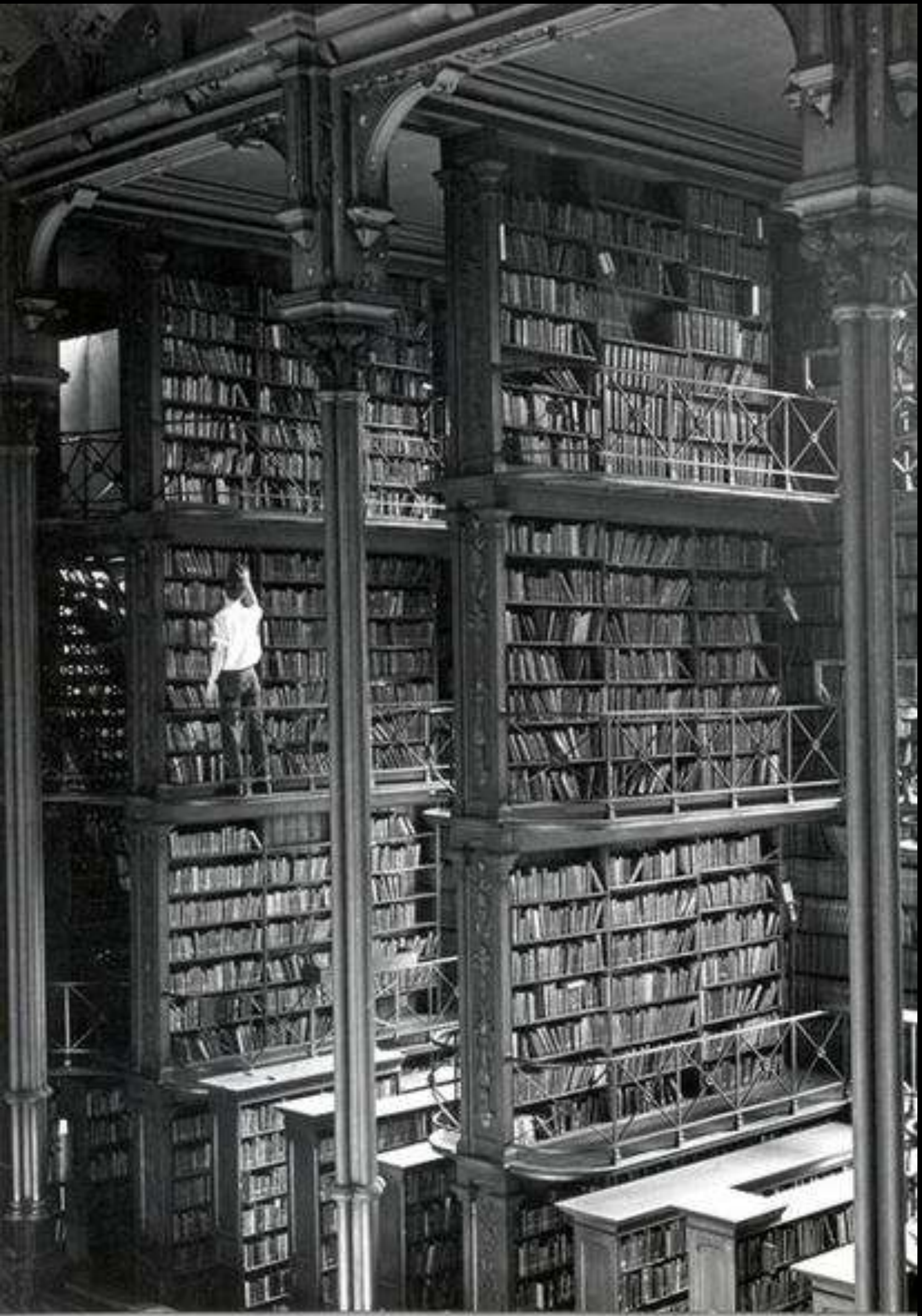


Organizing the Data Lake

December, 2016

Mark Madsen
www.ThirdNature.net
@markmadsen

A four part presentation in three parts

1. How the traditional data warehouse / business intelligence industry sees the market.
2. Architectural principles
3. Data lake functional and data architecture

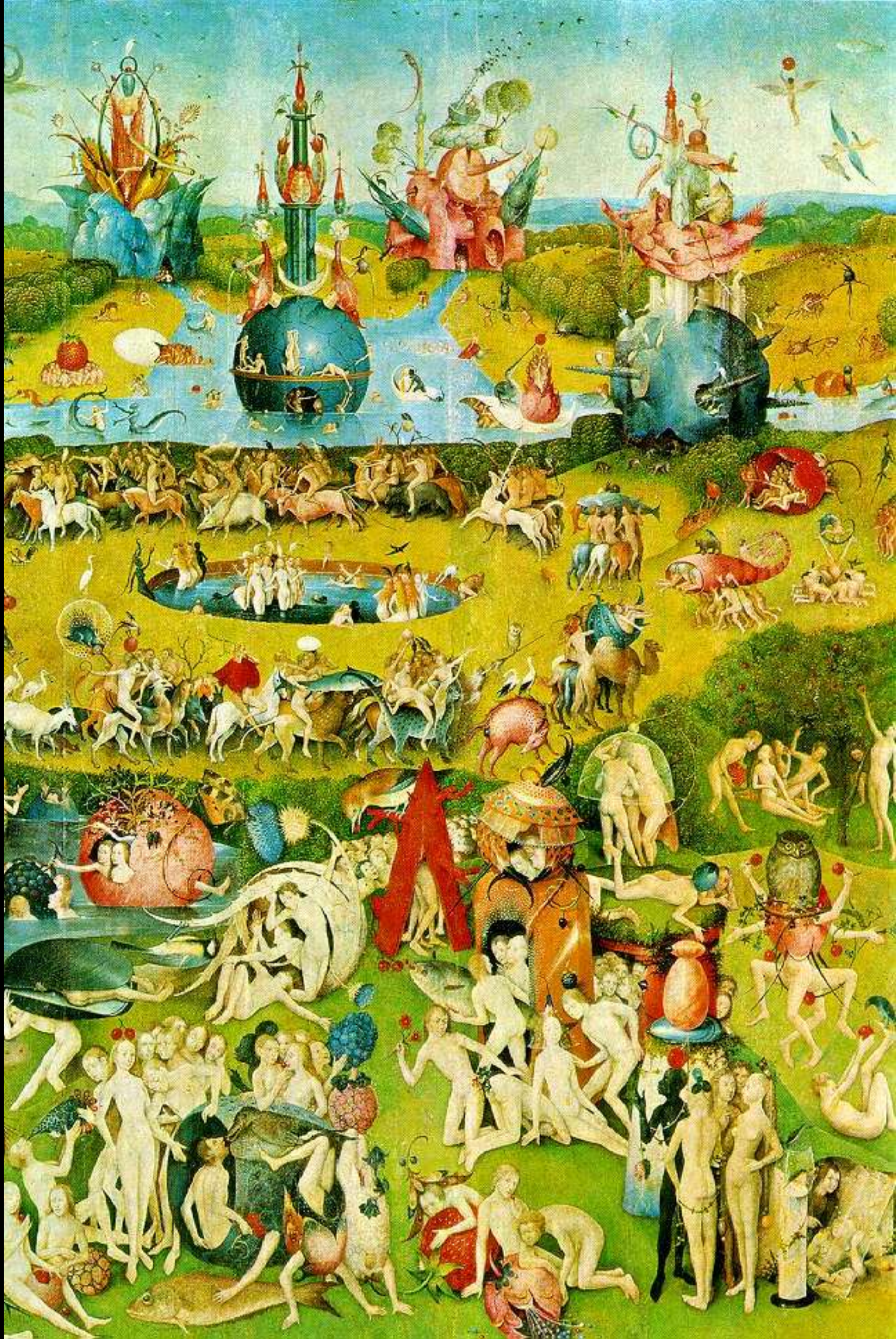
The big data market is:

… a leap forward

A leap in evolution to a more flexible way of gathering data and generating useful information.

- Decide if data is good enough at the time of use
- Files are flexible
- Save time collecting data
- “self service”

A *new approach* to managing and using data.



The big data market is:

… a step backward

A step backward to methods not capable of providing the quality, manageability, accessibility and reuse we need.

- Ensure data quality up front rather than after the fact
- Tables are stable
- Save time using data
- “self service”

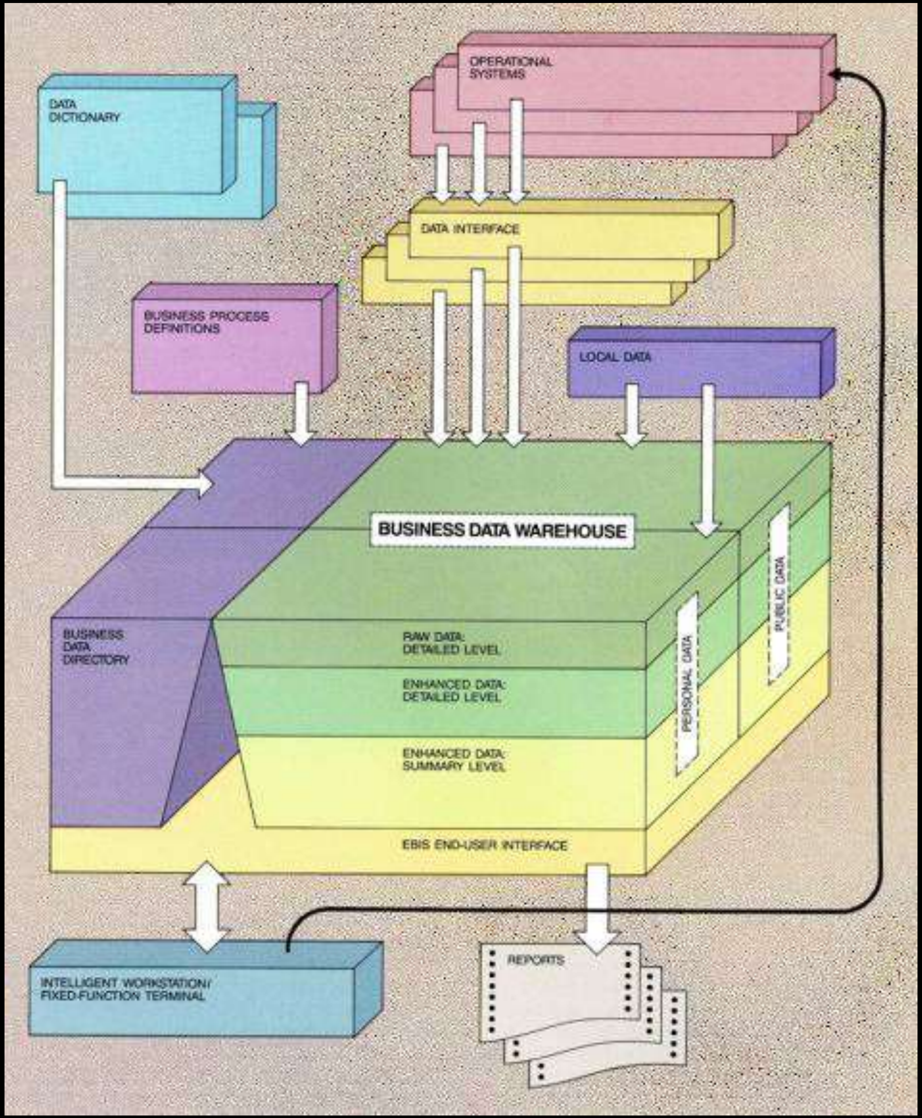
A *reinvention* of the wheel, going back to the manual era.

Third Nature

The architecture DWs are (still) using today

The general concept of a separate architecture for DW 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)

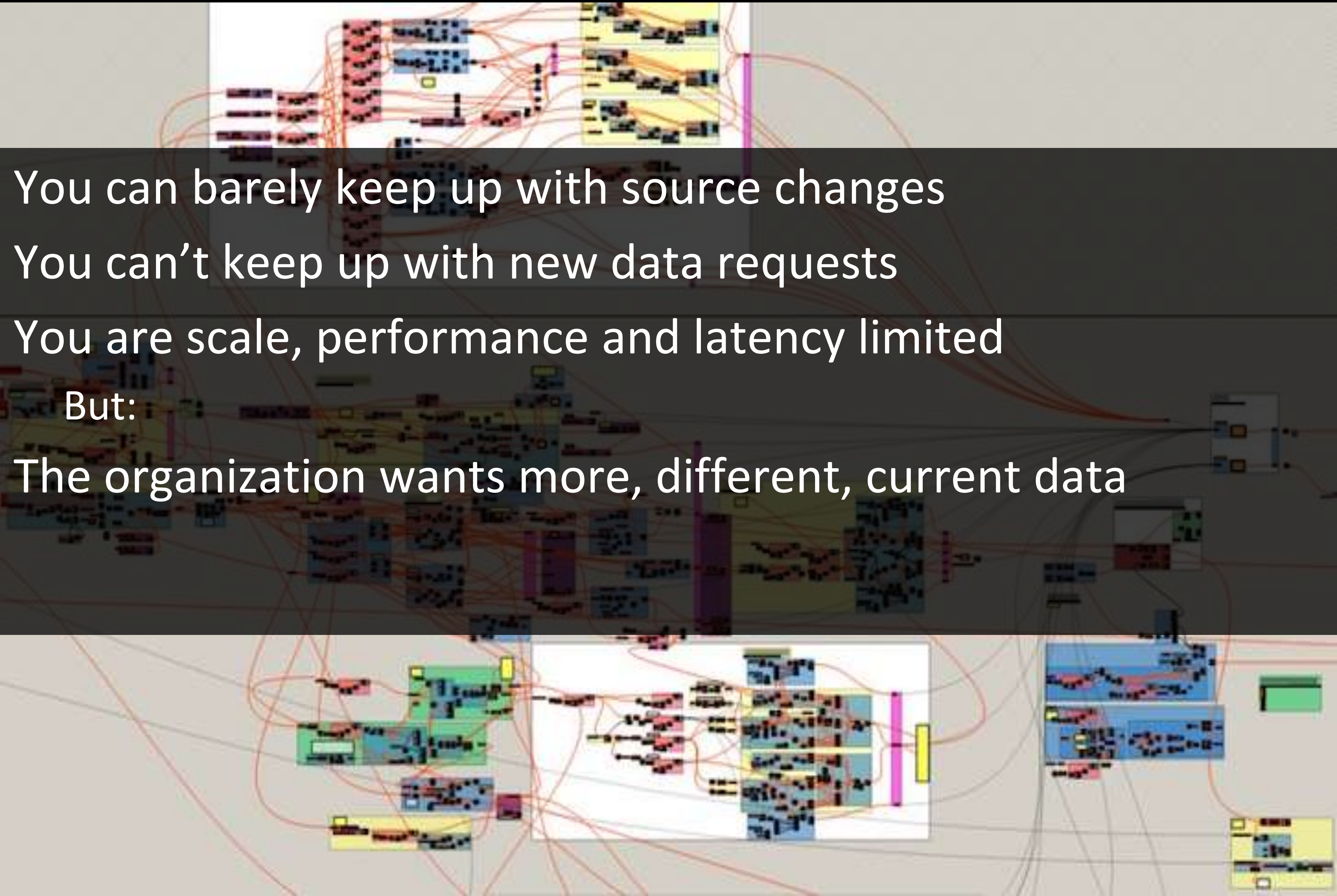


So we shifted to data publishing

Industrialized data delivery for self-service *<u>access</u>*.

But the creation and distribution of data is still a craft.

