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

Infinitech

D6.1 – Testbeds Status and Upgrades- I

Lead Beneficiary	**CP**
Due Date	2020-07-31
Delivered Date	2020-07-31
Revision Number	2.0
Dissemination Level	Public
Type	Report
Document Status	Release
Review Status	QA Reviewed
Document Acceptance	WP Leader Accepted and/or Coordinator Accepted
EC Project Officer	Pierre-Paul Sondag

Contributing Partners

Partner Acronym	Role[1]	Name Surname[2]
CP	Lead Beneficiary	Marinos Xynarianos
AGRO	Contributor	Gregory Mygdakos, Polymachi Symeonidou
AKTIF	Contributor	Orkan Metin
ATOS	Contributor	Jose Gato
BANKIA	Contributor	Raquel Fernan Pisonero
BOC	Contributor	Silvio Walser
BOI	Contributor	Eymard Hooper
BOS	Contributor	Jelena Milosevic
BPFI	Contributor	Gavin Purtill, Richard Walsh
DYN	Contributor	Michalis Michalakoukos
ENG	Contributor	Susanna Bonura, Domenico Messina
GFT	Contributor	Ernesto Troiano, Vittorio Monferrino, Maurizio Megliola,
HPE	Contributor	Alessandro Mamelli, Domenico Costantino
ISPRINT	Contributor	Aristodemos Pnevmatikakis
JRC	Contributor	Petra Ristau
JSI	Contributor	Maja Skrjanc
NBG	Contributor	Manos Silignakis, Nikos Droukas
NOVA	Contributor	Pedro Malo
PI	Contributor	Massimiliano Aschi
PRIVE	Contributor	Pablo Carballo, Teoman Onat
RB	Contributor	John Kaldis
UPRC	Contributor	Dimos Kyriazis, Dimittris Kotios,
WEA	Contributor	Carlos Albo Portero
ABILAB, CTAG, GEN, IBM, SIA	Contributor	
NUIG	Internal Reviewer	Martin Serrano
INNOV	Quality Assurance	Dimitris Drakoulis

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance

[2] Can be left void

Revision History

Version	Date	Partner(s)	Description
0.1	2020-05-31	CP	ToC Version
0.2	2020-06-15	CP, ALL	Initial Draft Version
0.3	2020-06-30	CP, ALL	Second Draft Version
0.4	2020-07-10	CP, ALL	Third Draft Version
1.0	2020-07-17	CP, GFT, NUIG	First Version for Internal Review
1.1	2020-07-22	CP, GFT, NUIG	Second Version for Internal Review after GFT Comments
2.0	2020-07-24	INNOV	Version for Quality Assurance
3.0	2020-07-31	CP	Version for Submission

Executive Summary

INFINITECH is a **joint effort** of Europe's leaders in ICT and **Finance/Insurance sectors** towards providing the technological capabilities, the **experimentation facilities** (**testbeds & sandboxes**) and the business models needed to enable European financial organizations, insurance enterprises and FinTech/InsurTech innovators to fully leverage the benefits of **BigData, IoT and AI technologies**. Several Large-Scale Innovative Pilots in Finance and Insurance, which will leverage both the technological developments of the project and the **testbeds/sandboxes** in order to deploy and validate novel use cases in real-life environments based on realistic datasets.

The development of BigData, IoT and AI-based innovations requires significant testing and validation efforts, such as testing for regulatory compliance and optimizing Machine Learning (ML) and Deep Learning (DL) models. Therefore, there is a need for advanced **experimentation infrastructures**, which shall provide access to resources for application development and experimentation, such as datasets, regulatory tools, libraries of ML/DL algorithms, Open APIs and more. Such **experimentation infrastructures** should be available in appropriate **testbeds**, based on the deployment of the relevant technical building blocks that will be used in various configurations.

The goal of task T6.1 "Testbeds Analysis, Customization and Continuous Upgrade" is the describing the analysis of **existing testbeds** (i.e. testbeds of incumbent organizations) in terms of their **existing resources** and **gaps** for supporting BigData, IoT, AI experimentation in-line with the **INFINITECH** approach. It will accordingly specify the ways they have to be **extended** in terms of **hardware and/or software** resources.

In particular the deliverable contains:

- a first analysis of the hardware & software, as well as any additional security or special requirements for the **experimentation facilities** (**testbeds & sandboxes)**, that will host the relative pilots
- Proposed extensions or upgrade of the testbeds infrastructure that will be required in hardware **and/or software** resources in order to support the development of **BigData, IoT and AI-based innovations**

The work related to task T6.1 will continue until Month 20, when the 2nd version of this deliverable will be submitted (D6.2), with the updates on the specification, and will end on Month30, where the last version will be submitted, as testbeds will be upgraded dynamically in order to fulfil the project's and the pilots' testbeds & sandboxes needs.

Table of Contents

List of Figures

List of Tables

Abbreviations

3DS	Three-Domain Secure
4MLD	Fourth Money Laundering Directive
API	Application Programming Interface
AWS	Amazon Web Services
BDVA	Big Data Value Association
CPU	Central Processing Unit
DB	Database
DDR	Double data rate,
DEV	Development
DL	Deep Learning
ESG	Environmental, Social, and Governance
ETF	Exchange-Traded Fund
GDPR	General Data Protection Regulation
gPRC	High-performance, open source universal RPC framework
GRIB2	Second version of the World Meterological Organization's (WMO) standard for distributing gridded data
HTTP	Hypertext Transfer Protocol
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
IoT	Internet of Things
JWT	JSON Web Token
KYB	Know Your Business
KYC	Know Your Customer
MiFID	Markets in Financial Instruments Directive
MiFIR	Markets in Financial Instruments and Amending Regulation
ML	Machine Learning
MPI	Message Passing Interface
NDA	Non-Disclosure Agreement
NETCDF4	Network Common Data Form
NIS	Network and Information Systems
OCR	Optical Character Recognition
OES	Operators of Essential Services

OKD	Distribution of Kubernetes optimized for continuous application development and multi-tenant deployment
P2PP	Peer-to-Peer Payment
PaaS	Platform as a Service
PAN	Primary Account Number
PCI DSS	Payment Card Industry Data Security Standard
PDF	Portable Document Format
PHP	Popular general-purpose scripting language (Hypertext Pre-processor)
PIA	Privacy Impact Assessment
POC	Proof of Concept
PSD2	Payment Service Directive 2
PSP	Payment Service Provider
PSU	Payment Service User
QTSP	Qualified Trust Service Provider
RA	Reference Architecture
RAM	Random Access Memory
REST	Representational state transfer
RTS	Regulatory Technical Standard
SA	Supervisory Authority
SCA	Strong Customer Authentication
SECaaS	Security-as-a- Service
SFTP	Secure File Transfer Protocol
SHARP	Smart, Holistic, Autonomy, Personalized and Regulatory Compliance
SIT	System Integration Test
SME	Small and Medium-Sized Enterprises
SSD	Solid State Disk
TBD	To be defined
TI	Threat Intelligence
TOTP	Time-Based One-Time Password
UAT	User Acceptance Test
VCPU	Virtual CPU
VDIH	Virtualized Digital Innovation Hub
WRF-ARW	Weather Research and Forecasting (WRF) - ARW (Advanced Research WRF)
WRFDA	Weather Research and Forecasting (WRF) model data assimilation system (WRFDA)
XML	Extensible Markup Language

1. Introduction

Task 6.1 provides the initial Specification for hardware & software, as well as any additional security or special requirements, of the existing or planned to be built testbeds, for hosting Pilots executions in terms of their existing resources and gaps for supporting BigData, IoT, AI experimentation in-line with the INFINITECH approach.

This deliverable intends to provide the details of the infrastructure that each testbed will use in order to support the Pilots execution, utilizing the initial Specification for the 10+2 testbeds, that will be hosted on premises at the finance organizations of the consortium and 2 (EU-wide) testbeds, as well as provide initial contribution for the required changes/enhancements that will be suitable in order all Pilots to be realized.

The project's results will be validated in the scope of 15 high impact pilots providing complete coverage of the sectors, including Know Your Customer (KYC), customer analytics, personalized portfolio management, credit risk assessment, preventive financial crime analysis, fraud anticipation, usage-based insurance, agro-insurance and more.

1.1. Objective of the Deliverable

This deliverable specifies the **initial analysis** for the relative infrastructure (hardware & software) of the 10 testbeds hosted from finance organizations of the consortium and 2 (EU-wide) testbeds, that will be implemented as part of the INFINITECH, taking into account the functionalities of the technological building blocks of the project, along with the relative data assets and needs for compliance to certain regulations (e.g., GDPR, PSD2, MiFiD, 4MLD). Also, **initial specification** of the **upgrades or modifications** of the **hardware/software resources** that will be available or being developed for validating autonomous and personalized solutions that will be implemented through the relative pilots and respective testbeds, including a unique collection of data assets for finance/insurance.

1.2. Insights from other Tasks and Deliverables

The deliverable is the first one that is released for WP6, so the contents of the initial **analysis** for the current status of the **infrastructure (hardware & software**) that will be used for all testbeds to host the Pilots of INFINITECH Project. Based on this description the other deliverables of WP6 will be conducted. Task 6.2 and 6.3 will provide deliverable **D6.4 Tools and Techniques for Tailored Sandboxes and Management of Datasets,** to be used from all testbeds**.** Tasks 6.4 and 6.5 **will put in place (in the field testbed and sandboxes)** the **tools and mechanisms** defined in Task 6.2 and 6.3. Task 6.6 will **specify and implement processes** for **certifying and standardizing digital finance/insurance solutions** in the project **tailored sandboxes and testbeds**. The following diagram (see Figure 1) depicts the interconnections between WP6 Tasks:

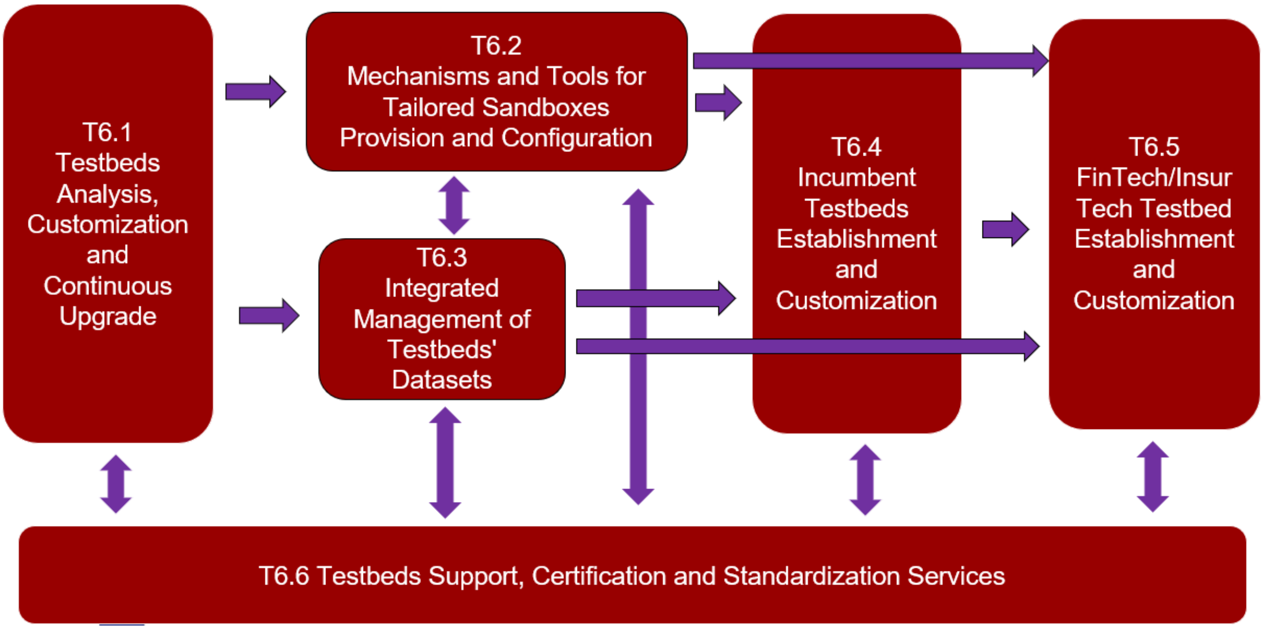
T6.1
Testbeds Analysis, Customization and Continuous Upgrade
T6.2
Mechanisms and Tools for Tailored Sandboxes Provision and Configuration
T6.3
Integrated Management of Testbeds' Datasets
T6.4
Incumbent Testbeds Establishment and Customization
T6.5
FinTech/InsurTech Testbed Establishment and Customization
T6.6 Testbeds Support, Certification and Standardization Services

Figure 1 – Schema of the links among Tasks of WP6

1.3. Structure

This deliverable is composed of five main sections. Chapter 1 is the introduction to the deliverable and includes the description of the objective, insights from other tasks and deliverables and the structure. Chapter 2 describes the methodology followed for the collection of the relative information from all INFINITECH Partners included in the deliverable. Chapter 3 contains the Initial analysis for the relative infrastructure (hardware & software) of the 10 testbeds hosted from finance organizations of the consortium and 2 (EU-wide) testbeds, that will be implemented as part of the INFINITECH. Chapter 4 describes the testbeds & sandboxes implementation General strategy that will be required to be followed from all Pilots. Finally, Chapter 5 reports some conclusions.

2. Methodology

The major source of insights of deliverable D6.1 is the INFINITECH Pilots and their contributions based on specific questionnaire that was distributed for collection of the relative information. The answers to the questionnaire are included in Chapter 3 and in order Testbed Hosts and Pilot Leaders to respond to that, were mainly based on the contents from other deliverables of WP2:

- D2.16 User Stories and Stakeholders' Requirements – I (Restricted)
- D2.17 Reference Scenarios and Use Cases – I (Restricted)
- D2.5 Specifications of INFINITECH Technologies - I
- D2.20 Security and Regulatory Compliance Specifications – I (Restricted)
- D2.22 Initial Specification of Testbeds, Data Assets and APIs – I (Restricted)

The contents of Chapter 4 -Testbeds & Sandboxes Implementation General strategy was provided from HPE which is leading WP6 and will provide all the require support regarding the technical and operational implementation for all Pilots Testbeds & Sandboxes implementation,

The results of those contributions are explained in the following sections.

3. Testbeds & Sandboxes Infrastructure Specification

3.1 Introduction

The INFINITECH Tailored Sandboxes and Testbeds INFINITECH will make available a number of **testbeds** for experimentation, testing and validation of BigData, AI and IoT solutions, including:

(A) **10 testbeds** that are established at the data centres of incumbent financial organizations and (B) **2 testbed** that will be provisioned and established in order to support the experimentation of the FinTech and InsurTech pilots and enterprises of the consortium.

These testbeds will be used for experimentation and validation of the INFINITECH pilots, while at the same time supporting demonstration and exploitation activities of the project as part of the market platform and the VDIH. Within each of the testbeds, the project will configure sandboxes tailored to the experimentation and validation requirements of the project’s pilots. Each of the sandboxes will include the following:

1) **INFINTECH technological building blocks** that will enhance the testbeds’ infrastructure with seamless data management and efficient analytics functionalities.
2) **Data assets** for training and validation of AI and BigData analytics algorithms (notably ML/DL algorithms), but also for validation and evaluation of pilot functionalities.
3) **Open APIs** for experimentation connected to each pilot, but also for supporting innovation beyond the specification pilot requirements (e.g., in the scope of hackathons). The Open APIs will be a customized version of the Open APIs of the INFINITECH technology enablers (i.e. the APIs for Data Management and Analytics).

4) **Regulatory Compliance Tools** that will boost the compliance of the INFINITECH developments based on combination of encryption, anonymization, data policy management, consent management and eIDAS integration.
5) **Selected ML/DL algorithms** from the INFINITECH library, i.e. algorithms suited to the pilot development at hand.

The selection of the proper resources from the INFINITECH data assets, technology building blocks and data governance building blocks will facilitate the provisioning of the sandboxes in-line with business and regulatory requirements, and based on a **DevOps and microservices approach**. Selection, provision and configuration of data and software resources in-line with the INFINITECH-RA will lead to the tailored sandboxes. The tailoring process will be made possible based on some of the INFINITECH developments, such as seamless access to data that will facilitate the selection of data resources and the proper configuration of the APIs that will empower the sandbox. Standard cybersecurity solutions will be also provided as part of each testbed and sandbox. Note that each of the ten (10) testbeds will comprise sandboxes tailored to the individual pilots of the financial institutions. However, the testbed that will be developed and provisioned for the FinTechs/InsurTechs of the consortium will comprise a broader set of data assets, APIs and a richer set of INFINITECH technologies in order to support all the relevant pilots. The list of testbeds of the project is provided in the following table 1:

Table 1 - INFINITECH Testbed Hosts & Pilots

Testbed Host	Pilot No	Title	Leader
Category 1 - Smart, Reliable and Accurate Risk and Scoring Assessment			
BANKIA (Spain)	Pilot 1	Invoices Processing Platform for a more Sustainable Banking Industry	BANKIA
NOVA (Portugal)	Pilot 2	Real-time risk assessment in Investment Banking	JRC
Category 2 - Personalized Retail and Investment Banking Services			
BOI (Ireland)	Pilot 3	Collaborative Customer-centric Data Analytics for Financial Services	BOI
PRIVE (Austria)	Pilot 4	Personalized Portfolio Management ("Why Private Banking cannot be for everyone?")	PRIVE
LIB (Spain)	Pilot 5a	Smart and Personalized Pocket Assistant for Personal Financial Management	LIB
BOC (Cyprus)	Pilot 5b	Business Financial Management (BFM) tools delivering a Smart Business Advise	BOC
NBG(Greece)	Pilot 6	Personalized Closed-Loop Investment Portfolio Management for Retail Customers:	NBG

Category 3 - Predictive Financial Crime and Fraud Detection			
Selected Bank	Pilot 7	Operation Whitetail – Avoiding Financial Crime	FTS
BOS(Slovenia)	Pilot 8	Platform for Anti Money Laundering Supervision (PAMLS)	BOS
AKTIF (Turkey)	Pilot 9	Analyzing Blockchain Transaction Graphs for Fraudulent Activities	AKTIF
ENG (Italy)	Pilot 10	Real-time cybersecurity analytics on Financial Transactions' BigData	PI
Category 4 - Personalized Usage-Based Insurance Pilots			
NOVA (Portugal)	Pilot 11	Personalized insurance products based on IoT connected vehicles	ATOS
NOVA (Portugal)	Pilot 12	Real World Data for Novel Health-Insurance products	SILO
Category 5 - Configurable and Personalized Insurance Products for SMEs and Agro-Insurance			
NOVA (Portugal)	Pilot 13	Alternative/automated insurance risk selection - product recommendation for SME	WEA
NOVA (Portugal)	Pilot 14	Big Data and IoT for the Agricultural Insurance Industry	GEN

Information on the software/virtualization infrastructure of each testbed are provided in the Testbed descriptions that follow.

