

Quantifarm

D3.1 Data Requirements and Interoperability

quantifarm.eu



D3.1 Data Requirements and Interoperability – initial version

Acronym	QuantiFarm
Full title	Assessing the impact of digital technology solutions in agriculture in real-life conditions
G.A. No.	101059700
Topic	HORIZON-CL6-2021-GOVERNANCE-01
Type of action	Research and Innovation Actions
Start date of project	1st July 2022
Duration	42 months
Author(s)/Organisation(s)	Jack Verhoosel/TNO, Nikos Kalatzis/NP, Han Kruiger/TNO, Christopher Brewster/TNO
Contributor(s)	All partners
Work Package	WP3 – Digital Tools
Delivery Date (DoA)	30/6/2023 (M6)
Actual Delivery Date	14/07/2023
Dissemination Level	PU
Abstract:	This deliverable describes the background, requirements and design of the QuantiFarm Common Semantic Model and the QuantiFarm Data Platform which are used to collect DAT generic and usage information to be fed into the various tools of the QuantiFarm Toolkit.

Document Revision History			
Date	Version	Author/Contributor/ Reviewer	Summary of main changes
1/03/2023	0.1	Nikos Kalatzis (NP)	Table of Contents
01/05/2023	0.2	Nikos Kalatzis, Vasilis Gkoles, Thanasis Manos (NP)	Specification of use cases and tech requirements
15/05/2023	0.3	George Papadopoulos, Alex Fournaropoulos (AUA)	Integration with SmartAKIS and FAIRChain
15/06/2023	0.4	Jack Verhoosel, Han Kruiger, Christopher Brewster (TNO)	Specification of QCSM and data platform
30/06/2023	0.9	Christopher Brewster (TNO), Nikos Kalatzis (NP)	Addressing comments from internal review.
14/7/2023	1.0	Christopher Brewster (TNO)	Final version addressing all comments from peer-reviewers



D3.1 Data Requirements and Interoperability – initial version

QuantiFarm Consortium			
Participant Number	Participant organisation name	Short name	Country
1	GAIA EPICHEIREIN ANONYMI ETAIREIA PSIFIAKON YPIRESION (GAIA)	GAIA	GR
2	NEDERLANDSE ORGANISATIE VOOR TOEGEPAST NATUURWETENSCHAPPELIJK ONDERZOEK TNO	TNO	NL
3	POLITECNICO DI MILANO	POLIMI	IT
4	NEUROPUBLIC AE PLIROFORIKIS & EPIKOINONION	NP	GR
5	CONSULAI, CONSULTORIA AGROINDUSTRIAL LDA	CONSULAI	PT
6	CONFEDERAZIONE GENERALE DELL AGRICOLTURA ITALIANA	CONFAGRICOLTURA	IT
7	FOODSCALE HUB GREECE ASSOCIATION FOR ENTREPREUNERSHIP AND INNOVATION ASTIKI MI KERDOSKOPIKI ETAIREIA	FSH	GR
8	PETERSON PROJECTS BV	PETERSON	NL
9	LUONNONVARAKESKUS	LUKE	FI
10	GEOPONIKO PANEPISTIMION ATHINON	AUA	GR
11	OKYS LTD	OKYS	BG
12	COMITE DES ORGANISATIONS PROFESSIONNELLES AGRICOLE DE L UNION EUROPEENNE COPA ASSOCIATION DE FAIT	COPACOGECA	BE
13	COMITE EUROPEEN DES GROUPEMENTS DE CONSTRUCTEURS DU MACHINISME AGRICOLE	CEMA	BE
14	. TEAGASC - AGRICULTURE AND FOOD DEVELOPMENT AUTHORITY	TEAGASC	IE
15	INSTITUTO TECNOLOGICO AGRARIO DE CASTILLA Y LEON	ITACyL	ES
16	HORTA SRL	HORTA	IT
17	KATHOLIEKE UNIVERSITEIT LEUVEN	KUL	BE
18	DELPHY BV	DELPHY	NL
19	INSTITUT DE L'ELEVAGE	IDELE	FR
20	AUGMENTA AGRICULTURE TECHNOLOGIES MONOPROSOPI IDIOTIKI	AUGMENTA	GR
21	ASOCIATIA NATIONALA A INDUSTRIILORDE MORARIT SI PANIFICATIE DIN ROMANIA	ANAMOB	RO
22	UAB ART21	ART21	LT
23	AGROSMART SIA	AgroSmart	LV
24	BENCO BALTIC DOO ZA SAVJETOVANJE IUSLUGE	BENCO	HR
25	FARM FRITES POLAND DWA SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA	FFP2	PL
26	AGROMAIS PLUS COMERCIO E SERVICOSAGRICOLAS S.A.	AGROMAIS	PT
27	KMETIJSKO GOZDARSKA ZBORNICA SLOVENIJE KMETIJSKO GOZDARSKI ZAVOD MURSKA SOBOT	KGZS	SI
28	TERRA LITTERA DOO	Terra	RS
29	ANYSOLUTION SL	AnySol	ES
30	A.M. FILAGROTIKI SYMVOULEFTIKI LTD	Filagro	CY
31	AGRIDEA SCHWEIZERISCHE VEREINIGUNG	AGRIDEA	CH
32	FLOX limited	FLOX	UK



D3.1 Data Requirements and Interoperability – initial version

LEGAL NOTICE

Funded by the European Union under Grant Agreement No 101059700. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



**Funded by
the European Union**

© QuantiFarm Consortium, 2022.

Reproduction is authorised provided the source is acknowledged.



Table of Contents

1. Introduction	9
1.1. Project Summary.....	9
1.2. Document Scope	9
1.3. Document Structure	9
2. State of the art analysis on relevant tech sectors	11
2.1 Background.....	11
2.2 Interoperability and data standards	12
2.3 Architectures for Data management/sharing/ Querying.....	12
2.4 Interoperability and legacy systems	13
3. Technical requirements for the QuantiFarm data platform	14
3.1 Data provision by Test Cases and other platforms.....	14
3.1.1 Test cases	14
3.1.2 External inventories of DATs	16
3.1.3 Open Datasets	18
3.2 Assessment Framework	19
3.3 Data needs for the QuantiFarm Tools	20
3.4 Technical Requirements specification	24
4. QuantiFarm Common Semantic Model	25
4.1 Design of the QCSM.....	25
4.1.1 Guiding principles	25
4.1.2 Reuse and extension of existing ontologies	25
4.2 Modelling digital agricultural technologies	27
4.3 Modelling sustainability parameters	28
4.3.1 Environmental sustainability parameters	29
4.3.2 Economic sustainability parameters	30
4.3.3 Social sustainability parameters	32
4.4 Modelling behavioural intentions	34
5. QuantiFarm Data platform design	36
5.1 Use cases for QuantiFarm Data platform.....	36
5.2 QuantiFarm Data platform functional architecture	37
5.3 Implementation	40
6 Conclusions	41
References	42
Annex A	45
Annex B	91



D3.1 Data Requirements and Interoperability – initial version

Annex C	96
Annex D	97
Annex E	99

List of tables

<i>Table 1 Test Cases summarization</i>	15
<i>Table 2 Generic test case properties</i>	16
<i>Table 3 QuantiFarm 's sustainability domains, categories and sub-categories</i>	20

List of figures

<i>Figure 1 Semantic standards and their relationships</i>	11
<i>Figure 2 High level data flow from sources via assessment framework to toolkit.</i>	14
<i>Figure 3 The FAIRshare inventory of DATS</i>	17
<i>Figure 4 The Smart-AKIS inventory of Smart Farming technologies</i>	18
<i>Figure 5 QCSM model with DAT concept and properties.</i>	28
<i>Figure 6 QCSM operations and the use of water, fertiliser and pesticides.</i>	30
<i>Figure 7 QCSM fertiliser and pesticide operations.</i>	32
<i>Figure 8 QCSM part of the FarmWorker and possible injuries.</i>	34
<i>Figure 9 Infographic of the behavioural intentions factors.</i>	35
<i>Figure 10 Use case diagram for QuantiFarm data platform</i>	36
<i>Figure 11 Functional architecture of QuantiFarm data platform.</i>	38
<i>Figure 12 Screenshot of QuantiFarm 's gitlab project.</i>	40



D3.1 Data Requirements and Interoperability – initial version

List of Abbreviations and Acronyms	
API	Application Programming Interface
AKIS	Agriculture Knowledge and Innovation Systems
CAP	Common Agricultural Policy
DATs	Digital Agriculture Technologies
DB	Database
EC	European Commission
EO	Earth Observation
EUPL	European Union Public Licence
FMIS	Farm Management Information System
GDPR	General Data Protection Regulation
ICT	Information and Communications Technology
JRC	Joint Research Centre
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
OWL	Web Ontology Language
TC	Test Case
VRA	Variable Rate Application
WP	Work Package
W3C	World Wide Web Consortium



Executive Summary

The QuantiFarm project focuses on advancing DATs to bolster the agricultural sector's sustainability (economic, environmental, and social) and competitiveness. A key component of this effort is the development of a comprehensive assessment framework for the independent evaluation of DATs' multifaceted costs and benefits. The project encompasses a wide range of stakeholders, including farmers, advisors, researchers, technology providers, and policy makers, working together across 20 European countries to ensure replicability and uptake of digital technologies in agriculture.

Among the key outcomes of the QuantiFarm project is the development of a web based QuantiFarm Toolkit, for Farmers, Advisors and Policy Makers. It will provide a user-friendly interface to the descriptions and assessments of the selected DATs (within and outside QuantiFarm) and tools will allow farmers and their advisors to make decisions. The overall functionality including the Advanced Decision Support Tools of the QuantiFarm Toolkit will be based on the theoretical model of the Cost Benefit Analysis and Sustainability Assessment.

A necessary step prior to the implementation of the toolkit is to analyse the underlying data needs and to build mechanisms for connecting with data repositories, standardize data formats and add appropriate meta data. This deliverable provides the design principles, technical requirements, and technical specifications of an integrated data management framework. To this end, the QuantiFarm Common Semantic Model (QCSM) is introduced which maximally reuses and combines existing standardised domain specific ontologies aiming to act as a common language for data derived from QuantiFarm Test Cases. The QCSM's design allows the integration of additional concepts supporting its ongoing adaptation to the changing needs of test cases, assessment framework and behavioural framework. In addition, the QuantiFarm Data Platform is introduced as is an Information System that exploits QCSM and provides all the necessary data collection, harmonisation, and management mechanisms in support of the QuantiFarm Toolkit.



1. Introduction

1.1. Project Summary

The QuantiFarm project focuses on supporting the further development of Digital Agriculture Technologies (DATs) in order to improve the sustainability (economic, environmental and social) and competitiveness of the EU's agricultural sector. To this end, QuantiFarm develops a comprehensive Assessment Framework for independent qualitative and quantitative assessments of the costs and benefits of digital agriculture technologies. The project will support the uptake of digital technologies by providing innovative tools and services, which will provide practical recommendations of relevance and practical utility to farmers, advisors, and policy makers across Europe. QuantiFarm is building its assessment and recommendation tools using data derived from 30 Test Cases (TCs) which span over 20 countries in 10 Biogeographical regions across Europe, thereby capturing multiple social, environmental and economic parameters. More than 100 farms of different types, sizes, ownership and operating conditions, have committed to participate in the project, both directly and through cooperatives and large umbrella organisations. In addition, the QuantiFarm Digital Innovation Academy will provide capacity building capabilities for advisors and other Agriculture Knowledge and Innovation Systems (AKIS) actors, by providing data on the various types of digital technologies available, their costs, benefits and impact on sustainability. QuantiFarm involves 32 partners, representing a variety of stakeholders, including 8 scientific organisations and 12 farmer representatives and consultants.

