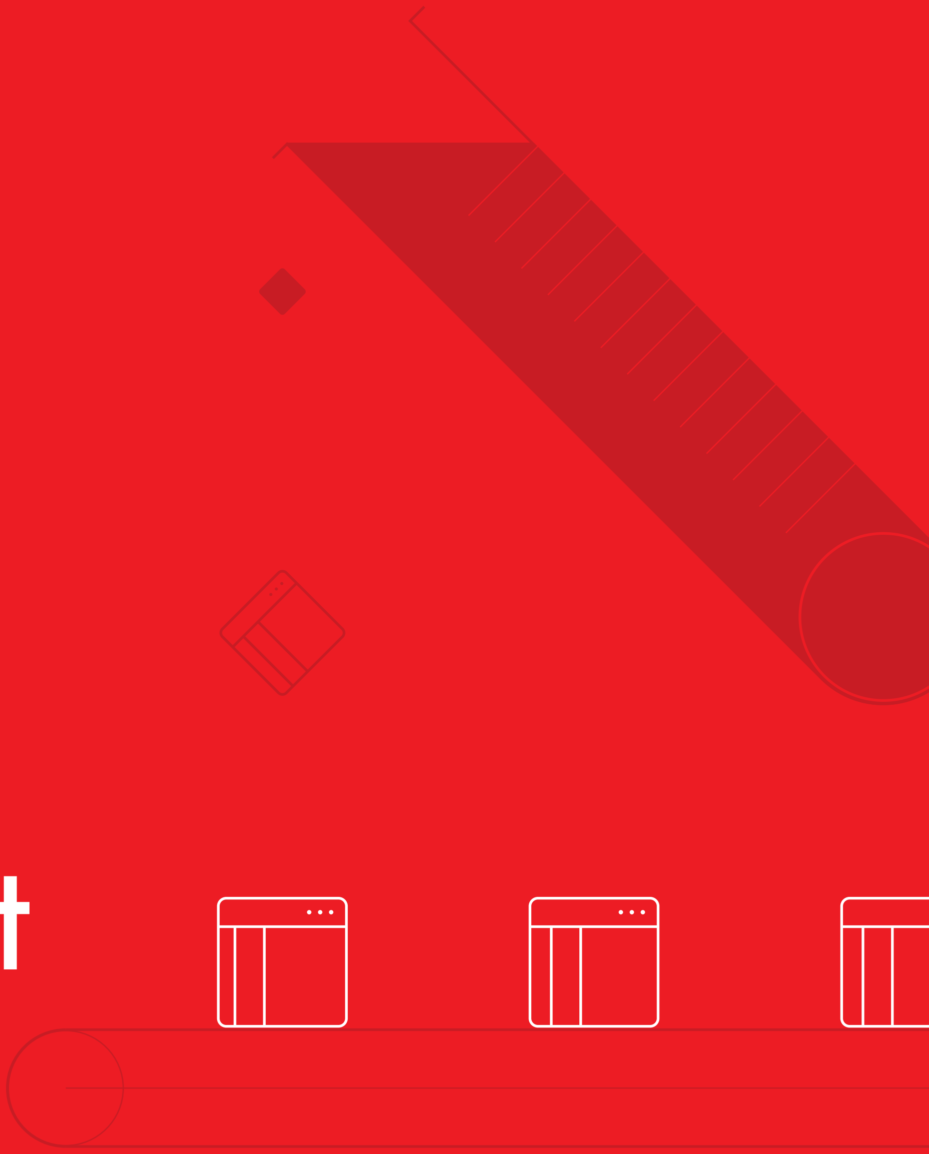
informatica

From Lab to Factory: The Big Data Management Workbook.



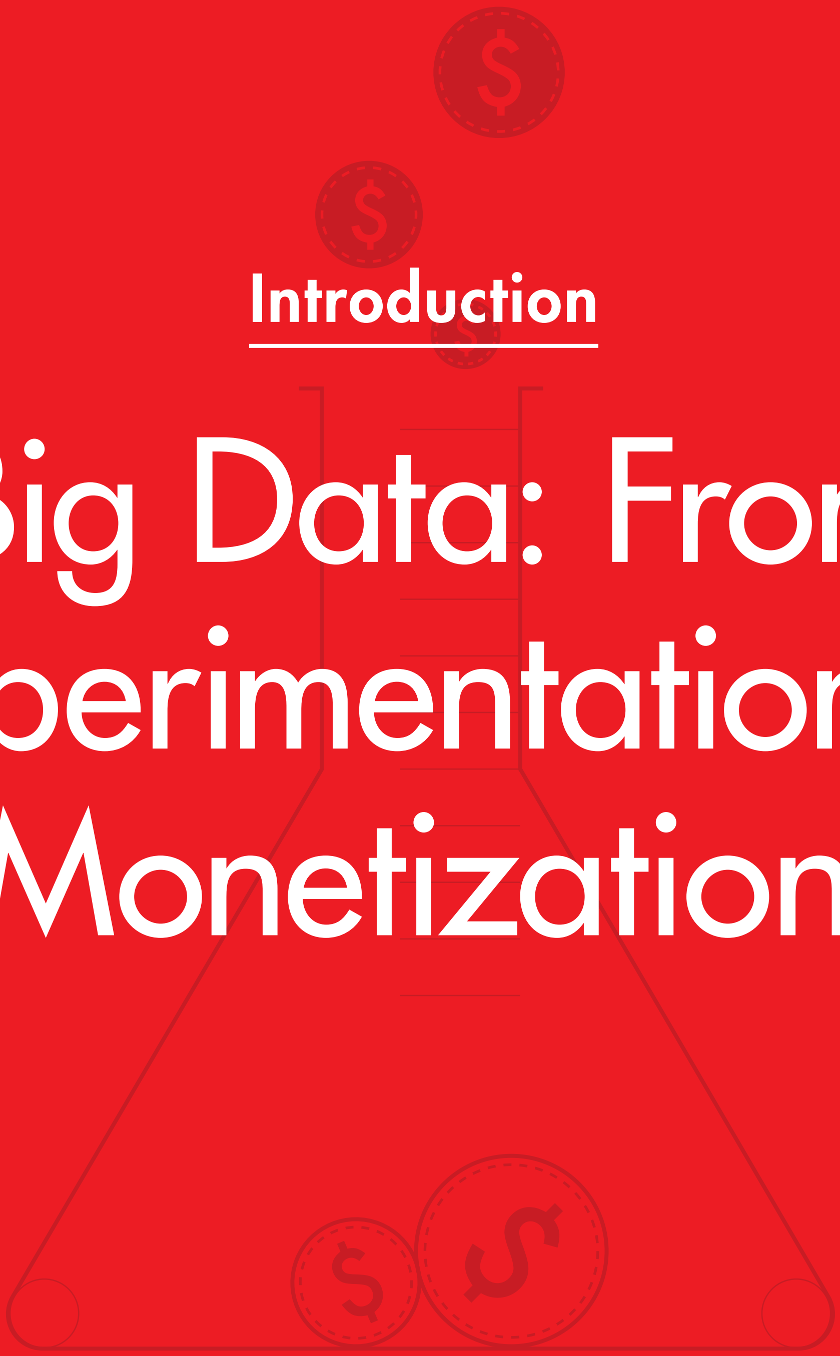
How to operationalize big data experiments in a repeatable way—and avoid failures.

Contents

Tip: Click on parts to jump to
the particular section you want

Introduction

Big Data: From Experimentation to Monetization.



Introduction

"Invention requires two things:

1. The ability to try a lot of experiments.

2. Not having to live with the collateral damage of failed experiments."[1]

Andy Jassy, SVP, Amazon Web Services

Big Data: From Experimentation to Monetization.

Beyond the hype, it's now clear that big data represents a serious opportunity for enterprise growth.

For one thing, it's increasing the effectiveness of existing business processes—fuelling more accurate and timely fraud detection[2], better predictive machine maintenance[3], and more relevant customer engagement[4].

But it's also enabling a great deal of innovation.

- Researchers at the **University of Pittsburgh Medical Center (UPMC)** electronically integrated—for the first time ever—clinical and genomic information on patients previously treated for breast cancer[5].
- Automaker **GM** uses telematics and broadband in its new models to fundamentally redefine the relationships between cars, drivers, and dealers[6,7].
- **Insurance provider Progressive** has been experimenting with big data to test innovations around its service, in one case rendering complex 3D images to record and monitor the condition of damaged vehicles[8].
- **A global mobile communications service provider** used advanced analytics to improve the 'take rate' of its underground Wi-Fi services with location-based marketing[9].

If your enterprise is serious about tapping into the *full* potential of big data, you need an enterprise architecture that's capable of serving two distinct purposes.

1. **Make data ready for analysis in a lab environment** where analysts can efficiently run meaningful experiments and pilots.
2. **Make data production ready in a factory environment** so it can be used for specific projects and products, as they're being operationalized.

The good news is that the requirements for these two purposes, while distinct, can be met by a common set of architectural components.

Even better, some early-movers have already built architectures that serve these dual capabilities.

The key lies in using a common set of data management standards and technologies to move projects from lab to factory environments in a smooth, predictable way.

This workbook will show you how.

Based on the insights, lessons, and best practices of these pioneers, we'll show you what to aim for, what to avoid, and what really matters when it comes to big data management.

Let's dive in.

Part 1

Riding the Elephant.

Confronting Hadoop's Limitations.

Investment and interest in Hadoop has never been higher[10]. That's great. Because Hadoop represents a crucial opportunity for enterprises to simultaneously reduce their data storage and processing costs, and leverage almost unlimited scale at the same time.

In no uncertain terms: Hadoop is essential to big data success.

But it's no silver bullet.

You can't expect to dump disparate data in and start analyzing it. In fact, the Hadoop ecosystem still has some non-trivial limitations:

Skills shortage

The biggest barrier to Hadoop success is still the significant shortage of skills for Hadoop and the myriad of sub-projects within Hadoop[11]. Not only does this make the staffing of your big data project more expensive, it can actually inhibit scalability from a project resource perspective.

And although you might be able to stand up a pilot project with a small group of experts, you can't efficiently operationalize and maintain all that code manually.

A lack of data management

Unstructured and schema-free data call for different methods. For instance, Hadoop does not retain much of the metadata that a traditional RDBMS contains. And it supports different storage and query paradigms such as Hive, HBase, Impala, Spark, etc. But even though it's constantly evolving, the Hadoop ecosystem lacks crucial data management capabilities around data governance, data quality, Master Data Management (MDM), metadata management, and support for ANSI-standard SQL.

Limited security

Even though Kerberos has established itself as a standard within Hadoop, authentication and authorization aren't enough to holistically protect a Hadoop implementation. This is especially true in production-grade environments that need to push and pull data from multiple systems and geographies.

As ever, there are new security authorization tools emerging (such as Sentry and Knox). But equally, more and more threats to data security are emerging. So it's essential that you look beyond the ecosystem itself for tools that can protect data even in the case of a security breach, such as data masking, encryption, and tokenization.

Confronting Hadoop's Limitations.

Constant change

The Hadoop ecosystem continues to change as new releases and new technologies emerge. But this is both an opportunity and an obstacle as far as enterprises are concerned.

For example, even though Spark is considered significantly faster than MapReduce for distributed processing[12], it does mean a lot of the hand-coded projects that made early bets on MapReduce now need to pivot.

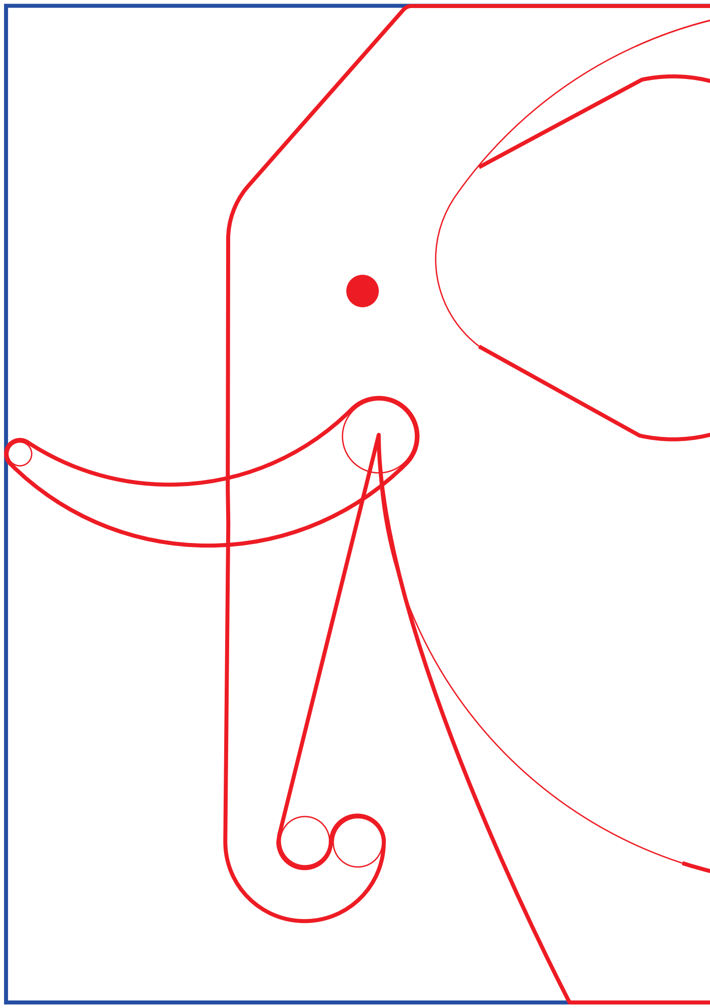
Spiraling costs

As we mentioned, the low starting cost of Hadoop and cloud storage make it relatively easy to stand up a pilot project. But at the point of operationalization, staffing, maintaining, and managing your environment is a whole other ball game.

There's a new realm of costs that come with scale and if you don't make smart moves around automation, you could end up trying to augment early hand-coding in an inefficient and expensive way.

To sum up: the Hadoop ecosystem is essential to the success of enterprise big data initiatives. But it is not sufficient to address all your big data challenges—especially when it comes to building a common architecture capable of serving both lab *and* factory environments.

In order to streamline, protect, and future-proof your architecture, you need to look beyond your data storage persistence layer.



Confronting the Limitations of Visualization and Analytics Tools

Like Hadoop, visualization and analytics tools have attracted a lot of attention in the context of big data. And like Hadoop, visualizations and analytics tools are only as useful as the data being fed into them. So while they're necessary to democratize access to data, they aren't sufficient to manage the data itself.

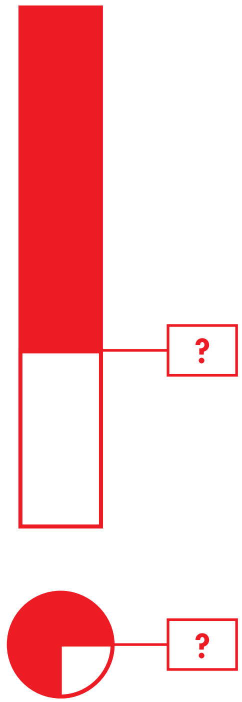
And without a reliable, repeatable, and efficient approach to big data management, you won't be able to get the most out of these tools.

Visualization tools can't visualize reliable insights if the data they're relying on contains too much noise. For instance, if you need to analyze the correlation between how much customers have bought and the cities they live in, you need to be able to trust the quality of your customer address data.

No matter how complex their analytical capabilities, analytics tools can't be relied on for accurate insights. For instance, you may use a complex attribution model to analyze which marketing channels deliver the best leads for sales. But without integrating data from all your marketing channels, you won't have a complete view of your prospects' buying journeys.

As they say, the only thing worse than no data is bad data. So it's important to remember that in the same way your data storage layer won't solve all your big data challenges, your analytics and visualization layer won't be able to solve all your challenges either.

The crucial layer between these two is big data management.



The Importance of Big Data Management.

It shouldn't come as a surprise that the key to big data success is smarter data management. But in all the excitement around Hadoop and new data sources, it's worryingly common for teams to rush in without an efficient way to clean, connect, and secure all their data.

So it's important that you define a clear and comprehensive data management strategy before you get started, so that you can:

Streamline data quality and governance
The primary function of your big data infrastructure must be to deliver data that is fit-for-purpose in a repeatable, efficient, and reliable way. To that end, you need to be able to integrate, cleanse, and master data based on the differing needs of your lab and factory environments.

In a lab environment, that will mean provisioning data and access to data that's 'good enough' to validate models and test hypotheses. In a factory environment, that will mean provisioning certified, high quality data that can be relied on for core business processes.

Leverage your existing talent
By employing the data management tools, standards, and interfaces your developers and data management professionals are already used to, you can leverage the skills they already have to clean, integrate, parse, and transform your data at scale.

But even beyond your data management talent, self-service data preparation tools allow your analysts and scientists to access and interrogate datasets *without* having to learn new skills like Java.

Make sure data-centric security is baked-in
As mentioned, in a factory environment, authentication and authorization aren't enough to secure all your data. So it makes sense to assess your security needs *before* your clusters become a free-for-all.

Beyond perimeter control, you also need to be able to secure your data in the event a security breach takes place. That means leveraging risk-centric data profiling, data masking, encryption, and tokenization.

Use your metadata strategically
By leveraging a global repository of metadata, you ensure your people have a reusable set of rules and business logic to deploy across both lab and factory environments. That increases developer and analyst productivity because it means they don't have to constantly reinvent the wheel. And it allows you to distribute standards and best practices across projects and environments.

The Importance of Big Data Management.

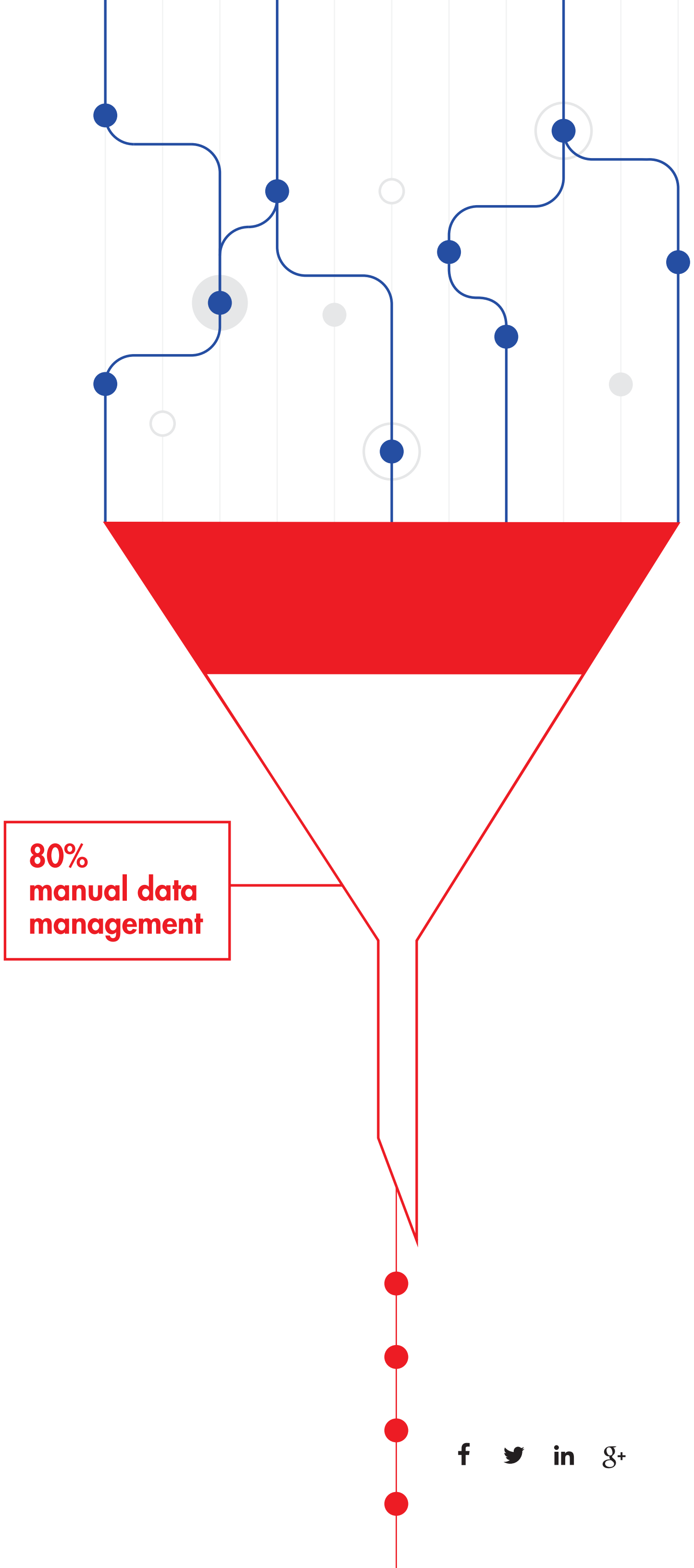
In fact, metadata management allows you to maintain standards, transformations, and best practices *regardless of changes to the runtime environment*.

So even if you wanted to, say, replace MapReduce with Spark, you wouldn't have to start from scratch because you wouldn't lose any of your rules, transformations, or logic.

Avoid expensive manual coding
When you're dealing with huge volumes of data, you can't expect to eyeball all your data, and you shouldn't try to manually manage all of it either.

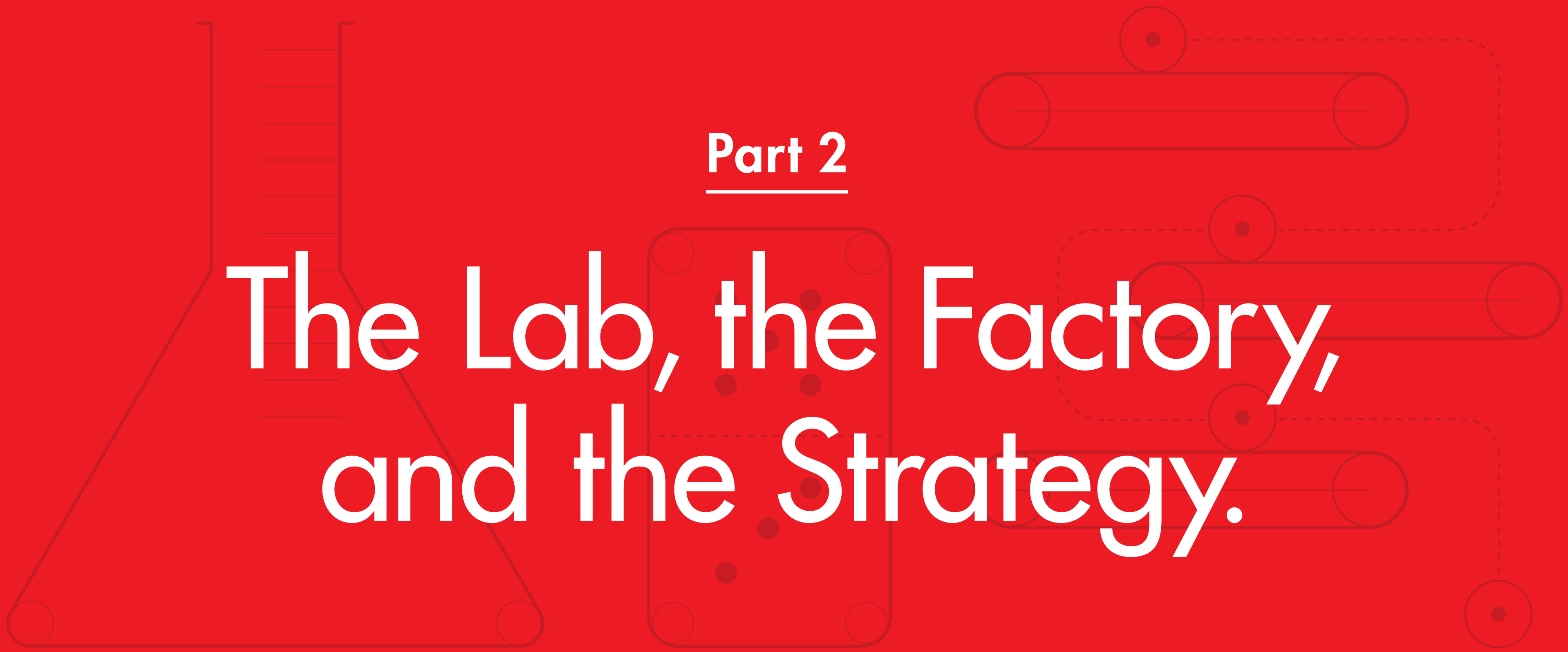
So data integration, governance, quality, security, and mastering are even more important in the context of big data. And automating these functions saves you time and money at the point of operationalization.

First, because it ensures your developers don't have to go through 20 pages of code every time they need to add or change a transformation. And second, because your analysts and scientists don't have to spend as much as 80 percent of their time[13] on manual efforts for accessing, cleansing, and wrangling data.



Part 2

The Lab, the Factory, and the Strategy.



Big Data Management: Two Sides of the Same Coin.

When companies treat big data management as a one-off, hand-coding exercise, they end up spending all their time maintaining and rewriting all that code as things change. A more strategic approach is to build an automated and standardized data management infrastructure that can support both lab and factory environments.

That is, to build a single platform that supports, on one side, self-service autonomy for analysts and scientists so they don't have to wait for IT to provision fully certified data to iterate through their hypotheses and validate their models.

And on the other side, flexible, scalable, and maintainable data pipelines so that IT can efficiently deliver data "fit-for-purpose" directly to end-users that meet business SLAs (service level agreements), detect data quality problems early, perform continual data audits, leverage scorecards and dashboards, and use existing ETL developers to build data pipelines in Hadoop.

We'll get into the specific requirements for both lab and factory environments in the coming sections. But for now, let's look at why it makes sense to build a common infrastructure that can support both sides.

It consolidates your data management investment
The same technology being used to provision data to analysts can be used at the point of operationalization. This rationalizes your technology stack and ensures that IT and the lines of business aren't spending twice for the same core capabilities. It also ensures that the data everyone needs isn't being stored and managed in silos.

It gives IT the benefit of shared metadata
The only way you can provision, profile, manage, and govern big data at scale across the enterprise is if you manage your metadata strategically. A global repository of shared metadata lets IT define enterprise-wide best practices and then distribute them across the company. And it gives developers and analysts a pre-defined set of reusable logic, transformations, and rules to apply quickly.

It makes it a lot easier to add new staff
When you rely on a small group of experts for a hand-coded approach to data management, the lack of standardization makes it incredibly hard to augment and even maintain that code when the team grows. By leveraging a common set of data management 'rules & tools', you onboard new employees and new data quickly and easily. You also avoid having to rely on rare, expensive big data talent to scale your infrastructure.

Building a Big Data Lab.

Before you can turn big data into operational assets—as data products or even as analytical environments—you need to understand what's possible and what makes most sense for the business. So before you can produce any meaningful innovation, your analysts and scientists need to have the freedom to experiment.

1. An HR analyst should be able to rapidly access performance data on new hires and hiring managers to see if it shows who in the company is best at identifying the right talent.

2. A sales analyst should be able to see if there are any correlations between social media activity and sales activity in different regions.

3. A marketing analyst should be able to see if there is any significant campaign sales lift due to new and targeted marketing campaigns.

4. A financial analyst should be able to quickly compare 'good enough' cost projections against sales forecasts to make an educated estimate.

Once, and really only once, they've validated their hypotheses and determined that their experiments are worth repeating, should IT start the process of operationalizing them.

The needs of the lab.

A lab environment for big data experiments has some distinct requirements:

People
In a big data lab, analysts and scientists need the freedom to access any data to autonomously test as many hypotheses as they need. IT's role, therefore, should be to provision 'good enough' data rapidly, give them the basic tools they need to visualize and share analyses, and 'get out of the way'.

This dynamic is crucial if you're going to foster experimentation. And it necessitates a move away from the paradigm where IT spends weeks preparing data while analysts wait.

Process
Until now, if analysts needed to run data experiments with traditional enterprise technology stacks, the process for IT has looked something like this:

- The analyst has an idea
- IT figures out where the data lives and how to access it
- IT talks to business users about how they intend to use it
- IT defines business rules to transform the data
- IT builds some schema for the data
- IT writes some ETL to transform and load it

Not only was this an expensive process that left business users waiting and in the dark for weeks at a time, there was actually no guarantee IT would get it right the first time. Needless to say, this isn't conducive to an environment of experimentation and rapid analysis.

What's needed is a big data platform that can handle the mass ingestion of multiple data sources and support self-service data discovery. So the process actually looks a little more like this:

- The analyst has an idea
- IT understands where to get the data
- IT loads it into the platform (if necessary with an initial schema)
- The analyst can go ahead and do their own self-service data discovery

The needs of the lab.

Technology

When it comes to building a big data lab, you need to ensure your big data platform supports rapid and easy experimentation. That means you need infrastructure that's built to support:

1. **Quick ingestion:** from multiple sources with pre-built connectors to streamline access to high volumes of data from new and old data sources (with real-time streaming for low-latency data).

2. **Self-service autonomy:** so analysts can easily interrogate the data without getting caught up in hours of data wrangling, using semantic search, automated profiling, self-service data discovery, and quick visualization tools.

3. **Proactive data management:** an analyst-driven lab environment that supports experimentation is useless if all the experiments are based on bad data. So you also need to make sure IT has the tools it needs to quickly integrate, profile, and audit data.

It's important that you also take into account the user experience you're enabling. Since analysts use Excel more than any other tool for analysis, that means equipping them with spreadsheet-based interfaces that are just as easy to use.

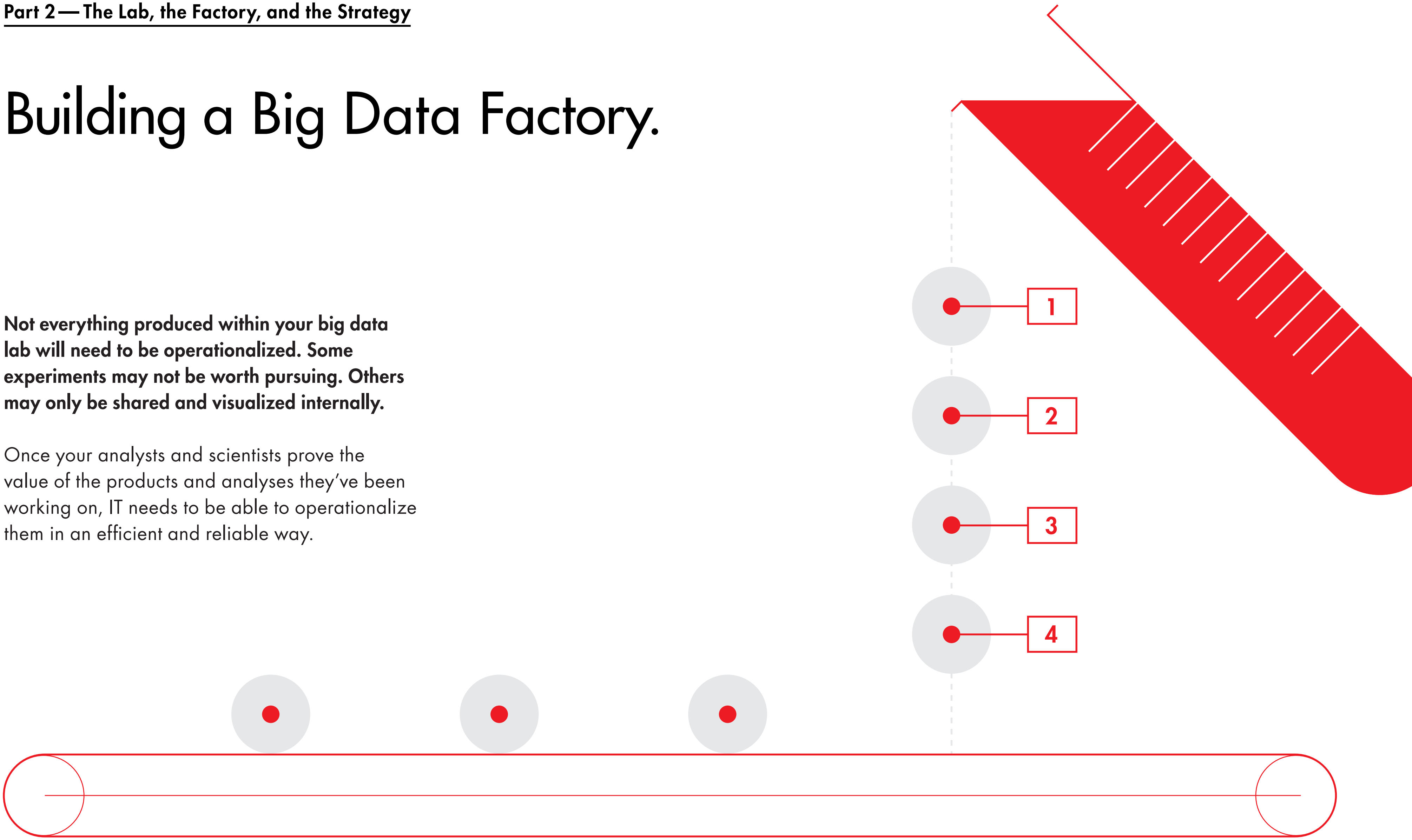
The aim should be for your analysts to be able to blend and merge data on-the-fly while also building complex analytical models that can take in new data.

Of course, when it comes to supporting both a lab and a factory environment, you need your data management infrastructure to do a whole lot more. Let's look at the more specific requirements for building a big data factory.

Building a Big Data Factory.

Not everything produced within your big data lab will need to be operationalized. Some experiments may not be worth pursuing. Others may only be shared and visualized internally.

Once your analysts and scientists prove the value of the products and analyses they've been working on, IT needs to be able to operationalize them in an efficient and reliable way.



The needs of the factory.

The factory side of your data management infrastructure is all about ensuring IT has the tools and investment it needs to build out what your business end-users and customers need.

People

The lab side of your data management infrastructure is all about empowering your analysts with a little help from IT to run their experiments on their own. But the factory side is about automating a data supply chain that turns the data and insights discovered in the lab into business value.

Here the role of the analyst becomes more about guiding IT and making sure what's being built actually fulfills the business need.

IT's role on the other hand is all about managing engineers and developers to optimize the capital investment required to productize, monetize, or operationalize the data assets.

Process

In the lab environment, you want your analysts to be able to build their own data pipelines on the fly. So if you give them tools that self-document transformations and data flows, you'll speed up the process of IT provisioning these data pipelines in a production environment. IT can just engineer based on the logic and objects used in the lab environment.

No two projects being operationalized are likely to be the same. So for now let's suffice it to say that here, IT needs to focus on managing DevOps and production support to make sure data is ingested, cleaned, and secured in a reliable, repeatable, and easily maintainable way.

The needs of the factory.

Technology
Most of the hype around big data has been around the data storage and data analysis layers. But even though technologies like Hadoop and data visualization (e.g. Tableau, Qlik) are crucial to success with big data, there is one essential layer in between them that simply cannot be ignored—the big data management layer.

This layer can be broken up into three distinct pillars to support both your factory and lab environments.

1. **Dynamic, at-scale big data integration:** to create flexible and scalable data pipelines that connect new data sources and empower analysts to create richer models.

2. **Collaborative, end-to-end big data governance:** to ensure both analysts and IT are confident that the data they're using is clean, complete, and timely.

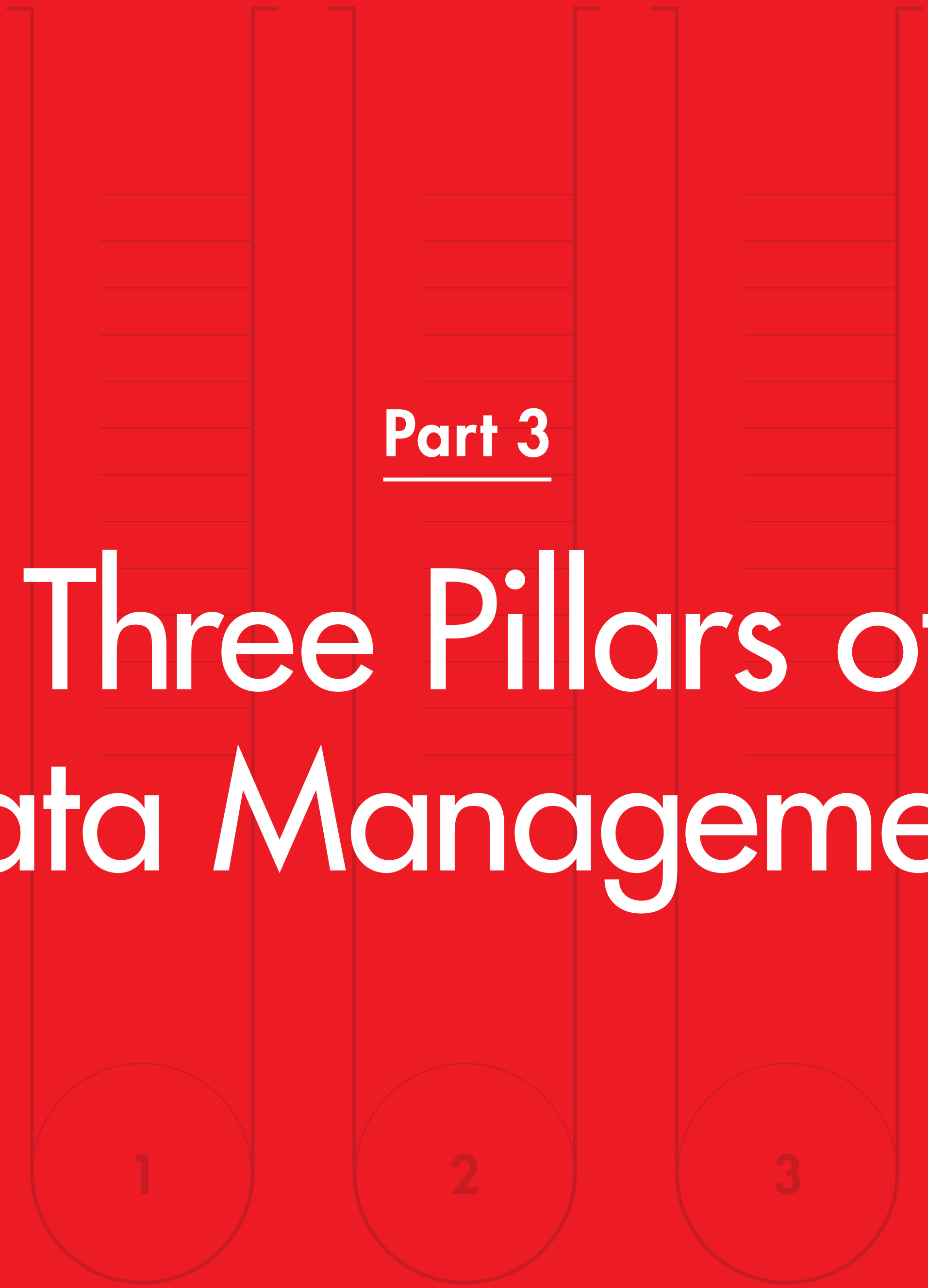
3. **Risk-centric big data security:** to proactively and comprehensively protect your sensitive data against the growing range of data security threats.

In the next section we'll go a little bit deeper to explain what's possible and what's important when you're building an infrastructure to support not only your production environments, but also your lab environments.

Specifically, we'll focus on the three pillars of big data management needed to successfully serve both lab and factory environments.

Part 3

The Three Pillars of Big Data Management.



Big Data Integration.

The first challenge for big data management infrastructure is about creating scalable, flexible, and intelligent data pipelines. On the lab side of things, this is all about making sure you have pre-built tools and simple and intuitive interfaces to ensure your analysts aren't spending all their time waiting for access to the data sources they need.

On the factory side of things, you need to make sure you're leveraging the more readily available IT skills you may already have around data integration to speed up the development and simplify the maintenance of your pipelines.

To that end, your big data platform needs to support:

Universal connectivity
In order to fuel a broad range of experimentation with big data, IT needs to be able to provide connectivity to a huge number of data sources. Versatility is key. That means high throughput data integration for multiple different schemas from multiple different data sources.

But equally, you also need to be able to provide real-time streams for low-latency data such as machine and sensor data.

Pre-built tools
When you're managing high volumes of data from multiple different sources, the biggest challenge is to provide access rapidly. So it makes a lot of sense to leverage pre-built connectors, transformations, and parsers that analysts can use when they need them.

And for data sources that don't come with pre-built connectors just yet, it helps to use self-service data discovery tools that rely on machine learning to suggest appropriate schema automatically.

Abstraction
The key to making scale work is leveraging all available hardware and distributed computing frameworks (e.g. MapReduce, Spark, etc.). Your production deployments need to be flexible enough to work across all available storage and infrastructure, regardless of runtime environment.

By abstracting all your rules, logic, and metadata away from the execution platform, you can ensure both IT and analysts have the freedom to reuse all that work across the platform.

Big Data Integration.

A brokerage model

Point-to-point integration was hard to maintain before big data. Today, it's impossible to manage. This is especially true when you're dealing with dozens of data sources, regions, business units, and users.

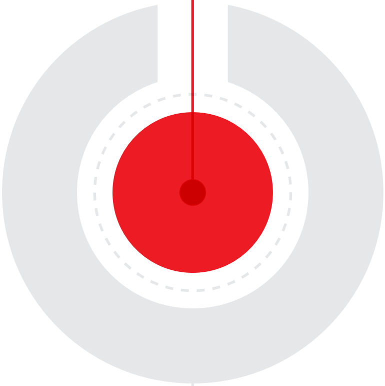
A crucial part of a big data strategy that supports both experiments and production environments is a hub-and-spoke brokerage model that centralizes management and governance of integrations, while distributing best practices.

That way, IT can orchestrate data flows while standardizing to reduce redundant development work and govern data when and where it's necessary.

Staging

All data was not created equal. So rather than burdening IT with the task of modifying and applying schemas to the data coming out of every single data source, it's smarter to prioritize some data over others depending on how the data will be used. For instance, you may need to mask some sensitive personally identifiable information (PII) but not modify sensor log data at all.

By dividing your platform into different staging areas, you can ensure different users and systems only have access to the data that's been appropriately modified (or not modified) for use in them.



How One Multinational Stages its Data

One of the first companies to build a big data platform capable of supporting both lab and factory environments is a large North American conglomerate with 24 business units.

The company used Hadoop to build a staging environment with four layers for its data based on the amount of modifications required to make that data usable. The four different layers it has built are:

Raw
For data that's pulled in right from the source, unchanged. By loading and storing it in Hadoop, the company has been able to lower its data storage costs by about two-thirds (of the cost per terabyte). And in many cases, it can actually leverage this raw data as is, directly from the source.

Published
For lightly modified data that's streamlined for use by the business. The data in this layer may be cleansed, it might have some PII removed and in some cases, the layer may need to handle slowly changing dimensions for data extracted from relational databases.

Both these layers are managed by the central IT function within the company, which is contracted by individual business units when they need to understand what new data needs to be pulled in and integrated.

Beyond these two layers, there are two more that were created to empower the business units to build what they need to.

Core
Here, business units can build new metrics, assets, and business rules to apply to the data. This third layer is built to be broadly applicable across multiple projects. For instance, the business units can build reusable metrics that join customer data with inventory data. But rather than central IT building common rules for use by every business unit, the business units create rules themselves.

Projects
The most tightly focused of the four layers, the project staging environment, data is stored and managed with specific projects and use cases in mind. Impressively, the team has been able to use its big data management to support multiple development channels through different repositories. That way, different teams, distributed across the business units can build domain-specific solutions while operating autonomously at their own pace.

By dividing the staging environments up into these four layers, the company has been able to make sure that developers from business units can work their own way—with different software development methodologies, release cycles, business rules—and produce their own assets.

While at the same time, the central IT function can focus on building out the platform, managing security, and profiling data quality.

Big Data Quality and Governance.

In order to make sure your analysts and developers are working with the appropriate levels of data quality they need—from fully-certified data for reporting to 'good enough' data to validate models—end-to-end data quality and governance is essential.

And since your big data platform needs to be able to manage high volumes of data coming from multiple different data sources and then distribute it to multiple target systems, the need for data governance is even greater.

That means it needs to be able to support:

Automated data quality
If the only thing worse than no data is bad data, then the only thing worse than no data quality rules is hand-coded and inconsistent data quality rules. By leveraging automated data quality tools and pre-built data quality rules, you ensure IT can centrally manage data quality at scale and consistently across the enterprise.

You can improve the productivity of your data stewards by ensuring quality rules are automatically applied and that they are alerted when data quality is out of tolerance. Most important, you can improve the productivity of your analysts by ensuring they don't have to doubt their data or manually search for errors.

Business context
When you're managing data quality across multiple teams, business units, systems, and regions, one of the biggest challenges is consistency. For instance, marketing and finance may define 'customers' differently because their analyses aim to arrive at different outcomes.

So it's crucial that you make it as easy as possible for data stewards to provision common business terms and definitions through business glossaries. That way both analysts and IT have the context they need to link what they're building to the business' needs.

Big Data Quality and Governance.

Easy discovery of exceptions

It's impossible to 'eyeball' several terabytes of data across clusters and data sources. So you need to have a reliable and repeatable approach to data profiling in place if analysts and developers are to have any confidence in the data they're using.

Formal data quality assessments, normalcy scorecards, and exceptions records management give data stewards the ability to react to anomalies and make adjustments when something goes wrong.

Data lineage

For IT to be able to quickly understand and audit the data integrations being built by analysts in your lab environment and developers in the factory, they need to be able to trace data lineage. Without data governance tools in place to catalog and manage your metadata, that isn't possible.

So when you're building your data management infrastructure, it's important that you give your users the ability to quickly determine data provenance and easily audit large datasets.

Master data and relationship management

In order to ensure your stack can infer and detect relationships between entities and domains at scale, it's important to leverage entity matching and linking that enriches your master data around crucial domains like products and customers.

It dramatically improves the speed with which analysts can find important patterns and trends. But most important, it gives them a holistic view of how those domains interact with each other. For instance, it allows them to understand which products customers are interacting with.

Big Data Security.

The more users, systems, business units, and partners are involved in your big data initiatives the harder it becomes to spot security breaches and oversights.

So rather than building your infrastructure first and then attempting to apply security policies and comply with regulations later, it's absolutely crucial that your platform has best-practice data security baked-in.

In practice, that means you need to enable:

Discovery and identification
A 360-degree view of sensitive data is crucial for a risk-centric approach to big data management. So IT needs to be able to discover, classify, and monitor sensitive data stores wherever they live and routinely profile that data for exposure to potential threats.

Additionally, you need non-intrusive data masking to protect your data assets even in the case of a perimeter security breach by de-identifying sensitive data in development and production environments.

Risk scorecards and analytics
A comprehensive view of all your sensitive data is crucial to detecting and responding to threats. Risk analytics and scorecards automate the detection of high-risk scenarios and exceptions based on modeling, score trends, usage, and proliferation analysis so IT is alerted instantly.

When it comes to big data security, speed is everything. The longer it takes IT to notice a security threat, the harder it becomes for you to reverse or even diagnose the damage.

Universal protection
Threats to big data security are evolving just as rapidly as data protection. So when it comes to securing your data across systems, users, and regions, shortcuts aren't really an option.

Your big data security strategy needs to be holistic enough to provide masking, encryption, and access control across data types (both live and stored data) and across environments (in production and non-production environments).

Centralized policy-based security
Operationally, it's important that big data security doesn't become a burden on IT or an obstacle to experimentation. You need to be able to create and monitor security policies centrally and then distribute them to users, systems, and regions.

At the scale of big data, this policy-based approach to security also makes compliance more manageable since certain privacy laws mandate location and role-based data controls.

Four Dimensions of Big Data Security

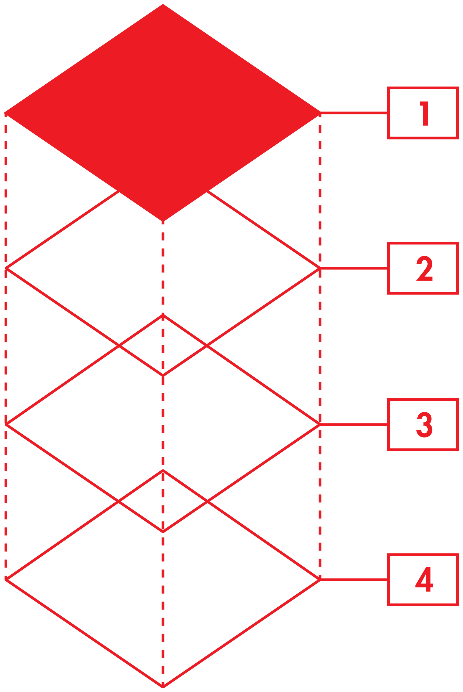
When most of your data no longer lives behind the firewall, you can't afford to rely on authentication and authorization alone for your big data security strategy. In fact, your big data security should ideally encompass four types of protection:

1. Authentication and Authorization
Tools like Kerberos, Knox, and Sentry are essential for perimeter control when multiple users are accessing your cluster.

2. Encryption
Initiatives like Intel's Project Rhino allow you to encrypt the data living in your data stores and manage keys. This is crucial when you need to ensure outsiders can't access sensitive data but insiders need to be able to decrypt and access the same data. And format-preserving encryption can be used to get as granular as the field level.

3. Tokenization
This is all about substituting sensitive data elements with non-sensitive equivalents or 'tokens'. It allows you to preserve data types and formats and protect against stolen keys.

4. Data masking
To persistently or dynamically protect data from certain users (e.g. protecting credit card data in dev and test environments), data masking can be used to de-identify data while still ensuring it looks like the original data. Unlike with encryption, the 'masked' data element can't be decrypted by insiders to reveal the sensitive information being protected.



A Reference Architecture for Big Data Management.

While no two enterprises will have exactly the same infrastructure needs, it is useful to consider the architectural choices made by other companies that have bridged the gap between their lab and factory environments.

Specifically, it's important to consider your technological requirements in the context of three crucial layers:

1. The visualization and analytics layer

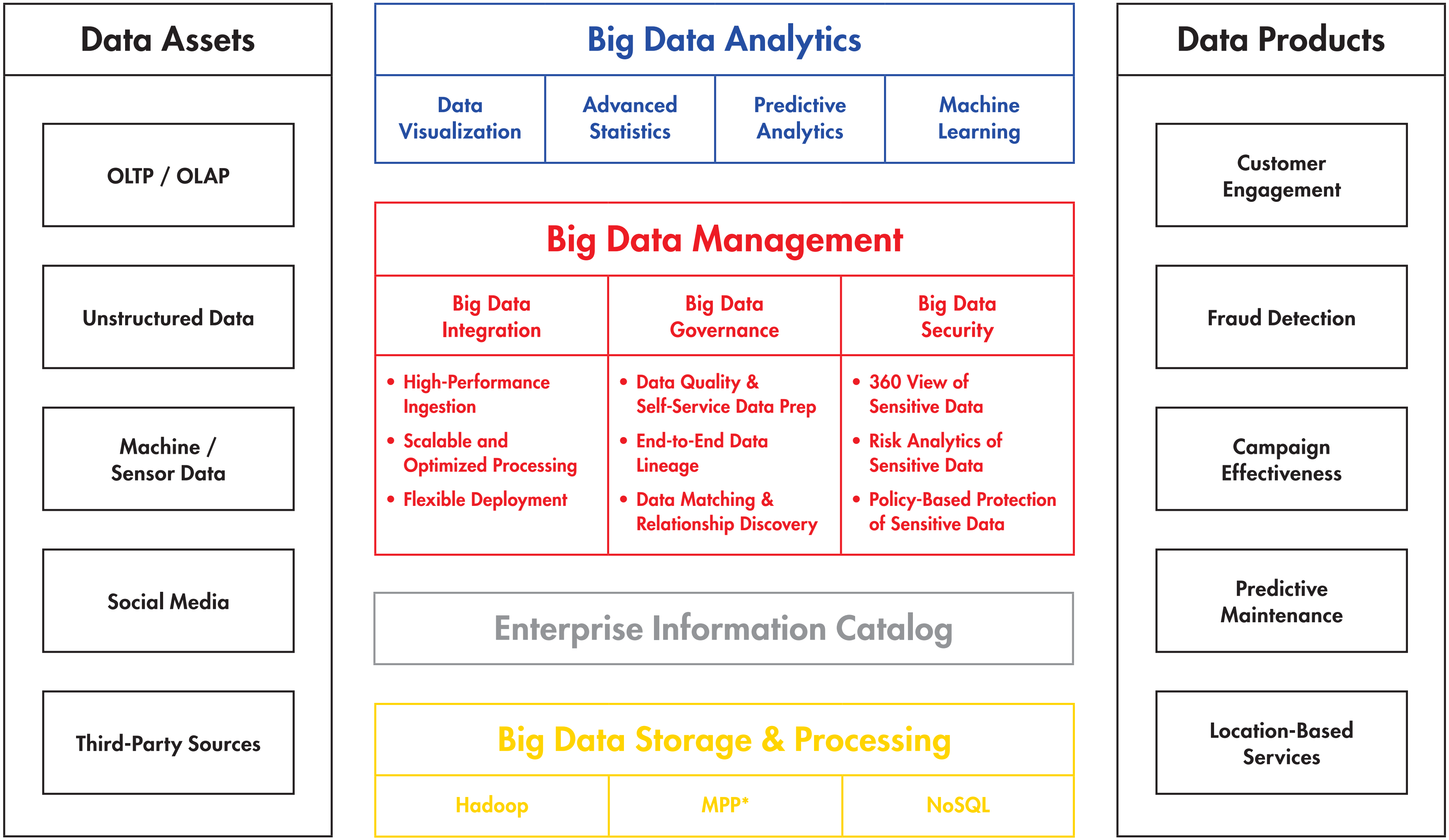
2. The big data management layer (encompassing big data integration, governance, and security)

3. The data storage persistence layer

Typically, enterprises focus on the first and third layers without plotting out what they need for the second layer. But as we've described, this second layer plays a crucial role in ensuring you can enable both a lab and factory environment.

In the following reference architecture, we've illustrated these layers and some sample tools and capabilities worth considering. Use it as a basis to determine your specific infrastructure requirements for a big data platform.

Big Data Reference Architecture



Data Assets

- OLTP / OLAP
- Unstructured Data
- Machine / Sensor Data
- Social Media
- Third-Party Sources

Big Data Analytics

- Data Visualization
- Advanced Statistics
- Predictive Analytics
- Machine Learning

Big Data Management

Big Data Integration

- High-Performance Ingestion
- Scalable and Optimized Processing
- Flexible Deployment

Big Data Governance

- Data Quality & Self-Service Data Prep
- End-to-End Data Lineage
- Data Matching & Relationship Discovery

Big Data Security

- 360 View of Sensitive Data
- Risk Analytics of Sensitive Data
- Policy-Based Protection of Sensitive Data

Enterprise Information Catalog

Big Data Storage & Processing

- Hadoop
- MPP*
- NoSQL

Data Products

- Customer Engagement
- Fraud Detection
- Campaign Effectiveness
- Predictive Maintenance
- Location-Based Services

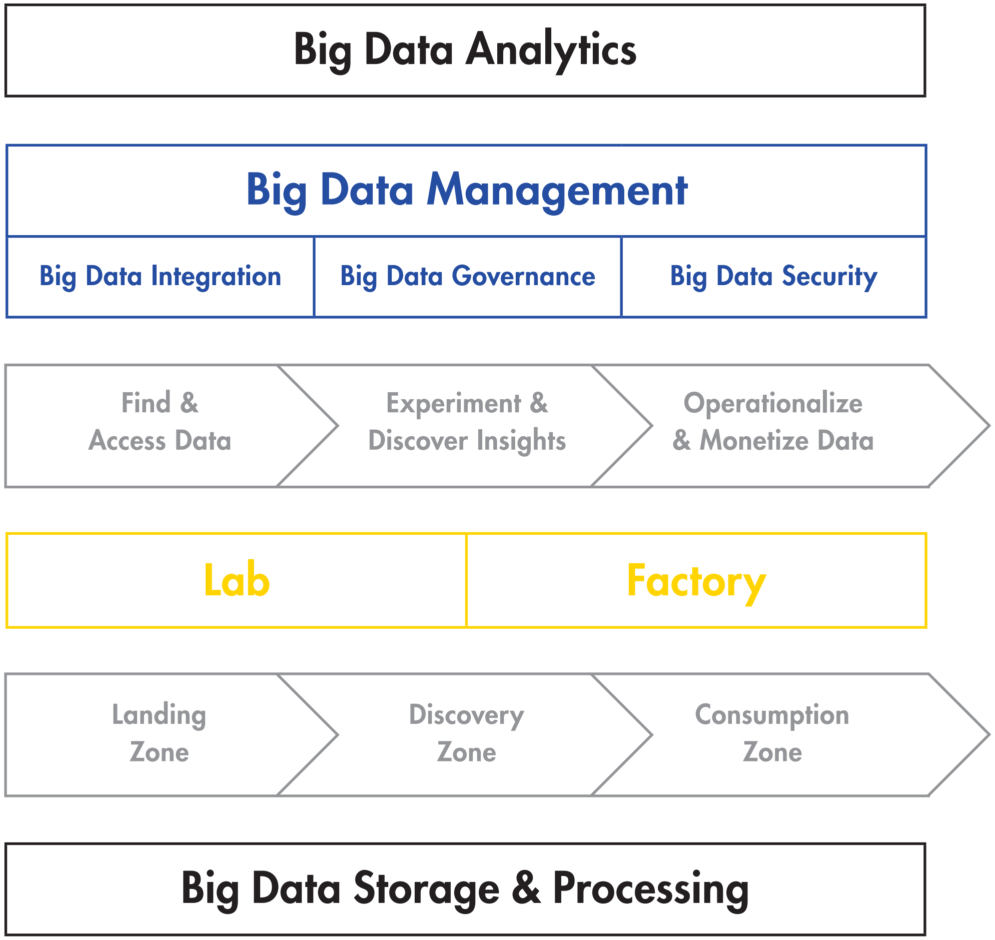
*Massively Parallel Processing

A Reference Architecture for Big Data Management.

The following diagram is a conceptual representation that depicts the various processes within the context of a big data infrastructure.

Source data is ingested into the landing zone, transformations are applied to prepare data for exploratory analysis, and automated workflows curate data for consumption.

This enables data scientists and analysts in the lab to quickly find and access the data they need to run their experiments and discover insights. Data engineers build automated workflows to operationalize these insights delivering trusted information that can then be used to monetize data assets.



Conclusion

Interrogate, Invent, Invest.

Interrogate, Invent, Invest.

Like ‘innovation’, ‘big data’ is a whole lot more than a buzzword. It’s an essential strategy for any enterprise seeking rapid and sustaining growth.

But corporate innovation has to start with an approach to experimentation that allows analysts and scientists to try new things without having to live with the collateral damage of failed experiments. The good news is that cost-effective and scalable storage and processing has shrunk the gap between idea and implementation.

A big data lab that can’t rapidly implement innovative solutions in a production-grade factory environment is only half-complete. And a data management infrastructure that can’t support self-service autonomy for analysts to experiment isn’t complete either.

We hope that with the help of the lessons and experiences we’ve shared here, you’ve seen that smart architectural and infrastructural decisions can help you de-risk experimentation and streamline production simultaneously.

The key is to remember the crucial layer between big data storage and big data analysis—big data management.

Implement the big data management three pillars of big data integration, governance, and security, and you won’t just be streamlining IT’s development and production processes—you’ll be giving the smartest scientists and analysts in your business the license they need to innovate.

Sources

1. **Medium**, The story of AWS and Andy Jassy's Trillion Dollar Baby

2. **Wall Street Journal**, Visa says big data identifies billions of dollars in fraud

3. **ComputerWeekly.com**, GE uses big data to power machine services business

4. **Datafloq**, T-Mobile USA cuts down churn rate by 50% with big data

5. **Informatica**, UPMC customer success story

6. **Datafloq**, Three use cases of how GM applies big data to become profitable again

7. **CIO Journal**, WSJ, GM grapples with big data, cyber security in vehicle broadband connections

8. **Forbes**, How big data is changing the insurance industry forever

9. **EMC InFocus**, It's not just big data…it's gigantic data: A Telecoms Case Study

10. **TDWI Best Practices Report**, Hadoop for the Enterprise, 2015

11. **TDWI Best Practices Report**, Hadoop for the Enterprise, 2015

12. **ComputerWeekly.com**, 'Spark versus MapReduce: which way for enterprise IT?' August 2015.

13. **New York Times**, For big data scientists, janitor work is the key hurdle to insights, 2014