QU4LITY

DIGITAL MANUFACTURING PLATFORMS FOR CONNECTED SMART FACTORIES

D3.4 HPC and Cloud Resources for ZDM

Deliverable Id:	**D3.4**
Deliverable Name:	**HPC and Cloud Resources for ZDM**
Status:	**FINAL**
Dissemination Level:	**PU**
Due date of deliverable:	**31/09/2021**
Actual submission date:	**29/10/2021**
Work Package:	**WP3**
Organization name of lead contractor for this deliverable:	**Engineering Ingegneria Informatica S.p.A. (ENG)**
Author(s):	**Angelo Marguglio**
Partner(s) contributing:	**Angelo Marguglio (ENG), Manfredi Giuseppe Pistone (ENG), Ales Ude (JSI)**

Abstract: This deliverable is devoted to presentation of QU4LITY customized HPC and Cloud Infrastructures for Digital Quality Management. In this context, it will be presented an HPC Infrastructure for state of art modeling, simulation, and data analysis to support autonomous decision-making. Moreover, it will be presented a private cloud infrastructure for supporting ZDM processes.

Co-funded by the Horizon 2020 Programme of the European Union

Contents

HISTORY 4

1 Executive Summary 5

2 Introduction 6

2.1 Objectives and Scope 6

2.2 Main changes after the first release 7

2.3 Methodology and workplan 7

2.4 Document Structure 8

3 Context and Requirements 10

3.1 Tech-Survey Questionnaire Results 11

3.1.1 PHILIPS: OneBlade shaving unit production line 12

3.1.2 SIEMENS: Control Products Quality Improvements 12

3.1.3 CONTI: Autonomous Quality in PCB Production for Future Mobility 14

3.1.4 WHR: Dryer Factory Holistic Quality Platform 15

3.1.5 MON: Zero defect and Autonomous Quality in Machinery Building for Capital Goods sector 16

3.1.6 KOL: Real-time injection molding process monitoring-control 17

3.1.7 THYS: Quality Management of Steering Gear based on Acoustic control 18

3.1.8 AIRBUS: Trade space framework for Autonomous Quality Manufacturing Systems' Design 19

3.1.9 GHI: Real-time cognitive hot stamping furnace 4.0 19

3.1.10 RiaStone: Autonomous Quality ZDM for "Ceramic tableware Single-firing 21

3.1.11 PRIMA: Additive Manufacturing Pilot Adaptive Control Technology 21

3.1.12 Danobat: Digital Machine for zero-defects at high precision cutting/grinding 22

3.1.13 FAGOR: Zero-Defects Manufacturing Digital Press Machine 23

3.1.14 GF: Digital machine and part twins for zero defect manufacturing 25

3.1.15 Overall assessment of Tech-Survey results 25

4 QU4LITY High Performance Computing Infrastructure 27

4.1 Main Features and Functional Description 27

4.2 Technologies and Architecture 28

4.3 Access Instructions 29

4.4 Quick Start Guide 29

4.5 Validation Campaign 35
4.6 Future Works 38
5 QU4LITY Cloud Solutions 39
5.1 Main Features and Functional Description 39
5.2 Technologies and Architecture 39
5.2.1 QU4LITY Cloud Bridge 40
5.2.2 Q-Ontology Enabler 41
5.2.3 Q-Ontology Enabler Customization 45
5.3 Access Instructions 45
5.3.1 REST API 46
5.3.2 Services Containerization 53
5.4 Quick Start Guide 55
5.4.1 QU4LITY Cloud Bridge 55
5.4.2 Q-Ontology Enabler 56
5.5 Content Available 57
5.6 Validation Campaign 58
5.7 Future Works 59
6 Conclusions 60
References 61
List of figures 62
List of tables 63
List of Abbreviations 64
Partners: 65

HISTORY

Version	Date	Modification reason	Modified by
0.1	30/07/2021	First ToC to be shared	Angelo Marguglio
0.2	02/08/2021	Initial inputs to the deliverable	Manfredi Pistone
0.3	14/09/2021	Initial JSI inputs added	Ales Ude
0.4	20/09/2021	Further refinements and inputs to the deliverable structure	Manfredi Pistone
0.5	27/09/2021	Final JSI inputs added	Ales Ude
0.6	09/10/2021	Minor improvements and further refinement, ready for the peer review	Angelo Marguglio
0.7	18/10/2021	Final updates after peer review	Manfredi Pistone
1.0	29/10/2021	Format quality check, final version for submitting	Diego Esteban

1 Executive Summary

The deliverable D3.4 "HPC and Cloud Resources for ZDM" will report on the work performed and final results of Task 3.2 (Customization of HPC and Cloud Infrastructures for Digital Quality Management). The scope of Task 3.2, as well as the content of this deliverable, is focused on the development of HPC and Cloud Solutions able to support ZDM processes.

Closed-loop integration in ZDM processes usually requires the integration of Cloud resources with artificial intelligence (AI) and machine learning capabilities. The AI, generally powered by High Performance Computing Infrastructures, gathers data from previous and current application integrations, analyzes that data and uses the results of the analysis to make informed recommendations or autonomously make decisions. In this way, closed-loop integration grants enterprise teams increased operational visibility into the whole ecosystem of application integrations. In this context, this report will provide an in-depth analysis of JSI HPC Infrastructure, providing a functional description and highlighting its features. Based on hybrid CPU and GPU infrastructures, HPC infrastructure will be provided with models and blueprints (quality management processes simulation, performance and failure predictions, defect causes detection, etc..) for executing ZDM simulations.

The Task 3.2 will also define a set of Cloud Solutions to support ZDM processes, making interoperable cloud-based applications, enterprise systems and quality management applications.

This document will also offer an overview of the validation campaigns put in place for the aforementioned solutions, highlighting pilot scenarios, their needs, their requirements and their solutions.

In a nutshell, this deliverable will describe and present a set of composable solutions and expertise in the two families of technological pillars built within the context of QU4LITY project and here discussed, High Performing Computing Infrastructure and Cloud solutions, as well as an overview of blueprints to pave the way for exploitation and dissemination phases realized in WP7-8-9.

2 Introduction

2.1 Objectives and Scope

The whole work-package 3 (WP3 in short) is developing and integrating a range of digital enablers, based on background technological developments of the partners and properly customized to the needs of ZDM, to support the QU4LITY Autonomous Quality paradigm as described in D2.3 (and in the following D2.4). Furthermore, these digital enablers will be integrated as part of ZDM solutions so as to emphasizes interoperability and flexibility and will be further exploited (even outside the project boundaries) through the QU4LITY marketplace (as part of the WP8 activities). Last but not least, the term "digital enablers" implied that each digital component will be reusable and accessible via an Open API, which will facilitate their use in ZDM processes and applications [Psarommatis20].

The challenge faced by QU4LITY is the requirement of interoperability among ZDM digital enablers which may deeply differ from each other. QU4LITY-based systems should, in fact, rely on a layer of abstraction, which to the extent possible obscures the system from the underlying implementation. In this matter, a cloud infrastructure is required to support ZDM processes and automation platforms of the Partners.

Almost all modern Digital Manufacturing Platforms (DMP) rely on cloud solutions, even if with a wide range of deployment options (e.g. on premise, on private clouds or even in public clouds). One characteristic that most cloud-based/cloud-supported systems have in common is that they rely on a distributed architecture (e.g. using a microservices architecture), consisting of several independent services providing the application functions. Such an architecture has several advantages over previous monolithic or bespoke component architectures: a) increased responsiveness on fulfilling the demands of the high-variable of modern businesses; b) increased scalability thanks to services modularization; and c) increased reliability as it is easier to swap out one component for another, assured it conforms to the same API and behavior. The last point is particularly relevant for the QU4LITY project as it facilitates a plug-and-play architecture, which is one of QU4LITY core objectives in any ZDM scenario.

Microservices architectures typically implement the different services in one, or a set, of container(s) in a virtualized environment. This feature gives the system the flexibility to instantiate preconfigured services from container templates and move containers around the physical IT infrastructure to strengthen the overall resilience of the system and to optimize its performance. As most major Cloud Service Provider (CSP) provide their tenant environments as a microservices architecture, QU4LITY Cloud Platform will be implemented following the same principles. Basing the QU4LITY system on a microservices architecture gives the manufacturing companies the flexibility to implement their solution in either environment, or a hybrid of the two. It also means that the QU4LITY system can inherit the added benefits of a microservices architecture, as outlined in the previous paragraphs. Whilst each CSP implements their environment in a proprietary way, the industry is starting to converge on a set of common architectural patterns and support for common services; by exploiting this convergence, the QU4LITY system can be made highly flexible in how and where it is deployed.

2.2 Main changes after the first release

Following the first iteration of Task T3.2 and its outcome represented by D3.3 and the recommendations received by the review team after the Midterm review, many changes have been implemented in this report D3.4.
Even if most of the content of the first release is still valid, a more in deep revision and fine tuning have been performed to provide stronger basis of the second and final iteration of the project execution. To this end, the following main changes/addendum have been implemented:

- Section 3.2 - Tech-Survey Questionnaire Results: results of a technical questionnaire, circulated among pilots during M28-M30, to better identify the needs and the requirements of pilots regarding Cloud Solutions and Infrastructures have been reported here.
- Section 4 - QU4LITY High Performance Computing Infrastructure: The core part of this specification has been updated to reflect the latest changes implemented on the JSI Infrastructure, as well as to introduce an exhaustive and detailed quick start guide to facilitate its implementation across pilots. Moreover, a validation campaign is proposed to present a real use case scenario to the reader, addressing comments received in the Review Report (RR) at Midterm review.
- Section 5 - QU4LITY Cloud Solutions: The core part of this specification has been updated to reflect the latest changes implemented by ENG in respect to pilots' needs and requirements, reported in section 3.2, as well as to introduce an exhaustive and detailed quick start guide to facilitate its implementation across pilots, and a brief overview of R-MPFQ Ontology Model, presented in D2.9 (and the following D2.10). Moreover, a validation campaign is proposed to present a real use case scenario to the reader, addressing comments received from RR at Midterm review.

2.3 Methodology and workplan

The methodology followed for the specification and development of the digital enablers for QU4LITY Cloud Solutions comprises the following phases:

- **Phase 1 – Requirements Analysis**: During this phase, the main pilots' requirements and needs to bring up the QU4LITY Cloud Solutions and QU4LITY HPC Infrastructure will be identified to prototype the most suitable platforms in accordance with the QU4LITY Reference Architecture.
- **Phase 2 – Existing Solution Analysis**: During this phase, existing solutions from QU4LITY partners will be identified and analyzed to keep in-line with QU4LITY vision and objectives.
- **Phase 3 – Prototyping**: During this phase, an initial prototype of QU4LITY Cloud Solutions will be implemented, with the aim to address potential architectural defects before starting the final implementation. QU4LITY HPC Infrastructure, on the other hand, will implement prototypal analytics workflows to evaluate analysis results.

- **Phase 4 – Customization and Enhancement**: During this phase, QU4LITY Cloud Solutions and QU4LITY HPC Infrastructure will be customized and enhanced based on pilots' feedback to address any possible flaw in its architecture and best suit it for production-ready use.

Furthermore, the activities within this task T3.2 did not start from scratch but has been conducted exploiting the experience gained from the Consortium partners in previews innovation and business activities. QU4LITY will be able to connect resources for modelling and simulation with the rest of digital enablers developed in WP3 (e.g. data analysis tools or simulation frameworks) tailoring these solutions for the specific needs in the context of ZDM. The application of these services to sensors data relevant for the product quality results in a further step towards Autonomous Quality (AQ).

2.4 Document Structure

D3.4 is divided in the following main parts:

- **Introduction**: This section identifies the tasks of the project related to the deliverable including information on objectives as well as a short description of the relationship of the current deliverable with the results of other tasks and work-packages.
- **Context and Requirements**: An analysis of other project context and especially the relationships with the QU4LITY Reference Architecture and business requirements, aligned to the overall Autonomous Quality vision pursued by the project.
- **QU4LITY High Performance Computing Infrastructure**: An analysis of the HPC cluster provided by JSI, serving as a reference to build HPC infrastructures for supporting ZDM processes.
- **QU4LITY Cloud Solutions**: An analysis of QU4LITY Cloud Solutions provided by ENG, serving as a reference to implement autonomous quality loops based on R-MPFQ Ontology Model for supporting ZDM processes.
- **Conclusions:** This section summarizes the achievements reached within T3.2 and paves the way to the evaluation, dissemination, exploitation and standardization of the produced assets/enablers in WP7-8-9.

An overall view of the document structure can be seen in the figure below.

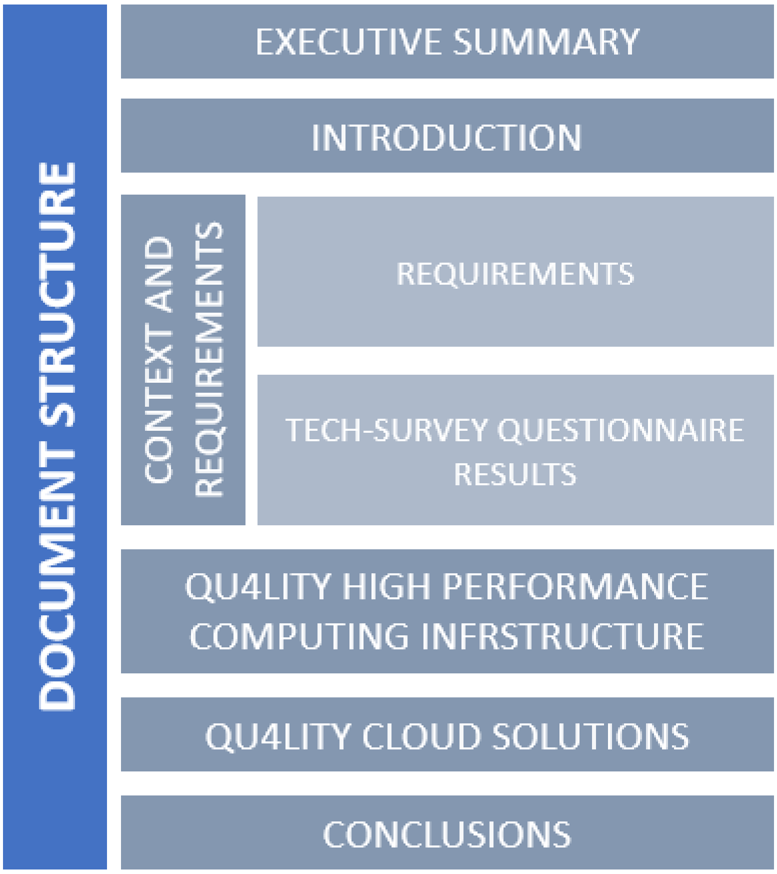

Figure 1 - D3.4 document structure

3 Context and Requirements

HPC and Cloud resources for ZDM are core parts of the QU4LITY Reference Architecture (Q-RA). This is depicted in Figure 2, which illustrates different components and views of the Q-RA.