3.2 Testbeds for Smart, Reliable and Accurate Risk and Scoring Assessment

3.2.1 Testbed for Invoices Processing Platform for a more Sustainable Banking Industry

3.2.1.1 Testbed Brief Description

Testbed for Invoices Processing Platform for a more Sustainable Banking Industry will host **Pilot 1** of INFINITECH Project that leader is **BANKIA (BANKIA SA - Spain)**, which will host the relative testbed at AWS Private Cloud Bankia.

The pilot will process invoices documents (thousands of documents coming from thousand different authors) by using AI technologies (mainly neural networks) to extract useful information from them (invoicing concepts and their prices). 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.

The main components that will be part of this pilot are:

- Invoices and TableBank databases.
- Data Collector.
- Data converters/formatters: Image pre-processor, PDF to Image converter and OCR Converter.
- DL models (neural networks): Table Detector and Concepts & Prices Extractor.
- Data Results Visualization and Process Monitoring.
- Data Validator.
- Data Tagger.
- Training and fine-tuning pipelines of DL models.
- Models Repository and Models Publisher.

3.2.1.2 Testbed Functional & Technical Specification

a. Functional Specification

The main objective of the pilot is to develop, integrate and deploy a data-intensive system to extract information from notary invoices and include the following modules:

- Solution to identify tables within invoices.
- Solution to extract tables from the invoices.
- Solution to extract relevant indicators from the invoices that will be used to determine the sustainability score.

b. Technical Specification

From a technical perspective, the system will be comprised of 4 main blocks.

1. A data lake that will facilitate the secure storage of digitized documents.
2. A system to parallelize different jobs that compose the AI pipeline to pre-process, process and post-process the documents and the outcomes. For instance:
    - Image pre-processing (cropping, adjusting brightness, contrast, etc.)
    - PDF to Text
    - OCR
    - Text correction
3. A computer Vision system to identify and extract tables from invoices; from these tables we will extract the sensible information to establish a sustainability scoring
4. A machine learning to extract information from the identified and extracted tables;

3.2.1.3 Sandboxes

There aren't sandboxes yet.

3.2.1.4 Datasets

The datasets that will be used for the specific pilot are:

- Dataset with real data of 32.300 invoices documents, from 3.000 different notaries extracted from Bankia systems.
- Dataset TableBank: Table Benchmark for Image based Table Detection and Recognition, 500.000 documents (https://arxiv.org/abs/1903.01949)

3.2.1.5 Compliance Requirements

Legal terms and conditions will be pending of special terms and conditions agreed between parts in the future and will be included in the next version of deliverable D6.1.

3.2.1.6 Additional Info

- **Kubernetes version:** (min versions)
  - **Kubernetes** v1.11.0+d4cacc0.
  - **OKD** v3.11.0+ec8630f-265, it is an opensource layer on top of Kubernetes.
  - **Docker** v1.13.1, build: 7f2769b/1.13.1

- **Source code management:**
  - **Gitlab Community Edition 11.7.5**

- **Docker registry:**
  - **OpenShift Container Registry** (v v3.11.0+ea42280): storage of OKD docker images
  - **Nexus Sonatype** (v 3.20.1-01, OSS Edition): storage of services/application docker images and DL models binaries and artefacts.
- **CI/CD software:**
  - **Jenkins v2.204.1**

3.2.2 Testbed for Real-time risk assessment in Investment Banking

3.2.2.1 Testbed Brief Description

Testbed for Real-time risk assessment in Investment Banking will host **Pilot 2** of INFINITECH Project that leader is **JRC (JRC Capital Management Consultancy & Research - Germany)**.

The testbed shall host all components necessary for the two use cases of Pilot 2:

a) real-time calculation/update of risk measures

This use case will calculate two standard risk measures, VaR and ES, for an existing portfolio of open positions and update them in near real-time. In addition, it shall evaluate financial news and other textual data concerning relevance and potential impact on positions contained in the portfolio and derive warning levels.

b) pre-trade analysis

This use case shall serve as a “what-if” study when a trader intends to add a new trade to the portfolio. It shall add the (not yet executed) new trade to the open positions and update the VaR and ES calculation. In addition, it shall update the sentiment-based warnings in case that the new trade is for a market that was not included in the portfolio before.

3.2.2.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Institute for the Development of New Technologies (UNINOVA) – Portugal – On premise

b. Testbed Location info:

Institute for the Development of New Technologies (UNINOVA) – Portugal

3.2.2.3 Testbed Functional & Technical Specification

a. Functional Specification

The Pilot Components that will be hosted from the specific testbed consist of

- *data base(s) for market data, trades, news and other textual data,*
- *data base for storing history of risk measures (traceability / auditability)*
- *connection to real-time market data feed*
- *connection to financial news data feed*
- *connection to electronic order platform*
- *access to internet sources / results of sentiment analysis (to be clarified)*

Name	Description	Layer	Available	Technology
Portfolio input-stream	Data streamer – executed order, trade data	Data management	no	API to be developed
Data windows	Create fixed length arrays extracted from input time data. (data windows = segment of time series)	Data management	yes (draft)	Java comp
Correlation matrix	Process that calculates the correlation matrix from the data windows (merging different data)	Data processing	yes (to be adapted)	Java comp
Scenario generation	Input for the Montecarlo simulation, input from scenario specifications (tbd)	Data processing	yes (to be adapted)	Python comp

Text extraction	to feed market sentiment extraction	Data processing	tbd	
Market sentiment extraction	standard procedures to analyse sentiment/behavioural	Data analytics/processing (tbd)	yes (to be adapted)	
Analytics	inputs from Montecarlo simulations (scenario generation) and makes calculations of the VaR and calculates risk related market implications from market sentiment.	Data analytics	yes (to be adapted)	Python comp
Configuration	Input from users to configure specifications for scenario generation (confidence level, time frame/segment, inputs to VaR formula)	User interaction		coding
User interface	to be defined - two numeric values and graph(s)			

The Testbed shall support the following functionalities:

- *Streaming and Batch data pipelines processing*
- *Data pipeline execution monitoring*
- *Data visualization customization through a set of graphs and query editor*
- *Graphical User Interface to easily create a dashboard for the different risk measures on different levels*

b. **Technical Specification**

Infrastructure	Data Management	Data Protection	Data Processing Architectures	Data Analytics	User Interaction	Data Visualization

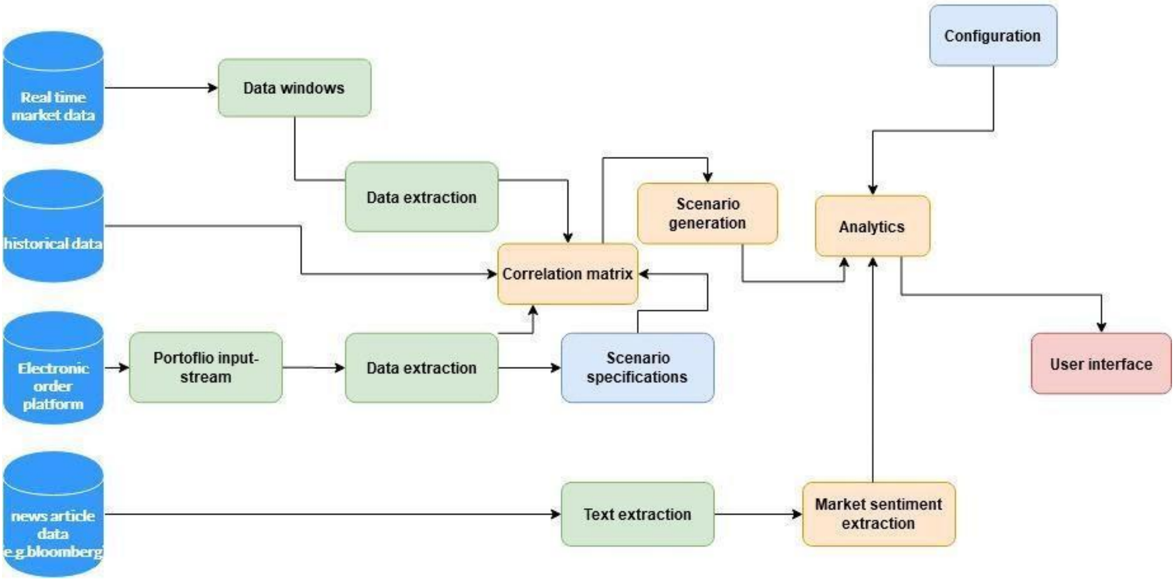

Figure 2 - JRC Testbed Technical Architecture Diagram

- **Hardware Requirements**
  - 16 GB RAM
  - 4 cores CPU
  - 500 GB HDD

- **Software Requirements**
  - Operating System: Windows Server (version will be defined on a later stage).
  - Layered Software: none
  - Application Software: none
  - Development Platform: Java, Python
  - Database: mySQL

3.2.2.4 Sandboxes

To be clarified, whether we will need one or two sandboxes, i.e. whether the use cases can share a sandbox or will need separate ones.

3.2.2.5 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

The dataset that will be used for JRC Testbed will be an operational datastore.
The dataset can be imported/migrated to the INFINITECH central repository.
Data updates on data will be migrated to the central repository, will be sync into the sandbox near real-time.

3.2.2.6 Compliance Requirements

Concerning deployment of regulatory compliance tools, pseudonymization of account numbers, accountability and auditability are requirements of medium to low priority.

Testbed & Sandbox environments will follow the Compliance & Regulatory Requirements already described in deliverable *D2.7 - Security and Regulatory Compliance Specifications.*

3.2.2.7 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.3 Testbeds for Personalized Retail and Investment Banking Services

3.3.1 Testbed for Collaborative Customer-centric Data Analytics for Financial Services

3.3.1.1 Testbed Brief Description

Testbed for **Irish Pilot Testbed – Digital Data Sharing Ecosystem for Customer, Account and Transaction Data** will host **Pilot 3** of INFINITECH Project that leader is **BOI (Bank of Ireland – Ireland).**

In trying to develop an ecosystem the Irish Pilot will have a significantly different footprint than other pilots.

The Irish pilot testbed will need to mimic a data sharing ecosystem by mimicking participants in that ecosystem and provide rules of engagement and highlighting the value exchanges between participants.

3.3.1.2 Testbed Functional & Technical Specification

a. Functional Specification

A digital ecosystem framework is described here to articulate testbed components required.

Digital Ecosystem Framework -

Organising Principle - An ecosystem, whether analog or digital, is based on an organizing principle, which represents the common need or interest that motivates participation. In this case 'financial data sharing capability'. This ecosystem may exist within the broader ecosystem of 'data sharing' in general and may overlap with other ecosystems e.g. 'Identity ecosystems'.

Shared Capabilities - Shared capabilities enforce rules and enable value exchange and participation. In a digital ecosystem, technology and standards to support participation and value exchange provide the bulk of shared capabilities. (See diagram below for shared capabilities)

Participants - A digital ecosystem includes a wide range of participants and roles: founders, leaders, providers, consumers, observers and others, including some who engage passively with the digital ecosystem. Any single participant can have multiple roles within a digital ecosystem. A digital ecosystem does not always have an owner, although many do:

- Founder (Organizing Principle): Every digital ecosystem has a founder, not an owner. The founder defines the organizing principle and the foundational shared capabilities, the rules for participation, the value to be exchanged and the shared capabilities.

- Leaders (Rules): The leaders (not the owners) are individuals or enterprises that support the founder's principles and guidelines by contributing to shared capabilities and evolving rules for participation.
- Providers (Shared Capabilities): Providers enable specific shared capabilities;
- Consumers (Value Exchange): Consumers are participants in value exchange.

In the case of the Irish pilot POC each participant will have different roles:

- Founder (Organizing Principle): BOI (a Bank), BPFI (coordinator of Banks).
- Leaders (Rules): BOI, BPFI, NUIG. E.g. BPFI partnership rules, BOI/NUIG data rules.
- Providers (Shared Capabilities): Mimic Bank A Providing Bank (might be On-Premise BOI or Cloud), BPFI as consortium data sharing service provider
- Consumers (Value Exchange): e.g. Mimic Bank B Consumer for KYC, Personal Consumer in Peer to Peer use case.

The Pilot for Digital Data Sharing Ecosystem for Customer, Account and Transaction Data have considered all elements above are part of investigation, however the Pilot cannot cover all capabilities or participants in a data sharing ecosystem, therefore the Pilot has been scoped to ensure the best validation of the overall ecosystem.

See below in Figure 3 for details of participants and shared capabilities.

- Red- Planned in scope of the Pilot.
- Orange- For further review and may be included in the Pilot, but will be investigated further.
- Blue – Will be investigated, but not in the Pilot build.
- Blank – Part of overall ecosystem, will be included in business model etc, but not in the Pilot.

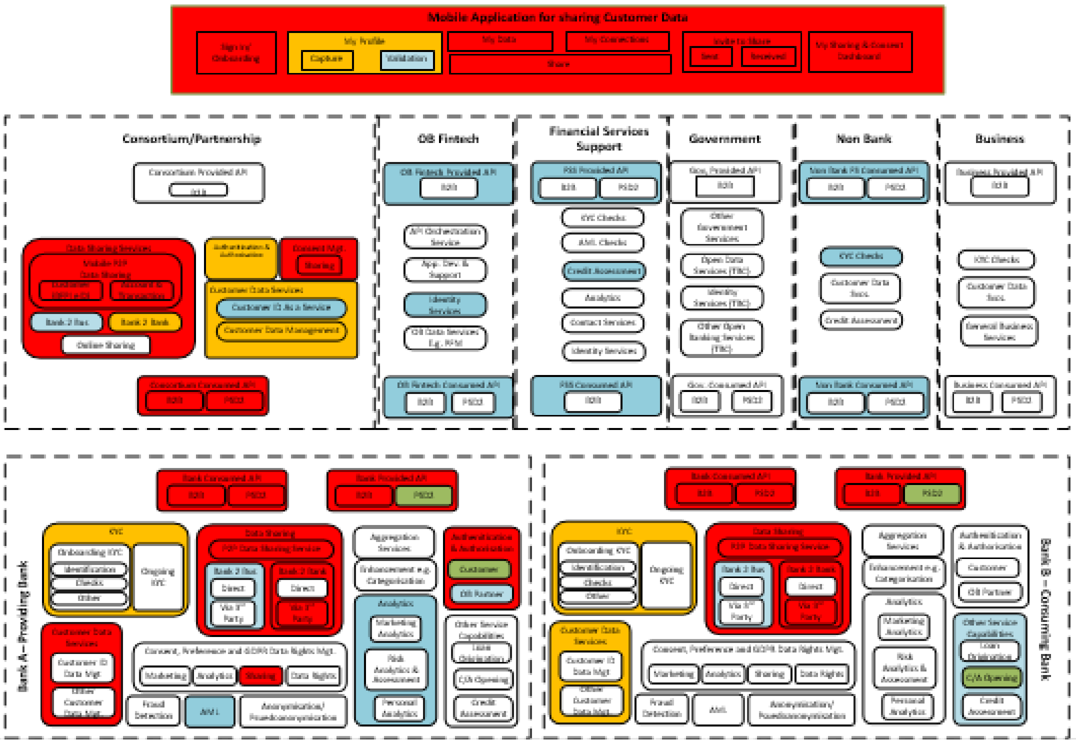

Figure 3 – BOI Testbed Participants and shared capabilities

The ecosystem description described above and data flow diagram for functional context and components etc.   Below is an example of KYC data flow. Previously submitted D2.13 – INFINITECH Reference Architecture - I document contains more ‘functional requirements per use case’ :

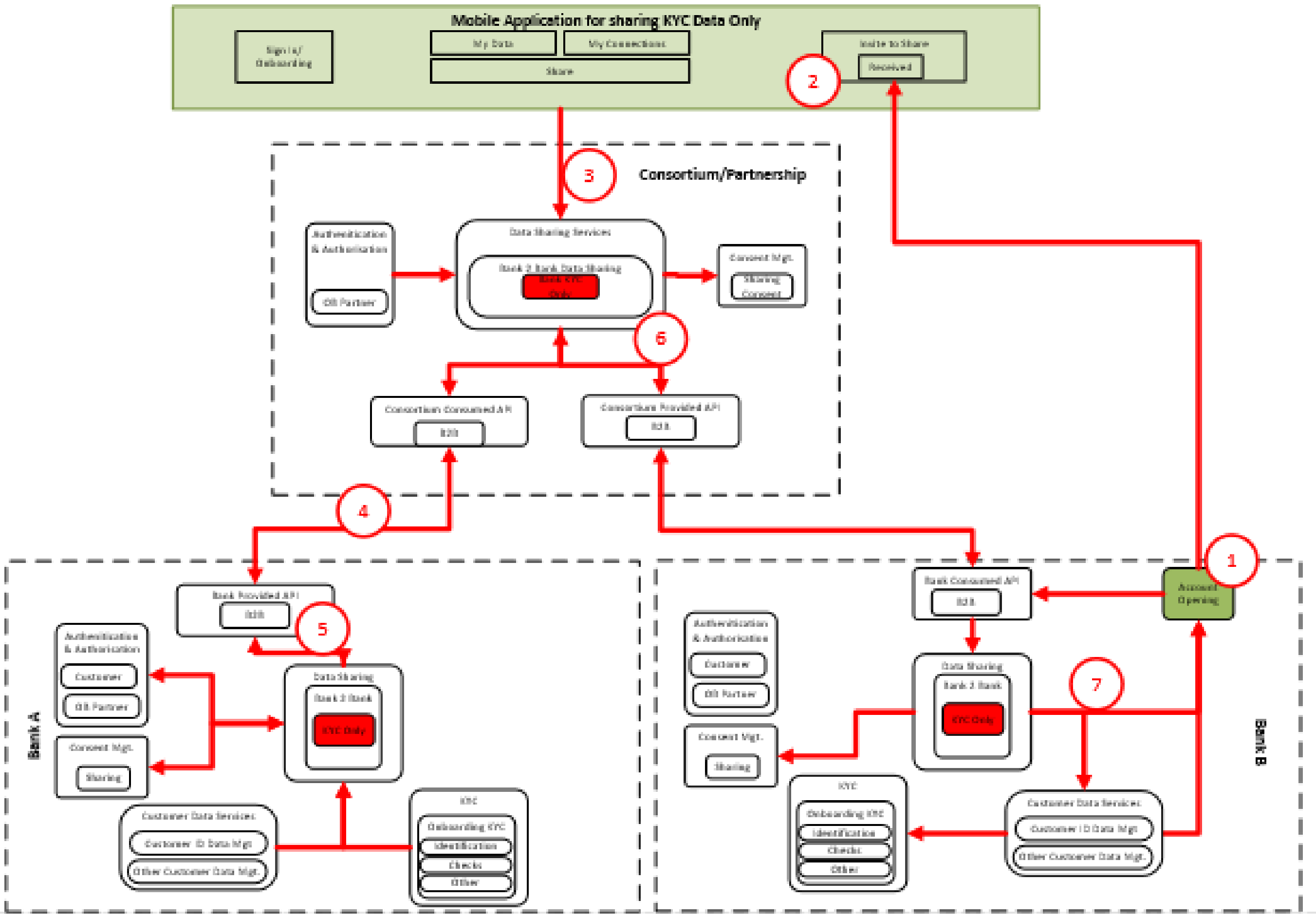

Figure 4 – BOI KYC Data Flow Diagram

a. Technical Specification

● Hardware Requirements

It is too early for Pilot 3 to specific exact technical architecture and unsure of value of hardware specification at this point and in context of cloud etc. and wiil be described in more detail in the next version of deliverable D6.1. However, to cover all bases we have included all realistic scenarios below. Blockchain components integrating with Pilot 3 components have also been discussed but are not included in

1. BOI On-Premise Testbed Scenario

Two (2) database and ingestion testbed environments could be utilised in BOI. The Enterprise Data Warehouse (EDW) and the Enterprise Data Lake (EDL). Both environments are on Teradata Hardware appliances on premise in BOI.
The Enterprise Data Warehouse (EDW) is on the Teradata EDW platform which is purpose build to support Teradata Database.
The Enterprise Data Lake (EDL) is a Cloudera Enterprise Data Hub. This is built to Teradata Intelliflex platform.