Growing complexity changed the IT environment



You can barely keep up with source changes

You can't keep up with new data requests

You are scale, performance and latency limited

But:

The organization wants more, different, current data

What do you mean, "3 months?"
I never said the "E" in EDW meant "everything"…

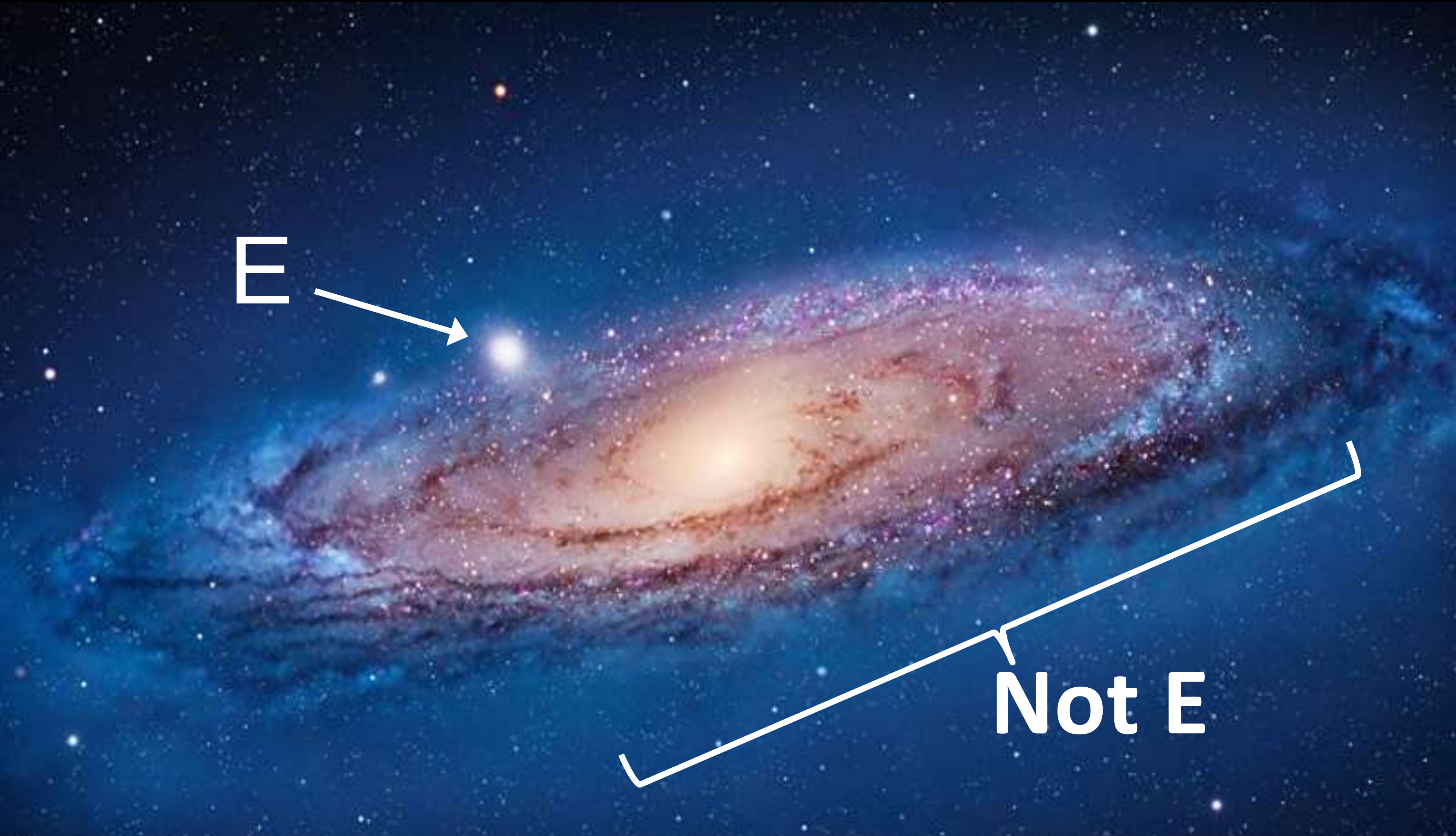
Meanwhile, IT has become the department of “No”

- Environments are more complex
- Data volumes are 10,000X larger
- Data is more complex
- More use of BI
- We have entirely new data uses

We’ve been struggling with performance problems and an inability to quickly meet new data and analytics requests for years, *yet we keep using the same designs.*

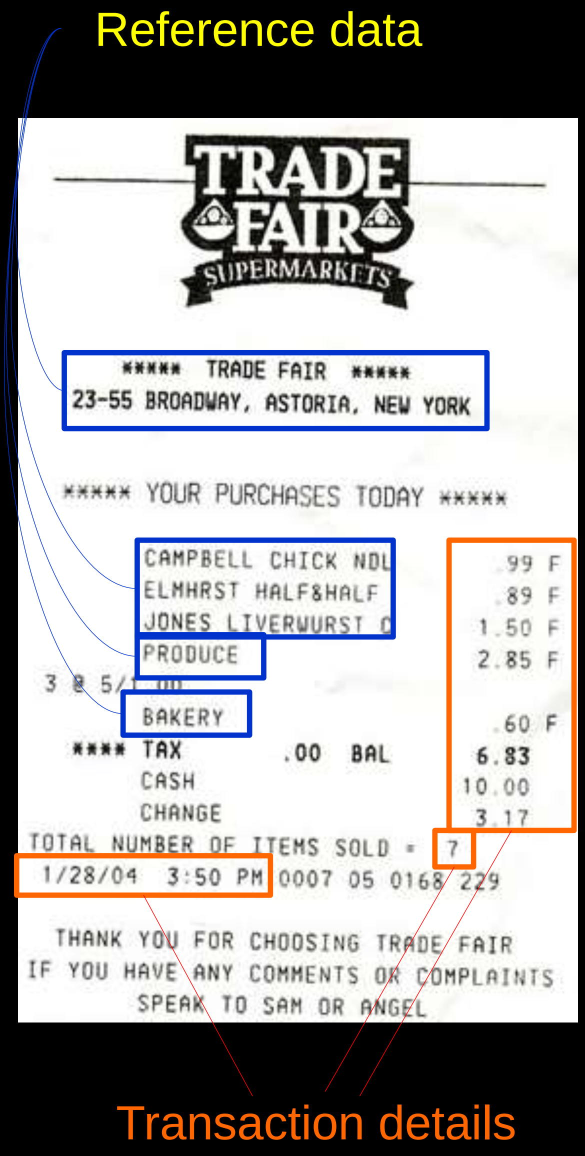
for prompt control of
senile agitation
THORAZINE*
'Thorazine' can control the agitated, belligerent senile and help the patient to live a composed and useful life.
Smith Kline & French Laboratories

Tell the DBAs it's going to get a *lot* worse



Conclusion: any methodology built on the premise that you must know and model all data before using it is untenable.

Transactions: what they are familiar with


Reference data
Transaction details

```
TRADE FAIR SUPERMARKETS

*****  TRADE FAIR  *****
23-55 BROADWAY, ASTORIA, NEW YORK

***** YOUR PURCHASES TODAY *****

       CAMPBELL CHICK NDL          .99 F
       ELMHRST HALF&HALF           .89 F
       JONES LIVERWURST O         1.50 F
       PRODUCE                    2.85 F
  3 @ 5/1.00
       BAKERY                      .60 F
  **** TAX         .00   BAL      6.83
       CASH                      10.00
       CHANGE                     3.17
TOTAL NUMBER OF ITEMS SOLD =  7
  1/28/04   3:50 PM 0007 05 0168 229

  THANK YOU FOR CHOOSING TRADE FAIR
IF YOU HAVE ANY COMMENTS OR COMPLAINTS
        SPEAK TO SAM OR ANGEL
```

The classic example of "structured data"

Transaction data includes:

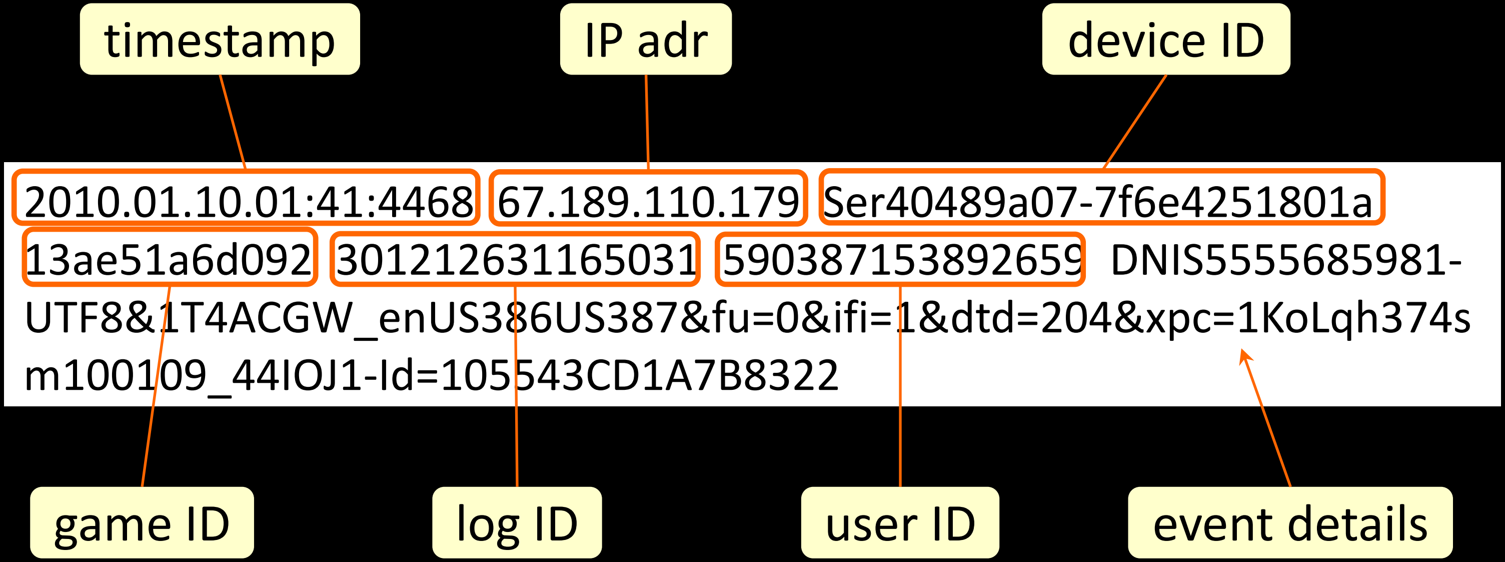
- quantification details (date, value, count)
- reference data for explanation (product, customer, account)
- Lots of meaningful information

Reference data is usually shared across the organization, hence its importance. There are two parts:

- identifier to uniquely identify the subject
- descriptive attributes with common or standardized value domains

Third Nature

Event streams contain mainly IDs referencing other data



```
2010.01.10.01:41:4468 67.189.110.179 Ser40489a07-7f6e4251801a
13ae51a6d092 301212631165031 590387153892659 DNIS5555685981-
UTF8&1T4ACGW_enUS386US387&fu=0&ifi=1&dtd=204&xpc=1KoLqh374s
m100109_44IOJ1-Id=105543CD1A7B8322
```

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?

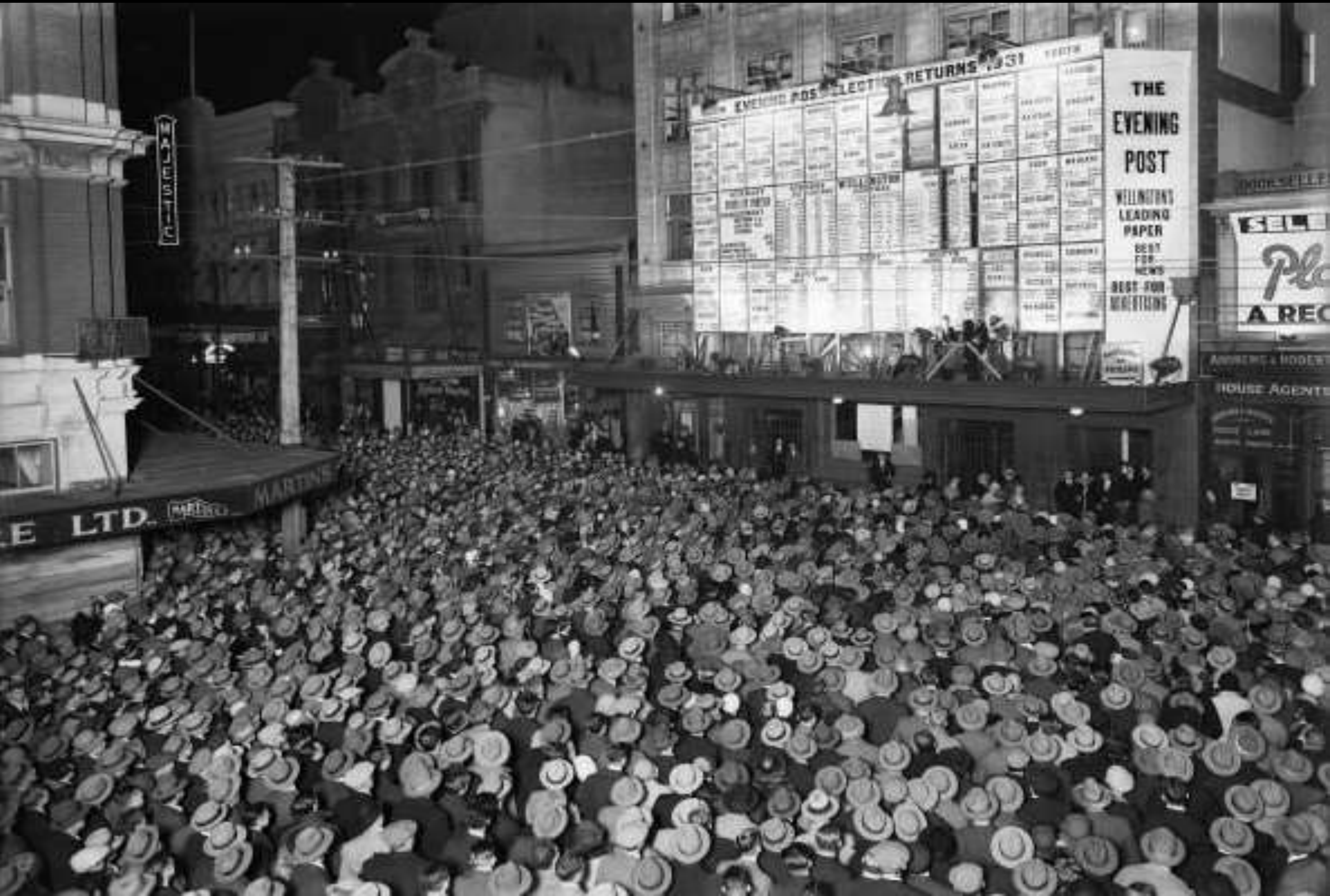
It’s just keys to other data.

e.g. remember that device ID 0 problem?

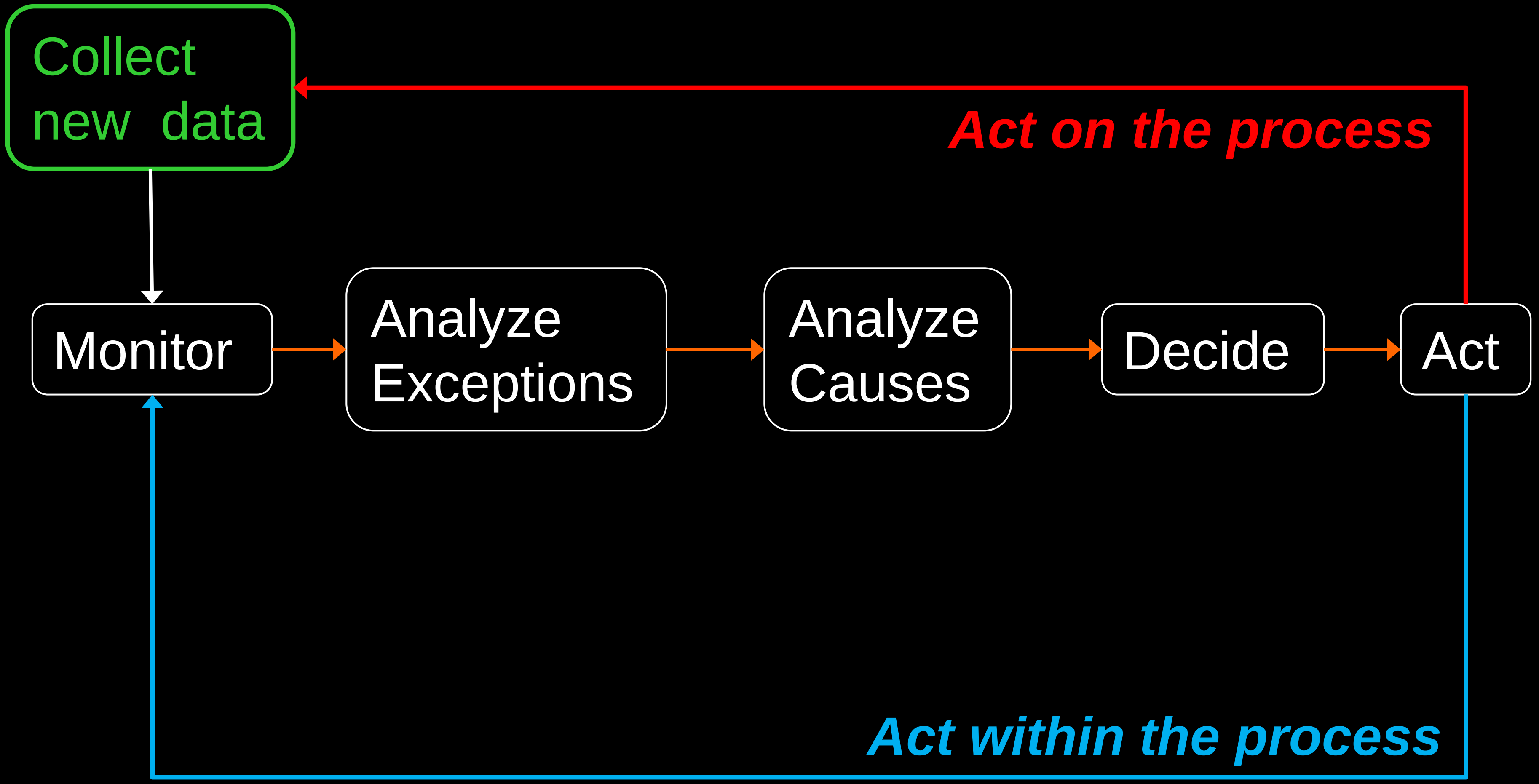
New uses: analytics embiggens the data volume problem

Many of the algorithms we need them to make sense of the observations and declarations are are $O(n^2)$ or worse.