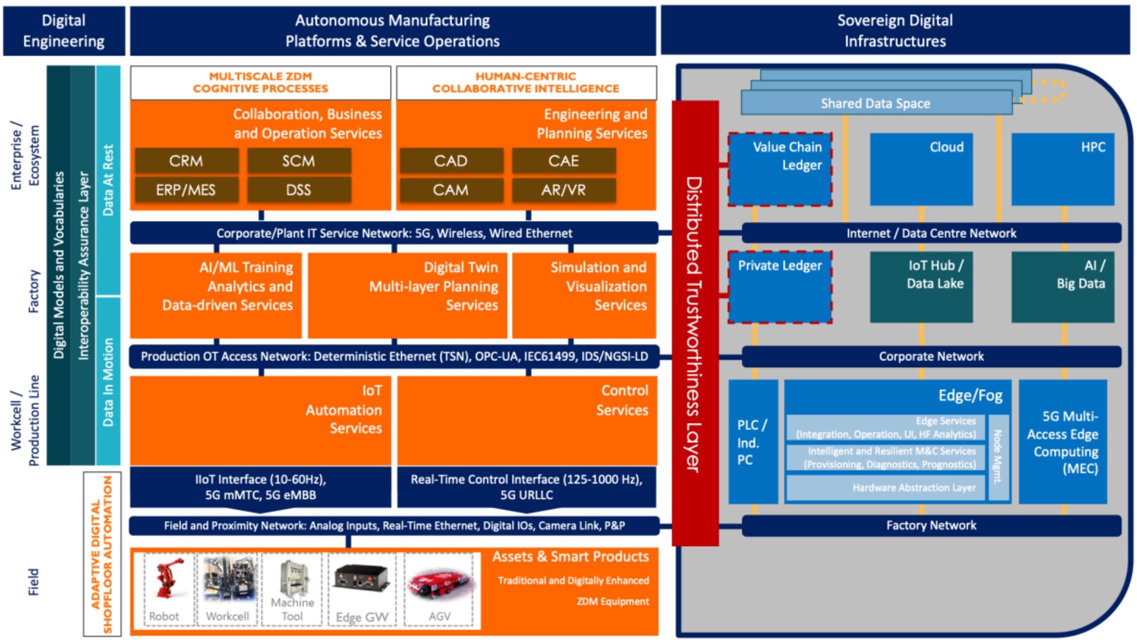

Figure 2 - QU4LITY Q-RA

In line with deliverable D2.12, the HPC and Cloud infrastructures are part of the **Digital Infrastructure Pillar.** This pillar is intended for the Fog/Cloud/HPC infrastructures required for the operation of the digital services pillar as well as communication and data distribution enablers to create direct interaction between the different layers. This vertical domain is therefore focused on the enablers for (big) data ingestion, processing and management both data in motion and data at rest.

The physical deployment of QU4LITY should benefit, in fact, from Cloud and HPC infrastructures. These infrastructures are essential for the implementation of the three-tier architecture pattern of the QU4LITY Q-RA. Likewise, HPC and Cloud resources will enable heavy-load computations as part of the Artificial Intelligence and industrial analytics cross-cutting functions.

The two components presented in this deliverable, QU4LITY High Performance Computing Infrastructure and QU4LITY Cloud Solutions, adhere to Q-RA specifications and interoperate with the remaining architectural components closing the gap toward the fully Autonomous Quality presented in WP2 reports.

The digital enablers under consideration in the current deliverables will have an impact in any of the project pilot. Most of the information collected from the pilot owners have highlighted the need for cloud resources to deal with the business objectives of the pilots themselves, in particular the following low-level requirements have been identified and analyzed in deliverables D2.1, and the following D2.2:

- Real-time or near real-time processing is required from Cloud Resources Provider;
- Both private and public cloud platforms are used;
- Fog/edge devices are needed to gather data, process models and return decisions/settings to the equipment;
- An interoperability layer is critical to support legacy systems.

In the project preparation phase, it became clear that pilot owners are further along with the development of Cloud than HPC Solutions. Hence the analysis in Section 3.1 focuses on Cloud Solutions, whereas the application of HPC in the context of ZDM was demonstrated in the frame of the Kolektor pilot. This is also reflected in the allocation of resources to both pillars.

3.1 Tech-Survey Questionnaire Results

To better identify the needs and the requirements of pilots regarding Cloud Solutions and Infrastructures, we've circulated a template among pilots to collect information regarding their use case so as to better focus the development of QU4LITY Cloud Solutions. To this end, the following lightweight template has been circulated and pilots' answers have provided a solid basis to complete this document.

Table 1 - Tech-Survey Questionnaire Template

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	
2	Which goals does your Cloud Platform help you achieving?	
3	How does your Cloud Platform address ZDM needs?	
4	How do Cloud Platform performances impact your workflow?	
5	Which QU4LITY components does your Cloud Platform interact with?	
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	
7	How could your Cloud Platform be improved?	
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	

9	Add free comments here	

3.1.1 PHILIPS: OneBlade shaving unit production line

Table 2 - PHILIPS Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Some test with MS Azure for example
2	Which goals does your Cloud Platform help you achieving?	Data coupling of several production lines
3	How does your Cloud Platform address ZDM needs?	Currently too slow for real-time production
4	How do Cloud Platform performances impact your workflow?	Security reasons of data and speed/complexity of big data sets
5	Which QU4LITY components does your Cloud Platform interact with?	TTTech edge device
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	No
7	How could your Cloud Platform be improved?	No standard yet
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	
9	Add free comments here	Currently no cloud service in place - data mirroring used for external parties.

3.1.2 SIEMENS: Control Products Quality Improvements

Table 3 - SIEMENS Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes. Siemens MindSphere. Connectivity from field to service application.

2	Which goals does your Cloud Platform help you achieving?	Quality management. OEE Reporting. Machine Analytics.
3	How does your Cloud Platform address ZDM needs?	With machine analytics the production process can be optimized to reduce failures.
4	How do Cloud Platform performances impact your workflow?	Workflow of production process is not impacted by Cloud Platform performances. On the other hand, quality process workflow is impacted by Cloud Platform performances, as it relies on analytics and model updates.
5	Which QU4LITY components does your Cloud Platform interact with?	Train new models and update
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	AML, xml
7	How could your Cloud Platform be improved?	
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	
9	Add free comments here	

3.1.3 CONTI: Autonomous Quality in PCB Production for Future Mobility

Table 4 - CONTI Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	AWS. Redshift, Dynamo DB, Problem Statement Our current architecture with DWH is outdated and not anymore sustainable and scalable in 55 Automotive Locations World Wide. In order to be able to provide the necessary support, updates and new features (Roll out) to our customers it has become very difficult and cost intensive. Therefore, we decided to bring our Manufacturing Data from today's MES to new world of cloud-based technologies at AWS. Benefits: • Data in any format • Big Data analytics • Data management + cataloging • Data-driven Apps • Improve products and services • Scalable solution • Identical HW resources for all plants
2	Which goals does your Cloud Platform help you achieving?	We use AWS to provide the necessary support, updates and new features (Roll out) to our customers
3	How does your Cloud Platform address ZDM needs?	Big Data analytics Data management + cataloging Data-driven Apps Improve products and services Scalable solution Identical HW resources for all plants
4	How do Cloud Platform performances impact your workflow?	Response time is not sufficient for some Use cases. Upload Volume/time is crucial. Scalability as well. Data generated by Automated driving applications requires large SPACE and fast Analytics in case of issues. Good performance and efficiency of the data transformation and analysis algorithm is required

5	Which QU4LITY components does your Cloud Platform interact with?	SINTEF and ATB solutions
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	Data in any format can be stored. Requirement was to have storage possibility for structured and unstructured data. In the prototype, data is provided via RESTful web API in CSV and JSON Formats to the connected components
7	How could your Cloud Platform be improved?	Provider should be from EU and not US
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	We still need on premise Infrastructure because of high demand in terms of response times
9	Add free comments here	

3.1.4 WHR: Dryer Factory Holistic Quality Platform

Table 5 - WHR Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes. Google Cloud Platform. It is providing Data Lake and analytical tool.
2	Which goals does your Cloud Platform help you achieving?	better access to manufacturing data and common application development
3	How does your Cloud Platform address ZDM needs?	Not yet implemented.
4	How do Cloud Platform performances impact your workflow?	Analytica tools and KPI visualization
5	Which QU4LITY components does your Cloud Platform interact with?	Not yet implemented.
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	No yet
7	How could your Cloud Platform be improved?	Implementing ontologies to normalize data ingestion in data lake
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	All Visualization and cross reference analytics
9	Add free comments here	

3.1.5 MON: Zero defect and Autonomous Quality in Machinery Building for Capital Goods sector

Table 6 - MON Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes, proprietary. We use RabbitMQ as message broker. Node-RED as data flow integration platform. MongoDB as message/data repository. API REST architecture to offer data from repository collections. Grafana as a process and deviation monitoring tool. The platform can connect to gather data from IIoT platforms at FAGOR ARRASATE and DANOBAT
2	Which goals does your Cloud Platform help you achieving?	Federated operations to identify anomalies and deviations in the different part of stages, IA and data analytics algorithms monitoring and gathering data from 2 manufacturing lines
3	How does your Cloud Platform address ZDM needs?	Gather data as standardized messages from different and heterogeneous industrial devices using a common message infrastructure (broker system) and architecture based on Asset Administration Shell communication specification, and present/monitor the results using Grafana. There are specific variables and data associated with the anomalies and deviation in manufacturing lines that help to carry out ZDM actions
4	How do Cloud Platform performances impact your workflow?	Clearly identify the problematics that could affect for other stages. Facilitates the monitoring of industrial processes by allowing the measurement of process times and phases. It also enables the discovery of anomalies by setting limits for critical process values. The platform has systems in place to assess whether these limits are being met.
5	Which QU4LITY components does your Cloud Platform interact with?	Our Cloud Platform is placed at the "IoT Hub/Data Lake" to support the "AI and Big Data" layer.
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	Yes, our messaging system is based on the asset administration shell communication specification to structure messages. Furthermore, API REST architecture to offer data from repository collection and interaction with standards/protocols such as OPC-UA, MQTT. Finally, the solutions are

		working to connect with IDS specification and architecture for manufacturing data spaces
7	How could your Cloud Platform be improved?	The cloud platform could be improved with features for advanced analytics using IA: Explainable AI, Transfer Learning and smart collaborations between industrial devices. Furthermore, we would like to have a stable version of IDS connectors and implementation of Asset Administration Shell specification for all of industrial devices. The implementation of secure and trustworthy federation of data infrastructure must be a must
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	IA algorithms, predictive anomalies for all of the casuistic, Federated Learning, Manufacturing Data Spaces, Autonomous maintenance
9	Add free comments here	

3.1.6 KOL: Real-time injection molding process monitoring-control

Table 7 - KOL Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	KOLEKTOR is not using cloud platforms for ZDM data processing
2	Which goals does your Cloud Platform help you achieving?	N/A
3	How does your Cloud Platform address ZDM needs?	N/A
4	How do Cloud Platform performances impact your workflow?	N/A
5	Which QU4LITY components does your Cloud Platform interact with?	N/A
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	N/A
7	How could your Cloud Platform be improved?	N/A

8	Which class of problems would you also like to address with Cloud Platform in future experiments?	N/A
9	Add free comments here	

3.1.7 THYS: Quality Management of Steering Gear based on Acoustic control

Table 8 - THYS Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	No, for now, every parameter correlation and system to monitor assets within the production line runs fully internally with the help of our current MEP and PLC systems.
2	Which goals does your Cloud Platform help you achieving?	N/A
3	How does your Cloud Platform address ZDM needs?	N/A
4	How do Cloud Platform performances impact your workflow?	N/A
5	Which QU4LITY components does your Cloud Platform interact with?	N/A
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	N/A
7	How could your Cloud Platform be improved?	N/A
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	N/A
9	Add free comments here	Nevertheless, we planned to integrate our own cloud (developed internally by Automotive headquarters) in few years in order to be able to get an overall monitoring of every production site in the world, this can also be the opportunity to use some of the features of QU4LITY to make a correlation between the entire logistic chain of internal half-finished goods suppliers.

3.1.8 AIRBUS: Trade space framework for Autonomous Quality Manufacturing Systems' Design

Table 9 - AIRBUS Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	AIRBUS is not using cloud platforms for ZDM data processing
2	Which goals does your Cloud Platform help you achieving?	N/A
3	How does your Cloud Platform address ZDM needs?	N/A
4	How do Cloud Platform performances impact your workflow?	N/A
5	Which QU4LITY components does your Cloud Platform interact with?	N/A
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	N/A
7	How could your Cloud Platform be improved?	N/A
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	N/A
9	Add free comments here	

3.1.9 GHI: Real-time cognitive hot stamping furnace 4.0

Table 10 - GHI Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes, GHI has its own Cloud platform. It is a platform of digitization, advisory and technical assistance services based on the collection of data in real time (Beyond 4.0). This 4.0 platform collects a large amount of data that is monitored and analyzed with advanced Big Data tools by the 4.0 engineering team.

2	Which goals does your Cloud Platform help you achieving?	All the information collected on the platform allows us to know the behavior of the equipment and control the processes, achieving: • Control and optimize the production process • Reduce the cost of raw material loads • Plan maintenance interventions • Identify and resolve anomalies efficiently • Reduce the risk of unexpected stops • Increase energy efficiency • Guarantee maximum performance of the facilities
3	How does your Cloud Platform address ZDM needs?	We can go from "I think" to talking about data, since complex relationships are detected, allowing processes to be optimized by making an inspection in real time.
4	How do Cloud Platform performances impact your workflow?	• Programming adapted to plant flow, minimizing delays • Control of capacities and operating times • Control of any phase of the process • Sequencing and optimal use of capacities • Production monitoring in real time, adjusting the schedule according to needs.
5	Which QU4LITY components does your Cloud Platform interact with?	• IDS connector for interoperability and secure data transmission • M3: final part quality data capture
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	At the moment it only connects with the data from the furnace
7	How could your Cloud Platform be improved?	Ideally, it should be able to control the entire production process in which GHI's furnaces are involved; that is to say, that all the information of the different assets can be collected in a single platform.
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	Take into account problems before and after the furnace and be able to relate them, thus achieving a global vision of the production processes and getting to predict problems.
9	Add free comments here	

3.1.10 RiaStone: Autonomous Quality ZDM for "Ceramic tableware Single-firing

Table 11 - RiaStone Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Due to the low latencies needed at shopfloor levels, RiaStone is not yet in the moment using cloud platforms for ZDM data processing
2	Which goals does your Cloud Platform help you achieving?	N/A
3	How does your Cloud Platform address ZDM needs?	N/A
4	How do Cloud Platform performances impact your workflow?	N/A
5	Which QU4LITY components does your Cloud Platform interact with?	N/A
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	N/A
7	How could your Cloud Platform be improved?	N/A
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	N/A
9	Add free comments here	

3.1.11 PRIMA: Additive Manufacturing Pilot Adaptive Control Technology

Table 12 - PRIMA Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Being pilot focused on machine enhancement, which is strictly dependent on client's on-premises infrastructures, PRIMA is not yet in the moment using cloud platforms for ZDM data processing
2	Which goals does your Cloud Platform help you achieving?	N/A
3	How does your Cloud Platform address ZDM needs?	N/A

4	How do Cloud Platform performances impact your workflow?	N/A
5	Which QU4LITY components does your Cloud Platform interact with?	N/A
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	N/A
7	How could your Cloud Platform be improved?	N/A
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	N/A
9	Add free comments here	