Detailed Technical Specification re. software and hardware are available from Teradata or Cloudera websites.
Diagrams of Architecture tiering and components are described below (Figure 5):

a) Enterprise Data Warehouse Architecture

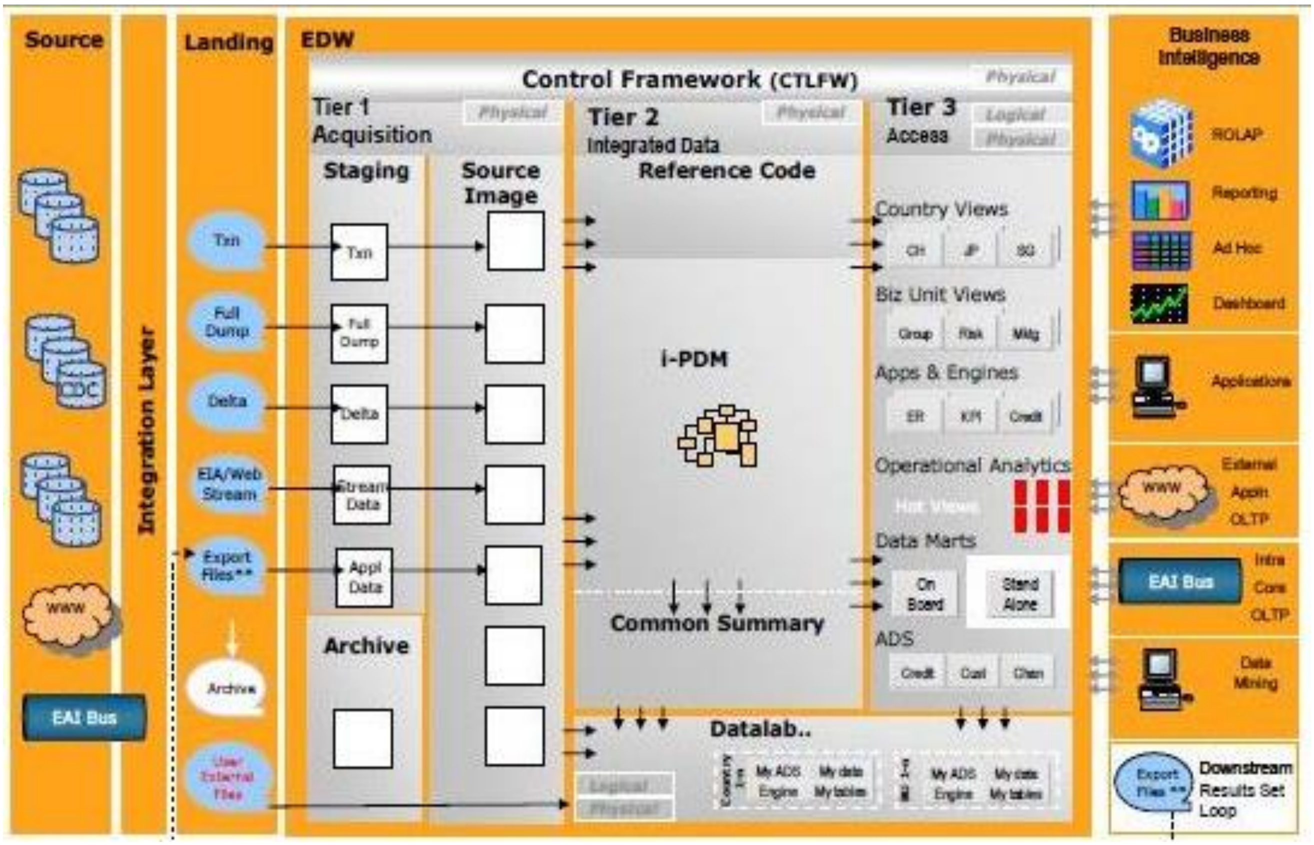

Figure 5 - BOI Testbed Enterprise Data Lake Architecture Scenario - Reference Architecture

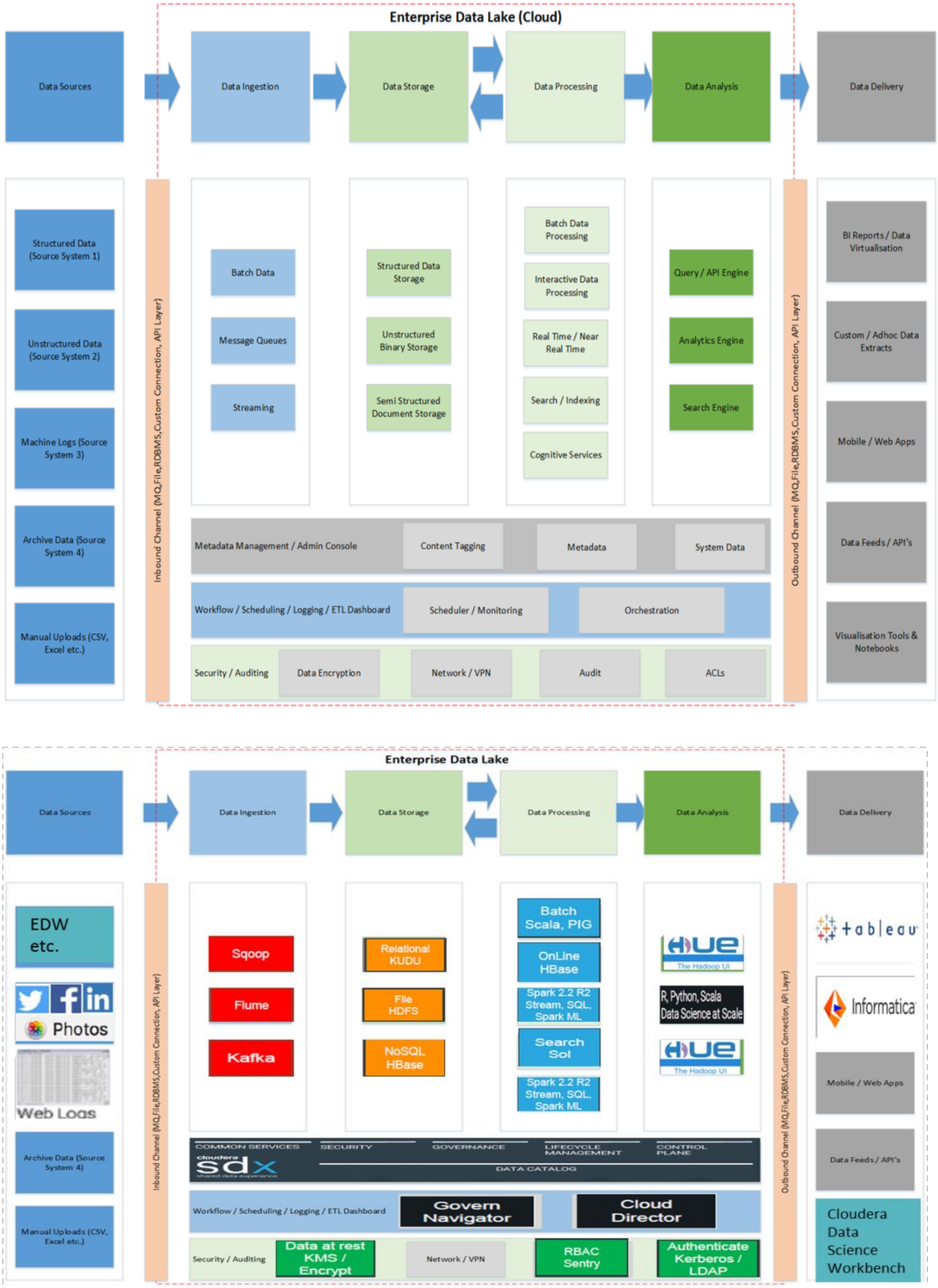
Enterprise Data Lake (Cloud)
Data Sources
Data Ingestion
Data Storage
Data Processing
Data Analysis
Data Delivery
Structured Data (Source System 1)
Unstructured Data (Source System 2)
Machine Logs (Source System 3)
Archive Data (Source System 4)
Manual Uploads (CSV, Excel etc.)
Inbound Channel (MQ,File,RDBMS,Custom Connection, API Layer)
Batch Data
Message Queues
Streaming
Structured Data Storage
Unstructured Binary Storage
Semi Structured Document Storage
Batch Data Processing
Interactive Data Processing
Real Time / Near Real Time
Search / Indexing
Cognitive Services
Query / API Engine
Analytics Engine
Search Engine
Outbound Channel (MQ,File,RDBMS,Custom Connection, API Layer)
BI Reports / Data Virtualisation
Custom / Adhoc Data Extracts
Mobile / Web Apps
Data Feeds / API's
Visualisation Tools & Notebooks
Metadata Management / Admin Console
Content Tagging
Metadata
System Data
Workflow / Scheduling / Logging / ETL Dashboard
Scheduler / Monitoring
Orchestration
Security / Auditing
Data Encryption
Network / VPN
Audit
ACLs
Enterprise Data Lake
Data Sources
Data Ingestion
Data Storage
Data Processing
Data Analysis
Data Delivery
EDW etc.
Photos
Web Logs
Archive Data (Source System 4)
Manual Uploads (CSV, Excel etc.)
Inbound Channel (MQ,File,RDBMS,Custom Connection, API Layer)
Sqoop
Flume
Kafka
Relational KUDU
File HDFS
NoSQL HBase
Batch Scala, PIG
OnLine HBase
Spark 2.2 R2 Stream, SQL, Spark ML
Search Sol
Spark 2.2 R2 Stream, SQL, Spark ML
HUE The Hadoop UI
R, Python, Scala Data Science at Scale
HUE The Hadoop UI
Outbound Channel (MQ,File,RDBMS,Custom Connection, API Layer)
tableau
Informatica
Mobile / Web Apps
Data Feeds / API's
Cloudera Data Science Workbench
COMMON SERVICES
SECURITY
GOVERNANCE
LIFECYCLE MANAGEMENT
CONTROL PLANE
cloudera SDX shared data experience
DATA CATALOG
Workflow / Scheduling / Logging / ETL Dashboard
Govern Navigator
Cloud Director
Security / Auditing
Data at rest KMS / Encrypt
Network / VPN
RBAC Sentry
Authenticate Kerberos / LDAP

Figure 6 - BOI Testbed Enterprise Data Lake Architecture - Tools Overlay

b) API Infrastructure

BOI Strategic API tooling is Mulesoft Anypoint Platform. MuleSoft provides the most widely used integration platform (Mule ESB & CloudHub) for connecting SaaS & enterprise applications in the cloud and on-premise. There is a lot of functionality in this platform. Please see website for functional detail and below for target BOI architecture in Figure 7:

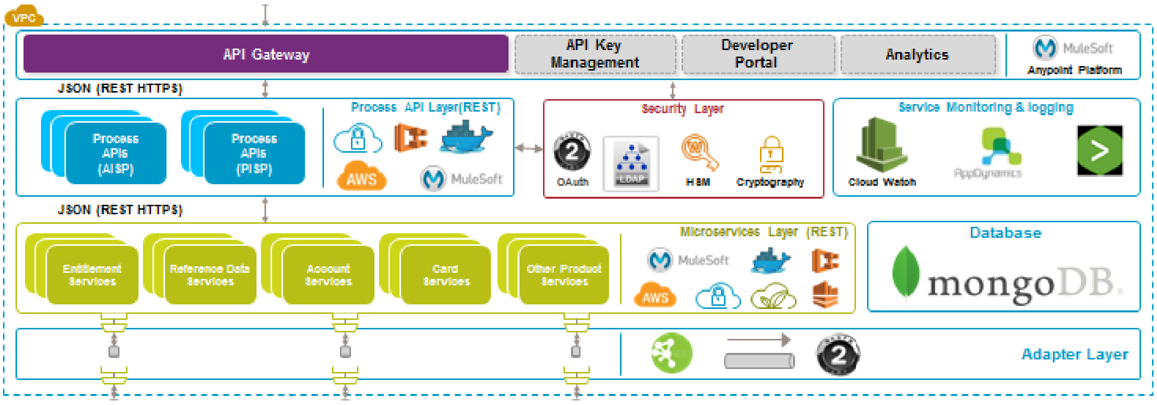

Figure 7 - BOI Testbed Target Architecture

c) User experience platform

BOI uses Temenos UXP which integrates other services.

2. Cloud Testbed Scenario

The current plan is to utilise Amazon Services. No decision on exactly which service will be utilised. The following are possible services utilise

<u>Amazon Relational DB, Key Value and In memory Databases being considered</u> -

- Amazon Aurora
- Amazon RDS
- Amazon Redshift
- Amazon Dynamo DB
- Amazon Elastic Cache

Also, Open Source database on Amazon being considered.

- My SQL
- Postgre SQL
- Cassandra
- Mongo DB (For JSON format data)

Currently there is no requirement for document or Graph DBs currently.

Amazon compute, containers. Integration and application development services being considered**Amazon EC2** - EC2 virtual machines gives you control of your server clusters and provide a broad range of customization options.

- **AWS Lambda** - Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume
- **Amazon ECR** - ECR compresses and encrypts your container images, making them fast to start and available to run anywhere.
- **Amazon ECS** - Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that provides the most secure, reliable and scalable way to run containerized applications.
- **Amazon EKS** - Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed Kubernetes service that provides the most secure, reliable, and scalable way to run containerized applications using Kubernetes.
- **AWS Fargate** -AWS Fargate is a serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Fargate removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation by design.
- **AWS Amplify** – Amplify provides a set of features such as Auth, Analytics, API (GraphQL and REST), Interactions, Predictions, PubSub, Storage, Push Notifications, and XR that enable you to build feature rich serverless applications powered by AWS services.
- **AWS AppSync** – Appsync simplifies application development by letting you create a flexible API to securely access, manipulate, and combine data from one or more data sources. AppSync is a managed service that uses GraphQL to make it easy for applications to get exactly the data they need.
- **Amazon API Gateway** - Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.

Example for possible configuration of services below in Figure 8:

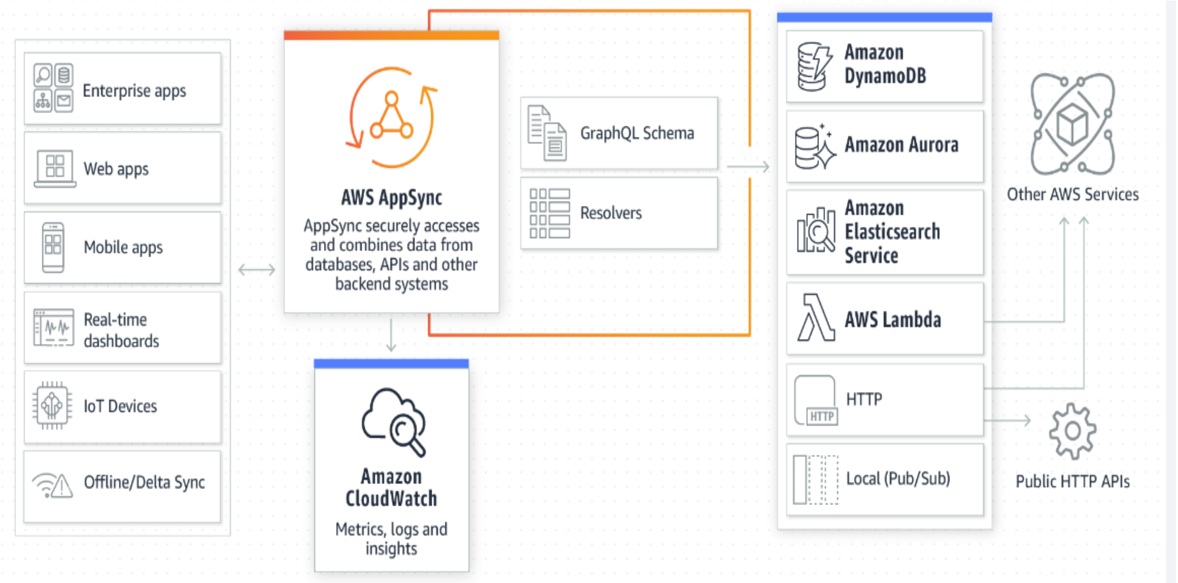

Figure 8 – BOI Testbed Cloud Scenario

3.3.1.3 Sandboxes

No existing sandboxes will be utilised. The mobile application front end will effectively be a sandbox for user interaction and user experience testing. As there may be API sandboxes to support integration with other pilots or other INFINITECH participants.

3.3.1.4 Datasets

The data sets used in the Pilot will be small and manually developed. The structure of the model and interoperability is more important than the content, but content will be mocked up to support use case and data flows.

1. They can be updated, but any history logs will be read only. Any updates will be through APIs or application front end. Data sets are already submitted above.
2. They will be available for use if required.
3. Depends what you classify as HTAP? and why must it be migrated?
4. Different participants in ecosystem may have different needs and may what to review different architectures.
5. Depends on the need of the central repository users. For POC purpose batch might be easier, unless someone wants to consume real-time.

3.3.1.5 Compliance Requirements

The compliance requirements already provided as part of deliverable *D2.7 - Security and Regulatory Compliance Specification.*

3.3.1.6 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment.

3.3.2 Testbed for Personalized Portfolio Management ("Why Private Banking cannot be for everyone?")

3.3.2.1 Testbed Brief Description:

Testbed for **Personalized Portfolio Management ("Why Private Banking cannot be for everyone?")** will host **Pilot 4** of INFINITECH Project that leader is **PRIVE (PRIVE Technologies – Austria).**

The main goal of Pilot 4 Personalized Portfolio Management is to develop and adapt within Privé Managers Wealth Management Platform an Optimization algorithm (further on called Privé Optimizer "AIGO"), as well as improving and expanding its capabilities as an artificial intelligence engine to aid 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.

