Tailored IoT & BigData Sandboxes and Testbeds for Smart, Autonomous and Personalized Services in the European Finance and Insurance Services Ecosystem

D2.15 - INFINITECH Reference Architecture - III

Title	D2.15 - INFINITECH Reference Architecture - III
Revision Number	3.0
Task reference	T2.7
Lead Beneficiary	GFT
Responsible	Ernesto Troiano - Szymon Ambroziak
Partners	AKTIF ATOS BOI BPFI CP CPH CTAG ENG FTS GFT GLA GRAD HPE IBM INNOV JRC LIB LXS NBG NOVA NUIG RB SIA UBI UNP UPRC
Deliverable Type	Report
Dissemination Level	PU
Due Date	2021-12-31 [M27]
Delivered Date	2022-02-22
Internal Reviewers	INNOV FTS
Quality Assurance	INNOV
Acceptance	Accepted by Coordinator
EC Project Officer	Beatrice Plazzotta
Programme	HORIZON 2020 - ICT-11-2018
	This project has received funding from the European Union's horizon 2020 research and innovation programme under Grant Agreement no 856632

Contributing Partners

Partner	Role[1]	Author(s)[2]	Section(s)
GFT	Lead Beneficiary	Ernesto Troiano - Szymon Ambroziak	0. 1. 2. 3. 3.3. 3.5. 4. 4.1. 4.2. 4.3. 4.3.1 4.4. 4.5. 4.6. 4.7. 4.9. 4.10. 6.1. 6.2. 6.3. 6.4. 6.5. 6.6. 6.7. 6.8. 6.9. 6.10. 6.11. 6.12. 6.13. 6.14. 6.15. 6.16.
GFT	Contributors		0. 10.
GFT INNOV LXS UBI UNP	Contributors		1. 2. 3. 3.3. 3.5. 4. 4.1. 4.2. 4.3. 4.3.1 4.4. 4.5. 4.6. 4.7. 5. 7. 8. 9.
GFT LXS NUIG	Contributors		4.9. 4.10.
ABILAB AKTIF ATOS BOI BPFI CP CPH CTAG ENG FTS GFT GLA GRAD HPE IBM INNOV JRC LIB LXS NBG NOVA NUIG RB SIA UBI UNP UPRC	Contributors		6.
BANKIA CP FBK GFT	Contributors		6.1.
GFT INNOV JRC	Contributors		6.2.
BOI BPFI NUIG TAH	Contributors		6.3.
PRIVE RB	Contributors		6.4.
BOC CP GFT UPRC	Contributors		6.5.
CP GFT NBG RB	Contributors		6.6.
CXB FBK FTS GFT	Contributors		6.7.
BOS GFT JSI	Contributors		6.8.
AKTIF BOUN GFT	Contributors		6.9.
ENG GFT PI	Contributors		6.10.
ATOS CTAG DYN GRAD	Contributors		6.11.
DYN GRAD ISPRINT RRD SILO	Contributors		6.12.
GFT RB WEA	Contributors		6.13.
AGRO GEN	Contributors		6.14.
ABILAB GFT	Contributors		6.15.
GFT NEXI	Contributors		6.16.
INNOV	Reviewer		all sections
FTS	Reviewer		all sections
INNOV	Quality Assurance		all sections

1. Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance
2. Can be left void

Revision History

Version	Date	Partners	Description
0.1	2021-11-01		Version updated from D2.14
0.2	2021-12-24		Version for Work in Progress
0.3	2022-01-07		Version Updated with new pilots
0.4	2022-02-01		Version with updated sections
1.0	2022-02-15		Version for Internal Review
2.0	2022-02-18		Version for Quality Assurance
3.0	2022-02-22		Version for Submission

Executive Summary

The present deliverable documents the Reference Architecture (RA) of the INFINITECH project aimed to develop Smart, Autonomous and Personalized Services in the European Finance and Insurance Services Ecosystem. It is the final version of the RA containing the work and the results developed within the first 27 months to design and build a suitable architecture that fits with the pilots' use cases.

The work on Reference Architecture has been conducted as teamwork and involved all Consortium partners.

D2.15 extends the previous version of the deliverable D2.14 - "INFINITECH Reference Architecture - II", taking into account all the improvements achieved between month 20 and month 27. In fact, D2.13 fulfilled the project milestone MS02 fully specifying the first RA of the INFINITECH platform, while D2.14 contained a refinement and D2.15 a further and final update on new pilots and solutions developed in the second period of the project.

The INFINITECH partners have selected a methodology to work on the RA, identifying it in the "4+1" architectural view model, which is presented in the document. The methodology is based on five different views, from which the structure of the system can be analysed (logical view, process view, development view, physical view and scenarios). Moreover, it will be demonstrated that all the functionalities of INFINITECH environment are properly covered by this model.

The State-of-the-Art survey underlines that some already existing Reference Architectures provide substantial input to INFINITECH, such as the pipelined and workflow approach to support the functionalities of the different Pilots and Use Cases of the project.

Relevant inputs to the task have been considered, in particular the input coming from use-cases considered in task T2.1 and a cross reference matrix. Finally, a layered and high-level reference view and a detailed logical view of the RA are presented. Different layers have been identified (infrastructure, data management and protection, data processing and architecture, analytics, interface, and presentation/visualization). The layers are mainly a mean of classification of the building blocks to form different workflows. The resulting RA provides a schema for building solid workflows and ensures full communication and interaction between all the building blocks, from the data source level (at the infrastructure level of the organizations) up to the Data Stores and Processing Analytics to presentation and visualization applications. High-Performance Computing (HPC) can be distributed at nodes within the platform supporting a high degree of scalability. Moreover, RA considers for external data sources such as public and private Data Lakes, IoT networks and Blockchains. A list of identified building blocks provides the basic functionalities of the INFINITECH reference sandbox for a more general class of use cases. Building blocks will be identified where existing technologies are available while other components will be designed, implemented, and integrated during the tasks belonging to work packages WP3, WP4 and WP5 of the project.

The validity of the RA has been proved by a mapping the workflows of the pilots of the projects, ultimately proving the conceptual approach of the INFINITECH RA.

The present document is a stable and final snapshot of the status of the RA at the date of submission of the deliverable. Even if deliverable D2.15 is a final version of the RA within the Task 2.7 and all the milestones related to the RA of the project have been achieved, the RA constitutes a living solution constantly verified during the continuous project development and in particular with the different pilots. Moreover, the Consortium will promote the RA, along with its methodology and technological advancements, during project dissemination as a more general solution applicable to a broader set of different use cases beyond the original scope in the Financial and Insurance sectors whenever Big Data and AI are to be considered.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Definition
ACID	Atomicity, Consistency, Isolation, Durability
AI	Artificial Intelligence
AML	Anti Money Laundering
AOI	Automatic Optical Inspection
API	Application Programming Interface
APP	Application, usually referred to the INFINITECH WEB application
AWS	Amazon Web Services
BDVA	Big Data Value Association
BFM	Business Financial Management
BI	Business Intelligence
BOC	Bank of Cyprus
BOI	Bank of Ireland
BOS	Bank of Slovenia
CAN	Campus Area Network|Controller Area Network
CCTV	Closed Circuit TeleVision
CD	Continuous Development
CEP	Center (for) Energy Policy
CI	Continuous Integration
CPU	Central Processing Unit
CRISP	Complex-Reduced Instruction Set Processor|Computer Resources Integrated Support Plan
CRM	Customer Relationship Manager
CRUD	Create Retrieve Update Delete - Basic Operations in DBMS
CSC	Common and Secure Communication
CSV	Comma Separated Value files
DB	Data Base
DL	Deep Learning
DLT	Distributed ledger technology
DM	Distributed Memory
DPA	Data Protection Authority
DPO	Data Protection Officer
DSS	Decision Support Systems|Distributed Software System
DoA	Description of Action (also DoW, description of Work, PART A of Grant Agreement)
EKS	Elastic Kubernetes Services (AWS Service)

Abbreviation	Definition
ERA	Earned Run Average
ERP	Enterprise Resource Planning
ESB	Emergency Service Bureau
ETF	Electronic Toll Fraud
ETL	Extract, Transform, Load
EU	European Union
FI	Financial Innovation (INFINITECH beneficiary)
FIBO	Financial Industry Business Ontology
FIGI	Financial Instrument Global Identifier
FIU	Financial Investigations Unit
GDPR	General Data Protection Regulation
GIS	Geographical Information System
GPS	Global Positioning System
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HDP	High Defition Progressive
HPC	High Performance Computing
HTAP	Hybrid transaction/analytical processing
HW	HardWare
IBM	International Business Machines
ICT	Information Communication Technologies
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute (of) Electrical (and) Electronic Engineers
IN	Intelligent Network
IOT	Internet of Things (also IoT)
IP	Internet Protocol
IRA	INFINITECH REFERENCE ARCHITECTURE
ISO	International Organization for Standardization
IT	Information Technology
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KYC	Know Your Customer
LEGO	Low End Graphics Option

Abbreviation	Definition
MDM	Mobile device management
ML	Machine Language
MPI	Message Passing Interface
MVP	Minimum Viable Product|Platform
N/A	Not Available / Not Applicable
NIST	National Institute of Standards and Technology (formerly NBS)
NLP	Natural language processing
OCR	Optical Character Recognition
ODS	Open Data Services|Overhead Data Stream
OLAP	On Line Analytical Processing
PDF	Portable Document Format (for Adobe Acrobat Reader)
PILOTS	Published International Literature on Traumatic Stress
PLM	Product Lifecycle Management|Programming Language for Microcomputers
PNG	Portable Network Graphics
POC	Proof of Concept
POD	Pay On Delivery|Point Of Distribution
PPS	PostPostScriptum|Product Performance Surveys|Public Packet Switching (Network)
PSD2	Second Payment Service Directive
RA	Reference Architecture
RDBMS	Relational Database Mangement System
REST	Representational State Transfer
RWD	Rear Wheel Drive
SA	Supervisory Authority
SAP	Service Access Point
SAR	Synthetic Aperture Radar
SME	Smalland Medium-Sized Enterprises
SQL	Structured Query Language
STR	Store Task Register|Synchronous Transmitter Receiver
SW	SoftWare
TB	Terabyte (1,000 gigabytes)
TXT	Text|TeXT file
UI	User Interface
UML	Unified Modelling Language|Universal Markup Language
URL	Uniform Resource Locator

Abbreviation	Definition
USD	United States Dollar
VAT	Value Added Tax
VM	Virtual Machine
VPC	Virtual Path Connection
VaR	Value at risk
WMS	Warehouse Management System
WP2	Work Package 2 dealing with Requirements and Specifications
WP3	Work Package 3 dealing with BigData/IoT Data Management
WP5	Work Package 5 dealing with AI Algorithms (ML/DL) and Blockchain
WP7	Work Package 7 dealing with Pilots
XAI	EXplicable Artificial Intelligence
XML	Extensible Markup Language

1 Introduction

1.1 Objective of the Deliverable

This deliverable extends its previous version (D2.14 - "INFINITECH Reference Architecture - II") and reports the status of the INFINITECH-RA of the project. The RA is driving the technological developments of the project, as well as the design and initial integration of the use cases. As part of these development and integration processes feedback on the relevance and appropriateness of the RA were solicited. This feedback was taken into consideration in the development of this final version of the RA. This version includes previous parts for self-containment.

INFINITECH is developing and validating BigData, IoT and Artificial Intelligence (AI) technologies for the finance and insurance sectors. In this direction, the project is advancing the state of the art in several technological development areas such as infrastructures for integrated, incremental and real-time analytics, semantic interoperability, as well as decentralized information sharing between stakeholders of the involved sectors. Furthermore, the project designs and implements a range of pilots and use cases that aim at validating these technologies in real-life scenarios of the two sectors. One of the main objectives of the project is to define a Reference Architecture (INFINITECH-RA), which will serve as a blueprint for developing, deploying and operating BigData, AI and IoT in the finance and insurance sectors. This reference architecture should define the structuring principles of the solutions, along with their main building blocks. In this way, it will facilitate solution architects in developing and documenting specific solutions. Likewise, the INFINITECH-RA will facilitate stakeholders' communications regarding BigData, AI and IoT systems for the finance and insurance sector, through providing a uniform set of terms and definitions that will be unambiguously understandable by stakeholders. The purpose of this deliverable is to introduce and describe the INFINITECH-RA. It will emphasize on the presentation of the structuring principles of the RA, building on experience and recommendations from other general-purpose architecture for BigData applications such as the RA of the BigData Value Association (BDVA). Moreover, the deliverable will illustrate why and how the presented RA can support the main use cases of the sector, including the use cases that are designed and implemented in the scope of WP7 of the project. It will consider business requirements, application requirements and regulatory requirements documented in other deliverables of this workpackage.

1.2 Insights from other Tasks and Deliverables

The INFINITECH Reference Architecture signals an important milestone for the INFINITECH work on WP2. Specifically, it provides the structuring principles that will drive the design, development and integration of BigData, AI and IoT use cases in the finance sector, taking into account the INFINITECH use cases and technologies, as well as wider requirements of the sector. Toward this direction INFINITECH is:

- Driven by earlier WP2 deliverables on user stories and stakeholders' requirements, namely deliverables D2.1 and D2.2 - "User Stories and Stakeholders' Requirements" (versions I and II). The INFINITECH-RA shall be able to fulfil the project's BigData/AI user stories for the finance sector, along with key stakeholders' requirements. Specifically, the user stories of the project will be used to validate the functionality and the versatility of the Reference Architecture, in terms of its ability to support a broad range of applications for the sector.
- Considering the security and regulatory compliance specifications presented in deliverables D2.7 and D2.8 - "Security and Regulatory Compliance Specifications" (versions I and II). The INFINITECH-RA shall boost compliance to the main regulations of the sector such as the 2nd Payment Services Directive (PSD2) and the 4th Anti-Money Laundering (AML) directive, while considering the compliance of financial sector applications to general, broadly applicable regulations such as the General Data Protection Regulation (GDPR).

- Closely related to the deliverables that describe the INFINITECH components and technologies, namely deliverables D2.5 and D2.6 - "Specifications of INFINITECH Technologies" (versions I and II). In practice, there has been a two-way interaction between D2.6 (/D2.5) and D2.14 (/D2.13). The structure and functionalities of the INFINITECH components in D2.6 (/D2.5) have been considered in the development of the reference architecture that drives their integration in compete BigData / AI use cases and pipelines. On the other hand, the Reference Architecture is also unveiling the need for new functionalities in the INFINITECH components, as well as for entirely new software and middleware elements needed to support the integration of INFINITECH technologies in turn-key solutions.

The following figure illustrates the interaction of the present deliverable with other key deliverables of WP2. Note that this interaction will take place as part of the methodology for the development of the architecture, which is illustrated in the next section. For example, the adoption of the 4+1 views methodology will enable the validation of the INFINITECH-RA against user stories developed in D2.1 and D2.2 of the project.

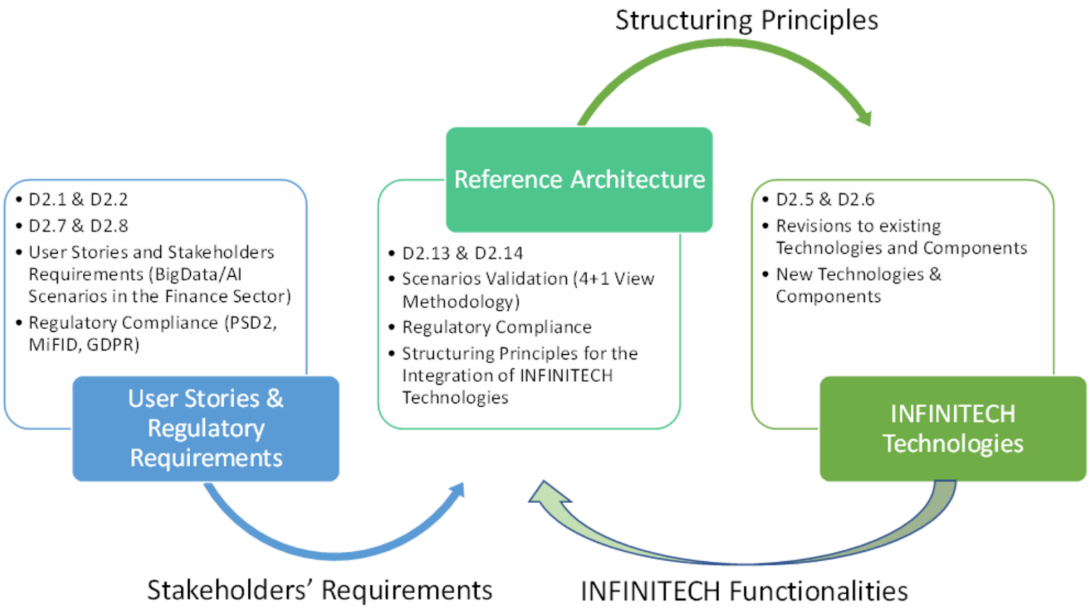


Figure 1 Interrelationship of INFINITECH-RA (D2.13) to other key deliverables of WP2

Beyond its cornerstone role in WP2, the present deliverable is closely linked to other technical developments of the project, namely developments of the technical workpackages of the project. Specifically, INFINITECH-RA will:

- Impact the design of sandboxes and testbeds in WP6 of the project, as these testbeds will be destined to host INFINITECH-RA deployments.
- Drive the integration of the project's pilots and use cases in WP7, as all pilots will align to the reference architecture of the project. The present deliverable provides already an early, INFINITECH-RA compliant design of the main BigData/AI pipelines that comprise the project's pilots.
- Provide insights for the detailed design of technical components in WP3, WP4 and WP5 of the project, including for example data management middleware, semantic interoperability elements, blockchain components for data sharing and decentralized application development, as well as Machine Learning (ML) workflows and algorithms. While the design and development of these components is in-principle independent of their use in a specific architecture, INFINITECH-RA will have an impact on the way these components interact between them. This will have a secondary influence on their design and development.

Overall, the present deliverable has an instrumental role in the project, as its drives many tasks and deliverables in the project. That's the reason why it is linked with one of the milestones of the project's workplan.

1.3 Methodology

The methodology for the development of the INFINITECH-RA is based on the following overlapping phases:

- Phase 1 – Review and Analysis (M2-M7): This phase was focused on the analysis of relevant technical and scientific information, ensuring that the INFINITECH-RA considers recent developments in BigData architectures in general and for the finance sector in particular. It analysed the BDVA RA, along with BigData architectures for digital finance applications that have been introduced and are widely used in the industry. In this context, BigData/AI environments and tools have been reviewed as well. Likewise, this phase considered the stakeholders requirements (from T2.1 of the project), as well as regulatory compliance requirements and the specifications of INFINITECH technologies reflected in D2.5 and D2.7 respectively. The overarching goal of this phase was to make sure that the design of the INFINITECH-RA aligns with key requirements of BigData applications in digital finance, as well as with the evolution of the state of the art in BigData in finance.
- Phase 2 – Architecture Design (M6-M12): This phase produced the initial design of the INFINITECH-RA, using the 4+1 view methodology [Kruchten95] for specifying software architectures. The selection of this proven and well-known methodology was motivated from the fact that INFINITECH pilot systems are essentially software systems. Hence, a methodology for specifying software architectures was applicable. In-line with the 4+1 architecture the INFINITECH-RA is described as a collection of complementary views (i.e. process, development, logical, physical) that represent dynamic behaviours and static behaviour of the systems, as well as relevant implementation aspects. Likewise, 4+1 signifies the need for confronting the architecture against a number of scenarios like the INFINITECH pilots. As part of the second phase of the INFINITECH development, a set of main views for the INFINITECH-RA were specified and described, as presented in subsequent paragraphs. Note that the phase extends beyond the delivery of the first version of the INFINITECH-RA (i.e. to M12 of the project), as several of the presented views will be revised in the coming months.
- Phase 3 – Fine Tuning and Updates (M12-M27): The third development phase of the INFINITECH-RA started after the submission of D2.13. Phase 3 aims at revising and fine-tuning the INFINITECH-RA based on its actual use in other work packages, including the sandboxes and pilot development work packages. It will receive and exploit stakeholders' feedback from the practical use of the INFINITECH-RA for developing BigData/AI systems for the finance sector. The feedback will be exploited towards improving aspects of the INFINITECH-RA, but also towards validating the architectural concepts introduced in practice. This Phase will lead to the final version of the INFINITECH-RA as part of deliverable D2.15 i.e. the third and last version/release of this deliverable.

The following figure illustrates the three development phases of the INFINITECH_RA and the main activities that they comprise.

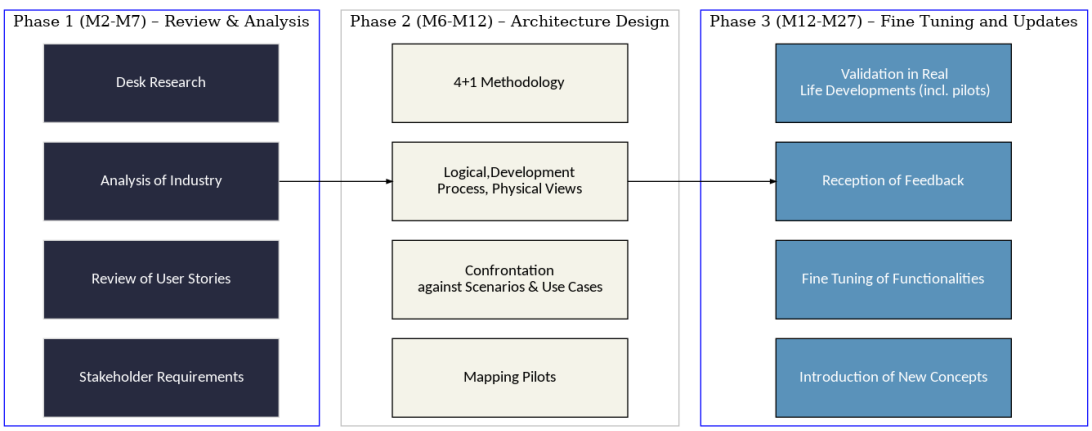


Figure 2 High-Level Overview of the Project's Phased Approach to INFINITECH-RA Development

1.4 Structure

The deliverable is structured as follows:

- Section 1 introduces the deliverable and includes the description of the objective, insights from other tasks and deliverables, the structure, and the updates from the previous version (D2.13).
- Sections 2 and 3 contain the description of the State of the Art. First, the earlier approaches for RA design of big data systems are reviewed. Then, utilisation of ML/AI techniques in SW engineering is discussed. Then, different architectural approaches are presented for realising ML utilisation in edge computing environments.
- Section 4 presents the RA design with different architectural views. Moreover, it shows how interoperability can be addressed within the RA.
- Section 5 analyses and discusses the RA.
- Section 6 provides detailed mapping of the Pilots workflows to the RA, which validates the RA conceptual designs and building blocks.
- Section 7 discusses unique points and propositions of the INFINITECH-RA.
- Section 8 presents the conclusions and future work.
- Appendices.

1.5 Input from Other Tasks

The specifications included in this deliverable take into account the requirements and specifications of all the previous tasks of WP2 [1-6] according to the diagram below.

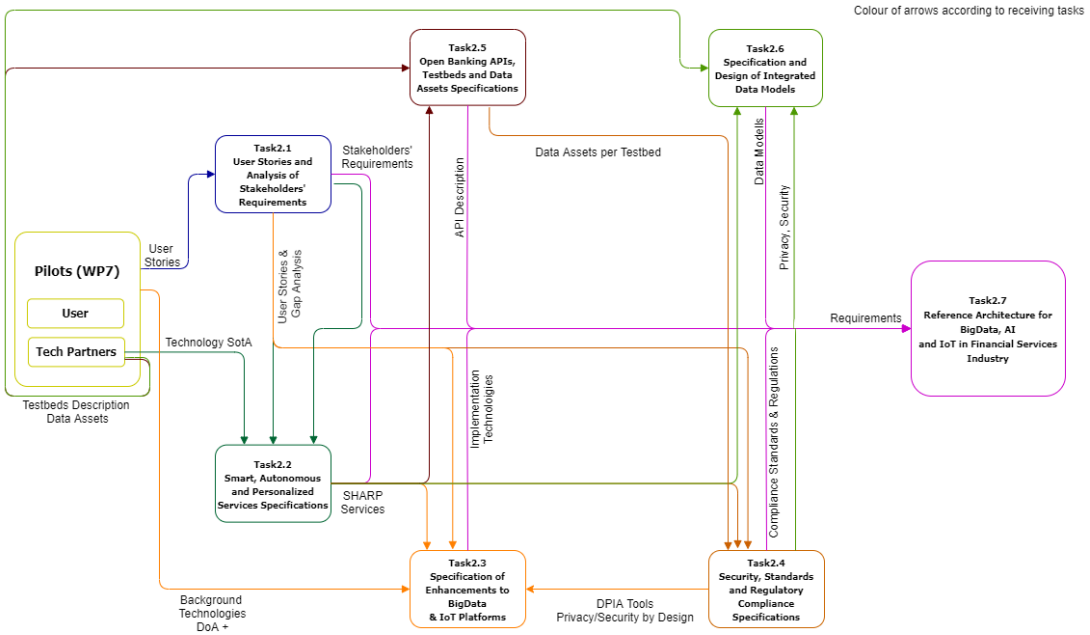


Figure 3 Relations between Tasks in WP2

1.6 Differences and updates from D2.14

D2.15 extends and enriches the content of its previous version (D2.14) with all the improvements achieved between month 19 and month 28. The main updates and changes with respect to D2.14 are summarized in the table below.

Table 1 Updates and additions with respect to D2.14

Section	Updated/New	Comments
3.3.4	New	Added StreamPipes in 3.3 Section on Pipelines Frameworks
4.9	New	Introduction to the Infinitech FinFlow concept
4.10	Update	Update on interoperability
6.x	Updated	Updates reporting the improvements in the allignments of pilots to the INFINITECH-RA.
6.1	Updated	Pilot1 set in frozen state
6.3	Updated	Updates pilot3
6.16	New	Added pilot16

2 Challenges in infrastructures for BigData, IoT and AI applications in the finance sector

2.1 General AI / BigData Challenges

2.1.1 Big Data

Over the past few years and in the context of big data, new challenges have emerged in data analytics, turning the attention to methodologies that raise the abstraction level and facilitate the convergence rate of algorithms. To this end, several big data frameworks have been developed by researchers and engineers and respective algorithms have been compiled, mainly being domain-specific. However, the development of scalable and distributed analytics solutions for extreme-scale analytics in the finance / insurance sector remains quite a complicated process given the diversity in datasets and data sources.

Data sources can be of different type (structural, semi-structural or completely unstructural), in various formats and data accessibility options. Due to the abovementioned factors, , modern enterprises rely on a variety of different and heterogeneous data management systems in order to handle with this data diversity: relational datastores that are used to store structural data compliant with an entity-relational model where the ensurance of transactional semantics and data consistency is of major concern, document-based data management systems to store semi-structural data, key-value stores that are considered efficient to store data coming from IoT sensors or logging information (i.e. when navigating among different web pages in the network, or simply logging the details of a finance transaction that took place) and even the use of HDFS data lakes is now considered prominent and facilitates the analysis of unstructural data that can be available. However, the analysis over a superset of the available data management system of an organisation is not a trivial task. Joining data across datasets is very cost demanding and difficult to be implemented efficiently in the application or data processing level. To overcome this barrier, various analytical frameworks have been proposed that provide polyglot capabilities and abstract this problem from data scientists, making the data management process transparent from their perspective.

Accessing heterogeneous data sources (a concept often addressed by data integration systems or multidatabases [Özsu11], [Tomasic98]) is a problem that has been widely studied in the literature and with the recent emerge of cloud database and big data processing, it has been evolved towards polystore systems. Early implementations [Minpeng11], [Ong14], [Simitsis12] relied on a single common model that the target datastores had to transform their schema to. A further improved presented by the polystore BigDAWG [Duggan15], [Gadepally16] which defines islands of information, which makes use of a single data model and language. Nowadays, Spark SQL [Armbrust15] exploits its advantage for massive parallel processing over a federation of different and heterogeneous stores. It defines the notion of dataframe and provides different connectors over a variety of supported datastores. By providing a single interface and a common query language, it pushes down the query execution on the target databases, when this is possible, and retrieves data into those dataframes that are being used for further processing. That requires however the retrieval of the data in the data analysis layer, which can be memory greedy. Similarly Presto allows for massive parallel processing where a coordinator orchestrates the query execution on several workers which make use of corresponding connectors, all of them implementing a common interface in order to hide the details of the query execution on the target database. Towards the same direction, Apache Drill and Impala maintain the notion of the data connectors that are able to retrieve data from external and heterogeneous datastores, and transform it to their own intermediate format and model that can be used for data processing in the upper layer of the framework. All those approaches provide polyglot capabilities, however, they are considered as additional frameworks that require the retrieval of the dataset in memory, and exploit their abilities for massive parallelism to be able to scale out adequately in order to deal with these requirements. The challenge lies on exploring the attributes

and statistics of each individual dataset that is stored in each store, optimize the query execution and retrieve only the minimum amount of data that is needed in each query operator, thus minimizing the need for memory consumption and data traffic in the network.

Besides the analytics level, running analytics over stale data is not ideal. For example, consider the case of a recommender system that aims to predict the behavioural patterns of a user, her preferences or dislikes, and provide personalised recommendations of relevant items. In a real-world scenario, user preferences change frequently, and new data continuously arrive in a real-time manner. A recommender system should, if possible, adapt to these changes as they happen, modifying its model to always reflect the current status, while requiring a single pass through the data. Furthermore, training new versions of the model in a continuously growing dataset is computationally inefficient, leading also to unnecessarily amplified infrastructure costs. Hence, data management approaches are required both to account for new datasets and to support these incremental analytics.

Apart from the scenarios where systems and analytical algorithms should continuously take into account the overgrowing datasets in order to update the results of their analysis, they will also need to analyse real-time data in order to respond to events and create alerts and notifications as they events occur. An example of such a scenario is fraud detection, where users must be notified when the fraud transaction occurs. However, dealing with data related with financial transaction raises a lot of concerns regarding data consistency and isolation. This is the reason why traditional relational database management systems are used, which ensure transactional (from a database theory perspective) semantics and ensure that a series of actions will be executed all or none (Atomicity), the data will be consistent (Consistency) and that they will not be stored breaking the rules that have been imposed by the database administrator (the balance of a bank account cannot be negative), parallel actions can be executed concurrently, but the result of their modifications will be likewise each action was executed in order (Isolation) and finally, that data will be preserved in terms of a system failure and all information can be recovered (Durability). Those four attributes consist of the ACID properties that traditional database management systems ensure and are of major importance in the finance sector. These databases are ideal for handling operational workloads, and they rely on the use of locks on data items that are being modified when a transaction access a data table. However, in order to perform a fraud detection, the system needs to read a variety of data by scanning the whole data table. In other words, to perform an analytical operation.

Operational and analytical operations are compatible and one blocks the other, as the existence of a lock will prevent the latter from being executed. Likewise, a long-running analytical operation will pose read locks in the whole data table, thus blocking all operational workloads from being executed: the database will not be able to serve customer needs as long as the long-running operation takes place. Due to this, modern enterprises take a snapshot of the operational database and send it to a data warehouse, thus separating those two workloads: The relational datastore will serve operational workloads by ensuring the ACID-ity, while the data warehouse will serve analytical query processing. The drawback of this approach is that detection of a fraud transaction will take place the following day, as moving data from one store to the other is cost-expensive operation and relies on heavy ETLs that are being performed during the night. Even exploiting modern approaches and apply micro-batching in the data movement process (i.e. the modifications of a transaction are being sent to a data queue, and a worker periodically gets data from the queue and sends them to the analytical warehouse) this allows for near real-time analytical processing, and cannot detect events that happen on real-time. Due to this, the analytics over operational data is still of great challenge.

2.1.2 Data Pipelines

An additional aspect and challenge refers to data management that needs to be enhanced in order to both facilitate the needs at extreme-scale (in terms of efficiently for data throughput and access) and to address the challenge of various federated and distributed systems that hold and corresponding datasets. What is more, one needs to consider the changes on the underlying systems (and datasets)

and thus the required level of dynamicity, raising the challenge for approaches that perform data pipelining from heterogeneous distributed systems towards analytics frameworks, while being adaptive to the aforementioned changes.

Tightly-coupled multistore systems trade autonomy for performance, typically in a shared-nothing cluster, to integrate structured (RDBMS) and HDFS data. Polybase [Minukhin18] is a feature of Microsoft SQL Server Parallel Data Warehouse to access HDFS data using SQL. It allows HDFS data to be referenced through external PDW tables and joined with native PDW tables using SQL queries. Hybrid systems are similar to tightly-coupled systems, e.g. integrating HDFS and RDBMS in a shared-nothing cluster, except that the HDFS data is accessed through a data processing framework like MapReduce. For example, QoX [Xu18] uses a dataflow approach for optimizing queries over relational (RDBMS and ETL) data and unstructured (HDFS) data, with a black box approach for cost modelling.

Moreover, when dealing with analytics performed on a variety of data sources, it is no less important to focus on the data aspects. Data-intensive distributed frameworks such as such as Apache Spark and Apache Drill can access multiple data stores using a unified API such as SQL. However, applications running on these frameworks have direct access to specific data stores and as a result to specific datasets, while frameworks such as Apache Ranger offer standardized access authorization to data stores, but only for a limited set of supported data stores and with limited policies.

What is required refers to an approach for data management that minimizes the data pipelining process and would enable a hybrid management of data, both for analytical and for transactional workloads. The latter would enable analytics to account for the different datasets made available as they are ingested in the data stores (mainly through the respective transactions).

2.2 Specific Challenges for the Finance Sector

2.2.1 Siloed Data and Business Operations

One of the most prominent challenges faced by banks and financial organizations is the fragmentation of data across different data sources such as databases, data lakes, transactional systems (e.g., e-banking) and OLAP (On Line Analytical Processing) systems (e.g., Customer Data Warehouses). This is the reason why financial organizations are creating BigData architectures that provide the means for consolidating diverse data sources. As a prominent example, the Bank of England has recently established a “One Bank Data Architecture” based on centralized data management platform . This platform facilitates the BigData Analytics tasks of the bank, as it permits analytics over significantly on larger datasets.

Note that the need for reducing data fragmentation has been also underlined by financial institutions following the financial crisis of 2008, where several financial organizations had no easy way to perform integrated risk assessments as different exposures (e.g., on subprime loans or ETFs (Exchange-Traded Fund)) were siloed across different systems. INFINITECH-RA must therefore provide the means for reducing data fragmentation and taking advantage of previous siloed data in integrated BigData Analytics and ML tasks.

2.2.2 Real Time Performance Requirements

Real Time Computing is when an IT system must respond to changes according to definite time constraints, usually on the order of milliseconds or seconds. In the realm of financial and insurance sectors, real time constraints apply where a response must be given to provide services to users or organizations and are in the order of seconds or more. Examples range from banking application to cybersecurity. Most of real-world financial applications are NOT real time as other industrial automation (plant control) and are usually solved putting more computing resources (cpu/gpu power, memory, ...) at the problem. However, in the case of ML/DL and BigData, algorithms can take significant amount of time and become useless in practical cases (responses arrive too late to be used). In those cases, a quantitative assessment of the computing time of algorithms is needed to configure resources to provide acceptable time.

2.2.3 Mobility

The digital transformation of financial institutions includes a transition to mobile-first banking [Bons12]. This refers to the interaction of customer and financial organizations through mobile channels as part of mobile banking. INIFINITECH must support mobile channels when visualizing BigData, AI and IoT applications for digital finance, but also when collecting and processing user's / customer's input.

2.2.4 Omni-Channel Banking - Multiple Channels Management

One of the main trends in banking and finance is the transition from conventional multi-channel banking to omni-channel banking [Komulainen18]. The latter refers to seamless and consistent interactions between customers and financial organizations across multiple channels. Hence, omni-channel banking/finance focuses on integrated customer interactions that comprise multiple transactions, rather than individual financial transactions. The INFINITECH-RA should provide the means for supporting omni-channel interactions through creating unified customer views and managing interactions across different channels based on integrated/consolidated information about the customer. Note that BigData analytics is the cornerstone of omni-channel banking as it enables the creation of unified views of the customers and the execution of analytical functions (including ML) that track, predict and anticipate customer behaviours.

2.2.5 Automation

Data intense applications, also in the banking and insurance sectors, are realized by specialized IT and RDBMS administrators. Recently, more and more data scientists and business analysts are involved in the development of such applications at a significant cost. The INFINITECH-RA should provide the means for orchestrating data intense application and management through easily creating workflows and data pipelines.

2.2.6 Transparency

During the last couple of years, financial organizations and customers of digital finance services raise the issue of transparency in the operation of BigData systems as a key prerequisite for the wider adoption and use of BigData analytics systems in financial, sector use cases. This is particularly important for use cases involving the deployment and use of AI/ML systems that operate as black-boxes and are hardly understandable by finance sector stakeholders. Hence, a key requirement for AI/ML financial sector applications is to be able to explain their outcomes. For example, a recent paper by bank of England [Bracke19] illustrates the importance of providing explainable and transparent credit risk decisions.

Overall, INFINITECH shall provide support for transparency in AI/ML workflows based on the use of Explainable Artificial Intelligence (XAI) techniques such as LIME (Local Interpretable Model-agnostic Explanations) [Ribeiro16], which develops/finds a local model around the prediction that is interpretable. XAI techniques should be supported as part of the ML techniques of the project, but also as an add-on to them as in several cases they will be used to interpret the outcomes of other ML/AI techniques (e.g., some classifiers).

3 Analysis and Alignment to State of the Art and Industrial Best Practices

3.1 Rationale for a Reference Architecture

RAs are designed for facilitating design and developments of concrete technological architectures, mostly in the IT domain, reducing risks with proven components, all while improveming overall communication within an organisation. Real drawbacks and benefits of RAs have been analysed with respect to the project's Pilots.

RA facilitated development of concrete IT architectures and reduced maintenance costs. In general, the value of RAs can be summarized in the following points:

- reduction of development and maintenance costs of systems
- facilitation of communication between important stakeholders
- reduction of risks

Typically, when a system is designed without a RA, an organisation may accumulate technical risks and end up with a complex and non-optimal implementation architecture.

In the industry, complex infrastructures for big data systems and high performance computing (HPC) have been developed and proved to sustain intensive data processing services (Netflix, Facebook, Twitter, LinkedIn etc.). The architectures and technologies of world class infrastructure have been published and RAs have been designed and proposed. However, very few solutions have been published for the Financial and Insurance sectors and this deliverable aims to partially cover the gap.

In the following sections, some relevant Reference Architectures and Models will be considered along with their relevance to the domain sector at which INFINITECH is aimed.

3.2 Overview of References Architectures for Data Driven Digital Finance Systems

3.2.1 Industry Vendors' Architectures for BigData in Digital Finance

3.2.1.1 IBM RA Architecture

The IBM Big Data RA capabilities view shown in Figure 4 is a vertically layered architecture. The left most layer are the data sources, both traditional ones and new ones, structured (such as relational DBMSs, flat files), semi-structured (such as XML) or unstructured (video, audio, digital etc). Traditional or back office applications execute and process the business transactions. These systems include the order processing, billing, marketing, product development and sales types of systems. New sources are systems that supply information to augment the information generated by the back-office applications. Those sources might include IoT data, video surveillance data, etc. The data sources provide data or raw materials to the analytics ecosystem that will be the foundation for analysis and knowledge.

The second left most layer is the data integration and analytics layer, both for data at rest and data at motion. Data Integration vertical is the data acquisition and transport pipeline between the Data Source Providers and the analytic ecosystem. The Real-Time Analytical Processing streaming layer supports data transfer at a steady high-speed rate to support many zero latency ("business real time") applications. The Data Warehousing provides raw and prepared data for analytics consumption. Shared Operational Data components own, rationalize, manage and share important operational data for the enterprise.