3.1.12 Danobat: Digital Machine for zero-defects at high precision cutting/grinding

Table 13 - Danobat Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes, DANOBAT works closely with Savvy Data Systems, a technological start-up focused on machine-monitoring and data analytics. In conjunction with them, DANOBAT has developed a Smart Box (edge), an industry-ready box for gathering machine data and stores these data in the Industrial Cloud platform of Savvy Data Systems so that it can be processed and analyzed.
2	Which goals does your Cloud Platform help you achieving?	The cloud gathers historical data from the machines. Data about the behavior of the machine can be shared in real time and historical data is used to analyze and develop functionalities about machine and quality.
3	How does your Cloud Platform address ZDM needs?	The platform is used to develop and deploy functionalities to ensure machine availability and production quality
4	How do Cloud Platform performances impact your workflow?	When the machine is connected to the Smart Box, large amounts of machine parameters data are continuously sent to the cloud in order to obtain the necessary information for making decisions that affect machine performance and availability (machine

		components life span, machine alarms, part defects...).
5	Which QU4LITY components does your Cloud Platform interact with?	It interacts with each of the machines of Danobat and other production software like the Danobat Control system. It also interacts with AI functions from ATLANTIS and line monitoring system from MGEP
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	Smart Box provide a REST API, which enables the automatic connection of third-party software systems to the Big Data environment managed by the Industrial Cloud platform of Savvy Data Systems. Thanks to this API, different types of systems such as MES, ERP, or similar, can automatically consume data safely and efficiently.
7	How could your Cloud Platform be improved?	Even though the API design is based on REST and it operates over HTTPS protocol, different types of systems can consume data collected by the platform in an automated way, this means that security must be a priority and should be improved.
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	Taking into account the previous question, security and log access control through the API should be a point to be addressed in the near future.
9	Add free comments here	

3.1.13 FAGOR: Zero-Defects Manufacturing Digital Press Machine

Table 14 - FAGOR Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	Yes. Own data platform that is being developed by FAGOR and IKERLAN. This platform, uses additionally cloud solutions provided by AZURE.
2	Which goals does your Cloud Platform help you achieving?	Monitoring and analysis of performance and status of different components from sensors to plants.
3	How does your Cloud Platform address ZDM needs?	The own cloud platform is continuously being developed and enhanced in order to provide solutions to improve the customer satisfaction and market positioning. This

		requires targeting ZDM. In terms of ZDM, the platform has been extended in QU4LITY with components and views that perform data analytics over the data from the machines, and that facilitate to the stakeholders adopting decisions and configuring system that directly impact in the quality and also reducing defectives and waste.
4	How do Cloud Platform performances impact your workflow?	Performance of the cloud platform is a key factor for FAGOR since it is used to monitor data and performance of different machines and sensors. Using this platform, FAGOR is able to improve remote management of their different machines, provide better customer service, avoid and minimize unplanned stops and also detect anomalies. The platform will be accessed by customers of FAGOR and consequently, the performance also impacts in customer satisfaction and experience.
5	Which QU4LITY components does your Cloud Platform interact with?	FALINK Cloud platform of FAGOR ARRASATE interacts with IKCLOUD, IKSEC, and VTT-OpenVA components that have been developed during the QU4LITY project. Additionally, FALINK platform itself has been extended with additional features for data analytics focused on ZDM and increasing quality.
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	For interoperability different components of the cloud solution interact with different standards/protocols such as OPC-UA, MQTT, REST API. The data model has been specified by FAGOR and is based on previously existing data models.
7	How could your Cloud Platform be improved?	The cloud platform could be improved with features for advanced analytics using IA: Explainable AI, Transfer Learning,
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	Enhance platform and advance towards prescriptive maintenance. Go towards autonomous maintenance and management of machines and plants.
9	Add free comments here	

3.1.14 GF: Digital machine and part twins for zero defect manufacturing

Table 15 - GF Tech-Survey Questionnaire results

#	Question	Answer
1	Are you already using some Cloud Platform? If yes, please describe the main technologies and technical advantages they provide in your manufacturing setting	MS Azure. Our Digital Infrastructure is set up in this environment, we develop applications for our connected machines and gather also manufacturing data from our production factories.
2	Which goals does your Cloud Platform help you achieving?	Provide services to our customers, by gathering data and sending back information and analytics for improving the customer processes
3	How does your Cloud Platform address ZDM needs?	It helped to gather process and component from our machine park, delivers analytics results for predicting quality and maintenance in seamlessly available applications
4	How do Cloud Platform performances impact your workflow?	It provides manufacturing KPIs in real time and helps to improve them with the help of predictive models
5	Which QU4LITY components does your Cloud Platform interact with?	Main QU4LITY components are our applications for predictive quality, predictive maintenance and multi-technology process optimizer.
6	Does your Cloud Platform adhere to any standardized data model/data protocol to interoperate with other components?	OPC UA and UMATI protocols
7	How could your Cloud Platform be improved?	Mainly adding high level analytics applications and security barriers which are recognized by customers
8	Which class of problems would you also like to address with Cloud Platform in future experiments?	Taking control of manufacturing processes in real time, with intelligent interaction with machine and robot controllers
9	Add free comments here	

3.1.15 Overall assessment of Tech-Survey results

Analyzing pilots' answers, it's clear that more emphasis on real-time network solution has to be put. The major drawback to adopting Cloud Solutions is, in fact, **network latency** which, in manufacturing environments, is critical to satisfy production needs. Congested networks, in fact, make impossible to reprocess salient information in a short period of time forcing plant and IT managers to rely on on-premises ad-hoc network solutions. Along with network performances, **interoperability across multiple data models** is crucial for building a comprehensive ZDM ecosystem for AQ as different solutions/assets, from different vendors, have to interoperate and communicate seamlessly.

Last but not least, fog/edge computing seems an interesting topic to the pilots as it enables to run AI models locally, boosting production and improving KPIs, without relying on the Cloud to close the loop. As a matter of fact, when speaking about fog/edge computing we also refers to hybrid Cloud Platforms where training of models is performed on high performance computing infrastructure and predictive and prescriptive analysis are run locally.

Addressing these needs, QU4LITY HPC Infrastructure and QU4LITY Cloud Solutions have been built keeping in mind two key factors: responsiveness and simplicity. The former will address the easing the design and deployment of resource intensive solution on a HPC cloud (once trained the model, it may be pushed back at the edge for fast processing close to the data generation). The latter, instead, goes behind the cloud services provisioning, enabling an easy interoperable mechanism for ZDM processes and solutions based on QU4LITY data models and approach.

A more in-depth technical description of the two solutions will be given in the following two sections.

4 QU4LITY High Performance Computing Infrastructure

4.1 Main Features and Functional Description

QU4LITY HPC solution at JSI provides the possibility of leveraging High Performance Computing (HPC) for training of deep neural networks, as discussed in D3.3. HPC will play an increasingly important role in helping deep learning achieve the next level of innovation fueled by neural network models [VanEssen15].

As part of the realization process, JSI has provided a service for seamless utilization of the HPC, when the data is available. The service is contract-based. As discussed in D3.3, only the training of the network is performed on the cluster, while the prototyping is done on local workcell workstations. The latter applies also for testing and deployment. This deliverable contains a blueprint for how to develop and train new models for ZDM on HPC infrastructure, a description of the implemented interfaces, and an example model that was developed using HPC to accelerate visual quality control by providing a feedback loop that predicts possible product that may arise in the production process.

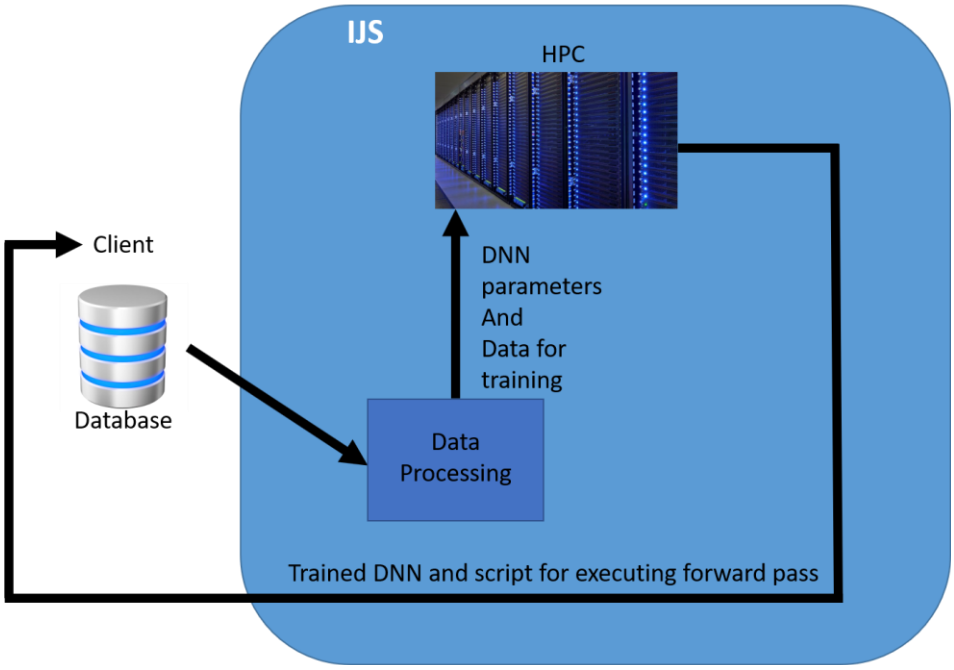

Figure 3 - HPC service schematic

Given annotated process input and output data, the provided service trains a deep neural network model that predicts the output based on new, previously unseen input data. The trained models can be used in production processes, just like in the Kolektor Pilot. In this pilot, the trained model is used to predict the likelihood of the known errors based on a set of production process parameters, i.e., it acts as a classifier that predicts the error most probable to occur from the given set of production parameters. Then, this can be exploited in the final visual quality control, where the potential errors with the highest probability are checked first. This reduces the average inspection times, because as soon as an error is confirmed with visual quality control, the process is stopped, and the product is discarded.

4.2 Technologies and Architecture

The implementation is running on the JSI high performance cluster. The cluster is called "Nova skupna gruča" (NSC) and is available to the QU4LITY Project. The NSC computing cluster operates using the Nordugrid Advanced Resource Connector (ARC) grid middleware and consists of 1984 cores, 16 GPUs, and has 9216 GB of RAM. Further details on the hardware setup were already provided in D3.3.

The service of training deep neural network models for ZDM or any other manufacturing tasks can be initiated by the client. After initial discussion with JSI, the client uploads the agreed-upon data via a web-service, potentially exploiting QU4LITY Cloud Solutions further described in Chapter 0.

The data is then processed by the personnel at JSI, and a dedicated script is implemented and uploaded to the cluster, where the input-to-output deep neural network model is trained. This cannot be automated because the data and neural network models are process-specific. The data needs to be processed, the network size and architecture should be adapted to the specific use case. Finally, the cyber-security measures require proper certification and identification prior to every access to the HPC. In the following we provide the blueprint to obtain a neural network ZDM prediction model and utilize the described HPC infrastructure to train it.

The architecture of the neural network is decided by the personnel at JSI. If the input data are just numeric values, a fully connected network is usually trained (but based on the further information about the data by the client, this can be easily changed). Different network sizes with different cost functions are tested until the desired performance is reached. In the pre-processing stage, the pairs of input/output data are prepared and transformed in the right format that can be used by the neural network training procedure. The preparation of data involves the normalization of data and its proper weighting so that different types of data given in different units can be properly taken into account by the training procedure. The starting point is the data provided by the client, which should be a file with comma-separated-values (.csv), with each row in the file a new data instance, and each column a different data parameter. A header line with names of the parameters should be on top of the data. The input and output data should both be provided in the same file, with input and output data marked in the header line.

On the other hand, if the input data are images and output some desired classifiers or parameters, convolutional neural networks are usually applied as they have proven to be the most robust for such data. Preparation of the input data is then done by resizing the images to the smaller and fixed number of pixels so that the trained network has smallest number of inputs but still does its task [Ridge20]. Just like in the case of other types of data, values of all the inputs and outputs are normalized and weighted. The data provided by the client should contain a folder with images labeled by numeric values, e.g. 00001.png, etc. It should also contain a file that relates the images with the output data, so that supervised training of the neural network can be performed.

After training, the trained neural network model is encapsulated in a Docker container and sent back to the client, who can utilize the model without installing any additional software.

4.3 Access Instructions

The service is contract-based, meaning that it needs to be negotiated and agreed with JSI. After an agreement, the means of data transfer are coordinated. Once the data is provided, the deep neural network model is proved by the JSI in the coordinated time, which depends on the complexity of the data.

4.4 Quick Start Guide

From the client side, the effort is in providing the input and output data. The data is different for every task. Depending on the customer contract, the JSI personnel can support transforming the data in the correct format. See Section 4.5 for the Kolektor pilot example.

In the following we provide a blueprint for the necessary steps on the service side.

Before working with the JSI NSC HPC, we need to obtain access. Several conditions need to be fulfilled through several steps for cybersecurity. This involves obtaining a certificate, in-person verification and confirmation by a virtual organization.

Once the request is approved, a certificate in .p12 format is issued to enable access to the HPC. This certificate must be copied in the home directory of the host computer.

Next, the host computer from which we issue commands to run a job on the cluster must be setup. Nordugrid repositories are added to the list of our sources and the appropriate key is copied:

wget -q -O - http://download.nordugrid.org/RPM-GPG-KEY-nordugrid-6 | apt-key add -

Finally, all the necessary packages are installed.

After this has been set up, we also need to register with nsc.ijs.si/VO virtual organization, which acts as a method to set user rights for SLING (Slovenian national super-computer network) as part of the JSI cybersecurity measures for authorization and verification.

In order to have these steps automated, we prepared a docker container with Ubuntu 18.04. A screenshot of the docker is shown in Figure 4.

Dockerfile (~/hpcdocker) - gedit

```
FROM ubuntu:focal

RUN apt-get -y update && apt-get -y install wget apt-utils gnupg2
RUN echo "deb https://download.nordugrid.org/repos/6/ubuntu focal main" >> /etc/apt/sources.list
RUN echo "deb-src https://download.nordugrid.org/repos/6/ubuntu focal main" >> /etc/apt/sources.list
RUN echo "deb https://download.nordugrid.org/repos/6/ubuntu focal-updates main" >> /etc/apt/
sources.list
RUN echo "deb-src https://download.nordugrid.org/repos/6/ubuntu focal-updates main" >> /etc/apt/
sources.list
RUN wget -q -O - http://download.nordugrid.org/RPM-GPG-KEY-nordugrid-6 | apt-key add -
RUN echo "deb http://repository.egi.eu/sw/production/cas/1/current egi-igtf core" >> /etc/apt/
sources.list
RUN wget -q -O - https://dist.eugridpma.info/distribution/igtf/current/GPG-KEY-EUGridPMA-RPM-3 | apt-
key add -

#needed for tzdata
ENV DEBIAN_FRONTEND="noninteractive" TZ="Europe/Ljubljana"
RUN apt-get -y update && apt-get -y install nordugrid-arc-client \
                nordugrid-arc-plugins-needed \
                nordugrid-arc-plugins-globus
RUN apt-get -y install ca-policy-egi-core fetch-crl

RUN mkdir ~/.arc
WORKDIR ~/.arc
#COPY usercert.p12 .
#RUN openssl pkcs12 -in usercert.p12 -clcerts -nokeys -out usercert.pem
#RUN openssl pkcs12 -in usercert.p12 -nocerts -out userkey.pem
COPY usercert.pem .
COPY userkey.pem .
RUN chmod 400 userkey.pem
RUN chmod 644 usercert.pem
RUN mkdir -p vomsdir
RUN echo "/C=SI/O=SiGNET/O=SLING/CN=voms.sling.si\n/C=SI/O=SiGNET/CN=SiGNET CA" > vomsdir/
voms.sling.si.lsc
RUN echo '"gen.vo.sling.si" "voms.sling.si" "15001" "/C=SI/O=SiGNET/O=SLING/CN=voms.sling.si"
"gen.vo.sling.slsi"' > vomses
RUN echo '"nsc.ijs.si" "voms.sling.si" "15004" "/C=SI/O=SiGNET/O=SLING/CN=voms.sling.si"
"nsc.ijs.si"' >> vomses
```

Figure 4 - Docker file that contains all the instructions for JSI NSC HPC access

After we have set up JSI NSC HPC access, we can start preparing the neural networks for training. For this purpose, the data must be stored in the /src folder within our working directory. For example, in Figure 5 we can see that the directory includes the file "src/NN_python_data.mat", where the data is stored. Besides input data, this file also contains the associated outputs of the neural network, for example annotations.