1.2. Document Scope

This document is part of the set of deliverables of WP3 of the QuantiFarm project and is the result of T3.1 on Data Requirements and Interoperability. This task has identified the relevant data sources for the assessment of DATs and worked on a data platform that allows to clean, integrate, link, harmonise, store and make available data coming from various heterogeneous data sources. Data formats have been standardised with the addition of suitable metadata, using semantic interoperability mappers and reusing and extending existing relevant ontologies.

The goal of this document is to describe the results of T3.1 activities, specifically focusing on the first versions of QuantiFarm Common Semantic Model (QCSM) and the QuantiFarm Data Platform that have been developed. Both the model and the platform are defined to collect data from various sources, including the 30 test cases of the project and other open data sources that are important for supporting farmers, advisors and policy makers about the use of Digital Agricultural Technologies. The Data Platform will provide collected data from all the sources in QuantiFarm Common Semantic Model format to the set of QuantiFarm tools to be developed in T3.2 and T3.3 to be provided to farmers, advisors and policy makers.

1.3. Document Structure

This document is comprised of the following chapters:

Chapter 1 provides a summary of the project, the document's scope and its overall structure.

Chapter 2 introduces the current landscape and background on agricultural interoperability standards and ontologies and existing data reference architectures to be reused.

Chapter 3 discusses the goal of the QuantiFarm data platform and the requirements from various sources on this platform and the common semantic model that is to be used by the data platform.



D3.1 Data Requirements and Interoperability – initial version

Chapter 4 presents the QuantiFarm Common Semantic Model (QCSM), the guidelines that have been used to define it, the existing ontologies that are being reused and examples of how DAT, sustainability and behavioural parameters are being modelled.

Chapter 5 describes the high-level design of the QuantiFarm data platform and its components based on use case definitions it needs to satisfy.

Chapter 6 gives a brief overview of the main ingredients of the document and indicates next steps to be taken.



2. State of the art analysis on relevant tech sectors

This chapter briefly describes the basic semantic technologies that are being used in the QuantiFarm Common Semantic Model and the QuantiFarm Data Platform. In addition, a brief summary is given of the existing agrifood interoperability standards and ontologies. Finally, a short overview is presented of existing data management architectures.

2.1 Background

In the last decades various methodologies for describing information in a semantically unambiguous way have been defined. The World Wide Web consortium (W3C) has been very active in this field, which has led to a number of standard mechanisms to define semantics, i.e., the Resource Description Framework (RDF), RDF Schema (RDFS) and the Web Ontology Language (OWL). See *Figure 1* for the various semantic standards and how they relate.

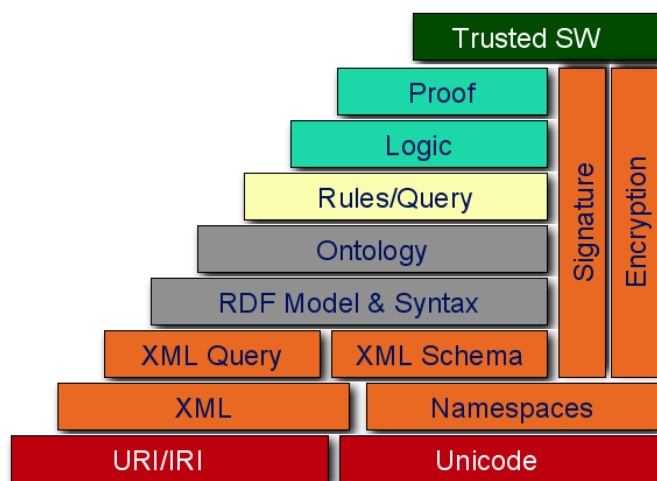


Figure 1 Semantic standards and their relationships

Figure 1: semantic standards and their relationships.

The Resource Description Framework (RDF) is a [World Wide Web Consortium](#) (W3C) standard originally designed as a data model for metadata. It has come to be used as a general method for description and exchange of graph data. RDF provides a variety of syntax notations and data serialisation formats, with [Turtle \(Terse RDF Triple Language\)](#) currently being the most widely used notation. RDF is a directed graph composed of triple statements. An RDF graph statement is represented by: 1) a node for the subject, 2) an arc that goes from a subject to an object for the predicate, and 3) a node for the object. Each of the three parts of the statement can be identified by a URI. An object can also be a literal value. This simple, flexible data model has a lot of expressive power to represent complex situations and relationships, while also being appropriately abstract. RDF was adopted as a W3C recommendation in 1999. The RDF 1.0 specification was published in 2004, the RDF 1.1 specification in 2014. SPARQL is a standard query language for RDF graphs. RDFS, OWL and SHACL are ontology languages that are used to describe RDF data.

The RDF Schema (Resource Description Framework Schema, abbreviated as RDFS) is a set of classes with certain properties using the [RDF](#) extensible [knowledge representation](#) data model, providing basic elements for the description of [ontologies](#). It uses various forms of RDF vocabularies, intended to structure RDF [resources](#). RDF and RDFS can be saved in a [triplestore](#), then one can extract some



D3.1 Data Requirements and Interoperability – initial version

knowledge from them using a query language, like [SPARQL](#). The first version [1,2] was published by the World-Wide Web Consortium (W3C) in April 1998, and the final [W3C recommendation](#) was released in February 2014 [3]. Many RDFS components are included in the more expressive [Web Ontology Language](#) (OWL).

The Web Ontology Language (OWL) is a family of [knowledge representation](#) languages for authoring [ontologies](#). Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains: the nouns representing classes of objects and the verbs representing relations between the objects. The OWL languages are characterised by [formal semantics and a first version was published in 2004](#). They are built using the [Resource Description Framework](#) (RDF). W3C announced the version 1.1 of OWL on 27 October 2009 [5]. This version, called OWL2, soon found its way into semantic editors such as [Protégé](#) and [semantic reasoners](#) such as Pellet, RacerPro, FaCT++ and HermiT. The OWL family contains many species, serialisations, syntaxes and specifications with similar names. OWL and OWL2 are used to refer to the 2004 and 2009 specifications, respectively.

2.2 Interoperability and data standards

There is a considerable body of work building ontologies for the food and agriculture domain which has gone hand in hand with the development of Linked Data (and “Linked Open Data”) in the agri-food domain. The major effort here has been AGROVOC developed by the FAO and maintained by a network of institutes around the world [6]. It is nowadays the most comprehensive multilingual thesaurus and vocabulary for agriculture. AGROVOC has now been partially mapped onto the US National Agricultural Library of the USDA and the CABI thesaurus in the form of the GACS ontology which has mapped and integrated the top 15,000 concepts [7]. Other recent work in this area has also focused on developing ontologies for sharing of research data including the Crop ontology initiative, the Agronomy Ontology (AgrO), and the Plant Trait Ontology (TO) supported by CGIAR [8]. FOODON integrates a number of existing ontologies but its focus seems to be again on research data although its ambition is to provide a mechanism for data integration across the food system. Considerable efforts have been put into extending and integrating the FOODON ontology with various other ontologies extending its utility to areas such as nutrition, and integrating it with the Foodex2 standard from EFSA [9, 10] As mentioned, most work on ontologies for the agrifood domain has up to now mostly been targeted towards the clear definition of domain concepts and terms in the form of a vocabulary for the annotation of research publications or research data sets [8]. As a result, there is little or no use yet of ontologies for supporting the actual sharing and exchange of data across the agrifood chain. Only a few papers about the use of ontologies for traceability and data analysis in the dairy sector [11, 12]. In addition, a few innovation projects have dealt with the use of ontologies in the horticultural supply chain and greenhouse domain [13]. However, the use of semantic standards for information exchange by standardisation organisations, like GS1, ISO or AgGateway is not yet common practice or even beginning to be picked up. More generally under the auspices of the EC funded projects like ATLAS (<https://www.atlas-h2020.eu/>) and especially DEMETER (<https://h2020-demeter.eu/>), there is support for greater use of semantic standards and technologies where the data and data models are explicitly specified, where URIs are widely used, and where data integration is consequently made far easier [14].

2.3 Architectures for Data management/sharing/ Querying

The use of semantics for data sharing or data exchange in general has a long history epitomised by the Linked Data initiative and more recently the FAIR data movement. There has been a tendency for this to be largely a focus for researchers rather than commercial applications with the exception of the life sciences. Within the food and agriculture sector, the majority of efforts at data sharing have focussed on ontologies for research data and assumed relatively centralised approaches to repositories (e.g. the



D3.1 Data Requirements and Interoperability – initial version

CGIAR platform for Big Data [8]), or else have been based on XML. Three major data standardisation efforts in the agri-food areas are ICAR (for livestock and dairy production), ISOBUS (for machine-to-machine data sharing) and AgGateway (for FMIS applications). These are currently available as XML standards (with codebooks) but are slowly moving towards JSON-LD versions of their standards. [15] make the case for using Linked Data principles and a variety of ontologies so that data can be integrated for farmer decision support. Other literature emphasises the importance of semantics for integrating IoT derived data in agriculture, while [16] similarly uses ontologies to ensure data integration for supply chain data. These papers, as do most others, assume centralised architectures. The Linked Pedigree architecture was proposed to enable data sharing across supply chains by formalising the GS1 EPCIS standards as ontologies and enabling SPARQL queries across distributed triple stores, and this work partly inspires the technological approach described below [17, 18].

2.4 Interoperability and legacy systems

The issue of making existing legacy systems interoperable is not new and has been researched as part of the data integration topic [19]. The goal of data integration is focused largely on the need for answering user queries over distributed relational data sources and using SQL as the query language. The data integration landscape has changed with the increase in (types of) data sources and applications that can benefit from it. Other approaches that use semantic technology propose query-rewriting of the overall query to integrate distributed heterogeneous data sources [20]. This is different from our approach in that the data sources are integrated using semantic domain knowledge instead of a user query. Our focus lies on composing web service APIs of the distributed legacy data sources. Applying semantic technologies to web services has been the topic of extensive research. Two topics can be distinguished, although they are often investigated together.

On the one hand, research has focused on giving semantic descriptions of the inputs/outputs and pre/post conditions of web services. to be able to automatically discover or select relevant services [21]. On the other hand, research focused on how to automatically compose multiple web services together [22]. Despite the many proposals to include semantics into the descriptions of web services [23–25], many of today’s web service APIs do not describe their formal invocation, pre- and post-conditions, and semantic input/output. The research on semantic web service composition in general has investigated several techniques to achieve this, and the approach is similar to rule-based ‘planning’ [26–28]. However, in contrast to these approaches, our approach uses graph patterns of a common semantic model to be able to automatically compose distributed heterogeneous data sources.

