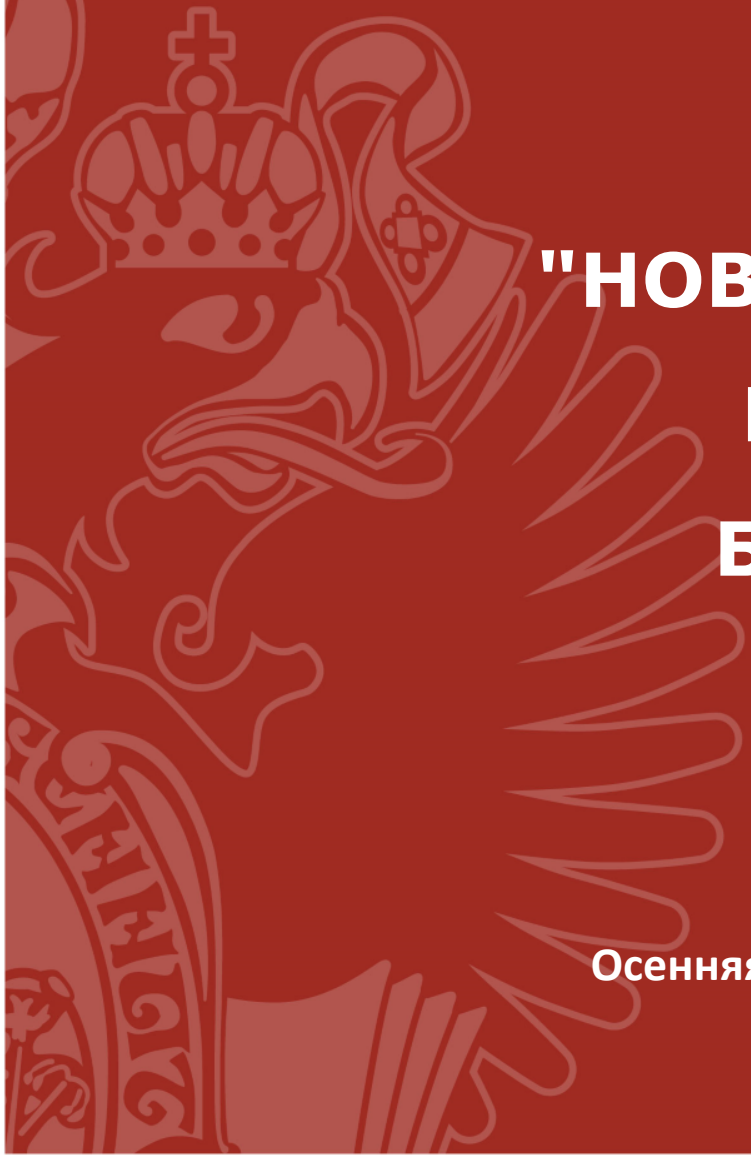


St Petersburg University
www.spbu.ru

"НОВЫЕ МЕТОДЫ ХРАНЕНИЯ ИНФОРМАЦИИ ДЛЯ БОЛЬШИХ ДАННЫХ"

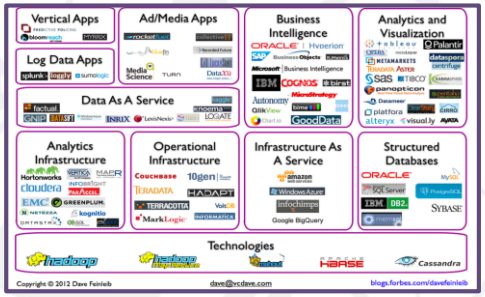
А.В. БОГДАНОВ

Осенняя Школа по информационным технологиям ОИЯИ

Дубна, 10 октября

Tools and frameworks for Big Data

Big Data Landscape 2012

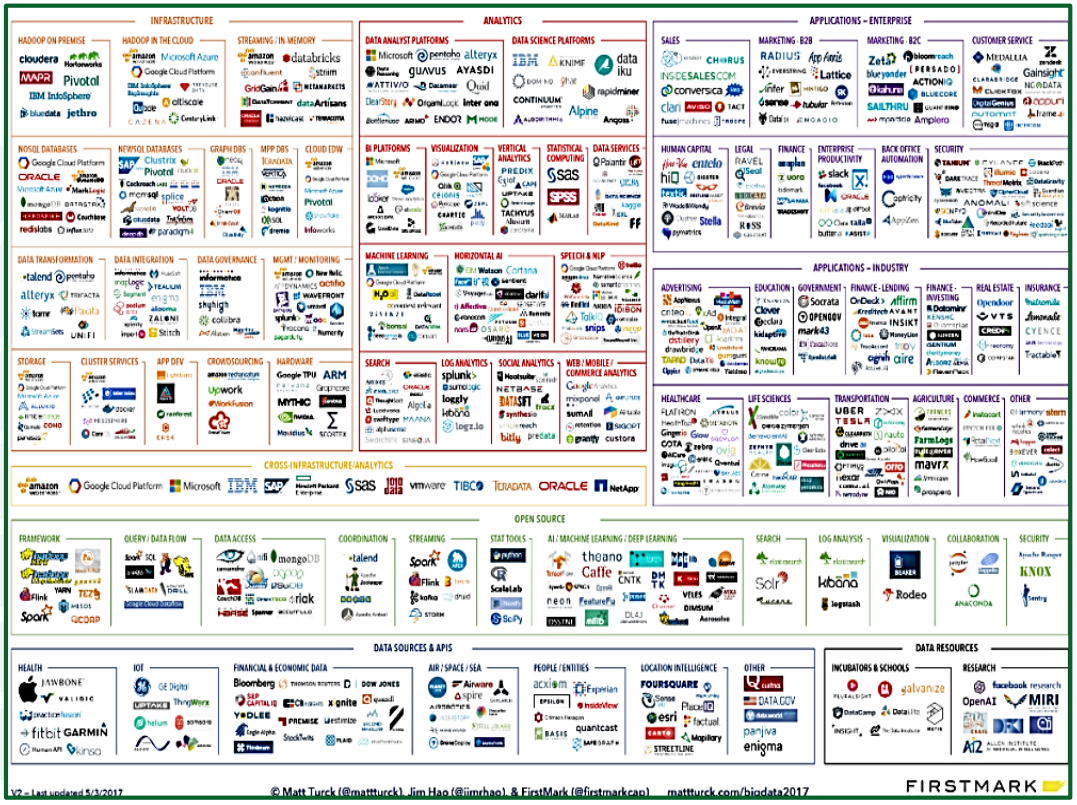


- A sharp increase in the **number** of technologies.
- Technologies develop on the principle **of integration with artificial intelligence**

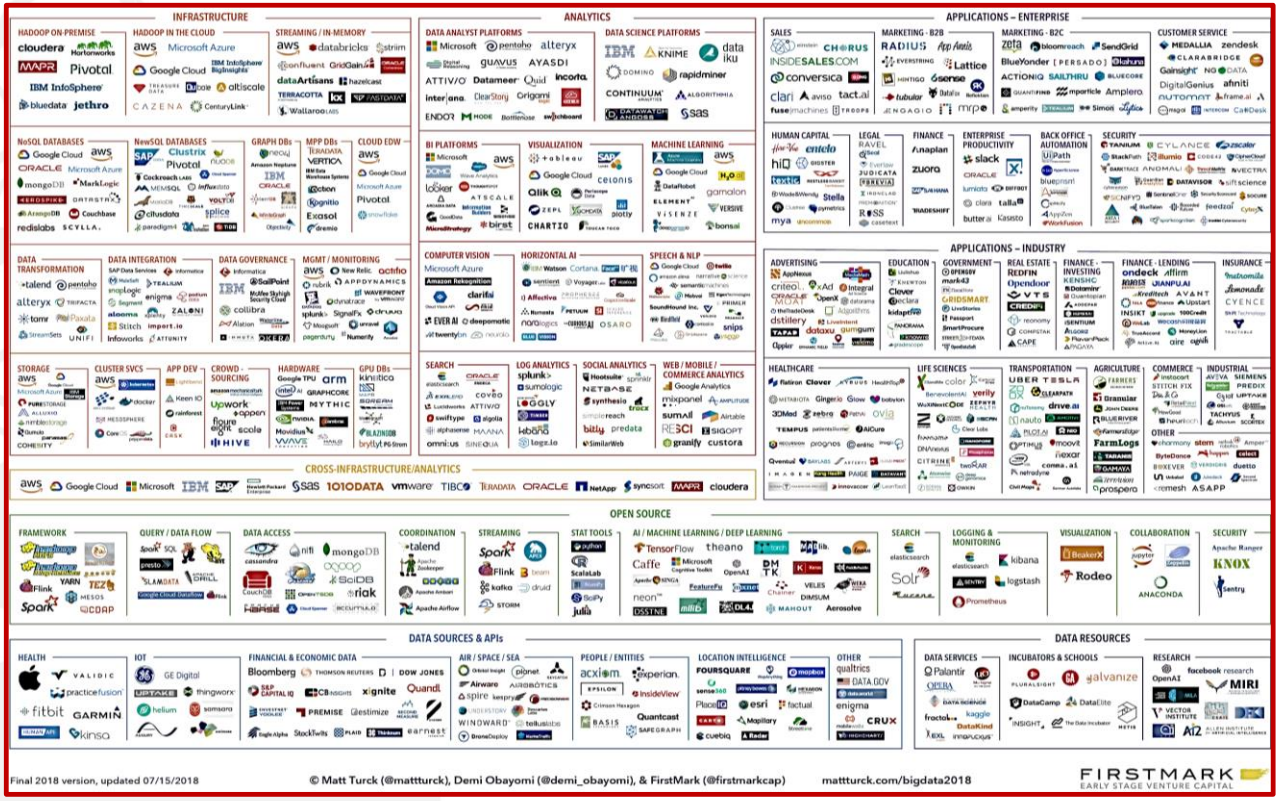
(Big Data + AI = The NewStack.

many tools created **specifically** to **solve new** business **problems**.

Big Data Landscape 2017



Big Data Landscape 2018



Improving the effectiveness of data-related technologies
(big data, data science, machine learning, artificial intelligence)
and their widespread adoption at various enterprises around the world.

The ideal big data technology and process architecture

Data Sources

- Relational databases
- Mainframe
- Documents and emails
- Social media, third-party data, log files
- Machine sensor
- Public cloud
- Private cloud

Data Ingestion

Batch load

Change data capture

Data Management

- Data integration
- Data quality
- Virtual data machine
- Data security
- Data warehouse

Data Delivery

Batch load

Data integration hub

Applications

- Visualization
- Mobile apps
- Analytics
- Business intelligence
- Real-time dashboards

How to choose ?

Distributed Data Network is a new paradigm

In order **to decentralize the monolithic data platform**, it is necessary to **change** our **understanding** of the **data**, its location and ownership.

Transferring data from domains to a lake or a centrally owned platform

Domains

- Ownership of the data sets is delegated from the central platform to the domains
- Domains host and maintain their data sets in an easy-to-use form
- The source domain data sets should be separated from the internal data sets of the source systems
- To provide data cleaning, preparation, aggregation and maintenance, as well as the use of the data pipeline
- The teams that manage the domains provide the ability to process their data to other specialists in the organization via APIs

Date

- Have a much larger volume, are invariable synchronized facts and change less frequently than their systems
- The source domain datasets are the most fundamental datasets and change less frequently, as business facts do not change so often
- Source domain datasets are raw data at the time of creation and are not customized or modeled for a particular consumer

! A secure and manageable global control of access to data sets should be implemented

To ensure a quick search for the required data, a registry must be implemented, a data catalog of all available data containing meta-information

Virtual Data Model Requirements

The **distributed data network** as a platform

is focused on domains belonging to independent groups

that have data processing engineers and data owners using a common data infrastructure as a platform for hosting, preparing and maintaining their data assets.

A **mesh data network platform** is a **specially** designed **distributed data architecture** with centralized management and standardization for interoperability that is provided by a common and consistent data self-service infrastructure.

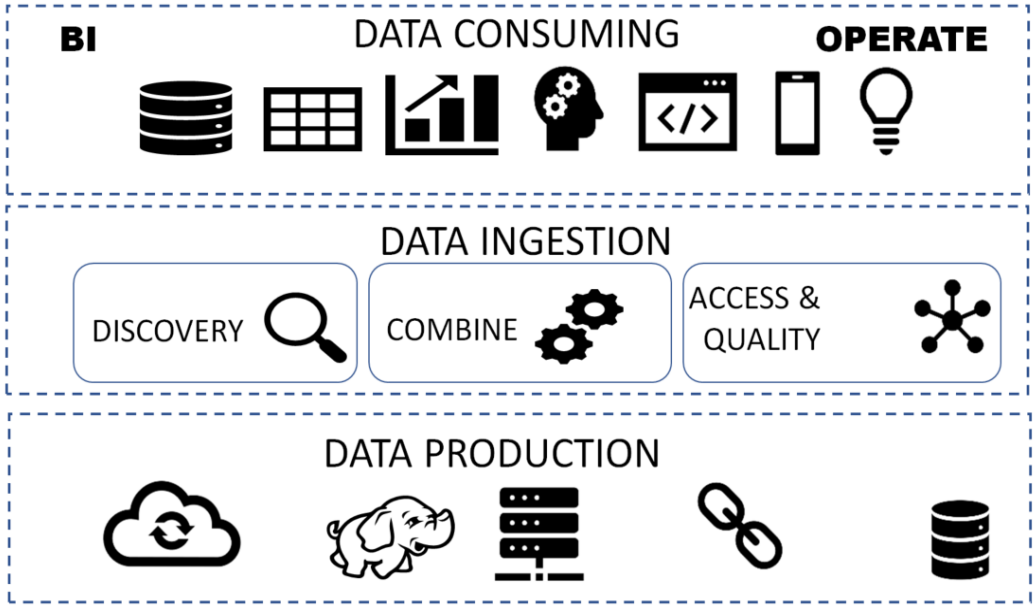
Formal requirements to form a virtual data model:

- **Abstract representation of data** in terms of the object model and its sections (rejection of a rigid structure due to a mesh data network).
- **Differential confidentiality** allowing to determine access parameters on the fly depending on the general role model.
- **API-centering** data management systems for loading data on demand.
- **Refusal from strict separation** of streaming and batch processing of data with the necessary switching on the fly, as a part of the implementation of the KAPPA architecture);Building a feedback system based on a generalized metadata model.

Data Virtualization

Logical Data Storages

(that may be accessed through SQL, REST and etc.)



This grants **access to data** from a large number of distributed sources and various formats, without the requirement for the users to know where it is stored.

This **eliminates** the necessity to **move data** or to allocate **resources for** its **storage**.

Apart from greater effectiveness and faster data access, data virtualization may give the necessary basis for fulfilling the requirements of **data management**.

Data Lake vs Data Warehouse

Data Lake - this is the concept of a centralized storage that allows you to store structured data from relational databases (rows and columns), semi-structured data (CSV, magazines, XML, JSON), unstructured data (emails, documents, PDFs) and binary data (images, audio, video) **with unlimited scalability.**

\+ This approach allows you **to store data in its original state**, without first having to structure the data and run various types of analytics - from dashboards and visualizations to big data processing, real-time analytics and machine learning to make the right decisions.

−
- Raw data is stored without content control.
- It is necessary to have certain mechanisms for cataloging and protecting data. Without these elements, data cannot be found or “credible”, which leads to the appearance of a “swamp”.
- Security issue.

It is essential that the data lake has management, semantic consistency and access control

Data API

To be able to work with BIG DATA it is necessary to have a set of tools, that we call Ecosystem, out of which the most important is API:

API is a business capability delivered over the Internet to internal or external consumers

- Network accessible function
- Available using standard web protocols
- With well-defined interfaces
- Designed for access by third-parties

The key features of API are management tools that make it:

- Actively advertised and subscribe-able
- Available with SLAs
- Secured, authenticated, authorized and protected
- Monitored and monetized with analytics

Data API: Unified approach to data integration

- Conventional APIs: Web, Web Services, REST API – not built for analytics;
- Database paradigm: SQL, NoSQL, ODBS and JDBC connectors – familiar to analysts;
- Database Metaphor + API = Data API;
- Specific API for every type of big data (every “V” and their combinations) – under a generic paradigm.

Methods for testing Big Data applications

Step 1: Data Staging Validation

- Data from various sources like RDBMS, weblogs etc. should be validated to make sure that correct data is pulled into system.
- Comparing source data with the data pushed into the Hadoop system to make sure they match.
- Verify the right data is extracted and loaded into the correct HDFS location

Step 2: "Map Reduce"Validation

- Map Reduce process works correctly
- Data aggregation or segregation rules are implemented on the data
- Key value pairs are generated
- Validating the data after Map Reduce process

Step 3: Output Validation Phase

- To check the transformation rules are correctly applied
- To check the data integrity and successful data load into the target system
- To check that there is no data corruption by comparing the target data with the HDFS file system data

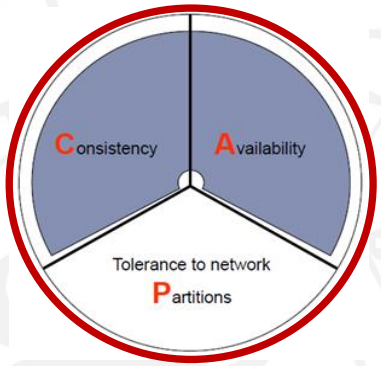
Step 4: Architecture and Performance Testing

- Data ingestion and throughput;
- Data processing sub-component performance.

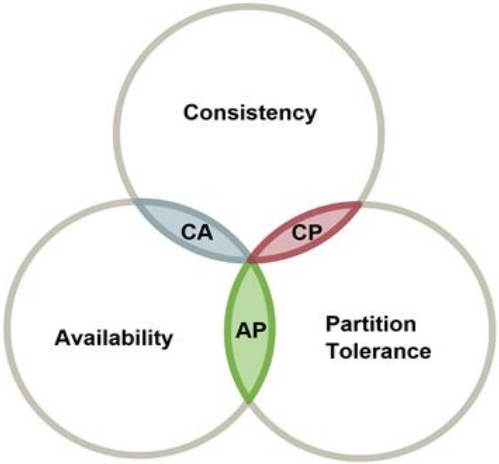
Big data type definition

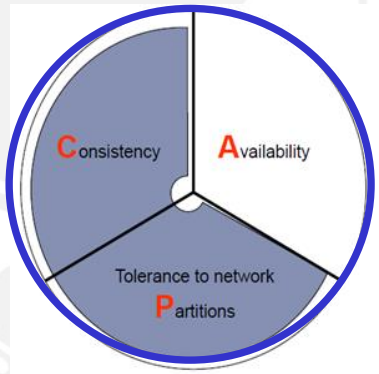
1) **Estimate the total system parameters** (maximum number of users for simultaneous operation, the ability to scale services, the availability of personalized access).
2) **Evaluate the project** (having its own server capacity, cost comparison with the cost of building rental of services).
3) **Evaluate time data access**, query performance evaluation for cloud infrastructures.
4) **Construct the automatic allocation system** and send requests in a distributed database.

Classification of systems based on CAP theorem

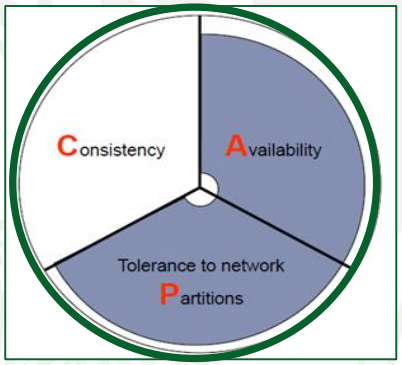


CA - System provides **high available consistency**. To manage data for several resources in these systems, methods such as, for example, two-phase commit are used.





CP - The system provides **strong alignment with the separation tolerance**. Pessimistic locking methods are used to manage data across multiple resources.

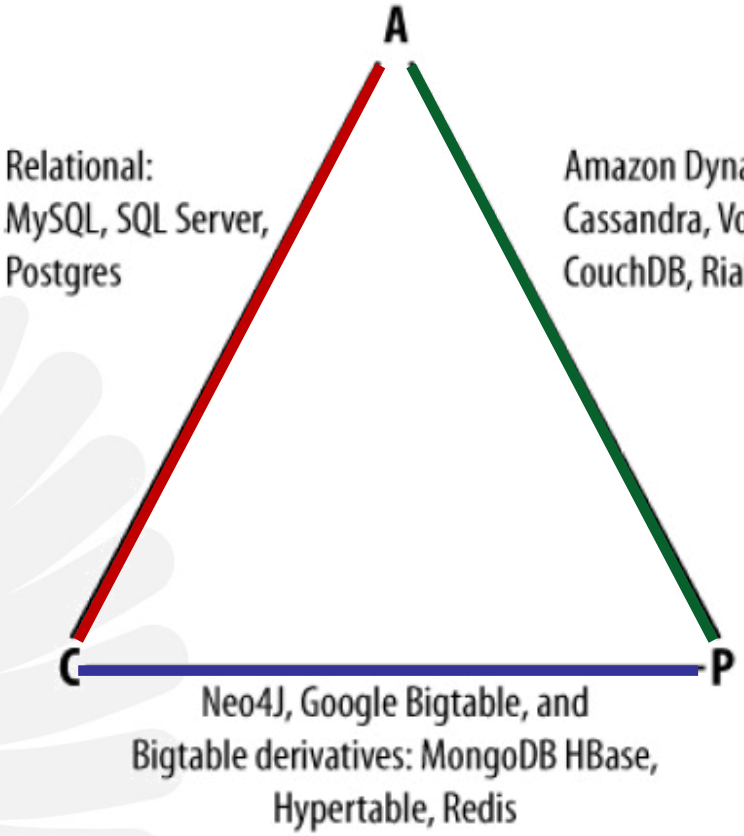


AP - The system assumes **full availability, weakened consistency**. Optimistic locking methods are used to manage data across multiple resources.

Types of Databases (CAP theorem)

Postgresql:

- Master-Slave Architecture
- Synchronization with Master in asynchronous / synchronous mode
- The transaction system uses a two-phase commit to ensure consistency.
- If separation occurs:
  - ✓ impossible to interact with the system;
  - ✓ the system cannot continue to work, but ensures strict consistency and is accessibility.



Cassandra:

- Uses master-master replication scheme.
- Network separation leads to the self-sufficient functioning of all nodes.

MongoDB:

- One Master node, which allows recording only in it, and this guarantees strong consistency
- Automatic change of the master, in case of separation from the other nodes.
- In the event of a network separation, the system will stop accepting records until it is satisfied that it can safely complete them.

Perfect Big Data

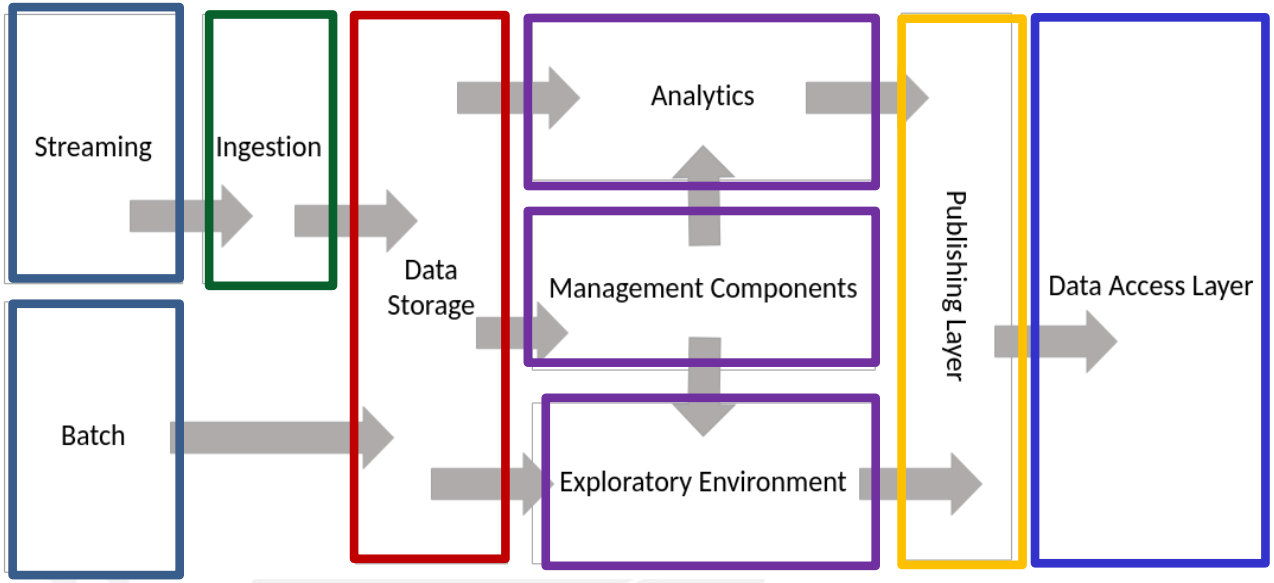
Specifications:

- Scalability
- Fault tolerance
- High Availability
- The ability to work with data in a wide access, but protected
- Support for analytics, data science and content applications
- Support for automation of data processing workflows
- Integration with popular solutions
- Self-healing ability

Ecosystem Tools Implementing:

- **Data collection**
- **Data storage:** the ability to organize heterogeneous data storage, providing real-time access to data, data storage using distributed storage (including the organization of data lakes**)**
- **Data research:** data extraction, formation of a complete picture of data, data collaboration
- **Data management:** creating directories, providing access control, data quality control and building information processing chains
- **Data production**: building a data infrastructure, data delivery, integration issues.

List of components



Data Source Layer:

Streaming – relational databases (RDBMS), social networks, web services, sensors, etc.

Batch – data received from the database and file system.

Ingestion layer – priority is given to data, they acquire category and sent to the **Data Storage Layer**, where they are stored in a very different form (depending on the type of storage selected).

Analytics Layer – basic data processing. **Management Components** – data cataloging, data processing chain building and auditing.

Exploratory Environment – the implemented functionality is tested, experiments are being conducted (storage replacement, data processing, data enrichment using additional data from the main storage).

Data from this level may be published on **Publishing Layer** – level of data presentation. Using a special API, access to the representation of this data is formed – **Data Access Layer**.

Modern Trends

- **It is impossible to completely move away from the concept of SQL** - this is due to the support of the databases that were used earlier, as well as the proposed convenient tools for working with data (the presence of joins, the creation of complex queries).
- **Companies** that are faced with new volumes of data **begin to implement their solutions at the local level**, which means that the same problem is solved repeatedly.
- Some companies are trying to switch to NewSQL, if possible and **need transactionality**.
- Large **companies** began to support **the development of data ecosystems** (data platforms) and tools for building such systems.
- Many existing **ecosystems have limited functionality** (for example, only for the Hadoop ecosystem tools of only their Hadoop add-ins).
- There is a tendency for software developers **to cover** as **many product use cases as possible.**

Virtual SuperComputer

The **virtual supercomputer (VSC)** is a **concept** of creating an application-centric computational environment with configurable computation and network characteristics **based on virtualization** technologies used **in distributed systems**.

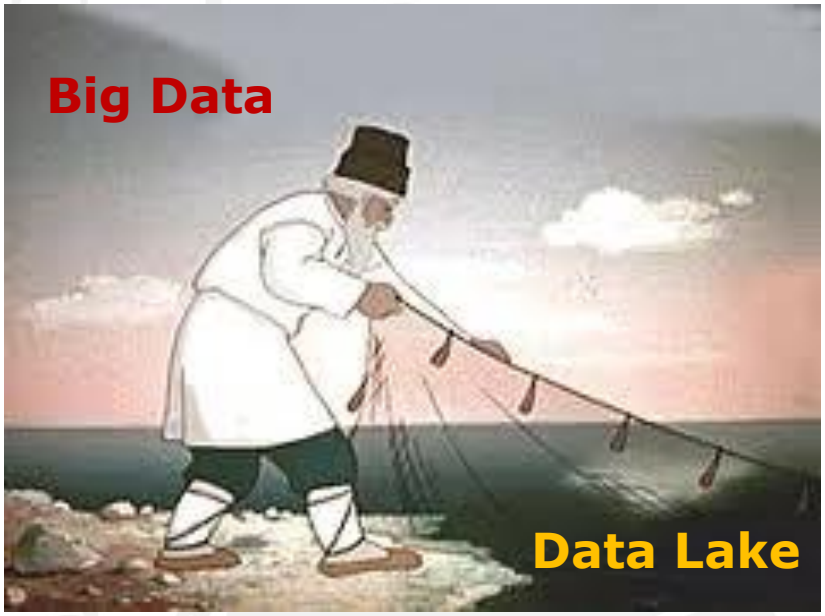
It **enables** flexible partitioning of available resources depending on application requirements and priorities of execution.

General principles

- A **VSC** is completely **determined by** its application programming interface (**API**): API is independent of the platform, takes form of a high-level programming language and is the only way of interacting with the computer.
- The **VSC API provides functions to integrate** with other such systems seamlessly. It allow us to scaling a VSC to solve problems that are too complex for one VSC .
- **Efficient data processing** by VSC is achieved by **distributing data** among available nodes and by running small programs (queries) on each host where corresponding data resides; it's helps not only run query concurrently on each host but also minimizes data transfers.
- Using light-weight **virtualization** is advantageous in terms of **performance**.
- **Load balance** is achieved using virtual processors with controlled clock rate, memory allocation, network access and process migration when possible.
- VSC uses complex grid–like **security mechanisms**, since proper combination of GRID security tools with cloud computing technologies is possible.

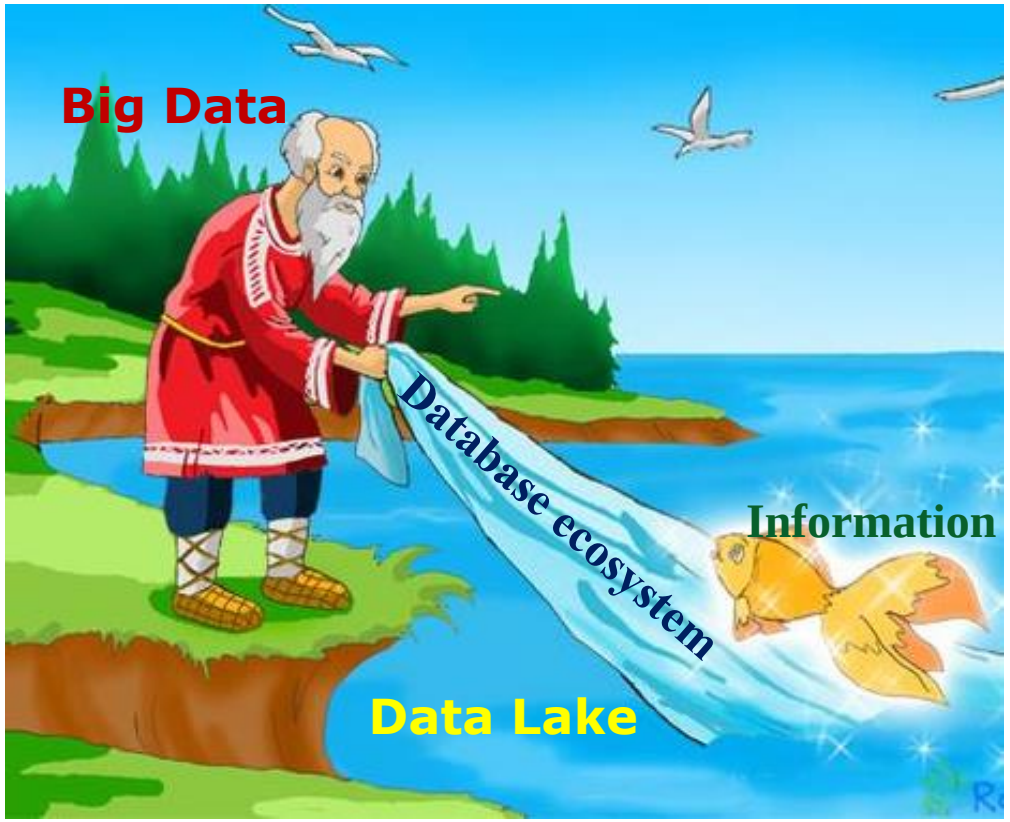
VSC is an API offering functions to run programs, to work with data stored in a distributed database and to work with virtual shared memory in the application-centric manner based on application requirements and priorities.

Big Data & Data Lake



Big Data technologies allow you to process a huge amount of information (which can be as large as hundreds of petabytes) in order to get new, useful information.

A **Data Lake** is a concept, an architectural approach to centralized storage that allows you to store all structured and unstructured data (from mobile applications, IoT devices, and social networks) with the possibility of unlimited scaling



Data Lake vs Data Warehouse

Characteristics	Data Lake	Data Warehouse
Data	Non-relational and relational IoT devices, websites, mobile applications, social networks and enterprise applications	Relationship from transactional systems, operating databases, and business applications
Scheme	Schema-on-read	Schema-on-write
Performance	Query results get faster with low-cost storage	Fastest query results using more expensive storage
Data quality	Any data that may or may not be curated (i.e.. Raw data)	Highly qualified data that serves as a “central version of the truth”
Main consumer	Data scientists, Data developers and business intelligence	Business intelligence
Analytics tools	Machine learning, predictive analytics, data discovery and profiling	Batch reporting, BI and visualization

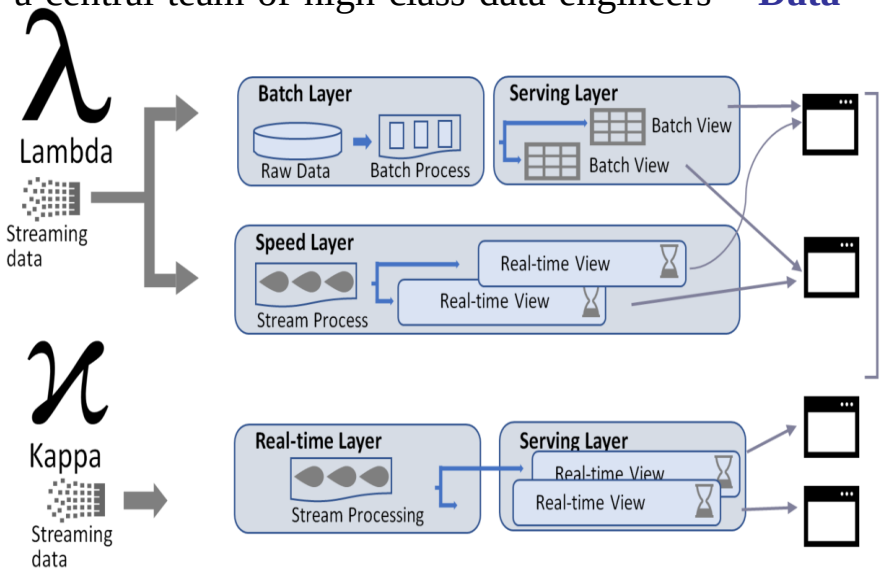
Data Storage Organization

Principal characteristics of modern data platforms: centralized, monolithic, with a tightly coupled pipelined architecture, managed by a group of highly skilled data engineers

1) **Proprietary corporate data warehouses** and **business analytics platforms**, which are awfully expensive solutions that only a small group of specialists understands, which leads to an underestimation of the positive impact of such a warehouse on the business.

2) **Big Data ecosystem** with **data lake**, managed by a central team of high-class data engineers – **Data Marketplace**.

3) **Existing solutions** are more or less similar to the previous generation, **with a bias towards streaming to ensure real-time data availability** with architectures such as Kappa, combining batch and streaming processing for data conversion with platforms such as Apache Beam, as well as fully managed cloud storage services, data pipeline mechanisms, and machine learning platforms.



Such a data platform **eliminates some of the problems of the previous ones**, such as real-time data analysis, but also reduces the cost of managing the Big Data infrastructure.

However, they **keep** part of the **problems** of **previous solutions**.

Data Networks and Data Marketplaces

Core approaches of data storage classification

1) Proprietary data storage.

Highly inflexible, very expensive solutions, involves a small group of specialists ⇨ wasted potential that this storage may have had on the business operations.

2) Big Data ecosystem.

It contains a data lake managed by a centralized team of highly specialized data engineers.

3) Data marketplace.

Are similar to the first two, but lean towards streaming of data and real-time access to insight. Batch and streaming data conversion processes are combined through plat-forms like Apache Beam, Kappa architectures are used, as well as fully controllable cloud storage services, data conveyor mechanisms, and machine learning platforms.

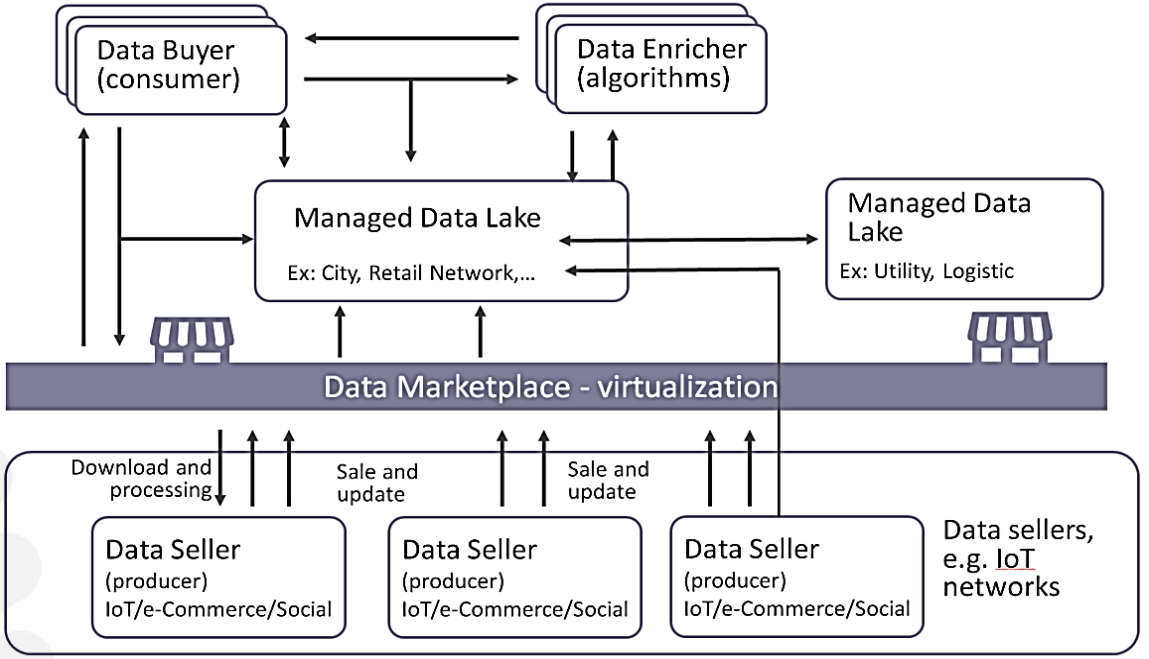
Main problems of using a centralized data platform architecture

1) **Real-time** analysis and **expensive** Big Data infrastructures
2) Continuous **emergence** of **new data** sources
3) Organizations seek to **combine data in different ways** to reflect their fluid business environments and demands. This leads to an increasing number of data transformations, aggregates, projections, and slicing ⇨ the response time rises.
4) When implementing data platform architectures, specialists are influenced by past architecture generations when identifying data processing stages.

Data Marketplace

The **new paradigm** for corporate data platform architectures is the **decentralized data network**.

This paradigm **requires** a **shift in the understanding of data**, its location and belonging.



Instead of **transferring data** from domains into lakes or from centrally owned platforms, there must be an easier way to **store** and **service data**, including duplicating data in **different domains** to allow **greater flexibility** in its **transformation**.

A recent example of such a decentralized platform for data storage is the **DGT Network**. It creates a **virtual data mesh, connecting different sources of data** across corporate information borders into unified analytics accessed by authorized users in a manner conductive to differential confidentiality.

Distributed Ledger Technologies Layer

Model's features for working with big data based on the F-BFT consensus

- Data processing is done in a hybrid consortium-based network built on a federative principle: nodes are grouped in clusters with changing leaders and network access is limited by a set of conditions
- Registry entry is done as a result of "voting" in a cluster and the subsequent "approvals" of an arbitrator node. Both "voting" and "approval" are a series of checks-validations in the form of calculations with binary results
- Each network node receives information and identifies informational objects as one of the Master Data classes
- If an object is new, then there is an attempt to initiate a specialized transaction to insert data into the corresponding registry through a voting mechanism of intermediary nodes
- The distributed data storage system (registry) takes the form of a graph database (DAG, Directed Acyclic Graph) that allows for coexistence of several transaction families for different object classes, while maintaining the network's horizontal scalability

The Artificial Intelligence Layer

The use of artificial intelligence allows for the resolution of important tasks:

- **Clearing text data** using Natural Language Processing (NLP) technologies and extract MD from loosely structured texts. NLP modules can determine the degree of correspondence between objects based on context;
- **Ensuring compliance against set standards** and Master Data management practices; conversion of MD into standard form;
- **High-speed comparison of datasets** (Entity Resolution) based on closeness metrics (most relevant for configurations);
- **Measuring data quality** directly based on support vector machine (SVM) algorithm.

The most in-demand **techniques** that **directly influence** the **quality of Big Data** and the measurement of quality attributes:

- Advanced technique for information objects discovery & identification;
- Data pattern recognition;
- Prediction analysis;
- Anomaly detection.

Conceptual DGT Quality Framework

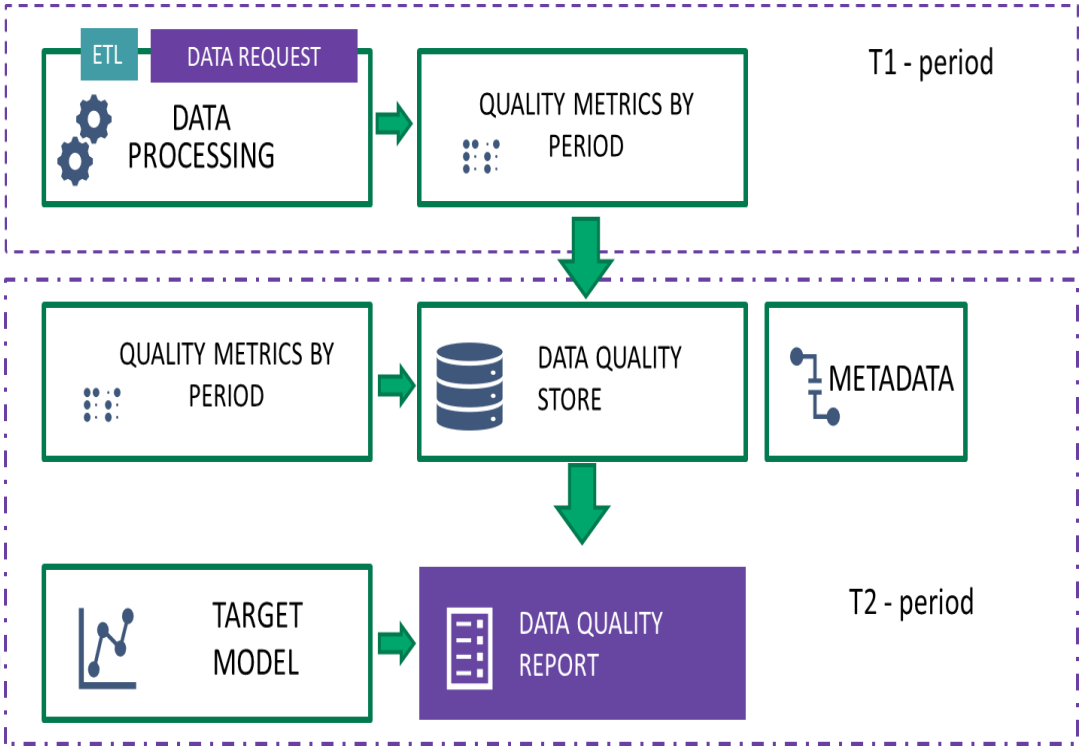
Master Data management styles	Information exchange properties that need to be taken into account
• Transaction-based • Centralized Mast	• The limitations of centralized solutions • Access to data in real time

In the framework of the approach being discussed, these problems are solved by utilizing innovational technologies that support great speed of decision-making and reduce losses due to data mismatch

- The integration layer of the system is built on a high-performance DGT core, which ensures the formation of a unified Master Data registry and its distribution between the participants of an information exchange
- Smart modules (oracles) that track data in real-time and participate in building recon-ciled datasets while simultaneously measuring quality metrics
- Developed API that can plug into not only the different corporate systems and analytic instruments, but also to a variety of instruments of data management and profiling

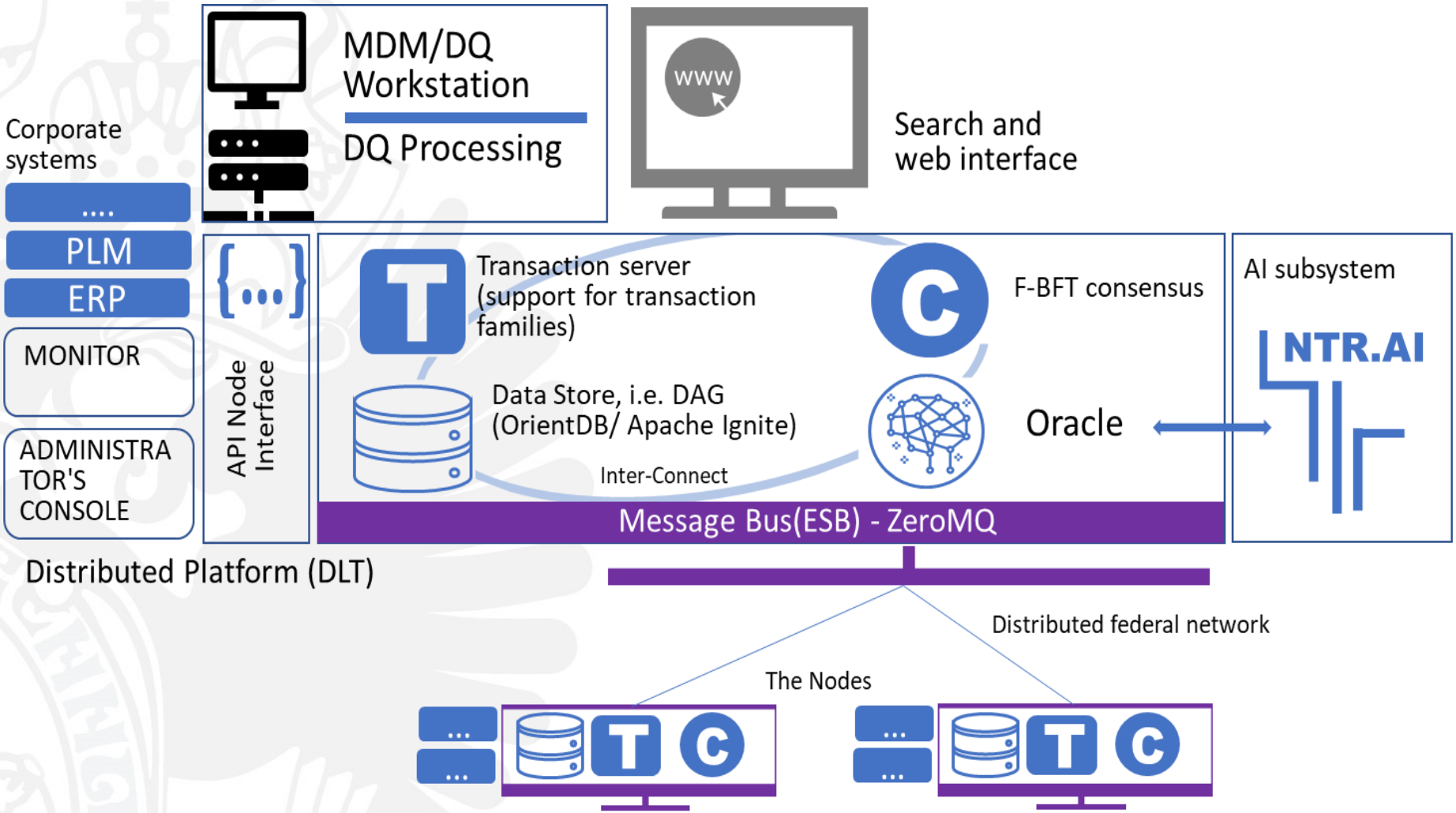
Quality Process Aligning

The total quality ratio can be calculated as weighted average by the following indicators:



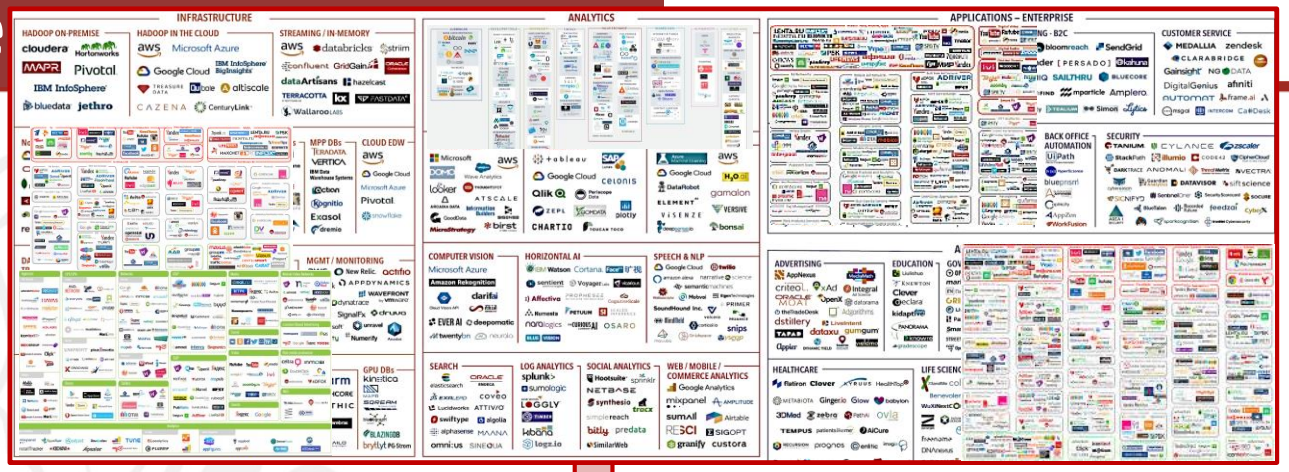
- Number of unidentified (unidentified) objects that have been recovered in the future;
- Data inaccessibility statistics based on frequency of requests;
- Processing data-gathering conflicts, including anomalies and going beyond data validation ranges;
- Distance between the initial and final data vectors;
- Coincidence with results from other sources;
- Timeline lengths and data latency;
- Estimates of cleaning time relative to the overall download cycle

DGT
Framework Implementation



DataLake

Technologies



The choice of technologies and the formation of a stack for data processing, taking into account all their characteristics

Establishing links between technologies, developing an interface for convenient user work

Ecosystem CA

Ecosystem C

Ecosystem CP

Ecosystem A

Ecosystem AP

Conclusion

Currently, the world is experiencing an “oversupply” of data and the main task that humanity now faces is to learn **how** to quickly and safely process it in order **to obtain new information**.

There are two globally intensively developed approaches –

Virtual Data Lakes and Big Data.

Both so far, to a large extent, remain concepts. Convenient and efficient **DataAPI** and the **Big Data Ecosystem** remain key issues. If these tasks are solved, then both approaches will naturally merge.

Thank you for attention!

St Petersburg University

spbu.ru