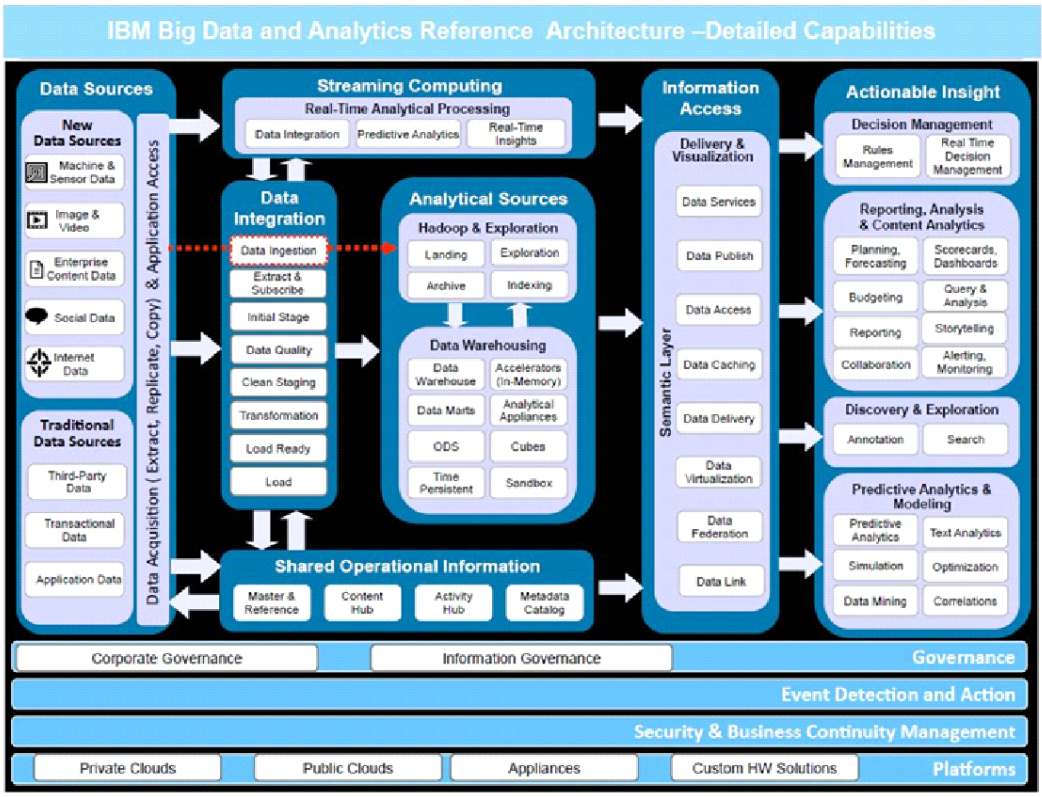


Figure 4 IBM Big Data and Analytics Reference Architecture

The third layer is the Information Access Layer, which is a semantic layer responsible for delivery and visualization of processed information to the next layer, the actionable insight layer. It enables the analytic applications to publish, locate and consume information without needing to understand exactly how and where it is physically stored and maintained. The goal is to provide a unified view of all information assets and can be divided into multiple components such as a database-resident Semantic Layer, data federation, data virtualization and data services. Finally, the Actionable Insight layer use data and models to provide insight to guide business decisions. It includes decision management, reporting, predictive analytics, planning and forecasting applications. On top of this layer enhanced applications can be provided, such as new business models, enhanced customer experience applications, risk-assessment applications, fraud and operational efficiency applications etc. Cross cutting across all those functional layers are the operational layers of information governance, security and business continuity and infrastructure platforms.

Every industry can leverage big data and analytics platform capabilities to its advantage, for example for banking industry it can be used to optimize offers and cross-selling, customer service and call service efficiency etc.

Relevance to INFINITECH RA

The IBM Big Data and Analytics RA suggests a collection of advanced data governance and security as well as analytics technologies. Data protection and management in a vertically layered approach accompanied with the respective infrastructure frameworks are considered absolutely necessary for business stability and development.

INFINITECH is intended to govern, secure and analyze advanced data in the financial industry through its custom developed enablers. Thus, IBM Big Data and Analytics RA is relevant to INFINITECH RA as it addresses advanced data governance and security together with analytics technologies. The following IBM Big Data and Analytics RA elements are acknowledged as applicable and relevant for the INFINITECH RA:

- Shared Operational Data: Process mining is destined to be of the fundamental capabilities in the overall INFINITECH. This IBM component should be taken under consideration when designing and developing governance, management and monitoring of system processes as well as operational data processing in INFINITECH.
- Information Access Layer: INFINITECH will implement data distribution and management when combining and maintaining dissimilar data from different sources, such as IoT and Blockchain. Information Access layer should be certainly considered in this manner.
- Actionable Insight layer: INFINITECH will envelope several types of data analysis, with predictive analytics being of highly important technology focus along the other types, such as incremental, declarative and parallel analytics. The Actionable Insight layer should be taken into account when organizing and planning the INFINITECH analytics.
- Data Integration, Real-Time Analytical Processing streaming layer and Data Warehousing: INFINITECH will be needed such tools when implementing algorithms for efficient data integration and preparation before delivery upon the data analysis infrastructure.

These IBM components and layer should be examined when determining continuous unification of INFINITECH modules for data transfer efficiency and arrangement for analytics applications.

3.2.1.2 Microsoft RA Architecture

The Microsoft Reference Architecture provides a Logical Banking Technology Architecture schema (Figure 5) (see also [MIRAB12]). It enables high value integration to other systems through a wide-array of industry-standard integration interfaces and techniques (such as ISO, BIAN, IFX) and reduces further the managing and business solution maintaining cost in the banking industry. Additionally, Microsoft RA offers an industry-leading set of robust functionalities defined and exploited in both the bank datacenters and cloud. This kind of functionalities extend across the overall IT stack from the crucial operations to the end-user and constitute a valuable framework for fraud detection in INFINITECH. Microsoft RA serves perfectly the Enterprise IT lifecycle of management and computing operations while simultaneously creates the next-generation customer experience through business productivity and cutting edge application and data services. Also, Microsoft RA's ability to manage and maintain huge volumes of critical transactions and processes as well as financial workloads only add value to the overall INFINITECH and how INFINITECH can exploit effectively those solutions. Diving deeper, the Microsoft RA provides Master Data Management (MDM), Data Quality Services (DQS) and pre-defined BI Sematic Metadata (BISM) which overlay BI capabilities delivered via pre-tuned datawarehouse configurations, near real-time analytics delivered through High-Performance technical Computing (HPC) and Complex Event Processing (CEP). Microsoft RA assets and commodities adopt jointly a high integration, which increases the overall architectural value and process efficiency especially when it comes to anti-money laundering. In this manner, components integration matrix costs and time costs of implementation and maintenance are efficiently reduced for any business through the highly-integrated Microsoft platform's optimal attributes exploitation. Overall, Microsoft RA is a leading provider of Data and Business Intelligence platforms providing a robust, end-to-end data, analytics and collaboration platform and constitutes a valuable part of the complete INFINITECH.

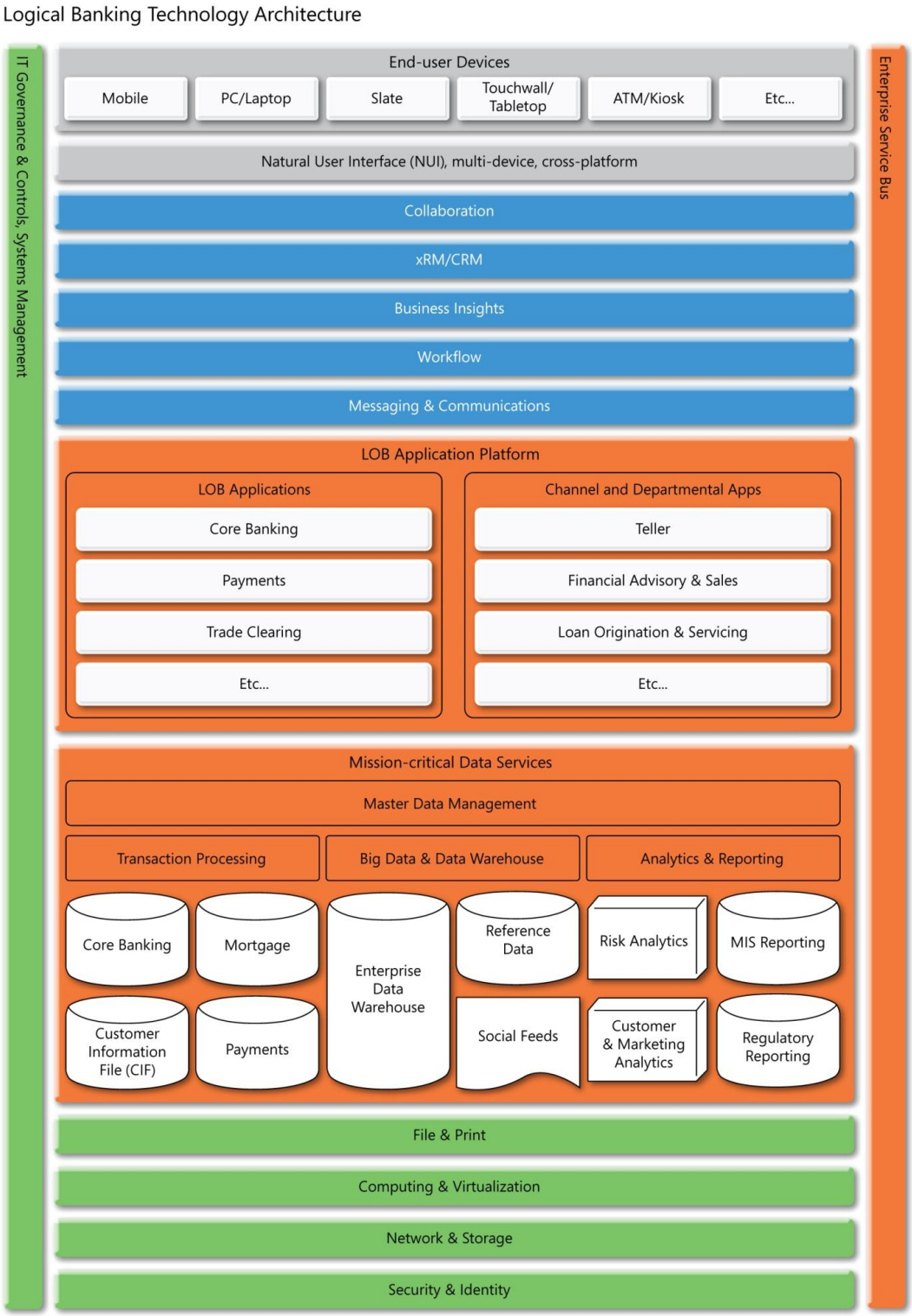


Figure 5 Microsoft RA Logical Banking Technology Architecture

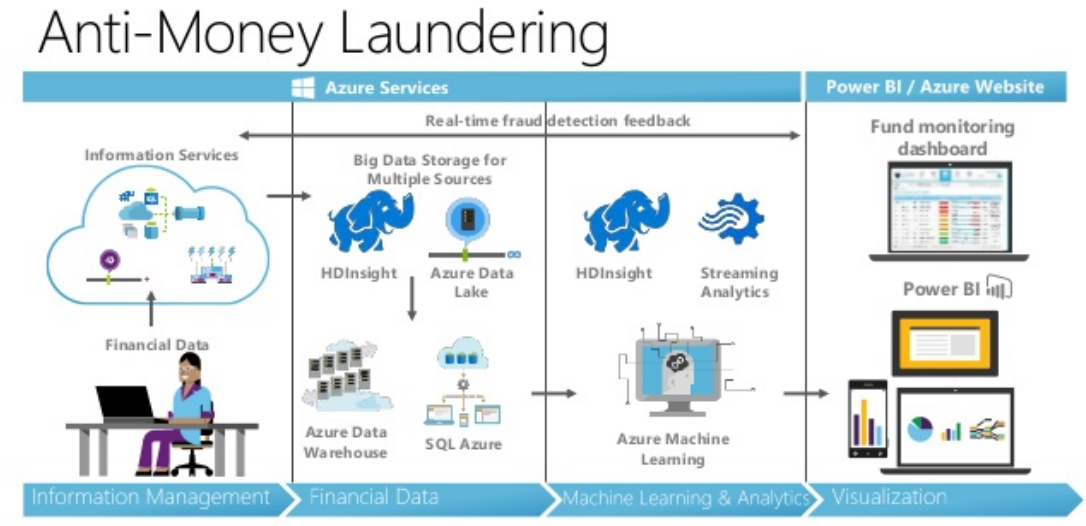


Figure 6 Microsoft RA Anti-Money Laundering Scheme

Relevance to INFINITECH RA

The Microsoft RA suggests a Logical Banking Technology Architecture and Anti-Money Laundering Scheme for the financial industry. Fraud detection as well as anti-money laundering are of utmost significance for the finance business.

INFINITECH is destined to address and eliminate fraud and money laundering in the financial sector through plenty of the specially designed pilots along with the overall solution of Testbeds and Sandboxes. Thus, Microsoft RA is relevant to INFINITECH RA as it focuses on organizing and sustaining massive volumes of transactions along with robust functionalities in bank datacenters. The following Microsoft RA elements are considered eventually related to the INFINITECH RA:

- Information Management: INFINITECH will implement information and data management of various financial corporations, such as banks, financial institutions, insurance-tech and fintech organizations. Process mining is destined to be of the fundamental capabilities in the overall INFINITECH. This Microsoft RA component (Anti-Money Laundering Scheme) should be taken into account in the context of designing and developing information services for the secure and scalable access of the type of data mentioned.
- Financial Data: In a similar manner, INFINITECH should consider thoroughly, for the same reasons, this Anti-Money Laundering layer in the context of storing this kind of financial data.
- ML & Analytics: INFINITECH will establish a variety of data analysis, including parallel, incremental, predictive and declarative real-time analytics. This layer of Microsoft RA should be definitely examined when preparing and constructing the analytics in Big Data, IoT and AI fields of INFINITECH.
- Visualization: INFINITECH will be using visualization tools and dedicated software components in order to extract the highly important information gained from the INFINITECH Analytics and conceive it in a human- and conclusion-friendly environment. This layer should be taken into account while assembling and formulating the visualization related INFINITECH modules.

3.2.1.3 WSO2 Architecture

WSO2 offers a modular platform that enables the implementation of connected applications for the financial service industry. The philosophy of the platform is to divide complex systems into simpler individual sub-systems that can be more flexibly managed, scaled and maintained. It emphasizes flexibility given the need to comply with a rapidly changing landscape of business and regulatory requirements. From an implementation perspective, it comprises various applications, data management systems and toolkits.

The platform architecture is illustrated in the following figure. It comprises a number of operational systems that feed a data warehouse to enable analytical processing and data mining. On tops of the data warehouse a number of enterprise applications are implemented including accounting applications and reporting applications.

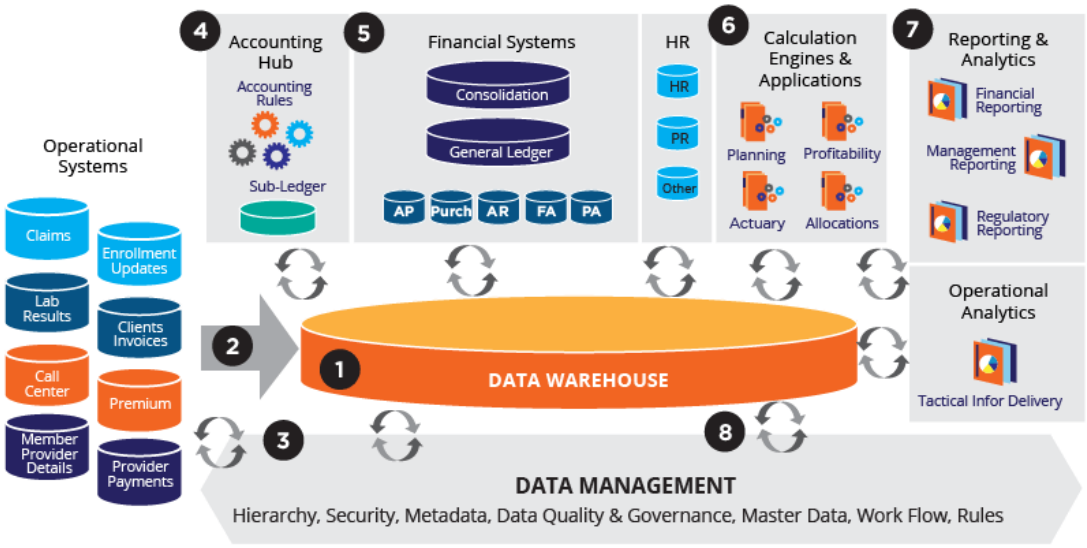


Figure 7 A next-generation financial infrastructure requirements

WSO2 sustains connected architectures in the financial domain through its determined professional and sophisticated product. Figure 7 illustrates how state-of-the-art business needs match into a connected financial reference architecture. In particular, WSO2 Private PaaS are eligible to be used in the data layer and the identity layers as well as in many financial clouds. WSO2 User Engagement Server qualifies as a builder of analytics needs, thus, it constitutes an essential factor on the different analytics automations that are needed in INFINITECH, while WSO2 Data Analytics Server aids in real-time processing of the stored statistics and data. Moreover, WSO2 Message Broker suits perfectly into managing communications under the business-specific financial protocols while WSO2 Business Process Server handles all the workflow processes which will be extensively required in INFINITECH since the big number of businesses participating while WSO2 DSS is ideal for data integration. WSO2 Applications Server suits for service development with defined standards, and WSO2 API Manager exposes the appropriately configured and developed financial APIs. Completely, WSO2 ESB constitutes the fundamental integration component rendering both latter components while being essential for the co-existence of standard, modular and configurable testbeds and sandboxes of INFINITECH. Lastly, the WSO2 middleware platform is designed in favour of specific standards related to electronic exchange of insurance data that regulate the (insurance) industry and is developed by the Association for Cooperative Operations Research and Development (ACORD), the global industry's nonprofit standards developer. In this manner, INFINITECH will benefit by the professional standards of the insurance industry. Overall, WSO2 is presenting a valuable solution and creates an abundance of opportunities for INFINITECH.

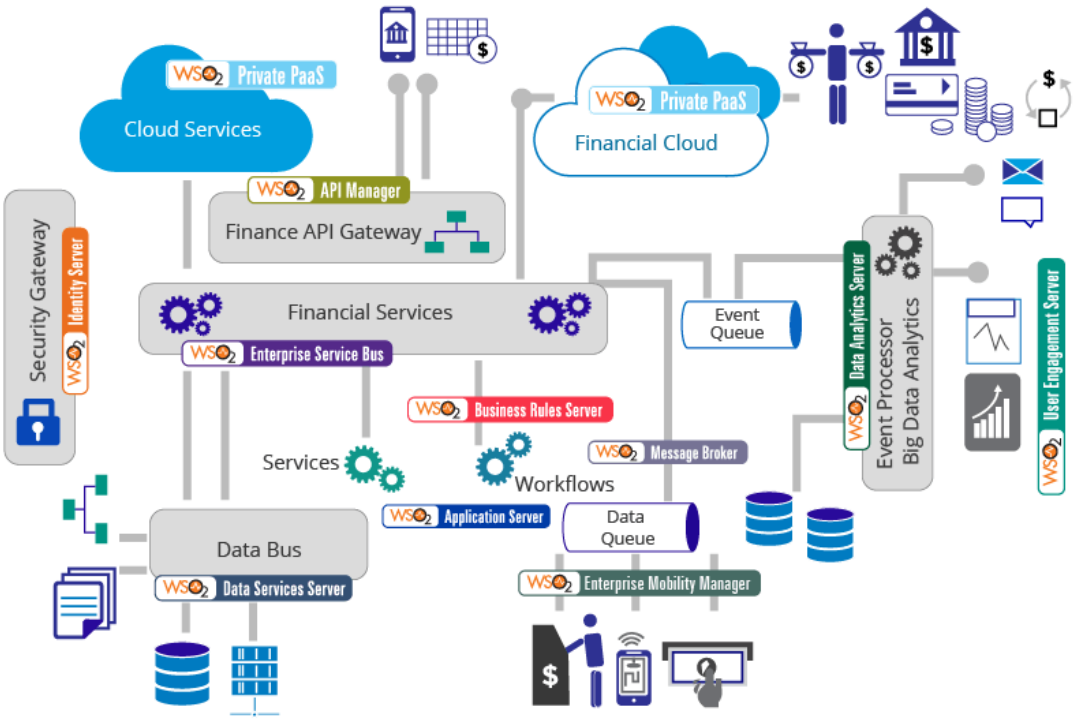


Figure 8 WSO2 Ecosystem Solution

Relevance to INFINITECH RA

The WSO2 Ecosystem accomplishes a sufficient solution for establishing an interconnected business followed by the necessary standards, components, and toolkits. The WSO2 RA upholds expertise when it comes to integration of internal components and it allows the performance of both in-house and foreign systems, while it manages the system components in an efficient and optimal way. INFINITECH promises to regulate, protect and evaluate advanced financial information and data in the finance industry over a variety of deliberately tailored services and components. WSO2 RA relates to INFINITECH RA as it focuses on integration of multiple and dissimilar modules, such as analytics, real-time data processing, financial clouds, and data exchange in the finance industry. The following WSO2 Ecosystem modules are determined highly relevant for the INFINITECH RA:

- WSO2 Private PaaS: INFINITECH is by definition focusing on the financial industry, thus, the WSO2 Private PaaS module should be evaluated as it contributes a good fit for many financial clouds.
- WSO2 User Engagement Server: INFINITECH will exploit a collection of analytics technologies. This WSO2 module should be considered for the INFINITECH analytics requirements types.
- WSO2 Business Process Server: INFINITECH will engage in a complex and intensive workflow due to the rich total of businesses participating. The WSO2 examined here, should be taken into account when organizing and cultivating INFINITECH workflow of processes and procedures.
- WSO2 ESB: INFINITECH will ultimately require essential module integration in the holistic approach of the project. WSO2 ESB could constitute a vital component when evaluating cooperation of the standard, modular and configurable testbeds and sandboxes of INFINITECH.
- WSO2 middleware platform: INFINITECH platform design would require specific standards related to electronic exchange of insurance data, thus, ACORD's design of the WSO2 middleware platform should be taken under consideration in this manner.

3.2.1.4 HortonWorks

Hortonworks Data Platform (HDP) provides the Hadoop Stack for Enterprise which is a centralized, enterprise-ready platform for storage and processing of any kind of data. When combined with a NoSQL database (e.g. Couchbase), HDP creates a huge volume of business value and intelligence. The currently demanded requirements are covered by this combination of the two. They include accuracy, in the context of bringing superior and precise analytic insights to Hadoop, and scalability, in the sense of providing comprehensive support from data-to-decision in order to boost the Hadoop's value across the business. Also, operational performance, in a manner of superior performance and high scalability data access across a range of business use cases, as well as economics, in the context of driving bottom-line benefits by boosting the value of operational and analytics infrastructure while reducing Total Cost of Ownership, are included. At a glance, state-of-the-art big data is constituted of the operational part (mostly NoSQL logic) and the analytical one (mostly Hadoop logic). Under the co-existence of both in finance, i.e. NoSQL together with HDP, several new scenarios are born. For instance, deep analytics when pulling data from Couchbase into Hadoop or training machine learning models and then cache them in Couchbase. Also, the integration with software like Sqoop, Kafka and Storm under the umbrella of Spark is now possible. Overall, the mentioned analytics have taken the form of Big Data Analytics as shown in Figure 9, and consist of three main phases: Data Pooling & Processing, Business Intelligence and Predictive Analytics.

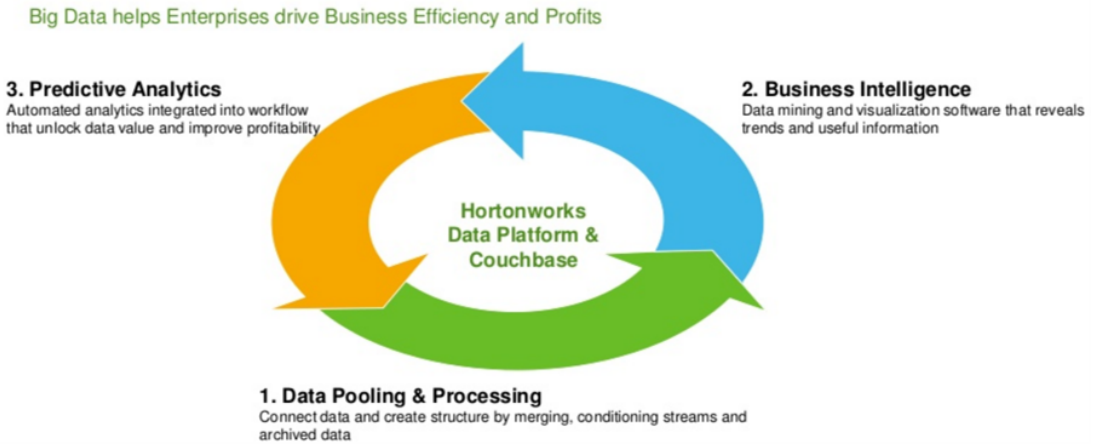


Figure 9 Big Data Adoption Lifecycle

In the finance sector, Big Data has transformed traditional banking. Internally, where the customer is unaware of, there are dissimilar pieces that form the whole Big Data picture. Risk management, security, compliance, digital banking, fraud detection and anti-money laundering constitute the essence of state-of-the-art banking which is perfectly built upon the HDP principles when combined with NoSQL (e.g. Couchbase). For instance, in a media and entertainment use case, Figure 10 shows how HDP and Couchbase can be used in order to integrate an Operational Data Store (ODS) and analytics capabilities in an enterprise environment.

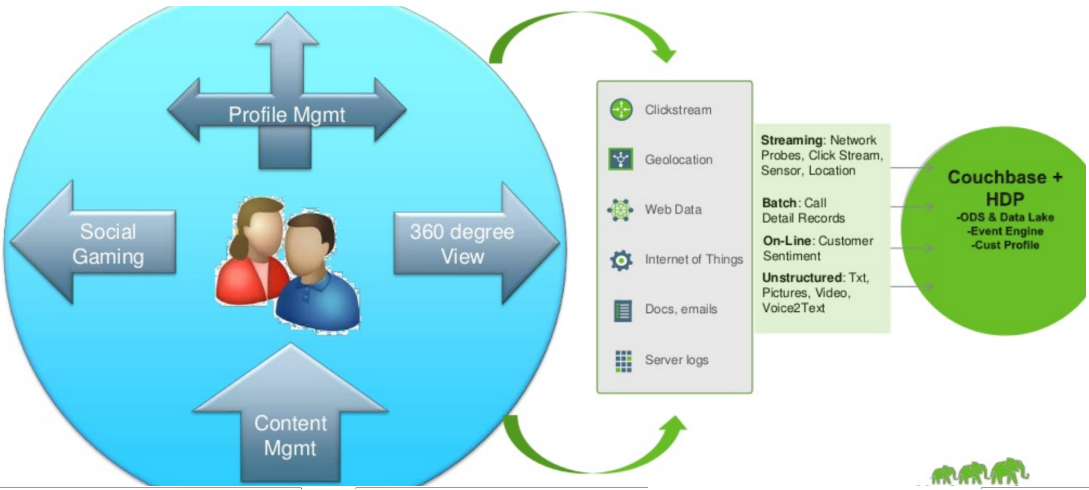


Figure 10 Media & Entertainment Use Case

In a similar manner, several concepts of the Hortonworks architecture could offer INFINITECH a plethora of benefits related to leading big data and artificial intelligence technologies. In particular, constant, and interoperable access to data lakes and NoSQL stored information, as well as integrated querying of data streams could be conveniently exploited in INFINITECH. Also, real-time analytics with open AI algorithms together with open finance APIs will be of immediate interest in INFINITECH as part of the regular standard mechanisms where the complete testbeds and sandboxes will be built on.

Relevance to INFINITECH RA

Hortonworks RA proposes a valuable Big Data Analytics. State-of-the-art big data analysis is crucial for any type of enterprise. Hortonworks is capable of satisfying big data needs and requirements of various use cases of dissimilar nature and prerequisites.

INFINITECH intends to make use of Big Data Analytics extensively and widely inside its core and surface structure. Hortonworks RA is absolutely relevant to INFINITECH's RA as it serves and reassures the needs and challenges which originate from big data analysis. The following Hortonworks elements are related to INFINITECH's RA:

- Data Pooling & Processing: INFINITECH will utilize big data for analysis, storage and distribution broadly, hence this first phase of Hortonworks Big Data Analytics should be taken into account when constructing, implementing and applying functions to INFINITECH data such as connecting, merging, conditionally streaming and archiving.
- Business Intelligence: In the context of big data analysis in INFINITECH, this second phase of Hortonworks Big Data Analytics should be considered when creating and establishing data operations such as result- and trend-driven mining and visualization.
- Predictive Analytics: INFINITECH will adopt different analytics processes among different scopes and pilot visions. Hortonworks third phase Big Data Analytics should be eligible when designing and composing automated result- and profit-driven integrated analysis into INFINITECH workflows.
- Operational Data Store (ODS) & Data Lake: INFINITECH will require storage capabilities and functionalities of big data scale. Hortonworks ODS and Data Lake should be taken under consideration when forming and building INFINITECH repositories and storehouses.

3.2.2 Reference Architectures from Standardization Bodies and Industry Associations

3.2.2.1 BDVA/DAIRO Reference Model

Big Data Value Association (BDVA/DAIRO) has presented a reference model for big data [BDVA/DAIRO]. The model has horizontal layers encompassing aspects of the data processing chain, and vertical layers addressing cross-cutting issues (e.g. cybersecurity and trust).

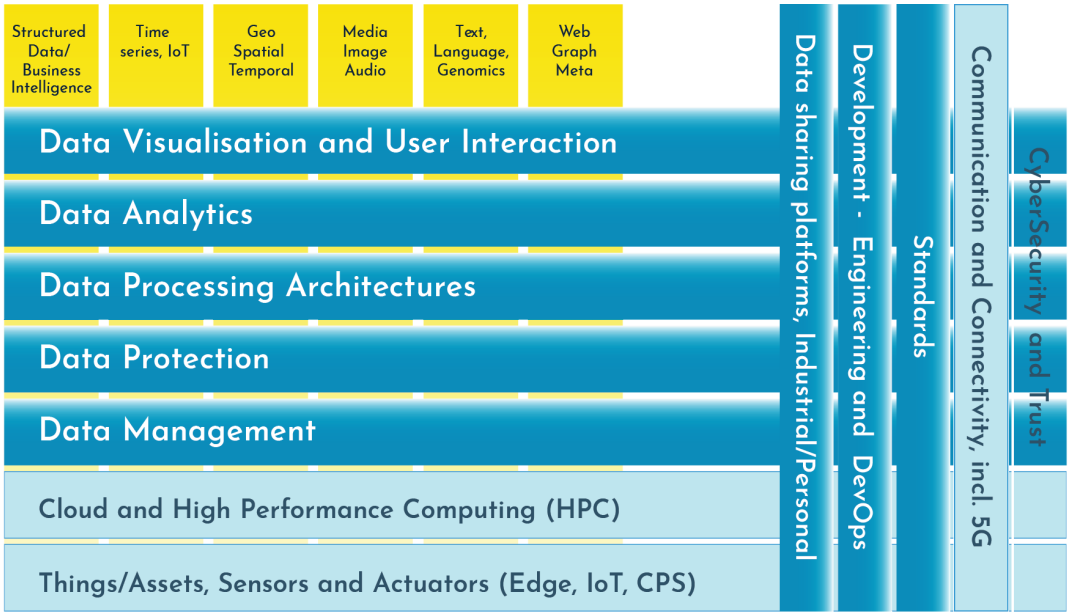


Figure 11 BDVA/DAIRO – Reference Model

The BDV Reference Model is structured into horizontal and vertical concerns.

- Horizontal concerns cover specific aspects along the data processing chain, starting with data collection and ingestion, and extending to data visualisation. It should be noted that the horizontal concerns do not imply a layered architecture. As an example, data visualisation may be applied directly to collected data (the data management aspect) without the need for data processing and analytics.
- Vertical concerns address cross-cutting issues, which may affect all the horizontal concerns. In addition, vertical concerns may also involve non-technical aspects.

Relevance to INFINITECH RA

BDV Reference Model (BDV RM) is not a Reference Architecture in the IT sense. However, the horizontal and vertical "concerns" of BDV RM are present in INFINITECH as layers, cross-cutting and data modelling sections. The BDV Reference Model serves as a common framework to locate Big Data technologies on the overall IT stack. It addresses the main concerns and aspects to be considered for Big Data Value systems. In the same way, the IRA consider a 1:1 correspondence with the horizontal "concerns" treated as layers in the INFINITECH RA conceptual schema.

In the same way that the BDV RM is not a layered architecture, as it does not impose hierarchy over sections or interdependencies in elements, IRA components can be freely pipelined (orchestrated) depending on the specific problem they are called to solve along other components. The most important consideration in IRA is that the all "concerns" are taken into account on any given solution. In the world of INFINITECH sandboxes, data components of each specific layers should be present as they add not only a level of specific processing but they assure that the specific problem is addressed (e.g. anonymization in data security, or user interface).

INFINITECH RA is largely influenced by the BDV Reference Model as it can be seen by this comparison figure. This will be made more evident in following Sections that introduce the INFINITECH-RA and its alignment to BDVA/DAIRO layers.

Figure 12 BDV – Reference Model vs INFINITECH Reference Architecture

3.2.2.2 NIST Big Data Reference Architecture (NBDRA)

National Institute of Standards and Technology (NIST) Big Data Program is developing a big data RA [Boid18]. The conceptual model is comprised of five functional components: data producer/ consumer, system orchestrator, and big data application/framework provider. Data flows, algorithm/ tool transfer, and service usage between the components can be described with different types of arrows. Activities and functional component views of the RA can be used for describing a big data system, where roles/subroles, activities, and functional components within the architecture are identified.

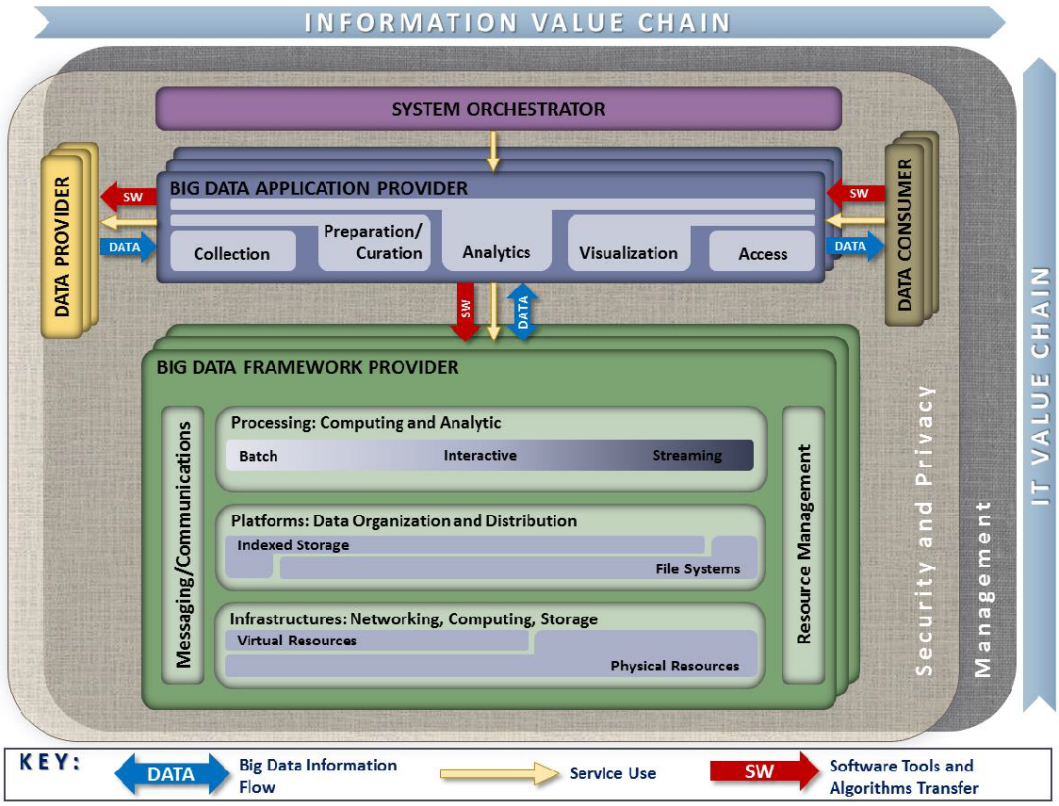


Figure 13 NIST Reference Architecture

Relevance to INFINITECH:

NIST approach is important for the INFINITECH RA as it identifies also roles and functional components. NIST is also considering the challenges in Big Data Architectures (as depicted in the following figure) which are very relevant to INFINITECH as well.

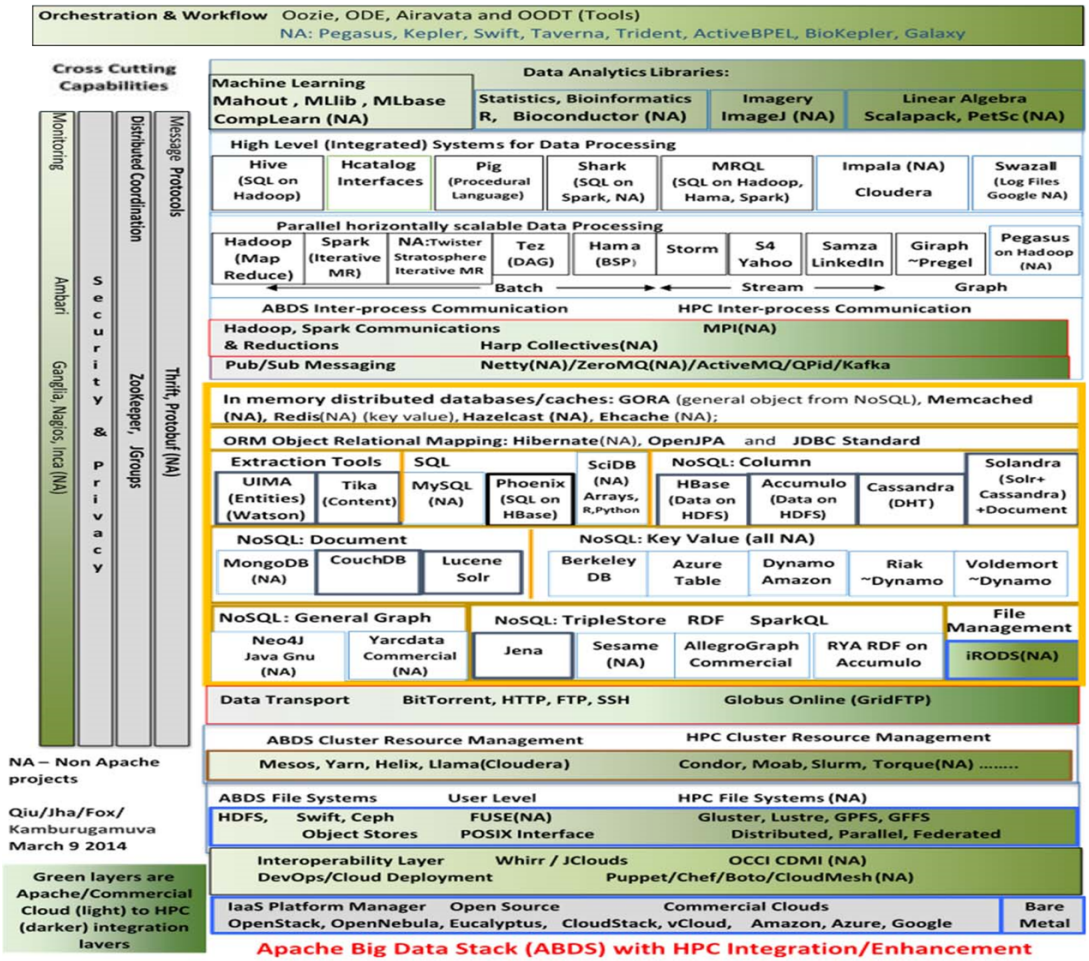


Figure 14 NIST Big Data Architecture and Infrastructure

3.2.2.3 Industrial Internet Reference Architecture (RA)

Internet of Things constitutes a fundamental territory of the INFINITECH Initiative since a vital part of the focus will be invested in it; thus, the Industrial Internet Reference Architecture (IIRA) will provide important value to INFINITECH as explained later in this section. The IIRA relates directly to Industrial Internet of Things (IIoT) systems as the latter establish the basic collections of IIRA interest. In particular, IIRA determines and describes the IIoT most critical architecture issues in the current industry while arranging them into the suitable viewpoints along with their respective stakeholders. Afterwards, the IIRA attempts to resolve these issues in the different viewpoints through efficient analysis by generating definite abstract architecture representations. With regard to INFINITECH, such an abstract structure will serve substantially in the arranging foundations of the project, as far as organization, alignment and management of the dissimilar existing formulations of IoT are concerned. In Figure 15, there is an illustration of the key ideas about the IIRA's architectural components and applications.

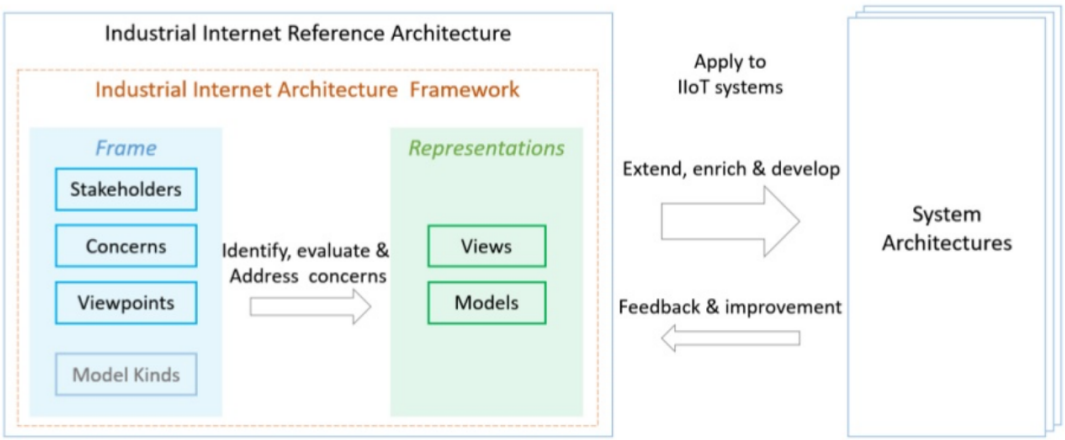


Figure 15 IIRA constructs and application

The IIRA's abstraction level omits architectural components who are assessing under concrete systems particularities. Thus, IIRA does not specify all the architectural components (constructs) of Figure 15, rather it adapts the ISO architecture specification (ISO/IEC/IEEE 42010) with minor rearrangements in two features as described in the following paragraphs.