In the area of Internet of Things, with the growth of IoT devices efforts have been devoted to overcoming interoperability challenges [29, 30]. In this context the W3C has launched the Web of Things Working Group (<https://www.w3.org/WoT/>) to counter the fragmentation arising from the use of IoT devices and support the use of semantic standards. This work largely builds on the Semantic Sensor Network (SSN) ontology (<https://www.w3.org/TR/vocab-ssn/>). The Open Mobile Alliance and a variety of telecommunication standards bodies have supported the Next Generation Service Interface (NGSI) [31]. The NGSI API and the associated NGSI context model has been adopted by key organisations in the IoT standardisation efforts.

In the work described below, we have used extensively the concepts and principles of semantic web technologies. This enables us to achieve a number of objectives: a) using semantics for the data model allows flexibility as the full range of parameters to be captured in assessment model; b) it ensure flexibility and adaptability when interacting with external data sources; c) it allows for easy construction of complex queries which may not be foreseeable at data model design stage; d) by representing all data collected in a knowledge graph, we can use a growing set of standardised tools for mining the knowledge graph.



3. Technical requirements for the QuantiFarm data platform

This section specifies the technical and data modelling requirements that will guide the implementation of the QuantiFarm data platform. Prior to the specification of these requirements, it is necessary to identify the involved data sources, the data properties that need to be modelled and the data needs of the QuantiFarm tools to be developed. *Figure 2* illustrates in a conceptual and high-level manner the overall flow of data starting from the “Data sources” to be integrated, the “Assessment Framework” that will perform the Economic, social and Environmental assessment of the DATs and finally the set of “Tools” that will consume these data aiming to provide more advanced decision-making services. As it is evident the heterogeneity of data derived by different sources needs to be harmonised in order to ensure the maximal reuse and applicability of the QuantiFarm Assessment Framework and Tools. For this reason the QCSM is introduced that will be utilised for enforcing interoperability at a semantic level.

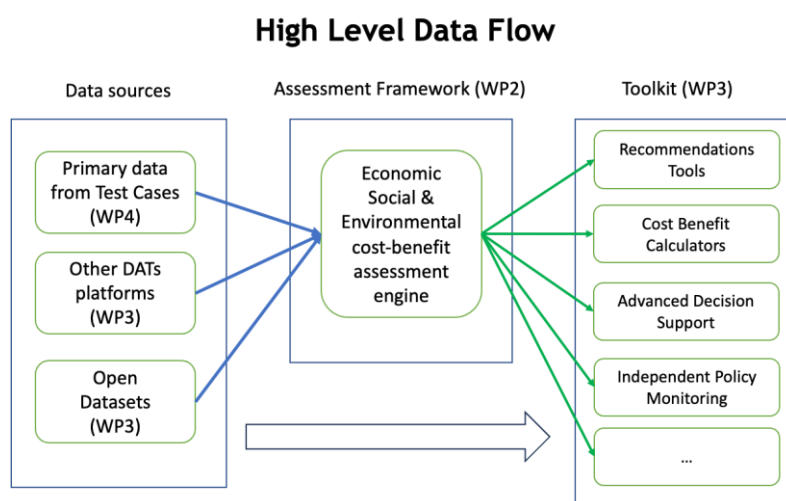


Figure 2 High level data flow from sources via assessment framework to toolkit.

3.1 Data provision by Test Cases and other platforms

3.1.1 Test cases

The project will collect data from 30 Test Cases (TCs) across multiple countries and biogeographical regions. Full details of the TCs are provided in “D4.1 Testing and Assessment Guidelines”. The TCs will enable the evaluation of different DATs in real life conditions, across 3 iterations/growing seasons. Each TC includes one or more farms using a specific DAT, and a set of equivalent and comparable farms not using any DAT. Care is being taken to find 1-to-1 matching so as to ensure consistent, contrasting yet comparable participants in each TC. An overview is provided in Table 1.

No	Sector	Crop/Animal	Digital Solution	Country	Biogeographical Region
1	Arable	Potatoes	SF DSS/ App	Greece	Mediterranean
2	Arable	Corn	Precision irrigation	Portugal	Mediterranean
3	Arable	Wheat	DSS/Agri-environmental Monitoring	Spain	Mediterranean



D3.1 Data Requirements and Interoperability – initial version

4	Arable	Catton	VRA add-on for old tractors	Greece	Mediterranean
5	Arable	Wheat	SF DSS/ App	Turkey	Anatolian
6	Arable	Wheat, onion, potato	Machinery with VRA, data analytics	Netherlands	Continental
7	Arable	Potatoes	SF DSS/ App	Poland	Continental
8	Arable	Wheat, Rapeseed, Rye, Barley	Drones, sensors, silo management, AI	Latvia	Boreal
9	Arable	Barley, Corn, Wheat	FMIS/ Financial Modelling	Slovenia	Alpine
10	Arable	Wheat	FMIS/ app	Romania	Steppe
11	Fruit	Olives	SF DSS/ App	Greece	Mediterranean
12	Fruit	Apples	Drones and soil sensors	Poland	Continental
13	Fruit	Grapes	SF DSS/ App	Italy	Mediterranean
14	Fruit	Strawberry, Blueberry	Precision Irrigation / Variable root pruning	Serbia	Pannonian
15	Fruit	Olives	SF DSS/ App	Cyprus	Mediterranean
16	Fruit	Apples	Drones and soil sensors	Netherlands	Continental
17	Fruit	Grapes	Harvesting robotic and SF DSS	Romania	Black Sea
18	Vegetables	Tomatoes	SF DSS/ App	Italy	Mediterranean
19	Vegetables	Tomatoes	Automated greenhouses	Netherlands	Continental
20	Fruit	Bananas	Precision Irrigation, Monitoring	Spain	Macaronesian
21	Vegetables	Tomatoes	Automated greenhouses	Finland	Boreal
22	Meat	Poultry	Cleaning robot, AI	UK	Atlantic
23	Meat	Cows	Heat box collar	France	Continental
24	Meat	Pigs	Automated monitoring, AI	Belgium	Continental
25	Dairy	Cows	Feeding robot	France	Continental
26	Dairy	Cows	Milking robot	Ireland	Atlantic
27	Dairy	Cows	Automated monitoring	Germany	Continental
28	Dairy	Cows	Livestock feeding DSS	Romania	Steppe
29	Apiculture	Bees	Automated monitoring	Lithuania	Boreal
30	Aquaculture	Oyster	Sensors for quality assessment	Croatia	Mediterranean

Table 1 Test Cases summarization

TCs will act as data providers for the Assessment Framework, collecting data for the assessment of the digital agriculture solutions they are using on their farms, and equivalently to assess the impact of not using DATs. To properly assess the impact of using a digital technology in agriculture in real conditions, the assessment needs to take into consideration the outcomes of the whole growing season. Given the fact that agricultural production is affected by many unpredictable factors such as extreme weather conditions, which can alter the results drastically, there needs to be enough growing seasons. Therefore, testing will start in January 2023 (M7) and will run for three (3) annual cycles, covering an equal number of growing seasons. The data collected by all 30 TCs will be integrated into QuantiFarm data platform and toolkit to support the respective assessment and decision support tools for farmers, advisors and policy makers with the necessary input.

All TCs include farms that are already using specific DATs. The DATs to be assessed in the context of the TCs are the DATs already being used by the farmers. No additional DATs will be developed or selected during the project. There is direct farmer involvement in all QuantiFarm TCs. The DAT is being applied for a particular part of the farm (a parcel or a group of animals) and its performance is being monitored and compared with a similar part of the farm (or another farm) for which the DAT is not applied. Table 2 provides more details on the general DAT information and other TC properties that are being collected per test case.



D3.1 Data Requirements and Interoperability – initial version

Information about TC	
Test Case Property	Data type
Crop / Animal	string
Month of start of the growing season	date
Month of end of the growing season	date
Biogeographical region	string
DAT	string
Total farm size	ha
Number of parcels with DAT(s)	integer
Number of parcels withOUT DAT(s)	integer
Initial investment for DAT(s)	€
Expected lifespan of each DAT	years
Time to install and make the DAT(s) operational	days
DAT(s) maintenance costs	€/year

Table 2 Generic test case properties

3.1.2 External inventories of DATs

One of the key objectives of the QuantiFarm project is to build on the results and lessons learnt from existing platforms that already maintain a large number of DATs. To avoid reinventing the wheel, the QuantiFarm Data Platform and Toolkit will be built on top of the databases of the Smart-AKIS and FAIRshare platforms. Although these two platforms maintain the descriptions of a large number of DATs, access to these is only available through web interfaces. To this end and in the context of the QuantiFarm project both platforms have been extended to facilitate data exchange through an API. This will create a unified, comprehensive data platform that brings together important information and tools from different sources. APIs will allow QuantiFarm to interact with various external platforms and consume their data and functionalities. At the same time, QuantiFarm 's own data and functionality can also be exposed and made accessible to these external platforms.

Integration with FAIRshare

One of the key platforms that QuantiFarm integrates with is FAIRshare (<https://fairshare-pnf.eu/>). FAIRshare's mission is to encourage farm advisors to utilise Digital Advisory Tools and Services¹ that contribute to more productive and sustainable agriculture. *Figure 3* provides a screenshot of FAIRShare's inventory of DATS. As it is illustrated the inventory currently maintains about 300 DATS and it also provides a set of filters that uses as input user defined preferences in order to refine and personalise the rendered results.

¹ In the context of FAIRshare a DAT is a “Digital Advisory Tool or Service” as opposed to the QuantiFarm definition as “digital technologies in agriculture”.



D3.1 Data Requirements and Interoperability – initial version

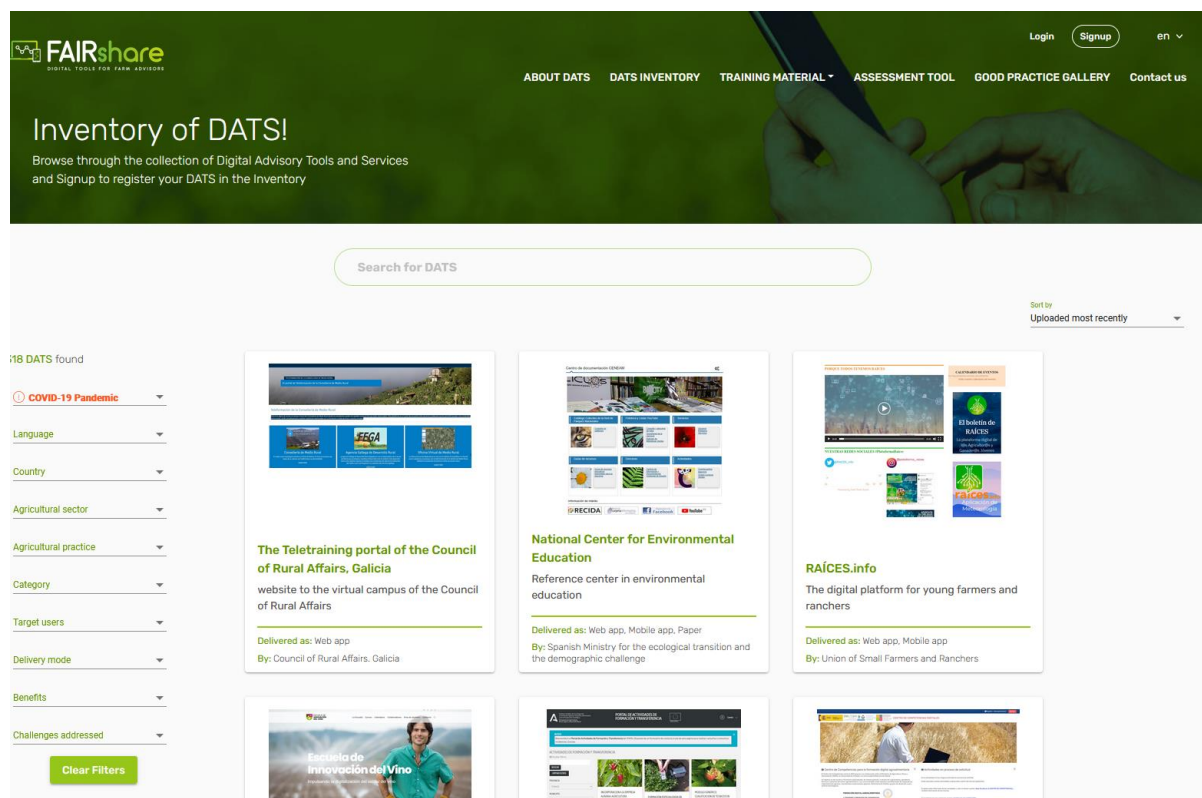


Figure 3 The FAIRshare inventory of DATS

The FAIRShare inventory provides the necessary functionality through the website for registered users to add, update, and remove DATS. The platform already had an internal API to allow communication between a front end (in Angular) and the backend (in Node.js and MongoDB).