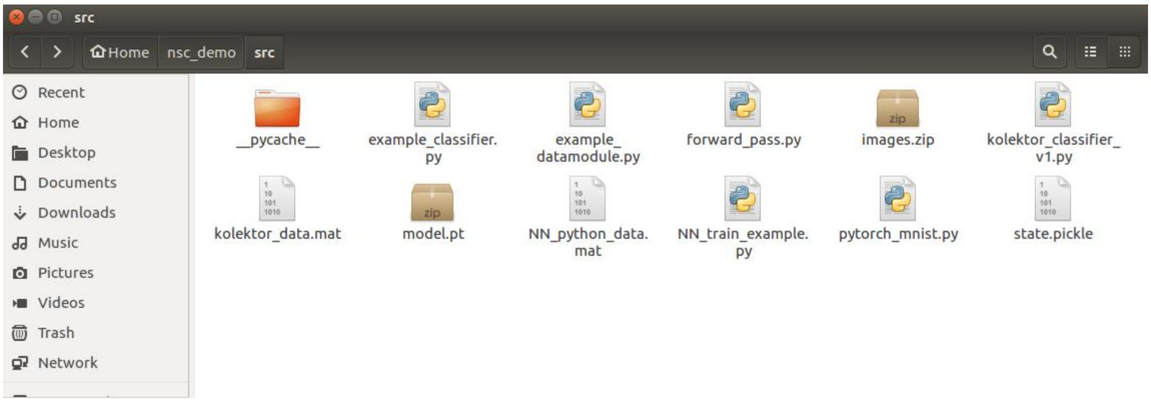

Figure 5 - Source folder with the data. Note the image.zip file and NN_python_data.mat file

If the images are also needed for the neural network training, they are provided in the file "src/images.zip", where images are stored as an image sequence (see Figure 6).

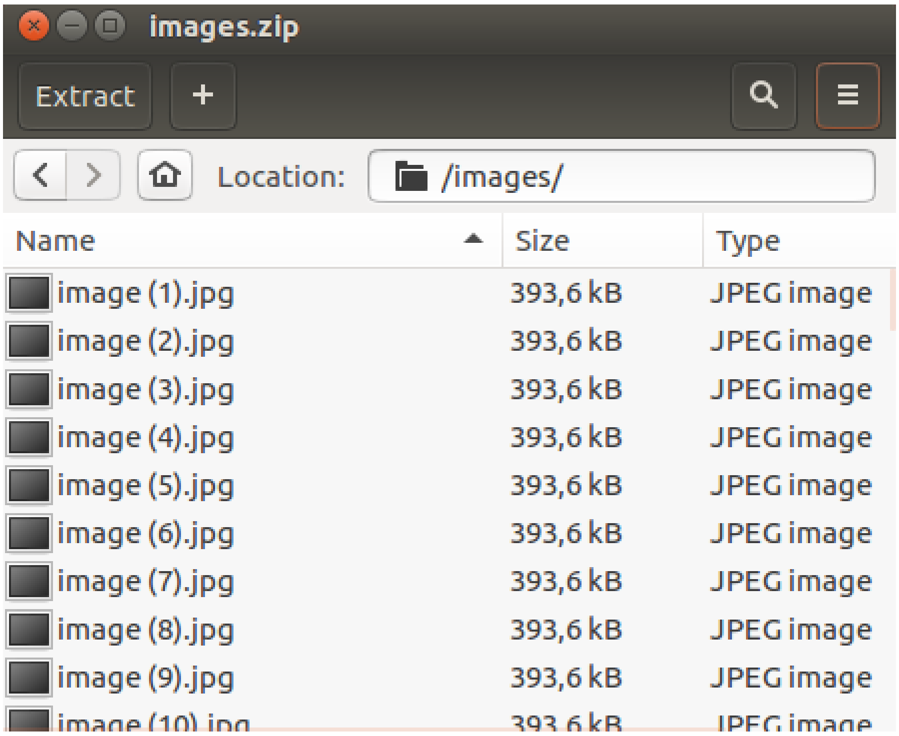

Figure 6 - Content of images.zip archive

The first step in preparing the neural network is to define the job name, files that will be sent to the cluster (installation files, scripts and also the data for training), bash script to execute, output files and some settings for the HPC (for example: how much memory it can take or how long it can run, etc.). This is accomplished by providing a .xrsl file shown in Figure 7. This file is specified by the JSI personnel.

```
&
(rsl_substitution=("JOBID" "NN_train_example"))
(count = 1)
(countpernode = 1)

(jobname = $(JOBID))
(inputfiles =
     ($(JOBID).sh scripts/$(JOBID).sh)
     ("Miniconda3-latest-Linux-x86_64.sh" "resources/Miniconda3-latest-Linux-x86_64.sh")
     ($(JOBID).py src/$(JOBID).py)
     ("NN_python_data.mat" "src/NN_python_data.mat")
     ("example_classifier.py" "src/example_classifier.py")
     ("example_datamodule.py" "src/example_datamodule.py")
     ("torch-1.8.0+cpu-cp37-cp37m-linux_x86_64.whl" "torch/torch-1.8.0+cpu-cp37-cp37m-linux_x86_64.whl")
     ("torchvision-0.10.0-cp37-cp37m-manylinux1_x86_64.whl" "torch/torchvision-0.10.0-cp37-cp37m-manylinux1_x86_64.whl")
     ("typing_extensions-3.10.0.0-py3-none-any.whl" "torch/typing_extensions-3.10.0.0-py3-none-any.whl")
     ("pytorch_lightning-1.4.2-py3-none-any.whl" "torch/pytorch_lightning-1.4.2-py3-none-any.whl")
     ("pandas-1.3.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl" "torch/pandas-1.3.2-cp37-cp37m-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl")
     )
(executable = scripts/$(JOBID).sh)
(outputfiles =
     (output.zip "")
     )
(stdout=$(JOBID).log)
(join=yes)
(gmlog=log)
(memory=4000)
(runTimeEnvironment = "APPS/BASE/TORCH-GPU")
(walltime="60")
(queue="gridlong")
```

Figure 7 - .xrsl file for setting up the JSI HPC task

The second important issue is the bash script that needs to be executed. This script sets up the environment and installs all the libraries that are necessary for the training of the neural network. Through this script, we can for example run a python program that defines and trains the neural network model. After the training has finished, this script renames and zips the result of training so that it can be downloaded for further processing. We installed PyTorch deep learning library for neural network training on the HPC infrastructure.

To obtain access to the HPC, the permission and SSH keys are needed. Once the access is approved by the administrator, we open the communication process using arcproxy:

```
(base) zvezdan@zvezdan-desktop:~$ arcproxy -S nsc.ijs.si
Enter pass phrase for private key:
Your identity: /C=SI/O=SiGNET/O=IJS/OU=E1/CN=Zvezdan Loncarevic
Contacting VOMS server (named nsc.ijs.si): voms.sling.si on port: 15004
Proxy generation succeeded
Your proxy is valid until: 2021-08-26 21:59:21
```

Figure 8 - Communication opening process with arcproxy

The next step is to submit the job to the HPC. This is done by sending the aforementioned .xrsl file:

```
(base) zvezdan@zvezdan-desktop:~/nsc_demo$ arcsub -c nsc.ijs.si jobs/NN_train_example.xr
sl
Job submitted with jobid: https://nsc.ijs.si:443/arex/OtuLDmHXPdznWeBumqyIfvmm6BJ3FoABFK
DmajFKDmuZLKDma2MEam
```

Figure 9 - Job submission

Each submitted job gets its own unique ID. The status of all of running jobs can be checked using arcstat:

```
(base) zvezdan@zvezdan-desktop:~/nsc demo$ arcstat -a
Job: https://nsc.ijs.si:443/arex/OtuLDmHXPdznWeBumqyIfvmm6BJ3FoABFKDmajFKDmuZLKDma2MEam
 Name: NN_train_example
 State: Running
```

Figure 10 - Health check using arcstat

Once the status of the job has switched to "Finished", which depends on the scheduling [Fan21]

```
(base) zvezdan@zvezdan-desktop:~/nsc demo$ arcstat -a
Job: https://nsc.ijs.si:443/arex/OtuLDmHXPdznWeBumqyIfvmm6BJ3FoABFKDmajFKDmuZLKDma2MEam
 Name: NN_train_example
 State: Finished
 Exit Code: 0
```

Figure 11 - Job finished

the results are downloaded from the HPC server using the arcget command and the unique job ID:

```
(base) zvezdan@zvezdan-desktop:~/nsc_demo$ arcget https://nsc.ijs.si:443/arex/OtuLDmHXPd
znWeBumqyIfvmm6BJ3FoABFKDmajFKDmuZLKDma2MEam
Results stored at: OtuLDmHXPdznWeBumqyIfvmm6BJ3FoABFKDmajFKDmuZLKDma2MEam
Jobs processed: 1, successfully retrieved: 1, successfully cleaned: 1
```

Figure 12 - Results fetch using arcget

This creates the folder with the same name as the job ID, where the log and the training results are stored:

Figure 13 - Results are stored in the marked folder, named after the job ID. Other folders contain supplementary files uploaded to the HPC

The trained neural network model is finally stored in the "output.zip" folder.

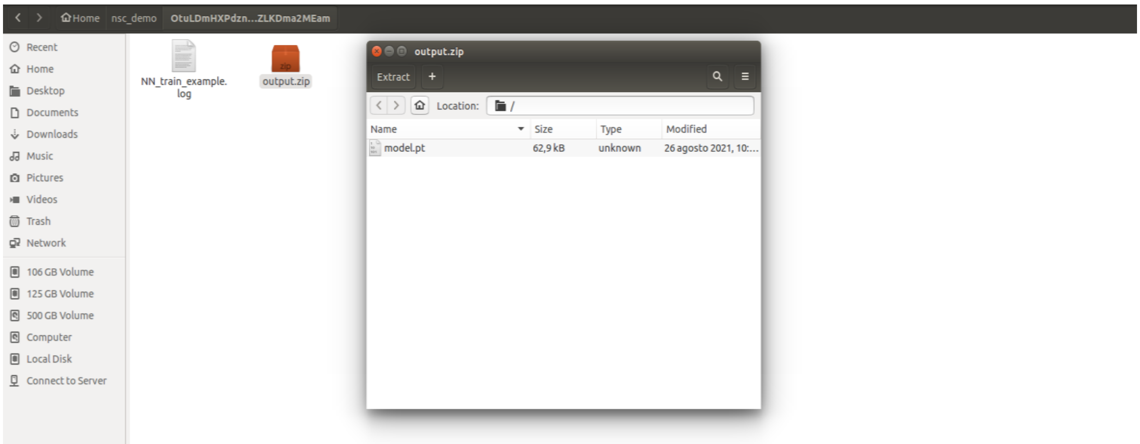

Figure 14 - Trained model is stored in output.zip. A log of all the commands executed on the HPC is provided as well

The model can be inspected on a local computer using a python script to predict the output values given the appropriately formatted input data. As discussed in D3.3, a similar functionality is provided by the GUI shown in Figure 15. Furthermore, the training results and progress can be post-festum viewed and analyzed on the local workstation using TensorBoard.

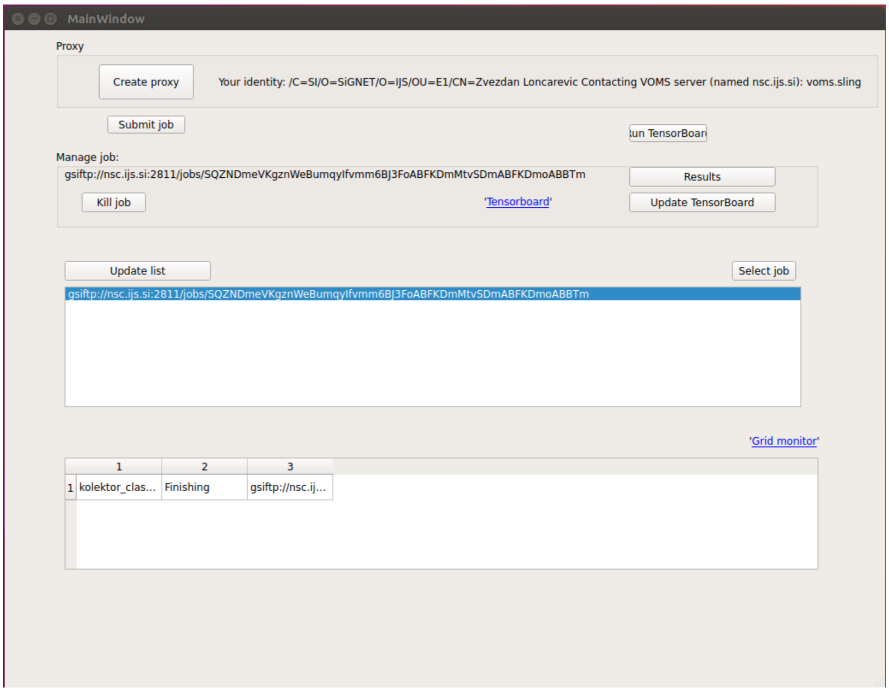

Figure 15 - GUI for JSI HPC access that provides the same command-line functionality as described above

The resulting network is sent to the user in the docker container. We use docker containers to ensure that all the dependencies and libraries are installed and that the client can easily set the trained neural network to output the predicted values.

4.5 Validation Campaign

The validation campaign is part of the Kolektor pilot, where we work on adaptive visual quality control to reduce the scrap rate of the Kolektor production line. The main aim is to accelerate the visual quality control by implementing a feedback loop that predicts possible product errors based on advanced analytics and artificial intelligence. By predicting the possible errors, we can generate a set of viewpoints for inspection on the fly. The robot then guides the camera to the set of predicted vantage points where images are taken and inspected.