Publishing is the primary view of BI, self service, big data

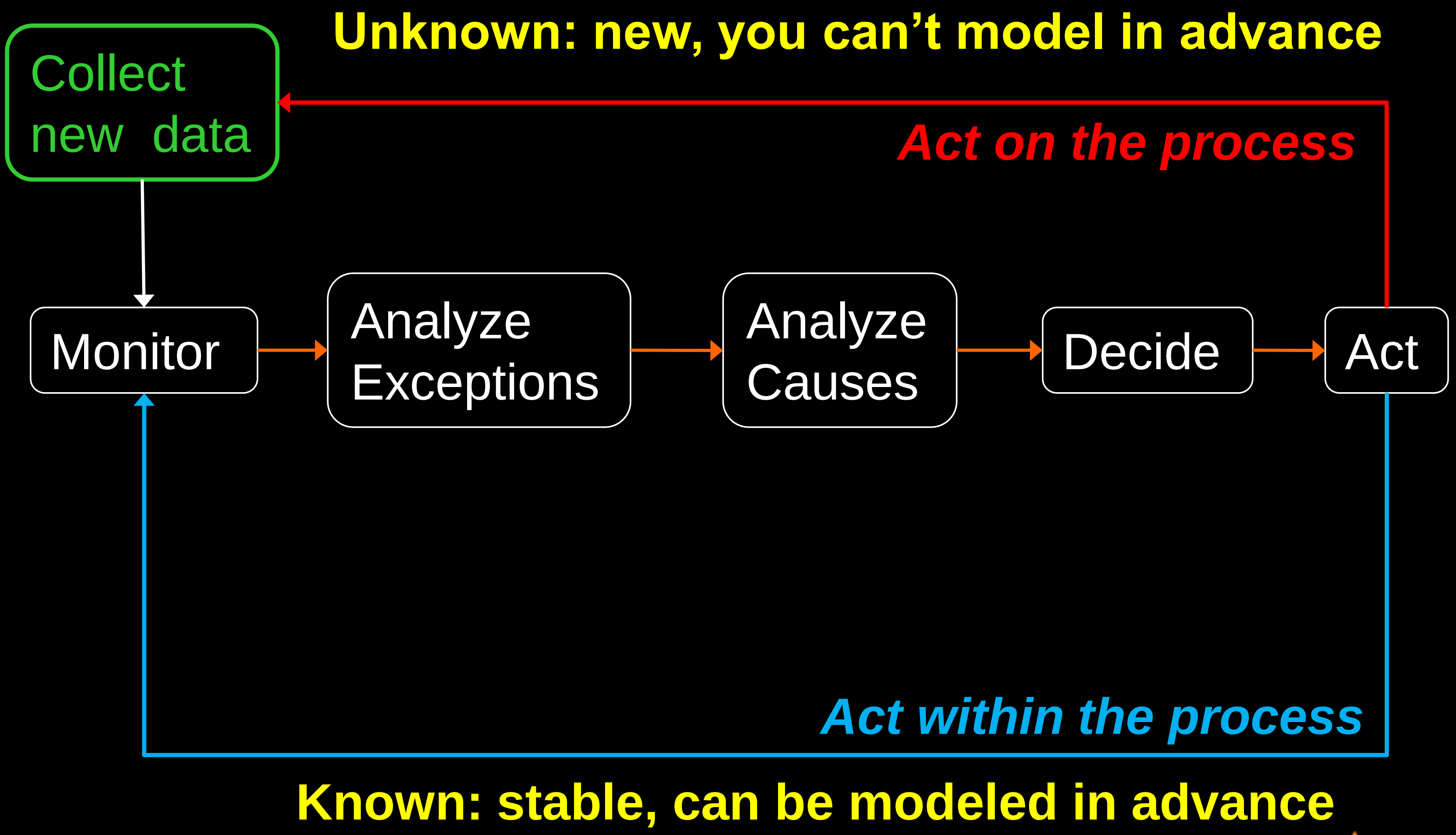


Data architecture requires understanding data use so we can build the right infrastructure



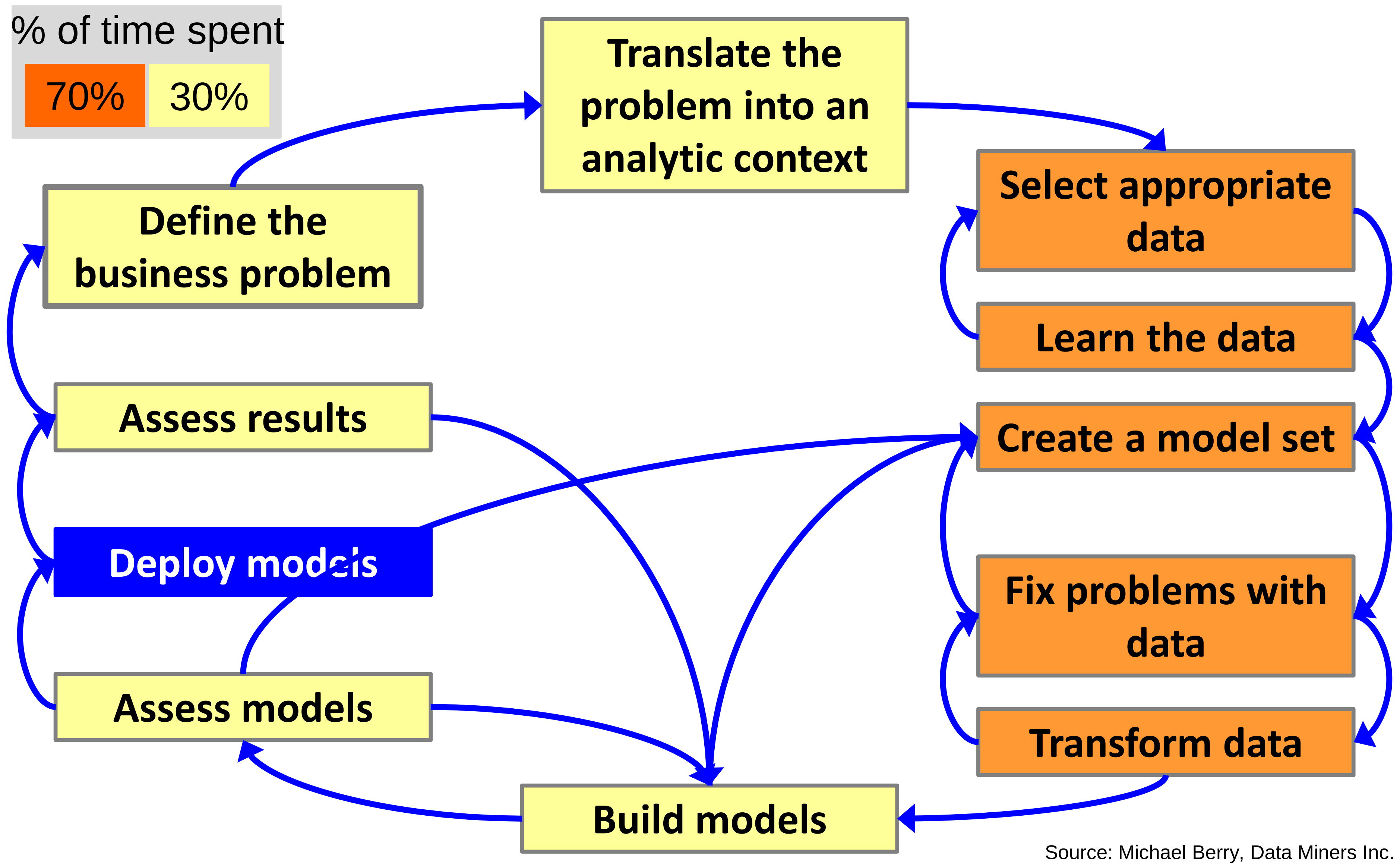
We need to focus on what people do with information as the primary task, not on the data or the technology.

Information is part of a dynamic system. There are feedback loops that can change both data and models.



Third Nature

And then there's analytics and data science…



Source: Michael Berry, Data Miners Inc.

What does all of this imply?

1. The data is not always known in advance, so it can’t be modeled in advance.
2. The data architecture must be read-write from both the back and front, not a one-way data flow.
3. The data written back may be repeatedly used, persistent data, or it may be temporary.
4. The data may arrive with any frequency, and the rate may not be under your control.

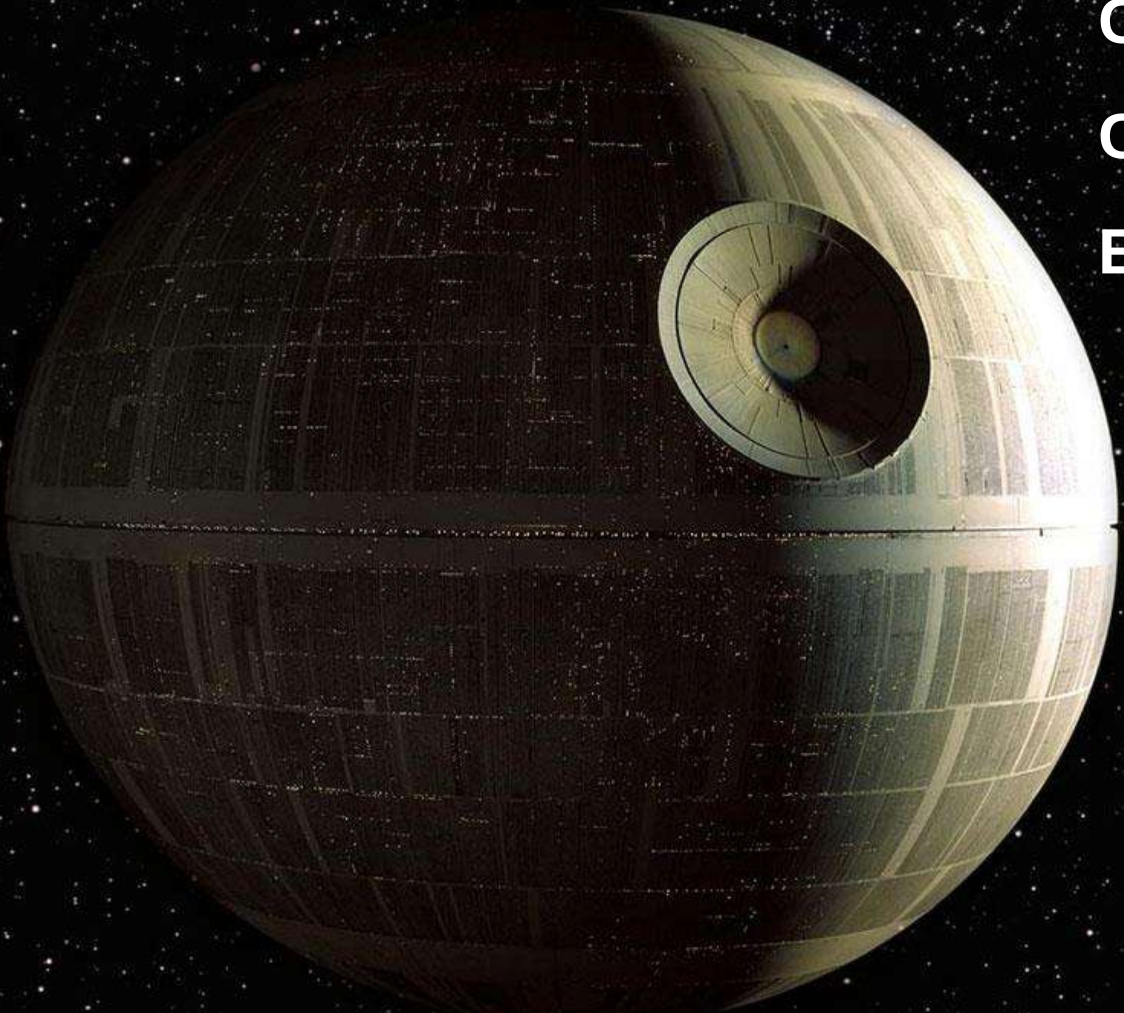
These are the opposite of assumptions in the architecture and methodology of a data warehouse.

The rate of change in enterprise data infrastructure is slower than the business. This can't be fixed by buying more technology.

The root cause of our problems is *change*

Third Nature

Data warehouse: centralize, that solves all problems!



Creates bottlenecks

Causes scale problems

Enforces a single model

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’s still a death star.

Third Nature

Eventually we run into the same problems

Third Nature

We have a design for stability. We need one for adaptability

Their view: nothing wrong with what we have. Just buy the latest product from the approved vendor

They see the big data market answer as…

The naïve data lake: just dump the data in!

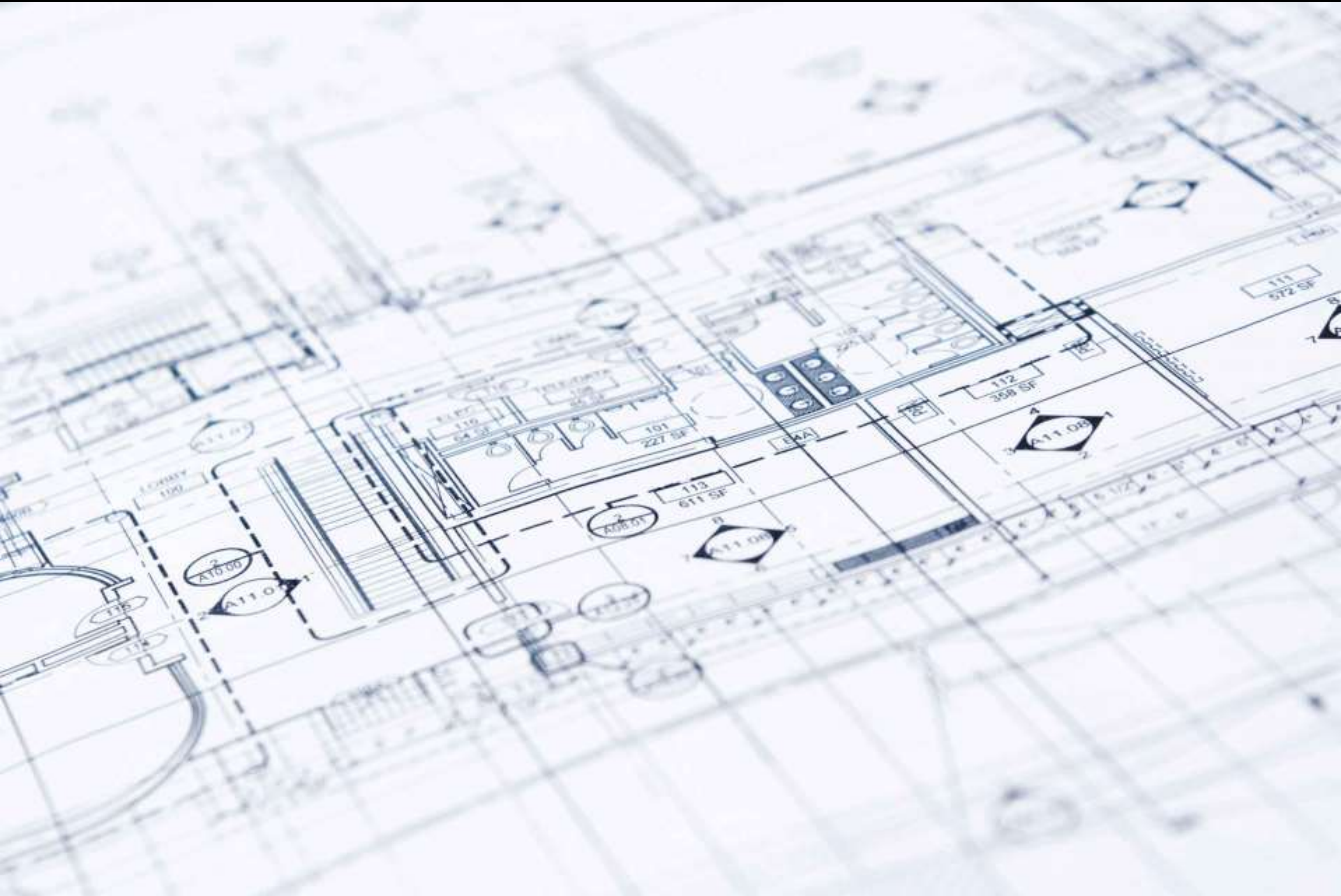
Combine with self-service: we'll figure it all out later!