To enable integration with QuantiFarm, FAIRshare's API underwent an extension of functionalities. The process involved addressing minor pre-existing issues, such as character encoding, and optimizing functionalities to prevent critical errors. The result was a more efficient and reliable API that can provide good results based on the keywords provided. The expanded API now includes critical calls like 'searchTools', which retrieves DATS that includes at least one of the provided keywords, and 'Tools', which enables the retrieval of the entire collection of DATS available on the FAIRshare database. To aid in this integration and for better understanding, a detailed documentation of the Fairshare API, presenting technical information and examples of the responses for each of the available requests can be accessed here: <https://documenter.getpostman.com/view/26161901/2s93JnW7Sx>.

In Annex C the properties used for describing a DATS are presented. As it will be described in section “4.2 Modelling digital agricultural technologies”, these properties are also incorporated within the QCSM ontology.

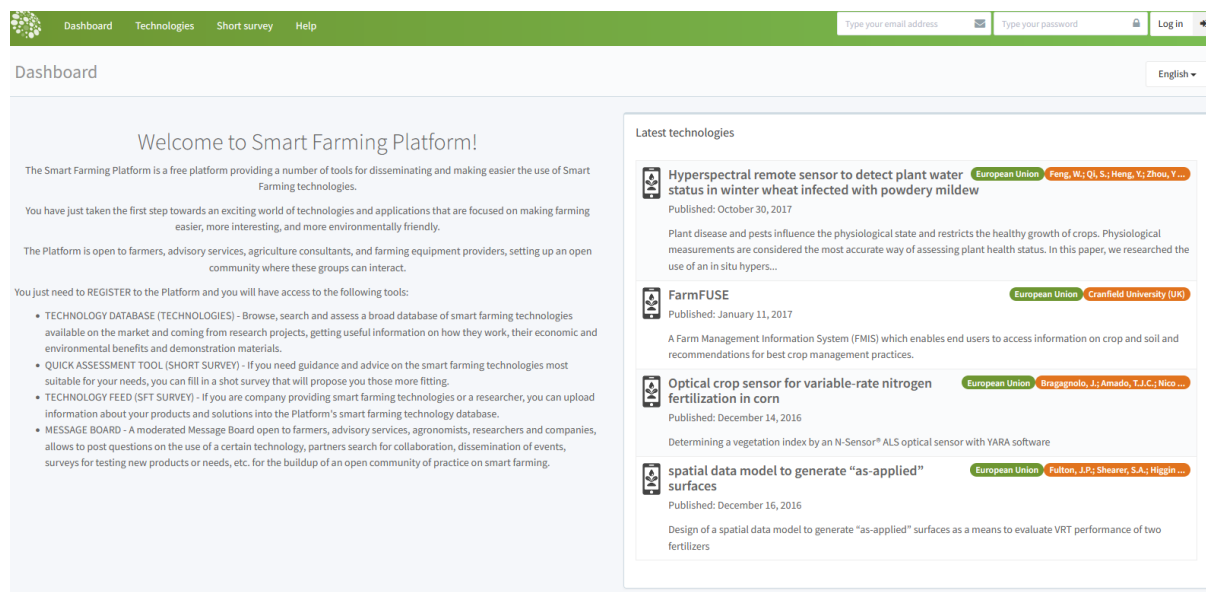
Integration with Smart-AKIS

Smart-AKIS is another external platform with which QuantiFarm has integrated. It is a free platform providing tools for the dissemination and application of Smart Farming technologies, available here <https://smart-akis.com/SFCPPortal/#/app-h/dashboard> and illustrated in Figure 4. Smart-AKIS dashboard provides a platform that allows users to browse, search and assess a broad database of smart farming technologies available on the market and coming from research projects, getting useful information on how they work, their economic and environmental benefits and demonstration materials. It must be noted that according to our research the registration of new technologies to the platform is



D3.1 Data Requirements and Interoperability – initial version

limited while the most recent update was in 2019. This presents a significant challenge as the digital agriculture sector is very fast moving.



The screenshot shows the 'Smart Farming Platform' dashboard. At the top, there is a navigation bar with 'Dashboard', 'Technologies', 'Short survey', and 'Help'. On the right, there are input fields for 'Type your email address' and 'Type your password', along with a 'Log in' button. Below the navigation bar, the main content area is titled 'Dashboard' and includes a language selector set to 'English'. The main content is divided into two columns. The left column contains a 'Welcome to Smart Farming Platform!' message, followed by a brief description of the platform's purpose and a list of tools available to users, such as 'TECHNOLOGY DATABASE (TECHNOLOGIES)', 'QUICK ASSESSMENT TOOL (SHORT SURVEY)', 'TECHNOLOGY FEED (SFT SURVEY)', and 'MESSAGE BOARD'. The right column is titled 'Latest technologies' and lists four recent publications: 'Hyperspectral remote sensor to detect plant water status in winter wheat infected with powdery mildew', 'FarmFUSE', 'Optical crop sensor for variable-rate nitrogen fertilization in corn', and 'spatial data model to generate "as-applied" surfaces'. Each entry includes a brief description, a publication date, and tags for 'European Union' and the authors.

Figure 4 The Smart-AKIS inventory of Smart Farming technologies

As with the case of FAIRShare, the Smart-AKIS platform only supports interactions with external entities through the GUI. In order to facilitate integration with the QuantiFarm's data platform, an API for Smart-AKIS platform was developed using the Laravel framework (<https://laravel.com/>). The Smart-AKIS API supports queries to the underlying database for retrieving smart farming technologies with the use of filters such as: 'Category options', 'Countries', 'Crop Systems', 'SFT Types', 'SFT Effects', 'Crop Types', and 'Technologies'. These calls enable the retrieval of different subsets of information. The detailed documentation for the Smart-AKIS API is accessible here: <https://documenter.getpostman.com/view/2626884/2s93m33iXy>.

In conclusion, the integration with external platforms such as FAIRshare and Smart-AKIS has provided a core dataset for QuantiFarm. It extends the range of available data and functionalities, thereby enhancing the value offered to end-users. Such integration efforts also underscore QuantiFarm's commitment to promoting interoperability and seamless data exchange in the agricultural domain. As will be described in the following sections, "Data Loaders" will periodically connect to SmartAKIS and FAIRShare databases in order to retrieve any available updates of digital agricultural technologies and store them within the QuantiFarm data platform.

3.1.3 Open Datasets

In some cases, it will be necessary to use datasets that are provided by third parties. For example, in order to capture the context that a DAT is operating in, it will be important in some cases to record weather parameters that have occurred during the cultivation period. In case weather monitoring HW is not installed in the Test Cases parcel then public data repositories with past and current measurements will be utilised. Especially for the operation of the Independent Monitoring Policy tool (more on this will be presented in Section 3.3) the Copernicus Climate Data store is being evaluated for integration. This source provides daily surface meteorological data for the period from 1979 to present as input for agriculture and agro-ecological studies. This dataset is based on the hourly ECMWF ERA5 data at surface level and is referred to as [AgERA5](#). This dataset is considered useful to identify short and long term climate change trends and anomalies affecting agricultural performance.



D3.1 Data Requirements and Interoperability – initial version

Other relevant sources are those providing data on national and European level relevant with common agricultural policy (CAP) performance indicators. Of interest for the QuantiFarm project are dataset and calculated indicators derived by the performance monitoring and evaluation framework ([PMEF](#)) assessing CAP 2023-27 national strategic plans. In a similar manner, the data from the Farm Accountancy Data Network ([FADN](#)) and its successor scheme Farm Sustainability Data Network ([FSDN](#)) will be considered for integration. Context indicators provide information on the agricultural sector and rural areas as well as general economic and environmental trends. Some of this information drills down to regional level (NUTS 2-3).

Other datasets that are likely to be useful and relevant to policy implementation include data sets on the boundaries of [environmental sensitive/important regions](#) such as: Natura2000 areas, Wildlife sanctuaries, Waterbodies, Biodiversity indicators, etc.

It should be noted that given the complexity and the large volume of these datasets, we will avoid duplicating such data within the QuantiFarm Data Platform. A more optimal approach will be followed by retrieving and integrating them directly according to the decision-making needs.

3.2 Assessment Framework

Among the main objectives of the QuantiFarm project is to develop an Assessment Framework for evaluating the value provided by DATs in agriculture. A mixed-method approach will be used including the farm as well as the wider society for the co-creation of the framework. The Assessment Framework will combine different quantitative and qualitative methodologies to conduct a comprehensive assessment of the economic, social and environmental benefits and costs of DATs. The first specification of the “Assessment Framework” is documented in D2.1 “Assessment Framework and Governance Mechanisms – initial version”. According to these specifications, the initial outcome of the assessment process will be a composite monetary index which will be complemented with a set of additional descriptive indicators on the impact of DATs. These are intended to avoid the risk of drawing simplistic conclusions when relying only on a single value to capture the complexity of multi-dimensional aspects.

The parameters that are being used to measure the performance of DATs are defined in D2.1. The “Assessment Framework” also defines sustainability categories and indicators in the environmental, economic and social domains. Each of the test cases will report repeatedly on all of these categories and indicators, in addition to some general information about the DAT. Seven categories have been identified in the environmental area; 4 in the economic domain and 2 in the social one. Twenty subcategories of performance measurement have been identified, for which there are 80 indicators (48 environmental, 16 economic and 16 social). The final set of indicators to be included in the assessment framework has taken into account the particularities of the test cases in terms of: crop/product, geographical location, harvesting season and DAT used, if any. The various sustainability domains, categories and sub-categories are summarised in [Table 3](#).

DOMAIN	DOMAIN ID	CATEGORY	CATEGORY ID	SUB-CATEGORY
Environmental	EN	Atmosphere	AT	Greenhouse Gases Air quality
		Water	WA	Water withdrawal Water quality
		Land	LA	Soil chemical properties



D3.1 Data Requirements and Interoperability – initial version

				Soil biological properties
		Energy & Inputs	EI	Energy use Renewable energy Nutrients use Pesticides use
		Waste	WS	Generated waste
		Biodiversity	BI	Biodiversity conservation
		Animal Health and Welfare	AHW	Animal health Animal welfare
Economic	EC	Profitability	PF	-
		Productivity	PD	-
		Efficiency	EF	-
		Food Quality	FQ	-
Social	SO	Internal social sustainability	IS	Education
				Working conditions
				Food safety
		External social sustainability	ES	Local community involvement and participation
Transparency and visibility				

Table 3 QuantiFarm 's sustainability domains, categories and sub-categories

The “Assessment Framework” and the respective algorithms that calculate the indicators reported in [Table 3](#) will be coded as a reusable software component that will support the dynamic provision of different input datasets referring to different DATs and respective contexts. The dataset to be utilised as input and the calculated performance indicators of each sub-category have been further detailed into measurable parameters that are being modelled in our QuantiFarm Common Semantic Model (QCSM). This is described in more detail in Chapter 4.

3.3 Data needs for the QuantiFarm Tools

Besides the “Assessment Framework” QuantiFarm will design and develop a set of software tools that will offer a variety of services. These services will be bundled in the “QuantiFarm Toolkit” and will be offered as **interactive and user-friendly** web-based tools. These tools are intended to fulfil the needs of specific user groups, viz. Farmers, Advisors and Policy Makers. The overall objective is to provide access to the assessment results and support decision-making across the various contexts that are analysed within the QuantiFarm 's project Test Cases. Through a personalised interface, the Toolkit will present information on the DATs' costs, benefits and sustainability impact, and thereby support both individual decision making and broader policy design. The detailed specifications and implementation details of the tools will be documented in D3.2 “Tools for DATs Assessment and Policy Monitoring” (to be delivered in December 2023). Nonetheless at this stage it is important to identify core requirements, especially regarding data. These requirements guide the specification of the data modelling properties of the QuantiFarm data model.

- a) Tool title: “**Cost and benefit calculator**”

Short description: A financial cost and benefit calculator for different sectors and different types of DATs will be available to farmers and advisors. After entering the field size, crop or livestock system



D3.1 Data Requirements and Interoperability – initial version

and the DAT, the calculator will be able to provide the potential benefits and costs (both running and investment costs) when using a specific DAT under certain conditions, compared to not using this DAT. These include different calculations for DATs in livestock (e.g. robotic milking, virtual fencing, automatic feeding, livestock grazing, etc), arable crops (e.g. VRA for fertilisers, robotic field operations for seeding, spraying, weeding, etc) and fruits and vineyards (e.g. precision irrigation, precision spraying, selecting harvesting, etc). In addition, the calculator will also calculate the risk in the investment or the increased risk in the production due to climate change adaptation. These calculations are described in “D2.5 Cost and Benefit Calculators Design”. According to these specifications the following set of required data types are identified, each focusing in a different application domain.

Application domain: “Crop Farming Systems”

- Calculator’s Module name: Investment Cost Formulas
Data types required: Years of Usage of DAT, DAT’s Initial Cost of Investment (€), Subscription Cost for 1ha (€), Monthly Subscription Cost (€), Annual Subscription Cost (€), Number of Units (sensors etc.), Total Cost of DAT Purchase (€)
- Calculator’s Module name: Yield Increase Calculator
Data types required: Yield increase (%), Current Yield (tons/ha), Market Price (€/1 Ton), Current Revenue (€ in 1 year), Increased Yield (tons/ha), Price of Increased Yield (€/ ha).
- Calculator’s Module name: Revenue Increase Calculator
Data types required: Price of Increased Yield (€/ha), Total Area (ha), Current Revenue (€/year)
- Calculator’s Module name: Fertiliser Use Calculator
Data types required: Fertilization saving (%) Current fertilizer usage (Kg/ ha) Current Fertilizer Cost (€/ 1 Kg) Current Fertilization Cost (€/ ha) Reduced Fertilizer usage (kg/ha) Fertilizer cost savings (€/ ha) Fertilization cost saving (€ in 1 year)
- Calculator’s Module name: Water Use Calculator
Data types required: Water saving (%), Current water usage (m3/ha), Current water cost (€/ m3), Current Irrigation Cost (€/ ha), Reduced water usage (m3/ha), Water cost savings (€/ ha), Water cost savings (€ in 1 year)
- Calculator’s Module name: Pesticide Use Calculator
Data types required: Pesticide saving (%), Current pesticide usage (kg or lt/ ha), Current pesticide cost (€/ kg or lt), Current pesticide cost (€/ ha), Reduced pesticide usage (kg or lt /ha), Pesticide cost saving (€/ ha), Pesticide cost savings (€ in 1 year)
- Calculator’s Module name: Labour Cost Calculator
Data types required: Labor saving (%), Current Labor Cost (€ in 1 year), Labor cost savings (€ in 1 year)
- Calculator’s Module name: FUEL Cost Reduction Calculator
Data types required: Fuel Saving (%), Current Fuel Cost (€ in 1 year), Fuel cost savings (€ in 1 year)

Application domain: “Livestock farming Systems Calculator”

- Calculator’s Module name: Investment Cost Calculator
Data types required: Number of animals, Years of Usage, Initial Cost of Investment (€), Start-up Fee per Animal (€), Monthly Cost per Animal (€), Number of Units (sensors etc.)
- Calculator’s Module name: Milk Yield Increase Calculator
Data types required: Milk Yield Increase (%), Average Milk Price per Litre (€/Litre), Average Litres of Milk Produced per animal per Day (Litre), The Price of Milk Yield in One Year (€)
- Calculator’s Module name: Energy Cost Calculator
Data types required: Energy Saving (%), Current Energy Consumption (in kWh), Cost of Energy per kWh (€), Current Cost of Energy (€/year), Energy Cost Savings (€/year)



D3.1 Data Requirements and Interoperability – initial version

- Calculator's Module name: Feed Cost & Waste Calculator
Data types required: Feed Saving (%), Current Feed Cost (€/year), Feed Cost Saving (€/year), Feed Waste Saving (%), Current Feed Waste Cost (€/year), Feed Waste Cost Saving (€/year)
- Calculator's Module name: Antibiotics Cost Calculator
Data types required: Antibiotics Saving (%), Current Antibiotics Cost (€/year), Antibiotics Cost Saving (€/year)
- Calculator's Module name: Mortality Cost Calculator
Data types required: Mortality Rate Decrease (%), Current Mortality Cost (€/year), Mortality Cost Saving (€/year)

b) Tool title: “Criteria-based interactive recommender tool”

Short description: A criteria-based interactive recommender tool will allow farmers (and farm advisors) to select the most appropriate solutions for their identified needs under a given set of conditions. Based on the criteria specified by the users, the tool will propose specific DATs that cover their needs and will provide specific quantified information on the costs, benefits and impact on sustainability of each solution. The user will be able to see the overall costs and benefits of each DAT in the form of a monetary value (in Euros) and with details provided on how this was calculated. The tool will be easy to use and the farmer will not need special training or the help of the advisor. Technically the tool will be based on a knowledge-based recommender system (building on existing systems and expertise) using a combination of reasoning and constraint satisfaction to provide a ranked list of the most suitable DAT for any given farm/crop/biogeographical region combination. The Recommender tool takes various data elements as input.

The first main category will be the characteristics of the requesting farmer and his/her context:

- Name of the farm
- Country of the farm
- Biogeographical region (flat land, mountains)
- Agricultural sector (arable, horticulture, meat, dairy, apiculture, aquaculture)
- Agronomic model (Organic or traditional/industrial/greenhouse etc.)
- Type of agronomic activity (crop type/livestock/dairy)
- Location (GPS coordinates: latitude, longitude, altitude – or derived from postal address)
- Size of parcels/greenhouse/animals (ha or number)
- Soil type (sand, clay, silt, loam, peat, chalk)

These characteristics are used to search for similar farms and contexts that have used or are currently using DATs which might be of interest to the requesting farmer.

The second category of input data are the parameters of a DAT the farmer is searching for:

- Functionality
- Target agricultural sector
- Target crop type
- Benefits/challenge to be solved

The Recommendation tool will take values for these parameters indicated by the farmer and searches for those DATs with fully matching or closely similar values. This list can then be presented to the farmer.

The third category of input data are the aspects that the farmer wants to know about the selected DATs:

- Initial investment costs
- Maintenance costs



D3.1 Data Requirements and Interoperability – initial version

- Expected ROI timeline
- Installation time
- Lifespan
- Amount of training needed
- Usability level
- Interoperability with other DATs
- Regional suitability
- Environmental sustainability

The output of the Recommendation Tool is a list of DATs and their characteristics that match the requested input criteria. In addition, for each DAT a calculation is made of the costs and benefits for the specific farm and its context. This is expressed in terms of monetary value. Upon request, the farmer can get insight into the specific sub-categories that are used to calculate this monetary value.

c) Tool title: “**Policy Monitoring tool**”

Short description: The “Policy Monitoring” tool will integrate and render -on a parcel and/or regional level- EO data products (e.g. crop type, NDVI, NDWI, land use), in-situ information (e.g. farm calendar exports describing applied farming practices, digital logs by utilised DATs) along with qualitative and quantitative results from questionnaires on evaluated DATs performance. The Policy Monitoring tool will provide policy makers with a visual policy monitoring dashboard, allowing the generation of analytical reports based on analytical queries, including summary tables as well as graphical charts. The Policy Monitoring tool will also facilitate data exchange through standardised API. A number of QuantiFarm test cases have been selected to evaluate their integration based on the technologies that will be assessed. An indicative list of TCs is: 1, 3, 4, 7, 8, 11, 17, 27, 29, 30 while the respective technologies that will be evaluated as ground truth data providers are Farm Management Information Systems, remote sensing with the use of UAVs , Variable Rate Application components applied on tractors, livestock automated monitoring, milking robots, automated monitoring for bees, robotic harvesting.

Data needs:

- Parcels location expressed in polygons with the use of coordinates,
- Farm/cultivation properties, applied farming practices and access to digital farm calendars,
- Digital data logs generated by the use of DATs.
- Earth Observation data products (e.g. parcels shapes, NDVI, crop type estimation)
- Other relevant open datasets/repositories for the areas of the test cases: climate baseline values, biodiversity indicators, polygons-coordinates for environmental sensitive areas (Natura2000, water bodies, wildlife sanctuaries).

The data outputs will be selected policy performance indicators. Examples: Fertilisers (N/P/K kg/ha), Pesticides (list of active substances, quantity per substance of product, frequency of applications), Irrigation (m³/hectare/year, total m³/year, frequency of irrigations), Land management (frequency of applied practices- ploughing, mowing), Harvested yield (number of harvests, quantity, quality).

As a final note, all tools are currently under specification and their design details including their data needs and calculated outputs will be documented in the respective deliverables. The flexible data modelling approach that the QCSM follows supports the necessary adaptations e.g. incorporation of additional domain specific ontologies. Hence, final specification of QuantiFarm Tools will dictate the final form of the QCSM.



3.4 Technical Requirements specification

The main purpose of the QuantiFarm data platform is to provide data about Digital Agricultural Technology (DAT) coming from various, heterogeneous sources in a uniform and unambiguous form to a set of QuantiFarm tools to be used by farmers, advisors and policy makers. This includes static data about DATs, such as title, description, provider, functionality etc., but also dynamic data about its usage in the context of a specific setting, such as the QuantiFarm test cases.

Functional requirements:

- A QuantiFarm database with DAT information including static and usage DAT data forms the core of the QuantiFarm data platform.
- The QuantiFarm database should provide data that is compliant to a common semantic model.
- The QuantiFarm tools can access the data in the QuantiFarm database upon request using a question-response mechanism.
- The QuantiFarm database collects information about DATs from other existing DATs repositories, such as FairShare and SmartAKIS.
- The QuantiFarm database collects all information of the QuantiFarm test cases about the usage of DATs in practical situations.
- The QuantiFarm database maintains a mandate scheme that is used for secure access to the QuantiFarm test case data.
- The QuantiFarm database provides an identification, authentication and authorization mechanism that supports secure access control to its data based on the mandate scheme.

Non-functional requirements:

- Performance: a query on the QuantiFarm database must respond within a reasonable time from the end-users perspective. The exact acceptable response time depends on the complexity of the query. If needed intermediate results need to be returned iteratively to satisfy the users request.



4. QuantiFarm Common Semantic Model

Based on the requirements for QuantiFarm data platform, a first version of a semantic model with common concepts and relations has been developed, the QuantiFarm Common Semantic Model (QCSM). In the next section, the design of the QCSM is described, its guiding principles and existing ontologies that have been reused and extended for modelling the required concepts and relations in the QCSM. Then, in the next subsections, the modelling of digital agricultural technologies and sustainability parameters in the QCSM is briefly identified. The complete QCSM and some of the reused and extended ontologies can be found in Annex A.

4.1 Design of the QCSM

The design of the QCSM has been guided by a number of principles described in the next subsection. Afterwards, the existing ontologies that have been reused and extended are being listed.

4.1.1 Guiding principles

For the design of the QCSM, the following guidelines have been used:

1. The QCSM is based on semantic technologies, like RDF and OWL, because it is currently the best way of intuitively defining formal semantics (OWL) and provides the flexibility for modular reuse of existing data models or extending them.
2. The QCSM should be a small, core model that covers the main common concepts in the agrifood domain with a focus on the farm, its digital agricultural technologies and sustainability performance.
3. The concepts and relations for the QCSM are selected from general properties of digital agricultural technologies on the farm and sustainability characteristics as identified by the assessment framework for the QuantiFarm test cases.
4. Existing ontologies that already define required concepts are reused by the QCSM as much as possible. In addition, proposals for extension of existing ontologies are being made as long as this fits into the scope of these existing ontologies.
5. Required concepts that can not be modelled in or added to an existing ontology will be defined within the QuantiFarm namespace for the QCSM, namely <https://quantifarm.eu/ontology/> prefixed as `qcsm`.
6. Existing ontologies are only reused when they have a clear formal OWL structure that is publicly available and accessible or downloadable in a .owl, .ttl or .rdf format, for instance at OBO foundry website, W3C website or the AgroPortal (<http://agroportal.lirmm.fr>). Consequently, no reuse of proprietary ontologies of different projects will be done.
7. Vocabularies and thesauri/taxonomies that simply define and list a large set of hierarchical terms will not be reused in the QCSM other than using the `rdf:isDefinedBy` property to point to the definition of the concept in a vocabulary.

A consequence of these guiding principles is that existing data models that are not proper ontologies are not explicitly inherited and extended by the QCSM. Unfortunately, extensive data models like NGS-LD and the Demeter AIM fall into this category.

4.1.2 Reuse and extension of existing ontologies

As a consequence of guiding principle 4 on reusing and extending existing ontologies as much as possible, the QCSM imports various existing ontologies. We therefore have built further on the work that has been done in the Ploutos project (<https://ploutos-h2020.eu>) in which already various ontologies have been reused and extension proposals for some of them have been generated. The specifications of the Ploutos Ontology and the semantic interoperability mechanisms are available in the scientific



D3.1 Data Requirements and Interoperability – initial version

publication “[Data Sharing in Agricultural Supply Chains: Using semantics to enable sustainable food systems](#)” [32] but also in the respective Ploutos Deliverables: [D4.2: Data interoperability for the agri-food sector](#) and [D4.3: Data interoperability enablers](#). Table 4 below sketches the reused ontologies and their scope.

Prefix	Name	Base URI	Scope
ENVO	Environment Ontology	http://purl.obolibrary.org/obo/envo.owl#	Ecosystems, environment, geographical components and processes
S4AGRI	SAREF4AGRI	https://saref.etsi.org/saref4agri/	Smart appliances and devices for agricultural purposes
SSN	Semantic Sensor Network	http://www.w3.org/ns/ssn/	Sensor networks and processes
SOSA	Sensor Observation Sample Actuator	http://www.w3.org/ns/saso/	Observations, actuations and samples from sensors and actuators
OM	Ontology of units of Measure	http://www.ontology-of-units-of-measure.org/resource/om-2/	Units hierarchy for measurements
Weather	BIMERR Weather Ontology	https://bimerr.oit.linkeddata.es/def/weather#	Weather phenomena and parameters
GS1	GS1 web vocabulary	https://gs1.org/voc/	Supply chain product information including certification
Ploutos	Ploutos Common Semantic Model	https://www.tno.nl/agrifood/ontology/ploutos/common#	Supply chain actors and operations

Table 4: Existing ontologies to be reused by the QCSM.

It should be noted that one of the main criteria for selecting the ontologies to be reused and integrated within the QCSM is to be accepted as an official standard or to be widely accepted by the community (de-facto standard). As it is evident in table 4 reported ontologies are supported by organisations such as the European Telecommunications Standards Institute ([ETSI](#)), the Genomic Standards Consortium ([GSC](#)), the World Wide Web Consortium ([W3C](#)), and the [GS1](#).



D3.1 Data Requirements and Interoperability – initial version

As part of the Ploutos use cases on traceability and certification, proposals have been made for the extension of the Saref4Agri, GS1 and Weather ontologies with new concepts and/or properties. In the context of the QuantiFarm project, reuse and extension proposals have been made again to Saref4Agri as will be discussed in the next sections.

4.2 Modelling digital agricultural technologies

Information about digital agricultural technologies are at the moment being collected and made available by a few platforms, specifically FairShare and SmartAkis. In addition, the QuantiFarm test case assessment framework defines another few properties of DATs. The collection of the most important DAT properties is defined in Table 5.

Name	Description	QCSM property
Title	Short title of the DAT	qcsm:DAT -> qcsm:hasTitle -> xsd:string
Description	Short description of the DAT and its purpose	qcsm:DAT -> qcsm:hasDescription -> xsd:string
Keywords	Keywords that characterise the DAT and its usage	qcsm:DAT -> qcsm:hasKeyword -> xsd:string
Provider name	Supplier of the DAT where it can be purchased	qcsm:DAT -> qcsm:hasProvider -> xsd:string
Functionality	Main services the DAT supports	qcsm:DAT -> qcsm:hasFunction -> xsd:string
Agricultural sector	Sector in the agriculture in which the DAT is meant to be used	qcsm:DAT -> qcsm:isTargetedtoSector -> xsd:string
Benefits	Main advantages of using the DAT	qcsm:DAT -> qcsm:hasBenefits -> xsd:string
Initial investment	Initial capital expenditures to purchase and start using the DAT	qcsm:DAT -> qcsm:hasInitialInvestment -> xsd:decimal
Maintenance costs	Operating expenditures for using the DAT	qcsm:DAT -> qcsm:hasMaintenanceCost -> xsd:decimal
Lifespan	Duration in which the DAT can be used normally functioning	qcsm:DAT -> qcsm:hasLifespan -> time:Duration



D3.1 Data Requirements and Interoperability – initial version

Installation time	Time interval in which the DAT can be made operational	qcsm:DAT -> qcsm:hasInstallationTime -> time:Interval
-------------------	--	---

Table 5: Main DAT properties modelled in the QCSM.

In the QCSM, the class of DATs is modelled as the concept *qcsm:DigitalAgriculturalTechnology* with various datatype properties of basic types like *xsd:string*, *xsd:date* and *xsd:decimal*. In addition, relations with other concepts are defined as object properties to for instance the *qcsm:TechnologyProvider*, *saref4agri:Farm* and *gs1:Country* classes. An overview of the DAT concept, its properties and relations with other concepts is visualised in Figure 5.

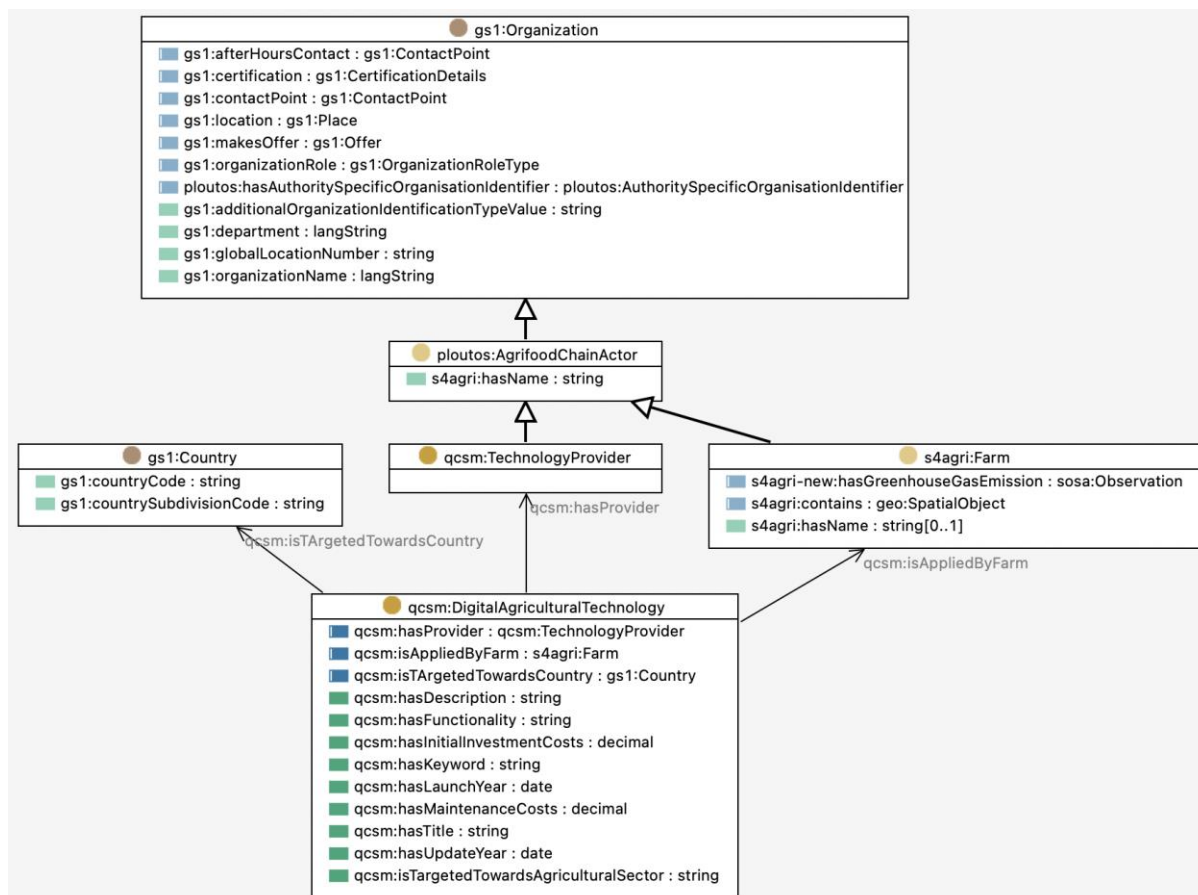


Figure 5 QCSM model with DAT concept and properties.

4.3 Modelling sustainability parameters

Within the QuantiFarm assessment framework three main pillars of sustainability parameters are being distinguished:

- Environmental sustainability, which includes sustainable treatment of the atmosphere, water, soil, waste, animals, etc.
- Economic sustainability, which includes long term viability of the farm, which includes sustainability parameters like production costs, sales revenues, land productivity and product quality.
- Social sustainability, which includes socially responsible sustainability parameters like proper education and treatment of personnel, working conditions, safety, involvement, transparency etc.



D3.1 Data Requirements and Interoperability – initial version

4.3.1 Environmental sustainability parameters

Table 6 contains the environmental sustainability indicators per category defined in the QuantiFarm assessment framework.

Category	Indicator	Unit of measurement
Atmosphere	Greenhouse gas emissions	
	Per greenhouse gas: CO ₂ , CH ₄ , N ₂ O, SF ₆ , etc.	t
Water	Water Consumption / productivity	m ³ /ha, l/m ²
	Volume of water applied for irrigation (or other purposes)	m ³ ; l
	Irrigated area	ha; m ²
	Crop yield	t; kg
Soil	Soil chemical concentrations	
	Total Soil Nitrogen	mg/kg, ppm
	Available Soil Phosphorus	mg/kg, ppm
	Available Soil Potassium	mg/kg, ppm
Energy & Input	Energy use	
	Fuel oil/diesel/propane consumption	l
	Gas consumption	m ³
	Electricity consumption	kWh
	Renewable energy share	kWh
Energy & Input	Fertiliser use, per Fertiliser:	
	Amount of nitrogen used	kg N / ha
	Amount of phosphorus used	kg P / ha
	Amount of potassium used	kg K / ha
Energy & Input	Pesticides applied (herbicides, insecticides, fungicides)	
	Amount of pesticides used	kg active ingr / ha
	Number of applications of pesticides in a growing season	-
Waste	Total amount of waste generated	kg; t
Animal health	Ease of movement	d
Animal health	Total indoor area per animal	m ² /animal
Animal health	Mortality rate	%
	Number of deaths in a year	-
	Average herd size	-
Animal health	Mortality rate at birth	
	Animals died in the first 24h	-
	Total number of animals born in year	-

Table 6: Main environmental sustainability indicators of the assessment framework.

These indicators and the information that is needed to determine them are modelled in the QCSM by extending concepts like water, soil, pesticides, fertilisers and operations with various properties. A specific part of the QCSM that models all the possible operations on a parcel, including irrigation operations, fertilisation operations and chemical control operations in which water, fertiliser and



D3.1 Data Requirements and Interoperability – initial version

pesticides respectively are added to the soil of a parcel. An overview of the *saref4agri:Parcel* concept, its *ploutos:Operations* and the use of water, fertiliser and pesticides is given in Figure 6.

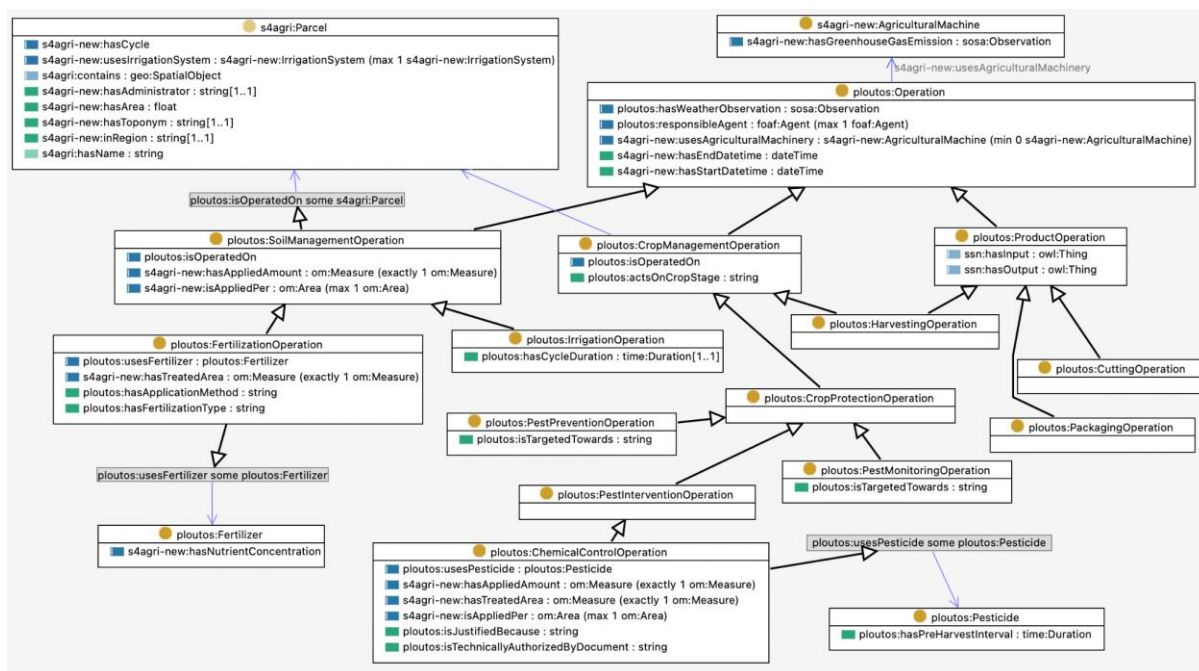


Figure 6 QCSM operations and the use of water, fertiliser and pesticides.

4.3.2 Economic sustainability parameters

Table 7 contains the economic sustainability indicators per category defined in the QuantiFarm assessment framework.



D3.1 Data Requirements and Interoperability – initial version

Category	Indicator	Unit of measurement
Profitability	Production costs	€/ha
	Cost for water	€/ha
	Cost for fertilisers	€/ha
	Cost for pesticides	€/ha
	Costs for seeds, seedlings and trees	€/ha
	Cost for fuel oil/diesel/propane	€/ha
	Cost for electricity	€/ha
	Labour costs	€/ha
	Cost of training for the use of DATs	€/ha
Profitability	Sales revenues	€
	Crop yield	t; kg
	Average Price	€/t; €/kg
Productivity	Land productivity	t/ha; kg/m ²
	Total production	t; kg
	Harvested area	ha
Efficiency	Feed conversion ratio	-
Efficiency	Rate of time for quality analysis	%
Efficiency	Rate of time fulfilled orders	%
Efficiency	Number of wrong orders	-
Food quality	Intrinsic product quality	

Table 7: Main economic sustainability indicators of the assessment framework.

These indicators and the information that is needed to determine them are modelled in the QCSM by adding new concepts like pricing, order, labour, training and extending concepts like feed and product with various properties. A specific part of the QCSM that models the operations on a parcel in which fertilisers and pesticides are added to the soil of a parcel is shown in Figure 7. It also includes the cost in Euros of a fertiliser or pesticide per unit of volume, e.g. litre or kilograms.



D3.1 Data Requirements and Interoperability – initial version

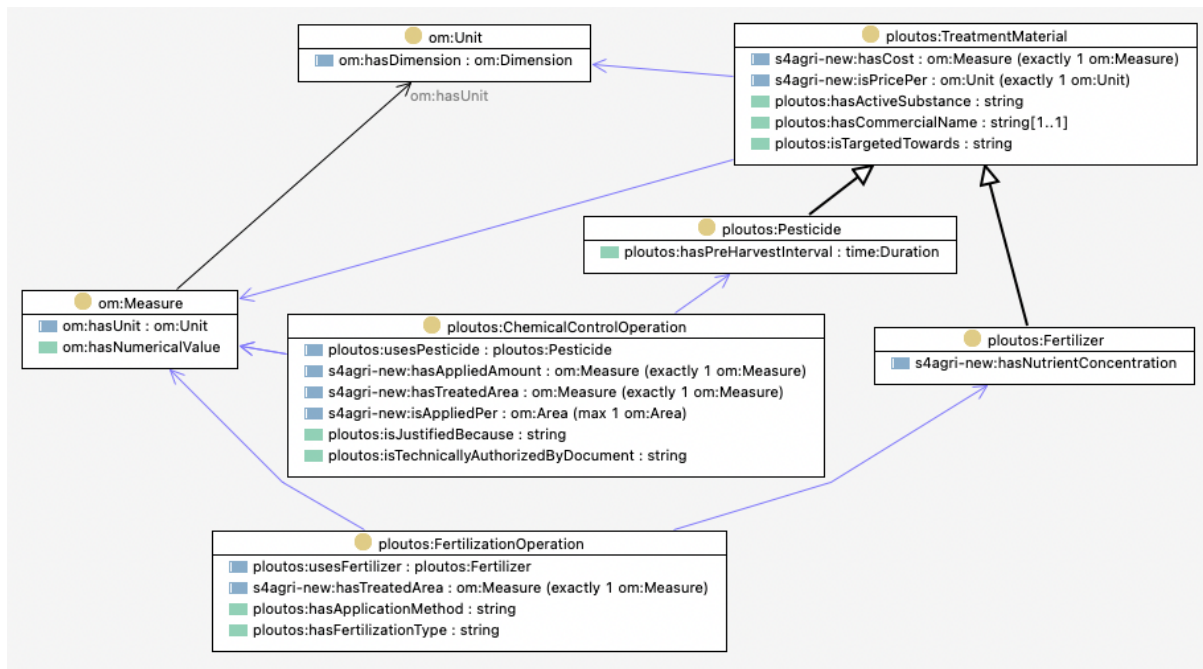


Figure 7 QCSM fertiliser and pesticide operations.

4.3.3 Social sustainability parameters

Table 8 contains the social sustainability indicators per category defined in the QuantiFarm assessment framework.



D3.1 Data Requirements and Interoperability – initial version

Category	Indicator	Unit of measurement
Education	Training hours	h/employee
	Number of training hours for all the employee	h
	Number of employee	-
Labour	Working time	h/kg/w
	Hours worked per type of worker (e.g., hours of full time employees)	h/kg
	Hours worked per type of worker (e.g., hours of seasonal workers)	h/kg
	Number of working weeks in a year	w
Working conditions	Frequency rate of occupational injuries	%
	Number of new injury cases	-
	Total number of working hours	h
Working conditions	Incidence of occupational injuries	%
	Total number of lost working hours due to occupational injuries	h
	Total number of working hours	h
Local community	Contribution to local employment	%
	Number of new local employees	-
	Total new employees hired	-
Transparency	Sustainability certifications and labels	-
	Total number of certifications and labels (directly linked to the DAT)	-

Table 8: Main social sustainability indicators of the assessment framework.