CyclePoint.f071.Value	CyclePoint.f9002.Value	CyclePoint.T821I.Value	CyclePoint.T831I.Value
2	92.051	24	170
7	92.056	24	170
15	92.064	25	170

Figure 16 - Snippet of production parameters from Kolektor, as shown in MS Excel

Next, we explain the data needed to train the neural network for predicting the vantage points. Figure 16 shows an excerpt of production parameters data of the Kolektor Pilot. There are 49 parameters in total, out of which we used 26 to generate the input data for neural network training, because the other parameters are not relevant. The 26 relevant parameters are:

1. CyclePoints.MOULDING_NEST.Value
2. CyclePoint.t4012.Value
3. CyclePoint.V4065.Value
4. CyclePoint.p4055.Value
5. CyclePoint.p4072.Value
6. CyclePoint.t4018.Value
7. CyclePoint.V4062.Value
8. CyclePoint.t4015.Value
9. CyclePoint.f071.Value
10. CyclePoint.T821I.Value
11. CyclePoint.T831I.Value
12. CyclePoint.T832I.Value
13. CyclePoint.T834I.Value
14. CyclePoint.T837I.Value
15. CyclePoint.T840I.Value
16. CyclePoint.T841I.Value
17. CyclePoints.MARKING_NEST.Value CyclePoints.MARKING_DMC_GRADE.Value
18. CyclePoints.COOLING_NEST.Value CyclePoints.BLASTING_NEST.Value
19. CyclePoints.EOLT_NEST_INSERTION.Value
20. CyclePoints.EOLT_NEST_WITHDRAW.Value
21. CyclePoints.EOLT_PD_DIA_AVG.Value
22. CyclePoints.EOLT_PD_DIA_LIM_HI.Value
23. CyclePoints.EOLT_PD_DIA_LIM_LO.Value
24. CyclePoints.EOLT_PD_DIA_MAX.Value

25. CyclePoints.EOLT_PD_DIA_MIN.Value
26. CyclePoints.EOLT_PD_ETALON_DIA_VAL.Value

This translates to:

1. Number of nest in tool
2. Time of cycle – actual value
3. Switch after volume – actual value
4. Maximal injection pressure – actual value
5. Switch after pressure – actual value
6. Injecting time – actual value
7. Material cushion – actual value
8. Time of dosing – actual value
9. Machine cycle counter
10. Temperature of feeder housing -actual value
11. Heating current tool 1 – actual value
12. Heating current tool 2 – actual value
13. Heating current tool 4 – actual value
14. Heating current tool 7 – actual value
15. Heating current tool 10 – actual value
16. Heating current tool 11 – actual value

26x7864 double

	1	2	3	4	5	6	7	8	9
1	-0.9957	-0.9957	-0.9978	-0.9935	-0.9914	-0.9978	-0.9957	-0.9978	-0.9957
2	-0.7365	-0.7257	-0.7257	-0.7257	-0.7279	-0.7257	-0.7257	-0.7257	-0.7257
3	-0.9914	-0.9914	-0.9914	-0.9914	-0.9914	-0.9914	-0.9914	-0.9914	-0.9914
4	0.0562	-0.0713	-0.0734	-0.0713	-0.0799	-0.0605	-0.0324	-0.0518	-0.0605
5	-0.1339	-0.1382	-0.1339	-0.1382	-0.1274	-0.1382	-0.1382	-0.1339	-0.1382
6	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611	-0.9611
7	-0.9935	-0.9957	-0.9957	-0.9957	-0.9935	-0.9957	-0.9957	-0.9957	-0.9957
8	-0.9136	-0.9114	-0.9114	-0.9114	-0.9136	-0.9114	-0.9114	-0.9114	-0.9114
9	-0.9957	-0.9849	-0.9676	-0.9849	-0.9719	-0.9827	-0.9762	-0.9698	-0.9827
10	-0.9482	-0.9482	-0.9460	-0.9482	-0.9460	-0.9482	-0.9482	-0.9460	-0.9482
11	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
12	-0.6307	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
13	-0.6307	-0.6328	-0.6350	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
14	-0.6307	-0.6328	-0.6350	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
15	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
16	-0.6307	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328	-0.6328
17	-0.9525	-0.9525	-0.9546	-0.9503	-0.9482	-0.9546	-0.9525	-0.9546	-0.9525
18	-0.9784	-0.9784	-0.9784	-0.9784	-0.9784	-0.9784	-0.9784	-0.9784	-0.9784
19	-0.9957	-0.9935	-0.9935	-0.9935	-0.9892	-0.9914	-0.9935	-0.9957	-0.9914
20	-0.9654	-0.9914	-0.9957	-0.9978	-0.9762	-0.9870	-0.9914	-0.9698	-0.9827
21	-0.9914	-0.9827	-0.9935	-0.9957	-0.9978	-0.9849	-0.9827	-0.9849	-0.9914
22	-0.9914	-0.9827	-0.9935	-0.9957	-0.9978	-0.9849	-0.9827	-0.9849	-0.9914
23	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591
24	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591
25	-0.9591	-0.9591	-0.9591	-0.9590	-0.9590	-0.9591	-0.9590	-0.9591	-0.9590
26	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591	-0.9591
27									
28									
29									
30									

Figure 17 - Extract of normalized input data, with 26 input parameters and 7864 input samples, in Matlab

The other parameters are self-explainable and were in this manner presented by Kolektor. The irrelevant parameters include absolute time of day, date, etc. Figure 17 shows an excerpt of normalized data, with values between -1 and 1.

The output data in the Kolektor pilot correspond to the type of errors that occur in visual quality control. They are thus related to the viewpoints from which the images have been captured. The dimension of the output data fits with the number of most common errors in the Kolektor Pilot production line. An image of a common error is shown in Figure 18.

The network we train on the cluster is multimodal and takes two types of inputs. One input is the image of the product at a certain stage of the production, always from the same viewpoint. This should not be mistaken with the final visual quality control, which is at the end of the production line. The other input, of a different type, are the production parameters.

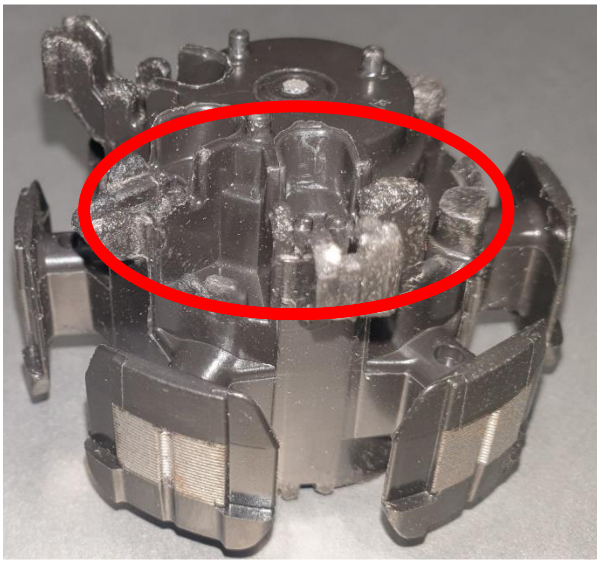

Figure 18 - Image of the product from the Kolektor Pilot with the error of the material being porous at the top

To combine the two inputs, we use a common structure for both of the input types, and then concatenate them as shown in Figure 19. The concatenated layer combines features extracted from the images as well as features extracted from the parameters and is followed by several fully connected layers and finally a softmax layer. The loss function plays an important role in classification. Cross-Entropy Loss function is typically used to train the fully-connected layers.

One part of the network takes images as input. Typical structures for image classification include several convolutional layers, each followed by a rectified linear unit (relu) layer and a pooling layer. Typically, 5-10 such combinations are used. The output is then flattened to get a vector of features. This vector is then concatenated with the features from the other part of the network.

For the part of the network that takes production process parameters as input, we take the dimensionality of the data to define the input and we define the number, dimensionality and activation functions of the hidden layers. For the concrete example we could be working with 3 hidden layers of size 50, 80 and 100 and assigned sigmoid activation functions in the first three and then a relu.

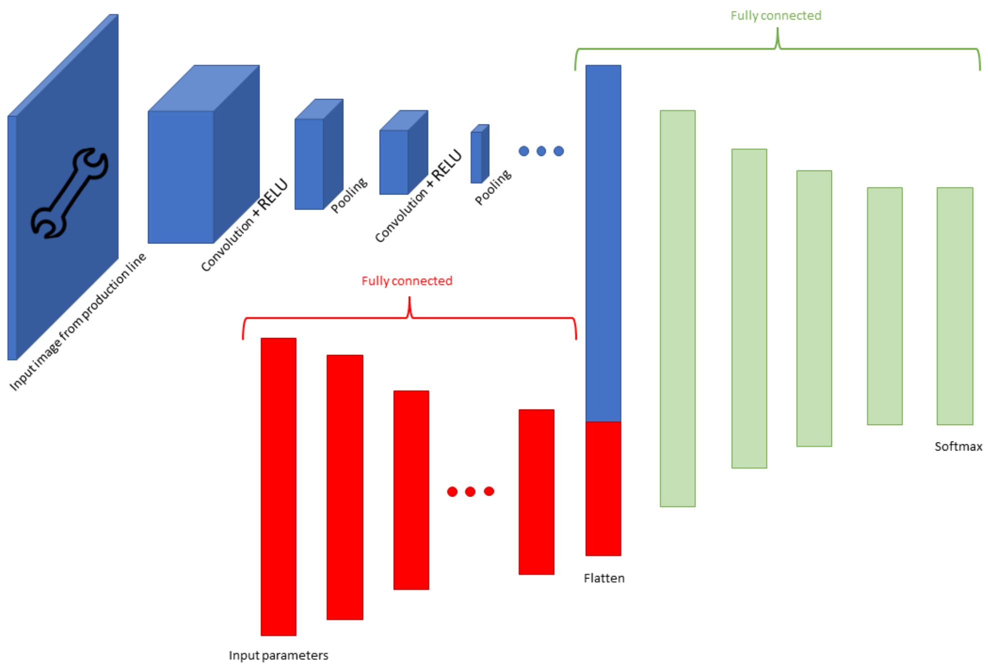

Figure 19 - Proposed structure of the neural network. Once trained, this constitutes the neural network ZDM prediction model

The output of the last hidden layer is then concatenated with the last layer of the image part of the network, as shown in Figure 19. The following hidden layers sizes and their number depends on the actual use-case, but 3-10 layers with up to several hundred neurons would typically suffice. The final output layer size is defined by the actual task. In our example, if 10 typical errors were expected, the output size would be 10.

Our initial results show that training is possible. We are, however, still collecting data with Kolektor for more complete training. The final visual quality control will be implemented when enough data and faulty parts are provided by Kolektor.

4.6 Future Works

We are currently exploring the possibilities for the increased automation of the provided blueprint, such as for automated data processing to generate the appropriate inputs and outputs for the neural network models. Note, however, that a complete automation is not possible because the data are task dependent. We have also been working on at least partial automation of the Deep Neural Network (DNN) architecture specification.

5 QU4LITY Cloud Solutions

5.1 Main Features and Functional Description

QU4LITY Cloud Solutions provide a seamless solution to exchange data using the QU4LITY Ontology Model (based on the R-MPFQ model developed in WP2), enabling a semantic enriched data exchange from on-premises data lakes to QU4LITY Cloud Data Storage using a time-based approach. Among the different needs and requirements that QU4LITY Pilots have highlighted, two solutions stand up for their interoperability and simplicity: QU4LITY Cloud Bridge and Q-Ontology Enabler. The former, the QU4LITY Cloud Bridge, eases the exchange of data across existing on-premises data lakes and QU4LITY Cloud Data Storage by the means of REST API. The latter, the so-called Q-Ontology Enabler, delivers a powerful set of scripts to enable semantic interoperability between legacy data and newly engineered QU4LITY R-MPFQ Ontology.

The two aforementioned QU4LITY Cloud Solutions, further described in the followings 5.2.1 and 5.2.2, delivers a solid basis to build a comprehensive ecosystem, from data gathering to data enrichment and visualization to satisfy QU4LITY pilot's needs, relying on distributed infrastructure.

5.2 Technologies and Architecture

The QU4LITY Cloud Solutions have been built keeping in mind two key factors: responsiveness and interoperability, thus, for this reason, the technological choices made during development phase have been crucial to deliver a set of high-quality digital enablers for AQ and ZDM.

The **QU4LITY Cloud Bridge** is built on top of Node.js, an event-driven, non-blocking I/O, JavaScript runtime, specifically designed for building fast and scalable network applications, perfect for data-intensive real-time applications that run across distributed devices. On top of that, Sequelize has been chosen to handle database connection at application level.

Sequelize is an open-source ORM Object/Relational Mapper) that provides easy access to MySQL, MariaDB, SQLite or PostgreSQL databases by mapping database entries to objects and vice versa. It has very powerful migrations mechanism that can transform existing database schema into a new version. It also provides database synchronization mechanisms that can create database structure by specifying the model structure.

The **Q-Ontology Enabler**, instead, consists of a set of tools/script built with Python, an interpreted, interactive object-oriented programming language suitable (amongst other uses) for distributed application development. Python makes the Q-Ontology Enabler a very versatile set of tools as every adjustment at code level does not require any compilation but a simple re-execution of the python interpreter to restart migration of existing on-premises data lakes towards QU4LITY Relational Ontology Model.

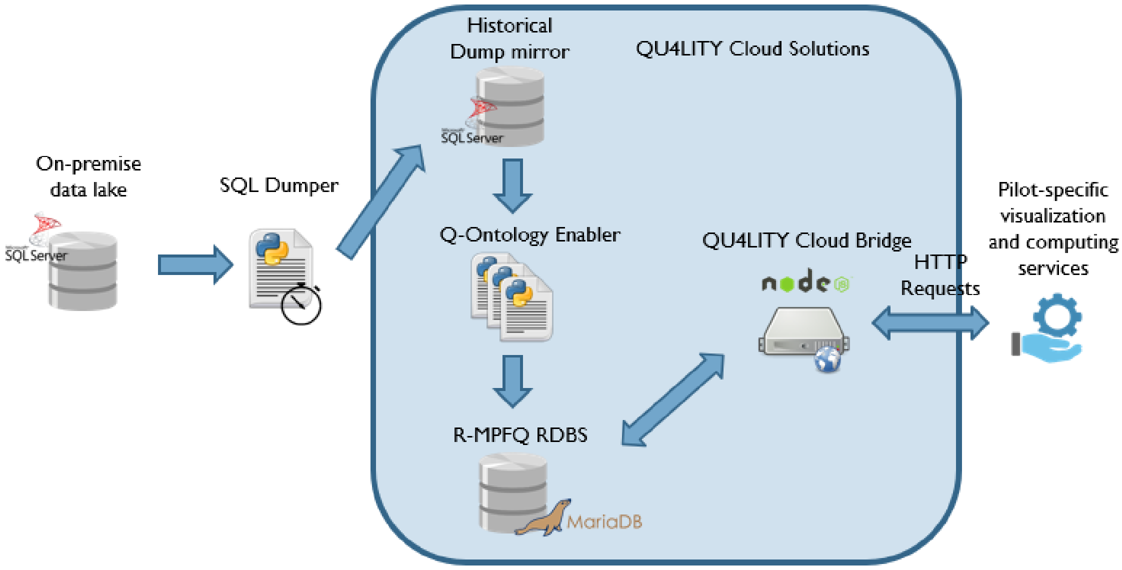

Figure 20 - QU4LITY Cloud Solutions Architecture

5.2.1 QU4LITY Cloud Bridge

QU4LITY Cloud Bridge provides a seamless solution to exchange data using the QU4LITY Ontology Model (based on R-MPFQ), enabling a semantic enriched data exchange from on-premise data lake to QU4LITY Cloud Data Storage using a time-based approach. The developed QU4LITY Cloud Bridge offers a REST API layer to ease the interfaces with other processing and visualization components taking care of any data decoding/encoding needs (i.e. IEEE754 data encoding [Brain15])

5.2.1.1 PaaS Model

A Platform as A Service (PaaS) provides to the consumer the capability to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. At the disadvantage of the lack of operational features, PaaS Model offers a reduced overall complexity and an easier maintenance and overall development of the applications that it provides.

QU4LITY Cloud Bridge, thus, offers to consumers a cloud infrastructure to support and enrich ZDM processes based on cloud-based deployments of enterprise systems and quality management applications, providing a medium-high level of abstraction of the underlying infrastructure.

5.2.2 Q-Ontology Enabler

Q-Ontology Enabler is a set of python scripts which ease the migration to the QU4LITY relational database performing a ETL processes on existing on-premises infrastructures. The relational transposition of R-MPFQ model does not represent a new Manufacturing Execution System (MES), but a way to organize in the most logical and structured way possible a varied reality although it may become that one day in the future.