3.3.2.2 Testbed Functional & Technical Specification

a. Functional Specification

Starting from a client's cash pool or current investments/portfolios, the user will select the fitness factors and constraints or preferences to perform the portfolio construction, basing themselves on the client's risk profile and his preferences. The tool will run on a preset universe of assets taking into account all the input data and constraints. The AI genetic algorithm will generate a new proposal, where the selected preferences and risk parameters have been recognized. The optimization tool can be run multiple times, after the necessary changes in initial parameters are made. This process can result in a UI proposal or a PDF generated investment proposal. The following diagram (Figure 9) describes the process workflow:

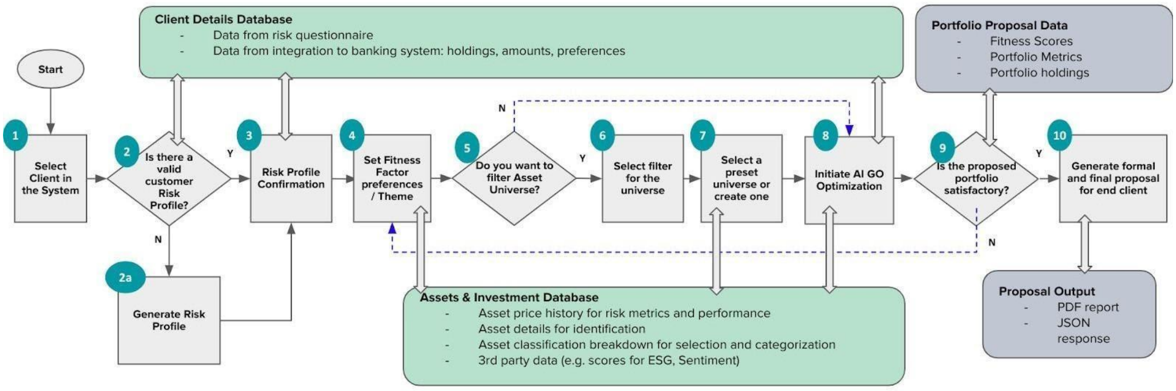
Client Details Database
- Data from risk questionnaire
- Data from integration to banking system: holdings, amounts, preferences
Portfolio Proposal Data
- Fitness Scores
- Portfolio Metrics
- Portfolio holdings
Start
1
Select Client in the System
2
Is there a valid customer Risk Profile?
Y
N
2a
Generate Risk Profile
3
Risk Profile Confirmation
4
Set Fitness Factor preferences / Theme
5
Do you want to filter Asset Universe?
N
Y
6
Select filter for the universe
7
Select a preset universe or create one
8
Initiate AI GO Optimization
9
Is the proposed portfolio satisfactory?
Y
N
10
Generate formal and final proposal for end client
Proposal Output
- PDF report
- JSON response
Assets & Investment Database
- Asset price history for risk metrics and performance
- Asset details for identification
- Asset classification breakdown for selection and categorization
- 3rd party data (e.g. scores for ESG, Sentiment)

Figure 9 - PRIVE Optimizer “AIGO” Process Flow Diagram

b. Technical Specification

● Hardware Requirements

- *3 VM instances, each with the following: CPU: Intel Xeon 3 GHz or faster*
- *Core: minimum 2 Core 4 threads*
- *Memory: 32 GB DDR4 1600 or 1866*
- *Hard Disk: 16 GB SSD*

The following diagram (see Figure 10) describes the Technical Architecture of the Testbed that will be available from PRIVE:

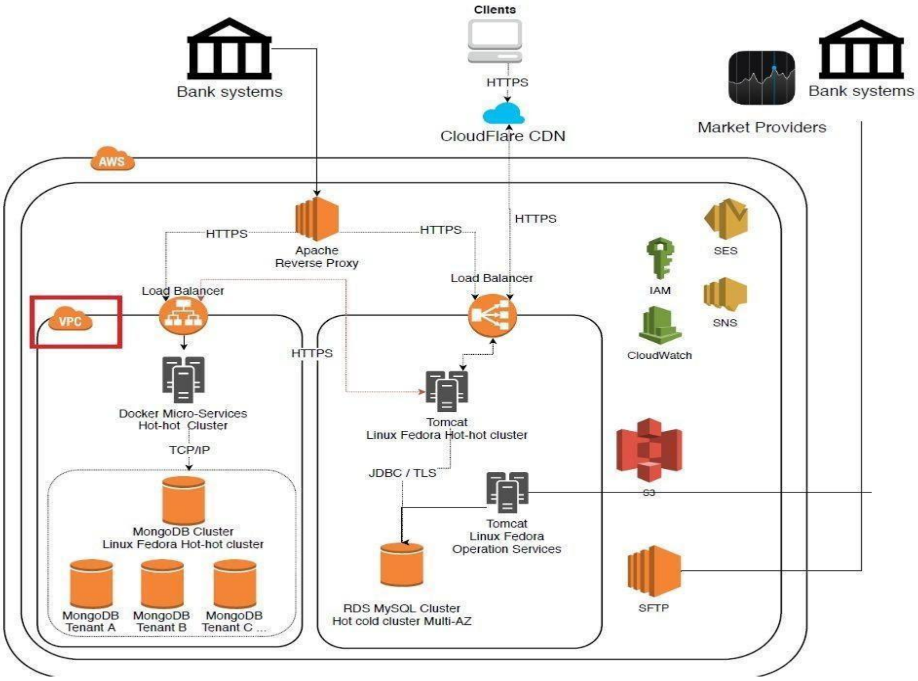

Figure 10 - PRIVE Testbed Technical Architecture

- **Software Requirements**

a. *Operating System / Application Software / Layered Software*

The SaaS platform runs in multiple data centres with active-active setup to achieve high availability. Privé has the following environments: DEV, SIT, UAT and PROD. Data can be transferred via SFTP, FIX or API. Most Privé APIs are REST, but SOAP and GraphQL are also supported. The architecture is based on microservices, using the following specification:

- Operating System: Ubuntu 18.04 LTS
- Framework: SpringBoot: 2.2
- Application Server: Tomcat: 7.0.103
- Database: MySQL: 5.6.47
- Database: MongoDB: 3.6
- Language Runtime: Java: OpenJDK 8u242

b. *Development Platform*

Html5/ReactJS for frontend. Our platform is written in Java, with Spring MVC, Spring boot, and hosted with Apache Tomcat.

3.3.2.3 Sandboxes

There may not be any sandbox as Pilot will be running on the PRIVE Infrastructure on Amazon Web Services.

Interested third parties or project partners can access the AI GO functionalities also via APIs, which can be provided on request.

3.3.2.4 Datasets

The imported datasets from potential bank-partners or customers are in normal setups read-only. An update of these datasets is always imported overnight in a daily data upload process. This ensures that the AIGO functionality within Privé Managers platform and also within other API integrations has every day the most current portfolio data of serviced customers available.

The datasets will be maintained on-premise, as well as the Datalakes/Data Warehouses that will be used for the specific Testbed. There is no data sync, because there is no intraday sync, as already described a daily update process is performed on a daily basis.

The datasets to be used by this pilot will be Customer Transactions Data, Financial Market Price Data, Financial Market Asset Master Data, Customer Risk Profile Data, Mutual Fund, ETF and Structured Products Breakdown Data, Customer Economic Outlook as well as single Account & Investors Data. All datasets will be stored within the 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. The following list describes in more detail the data that will be used from Privé Optimizer or “AIGO” during processing:

- Data from Risk Questionnaire (Risk Classification)
- Data from integration to banking system: holdings, amounts, preferences.
- Customer Risk Profile. Set Fitness Factor Preferences/Theme.
- Asset Price History for Risk Metrics and Performance.
- Asset Details for Identification.
- Asset Classification Breakdown for Selection and Categorization.
- 3rd Party Data (eg. Scores for ESG, Sentiment,)
- Customer Transactions Data fetched directly from the Bank or an Asset Manager.
- Financial Market Price Data fetched from several Market Data Providers.
- Financial Market Asset Master Data fetched from several Market Data Providers.
- Customer Risk Profile Data fetched directly from the Bank or an Asset Manager.
- Mutual Fund, ETF and Structured Products Breakdown fetched from several Market Data Providers.
- Customer Economic Outlook fetched directly from the Bank or an Asset Manager based on questionnaires and Customer Profiles.

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.

3.3.2.5 Compliance Requirements

To be compliant with all GDPR rules and conditions any user for testbed cases for the Privé AIGO module will have to sign a standard Mutual NDA (Draft can be provided on request) and for further use-cases with real-data (provided by a selected bank or partner) a SaaS license agreement or an API license agreement (depending on the technical implementation by the partner) shall be concluded beforehand, where a 3-months test-period can be agreed on.

3.3.2.6 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components’ deployment within a CI/CD environment. GitLab will be used to manage component’s code.

3.3.3 Testbed for Smart and Personalized Pocket Assistant for Personal Financial Management

3.3.3.1 Testbed Brief Description:

Testbed for **Smart and Personalized Pocket Assistant for Personal Financial Management** will host **Pilot 5a** of INFINITECH Project that leader is **LIB (Liberbank – Spain).**

Provide bank's customers with Smart services based on PFM categorization of historical account transactions.

- Smart alerts: prevent possible overdrafts
- Smart automations: identify recurrent payments
- Smart expense advisor: categories compared with other "similar" customers
- Smart recommendations of bank's products: only the ones relevant for each customer at each life moment
- Smart sentinel: provide customer protection based on alerting on potential anomalies associated with double payments, high value charges or bills, geographic impossible transactions

3.3.3.2 Testbed Technical Specification

a. Technical Specification

- **Hardware Requirements**
  - **Bastion Machine** – Standard machine with external connectivity, this machine is used for infrastructure administration. (vCPU: 1, RAM: 2GB)
  - **Kubernetes Cluster** – The Kubernetes API is not exposed outside the TestBed, it must be managed through the Bastion Machine.
    - Masters – 3 Nodes – If High availability is required.
      The number of worker nodes will determine master node sizes. For 5 Kubernetes worker nodes.

Nodes	RAM(GB)	CPU	DISK
3	8	2	SSD 500GB

- Worker Nodes – 5 - CPU: 8 RAM: 32GB

Nodes	RAM(GB)	CPU	DISK
5	8	32	SSD 500GB

- **Network attached Storage**
  - 2 x 1T B
- **Load Balancer**

  With a multi-master scenario, a Load Balancer between worker nodes and master nodes is required

  For a High-performance NGINX setup, 1,000,000 Requests per Second hardware recommendation is:

Nodes	RAM(GB)	CPU	DISK
2	8	32	1TB

- **Software Requirements**

  - **Operating System**: Ubuntu 16.04+ (all nodes)
  - **Kubernetes**
    - The master nodes are composed of four basic services:
      - **kube-apiserver** – Exposes a REST API for cluster management and communication between nodes.
      - **Etcd** – Key-value database where all the configuration and deployment instructions are stored.
      - **kube-scheduler** – Decides which node the resource goes.
      - **kube-controller-manager** – Ensures that the state of the cluster matches the configuration in etcd.
    - **kube-dns** is the internal DNS service for Kubernetes.
    - **Worker Nodes**: The container runtime (e.g. Docker), the kubelet, and cAdvisor.

  - **Load Balancer:** For setting up a HA Load Balancer NGINX Plus could be used
  - **Service Mesh:** Istio makes it easy to create a network of deployed services with load balancing, service-to-service authentication and monitoring
    - Automatic load balancing for HTTP, gRPC, WebSocket, and TCP traffic.
    - Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection.
    - A pluggable policy layer and configuration API supporting access controls, rate limits and quotas.
    - Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress.
    - Secure service-to-service communication in a cluster with strong identity-based authentication and authorization.

  - **Kubernetes package manager:** Helm
    - **Helm** Charts help you define, install, and upgrade even the most complex Kubernetes application.
      - Prometheus
      - Grafana

- Confluent Kafka Operator
- Spark Operator
- ElasticSearch operator
- Fluentd

- **GitOps**

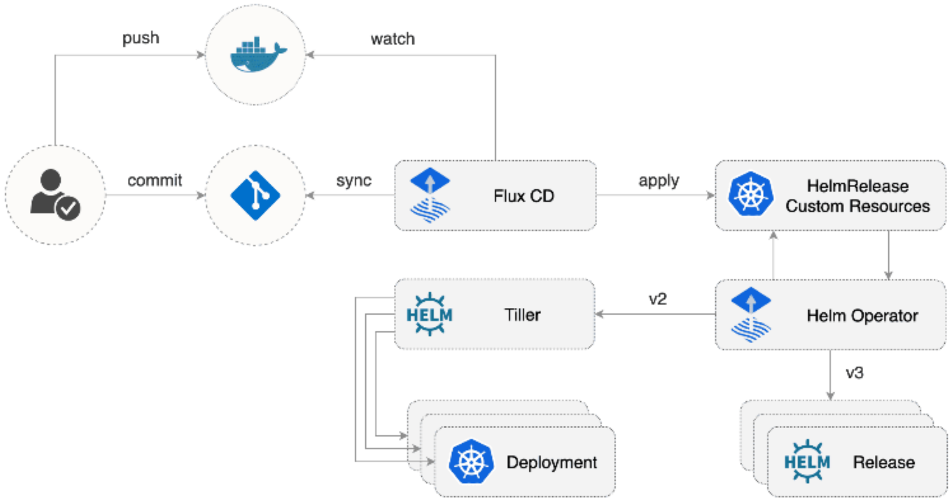

Figure 11 – Liberbank Testbed Technical Deployment Architecture

Flux monitors all of the container image repositories that you specify. It detects new images, triggers deployments, and automatically updates the desired running configuration of your Kubernetes cluster—and does so within the bounds of a configurable deployment policy.

- You declaratively describe the entire desired state of your system in git.
  - This includes the apps, config, dashboards, monitoring and everything else.
- What can be described can be automated.
  - Use YAMLs to enforce conformance of the system.
  - You don't need to run kubectl, all changes go through git.
  - Use diff tools to detect divergence between observed and desired state and get notifications.
- You push code not containers.
  - Everything is controlled through pull requests.

3.3.3.3 Sandboxes

There is no existing sandbox.

3.3.3.4 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

3.3.3.5 Compliance Requirements

Testbed & Sandbox environments will follow the Compliance & Regulatory Requirements already described in deliverable *D2.7 - Security and Regulatory Compliance Specifications.*

3.3.3.6 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.3.4 Testbed for Business Financial Management (BFM) tools delivering a Smart Business Advise

3.3.4.1 Testbed Brief Description:

Testbed for **Business Financial Management (BFM) tools** will host **Pilot 5b** of INFINITECH Project that leader is **BOC (Bank of Cyprus – Cyprus).**

The objective is to build AI powered Business Financial Management (BFM) tools that assists Small Medium Enterprises (SMEs) clients of the Bank of Cyprus (BOC) in managing the SMEs financial health. In particular, the BFM tools are expected to provide assistance with the SMEs activities in the area of cash flow management, continuous spending/cost analysis, budgeting, revenue review, or VAT provisioning.

The main components that will be part of this pilot are:

- Data Collectors (Internal from BOC Real-Time & Historical, External) for collection and transformation.
- Data Management (Data Repository, Data Streaming).
- Gateway for IRA data access services.

3.3.4.2 Testbed Functional & Technical Specification

Testbed does not exist at the moment.

a. Functional Specification

To support the above stated objective Testbed brief description several engines are planned to be developed such as:

- a Cash Flow (CF) Prediction engine delivering holistic Cash Flow management experience and actionable advice,
- a Budget Prediction engine that allows setting easily budget targets through the provision of suggested target values as well as simple budget monitoring,
- a KPI engine leading to valuable insights on the SME financial health and performance,
- a transaction monitoring engine that watches out for potential anomalies and savings.
- a Transaction Categorization delivering powerful auto-categorization to tag income and expense transactions by utilizing a continuous learning mechanism,
- a Invoice Processing engine that generates meaningful invoice background info to other components (e.g. Cash Flow Prediction) and SMEs. This applies if respective data can be obtained,
- a Benchmark engine supporting comparisons to other SMEs with similar profiles and
- a Recommender engine generating actionable insights.

BOC will provide business requirements, UPCR develop the respective AI models and GFT/Consortium partners deploy respective coding/integration. A Functional Specification Diagram (Figure 12 below) is following:

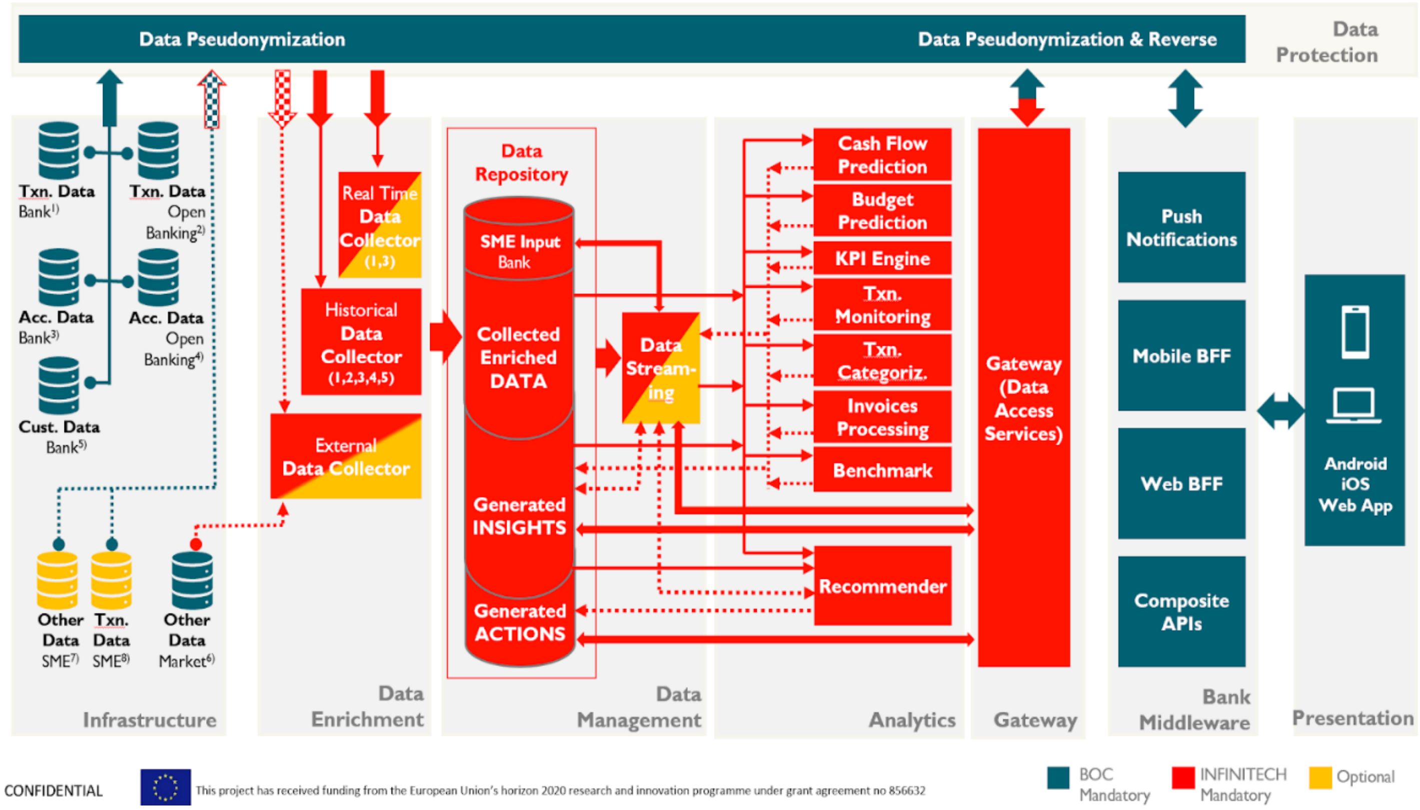

Figure 12 - BOC Testbed Functional Specification Diagram

b. Technical Specification

- **Hardware Requirements**
  - **AI models:** 8-Core CPU, 32 GB RAM and a GPU with 8 (preferably 16) GB RAM with the ability of GPU-enabled instances for deploying deep learning models.
  - **LeanXcale**: To be defined on later stage and confirm in the next version of deliverable.
  - **GFT**: To be defined on later stage and confirm in the next version of deliverable.
  - **Other**: To be defined on later stage and confirm in the next version of deliverable.
  - **Relative cloud configuration:**
    - ○ **GFT**: Cloud Design to be provided and deployed.