The first feature is that IIRA does not particularly determine the model kind as a key component of the whole framework; rather it roughly uses this idea during the concern analysis in the abstract representation development.

The second feature is that IIRA does not particularly consider specific architectural components, such as Correspondence Rules, Correspondence and Architecture Rationale; it simply allows the concrete systems development to handle them. The models and their views' representations in the IIRA are carefully selected in this manner since they display the respective concerns at the corresponding level of abstraction and simultaneously determine the key ideas of this reference architecture. Nevertheless, they are not the only models and views for addressing concerns in the respective viewpoints or appropriate for a production system. These views can help initiate a concrete architecture, and then can be enhanced properly towards the specifications of the required IIoT system. Architecture developers that are eager about the ISO/IEC/IEEE 42010 Architecture Description for their concrete systems, can use the fundamental framework of IIRA in the beginning, then augment and upgrade the components offered that are applicable to the their individual needs and finally, develop those components that are outside the IIRA in the context of the architectural operation. The architectural description mentioned above will certainly increase the ability to regulate and supervise plentiful modules, as well as untap the complete potential of component development and rearrangement regarding IoT in INFINITECH. With an INFINITECH vision to a large but simple architecture of numerous requirements, IIRA will help INFINITECH to inspect its IoT and other needs from a broader and wider perspective, and simultaneously deliver and address productive, beneficial and profitable solutions with competence and dexterity on the field of architectural description and design approach.

Relevance to INFINITECH RA

The IIRA recommends a fundamental framework of abstract architecture design and deployment. IIRA uses a standard way to apply the Industrial Internet Architecture Framework (IIAF) capabilities to IoT systems and extend, enrich and develop system architectures.

INFINITECH is a humongous project that consists of various dissimilar and independent ideas while needs IoT technology integration and management. IIRA can be relevant to INFINITECH as it focuses on the conceptualization of different notions and components and is directly related to IIoT. The following IIRA aspects are suitable for acknowledged as applicable and relevant for the INFINITECH RA:

- IIoT systems: INFINITECH will require organization, adjustment and administration of disparate formulations of IoT. IIRA should be considered when defining and constructing the IoT fundamentals of INFINITECH.
- Abstract architecture representations and abstract structures: INFINITECH conceptualization should be performed with discretion, prudence and experienced organizing. IIRA abstract structure should be taken into account when conceptualizing INFINITECH's architecture.

3.2.3 Architectures of Relevant EU Projects and Research Initiatives

3.2.3.1 H2020 BigDataStack

The BigDataStack project focuses on the provision of an architecture that drives resource management decisions based on data aspects, such as the deployment and orchestration of services [Kyriazis18]. A relevant architecture is presented in the following figure, including the main information flows and interactions between the key components.

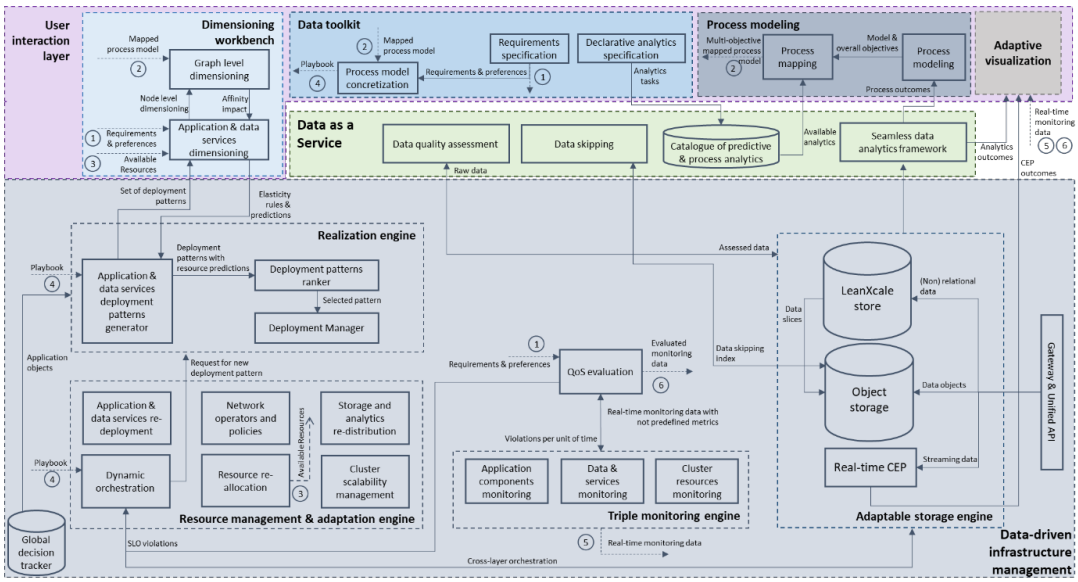


Figure 16 BigDataStack architecture model

As cited in the figure, raw data are ingested through the Gateway & Unified API component to the Storage engine of BigDataStack, which enables storage and data migration across different resources. The engine offers solutions both for relational and non-relational data, an Object Store to manage data as objects, and a CEP engine to deal with streaming data processing. The raw data are then processed by the Data Quality Assessment component, which enhances the data schema in terms of accuracy and veracity and provides an estimation for the corresponding datasets in terms of their quality. Data stored in Object Store are also enhanced with relevant metadata, to track information about objects and their dataset columns.

Those metadata can be used to show that an object is not relevant to a query, and therefore does not need to be accessed from storage or sent through the network. The defined metadata are also indexed, so that during query execution objects that are irrelevant to the query can be quickly filtered out from the list of objects to be retrieved for the query processing. This functionality is achieved through the Data skipping component of BigDataStack providing the relevant data skipping objects. Furthermore, the overall storage engine of BigDataStack has been enhanced to enable adaptations during runtime (i.e. self-scaling) based on the corresponding loads. The latter is of major importance

for INFINITECH given that the LeanXcale database will also be exploited in the project and the provided self-scaling capabilities will ensure the seamless provisioning of services (i.e. analytics) towards the financial / insurance data intensive applications.

Given the stored data, decision-makers can model their business workflows through the Process Modelling framework that incorporates two main components: the first component is Process modelling, which provides an interface for business process modelling and the specification of an end-to-end optimisation goals for the overall process (e.g. accuracy, overall completion time, etc). The second component refers to Process Mapping. Based on the analytics tasks available in the Catalogue of Predictive and Process Analytics and the specified overall goals, the mapping component identifies analytics algorithms that can realise the corresponding business processes. The outcome of the component is a model in a structural representation e.g. a JSON file that includes the overall workflow, and the mapped business processes to specific analytics tasks by considering several (potentially concurrent) overall objectives for the business workflow. Following, through the Data Toolkit, data scientists design, develop and ingest analytic processes/tasks to the Catalogue of Predictive and Process Analytics. This is achieved by combining a set of available or under development analytic functions into a high-level definition of the user's application. For instance, they define executables/scripts to run, as well as the execution endpoints per workflow step. Data scientists can also declare input/output data parameters, analysis configuration hyper-parameters (e.g. the k in a k-means algorithm), execution substrate requirements (e.g. CPU, memory limits etc.) as service level objectives (SLOs), as well as potential software packages / dependencies (e.g. Apache Spark, Flink etc.). The output of the Data Toolkit component enriches the output of the previous step (i.e. Process Modelling) and defines a BigDataStack Playbook.

The generated playbook (including operation sequence templates) is utilized by the Application and Data Services Deployment Patterns Generator. The component creates different arrangements (i.e. patterns / configurations) of deployment resources for each application and data service pod. To enable the latter, data from the Global Decision Tracker are obtained. These data refer to application objects that emerge following the registration of a new application via a playbook, resulting to the creation of an application object and a series of object templates representing the different parts of an application. Based on the above, a set of candidate deployment patterns are passed to the Application Dimensioning Workbench, along with an end-to-end optimization objective and the information on the available resources, in order to estimate resource usage and QoS performance prior to actual deployment. The primary output of the Application Dimensioning Workbench is an elasticity model, which defines the mapping of the input QoS parameters to the concrete resource needed (such as the number of VMs, bandwidth, latency etc.). These decisions are depended on data-defined models. Thus, based on the obtained dimensioning outcomes, deployment patterns are ranked by the Deployment Patterns Ranker and the optimum pattern is selected for deployment, making the concluding arrangement of services data-centric. The Deployment Manager administers the setup of the application and data services on the allocated resources. This is a key component that automated and optimizes the deployment process and could be exploited in INFINITECH for the different deployment configuration of the pilots of the project.

During runtime, the Triple Monitoring engine collects data regarding resources, application components (e.g. application metrics, data flows across application components, etc.) and data operations (e.g. analytics / query progress, storage distribution, etc.). An advancement comparing to the previous version of the architecture is that these metrics are not predefined and are identified during runtime so as to optimize the metrics to be collected and thereafter their evaluation. The collected data are evaluated through the QoS Evaluation component to identify events / facts that affect the overall quality of service (in comparison with the SLOs set in the toolkit). The evaluation outcomes are utilised by the Runtime adaptation engine, which includes a set of components (i.e. cluster resources re-allocation, storage and analytics re-distribution, network operators and policies enforcement, application and data services re-deployment, and dynamic orchestration patterns), to trigger the corresponding runtime adaptations needed for all infrastructure elements to maintain QoS. It should be noted that the dynamic orchestration employs a reinforcement-based logic that leads to cross-layer orchestration and optimization addressing both the resources and the data services as depicted in the figure.

Moreover, the architecture includes the Global decision tracker, which aims at storing all the decisions taken by the various components. In this context, the tracker holds additional information such as application logging data, Candidate Deployment Patterns, QoS failures, etc. Thus, as a global state tracker, provides the ground for cross-component optimisation, as well as tracking the state and history of BigDataStack applications.

Finally, the architecture includes the Adaptive Visualisation environment, which provides a complete view of all information, including raw monitoring data (for resource, application and data operations) and evaluated data (in terms of SLOs, thresholds and the evaluation of monitoring in relation to these thresholds). Moreover, the visualization environment acts as a unique point for BigDataStack for different stakeholders, actors, thus, incorporating the process modelling environment, the data toolkit and the dimensioning workbench. These accompany the views for infrastructure operators (e.g. regarding deployment patterns).

Relevance to INFINITECH RA

H2020 BigDataStack shares similar concepts and approaches on what concerns the data lifecycle and the automation of the deployment of the components. Regarding the data lifecycle, they both treat data at rest and streaming data. Data are stored in different silos on premise of the application owner and need to be ingested inside the platform. This process takes places in BigDataStack when the data arrives to the gateway component and a data ingestion process takes places to migrate the data inside the data repositories of the project. Data can live inside the operational database of the platform or inside a data warehouse. BigDataStack provides a seamless analytical framework that exposes a central endpoint and a common query language to access data that might be resigned in both stored, by hiding the complexity of joining the data from the two heterogeneous stores from the data user. Moreover, streaming data are also processing in the real-time using the Context Event Processing (CEP) component of the platform that also allows for querying the data at rest performing complex analytical query processing by ensuring data ensuring.

This is ensured by the INFINITECH RA by exploiting the Declarative Real-Time Analytics framework that is offered by the data management layer. What is more, H2020 BigDataStack provides a set of different analytical tools that access the central repository in a seamless way. This is similar to the libraries of ML/DL algorithms that the INFINITECH RA provides that access data coming from the data management layer in a seamless manner, by additionally ensuring data interoperability between different datasets via the corresponding semantic interoperability framework of INFINITECH. Finally, the use of open source technologies for the automation of the deployment and the software lifecycle management in terms of fault tolerance and elastic scalability that the H2020 BigDataStack is relying on, is towards the same direction with INFINITECH notion of sandboxes.

3.2.3.2 H2020 BOOST

BOOST 4.0 aims to be an open standardised and transformative shared data-driven Factory 4.0, demonstrating how European industry can build unique strategies and competitive advantages through big data across all phases of the product and process lifecycle.

The main objectives of BOOST 4.0 are to establish 10 big data lighthouse smart connected factories, provide the RAMI 4.0 and IDS based BOOST4.0 open EU framework and governance model, for both services and data assets and put together methodologies, assets, models and communities in order to maximise visibility, mobilization, replication potential and impact.

The BOOST 4.0 Reference Architecture consists of a number of layers at its core, alongside with a Factory dimension and a manufacturing entities dimension. More detailed information about these layers can be found in the D2.5 – BOOST 4.0 Reference Architecture Specification v1 [BOOST4.0-D2.5]].

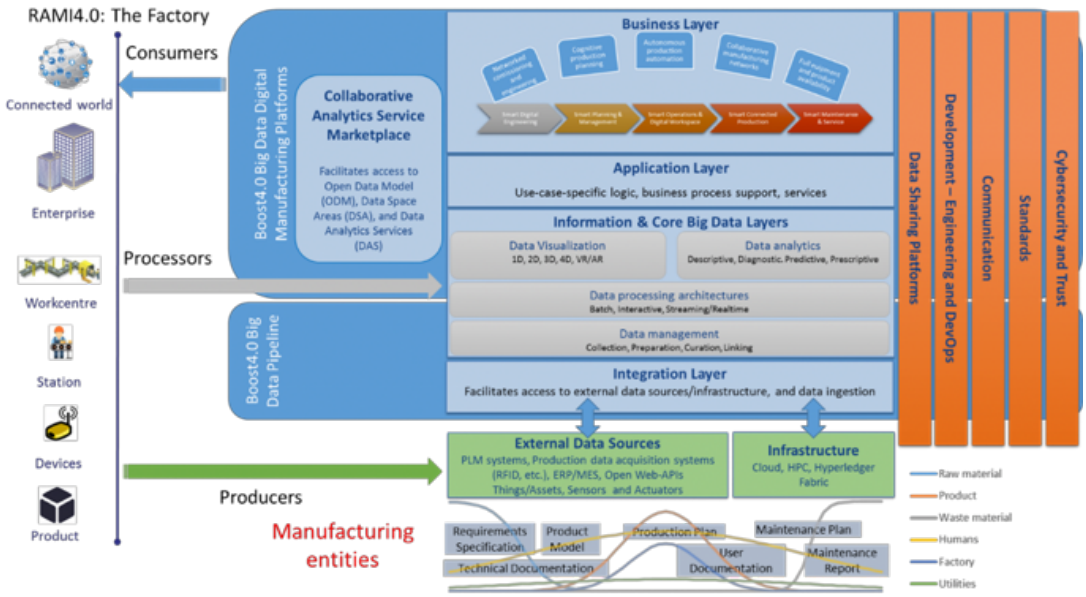


Figure 17 The Boost 4.0 Reference Architecture

The core layers represent a collection of functionalities/components performing a specific role in the data processing chain, and consist of Integration layer, Information and Core Big Data layers, Application layer and Business Layer.

The Integration layer facilitates the access and management of external data sources such as PLM systems, production data acquisition systems, Open web API's, and so on.

The Information and Core Big Data layer is composed of components belonging in four different sublayers:

- Data management groups together components facilitating data collection, preparation, curation and linking.
- Data processing groups together architectures that focus on data manipulation.
- Data analytics groups together components to support descriptive, diagnostic, predictive and prescriptive data analysis.
- Data visualization groups together algorithms components to support data visualization and user interaction.

The Application layer represents a group of components implementing application logic that supports specific business functionalities and exposes the functionality of lower layers through appropriate services.

The Business layer form the overall manufacturing business solution in the BOOST 4.0 five domains (networked commissioning & engineering, cognitive production planning, autonomous production automation, collaborative manufacturing networks, and full equipment & product availability) across five process life-cycle stages (Smart Digital Engineering, Smart Production Planning & Management, Smart Operations & Digital Workplace, Smart Connected Production, Smart Maintenance & Service) Alongside the core layers, there are a number of other cross-cutting aspects that affect all layers:

- Communication aims to provide mechanisms and technologies for reliable transmission and receipt of data between the layers
- Data sharing platforms allow data providers to share their data as a commodity, covering specific data, for a predefined space of time, and with a guarantee of reversibility at the end of the contract.
- Development-Engineering and DevOps covers tool chains and frameworks that significantly increase productivity in terms of developing and deploying big data solutions.
- Standards covers the different standard organisations and technologies used by BOOST 4.0
- Cybersecurity and trust cover topics such as device and application registration, identity and access management, data governance, data protection and so on.

Factory dimension (as defined in RAMI 4.0 (Reference Architecture Model Industry 4.0)) is used to identify the roles of various entities in the context of the Reference Architecture (producer, consumer and processor). Finally, the dimension of manufacturing entities allows the specification of such elements across the manufacturing process lifecycle stages and adds the relevant data types and ontologies for them. On the other hand, Manufacturing entities dimension provides a unified semantic model covering the multi-domain knowledge of the BOOST 4.0 pilots.

3.2.3.3 H2020 FINSEC

H2020 FINSEC project is targeted for cyber and physical security. The FINSEC platform RA aims to provide a general approach to the overall security of organizations, especially in the Financial Sector. The FINSEC RA platform is devised to be above the infrastructure (both IT and Physical) of a typical use case, e.g. of a company organization which already has its own infrastructure and applications and procedures to monitor the IT assets from a cyber-security standpoint. The following figure illustrates the actual design of the FINSEC RA

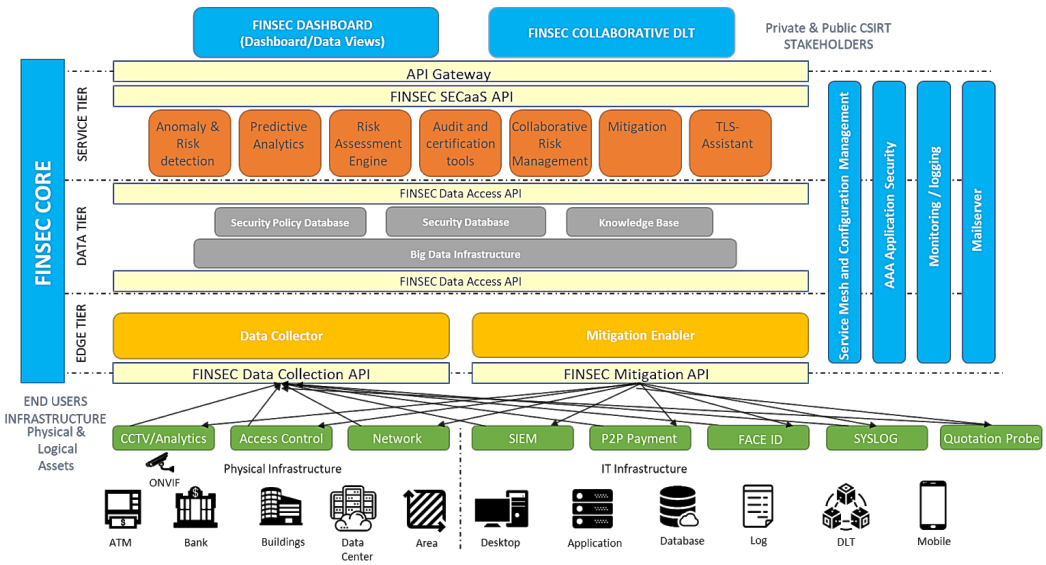


Figure 18 The FINSEC Reference Architecture version 3.0

The FINSEC RA can be viewed as an N-tier architecture composed by several layers:

- Field Tier - data from the physical and logical assets to be protected against threats. For example, CCTV analytics and SIEM are involved in this layer to give useful information about potential attacks to the upper tiers.

- Edge Tier - Data Collection module, which is necessary to filter the needed information during their flow towards the upper levels. The Actuation Enabler is responsible to allow some actions requested from the upper layers to be performed onto the probes, such as the shutdown of a server in case of threat or the closure of an automatic door of a protected room.
- The Data Tier is where information is stored and is organized into three different storage infrastructures, providing consisting data access API to all other modules.
- The Service Tier is where the kernel applications of the FINSEC and the security toolbox will be running, also exposing the functionality to external entities.
- The Presentation and Communication tier offer interfaces to the rest of the world. This tier is where the dashboard will be provided to monitor data and where assets and the FINSEC Collaborative Module will be available to share information with other FINSEC instances.

Relevance to INFINITECH RA

The FINSEC RA, is an example of layered bottom-to-top hierarchical IT architecture, suited for data flow from field to presentation and communication layers of applications. Data can flow also in other directions, especially from Analytics and Mitigation back to the fields with information that can transform also the data and the underlying infrastructure. Although the scenarios and the use cases of FINSEC are very different from INFINITECH, the two worlds share the same Financial and Insurance business sectors and therefore most of the constraints and regulatory frameworks apply in both architecture reference solutions.

Furthermore, the FINSEC -RA makes provisions for decentralized data exchange between stakeholders of the financial supply chain. It promotes a novel decentralized data sharing approach based on a blockchain infrastructure. As such it also provides a vision for exploiting blockchain technology in the finance sector beyond conventional blockbuster cryptocurrencies (e.g., BitCoin, Ethereum). INFINITECH could build on this approach towards supporting data sharing and data exchange in a broader range of use cases, beyond security data exchange.

3.3 Pipeline architectures

Managing the flow of information forms an integral part of every enterprise looking to generate value from their data. This process can be complicated due to the number of sources and data volume. In these situations, pipelines can be of help as they simplify the flow of data by eliminating the manual steps and automating the process. [N2K]

The pipeline architecture is a simple and powerful architecture inspired by the Unix technique of connecting the output of an application to the input of another via pipes on the shell [OR]. It is particularly suitable for applications that require a series of independent computations to be performed on data. Any pipeline consists of filters connected by pipes [SA].

A filter is a component that performs some operation on input data to transform them in output data. The latter is passed to other component(s) through a pipe. The pipe is a directional connector that passes stream of data from one filter to the next. A pump is a data source and is the first element of the pipeline.

A pump could be for example a static file, a data lake, a data warehouse, or any device continuously creating new data. Data can be managed in two ways: batch ingestion and streaming ingestion [DPA, N2K]. With batch ingestion, data are extracted following an external trigger and administered as a whole. Batch processing is mostly used for data transfer and is more suitable in cases where acquiring exhaustive insights is more important than getting faster analytics results. Instead, with the streaming ingestion, sources transfer unit data one by one. Stream processing is suitable in case of real-time data is required for applications or analytics.

Finally, the sink is the last element of the pipeline. It could be another file, a database, a data warehouse, a data lake or a screen. A pipeline is often a simple sequence of components. However, its structure can also be very complex: in fact, in principle, a filter can have any number of input and output pipes.

The pipeline architecture has different advantages [AS]:

1. It is easy to understand the overall behavior of the system, since it is a composition of the behaviors of the components;
2. It supports the reuse of the filters defined;
3. It is easy to add, replace, or remove filters from the pipeline. This makes the system easy to maintain and enhance;
4. It supports concurrent execution.

Data pipelines carry raw data from different data sources to data warehouses for data analysis and business intelligence. Developers can build data pipelines by writing code and interfacing with Saas platforms manually. However, nowadays, data analysts prefer using Data Pipeline as-a-service (DPaaS), which does not require coding. While using data pipelines, businesses can either build their own or use a DPaaS [DPA]. Developers write, test, and maintain the code required for a data pipeline using different frameworks and toolkits, for example management tools like Airflow and Luigi. Otherwise, solutions like KNIME enable to handle pipelines without the need for coding.

3.3.1 Apache Airflow

Apache Airflow [AA] is an open-source tool for authoring, scheduling and monitoring workflows. Airflow can be used to author workflows as Directed Acyclic Graphs (DAGs) of tasks. Apache Airflow has an airflow scheduler that executes your tasks on an array of workers while following the specified dependencies. Its rich command line utility enable to easily make performing complex surgeries on DAGs. Moreover, its rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed. When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative. Airflow provides a simple query interface to write SQL and get results quickly, as well as a charting application letting you visualize data. The Airflow scheduler monitors all tasks and all DAGs, and triggers the task instances whose dependencies have been met. Behind the scenes, it monitors and stays in sync with a folder for all DAG objects it may contain, and periodically (every minute or so) inspects active tasks to see whether they can be triggered. Airflow is based on the following principles:

- Dynamic: Airflow pipelines are configuration as code (Python), allowing for dynamic pipeline generation. This allows for writing code that instantiates pipelines dynamically.
- Extensible: Easily define your own operators, executors and extend the library so that it fits the level of abstraction that suits your environment.
- Elegant: Airflow pipelines are lean and explicit. Parameterizing your scripts is built into the core of Airflow using the powerful Jinja templating engine.
- Scalable: Airflow has a modular architecture and uses a message queue to orchestrate an arbitrary number of workers. Airflow is ready to scale to infinity.

3.3.2 Luigi

Luigi [SL] is a Python package useful for building complex pipelines of batch jobs. It was developed at Spotify, where it is used to run thousands of tasks every day, organized in complex dependency graphs. It enables to handle dependency resolution, workflow management, visualization, handling failures, command line integration, and much more.

The purpose of Luigi is to address all the plumbing typically associated with long-running batch processes. It is suitable to chain many tasks and automate them. The tasks can be anything, but are typically long running things like Hadoop jobs, dumping data to/from databases, running machine learning algorithms and more.

Luigi helps to stitch many tasks together, where each task can be a Hive query, a Hadoop job in Java, a Spark job in Scala or Python, a Python snippet, dumping a table from a database, or anything else. It makes it easy to build up long-running pipelines that comprise thousands of tasks and take days or weeks to complete. Since Luigi takes care of a lot of the workflow management, the user can focus on the tasks themselves and their dependencies.

Luigi also provides a toolbox of several common task templates. It includes support for running Python MapReduce jobs in Hadoop, Hive, and Pig, jobs. It also comes with file system abstractions for HDFS, and local files that ensures all file system operations are atomic. The visualiser provides an overview of the dependency graph of the workflow, where nodes are tasks which has to be run. Green nodes represent completed tasks, while yellow nodes represent tasks yet to be run.

The exact number of users is not known, because Luigi is open source and without any registration walls. However, based on the number of unique contributors, it is used by hundreds of enterprises.

3.3.3 KNIME

There are other solutions that enable to handle pipelines without the need for coding. One of them is the KNIME Analytics Platform [KAP]. KNIME is an open source software suitable for designing data science workflows and reusable components accessible to everyone. It is very intuitive, as it enables to create visual workflows with a drag and drop style graphical interface.

The KNIME Hub offers a library of components that enable to:

- Blend data from any source: simple text formats (CSV, PDF, XLS, JSON, XML, etc), unstructured data types (images, documents, networks, molecules, etc), or time series data. It also enables to connect to a host of databases and data warehouses to integrate data from Oracle, Microsoft SQL, Apache Hive, and more. Load Avro, Parquet, or ORC files from HDFS, S3, or Azure. Moreover, it is possible to access and retrieve data from sources such as Salesforce, SharePoint, SAP Reader (Theobald Software), Twitter, AWS S3, Google Sheets, Azure, and more.
- Shape data: derive statistics (mean, quantiles, and standard deviation), apply statistical tests to validate a hypothesis, or make correlation analysis, and more into workflows. Many components are available to clean, aggregate, sort, filter, and join data either on local machine, in-database, or in distributed big data environments. In addition, features can be extracted and selected to prepare datasets for machine learning with genetic algorithms, random search or backward and forward feature elimination.
- Leverage Machine Learning & AI. KNIME enables to:
    - Build machine learning models for classification, regression, dimension reduction, or clustering, using advanced algorithms including deep learning, tree-based methods, and logistic regression.
    - Optimize model performance with hyperparameter optimisation, boosting, bagging, stacking, or building complex ensembles.
    - Validate models by applying performance metrics including Accuracy, R2, AUC, and ROC. Perform cross validation to guarantee model stability.
    - Explain machine learning models with LIME, Shap/Shapley values. Understand model predictions with the interactive partial dependence/ICE plot.
    - Make predictions using validated models directly, or with industry leading PMML, including on Apache Spark.

- Discover and share insights: visualize data with classic and advanced charts (bar chart, scatter plot, parallel coordinates, sunburst, network graph, heat map), display summary statistics, export reports, store processed data or analytics results in files or databases.

Moreover, open source extensions (developed by KNIME, partners, and the community) are available to provide additional functionalities, such as access to and processing of complex data types, as well as the addition of advanced machine learning algorithms. In addition, open source integrations provide seamless access to open source projects such as Keras for deep learning, H2O for high performance machine learning, Apache Spark for big data processing, Python and R for scripting, etc.

Its modularity and drag and drop style makes KNIME a relevant solution for INFINITECH. These characteristics can be found in other tools, like the commercial product C3 AI Ex Machina [C3] and the free distributed LONI Pipeline [LP].

3.3.4 Apache StreamPipes

Apache StreamPipes [SP] enables flexible modeling of stream processing pipelines by providing a graphical modeling editor on top of existing stream processing frameworks.

It empowers non-technical users to quickly define and execute processing pipelines based on an easily extensible toolbox of data sources, data processors and data sinks. StreamPipes has an exchangeable runtime execution layer and executes pipelines using one of the provided wrappers, e.g., standalone or distributed in Apache Flink.

Pipeline elements in StreamPipes can be installed at runtime. Pipeline elements are standalone microservices that can run anywhere - centrally on servers, in a large-scale cluster or close at the edge.

StreamPipes is based on a few core concepts, illustrated in the following.

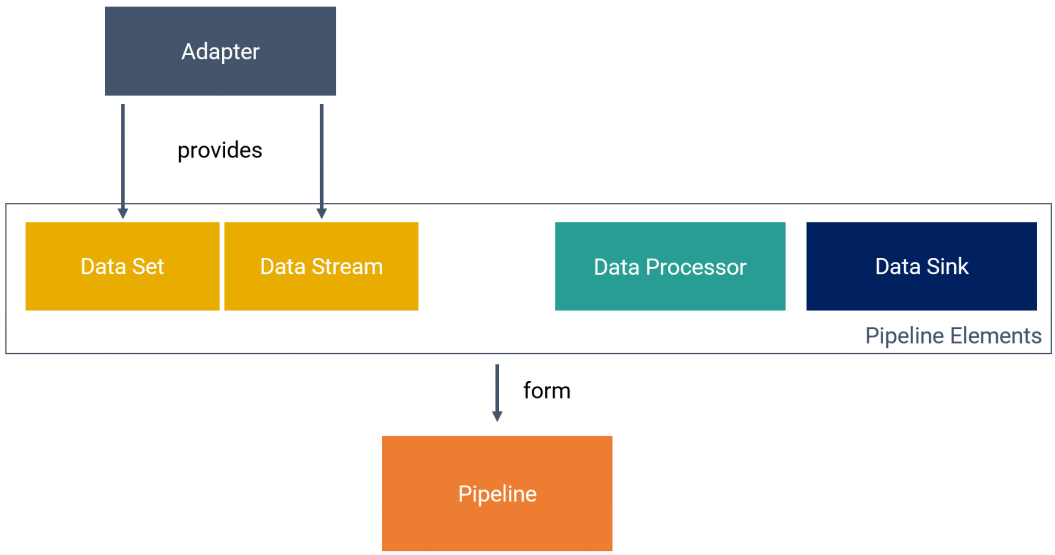


Figure 19 StreamPipes Overview of Concepts

- An **Adapter** connects to any external data source and forwards received events to the internal StreamPipes system. Within StreamPipes, the output of adapters are available in form of the two primary building blocks **Data Set** and **Data Stream**.
- **Data Streams** and **Data Sets** represent the primary source for working with events in StreamPipes. A stream is an ordered sequence of events, where an event typically consists of one or more observation values and additional metadata. The "structure" (or schema) of an event provided by a data stream or set is stored in the internal semantic schema registry of

StreamPipes. While data streams are typically unbounded, data sets have a fixed end and are internally "replayed" by the system from beginning to end once they are used as part of a pipeline. Most concepts also apply for data sets.

- A **Data Processor** in StreamPipes transforms one or more input data streams into an output data stream. Such transformations can be rather simple, e.g. filtering based on a predefined rule or more complex, e.g. applying rule-based or learning-based algorithms on the data. Data Processors can be applied on any data stream that matches the input requirements of a processor. In addition, most processors can be configured by providing user-defined parameters directly in the user interface. Processing elements define stream requirements that are a set of minimum properties an incoming event stream must provide. Data processors can keep state or perform stateless operations. At runtime, data streams are processed by using one of the underlying runtime wrappers.
- **A Data Sink** consume event streams similar to Data Processors, but do not provide an output data stream. As such, data sinks typically perform some action or trigger a visualization as a result of a stream transformation. Similar to data processors, sinks also require for the presence of specific input requirements of any bound data stream and can be customized. StreamPipes provides several internal data sinks, e.g., to create notifications, visualize live data or persist historical data of incoming streams. In addition, various data sinks are provided to forward data streams to external systems such as databases.
- A **Pipeline** describes the transformation process from a data stream to a data sink. Typically, a pipeline consists of at least one data stream (or data set), zero or more data processors and at least one data sink. Pipelines are built by users in a graphical way using the **Pipeline Editor** and can be started and stopped at any time.

3.4 Overall INFINITECH-RA Positioning

Overall, the INFINITECH-RA will be built on several concepts that have been introduced by other reference architecture models, including models developed by industrial organizations (e.g., large IT enterprises) and associations (e.g., BDVA), as well as models produced by other projects. Moreover, it considers the structuring principles of BigData and AI applications in digital finance, as these principles are reflected in industrial reference architectures. Nevertheless, the INFINITECH-RA aims at introducing a more flexible approach: Instead of providing a rigorous (but monolithic) structure of BigData/AI applications, it defines these applications as collections of data-driven pipelines. The latter are built based on INFINITECH components, notably the components specified in D2.5 of the project. Hence, the INFINITECH-RA provides a number of layered architectural concepts and a rich set of digital building blocks, which enable the development of virtually any BigData or AI application in Digital Finance. The project will offer increased flexibility in defining data-driven applications in the sector, in a ways that covers'/subsumes most of the rigorous architectures outlined in earlier paragraphs.

4 The INFINITECH Reference Architecture

The purpose of a Reference Architecture is to provide a conceptual and logical schema for solutions to a large class of problems. In the INFINITECH project the domain is as vast as the Financial and Insurance Sectors where most of the applications are data-driven. The class of problems of the project (pilots and use cases) and in general the service management of financial institutions and insurance companies are largely based on data that should be managed in the safest and protective way.

In these domains customers' enormous data sets must be processed to derive information with the purpose to provide better and more competitive services respecting the complex and sometimes conflicting regulatory frameworks such as privacy, security, interoperability, etc.

Therefore, a reference architecture should have explored in advance the specific domains in which the class of problems must find solutions providing a general model to which stakeholders (end-users, business owners, designers, data scientists, developers, maintainers etc.) can refer for best practices in the specific problem-solution space. In information technology, a RA can be used to check solutions to a particular problem in that class against the best practices and specific technologies. The INFINITECH RA is no exception and it is the result of the analysis of the significant number of use cases in the project's pilots, their requirements (users' stories) and constraints (regulatory, sector and technological) as well as the state-of-the art technologies and similar architectures.

It is important to state what is the RA in the INFINITECH project:

- A set of views for the Logical, Process, Development and Physical implementations
- A set of common scenarios referring to generic use cases
- A way to verify the use cases' scenarios and solutions
- A way to speak the same language among stakeholders
- A way to leverage solutions referring to best practices and building blocks
- A way to verify if constraints in requirements, regulatory, technical and logical have been addressed properly

It is also important to state what the INFINITECH RA is NOT:

- A ready-to-deploy technological IT framework
- A rigid and unmutable set of connecting building blocks
- A set of mandatory rules for development and integration
- A manual for implementation and rollouts
- A one-size-fits-all recipe for all business cases

The INFINITECH RA, is largely based on the concept of Data Pipelines used in Data Science and Data Manipulation. This model will allow to consider the blocks which make the solution by-design comply with rules and constraints. In that respect, ML and DL are emerging as key technologies to be developed, tested and evaluated in their specific domains. However, the utilization of machine learning (ML) as part of the computing infrastructure is still an area for further research [8]. Therefore, at the heart of the project as well as a pillar concept of the RA is to leave great flexibility in the data management workflow to experiment with different solutions.

Particularly, the RA and the workflows based on it should be used to understand how data can be collected, and how models and technologies should be developed, distributed, and deployed for decision making purposes in the specific use cases.

Therefore, the goal of this deliverable is to provide a RA for facilitating the design of the INFINITECH Pilots' workflows which encourages the utilization of standard data-processing components and ML techniques as well as big data architectures and infrastructure.

Another important fact, that has been addressed is how development and deployment of ML models may be related to actors (e.g. Data scientist, ML engineer) at the business layer.

In general, the value of RAs can be summarized in the following points:

- reduction of development and maintenance costs of systems
- facilitation of communication between important stakeholders
- reduction of risks

Also, when a system is designed without a RA, an organization may accumulate technical risks [11] and end up with a complex and non-optimal implementation architecture.

The presented reference architecture (RA) has been created inductively based on the Pilots logical workflows. The RA has been extended based on an earlier published RA for big data systems [4]. Particularly, development and deployment views have been focused on big data computing environments.

High level and deployment environment views of the RA may facilitate architecture design of workflows in the future.

The following sections contain the different views of the INFINITECH Reference Architecture. It is inspired by BDVA Reference Model and other logical architectures for big data presented in the section related to the State of the Art.

The INFINITECH Reference Architecture or IRA for short, is also derived from the use cases defined in the project and provides a general framework for the solutions to be developed and deployed in the pilots.

Most of the logical views of IT architectures, and not only in H2020 projects, present schemes with links and relations between the different building blocks from which it should be evident to understand the logical flow of data and the interdependencies between blocks.
However, most of these schemes impose rigid mechanisms to how data can be managed which in real world scenarios complicates the solution instead of simplifying it. What data scientists want is to experiment with data (especially big data) in a flexible way to play with algorithms and technologies to select the best combination to solve real use cases, like those proposed in INFINITECH.

High level and deployment environment views of the RA may facilitate architecture design of workflows in the future.

4.1 Methodology: Architectural View Model

The INFINITECH Reference Architecture uses the “4+1” architectural view model as the methodology to cover all the aspects of the different problems the Project and the Consortium want to address.

The “4+1” architectural view model is a methodology to design an architecture for a software platform, having the main capacity of describing it from 5 concurrent “views”.

These views represent different stakeholders who could deal with the platform and the architecture, from the management, development and user perspectives. The four main views which facilitate the definition and design of the architecture are the logical, process, development and physical ones, while the “+1” view is represented by the use cases or scenarios, thus making this model an abstraction of the developed solution/platform and the basis for the development. In the following, the meaning of the different views is explained.

1. Logical view: it represents the range of functionalities or services that the system provides to the end users, and can be shown as block diagrams
2. Process view: it represents the system processes and data flows and how the different processes and building blocks communicate between each other, with details on the runtime behaviour of the system.
3. Development view (or implementation view): it illustrates the software management aspects of the system, from the programmer’s point of view. It is represented by UML components and package diagrams.

4. Physical view (or deployment view): it describes the lower levels of the architecture, dealing with the physical infrastructures where the software components run and the physical connections between them. It can be represented by deployment diagrams.
5. Scenarios: the architecture of the system can be explained from the end user's point of view too, with the description of use cases. The use cases or "scenarios" aim at describing some possible functioning situations of the system and interactions between components. The validation and assessment of functionalities are usually performed by this view.

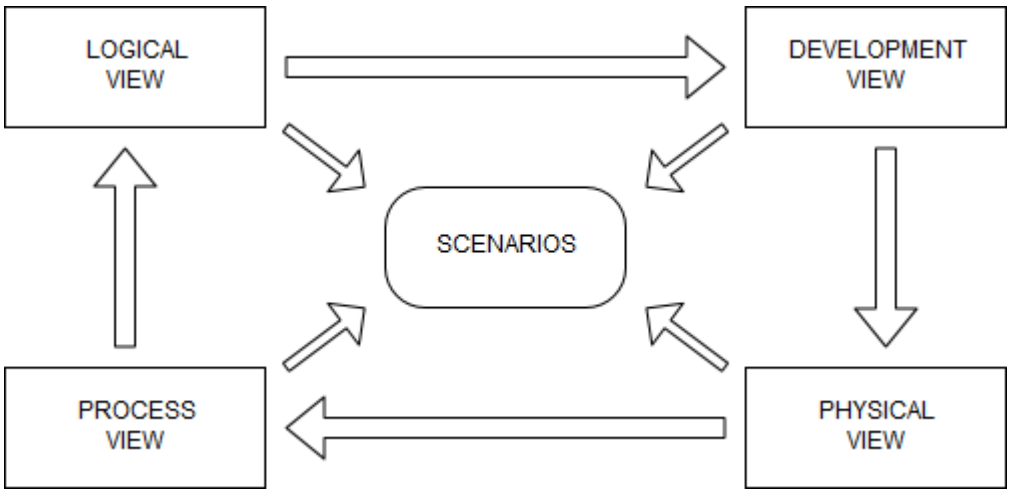


Figure 20 "4+1" views

The present document describes in particular how the Logical view and the Process view were designed by the Consortium, and analyses afterwards the Development view and the physical view.

