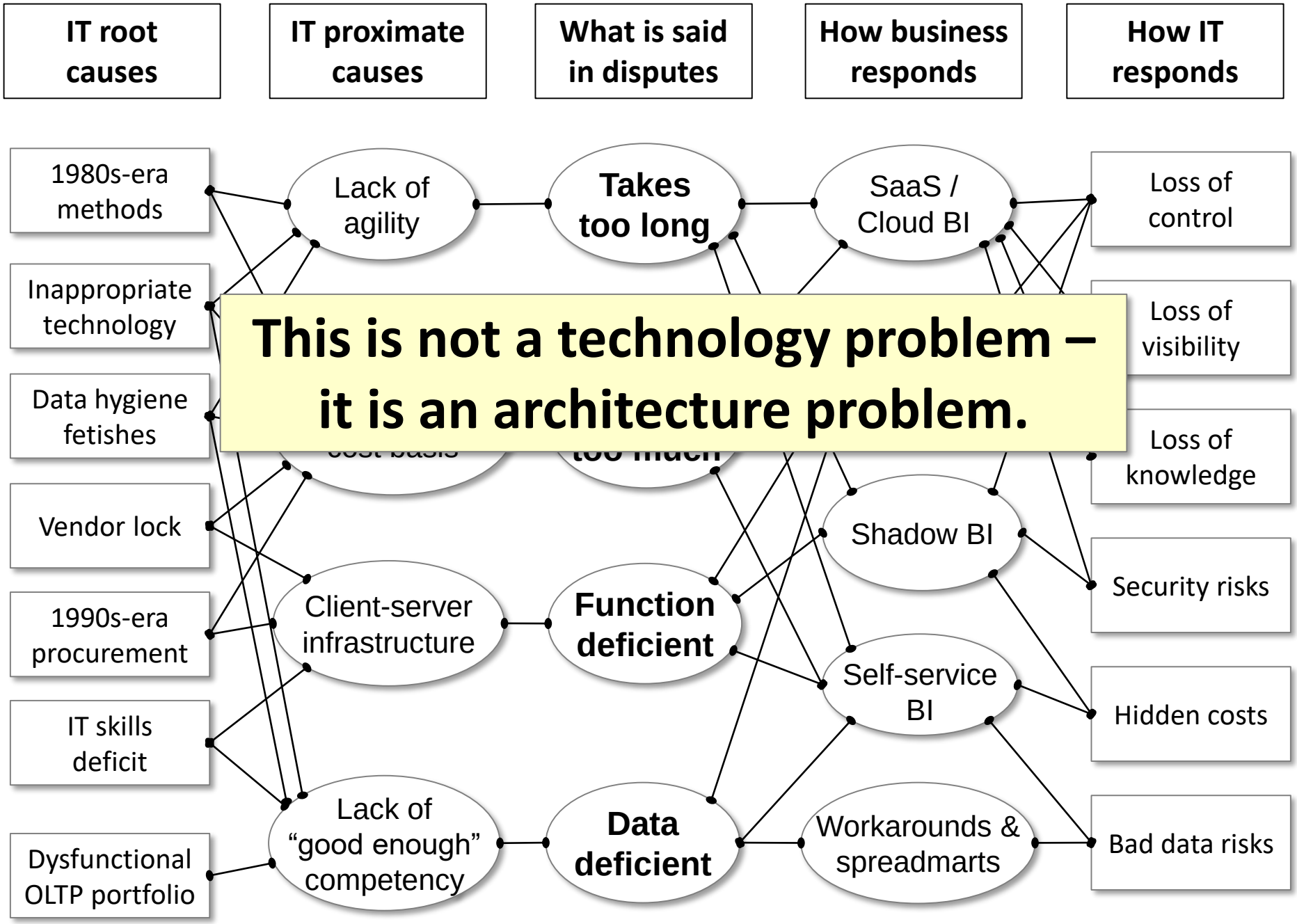
Survey Analysis: BI and Analytics Spending Intentions, 2017

Gartner®

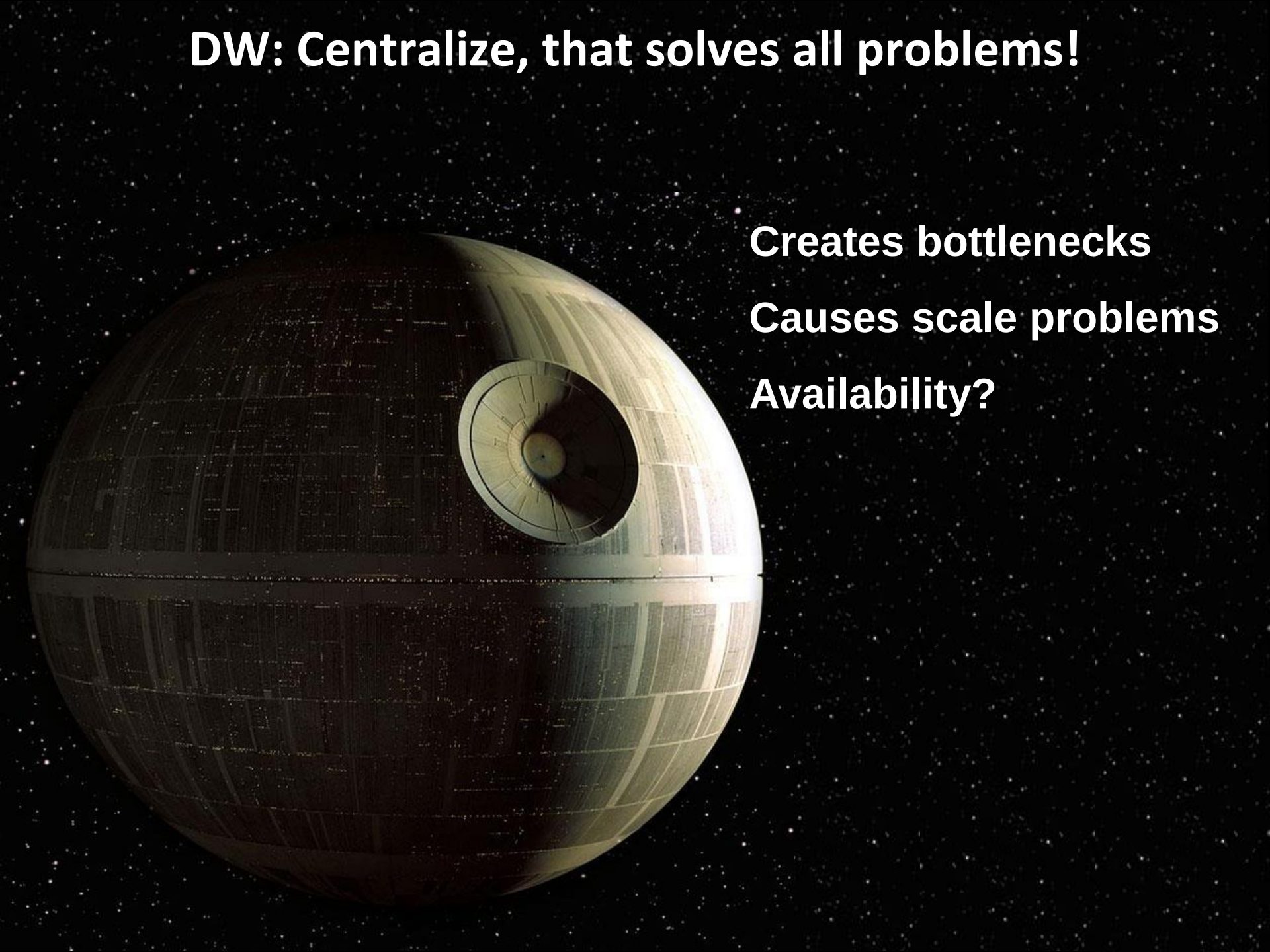
The business complaints about data and analytics







DW: Centralize, that solves all problems!

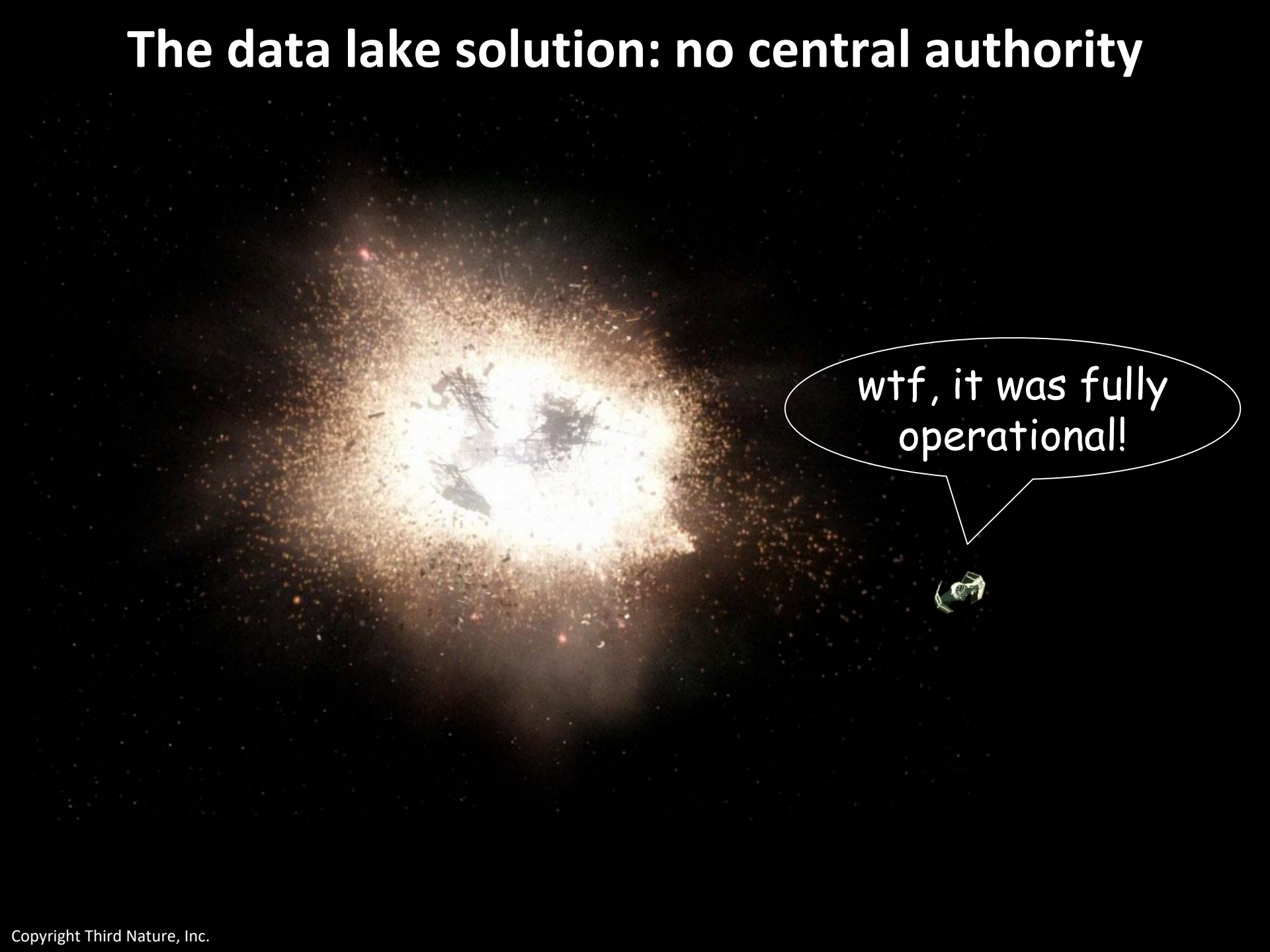


Creates bottlenecks

Causes scale problems

Availability?

The data lake solution: no central authority

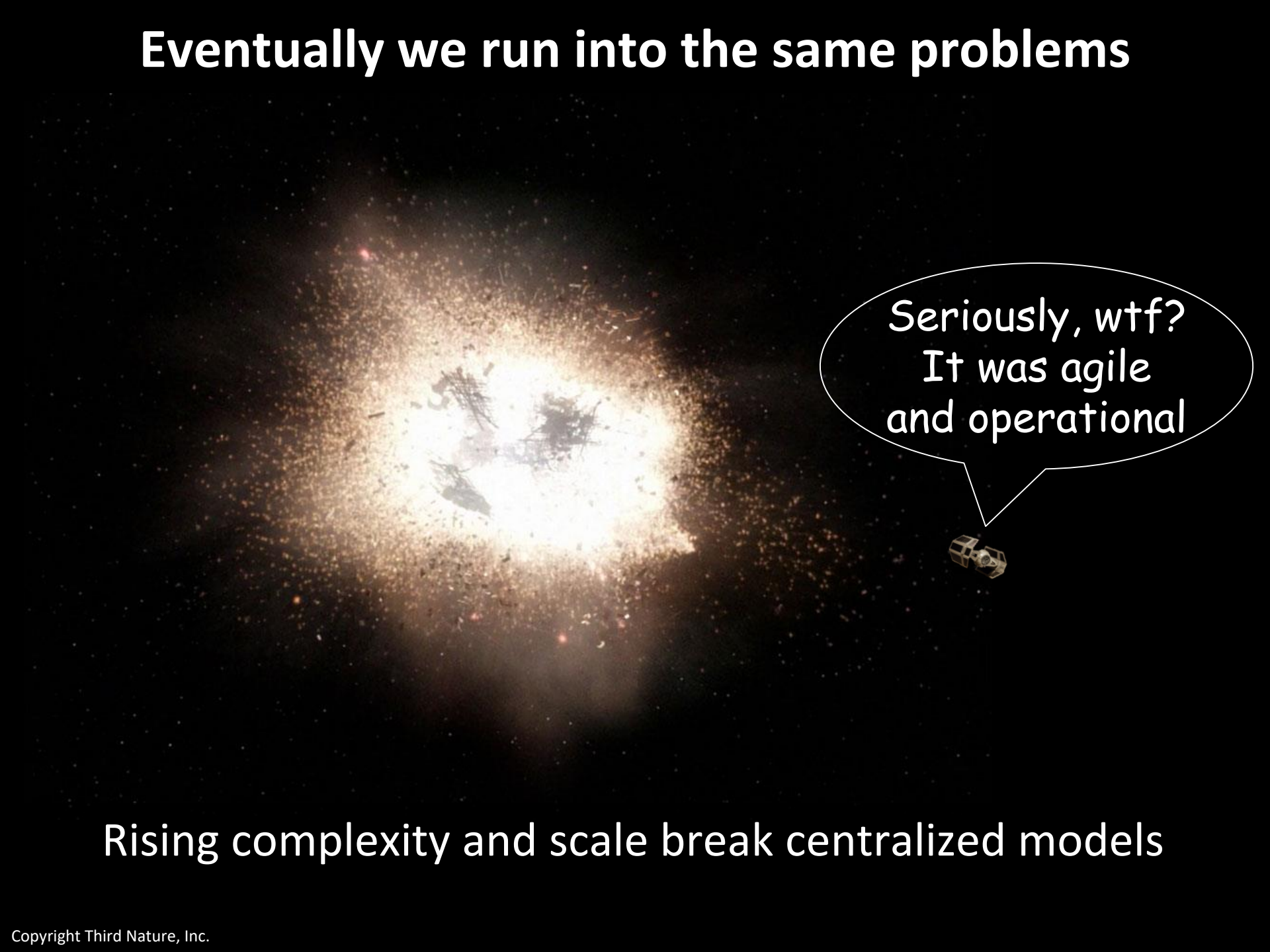


The data lake solution?

There's a problem: as the lake is envisioned, it is still a centralized data architecture, but this time there is no single global model. Instead it's files and not modeled. It can be operational while under construction.

It's still a death star.

Eventually we run into the same problems



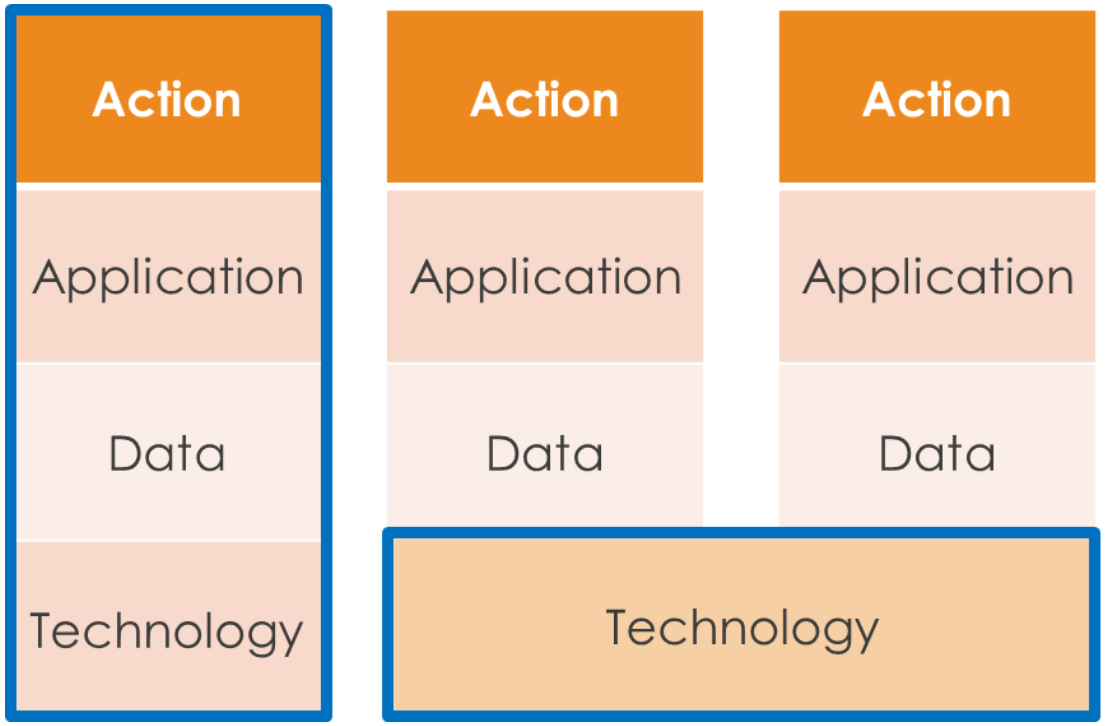
Rising complexity and scale break centralized models

“The problem is the tools. Standardize on one tech!”

It’s called stack think. Pick your vendor. Cede all architecture.

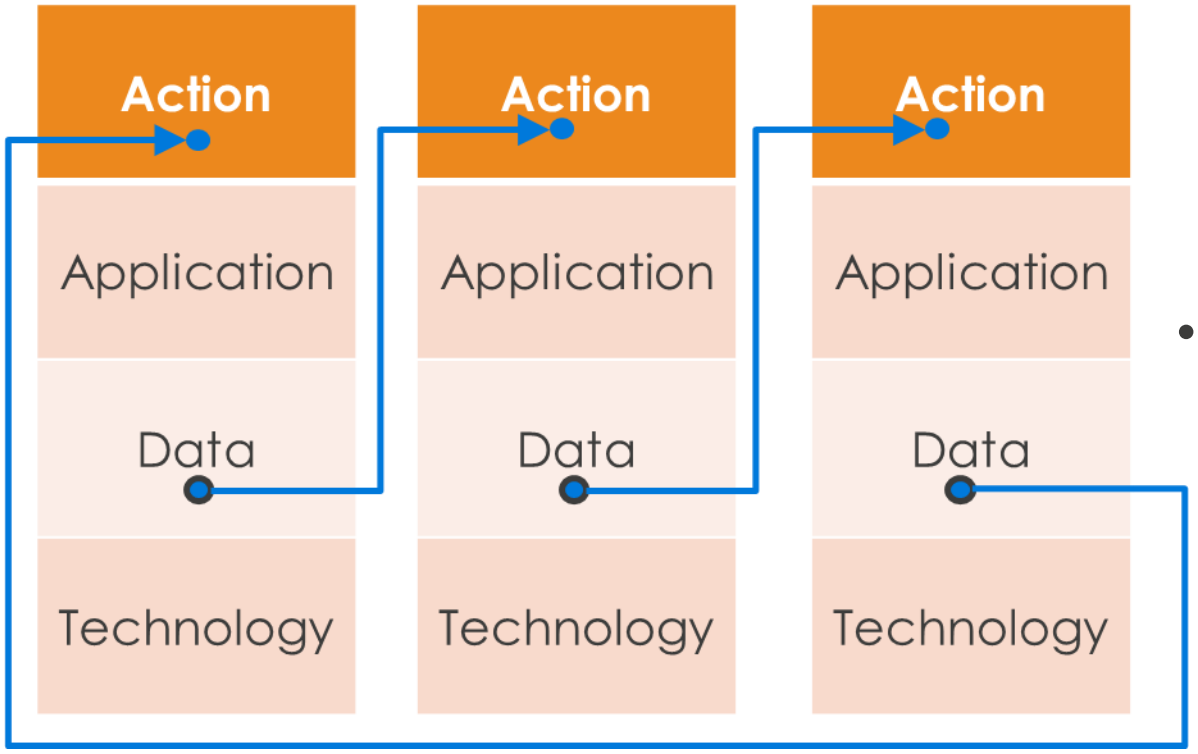
The problem is methods and process: agile your way into the Analytics Shantytown

“Infrastructure one PoC at a time”



- Use case driven
- Leverage (any) new technology
- Re-use of the technology stack
- Data is captive to the application or to the technology stack
- Developers tend to think “function first”
- Analytics people also think “function first”, but generally require integrated data

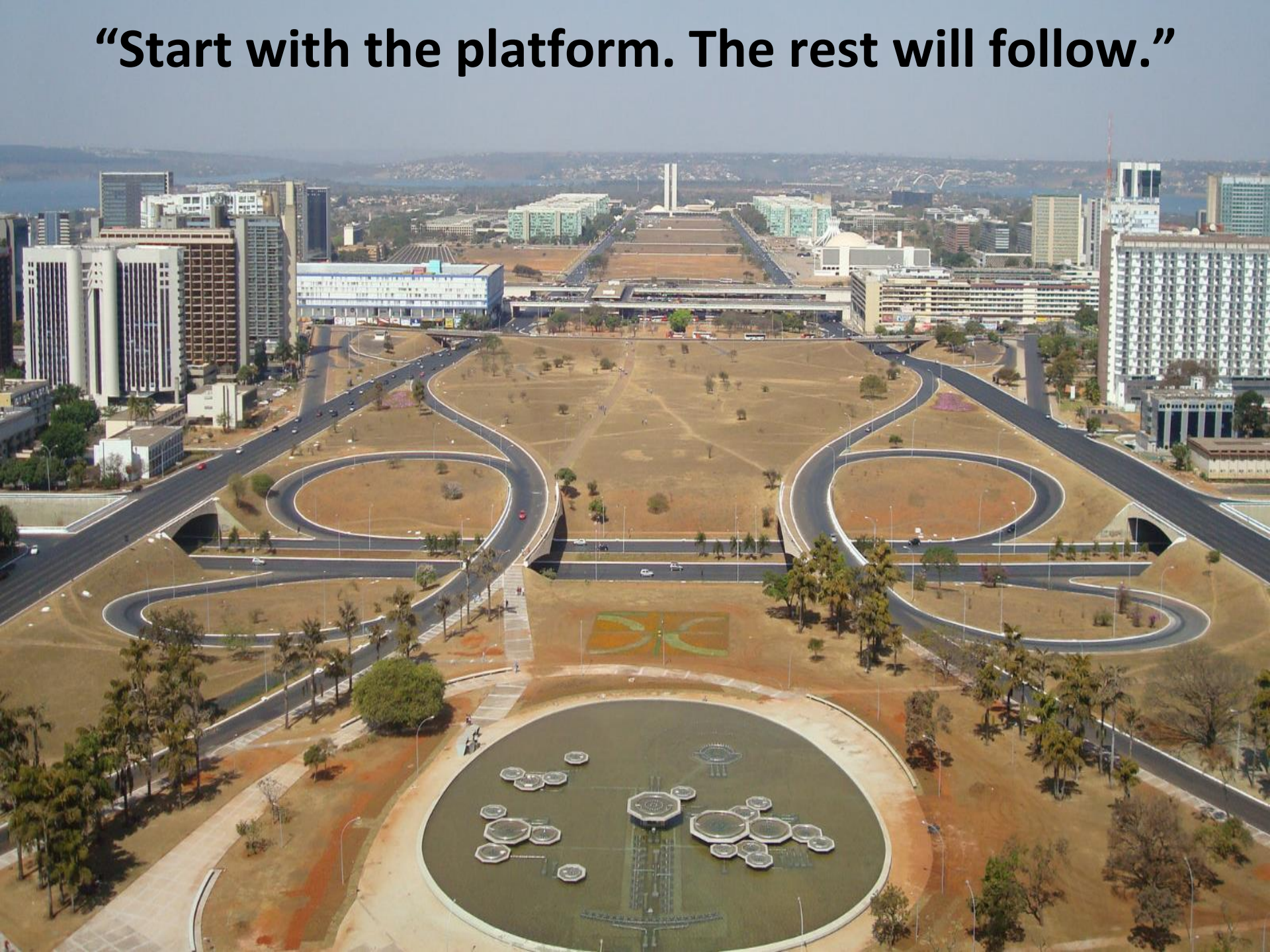
A key aspect of operational analytics is “end-to-end”



- Most data applications are organized around a process, not a task or function.
- Real-time analytics systems pass their data over the network, but the dependencies can cross application boundaries, as well as requiring persistence.
- Developers are being forced to think of data outside the local context.

E2E does not allow you to take a random walk through technologies and BYOT because the complexity of integration leads to a mountain of technical debt.

“Start with the platform. The rest will follow.”



Big data promise: What you see

Big data reality: What users see

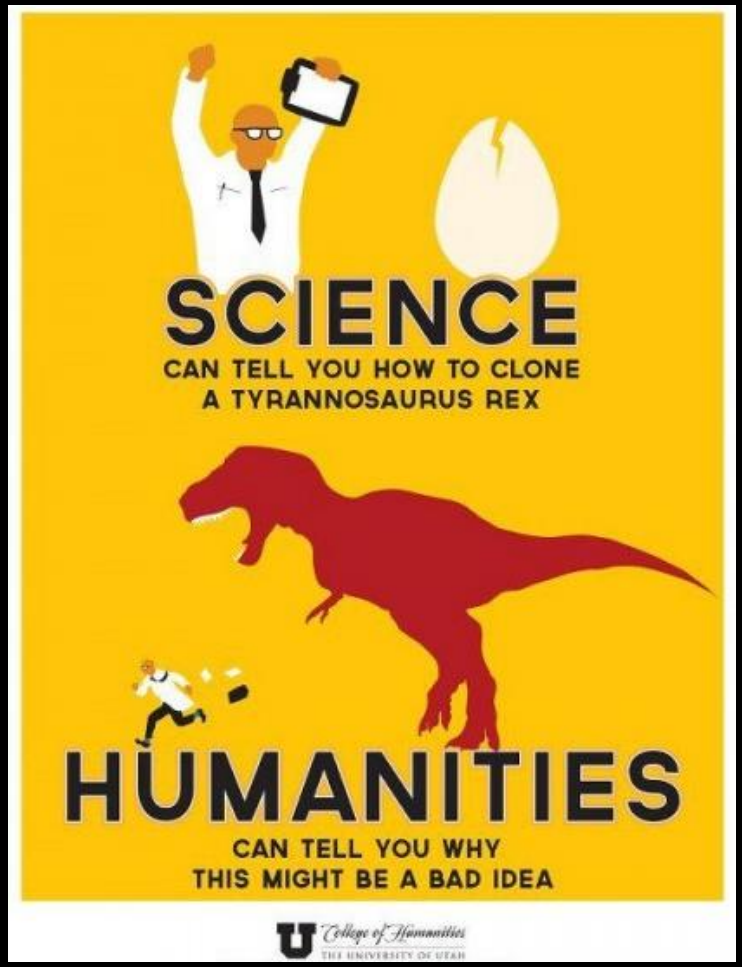

Persisting data is not the end of the line.
If you stop here you win the battle and lose the war

“Begin with the end in mind”

The starting point *can’t be* with technology. That’s like starting with bricks when designing a house. You may get lucky but…

The goals and specific uses are the place to start

- Use dictates need
- Need dictates capabilities
- Capabilities are solved with technology



This is how you avoid spending $2M on a Hadoop and spark cluster in order to serve data to analysts whose primary requirements are met with laptops.

We don't have an analytics problem, just like we didn't have a BI problem

The origin of analytics as "business intelligence" was stated well in 1958:

> "…the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal." ~ *H. P. Luhn*

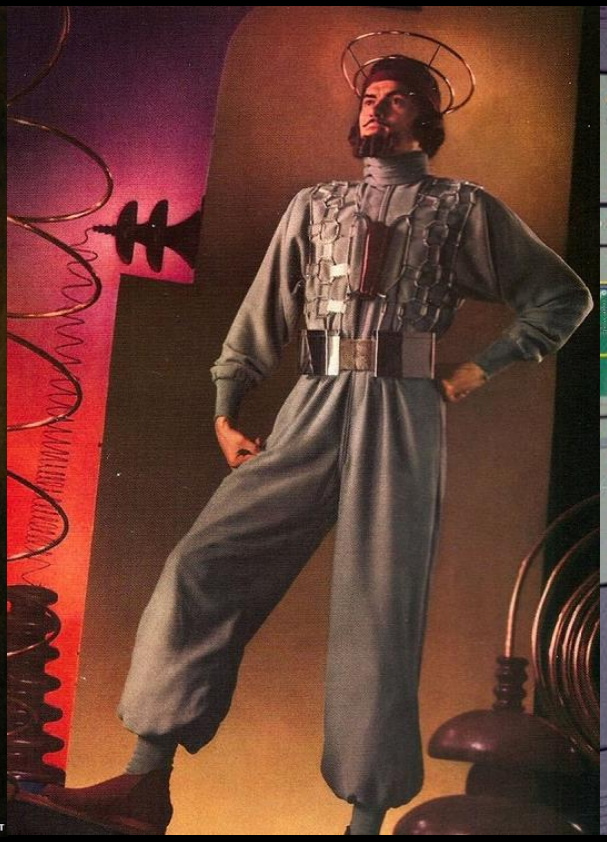
Our goal is analytics as a capability, not a technology

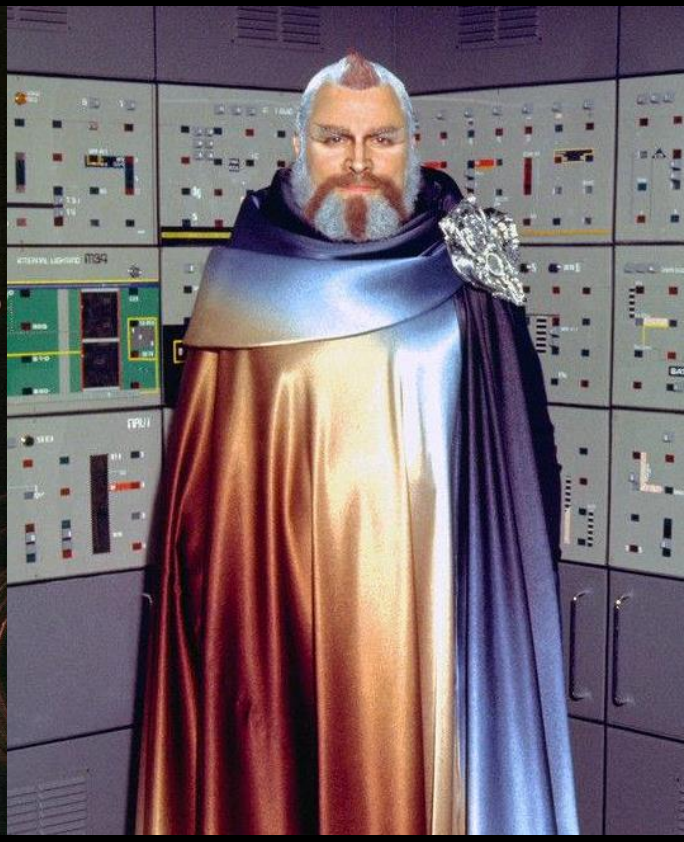
"A Business Intelligence System", http://altaplana.com/ibmrd0204H.pdf

B I

The old problem was access, the new problem is analysis

Three constituencies





Stakeholder

aka the recipient

Analyst

aka the data scientist

Builder

aka the engineer

Starting points for analytics strategy

Many organizations choose to start with the analysts. Create a data science team. Turn them loose to find a problem.