- **Software Requirements**
  - **GFT:** TBD.

3.3.4.3 Sandboxes

There is no existing sandbox.

3.3.4.4 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

3.3.4.5 Compliance Requirements

BOC is the end user for this pilot, consequently as per the banks procedures and policies as well as respective regulation related confidentiality and security procedures must be followed with no exceptions and respective agreements be in place.

For the above reason the pilot will be developed on a dedicated infrastructure.

3.3.4.6 Additional Info

- **Kubernetes version:**
  TBD as per public cloud offering. To be defined on later stage and confirm in the next version of deliverable.

- **Source code management:**
  TBD as per public cloud offering. To be defined on later stage and confirm in the next version of deliverable.

- **Docker registry:**
  TBD as per public cloud offering. To be defined on later stage and confirm in the next version of deliverable.

- **CI/CD software:**
  TBD as per public cloud offering. To be defined on later stage and confirm in the next version of deliverable.

3.3.5 Testbed for Personalized Closed-Loop Investment Portfolio Management for Retail Customers

3.3.5.1 Testbed Brief Description:

Testbed for **Personalized Closed-Loop Investment Portfolio Management for Retail Customers** will host **Pilot 6** of INFINITECH Project that leader is **NBG (National Bank of Greece – Greece).**

The goal of this pilot 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).

Expected Outcomes

- BigData/AI system for personalized investment recommendations for the retail customers of the bank.
- Development of a closed-loop system that continuously learns, improves itself and provides better recommendations.
- The system shall improve productivity of investment consultants of the bank, through enabling them to access faster recommendations tailored to their retail customer needs.

3.3.5.2 Testbed Functional & Technical Specification

a. Functional Specification

Data from NBG Datawarehouse related to Investment Products 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.

Based on the Risk profile of each customer, the relative financial instruments mapping with each customer cluster profile, using the Personalised Investment Recommendation Engine, will be possible for each relationship manager, thru the relative visualisation application, to provide a personalized investment proposal for each individual customer that will be interested for investment. The personalized investment proposal will be refined based on news & social feed data as sentiment analysis data, in order to take in consideration in real time any factors that may affect the relative mixture or financial instruments. The investment proposal will provide all the alternative financial instruments that are suitable for the specific customer, with specific grading (percentage) that fulfil both customer's expectations, in combination with bank's risk profile analysis. The following diagram (Figure 13) depicts the Functional Specification flow described above:

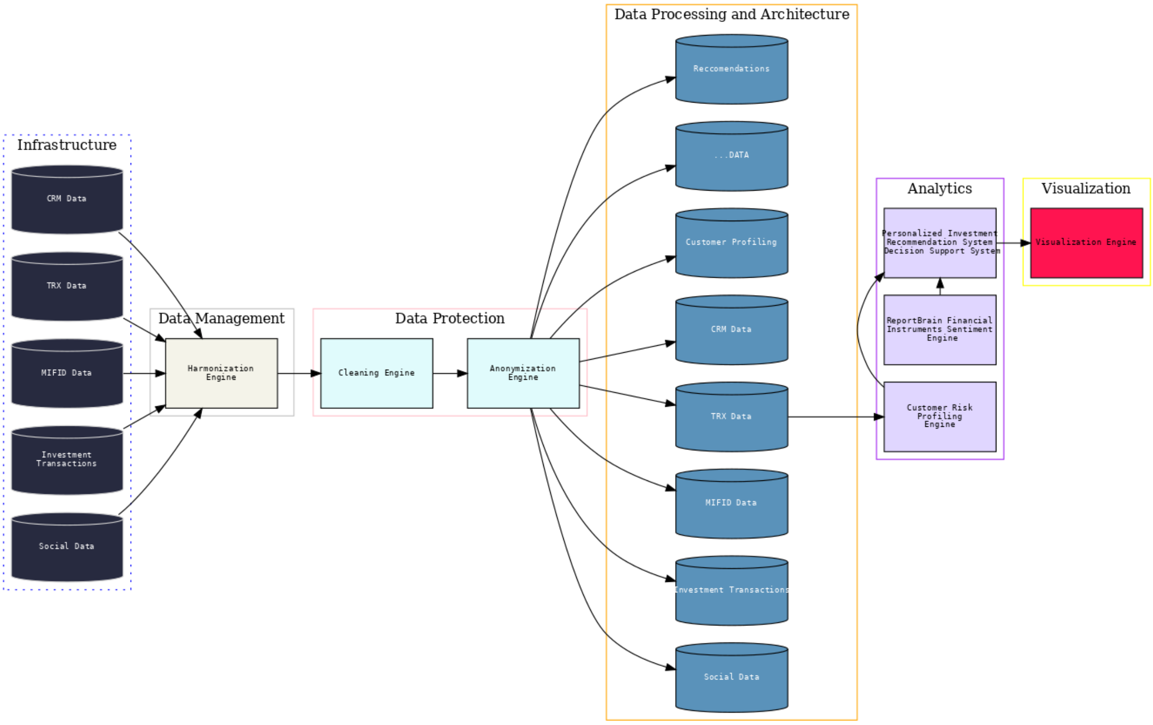

Figure 13 - NBG Testbed Functional Specification

b. Technical Specification

Component	vCPUs	RAM	Storage	Software
UBITECH Harmonization Tool	vCPUs: >= 4	>= 32GB	>= 250 GB	Java, Docker, Linux OS
UBITECH Cleansing Engine	vCPUs: >= 4	>= 32GB	>= 250 GB	Java, Docker, Linux OS
UBITECH Anonymization Engine	vCPUs: >= 4	>= 32GB	>= 250 GB	Java, Docker, Linux OS

LeanXcale Datastore (2 Data Nodes, 1 Query Data Node, 1 Metadata Node)	vCPUs: >= 4	>=8GB min >=32GB max	>=40GB	Docker, Linux OS
Visualization API & Application (2 Servers)	vCPUs: >= 4	>=16GB	>=50GB	Java Spring Boot, Angular, Express, Node JS, Docker, Linux OS
Retail Customers Risk Profiling	vCPUs: >= 4	>= 32GB.	>= 250 GB	Python Libraries
Personalized Investment Recommendations/ Decision Support System	vCPUs: >= 4	>= 32GB.	>= 250 GB	Python Libraries

3.3.5.3 Sandboxes

There is no existing sandbox.

3.3.5.4 Datasets

The initial population in which the Pilot will initially focus is about **150.000 customers**.

The following are the sizes:

- **Table of historical prices for 2 years**: 255 b per record on 1,745,281 entries = 246084621 b = 0.23 Gb
- **Deposit account trading table for 2 years**: 210 b per registration for 1,900,000 for 24 = 8.91 Gb
- **2-year card trading table**: 272 b per 1.2 million transactions per month for 24 = 7.3 Gb
- **2-year investment trading table**: 248 b per 50 thousand transactions per month for 24 months =
- **Table of investment products** is negligible size
- **CRM data table**: 351 b per 150,000 people = 0.05 Gb. For another varchar field above volume is 10 Mb, so as many CRM fields and add will stay below 0.5 GB.

3.3.5.5 Compliance Requirements

Testbed & Sandbox environments will follow the Compliance & Regulatory Requirements already described in deliverable *D2.7 - Security and Regulatory Compliance Specifications.*

3.3.5.6 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.4 Testbeds for Predictive Financial Crime and Fraud Detection

3.4.1 Testbed for Operation Whitetail – Avoiding Financial Crime

3.4.1.1 Testbed Brief Description:

Testbed for **Operation Whitetail – Avoiding Financial Crime** will host **Pilot 7** of INFINITECH Project that leader is currently FTS **(FUJITSU Technology Solutions – Germany)** as **Selected Bank** that the relative testbed will be hosted **is not currently confirmed.**

The Bank's customers make millions of transactions a day. The bank needs to be able to identify transactions that are moving the proceeds of crime (money laundering). Using advanced computing techniques could start creating a more up-to-date, accurate, versatile and complete view of the customer's profile and behaviour based on the near real-time analysis of transactions. This level of information could reveal abnormal behaviour earlier, and more accurately and enable the business to intercept criminal behaviour more effectively.

Within the pilot processes like

(i) Know Your Customer (KYC), where powerful processing capabilities will allow for screening the vast amount of available internal and external data sources (including next gen. information sharing capabilities) in near-real time, to ensure that KYC data is automatically updated to the most recent information available on the customer;

(ii) Customer risk profiling, based on feeding the transaction-based behavioural profile data (Know Your Transactions - KYT) and KYC results into an advanced financial crime risk model 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 are addressed.

The major modules, which are involved and need to be developed, are:

- Advanced KYC – Data Join and Enrichment in a new KYC data lake
- KYC Data Extraction – extracting the relevant customer data from the advanced KYC data lake
- KYT Transaction Pattern Extraction – Extracting the behavioural transaction pattern
- Advanced FC Risk Model – Scoring the Financial Crime (FC) risk
- FC Dashboard for project and testing purposes

The complete pilot is based on synthesized data (customer internal & external, transaction data). The relevant modules for generation of synthetic data need to be implemented yet.

3.4.1.2 Testbed Functional & Technical Specification

a. Functional Specification

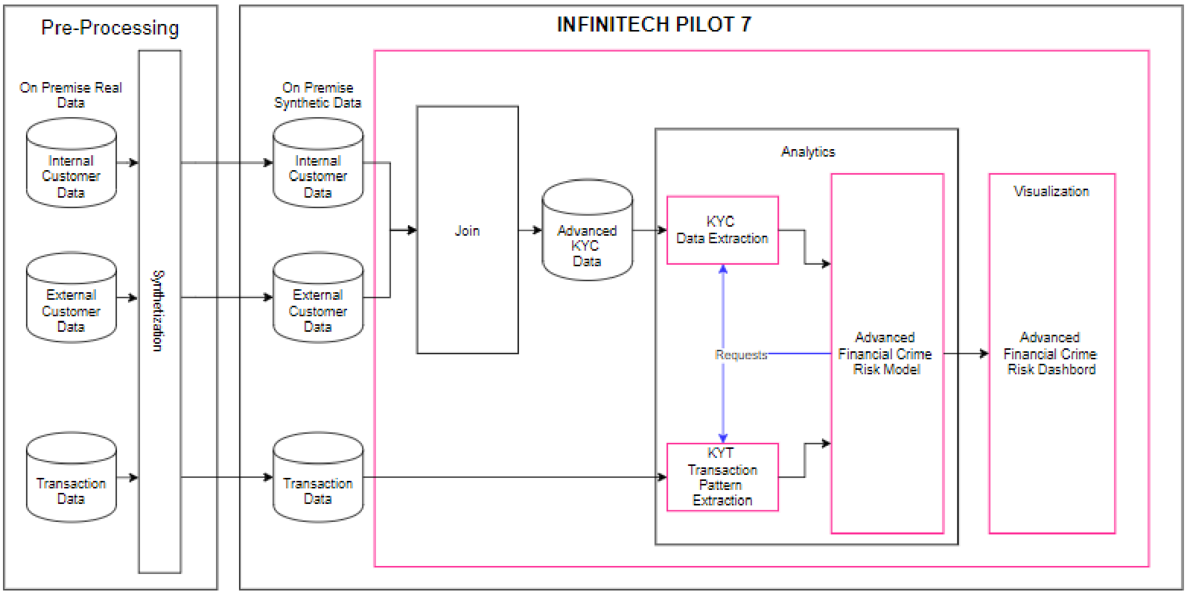

Figure 14 – FBK Testbed Functional Specification

b. Technical Specification

These details depend on the amount of data and the technologies, which will be applied in the pilot. Therefore, a detailed infrastructure specification will need updated information with the Selected Bank (when INFINITECH Consortium participation will be confirmed).

- **Hardware Requirements**

Not yet clearly defined, will be included in next version of deliverable D6.1

- **Software Requirements**

Not yet clearly defined, will be included in next version of deliverable D6.1

3.4.1.3 Sandboxes

There is no existing sandbox.

3.4.1.4 Datasets

The KYC data lake will be updated slowly. Only a few changes related to external sources are expected.

The synthetic KYT data shall be updated regularly for development and testing by generated synthetic transactions.

Most probably, the data will be hosted on premise at the selected bank due to bank's compliance requirements. Specific synthetic data set may be used for development of the KYC/KYT extraction and risk scoring models. Sharing of these data with the developing parties needs to be defined and will be included in the next version of deliverable D6.1.

3.4.1.5 Compliance Requirements

Considering the Ethics requirements, the pilot will utilize synthetic data. It is planned that synthesizing the data will be performed by the bank on premise. Therefore, no real data will be used in the pilot.

3.4.1.6 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.4.2 Testbed for Platform for Anti Money Laundering Supervision (PAMLS)

3.4.2.1 Testbed Brief Description:

Testbed for **Predictive Financial Crime & Fraud Detection - Platform for Anti Money Laundering Supervision (PAMLS)** will host **Pilot 8** of INFINITECH Project that leader is **BOS (Bank of Slovenia – Slovenia).**

The pilot will develop a platform with several functionalities/tools for improved AML supervision on a risk-based approach (see Figure 16 below):

- **Screening tool:** tool will enable to detect high-risk behaviour based on the automated screening of the relevant data, predefined scenarios and additional scenarios identified by the system itself.
- **Search Engine:** tool will enable a search for a specific transactions, pattern or financial institution (FI).
- **Risk Assessment tool:** tool will implement risk assessment methodology and enable automated calculation of the FI risk and sector risk based on timely and up to date data. Based on different triggers coming from internal and external sources (screening tool, distribution channel) risk assessment tool will be informed to update the existing risk assessment.
- **Distribution Channel:** third parties will send/upload data to the distribution channel tool. Received data (especially from the FI) will be automatically integrated into the risk assessment tool.

3.4.2.2 Testbed Functional & Technical Specification

a. Functional Specification

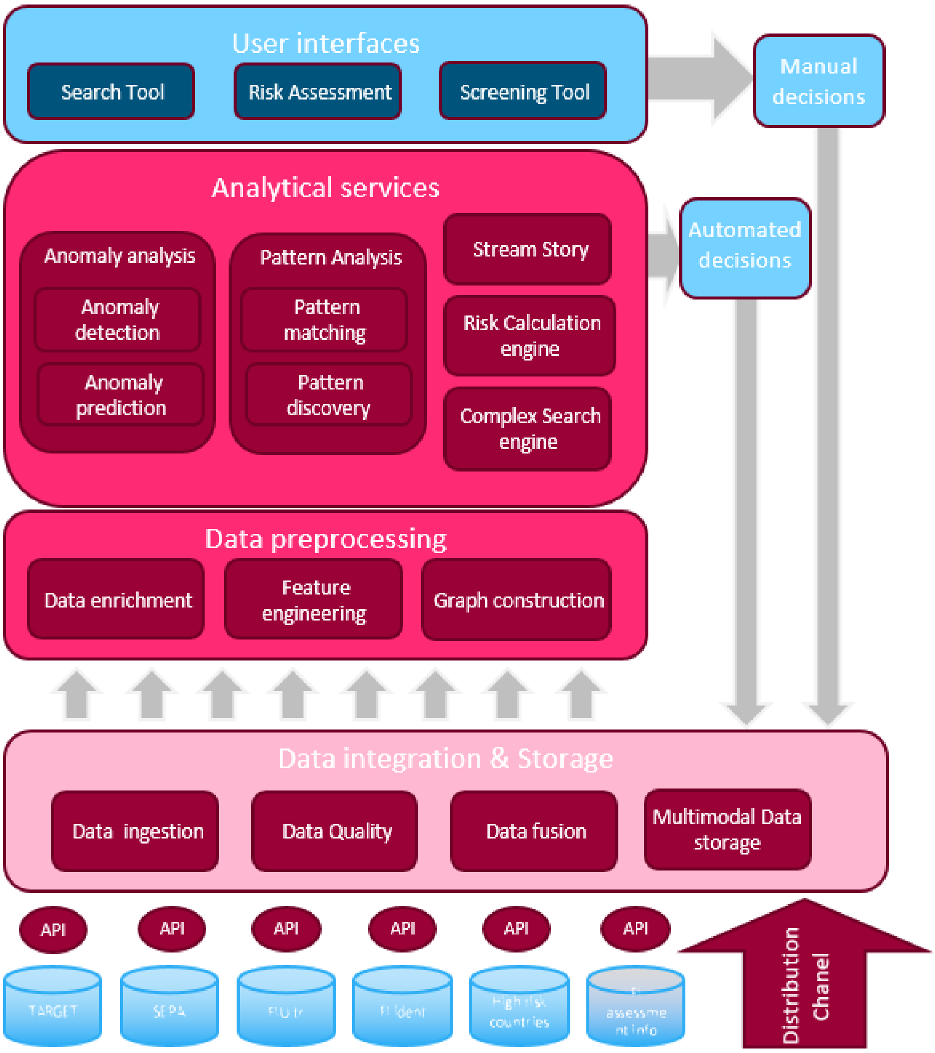

Figure 15 – BOS Testbed - PALMS high level architecture

PALMS High-Level architecture (see Figure 15 above) consist of 4 main layers, which corresponds to BDVA:

a) Data Integration and storage, which provides four main functionalities:
   a. Data Ingestion,
   - Data Quality
   - Data fusion and
   - provides Multimodal NSQL data storage;

b) Data Pre-processing, which provides specific data preparation for advanced analytics;

c) Data Analytics, which consist of several main components:
   - Anomaly analysis, which will provide functionalities for anomaly detection and prediction for time series data,
   - Pattern analysis, which will provide analytical services on data graphs, including detection of complex patterns on data graphs.,

- Stream story is 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 timeseries.
- Risk Calculation engine and Complex search services, which will be implemented specifically for Pilot8 requirements and therefore will be tailored to BOS specific and

d) User Interfaces.

b. Technical Specification

- **Hardware Requirements**

Hardware Description	
HP Z4 G4 WKS	
CPU:	Intel XeonW-2125 4.0 4C
RAM:	256GB (8x32GB) DDR4
Graphic:	NVIDIA Quadro P400 2GB (3)mDP Graphics
Disk:	Z Turbo Drv 1TB PCIe NVMe OPAL2 TLC SSD

- **Software Requirements**

Preliminary Software Requirements for PALMS platform:

- Linux (or Windows) operating system
- Libraries:
  - QMiner (machine learning / signal processing)
  - SNAP (graph analysis)
- External tools (candidates):
  - PostgreSQL
  - Elastic Search
- Programming tools:
  - C/C++ compilers (GNU or Microsoft)
  - Python
  - Node.js

3.4.2.3 Sandboxes

There is no existing sandbox.

