



St Petersburg  
University  
[www.spbu.ru](http://www.spbu.ru)

# **STORAGE OPTIMIZATION VIA DATA VIRTUALIZATION**

**Bogdanov A., Degtyarev A.,  
Shchegoleva N., Khvatov V.,  
Korkhov V.**

**Saint Petersburg State University DGT  
Technologies AG  
Plekhanov Russian University of Economics**

# Introduction

The fact that over 2000 programs exist for working with various types of data, including Big Data, makes the issue of **flexible storage** a quintessential one.

**Storage** can be of **various types**, including portals, archives, showcases, data bases of different varieties, data clouds and networks. They can have synchronous or asynchronous computer connections.

Because the **type of data** is frequently **unknown** a priori, there is a necessity for a highly **flexible storage** system, which would allow to easily switch between various sources and systems.

Combining the **concept of virtual personal supercomputer** with the classification of Big Data that accounts for different storage schemes would solve this issue.

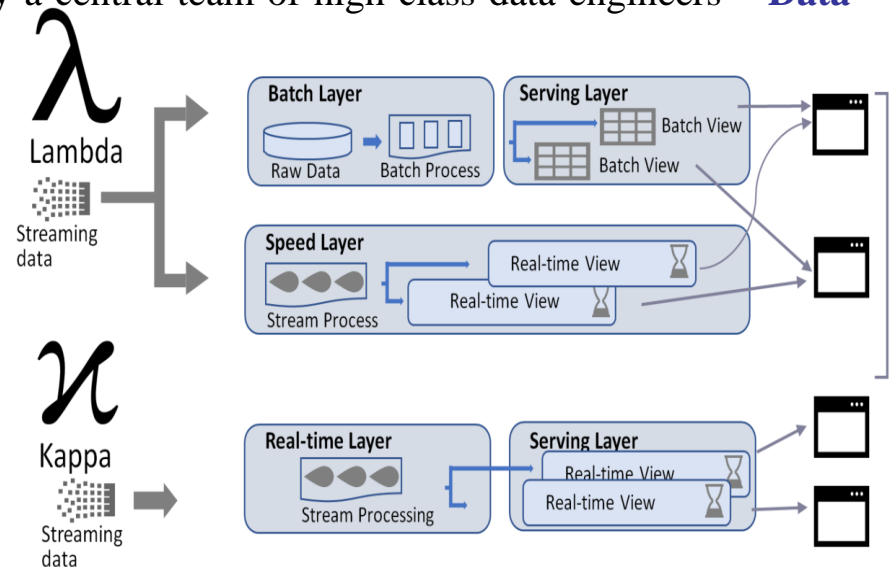
# Data Storage Organization

**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**.

# Centralized Data Platform Problems

## 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.

# 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 do-mains belonging to independent groups**

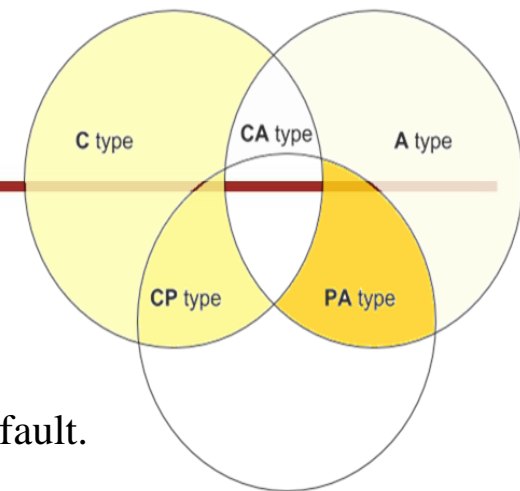
that have data processing engineers and data owners using a common data infrastructure as a platform for hosting, preparing and maintaining their data assets.

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.

# Big Data Types



**Only 5 classes out of 6 potential are possible:**

A **P-class** cannot exist by itself.

Modern corporate architectures are distributed - that is, divided - by default.



the allocation of the P-class as a separate does not make sense, because no such systems are created.

**C - class (consistency)** It is characterized by data that:

- agreed - this is a guarantee that simultaneous reading from different places will return the same value. That is, the system does not return outdated or conflicting data;
- stored in one place (usually);
- may not have backups (there is too much data to do backup for them);
- often analytical data with a short life span.

**A - class (availability)** It is characterized by data that:

- should always be available;
- can be stored in different places;
- have at least one backup or at least one other storage location;
- are important data, but do not require significant scaling.



### CA – class:

- data must be consistent and accessible;
- potentially a monolithic system, without the possibility of scaling or scaling under the condition of instant exchange of information about the changed data between the master-slave nodes;
- there is no resistance to distribution, if scaling is provided for (branches), then each branch works with a relatively independent database.

**In this case, the CA class is divided into 3 subclasses:**

**1). Big data of large sizes** that cannot be represented in a structured way or they are too large (stored in Data Lake or Data Warehouse):

- data has any format and extension (text, video, audio, images, archives, documents, maps, etc.);
- whole data collected, the so-called "raw data";
- large data that is unreasonable to place in the database (unstructured data in the case of data warehouses);
- multidimensional data.

**Example:** medical data that cannot be stored in tabular form (x-ray, MRI, DNA, etc.)



**2). Data of a specific format** that can be represented in a structured form (biological data, DNA and protein sequences, data on a three-dimensional structure, complete genomes, etc.)

- multidimensional data;
- data must be analyzed and their sizes reach gigantic values.

**Example:** medical and bioinformatics data that need to be searched and stored in a relational table. Examples of extensions are xml, json, etc.

### **3). Other data well presented in relational databases**

- have a clear structure or can be represented in the concept of a relational database;
- the size of the stored data does not matter (provided that lightweight objects or links to large objects are stored in the storage);
- transactional required;
- “Raw” data may be, but is not recommended (an exception - if the logs are stored).

**Example:** customer data, logs, clicks, weather statistics or business analytics, personal data, rarely updated, customer base, etc.

**CP – class** It is characterized by data that :

- must be consistent and at the same time there is support for the distributed state of the system, which has the potential for scaling;
- structured, but can easily change their structure;
- must be presented in a slightly different format (graph, document), that is, data for social networks, geographic data and any other data that can be presented in the form of a graph;
- have a complex structure, because of which there is a potential need for storing files in a document-oriented format;
- they accumulate very quickly, so a distribution mechanism is needed;
- no permanent availability requirements.

**AP – class** It is characterized by data that:

- should be available and at the same time there is high support for the distributed state of the system, which has the potential for scaling;
- have a complex structure, the potential need to store files in a different format with the ability to change the scheme without the need to transfer all the data to a new scheme (Cassandra);
- accumulate quickly.

**Example:** this class is suitable for data that is historical in nature. The main task here is to store large amounts of data with the potential growth of this information every day, statistical and other processing of information online and offline in order to obtain certain information (for example, about the interests of users, mood in conversations, to identify trends and etc.)

# 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

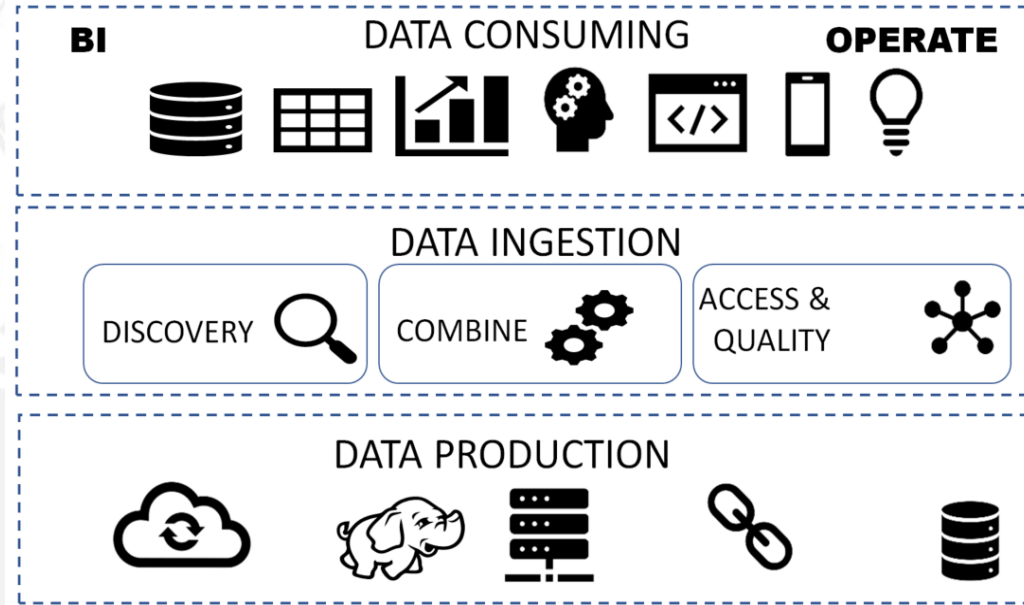
- A **VSC** is completely **determined by** its application programming interface (**API**): API is independent of the platform, takes form of a high-level programming language and is the only way of interacting with the computer.
- The **VSC API provides functions to integrate** with other such systems 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.**

