WHITE PAPER

PLATFORA SOLUTION ARCHITECTURE

Implementing a Big Data Discovery Solution with Platfora

PLATFORA SOLUTION ARCHITECTURE

Implementing a Big Data Discovery Solution with Platfora

INTRODUCTION

Big data is rapidly transforming not only how business is done but how businesses are organized around data. And in an increasingly data-driven economy, how businesses are organized around data is becoming synonymous with how businesses are organized overall. It isn't just the technology that has changed. All of the old assumptions about how data is accessed and used—assumptions that have held true for decades—have changed in the wake of the big data revolution.

Who owns the data? Who can have access to it? Who owns the questions? And who is responsible for finding the answers? As the emerging capabilities of big data and modern analytics architectures are fully realized, changing what organizations expect and what they demand, businesses are finding innovative and often unexpected answers to these questions. Driving such change is a combination of business and technology drivers, with technology opening up new business capabilities that, once leveraged, demand even more capability from both systems and processes.

In the face of this upward spiral of both requirements and capabilities, one thing is very clear: to be effective, a solution framework for big data discovery has to respond to both technical requirements and broader business needs. Businesses require access to all of their data, not subsets, which means handling massive data volumes. They need to manage a wide variety of data structures and types. They need to find correlations across these large volumes and multiple types of data to generate new insights, and make those insights more broadly available across the organization than ever before. These are all business needs as well as technical, and the old traditional models won't scale any better than the old technologies. New technologies and a new business reality demand a fundamentally different architecture for delivering data analysis.

Evolution of Modern Architectures

In the traditional enterprise data warehouse (EDW) environment with a BI front end, it was understood that an organization would access and analyze only a portion of its data. The limitations occurred in the carrying capacity of the data warehouse infrastructure and the modeling requirements of the BI tool. Going in, analysts needed to know exactly what question or questions they wanted to ask—these had to be defined up front. Beginning with a new dataset, the process of finding the answer to a single question required the participation of an entire team and could take from three to six months. Organizationally, this workflow required ETL programmers, Data Warehouse Architects and Administrators, and BI Architects and Administrators. After an extensive ETL process, 10% of the organization's data would be available within the data warehouse and BI tools. Because of the complexity of the process and the long time frames involved, iterating with follow-up questions was hard to do effectively, if at all.

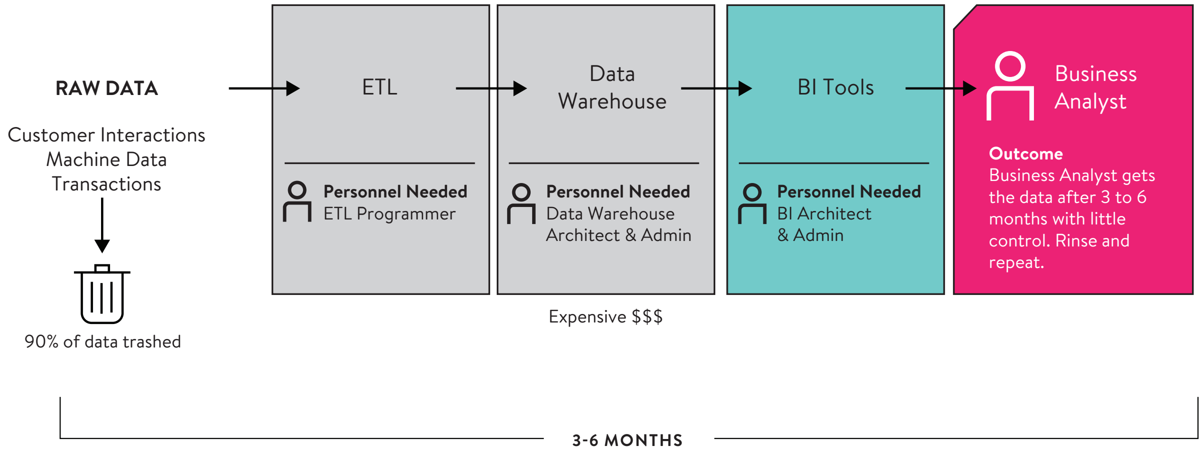


Figure 1: Analytics Workflow for Traditional Data Warehouse / BI Environment

Hadoop kicked off a revolution in how data is stored and processed by providing a 1000x savings for storing data over an EDW, thus eliminating the previous limitations around data volume and data type. Now organizations can put all of their data together in a common repository at a small fraction of what it would have cost to store it in a traditional EDW. These are big steps forward, but on their own they do not address the complexity and latency issues encountered in the data warehouse model. In order to perform SQL queries on Hadoop data, businesses have to use tools like Pig, Hive, and MapReduce. This adds time and complexity to the process and requires the involvement of a team with specific knowledge of each of these areas. The team to support such an environment typically includes Hadoop experts and Data Scientists in addition to the BI and Data Warehouse and Architects and Administrators already mentioned.

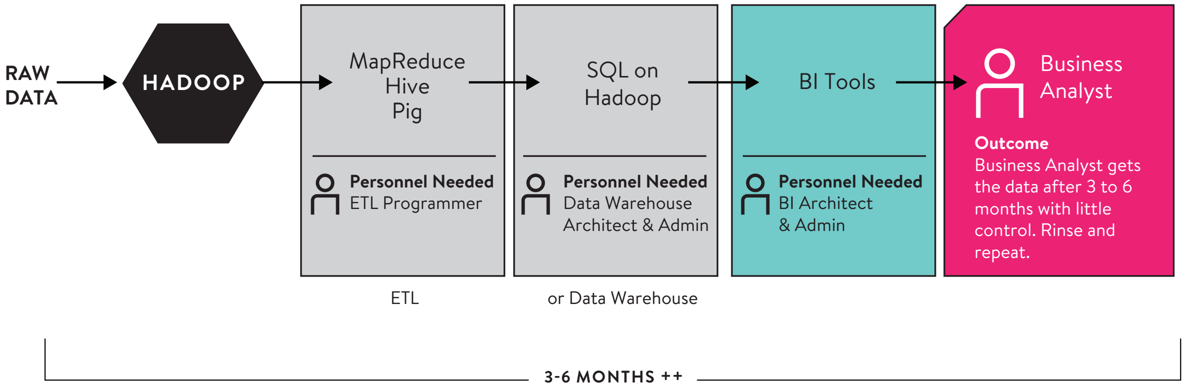


Figure 2: Analytics Workflow for Typical Hadoop Environment

Today, Platfora big data discovery provides a new architectural approach that thoroughly disrupts the traditional workflow. Native to Hadoop, Platfora eliminates the latency and complexity inherent in previous approaches. Users can visually and interactively interrogate data up to petabytes in size. The environment is easily accessed by existing data administrators and Data Scientists without the need for a team of specialists to prepare or manipulate the data.

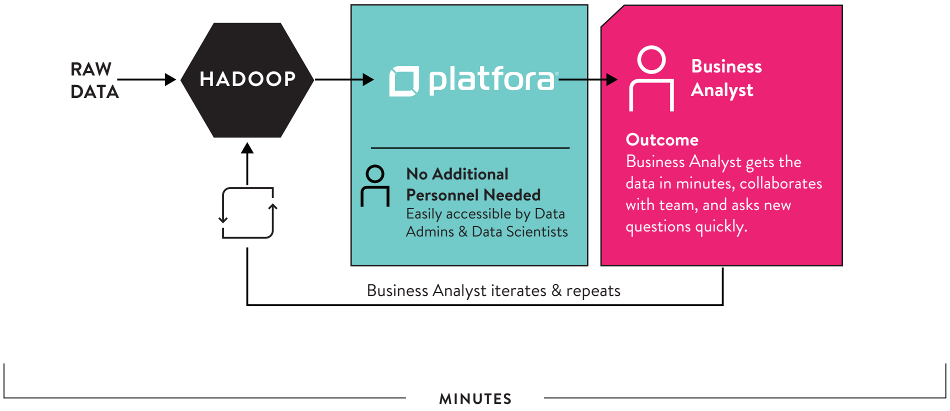

RAW DATA
HADOOP
platfora
No Additional Personnel Needed
Easily accessible by Data Admins & Data Scientists
Business Analyst
Outcome
Business Analyst gets the data in minutes, collaborates with team, and asks new questions quickly.
Business Analyst iterates & repeats
MINUTES

Figure 3: New Analytics Workflow Enabled by Platfora

The Platfora Difference

This document outlines the Platfora Solution Architecture, detailing how Platfora big data discovery supports an entirely new kind of analytics environment. It provides a breakdown of the architecture, showing how it serves as the basis for Platfora's most important differentiators from other solution types. Platfora's core differentiators include the following:

Platfora provides an **end-to-end solution**, as contrasted with approaches that require cobbling together a working solution out of disparate parts.

Platfora's self-service capability puts analysis **into the hands of decision-makers**, the businesspeople who need access to the data; this is distinct from solutions that require frequent, multiple hand-offs between business analysts and technical staff.

Platfora **lenses** are in-memory extracts of raw data with analysis pre-applied. They provide fast access to big data and are updated automatically to avoid lag times to and reduce duplication.

The heart of Platfora is an **in-memory scale-out engine** that supports an environment of any size, unlike traditional environments that put definite limits on how much data can be supported in memory.

The Platfora **Data Catalog** keeps track of all the datasets in the data lake, enabling data governance and tracking of data lineage.

With added support for **Apache Spark**, the Platfora solution architecture provides new capabilities that build out the value proposition even further, including out-of-the-box advanced analytics, simplified access of data, accelerated results, and the option to manage analysis using approaches other than traditional SQL tools.

ARCHITECTURAL OVERVIEW

This section details the makeup of the overall Platfora platform as well as the relationships between the various components and how they are deployed. First we address the High-Level Architecture, which is then broken down into two further views: Deployment Architecture and Server Architecture.

High-Level Architecture

Figure 4 below shows the Platfora High-Level Architecture. As the diagram shows, Platfora supports a truly end-to-end solution—from the raw data produced by transactions, customer interactions, and machines to an end-user environment accessible by business analysts. The raw human- and machine-generated data are made available via appropriate connectors (as needed) through HDFS and other data sources. Native to Hadoop, Platfora is leverages an open source infrastructure of proven and emerging technologies, including HDFS, MapReduce, and now Apache Spark. Platfora's scale-out, in-memory processing capability provides for accelerated data access across an unlimited number of nodes.

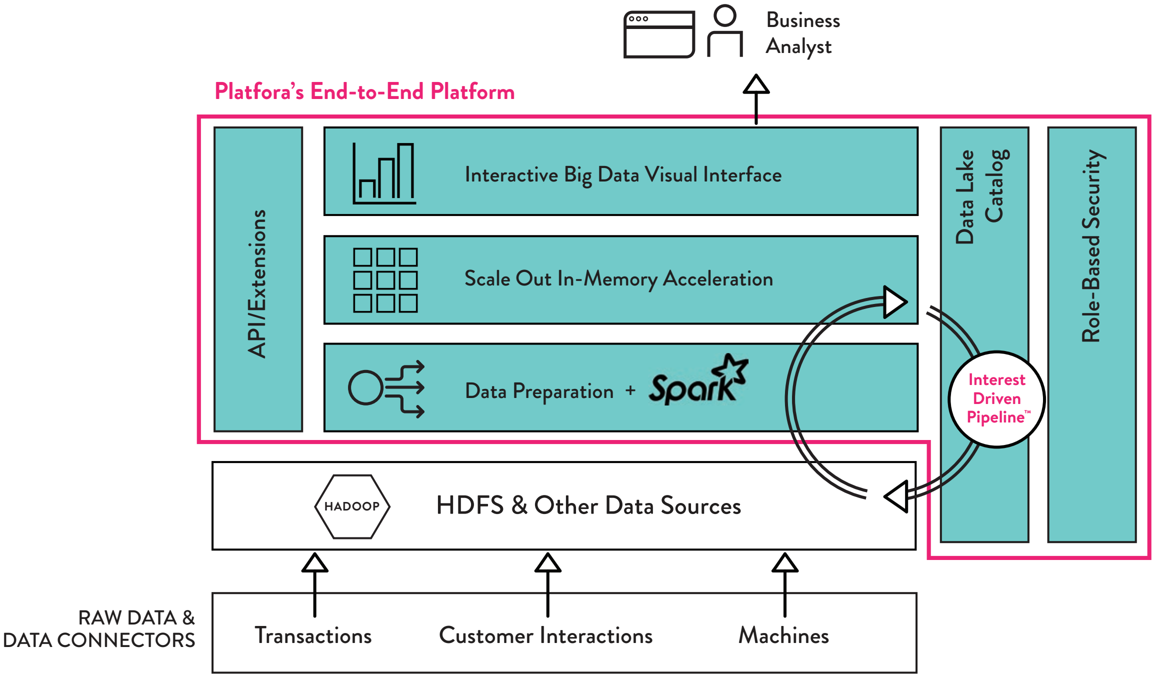

Business Analyst
Platfora's End-to-End Platform
API/Extensions
Interactive Big Data Visual Interface
Scale Out In-Memory Acceleration
Data Preparation + Spark
Data Lake Catalog
Role-Based Security
Interest Driven Pipeline™
HADOOP
HDFS & Other Data Sources
RAW DATA & DATA CONNECTORS
Transactions
Customer Interactions
Machines

Figure 4: Platfora High-Level Architecture

API support and Platfora extensions make for a more flexible and adjustable solution environment, enabling businesses to adapt their solution to the specific questions they most need to ask. Platfora's Data Catalog spans the end-to-end workflow providing a semantic view of data from its raw state in storage all the way to how it is used in analysis. The catalog provides a searchable interface for users to find and use the data they need.

Platfora provides a comprehensive and flexible security model. With a three-pronged approach to security that includes data level security, user-created object-level security, and role-based access, Platfora provides security at both the implementation and operational stages. End users of Platfora access visual and highly intuitive analytics via an HTML5 compliant web browser.

Deployment Architecture

Platfora software runs on a scale-out cluster of servers, which can be on premise or cloud-based, physical or virtual. Platfora uses native Hadoop interfaces to connect to the distributed file system and data processing services of Hadoop.

On-Premise

For on-premise deployments, Platfora servers reside on dedicated hardware co-located in the same data center as the Hadoop cluster. Platfora users access the Platfora master node using an HTML5-compliant web browser. The Platfora master node accesses the HDFS NameNode and the YARN Resource Manager or MapReduce JobTracker. The Platfora worker nodes access the HDFS data nodes directly as well. Platfora software can run on a wide variety of server configurations—on as little as one server or scale across multiple servers.

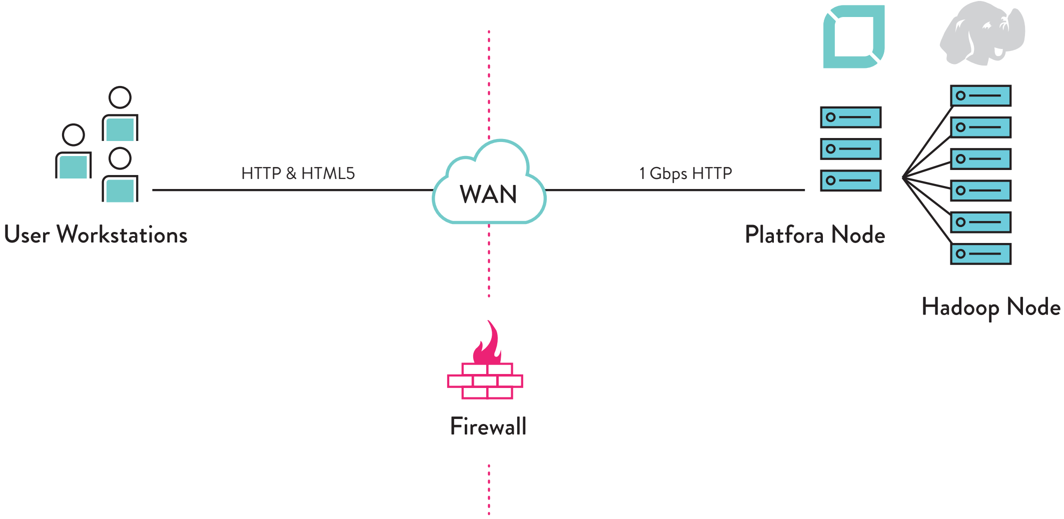


Figure 5: Platfora On-Premise Deployment

Cloud

Platfora can also be deployed in a variety of cloud environments with Amazon Web Services. It uses Amazon Simple Storage Service (S3) to access the raw data and uses Amazon Elastic MapReduce (EMR) to run its data processing jobs (lens builds). The results of the lens build jobs are also written back to S3.

Another option is to use cloud infrastructure including Amazon EC2 instances in the same way as an on-premise deployment.

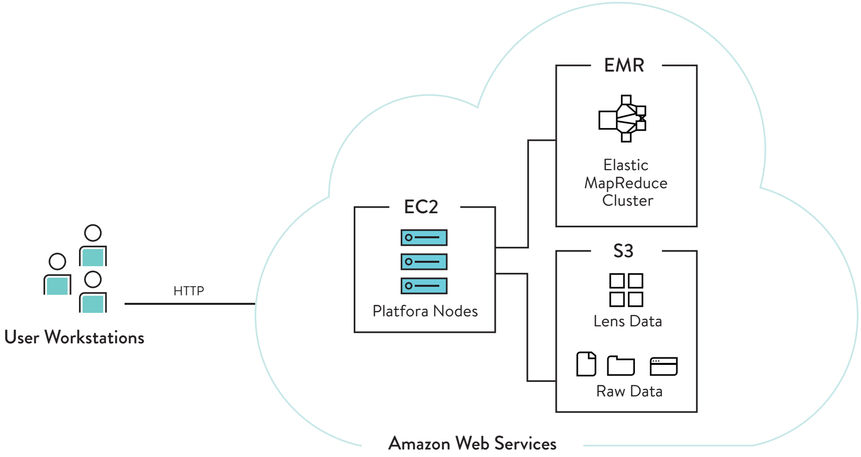


Figure 6: Platfora Cloud Deployment

Server Architecture

Master Node

The master node manages the following Platfora services:

Metadata Catalog: The Platfora metadata catalog contains all of the information about the data managed by Platfora, including datasets, lenses, and vizboards. The catalog is backed by a relational database that runs on the Platfora master node; it is accessed by all nodes in the Platfora cluster.

Lens Builder: The lens builder interfaces with the data processing services of Hadoop. It translates requests to prepare data for analysis from the Platfora application into a series of custom Spark and MapReduce jobs. These are submitted to the YARN Resource Manager or Hadoop Job Tracker for execution. Once the data is extracted and transformed within Hadoop, the job results are written back to the Hadoop file system as Platfora lens. The lens is a Platfora columnar file format for storing high performance in-memory extracts of data with analysis applied.

On-Disk Storage: Finished lenses are immediately copied from the Hadoop file system to on-disk storage of the Platfora worker nodes. The data of a lens is distributed across all of the available worker nodes in a Platfora cluster using a sharding technique. However, in a multi-node configuration, the lens data doesn't get distributed to the master. The master gets only the information it needs to act as the query coordinator.

In-Memory Query Engine: When users explore and analyze data in Platfora, they are actually generating queries that run against a lens. Platfora returns the result of a lens query as a data visualization, a table, an API response, or by writing results to disk. The in-memory query engine has two kinds of processes that work on a query. The query coordinator process runs on the master node only, and translates actions made in the Platfora application into queries. The query worker process runs on the worker node.

Web Application Server: The Platfora user interface runs as a web application in the network. Users connect to Platfora using any HTML5-compliant browser. Through the browser, users interact with data in Hadoop as easily as browsing a website.

Worker Nodes

The Platfora worker nodes are used to distribute lens storage capacity and query processing workload. As users work with more and bigger lenses in Platfora, more memory and processing power are leveraged to render visualizations quickly. Administrators can add additional worker nodes to scale out lens storage capacity and performance.

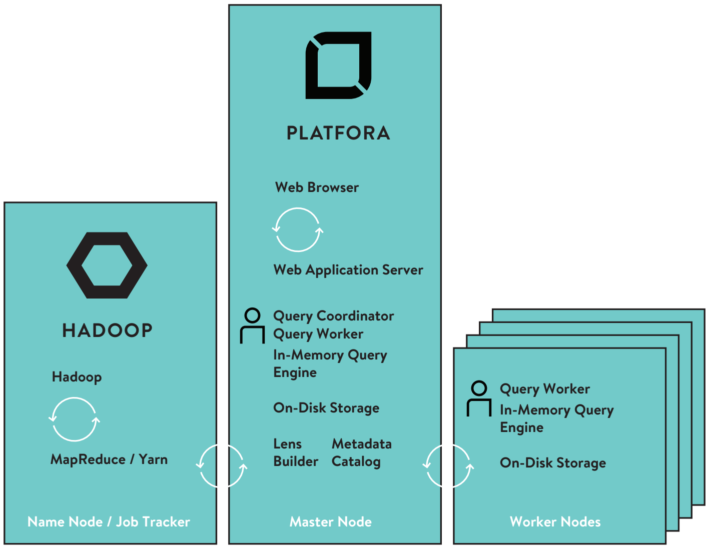


Figure 7: Relationship of Platfora Master Node to Worker Nodes

MPP Query Architecture

Platfora supports Massively Parallel Processing (MPP) for managing big data queries. As Figure 8 shows below, the master node takes the query and breaks it down into smaller tasks and assigns those tasks to the worker nodes that have that part of the data. This enables Platfora to take advantage of all of the resources of the various worker nodes to execute the query. Because of this approach, the size of data being queried can be orders of magnitude larger than the memory of any one machine.

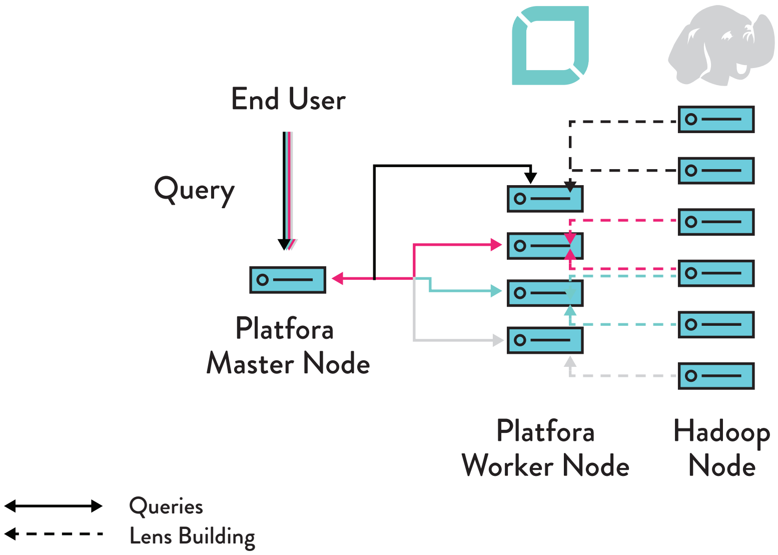


Figure 8: Platfora MPP Architecture

DATA ACCESS

This section provides a description of the Platfora infrastructure for data access via the various appropriate data sources.

Platfora takes an open approach to the makeup of the user organization's data lake. From the Platfora perspective, it is that collection of data that matters—not, for example, which Hadoop distribution is used. Platfora connects to a data lake and uses Hadoop to process the data.

This approach is in contrast to other solutions that require a SQL connector to Hadoop. For example, it is common to provide access to Hadoop with Hive queries. Such an approach puts a large burden on the enterprise. It requires the business to know all the questions up front, with Hive tables providing some of the same kind of structure that a data warehouse would. This kind of environment is not iterative and it is difficult to change. It limits both the data that can be accessed in the data lake and the users who can access the data; standard business users usually don't have strong SQL skills.

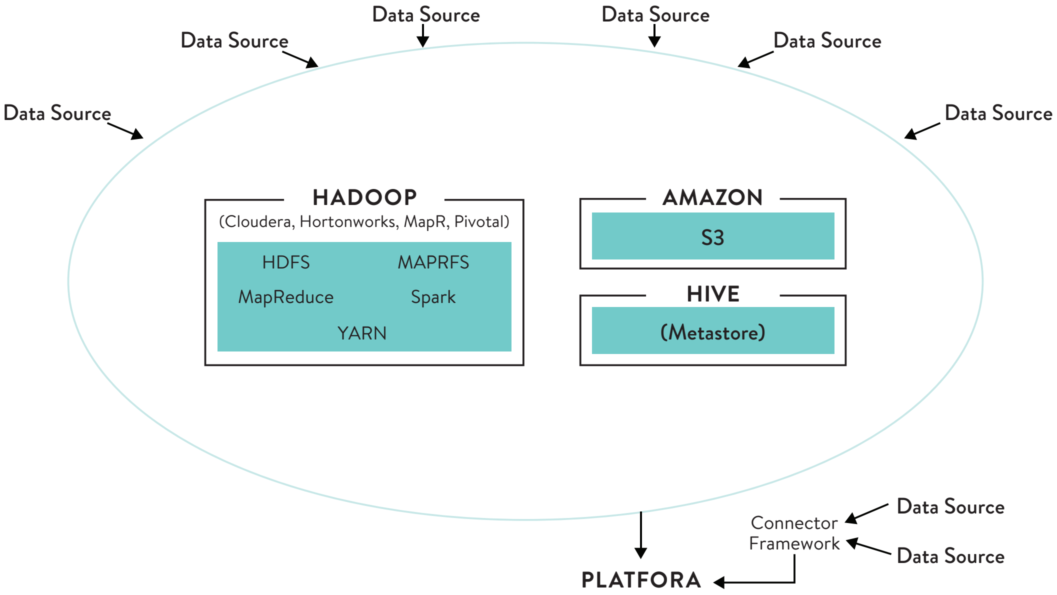


Figure 9: Platfora Data Lake Architecture

Hadoop

Platfora provides fully native support to Hadoop, and can access data from a Hadoop cluster by several different approaches.

HDFS

Platfora connects to the HDFS NameNode server and using the HDFS file system as its primary data source. Platfora supports all major Hadoop distributions, including Cloudera, Hortonworks, MapR, and Pivotal.

MapR

Platfora supports MapRFS and can be configured to connect to a MapR Container Location Database (CLDB) server using the MapR file system as its primary data source.

Amazon
Platfora supports Amazon Simple Storage Service (S3) as a data source for users who run their Hadoop clusters on Amazon EC2 or who use the Amazon EMR service. Platfora supports only the S3 Native File System (s3n). Depending on the configuration, HDFS can also be used with Amazon cloud storage.

Hive
Platfora does not run Hive queries for data access in Hadoop environments, but it does access the Hive metastore to import metadata as required.

Data Connector Framework

With data connectors developed via the Data Connector Framework, a Platfora environment can be connected to virtually any other data source. Occasionally, analytics requirements emerge that require integrating data from outside the data lake. Connectors close that gap. Data connectors are not scalable in the way that Hadoop is, so they are ultimately not a technology to support a full big data analytics environment; however, they do add important functionality for some use cases.

Execution Frameworks

Platfora leverages Apache YARN as the resource allocation and management framework for all jobs run within the Hadoop cluster. Until fairly recently, MapReduce was similarly the default choice for distributed processing on the cluster. Platfora has always supported MapReduce, and now also supports Apache Spark. With in-memory storage and data processing Spark can provide faster performance for some use cases. More importantly, Spark provides higher-level operators and rich analytics libraries for Platfora to leverage to prepare and transform large datasets.

DATA PREPARATION AND LENS CREATION

One of the fundamental differences between the Platfora approach to big data and the approach taken by most solution providers is that Platfora requires no external process to make big data smaller before it can be analyzed. Traditional BI methodologies and processes involve the creation of cubes, external datamarts, and other subsets or reproductions of some part of the data to be analyzed. This is usually done by dedicated IT or analytics staff. Platfora works with the full dataset, allowing for iterative access on the same data, limiting data movement and preparation time. And when all that is needed is a subset, business analysts can do the work for themselves.

This section provides an overview of Platfora's data preparation functions, including both the native Platfora functions and new capabilities incorporated through Spark and explains how, once data preparation is complete, lenses can be created and deployed.

Native Platfora Data Preparation Functions

Raw File Access
Platfora works directly with raw files as they are generated and stored in Hadoop. Whereas a typical dataset could be derived from a single file, big data discovery often involves working with datasets that are made up of hundreds, thousands, or even millions of files generated by systems like web servers, security systems, or other machines. A Platfora Dataset contains the information on how Platfora will automatically process the raw data files—even if they are in multiple raw formats and schema—into a single usable dataset for analysts.

Format and Type Inference

Platfora automatically infers the format of raw files and “wrangles” them into a structured, columnar state. Common data formats such as delimited (i.e. CSV), JSON, XML, Avro, and others are supported. Accurately inferring file format makes it easy for business users to work with raw data without first writing parsing rules. Platfora provides a preview of a sample of the parsed raw files to the user to ensure that data is being properly understood and allows the user to override settings as required.

Nested Data Handling

Data formats which support nested elements like XML and JSON can also be wrangled into a columnar format. Like leaves on a tree, users can pick and choose which elements should be pulled from the raw records

Data Profiling

Platfora automatically reviews raw datasets to provide descriptive statistics about the makeup of the data as well as suggestions for how it can be prepared.

Computed Fields

Computed fields are useful for deriving meaningful values from base fields (such as calculating someone’s age based on their birthday), doing data cleansing and pre-processing (such as grouping similar values together or substituting one value for another), or for computing new data values based on a number of input variables (such as calculating a profit margin value based on revenue and costs).

Platfora has an extensive library of built-in functions that can be used to define data processing tasks. There are specific types of computed fields designed to handle binned data, location values, date/time fields, and time series data.

Platfora provides a special type of computed field called a Meaure(s). Meaure(s) provide the basis for quantitative analysis in a visualization or lens query. A Meaure(s) is a numeric value representing an aggregation of values from multiple rows. For example, Meaure(s) contain data such as total dollar amounts, average number of users, count distinct of users, and so on.

References and Data Modeling

Creating a Reference(s) allows the datasets to be joined when building lenses and executing queries, similar to a foreign key to primary key relationship between tables in a relational database. Using Reference(s), users can model their data into a full snowflake schema.

Once a Reference(s) is created, the fields of all downstream datasets are available through the dataset where the reference was created. Users can define computed expressions using downstream dimension fields, and they can choose downstream dimension fields when they build a lens.

Data Preparation with Spark

Support for Apache Spark adds a whole new level of capability and performance to data preparation with Platfora. Spark provides the ability to sample data and to instantly see the results of a proposed change. Spark makes it possible to search data for a particular value or type and filter for that in the context of data ingest. Spark transformation extensions make it possible to significantly expand the data transformation capabilities that Platfora already offers.

Additionally, Spark represents a growing technology base and is likely to become an increasingly important part of the big data landscape. Spark support adds a wide variety of options for interactive data exploration and preparation. There is a growing community of Spark analysts and developers providing technology which can be used seamlessly with Platfora. For example, SparkR provides access to R functions for Hadoop. This makes it easy for Data Scientists and others who are used to working with R to continue using the framework they are familiar with. Spark opens the door for Data Scientists to use tools Data Scientists do.

Lens Creation

A key differentiator of the Platfora solution architecture is that it supports a workflow in which the analyst does not have to call IT to build a data mart. The Platfora lens is built within Hadoop, leveraging raw data. The transformed data is projected into a high performance columnar in-memory format with Platfora to support rapid visual query access.

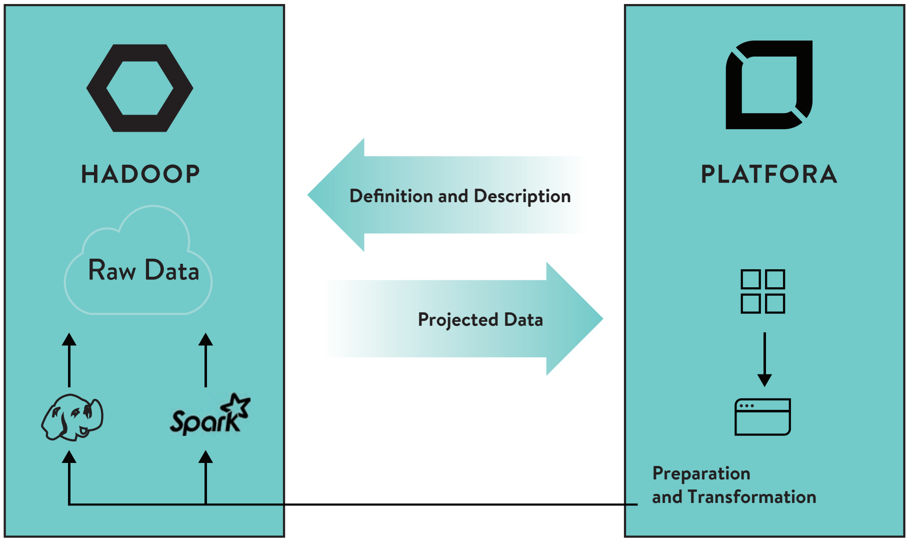


Figure 10: Lens Workflow

Platfora leverages both MapReduce and Spark in building lenses. The time window required to build a lens is determined by the size of Hadoop cluster: the amount of data plus the number of nodes. Once set up, lenses can be built out incrementally. As new data arrives, the lens is automatically updated without having to be rebuit from scratch.

IN-MEMORY, SCALE-OUT ENGINE

As discussed in section 1 of this document, changing technologies and changing business requirements have been the major drivers behind redefining the analytics workflow. In section 4 we showed how, unlike traditional solutions, Platfora does not require the development of data marts or other silos to enable the implementation of an analytics environment. Instead, Platfora addresses the need for a middle tier of in-memory data to make big data accessible without compromising the full dataset or adding unneeded complexity and redundancy to the environment.

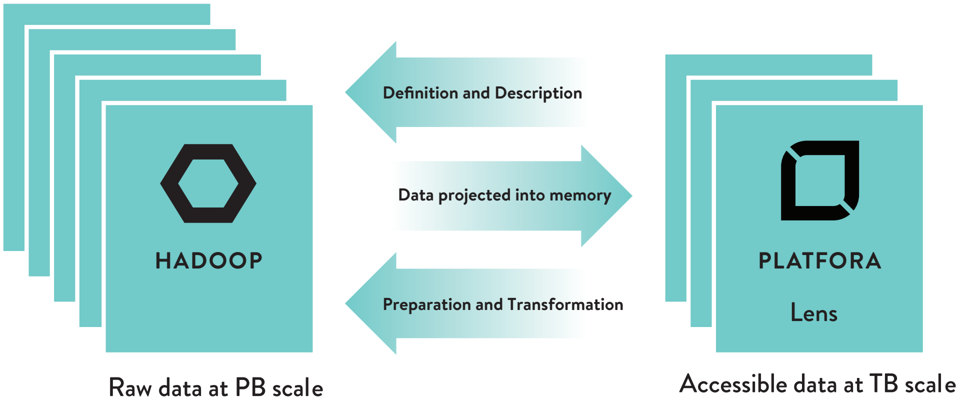

HADOOP
Raw data at PB scale
Definition and Description
Data projected into memory
Preparation and Transformation
PLATFORA
Lens
Accessible data at TB scale

Figure 11: Scale Out Architecture

The in-memory approach is important not just because it simplifies the environment, but because it supports an interactive approach for business users. Users can ask questions iteratively, building their understanding on previous answers, without the lag times and need to reach out for help that would be involved in traditional environments.

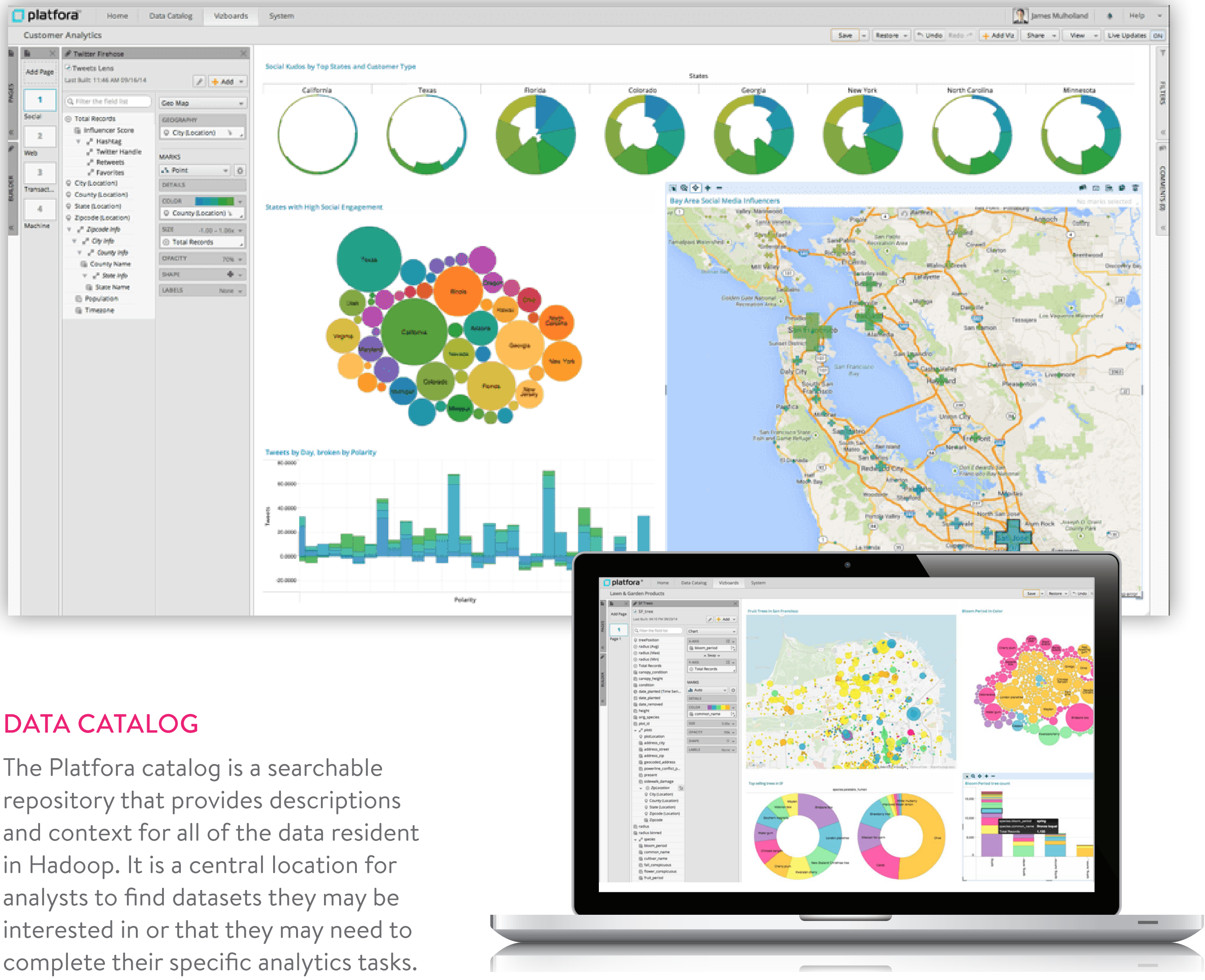
The Platfora architecture provides a scale-out engine that functions like a caching layer. This makes it easy to go back to the source when the data you need is not located in-memory. The scale-out engine swaps data into memory from disk automatically.

Perhaps most importantly, this is all completely transparent to the user. Users never have to interrupt their workflow to ask whether the data is in memory or whether the scale is appropriate. In a traditional system, many cycles are typically spent working out these kinds of issues.

DATA VISUALIZATION

Backed by in-memory data, Platfora's visualization layer supports iterative, ad-hoc visual analysis that is fully interactive. Platfora Vizboards are 100% HTML5 and utilize a custom rendering engine. While available engines like D3 scale to small data sizes, the Platfora Vizboard engine allows interactive analysis at the scale of big data—rendering millions of data points on the screen at once yet retaining the ability to select, highlight, pan, zoom, and drill into data.

The Platfora visualization builder is designed to allow business users to visualize almost any data in a lens with drag-and-drop simplicity. Whether it is selecting the Measure fields to quantify, dimension fields to group by or a time series to draw the user has complete flexibility to explore all of their data. Users can select the type of mark—for instance a line or a bar—and then begin to encode more data in their visualization using the color, size, shape, opacity, and labels on that mark. As part of the visualization process, the analyst can filter, sort or limit, or drill into data interactively.



DATA CATALOG

The Platfora catalog is a searchable repository that provides descriptions and context for all of the data resident in Hadoop. It is a central location for analysts to find datasets they may be interested in or that they may need to complete their specific analytics tasks.

The Data Catalog is populated by many of the data preparation functions described in previously. Data preparation is the process of defining what the catalog sees. It is important to remember that, in data preparation and in implementing the catalog, Platfora does not do anything to the data per se. The end state of data preparation is the catalog entry, which can be modified at any time.

That catalog is flexible to cover everything in the Hadoop data lake; however, only sets of data that have been identified as described above are included in the catalog. The catalog presents an example of the Hadoop tenet of Schema on Read, meaning it represents the way data will be used rather than the way it is stored. Multiple versions of ways to use the same data are possible.

One feature that makes the Platfora catalog unique is that it is entity-centric. In other words, the catalog is not centered around rows or columns, but rather objects or entities (customers, products, etc.) defined within the data. In order to be used in analytics, entities have to be defined by references. Users have to define the relationships between entities in order to leverage them within analytics.

Defining entities makes it possible to search for classifications rather than records. The entity concept is also very important for event-series data. So, for example, it is possible to classify entities by events. In a marketing analysis scenario for retail, users might be interested in knowing who clicked on the homepage and then put a particular item in their shopping cart. Platfora makes this kind of complex analysis very straightforward and simple.

The catalog also has an important role to play in managing data governance and lineage. Because the data always remains where it is, under the control of IT or the analytics group, correct data governance is much easier to ensure in a Platfora environment than a traditional BI/data warehouse environment. And no matter how many changes the data is subject to in memory, the raw data remains in place and unaltered, making data lineage very easy to track.

SECURITY

To ensure maximum security throughout the Platfora deployment, the Platfora security models is built on three axes: data, object/document, and role.

Data Security

Platfora's data security scheme is applied directly to the raw data in Hadoop and is designed to keep data out of the hands of people who shouldn't have it. Platfora can operate in two modes: one where permissions are inherited from the Hadoop filesystem and the other where permissions are set from within Platfora.

Object/Document Security

Platfora ensures object/document security by managing the owner, editor, and viewer collaboration roles. Similar to a web document, this security model invites collaboration between groups, but defaults to data security to prevent sharing data that should not be shared.

Role-Based Security

Within the Platfora system, permissions are granted based on roles. So, for example, some roles will be assigned the ability to add datasets and create lenses where others will allow only for the access of (some) Vizboards. This role-based approach ensures both that critical security requirements are addressed while still providing tremendous flexibility in extending analytics as broadly and deeply into the organization as required.

Other Security Features

In addition to the three axes of security detailed above, Platfora provides the following security capabilities:

Authentication
Also included are plugable authentication modules that allow the environment to integrate with systems such as SAML. Platfora also integrates with Kerberos for authentication. Platfora provides integration with enterprise LDAP/active directories for groups and roles.

Identity Management
For purposes of identity management, LDAP and the active directories don't actually log users in. They are used to disambiguate identities across multiple systems, keeping users in sync across different systems and platforms.

API

Platfora is a multi-tiered system, with all communication between tiers performed via a RESTful API. Most system functions have a public RESTful API interface that can be programmed to. Platfora enables programmatic query access via RESTful API, that is, querying without using the Vizboard layer. This makes it possible to integrate with third party data science tools such as R and other technologies, including complex workflow engines, business rule engines, etc.

CONCLUSION

Platfora provides a solution framework for big data discovery that addresses both the technical requirements and the broader business challenges that organizations face today. Platfora does more than enable data discovery within large data sets. Managing a wide variety of data structures and types, Platfora enables analysts and business users to generate new insights from these large volumes and multiple types of data, and make those insights more broadly available across the organization than ever before. Addressing a new business reality, Platfora provides a fundamentally new architecture for delivering data analysis.

ABOUT PLATFORA

Platfora is the #1 Big Data Analytics platform built natively on Hadoop and Spark. Platfora enables business users and data scientists to visually interact with petabyte-scale data in seconds, allowing them work with even the rawest forms of transaction, customer interaction, and machine data to find new opportunities and manage risk. Platfora is transforming the way businesses unlock insights, make decisions, and produce better outcomes through the use of its industry-defining Customer Analytics, Security Analytics, and Internet of Things solutions. Platfora Big Data Analytics is creating data-driven competitive advantages for leading organizations such as Citi, Sears, AutoTrader, Disney, Edmunds.com, Opower, Riot Games, Vivint, and TUI Travel. To learn more about Platfora, read our blog or go to **www.platfora.com**