3.4.2.4 Datasets

Dataset Name	Dataset (short) description
TARGET2/SEPA transactions	Data on international and domestic transactions including data on payer, payee, payment service provider, account no, amount, date
FIU transactions	Data on international transactions to high risk countries published by the Slovene Financial Intelligence Unit (FIU) including data on payer, payee, payment service provider, account no, amount, date
FI data	Financial Institution (FI) data: name, address, ID of the FI and statistic data about its inherent risk and control environment
eRTR	Slovenian Transactions Accounts Register (eRTR) IBAN, transaction account no.
ePRS	Slovenian Business Register (ePRS) Legal entity identification data: name, address, ID

3.4.2.5 Compliance Requirements

Pilot will develop Platform for AML Supervision for end user BOS, who is obliged to strict security protocols and confidentiality due to highly sensitive data therefore:

- cooperation between BOS and JSI (technical partner on the pilot) is defined in the NDA;
- PAMLS will be developed on the BOS premises and on a dedicated infrastructure;
- when required by respective laws data will be anonymized.

3.4.2.6 Additional Info

- Kubernetes version: Currently Pilot do not use Kubernetes.
- Source code management: Use public and private repositories on GitHub.
- Docker registry: Do not use proprietary docker registry; Usage of Docker's Hub.
- CI/CD software: Usually usage of custom installations of Jenkins and utilize GitHub Actions.

3.4.3 Testbed for Analysing Blockchain Transaction Graphs for Fraudulent Activities

3.4.3.1 Testbed Brief Description:

Testbed for **Fraudulent Blockchain Transactions Detection** will host **Pilot 9** of INFINITECH Project that leader is **AKTIF (Aktifbank – Turkey).**

The current existing testbed functionality includes only data collection from the blockchains. Raw blockchain data from Ethereum mainnet is available. A bitcoin node is also run in order to sync bitcoin data. A blacklist of blockchain addresses manually compiled from the Internet is maintained.

3.4.3.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Dept. of Computer Eng. Bogazici University Istanbul, Turkey

b. Testbed Location info:

Ethereum raw blockchain data is stored on Amazon cloud on EBS storage as part of nano machine instances. A local node at Dept. of Computer Eng., Bogazici University is also running Bitcoin node in order to sync bitcoin blockchain data

3.4.3.3 Testbed Functional & Technical Specification

a. Functional Specification

The following diagram (see Figure 16 below) is the overall data flow diagram of the system that will be developed in INFINITECH project. Currently, the top raw data retrieval modules are available for Ethereum. The transaction analysis system in the lower part will be built in INFINITECH project.

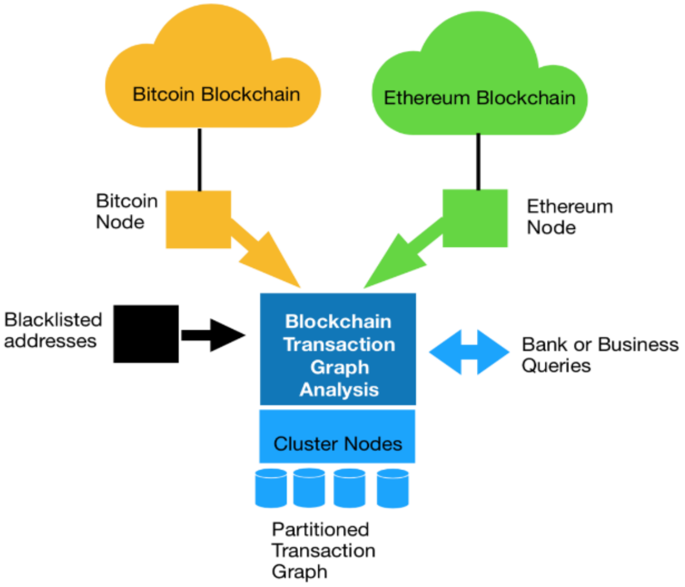

Figure 16 – AKTIF Testbed – Functional Specification Diagram

b. Technical Specification

Technical Architecture Diagram (see Figure 17 below) for the system that will be implemented in INFINITECH Project:

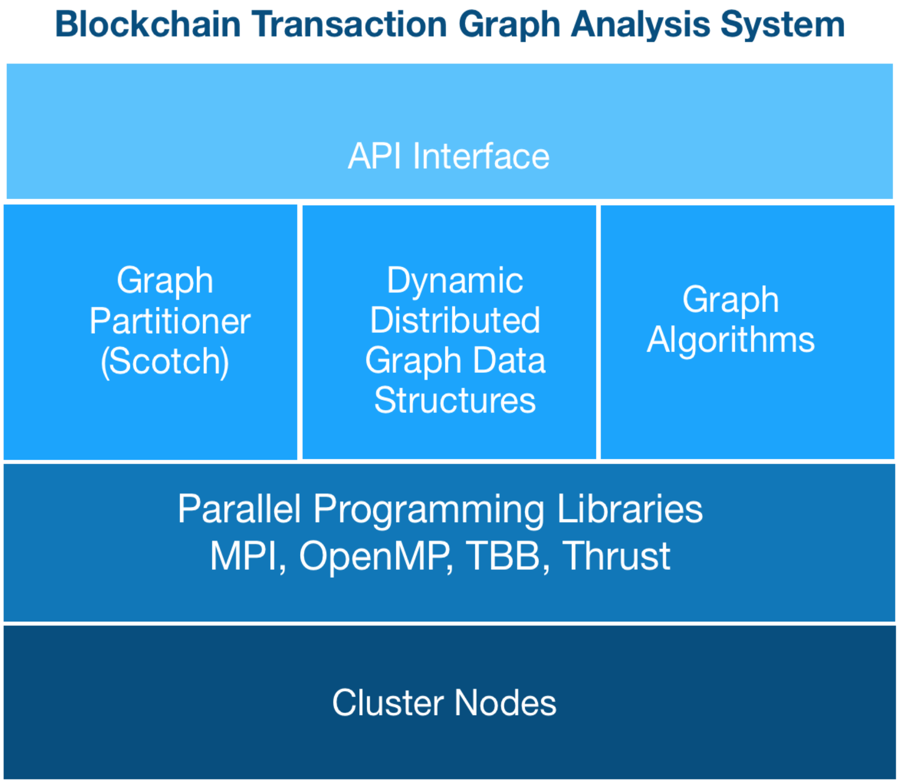
Blockchain Transaction Graph Analysis System
API Interface
Graph Partitioner (Scotch)
Dynamic Distributed Graph Data Structures
Graph Algorithms
Parallel Programming Libraries MPI, OpenMP, TBB, Thrust
Cluster Nodes

Figure 17 - AKTIF Testbed – Technical Architecture Diagram

- **Hardware Requirements**

Cluster with at least 8 to 16 nodes with extra-large memory capacity (similar to Amazon large instances). Installation will be able to partition the transaction graph and scale the number of nodes used.

- **Software Requirements**

The following operating system and open source software will be used:

- Linux operating system
- Scotch graph partitioning software
- GNU C/C++ compilers supporting OpenMP and TBB, Thrust libraries may also be needed (for shared memory programming)
- MPI Message passing libraries (for distributed memory programming)
- Python
- Node.js
- Bitcoin client (parity-bitcoin)
- Ethereum geth and/or parity clients

3.4.3.4 Sandboxes

There is no existing sandbox.

3.4.3.5 Datasets

The following data is collected by pilot partner, Bogazici University (BOUN).

- Raw blockchain data from Ethereum mainnet is available.
- A bitcoin node is also run in order to sync bitcoin data.
- A blacklist of blockchain addresses manually compiled from the Internet is maintained
- Ethereum contract addresses of popular ERC20 tokens that will be used to extract ERC20 token transfer transactions.

3.4.3.6 Compliance Requirements

Since Ethereum and Bitcoin blockchain data that is used in the testbed is publicly available, no compliance requirement is needed from other INFINITECH partners that will use the system.

3.4.3.7 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitHub will be used to manage component's code.

3.4.4 Testbed for Real-time cybersecurity analytics on Financial Transactions' BigData

3.4.4.1 Testbed Brief Description:

Testbed for **Real-time cybersecurity analytics on Financial Transactions' BigData** will host **Pilot 10** of INFINITECH Project that leader is **PI (Poste Italiane – Italy).**

With regard to the Pilot #10 design and execution, the testbed definition (that is the setting of hardware resources, like Storage, Compute and Network…) aims to take into account the deployment of an instance of ALIDA asset (https://home.alidalab.it/), a micro-service based platform for data management and composition, deployment, optimisation, execution and monitoring of big data analytics data pipelines (covering ingestion, preparation, analysis and visualization), which has been developed by ENG in previous and ongoing research activities.
The recommended set of resources needed is described in the following picture (see Figure 18):

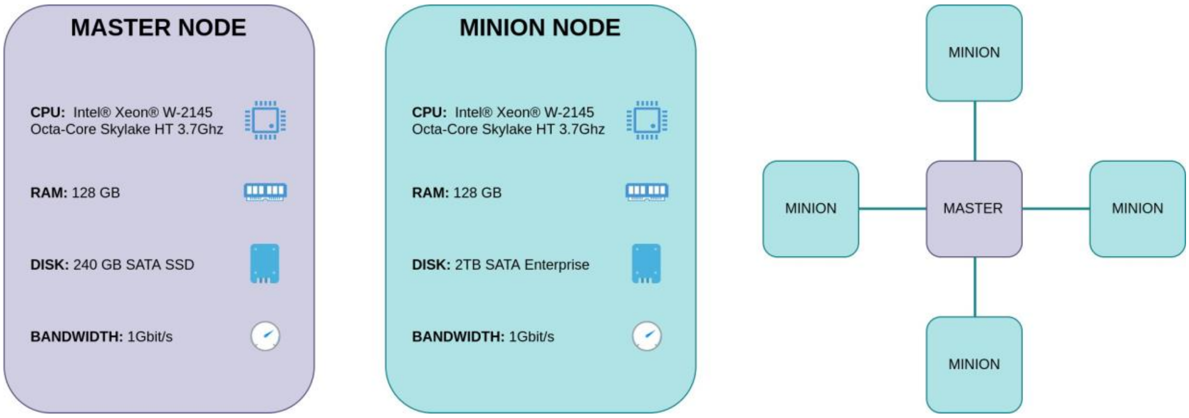

Figure 18 – PI/ENG Testbed Resources Diagram

The infrastructure setup consists in a single-master four-worker node running as Kubernetes on-premise cluster, each node has a wide enough set of allocable resources to run the testbed safely and without running into disk pressure and memory pressure issues for the expected workload. In any case, the system can be scaled both horizontally and vertically. The machines are equipped with 128 GB of RAM, 2TB of storage and one octa-core 3.7Ghz processor.

3.4.4.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

- Engineering – Palermo – ITALY

b. Testbed Location info:

- HETZNER Data Centre

3.4.4.3 Testbed Functional & Technical Specification

a. Functional Specification

The Testbed aims at supporting the following functionalities:

- Streaming and Batch data pipelines processing
- Catalogue of Big Data Analytics services (covering ingestion, preparation, analysis, visualization) for various data analytics scenarios
- Graphical editor for building data pipelines
- Data pipeline deployment by means of modern resource orchestrators such as Kubernetes
- Data pipeline execution monitoring
- Data visualization customization through a set of graphs and query editor
- Graphical User Interface to easily create and redistribute new custom BDA services

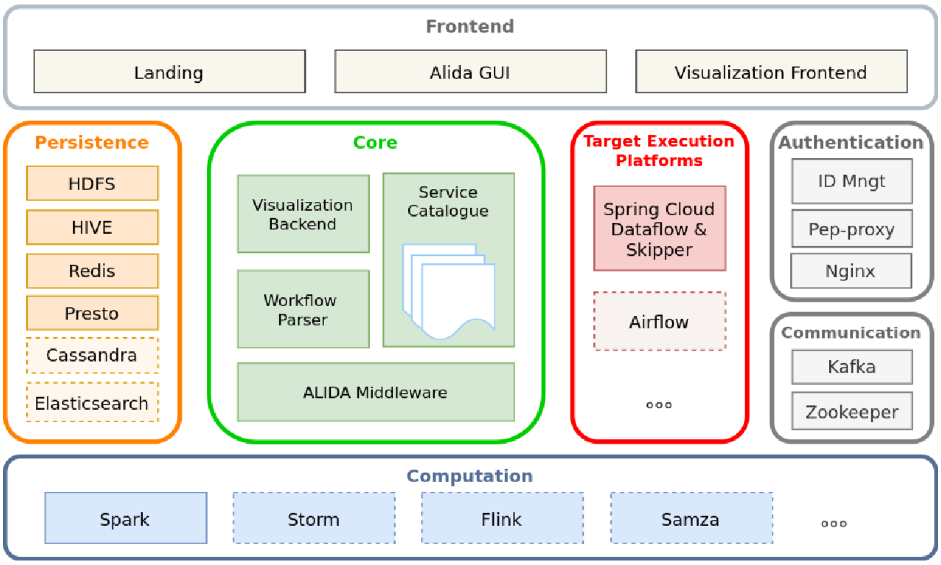

Figure 19 – PI/ENG Testbed Functional Specification Diagram

To meet this goal, an ALIDA instance will be deployed. ALIDA is a micro-service-based platform for composition, deployment, optimisation, execution and monitoring of pipelines of Big Data Analytics (BDA) services.

It is assumed that some Regulatory Compliance Tools will be properly deployed into the testbed (for Pseudonymization, Discretionary Data Access Control, Encryption of data at rest and during transfers, Accountability, Auditability). In fact, even though the datasets that will be used in Pilot 10 are Fully Synthetic ones, therefore no specific regulation could be applied to them, we are considering our data as real.

b. Technical Specification

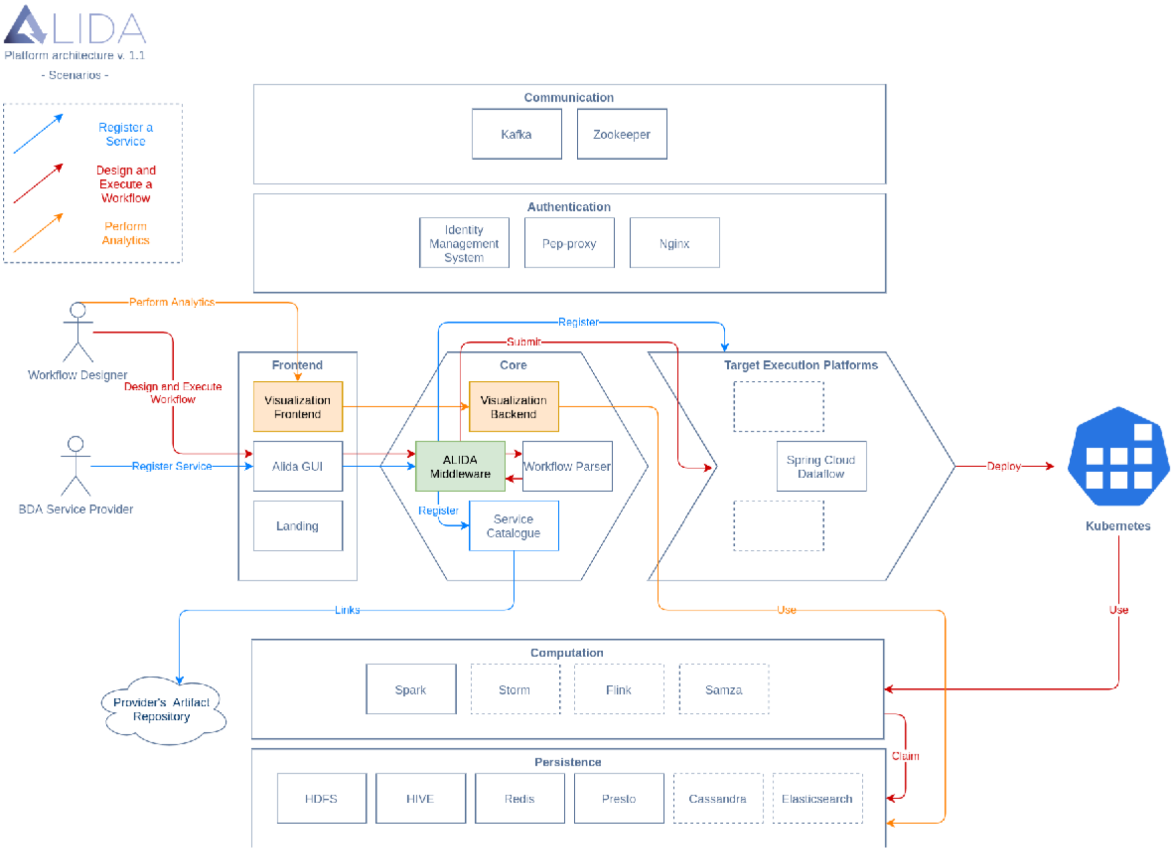

Figure 20 – PI/ENG Testbed Technical Specification Diagram

ALIDA presents a microservice architecture, where microservices are deployed in containers, whose management is largely simplified by Kubernetes, a container orchestrator which automates the deployment, management, scaling, and networking of containers.

Basically, the picture shows three flows: service registration (blue flow), data pipeline design and execution (red flow) and visualization (orange flow).

The key flow is the data pipeline design and execution: by means of the GUI user designs a data pipeline and executes it. In this case, the core components of ALIDA submit the request of execution and deployment of the pipeline to the candidate orchestrator. Currently just Spring Cloud Data Flow (https://spring.io/projects/spring-cloud-dataflow) is adopted as pipeline orchestrator in ALIDA.

Spring Cloud Data Flow (SCDF) is a Microservice based Streaming and Batch data processor that can make use of a variety of container orchestrators.

In the ALIDA platform, the SCDF engine instance uses an implementation of the Spring Cloud Deployer for Kubernetes. Then, Kubernetes uses the computation resources available into the platform and the persistence layer to manage data storage and workloads.