5.2.2.1 R-MPFQ Ontology Model

The management of the complexity of manufacturing processes is one of the most interesting challenges in the industrial world in contiguous transformation. To do this arises the need to develop digital model able to represent reality, to process data, states and processes, and the quality trend evolution. This done through new forms of functional, logical and operational correlations.

The manufacturing process in household appliances, can be represented in these macro areas:

- Incoming Materials
- In-House Parts
- Pre-Assembly
- Assembly
- 100% Testing
- Repairing
- Statistical Test and Packaging

These macros can see the areas integrated into a single vision through the following outline:

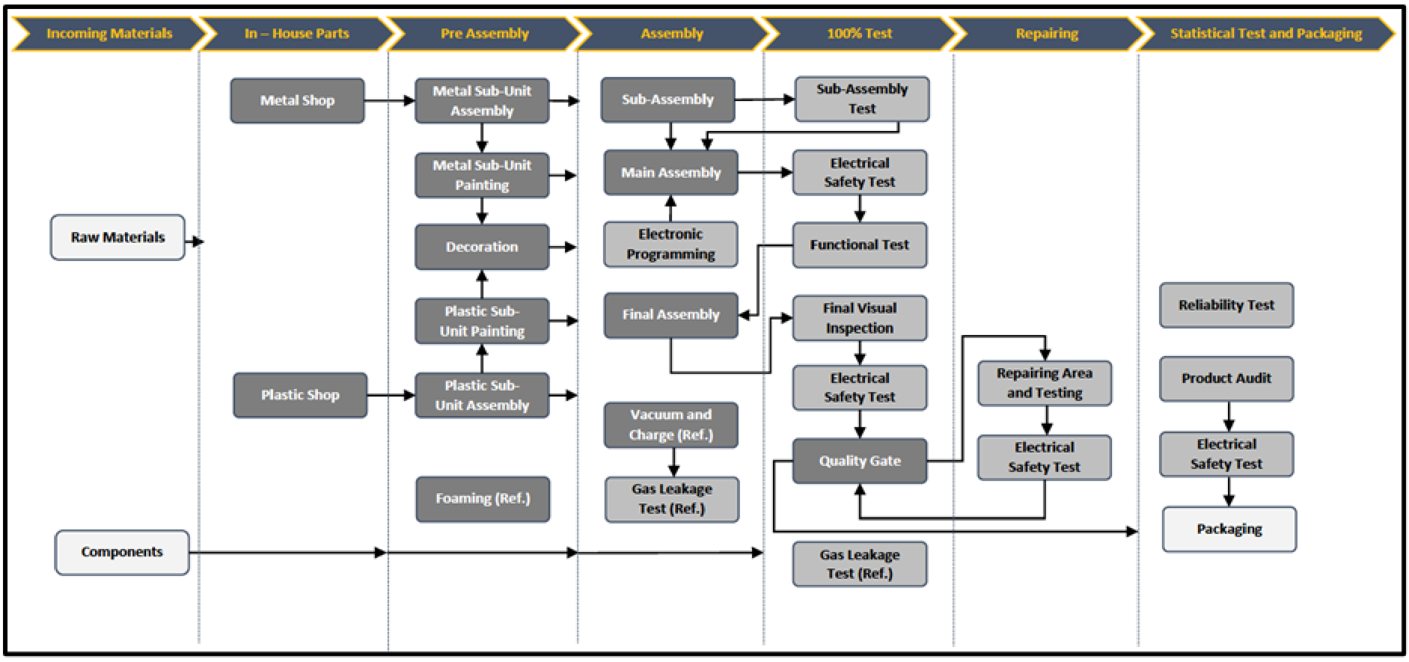

Figure 21 - Manufacturing process outline

To integrate all relevant elements that affect product and process quality, a RMPFQ-model is developed in Task T2.5 which focuses on "QU4LITY Digital Models and Vocabularies". This model is based on the previously developed functional models introduced in past projects (e.g.: GRACE FP7-NMP-2009-SMALL-3 [Leitão12]) and on enhancements designed and applied during QU4LITY lifetime. The previous MPFQ model [Foehr13] only covers the processes of assembly manufacturing covering Material, Process, Function and Quality. In QU4LITY project, pilots focusing on both machining and assembly processes are included. Therefore, the R-MPFQ model is developed by adding a Resource element. The definition of each element is listed as follows:

- Manufacturing Resource, according to ISO 15531 [Cutting07], represents the devices, tools and means, at the disposal of the enterprise to produce goods and services, but except raw material and final product components,
- Material represents everything that is needed to produce a certain product or product component, which may include raw materials, pre-products, consumables, operating supplies, product components and assemblies,
- Manufacturing Processes are defined as processing and transforming materials into the final goods by using machines, tools and human labour. This process is defined within the plant engineering,
- Product Functions / Features represent the distinguished characteristics of a product item, which may include functionalities like specific tasks, actions or processes that the product is able to perform; and/or other features like performance,
- Product Quality (Q) is defined as, according to DIN EN ISO 9000 [Johnson93], the degree of conformance of final product functions and features to designed requirements.

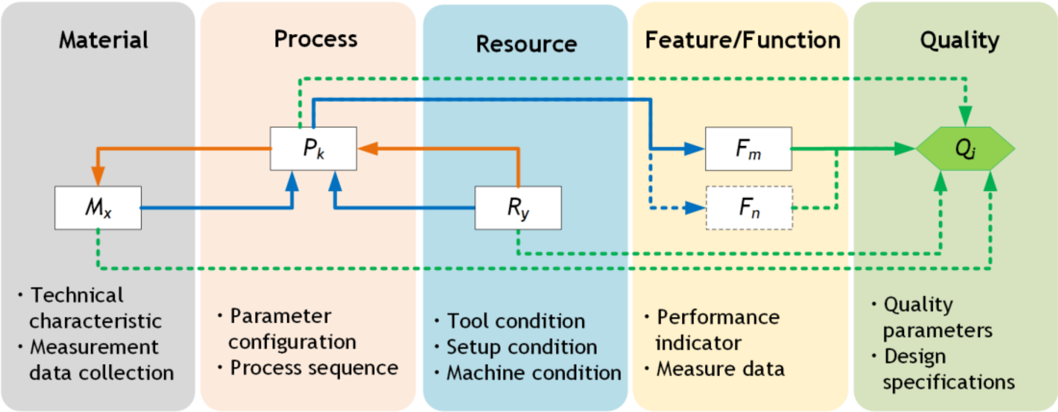

Figure 22 - R-MPFQ model elements and their interrelations

Figure 22 shows the elements of the proposed RMPFQ-model and their interrelations, as well as some data related to these elements. First, a given workpiece (M) is machined by machining resources (R), e.g. a given setup (fixturing and associated tooling) and a cutting tool (R), through a planned machining process (P), composing the RPM interactions (marked with orange lines). Second, the machining process (P) uses input material (M) and resources (R) to produce one or more features (F), composing the RPMF interactions (marked with blue lines). Moreover, all the RMPF-elements may also have straightforward impact on the quality (Q) of the machined workpiece (marked with green lines). There also exist relations among different resources, i.e. machine, setup, and cutting tool.

This model covers the most critical factors that affect the quality of a product or process. Each of the elements can be further decomposed into lower-level components in practical applications (Figure 23).

The R-MPFQ Quantitative modelling role, in this extent, is to define the correlation between the R-MPFQ parameters (resource, material, process, function) and quality KPI. The correlations are found dynamically through a data-drive approach (machine learning techniques), in this way it is possible to carry out quantitative analysis that enhances the decision support system.

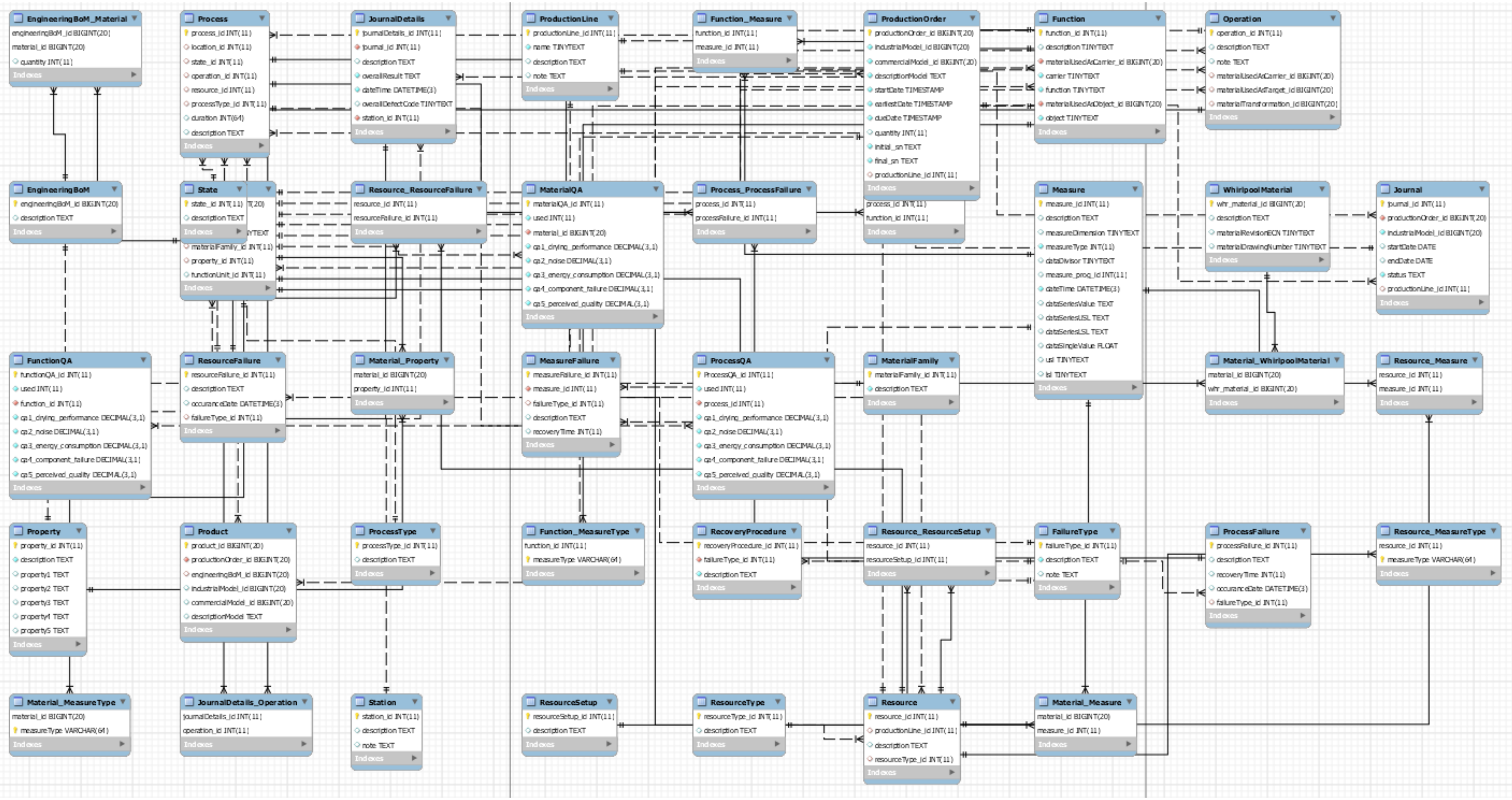

Figure 23 - R-MPFQ Relational Model

5.2.3 Q-Ontology Enabler Customization

Q-Ontology Enabler, being a set of python scripts, can be easily customized editing the source files. The most important thing to keep in mind, when customizing Q-Ontology Enabler scripts, is that it performs ETL processes, thus, only inputs have to be modified not to break R-MPFQ Ontology compliance.

Q-Ontology Enabler scripts follow the same structure:

Figure 24 - Generic Q-Ontology Enabler script structure

1. File opening
2. CSV rows parsing loop
3. Input Extraction and Transformation
4. Load processing
5. File closing

Customizing Q-Ontology Enabler scripts thus require only to properly change section 1 and 3 to fit .csv input file name and structure.

5.3 Access Instructions

QU4LITY Cloud Bridge and Q-Ontology Enabler are both offered as Open-Source software, licensed under the GNU Affero General Public License v3.0 [German09], at the following repositories on GitHub:

- GitHub - qu4lity/qu4lity-cloud-bridge

- GitHub - qu4lity/q-ontology-enabler

QU4LITY Cloud Bridge can be deployed effortlessly by taking advantage of the facilities offered by Docker as well as installed locally with the help of node.js.

Q-Ontology Enabler can be run using a Python 3 compiler.

5.3.1 REST API

QU4LITY Cloud Bridge offers a rich set of API, which can be consulted below:

Table 16 - QU4LITY Cloud Bridge APIs

HTTP Method	Service	Description
POST	/drum/sensor/fetch/one	Returns given drum sensor measure by passing measure_id as body parameter
POST	/drum/sensor/fetch/all	Returns all drum sensor measures. You can pass the followings as body parameter: • type: String - MANDATORY • from: Date • to: Date • limit: Int • offset: Int • decoded: Bool
POST	/drum/test/fetch/one	Returns given drum test measure by passing measure_id as body parameter
POST	/drum/test/fetch/all	Returns all drum test measures. You can pass the followings as body parameter: • type: String - MANDATORY • onlyFailures: Bool • from: Date • to: Date • limit: Int • offset: Int
GET	/engineeringBoM/list	Returns all engineeringBoMs

HTTP Method	Service	Description
POST	/engineeringBoM/fetch/one	Returns an engineeringBoM. You **must** pass the followings as body parameter: • engineeringBoM_id: Int - MANDATORY
POST	/engineeringBoM/fetch/all	Returns a subset of all engineeringBoMs exploding the nested inclusions. You can pass the followings as body parameter: • engineeringBoM_id: Int - MANDATORY
GET	/function/list	Returns all functions
POST	/function/list/by	Returns a subset of functions. You can pass the followings as body parameter: • function_id: Int - MANDATORY • function: String • materialUsedAsObject_id: Int • materialUsedAsCarrier_id: Int • limit: Int • offset: Int
POST	/function/fetch/one	Returns a function. You **must** pass the followings as body parameter: • function_id: Int - MANDATORY
POST	/function/fetch/all	Returns a subset of all functions exploding the nested inclusions. You can pass the followings as body parameter:

HTTP Method	Service	Description
		• function_id: Int - MANDATORY • function: String • materialUsedAsObject_id: Int • materialUsedAsCarrier_id: Int • measureType: String • from: Date • to: Date • limit: Int • offset: Int
POST	/function/fetch/measureType	Returns measure types for functions. You can pass the followings as body parameter: • function_id: Int - MANDATORY
GET	/functionQA/list	Returns all functionQAs
POST	/functionQA/fetch/one	Returns a functionQA. You **must** pass the followings as body parameter: • function_id: Int - MANDATORY
GET	/journal/list	Returns all journals
POST	/journal/fetch/one	Returns a journal. You **must** pass the followings as body parameter: • journal_id: Int - MANDATORY
POST	/journal/fetch/all	Returns a subset of all journals exploding the nested inclusions. You can pass the followings as body parameter:

HTTP Method	Service	Description
		• journal_id: Int - MANDATORY • productionOrder_id: Int
GET	/journalDetails/list	Returns all journalDetails
POST	/journalDetails/fetch/one	Returns a journalDetails. You **must** pass the followings as body parameter: • journalDetails_id: Int - MANDATORY
POST	/journalDetails/fetch/all	Returns a subset of all journalDetails exploding the nested inclusions. You can pass the followings as body parameter: • journalDetails_id: Int - MANDATORY • productionOrder_id: Int
GET	/material/list	Returns all materials
POST	/material/fetch/one	Returns a material. You **must** pass the followings as body parameter: • material_id: Int - MANDATORY
POST	/material/fetch/all	Returns a subset of all materials exploding the nested inclusions. You can pass the followings as body parameter: • material_id: Int - MANDATORY • measureType: String • from: Date • to: Date • limit: Int

HTTP Method	Service	Description
		• offset: Int
POST	/material/fetch/measureType	Returns measure types for materials. You can pass the followings as body parameter: • material_id: Int - MANDATORY
GET	/materialQA/list	Returns all materialQAs
POST	/materialQA/fetch/one	Returns a materialQA. You **must** pass the followings as body parameter: • material_id: Int - MANDATORY
GET	/operation/list	Returns all operations
POST	/operation/fetch/one	Returns an operation. You **must** pass the followings as body parameter: • operation_id: Int - MANDATORY
POST	/operation/fetch/all	Returns a subset of all operations exploding the nested inclusions. You can pass the followings as body parameter: • operation_id: Int - MANDATORY • materialUsedAsObject_id: Int • materialUsedAsTarget_id: Int • materialTransformation_id: Int
GET	/process/list	Returns all processes

HTTP Method	Service	Description
POST	/process/fetch/one	Returns a process. You **must** pass the followings as body parameter: • process_id: Int - MANDATORY
GET	/processQA/list	Returns all processQAs
POST	/processQA/fetch/one	Returns a processQA. You **must** pass the followings as body parameter: • process_id: Int - MANDATORY
POST	/process/fetch/all	Returns a subset of all processes exploding the nested inclusions. You can pass the followings as body parameter: • process_id: Int - MANDATORY
GET	/productionLine/list	Returns all productionLines
POST	/productionLine/fetch/one	Returns a productionLine. You **must** pass the followings as body parameter: • productionLine_id: Int - MANDATORY
GET	/productionOrder/list	Returns all productionOrders
POST	/productionOrder/fetch/one	Returns a productionOrder. You **must** pass the followings as body parameter: • productionOrder_id: Int - MANDATORY
GET	/resource/list	Returns all resources

HTTP Method	Service	Description
POST	/resource/fetch/one	Returns a resource. You **must** pass the followings as body parameter: • resource_id: Int - MANDATORY
POST	/resource/fetch/all	Returns a subset of all resources exploding the nested inclusions. You can pass the followings as body parameter: • resource_id: Int - MANDATORY • measureType: String • from: Date • to: Date • limit: Int • offset: Int
POST	/resource/fetch/measureType	Returns measure types for resources. You can pass the followings as body parameter: • resource_id: Int - MANDATORY
GET	/station/list	Returns all stations
POST	/station/fetch/one	Returns a station. You **must** pass the followings as body parameter: • station_id: Int - MANDATORY
GET	/whr_material/list	Returns all whr_materials
POST	/whr_material/fetch/one	Returns a whr_material. You **must** pass the followings as body parameter: • whr_material_id: Int - MANDATORY

5.3.2 Services Containerization

QU4LITY Cloud Bridge is offered as containerized solution under Docker at the aforementioned project repository, Q-Ontology Enabler, on the other hand, being just a set of Python ETL scripts, doesn't offer any containerization facilities.

The docker-compose.yaml file to instantiate QU4LITY Cloud Bridge as well as a MariaDB instance and a nginx web server is reported below:

```yaml
version: "3"

services:
    nginx:
        hostname: nginx
        image: nginx:latest
        networks:
          - hostnet
        ports:
          - "8080:80"
        depends_on:
          - qu4lity_cloud_bridge
        volumes:
          - ./nginx_conf/.htpasswd:/etc/nginx/.htpasswd
          - ./nginx_conf/nginx.conf:/etc/nginx/nginx.conf

    qu4lity_cloud_bridge:
        hostname: qu4lity_cloud_bridge
        image: node:14
        networks:
          - hostnet
        expose:
          - "9000"
        user: "node"
        working_dir: /home/node/app
        depends_on:
          - mpfq_mariadb
        environment:
          - NODE_ENV=production
          - PORT=9000
          - MPFQ_MARIADB_HOST=mpfq_mariadb
          - MPFQ_MARIADB_PORT=3306
          - MPFQ_MARIADB_DB=whr_mpfq_relational
          - MPFQ_MARIADB_USER=USER
          - MPFQ_MARIADB_PASSWORD=PASSWORD
        volumes:
          - ./node_conf/qu4lity-cloud-bridge:/home/node/app
        command: "npm start"

    mpfq_mariadb:
        hostname: mpfq_mariadb
        image: mariadb:latest
        networks:
          - hostnet
        ports:
          - "3306:3306"
        volumes:
          - container-volume:/var/lib/mysql
          - ./mariadb_conf:/docker-entrypoint-initdb.d/
        environment:
          - MYSQL_ROOT_PASSWORD=PASSWORD

volumes:
  container-volume:

networks:
   hostnet:
```

Figure 25 - QU4LITY Cloud Solutions docker-compose file

The extensive use of environments makes possible to reconfigure the whole setup with ease. Additionally, Docker secrets could be used to manage sensitive data in untrusted environments.

5.4 Quick Start Guide

5.4.1 QU4LITY Cloud Bridge

For a test deployment the following scripts and sample data are offered:

- README.md: a text file containing instructions on how to install/deploy the platform as well as a list of all the REST API offered by the infrastructure;
- docker-compose.yml: a YAML file containing the configurations of the various Docker containers (Docker images, environment variables, networking, data volumes configuration etc.) see Figure 25;
- nginx_conf: a folder containing a sample configuration for nginx
- mariadb_conf: folder containing all the .sql import scripts needed to prepopulate the databases with for demonstration purposes.

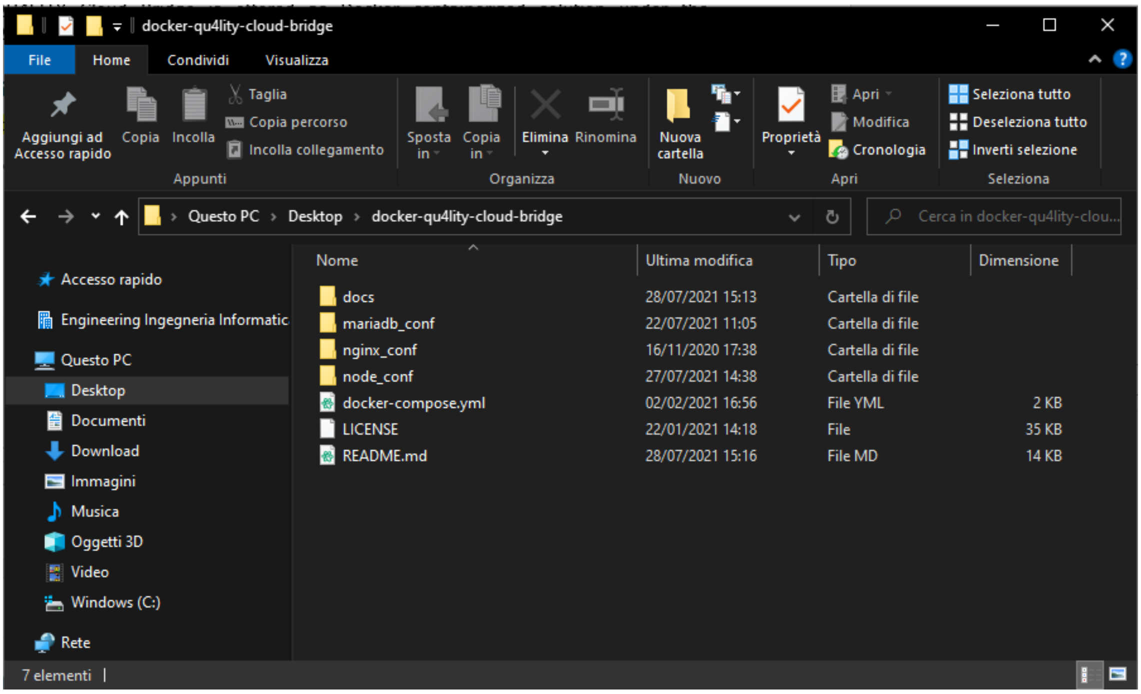

Figure 26 - QU4LITY Cloud Bridge root folder

To instantiate the whole test environment QU4LITY Cloud Bridge features a docker-compose file which will take care of the creation of 3 containers namely: nginx, MariaDB and node.js. Alternatively, you can locally run the infrastructure with the help of node.js and an existing MariaDB Server.

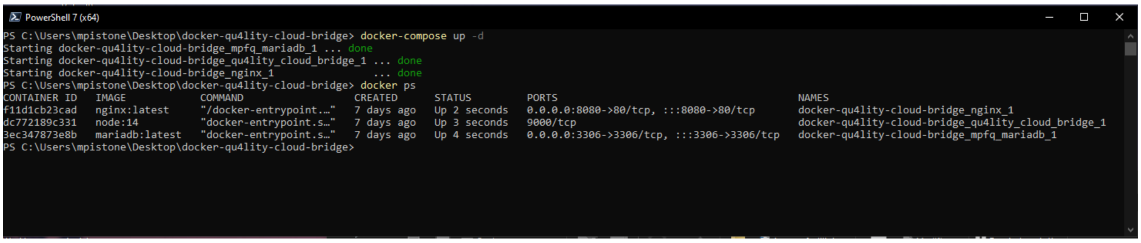

Figure 27 - QU4LITY Cloud Bridge running on Docker

5.4.2 Q-Ontology Enabler

The following scripts are offered to help you performing ETL operations with Q-Ontology Enabler:

- README.md: a text file containing instructions on how to customize ETL scripts to successfully export existing data toward R-MPFQ Ontology Relational Database

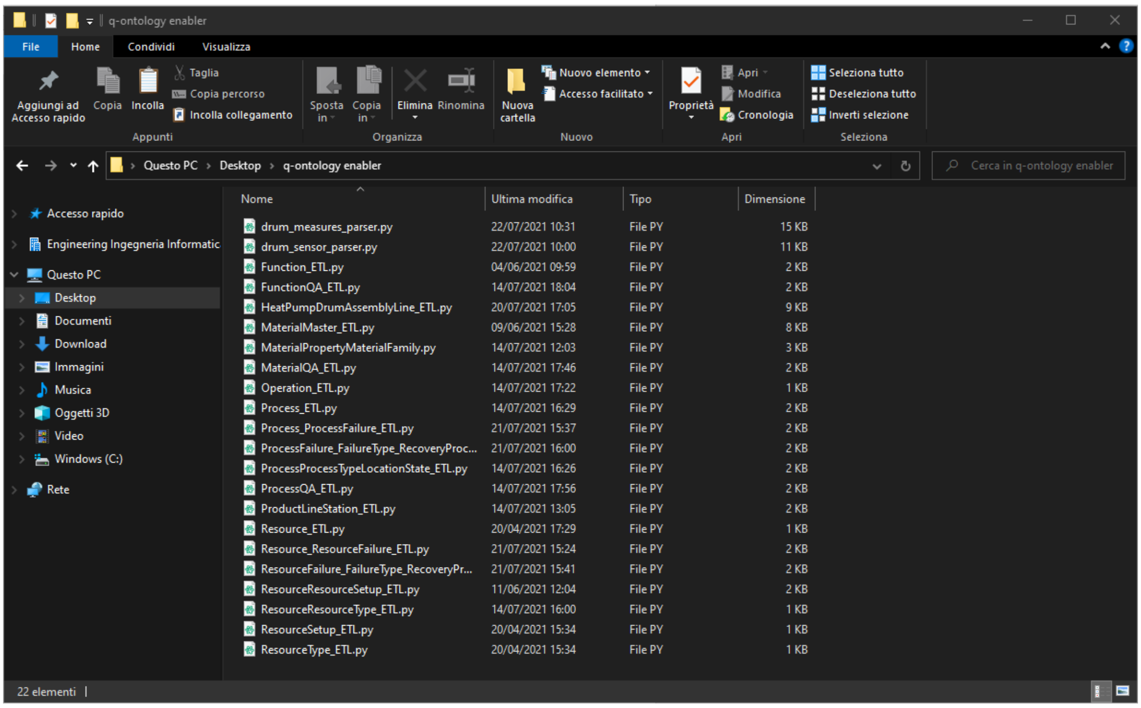

Figure 28 - Q-Ontology Enabler root folder

To start performing an ETL process, after having customized one of the existing scripts, you just need to run a Python3 Client and pass the ETL script as input. The output will be a .sql import script which you can run on your existing database instance.