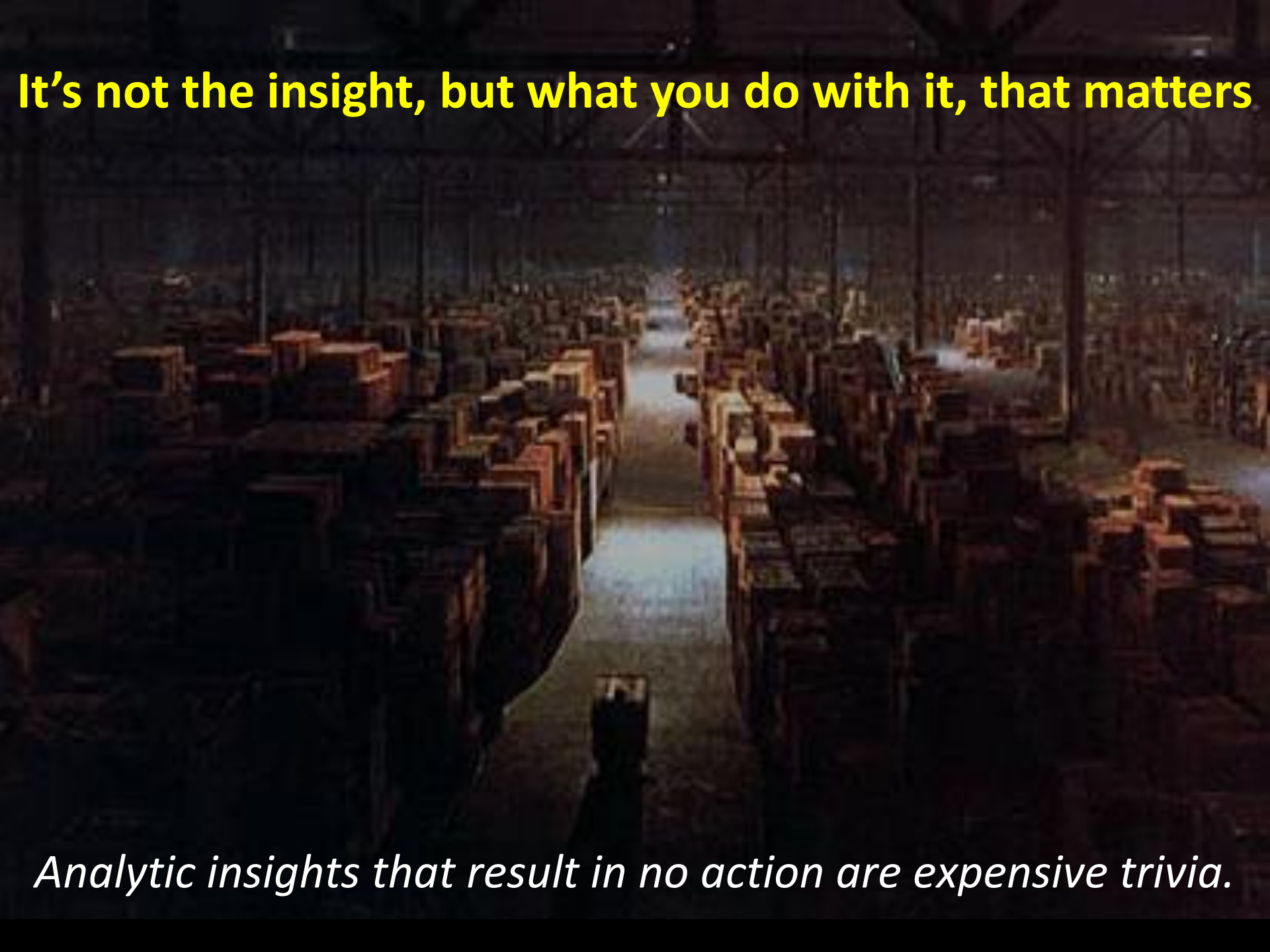
Many more start with builders: technology solutions looking for problems, e.g. 65% of the IT driven Hadoop and Spark projects over the last five years.

The right place to start? Stakeholders. The goal to achieve, the problem to solve.

Each constituency has their own set of problems to deal with

NATURE OF THE PROBLEM FROM THE STAKEHOLDER'S PERSPECTIVE

The myth that still drives analytics – analytic gold


It’s not the insight, but what you do with it, that matters
Analytic insights that result in no action are expensive trivia.

Applying analytics is not an analytics problem

Applying analytics is not in the analyst's control.

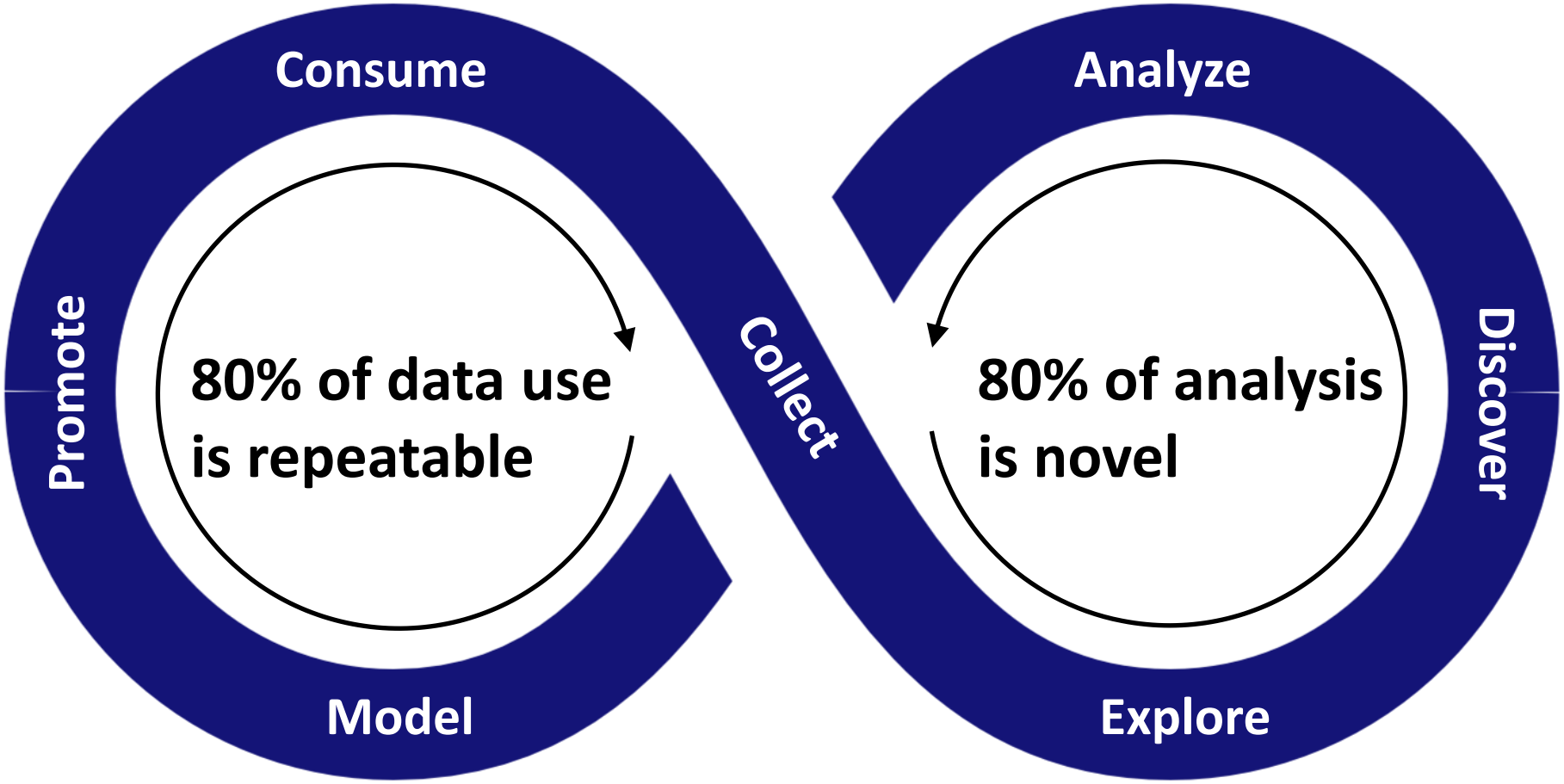
It's not in the engineer's control.

It's in the control of the people involved in the process.

Failures are often in execution, not in analytics development.

For example, we saw unexpectedly poor performance in a number of geographies. Was it the new analytics we tried? Was it a data problem? No, it was a simple compliance problem.

BI and Analysis: Repeatability and Discoverability



Focus is on repeatability

Application cycle time

90% of users *just want answers*

Focus is on discoverability

Analyst cycle time

9% analysts, 1% data scientists

NATURE OF THE PROBLEM FROM THE ANALYST'S PERSPECTIVE

The analytics process at a high level

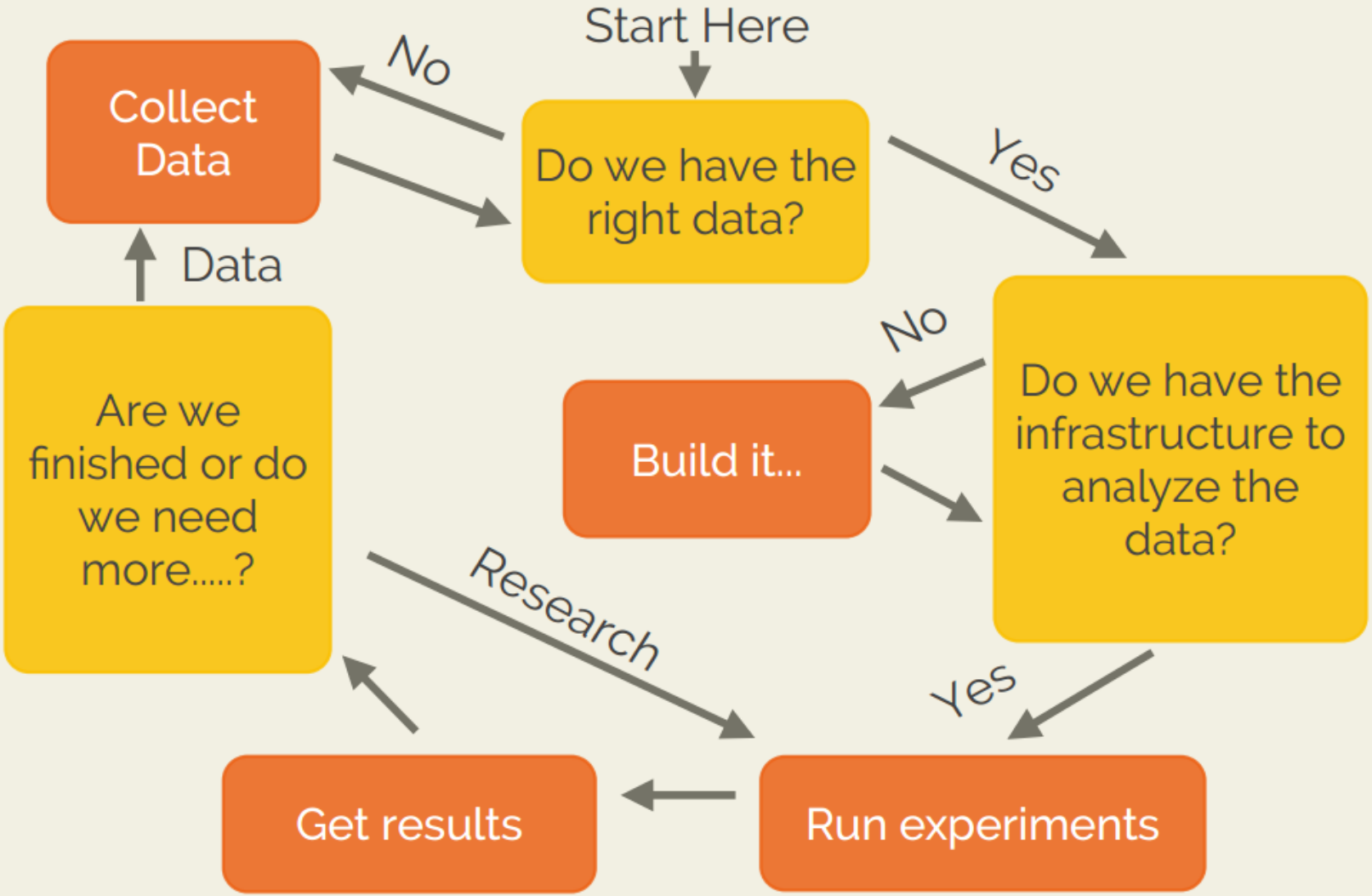
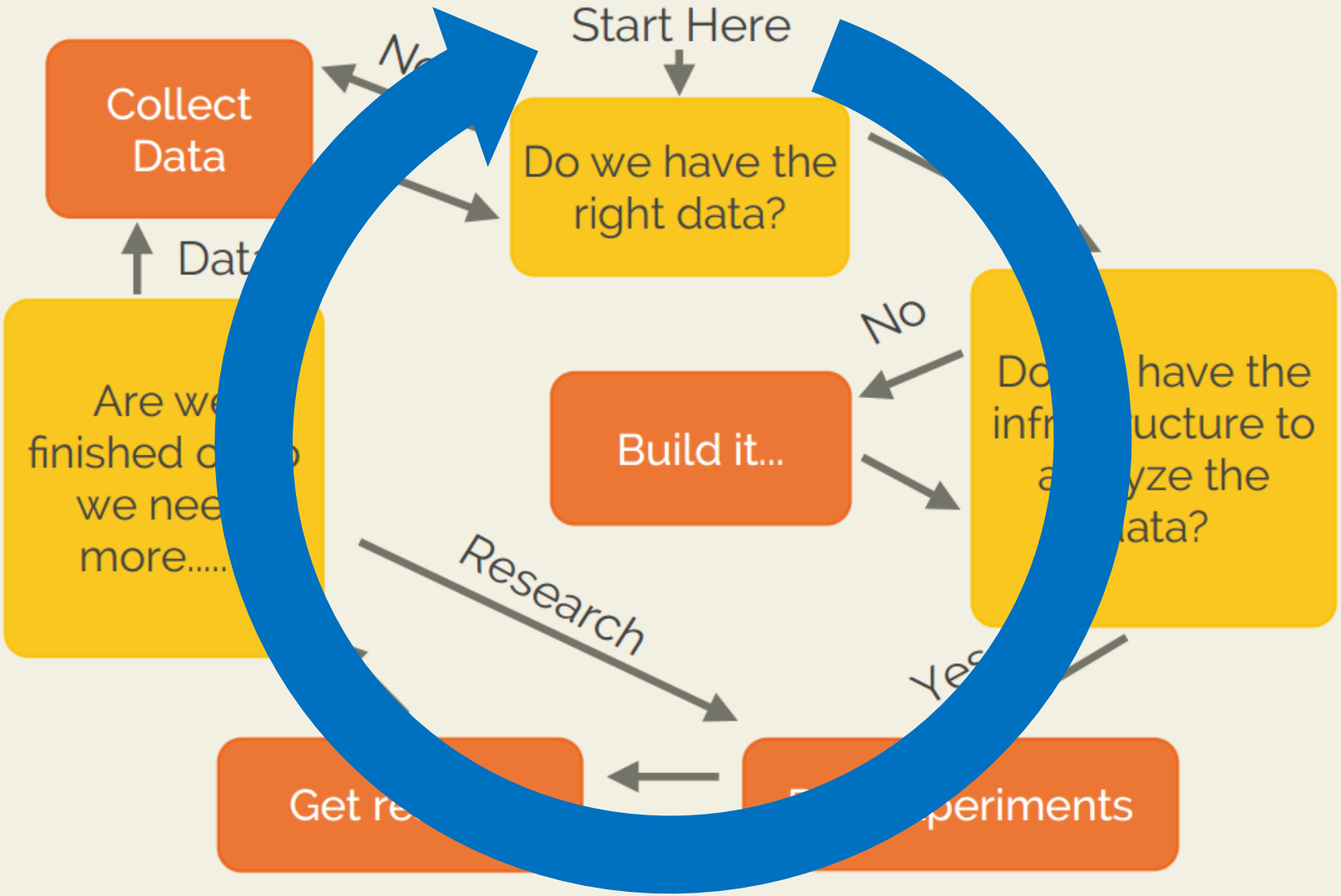


Diagram: Kate Matsudaira

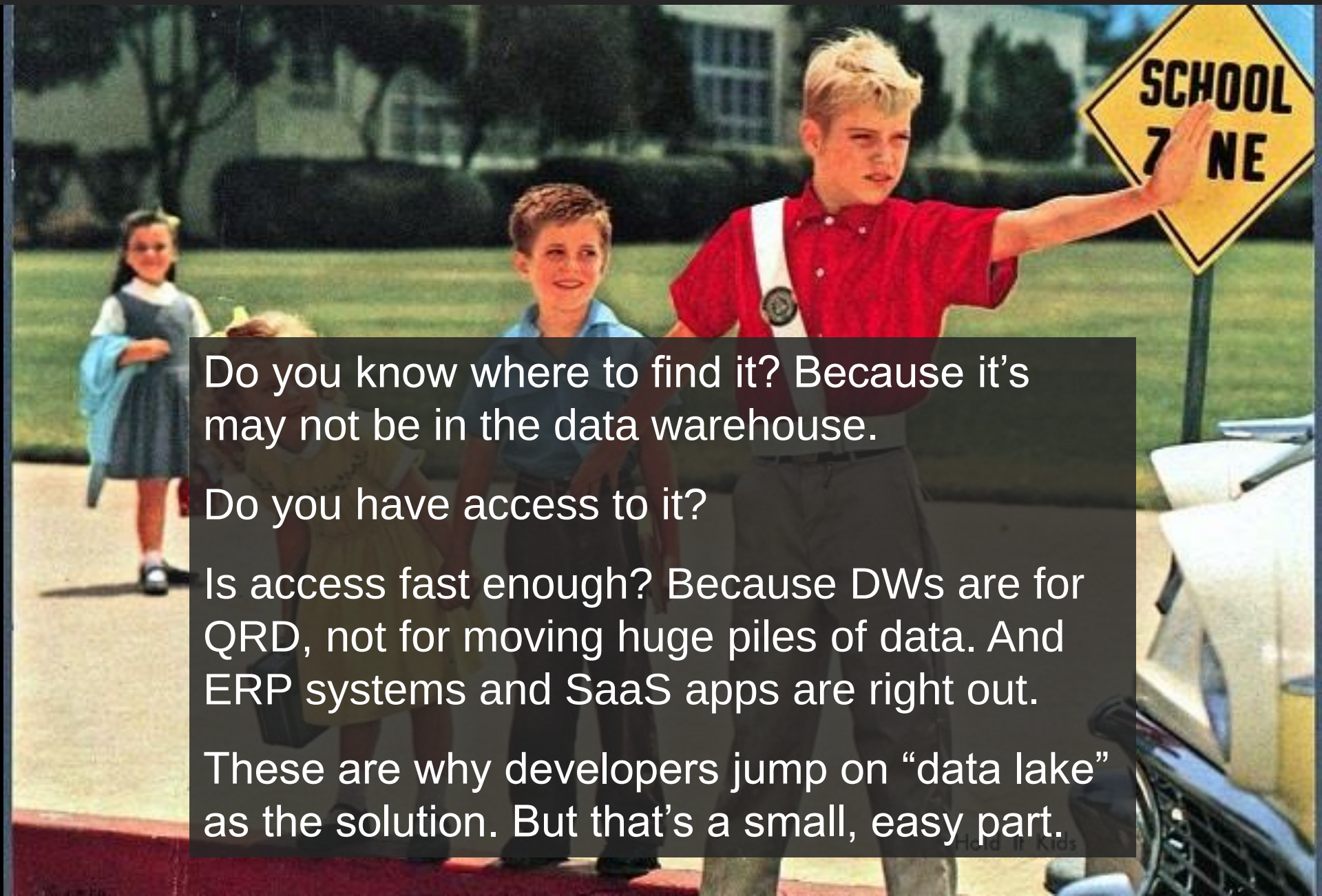
Repeat for each new problem (unlike BI)



The nature of analytics problems is researching the unknown rather than accessing the known.

Diagram: Kate Matsudaira

The main hurdle: just getting the data



Do you know where to find it? Because it's may not be in the data warehouse.

Do you have access to it?

Is access fast enough? Because DWs are for QRD, not for moving huge piles of data. And ERP systems and SaaS apps are right out.

These are why developers jump on "data lake" as the solution. But that's a small, easy part.

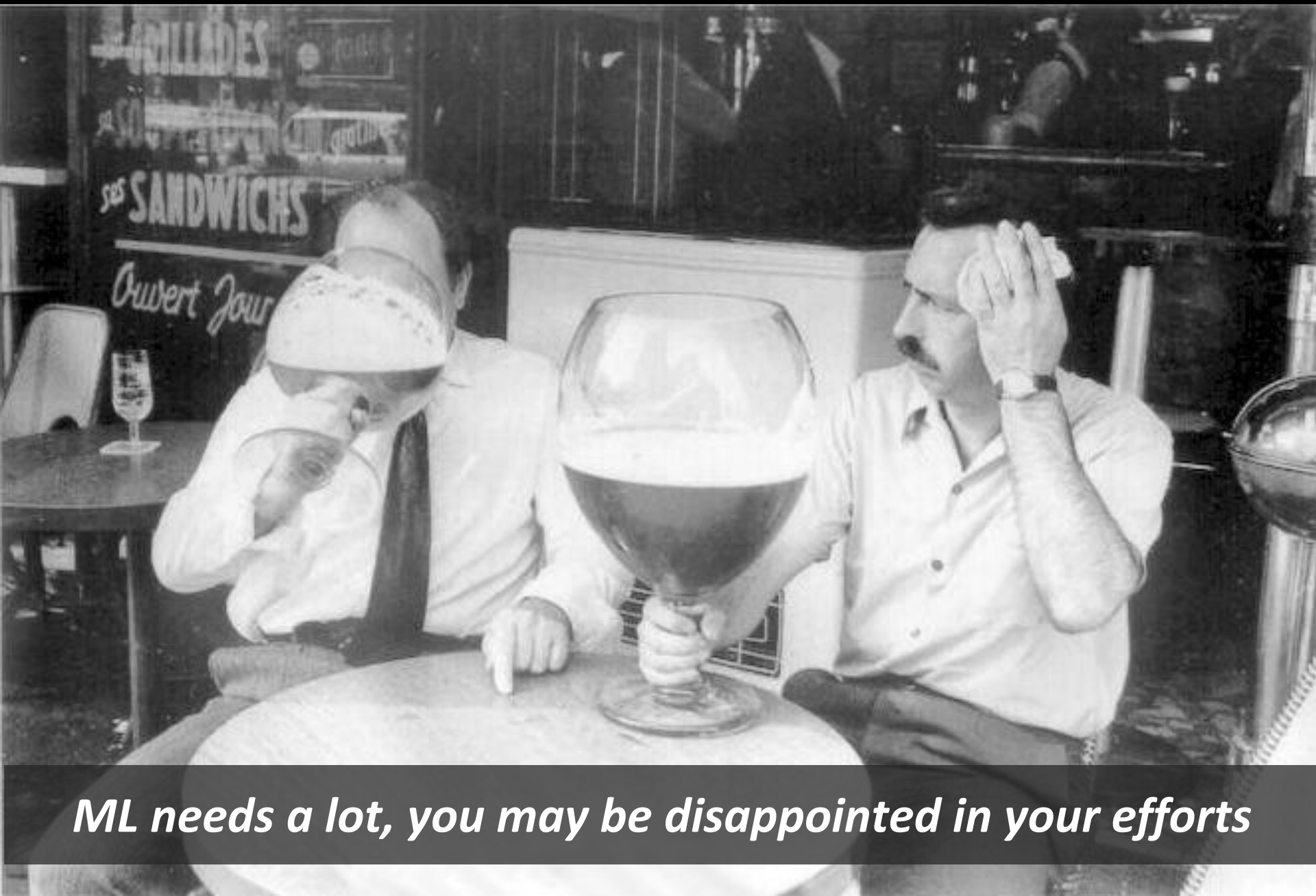
Do you have the right data?

Many machine learning techniques require labeled (known good) training data:

Supervised learning: a person has to define the correct output for some portion of the data. Data is divided into training sets used for model building and test sets for validating the results.

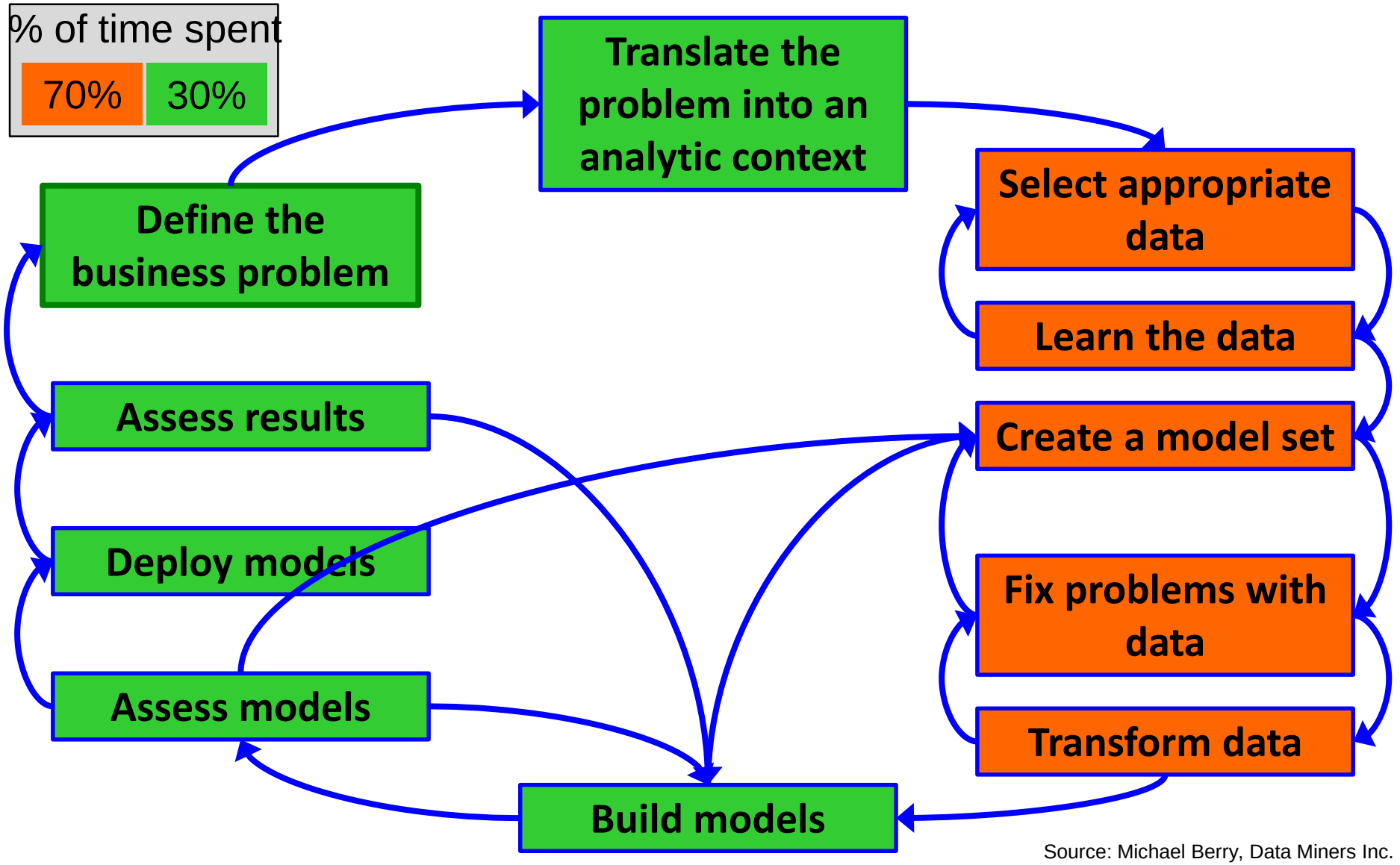
- *What is spam and what isn't?*
- *What does a fraudulent transaction look like*

Do you have enough of the right data?



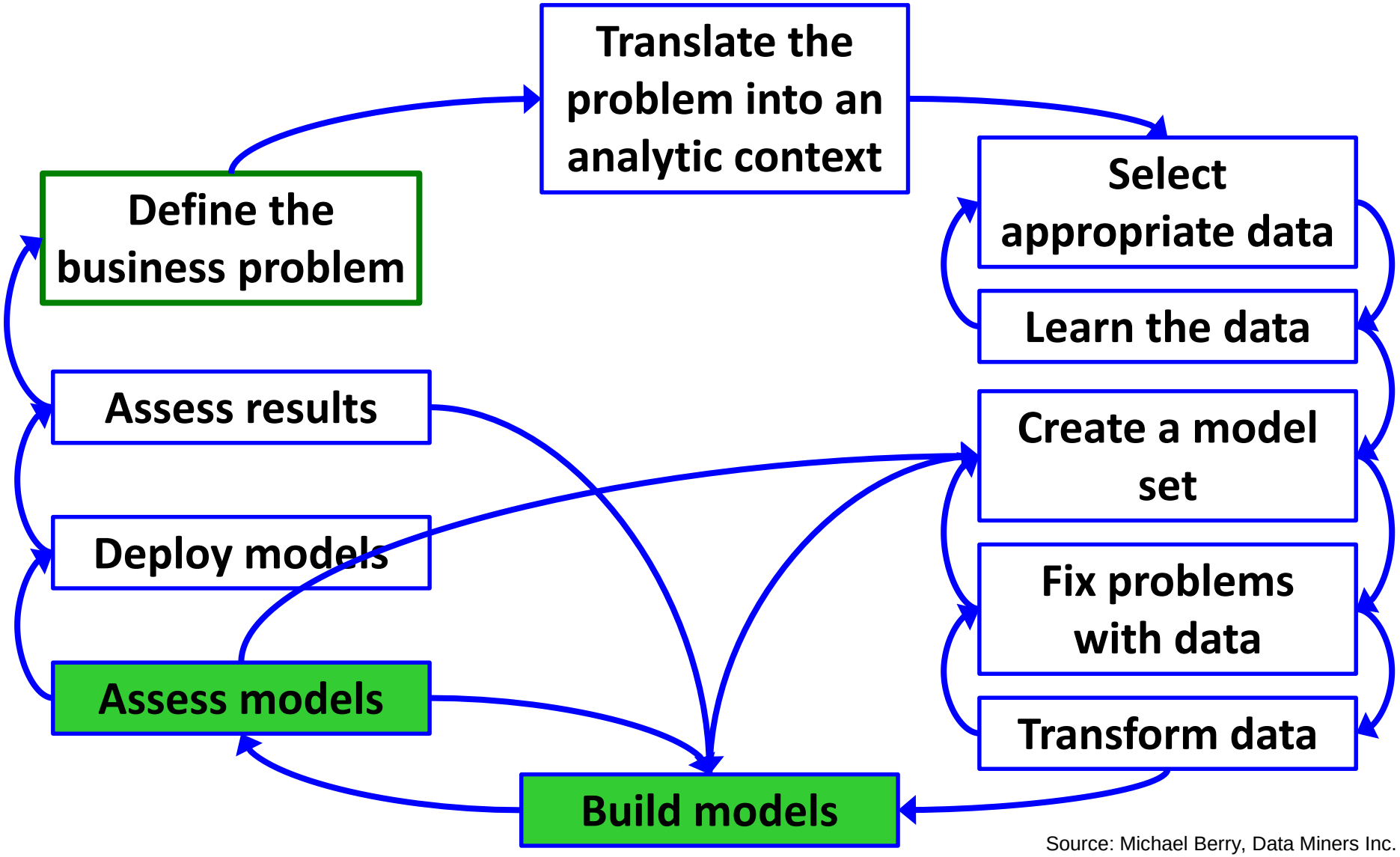
ML needs a lot, you may be disappointed in your efforts

Where do analysts spend their time? *mostly data work*



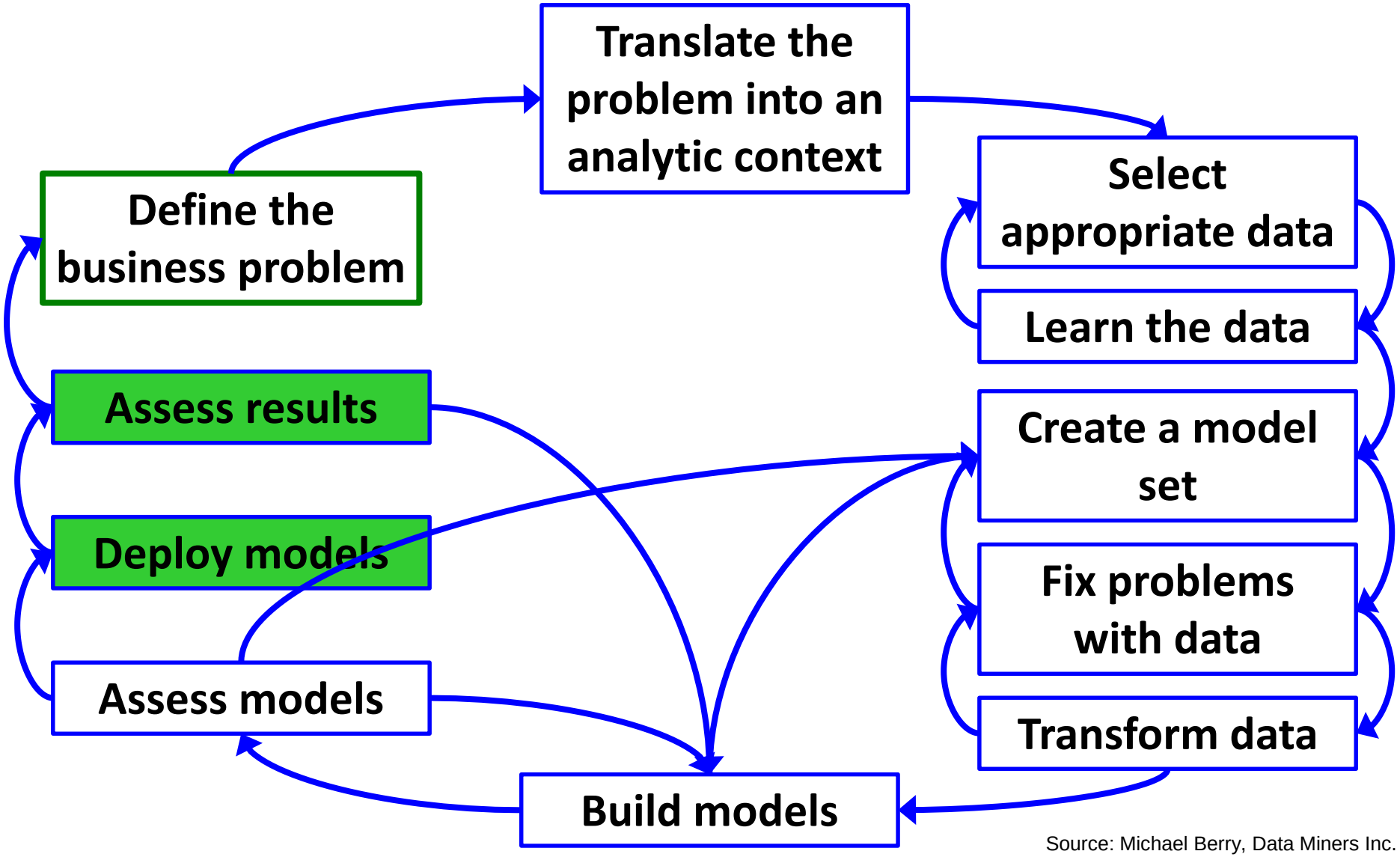
Source: Michael Berry, Data Miners Inc.