# 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

**Virtualization features that support the scalability and operational efficiency required for big data environments**

- **Partitioning**: sharing resources and moving to streaming data.
- **Isolation**: Transition to the object representation of data with reference to the domain model.
- **Encapsulation**: Logical storage as a single entity

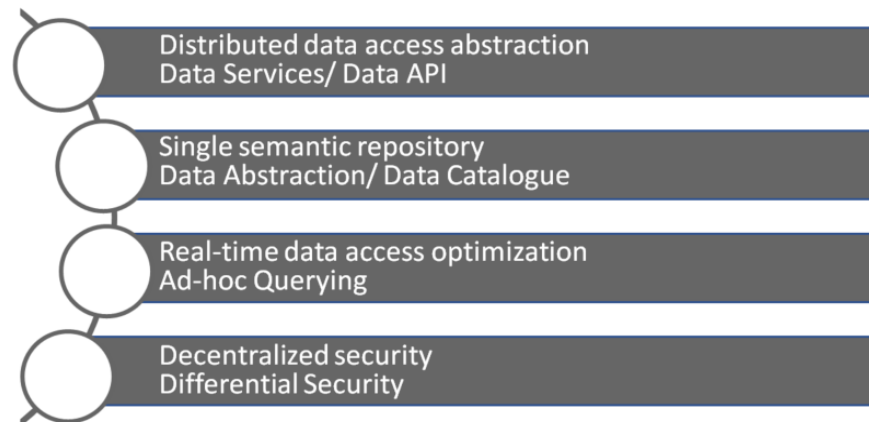
This solution **changes** the general **approach to data** in data access abstraction, semantic storage, real-time data access and decentralized security

**Data virtualization:**

1. Partitioning
2. Isolation
3. Encapsulation



*changes approach  
to data*



# 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

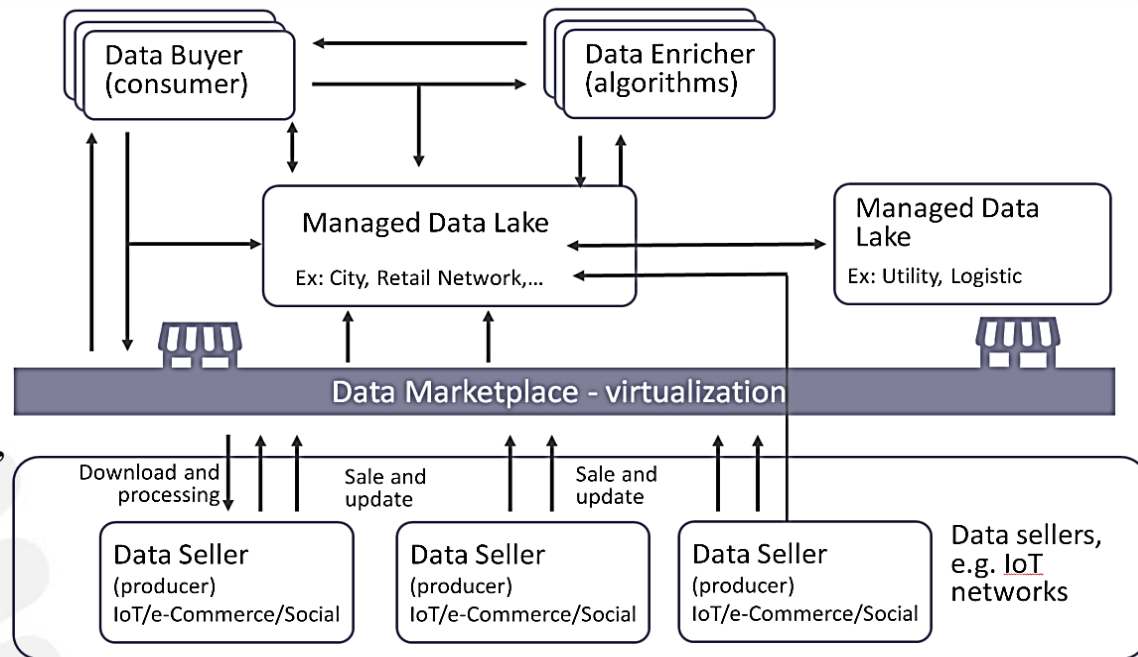
- 1) **Real-time** analysis and **expensive** Big Data infrastructures
- 2) Continuous **emergence** of **new data** sources
- 3) Organizations seek to **combine data in different ways** to reflect their fluid business environments and demands. This leads to an increasing number of data transformations, 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 conducive to differential confidentiality.