The platform currently supports several frameworks such as Spark, H2O, Flink, mainly used for BDA service development and registration. Other frameworks may be deployed, and relevant machine learning libraries available for Python can be used.

It is worth to notice that BDA services may or may not be implemented working on these frameworks. The only requirement that the BDA services must match is that they have to be implemented as a Spring Boot application shipped within a Docker image that can optionally contain a Python application.

In ALIDA, most of the BDA services developed are based on Spark, so they can process a large volume of data, in a distributed environment. Spark is used for both batch and streaming applications thanks to Spark Streaming.

With regard to the persistence layer, Hive and Redis play several roles.

Hive is used both to access the data resulting from the execution of the workflows through the visualization component, and to ensure that SPARK can write and read the datasets stored into HDFS.

Redis is used by ALIDA to store some properties related to the platform and to serve the visualization component with the aim to create graphs and visualizations with real time updating capabilities. Last but not least, Kafka is adopted on various fronts: the platform, the core component of ALIDA, uses it to transmit and update the properties during the workflow execution phases. SCDF itself exploits it in the management of streams pipelines.

ALIDA is cloud native software, this means that it can be seamlessly deployed both in on-premise environments and on the cloud, environments provisioned by the widely known providers such as Microsoft Azure, Amazon AWS and Google Cloud Platform.

To summarize, the software requirements to get the sandbox up and running are:

- Helm and Tiller 2.14+
- Kubernetes 1.14+
- To enable ingresses, a valid ingress provider is required, Traefik is recommended
- A DNS service provider is recommended to use ingresses with Traefik
- A persistent volume provisioner support in the underlying infrastructure

3.4.4.4 Sandboxes

There is no existing sandbox aτ the moment, but based on that ALIDA software will be the main software used

3.4.4.5 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

3.4.4.6 Compliance Requirements

Testbed & Sandbox environments will follow the Compliance & Regulatory Requirements already described in deliverable *D2.7 - Security and Regulatory Compliance Specifications.*

3.4.4.7 Additional Info

As already mentioned, the software requirements to get the sandbox up and running are:

- Helm and Tiller 2.14+
- Kubernetes 1.14+
- To enable ingresses, a valid ingress provider is required, Traefik is recommended

- A DNS service provider is recommended to use ingresses with Traefik
- A persistent volume provisioner support in the underlying infrastructure

The development process of the sandbox is supported by the following tools:

- Gitlab (source code versioning and CI/CD)
- Docker Registry
- Grafana as monitoring tool

3.5 Testbeds for Personalized Usage-Based Insurance Pilots

3.5.1 Testbed for Personalized insurance products based on IoT connected vehicles

3.5.1.1 Testbed Brief Description:

Testbed for **Personalized insurance products based on IoT connected vehicles** will host **Pilot 11** of INFINITECH Project that leader is **ATOS (ATOS – Spain)**.

The testbed will be focused on deriving driving profiles from captured driving routes and classifying different driving styles, according to inferenced profiles and related context information. Different ML/DL techniques will be used to define and manage these profiles, which require from routes storage capabilities to execute training, testing and inferencing processes. To capture car routes, that include technical connected-vehicles data (such as speed, acceleration, location, etc.) a connected car framework, based on FIWARE architecture would be also deployed.

3.5.1.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Institute for the Development of New Technologies (UNINOVA) – Portugal – On premise

b. Testbed Location info:

Institute for the Development of New Technologies (UNINOVA) – Portugal

3.5.1.3 Testbed Functional & Technical Specification

a. Functional Specification

The module's description that conforms Personalized insurance products based on IoT connected vehicles Pilot and corresponds to the diagram below (see Figure 21):

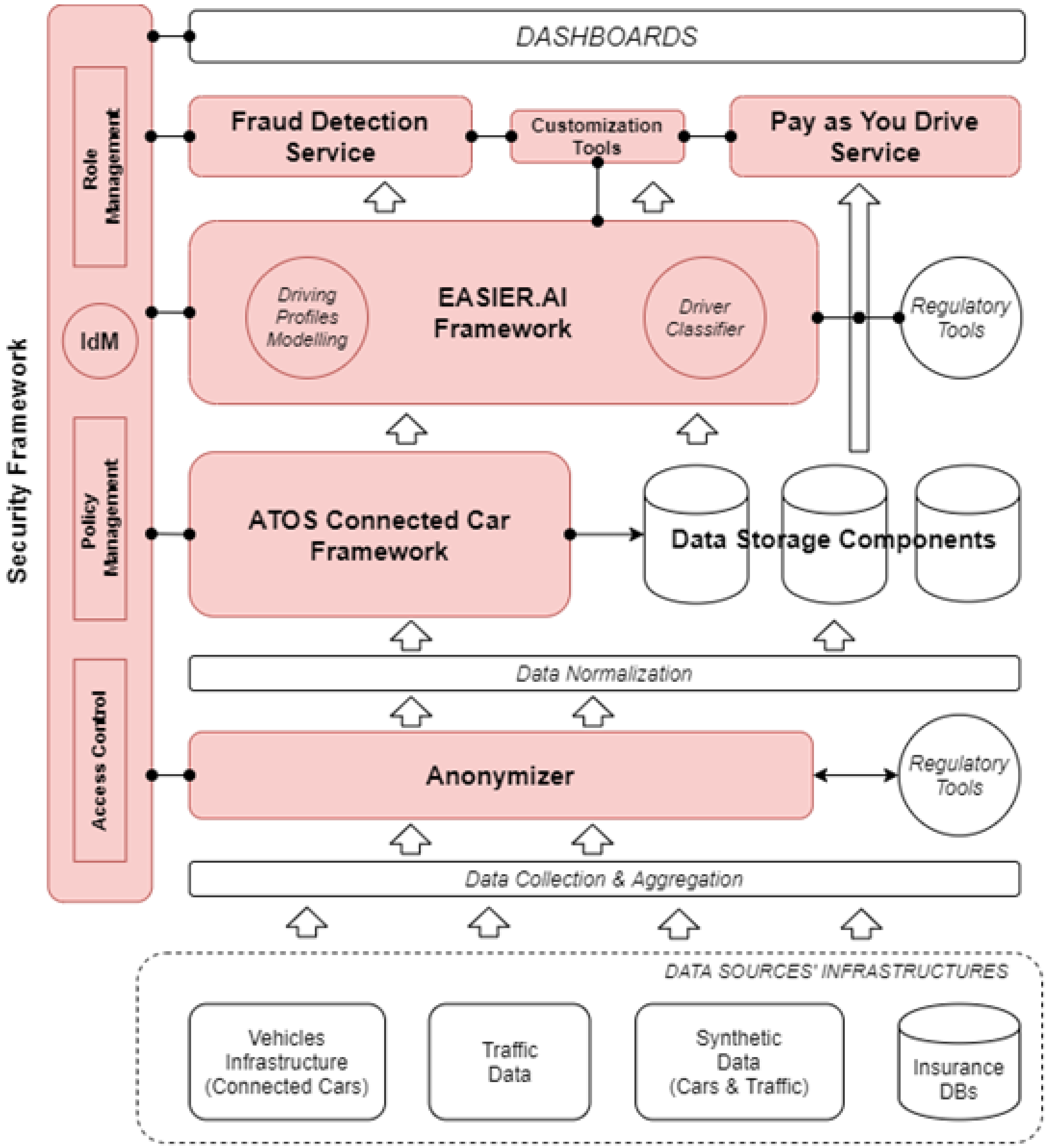

Figure 21 – ATOS Testbed Functional Specification Diagram

Components painted with red would be (initially) deployed in the testbed (NOVA) and corresponds with:

- Data anonymisation process (Anonymiser): that anonymise data coming from the data sources to guarantee GDPR compliance
- Routes (and Vehicle's data) collection (Connected Car Framework): to gather, homogenise and store data coming from the connected cars and the synthetic routes generator
- AI modelling and exploitation (EASIER.AI framework plus AI Services): that exploit captured data to define and obtain driving profiles and infer driving styles classification
- Access control management components to define users and roles to access the components stores data and configuration management

b. Technical Specification

- **Hardware Requirements**

Technical Components					
Platform	Nodes	Cores by Node	Memory	Storage	Special Hardware
EASIER.AI	5 = (1)xSmall Node + (2)xMedium Node + (1)xLarge Node + (1)xXXL Node	Small 1 Nodes 1 Core Medium 2 Nodes 4 Core Large 1 Nodes 6 Core XXL 1 Node 6 Core	Small 1 Node 4GB Medium 2 Nodes 8GB Large 2 Nodes 16GB XXL 1 Nodes 16GB	Small 1 Node 100GB Medium 2 Nodes 500GB Large 1 Nodes 2TB XXL 1 Node 10TB	Some nodes would have access to GPU, FPGA, TPU, or other special AI hardware. AVX2 Instructions set support
Connected Car	3 = (1)xSmall Node + (2)xMedium Nodes	Small 1 Nodes 1 Core Medium 2 Nodes 4 Core	Small 1 Node 4GB Medium 2 Nodes 8GB	Small 1 Node 100GB Medium 2 Nodes 500GB	
Anonymiser	1=(1)xMedium Node	1 Node 4 Core	1 Node 16GB	1 Node with 1TB	
Trainners/Inferencers AI	TBD later, in a first stage we will use the EASIER.AI infrastructure to also run trainers and inferencers				
TOTAL	9 Nodes	34 Cores	104 GB	15,2 TB	GPU, FPGA, TPU and AVX2 instructions support

- **Software Requirements**

Components to be deployed within the testbed will run on top of Linux OS. Connected Car Framework requires from MongoDB 3.2 (or upper). EASIER.AI component is based on ElasticSearch/Keras approach supported by a TensorFlow backend. Kubernetes framework is used to run all these components.

3.5.1.4 Sandboxes

There is no existing sandbox.

3.5.1.5 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.* Synthetic data managed within this pilot is also described in deliverable *D5.1 - Datasets for Algorithms Training & Evaluation.*

3.5.1.6 Compliance Requirements

Legal terms and conditions will be pending of special terms and conditions agreed between parts in the future.

3.5.1.7 Additional Info

The pilot will use Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.5.2 Testbed for Real World Data for Novel Health-Insurance products

3.5.2.1 Testbed Brief Description:

Testbed for **Personalized life insurance products based on IoT connected devices,** will host **Pilot 12** of INFINITECH Project that leader is **ISPRINT (Innovation Sprint Sprl – Belgium)**.

The testbed will be focused on deriving health profiles using real world data collected from activity trackers, and questionnaires. Different ML/DL techniques will be used to define and manage these profiles. To capture user´s daily activity, and answers to questionnaires a connected clinical platform (Healthentia) will be used.

3.5.2.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Institute for the Development of New Technologies (UNINOVA) – Portugal – On premise

b. Testbed Location info:

Institute for the Development of New Technologies (UNINOVA) – Portugal

3.5.2.3 Testbed Functional & Technical Specification

a. Functional Specification

The module's description that conforms Pilot 12 is shown in the diagram presented below (see Figure 26):

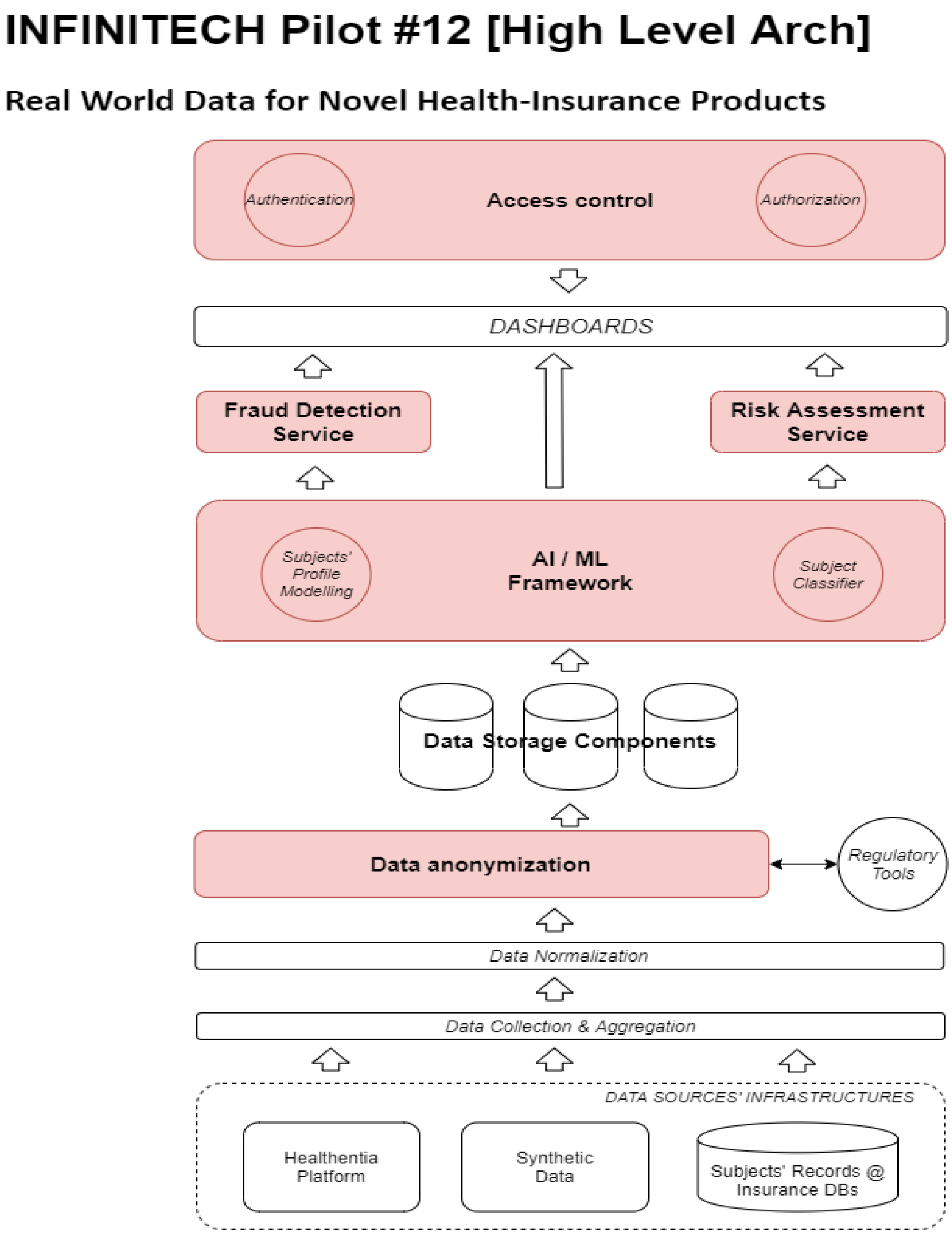

Figure 22 – ISPRINT Testbed Functional Specification Diagram

Components painted with red would be deployed in the testbed (NOVA) and corresponds with:

- Data anonymisation process (Anonymiser): that anonymise data coming from the data sources to guarantee GDPR compliance
- AI modelling and exploitation (AI/ML Framework): captured data will be analysed in order to discover different profiles and infer life styles classification.

- Access control management components to define users and roles to access the components stores data and configuration management

b. Technical Specification

- **Hardware Requirements**

Platform	Processors	Cores	Memory	Storage	Special Hardware
ML server	1	6 Cores / processor	32 GB	2TB	Access to GPU special AI hardware is very welcome
Application server	1	6 Cores / processor	32 GB	12TB	None
Anonymization server	1	4 Cores / processor	16 GB	1TB	None
TOTAL	**4**	**18**	**64 GB**	**15TB**	

- **Software Requirements**

Platform	OS	SW
ML server	Ubuntu 18.04	To be managed by Innovation Sprint
Application server	Ubuntu 18.04	To be managed by Innovation Sprint
Anonymization server		

3.5.2.4 Sandboxes

There is no existing sandbox.

3.5.2.5 Datasets

Datasets Requirements are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.* Synthetic data managed within this pilot is also described in deliverable *D5.1 - Datasets for Algorithms Training & Evaluation.*

3.5.2.6 Compliance Requirements

Legal terms and conditions will be provided, pending special terms and conditions agreed between parties in the future.

3.5.2.7 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.6 Testbeds for Configurable and Personalized Insurance Products for SMEs and Agro-Insurance

3.6.1 Testbed for Alternative/automated insurance risk selection - product recommendation for SME

3.6.1.1 Testbed Brief Description:

Testbed for **Alternative/automated insurance risk selection - product recommendation for SME** will host **Pilot 13** of INFINITECH Project that leader is **WEA (Wenalyze Sl.- Spain)**.

The targets or initial data of the companies are loaded, basically name, address and tax identification number. Automatically and in real time, the robots of the platform search for information on these companies in the online open sources
For the management of the information we will do it through data aggregators that then to pass to a manager of these data
Once the information extraction phase is finished, we move on to the architecture part of the data process which would be done on databases in NoSQL and SQL format, here a polyglot module would be used for the analysis in different languages of the database
The information once processed will go to the user interaction part and its visualization

3.6.1.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Institute for the Development of New Technologies (UNINOVA) – Portugal – On premise

b. Testbed Location info:

Institute for the Development of New Technologies (UNINOVA)) – Portugal

c. Testbed Responsible Contact Persons:

To be filled by NOVA in a later stage.

3.6.1.3 Testbed Functional & Technical Specification

a. Functional Specification

The Pilot 13 will implement an automation of the subscription process 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.

- In Nova hosting just will be implement the data base system by LeanXcale

- The solution consists in make transactions and be storage and manage in non-relational databases

The diagram below (see Figure 23) describes the system and the relevant components that will be implemented in INFINITECH Project. The components of the architecture that will be in Nova are the ones inside the red square

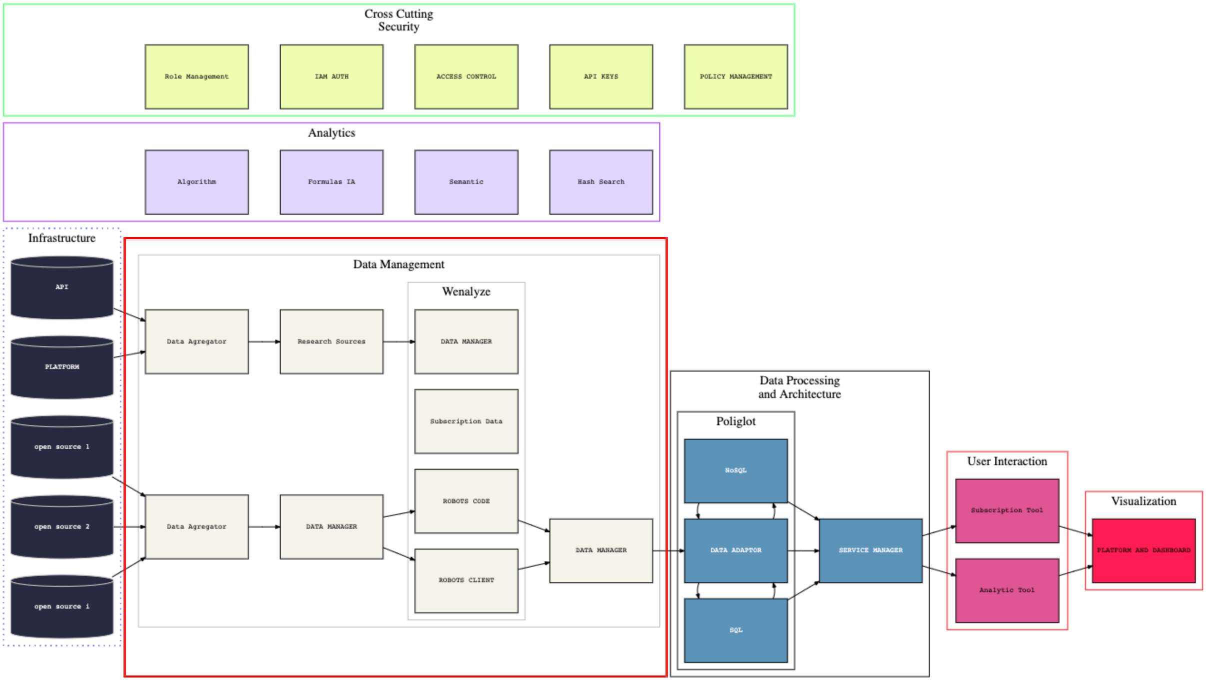

Figure 23 - WEA Testbed Hardware Requirements Diagram

b. Technical Specification

- **Hardware Requirements**

An instance similar to AWS with:

- CPUs, 8 cores
- Architecture, X86_64

- Memory, 32 Gb of RAM
- Storage, 500 Gb
- Storage type, SSD
- Network performance 25 Gigabit

- **Software Requirements**

  The following operating system and open source software will be used:

- Linux operating system (Debian like Ubuntu, no GNU)
- SSH access
- Node.js and NPM
- GITHub
- NginX server

3.6.1.4 Sandboxes

There is no existing sandbox.

3.6.1.5 Datasets

Data will be extracted from open sources such as company websites, official registers, social networks, opinion forums, etc

SMEWIF	SMEs website information and functionalities
ROPS	Review and opinions platforms
EUBD	European SMEs Business Directories
GIO	SMEs geolocation information and characteristics
SMSIP	Social media SMEs information and presence

For more information are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

Data will include 150.000 SME targets with 50.000.000 data fields

3.6.1.6 Compliance Requirements

Legal terms and conditions will be pending of special terms and conditions agreed between parts in the future and will be included in the next version of deliverable D6.1.

3.6.1.7 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

3.6.2 Testbed for Big Data and IoT for the Agricultural Insurance Industry

3.6.2.1 Testbed Brief Description:

Testbed for **Big Data and IoT for the Agricultural Insurance Industry** will host **Pilot 14** of INFINITECH Project that leader is **GEN (Genillard&Co GmbH – Germany).**

The targets or initial data of the companies are loaded, basically name, address and tax identification number. Automatically and in real time, the robots of the platform search for information on these companies in the online open sources
For the management of the information we will do it through data aggregators that then to pass to a manager of these data
Once the information extraction phase is finished, we move on to the architecture part of the data process which would be done on databases in NoSQL and SQL format, here a polyglot module would be used for the analysis in different languages of the database
The information once processed will go to the user interaction part and its visualization

3.6.2.2 Testbed Hosting Partner Details

a. Hosting Partner Information Data:

Institute for the Development of New Technologies (UNINOVA) – Portugal – On premise

b. Testbed Location info:

Institute for the Development of New Technologies (UNINOVA) – Portugal

3.6.2.3 Testbed Functional & Technical Specification

a. Functional Specification

Find below the main components that will be used or developed as part of the INFINITECH Project for Pilot 14 and hosted to the specific testbed:

Octopush EO Service	
Description	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.

Icon	octopush
Owner	AgroApps PC
BDVA Layer	Infrastructure/Data, Management/Data, Processing
Input	Sentinel SAR and L2A, MODIS Products, ProbaV, geospatial information
Output	Vegetation indices, Biophysical parameters, FAPAR/NDVI/DMP Anomalies, SAR indices
Already available	Yes
Technology to be used	Python, PHP, SAGA tools and Orfeo toolboxes, Docker, R Library, RESTful API
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

AgroApps Weather Intelligence Engine (WIE)	
Description	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.
Owner	AgroApps PC
BDVA Layer	Infrastructure/Data, Management/Data, Processing
Input	Copernicus Climate Data, WRF
Output	Crop Specific Climate Indices, 4D Weather Information, AOI cropping
Already available	Yes
Enhancements needed	Describe what more needs to be done to fulfil the pilot use case needs
Technology to be used	WRF-ARW, UPP, GRIB2, NETCDF4, MPI, WPS-ARW, WRFDA, RESTful API, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

Data Integrator

Description	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.
Owner	AgroApps PC
BDVA Layer	Data Integrator
Input	Octopush EO Service and WIE metadata, AOI
Output	Zonal Statistics, Alerts
Already available	Yes
Technology to be used	Python, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

Business DB	
Description	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.
Owner	AgroApps PC
BDVA Layer	Architecture/Data, Analytics/User, Protection/Data, Storage/Data
Input	User input
Output	Business data
Already available	Yes
Technology to be used	PostgreSQL, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	Yes: It will help to ensure that the personal data of the pilot are processed in compliance with GDPR.

Geospatial DB	
Description	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.
Owner	AgroApps PC
BDVA Layer	Storage/Data
Input	geometries, zonal statistics
Output	alerts, vector sources
Already available	Yes
Technology to be used	PostGIS, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

WMS Server	
Description	WMS (Web Map Server) is responsible for rendering and serving of the GIS layers to the User Interface.
Owner	Open Geospatial Consortium
BDVA Layer	Interaction/ Visualisation, Server
Input	Vector source
Output	maps
Already available	Yes
Technology to be used	Geoserver, Docker
License	Open Geospatial Consortium
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

RESTful API	
Description	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.
Owner	AgroApps PC
BDVA Layer	Data Integrator
Input	Business Data, Alerts, Maps
Output	RESTful API, API endpoints
Already available	Yes

Technology to be used	PHP, Laravel PHP framework, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	No

User interface	
Description	The front-end user interface is the gateway responsible to present all the system data through user-friendly controls and web mapping interfaces.
Owner	AgroApps PC
BDVA Layer	Visualisation
Input	Business data, alerts, vector source, maps
Output	interactive map controls, filter controls, sortable table controls, export of data (i.e. xls, csv, etc.)
Already available	Yes
Technology to be used	Javascript (Vue.js), OpenLayers, Docker
License	AgroApps Proprietary
Part of INFINITECH Core?	Can be offered as part of the INFINITECH architecture for the pilot #14
Part of Regulatory Compliance Tool?	Yes: It will help to ensure that the personal data of the pilot are processed in compliance with GDPR.

b. Technical Specification

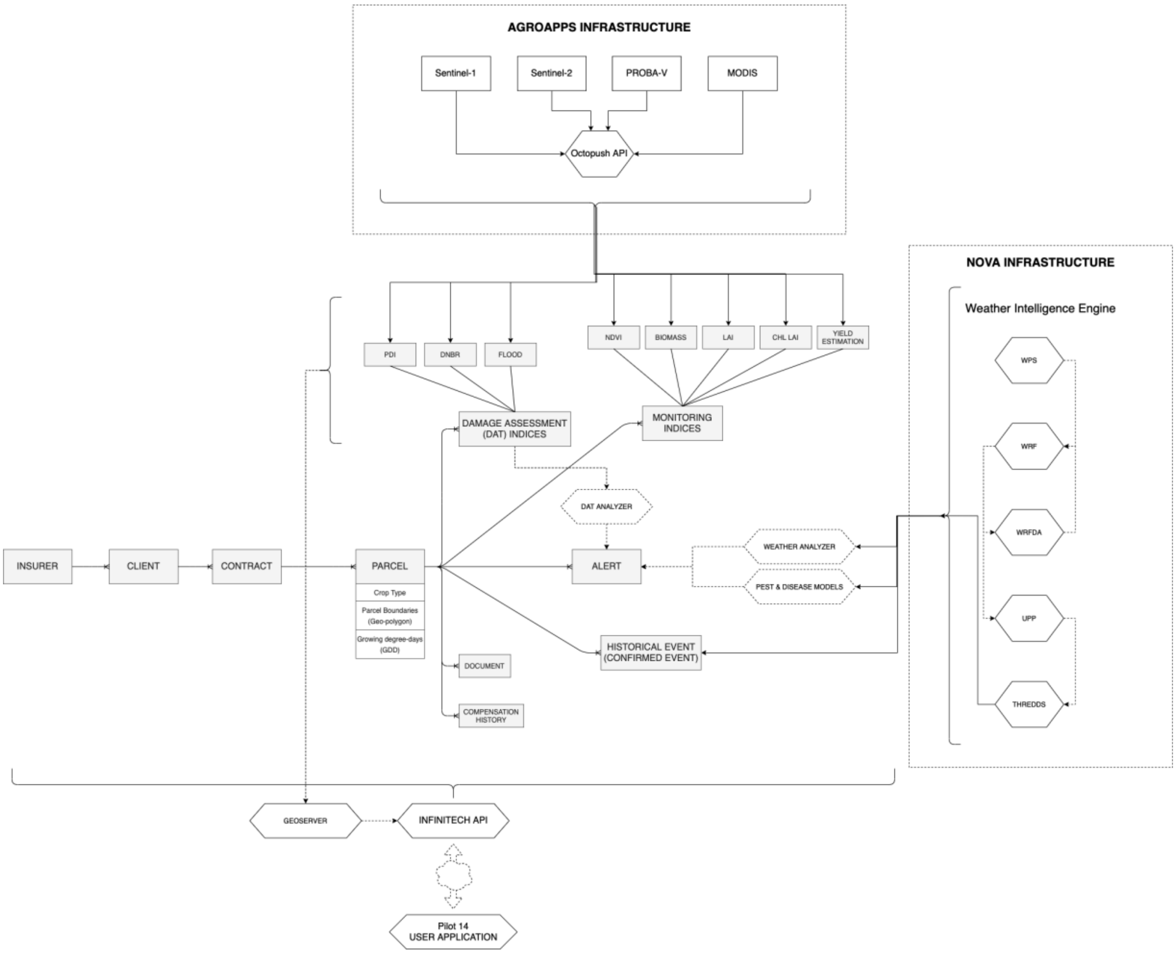

Figure 24 - AGRO Testbed Technical Specification Diagram

- **Hardware Requirements**
    - 240 CPU cores (720 CPUh/forecast cycle, 2 cycles/day),
    - 180GB RAM,
    - 500GB SSD Storage
- **Software Requirements**
    - CentOS OS
    - Libraries:
        - HDF5, NETCDF4 (C, FORTRAN), JASPER
    - Data Bases:
        - PostGIS, PostgreSQL, THREDDS
    - Programming Tools:
        - NCL, Intel Compiler (C, FORTRAN), Python

3.6.2.4 Sandboxes

There is no existing sandbox.

3.6.2.5 Datasets

Dataset Name	Dataset (short) description
Gridded Climate Indices (1/1/1979 to 1/12/2019)	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

For more information are included in the deliverable *D2.9 - Initial Specification of Testbeds, Data Assets and APIs.*

3.6.2.6 Compliance Requirements

Testbed & Sandbox environments will follow the Compliance & Regulatory Requirements already described in deliverable *D2.7 - Security and Regulatory Compliance Specifications.*

3.6.2.7 Additional Info

The pilot partners are considering the use of Kubernetes to manage system components' deployment within a CI/CD environment. GitLab will be used to manage component's code.

4. Testbeds & Sandboxes Implementation General strategy

The INFINITECH Reference Architecture (cfr. INFINITECH D2.13 deliverable) was developed having in mind a specification of the related development and deployment views following a microservices based approach, with services interacting among them through REST APIs. This choice implies, the necessity to change the approach in the app creation, moving from design of monolithic applications where the various components (generally UI, Business logic and Data-layer) were strongly coupled among them, to microservice applications where the various components (microservices) are decoupled from each other.

The nature of the microservices, if from one side provide many benefits:

- Simple to develop
- Simple to upgrade
- Simple interaction
- Simple to scale

From other side, a software previously managed as a single piece indivisible is split into several dozen of microservices (containers), making them difficult to manage for some aspects:

- Deployment is more complicated
- Monitoring each microservice requires to have many metrics and logs to manage it
- Debugging, a monolithic application is much easier to debug and test due to fact is composed by a single indivisible unit.

In this context, many tools like Docker Swarm, Apache Mesos, OpenStack Magnum and so on have been created in order to manage the entire life-cycle of the microservices (deployment, scaling, and management), but today the most famous tool is Kubernetes.

Kubernetes is a portable, extensible open-source platform for managing containerised workloads and services, that facilitates both declarative configuration and automation [kubernetes.io documentation]

We 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:
   a. **Quote** sets the limits on how many HW resources can be consumed by all objects
   b. **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, as is shown in the picture below, 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.

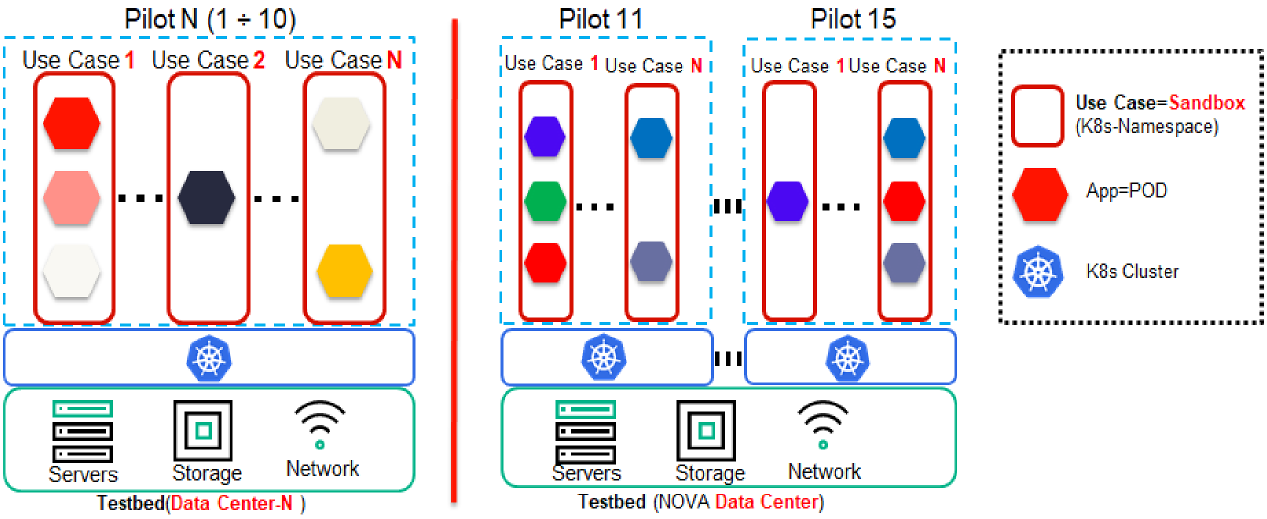

Figure 25– Testbed Proposed Architecture

However, the microservices approach, requires not only a technological change but also a cultural change. Its use involves to pass from creating, building, testing and distributing a version of an application 10 times a year to 10 times a day. This required the development of a new methodology that creates a strong synergy and exchange of information between developer and operation people, that methodology is called DevOps

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, WP6 (and INFINITECH related tasks in general) will adopt a Continuous Integration approach, using open source tools such as Jenkins. Every time a developer pushed changes to the source code repository, Jenkins automatically triggered a new build of the component and deployed the updated container to an integration environment on Kubernetes. This allowed 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 is deployed to target environments, typically to test environments, in a repeatable and automated fashion.

Moreover, in WP6 tasks and activities we intend to enhance the 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 developers quick feedback if the latest changes introduced 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 (a compound of Machine Learning and IT Operations) is focused 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

In order to automate porting the machine learning algorithms, as much as possible, in production environments.

Putting this into practice is often very complicated, because ML works in heterogeneous environments, for example often the models are 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 this the Kubernetes and 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 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 WP6 tasks.

Moreover, in order to simplify development and integration by each partner, a blueprint will be created on a cloud provider. In particular on the cloud provider will be created a cluster Kubernetes on which will be implemented tools for CI/CD like:

- Gogs(Go Git Service) as software version
- Jenkins as pipeline manager
- Harbor as Docker registry

Finally, a proper development environment will be created to test the INFINITECH services/components.

Further details and Specification about the previous concepts will be reported in the upcoming WP6 deliverable D6.4 “Tools and Techniques for Tailored Sandboxes and Management of Datasets I”.

5. Conclusions and next steps

The scope of task T6.1 is the analysis of the existing testbeds (i.e. testbeds of incumbent organizations) in terms of their existing resources and gaps for supporting BigData, IoT, AI experimentation in-line with the INFINITECH approach. It will accordingly provide the necessary feedback in order to specify the ways they have to be extended in terms of hardware and/or software resources, based on the progress of the pilot development and execution.

The current version of the deliverable is based on the initial feedback from all Pilots, based on the information already available at this stage. Based on the initial analysis the process will be iterative, as testbeds will be upgraded dynamically in order to fulfill the project's and the pilots' needs.

Based also on the preparation of a reference testbed that will be part of the deliverables that will be part of WP6, will be able all Pilot's to adjust their hardware and/or software resources to be compliant with the reference testbed, in order to be able that the relative tools, procedures, deployment and support procedures to be unified. The testbeds and the relative sandboxes deployment process will be enhanced accordingly based on the reference blueprint testbed and the details of each pilot, using the feedback of the relative partners that will be responsible for the provision of the relative technical support, as part of the rest of WP6 tasks.

As we are running the initial phases of the INFINITECH Project the deliverable depicts the current specifications of hardware and/or software resources that each Pilot and the relative Testbeds will use for deployment. As for some pilots the relative infrastructure is not yet finalized the relative information are minimum and will be updated based on the feedback partners that participate in each pilot, in the next phases of Pilots' development and this deliverable next version. Also, as 5 of the Pilots will be hosted in on Testbed provided from NOVA, the relative information will be expanded with more details based on the feedback of the relative team that will be responsible for providing the necessary infrastructure and support.

Based on the outcomes of this deliverable, the next steps of task T6.1 are:

- Specify in more detail the specifications necessary of hardware and/or software resources that will be used from all INFINITECH Pilots and the Tailored Experimentation Infrastructures (Testbeds and Sandboxes) for different BigData, IoT and AI applications in the Financial and Insurance Sectors. The feedback will be based on the progress and realization of the Pilots deployment progress including datasets, Open APIs for experimentation and innovation, ML/DL algorithms and regulatory compliance tools.

- Based on the updated information for the necessary hardware and/or software resources, as well as the reference testbed infrastructure that will be available as part of the WP6 of INFINITECH Project, to describe the necessary amendments or gaps that will be required in order to fulfil all and have a unified way of deployment for all INFINITECH Pilots.