Aren't we back where we started?

Third Nature

The solution to our problems isn't technology, it's architecture.

Blueprints are not architectures



Bricks are not buildings

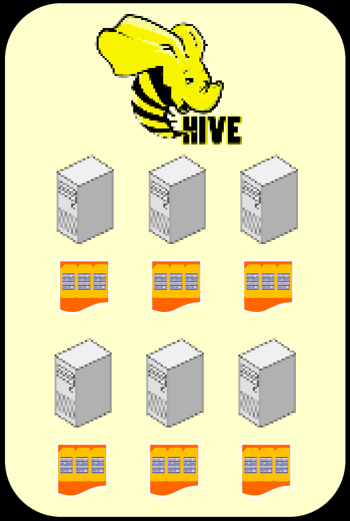
We don’t think this is equivalent to this

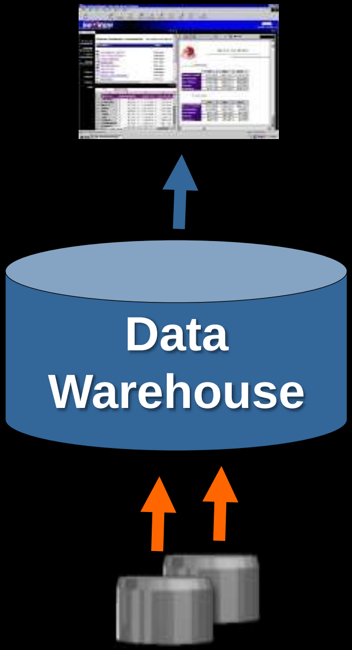
Architecture is not technology. It’s not a product you can buy.

Third Nature

An idea promoted by big data vendors

These do exactly the same thing:

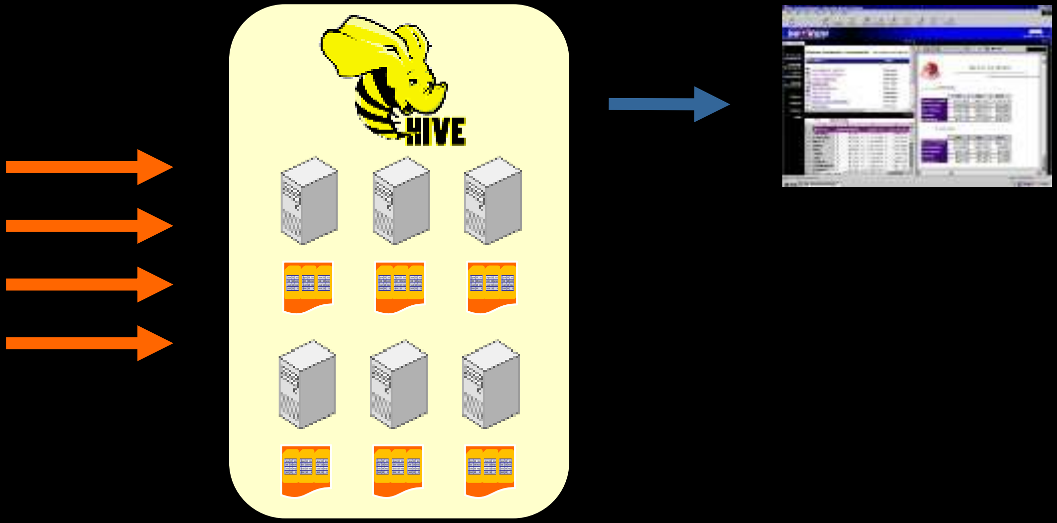




One is a set of technologies. One is an architecture.

Another idea promoted by big data vendors

This is a data lake



This is a set of technologies, not an architecture.

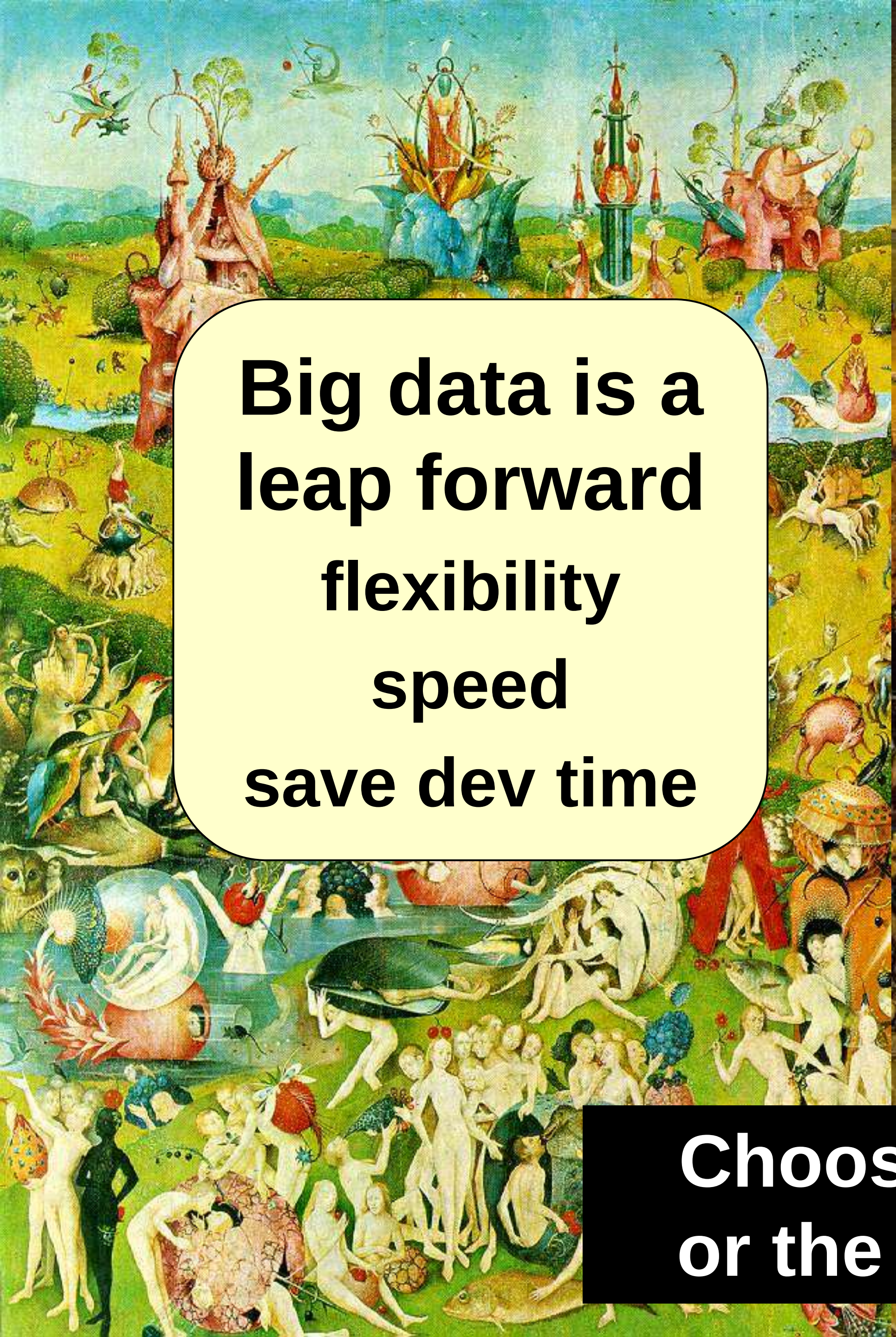
Data hoarding is not a data management strategy

Metadata and the Data Lake

If you want to use the lake for more than one application or set of analyses, you need to know:

- Who requested it? Why?
- Who’s using it? What are they using it for?
- What is this data? What are it’s parameters of use?
- Profile: size, age, refresh mechanism, rate, statistics
- Structure: the format (json, avro, table) and schema
- Form (the type of structure everyone forgets)
- How is it used? Ad-hoc, production pipeline, no use
- Where did the dataset come from?
- What are the security policies for this dataset? Expiration / archive policy?

Third Nature



Big data is a leap forward

flexibility

speed

save dev time

Big data is a fall backward

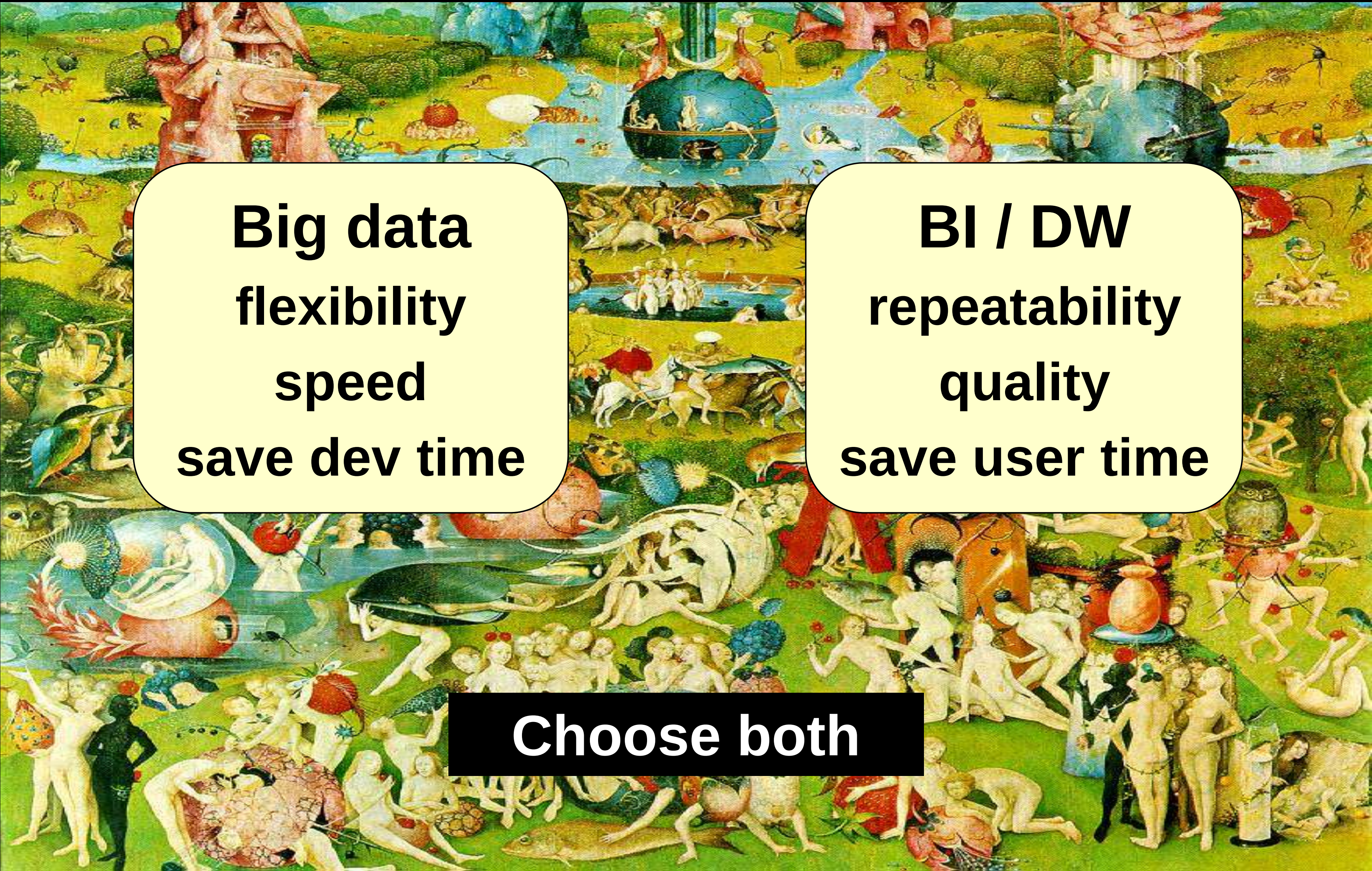
repeatability

quality

save user time

Choose one or the other.

What if we could reconcile two opposing ideas using ideas borrowed from other domains?



Big data

flexibility

speed

save dev time

BI / DW

repeatability

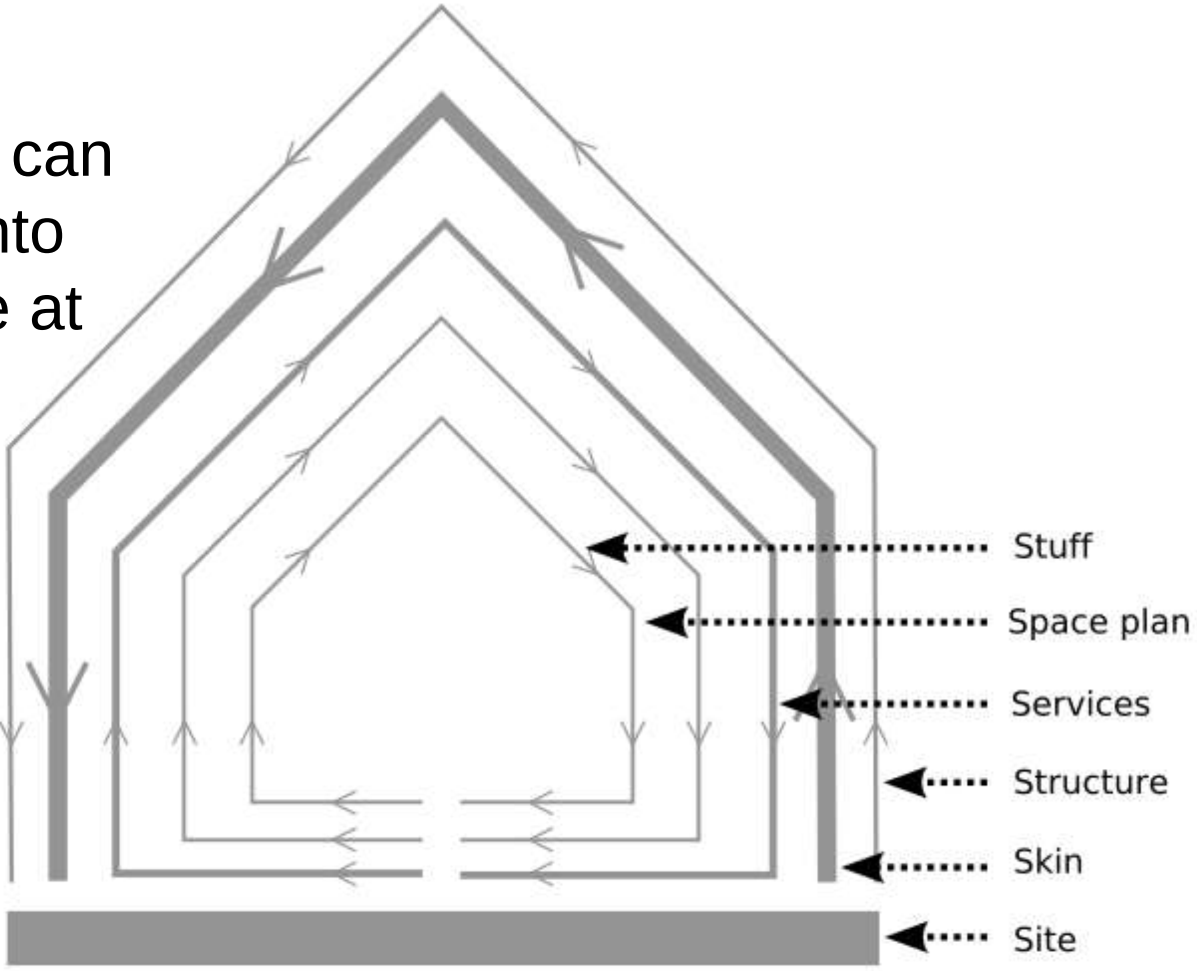
quality

save user time

Choose both

Separating fast from slow: pace layering and change in buildings

Complex systems can be decomposed into layers that change at different rates.



"How Buildings Learn", Stewart Brand

fast layers:

absorb change
propose solutions
learn

get all the attention: fixtures

slow layers:

integrate change
constrain options
remember

do all the work: plumbing

The focus for infrastructure needs to be on repeatability - where it can be supported

Architecture: components and layering

Components above: flexibility, repurposing, quicker change

Application

Layers below: stability, reuse, slow predictable change

Infrastructure

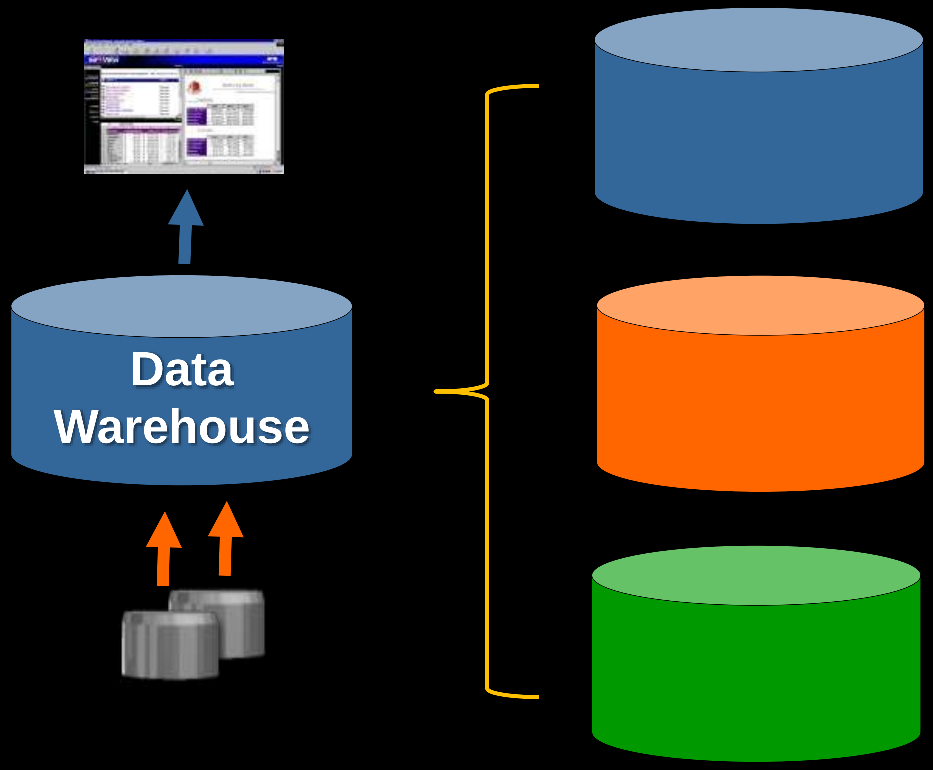
We thought this was the schema…

FOSTER'S
SANYO

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

Decoupling: design for isolation of unrelated change

A data warehouse has the right pattern, but the wrong implementation



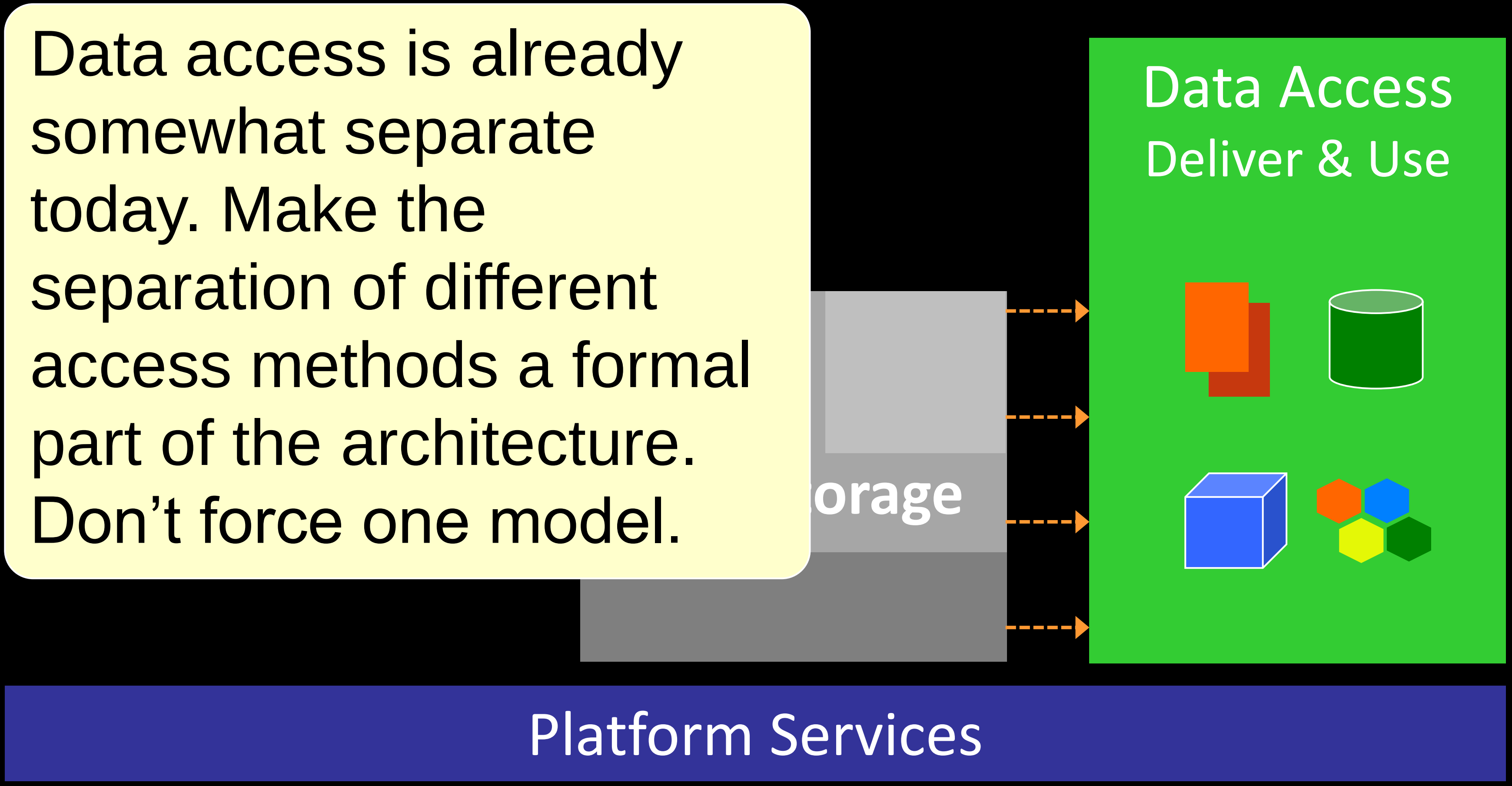
There are three things happening inside a DW:

- Data acquisition
- Data management
- Data delivery

Isolate them and their uses of data from one another.

Separate the component systems to isolate unrelated change

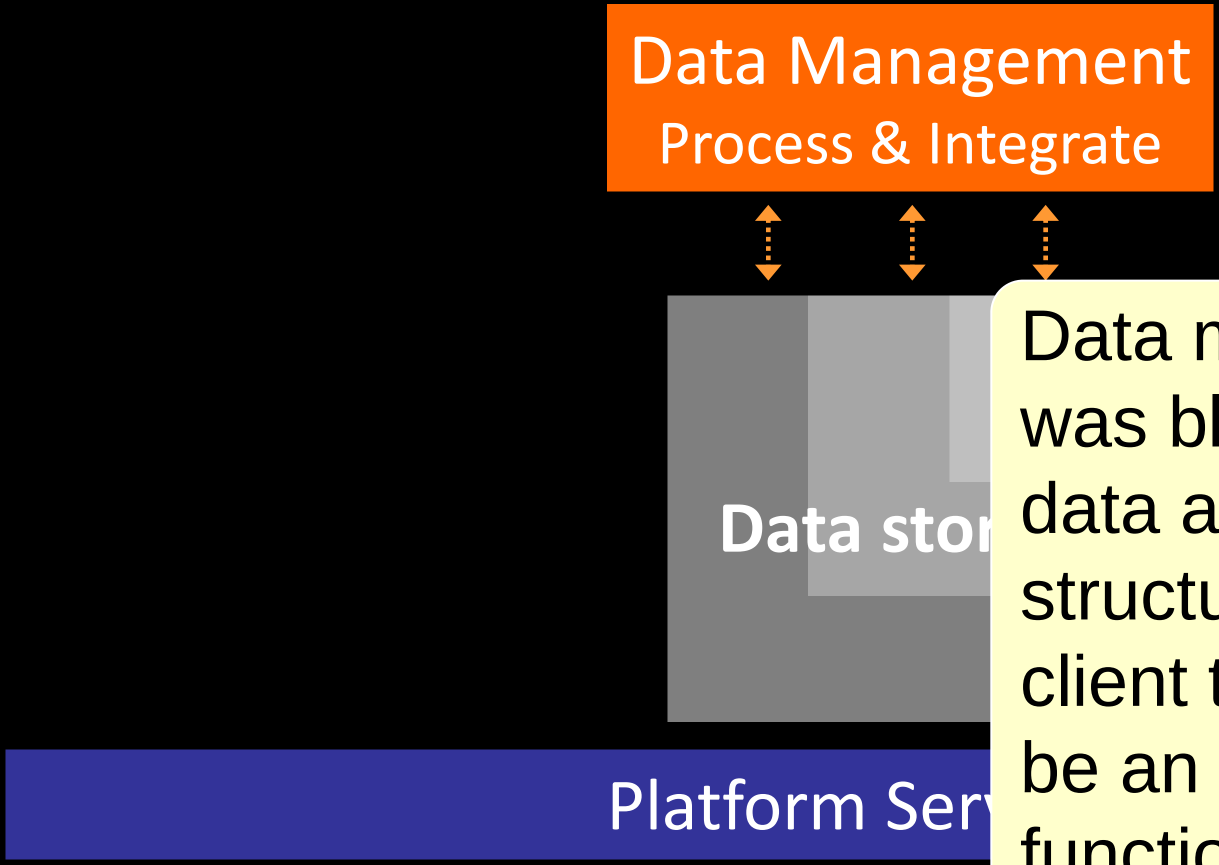
The goal is to decouple: solve the application and infrastructure problems separately



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.
orage
Data Access
Deliver & Use
Platform Services

This separates BI from other uses of data , 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

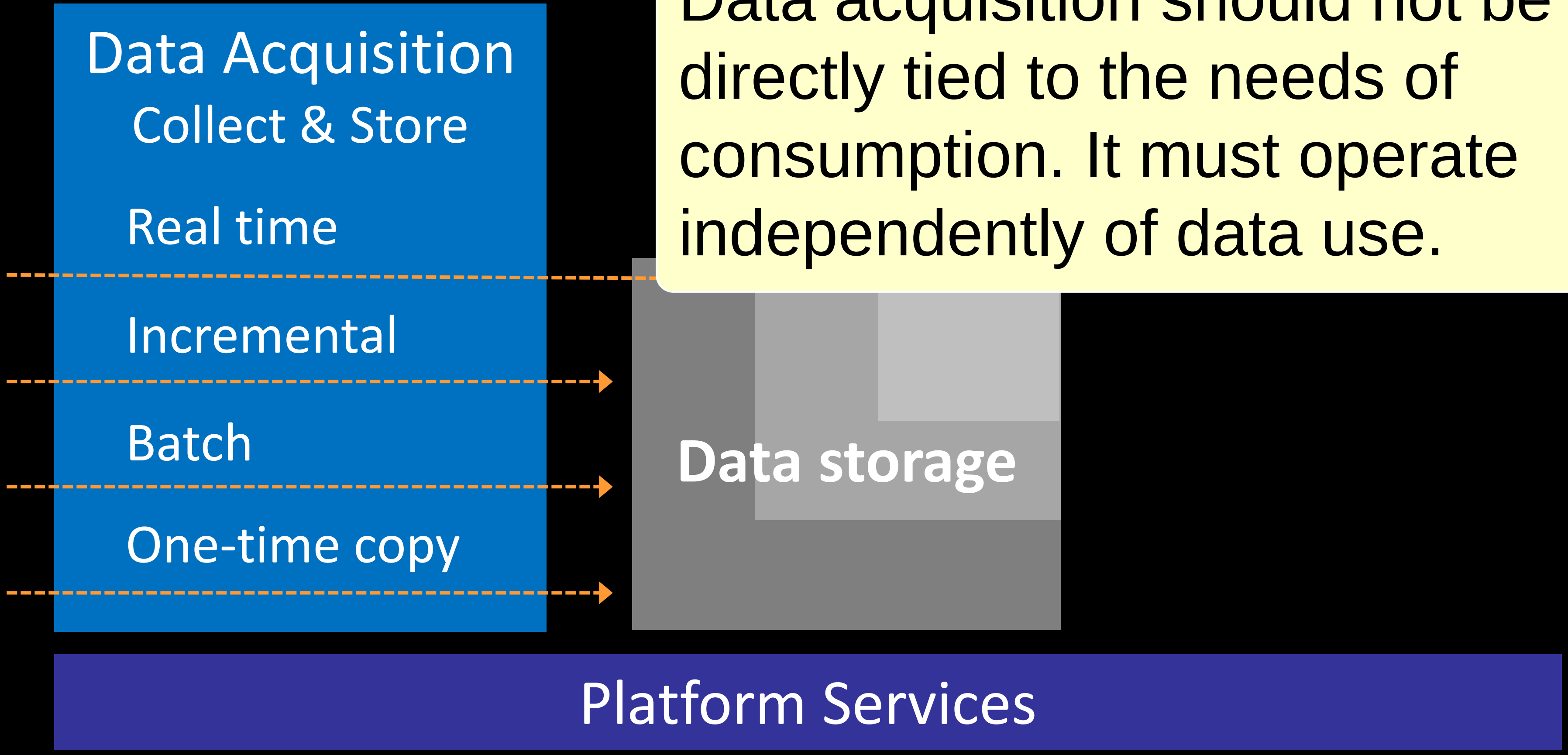


Data management was blended with both data acquisition and structuring data for client tools. It should be an independent function.

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

Third Nature

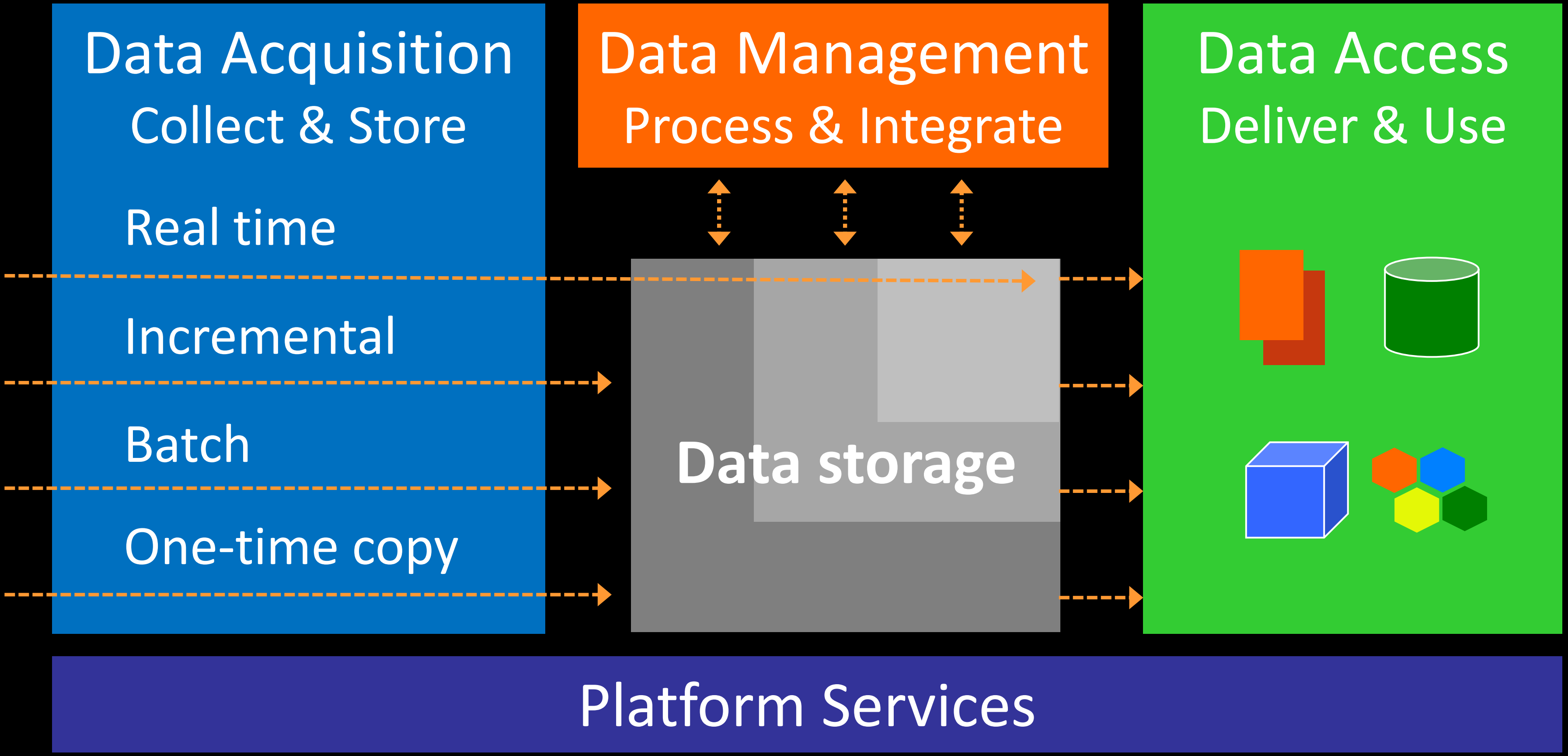
The goal is to decouple: solve the application and infrastructure problems separately



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 the data warehouse, and extends them



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

Splitting the architecture addresses three goals

Production

Creation, collection, storage of new data

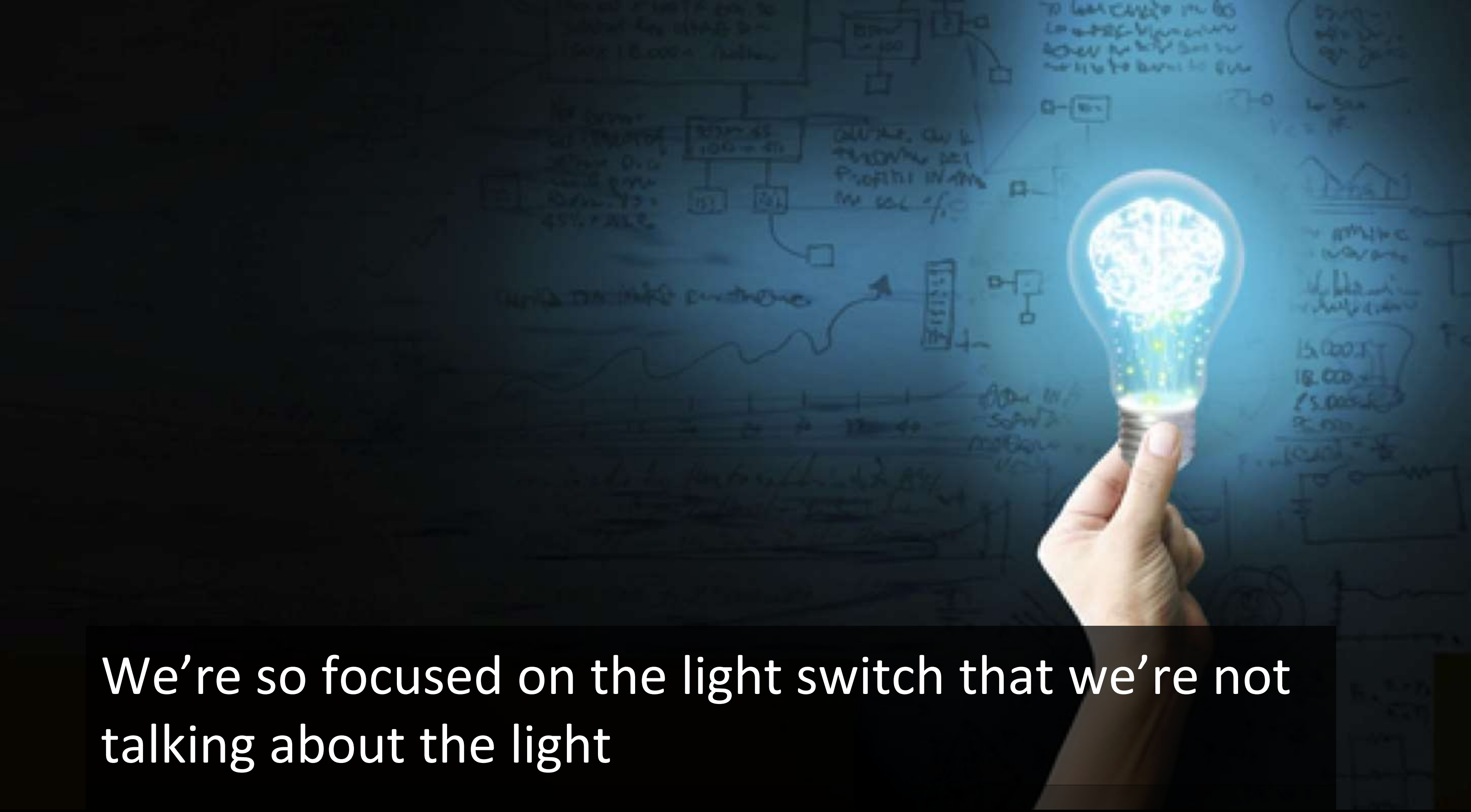
Distribution

Organization and distribution of data to multiple points of use

Consumption

Direct support of data use

Separation of concerns, coordination of process



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

DATA ARCHITECTURE

As with the code, decouple the data architecture

The core of the data warehouse isn’t the database, and the core of the data lake isn’t Hadoop it’s the data architecture that the tools implement.

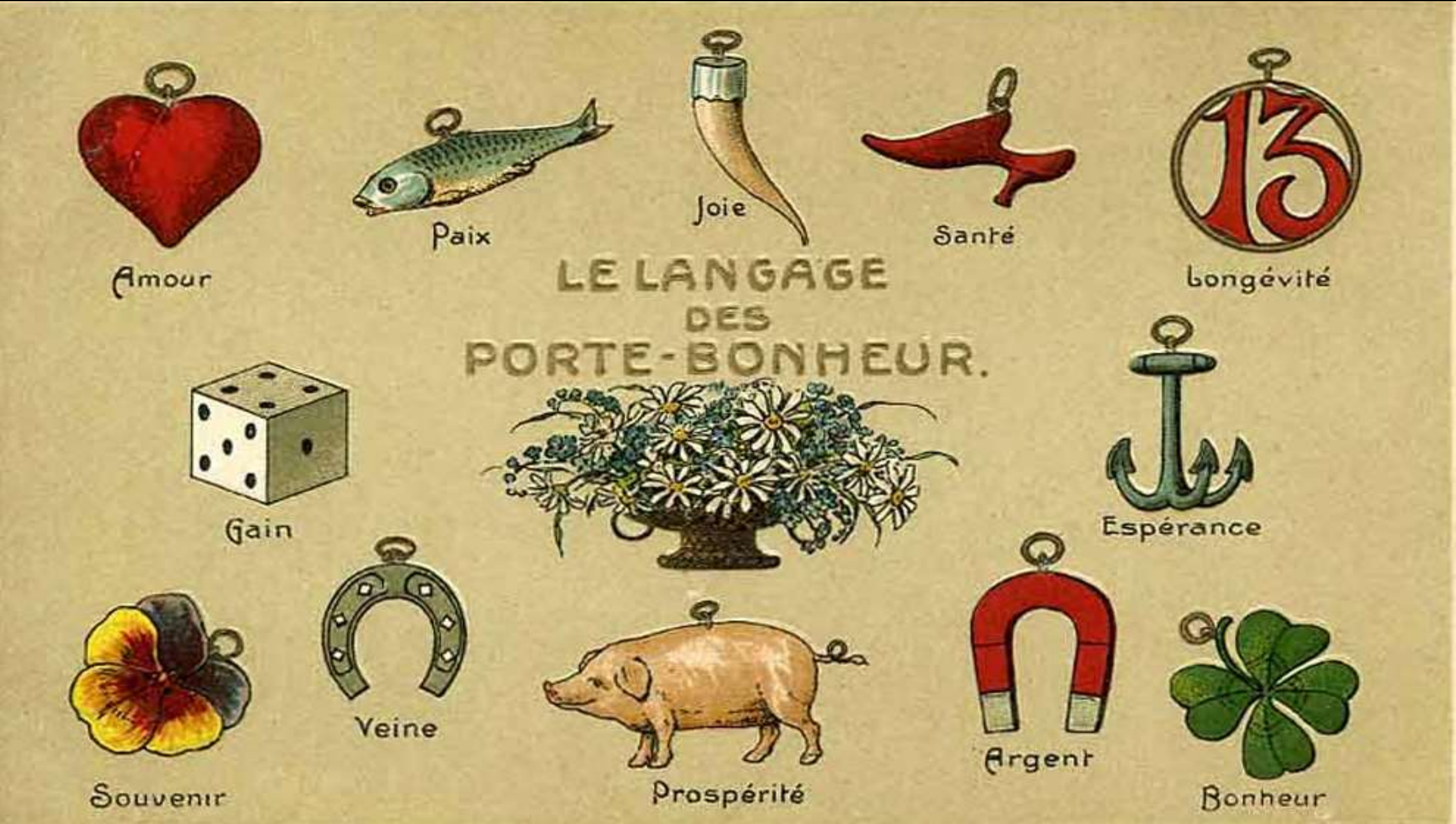
We need a data architecture that is not limiting:

- Deals with data and schema change easily
- Does not always require up front modeling
- Does not limit the format or structure of data
- Assumes a full range of data latencies, from streaming to one-time bulk loads, both in and out
- Supports different uses of the same data

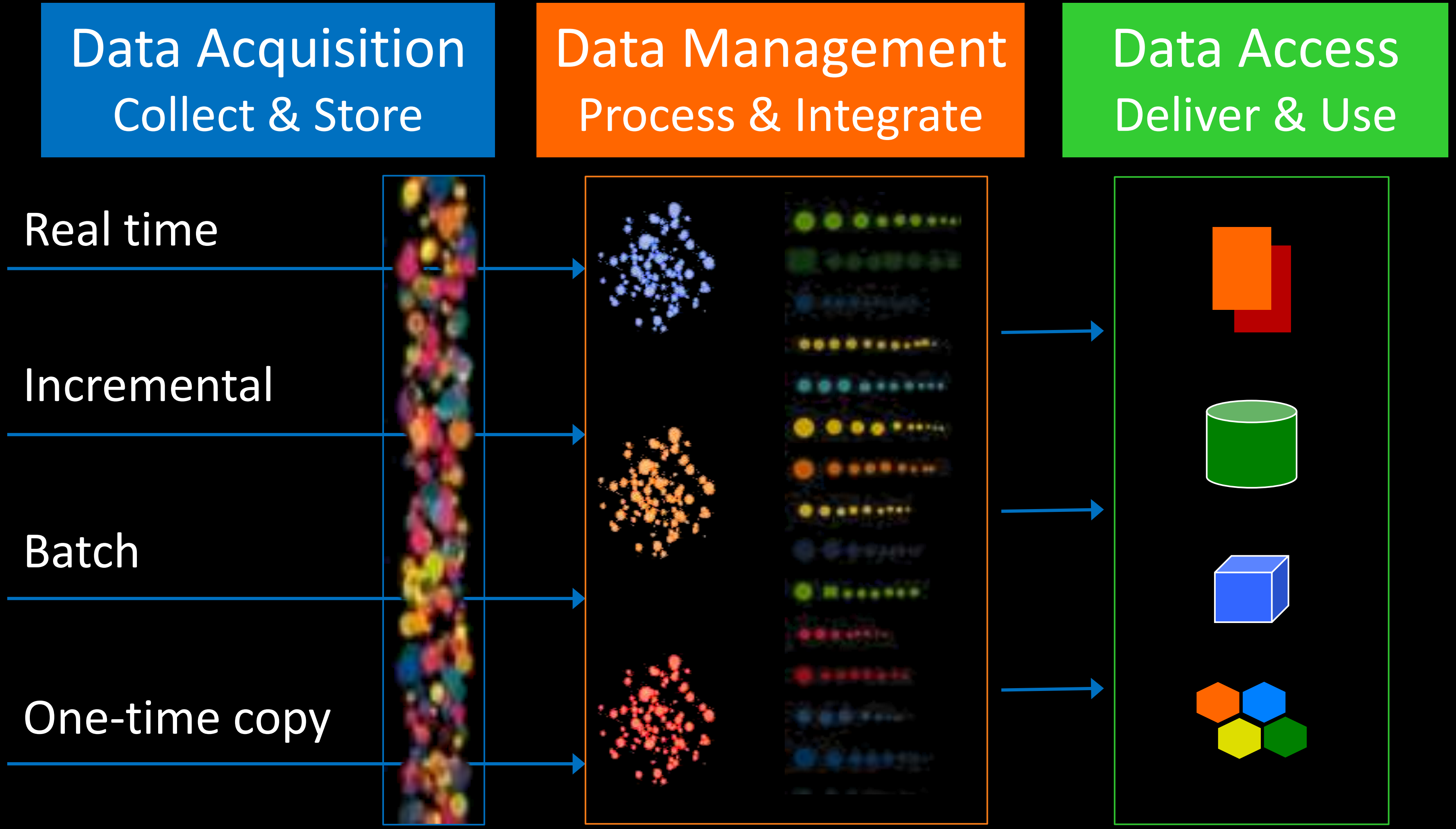
Third Nature

The new normal is distributed repositories

Data is collected in different places for different purposes. The architecture must acknowledge this



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



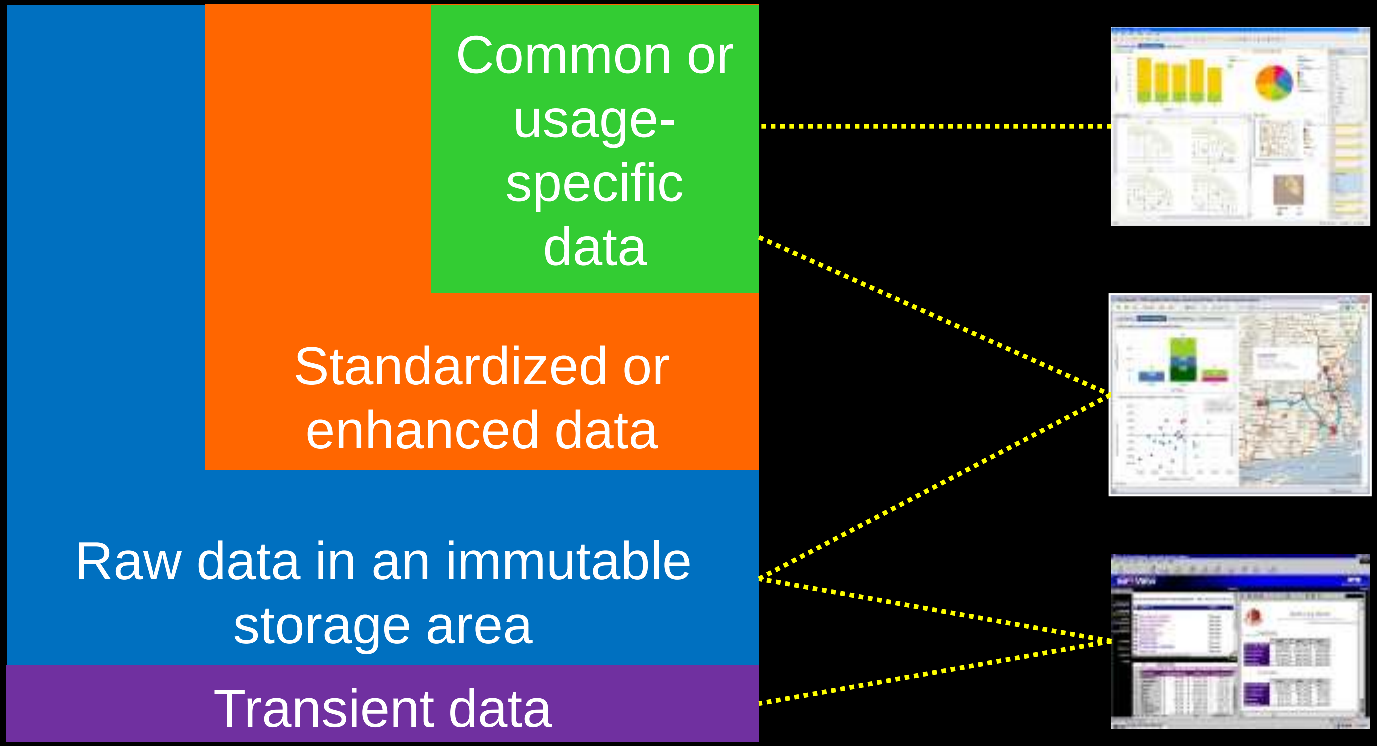
Data architecture is part of the mechanism for change isolation

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.


Common or usage-specific data
Standardized or enhanced data
Raw data in an immutable storage area
Transient data

Food supply chain: an analogy for data

Multiple contexts of use, differing quality levels

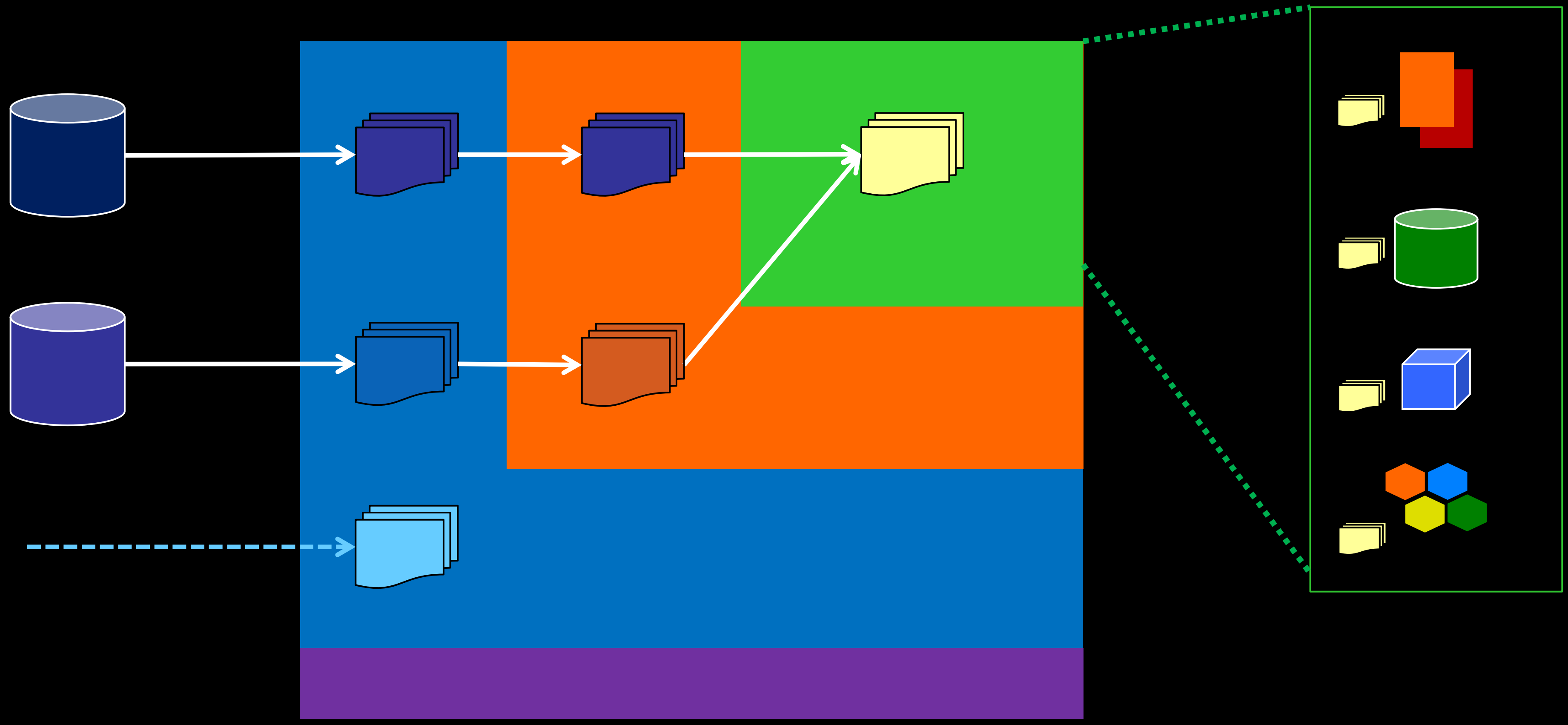
You need to keep the original because just like baking, you can’t unmake dough once it’s mixed.

Third Nature

Data can live in more than one place,

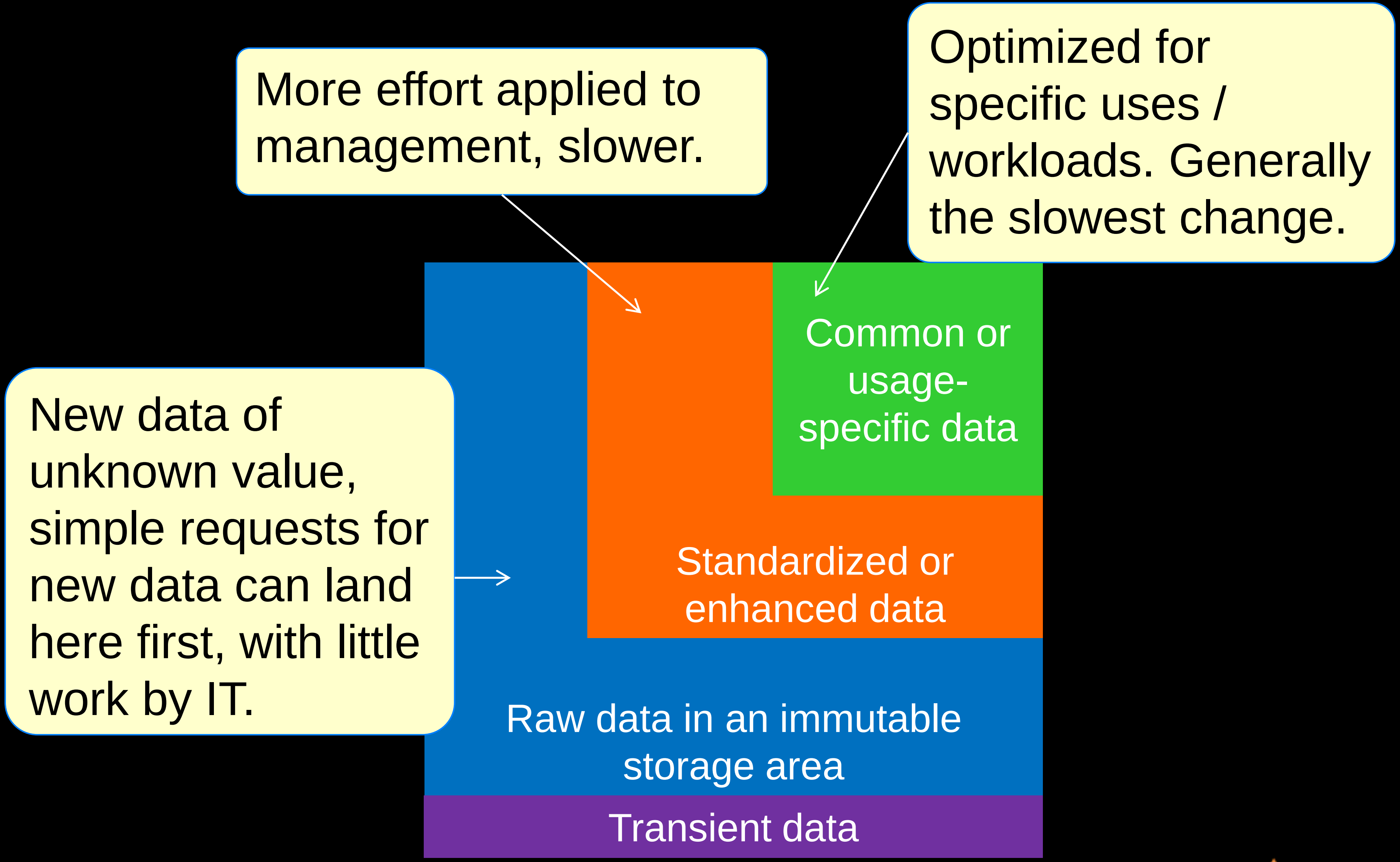
in more than one zone,

in more than one form

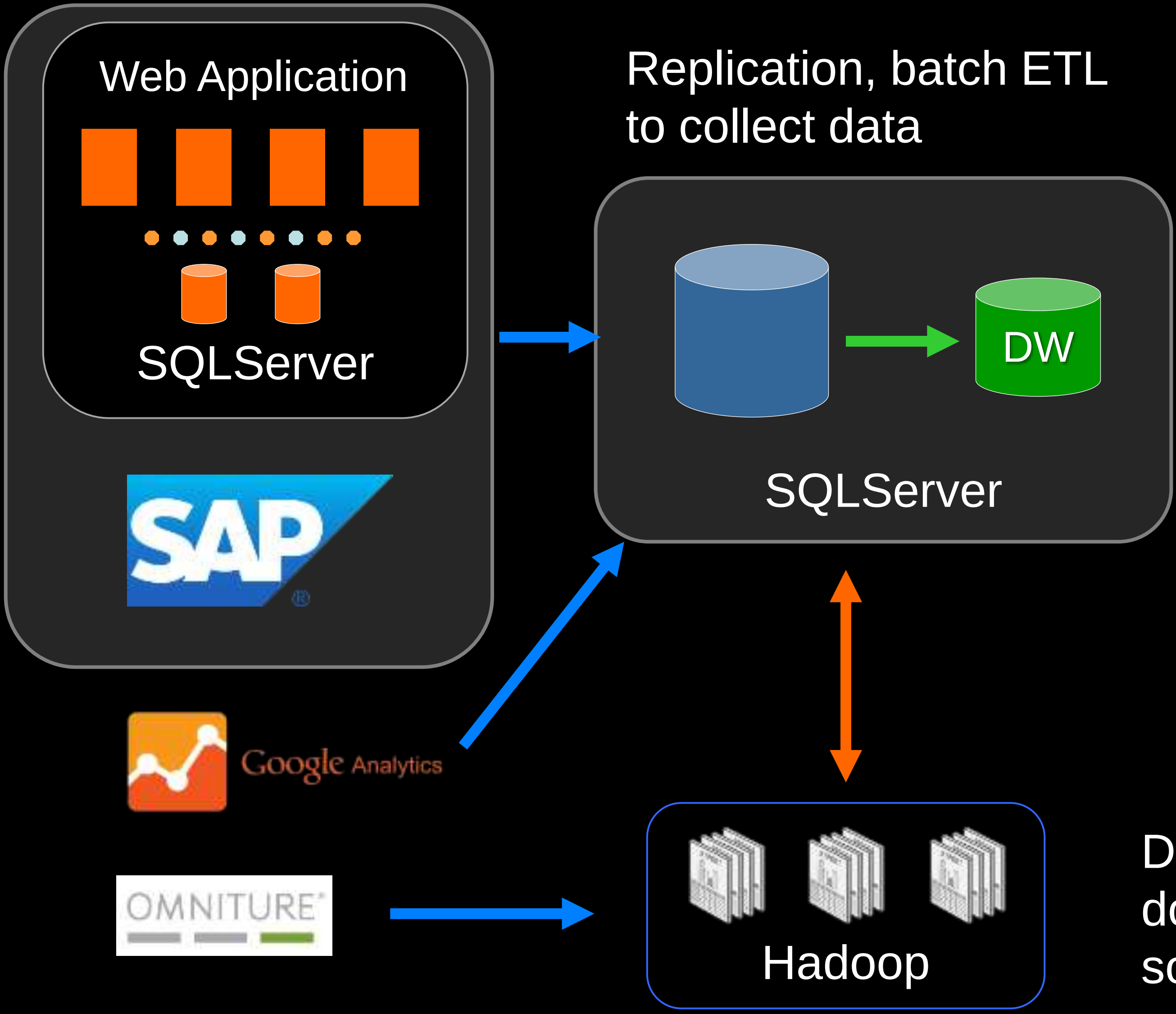


This is not a single global data model

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.
Common or usage-specific data
Standardized or enhanced data
Raw data in an immutable storage area
Transient data

Example: data environment, mid-size retailer



Replication, batch ETL to collect data

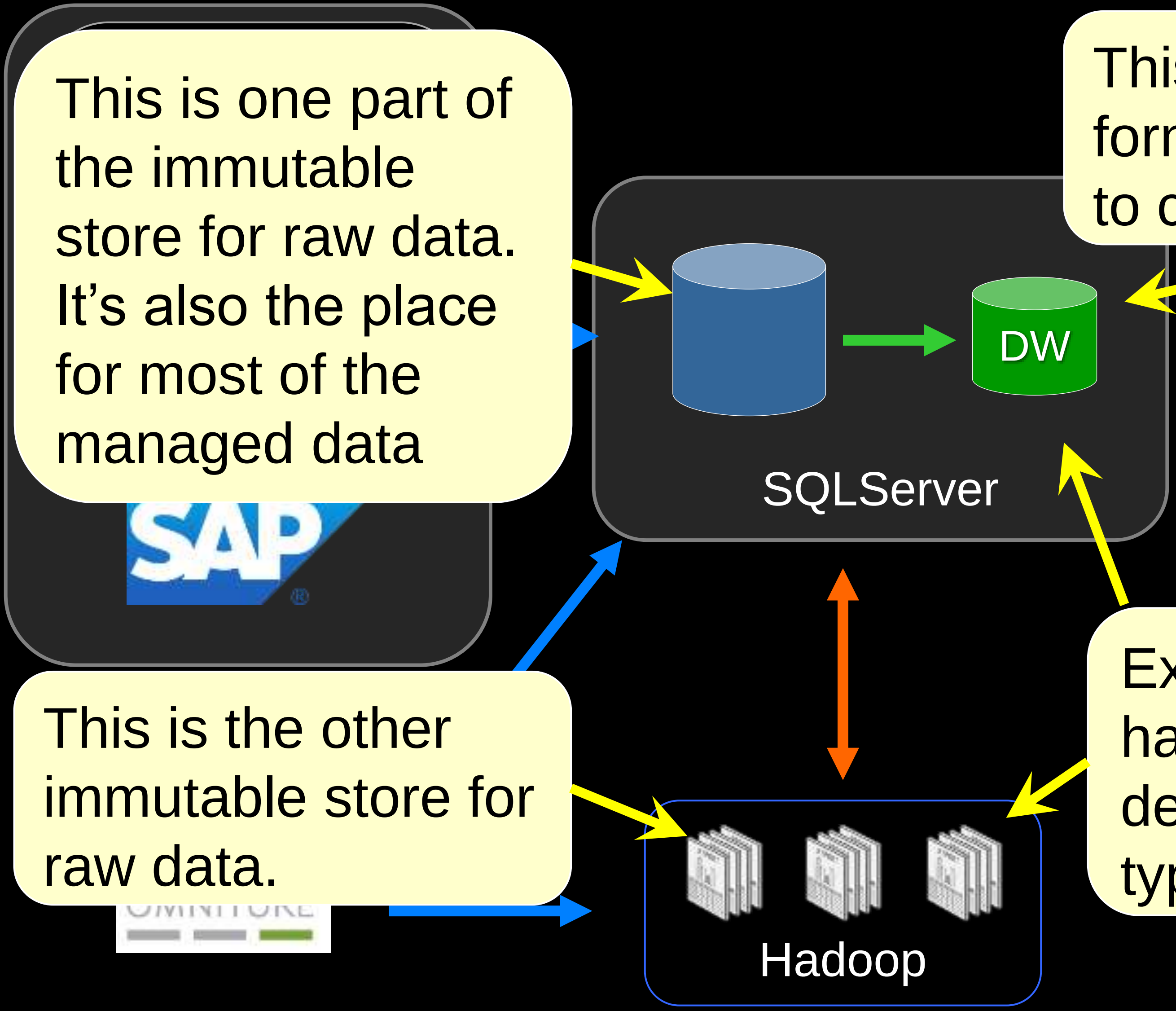
Persistent store, data warehouse on same DB in different schemas

Discovery work usually done in Hadoop, sometimes in database.

Log file fetch & load for clickstream, summaries sent to reporting env

Third Nature

Example: data environment, mid-size retailer



This is one part of the immutable store for raw data. It's also the place for most of the managed data

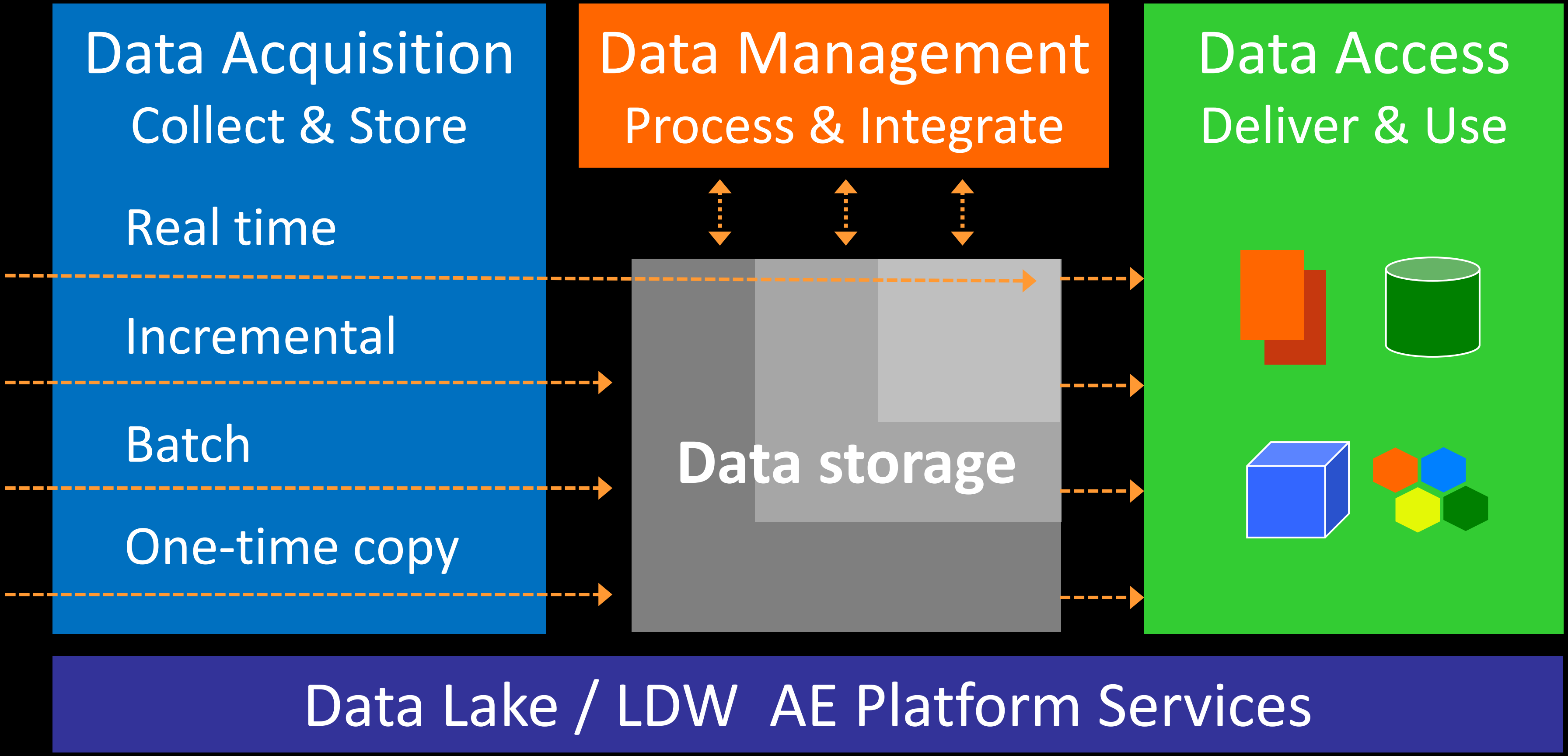
This is the data formatted for delivery to consumer via QRD

This is the other immutable store for raw data.

Exploratory work happens both places depending on the type of workload

New environment enables adaptive use of data

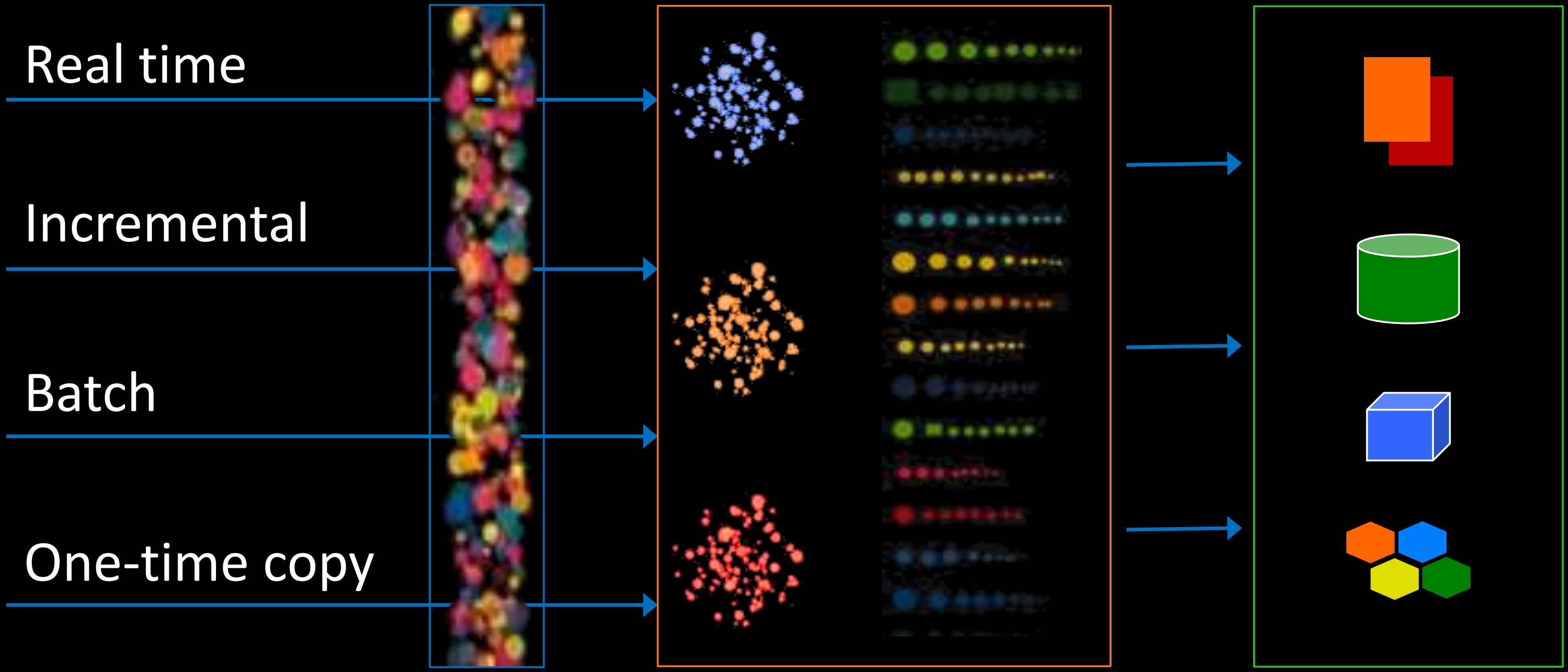
Data needs change, so you need a system for evolution as well as data infrastructure that provides stability.



You can't build this all at once. You need to grow it over time.

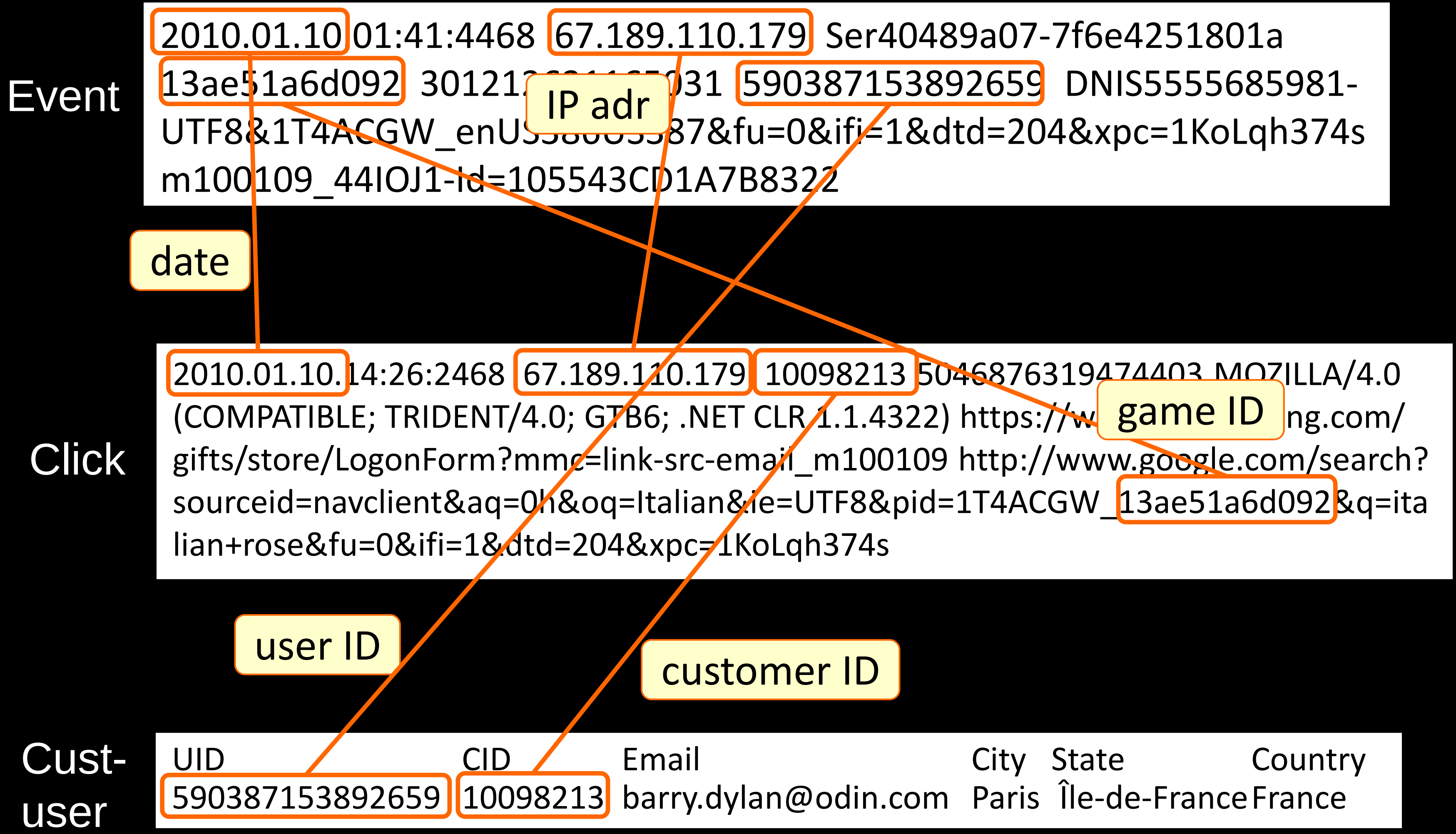
The data architecture has to manage independent datasets, key to ad-hoc flexibility is to manage keys not attributes*

Data Acquisition	Data Management	Data Access
Collect & Store	Process & Integrate	Deliver & Use
Schema on read	*Flexible, canonical data*	*Schema on write*



** the attributes have to be managed upstream, in applications*

Analysis is exploratory, dataset linking on demand
If you want to link datasets then you must manage the keys
You need canonical forms for common data too



Event

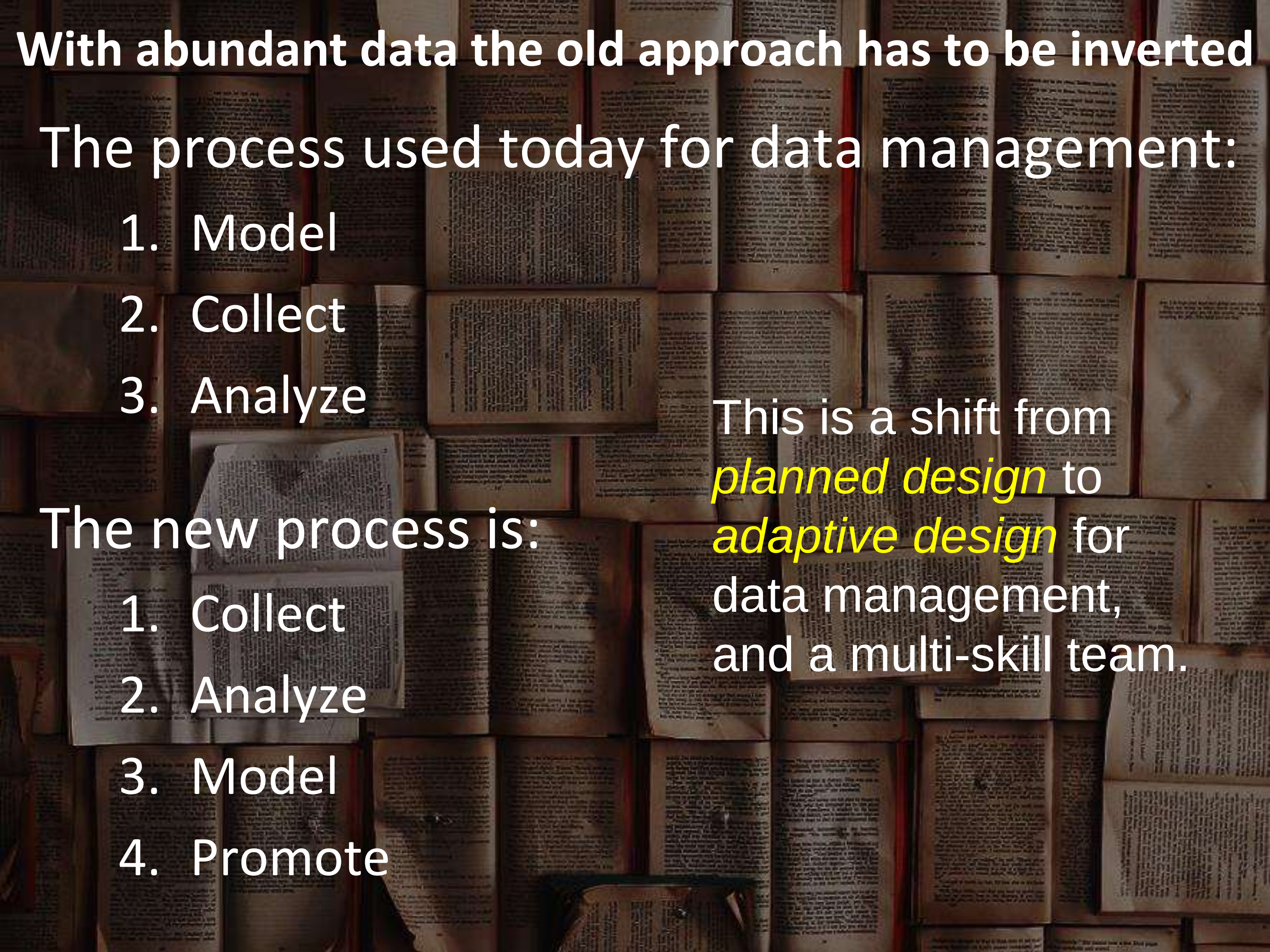
2010.01.10 01:41:4468 67.189.110.179 Ser40489a07-7f6e4251801a
13ae51a6d092 30121[illegible]31 590387153892659 DNIS5555685981-
UTF8&1T4ACGW_enU[illegible]87&fu=0&ifi=1&dtd=204&xpc=1KoLqh374s
m100109_44IOJ1-Id=105543CD1A7B8322

Click

2010.01.10.14:26:2468 67.189.110.179 10098213 5046876319474403 MOZILLA/4.0
(COMPATIBLE; TRIDENT/4.0; GTB6; .NET CLR 1.1.4322) https://w[illegible]ng.com/
gifts/store/LogonForm?mmc=link-src-email_m100109 http://www.google.com/search?
sourceid=navclient&aq=0h&oq=Italian&ie=UTF8&pid=1T4ACGW_13ae51a6d092&q=ita
lian+rose&fu=0&ifi=1&dtd=204&xpc=1KoLqh374s

Cust-user

UID	CID	Email	City	State	Country
590387153892659	10098213	barry.dylan@odin.com	Paris	Île-de-France	France



With abundant data the old approach has to be inverted

The process used today for data management:

1. Model
2. Collect
3. Analyze

The new process is:

1. Collect
2. Analyze
3. Model
4. Promote

This is a shift from *planned design* to *adaptive design* for data management, and a multi-skill team.

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 event stream data.

The data management role is changing

We no longer control so much as we guide

We aren't designers of

perfect governance and control,

we are designers of an environment that is resilient in the face of change

Help them to be the department of Yes

Instead of trying to model everything in advance, *collect it*.

Instead of trying to control change, *accept it.*

Instead of trying to control what people do with data, *focus on visibility*.

CC Image Attributions

Thanks to the people who supplied the creative commons licensed images used in this presentation:

pyramid_camel_rider.jpg - http://www.flickr.com/photos/khalid-almasoud/1528054134/
House on fire - http://flickr.com/photos/oldonliner/1485881035/
glass_buildings.jpg - http://www.flickr.com/photos/erikvanhannen/547701721
Building demolition - https://www.flickr.com/photos/gregpc/4429888820
peek_fence_dog.jpg - http://www.flickr.com/photos/webwalker/114998078/
donuts_4_views.jpg - http://www.flickr.com/photos/le_hibou/76718773/
chinese garden gate.jpg - http://www.flickr.com/photos/musaeum/420467301/

About the Presenter

Mark Madsen is president of Third Nature, a technology research and consulting firm focused on business intelligence, 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, a contributor to Forbes Online and on the O'Reilly Strata program committee. For more information or to contact Mark, follow @markmadsen on Twitter or visit http://ThirdNature.net

Third Nature

About Third Nature

Third Nature is a research and consulting firm focused on new and emerging technology and practices in analytics, business intelligence, information strategy and data management. If your question is related to data, analytics, information strategy and technology infrastructure then you‘re at the right place.

Our goal is to help organizations solve problems using data. We offer education, consulting and research services to support business and IT organizations as well as technology vendors.

We fill the gap between what the industry analyst firms cover and what IT needs. We specialize in product and technology analysis, so we look at emerging technologies and markets, evaluating technology and hw it is applied rather than vendor market positions.