# Big Data Virtualization Applications

Studies have shown that data marketplaces already offer several important advantages for companies seeking to put their data to effective work. One of these uses is the construction of ecosystems:

- **Data marketplace powered by the DGT Network** which allows for horizontal integration by creating distinctive clusters of enterprise-operated nodes, which exchange data through a secure F-BFT Protocol and record it in a unified ledger (the Direct Acyclic Graph ledger).
- **Data Marketplace launched by the IOTA Foundation** is an open-source distributed ledger that connects Internet of Things devices to process microtransactions in exchange for cryptocurrency.

**Prospects** As part of by McKinsey research on the underutilization of IoT data by corporate enterprises, they note a typical example where one oil rig with **30,000 sensors examines only 1% of the data** collected to detect and control anomalies, ignoring the greatest value that comes in predictive analytics and optimization.

# Data Quality in a Decentralized Environment

- A study done by the Gartner Group in 2008 shows that on average **organizations lose at least \$8.2 M per year** due to **bad data quality**, operational ineffectiveness and lost opportunities. According to another study [1], **only 12% of organizations use** their data's potential to attain strategic advantages.
- The **situation** is made **worse** by the influence **of Big Data** and by the largely **decentralized nature** of modern business. However, **modern technologies can help** overcome the problem of data quality.
- According to [2], the use of **decentralized registry** and **artificial intelligence** technologies **will elevate data quality** by 50% by 2023. It is true that the inherent qualities of blockchains render them capable of controlling consistency and integrity, which are some of the most important data quality attributes.

- 1) Assessing Your Customer Intelligence Quotient, Forrester, <http://www.forrester.com/Assessing+Your+Customer+Intelligence+Quotient/fulltext/-/E-RES53622?docid=53622>
- 2) Predicts 2020: Data and Analytics Strategies — Invest, Influence and Impact, Gartner Report, 2019

# Measuring Data Quality

## The main DQ Attributes by ISO 9000:2015



## Quality of Big Data

Size, Speed, Diversity, Accuracy and Cost

Unified information frequently includes several data types (structured, half-structured, unstructured)

Semantic differences in definitions may lead to the same positions being filled differently

Differences in formats and syntaxes

Brewer's CAP theorem

# Data Quality Structure Trends

## Crucial trends

### Decentralization

- The necessity of adapting distributed registry technologies to control data quality

### Data Virtualization

- The necessity of abandoning verification according to a given data structure (since it may vary)

A constructive approach to the calculated metrics of data quality is the **selection of stable information** objects and the **application of validation rules** in accordance with them in real time

### Two-phase approach to data processing

Pre-processing of  
incoming data with  
the identification of  
main information  
objects and  
validation of their  
attributes

Processing of quality  
attributes across the  
entirety of available data,  
taking into account the  
discrepancies in versions  
of transactional  
information

It's will **prevent “data depletion”** as a result of loading and **adapt the standard mechanism** for calculating quality under **decentralized data processing** and almost equivalent to creating a reliable master data management system in a big data paradigm and storing information about transaction transactions in as many versions as possible.

# Information Types

- **Transactional (or operational) data** – a rapid stream that describes the changes in statuses of information objects, such as money transfers, product shipments, sensor indicators;
- **Analytic data** – slices of operational data prepared for decision-making;
- **Master Data** – which is necessary for identifying information objects; these are sets of data with a relatively slow rate of change, including normative-directive information, metadata, parameters and configuration of informational objects.