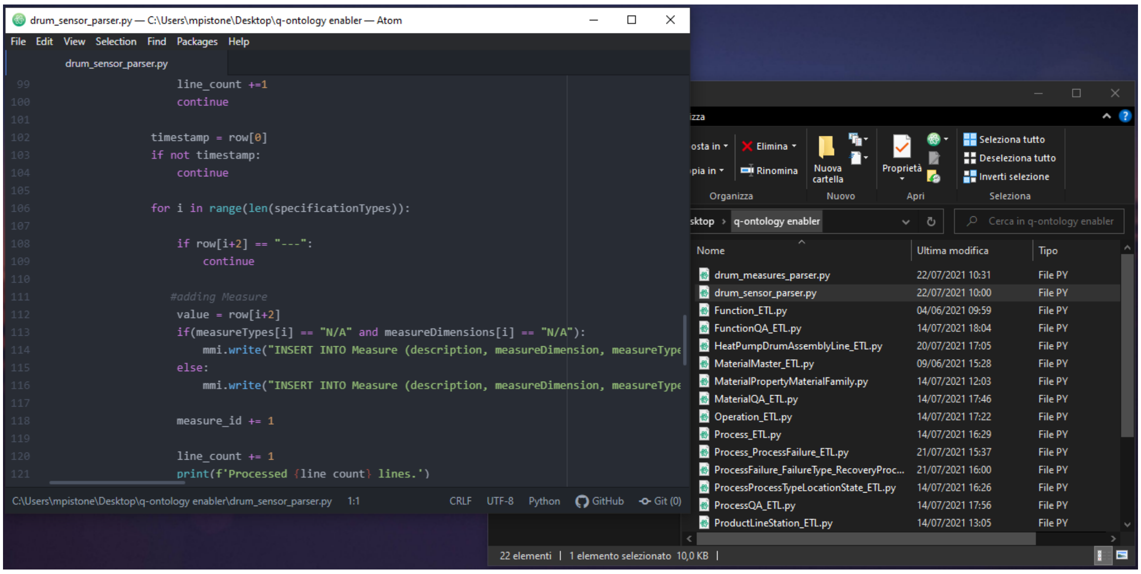

Figure 29 - Q-Ontology Enabler customization

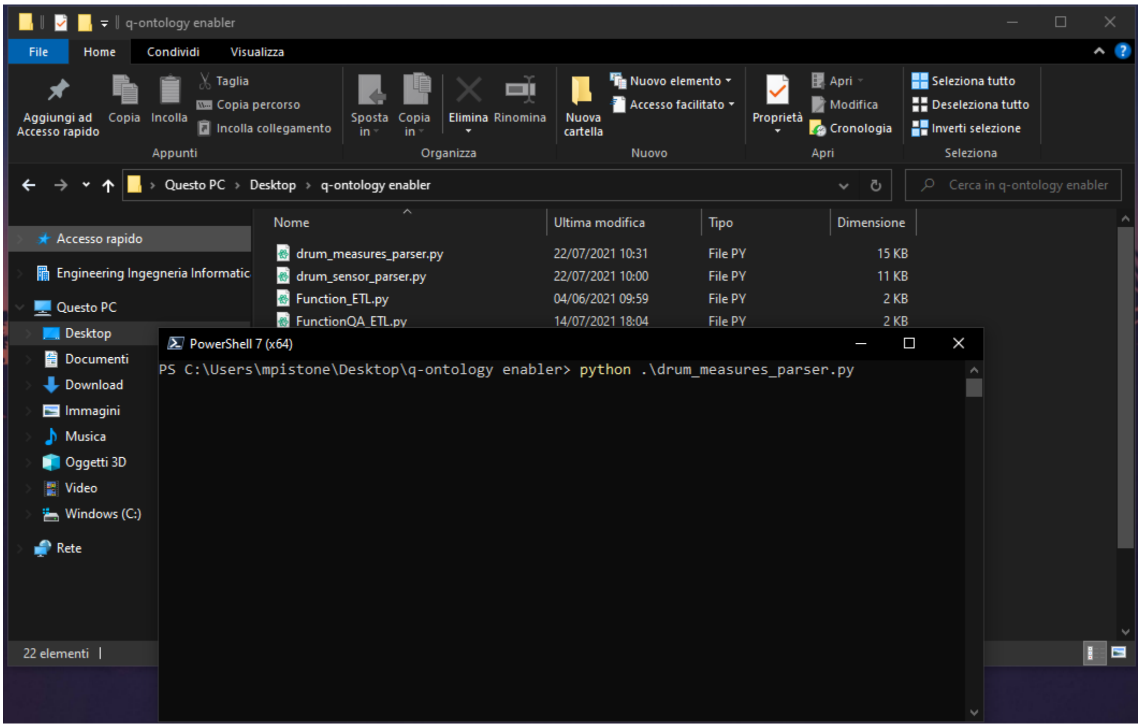

Figure 30 - Q-Ontolgy Enabler run

5.5 Content Available

Data is evaded from QU4LITY Cloud Bridge by the means of REST API in JSON format. Requests differ depending on the needs of pilot-specific visualization and computing tools, they can return full chain of linked data as well as single bits of information. Table 16 shows a detailed description of the API calls that can be performed trough REST interface. API responses returned from QU4LITY Cloud Bridge reflects data models defined on R-MPFQ Ontology model simplifying integrations activities extensively as shown in Figure 31 and Figure 32.

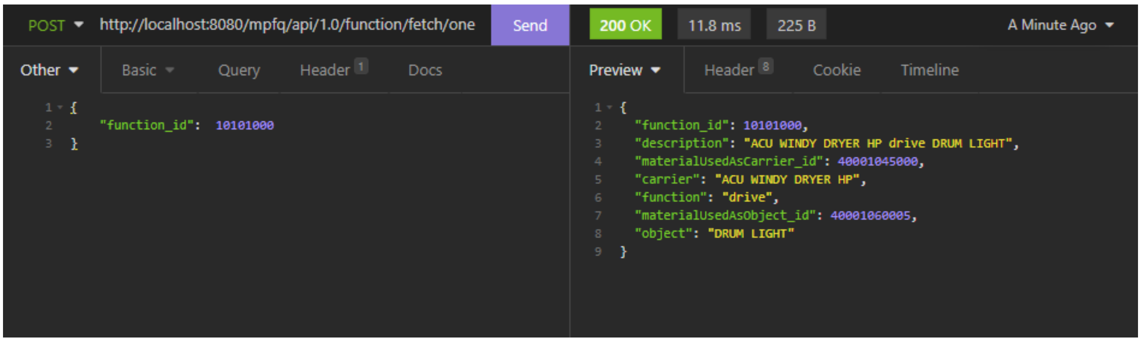

Figure 31 - Example of QU4LITY Cloud Bridge API Call response

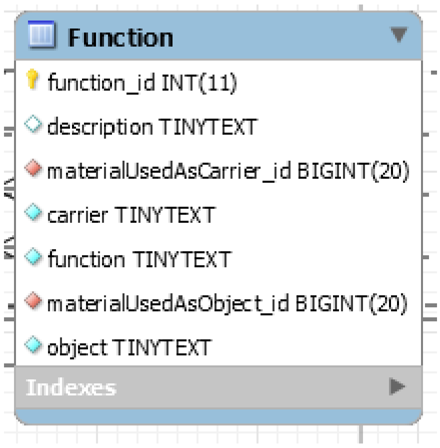

Figure 32 - Function Model in R-MPFQ Relational Model

5.6 Validation Campaign

Whirlpool is opening a green field plant in Lodz where a clothes dryer will be produced. The manufacturing process for building a clothes dryer includes many steps: combination of automatic equipment and manual operations and, throughout the production process, several quality stations are installed to perform measurements, detect defective parts, filter them, or repair them.

Quality data are currently managed as islands and with low correlation between the many data generated at each Quality gate. In particular, the following class of problems can be found in Whirlpool pilot:

- **Lack of a common and a holistic semantic model** able to represent concepts at different stages of product lifecycle
- **Lack of standard methods** and tool to **gather**, **store** and **share data**
- **Lack of** flexible and user-friendly **analytical tools**
- **Lack of** a comprehensive **way to share** results or **data analysis** and **link** them **to business priorities**.
- The full potential of **data generated** at each **gate** is **not exploited yet** and that any attempt of using the data is currently requiring a strong specialization and specific knowledge of each gate plus a **time-consuming effort to query database** and **manually correlate and analyze data**.

The pilot project will leverage the results of a previous research project (GRACE FP7-NMP-2009-SMALL-3 [Leitão12]) and integrate the QU4LITY enablers and digital platforms (through APIs) and AQ control circuits. The main innovation will be represented by the introduction into production of the MPFQ model fused with the AQ control circuits: Functional Integration and Correlation between Material, Quality, Process and Appliance Functions. QU4LITY digital enablers will augment human and machine decision product and processes, correlating consumer needs and perceived Quality with production parameters, thus, improving the speed of decisions and their effectiveness. This will impact both the Quality related metrics, reducing defective products on the market, and the Productivity, metrics letting operator spend less time in expensive analysis and moving as much as possible decision steps from them to machines.

5.7 Future Works

QU4LITY Cloud Solutions although being fully developed and ready to be adopted in production environments, could be further enhanced taking into account fine-tuning and functionalities requests from pilots who adopt them.

One major enhancement, that could take place, would consist in the integration of major Cloud Storage Platforms into the lifecycle of QU4LITY Cloud Bridge, thus, offering the possibility to store data on AWS, Azure Cloud Services or Google Drive instead of relying on a private relational database system. Therefore, enhancing interoperability aspects of QU4LITY Cloud Bridge, it may become more attractive to the end-users.

6 Conclusions

This deliverable outlines the results of the tasks T3.2 providing an in-depth analysis and overview of the functionalities and the capabilities offered by QU4LITY High Performance Computing Infrastructure and QU4LITY Cloud Solutions.

The deliverable outlines the functionalities of the prototype implementations of the presented digital enablers. Such enablers have been used in at least one QU4LITY pilots in WP7 proving their efficiency and goodness.

QU4LITY HPC Cloud Infrastructure delivers an innovative approach to multimodal network training on a cluster, combining two different types of inputs: images and parameters. The concatenated layer combines features extracted from the images as well as features extracted from the parameters and is followed by several fully connected layers. Moreover, we have been exploring the possibilities for an increased automation of the data processing to generate the appropriate inputs and outputs for the network. Note, however, that a complete automation is not possible because the data is task dependent.

QU4LITY Cloud Solutions delivers a powerful set of tools to enable semantic interoperability between legacy data and newly engineered QU4LITY R-MPFQ Ontology, and data exchange toward QU4LITY Cloud Data Storage by the means of an easy-to-use REST Interface. We are also considering some improvements to the interoperability aspects of QU4LITY Cloud Bridge, by integrating major Cloud Storage Platforms into the lifecycle of QU4LITY Cloud Bridge; thus, offering the possibility to store data on AWS, Azure Cloud Services or Google Cloud instead of relying on a private relational database system. Further improvements to the Q-Ontology Enabler will be considered case by case as it strictly depends on pilot-specific legacy data.

QU4LITY Cloud Solutions have also been added to the QU4LITY Market Platform in WP8, i.e. offering a whole new way to interact with legacy and smart data to the ZDM community. To this extent, a series of dissemination activities (e.g., webinars and digital talks) have been organized and solutions have been proposed as entry point for other research projects and initiatives (DFA Zero-X Manufacturing), offering feedbacks and ideas for future development and fine-tuning of aforementioned assets.

QU4LITY HPC Infrastructure, on the other hand, is distributed by JSI which has an in-house Centre for Technology Transfer and Innovation that takes care of relations with industrial partners and dissemination of results. In any case, JSI is considering the possibility to join DFA initiative (WP9).

Overall, the present deliverable has presented a pool of lightweight and responsive solutions for digital quality management, notably assets that fall in the realm of QU4LITY Cloud Solutions and a strong, resilient, efficient and secure infrastructure for training deep neural network models for ZDM on a High-Performance Computing Cluster.

References

[Brain15] Brain, M., Tinelli, C., Rümmer, P., & Wahl, T. (2015, June). An automatable formal semantics for IEEE-754 floating-point arithmetic. In 2015 IEEE 22nd Symposium on Computer Arithmetic (pp. 160-167). IEEE.

[Cutting07] Cutting-Decelle, A. F., Young, R. I., Michel, J. J., Grangel, R., Le Cardinal, J., & Bourey, J. P. (2007). ISO 15531 MANDATE: a product-process-resource based approach for managing modularity in production management. Concurrent Engineering, 15(2), 217-235.

[Fan21] Fan, Y., Lan, Z., Childers, T., Rich, P., Allcock, W., & Papka, M. E. (2021). Deep Reinforcement Agent for Scheduling in HPC. *arXiv preprint arXiv:2102.06243*.

[Foehr13] Foehr, M., Jäger, T., Turrin, C., Petrali, P., & Pagani, A. (2013, February). Methodology for consideration of product quality within factory automation engineering. In 2013 IEEE international conference on industrial technology (ICIT) (pp. 1333-1338). IEEE..

[German09] German, D. M., & González-Barahona, J. M. (2009, June). An empirical study of the reuse of software licensed under the GNU General Public License. In IFIP International Conference on Open Source Systems (pp. 185-198). Springer, Berlin, Heidelberg.

[Johnson93] Johnson, P. L. (1993). ISO 9000: Meeting the new international standards. New York: McGraw-Hill..

[Leitão12] Leitão, P., Rodrigues, N., Turrin, C., Pagani, A., & Petrali, P. (2012, October). GRACE ontology integrating process and quality control. In IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society (pp. 4348-4353). IEEE.

[Psarommatis20] Psarommatis, F., May, G., Dreyfus, P. A., & Kiritsis, D. (2020). Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. International Journal of Production Research, 58(1), 1-17.

[Ridge20] Ridge, B., Gams, A., Morimoto, J., & Ude, A. (2020). Training of deep neural networks for the generation of dynamic movement primitives. Neural Networks, 127, 121-131.

[VanEssen15] Van Essen, B., Kim, H., Pearce, R., Boakye, K., & Chen, B. (2015, November). LBANN: Livermore big artificial neural network HPC toolkit. In *Proceedings of the workshop on machine learning in high-performance computing environments* (pp. 1-6).

List of figures

Figure 1 - D3.4 document structure 9
Figure 2 - QU4LITY Q-RA 10
Figure 3 - HPC service schematic 27
Figure 4 - Docker file that contains all the instructions for JSI NSC HPC access 30
Figure 5 - Source folder with the data. Note the image.zip file and NN_python_data.mat file 31
Figure 6 - Content of images.zip archive 31
Figure 7 - .xrsl file for setting up the JSI HPC task 32
Figure 8 - Communication opening process with arcproxy 32
Figure 9 - Job submission 32
Figure 10 - Health check using arcstat 33
Figure 11 - Job finished 33
Figure 12 - Results fetch using arcget 33
Figure 13 - Results are stored in the marked folder, named after the job ID. Other folders contain supplementary files uploaded to the HPC 33
Figure 14 - Trained model is stored in output.zip. A log of all the commands executed on the HPC is provided as well 34
Figure 15 - GUI for JSI HPC access that provides the same command-line functionality as described above 34
Figure 16 - Snippet of production parameters from Kolektor, as shown in MS Excel 35
Figure 17 - Extract of normalized input data, with 26 input parameters and 7864 input samples, in Matlab 36
Figure 18 - Image of the product from the Kolektor Pilot with the error of the material being porous at the top 37
Figure 19 - Proposed structure of the neural network. Once trained, this constitutes the neural network ZDM prediction model 38
Figure 20 - QU4LITY Cloud Solutions Architecture 40
Figure 21 - Manufacturing process outline 42
Figure 22 - R-MPFQ model elements and their interrelations 43
Figure 23 - R-MPFQ Relational Model 44
Figure 24 - Generic Q-Ontology Enabler script structure 45
Figure 25 - QU4LITY Cloud Solutions docker-compose file 54
Figure 26 - QU4LITY Cloud Bridge root folder 55
Figure 27 - QU4LITY Cloud Bridge running on Docker 56
Figure 28 - Q-Ontology Enabler root folder 56
Figure 29 - Q-Ontology Enabler customization 57
Figure 30 - Q-Ontolgy Enabler run 57
Figure 31 - Example of QU4LITY Cloud Bridge API Call response 58
Figure 32 - Function Model in R-MPFQ Relational Model 58

List of tables

Table 1 - Tech-Survey Questionnaire Template 11
Table 2 - PHILIPS Tech-Survey Questionnaire results 12
Table 3 - SIEMENS Tech-Survey Questionnaire results 12
Table 4 - CONTI Tech-Survey Questionnaire results 14
Table 5 - WHR Tech-Survey Questionnaire results 15
Table 6 - MON Tech-Survey Questionnaire results 16
Table 7 - KOL Tech-Survey Questionnaire results 17
Table 8 - THYS Tech-Survey Questionnaire results 18
Table 9 - AIRBUS Tech-Survey Questionnaire results 19
Table 10 - GHI Tech-Survey Questionnaire results 19
Table 11 - RiaStone Tech-Survey Questionnaire results 21
Table 12 - PRIMA Tech-Survey Questionnaire results 21
Table 13 - Danobat Tech-Survey Questionnaire results 22
Table 14 - FAGOR Tech-Survey Questionnaire results 23
Table 15 - GF Tech-Survey Questionnaire results 25
Table 16 - QU4LITY Cloud Bridge APIs 46

List of Abbreviations

Abbreviation	Explanation
AQ	Autonomous Quality
ARC	Advanced Resource Connector
AWS	Amazon Web Services
CNN	Convolutional Neural Network
CSP	Cloud Service Provider
CPU	Central Processing Unit
DMP	Digital Manufacturing Platform
DNN	Deep Neural Network
GCP	Google Cloud Platform
HPA	Horizontal Pod Autoscaler
HPC	High Performance Computing
IaaS	Infrastructure as a Service
IT	Information Technology
JVC	New Joint Cluster
MES	Manufacturing Execution System
PaaS	Platform-as-a-Service
QoS	Quality of Service
Q-RA	QU4LITY Reference Architecture
SLA	Service Level Agreement
SOTA	State Of The Art
SPT	Security Privacy and Trust
ToC	Table of Content
WP	Work-package
ZDM	Zero Defect Manufacturing

Partners: