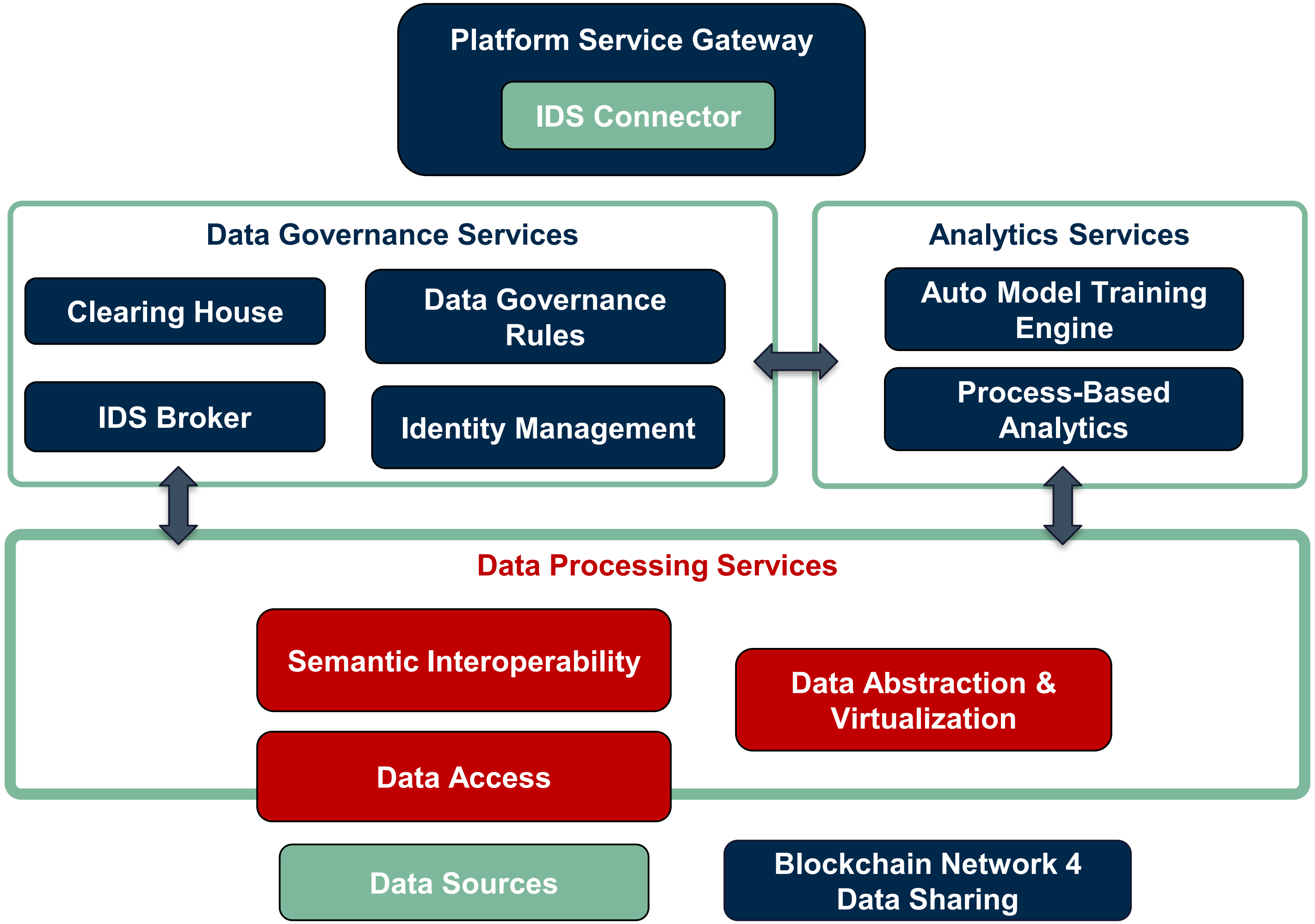
Data Processing and Interoperability Services in DataPorts

Achilleas Marinakis
ICCS / NTUA

DataPorts
Data Platform for the Connection of Cognitive Ports

29 November 2022

DataPorts Architecture



Semantic Interoperability

- **Scope**
  - **To define an unambiguous way to interpret the data that are being exchanged between computer systems**
- **Goal**
  - **To simplify the reuse of data by other applications and components, within the smart ports domain**
- **Challenge**
  - **Enabling interoperability not a trivial task, because different organizations in transportation and logistics do not follow a common standard**
    - **They have their own vocabularies, with poor definition of semantics, or no explicit semantic formulation at all**