Where do most of the analytics tools focus?



Source: Michael Berry, Data Miners Inc.

Where do most of the analytics aaS focus?



Source: Michael Berry, Data Miners Inc.

The analyst's workspace needs to be more like a kitchen than like BI vending machines

Analytics work is a production workload, but…

Do you treat this as a production workload? Storage control, space limitations, resource limitations, production lockdown

NATURE OF THE PROBLEM FROM THE BUILDER'S PERSPECTIVE

IT and Ops people want to know “what to build?”

Giant data platform?

Self service tools?

Analytics has different processes and workloads

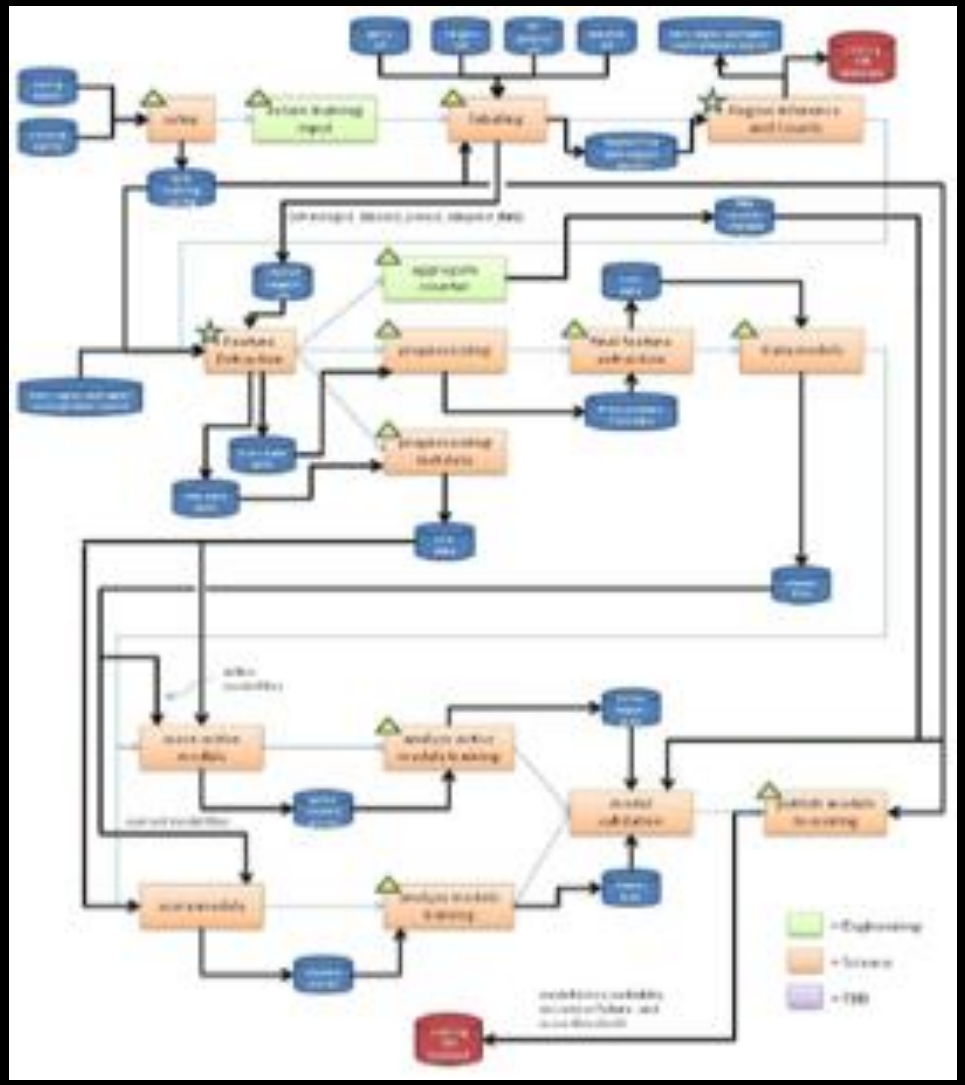
None of this analytics work is the same as what IT considered "analysis" to be, which is usually equated with BI or ad-hoc query.

Ad-hoc analysis ≠

Exploratory data analysis ≠

Batch analytics ≠

Real-time analytics



A real analytics production workflow

Hatch, CIKM '11

Things engineering and operations worry about

Engineering time and effort

- Introduction of new technology, complexity
- Integration - Deployment of models requirements linking different types of environments, creating supportable workflows for the analysts
- Ability to develop and deploy at the required speed

Supportability

- Automation
- The environment requires additional monitoring, other technology and processes, particularly for customer-facing work
- Support costs (time and money)

SLAs:

- *Availability* – if analytics are tied to production operations, particularly customer facing, this becomes important and difficult because it's not standard application work
- *Performance and scalability* – have to manage unpredictable workloads, resource conflicts between model development with model execution

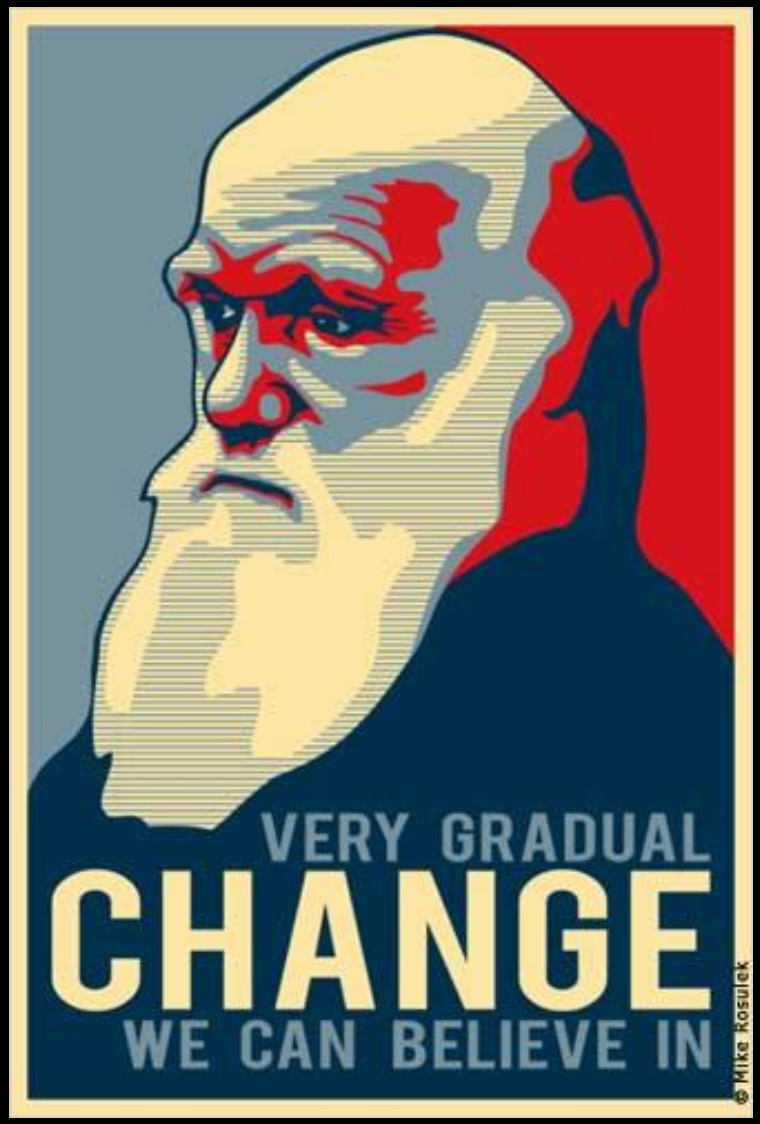
The world changes, do the models?

In BI you maintain ETL and schemas, in ML you maintain models and maybe pipelines.

"Model decay" happens as the assumptions around which a model is built change, e.g. spam techniques change.

When you adjust the model you need to know it is normal again

- Better save the data used to build the model
- Better save the model
- Baseline and measurements



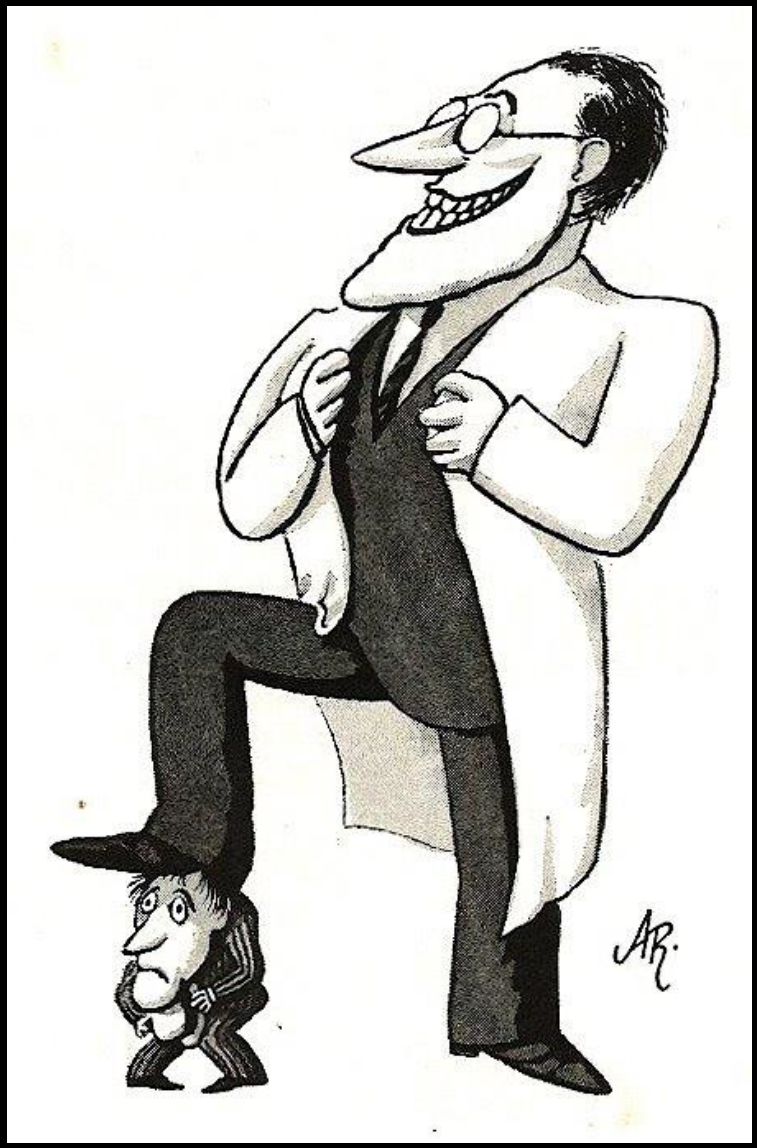
There are requirements from all constituents. You need to put them together to have a complete picture of what's needed.

THREE PERSPECTIVES, ONE SOLUTION?

The missing stakeholder

There is another stakeholder: analytics management - the CAO, CDO, VP of analytics, aka “your boss” if you’re a data scientist.

This is the perspective and problems of the person responsible for oversight of the team and efforts is across the organization and across multiple projects



Job #1 - Repeatability

Job # 2 - Operational predictability

Job #3 - Reproducibility

They need a system of record for analytics

There is an extensive list of requirements to support

Primary requirements needed by constituents	S	D	E
Data catalog and ability to search it for datasets		X	X
Self-service access to curated data		X	
Self-service access to uncurated (unknown, new) data		X	X
Temporary storage for working with data		X	
Data integration, cleaning, transformation, preparation tools and environment		X	X
Persistent storage for source data used by production models		X	X
Persistent storage for training, testing, production data used by models		X	X
Storage and management of models		X	X
Deployment, monitoring, decommissioning models			X
Lineage, traceability of changes made for data used by models		X	X
Lineage, traceability for model changes	X	X	X
Managing baseline data / metrics for comparing model performance	X	X	X
Managing ongoing data / metrics for tracking ongoing model performance	X	X	X

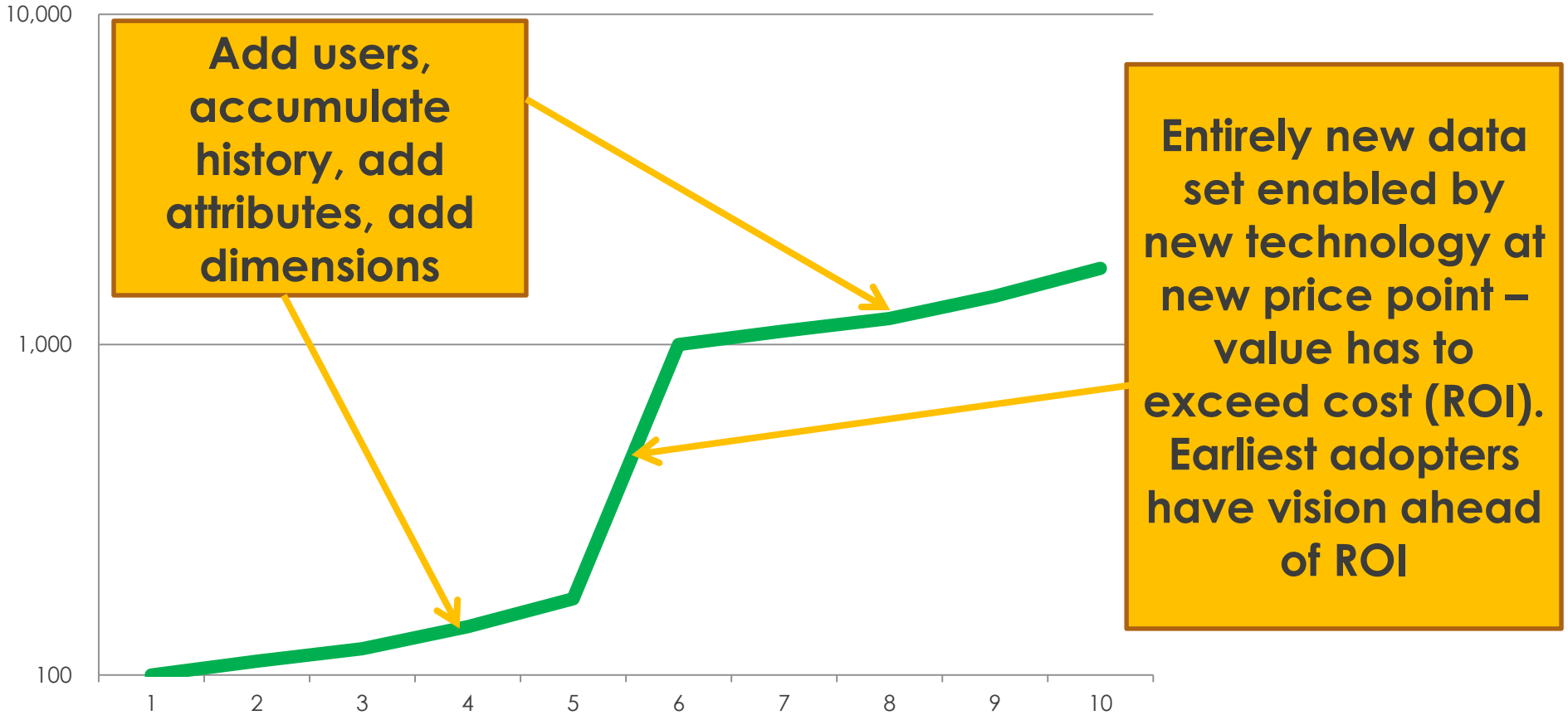
S = stakeholder, user, D = data scientist, analyst, E = engineer, developer

How did we get to this state with BI & analytics?

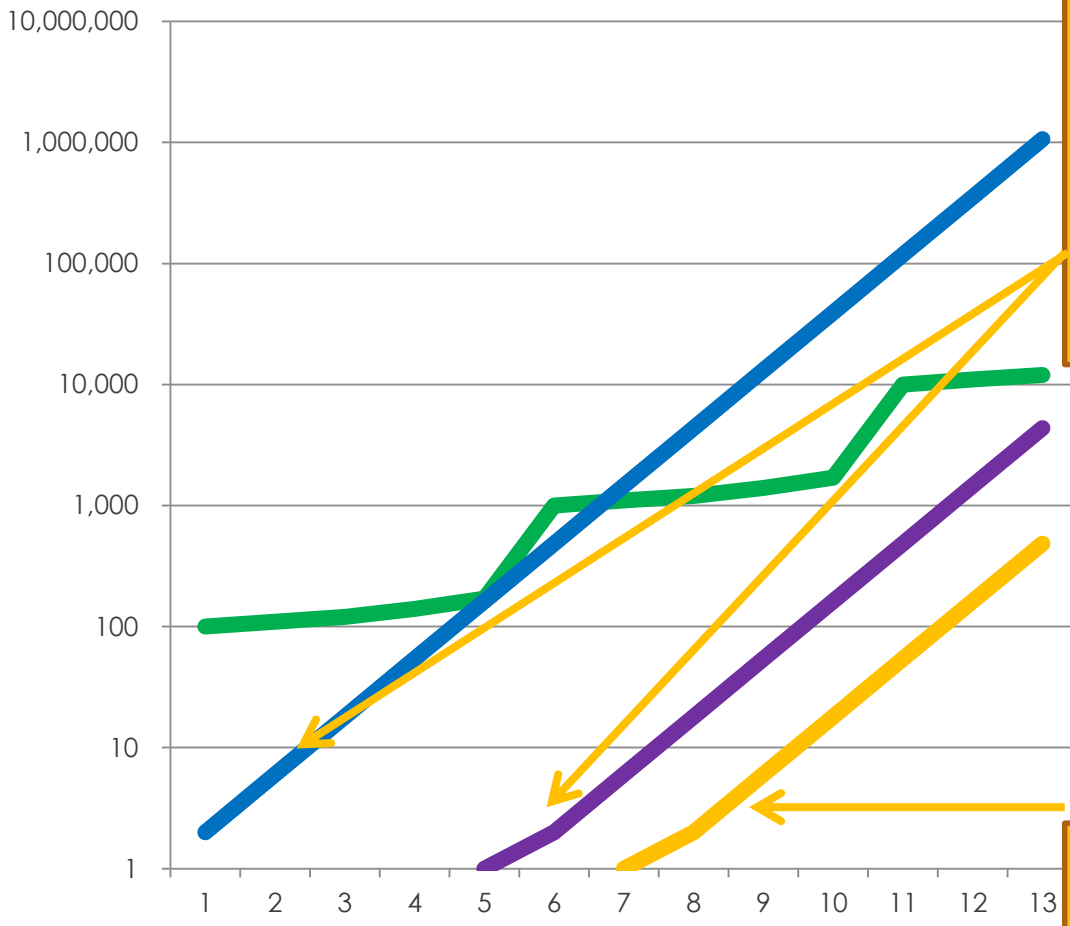
There's a difference between having no past and actively rejecting it.

Third Nature

Customer Data Plateaus



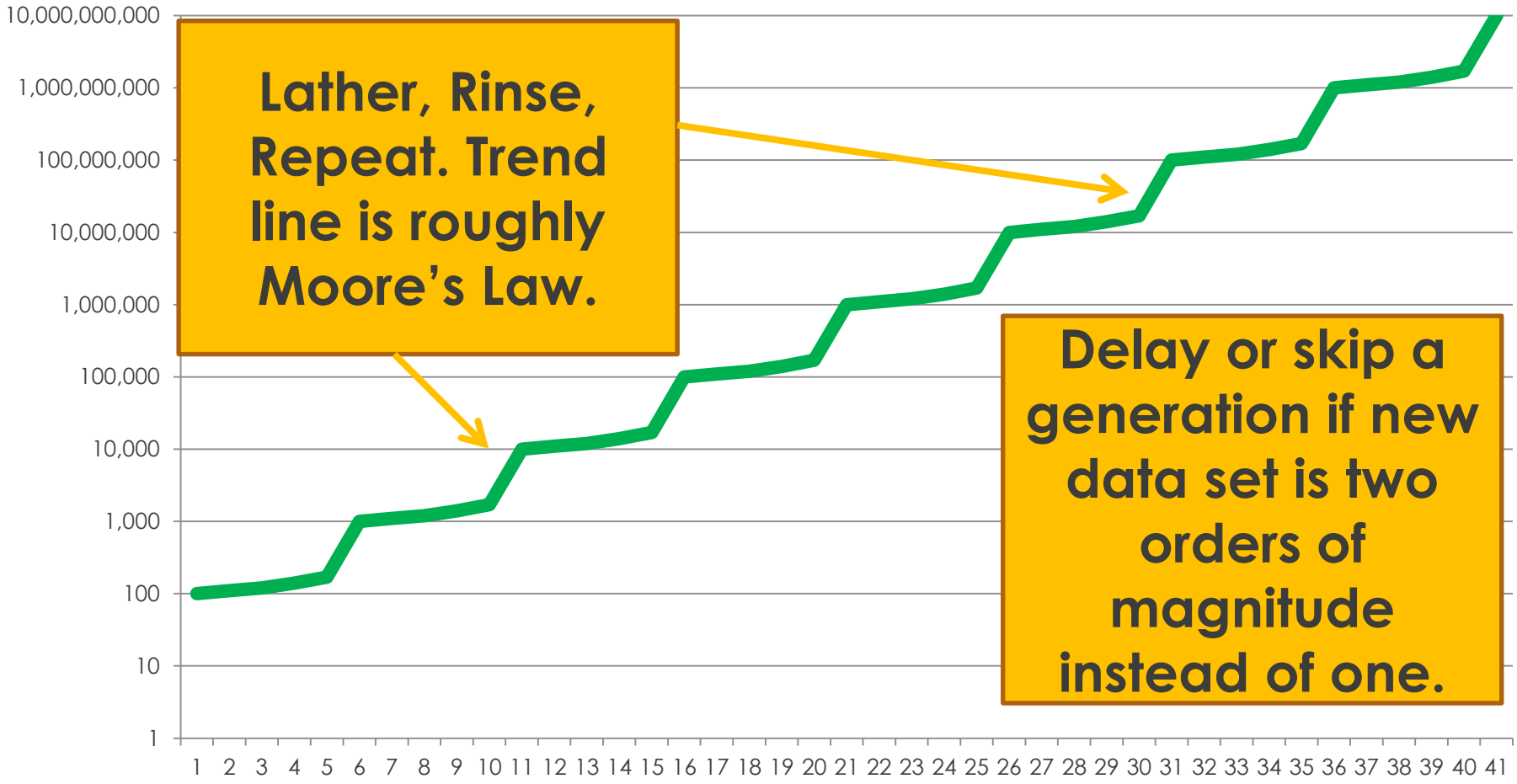
User Adoption of New Data



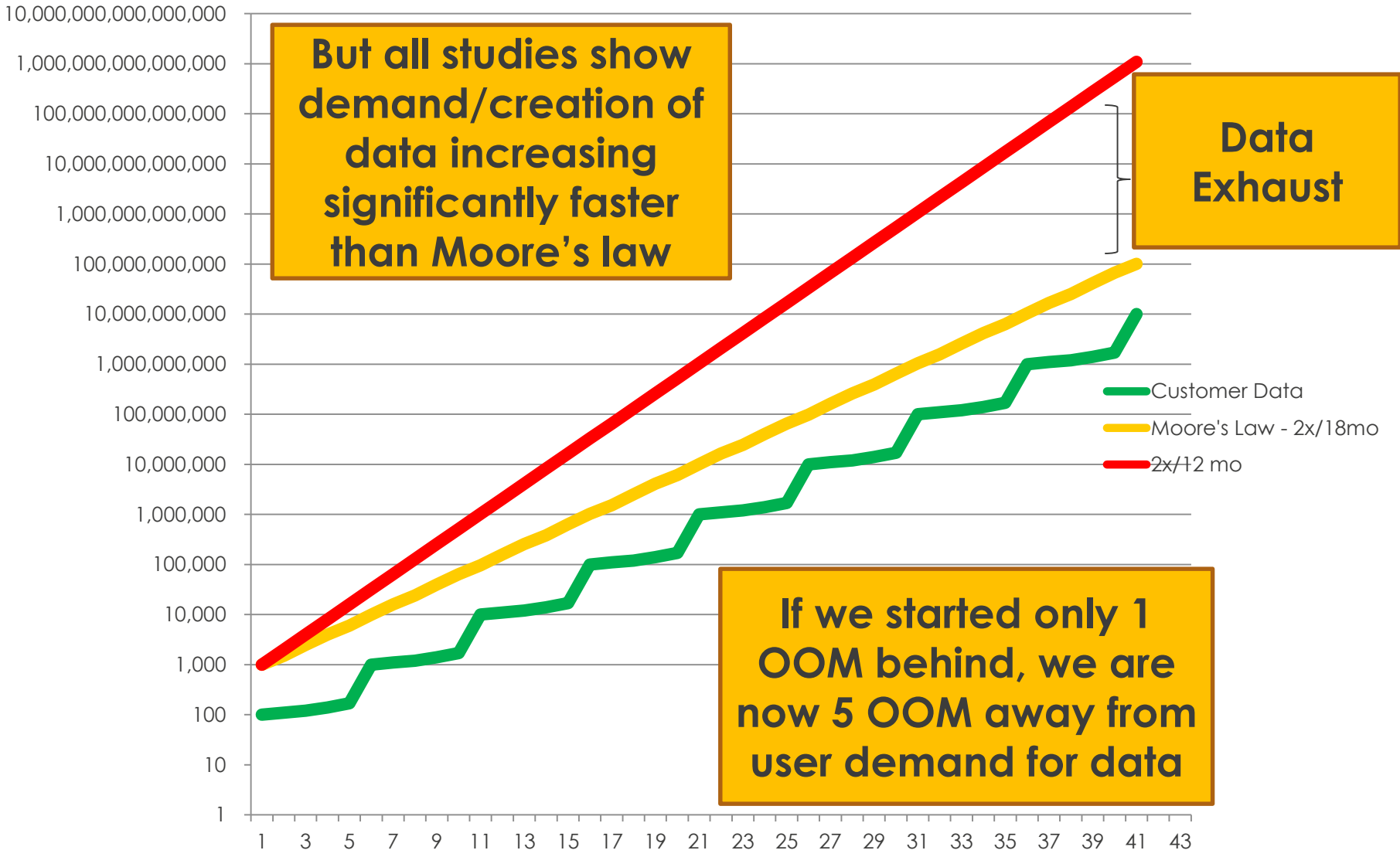
User adoption of new data sets starts over. Very small number of experts growing to wider audience, sophisticated users moving to business analysts, then business users, B2B customers and even to consumers

Greater value is derived when data sets are linked – see bigger picture (eg who buys what, when). Comes after initial extraction of easy value from standalone data

Data Size Plateaus over Time

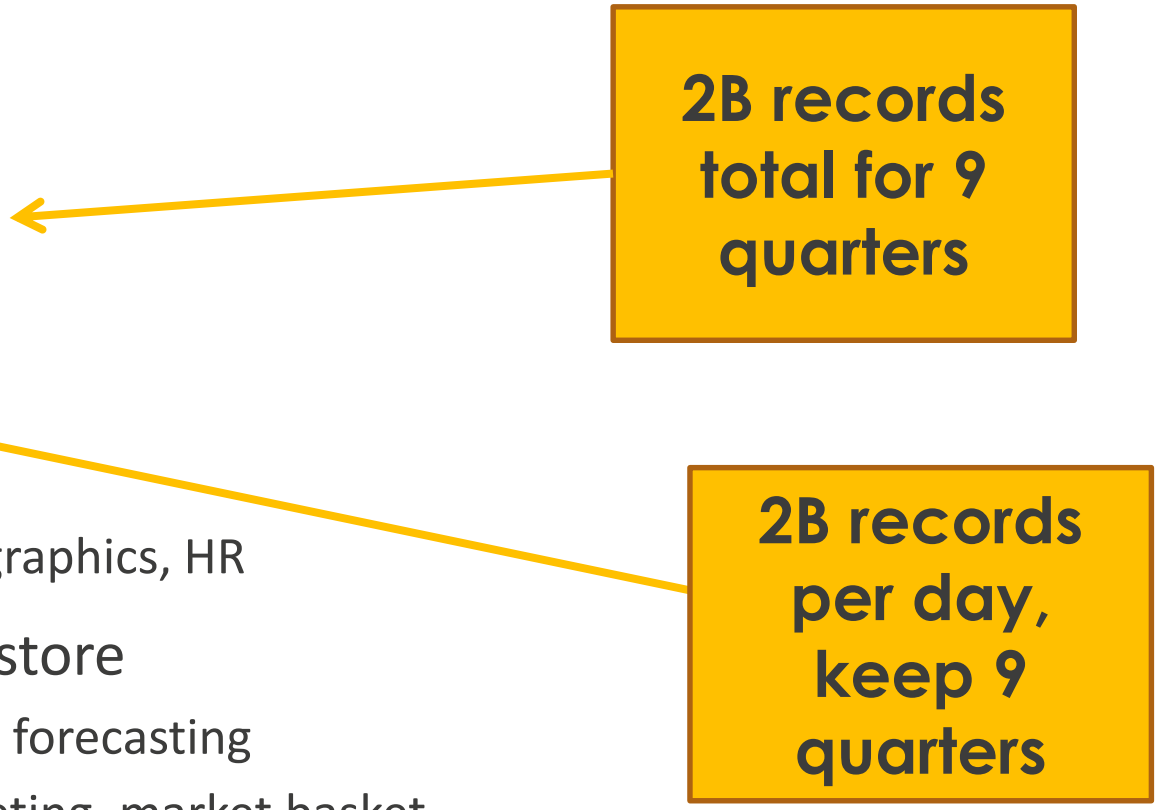


Customer Data Size vs. 2x per 12mo



Retail Plateaus

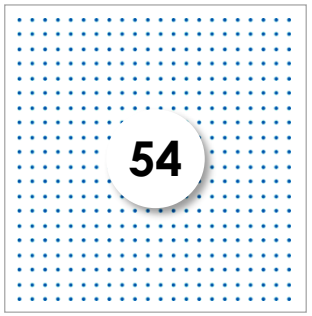
- <Store, Item, Week>
- <Store, Item, Day>
  - Simple aggregations
- Market Basket
  - Affinity
  - Link to person, demographics, HR
- Inventory by SKU by store
  - Temporal, time series, forecasting
  - Link to product, marketing, market basket



Retail Plateaus

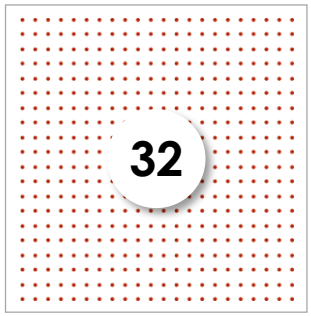
- Web Logs and traffic
  - Behavioral patterns – eg path linked to person, offers, other channels
  - Operations of the web site
- Supply chain sensors – sampled at major event
  - Activity Based Costing
  - Link to customer, product, HR, planning
- Social Media
  - Text analysis, Filtering, languages
  - Link to customer, sales, other channel interactions
- Supply chain sensors – sampled at minutes or seconds
  - Telematics
  - Real time, Event detection, trending, static and dynamic rules
  - Link to HR, thresholds, forecasts, routing, planning

How many batteries are in inventory by plant?



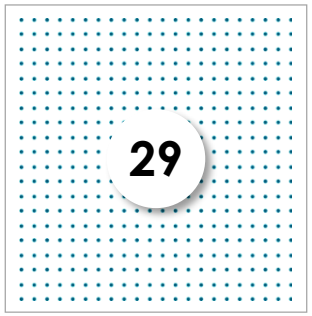
OPERATIONS
Inventory
Returns
Manufacturing
Supply Chain

What is the trend of warranty costs?



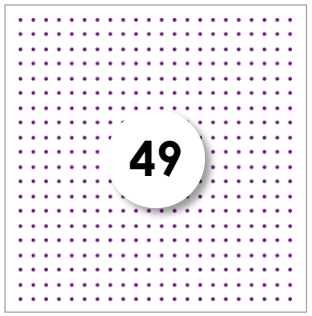
FINANCE
Revenue
Expenses
Customers

How many people made a warranty claim last week?



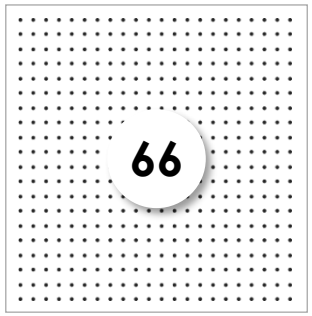
CUSTOMER CARE
Customer
Products
Orders
Case History

How many sales have been made quarter to date?

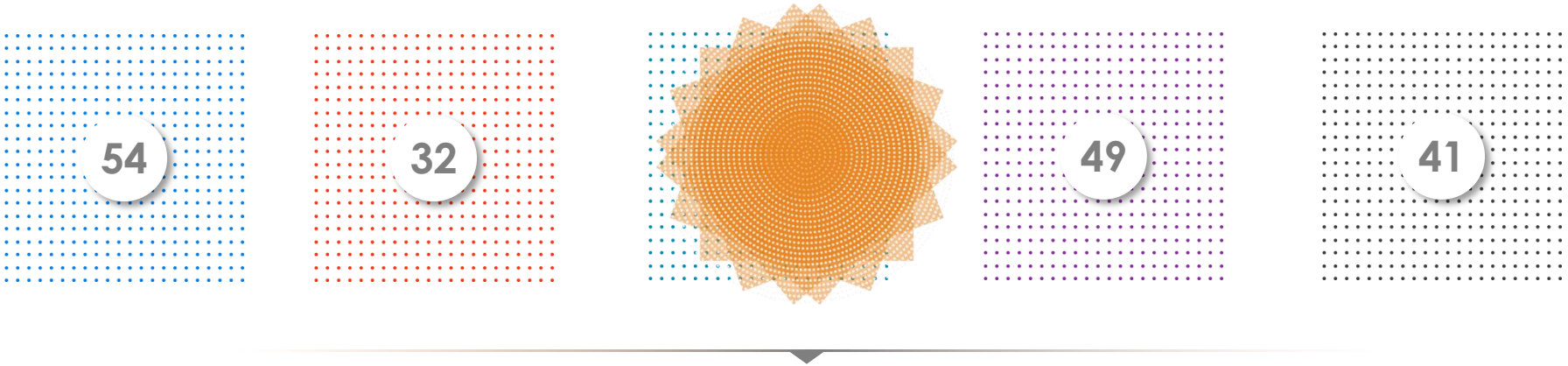


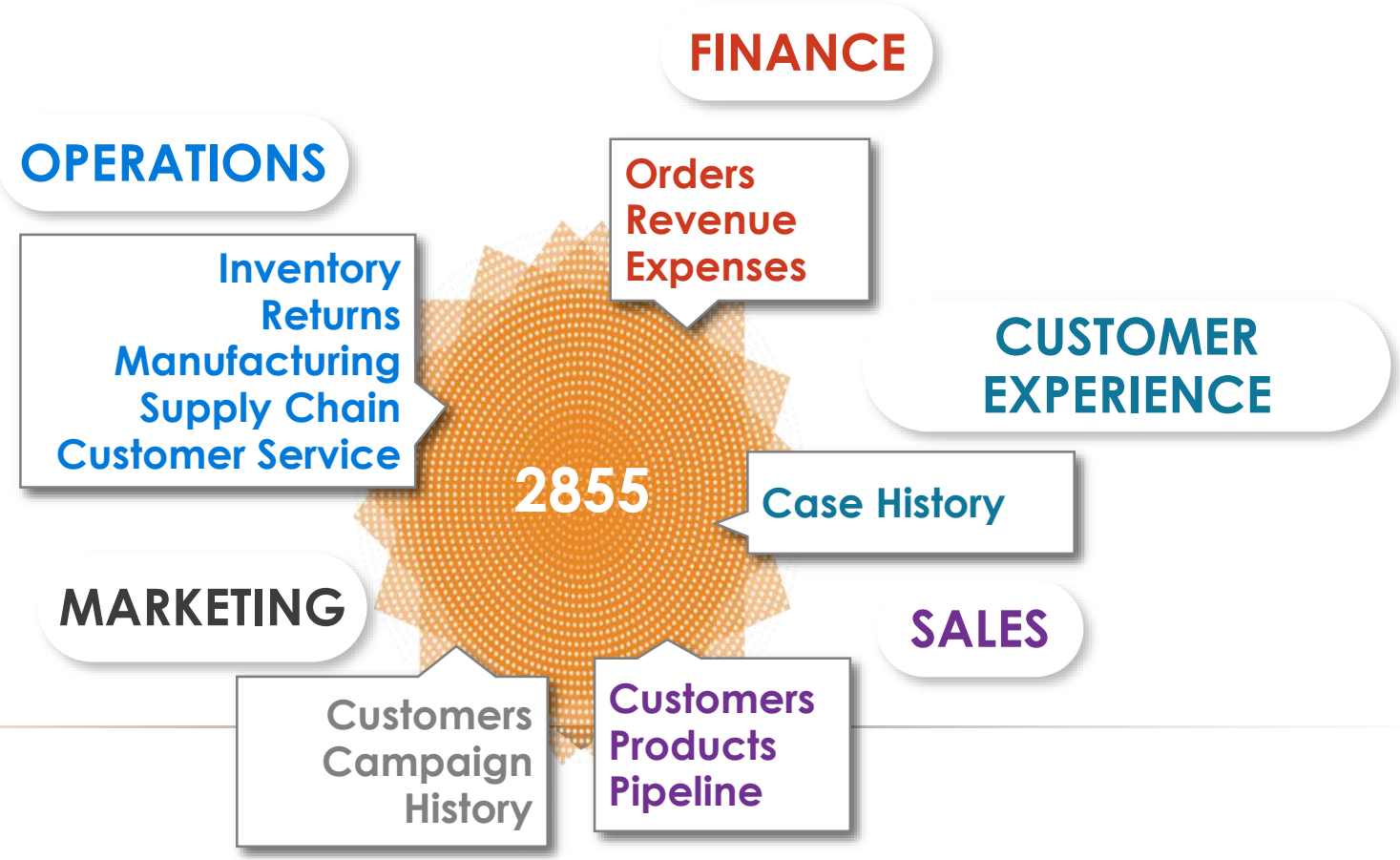
SALES
Orders
Customers
Products

Which customers should get a communication on extended warranties?



MARKETING
Customers
Orders
Campaign
History


54
32
49
41



Given the rise in **warranty costs**, isolate the problem to be a **plant**, then to a **battery lot**.

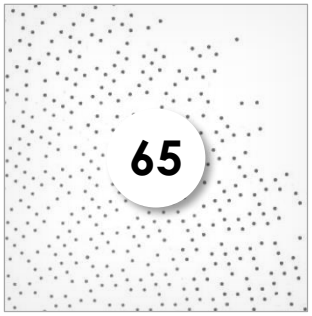
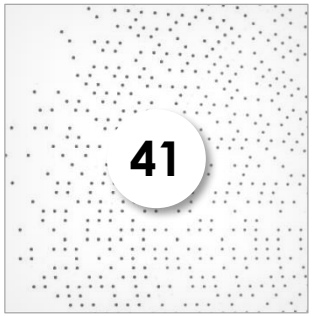
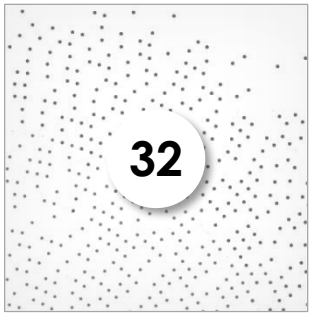
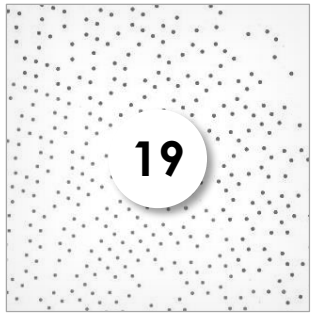
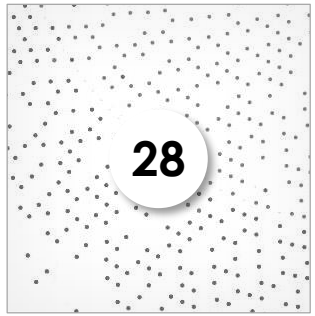
Communicate with **affected customers**, who have not made a warranty claim on batteries, through **Marketing** and **Customer Service** channels to recall cars with affected batteries.

Manufacturing: Data Overlap Analysis

New Business Improvement Opportunities through Data Leverage

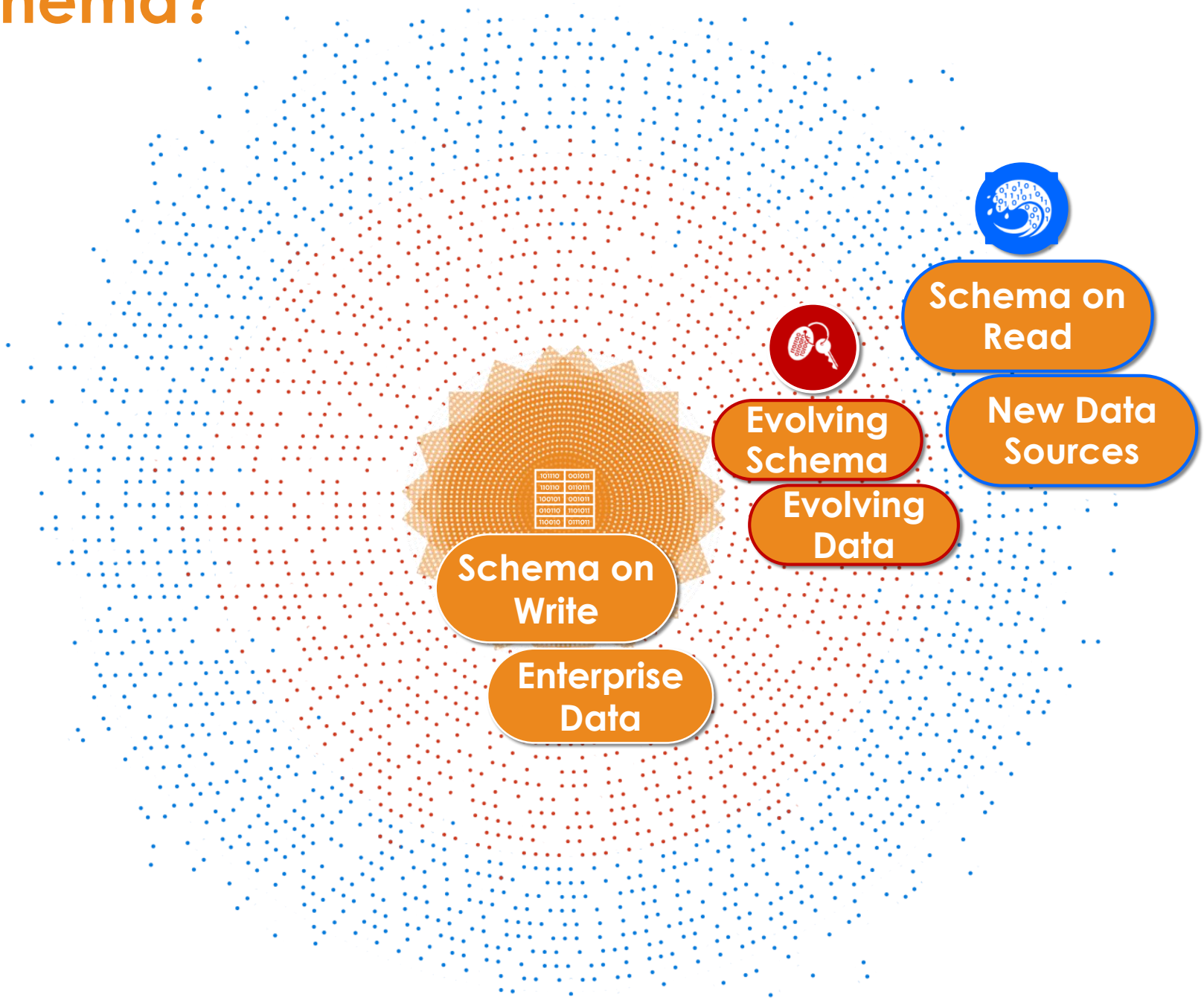
Then \ If	Sales Force Profitability Analysis	Transportation Planning	Production Planning	Vendor Managed Inventory	Global Pricing Rationalization	Fulfillment (Perfect Order)	Manufacturing Quality Optimization	Preventative Maintenance Analysis	Warranty Claims Analysis	Quality Life Cycle Improvement
Sales Force Profitability Analysis	**100%**	**80%**	66%	24%	41%	66%	0%	24%	0%	24%
Transportation Planning	11%	**100%**	**87%**	28%	64%	56%	22%	34%	13%	45%
Production Planning	6%	57%	**100%**	19%	**83%**	35%	17%	28%	9%	40%
Vendor Managed Inventory	12%	**100%**	**100%**	**100%**	**78%**	**100%**	61%	**100%**	39%	**100%**
Global Pricing Rationalization	4%	50%	**100%**	17%	**100%**	27%	15%	29%	11%	41%
Fulfillment (Perfect Order)	16%	**94%**	**90%**	48%	58%	**100%**	35%	53%	19%	56%
Manufacturing Quality Optimization	0%	**76%**	**88%**	60%	66%	**71%**	**100%**	**79%**	43%	**73%**
Preventative Maintenance Analysis	7%	**75%**	**93%**	61%	**80%**	68%	50%	**100%**	27%	**82%**
Warranty Claims Analysis	0%	**94%**	**100%**	**83%**	**100%**	**83%**	**94%**	**93%**	**100%**	**98%**
Quality Life Cycle Improvement	4%	57%	**75%**	35%	64%	41%	26%	47%	16%	**100%**


BIOS

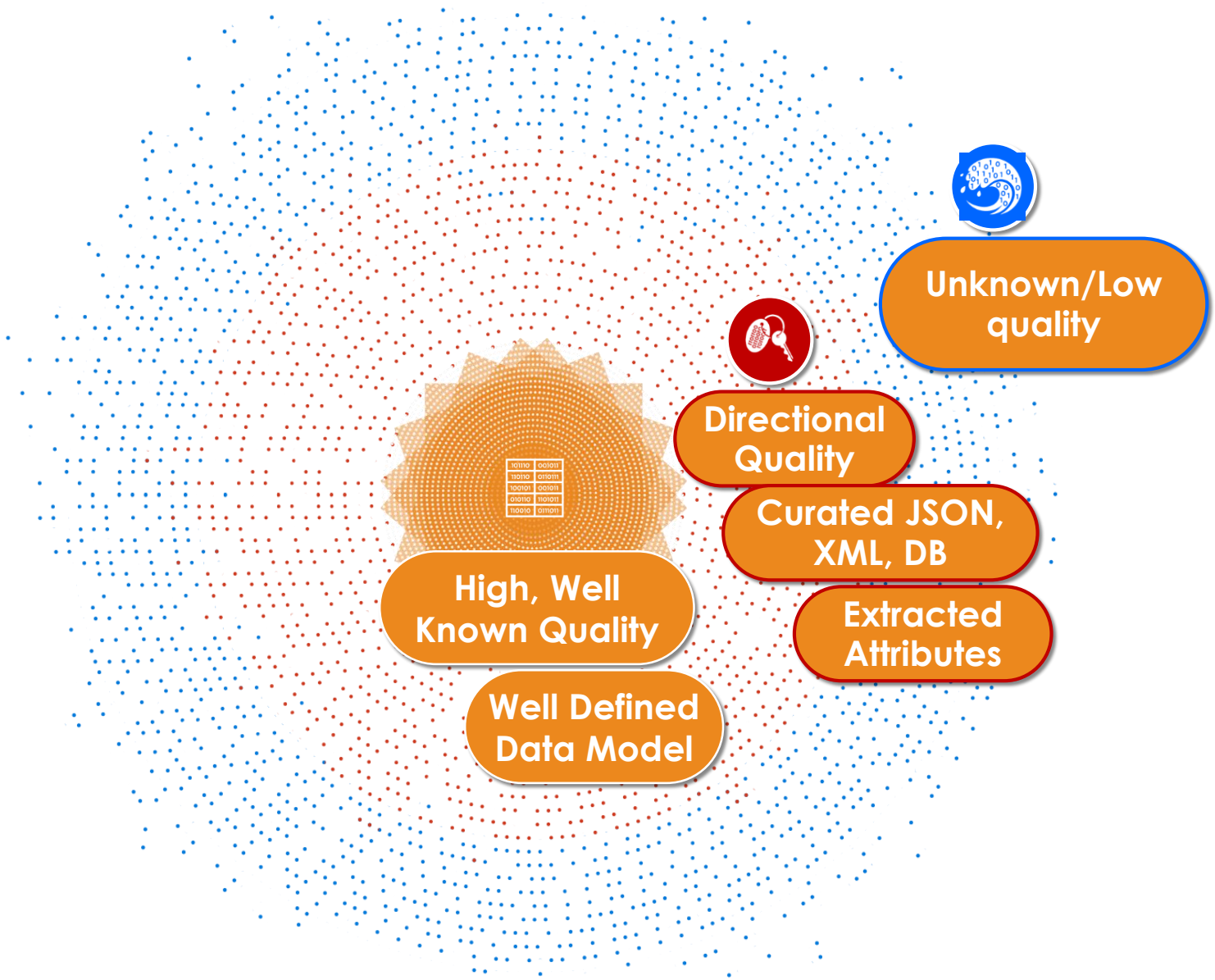
How many visitors did we have to our hybrid cars microsite yesterday?	What are the temperature readings for batteries by Manufacturer?	What is the sentiment towards line of hybrid vehicles?	Which customers likely expressed anger with customer care?	Which ad creative generated the most clicks?
				
CLICKSTREAM	PRODUCT SENSOR	SOCIAL MEDIA	CUSTOMER CARE AUDIO RECORDINGS	DIGITAL ADVERTISING

Enterprise Data

Schema?

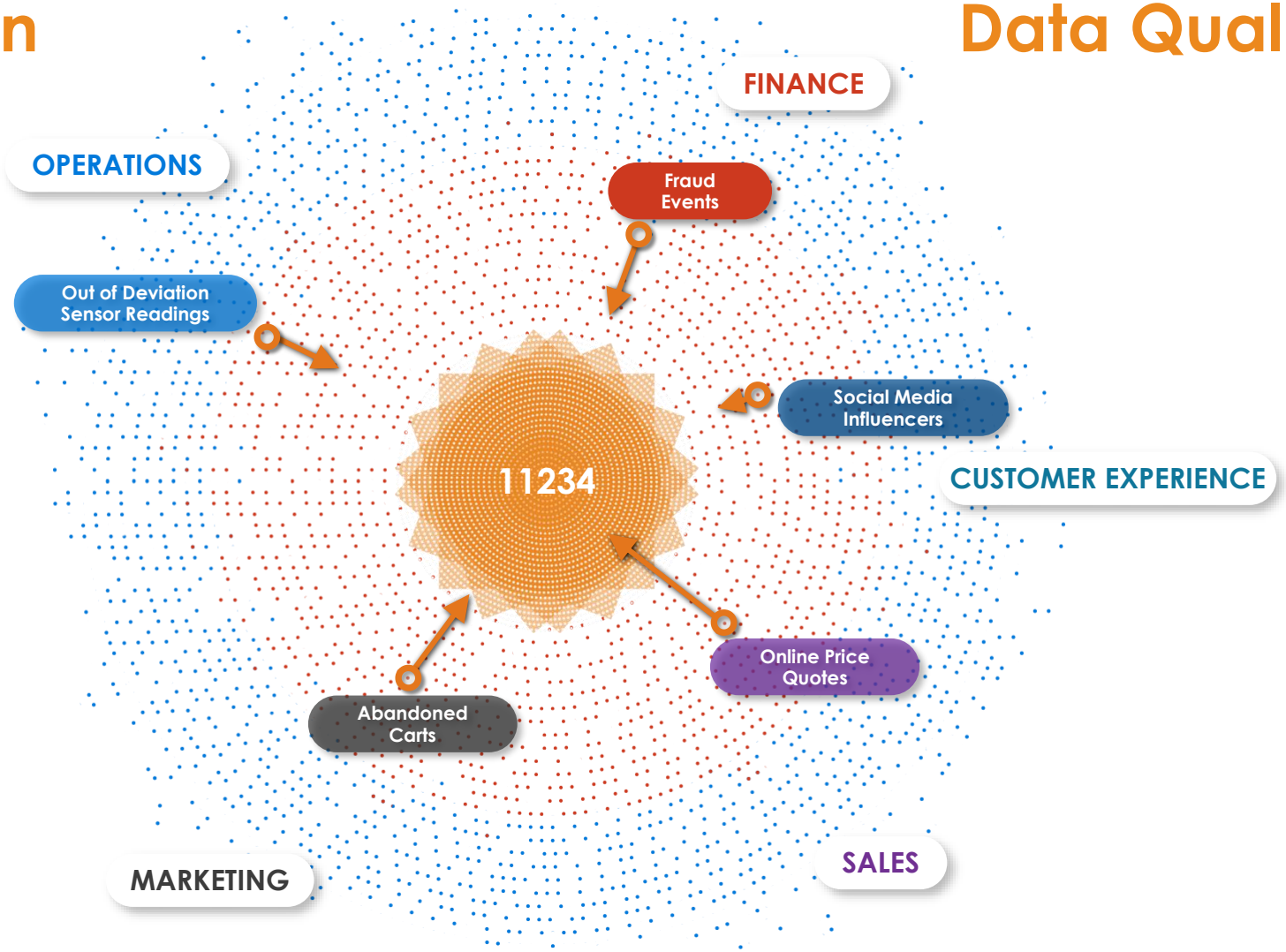

Schema on Read
New Data Sources
Evolving Schema
Evolving Data
Schema on Write
Enterprise Data

Curation Required

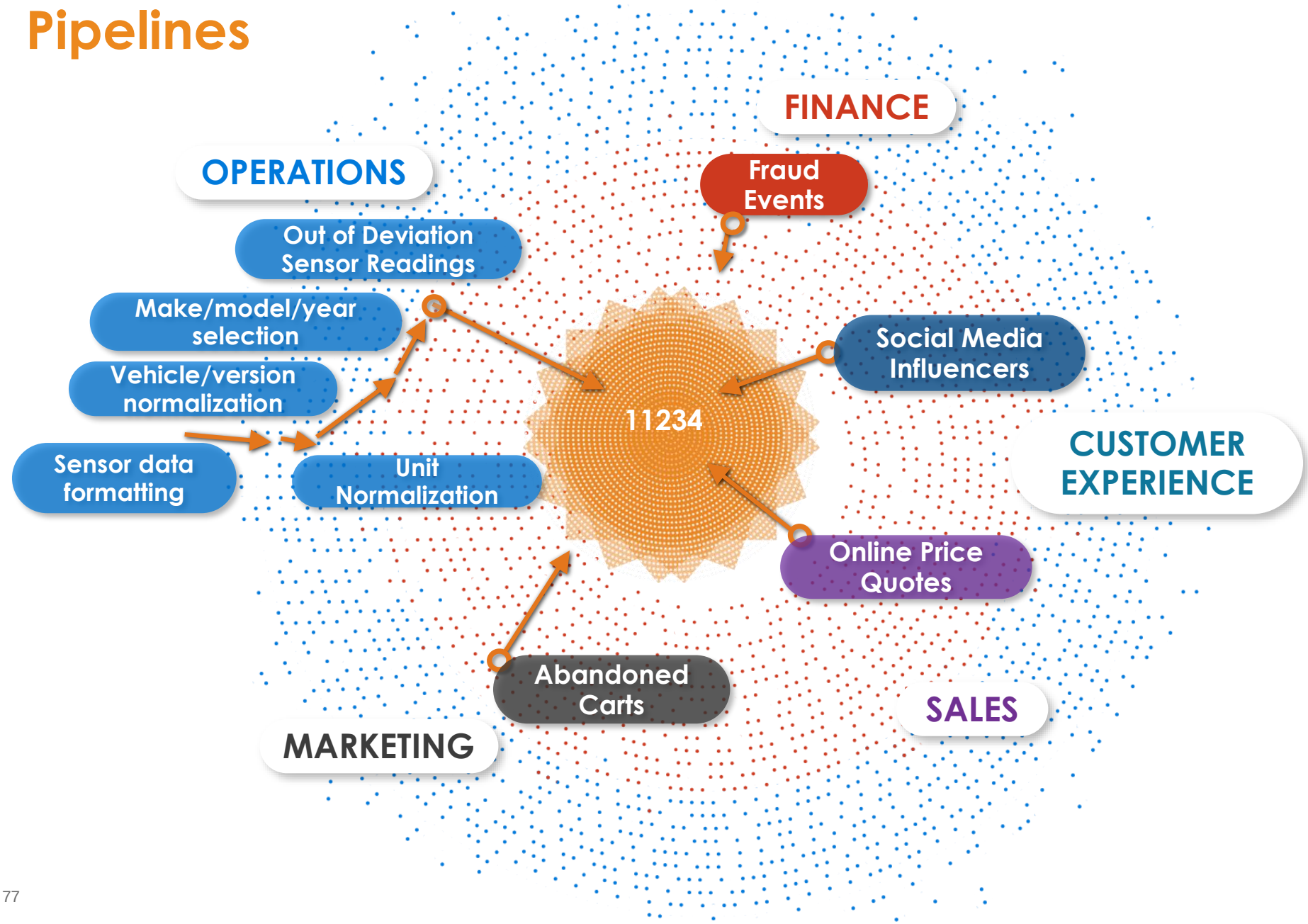


Minimum Viable Curation

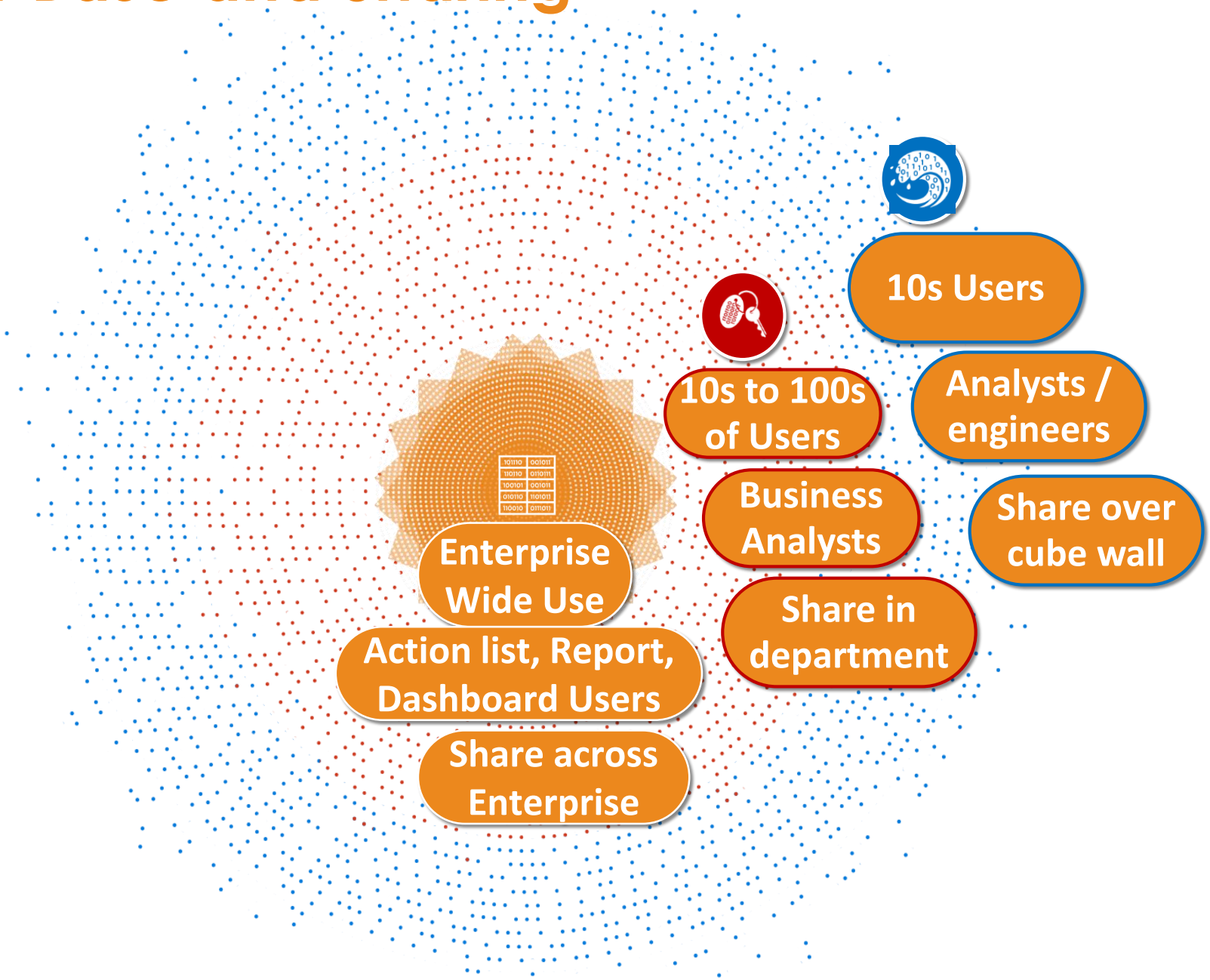
Minimum Viable Data Quality



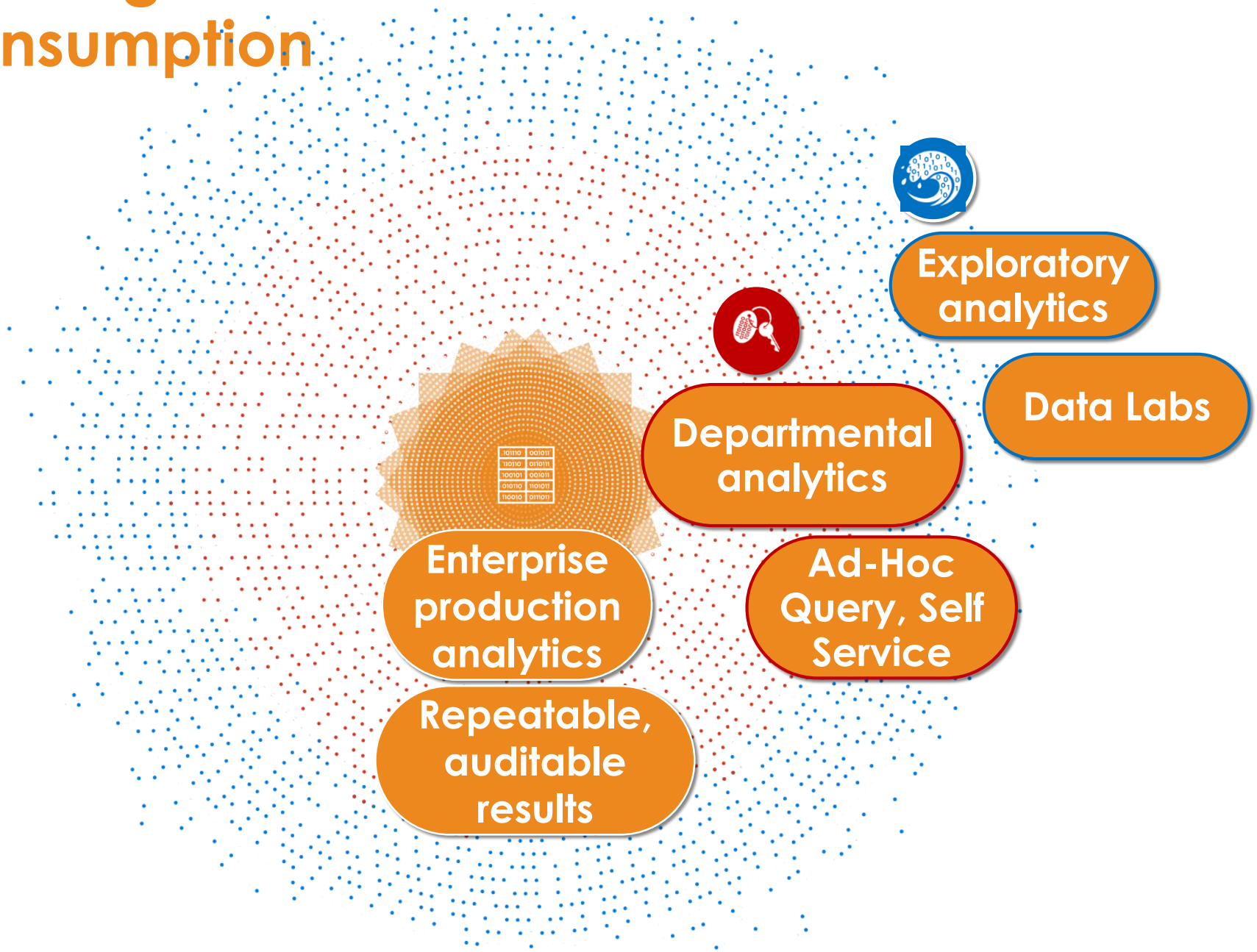
Data Comprehension, Pipelines



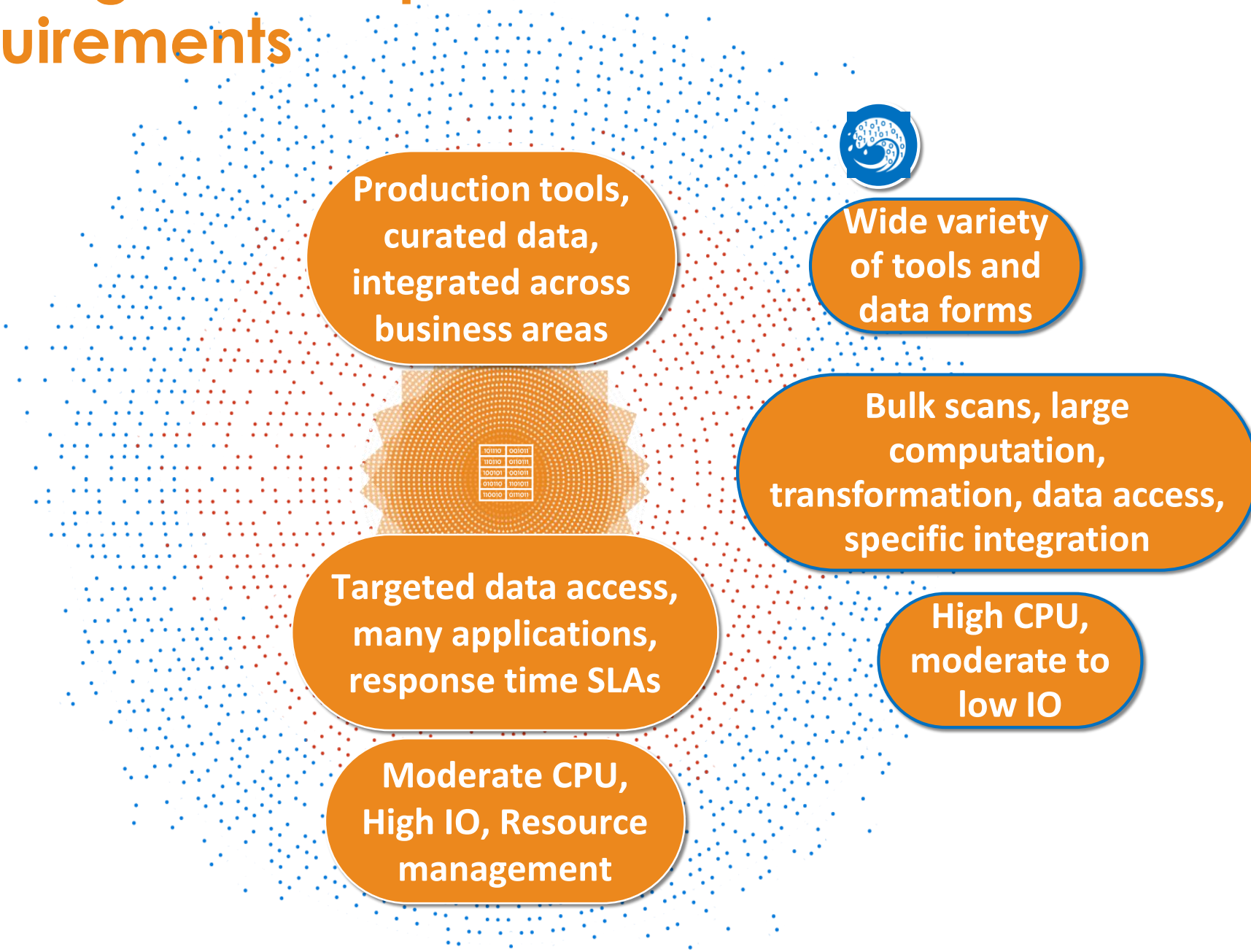
User Base and Sharing



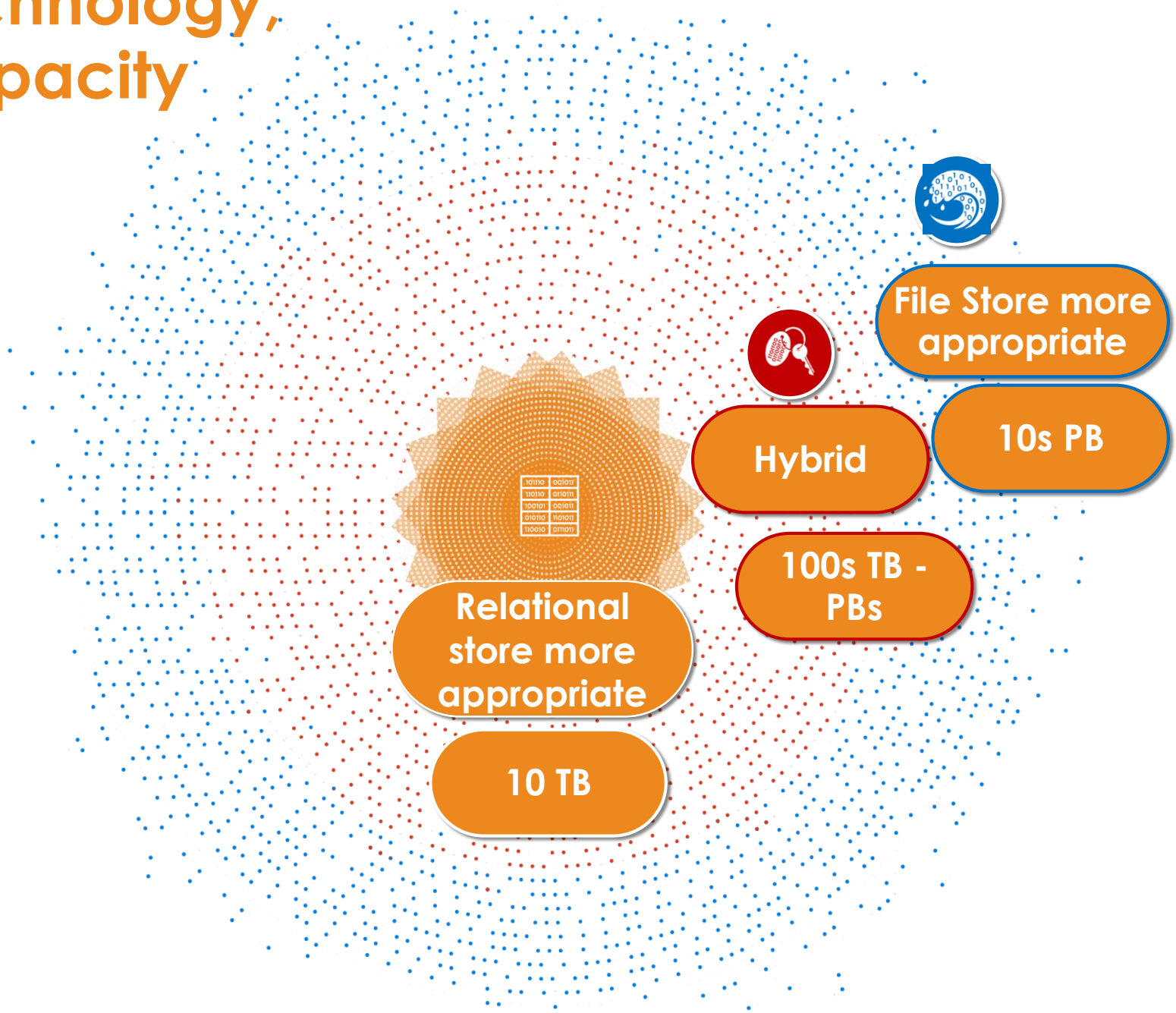
Evolving Consumption



Evolving Consumption Requirements



Technology, Capacity

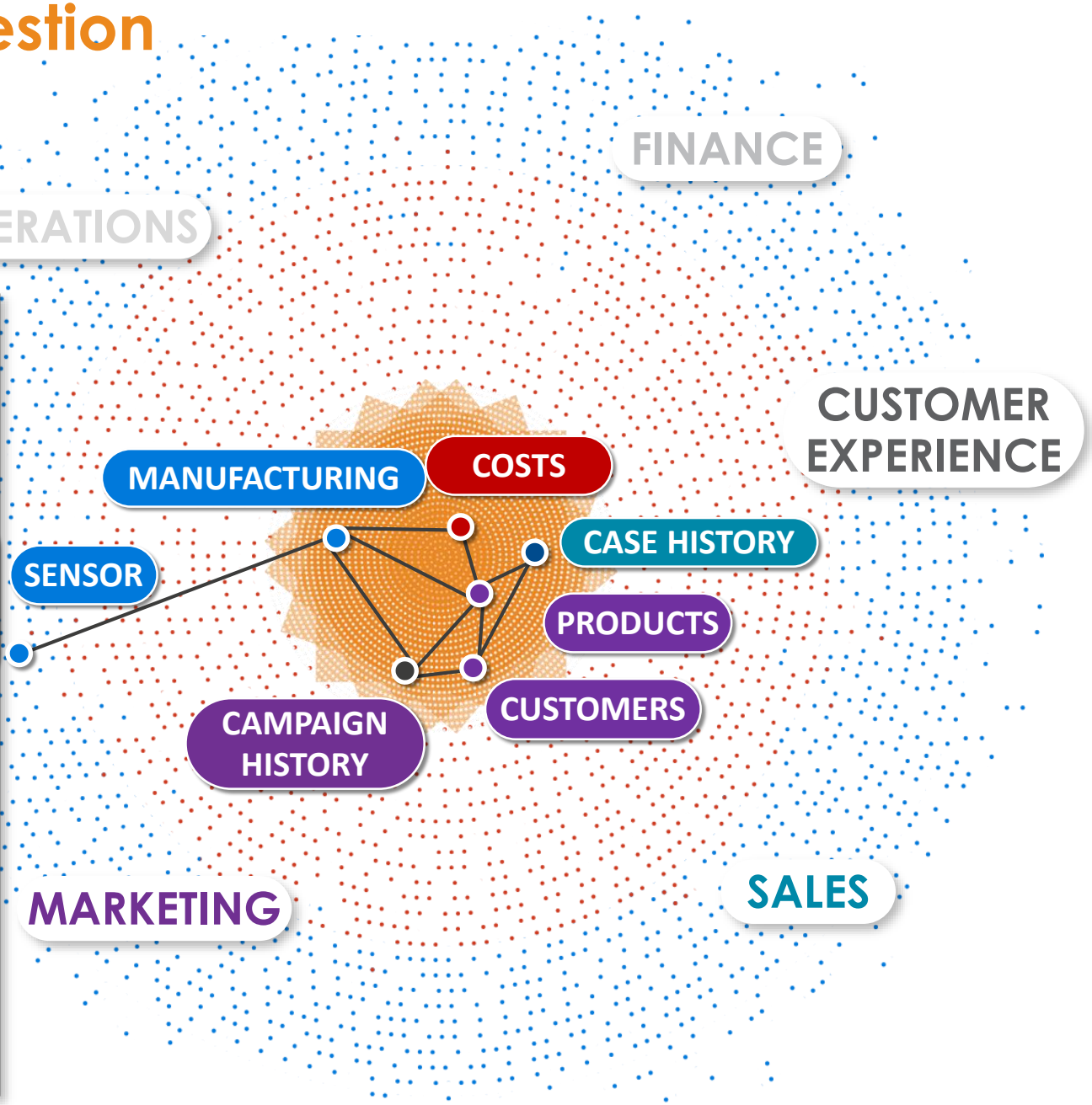


Access Wide Variety of Data to Answer a Question

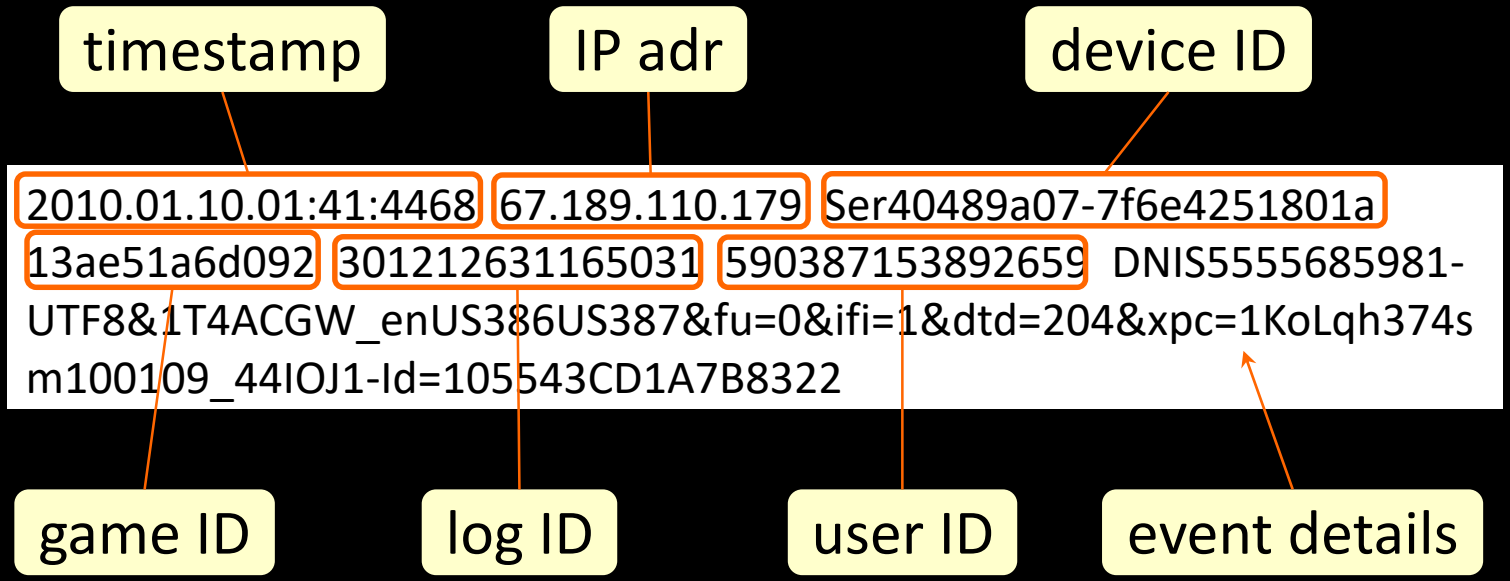
Given the rise in warranty costs, isolate the problem to be a plant and the specific lot.

Exclude 2/3rd of the batteries from the lot that are fine.

Communicate with affected customers, who have not made a warranty claim, through Marketing and Customer Service channels to recall cars with affected batteries.



An event contains mainly IDs…that reference other data



Log de-referencing and enrichment is difficult since you can't enforce integrity like you can in a DB.

What's the glue that holds it together?

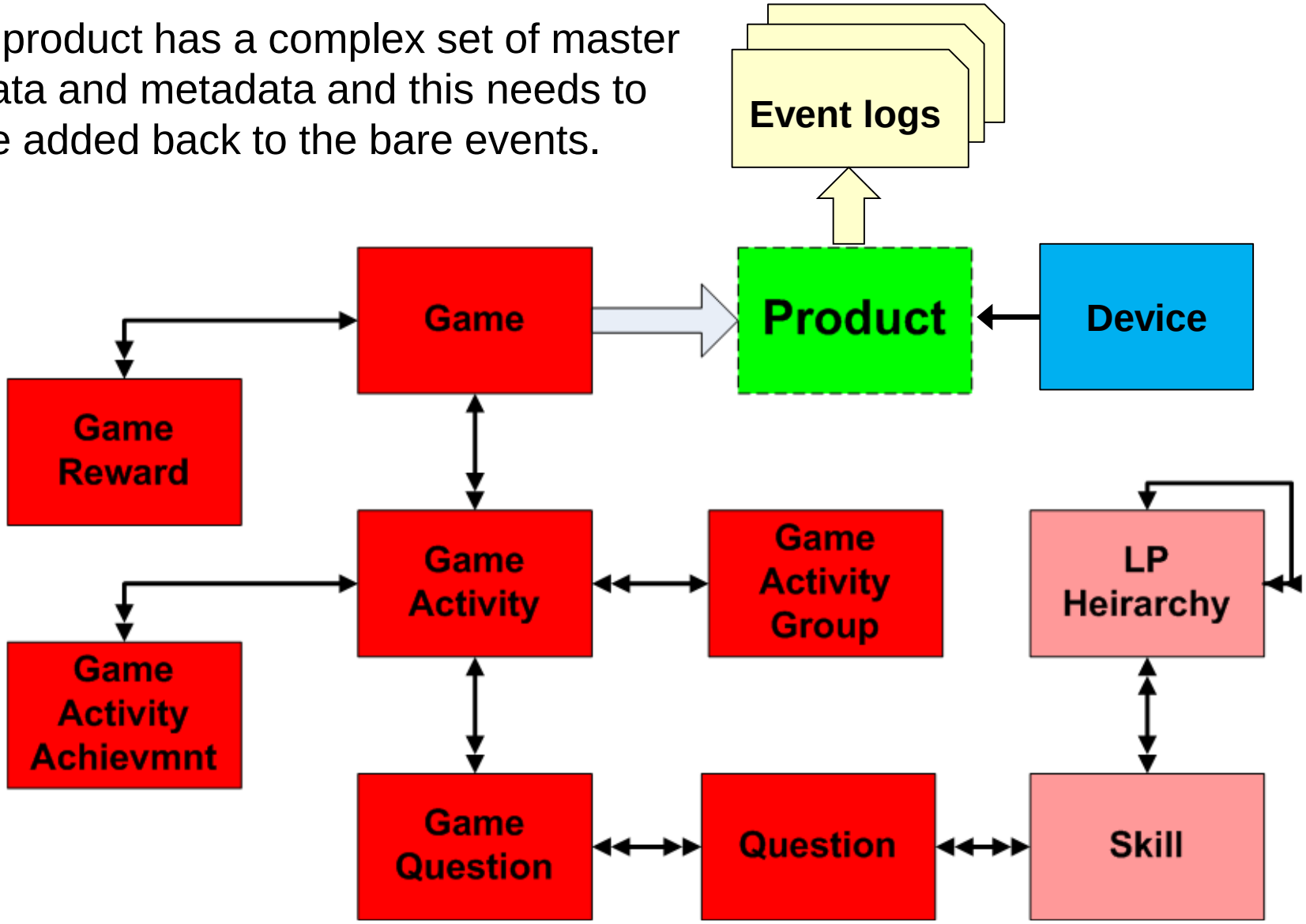
It's just keys to other data.

e.g. remember that device ID 0 problem?

Third Nature

It’s not just the log data, it’s big data + small data

A product has a complex set of master data and metadata and this needs to be added back to the bare events.



Third Nature

Where does the reference data come from?

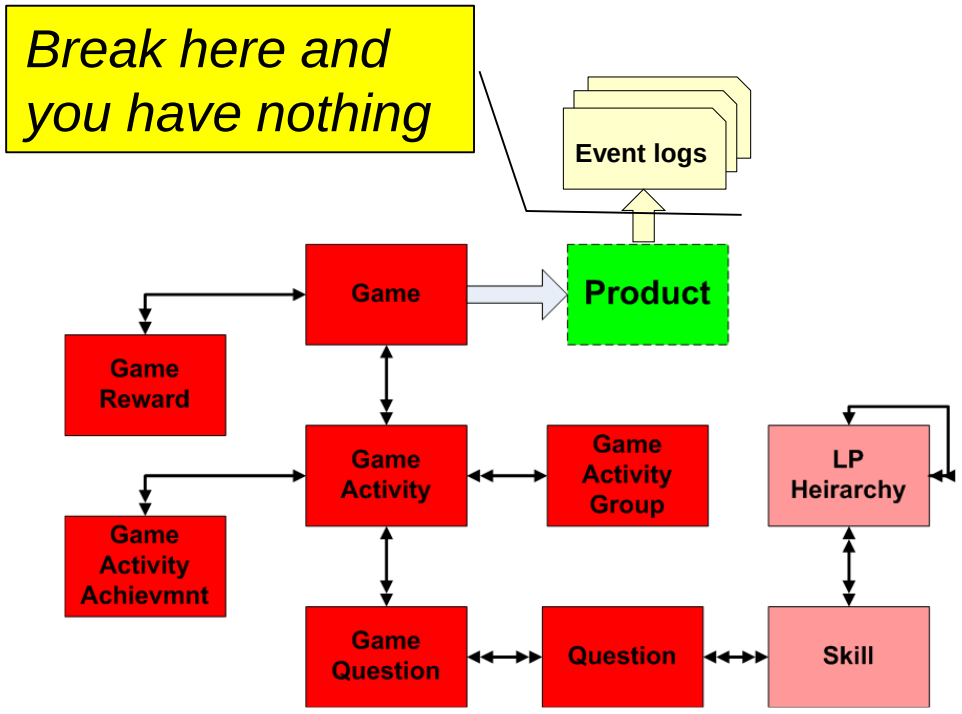
The keys come from somewhere. You don't just make up system-wide unique identifiers in your code.

The lack of local lookup data at event generation leads to development practices that lead to inconsistencies.

Problem we had: product identifiers that didn't match any known products

Had to fix by analyzing each log of bad-ID events.

Because developers used config files they copied from the PMS and put in the code

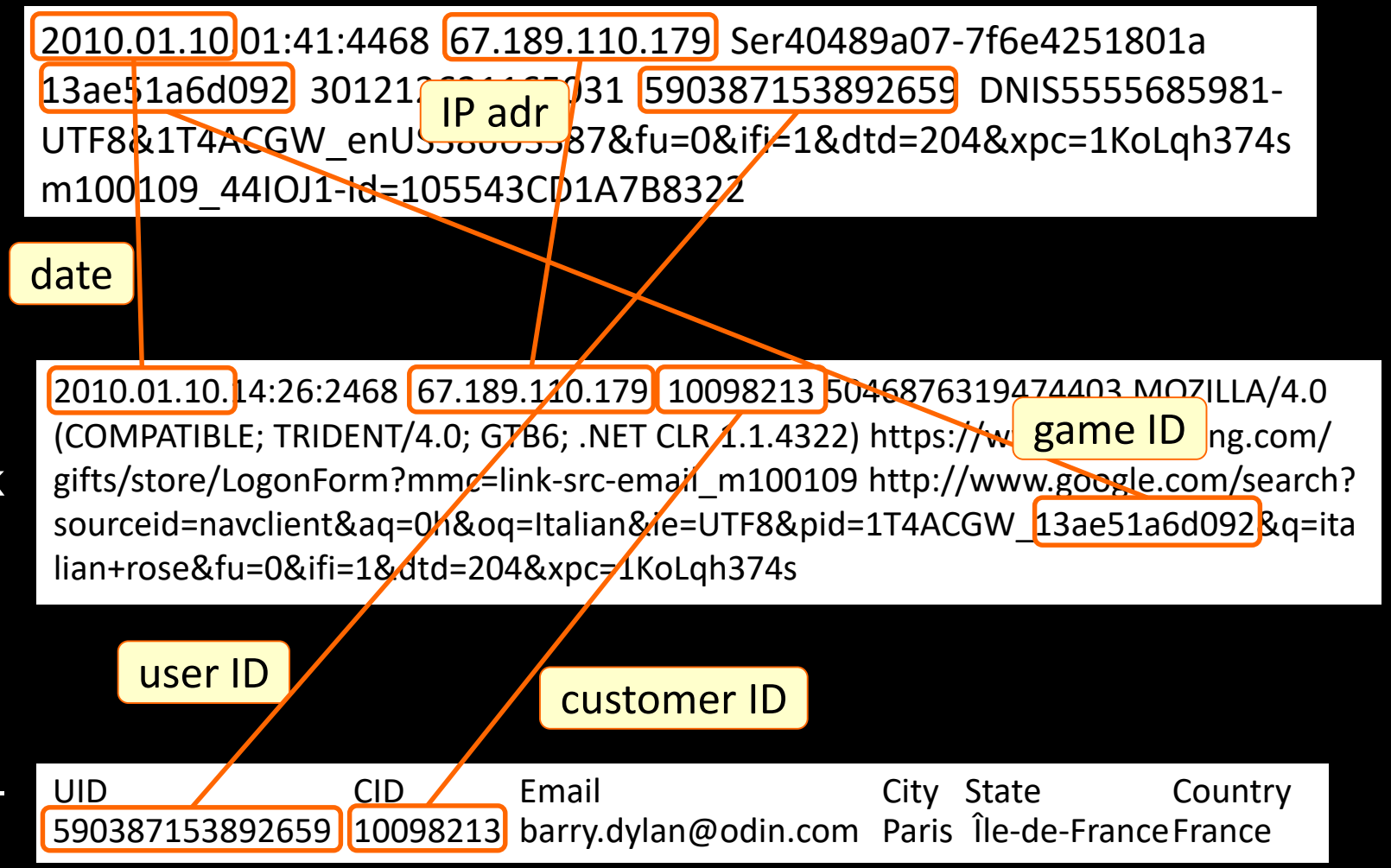


Third Nature

MDM again

If you want to link datasets then you must manage the keys

You need canonical forms for common data (in code too)



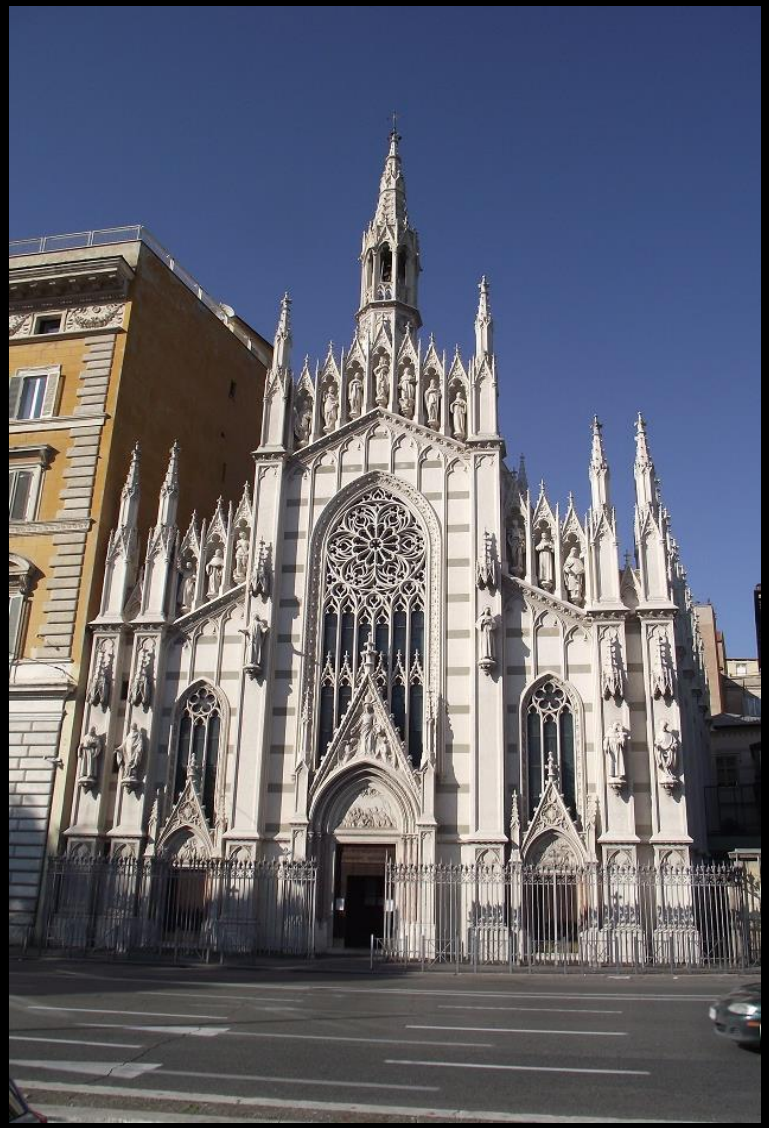
Third Nature

Data hoarding is not a data management strategy

The missing ingredient from most big data

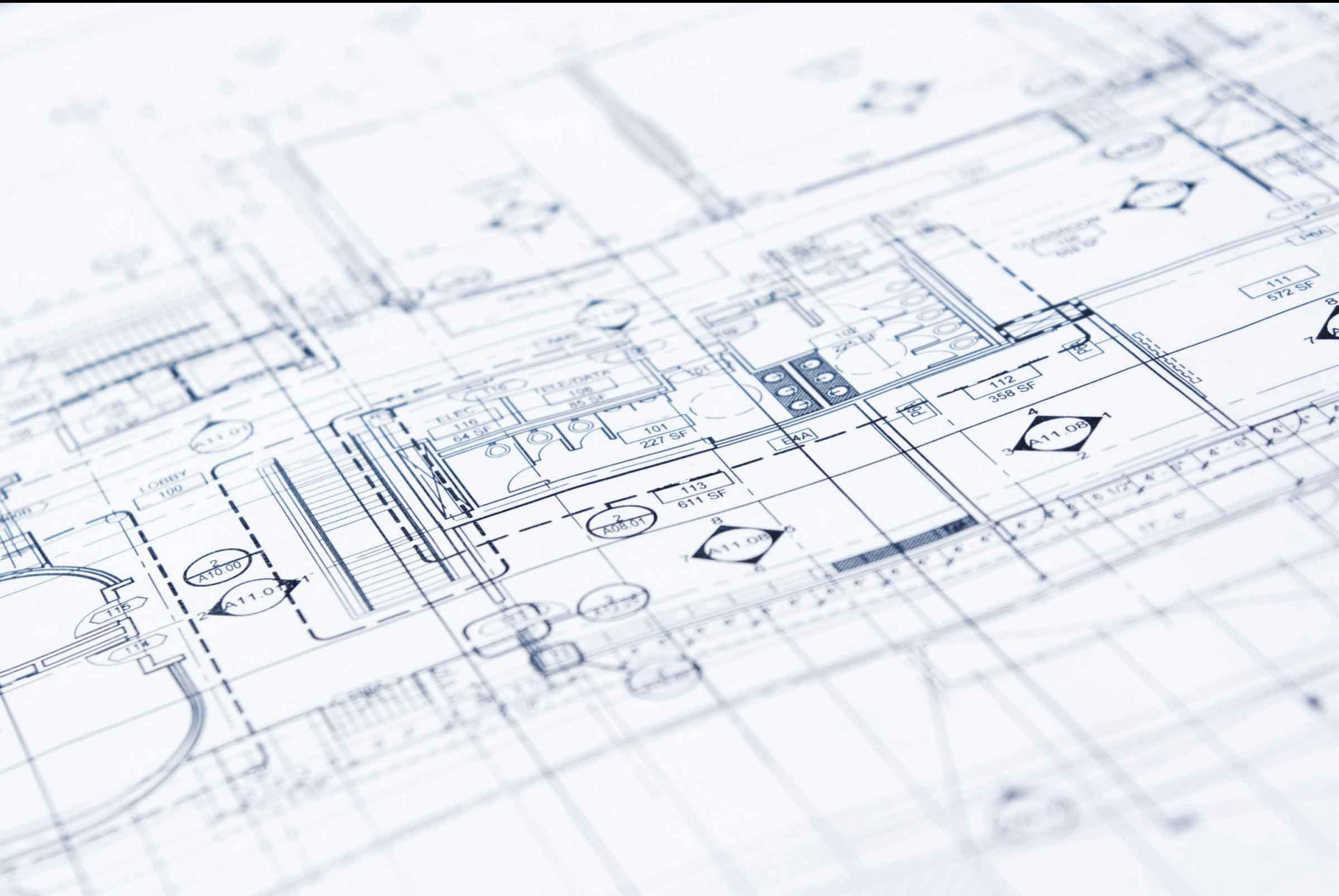
Third Nature

Architecture is an abstraction – it's a purpose



Third Nature

Blueprints are not architectures

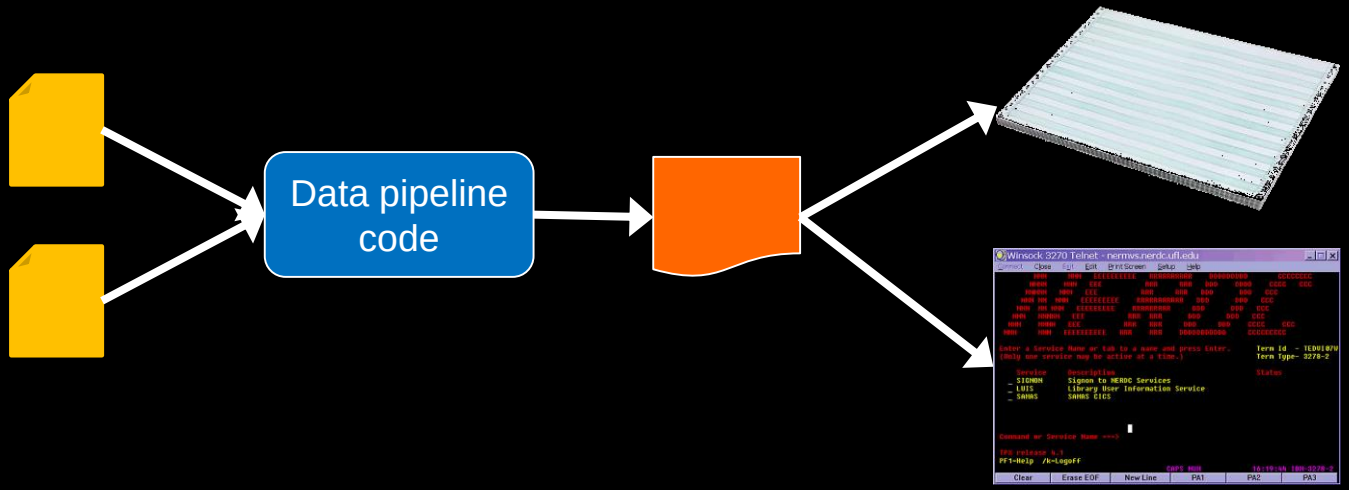


History: This is how BI was done through the 80s

First there were files and reporting programs.

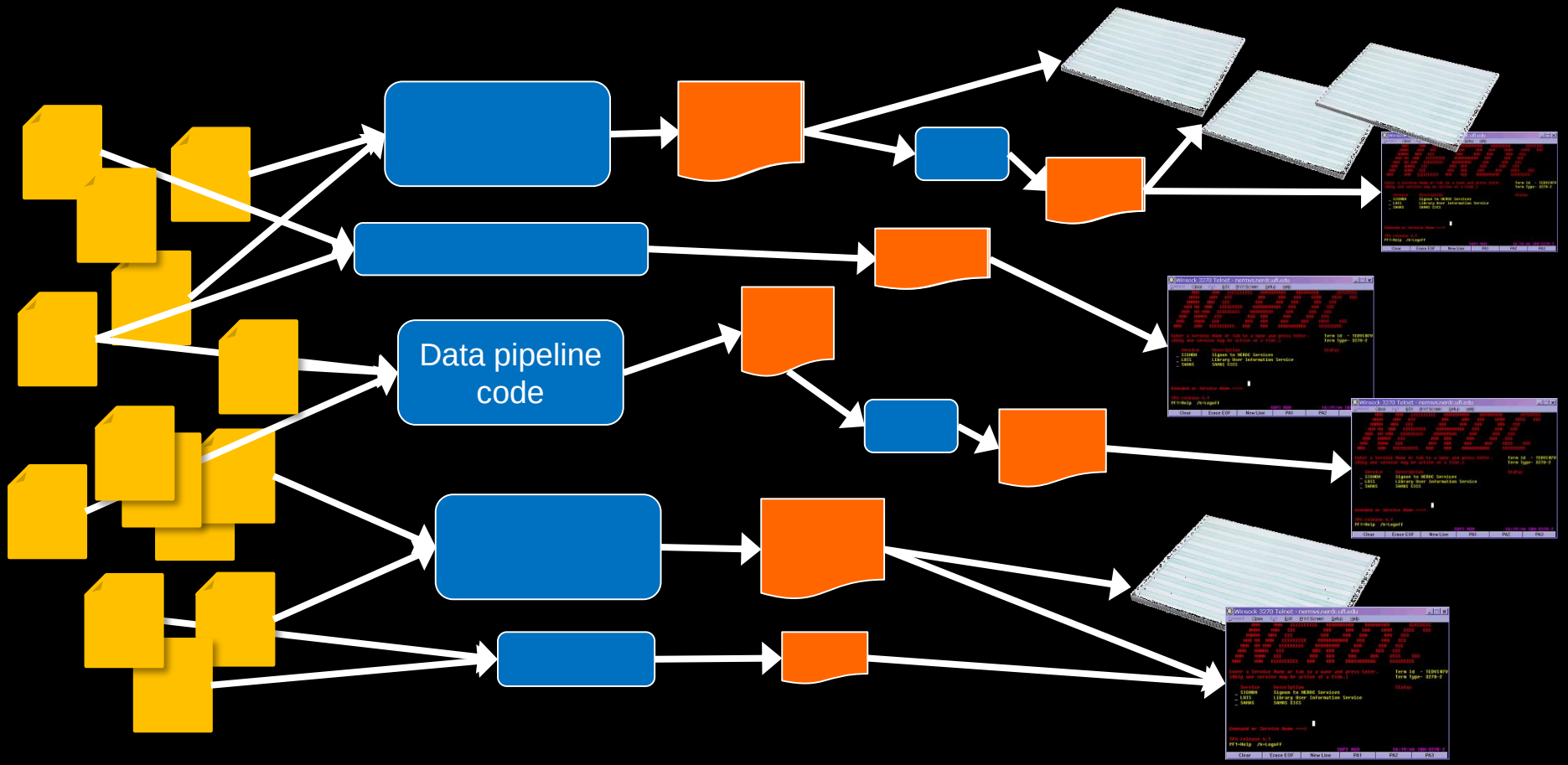
Application files feed through a data processing pipeline to generate an output file. The file is used by a report formatter for print/screen. Files are largely single-purpose use.

Every report is a program written by a developer.



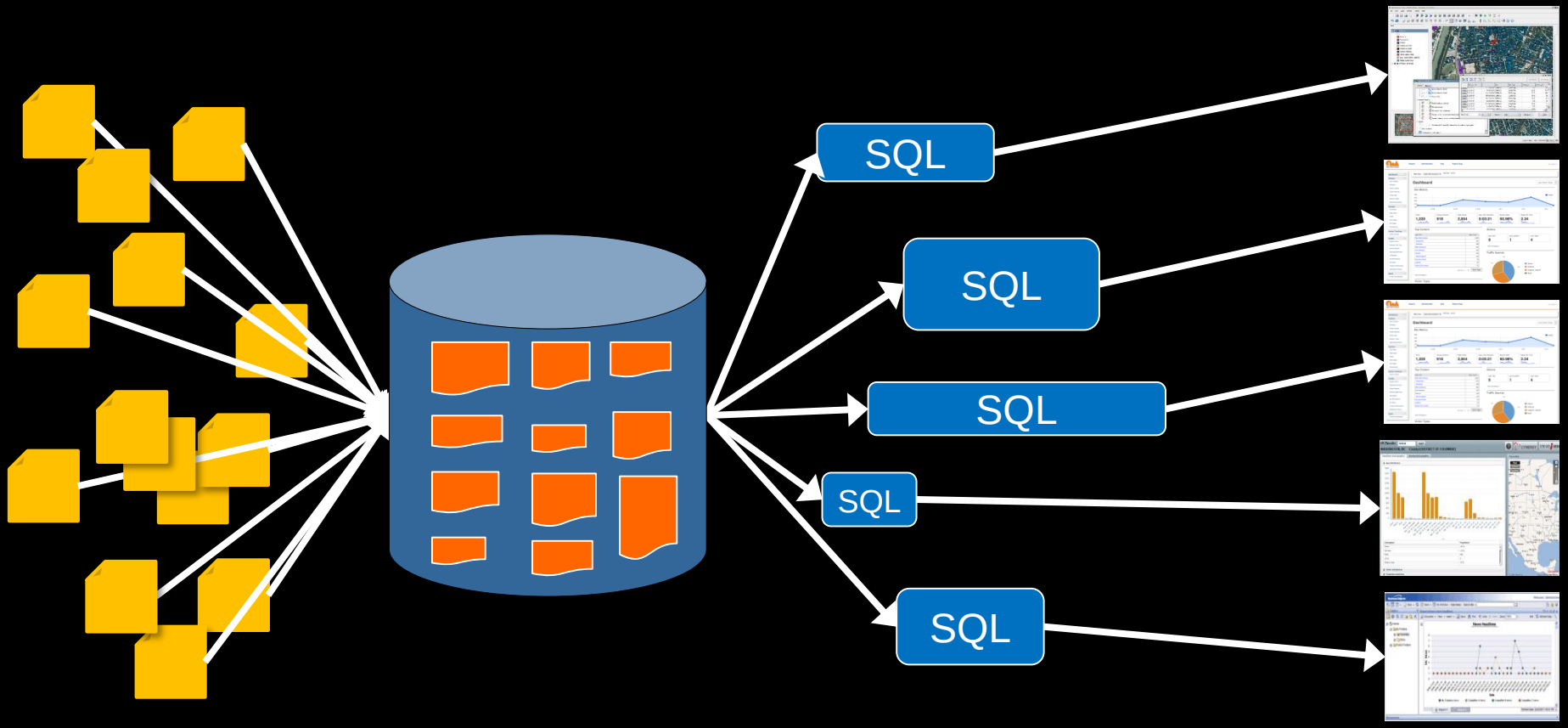
History: This is how BI ended the 80s

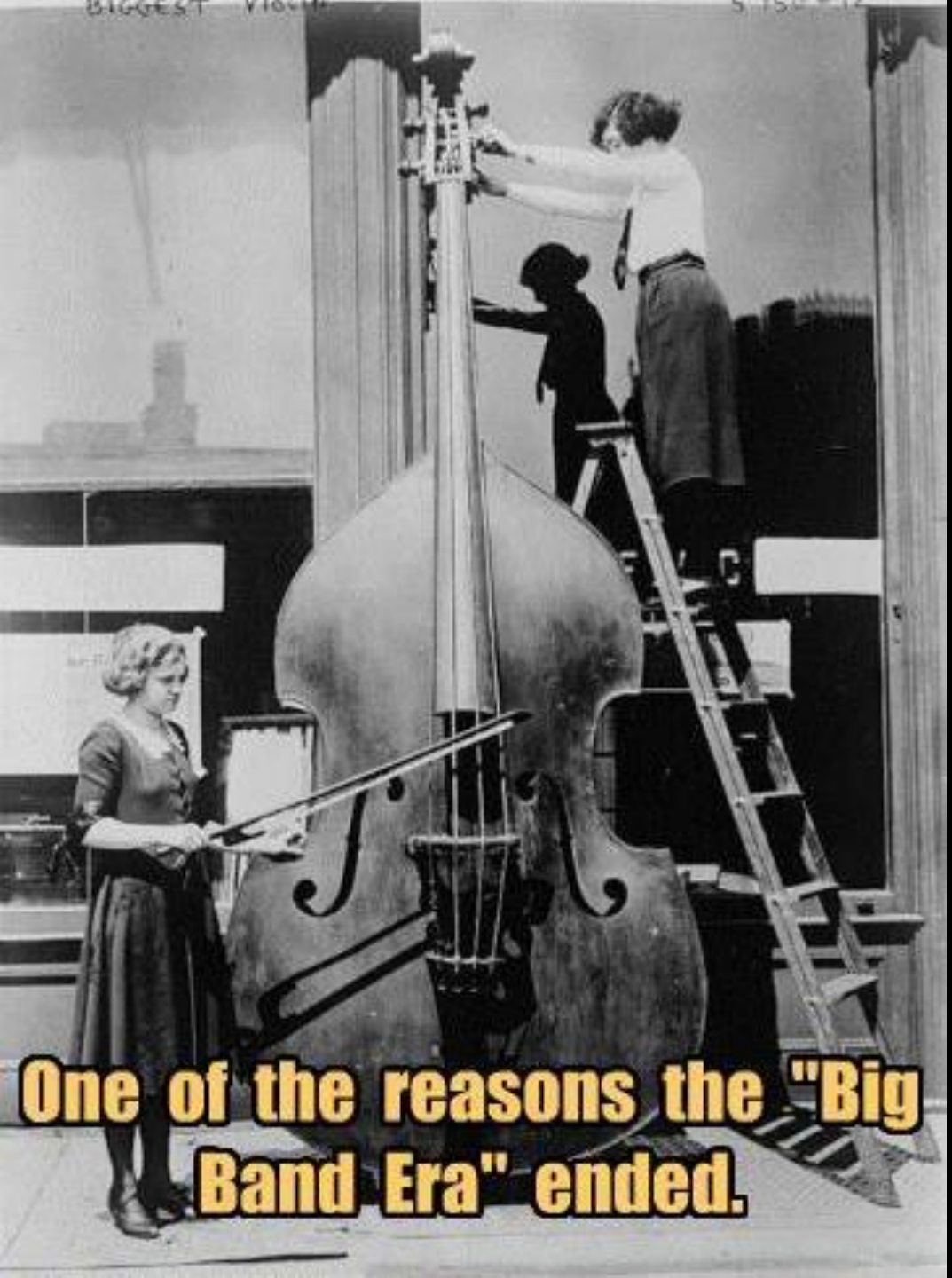
The inevitable situation was...



History: This is how we started the 90s

Collect data in a database. Queries replaced a *LOT* of application code because much was just joins. We learned about “dead code”





Pragmatism and Data

Lessons learned during the ad-hoc SQL era of the DW market:

- When the technology is awkward for the users, the users will stop trying to use it.
- Even “simple” schemas weren’t enough for anyone other than analysts and their Brio…

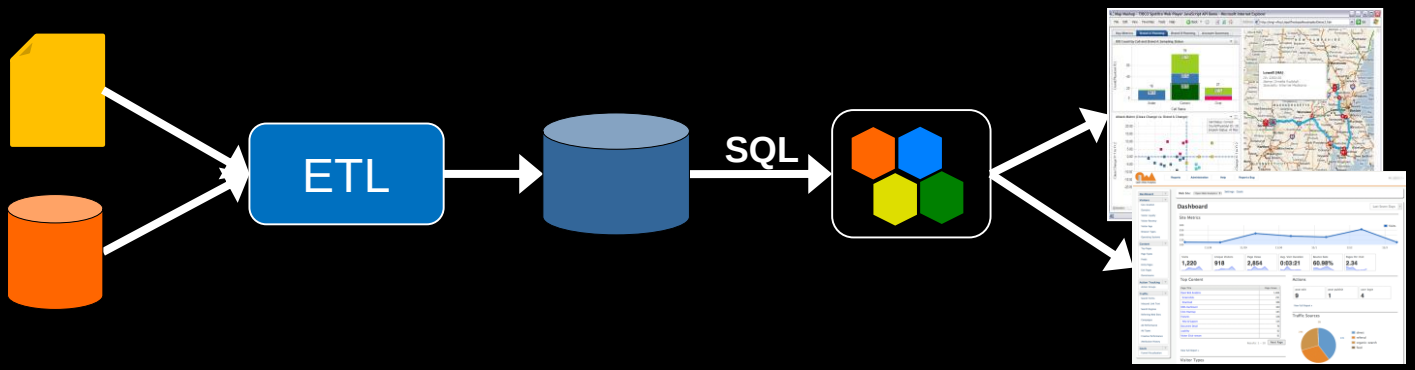
Led to the evolution of metadata-driven SQL-generating BI tools, ETL tools.

BI evolved to hiding query generation for end users

With more regular schema models, in particular dimensional models that didn't contain cyclic join paths, it was possible to automate SQL generation via semantic mapping layers.

We developed data pipeline building tools (ETL).

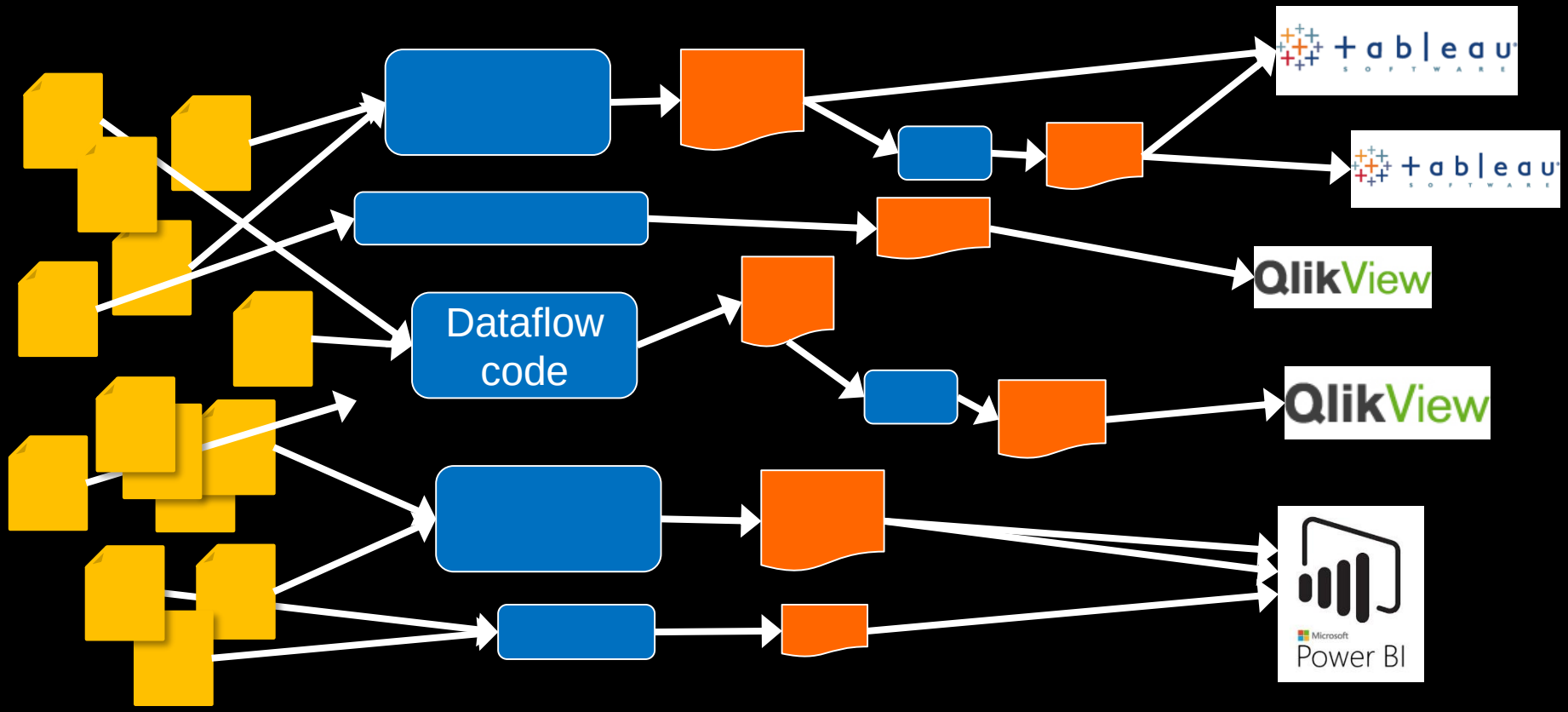
Query via business terms made BI usable by non-technical people.



Life got much easier…for a while

Today's model: Lake + data engineers, looks familiar…

The Lake with data pipelines to files or Hive tables
is exactly the same pattern as the COBOL batch..



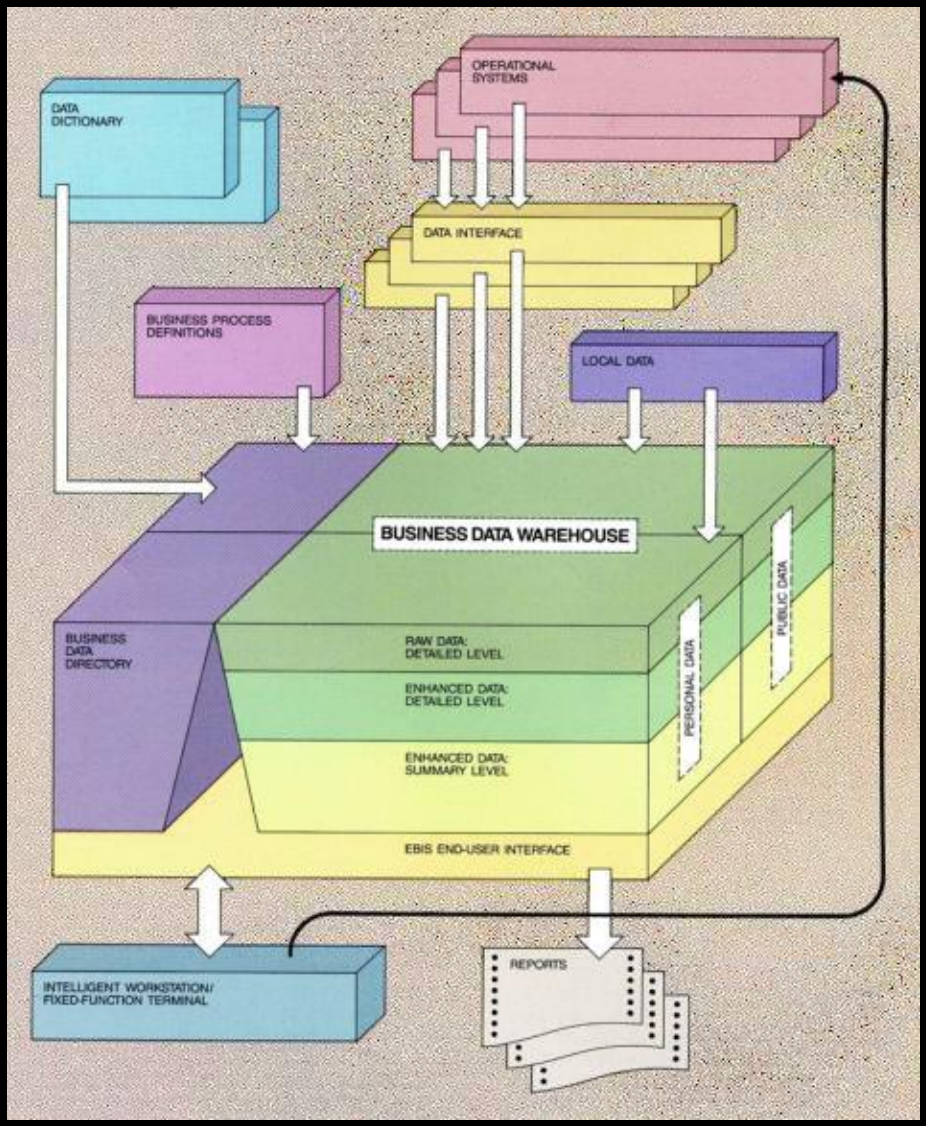
We already know that people don't scale…

We’re so focused on the light switch that we’re not talking about the light

DATA ARCHITECTURE

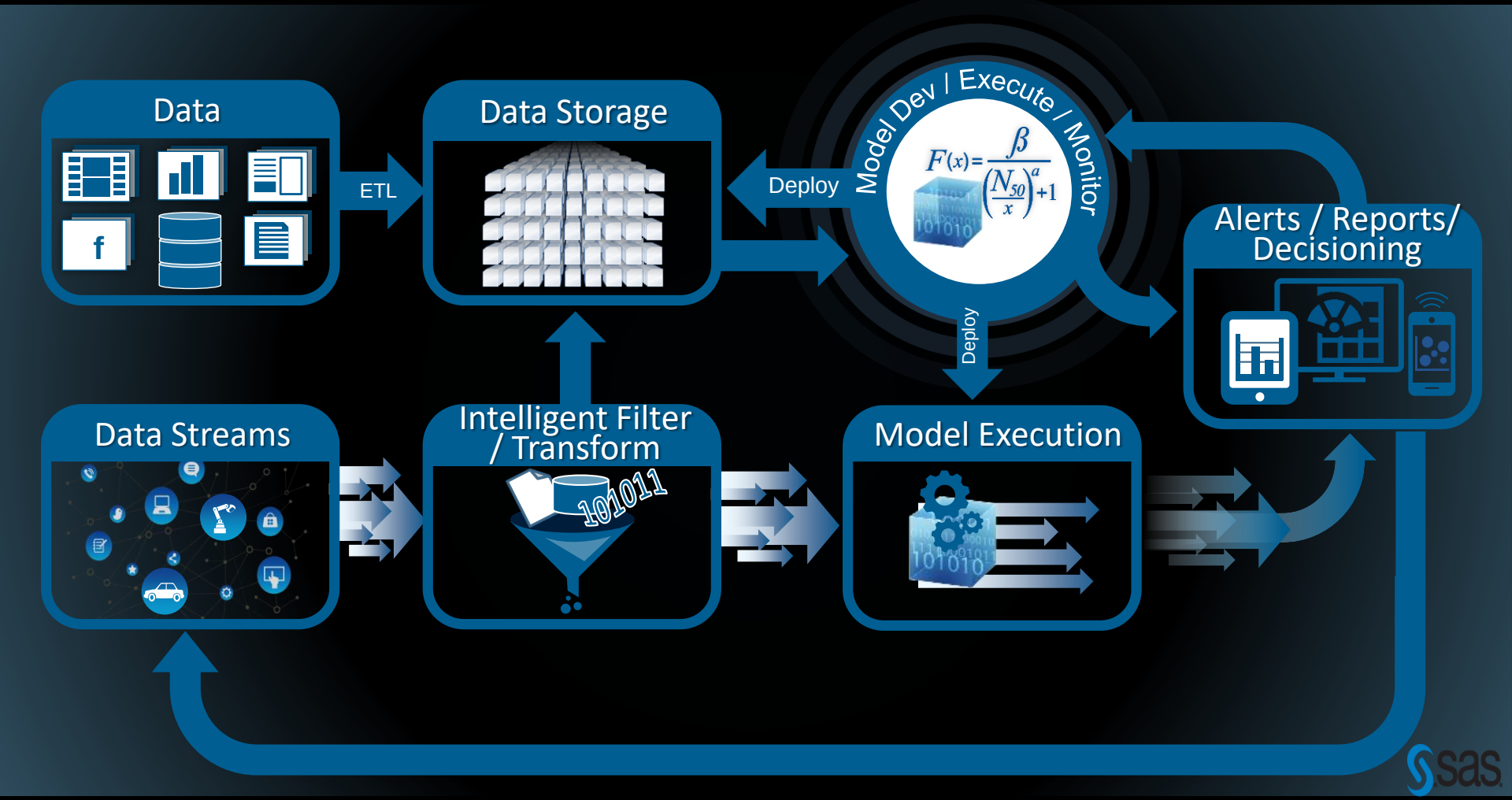
The architecture from 1988 we SHOULD HAVE BEEN USING

The general concept of a separate architecture for BI has been around longer, but this paper by Devlin and Murphy is the first formal data warehouse architecture and definition published.



"An architecture for a business and information system", B. A. Devlin, P. T. Murphy, IBM Systems Journal, Vol.27, No. 1, (1988)

But 30 years ago we did not expect so many different models of deployment, execution and use. Needs change



Analytics for eyeballs and analytics for machines are different

Decouple the Architecture

The core of a data warehouse isn't the database, it's the data architecture that the database and tools implement.

We need a new data architecture that is not limiting:

- Deals with change more easily and at scale
- Does not enforce requirements and models up front
- Does not limit the format or structure of data
- Assumes the range of data latencies in and out, from streaming to one-time bulk
- Allows both reading and writing of data
- Makes data linkable, and provide governance where required
- *Does not give up the gains of the last 25 years*

Complexity requires a shift: not buildings, the block

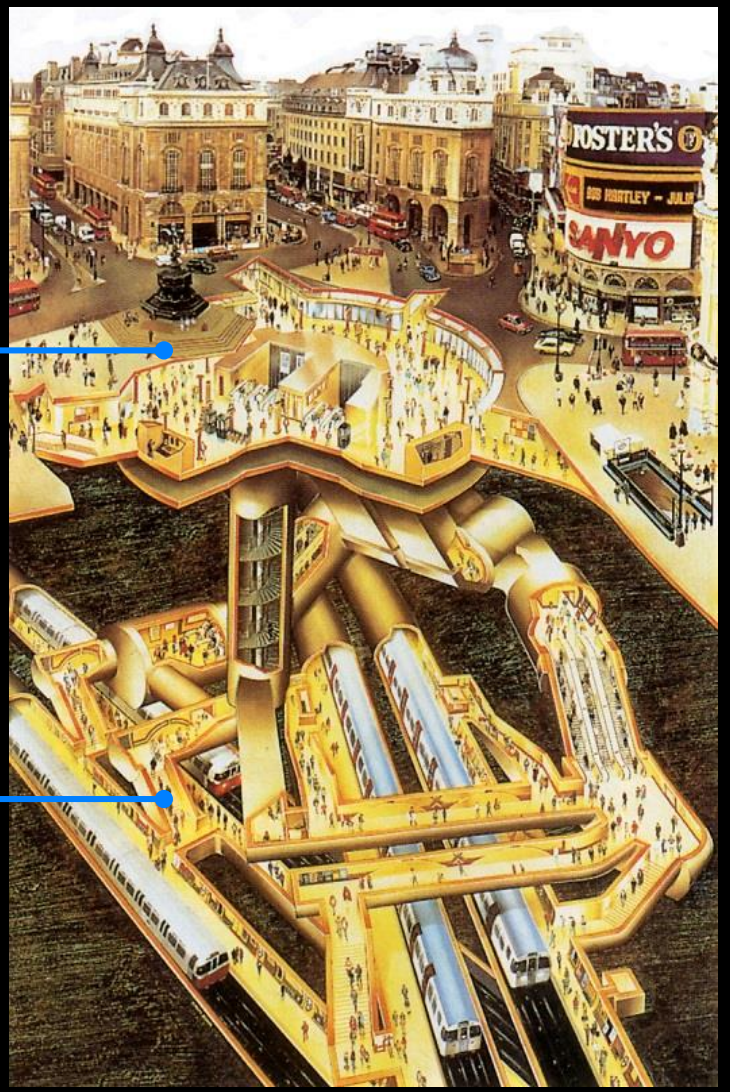
Components above: flexibility, repurposing, quicker change above

Application

Layers below: stability, reuse, slow predictable change below

Infrastructure

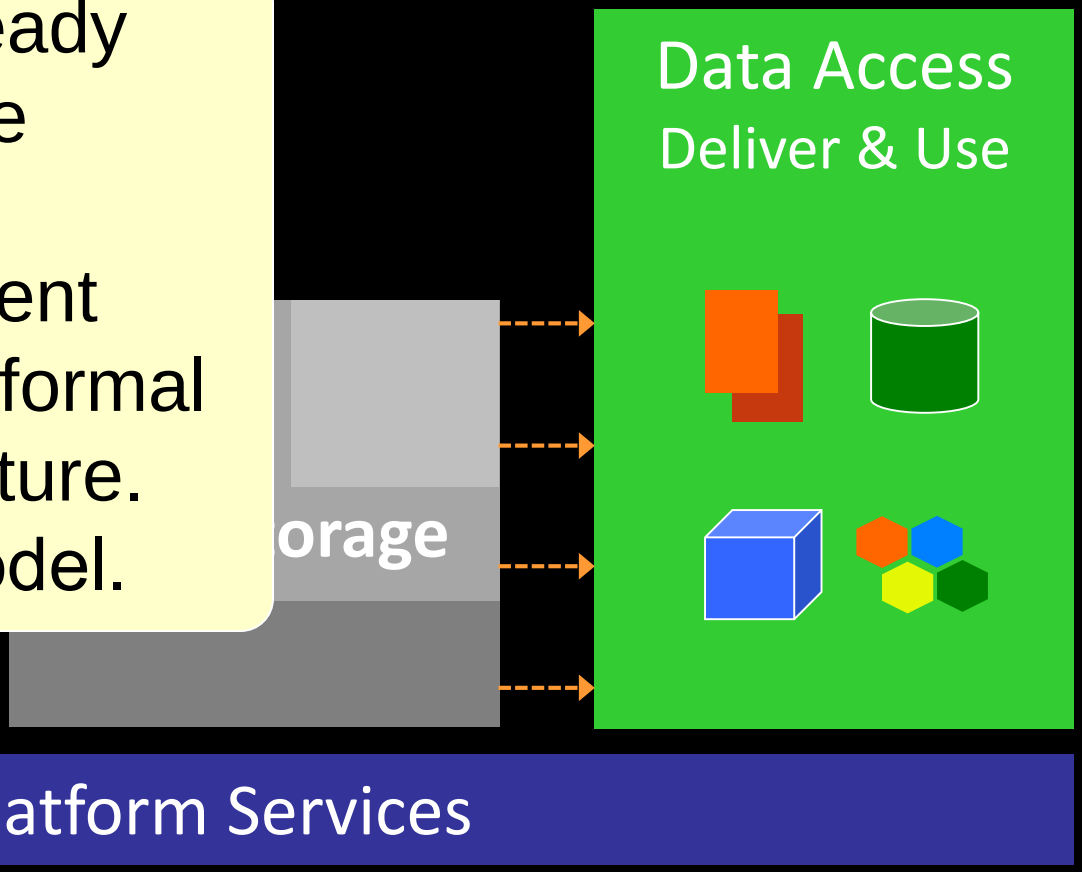
We thought this was the DW…



Infrastructure is just a layer carefully chosen after a lot of experience

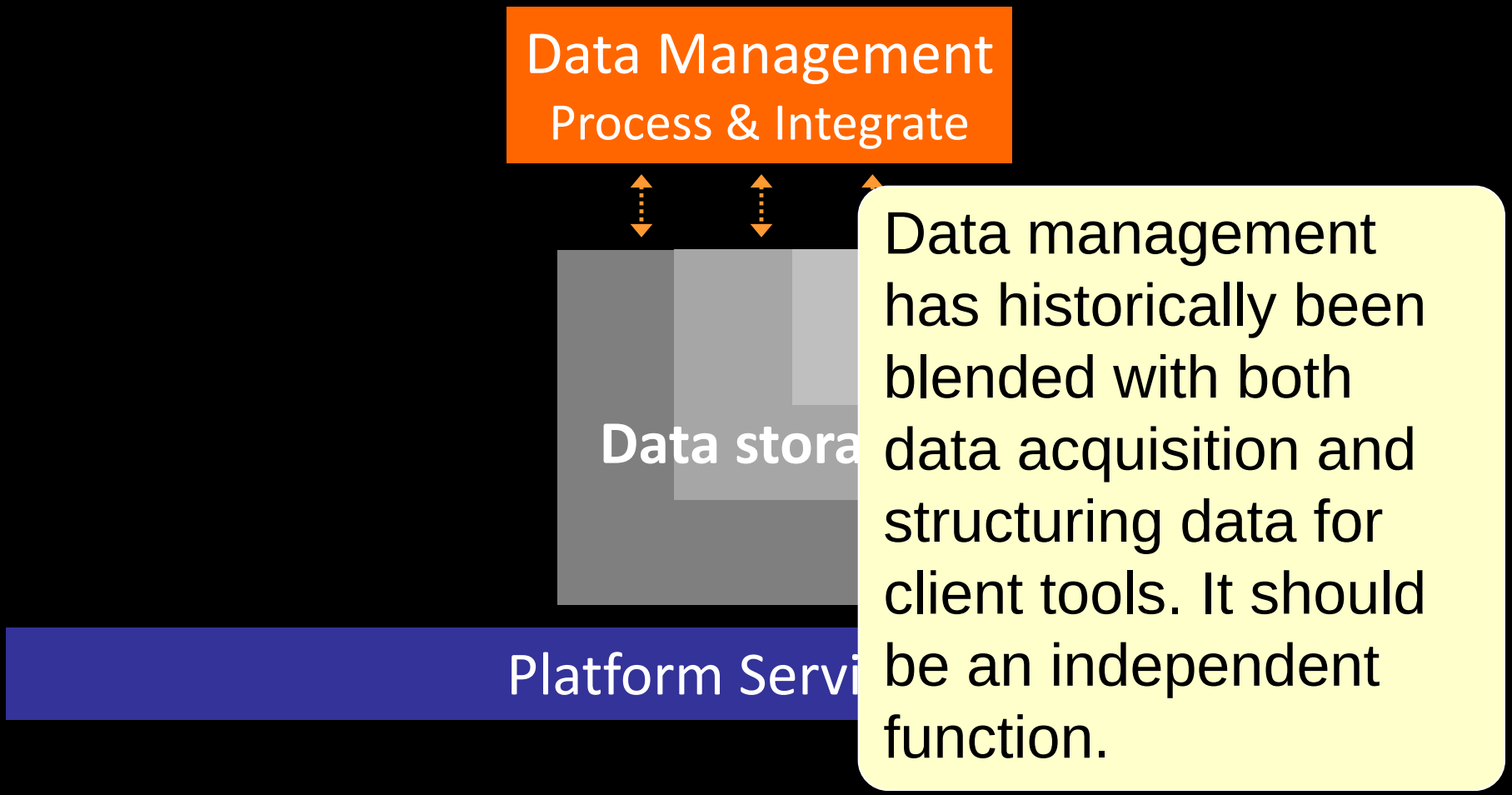
The goal is to decouple: solve the application and infrastructure problems separately, independently

Data access is already somewhat separate today. Make the separation of different access methods a formal part of the architecture. Don't force one model.



This separates uses of data from each other, allowing each type of use to structure the data specific to its own requirements.

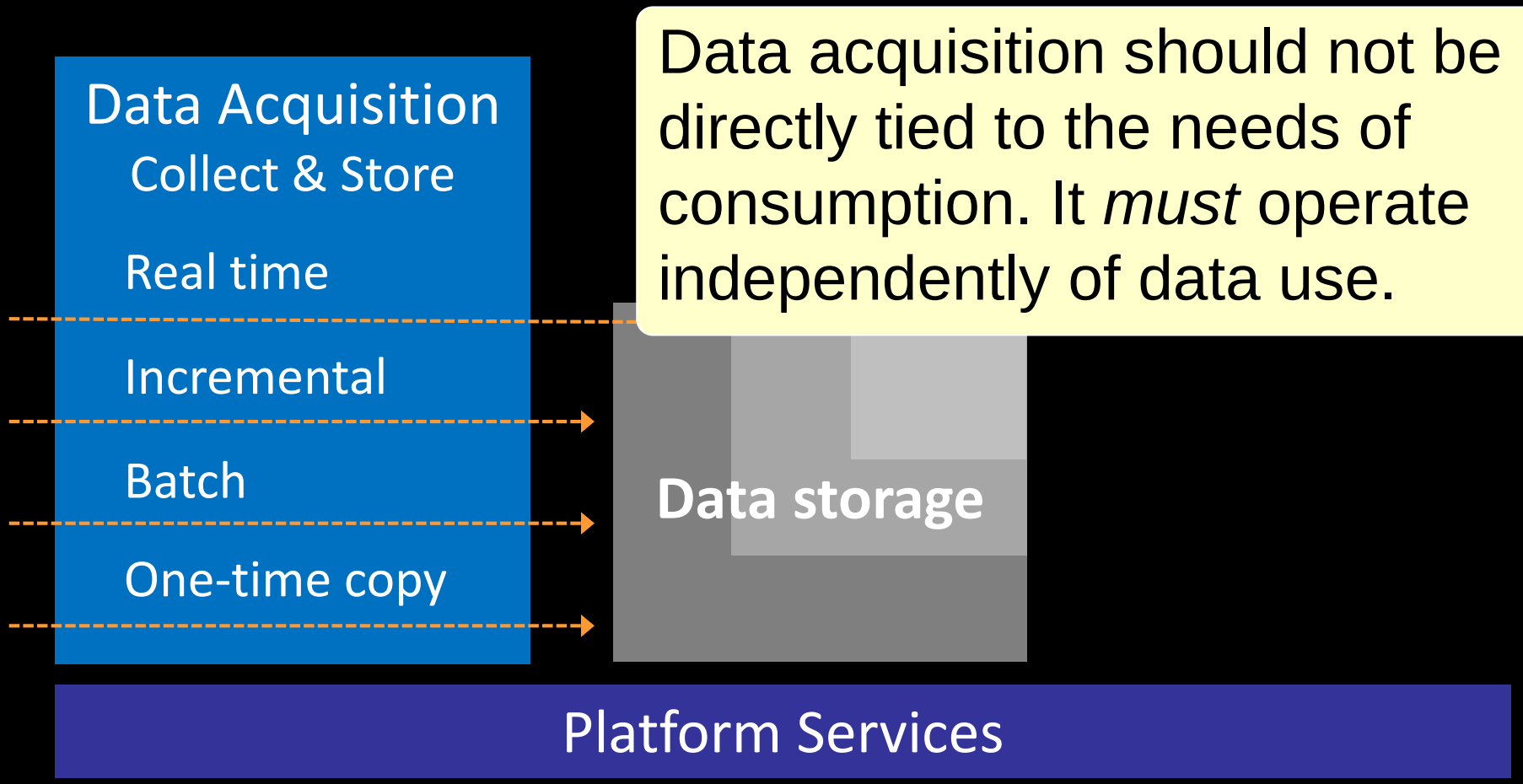
The goal is to decouple: solve the application and infrastructure problems separately, independently



Data management has historically been blended with both data acquisition and structuring data for client tools. It should be an independent function.

Data management should not be subject to the constraints of a single use

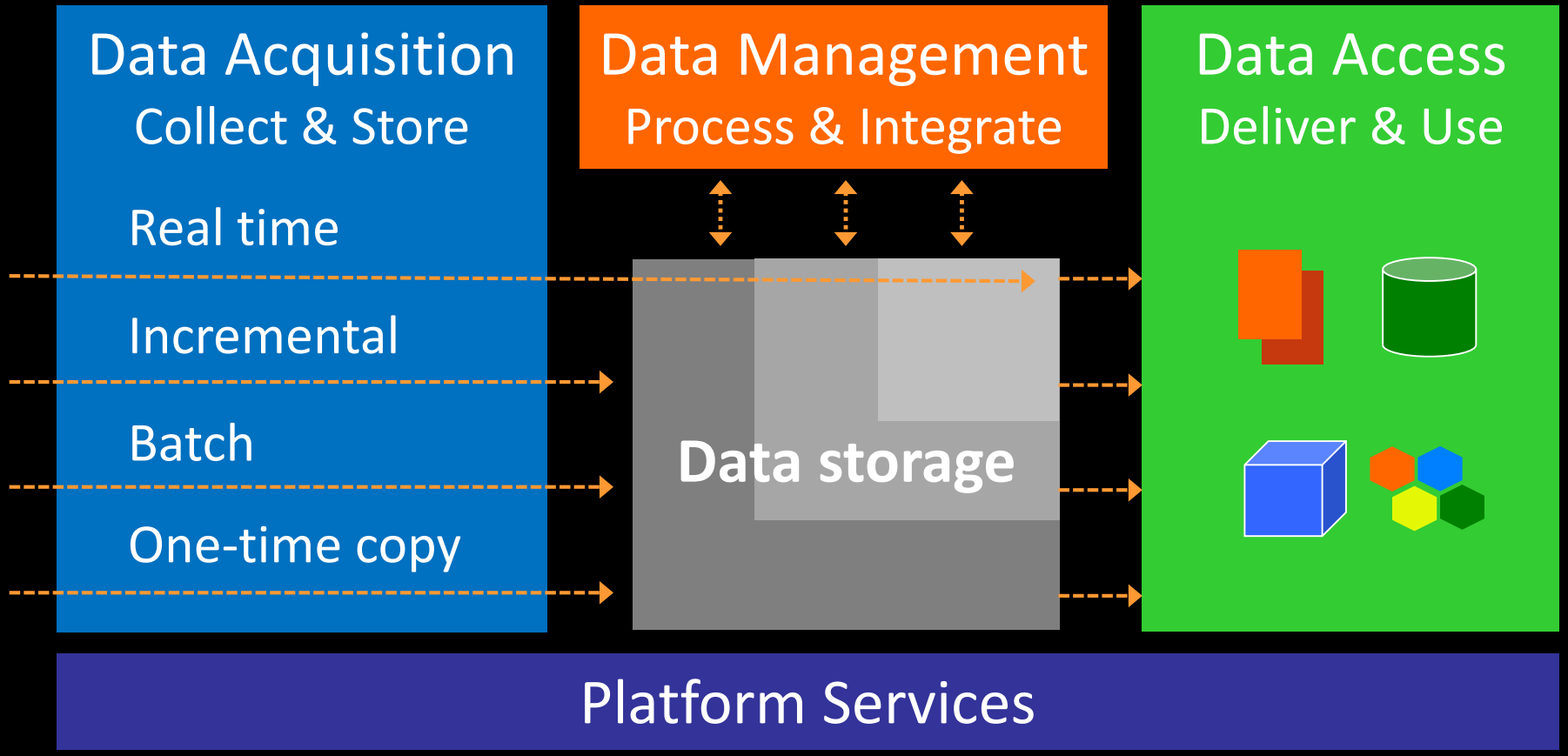
The goal is to decouple: solve the application and infrastructure problems separately, independently



Data acquisition should not be directly tied to the needs of consumption. It *must* operate independently of data use.

Data arrives in many latencies, from real-time to one-time. Acquisition can't be limited by the management or consumption layers.

The full analytic environment subsumes all the functions of a data lake and a data warehouse, and extends them



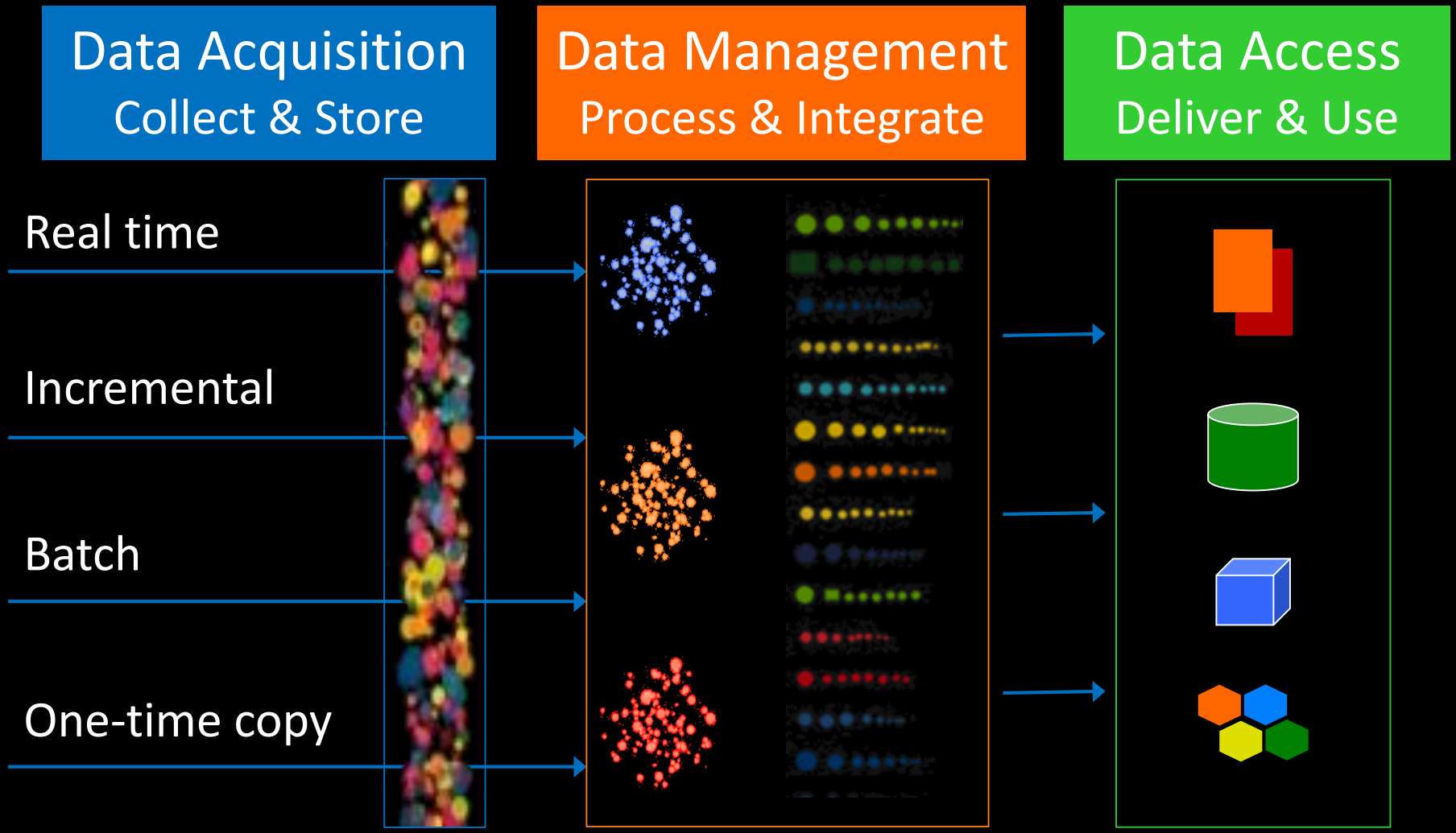
The platform has to do more than serve queries; it has to be read-write.

Food supply chain: an analogy for analytic data

Multiple contexts of use, differing quality levels

You need to keep the original because just like baking, you can't unmake dough once it's mixed.

The data architecture must align with system components because each of them addresses different data needs



Separating concerns is part of the mechanism for change isolation

The design focus is different in each area

Ingredients

Goal: available

User needs a recipe in order to make use of the data.

Pre-mixed

Goal: discoverable and integrateable

User needs a menu to choose from the data available

Meals

Goal: usable

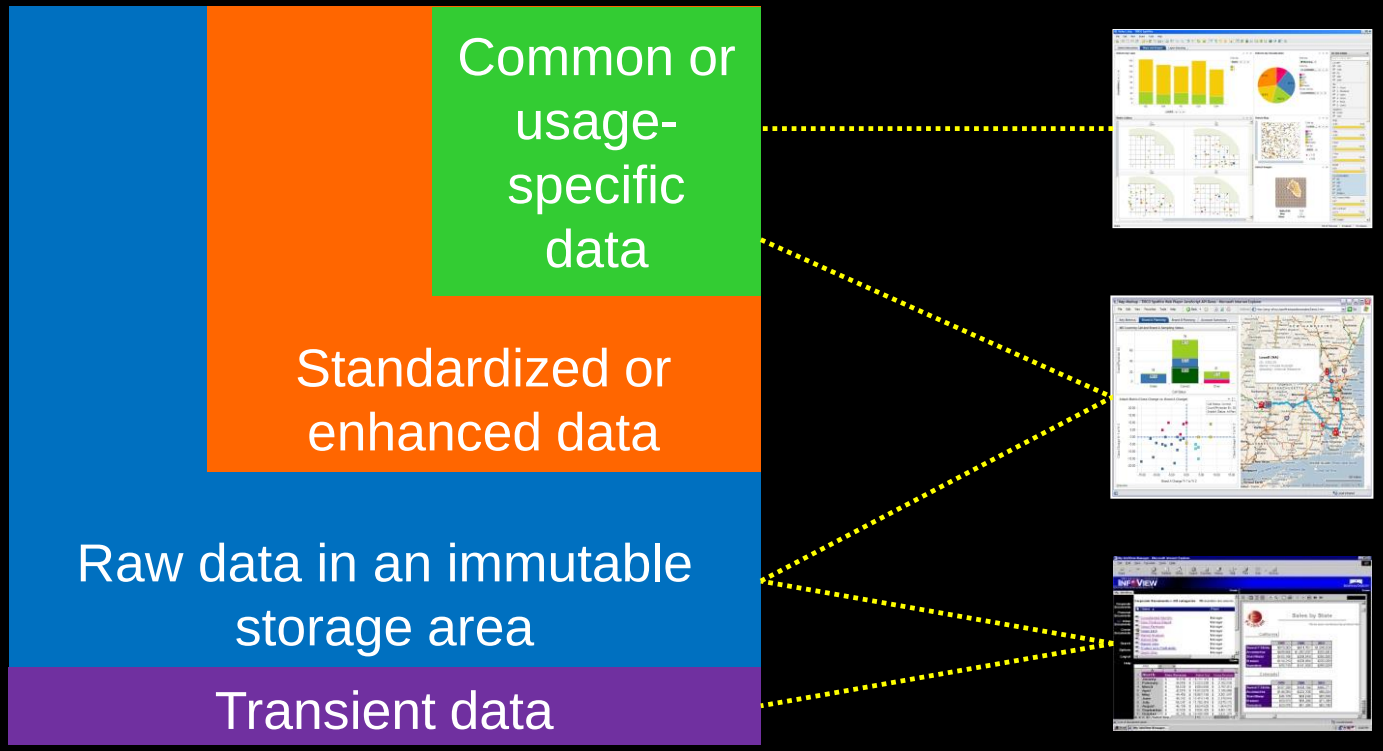
User needs utensils but is given a finished meal

The data is in zones of management, *not* isolating layers

Relax control to enable self-service while avoiding a mess.

Do not constrain access to one zone or to a single tool.

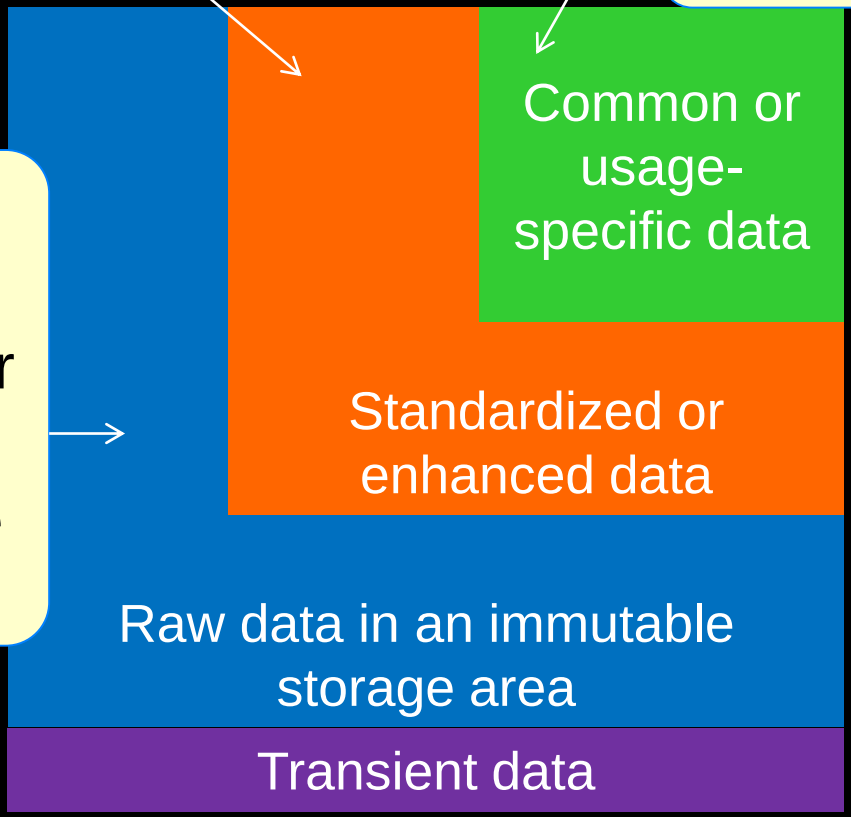
Focus on visibility of data use, not control of data.



This data architecture resolves rate of change problems

More effort applied to management, slower.

Optimized for specific uses / workloads. Generally the slowest change.



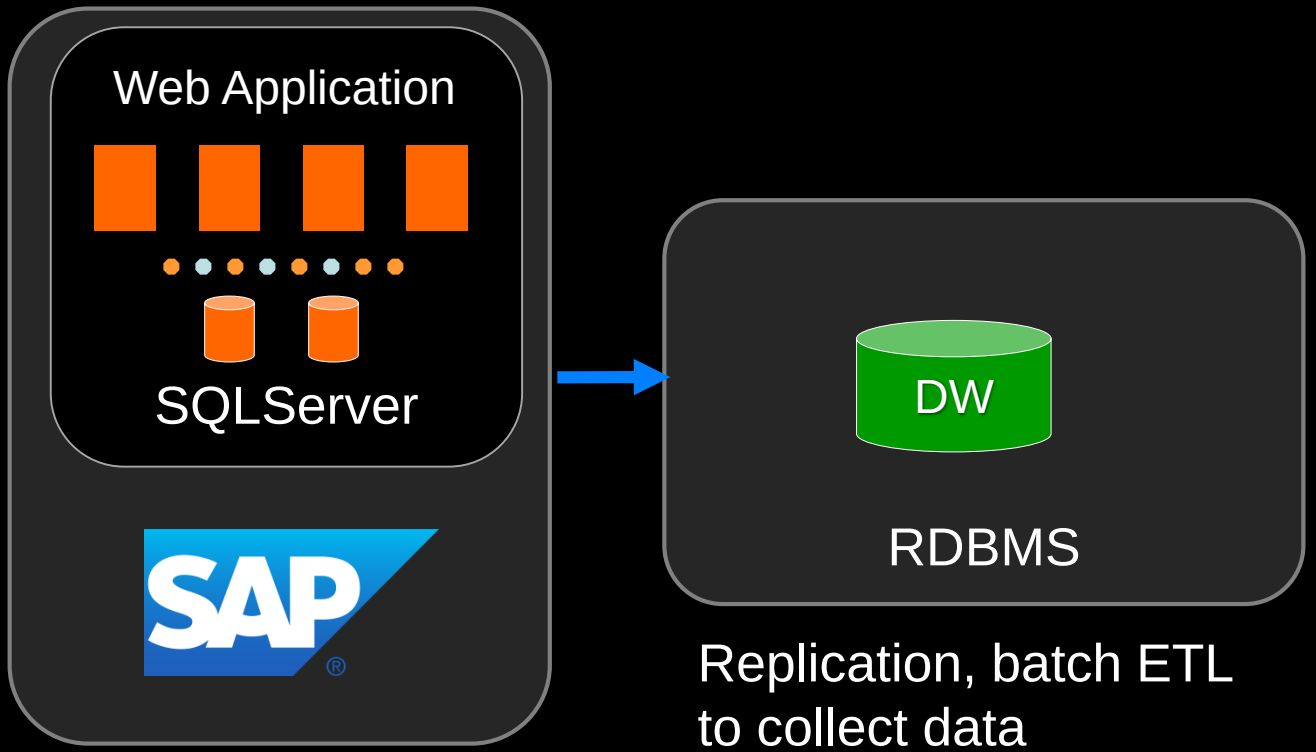
New data of unknown value, simple requests for new data can land here first, with little work by IT.

Not fast vs slow: *fast vs right*

Not flexibility vs control: *flexibility vs repeatability*

Agile for structure change vs agile for questions / use

Example: data environment, mid-size retailer



Replication, batch ETL to collect data

Standard retail / wholesale data warehouse model.

Nightly batch model from stage, contains ~ 1 TB of data

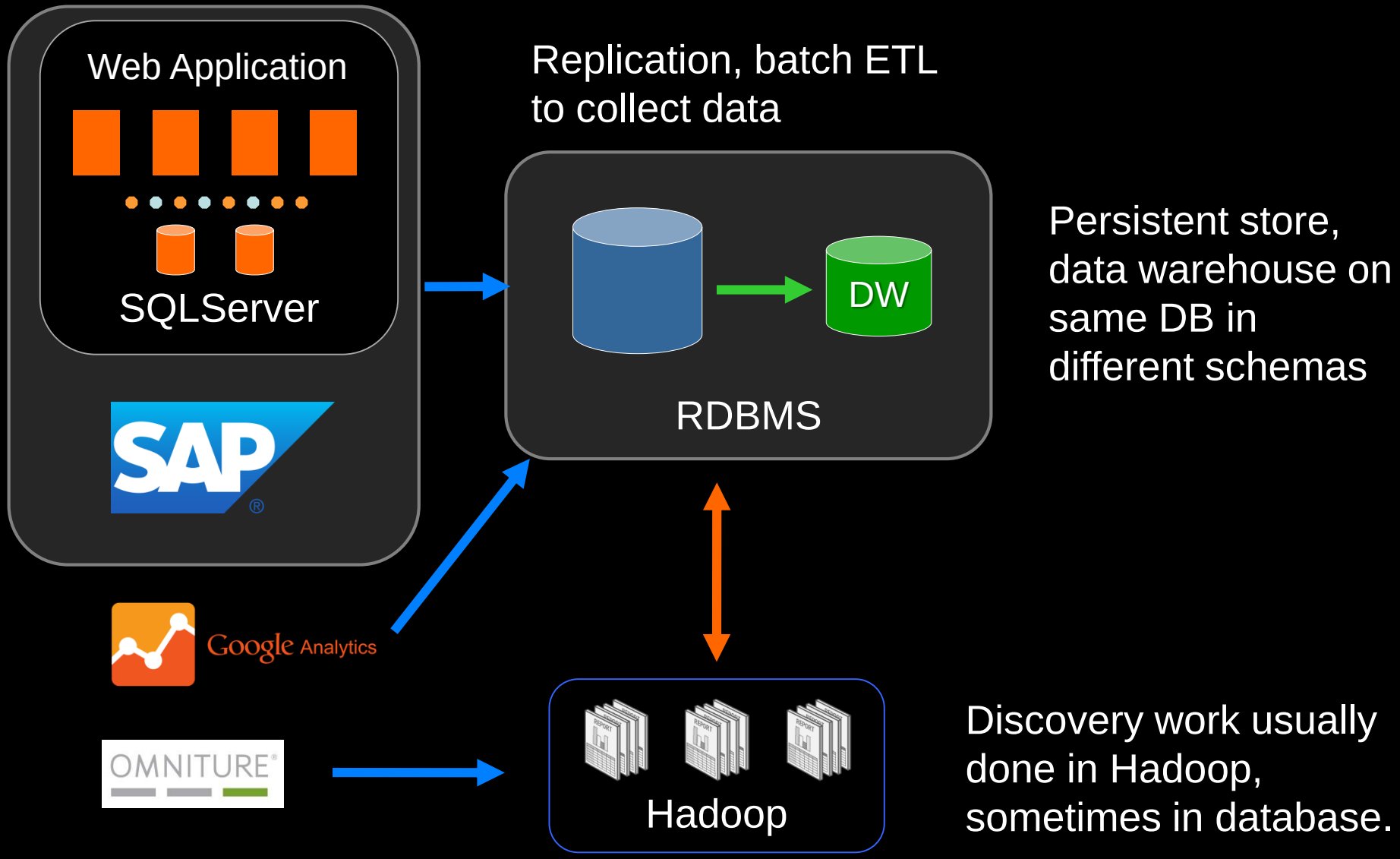
New requirement:

OMNITURE

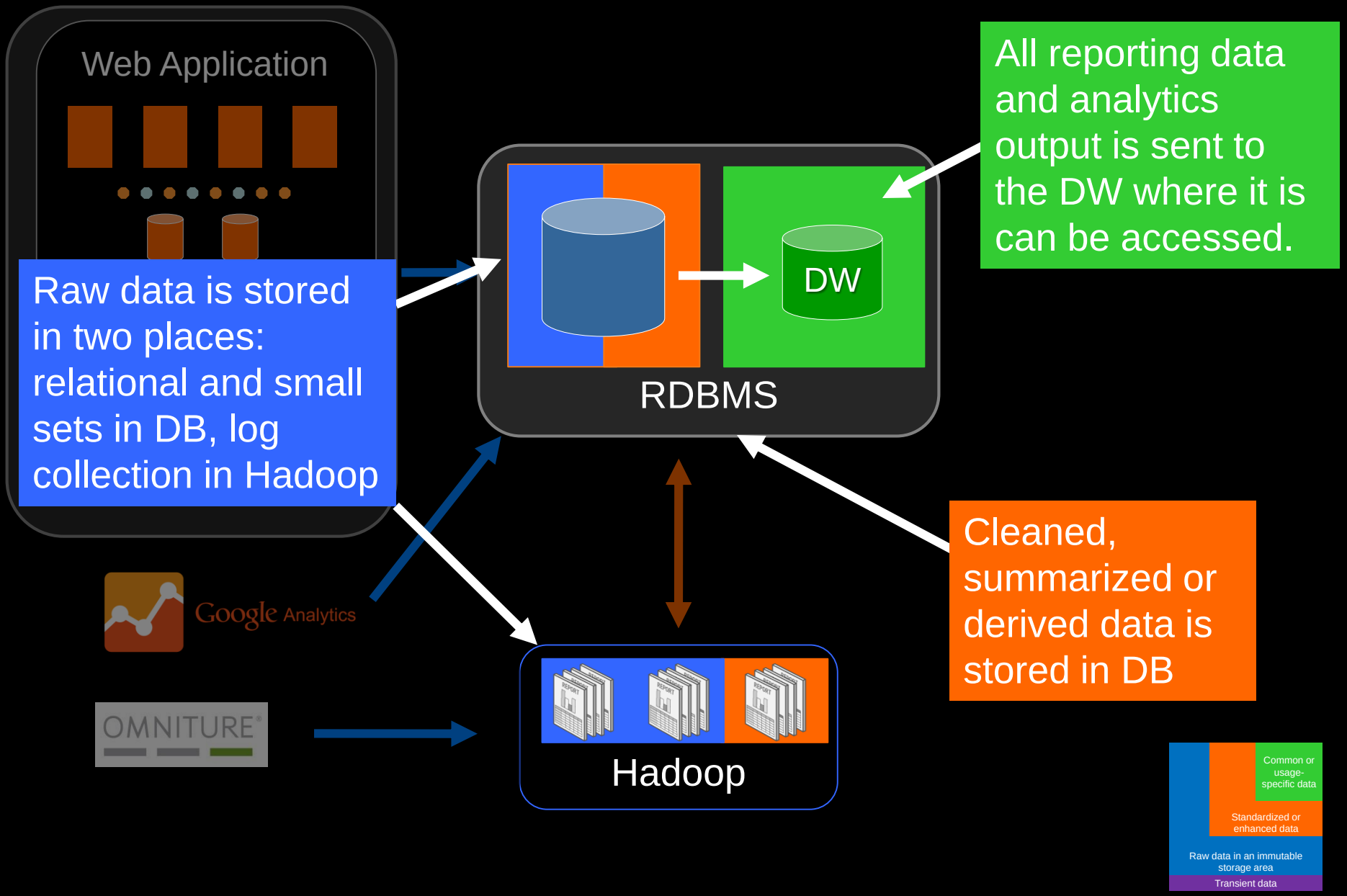
Load all the online marketing and web activity for use in analytic models (not simple web analytics)

Will add ~10 TB of data, continuous or hourly loads

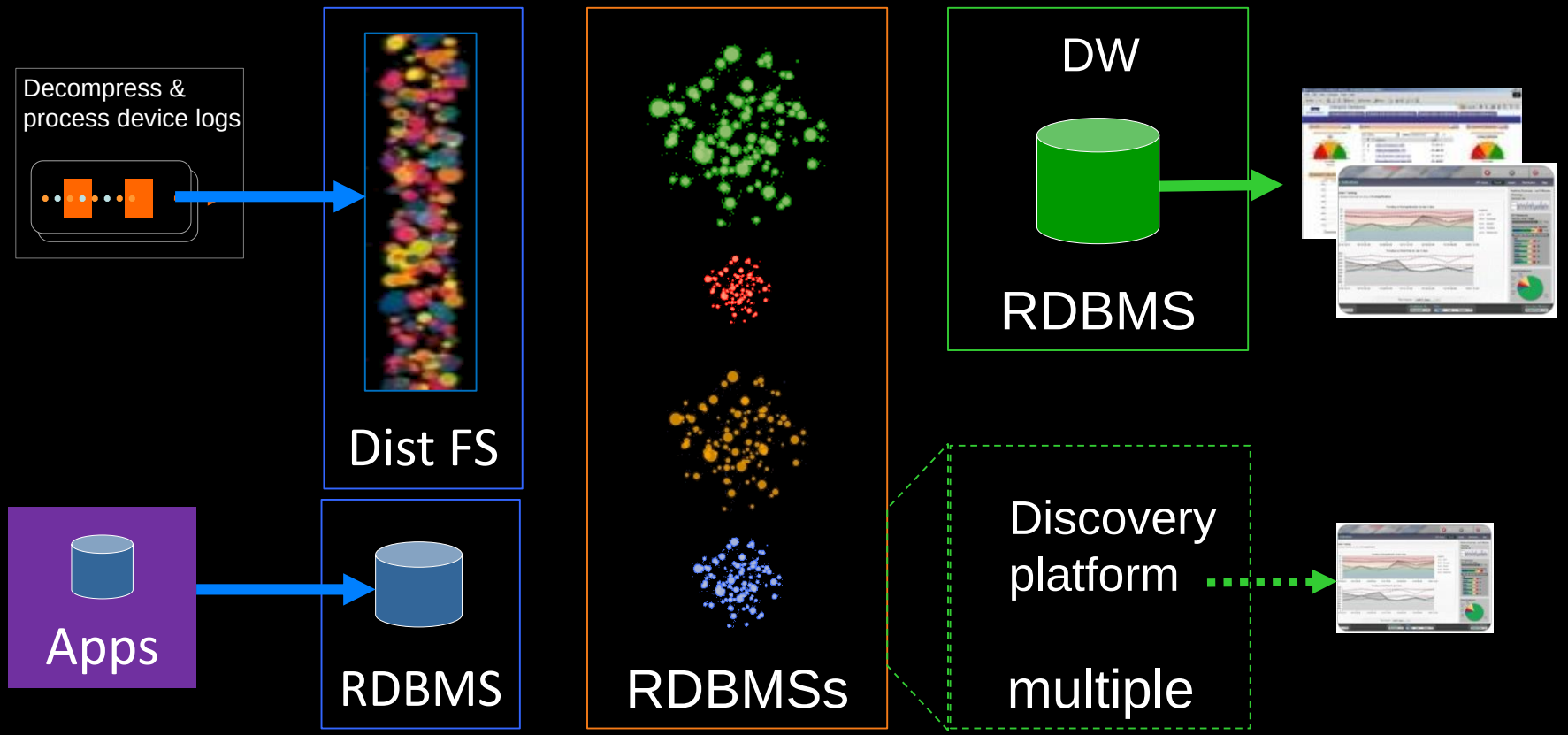
Example: data environment, mid-size retailer



This data architecture uses the 3 zone pattern



The concept of a zone is not a physical system. It's data architecture



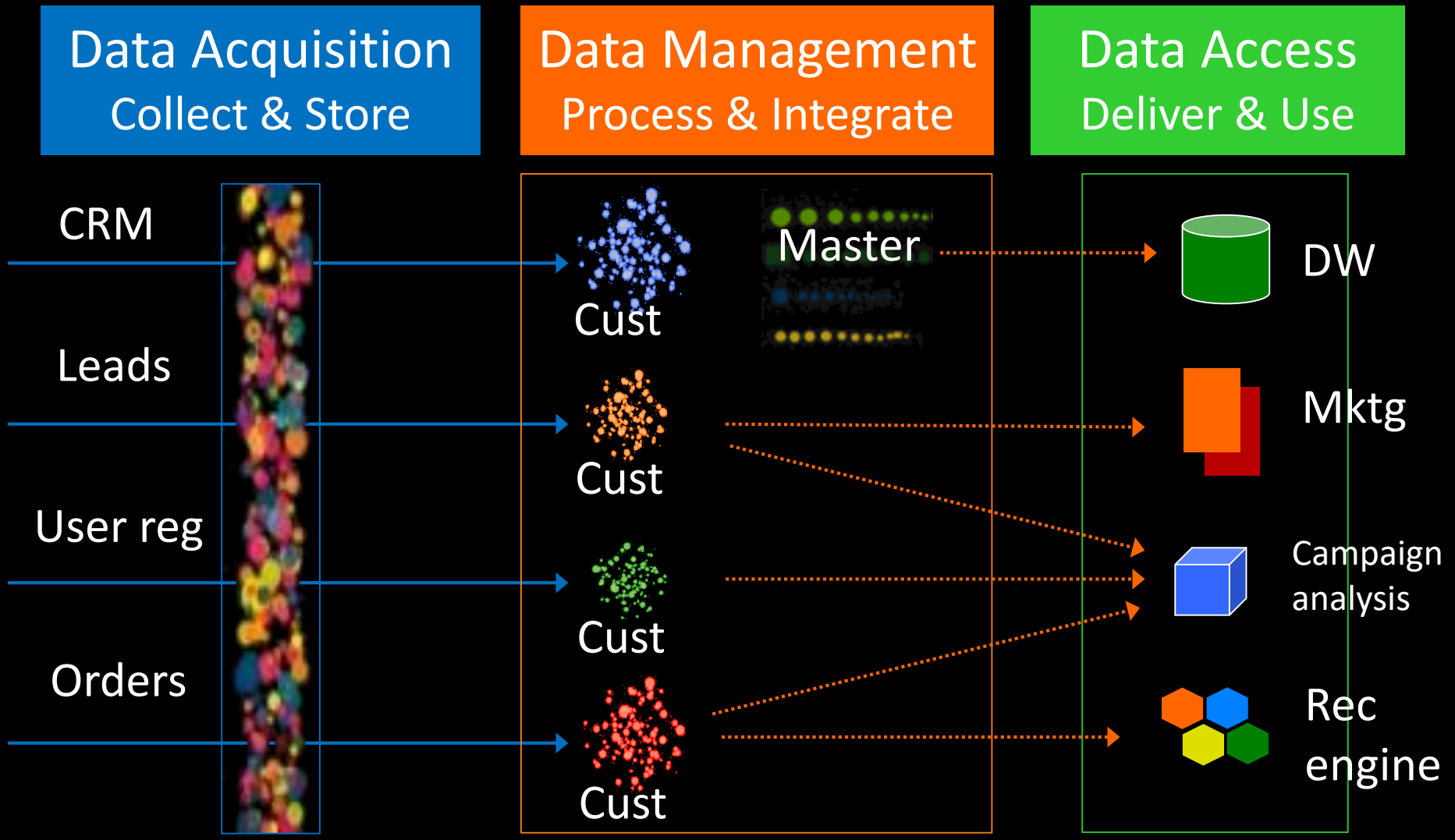
The biggest decision is to separate all data collection from the data integration from consumption.

Physical system/technology overlays are separate, depend on the specific use cases and needs of the organization.

Third Nature

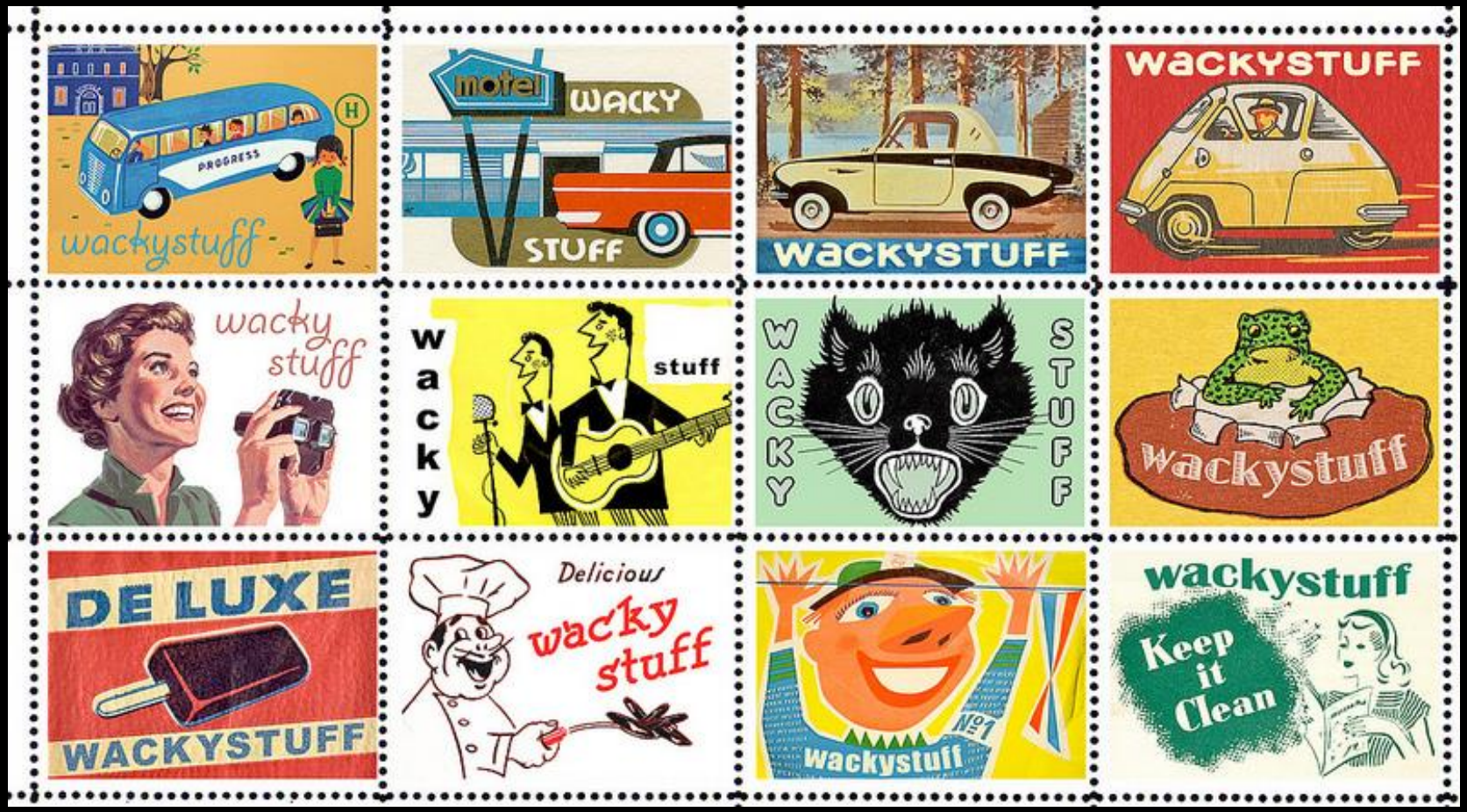
Data has to be moved, standardized, tracked

There is a lot of data policy and governance to think about



Third Nature

What about the technology? Do I need an <X>?



Third Nature

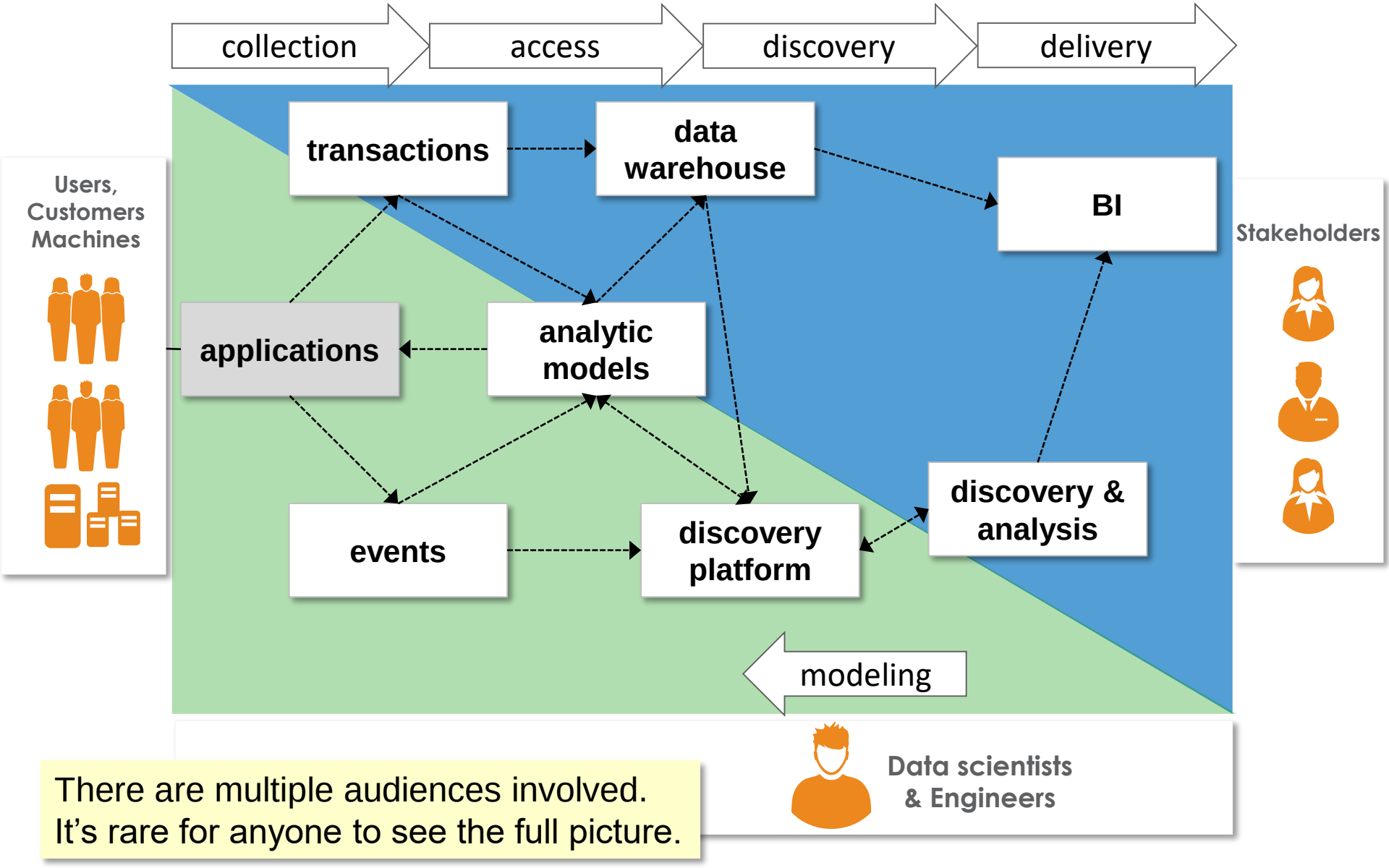
TANSTAAFL

When replacing the old with the new (or ignoring the new over the old) you always make tradeoffs, and usually you won't see them for a long time.

Technologies are not perfect replacements for one another. Often not better, only different.

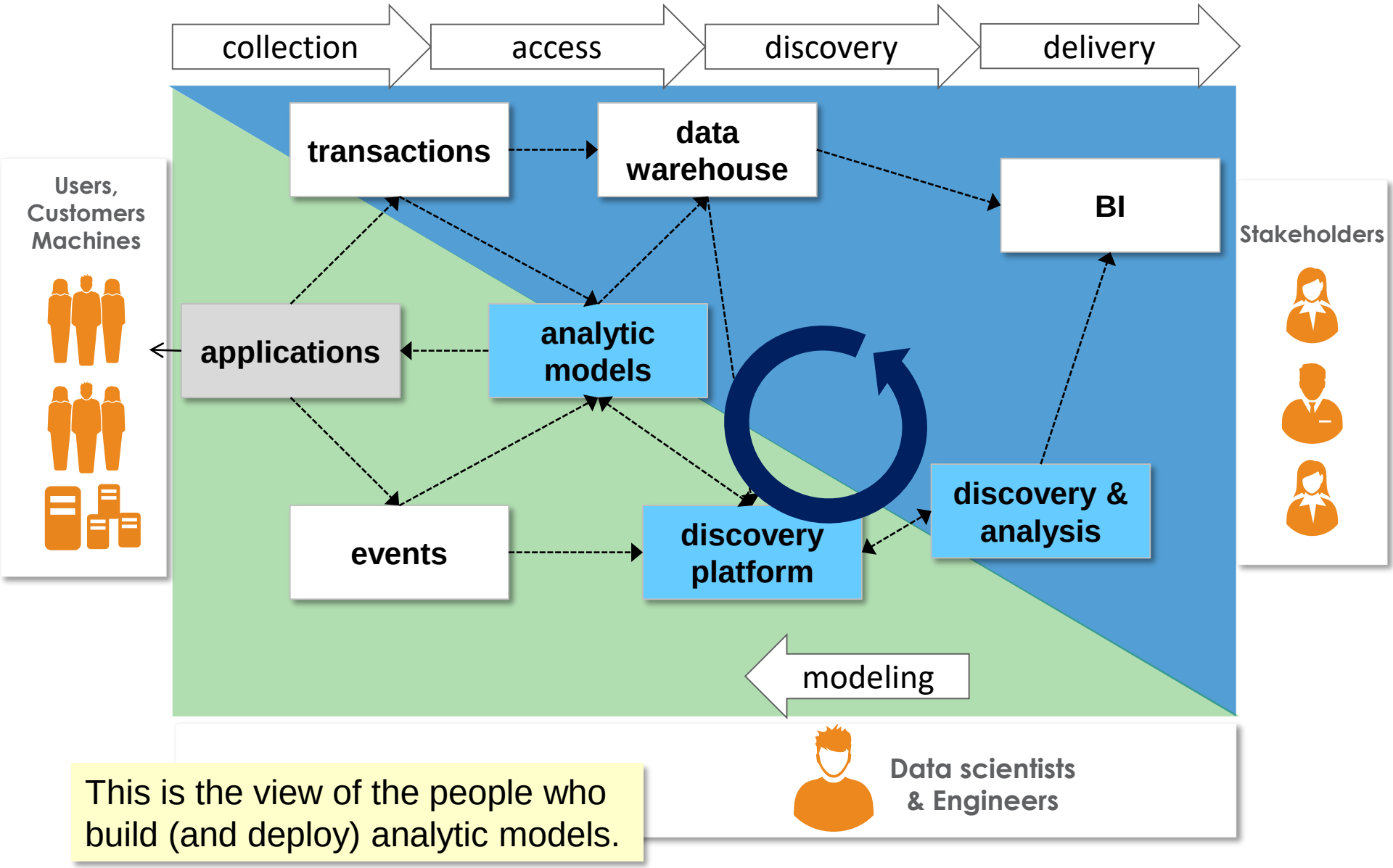
Third Nature

You need to see the full context in order to build a platform.
The analytics environment is more than just silos of activity



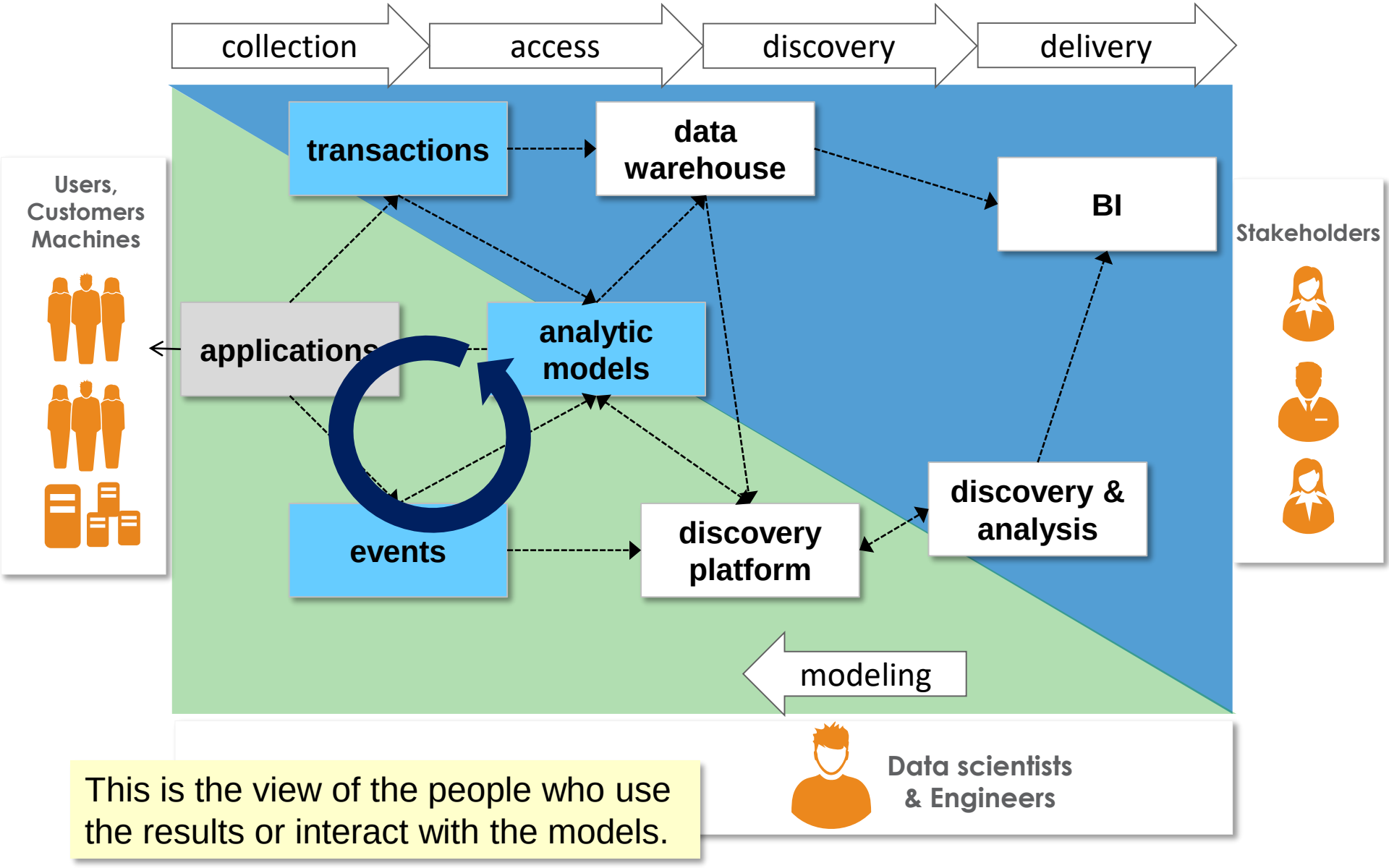
There are multiple audiences involved.
It's rare for anyone to see the full picture.

Environment and workflows: Modeling / Analysis Loop



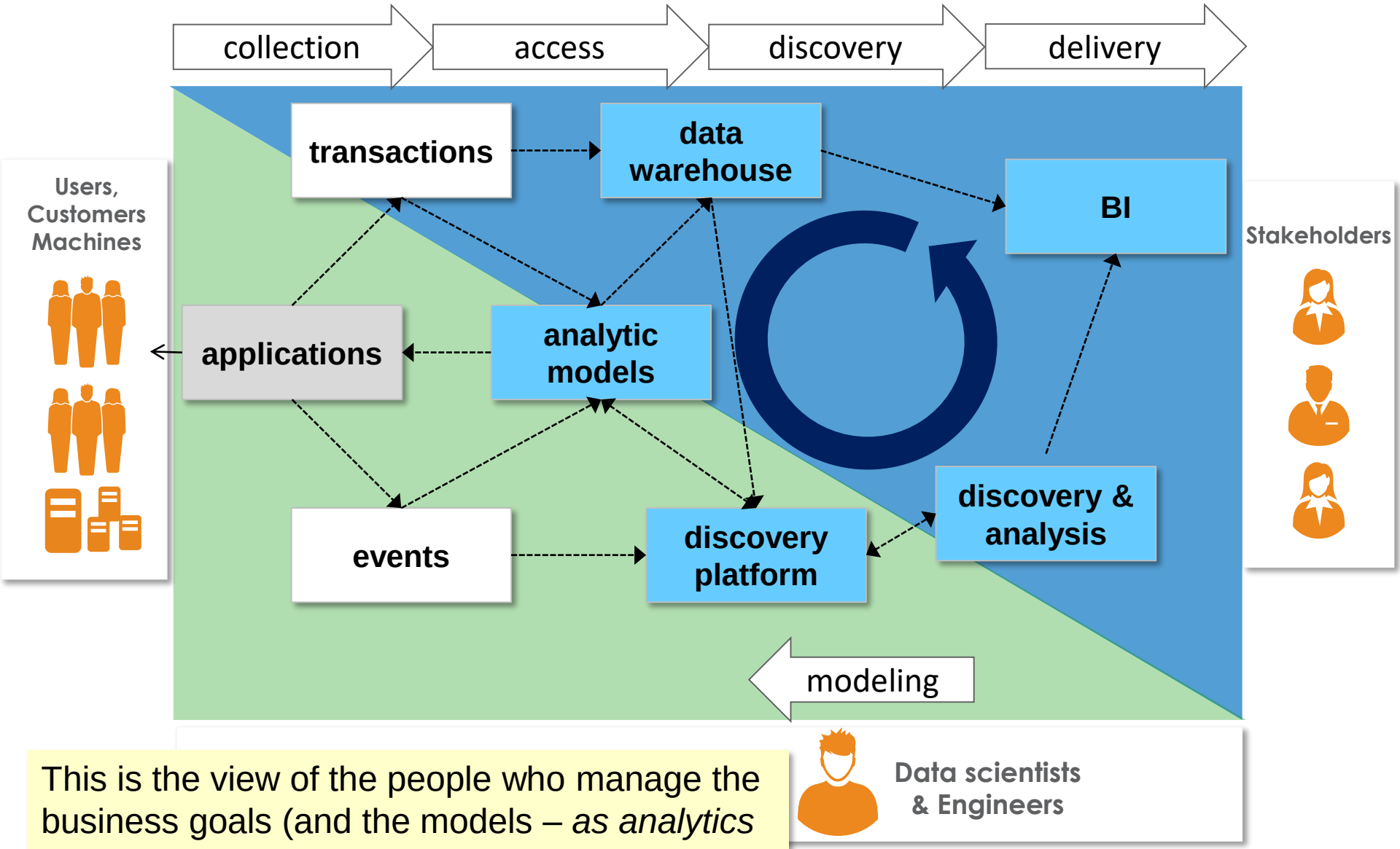
This is the view of the people who build (and deploy) analytic models.

Environment and workflows: Execution Loop



This is the view of the people who use the results or interact with the models.

Environment and workflows: Monitoring and Managing



This is the view of the people who manage the business goals (and the models – *as analytics is operationalized, it becomes the same thing*).

Blended Architectures Are a Requirement, Not an Option

Data Warehouse + Data Lake

On Premise + Cloud

RDBMS + S3 + HDFS

Commercial + Open Source

You can't just buy one thing platform from one vendor. We aren't building a death star.

Each of the zones is likely to have products specific to that zone's usage. The uses differ, the people using them differ, shouldn't the tools should differ too?

In piena foresta indiana, un uomo aspetta il treno vicino alla linea ferroviaria. Improvvisamente un boa assale il malcapitato, stringendolo nelle proprie spire potenti. Ma ecco una tigre slanciarsi a sua volta contro l'enorme rettile il quale avvolge, allora, anche la belva nella stretta mortale. Sul mostruoso groviglio sopraggiunge, frattanto, il treno. Il viluppo è spezzato sanguinosamente dalle ruote del convoglio. (Disegno di A. Beltrame)

Manage your data (or it will manage you)

Data management is where developers are weakest.

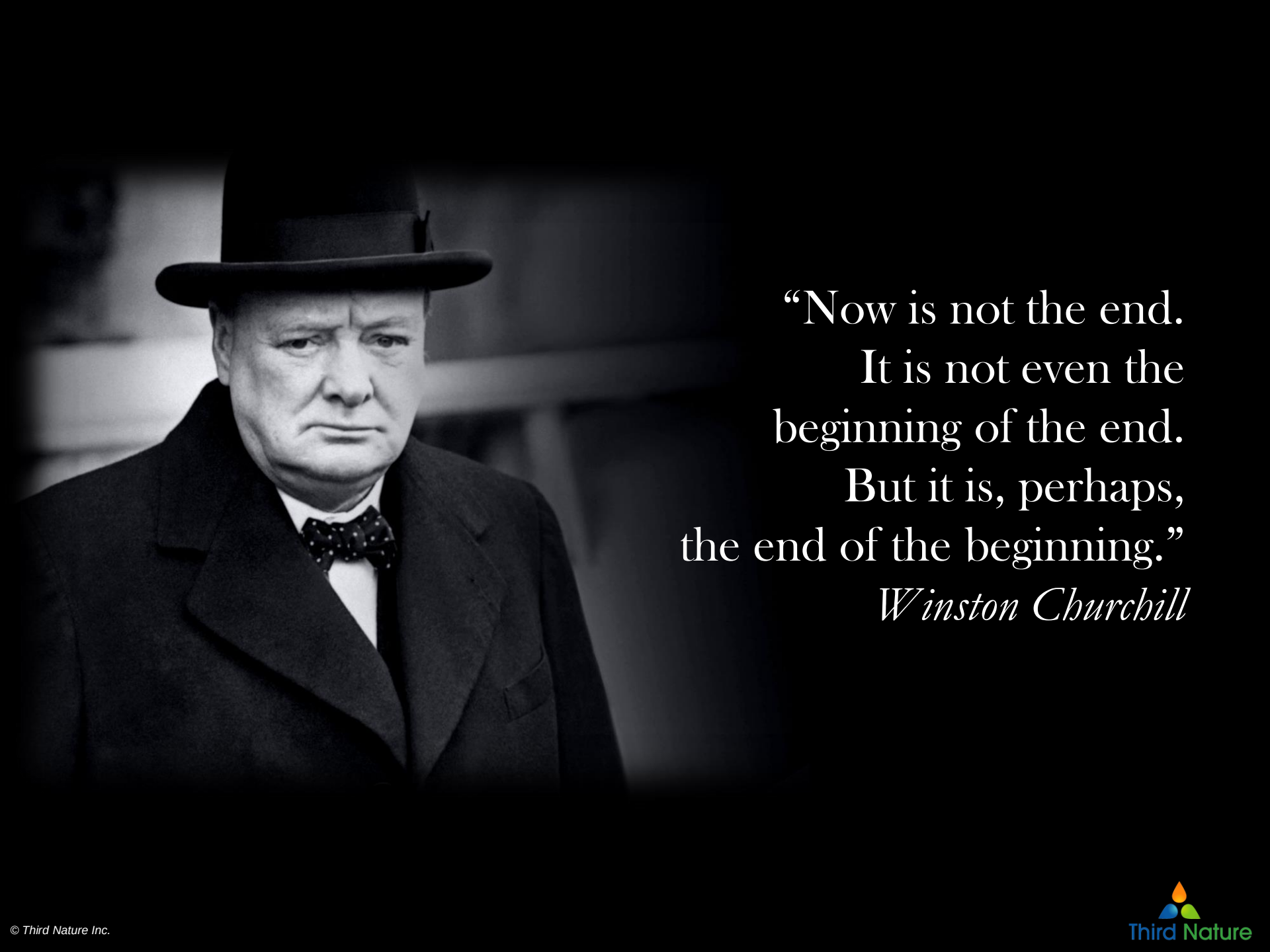
Modern engineering practices are where data management is weakest.

You need to bridge these groups and practices in the organization if you want to do meaningful work with data. Remember Conway's Law when you build.

Third Nature

Conclusion

1. It’s not about storing data, it’s about using it
2. Use drives architecture. Understand the uses, what you are designing for, to drive decisions.
3. Put data at the center, not technology. Don’t let the tech define what you can do or how you do it.
4. The death star is not the answer. The data model is not a flat earth. You are not building a monolith.
5. Know your history. Avoiding wheel reinvention saves time, money, careers.



Third Nature

About the Presenter

Mark Madsen is the global head of architecture at Think Big Analytics, Prior to that he was president of Third Nature, a research and consulting firm focused on analytics, data integration and data management. Mark is an award-winning author, architect and CTO whose work has been featured in numerous industry publications. Over the past ten years Mark received awards for his work from the American Productivity & Quality Center, TDWI, and the Smithsonian Institute. He is an international speaker, chairs several conferences, and is on the O'Reilly Strata program committee. For more information or to contact Mark, follow @markmadsen on Twitter.

Todd Walter

Chief Technologist - Teradata

- Chief Technologist for Teradata
- A pragmatic visionary, Walter helps business leaders, analysts and technologists better understand all of the astonishing possibilities of big data and analytics
- Works with organizations of all sizes and levels of experience at the leading edge of adopting big data, data warehouse and analytics technologies
- With Teradata for more than 30 years and served for more than 10 years as CTO of Teradata Labs, contributing significantly to Teradata's unique design features and functionality
- Holds more than a dozen Teradata patents and is a Teradata Fellow in recognition of his long record of technical innovation and contribution to the company

TERADATA.