4.2 Key Assumptions

The INFINITECH RA is inspired by state-of-the-art solutions and technologies. The following assumptions are valid as general requirements for the IRA:

- The BDVA Reference Model is a "reference" also for the INFINITECH project. More information are presented in the section 3.2.2.1 and can be found at SRIA BDVA [BDVA-RM]
- The INFINITECH Reference Architecture (RA) should provide an abstraction to solve a general class of use cases and in particular those arising from the pilots.
- INFINITECH RA should provide a mapping of SotA technologies, development methodologies, deployment and testing environments, production.
- INFINITECH RA will exploit Microservices technologies and DevOps methodologies
- Data Components should be mapped/encapsulated in Microservices
- Loosely coupled microservices are preferred over tightly coupled intertwined applications
- Data can be pipelined into different microservices to perform different type of processing
- Data can be streamed and/or stored in data storages
- Data Movements should be limited as much as possible (streaming in favour of storing)
- Cross cutting services will provide added functionalities across the different layers

4.2.1 Definitions

- In the INFINITECH RA a node is a unit of data processing.
- A node exhibits interfaces (APIs) for data management in particular for consuming and producing data (IN and OUT)
- In a microservice implementation of IRA, nodes are microservices by preferable exposing a REST API

- IRA has layers that can be referred to the BDVA Reference Model
- Nodes belong to layers in the RA
- Nodes with IN OUT interfaces can be stacked up to form data pipelines (LEGO concept)
- Nodes are loosely coupled in the RA: i.e. they are not connected until a pipelined is created
- Data can flow in all directions (e.g. a ML can also push back data into a Data Layer)
- Node stacks with other compatible nodes (stackability depends on the IN-OUT interfaces)
- As a consequence of the previous point, RA can be segmented into vertical layers (called bars) to group compatible nodes.
- A node can belong to one or more vertical bar
- Node can be orchestrated to form solutions (sandboxes)

Therefore, the IRA can be considered:

- Layered: RA identifies Layers as way of grouping nodes in the same way as DBVA Reference Model has “concerns”
- Loosely Coupled: no rules to connect nodes in a predefined way or in a rigid stack
- Distributed: computing and nodes can be distributed anywhere (on-prem, on cloud, on different clouds)
- Scalable: nodes can distribute computing at edges, at HPC nodes (GPUs) or centrally
- Multi workflow: RA allows for simple pipelines and/or complex dataflows
- Interoperable: nodes can be connected to other nodes with compatible interfaces
- Orchestrable: nodes can be composed in different ways allowing creation of virtually infinite combinations (the sandbox concept)

4.3 Logical View

The Logical Views of the IRA are presented in the schemes below. Rather than identifying and specifying functional blocks and how they are interconnected, the INFINITECH Reference Architecture is defined by a set of “principles” to build pipelines or workflows. These principles constitute the guidelines that will drive any specific implementation of INFINITECH solutions (Pilots implementation, Minimum Viable Platform, Sandboxes, etc). At a first level the INFINITECH RA can be seen as a pipelined mapping of nodes referring to the BDV Reference Model and cross cutting layers as in the schema below:

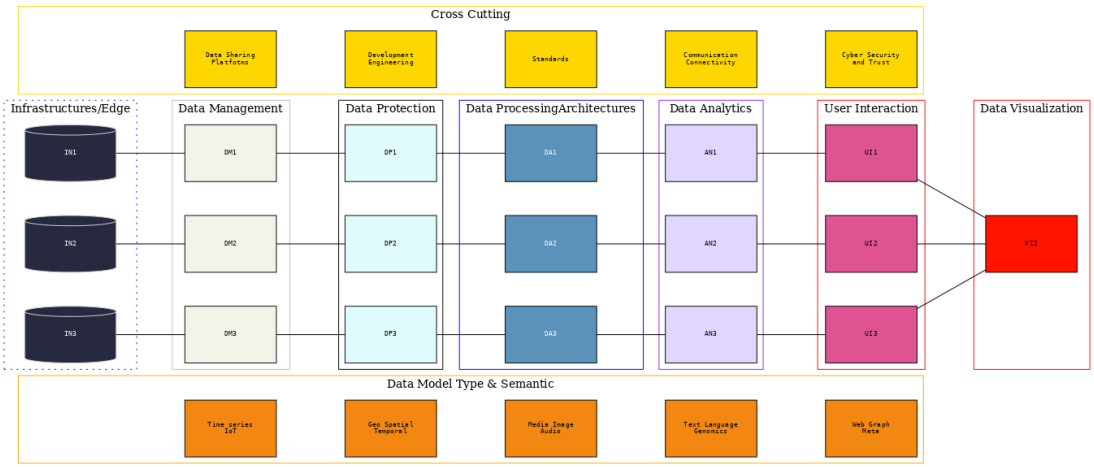


Figure 21 INFINITECH Reference Architecture Mapping with BDVA Reference Model

The following schema is a High Level Logical View that depicts the INFINITECH RA. The generic nodes are only examples and do NOT refer to any specific tool or applications but to generic services that belong to a general class of application performing the functionality of the corresponding layer in the BDVA Reference Model.

Specific Nodes will be defined and mapped in specific solutions for all the Use Cases of the project. However, RA is generic and will leave nodes a great level of flexibility depending on the implementation.

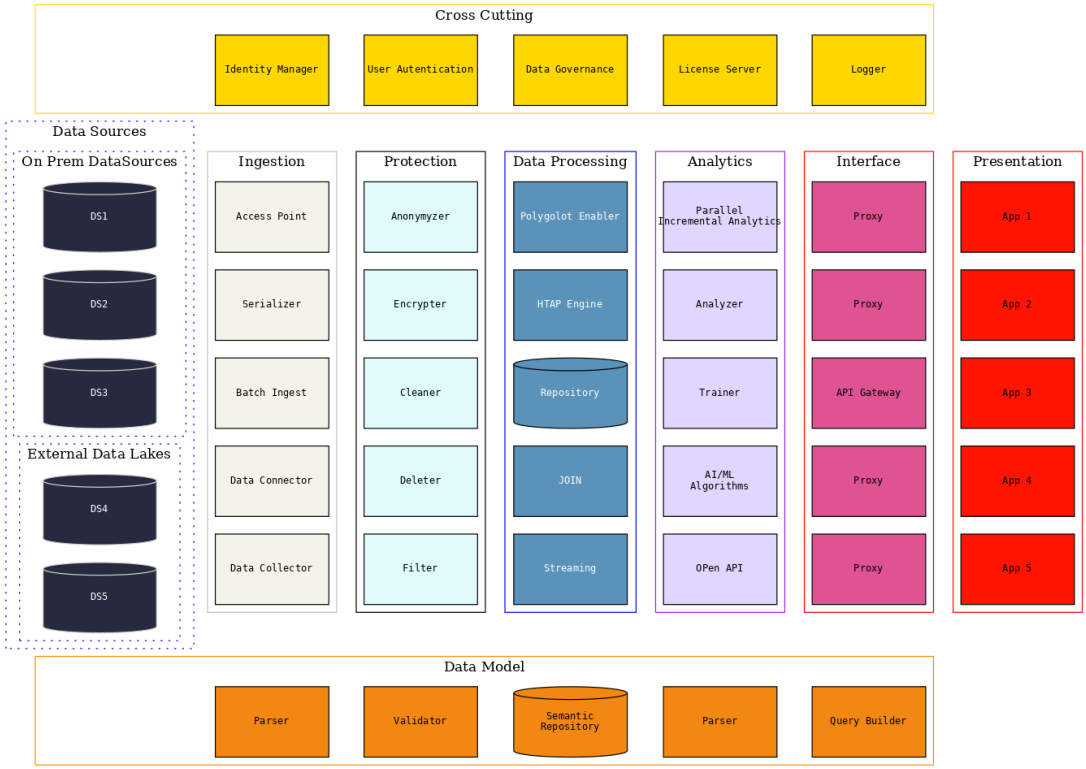


Figure 22 INFINITECH Reference Architecture Logical View – Example Mapping

The IRA defines layers as a way to logically group components. The identified layers are:

- Data Sources: at the infrastructure level there are the source of data (database management systems, data lakes holding non-structural data, etc)
- Ingestion: a layer of data management usually associated with data import, semantic annotation and filtering from data sources
- Security: a layer for management of the clearance of data for security, anonymization, cleaning of data before any further storing or elaboration
- Management: a layer responsible for the data management aspects, including the persistent storage in the central repository and the data processing enabling advanced functionalities such as Hybrid Transactional and Analytical Processing (HTAP), polyglot capabilities, etc
- Analytics: a layer for the AI/ML/DL components
- Interface: a layer for the definition data to be produced for user interfaces
- Cross Cutting: a layer with service components that provides functionalities orthogonal to the data flows (e.g. Authentication, Authorization, ...)
- Data Model: a cross cutting layer for modelling and semantics of data in the data flow
- Presentation/Visualization: a layer usually associated with the presentation applications (desktop, mobile apps, dashboards and the like)

It should be noted that the IRA does not impose any pipelined, or sequential composition of nodes. However, it is recommended to consider each different layer and the relative components to solve specific problems of the use case.

4.3.1 Logical Components Grouping

Deliverable D2.6 of the INFINITECH Project has made a preliminary list of components grouped in categories which can be referred to the IRA presented. These components provide the building blocks that can be combined and used together to create BigData, IoT and AI pipelines for Digital Finance applications in-line with the logical view of the INFINITECH-RA. These components are listed in the table below to facilitate understanding of the INFINITECH-RA examples and scenarios given in subsequent section. For the documentation of the functionalities and the use (inputs/outputs) of these components, readers should consult deliverable D2.6. Note that the presented list of components is extensible i.e. it is likely to be augmented during the evolution of the technical developments of the project.

Table 2 List of INFINITECH Components and Technologies

Component	Category	Infinitech Layer
Relational Database	Data Management	Data Processing
Polyglot Query Processing	Data Management	Data Processing
Incremental Analytics	Data Management	Data Processing
OneHotEncoder	Data Management	Data Processing
Stream Processor	Data Management	Data Processing
Online Aggregates	Data Management	Data Processing
Data Collection	Data Management	Ingestion
Anonymization tool	Data Protection	Protection
DUOS (Digital User On-boarding Services)	Data Protection	Data Processing
DPO (Data Protection Orchestrator)	Data Protection	Data Processing
Blockchain Reader	Blockchain	Data Processing
Blockchain Writer	Blockchain	Data Processing
Smart Contract Executor	Blockchain	Data Processing
Blockchain Data Visualizer	Blockchain	Data Processing
Blockchain Authenticator	Blockchain	Data Processing
Blockchain Encryptor	Blockchain	Data Processing
Blockchain Decryptor	Blockchain	Data Processing
Blockchain Transaction Dataset Preparation Component		Data Processing
Scalable Transaction Graph Analysis Component		Data Processing
Semantic Streams Analyzer	Semantics & Graph Data Model Tools	Data Processing
Semantic Reasoner	Semantics & Graph Data Model Tools	Data Processing

Component	Category	Infinitech Layer
Ontology Mapping	Semantics & Graph Data Model Tools	Data Processing
Semantic Annotator-Preprocessing	Semantics & Graph Data Model Tools	Data Processing
Smart Fleet (IoT Context Management and Historical data component)		Data Processing
Fraud Detection Service Training	Analytics & ML Algorithms.	Analytics
Fraud Detection Service Execution	Analytics & ML Algorithms.	Analytics
Pay As You Drive Service Training	Analytics & ML Algorithms.	Analytics
Pay As You Drive Service Execution	Analytics & ML Algorithms.	Analytics
Investment Recommendation Engine Training	Analytics & ML Algorithms.	Analytics
Investment Recommendation Engine Execution	Analytics & ML Algorithms.	Analytics
Recommender	Analytics & ML Algorithms.	Analytics
Cash Flow Prediction	Analytics & ML Algorithms.	Analytics.
Budget Prediction	Analytics & ML Algorithms.	Analytics
KPI Engine	Analytics & ML Algorithms	Analytics
Transaction (Txn) Monitoring	Analytics & ML Algorithms	Analytics
Transaction (Txn) Categorization	Analytics & ML Algorithms	Analytics
Invoice Processing	Analytics & ML Algorithms.	Analytics
KMeans	ML Algorithm	Analytics
Random Forest (Model)	ML Algorithm	Analytics
Random Forest (Predict)	ML Algorithm	Analytics
Client Contextual Information	Analytics/ML	Analytics
Financial Fraud/Crime Risk Score	Analytics/ML	Analytics
Anomaly Analysis	Analytics & ML algorithms	Analytics
Pattern Analysis	Analytics	Analytics
Stream Story	Analytics & ML algorithms	Analytics
Open API Gateway	Interface	Interface
User Interface for Blockchain Transaction Reports and Visualization Component		Visualization
Visualization Preparation	Data Management	Data Processing
Real Time Visualization	Visualization	Visualization
INFINISTORE	Data Management	Data Processing
UI Risk Assessment based on VaR	Analytics & Visualization	Visualization
Pseudo-anonymization tool	Data protection	Data Processing
Health insurance risk assessment service	Analytics & ML Algorithms	Analytics
Health insurance fraud detection service	Analytics & ML Algorithms	Analytics
Well-being outlook classifiers	Analytics & ML Algorithms	Analytics
Synthetic RWD for well-being analytics	Analytics & ML Algorithms	Analytics
Open Banking Agreggator Solution	Open APIs	Data Processing
BADP (Big data analytics platform)	Category used	Data Processing

Appendix B contains a matrix mapping of RA modules versus User Requirements of D2.1 and D2.2.

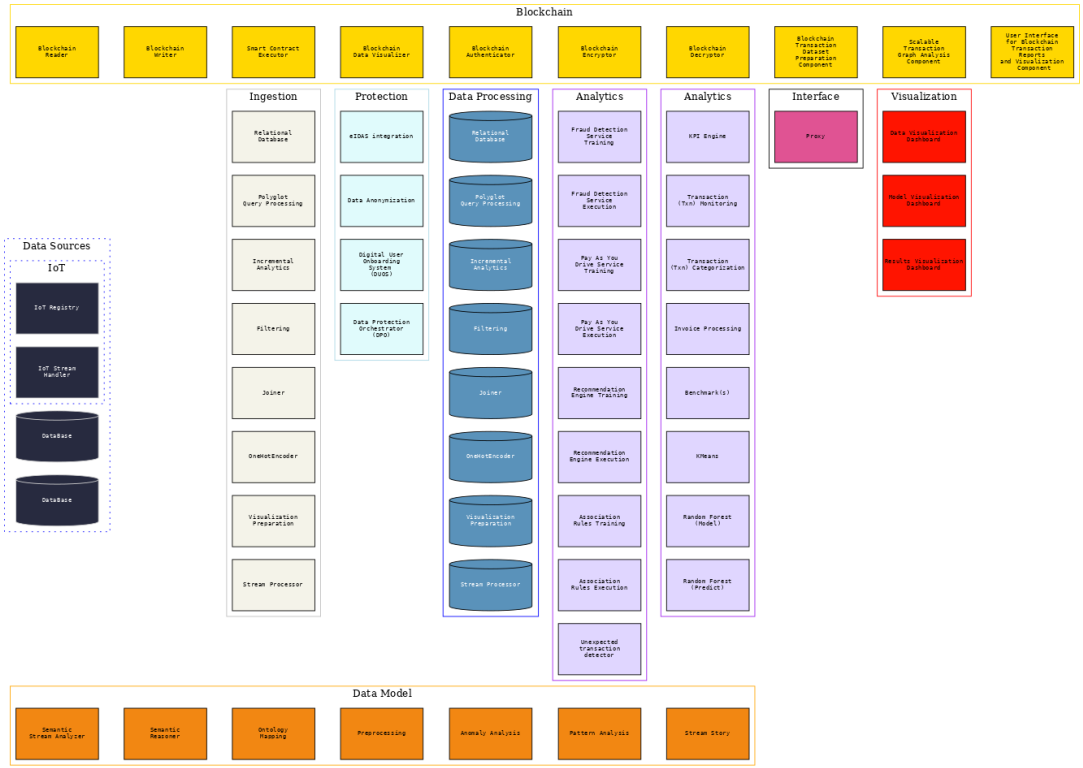


Figure 23 INFINITECH Components

4.4 Process View

In the context of the "4+1" methodology, the runtime behaviour of the workflow will be analysed, to give a representation of the Process View in Section 6. In the specific case of INFINITECH project, the data-driven architecture to design the proposed solutions suggests to identify the Process View with the flow of the data through the different tiers already exposed for the Logical View. These data can be referred to simple events, or more complex situations such as security, analytics, data processing, and can be exchanged among the data components such as data collector, databases, services, dashboard, etc.

A schematic representation of the pilots at runtime is given in the Section on PILOTS, through a number of RA diagrams, each representing a different relevant workflow of the proposed use cases.

4.5 Development View

The INFINITECH RA is designed considering a microservices architecture implementation, with services interacting among them through REST APIs. In that respect, all microservices run in Docker containers on the Kubernetes platform. The following explains the technology background and the development view adopted in the project.

In recent years the DevOps (Development and Operations) methodology has been developed in a sense that is trying to bring, as much as possible, the world of those who develop applications (Developer) and the world of those who manage them (Operations). By bringing the two worlds closer it allows both to understand the problems faced by both sides. It allows them the possibility of being faster and more efficient in the development of an application and its software lifecycle management, because the developers specialize in the environment in which the software will be deployed and the Operation people know best its characteristics before still to put it into production. This type of approach requires not only a technological change but above all a cultural change.

The practical implementation of DevOps goes through the CI/CD. The CI/CD stands for the combined practices of Continuous Integration (CI) and Continuous Delivery (CD).

In order to support the development and testing activities, we have adopted a Continuous Integration approach, using open source tools such as Jenkins. Every time a developer pushes changes to the source code repository, Jenkins automatically triggers a new build of the component and deploys the updated container to an integration environment on Kubernetes. This will allow the team to continuously test their components against an up-to-date environment, speeding up development and avoiding painful integration problems at the end of the cycle. Continuous Delivery is an extension of that process. It's the automation of the release process so that new code can be deployed to target environments, typically to test environments, in a repeatable and automated fashion.

GitLab[1] have been chosen as the source code repository. This tool organizes permissions and accesses into objects called **groups**. A group is accessible by a set of developers that can then create projects within it. Such projects contain the actual source code.

There are two kind of groups, the Pilot Groups and the RA Groups: in the Pilot groups the pilot developers store their projects for a specific Pilot, while the RA Groups contains the projects for all general components that are not pilot specific and can be reused by the different pilots.

The following pictures depict respectively an example of the RA Groups and Pilot Groups structures.

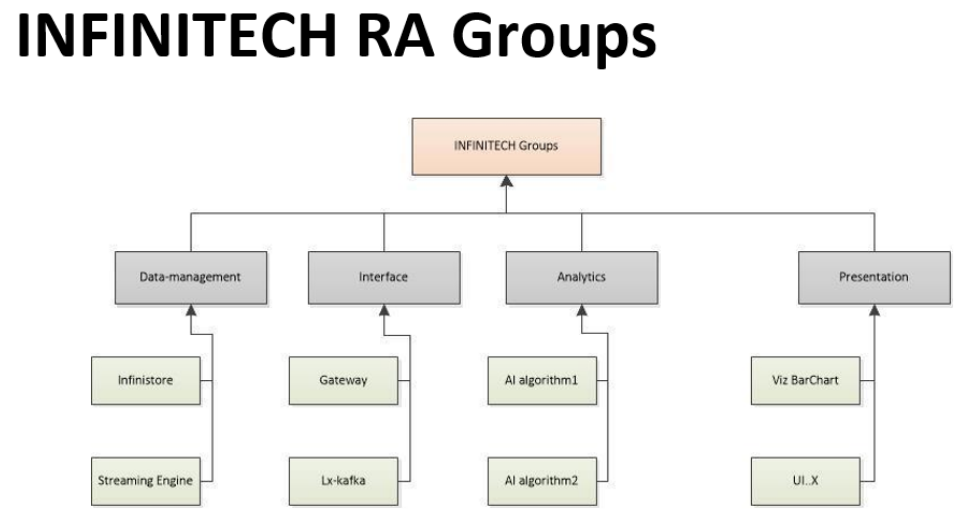


Figure 24 INFINITECH RA Groups

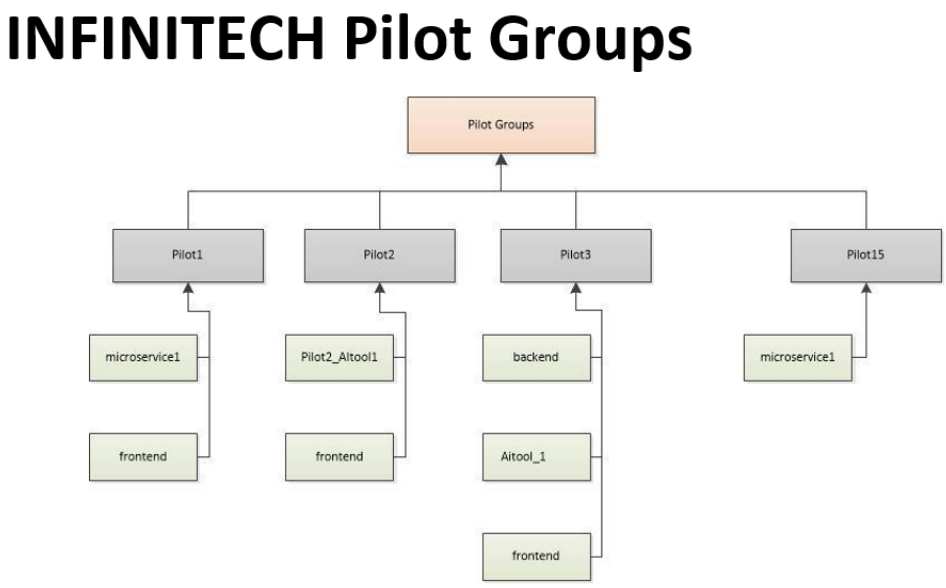


Figure 25 INFINITECH Pilot Groups

We intend to enhance this process by adopting a DevSecOps approach and including the related tools in our CI/CD pipeline.

DevSecOps aims at including security in the software development life cycle since the beginning, following the same principles of DevOps. Since security is considered throughout the process and not just as an afterthought at the end of it, products get a safer, more solid design and teams avoid costly release delays and rework due to non-compliance detected only late in the process.

DevSecOps project teams include security experts, who work with developers and operations to make sure that security requirements are properly addressed and best practices followed, in addition to validating product design and architecture.

Furthermore, based on the DevOps core principle of automation, DevSecOps introduces several security tools in the CI/CD pipeline, so that different kinds of security checks are executed continuously and automatically, giving to developers quick feedback whether their latest changes introduce a vulnerability that must be corrected.

If the DevOps methodology has tried to unify the world of the developer and that of the operation, the MLOps[2] is focusing on:

- Facilitate communication and collaboration between teams
- Improve model tracking, versioning, monitoring and management
- Standardize the machine learning process to prepare for increasing regulation and policy

Putting this into practice is often very complicated, because ML works in heterogeneous environments, for example the models are often on Data Scientist's notebooks, its training is done in the cloud to take advantage of available resources, to finish with the execution of the software in production on premises.

Therefore, the first step towards MLOps requires standardizing these environments as much as possible and in order to achieve this, the Kubernetes along with the Docker containers provide the abstraction, scalability, portability, and reproducibility required to run the same piece of software in all these environments. As a second step it is necessary to make standard the workflow used for construction and building of the ML models.

In this sense, software like Kubeflow[3] provides an infrastructure to build models and it enables the portability of these models and workflows. In particular ML workflows are defined as Kubeflow pipelines. A pipeline consists of various steps:

- Data preparation
- Training
- Testing
- Serving

Each step is a container and the output of each step is the input of the following step. Once compiled, this pipeline is portable across environments. However, the possible usage of MLOPs technologies will be assessed in detail within the next phase of WP6 tasks, in order to enhance the "INFINITECH way" blueprint guidelines for testbeds and sandboxes (detailed in the upcoming WP6 deliverable D6.5).

[1] Gitlab: https://gitlab.com/

[2] MLOps is a compound of Machine Learning and IT Operations. MLOps is a practice for collaboration and communication between data scientists and operations professionals to help manage production ML lifecycle (source Wikipedia)

[3] Kubeflow is a free and open-source machine learning platform built by developers at Google, Cisco, IBM, Red Hat, CoreOS and CaiCloud, and first released at Kubecon North America in 2017 (Wikipedia)

4.6 Deployment View

In the last few years there has been a strong transformation, due to container spread which led to rethinking both how to manage own infrastructure and how to design and build the applications. As we said before the INFINITECH platform has been devised on the concept of a microservice architecture.

The nature of the microservices that is characterized by atomic components, makes them easier to develop, update, and deploy the entire solution, but if on the one hand they allow to have applications that quickly scale according to the requests and that could be easily updated, on the other hand they consider that a software has been previously managed as a single piece indivisible that is split into several dozen of microservices (containers), making it difficult to manage them.

In this context, arose the necessity to develop a tool that was able to manage the life-cycle of the microservices (deployment, scaling, and management), this tool was developed in Google with the name of "Project Seven of Nine "and released as open source software in 2014. Today that tool is known as Kubernetes.

In particular, it will use two main concepts available in Kubernetes in order to implement “Sandbox” in INFINITECH, these concepts are:

1. **Namespaces**: They are a logical grouping of a set of Kubernetes objects to whom it’ possible to apply some policies, in particular:
    - **Quote** sets the limits on how many HW resources can be consumed by all objects
    - **Network** defines if the namespace can be accessed or can access to other Namespace, in other word if the Namespace is isolated or not

    Different namespaces can be given different policies.
2. **POD** is the simplest unit in the Kubernetes object. A Pod encapsulates one container, but in some cases (when the application is complex), a POD can encapsulate more than one container. Each POD has its own storage resources, a unique network IP, access port and options related to how the container/s should run.

The Kubernetes Namespace allows us to isolate logically the objects (mainly PODs) inside it from other Namespace, in other words each Namespace will be one use case of a specific pilot.

Regarding the “Testbed”, the set of hardware resources of each Data Center (Storage, Compute and Network) or VPC (Virtual Private Cloud) on the cloud provider will be considered a Testbed.

Therefore each dedicated Testbed will have only one cluster Kubernetes with many Namespace as many as are the use case to implement, instead for a shared testbed will have many clusters Kubernetes as are the pilots to manage.

Moreover, in order to simplify the development and integration by each partner, a blueprint deployment has been created on Amazon web services (again, following the INFINITECH blueprint guidelines for testbeds and sandboxes, see Deliverable D6.5). In particular, on the cloud provider has been created a cluster Kubernetes (EKS[1]) upon which has been implemented a set of tools for CI/CD:

- GitLab(Git) as software version (see section 4.5)
- Jenkins as pipeline manager
- Harbor as Docker registry

Finally, we’ve developed an environment in order to test the INFINITECH services/components as a separate Namespace in the same Kubernetes cluster where CI/CD infrastructure runs. Such development environment is fully integrated with the CI/CD tools.

After that, for all the pilots partners that will choose to use the same cloud provider (AWS[2]) or another supported one (e.g. Microsoft Azure, etc.) for their final deployment, we planned to provide the automated replication feature, i.e. possibility to replicate a similar environment using Terraform[3],

scripts, tool which allows to create the “infrastructure as code”. Regarding the pilots on premise deployment environments, we will try to automate as much as possible the recreation of the blueprint environment using tools like Terraform, Vagrant[4] and Rancher[5].

The general flow for an on premise architecture based on vSphere is depicted in Figure 25.

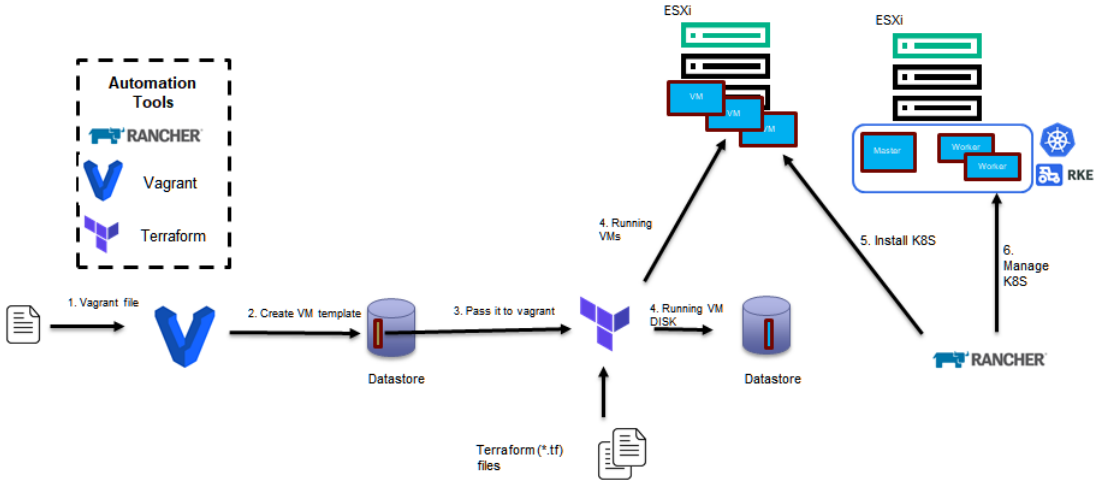


Figure 26 Automated K8S cluster creation with Rancher

1. We will use Vagrant to create the VM template on vSphere.
2. We will use Terraform to automate the deployment of VMs based on templates created in the previous step.
3. We will use Rancher to deploy the Kubernetes clusters on the Virtual Machines created in the previous step.

Further details and specifications, about the previous concepts has been reported in the WP6 deliverables D6.4 “Tools and Techniques for Tailored Sandboxes and Management of Datasets I” and D6.10 “Sandboxes for FinTech and InsuranceTech Innovators- I”, and will actually be realized in the next phase of WP6 work.

[1] EKS: Elastic Kubernetes Service (https://aws.amazon.com/eks/) is Amazon Kubernetes managed service

[2] AWS: Amazon Web Services (https://aws.amazon.com/) is Amazon cloud offer

[3] Terraform : https://www.terraform.io/

[4] Vagrant: https://www.vagrantup.com/

[5] Rancher: https://rancher.com/

4.7 Use of INFINITECH-RA in Common BigData and AI Scenarios – Initial “Scenarios” View

The following subsections present some sample reference solutions for common use case scenarios in the sector.

4.7.1 Simple Machine Learning Pipeline

The reference architecture enables the design and execution of classical machine learning pipelines. Typical machine learning applications are implemented in-line with popular data mining methods like the CRISP-DM (Cross Industry Standard Process for Data Mining) [Shearer00]. CRISP-DM includes several phases, including the phases of data preparation, modelling and evaluation. In typical Machine Learning pipeline, data are acquired from various sources and prepared in-line with the needs of the target Machine Learning model. The preparation of the data includes their segmentation

into training data (i.e. data used to train the model) and test data (i.e. data used to test the model). The model is usually evaluated against the requirements of the target business application. Figure 26 illustrates how a typical machine learning pipeline can be implemented/mapped to the layers and functionalities of the INFINITECH-RA. Specifically, the following layers/parts are envisaged:

- Data Sources: At this level reside the data sources of the BigData/AI application. These may be of different types including databases, datastores, datalakes, files (e.g., spreadsheets) and more.
- Ingestion: At this level of the pipeline data are accessed based on appropriate connectors. Depending on the type of the data sources, other INFINITECH components for data ingestion can be used such as data collectors and serializers. Moreover, conversions between the data formats of the different sources can take place.
- Data Processing: At this level data are managed. Filtering functions may be applied and data sources can be joined towards forming integrated data sets. Likewise, other preprocessing functionalities may be applied, such as partitioning of datasets into training and test segments. Furthermore, if needed, this layer provides the means for persisting data at scale, but also for accessing it through user friendly logical query interfaces. The latter functionalities are not however depicted in the figure.
- Analytics: This is the layer where the machine learning functions are placed. A Typical ML applications entails the training of Machine Learning model based on the training datasets, as well as the execution of the learnt models based test dataset. It may also include the scoring of the model based on the test data.
- Presentation: This is the layer where the model and its results are visualized in-line with the needs of the target application.

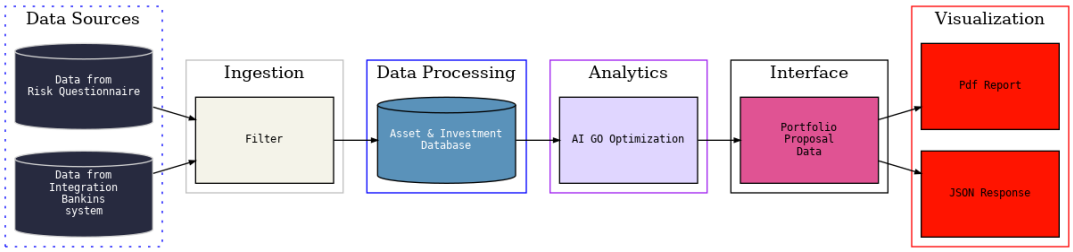


Figure 27 Simple Machine Learning Workflow Example implemented in-line with the INFINITECH-RA

4.7.2 Blockchain Data Sharing and Analytics

INFINITECH platform can also facilitate the execution of blockchain empowered scenarios for the financial sector, where e.g. several discrete entities (including banks, insurance companies, clients etc.) may be engaged in data sharing activities, thus empowering these financial organizations to update customer information as needed, while being able to access an up-to-date picture of the customer's profile at all times. This is extremely significant for a plethora of FinTech applications, including yet not limited to Know Your Customer / Business, Credit Risk Scoring, Financial Products Personalization, Insurance Claims Management and more. Let's consider a use case in which a Financial Organization A (e.g. a bank, or an insurance company) wishes to be granted access to a customer of another financial organization (e.g. another bank within the same group, or a different entity altogether). Supposing that legal constraints are properly handled (e.g. consents for the sharing of the data has been granted, and all corresponding national and/or European regulations such as GDPR are properly respected and abided by), and that all data connectors (required for the retrieval of the data from the raw information sources) and data curation services (e.g. the Cleaner, required for cleaning the raw data sources, the Anonymizer, required for anonymizing the same information

sources, and the Harmonizer, required for mapping the cleaned, and properly anonymized information sources to the Common INFINITECH Information Model) are in place, enabling the collection and pre-processing of the raw information sources, the following flow is envisaged:

1. The Blockchain Encryptor encrypts the properly pre-processed data so that they can be saved in the private collection of Organization A.
2. The Blockchain Authenticator authenticates (the user of) Organization A, so that access to update the ledger is granted.
3. Once authenticated, the Blockchain Writer inserts the encrypted data to the private collection of Organization A and the hash of the data is submitted to the ledger.
4. Organization B requests access to the data of a customer from Organization A.
5. The Blockchain Authenticator authenticates (the user of) Organization B that initiates the request so as to grant or decline the request to access to private-actual data.
6. Once (the user of) Organization B is authenticated, the Smart Contract Executor translates the Query submitted and queries the private collection of Organization A.
7. The Blockchain Authenticator authenticates (the user of) Organization B, so that access to read the ledger is granted.
8. The Blockchain Reader retrieves the queried information from Organization's A private collection.
9. The Smart Contract Executor generates a new transaction and triggers the Blockchain Authenticator so as to authenticate (the user of) Organization A, in order to grant access to update the ledger.
10. Once authenticated, the Blockchain Writer submits the encrypted data to the ledger, and a new record on the same ledger is created, containing the metadata of the contract (organizations involved, data created, metadata of the encrypted data transmitted, validity of the contract) and the actual encrypted data.
11. Organization A sends out of band the decryption key to the Organization B, and the Blockchain Decryptor decrypts the queried data.

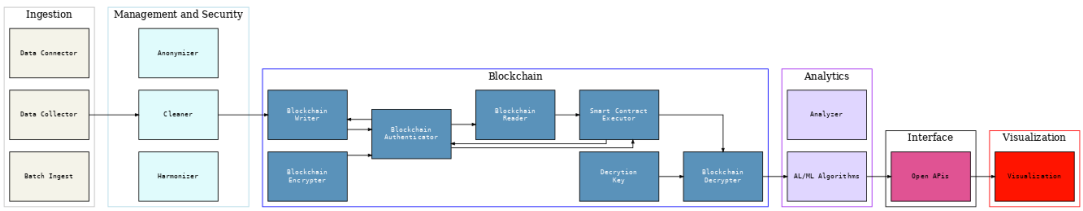


Figure 28 Blockchain Data Sharing and Analytics Pipeline

4.7.3 Data Ingestion, Anonymization, Analytics and Visualization Pipeline

One of the most common operation in BigData and AI pipelines involves data acquisition, anonymization and processing using data mining or machine learning technique. Data acquisition is usually the starting point for data scientists and AI/ML developers.

In ML scenarios (research, PoCs, experiments, etc) the importance of keeping data privacy and being compliant with GDPR is crucial especially in context like insurance, banking, and financial services. These scenarios remark the importance of anonymization data as a part of AI pipeline development, during the data preparation phase. However, anonymization is not only about hiding personal data . More complex situations would infer personal data from data in principle not so directly related to a person.

As an example example: information from GPS positioning would infer movement patterns that would easily point to a very concrete person. In this case, if GPS information is crucial for the model, GPS coordinates need to be converted to not so accurate information. For example, converting to GPS coordinates to just generically point to the city where the events happened.

With the data prepared, privacy respected and features ready, the AI developer will start creating new models to infer new information from the current existing ones. ML phase could be divided, at least, in two components the model training and the model deployment. The model training uses the input data to learn from the past, looking for patterns, that would infer or predict future situations. With the model ready, this is deployed in order to provide a concrete service: Is it going to rain today? Are my stocks going to fall? Am I a good driver?

The visualization shows the results of querying these deployed models to show the responses for a final user. Visualization would be a very specific GUI for that model, or part of a more complex existing dashboard. In this point, INFINITECH fosters REST APIs and JSON data formats for quick and robust integrations. INFINITECH WP4 is in charge of providing the (semantic) data models making these integrations scalable and replicable in different situations and scenarios.

4.7.4 Data Exchange and Semantic Interoperability Pipeline

The INFINITECH reference architecture addresses the need for defining the methods and conventions alike the recommendations for naming, identification, categorising about the use, manipulation and inclusion of data sets in the INFINITECH project. There are regulatory aspects about the data that are not addressed in this document but in data management plan, rather this section focus on the operational part of the data, looking closely at semantic and interoperable aspects within the project.

This section related to data exchange and semantic interoperability pipeline also identifies the different important reference use cases addressing general INFINITECH Pilot aspects as reference use cases across different Pilots. The data exchange and interoperability aspects are according to the WP4 implementation work in accordance with the signed Grant Agreement of the INFINITECH project and with respect to EC Horizons 2020 recommendations for operating data.

The information provided in this section is meant to define common grounds in relation to data identification, data annotation, data sharing, data exchange and data sharing and must be used for the foundations of successful internal and external Pilot data interoperability aspects in order to motivate participation and collaboration within the INFINITECH consortium partners at the mainly at the pilot level but also between and amongst external partners or participants in the project by directing the developments and integration of semantic interoperability and data exchange aspects.

The following pipeline is similar to the classical data collection / data analytics pipeline, yet it includes INFINITECH components and data transformation that facilitate semantic interoperability.

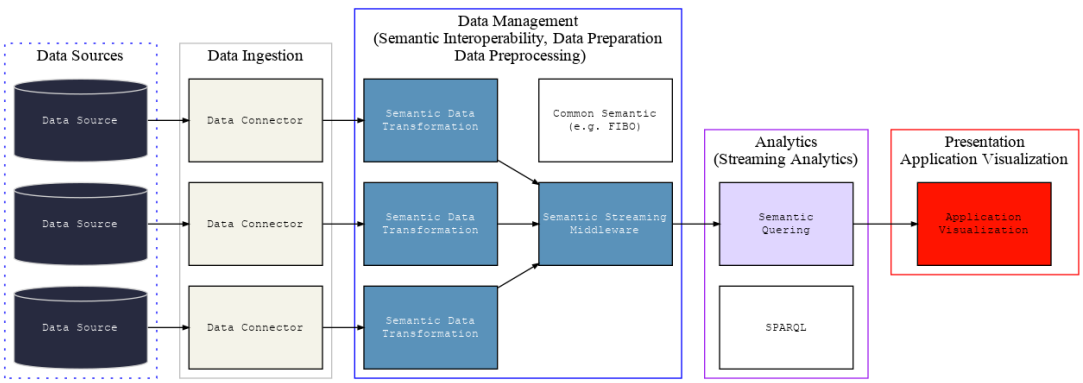


Figure 29 Pipeline for a Scenario involving Semantic Interoperability Across Diverse Data Sources

The Semantic Interoperability pipeline and its pre-conceived components (specifically semantic data transformation components, semantic streaming middleware, semantic querying and visualisation of semantic data) are under development in WP4 and are designed and intended to be used to unify the

semantics of diverse data sources. This approach can accordingly enable the use of the INFINITECH Semantic Streaming Analytics components in the pipeline enabling multiple general semantic interoperability operations and its particular use cases, some of them are described as follow:

- **Data Identification:** The data identification points in INIFINITECH are where data will be used, a part of any generalization and/or categorization the data is a representation of a fact (virtual or physical) and thus its identification and intended usage modify its way for communication. Identification is a process where the data is not only identify but categorised and classify with the purpose of:
    - Outline the discoverability of data (metadata provision)
    - Outline the use of identifiers of data and refer to standard identification mechanism. e.g. Do you make use of persistent and unique identifiers such as Digital Object Identifiers?
    - Outline naming conventions used
    - Outline the approach towards search keyword
    - Outline the approach for clear versioning

  Example use case or application use: Identification of the data in INFINITECH can be useful when the process of joining specified datasets with the updated new datasets version is required, by maintaining versions of this datasets multiple benefits of interoperability are activated. For example main categories of data assets can be identified by INFINITECH pilots and comprise both the data created previous any operation via INFINITECH solution, data sets used during the operation of the INFINITECH Pilots solutions and the data produced as result of the INFINITECH services process can be also identified.
- **Data Annotation**: In INFINITECH annotation procedures shall be used to coordinate at the different deployment sites on how to collect the data locally and cross pilot, if required. In some cases, it is a questionnaire presented to the user before and after the pilot; in other cases, is the result of tracking the usage of the system. Data annotation is a process for gathering the information according to reference vocabularies or conventions from domain professionals to enhance the available data and make it machine understandable; etc. Therefore, it is expected that in INFINITECH each deployment site will have its own data format and collection method that will be connected INFINITECH Open Data Model as a collection online available tool, and if desired all sites will share the same tool for the same data format, as well as the same procedure to annotate the data. Annotation is a process where the data is enhanced with the purpose of:
    - Data consumed and/or handled within the INFINITECH project by means of electronic and non-electronic forms.
    - Data Registered using external information to the project that will be derived to the INFINITECH project by means of electronic and non-electronic forms.
    - Have control over the processes following particular data lifecycle.
    - Data Exposition by means of data that has been registered and or produced in the INFINITECH project and could be distributed by means of electronic and non- electronic forms.
    - Data Management by means of data and information that can be registered and produced within INFINITECH mediums, by means of, electronic and non-electronics forms that is used within the consortium for processes, operations and or simply usage for the normal activity within the project.

  The following are the envisage use cases involved in data capture annotation: When data sets that will be produced as result of management and operations with other sources of information. When data sets that will be provided to the INFINITECH project to store, handle

and or processing. This is the primary activity in INFINITECH. And finally if data sets need to be make them accessible via interfaces to get the information used and/or generated by INFINITECH providing that those data sets are used for informative process.

- **Data Sharing**: Sharing data sometimes is equally treated as the process of exchanging data, however in INFINITECH and following dataflows best practices, this is a process to guarantee a durable threat for the available (stored or not) data. In a wider definition data sharing of data defines the process of formulating and defining the ways that guarantee the data will remain during the time in different instances or data holders with all the guarantees of identification, traceability and ownership and to ensure that the data will be useful for the same purpose which it was created. Data Sharing is a process where the data is treated with the purpose of:
    - Migrate the data to best format
    - Migrate the data to another medium or a more suitable medium
    - Back-up stored data to preserve the information
    - Create metadata and documentation for the stored data
    - Archive data in a defined physical or virtual medium

  Example use case or application use: In INFINITECH, pilot-generated data and/or external data to the project that will be necessary used to derivate to INFINITECH project services by means of electronic and non-electronic data records. Data sets that will be provided by INFINITECH Pilot(s) to be stored, handled and or processed. Data captured by the INFINITECH pilot infrastructure defined on each pilot with the particularity that in the case of the data generated by the INFINITECH Pilot solutions must follow a defined semantic description approach based on INFINITECH standard-adopted formats.
- **Data Exchange**: Enabling data exchange defines the adequate ways to access the data but also the treatment of the data by its different involved entities or subjects to operate the data. Data Exchange is one of the main challenges in INFINITECH project. Data Exchange is a complex process because based on the demand for the data is the way to facilitate data access. If the demand is coming from multiple parts the access process is multiplied, first because there are activities associated that are specific for every part demanding access to the data and second because if the data is not prepared to be used by multiple parts it has to be pre-processed to repeat parts of the lifecycle in generating new data. Additional to that data exchange imply Data Access making necessary to define data values against the data. Data exchange is a process where the data is not only identified, categorised, classified and annotated with the sole but not limited purpose of :
    - Distribute Re-distribute the data
    - Share and format of the data
    - Mechanisms for control and accessibility
    - Establish copyright for the data
    - Promote data

  The following are the envisage use cases for data exchange that can be applicable in INFINITECH, Users of the INFINITECH pilot looking at data sets for public evidence e.g. profile building, liability aspects, ownership of data etc. INFINITECH users and professionals can benefit from Data Exchange by being informed about the data exchanged and shared with evidence and impact rather than just a reporting or informative element. INFINITECH users at the Pilot level using data dashboard(s) which will provide the view and access to data.
- **Data Discovery**: INFINITECH data discovery focuses on providing technological means for uncover aspects of the data and the data itself by using semantic annotation and descriptions that are not considered evident or as part of the data itself. Discover relies in the INFINITECH lifecycle simply because the demands for the information is to generate a meaningful outcome that helps to inform, observe, visualize features of the data. Data Discovery will ensure the re-usability of the data and the ways to transform it into meaningful results. In INFINITECH Data

Discovery provides mechanism for data transformation, correct data collection and data re-usability and that the data can be repurposed. Data Discovery is also process where the data is treated with the purpose of:

- Follow-up data along other data places or data sources
- Identify or define new Data lifecycles
- Undertake information about the produced data
- Find points for storing/improving data
- Learn from the shared data
- Identify better mechanisms for sharing data
- Define post process beyond re-using and transforming data

Example use case or application use: Discovery can look at the data processes in the INFINITECH lifecycle in order to enrich the capacity to look at those data characteristics and operations. Discovery is one of the main objectives in INIFNITECH for enabling analysis having more implications based on a specific demand of the data discovery following the original representation and basic information provided when the data was created.

4.8 Mapping Methodology for Use Cases in the RA

The goal of INFINITECH RA is to create a RA for facilitating the realization of Big Data pipelines dealing also with IoT and Blockchain data sources utilizing ML techniques in distributed computing environments.

As stated before, the IRA is NOT a one-size-fits-all solution but rather a schema to compare the solutions against best practice and standard flows and components. A methodology to check the Use Case with the IRA is suggested as follows:

- A Use Case will be considered as a transformation from some Data Sources to some Data Destinations. This is called the “stretching phase” where a pipeline or workflow is defined from the different sources to the resulting data.
- The INFINITECH RA is the total transformation of the sources into destinations once the appropriate workflow is defined and implemented.
- Each Data Sources are ingested into the INFINITECH platform via a first layer of Data Management components (serializers, database connectors, …)
- Data Sources can be accessible from different types of data management technologies (SQL, NoSQL, IOTs streams, Blockchains’ data but also raw data such as Text, Images, Videos, etc.) and a connector must be provided in order to support all the variety of the supported target datastore management systems.
- Data should not be stored into the INFINITECH platform data layers, unless they are “cleared”. Clearing can involve filtering, deletions, anonymization, encrypting of raw data. These transformations are different among the various datasets. This “clearing” can be managed via the security layer of the RA model or the cross cutting services.
- The Data Model provides reference for ontologies and semantics across the nodes.
- In most use cases, most of the elaborations, it is required to store data (filtered by the processes beforehand) in a data store. The datastore of the IRA can be a very complex infrastructure able to manage big data and should be able to scale out accordingly in order to support the diversity of data and work load.
- A generic data component should provide the basic CRUD (create, retrieve, update, delete) endpoints along with advanced endpoints (like streaming in Real Time, HTAP capabilities) with standard API. When standard RESTful APIS are provided components can be stacked (piled up as the LEGO concept).

- Analytics work on top of the Data (stored or streamed), consume and might produce data, which is stored into the data repository as well.

However, many other points must be considered. Most of the real-world use cases can be much more complicated than the methodology proposed above. They are intertwined with existing infrastructures and data processing components and cannot be easily stretched into a “pipelined” workflow as suggested. In those cases, it should be possible to identify the boundaries of “computational nodes” that provide basic interfaces. A complex infrastructure can be more than one microservice with many endpoints distributed among the different microservices in order to keep the functionalities homogeneous. On the other hand, a huge infrastructure can be encapsulated (wrapped) into a microservice interface to exhibit basic functions. When this exercise is completed, a map of a sandbox to the IRA can be provided as a reference schema to guide (or at least to benchmark) the implementation.

4.9 Infinitech Flow

The section presents a framework to manage financial data applications to manage infinite use cases to speed up development, deployment and testing exploiting the concepts of sandboxes and testbeds. Moreover, it presents an idea for a web application (WEB APP) to create complex workflows pipelining basic components to perform transformations on datasets and data streams. Components must be interoperable with standard mechanisms to exchange data in and out. More specifically a precise definition of dataset is what it takes to exchange data, independently of the type and semantic of data.

The following concepts have been theorized in the INFINITECH RA, but has not (yet) implemented a tool to support the creation, development and deployment of INFINITECH pipelines, in the same way frameworks like KMINE, StreamPipes described in section 3.3 provide.

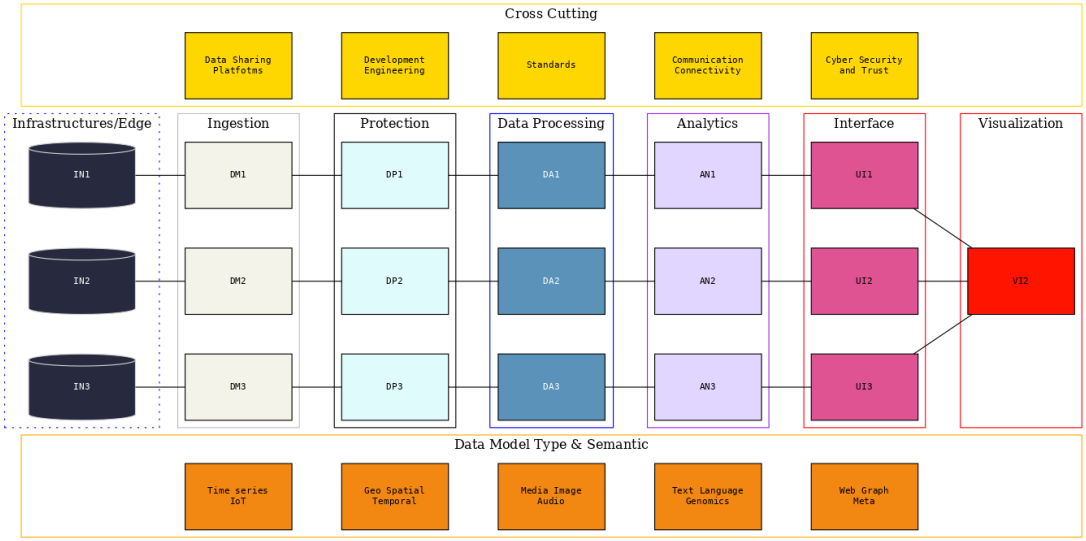
4.9.1 Basic Concepts

Ideally a data scientist would like to solve problems of data manipulation by putting in cascade nodes that transform data, from source to destination. Sources are typically databases but recently BigData from different sources (large file systems) and IoT (data coming from network of devices) should be considered. Destinations comprise applications for graphical or textual presentation of results, like documents and reports or simply data stored in another database for further elaboration. In between these extremes, there are different types of transformations, filtering like anonymization, manipulation like conditional combination of data and more importantly analytics and ML algorithms that dig into data producing a completely different set of data.

4.9.2 From RA to Implementation

Most, if not all, the use cases in data science can be reduced to a workflow of data processed in steps from source to destination. The INFINITECH RA, proposes a left-to-right elaboration with nodes in cascade responsible for typical elaborations from ingestion to presentation, stepping through filtering, data management, analytics and visualization. The RA serves as a conceptual schema where the problems and the solutions can be described in terms of data manipulation.

The RA actually does not impose how nodes must exchange data. These are left to implementation and are use case dependent. However, an approach where nodes adapt to the specific problem does not scale and implies “reinventing-the-wheel” most of the time. A better approach is where nodes follow a standardized and uniform interface that allows them to interoperate with other nodes that follow the same conventions.



To be reusable a block must be compatible with other blocks, especially in the way it manages the data flow. Therefore the problem to be solved are:

1. How data are ingested into a block (DATA IN MECHANISM)
2. How data are filled into the next block in the chain (DATA OUT MECHANISM)
3. How the node behaves to work on the data IN to produce data OUT (NODE CONFIGURATION)
4. How a node knows about the syntax/semantic of the DATA IN and how the node produce syntax/semantic of DATA OUT to be exploited by other nodes in the chain

To concrete these concepts a standardization of what is in a dataset must be adopted. This is defined in the following section.

4.9.3 WHAT'S IN A NAME

Data can be managed in various ways (SQL and NoSQL Databases, JSON objects, CSV files etc). In order to be uniformly treated data can be represented in the form of DATASET/ DATARECORD/ DATAID/ DATAVALUE. The following is the assumption in INFINITECH.

Data can be managed in a SQL (rigid schema) Database, in a NoSQL (more flexible) Database, in a JSON array with key/value pairs objects, in CSV/Excel files with rows and columns and more.

INFINITECH	**SQL**	**NoSQL (eg. MondoDB)**	**JSON**	**CSV/EXCEL**
DATASTREAM	-	stream watch	JSON Array []	File
DATASET	Table	Collection	JSON Array []	File
DATARECORD	Row	Document	JSON object { }	Row
DATAID	Column name	Field id	key	Column
DATAVALUE	Column value	Field value	value	value

In INFINITECH a representation at a syntactic level is as follows:

> **dataset/key1/key2/key3/...**

Or simply:

> **dataset**

In INFINITECH the underlying assumption of the nature of the data is something else knows about the data and in case this information is supplied to the node that must process them. The most important thing for a node is to know which dataset(s) are input, and which is the next node in the cascade to supply the output. This is the work of data scientists when they model the solution. The rest is left to the underlying implementation of the "INFINITECH way" to develop and deploy a node.

4.9.4 The IMPLEMENTATION view

For any kind of manipulation, even the simplest, knowing the name of the dataset or the keys is not sufficient. More information is needed, e.g. where the data are, what type, how many etc. In the proposed schema these information have defaults:

- DATASET is the name of a TABLE/COLLECTION to be found in the underlying default database
- DATAIDs are the column/field in tables/collections
- DATAVALUEs are of type "string" with a unicode string representation of values.

Moreover the INFINITECH solution relies on (transparent) components which supply the information on request:

- A **default database** is present and a node is preconfigured to read and write data. This is either SQL or NoSQL and in INFINITECH is the LXS INFINITECH DATASTORE
- A **configuration server** provides configuration about the specific node in its current deployment. For instance a join component when deployed needs to receive which operation (INNER, LEFT, RIGHT join) on which column to operate.
- An **ontology server** provides more datatype/semantic information about a dataset supplying metadata on demand. An INFINITECH node can adapt its behavior to the nature of the data.

On these assumptions, the implementation view is as follows. In INFINITECH a standard node manages DATASETS and in theory all he needs to know is what is the name of the dataset in input and the dataset produced to be communicated to the next node in cascade. The datasets names are communicated in the REST API of the node implementation as a microservice.

As an example, a node is triggered by an API (e.g. a POST) that has something to do with a DATASET received in input. After elaboration it produces for the next node in the chain a trigger (an POST) with the DATASET in output.

However, different problems both in the Logical and Implementation view must be solved.

1. What is in the data?
2. Where are the data?
3. When are the data available?

To solve 1) a node receiving only the dataset "name" can ask an ONTOLOGY server what is in there. When it produces data, the node will write to the Ontology server the metadata concerning the dataset, like key names, key types, ranges, etc. An ontology about the data is therefore another dataset with the metainformation about the DATASET it describes.

4.9.4.1 When data are available

When a node connects to another node, the data will be available depending on the type of API call. Two types are supported:

- Batch elaboration (use with GET and POST API)
- Streaming (i.e. realtime PUT API)

4.9.4.2 INFINITECH NODE REST API

An infinitech node, by definition, has the following standard API to communicate with other standard nodes.

HTTP Standard Methods	REST API	Parameters in URL	Parameters in object	Action performed
GET	*get**	*dataset/key1/key2/ key3 ...**	**None**	**Work on data**
POST	*post**	*dataset {}**	**[{key1:value1, key2:value2},..]**	**Work on data specified in the object**
PUT	*put**	*dataset/key1/key2/ key3...**	**{key1:value1, key2:value2},..**	**Stream the data in object**
DELETE	*delete**	*dataset/key1/key2/ key3...**	**N/A**	**Dataset and/or keys are not available anymore**
OPTIONS	*options**	*key1/key2/...**	**Can be specific**	**Can be specific**
PATCH	**N/A**	**N/A**	**N/A**	**Not used**
CONNECT	**N/A**	**N/A**	**N/A**	**Not used**

4.9.5 Ontology server

Another important point to handle is interoperability. How can the components understand the information received from other components? The idea proposed is to make use of a simple key-value pairs format without defining a particular syntax. Then, an Ontology Server will support the other services providing information on the terminology used. To better explain the concepts behind this idea, we will consider an example scenario, in which an Analytics components needs to process information on transactions whose amount is greater than 10 thousands euros and perform on January 2021. The information are collected from three data sources: *S1*, *S2*, and *S3*. Suppose that *S1* is a SQLServer database, while *S3* is a MongoDB. The Data Collector interacts with the Ontology Server to understand how to query the different data sources. The example scenario is depicted in the following figure.

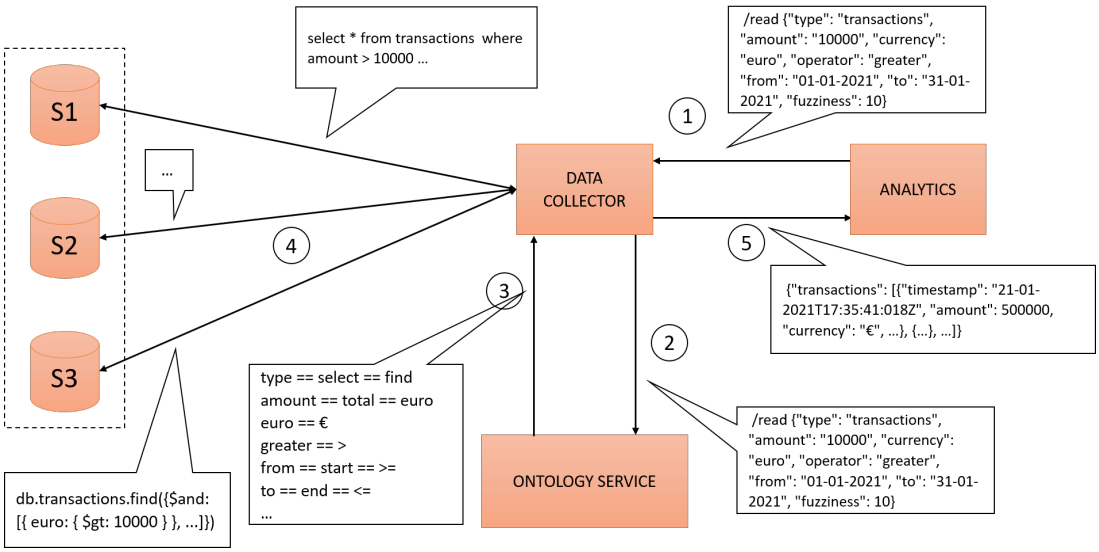


Figure 30 Ontology Server

- The Analytics component performs a request (/read) to the Data Collector component to collect from the data sources *S1*, *S2*, and *S3* all the information related to the transactions with amount greater than 10000 euros performed between the 01/01/2021 and the 31/01/2021. Moreover, the Analytics specifies a level of fuzziness;
- The Data Collector performs a request (/read) to the Ontology Server to understand the meaning of the request performed by the Analytics;
- The Ontology Server provides the Data Collection with the following indications:

    1. The *concept* underlying the request is the transaction: this means that, for example the word transaction (or a synonym) could the *from* in a query to *S1* (SQLServer), or the collection for the find operation on *S3* (MongoDB);

2. The amount, the currency, from, and to are *filters*: for example, these keywords could appear in a *where* (or a *on*) of a query to *S1* (SQLServer), or as a composition of query criteria of the find operation on *S3* (MongoDB);
3. from and to are two dates, then the two keywords could define a time window;
4. “amount” could be a synonym of “total”, or “euros”;
5. “euros” is synonym of “€”;
6. ...

- The Data Collector can then prepare the queries to the different data sources according to the type of source, the information regarding the database (for example, in case of SQL database, the database schema), and the information provided by the Ontology Server. Suppose that the Data Collector knows that the table *transactions* in *S1* contains information on the transactions. The Data Collector could then compose the following query for *S1*:

```
SELECT  *
FROM      transactions
WHERE    amount > 10000 AND currency = "€"
            AND timestamp >= Convert(datetime, '2020-01-01' )
            AND timestamp <= Convert(datetime, '2020-01-31' )
```

- Instead, for *S3* , it compose the following command:

```
db.transactions.find({
  $and: [
    {euro: {$gt: 10000}},
    {timestamp: {$gt:ISODate('2021-01-31T00:00:00.000+00:00'),
      $lt:ISODate('2021-01-31T23:59:59.999+00:00')}}
  ]
})
```

- Once the Data Collector prepared the queries to the different sources, it queries the different databases using their own query;
- The Data Collector collects the information returned by the different data sources in a simple key-value pair format.

Notice that the response of the Ontology Server (and then the queries composed by the Data Collector) depends on the level of fuzziness specified. When the fuzziness is 0, the queries will contain exactly the same words specified by the Analytics. The higher the fuzziness value, the more the Ontology Service returns matches (synonyms, etc.).

4.9.6 Configuration server

When deployed to perform a transformation over dataset(s) in input and to produce output, a node needs (most of the time) to be configured. Example a component that performs a JOIN of 2 datasets

4.9.6.1 Example JOIN

This node combines two tables similar to a join in a database. It is used to combine rows from two or more tables, based on a related column between them. The configuration gives the node the parameters to perform (keys, type of JOIN (inner, left, right, full).

This results in the equivalent SQL operation:

SELECT key1a, key1b, key2a, key2b, ...

FROM DatasetA

INNER JOIN DatasetB ON DatasetA.key1a=DatasetB.key1b;

4.9.7 User Interface

The following presents the concept of a WEB Application to manage the creation of dataflows in the INFINITECH context.

The application presents a graphic user interface with a palette of nodes that can be dragged and dropped on a canvas and connected to create flows.

- Node list on the left (grouped by layer)
- Canvas to create data flows
- User drags and drop from Node list to Canvas and connect them
- Connections are different:
    - Pipeline (GET)
    - Move Data (POST)
    - Stream (PUT)
    - Configure (OPTIONS)

In the Infinitech terminology this operations are to create SANDBOXES to be deployed in TESTBEDS

Once created the canvas a user can:

- Connect the nodes (connect the dots, in different ways)
- Configure nodes and DB
- Add metainformation to nodes about the datasets
- Generate the scripts to trigger the CI/CD deployment)
- Export the scripts
- Run the scripts into one or more cloud infrastructures (i.e. run the sandbox)

4.9.8 Conclusion

The section has described a more advanced Framework for data Pipeline management, in the context of INFINITECH. The advantages with respect to other frameworks is that it solves the interoperability and configurability cumbersome activities introducing configuration and semantic servers that relies on conventions and ontologies to autoconfigure components in pipelines.

The disadvantage, is that it is not (yet) implemented, and this section is to set the basis for such an implementation to solve the gap between INFINITECH Pilots design and implementation in testbeds and sandboxes.

4.10 RA and Building Blocks Interoperability

The INFINITECH Reference Architecture (IRA) supports the idea to create solutions piling components from different groups to transform information to satisfy the requirements of the business case while, at the same time, proposing a layered approach to take into account specific data management procedures like ingestion, protection, data storage and semantic interoperability, analytics, interaction, presentation and more. In this respect the IRA provides not only a conceptual framework but a concrete tool to solve large class of problems with best practices approaches and solutions used in different business scenarios.

Besides the correspondence with the BDVA Reference Model, the idea to solve data problems with pipelines and workflows is close to the way data scientists work and resembles the powerful concept of neural networks. Nodes fire other nodes with data and transform the sources in the desirable result along the way.

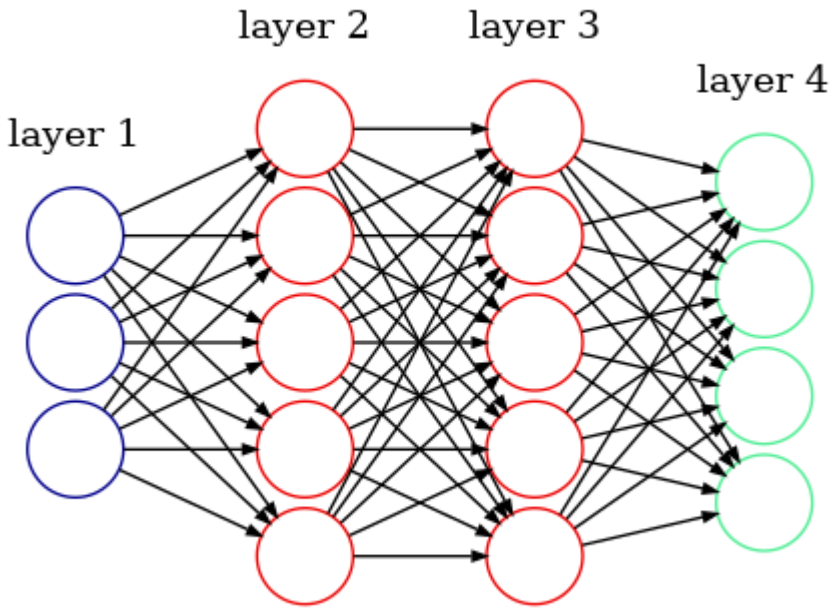


Figure 31 Example of Neural Network the IRA Analogy

In the same way most of AI frameworks (like PyTorch or TensorFlow) interoperate, the IRA proposes an interoperability schema that leverages the best practices and technologies to create blocks that can be linked to transform data end-to-end from one block to the other.

Actually some of the solutions considered in the State-of-The Art section [Section 3], have similar interoperability issues. Most of them are specific and do not solve generic class of problems. Interoperability is often a matter of configuring the specifically the parameters involved in the inter-process communication. INFINITECH RA wants to go further proposing a reference **Interoperability Schema**.

In the development/deployment view, each component is a service (technically a microservice) and can be considered as a black box that operates on data consuming data from one end and producing data to the other end. Therefore the RA suggest to pipeline the building blocks, albeit there can be possible many different flows from blocks in any arbitrary connection.

However, when it comes to connect blocks together, there are still some questions the Infinitech Reference Architecture leaves open or implementation dependent:

1. How do the components communicate with each other?
2. How components understand data they exchange?

There can be a **Reference Way** to answer to 1 and 2?

The more standardization of the layered building blocks the more powerful and flexible the implementation will be. It is worth to propose a standard interoperability model into the RA and to suggest to implement the building blocks accordingly. The following sections propose the INFINITECH WAY to interoperate.

4.10.1 Communication between the components

In principle, the IRA does not define any relationship among the blocks: any component can be connected to any other component. Since the IRA does not define a specific data flow, a data scientist in principle can connect components to design a solution for a specific use case.

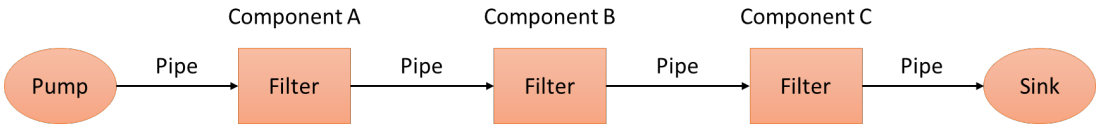


Figure 32 Lego Concept

What does it mean to draw a line or an arrow between two components?

In the INFINITECH reference implementation the common way is to use REST API among microservices. Two options could be considered:

1. Component specific API
2. Standard API

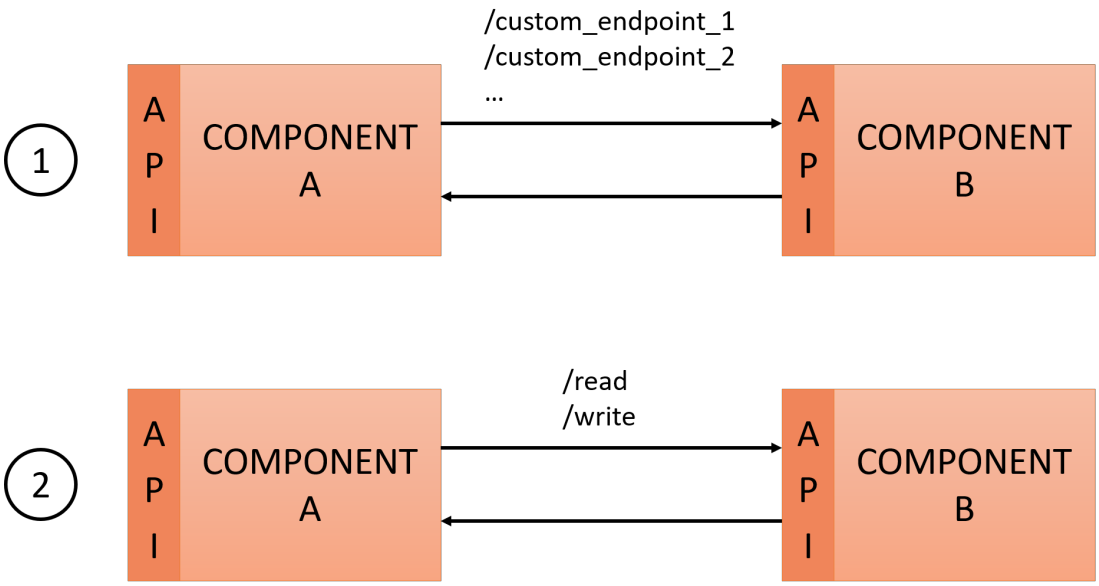


Figure 33 Component API

For 1. to work the two components MUST know the other endpoints' syntax and semantic of the data exchanged. This makes 1. very difficult to generalize and allows only limited pipelines compositions.

On the other hand, a predefined set of standard endpoints combined with Ontology Server feature of abstracting data schemas makes a component reusable and plug and play to connect with another block with the same characteristics. With this approach, any component knows the endpoint exposed by all the other components through their API. As explained in Section 3.3, the pipeline architectures

are inspired by the Unix technique of connecting the output of an application to the input of another one via pipes on the shell. Taking inspiration from the Unix cat command, which reads from standard input and writes in standard output, two basic endpoints can be defined: /read and /write. Therefore if any block would exhibit standard /read and /write endpoints (standard from the point of view of REST API syntax) the connection between the components would be straightforward.

However, the component-to-component is not the only possible type of communication. In fact, for example, it is possible that two components communicate through a third standard component like a data-store or database, as B and C in the figure below. In particular, component C consumes the data inserted by B in a SQL database.

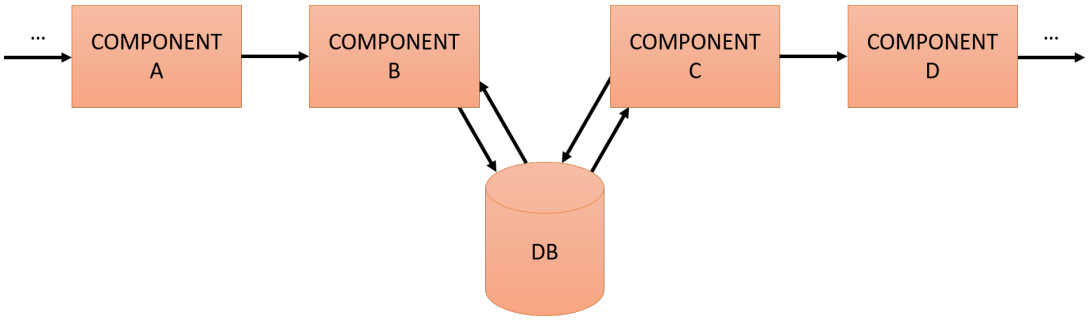


Figure 34 Communication through DB

From the IRA point of view, a database is just another component, a black box like any other building blocks. Then, the architecture in the previous figure can be seen as two distinct pipelines, as shown in the figure below.

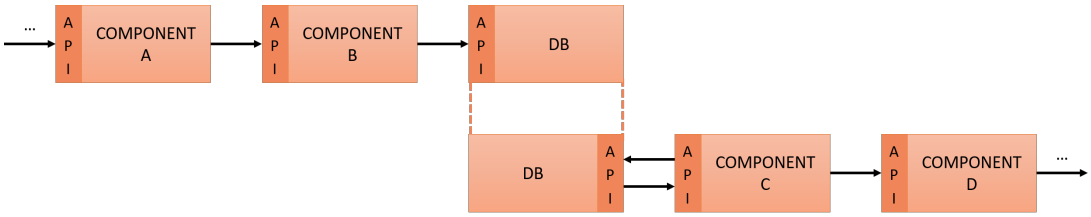


Figure 35 Decomposition in two pipelines

The first pipeline includes components A and B and the database. Component B writes in the database, which is the sink of the pipeline. The second pipeline is composed by the database, component C, and component D. In this case, the database is the pump of the pipeline.

There is still an issue to address: the representation used in the following figure is ambiguous. In fact, the arrow between Component A and Component B indicates that the two blocks interact, however they both expose the /read and /write endpoint. Thus they could communicate in two different ways:

1. Component A could send some data to Component B through Component B /write endpoint, or
2. Component B could retrieve some data from Component A through Component A /read endpoint.

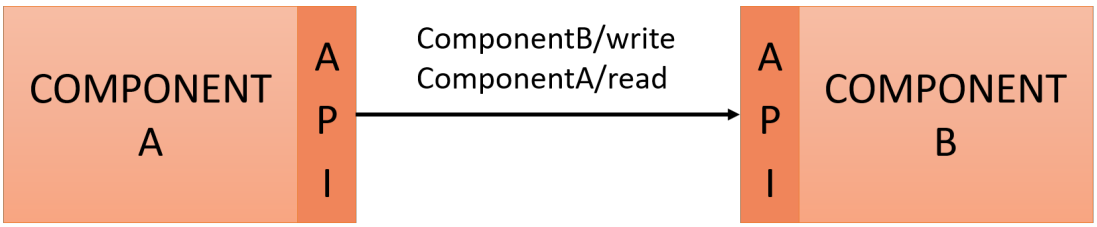


Figure 36 Ambiguity of endpoints

In the figure above the sole arrow does not enable to distinguish these two types of communication. It is then necessary to find a representation that makes the type of communication explicit. To this aim, we introduce two types of arrows:

1. The normal arrow: it indicates that a /write endpoint is used;
2. The tailed arrow (having a circle in its tail): it indicates that a /read endpoint is used.

The figure below shows how the two arrows are used in a simple pipeline which represents the functioning of cat command. In this example, the block BETA retrieves some data from ALPHA through the ALPHA/read endpoint. This is represented through a tailed arrow that starts from ALPHA and points to BETA. Then, BETA elaborates the input and provides its output to GAMMA through the GAMMA/write endpoint. This is represented by a normal arrow which starts from BETA and points to GAMMA.

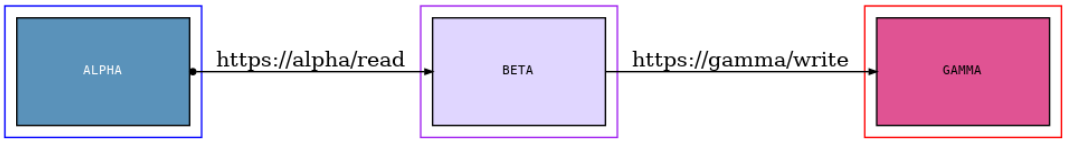


Figure 37 Cat Pipeline

Instead, the following figure shows a classic pipeline, in which ALPHA pushes data to BETA through the BETA/write endpoint and BETA pushes its output to GAMMA through the GAMMA/write endpoint. In both cases, the normal arrow explicits that the /write endpoint is used.

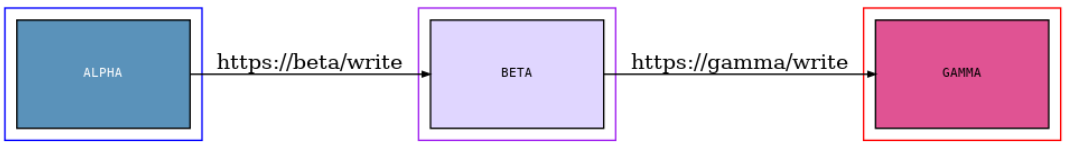


Figure 38 Classic Pipeline

For the sake of completeness, the following figure depicts all the possible interactions between two components, considering the type of endpoint used as well as the direction (from left block to right block and viceversa).

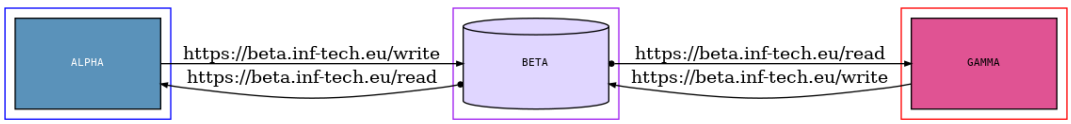


Figure 39 Possible types of interaction between components

5 Addressing Stakeholders' requirements

The INFINITECH-RA provides the means for addressing the identified requirements and challenges of BigData and AI applications in the target sectors. This is illustrated in the following tables where the RA is confronted to the earlier listed requirements and challenges. Note however that several of these requirements are not addressed by the RA per se, but rather based on some of the INFINITECH components and technologies such as the components presented in D2.5. The latter provide the technical instruments that substantiate the high level concepts of the RA in concrete solutions.

Table 3 INFINITECH-RA vs Key Requirements for BigData and AI Applications in Digital Finance

Requirements	INFINITECH RA
Siloed Data and Business Operations	The INFINITECH-RA is versatile in supporting data collection and ingestion from diverse sources (e.g., databases, data lakes). Furthermore, it offers a rich set of data pre-processing and data preparation functionalities, range from simple join of relational/tabular data to semantic interoperability functionalities.
Real Time Performance Requirements	Based on the INFINITECH-RA and the project's data management technologies it is possible to support integrated data management (ingestion, access) pipelines that improve data access performance. Additional performance gains are obtained based on the use of INFINITECH analytics technologies like incremental analytics.
Mobility and Multiple Channels	INFINITECH-RA decouples the visualization and application development parts of an application from the data management parts. As such it facilitates the implementation and delivery of multi-channel, mobile first applications.
Automation	INFINITECH-RA Automates the development, deployment, and operation of BigData and AI applications in Digital Finance based on the use of piplelines. The latter eliminate manual steps and process, through enable seamless flow of data from the sources to the applications. INFINITECH technologies have been structured as pipelines' building blocks to facilitate this approach to automation. The pipelines of the INFINITECH-RA provide a sound basis for further automation based on the use of AutoML frameworks (e.g., MLBOX, AutoSklearn or TPOT)
Transparency	Pipelines provide transparency on the data processing. They include explainable rule-based algorithms (e.g., random forests) that can be used when transparency at the level of ML algorithms is required. Likewise, INFINITECH-RA allows for the implementation of transparent and explainable AI techniques.
Security and Regulatory Compliance	The INFINITECH-RA specifies dedicated layers where security and regulatory compliance functions can be placed. This is for example the case with the anonymization functionalities that have been exemplified earlier. Furthermore, a group of INFINTIECH Security and Privacy Components and technologies is available for use in pipelines with stringent security and data protection requirements.
Ethical and Responsible AI	The above-listed transparency, security and privacy friendliness features boost the project's ability to support Ethical and Responsible AI applications, such as applications that are secure, trustworthy and transparent in terms of their AI operations.

Table 4 INFINITECH-RA vs Key Challenges for BigData and AI Applications in Digital Finance

Challenge	INFINITECH RA
Limited Integration	The INFINITECH RA and components attack integration challenges from multiple complementary perspectives, including: (i) Integration of data and ML operations through workflows and pipelines; (ii) Integrated data modelling, including semantics; (iii) Continuous integration in-line with a DevOps and MLOps paradigm.
Poor Stakeholders Collaboration in Financial Services	The INFINITECH RA specifies a module/approach for decentralized and transparent movements of data, which is a foundation for more efficient collaboration between stakeholders of the financial services supply chain. Specifically, the blockchain and information sharing components that support the RA will boost the collaboration of stakeholders in the Financial Services supply chain.
Compliance with Stringent Regulatory Requirements	The INFINITECH-RA pipelines are built based on components that support security, data protection and regulatory compliance requirements. Relevant provisions are made in the layers of the INFINITECH-RA. Furthermore, INFINITECH specifies the security and data privacy groups of components that facilitate compliance at the technical level.
Lack of Flexibility	INFINITECH RA leverages industry best practices for the development of flexible data driven pipelines based on DevOps/MLOps approaches. It leverages microservices and other design principles that boost automated testing and continuous integration. It will be also augmented with constructs that will facilitate its evolution to MLOPs that enable continuous updates of Machine Learning and AI related pipelines. Also, INFINITECH technologies provide flexibility at the levels of data ingestion and data sources integration, including support for relevant ontologies.

6 Pilots' Reference Designs and Initial Alignment to the INFINITECH-RA

The RA is grounded by the Pilots workflows as references. The following sections present the mapping of the Pilots' Use Cases in the logical view and components identification of the INFINITECH Reference Architecture. For every pilot a basic description of the objectives, sources and workflow along with the expected results and interfaces are described. With respect to D2.13, the following sections are updated with the improvements achieved between month 10 and month 19.

The Reference Designs presented are not intended to be exhaustive and complete and in most cases do not present the different phases of the deployments, like the training and the run phase of the Machine Learning algorithms. They are intended to demonstrate the viability and versatility of the INFINITECH-RA approach. Hence, the mappings of the pilot systems to the INFINITECH-RA concepts and components are at this stage by no means exhaustive and complete. The last release of the INFINITECH-RA (i.e. deliverables D2.15) will provide more complete mappings of the pilot systems, both in terms of the mapping of different viewpoints and in terms of the use of more INFINITECH components and technologies.

6.1 Pilot#1: Invoices Processing Platform for a more Sustainable Banking Industry

6.1.1 Pilot Objectives

The main objective of pilot #1 is to develop, integrate and deploy a data-intensive system to extract information from notary invoices, in order to:

- Establish the sustainability index of each notary based on the number of physical copies that are issued.
- Provide financial institutions the information (properly indexed) about the documents that are finally generated by notarial services required by the bank.
- Promote notarial services from those with the higher sustainability score.

NOTE: the Pilot#1 is suspended as BANKIA, which was the financial organization leader of the Pilot is no more a partner of the Consortium, due to a merge-acquisition to another financial group. The concept of the Pilots and the preliminary study and designs remains valid and will be as a reference to similar use cases.

6.1.2 Data Sources

Data from 32.300 real invoices documents and from 3.000 different notaries extracted from Bankia systems are the source of the Pilot. Invoice documents to be digitalized in PDF format or may also arrive already digitalized from other channels (email attachments, bulk sftp, etc.). Data type will be: PDF/Image/Text. Data format will be: PDF/ PNG/ TXT. Estimated data volume will be: 2 TB. The dataset TableBank, which consists of 500.000 documents, will be used as Table Benchmark for Image based Table Detection and Recognition.

6.1.3 Data Produced

Digitization of contracting and invoicing processes will allow an automated analysis of the digitized documents enabling a smart and autonomous scoring of notary services. Rating notaries based on a "Sustainability Index Score" will provide a new criterion to be applied when contracting these services impacting positively in the short and long-term in the amount of paper used and the economic fees applied.

6.1.4 Explainable Workflow

Invoice documents will be securely storage in a data lake. The system will parallelize different jobs to pre-process, process and post-process the documents and the outcomes. For instance: Image pre-processing (cropping, adjusting brightness, contrast, etc.); converting PDF to Text; OCR; text correction. A computer Vision system will identify and extract tables from invoices that will allow extracting sensible information to establish a sustainability scoring. And using machine learning we will extract information from the identified and extracted tables.

The extracted information will be displayed so it can be validated and re-introduced to the system. The AI models will be trained (offline process) with a combination of public huge datasets and specific invoices samples. Trained models will be published to the runtime processing time after an expert evaluation.

6.1.5 Logical Schema

The following figure illustrates the logical architecture of the pilot in-line with INFINITECH-RA constructs and approach.

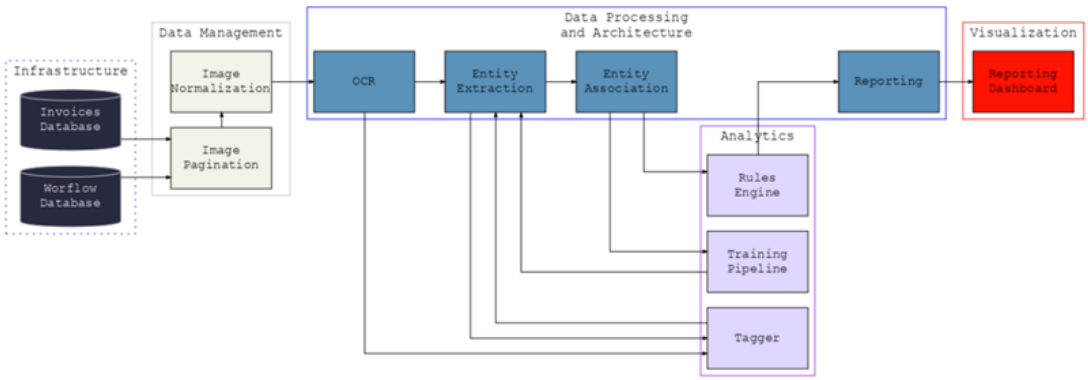


Figure 40 Invoices Processing pilot pipeline in-line with the IRA

6.1.6 Components

The main technological components that will be implemented and integrated as part of this pilot are:

- Invoices and invoicing workflow database.
- Document ingestion.
- Document pre-processing: document pagination, PDF to image conversion, image normalization, OCR.
- Document entities and region-of-interest extraction: machine learning models and Natural Language Processing extractors for the identification and extraction of entities of interest: billable concepts, prices, headers, addresses, etc.
- Entity association: graph deep neural networks for the identification of related concepts: e.g. that a certain billable concept corresponds with an identified price and identified.
- Business rules engine: application of compliance business rules for the generation of alerts and reports
- Data Tagger: for the tagging of training invoices examples
- Document validator: for the verification of processed invoices
- Training and inference orchestrated pipelines.
- MLOps tools: Models and data Repository, code repository.
- Reporting business dashboards and operational databases

6.2 Pilot#2: Real-time risk assessment in Investment Banking

6.2.1 Pilot Objectives

The Pilot#2 implements a real-time risk assessment and monitoring procedure for two standard risk metrics: VaR (Value-at-Risk) and ES (Expected Shortfall). The main outcome is the measurement of market risks of assets portfolios. In addition, the pilot will evaluate what-if scenarios allowing pre-trade analysis, i.e. estimating changes in risk measures before a new trading position is entered. Moreover, the pilot will implement a sentiment-based decision support indicator derived from financial and economic news data and social media channels. While VaR and ES are quantitative risk measures based on numerical price data, the market sentiment will be derived from financial and economic news data and social media channels.

6.2.2 Data Sources

Data will be extracted from several data sources: real-time market data, historical market data, synthetic electronic order platform (trades data), and financial news/article data. The pilot will leverage FOREX (FX) data provided by the JRC Platform and other trading platforms via Forex APIs. The data will include:

- Trade Data (i.e. data with the assets' positions) of the user that will be used to define the portfolio(s) of the user and their VaR/ES;
- Tick Data (i.e. Historical market data) that will be used in the different methods for VaR calculations including standard methods such as Monte Carlo simulations, Variance-Covariance, Historical Simulation and a novel one based on deep neural networks, the so-called DeepVaR.
- Alternative data (e.g., data from news feed) that will be used for market sentiment analysis based on NLP (Natural Language Processing Techniques). Such data will be obtained from Open API's (e.g., Google News API, Twitter API and Interactive Brokers API).

Trade Data and Tick Data will contain information such as: the name of the instrument in FOREX trading (ex. GBPUSD for the exchange of GBP to USD), Timestamp that denotes when the trading took place, the Quantity and the Closing Price.

6.2.3 Data Produced

The main data computed and produced include the VaR (Value-at-Risk) and ES (Expected Shortfall) estimations. In addition, the injected real time data are both processed and saved as historical market data as it is (ticker data) and processed (i.e., aggregated market data in frequencies of 1 min, 5 min, 1 hour,1 day). Moreover, the pilot's sentiment-based decision support indicator derived from financial and economic news data and social media channels will produce a sentiment score (positive, neutral, negative) for each article/description coming from the news feed.

6.2.4 Explainable Workflow

Data from the real-time market database and the news feed databases is injected into the Data Management layer through a stream processing component which is capable of handling large volumes of data that feature very high ingestion. The real-time data is initially concatenated with the historical data and then is appropriately transformed using a data windows component (i.e., the Online Aggregates Component), creating segments of time series. Data from the electronic order platform are managed using a data extractor. These data will also serve as input for both the correlation matrix and the scenario specifications components. The processed market data (historical and real-time) will then feed the correlation matrix component together with the processed data from the electronic order platform database. The correlation matrix processes and calculates the ingested data, merging the different data sources. The output will then serve as input, together with

the scenario specifications component, for the scenario generation, the basis for the Monte Carlo simulation. The processed data will then go into the Analytics component where VaR/ES estimation takes place.

On the other side, data from the news article database are processed using the text extraction component and then market sentiment extraction one. Therefore, sentiment and behavioural analysis will be performed, serving as well as input for the Analytics component.

The analytics component will perform calculations on the data from the above-described flows and from the inputs of the configurator. The latter involves interaction of the user, in order to configure specifications for the scenario generation.

The results are depicted in the User Interface which is responsible not only to visualize the VaR/ES predictions but also to perform pre-trade analysis leveraging the developed risk assessment models.

6.2.5 Logical Schema

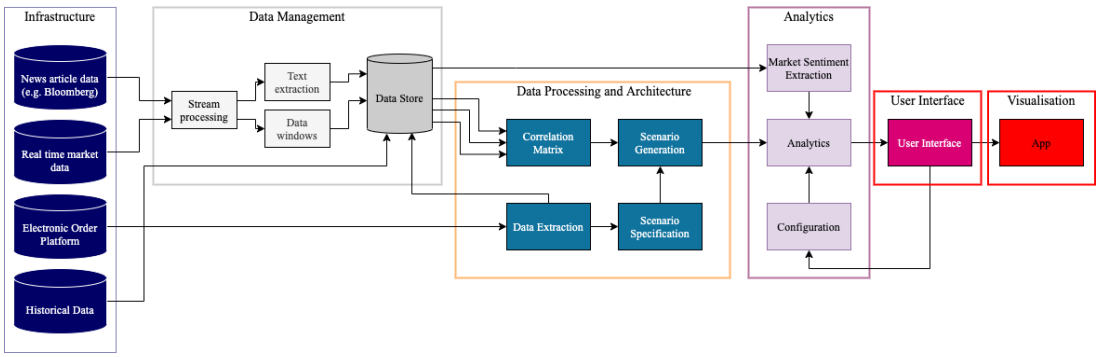


Figure 41 Real-Time Risk Assessment pilot pipeline in-line with the IRA

6.2.6 Components

The workflow leverages the following components:

- BigData Management Layer i.e. INFINISTORE and Online Aggregates (Data Management in RA)
- Custom Injection Simulator (Data Ingestion in RA)
- Kafka (namespace Cross Cutting in RA)
- Zookeeper (Cross Cutting in RA)
- AI model for VaR prediction (Analytics and Machine Learning in RA)
- UI Risk Assessment based on VaR (Interface in RA)
- Sentiment Analysis for financial news (Analytics and Machine Learning in RA)

6.3 Pilot#3: Collaborative Customer-centric Data Analytics for Financial Services

6.3.1 Pilot Objectives

This pilot would examine how banks and fintech(s) in collaboration with research organisations and NGOs can develop an AI driven capability using transactional data generated by the financial activities that identifies money-related profiles based on the transactional data generated. Data profiles e.g. from social media then can be associated to human profiles base on their financial activity. These profiles will be built into the available AI engine and will be combined with existing technology and data sourced from the TAH human trafficking platform. The results will produce a complete picture of people profile, people trafficking routes and the corresponding money flows back to the criminal organizations.

This pilot will utilize a combination of open-banking, social and internal-bank generated data sources to establish a high-volume and high-quality view of the customer to be used for a range of data analytics performed on big data platforms. The use of analytical methods could include link analysis to support permission-based customer relationship analytics on behalf of customer and bank, or transaction monitoring to support credit risk management for bank, but also that provide value for customers.

The Pilot#3 will need to simulating a data sharing ecosystem by mimicing participants in that ecosystem and provide rules of engagement and highlighting the value exchanges between participants. A digital ecosystem framework is described here to articulate testbed components required.

6.3.2 Data Sources

Pilot #3 will consider two sources of data:

- Operational Data Sources – We will not using existing BOI Ops. Data sources, because of confidentiality issues even if anonymized and also data consistency issues. Instead Proof of concept data sources are 'synthetic' customer, account and transactions data designed to mimic real world data scenarios from financial services.
- Captured data from data entry in application including consent or metadata exhaust from sharing process.

6.3.3 Data Produced

For Pilot 3 a better representation of the data lifecycle might be as follows:

1. Data utilized/transferred (E.g. data sharing payload – Customer/KYC, Account or Transaction Data),
2. Data transformation (e.g. any data changes) ,
3. Data produced (e.g. new data) &
4. Data deletion (e.g. revoked consent) etc.

6.3.4 Explainable Workflow

The whole premise of the pilot purposed is to enable unlimited use cases between any participants via a single application creating a single ecosystem. Specific back end data services might be built to support a particular use case e.g. KYC. Below is a data flow of KYC use case in terms of business process/data flow and technical data flow.

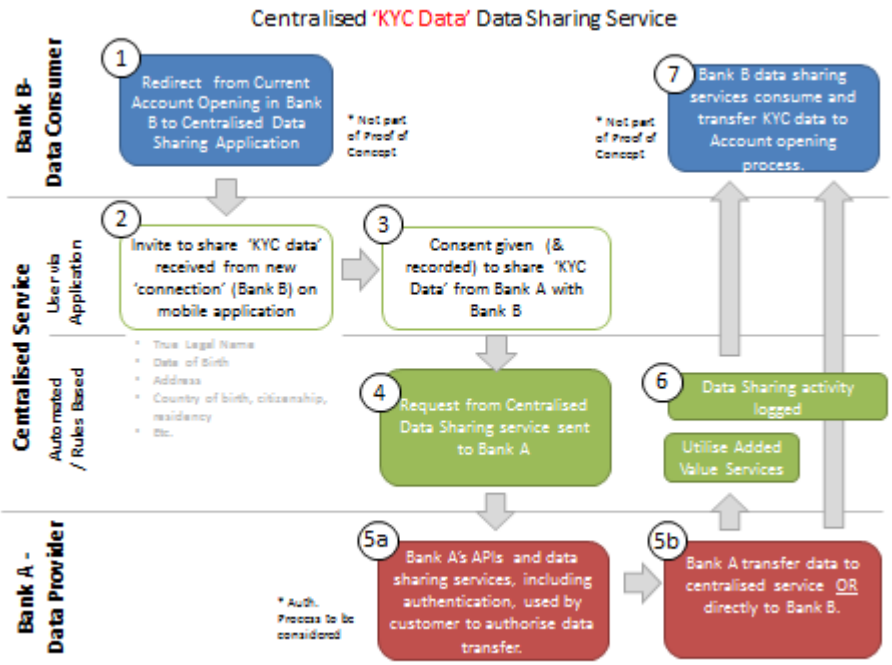


Figure 42 Customer-Centric Data Analytics pilot workflow – KYC Data Sharing Process – Business Workflow

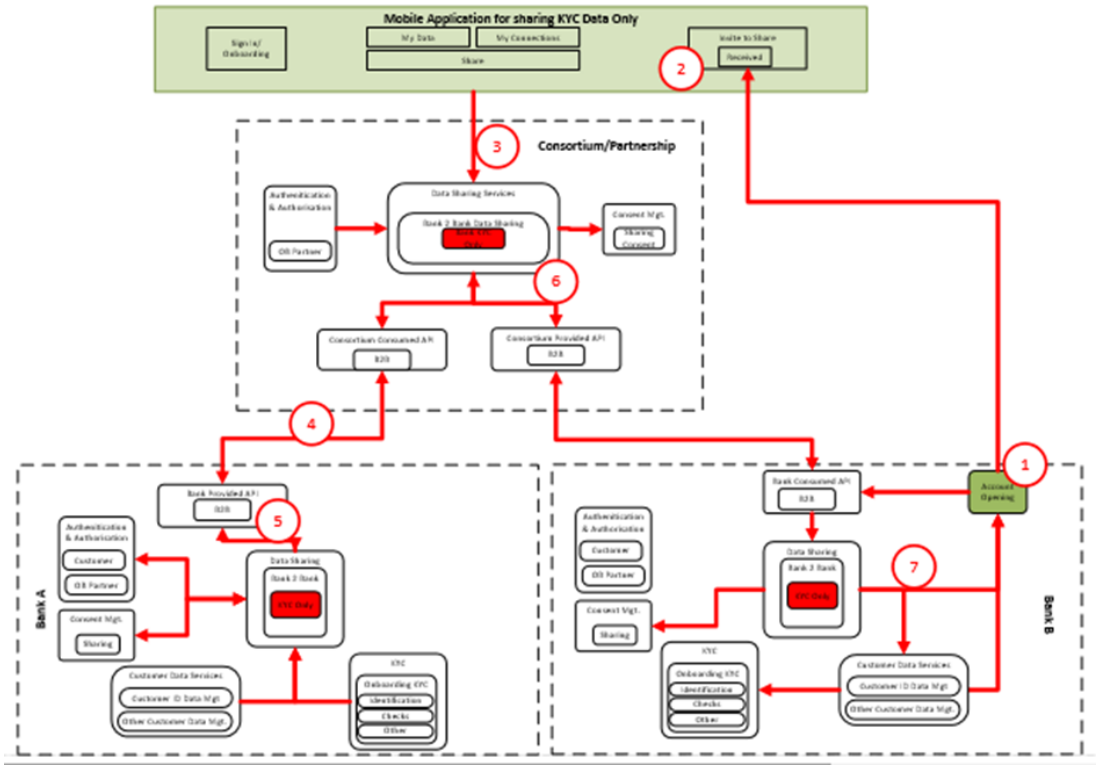


Figure 43 Customer-Centric Data Analytics pilot workflow – KYC Data Sharing Process – Technical Workflow

6.3.5 Logical Schema

The following figure refactors the components of the above-listed workflows towards illustrating the pilot logical architecture in-line with the INFINITECH-RA.

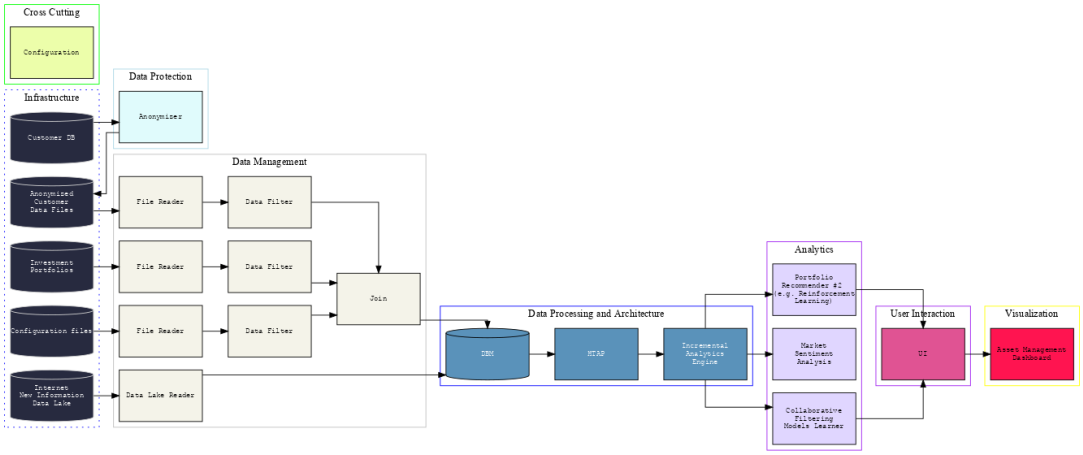


Figure 44 Customer-Centric Data Analytics pilot pipeline in-line with the IRA

6.4 Pilot#4: Personalized Portfolio Management ("Why Private Banking cannot be for everyone?")

6.4.1 Pilot Objectives

The main goal is to develop and adapt within SaaS based Privé Managers Wealth Management Platform a Portfolio Optimization algorithm (further on called Privé Optimizer or "AIGO"), as well as improving and expanding its capabilities as an artificial intelligence engine to support better investment propositions for retail clients.

This pilot will explore the possibilities of AI-Based Portfolio construction for Wealth Management processes, regardless of the amount to be invested (therefore the slogan "Private Banking could be for everyone"). The AI-Based Portfolio Construction will enable advisors and/or end-customers, to use the existing Wealth Management Platform "Prive Managers" and make use of its risk-profiling and investment proposal capabilities, starting from his/her personal risk-awareness. AIGO allows for a variety of use cases which cater to the needs of financial advisors, end-clients and financial services companies. The innovative AIGO genetic algorithm can be used for proposing investments and evaluating them given an easy-to-use, personalizable set of criteria, in the form of so-called fitness factors. These fitness factors will be used to generate "health" scores for portfolios, which are used to define the "fittest" investments.

6.4.2 Data Sources

The data to be used by this pilot will be:

- Customer Transactions Data: customer securities and cash transactions through their deposit accounts. They are fetched directly from the Bank or an Asset Manager;
- Financial Market Price Data: price data for Stocks, Bonds, Mutual Funds and or other assets like certificates/warrants. They are fetched from several Market Data Providers;
- Financial Market Asset Master Data: asset related characteristics (e.g. expiration date, minimum investment amount, asset class breakdowns). They are fetched from several Market Data Providers;
- Customer Risk Profile Data: customer Risk Profile Data through their account data and profiling, based on B2B customers parameters. They are fetched directly from the Bank or an Asset Manager;
- Mutual Fund, ETF and Structured Products Breakdown: asset breakdowns based on bank data or market data providers breakdown. They are fetched from several Market Data Providers;

- Customer Economic Outlook: they are fetched directly from the Bank or an Asset Manager based on questionnaires and Customer Profiles;
- Single Account & Investors Data: 19484 accounts for about 15400 investors (live data) 94.407 different securities available; Investors serviced by 309 different advisor companies; Accounts in 28 different custodian banks (Data from 2019). All datasets will be stored within Privé SaaS solution in a cloud setup. Asset data and Client data are fetched from 3rd party databases and partially from selected market-data providers. Risk metrics are calculated in the historical backtesting component for each single portfolio. A Genetic Algorithm component evaluates different Fitness Factors and generates a customised portfolio proposal.

6.4.3 Data Produced

JSON files will be produced from Privé API (if other 3rd party solutions address to this Portfolio Optimisation functionality, and PDF files can be generated for UI display and customer documentation.

The output data consists of the single portfolio holdings, their weights and amounts to decide about the Proposed Portfolio. Fitness Factors Scores and Total Fitness Score will be output for both the current and proposed (optimised) portfolio. For both Portfolios also Risk and Return metrics will be shown: 5 year annualized return, volatility and sharpe ratio.

6.4.4 Explainable Workflow

Starting from a client's cash pool or current investments/portfolios, a risk profile is created or an existing one is updated (Steps 1 to 3 on Figure 44). Then the user will select the fitness factors and constraints or preferences to perform the portfolio construction, based on the client's risk profile and preferences (Step 4). The optimisation tool will run on a pre-set universe of assets taking into account all the input data and constraints (Steps 5 to 7). The AI genetic algorithm will generate a new proposal, where the selected preferences and risk parameters have been recognised (Step 8 and 10). The optimisation tool can be run multiple times, after the necessary changes in initial parameters are made, based on that the proposed portfolio is satisfactory or not (Step 9). This process can result in a UI proposal or a PDF generated investment proposal.

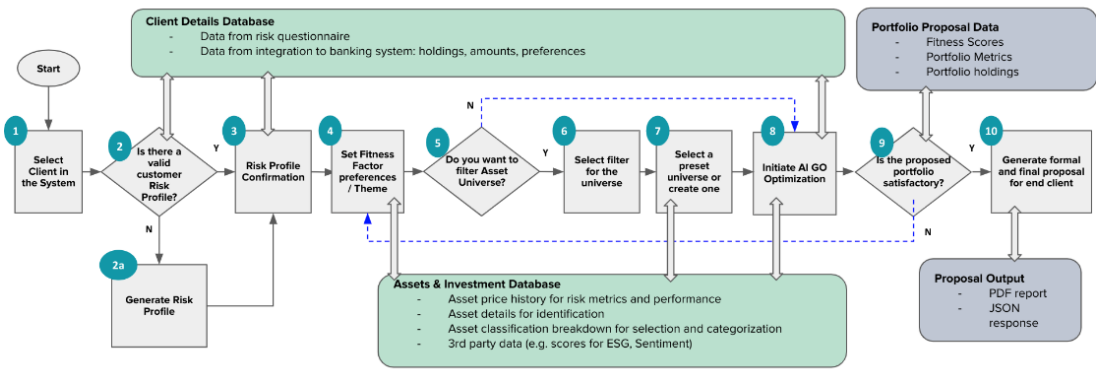


Figure 45 Personalized Portfolio Management pilot workflow

6.4.5 Logical Schema

An initial mapping of the explainable workflow of the pilot to the INFINITECH-RA layers and constructs is depicted in the following figure.

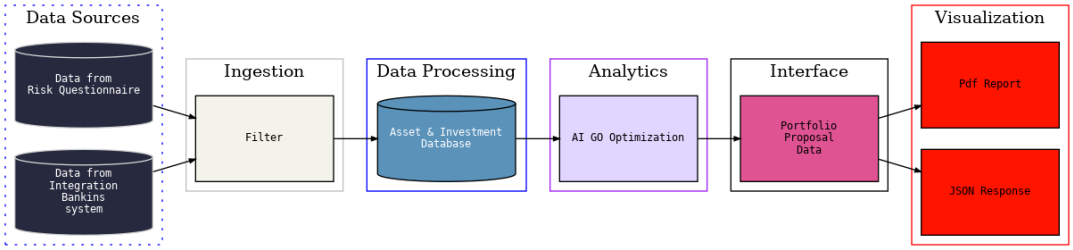


Figure 46 Personalized Portfolio Management pilot pipeline in-line with the IRA

6.4.6 Data Components

This section links the software components with the corresponding Reference Architecture layers, providing some details about their implementation. In this sense:

- Data Collection (Data Management layer of the RA) based on customers cash pool or current investments/portfolios data
- Customers' & investments/portfolio Data quality check (Data Processing layer in RA): according to specific data models, in order to perform the data preparation for portfolio construction, based on the client's risk profile and preferences.
- AI Based Portfolio Optimization Process (AIGO, Analytics layer in the RA) that will be developed from the Pilot based on AI Algorithm, will run on a pre-set universe of assets taking into account all the input data and constraints, generating a new proposal, where the selected preferences and risk parameters for a specific customer.
- New proposal for the personalized portfolio will be visualized through a PDF report generation or a JSON extract that will be able to be imported in any relevant portfolio management tool.

6.5 Pilot #5b: Business Financial Management (BFM) tools delivering a Smart Business Advise

6.5.1 Pilot Objectives

Most of today's Financial Management tools for Small Medium Enterprises (SMEs) are geared towards analysing only past transactions, making such tools inadequate in today's world. Today, SMEs and their customers alike demand just-in-time processing, transparency and personalized services to assist SME owners not only in understanding better their SME business/financial health but also to be able to decide on the next best action to take. Thus, Pilot#5b aims to assist SME clients of Bank of Cyprus (BOC) in managing their financial health in the areas of cash flow management, continuous spending/cost analysis, budgeting, revenue review and VAT provisioning, all by providing a set of AI powered Business Financial Management tools and harnessing available data to generate personalized business insights and recommendations. Machine learning algorithms, predictive analytics and AI-based interfaces will be utilized to develop a kind of smart virtual advisor with the aim to minimize SME business admin effort, to focus on growth opportunities and to optimize cash flows performance.

6.5.2 Data Sources

The following data sources will be integrated and used in the pilot:

- Transaction Data from BOC: a .csv file with around 500MB and 3.5 millions of transactions between 2018 and 2019;
- Transaction Data from Open Banking (i.e. PSD2 data)
- Transaction Data from SMEs (optional)

- Other Data (Market)
- Other Data from SMEs (optional)
- Accounts Data from BOC: maps accounts with the transactions;
- Accounts Data from Open Banking
- Customer Data from BOC: links customer to accounts and the available NACE code is used in the transactions' categorization model;
- Direct Input from SMEs (e.g. feedback loop for transaction categorization)

6.5.3 Data Produced

The pilot will combine the abovementioned diverse datasets in order to produce personalized business insights and recommendations for SME customers of BOC. Output data, as shown below, will be generated by the various engines in relation to cash flow predictions, budgeting, KPIs, benchmark(s) and transaction monitoring and categorization. The data will be stored in the common datastore and be available to the end user (SME) via the Infinitech Reference Architecture (IRA) gateway (and the banks middleware). To this end a pilots-specific REST API will be developed leveraging different endpoints for each specific service. The output data/endpoints include:

- A JSON containing the obtained insights and recommendation to be provided to the respective SME
- A JSON containing the obtained cash flow related data to be provided to SME directly or indirectly
- A JSON containing the derived budget target for each category used by the respective SME
- A JSON containing results on Financial Health and Performance matrix
- A JSON containing results on abnormal transactions and suspicious expenses
- A JSON containing Matrix with invoice information and payment prioritization
- A JSON containing benchmarks that allow the SME to compare to likewise businesses.

6.5.4 Explainable Workflow

Some of the available datasets require real time data collection, while in others historical data collection is sufficient to provide actionable business insights. In detail, transaction and account data related to the respective SME will be drawn from BOC's repository by a real time/historical data collector as well as transaction and account data from Open Banking (PSD2), as well as BOC customer data, will utilize a historical data collector. Furthermore, a way of handling batch of data is needed to provide as there should be an option of pushing data to the Infinistore once a day by the bank (e.g. in cases where the real-time connection is lost or for the purpose of uploading history data). To this end, the bank IT team will be capable of uploading a batch of data in CSV format directly to the pilot specific cloud sandbox. IIn addition, an external data collector will also be used in order to integrate other related Open Banking/macroeconomic data. The SMEs data source (e.g. ERP/Accounting system) utilization remains optional as consent is required for the collection and processing of such data and its cloud availability being required. All data except external macroeconomic data will be pseudoanonymized (by tokenization) before being uploaded to the IRA. The cloud Data Repository (within IRA) will then store all collected data, along with the generated insights, past SME financial actions (to measure at what degree the SME actions reflect the recommended insights), as well as minimum user input which is required. A continuous data streaming will connect the Data Repository with the various deployed BFM tools (machine learning algorithms), which would allow the retraining of the respective AI models and the generation of useful insights and recommended actions. A reverse data pseudoanonymization will then be applied before the processed data move to the bank middleware component that contains composite APIs and produces push notifications, all which will be offered to the SMEs via Android, iOS and web apps. Upon SME user login the IRA is also accessed, insights/recommendations picked up from the cloud data repository and provided to the SME user. To this end, a prototype component will be developed in order to digest and properly present the results of the corresponding analytics components. The pilot's workflow is depicted in the figure below.

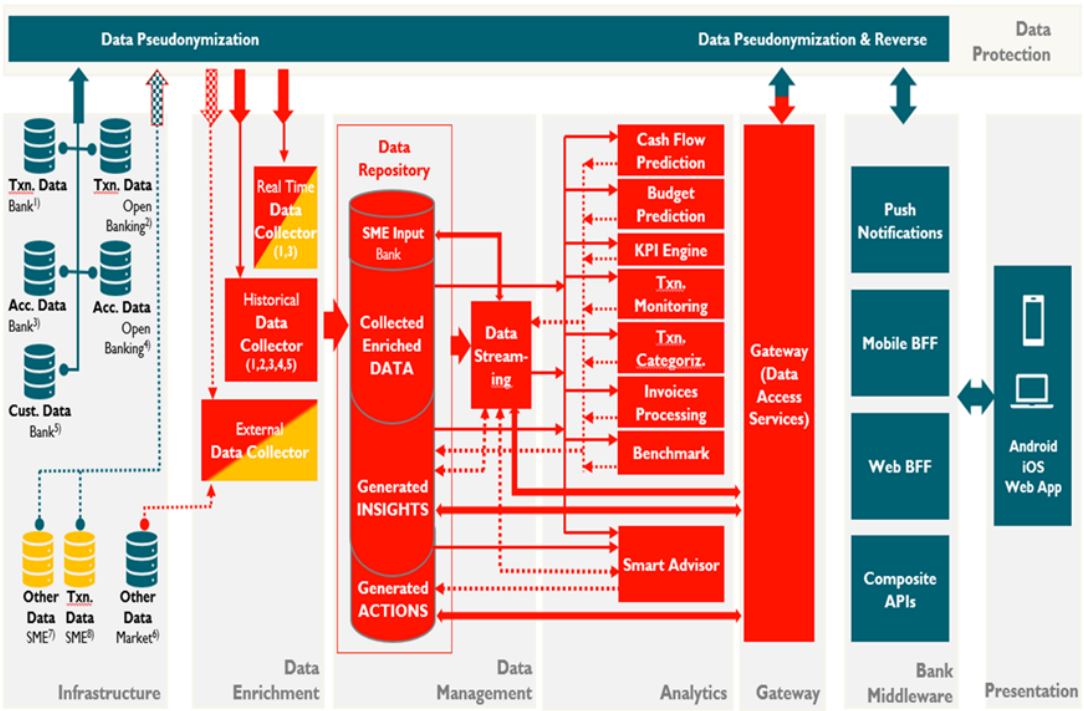


Figure 47 Business Financial Management pilot workflow

6.5.5 Logical Schema

The following figure illustrates a logical view of the pilot system architecture in-line with the INFINITECH-RA.

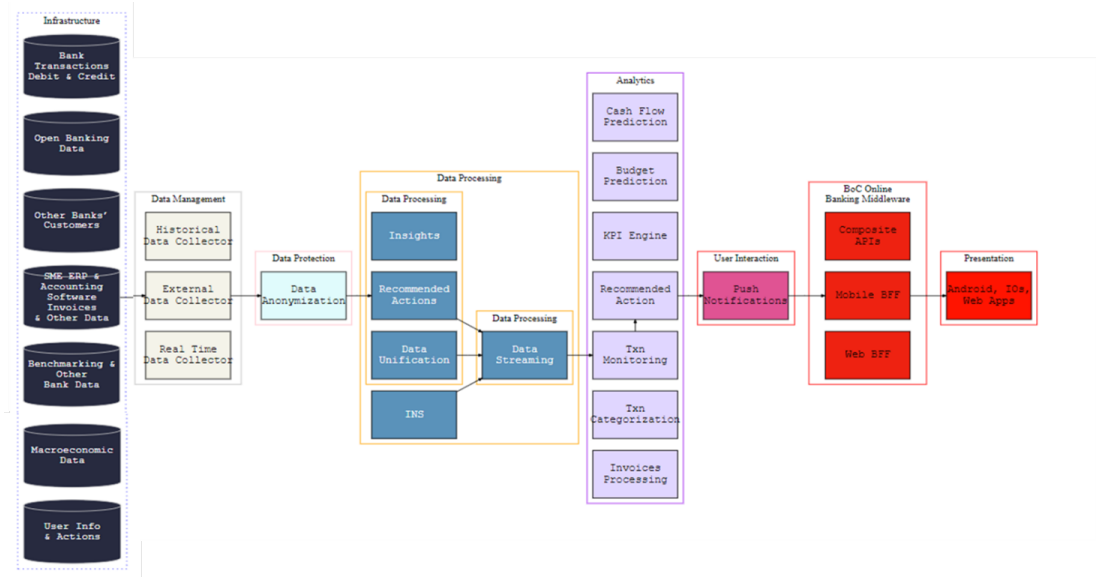


Figure 48 Business Financial Management pilot pipeline in-line with the IRA

6.5.6 Components

The following components will be deployed and used in the pilot pipelines:

- Transaction Categorization Engine (Analytics layer in the RA): key component in charge of labelling the transactions of selected SME customers of Bank of Cyprus into 20 main categories (with around 80 respective subcategories to be implemented soon);;
- Cash Flow Prediction component (Analytics layer in the RA): based on a propabilistic Deep Neural Network (implementation of DeepAR model) to predict the expenses of certain categories of a given account in a time horizon of 12 weeks;
- Budget Prediction engine (Analytics layer in the RA): allows setting easily budget targets through the provision of suggested target values as well as simple budget monitoring;
- KPI engine (Analytics layer in the RA): leading to valuable insights on the SME financial health and performance;
- Transaction monitoring engine (Analytics layer in the RA): watches out for potential anomalies and savings; To this end Graph analysis apporaches is being explored and implemented;
- Invoice Processing engine (Analytics layer in the RA): generates meaningful invoice background info to other components (e.g. Cash Flow Prediction) and SMEs. This applies if respective data can be obtained from SME relative ERP system;
- Benchmark engine (Analytics layer in the RA): supporting comparisons to other SMEs with similar profiles;
- Smart Advisor (Analytics layer in the RA): generating actionable insights for a SME that will allow to perform better.

6.6 Pilot#6: Personalized Closed-Loop Investment Portfolio Management for Retail Customers

6.6.1 Pilot Objectives

The goal of Pilot#6 is to create a system for personalized investment recommendations for the retail customers of the bank. NBG will leverage large customer datasets and large volumes of customer-related alternative data sources (e.g., social media, news feeds, on-line information) in order to make the process of providing investment recommendations to retail customer more targeted, automated, effective, as well as context-aware (i.e. tailored to state of the market).

6.6.2 Data Sources

Data that will be used for this pilot will be extracted and anonymized in CSV files from NBG Datawarehouse and several data sources:

- Deposit Account Transactions: Data of Deposits accounts transactions for retail customers are extracted for the last two (2) years (8,91G),
- Cards Transactions: Data of Transactions related to Cards for retail customers for the last two (2) years (7,3GB).
- Instruments Historical Prices: Data for Instruments Historical Prices for the last two (2) years (0,23GB).
- Investment Related Transactions: Data of Investment Related Transactions for last two(2) years (0,3GB).
- Instruments Characteristics: Data for Instrument characteristics for matching with customers profiles, including asset class, currency, ISIN, maturity etc. (0,01GB).
- CRM Data: 150.000 Customers related data like demographics, product ownership and responses to MIFID questionnaires (0,05GB) .
- Sentiment Analysis for each instrument proposed from Data Analysis as recommendation using RB information from the news or/and social media to provide to NBG customers with clearer and real-time risk results.

6.6.3 Data Produced

Personalized investment recommendations for the retail customers of the bank, based on their Risk and transactions profiles. Banks relationship managers based on each customer risk and transactions profile, will be able to propose the possible alternatives of financial instruments that a customer will be interested to invest, with the relative prioritization. The proposed recommendations will be based on the instruments available from the bank, with the necessary input data for sentiment analysis for each financial instrument, based on the news & social feed, for the specific instrument (e.g. stock, bond, etc), or the relative instrument category.

Existing Landscape in Financial Institutions and particularly Banks has set as priority the identification of targeted Customer propositions and especially in investments sector. Driven both by Competition as well as Customer needs, depiction of each Customers potential and risk appetite in combination with interesting for the Customer recommendations, may lead in the increase of each Customer's share of wallet and at the end increase of Bank's Market share in the specific Sector.

6.6.4 Explainable Workflow

Data from NBG Datawarehouse related to Investment Products Retail Clients (CRM Data, Deposit Account Transactions, Cards Transactions, Investment Related Transactions,) will be extracted in CSV files and utilizing the relative tools for data processing, anonymization and quality checking and cleansing will be imported to Leanxcale Datastore. Based on the data extracted for NBG Clients, through the Customer Risk Profile engine using data analysis tools, will be able to divide all customers in specific profile clusters based on the investment & banking behaviour (MIFID questionnaires), deposit and card transactions, as well as investments transactions. Also, NBG will provide for each investment customer cluster profile the relative instruments that will be suitable for investment.

6.6.5 Logical Schema

A first approach to mapping the pilot architecture to the INFINITECH-RA layers and pipelines approaches is illustrated in the following figure.

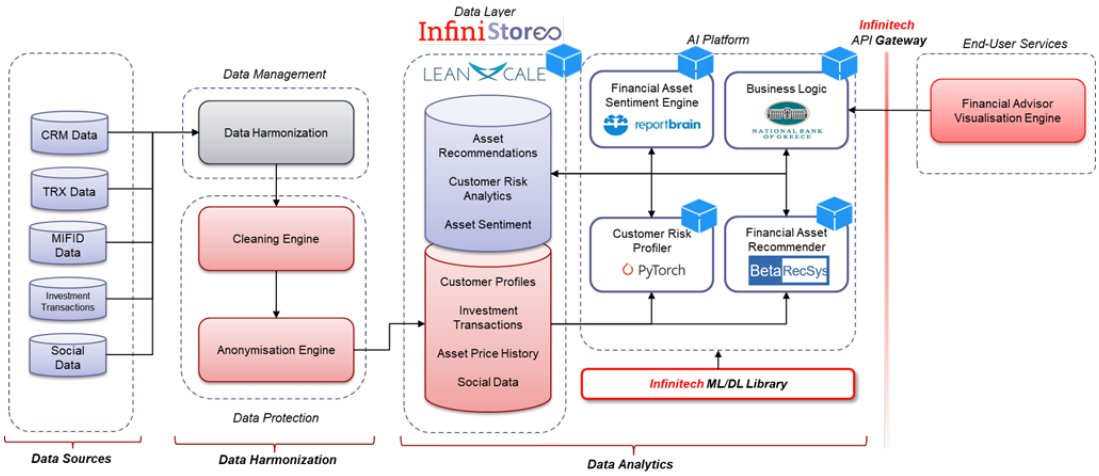


Figure 49 Personalized Closed-Loop Investment Portfolio Management pilot pipeline in-line with the IRA

6.6.6 Components

The following components will be deployed and used in the pilot:

- DataStore (Leanxcale) (Data Sources in RA)
- NBG Datasets (Data Sources in RA)
- Data Collection (UBI Icarus) (Data Management in RA)
- Data Normalization (UBI Icarus) (Security inRA)
- Customer Risk Profile Cluster (Analytics in RA): classify customers into 4 risk profiles: Conservative, Income Seeking, Balanced, Growth Seeking
- Personalized Investment Recommendation AI engine (Analytics in RA)
- Customer initiation and personalized recommendation UI Application (Presentation in RA)

6.7 Pilot #7: Operation Whitetail – Avoiding Financial Crime

6.7.1 Pilot Objectives

Pilot 7 ("Operation Whitetail - Avoiding Financial Crime") is managed by CXB. Due to a change in pilot partners, it is in an early stage of development. The goal of Pilot 7 is to explore more accurate, comprehensive and near real-time pictures of suspicious behavior in Financial Crime, Fraud, in the use case of instant loans. Such loans can be requested online and are subject to fraud and crime, e.g. identity theft. Based on comprehensive data including KYC and transaction data a financial crime risk score is calculated by AI/ML algorithms. This way the instant loan can be approved or denied related to this score.