Data Modeling

- ## Goal
  - To describe ports data in a systematic and standardized way, with mappings to standard vocabularies in order to facilitate interoperability with other systems

- ## Process
  - Identify and analyze the different data sources to be integrated in the DataPorts platform, considering the meaning and format of the data
  - Review of existing ontologies and vocabularies in the application domain for potential reusability

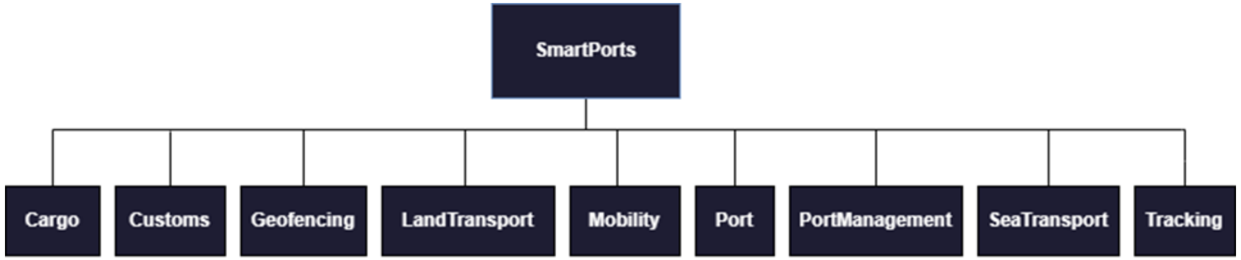
- ## Tangible Outcome
  - DataPorts Common Data Model

DataPorts Common Data Model (1/2)

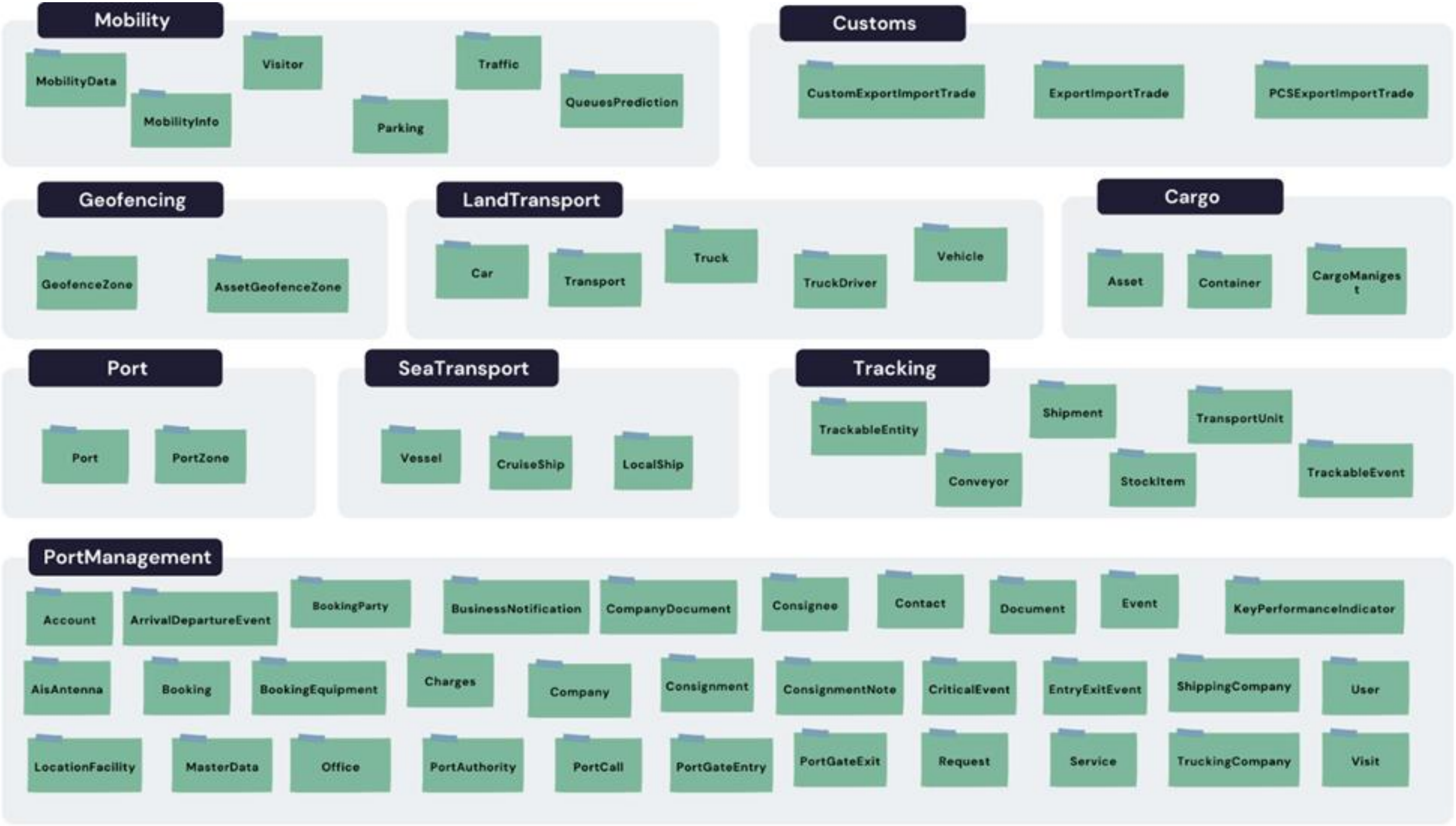
- **Based on Fiware Smart Data Models guidelines**

- **Also fully compatible with other standards and initiatives**
  - **e.g. the Context Information Management API standard (NGSI-LD), set by the European Telecommunications Standards Institute (ETSI)**

- **The *SmartPorts* domain includes many subjects, each one contains different entity types**
  - **e.g. *SeaTransport subject* provides access to *Vessel, CruiseShip, LocalShip entities***



DataPorts Common Data Model (2/2)



Mobility

- MobilityData
- MobilityInfo
- Visitor
- Parking
- Traffic
- QueuesPrediction

Customs

- CustomExportImportTrade
- ExportImportTrade
- PCSExportImportTrade

Geofencing

- GeofenceZone
- AssetGeofenceZone

LandTransport

- Car
- Transport
- Truck
- TruckDriver
- Vehicle

Cargo

- Asset
- Container
- CargoManigest

Port

- Port
- PortZone

SeaTransport

- Vessel
- CruiseShip
- LocalShip

Tracking

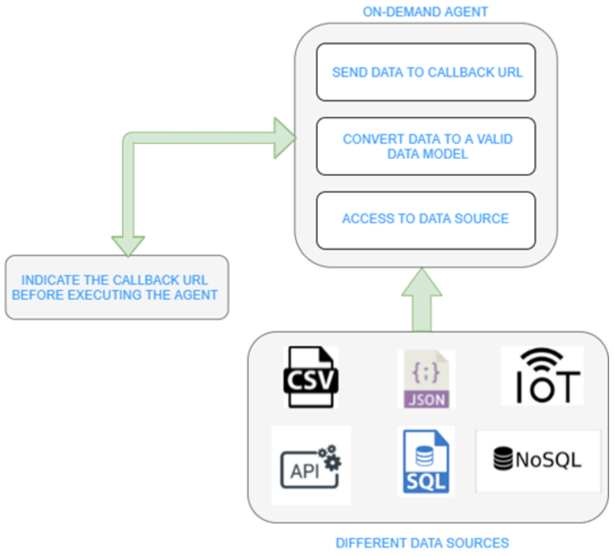
- TrackableEntity
- Conveyor
- Shipment
- StockItem
- TransportUnit
- TrackableEvent

PortManagement

- Account
- ArrivalDepartureEvent
- BookingParty
- BusinessNotification
- CompanyDocument
- Consignee
- Contact
- Document
- Event
- KeyPerformanceIndicator
- AisAntenna
- Booking
- BookingEquipment
- Charges
- Company
- Consignment
- ConsignmentNote
- CriticalEvent
- EntryExitEvent
- ShippingCompany
- User
- LocationFacility
- MasterData
- Office
- PortAuthority
- PortCall
- PortGateEntry
- PortGateExit
- Request
- Service
- TruckingCompany
- Visit

Data Access Component

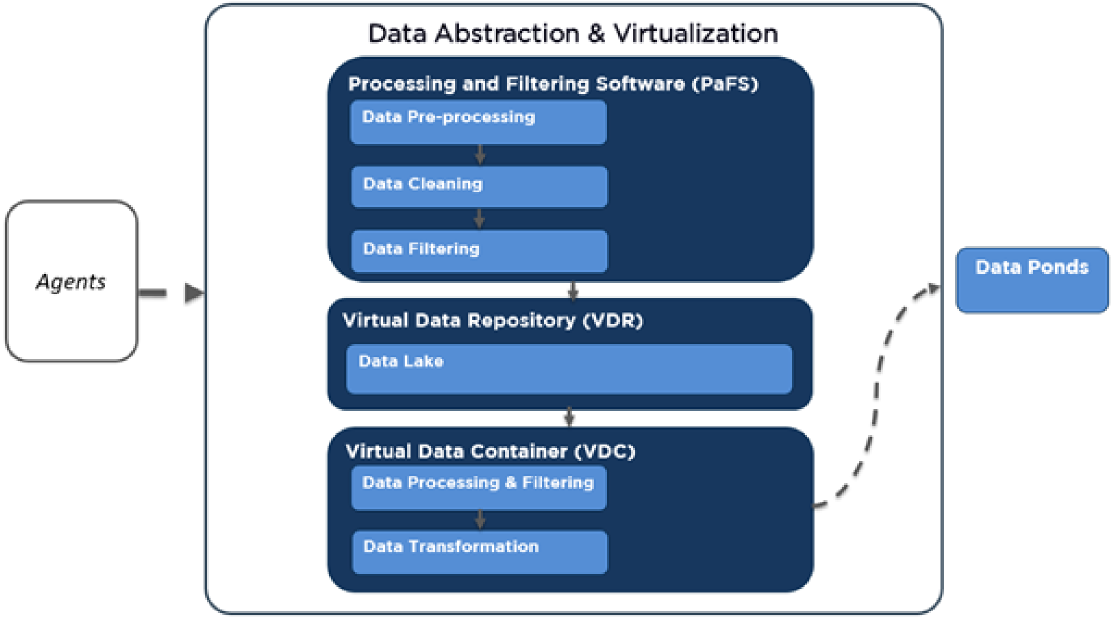
- **Scope**
  - **To connect heterogeneous (in terms of interfaces, formats and models) data sources to the DataPorts platform**
- **Goal**
  - **To develop a scalable software component that obtains data from a source, translates them into the DataPorts Common Data Model and make them available to the processing layer via a unified API**
- **Tangible Outcome**
  - **Dockerized agents to access Data Sources, through a GUI and SDK management framework**


ON-DEMAND AGENT
SEND DATA TO CALLBACK URL
CONVERT DATA TO A VALID DATA MODEL
ACCESS TO DATA SOURCE
INDICATE THE CALLBACK URL BEFORE EXECUTING THE AGENT
CSV
JSON
IoT
API
SQL
NoSQL
DIFFERENT DATA SOURCES

Data Abstraction and Virtualization

- **Scope**
  - **To offer the translated and harmonized Data as a Service, putting emphasis on QoD and QoS**

- **Goal**
  - **To provide an abstraction layer between data providers and consumers**
    - **Let the application developers just define the content and the format of the needed data**
    - **Rely on the proposed middleware to deliver those data, encapsulating all the underlying technical complexity**