Therefore, **Master Data** is a **conditionally constant set of data** that **defines** the **composition of the domain** being automated and the **basis** for describing the **business logic** of the application. Master Data can have a flat, hierarchal or network structure depending on the existing business processes.

**Master Data** conditionally includes such **subgroups** as directors, metadata and configurations, depending on the existing models of information management and object life cycles.

**Master Data** has a **direct influence** on the **quality of information** and in the context of **distributed decentralized systems**, it requires a **process of agreement hosts** that would represent the different sides of the informational exchange.

# Conceptual DGT Quality Framework

## The approach

### Master Data management styles

- Transaction-based
- Centralized Master Data
- Shared Master Data

### Information exchange properties that need to be taken into account

- The limitations of centralized solutions
- Access to data in real time
- Smart data processing
- Smart data processing

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 reconciled 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



# **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.

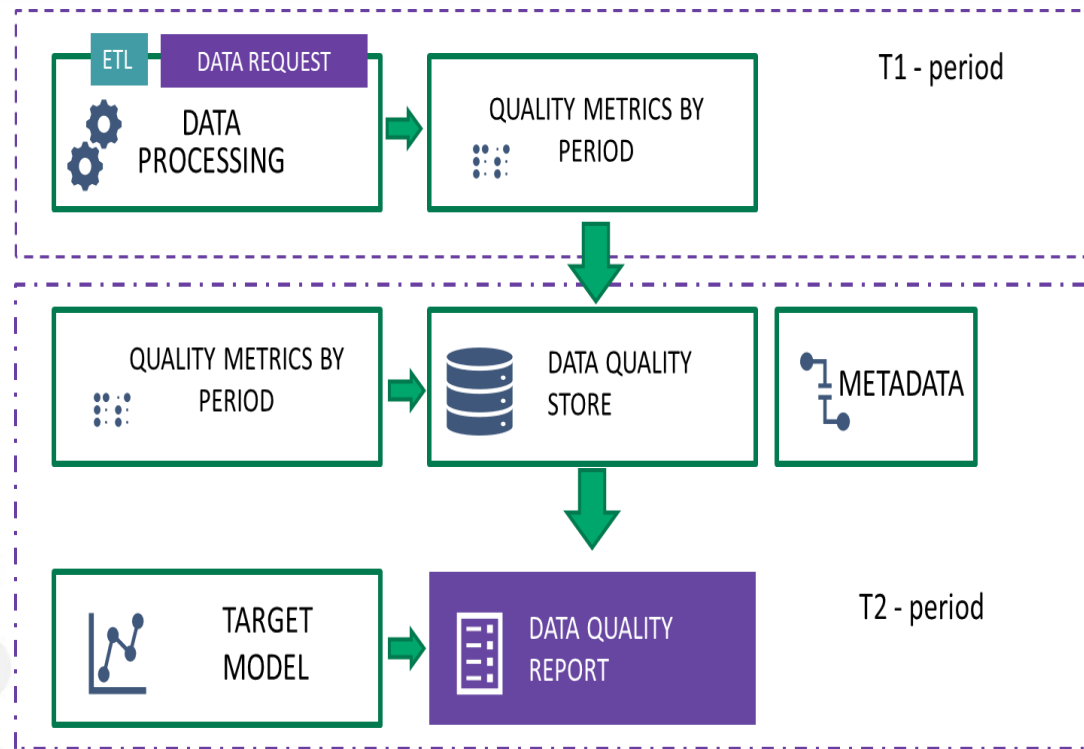
The most in-demand **techniques** that **directly influence** the **quality of Big Data** and the measurement of quality attributes:

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

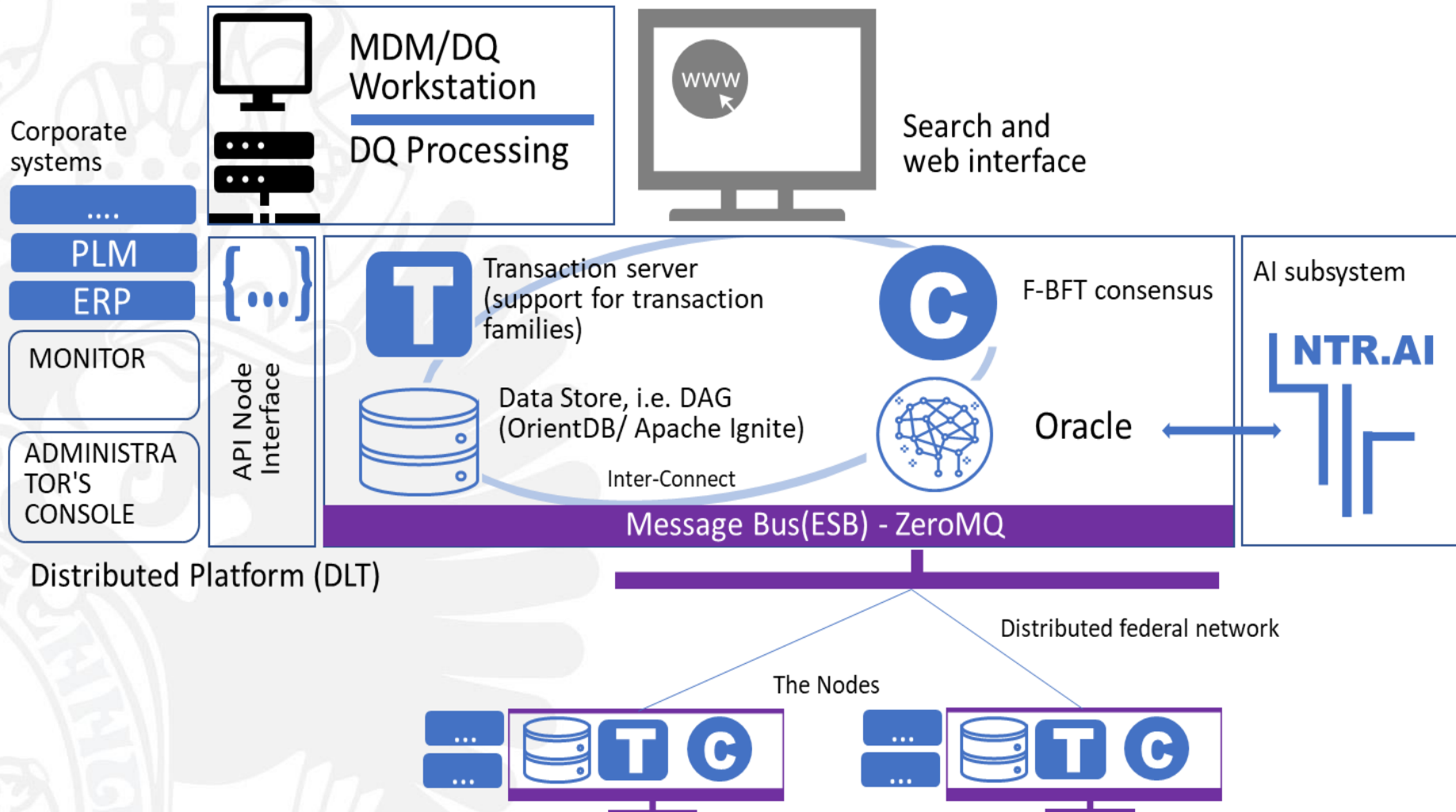
# Quality Process Aligning

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



# Conclusion

**Applying distributed ledger technologies** to the task of maintaining Master Data between organizations will **provide** a **united information space** for **groups of companies** integrated horizon-tally or vertically, allow real-time **quality indexes** to be **calculated** and effective information **exchange** in operational data, **improve** the **quality** of analytical **data**, and ultimately make the **decision-making process itself a quality**.

# Conclusion

- 1) **Data virtualization** is a method of organizing access to data **without requiring** the information about its **structure or place** in any particular information system.
- 2) **The main goal** is to **simplify access** and **use** of data by turning it into a service, essentially **shifting** the **paradigm from storage to usage**.
- 3) **Core characteristics of virtualization** that support scalability and operational effectiveness necessitated by Big Data environments
  - portioning, which is the division of resources and a shift to streamed data;
  - isolation, which is an object-oriented approach to data with domain application in mind;
  - encapsulation, keeping the logical storage as a singular object.
- 4) **Differential security and privacy** are achieved.

**Data virtualization is more than just a modern approach,  
it is an entire new way of seeing data.**



**Thank you for attention!**

St Petersburg University  
**spbu.ru**