6.7.2 Data Sources

The pilot will use synthetic or anonymized data as source. In the bank internal data pool sources will be accessed. This data pool also includes bank internal and external KYC data and internal transactional data. These data shall be joined in an advanced KYC data source and the relevant data for the use case will be extracted from that Due to compliance rules, these data need to be treated confidential. In a 1st step use related data representing customer profiles will be extracted facilitating the development of synthesized data sets giving insight to the financial crime risk score and facilitating the development of AI/ML models.

6.7.3 Data Produced

The pilot will produce data giving insight to the financial crime, i.e. instant loan, risk score. This may include a risk score, customer data, transaction patterns and details. The detailed data, which will be presented, are yet to be specified depending on the advice of Financial Crime experts in the bank.

6.7.4 Explainable Workflow

Within the pilot the following processes are addressed:

- KYC (Know Your Customer), for screening the available data sources in near-real time, to ensure that KYC data is automatically updated to the most recent information available on the customer facilitating data quality.
- Customer risk profiling, based on feeding the transaction-based customer's behavioural profile data and KYC results leading to an advanced risk score that could provide a holistic customer risk profile and will enable the business to respond quicker to newly identified risk and changes in criminal behaviour.

6.7.5 Logical Schema

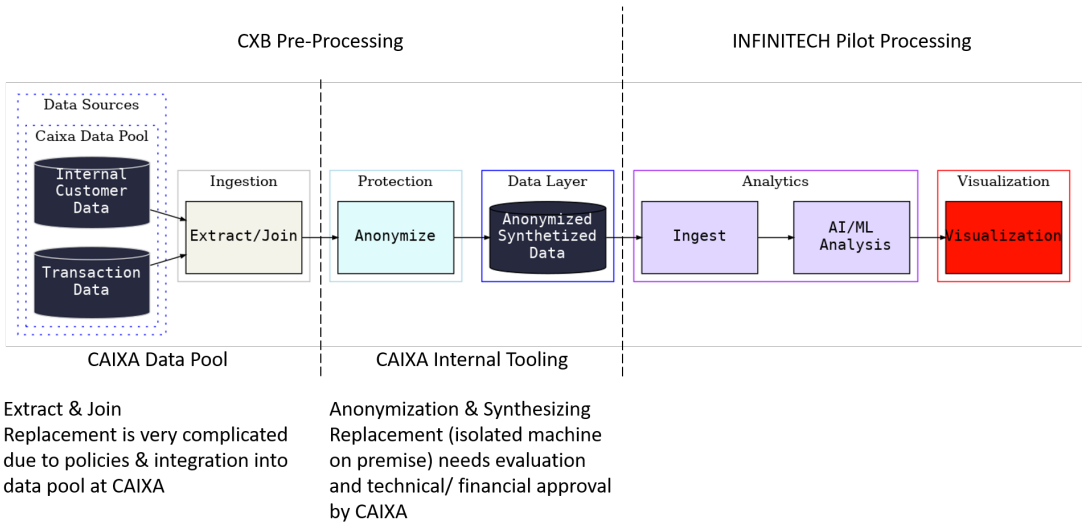


Figure 50 Financial Crime pilot pipeline in-line with the IRA

6.7.6 Components

Due to strict compliance and approval procedures in CXB the pilot operations are facilitated splitting the tasks in a pre- and pilot processing part. The pre-processing part may be mimicked by INFINTECH tools based on beforehand synthesized/anonymized data. However, for a smooth progress of the pilot development, a bank internal and an INFINITECH process will be considered as a first step.

A List of the main components to be deployed and used in the pilot follows:

Pre-processing - Inside the bank by bank approved tools:

- Bank Data Pool (Data Sources in the RA)
- Bank Data Pool Extraction (Data Management in the RA)
- Bank Data Pool Join (Data Management in the RA)
- Data synthesation / anonymization (Data Source in the RA)

Pilot-Processing - The synthesized / anonymized data then are used in the INFINITECH Pilot

- Synthesized / anonymized data (Data Source in the RA)

- Data Ingestion (Ingestion in the RA)
- Data Analytics / Scoring (Analytics in the RA)
- Visualization (Presentation in the RA)

6.8 Pilot#8: Platform for Anti Money Laundering Supervision (PAMLS)

6.8.1 Pilot Objectives

The objective of the Pilot, is to develop a Platform for anti-money laundering Supervision (PAMLS), which will improve the effectiveness of the existing supervisory activities in the area of anti-money laundering and combating terrorist financing (AML/CTF) by processing large quantity of data (Big Data) owned by the Bank of Slovenia (BOS) and other competent authorities (FIU).

PAMLS will contain the following functionalities:

- Risk assessment tool: to assess the money laundering and terrorist financing (ML/FT) risks of financial institutions (FIs) and the risk of a whole sector to support risk based supervision,
- Screening tool: for screening payment transactions, enriched with data from business register (ePRS) and transactions accounts register (eRTR), to recognize unusual patterns that could indicate typologies and risks of ML/FT at level of individual FI or the whole sector,
- Search engine: allowing supervisor to look for a specific transaction or a sample of transactions,
- Distribution channel: for secure gathering data that will feed risk assessment tool and screening tool..

6.8.2 Data Sources

Relevant datasets, planned to be analyzed within PAMLS are:

- TARGET2 transactions:
  - transactions executed by the Slovenian payment institutions within TARGET2 (Trans-European Automated Real-time Gross Settlement Express Transfer System)
  - high value (above 50.000 EUR), urgent transactions in EUR
  - transactions processed through BOS payment systems (responsible BOS Payment Settlement and Systems department - PPS) Confidential data.
- SEPA transactions:
  - transactions executed by the Slovenian payment institutions within SEPA (Single Euro Payments Area)
  - domestic and international transactions within SEPA area in EUR under 50.000EUR value
  - transactions processed through payment systems by third party provider Confidential data.
- FIU transactions (public data):
  - transactions related to high risk countries above 15.000 EUR reported to the Slovene Financial Intelligence Unit (FIU) Public data.
- FI identification data
  - identification information about Financial Institution (FI)
  - aggregated statistical data on the FI inherent risk and control environment (number of clients, number of Suspicious transactions reports (STR) etc.)
  - FI reports to the BOS (reports are confidential) Confidential data.
- ePRS data
  - Slovenian Business Register (public data on legal entities) Public data.

- eRTR data
  - Slovenian Transactions Accounts Register (public data on legal entities) Public data.
- High risk country list
  - List of countries defined as high risk due to lack of or not effective AML/CTF system
  - List is managed and published by the Slovene FIU (public data) Public data. Personal data will be anonymized by the source, prior data delivery to PAMLS.

6.8.3 Data Produced

Ongoing risk assessment for the purpose of the Anti-Money Laundering and Combating Terrorist Financing Supervision over the FI and FI sector.

6.8.4 Explainable Workflow

PAMLS will use various data sources, which we can divide in to three groups:

- The first group consists of transactional data process through payment service providers in Slovenia (TARGET and SEPA transactions). This group will first be enriched with ePRS and eRTR data, and then pseudo-anonymized (for end user anonymized). Before data will be stored in PAMLS internal data storage it will also be joined with High risk country List.
- The second set of data sources represents public data (FIU transactions) that will also be enriched with ePRS and eRTR data and than joined with High risk country List and stored in PAMLS internal data storage.
- Third group of data sources represents FI data (data on FI inherent risk and control environment), which will stored in PAMLS internal data storage after positive Data Quality Check.

After data is ingested in PAMLS platform, it needs to be preprocessed in a way, that information is properly enriched and it needs to be provided in a suitable data format (vectors, graphs). Process of feature engineering, tailored to specific goals, will follow. PAMLS will develop and test novel approaches for detecting unusual patterns of ML/TF, which could be labelled as high risk later in the process and will have an effect on final FI risk assessment. Part of the PAMLS is also Risk Calculation engine. There the risk calculation will be continuously calculated on a level of a sector or a particular FI, using predefined Risk Assessment methodology.

To empower bank analysis, to develop and test novel approaches, PAMLS provides three components: Stream story, Pattern discovery & matching, Anomaly detection & prediction. With introduction of enriched graph topologies and a hierarchical Markov chain models, PAMLS will capture the qualitative behaviour of the systems' dynamics and enable analyst to discover new regularities and correlations on a larger scale.

These components will enable iterative development of potentially new upgrades to existing Risk Assessment methodology and discovery of novel or additional money laundering and terrorist financing typologies. PAMLS will also provide three different user interfaces, which corresponds to 3 different use cases.

6.8.5 Logical Schema

The following figure illustrates an initial logical mapping of the pilot components to the layers and pipelines approach of the INFINITECH-RA.

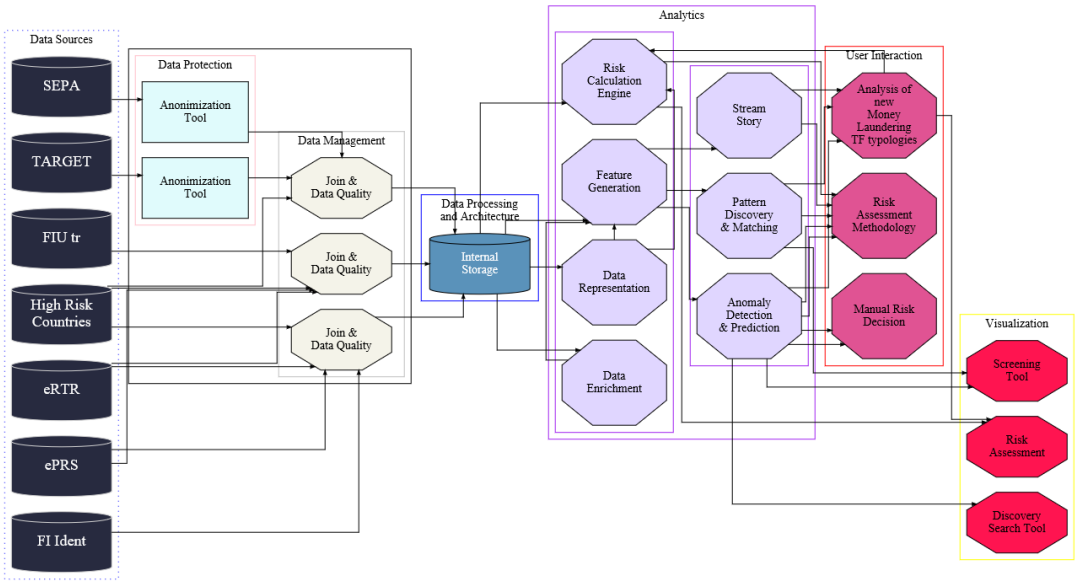


Figure 51 PAMLS pilot pipeline in-line with the IRA

6.8.6 Components

The pilot will use the following components:

- Risk Calculation engine and Complex search services (Analytics in the RA)
- Anomaly detection and prediction component (Analytics in the RA): will provide functionalities for anomaly detection and prediction for time series data including Pattern analysis. The latter will provide analytical services on data graphs, including detection of complex patterns on data graphs;
- StreamStory component (Analytics in the RA): a component for the analysis of multivariate time series. It computes and visualizes a hierarchical Markov chain model which captures the qualitative behaviour of the systems' dynamics, where system is described with a group of time series;
- Pattern discovery and matching component (Analytics in the RA)
- Pseudo-anonimization tool (Data Managenent in the RA)
- PostgreSQL (Data Managenent in the RA)
- ElasticSearch (Data Managenent in the RA)
- NEO4J (Presentation in the RA)

6.9 Pilot #9: Analyzing Blockchain Transaction Graphs for Fraudulent Activities

6.9.1 Pilot Objectives

There can be blockchain crypto currencies and tokenized assets (e.g. USD, EUR, TRY tokens) that are obtained fraudulently as a result of ransomware and theft of funds. These fraudulent assets can go through various transfers on the blockchain and enter the regulated environments in different jurisdictions. As a result, it is possible that a company may accept deposits of crypto currencies and tokens that can be traced to addresses involved in fraudulent activities. Pilot #9 is developing a parallel and scalable transaction graph analysis system that can construct and operate on the massive Bitcoin and Ethereum blockchain transaction graph with distributed dynamic data structures on an HPC cluster. During Period 1 of the project, the pilot has implemented parallel graph algorithm based

fraudulent activity analysis. In the Period 2 of the project, it has also initiated implementation of machine learning based analysis algorithms. The pilot is also providing a user interface that provides various queries and visualization of results using graph drawing package.

6.9.2 Data Sources

Pilot #9 will use the following data sources:

- Public Bitcoin Blockchain Data (BOUN)
  - Bitcoin transfers (send transactions);
- Public Ethereum Blockchain Data (BOUN)
  - – Ether transfers (send transactions) and ERC20 Token Smart contract transactions (major popular tokens including stable coins like EURS, GUSD, USDT, TRYB, PAX,TUSD, QCAD, XAUT)
- Bitcoin and Ethereum Addresses Database (AKTIF)
  - Database of all Bitcoin (within block ranges 0-674999) and Ethereum addresses (within block ranges 0-10199999) are maintained as a database with capability to label each address with features.
  - Blacklisted Bitcoin and Ethereum blockchain addresses that are obtained from the Internet by manual search for published hacked/fraudulent accounts and addresses involved in ransomware activities.

6.9.3 Data Produced

Pilot #9 can generate the following data:

- Extracted Ethereum and major ERC20 token transaction data that is also made available at https://zenodo.org/record/4718440#.YXkLhtZBw1I. It can be downloaded by researchers and businesses;
- Paths and subgraphs that show tracing of blockchain addresses to blacklisted addresses;
- Importance values of addresses computed by running parallel Pagerank algorithm is produced as data. This rank data can be used as an important feature in the machine learning algorithms.

6.9.4 Explainable Workflow

Pilot #9's Blockchain Transaction Dataset Preparation Component parses raw blockchain data and extracts Bitcoin, Ethereum and major ERC20 token transactions (such as Gemini USD (GUSD), Tether USD (USDT), Tether Gold (XAUT), Statis Euro (EURS) and Turkish BiLira (TRYB)) that come from the Bitcoin and Ethereum Mainnet blockchains. After retrieving all the blocks up until now, this component is run periodically to retrieve newly generated blockchain blocks during the period.

Scalable Transaction Graph Analysis Component of the pilot takes the full bitcoin and Ethereum public transaction dataset. Graph traversal algorithms are used to analyze the data. Parallel graph traversals are used to extract features that are in the form of subgraphs. Since the transaction graph size is massive and dynamically growing, it constructs distributed and partitioned transaction graph in parallel using MPI message passing libraries in order to achieve scalability. Graph analysis service is interacted through a message queue that takes commands in YAML format. The outputs of the service are in the form of graph paths or subgraphs that show tracing of Blockchain addresses to blacklisted addresses. In the second period of the project, machine learning algorithms have been started to be developed. Bitcoin and Ethereum transaction data and blacklisted address lists as well as pageranks that are computed in parallel are used in machine learning algorithms.

Finally, the User Interface for Blockchain Transaction Reports and Visualization functional service interacts with the Scalable Transaction Graph Analysis and presents results in a web browser. When subgraphs are returned that trace customer addresses to blacklisted addresses, these subgraphs are output in vis.vj graph visualization software format for viewing in browsers. The business service is

provided through a RabbitMQ message queue that takes commands in the YAML format. Visualization of transaction graph traces as well as a simple address score based on shortest path from blacklisted addresses is also provided.

6.9.5 Logical Schema

The following figure illustrates how the pilot architecture can be expressed in terms of the layers and the pipelines approach of the INFINITECH-RA.

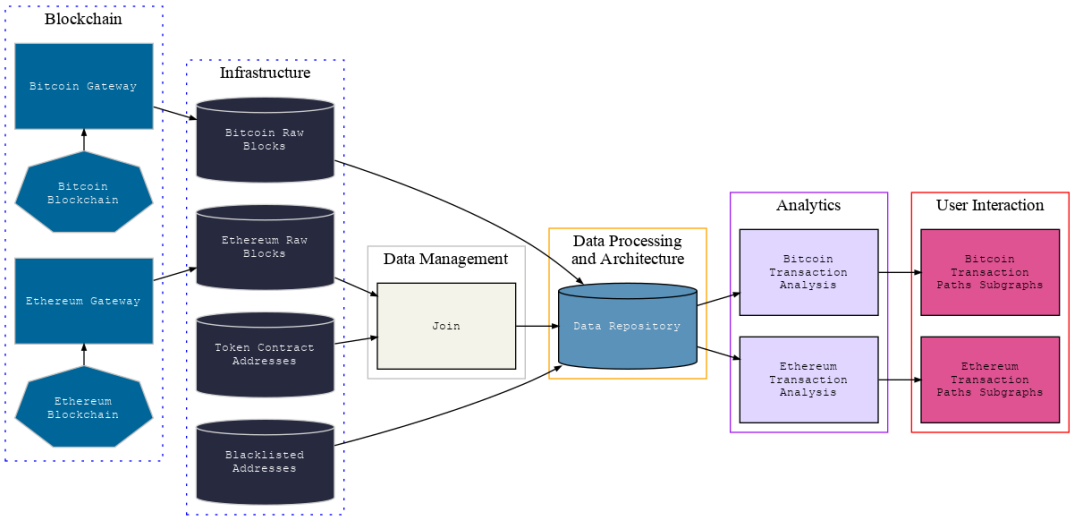


Figure 52 Blockchain Transaction Graphs Analysis pilot pipeline in-line with the IRA

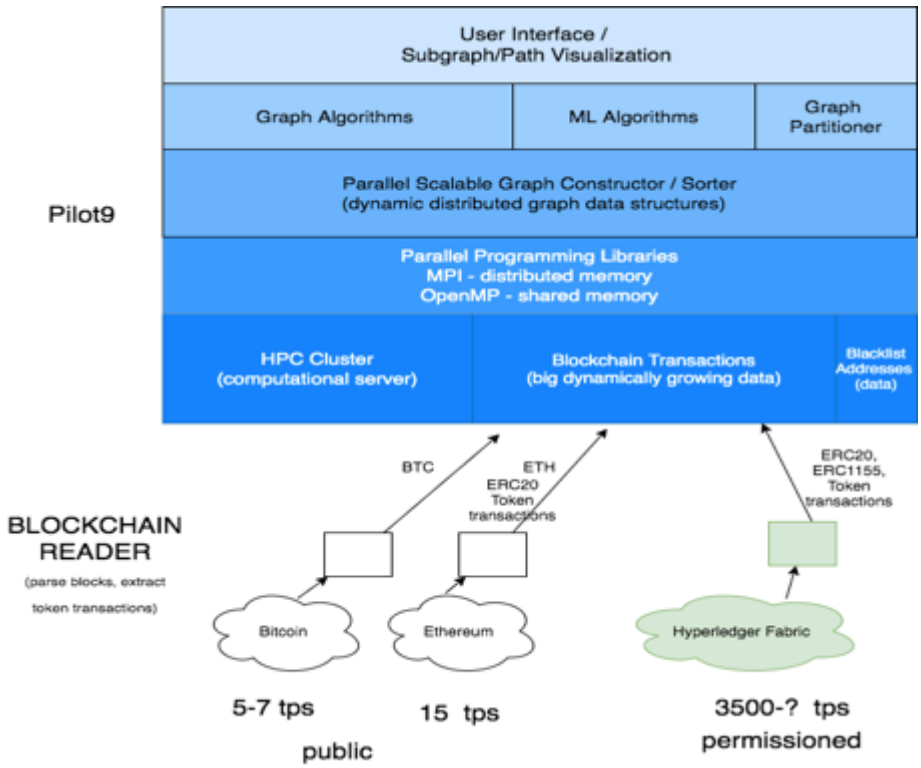


Figure 53 Layered architecture of the scalable blockchain transaction graph analysis system

6.9.6 Data Components

The following components will be developed, deployed and used in the pilot:

- Blockchain Transaction Dataset Preparation Component (Data ingestion in RA)
- Scalable Transaction Graph Analysis Component (Data Management and Analytics in RA)

- User Interface for Blockchain Transaction Reports and Visualization Component (Interface and Analytics in RA). A database of bitcoin and ethereum addresses as well as blacklisted addresses is also managed by this component.

6.10 Pilot #10: Real-time cybersecurity analytics on Financial Transactions' BigData

6.10.1 Pilot Objectives

In Pilot#10 a fraud detection system is proposed to meet two goals:

- The early detection of new and subtle types of frauds. Since fraudsters keep innovating novel ways to scam people and online systems, it becomes crucial to apply AI/ML methods to detect outliers in large transactional datasets and be robust to changing patterns.
- The reduction of the number of false positives which are usually analyzed to understand if they are real fraud attempts or not. To this aim, it is very important to be able to train, validate and test ML models to make the most accurate ones operational.

6.10.2 Data Sources

The data sets in input of the batch workflow are related to several types of transactions: - Bank Transfer SEPA (The Single Euro Payments Area (SEPA): a payment-integration initiative of the European Union for simplification of bank transfers denominated in euro. SEPA covers predominantly normal bank transfers.

A data generator, implemented by ENG, will simulate real-time transactions (SEPA) which includes informations about the emission date, the beneficiary and the orderer accounts, the amount, the IP address of the orderer's connection and its location (futher informations on Explainable Worflow section). These data will be collected and stored on a dataset to later retrain machine learning models batch-wise; at the same time they are analyzed at real-time for fraud detection with previously trained models.

6.10.3 Data Produced

The data produced consist of a list of suspected fraudolent transactions with an associated probability of actually being frauds exstimated based on the models used.

6.10.4 Explainable Workflow

To meet the abovementioned goals, Pilot#10 envisages two layers (batch and stream layers) implementing the following ML pipelines:

- Unsupervised training (batch) of an outilier detection model (Isolation Forest) on all the collected data.
- Supervised training (batch) of a classifier on the data labeled by the domain expert user.
- Real-time detection (stream) of ouliers: which consists of both data preparation services and the application of the Isolation Forest model.
- Real-time detection (stream) of fraudolent transactions using the supervisely trained model.

For the first training the goal is to try to identify the outliers using the collected data over time and a method called Isolation Forest, an unsupervised teqnique that identifies anomalies isolating points in a n-dimensional space using binary trees. These points are not necessarily fraudolent transactions, but assuming that the illegitimate ones are a very small percentage of the whole dataset, it is likely that they are as well outliers; therefore it is important to collect outliers and make them available to the domain expert for further analisys. While analising them, he will also label the data, distinguishing between true positive and false positive fraudolent transactions.

The second training consists of the generation of a supervised classifier model. Since the domain expert labels a portion of the data at real-time, and those are collected in batches, we can exploit the work done so far in order to offer a second exstimation of the probability of the transaction being illegal or not. This second exstimation is going to help the system in filtering what transactions the domain expert must analyze and what are the ones he can ignore, reducing his work but at the same time tring not to reduce the reliability of the fraud detection mechanism.

At the same time a real-time analysis is needed. Before the real-time detection the data pass through a process of cleaning and filtering in order to create new features that will be more useful in the predictive model or to enhance other features, improving model performance. We are supposing that it will be needed to analyze datasets made of mixed-type data, where numeric and nominal features coexist. These data must be then elaborated: e.g., instead of dates, time intervals might be more interesting; instead of user names or IP addresses, their location might be more useful during the model's training.

The real-time detection of outliers consists on the application of the unsupervised model described before; at the same time the supervised model is used for inference and a second estimation of the probability that the transaction's data belong to a fraud is produced.

The results of both predictions are analized by the domain expert which can take action and block the transaction in time, while at the same time all the data produced are collected, including the labels generated. These data are used on the next cycle of batch tranings, improving models performance over time.

6.10.5 Logical Schema

An initial mapping of the pilot's components and modules to the INFINITECH-RA pipelines approach is illustrated in the following figure.

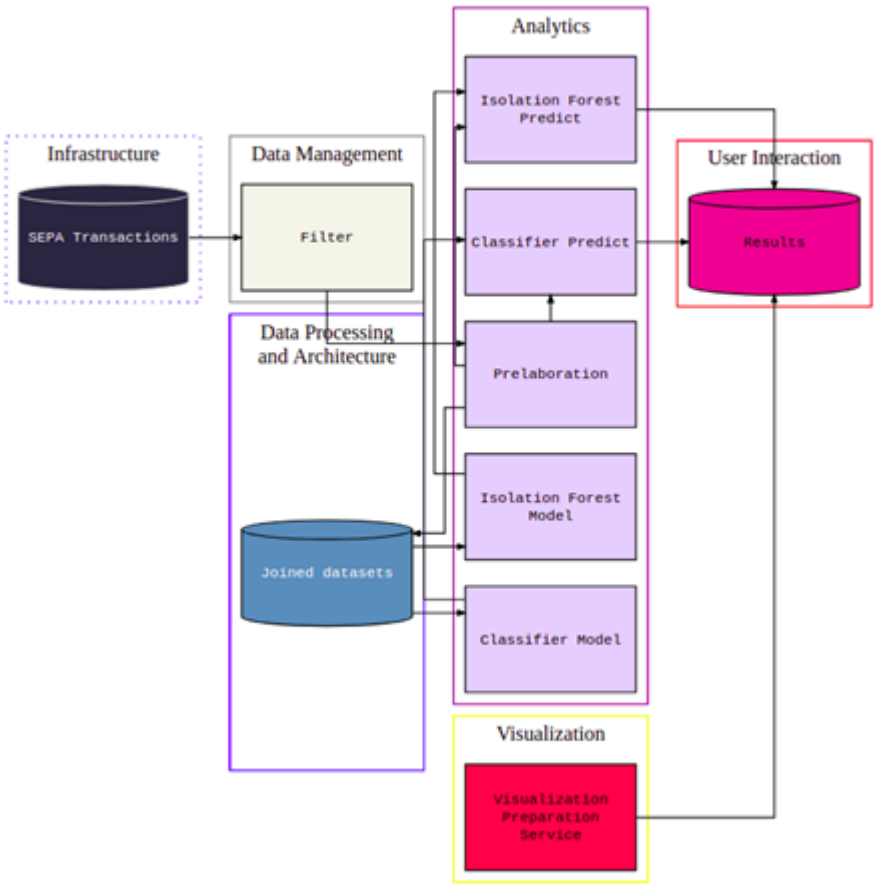


Figure 54 Real-time cybersecurity analytics pilots pipeline in-line with the IRA

6.10.6 Components

The list of main components to be deployed and used in the pilot includes:

- Filter: Filtering component to remove specific rows and columns
- Join: Service to join two or more datasets where at least one column must be the same
- Prelaboration: Service to transform categorical variables into numerical ones throgh different calculation
- Outliers detection (Isolation forest): Given a set of observations (x1, x2, ..., xn), where each observation is a d-dimensional real vector, isolation forest associate to each• Fraudolent transaction detection: exploiting a supervised classifier algorithm (e.g., random forest classifier, neural network classifier), classify incoming data in two categores: suspected frauds or clean transactions. observation a value that expresses how much it differs from the distribution calculated on each dimension.
- Fraudolent transaction detection: exploiting a supervised classifier algorithm (e.g., random forest classifier, neural network classifier), classify incoming data in two categores: suspected frauds or clean transactions.
- Results: Storage that contains all the processed data elaborated by the workflow published to be visualized.
- Visualization: The service that gets the resulting datasets to be delivered to the visualization.

6.11 Pilot #11: Personalized insurance products based on IoT connected vehicles

6.11.1 Pilot Objectives

In a few words, this pilot aims to develop new services for driving insurance companies, based on the information gathered from a connected vehicle, as an IoT ecosystem. Current driving insurance services try to reward good drivers against the “bad one”, but based on very static or historical information: your age, colour of your car, incidents by year, etc. A new approach, more dynamic, adapted and custom services are needed. You pay as you drive, in a similar approach to a cloud word, where you pay as you consume. Complementary to this, a second service will help to detect possible fraud’s situation. Fraud causes not fair costs to the company that would affect indirectly to the good/ honest drivers.

In both use cases the underline technology is based on connected vehicles, IoT and BigData, because of the expected amount of data to be managed. The business analysis part, which determines how good driver you are, and the detection of possible frauds, will be based on AI and ML techniques. Due to the personal data managed in the pilot, security and privacy will be also a technology challenge to achieve.

6.11.2 Data Sources

The main data source in the pilot is produced by the connected vehicle, with about 20 vehicles. It is under study, the inclusion of some historical data provided by vehicles from other previous project; if legally possible. The data produced by the connected vehicle includes: CAN data, traffic events, gps, speed, etc. Complemented with data provided by the city of Vigo.

Finally, the data will be complemented with some synthetic/simulations of vehicles trips. Based on an opensource tool, SUMO and a custom developed adaptor to integrate and transform the data, according to expected data pipeline and data standards.

About data standards, the Smart Fleet platform layer, in charge of gathering, homogenizing, filtering, etc, is based on a FIWARE platform. Therefore, FIWARE Data Models will be used during the project. In that point, it is expected to contribute back to these standardization efforts fostered by the FIWARE Foundation. Some models would be adapted, or new ones would be created.

The pilot will make use of the following dataset:

- Simulated Urban Mobility Dataset (ATOS, ~368 GB): Simulated Urban mobility data (mainly vehicles CAN Signals) through different scenarios (cities). Captured from SUMO tool
- CAN Data (Historical Data) (CTAG): Data collected from vehicle's CAN Bus (20 vehicles driving 4 h/day 1 year). Historical data coming from existing deployments
- Traffic Events (Historical data) (CTAG, ~900 GB): Traffic events published by the city of Vigo and DGT (Historical data related to captured CAN Data)
- NMEA Data for vehicles (Historical) (CTAG, ~120 GB): Complementary location (GPS, Timestamp, speed, heading…) for Vehicles' CAN Data (Historical data related to captured CAN Data)
- CAN Signals (Live) (CTAG, ~150 GB): CAN data + Driving style info (revolutions, gear, hard breaking…)+ Parking (close doors, windows…) + Maintenance
- Traffic Events (Live) (CTAG, ~250 GB): Traffic events published by the city of Vigo and DGT
- NMEA Data for vehicles (Live) (CTAG, ~50 GB): Complementary location (GPS, Timestamp, speed, heading…) for Vehicles' CAN Signal
- Motor Insurance Data (DYN, ~500 MB): Data concerning motor insurance including data from the policies (duration, covers), data from vehicles (licence No, VIN etc.) and data from drivers (age, experience etc.)

6.11.3 Data Produced

Two main business services will be produced during the pilot's implementation. Therefore, it is not so focused on producing data, but, to provide services use. These services will be used, internally, by the insurance company. In any case, the data produced (or the results) by these services would be considered as data produced, that can be stored in a database, to feed new chains/workflows.

Pay-As-You-Drive service:

- Input: driver's trip info
- Output: a value from 0 to 100 about the driver's behaviour.

Fraud detection:

- Input: driver's trip info
- Output: a kind of driver.

It would be used to compare the kind of driver against an historical register. Example of usage in case of an accident: it would check if the kind of driver differs from previous days (stolen vehicle, identity theft).

6.11.4 Explainable Workflow

The data collected from the vehicle is transmitted to the INFINITECH Testbed, where the data is pipelined into a workflow with a set of steps. Before going to the Smart Fleet platform, data is prepared about regulation and anonymized to protect the driver's privacy. With the data prepared to be managed, the Smart Fleet Platform homogenize, filter, clean, and standardize the data (based on FIWARE Data models). Here the data is prepared as time series for real time management, or, it is stored as historical information. Looking at the AI Platform, it is expected to develop/train two different ML models for the two business services. Once the models have been implemented and these are available in the platform, these models will be trained, supported by the previous data gathering workflow. Getting the training data from the Smart Fleet Platform. It is important to clarify that the training process is not a matter of getting data, training and finish. The model will be constantly trained according to specific scheduling. The model will be always updated with the new data that constantly is generated by the connected car: (1) Data Management: data produced -> pre-prepared -> gathered -> streamed of store (2) (scheduling time raises) (3) ML Model training: data features extraction -> training model -> store the model The usage of the model, or inference service, or business service, it is an independent workflow. It just deploy a service that will exploit the previously trained model. These are interconnected, the first time the services are deployed with the model, this is linked to future training. When new training succeeds with more accurate models, the inference service will update the resulting model automatically.

6.11.5 Logical Schema

An INFINITECH-RA compliant architecture of the pilot is depicted in the following figure:

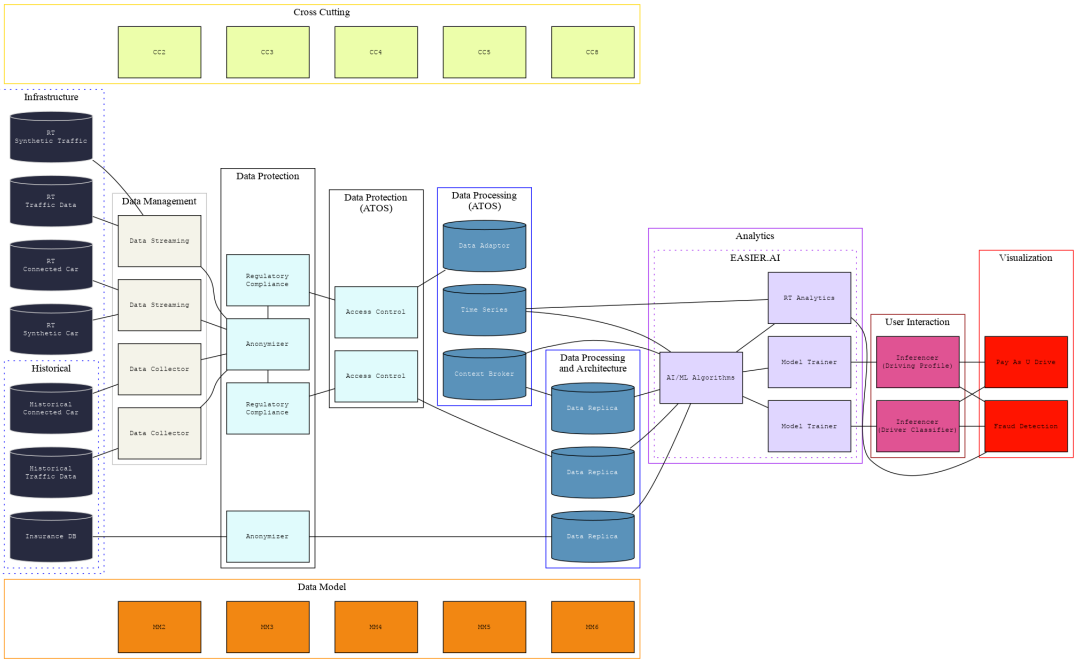


Figure 55 Personalized insurance products based on IoT connected vehicles pilot pipeline in-line with the IRA

6.11.6 Components

The main components to be used in the pilot include:

- Smart Fleet Framework (Context Broker) (Data Management in RA)
- Smart Fleet Framework (PeP Proxy) (Data Security and Privacy in RA)
- Smart Fleet Framework (Historical DB: CrateDB) (Data Source in RA)
- Smart Fleet Framework (Context DB: Mongo) (Data Source in RA)
- Smart Fleet Framework (QuantumLeap) (Data Source in RA)
- Smart Fleet Framework (Weather Injector) (Data Ingestion in RA)
- Smart Fleet Framework (IoT Agent) (Internet of Things in RA)
- Smart Fleet Framework (Grafana) (Interface in RA)
- Security Framework (IDM) (Data Security and Privacy in RA)
- Anonymiser (GRAD Anonymiser) (Data Security and Privacy in RA)
- EASIER.AI (Elasticsearch) (Data Management in RA)
- EASIER.AI (kibana) (Analytics and Machine Learning in RA)
- EASIER.AI (logstash) (Data Ingestion in RA)
- Pay as You Drive Service (Interface in RA)
- Fraud Detection Service (Interface in RA)

6.12 Pilot #12: Real World Data for Novel Health-Insurance products

6.12.1 Pilot Objectives

Risk assessment is an integral part of the insurance industry, but it is usually static, done at the beginning of a contract with a client. The continuous estimation of risk factors is the aim of this pilot, an estimation based not just on medical history, but on lifestyle and behaviour, as they are continuously monitored. This allows the insurance companies to offer personalized dynamic products,

where clients' premiums are calculated dynamically based on their habits. Complementary to this, a second service will help to detect possible fraud's situation. Fraud causes not fair costs to the company that would affect indirectly to the good/honest clients.

In both use cases the underline technology is based on analysing Real-World Data (RWD) of the clients. The business analysis part, which determines how healthy a client's lifestyle is, and the detection of possible frauds, will be based on ML techniques. Due to the personal data managed in the pilot, security and privacy will also play an important role.

6.12.2 Data Sources

The main data source in the pilot is the RWD collected by Healthentia. Healthentia is a platform for measuring and reporting RWD. Measurements are based on sensors on smartphones or IoT wearable devices. Reports employ questionnaires that the clients periodically answer utilising the Healthentia app. A secondary source of data is the records of the clients of the health insurance companies. Finally, the data will be complemented with synthetic/simulated data.

- Healthentia Live (average 720kB per user per week): Measured physical activity (steps, floors, sleep and heart rate) and user reported data from users of Healthentia SaaS who have given consent
- Healthentia Simulated (average 720kB per user per week): Simulated physical activity and reported data

6.12.3 Data Produced

Two main business services will be produced during the pilot's implementation. Therefore, it is not focused on producing data, but on service provision. These services will be used by the insurance company. To have these services, the ML module will be producing models which can be considered as data produced, that can be stored in a database, to feed new chains/workflows.

Risk assessment service:

- Input: client's lifestyle, enumerated by long-term, short-term averages and trends of physiological parameters that have to do with activity, sleep, the heart, nutrition, hydration, body signals (blood presure, temperature), weight and symptoms (pain, fatigue, diarrhea, nausea, cough).
- Output: decisions on health outlook are accumulated across time, forming a health assessment ranginh from -100 to +100.

Fraud detection:

- Input: client's lifestyle enumerated as above, models of all clients.
- Output: probability of fraud, enumerating mismatch of current behavior from past behavior of client and other clients.

6.12.4 Explainable Workflow

The RWD collected from the client using Healthentia and the secondary sources is transmitted to the INFINITECH Testbed, where they are aggregated together, anonymised for protection and stored. Stored data are either used to (re)train the risk and fraud assessment models. The trained models are used by the services on input data without anonymisation to provide the risk and fraud assessments. The outputs of the services are offered to the health insurance professionals via the presentation layer of the pilot, together with all collected RWD for human insights/verification. The presentation layer is the Healthentia portal app.

6.12.5 Logical Schema

An initial mapping of the pilot architecture to the INFINITECH-RA is depicted in the following figure.

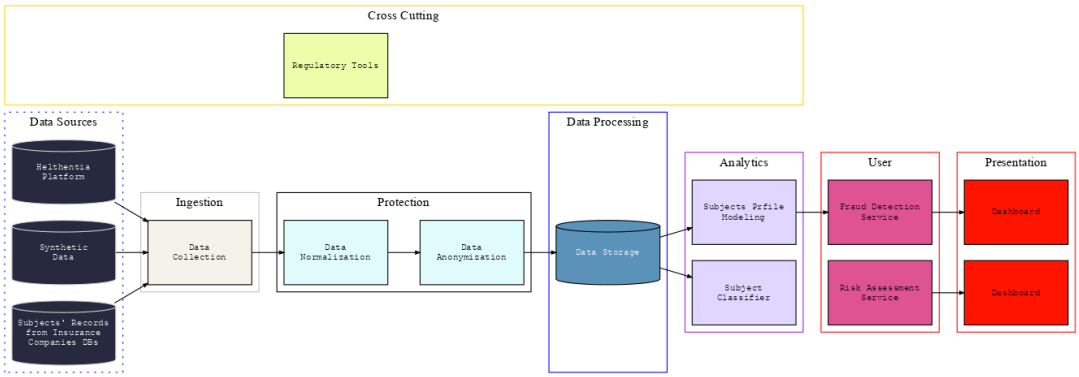


Figure 56 Real World Data for Novel Health-Insurance products pilot pipeline in-line with the IRA

6.12.6 Components

The following components will be deployed and used as part of the pilot:

- UBITECH Data Capturing Tool (Data Ingestion in RA)
- LeanXcale Database (Data Management in RA)
- Innovation Sprint's ML services (risk assessment and fraud detection) (Analytics and Machine Learning in RA)
- ATOS Regulatory tool through Data protection Orchestrator (DPO) (Data Security and Privacy in RA)
- GRAD Regulatory tool through Anonymization Component (Data Security and Privacy in RA)

6.13 Pilot #13: Alternative/automated insurance risk selection - product recommendation for SME

6.13.1 Pilot Objectives

The pilot will implement an automation of the subscription process that helps the insurance company reduce costs. In addition, being able to verify that the data entered is correct with a double verification avoids possible errors in the cost of the insurance premium The monitoring and identification of real-time risk changes allows the company to know if the insurance cost really corresponds to the real risk of the SME or if it should increase or decrease it to adapt it to its current situation.

6.13.2 Data Sources

Data will be extracted from open sources such as company websites, official registers, social networks, opinion forums, etc. Data will include 150.000 SME targets with 50.000.000 data fields.

- SMEWIF: SMEs website information and functionalities. Description of the text containing in the website of the companies, services and structure of the company
- ROPS: Review and opinions platforms. Reputation information and opinions of clientes about productos ans services
- EUBD: European SMEs Business Directories. Oficial and legal information about the comanies, social object, activitie, other companies where they have equity
- GIO: SMEs geolocation information and characteristics, images and geographical information

- SMSIP: Social media SMEs information and presence.
- I&R: Key performance indicators and insurance needs

The Pilot will also use synthetic data. P13-Alternative/automated insurance risk selection - product recommendation for SME SMEs synthetic raw data

6.13.3 Data Produced

The output will be the ERP (Enterprise Risk Profile) and EIAU (Enterprise Insurance Automated Underwriting).

With these two outputs, not only the risk profile and its levels of any SME company are obtained, but also the information for the automation of the subscription and the application of rules to obtain the price automatically.

6.13.4 Explainable Workflow

Pilot #13 is a "Big Data" data analysis platform applying ML (Machine Learning) and AI (Artificial Intelligence) technologies to better predict the insurance needs of SMEs.

Well, this system must be prepared to offer a commercial use to the companies, so it must have a user interface so that they can manage the information (Frontend) and a management layer at a logical level (Backend).

The companies (enterprises) will access our platform through a registration process and subsequent validation by assigning a package of number of customers, the basic and commercial information will be recorded in Amazon Cognito and the logical information of the company will be recorded in a table of DynamoDB called Enterprises.

With regard to the use of the information by the companies, the user must load the information they have stored in their systems in our platform, this will receive the name of raw data (crude-data). The raw data will be uploaded to the platform as structured information in CSV format. The companies that use our service will have a limited amount of clients loaded in crude-data, for this, the fields of the Enterprises table, limit, clients_uploaded, total_clients_uploaded will be used in a monitored way.

Each row of this document will identify a client, which can be target in different sources of information on the Internet and other open sources in real time, depending on the information available (the quality of information depends on the company), which will be recorded in the DynamoDB Targets table (Infrastructure).

Once the robots can obtain the least updated target for a particular source, they will proceed to obtain the information and subsequent storage in big data, this information container will be in Amazon S3 in a loop called big-data as well as stored in the folder with the name of the company's identifier. The information obtained from the source will be stored in the folder mentioned above in a JSON document whose name will be the user's identifier. (Data Management) From this layer the analysis algorithms will be applied and the results of the analyzed companies will be shown with the indicators of risk levels and the configuration and automation of the underwriting obtaining the ERP (Enterprise Risk Profile) and EIAU (Enterprise Insurance Automated Underwriting) (Data Processing and Visualization).

It is important to note that the quotes to services provided by AWS are for illustrative purposes and provided they have the same technical and technological characteristics they can be replaced by another supplier, as in the case of the project under consideration may be NOVA and LeanXcale.

6.13.5 Logical Schema

The following figure provides an INFINITECH-RA compliant logical view of the logical architecture of the pilot.

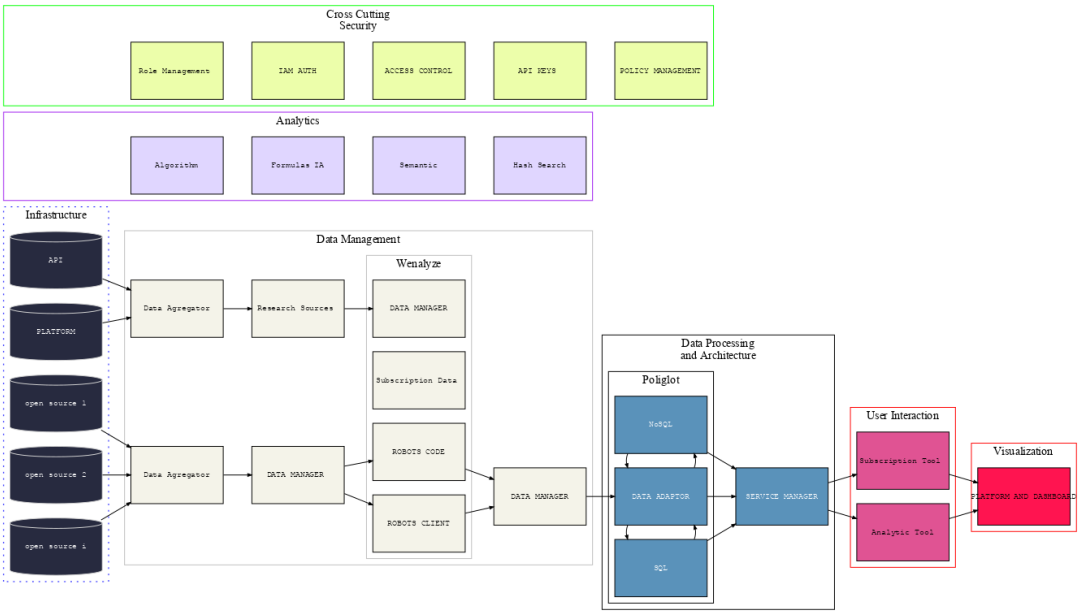


Figure 57 Alternative/automated insurance risk selection - product recommendation for SME pilot pipeline in-line with the IRA

6.13.6 Components

The following INFINITECH component will be used as part of this pilot:

- Data Acquisition Layer (Data Ingestion in RA); The data acquisition layer is composed of micro-robots that roam the data sources. The deployment of the micro-robots is discretionary depending on speed and analysis needs, being a flexible and scalable deployment;
- INFINISTORE (HTAP data store and the polyglot engine) (Data Management in RA);
- Analytics Layer (Analytics and Machine Learning in RA);
- Connectivity layer through API-Rest.

6.14 Pilot #14: Big Data and IoT for the Agricultural Insurance Industry

6.14.1 Pilot Objectives

The objective of Pilot #14 “Big Data and IoT for the Agricultural Insurance Industry” is to deliver a commercial service module that will enable insurance companies to exploit the untapped market potential of Agricultural Insurance (AgI), taking advantage of innovations in Earth Observation (EO), weather intelligence & ICT technology. EO will be used to develop the data products that will act as a complementary source to the information used by insurance companies to design their products and assess the risk of natural disasters. Weather intelligence based on data assimilation, numerical weather prediction and ensemble seasonal forecasting will be used to verify the occurrence of catastrophic weather events and to predict future perils that could threaten the portfolio of an agricultural insurance company. The INFINITECH AgI-module derived indices will allow and enable the agricultural insurance industry to enlarge its market, while delivering a larger portfolio of products at lower costs and serve areas, where classical insurance products could not be delivered.

6.14.2 Data Sources

The main data source for the pilot is produced by satellite and a weather intelligence engine. The Earth Observation (EO) data will be derived from the satellites Sentinel-1,2,3, LandSat-8, MODIS and PROBA-V). Also, numerical weather predictions for the pilot areas (gridded data) are generated each day and will replace the previous prediction. Lastly, gridded climate indices based on ERA-5 Land and ERA-5 Reanalysis Data will be used for the pilot.

Following, the list of datasets is presented (GEN):

- Gridded Climate Indices (1/1/1979 to 31/12/2019): Climate Indices based on the ERA-5 Land and ERA-5 Reanalysis Data
- EO Data: Earth Observation Data (Sentinel- 1,2,3/LandSat-8, MODIS, PROBA-V) for remote damage and crop loss assessment
- Numerical Weather Predictions: Very High-Resolution Weather Predictions for the Pilot Areas

6.14.3 Data Produced

The data produced will result in a solution for Agricultural Insurance companies allowing them to efficiently couple EO satellite data and weather/climate data with any type of complementary data (from separated drone shots to ultra-high-resolution SAR imagery). The INFINITECH AgI module will enable Insurance companies to alleviate the effect of weather uncertainty when estimating risk for AgI products, reduce the number of on-site visits for claim verification, reduce operational & administrative costs for monitoring of insured indexes and contract handling, & design more accurate & personalized contracts. By deriving impartial indices on top of a multitude of data, the module will allow insurers to reduce significantly the time needed for handling and verification of claims and the costs imposed by fraud, moral hazard and adverse selection.

6.14.4 Explainable Workflow

The data produced by the Octopush EO Service (Crop Monitoring, Pest & Disease Services, Damage Assessment Services) and the Agro Apps Weather Intelligence Service (Weather Forecast Services, Climate Services) is pipelined into a Data integrator. The integrator feeds back Metadata and information on the Area of Interest (AOI) to the data producing services. The integrator itself is retrieving the AOI & Metadata from a Business DB storage layer. The geospatial data storage and data persistence mechanisms allows the storage of the geometries and zonal statistics and provides the essential functionality for querying and retrieving data via an API (alerts) or WMS server components (vectors source). The WMS server is then responsible for rendering and serving of the GIS layers to the User Interface. The restful API will act as a communication and data exchange bridge, that allows the platform to share processed and structured content internally, between the different components. The front-end user interface is the gateway responsible to present all the system data through user-friendly controls and web mapping interfaces.

6.14.5 Logical Schema

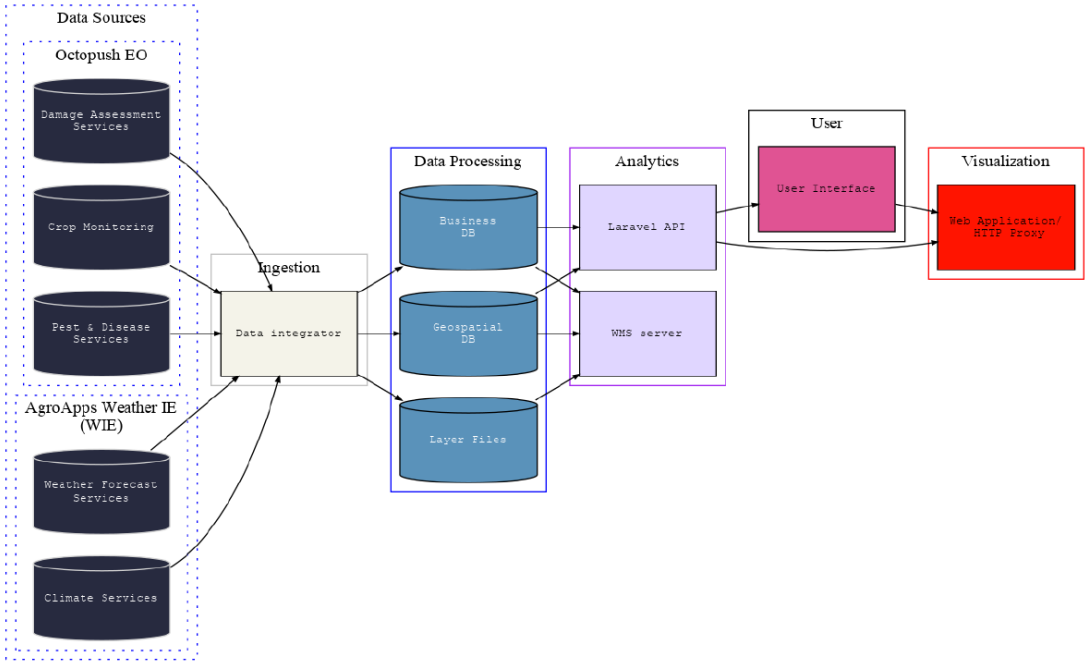


Figure 58 Big Data and IoT for the Agricultural Insurance Industry pilot pipeline in-line with the IRA

6.14.6 Components

The pilot comprises ICT modules and services for the insurance sector.

ICT Modules:

- Octopush EO Service (Data Source in RA): Octopush EO Service is an integrated satellite derived software service, which collects earth observation, geospatial, in-situ and other geo-referenced data. It applies appropriate processing algorithms and returns the results in a ready-to-use format.
- AgroApps Weather Intelligence Engine (AgroApps WIE) (Data Source in RA): The WIE is an integrated weather derived software service which collects weather information from several resources and along with the geo- referenced data, it applies appropriate processing algorithms and returns the results in a ready-to-use format.
- Data integrator (Data Ingestion in RA): The Data Integrator acts as a bridge between the WebGIS subsystem, Octopush EO service and WIE. It is responsible for performing the essential scheduled calls to the data providers in order to fetch and process the desired EO and weather information. It is able to run calls on demand or daily data integration tasks by retrieving EO data and weather products from Octopush EO service and WIE and transforms, binds, injects those into the WebGIS database.
- Business and Geospatial DB (Data Management in RA): Business DB offers a storage layer essential to carry the business logic and relevant information/ data stored and managed by API. It also stores, retrieves and provides information related to user accounts, settings, actions and preferences. The geospatial data storage and data persistence mechanisms allows the storage of the geometries and zonal statistics and provides the essential functionality for querying and retrieving data via an API or WMP server components.
- Web Map Server (WMS Server) (Analytics and Machine Learning in RA for Geoserver and Interface for Apache Tomcat and RESTful API): WMS is responsible for rendering and serving of the GIS layers to the User Interface.

- RESTful API (Interface in RA): The API will act as a communication and data exchange bridge, that allows the platform to share processed and structured content internally, between the different components.
- User interface (Interface in RA): The front-end user interface is the gateway responsible to present all the system data through user-friendly controls and web mapping interfaces.

Services for the Insurance Sector:

- Remote Damage Assessment for drought and hail
- Flood and wildfires mapping
- Short and medium range weather forecasts
- Seasonal Climate Forecasts of Agroclimatic Indicators
- Climate Risk Assessment

6.15 Pilot 15 Open Inter-banking Pilot

6.15.1 Pilot Objectives

Pilot 15 main objective is to deliver a prototype to address and tackle business pains shared within banking institutions leveraging Machine Learning and Natural Language Understanding paradigms. The model aims at reading and analyzing extensive internal documents of banks in real time to highlight the main concepts and compare them with a reference taxonomy to build a common business glossary in order to:

- provide banks with a tool able to standardise the documentation analysed;
- Increase Automation and Intelligence based on data processing leveraging data governance processes;
- improve the analysis and comprehension capabilities of internal documents and contents.

6.15.2 Data Sources

Data will be extracted from a large set of bank's internal documents in pdf, word and/or txt format, provided by the banks involved in the pilot. The documents will focus on the following areas: KYC, entering into a relationship with the customer and the Markets in Financial Instruments Directive (MIFID). In addition to documents related to the three specific areas, other data sources includes:

- additional documents relating to different areas and identified randomly within the document base:
- internal dictionaries, internal glossaries, internal taxonomies useful for the development of metadating techniques.
- ABI Lab architectural framework (reference taxonomy)

6.15.3 Data Produced

The advanced document processing will allow real-time useful information via searching semantically relevant text according to the semantic metadata, increasing automatisation, easiness of use and usefulness of outcomes.

6.15.4 Explainable Workflow

STAGE 1 – study and research @ ABI Lab controlled environment

- Data ingestion/preparation, including technical components aimed at normalising and aggregating the data that we need for our specific analytical purposes, preparing the information to be processed by the Machine Learning tools;
- Data storage, including tools and infrastructures aimed at data collection from different sources and in different formats, and their storage;
- machine learning engine optimisation, enabling continuous Natural Language Understanding algorithms optimisation, following the use case experimental purposes
- semantic model design A data visualisation layer, including tools and methods to display results to different users and stakeholders.

STAGE 2 – test and validation @ Infinitech testbed (Model based on BDVA RA)

6.15.5 Logical Schema

The following figure illustrates the logical architecture of the pilot in-line with INFINITECH-RA constructs and approach.

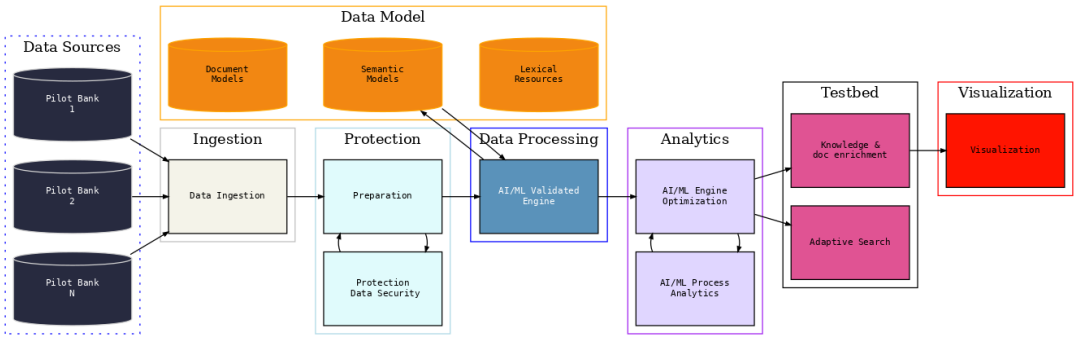


Figure 59 Open Inter-banking pilot pipeline in-line with the IRA

6.15.6 Components

The main technological components that will be implemented and integrated as part of this pilot are:

- A data storage layer, including tools and infrastructures aimed at data collection from different sources and in different formats, and their storage;
- A data ingestion/preparation layer, including technical components aimed at normalising and aggregating the data that we need for our specific analytical purposes, preparing the information to be processed by the Machine Learning tools;
- A machine learning engine layer, including Natural Language Understanding algorithms, opportunely configured for the use case purposes.

6.16 Pilot 16: Data Analytics Platform to detect payments anomalies linked to money laundering events

6.16.1 Pilot Objectives

Nexi, as the Italian paytech leader, owns and manage a large, big data ecosystem, which includes information regarding cardholders, merchants, organizations, and digital payment authorizations and transactions. The pilot will build a data analytics platform to help Nexi AML team to discover, monitor and analyze suspicious scenarios related to money laundering through digital card payments.

The pilot purpose is to preside anomalous scenarios linked to money laundering, adhering to European AML regulatory compliance policies, by notifying detected cases to the Italian Financial Intelligence Unit (FIU). The innovation potential of current pilot lies in introducing novel technologies like, machine learning, artificial intelligence, graph database to detect anomalous scenarios, which allows to automatically detect complex anomalous money-laundering scenarios.

The adoption of pilot platform will improve quality and efficiency of AML users work and, at the same time, will concur in reducing risk of unmatched scenarios related to money laundering events.

6.16.2 Data Sources

The following data sources will be integrated and used in the pilot:

- Cardholders transaction operations
- Cardholders information registry
- Merchants transaction operations
- Merchants information registry
- AML Anomalies Features Store
- AML Suspicious Activities Report (SAR) practices collection
- Master and reference data

All above-mentioned data sources are in an anonymized format and are stored and collected into a Data Lake environment to enable agile development and processing.

The pilot will use a graph database, to model many-to-many relations that belongs to anomalous events linked to money laundering; thanks to this technology we can find out any relationship occurring between a suspicious payment events and individuals or merchants.

6.16.3 Data Produced

The three data outputs produced during the pilot are:

1. Anomalous subjects
2. Cluster of anomalous subjects
3. Anomaly risk score for each subjects

Decision rules developed into the graph database, based on many-to-many associations, will produce periodically (monthly or quarterly accordingly) anomalous subjects (1) or groups, clusters, of anomalous subjects (2) intercepted.

During the pilot we will develop an algorithm that update anomaly risk scores (created with machine learning supervised classification algorithm) associated to any subject: those updated score is the last data product generated (3).

Subjects can be, cardholders, legal entities, organizations, legal representatives. Any sensitive information, such as person ID, is anonymized so that it’s not traceable to physical or legal person.

The output format will be a csv text or a parquet file stored into the Data lake. Those formats allow not to be bounded to any particular database technology. In a post-processing phase, we can then accordingly choose how to use and model those data: a relational database to perform SQL queries and to be the back-end of a data visulization tool, or just a plain .csv file to perform exploratory data analysis with data science frameworks.

6.16.4 Explainable Workflow

The data workflow considers all steps and data shapes needed to develop a ML solution to find anomalous scenarios : the collection of data (listed in the previous paragraph), the processing step to create a training, validation and test set, the training of risk score with ML algorithms and the presentation layers to communicate results.

As a first step, we collect in anonymized format historical data about Nexi clients behaviour, such as transaction payments, withdrawals, merchants information, reversals, money transfer, past SAR reported to Italian Financial Unit (IFU), into a Data Lake.

Afterwards, we apply the Transformation step of a typical Extract, Load, Transform(ELT) workflow to create the Feature Store ; that is, a dataset containing Machine Learning features for each cardholder (or organization) together with the target variable, the outcome of the ML algorithm, that in this case is binary variable, representing whether a cardholder has been notified to IFU. It is updated monthly in batch mode.

Once the Feature Store is ready, we follow these steps:

1. Training machine learning model and, based to the predictions generated, we get the anomaly risk score for each cardholders (or organization).
2. Create a graph database, inserting cardholders, organizations, legal representatives behaviours data (from both Data Lake and feature store) and ML based risk score
3. Perform a Personalized Page Rank algorithm to adjust risk scores, taking into account the many-to-many relationships modelled with graph data structures.
4. Define rule based anomaly events detection to find customers of groups of customers (clusters) to whom AML users would pay attention

All the steps mentioned above are then stored into the Data Lake in a file format (.csv) or compressed file like *parquet*, and then modelled into relational database tables or views to make those accessible to analytics users.

Finally, a visualization dashboard allows to end users (Customers Due Diligence team members) to explore and visualize outputs of the data workflow, and so can discover and analyse riskier subjects as suggested by algorithms developed.

6.16.5 Logical Schema

The explainable workflow of the pilot to the INFINITECH-RA layers is in the following figure.

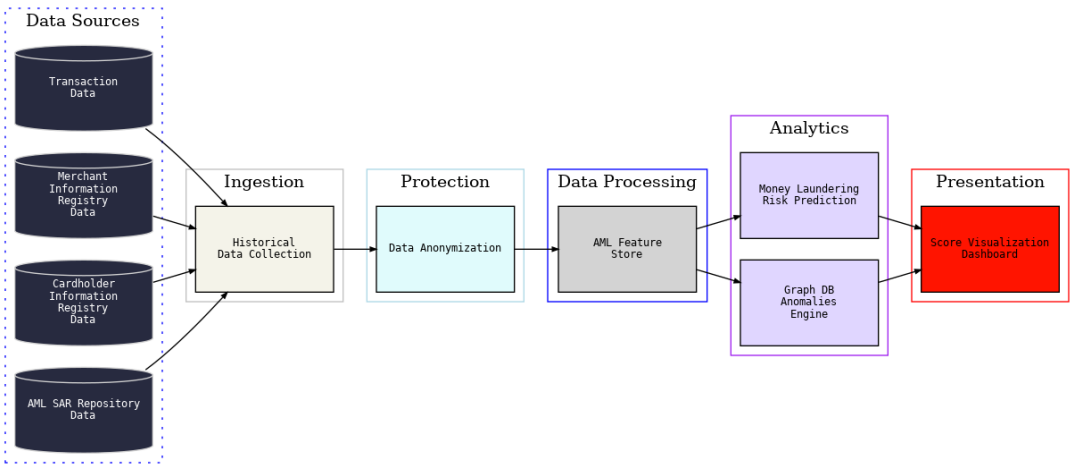


Figure 60 Pilot 16 pipeline in-line with the IRA

6.16.6 Components

The following main components will be deployed and used in the pilot pipelines:

- BigData Management Layer to collect and process data (Data Management in RA)
- Money Laudering Risk Prediction supervised classification model and Graph database engine to adjust risk scores (Analytics and Machine Learning in RA)
- Visulaization dashboard of customers wirh higher risk of money laundering event (Visualization in RA)

7 INFINITECH-RA positioning

7.1 Unique Points and Propositions of INFINITECH-RA

RA seems generic and high level. Shouldn't it be more concrete?

The INFINITECH RA presented in the document is described in High Level on purpose. It does not specify which node does what and does not put constraints on connections between nodes. It describes the principles upon which nodes with specific functions can be integrated in the INFINITECH reference implementations to create a large class of solutions. It allows a classification of nodes based on their functions to be mapped with a well established Reference Model (BDVA RM) and enables the generalization of use cases.

A specific implementation, however, can impose strict constraints on nodes interfaces (like REST APIs) to make nodes interoperable (interoperable, stackable, orchestrable).

Where is the Innovation?

The RA is a model and a set of principles on how to integrate specific functionalities in any instantiation of the INFINITECH platform. However, the principles are quite innovative and powerful as they imply the possibility to build solutions out-of-the-box with standard components taken out from a library of services (nodes). INFINITECH nodes which comply with the INFINTECH RA, will be interoperable, stackable, exchangeable, orchestrable. It allows the integration of nodes that performs functionalities like Data Collector to different data sources, Data Anonymization, Data Ingestion, Data Processing, Data Streaming, Data Abstractions and Ontologies, Data Analytics, Data Manipulation and Presentation.
An INFINITECH implementation based on the DevOps, Microservice Architecture and Kubernetes Deployment is highly powerful and innovative as it allows the creation of specific solutions to solve a virtually infinite class of use cases. The RA contains the principles to build Library of Standard Components, to populate a MarketPlace, to work with an Orchestrator Tool, to experiment with Sandboxes, to create powerful solutions out of the box.

This has very few competitors in the market now.

What are the specific features for the Financial and Insurance Sector?

The IRA is a general model and not specific to any sector. However, it is modeled after the INFINITECH requirements from use cases and pilots and complies with the Financial and Insurance Sector regulations and best practices. The specific features for banks, insurance company and other organizations are in the specific building blocks and solutions for data manipulation (from ingestion to analytics and to presentation) developed to solve the use cases of the project. The project will develop semantic engines and Machine Learning Algorithms which will be vertical using ontologies of the financial World (FIBO, FIGI, etc). The RA offers a general framework reusable in large domains of financial sector's scenarios.

How can the RA cope with Business Requirements, Technological Requirements, Regulatory Requirements, etc.?

Requirements of all types form a checklist against which the IRA and its implementation will be checked. For example, a business requirement will be assessed in a specific pipeline/sandbox implementation. A Regulatory Constraints will be checked both in the General Framework (there is a specific layer that complies with privacy or security in IRA) and in the specific implementation (is the solution compliant with Regulatory X?) Moreover, cross cutting services at different layers of the RA can implement automatic checking for requirements as stackable/pipelinable nodes. This will be quite innovative and will represent a unique selling point of the INFINITECH project, as for instance to Anonymization as-a-service, GDPR as-a-service, etc.

Where are the building blocks described in the DoA and how can they be mapped in the RA?

Specific tools and applications can be mapped in the IRA layered model. LeanXcale powerful database management solution, for example, can have a central role in the Data Processing Architecture, providing a generalization for Data Repository with streaming endpoints for Analytics.

Anonymization is also very easy to map and fits into the IRA as a pipeline service. Anonymization can be implemented as a node which gets data from one source, transforms and produces data sent to another node (a data repository such as LeanXcale repo). It is as simple as two endpoints and it can be configured by cross cutting services. There could be many types of anonymization services specific to the particular data sets but the endpoints could be the same allowing for exchangeability and interoperability of the service.

How can the IRA solve Pilots and Use cases?

The IRA provides a general framework and each use case will be different. A mapping of a use case is basically the operation to identify which functions are needed at different layers, which tools and applications will solve them and to define a pipeline of services (nodes) able to manipulate the datasets of the use case to provide the expected results. Some examples of Use Cases are provided and this document will not be complete until all the use cases are mapped to the RA. No use case left behind.

I am an End User and I cannot see how my Use Case can fit in the RA.

Most of the use cases map smoothly in the pipelined approach of the IRA. For instance, a simple case is to ingest all the data from different data sources (siloed in data lakes) into a central repository after collection, anonymization, categorization. Then a level of abstraction (polyglot and ontologies) can allow analytics to work on this aggregated set of data. Applications like dashboards or Visualization tools will work out of the produced data.

Another case is to have intelligent processing as close to the datasource as possible. In this case analytics will query data directly and produce data for presentation.

Distributed Ledgers (blockchains) pose a challenge to the IRA which can be solved in many different ways (see below).

However, the IRA does not contain any specific solution but a general framework to create different ways to build them up.

Is the RA being implemented in the Cloud? Will my data be distributed outside my organization? How can I preserve the classification of my own data?

As far as the INFINITECH Project is concerned, the principle is that Data that belongs to Financial Organizations stays in the Financial Organizations. The IRA is not imposing anything like exporting data outside any organization. It is not the specification of a Cloud Service. The project will organize on-premise testbeds and the pilots will deploy an instance of the IRA in sandboxes in the infrastructure of the organizations. There are cases where synthetic data can be used which will not require on-premise infrastructure but this is specific to Pilots. However, the IRA, can be implemented as Sandbox-As-A-Service on a public Cloud using for instance synthetic data, preserving the confidentiality and security of the data of the tenants.

Blockchain Technology does not seem to fit in the RA. How can it be managed?

Indeed, a distributed ledger poses challenges to the central data processing layer of the IRA and to the underlined pipeline concept. However, this is only apparent as a blockchain node can be accessed to gather the relevant information and produce data that can go as input to other nodes. A Blockchain node can be considered as any other datasource at the infrastructure layer and accessed through a specific Data Collector (see Web3.js APIs).

DLT can also find a place in the central layer of the IRA and even as a cross cutting service, for peer-to-peer no-man-in-the-middle authorization, and even for licensing services.

DLT is central to INFINITECH project and the IRA provides many ways to address them. More will be defined in the specific use cases.

8 Conclusions

A reference architecture for BigData systems in digital finance can greatly facilitate stakeholders in structuring, designing, developing, deploying and operating BigData, AI and IoT solutions. It serves as a stakeholders' communication device, while at the same time providing a range of best practices that can accelerate the development and deployment of effective systems. This deliverable has introduced the final version of such a reference architecture, namely the INFINITECH-RA. The latter adopt the concept and principles of data pipelines, which are in-line with the state of the art in BigData and Artificial Intelligence systems. It is also in-line with the principles of the reference architecture of the BDVA. In practice, it extends and customizes the BDVA RA with constructs that permit its use for digital finance use cases. The INFINITECH-RA defines the structuring principles that drive the integration of the INFINITECH technical components and technologies, which have been documented in deliverable D2.6.

As part of the deliverable, we have illustrated how INFINITECH-RA can be used to support the development of common BigData/AI pipelines for digital finance applications. In this direction the deliverable has provided some examples of popular pipelines. It has also provided an initial mapping of most pilots of the project to the INFINITECH-RA, using INFINITECH components and technologies. The presented RA will serve as a basis for the implementation of the final versions of the MVP (Minimum Viable Product) of the pilots. The completion of the final version of the INFINITECH-RA here reported signifies the accomplishment of several of WP2 objectives as listed In the following table.

As per the specific KPIs set for the project the following table addresses the specific indexes listed in the DoA.

KPI	Description	Comment
KPI 1	RA to be specified >= 1	This Deliverable documents the RA of the project and the RA logical schema of the pilots of the project. Therefore the KPI is fully achieved
KPI 2	Supported types of different types of databases, data sources and data stores >= 12	The RA in principle manage different kind of Data sources (RDBMS SQL and NoSQL, Data Lakes, IOT and Blockchain datasources). The number is far more than the 12 basic KPI

Table 5 Mapping of INFINITECH DoA/Task KPI with Deliverable Achievements

The INFINITECH-RA presented in this and previous deliverables, has received numerous stakeholders' feedbacks from the application of the concept to the actual usage as a tool to simplify design and implementation of a number of POC and Applications exploiting BigData/AI systems for the Finance and Insurance Sectors. The INFINITECH-RA has been refined over time, based on its actual use in other workpackages, including the sandboxes and pilot development workpackages. Along with other methodologies and technologies it is definitely part of the INFINITECH WAY.

Nevertheless, the INFINITECH-RA is a non-static concept and will evolve in the future, continuously adapting and benefiting from different use cases and more importantly from the end users in a continuous *design thinking* process. We are confident that the foundations and the concepts that were set as pillars were correct and have helped and could help and support the Financial and Insurance sector in the future.

9 Appendix A: References

[Armbrust15] M. Armbrust, R. Xin, C. Lian, Y. Huai, D. Liu, J. Bradley, X. Meng, T. Kaftan, M. Frank-lin, A. Ghodsi, M. Zaharia, “Spark SQL: relational data processing in Spark”, in ACM SIGMOD, 2015, pp. 1383-1394.

[BDVA-RM] Big Data Value Association. BVA SRIA—European big data value strategic research and innovation agenda. 2017. http://bdva.eu/sites /defau lt/files /BDVA_SRIA_v4_Ed1.1.pdf. Accessed 18 Feb 2020

[Boid18] Boid D, Chang W. NIST Big Data Interoperability Framework: Volume 6, RA Version 2. NIST Big Data Program. 2018. https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-6r1.pdf

[Bons12] Bons, R.W.H., Alt, R., Lee, H.G. et al. Banking in the Internet and mobile era. Electron Markets 22, 197–202 (2012). https://doi.org/10.1007/s12525-012-0110-6

[FINSEC19-D2.5] FINSEC Reference Architecture II – Available at https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5ce3a941d&appId=PPGMS

[BOOST4.0-D2.5] BOOST4.0, D2.5 – BOOST 4.0 Reference Architecture Specification v1, Available at: https://cordis.europa.eu/project/id/780732/results [Accessed: 15-May-2020]

[Bracke19] Philippe Bracke, Anupam Datta, Carsten Jung and Shayak Sen, “Machine learning explainability in finance: an application to default risk analysis”, Bank of England, Staff Working Paper No. 816, August 2019

[Duggan15] J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, S. Zdonik, “The BigDAWG polystore system”, SIGMOD Record, vol. 44, no. 2, pp. 11-16, 2015.

[Gadepally16] V. Gadepally, P. Chen, J. Duggan, A. J. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson, M. Stonebraker, “The BigDawg polystore system and architecture”, in IEEE High Performance Extreme Computing Conference (HPEC), 2016, pp. 1-6.

[Kruchten95] Kruchten, Philippe (1995, November). Architectural Blueprints — The “4+1” View Model of Software Architecture. IEEE Software 12 (6), pp. 42-50.

[Komulainen18] Komulainen, H., Makkonen, H. Customer experience in omni-channel banking services. J Financ Serv Mark 23, 190–199 (2018). https://doi.org/10.1057/s41264-018-0057-6

[Kyriazis18] D. Kyriazis, et al, “BigDataStack: A Holistic Data-Driven Stack for Big Data Applications and Operations”, BigData Congress, San Francisco, CA, USA, 2018: 237-241

[Minpeng11] Z. Minpeng, R. Tore, “Querying combined cloud-based and relational databases”, in Int. Conf. on Cloud and Service Computing (CSC), 2011, pp. 330-335.

[Minukhin18] S. Minukhin, V. Fedko and D. Sitnikov, "SQL-On-Hadoop Systems: Evaluting Performance of Polybase for Big Data Processing," 2018 International Scientific-Practical

Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 591-594. [MIRAB12] Microsoft Industry Reference Architecture for Banking (MIRA-B), Microsoft Corporation Whitepaper, May 2012

[Ong14] K. W. Ong, Y. Papakonstantinou, and R. Vernoux, “The SQL++ semi-structured data model and query language: a capabilities survey of SQL-on-Hadoop, NoSQL and NewSQL databases”, CoRR, abs/1405.3631, 2014.

[Özsu11] T. Özsu, P. Valduriez, Principles of Distributed Database Systems, 3rd ed. Springer, 2011, 850 pages.

[Ribeiro16] Ribeiro, Marco & Singh, Sameer & Guestrin, Carlos. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. 1135-1144. 10.1145/2939672.2939778.

[Shearer00] Shearer C., The CRISP-DM model: the new blueprint for data mining, J Data Warehousing (2000); 5:13—22.

[Simitsis12] A. Simitsis, K. Wilkinson, M. Castellanos, U. Dayal, “Optimizing analytic data flows for multiple execution engines”, in ACM SIGMOD, 2012, pp. 829-840.

[Tomasic98] A. Tomasic, L. Raschid, P. Valduriez, “Scaling access to heterogeneous data sources with DISCO”, IEEE Trans. On Knowledge and Data Engineering, vol. 10, pp. 808-823, 1998.

[Xu18] H. Xu, M. Chen, Y. Zhou, B. Du and L. Pan, "A Novel Comprehensive Quality Index QoX and the Corresponding Context-aware System Framework," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018, pp. 2415-2419.

[OR] https://www.oreilly.com/library/view/software-architecture-with/9781786468529/ch08s04.html

[SA] http://www.dossier-andreas.net/software_architecture/index.html

[AS] http://www.se.rit.edu/~emad/teaching/slides/CISC322_06_ArchitectureStyles_sep20.pdf

[DPA] https://sarasanalytics.com/blog/data-pipeline-architecture

[NTK] https://www.astera.com/type/blog/data-pipeline-architecture

[AA] https://airflow.apache.org/docs/apache-airflow/stable/index.html

[SL] https://luigi.readthedocs.io/en/stable

[KAP] https://www.knime.com/

[C3] https://c3.ai/products/c3-ai-ex-machina/

[LP] https://pipeline.loni.usc.edu/learn/user-guide/building-a-workflow/

[SP] https://streampipes.apache.org/ (https://streampipes.apache.org/)

10 Appendix B: Functional requirements

Funct. Reqs.	DB	Lakes	Files	IoT	Block chains	Collect	Connec	Serializer	eIDAS	Anonym	DUOS	DPO	API	Proxy	Data	Model	Results
REQ-P1-F-B-M-O-001			X														
REQ-P1-F-B-M-O-002		x															
REQ-P1-F-B-M-O-003										X							
REQ-P1-F-B-M-O-004																	
REQ-P1-F-B-M-O-005						X	X	X									
REQ-P1-F-B-M-O-006																	
REQ-P1-F-B-M-O-007																	X
REQ-P2-F-B-M-O-008																	
REQ-P2-F-B-M-O-009									X	X							
REQ-P2-F-B-M-O-010																	
REQ-P2-F-B-M-O-011															X		X
REQ-P3-F-B-M-O-012																	
REQ-P3-F-B-M-O-013																	
REQ-P3-F-B-M-O-014																	
REQ-P3-F-B-M-O-015																	

REQ-P3-F-B-M-O-016									X								
REQ-P3-F-B-M-O-017									X	X							
REQ-P3-F-B-M-O-018																	
REQ-P3-F-B-M-O-019									X								
REQ-P3-F-B-M-O-020																	
REQ-P4-F-B-M-O-021																	
REQ-P4-F-B-M-O-022									X								
REQ-P4-F-B-M-O-023																	
REQ-P4-F-B-M-O-024																	X
REQ-P6-F-B-M-O-043	X					X	X	X									
REQ-P6-F-B-M-O-044		X	X			X	X	X									
REQ-P6-F-B-M-O-045																	
REQ-P6-F-B-M-O-046																	
REQ-P6-F-B-M-O-047																	
REQ-P9-F-B-M-O-061	X					X	X	X									
REQ-P9-F-B-M-O-062									X	X							
REQ-P9-F-B-M-O-063																	

REQ-P9-F-B-M-O-064															X		X
REQ-P10-F-B-M-O-065	X						X										
REQ-P10-F-B-M-O-066						X			X	X							
REQ-P10-F-B-M-O-067						X											
REQ-P10-F-B-M-O-068																	
REQ-P10-F-B-M-O-069															X		
REQ-P10-F-B-M-O-070																	
REQ-P10-F-B-M-O-071						X	X	X	X	X							
REQ-P10-F-B-M-O-072					X	X	X	X									
REQ-P10-F-B-M-O-073					X												
REQ-P10-F-B-M-O-074	X																
REQ-P10-F-B-M-O-075																	
REQ-P10-F-B-M-O-076															X	X	X
REQ-P11-F-B-M-O-077				X		X	X	X									
REQ-P11-F-B-M-O-078		X	X			X	X	X									
REQ-P11-F-B-M-O-079		X	X			X	X	X									
REQ-P11-F-B-M-O-080										X							

REQ-P11-F-B-M-O-081																
REQ-P11-F-B-M-O-082																
REQ-P12-F-B-M-O-083				X		X	X	X								
REQ-P12-F-B-M-O-084									X	X						
REQ-P12-F-B-M-O-085																
REQ-P13-F-B-M-O-086		X	X			X	X	X								
REQ-P13-F-B-M-O-087																
REQ-P13-F-B-M-O-088																
REQ-P13-F-B-M-O-089																
REQ-P14-F-B-M-O-090		X	X			X	X	X								
REQ-P14-F-B-M-O-091																
REQ-P14-F-B-M-O-092		X	X			X	X	X								
REQ-P14-F-B-M-O-093																
REQ-P14-F-B-M-O-094																
REQ-P14-F-B-M-O-095																
REQ-P14-F-B-M-O-096																
REQ-P14-F-B-M-O-097																

REQ-P14-F-B-M-O-098																	
REQ-P14-F-B-M-O-099																	