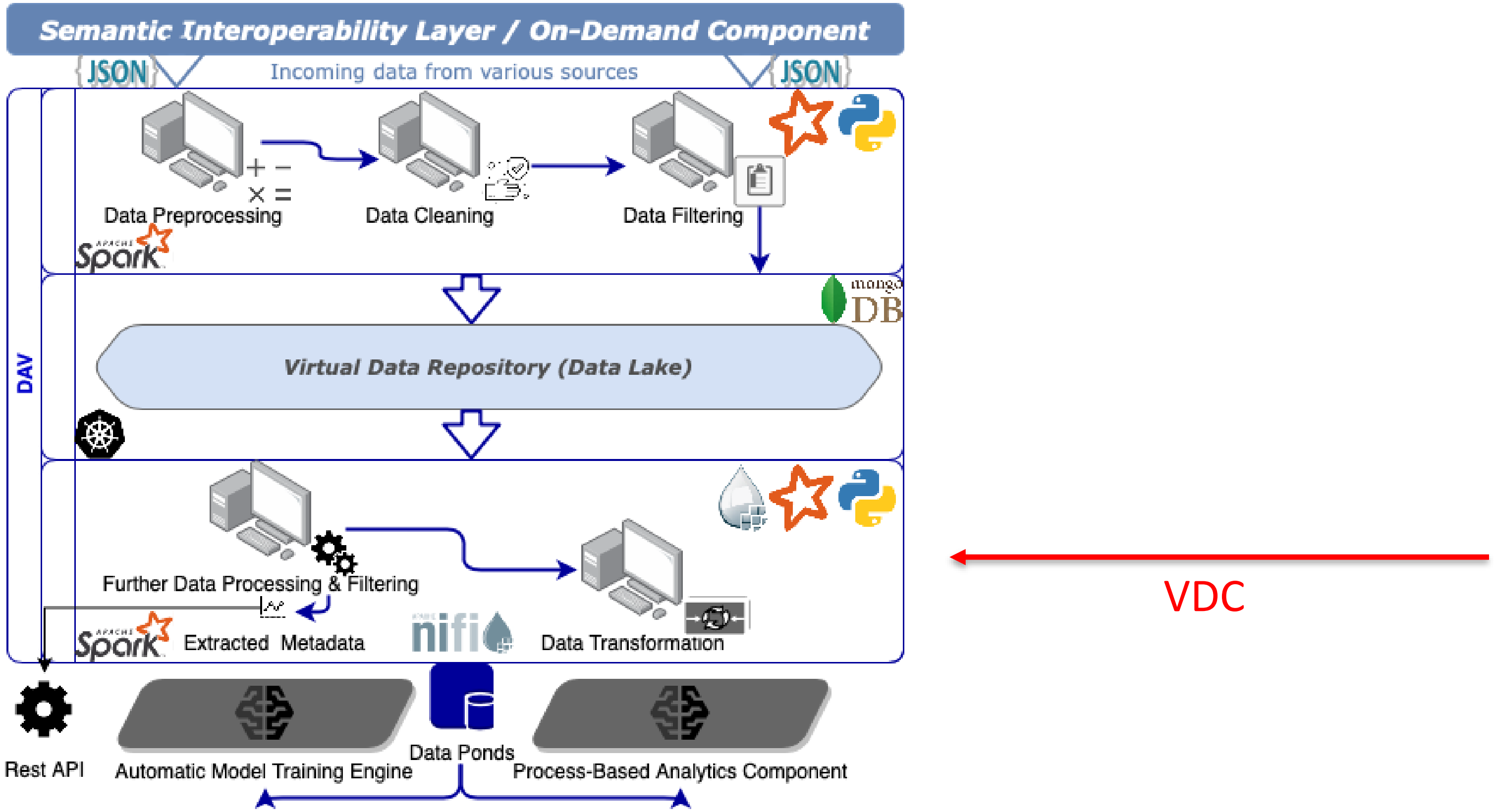

Data Abstraction & Virtualization
Processing and Filtering Software (PaFS)
Data Pre-processing
Data Cleaning
Data Filtering
Virtual Data Repository (VDR)
Data Lake
Virtual Data Container (VDC)
Data Processing & Filtering
Data Transformation
Agents
Data Ponds

Data Processing and Storage

- ## Pre-Processing and Filtering Software
  - Responsible for the initial pre-processing, cleaning and filtering of the incoming data
  - To improve the quality of the data and thus to increase the performance of the applications built on top of them
- ## Virtual Data Repository
  - Distributed infrastructure to store the pre-processed, cleaned, and filtered data
    - Data Lake
  - Capable of upscaling or downscaling in a fully automated way according to the workload
    - Resource optimization
    - Low response times
    - High availability

Virtual Data Container - Overview

- The interface between the middleware and the data consumers (analytics)
- Further process, filter and then serve the needed information as data ponds, by applying specific filtering rules to the data
- Transforms the data to the requested format (JSON, Parquet or CSV)


Semantic Interoperability Layer / On-Demand Component
JSON
Incoming data from various sources
JSON
Data Preprocessing
Data Cleaning
Data Filtering
Spark
mongoDB
DAV
Virtual Data Repository (Data Lake)
Further Data Processing & Filtering
Spark
Extracted Metadata
nifi
Data Transformation
Rest API
Automatic Model Training Engine
Data Ponds
Process-Based Analytics Component
VDC

Virtual Data Container – Filtering Rules

- ## Goal
  - To enable the consumers retrieve only the portion of data they are interested in, thus minimizing the processing and the network workload
  - To detect and remove wrong values, most probably caused by sensors malfunction (e.g. temperature at minus 50 degrees Celsius)

- ## Structure
  - Array of JSON Objects
  - Follow the *subject* – *operator* – *object* syntax (e.g. vessel_arrival_time >= 2022-11-29T08:00:00.000+00:00)
  - Kept simple so that the rules can be specified not only by data scientists and experts, but also by users with limited or no technical background within a port ecosystem

- ## Tangible Outcome
  - A common access layer, where the data consumers simply define queries (in the form of rules) in a unified format
  - VDC designer is responsible for developing the suitable module to execute those queries, according to the selected storage technology (MongoDB in the case of DataPorts)

Thank you for your Attention

Achilleas Marinakis
Research Engineer
ICCS / NTUA
achmarin@mail.ntua.gr