These indicators and the information that is needed to determine them are modelled in the QCSM by adding new concepts like labour, education, training, injuries with various new properties.

A specific part of the QCSM is depicted in Figure 8 that models the *s4agri-new:FarmWorker* that works at a *s4agri:Farm*, executes various *ploutos:Operations* and can get hurt by some *s4agri-new:Injuries*.



D3.1 Data Requirements and Interoperability – initial version

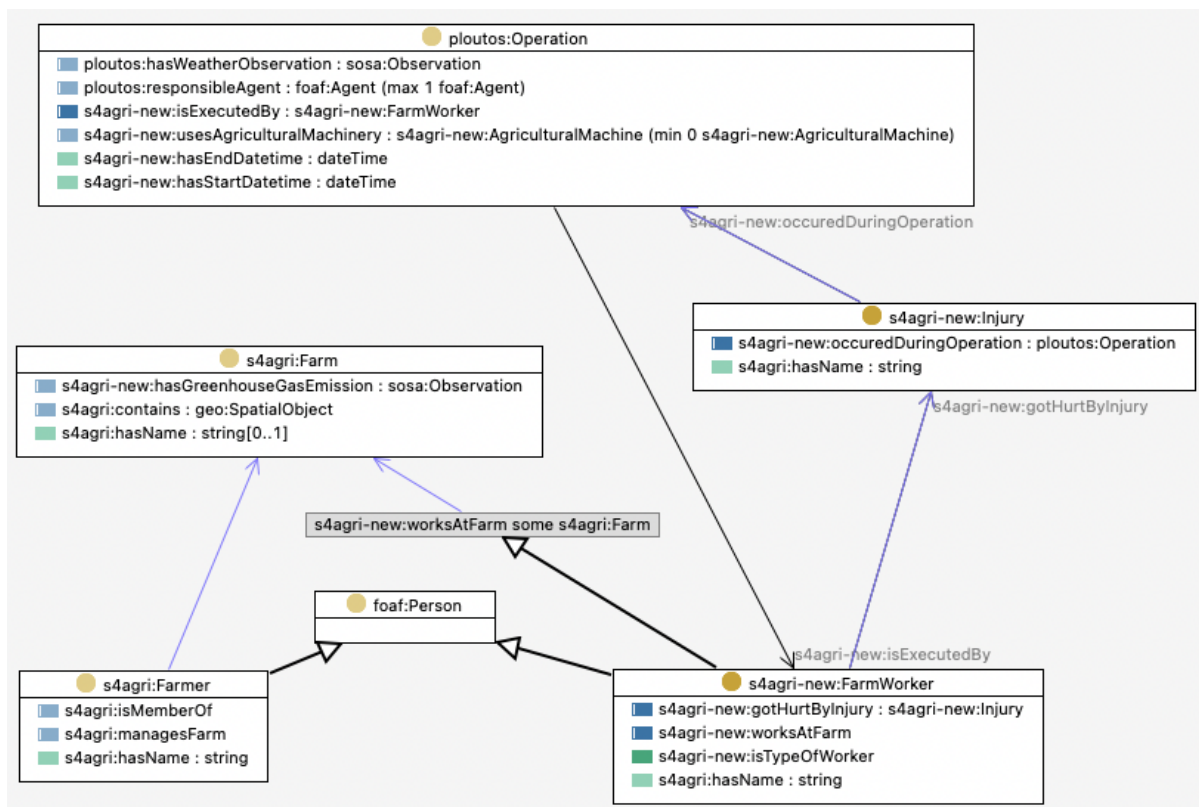


Figure 8 QCSM part of the FarmWorker and possible injuries.

4.4 Modelling behavioural intentions

Besides DAT parameters and environmental, economic and social sustainability indicators, also behavioural intentions influence the consideration, uptake and usage of DATs. These behavioural intentions and the factors that determine them are being studied by work package 1 of the QuantiFarm project and are currently being described in D1.1. The infographic that is made for an overview of these factors is presented in Figure 9.

When studying this infographic, it can be seen that some of the factors can be determined objectively, such as age, gender, size of business or type of farm. However, other factors are more subjective and denote the opinion of the DAT user, e.g., ease of use, prediction of risks or difficulty of learning. Nevertheless, both types of factors are important to measure during the process of encountering, consideration, implementation and usage of a DAT.



D3.1 Data Requirements and Interoperability – initial version

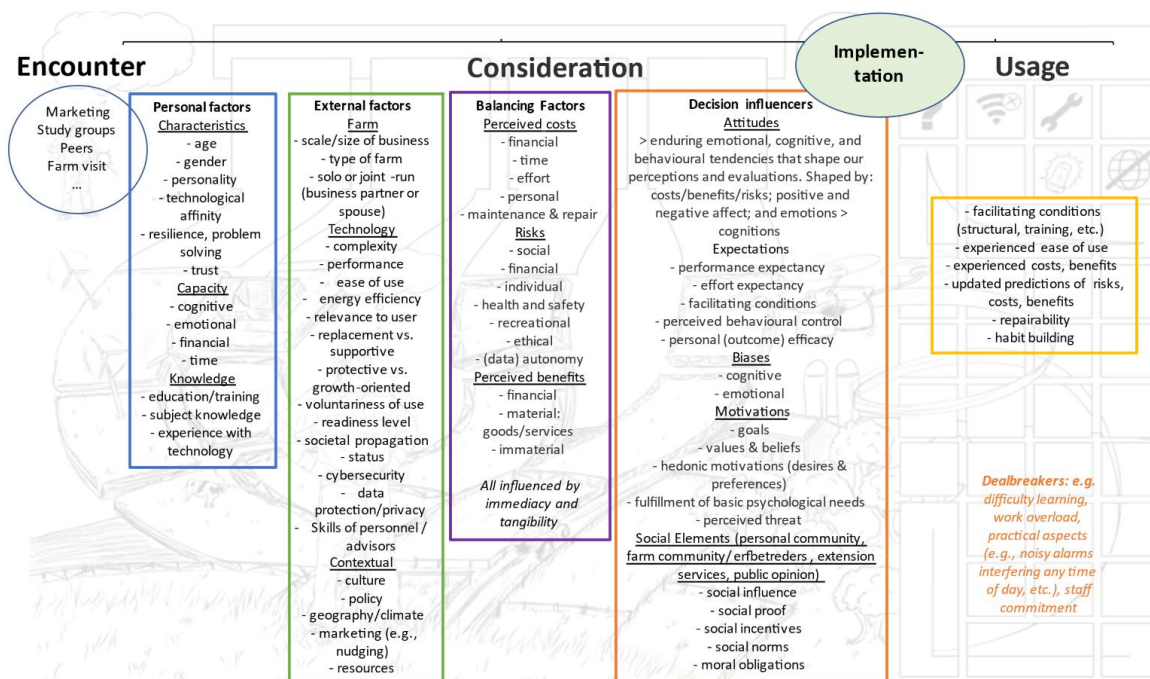


Figure 9 Infographic of the behavioural intentions factors.

With respect to adding these parameters to the QCSM, a subset is already present or can easily be added to existing concepts. The Personal Factors can be added as properties to the Farmer concept, the External Factors of the farm can be added as properties to the Farm concept and technology parameters as properties to the DAT concept. Most of the Balancing factors about costs, risks and benefits have overlap with the economic sustainability indicators and can be added to these parts of the QCSM. New categories of parameters for the QCSM are Contextual, Attitudes, Expectations, Biases, Motivations, Social elements and Usage parameters. Together with work package 1 these parameters will be further detailed in the second half year of 2023 to be added to the QCSM.



5. QuantiFarm Data platform design

5.1 Use cases for QuantiFarm Data platform

A crucial step for the design of a software system - the QuantiFarm data platform in our case- is the specification of the “Use Cases²” in Unified Modeling Language (UML) format. UML is a standardised modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualising, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems. A UML use-case diagram models the behaviour of a system and helps to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system.

To this end, Figure 10 illustrates the external entities expected to interact with the QuantiFarm data platform and most important interactions.

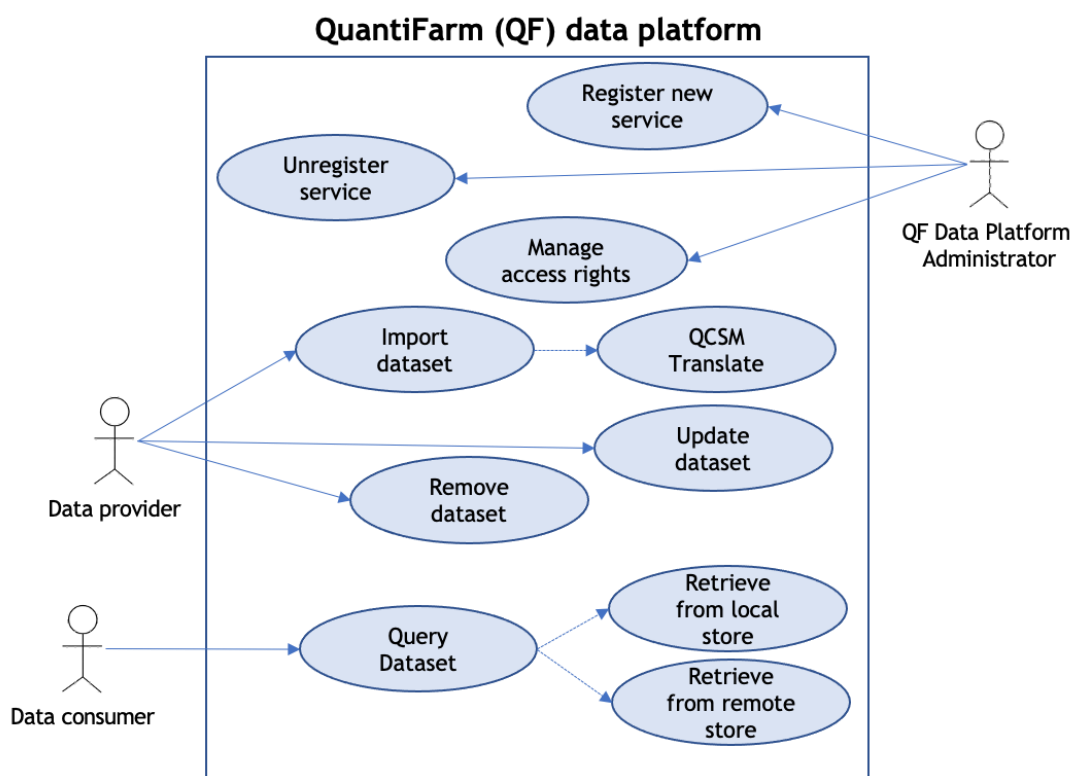


Figure 10 Use case diagram for QuantiFarm data platform

At this stage three main actors of the QuantiFarm data platform are identified:

- QuantiFarm data platform administrator: This actor represents an entity (actual person or a service) allowed to register a new data provider, remove an existing one and to update/configure the access rights of a registered data provider.

² “IBM Rational Software Architect documentation” <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>



D3.1 Data Requirements and Interoperability – initial version

- Data provider: This actor represents data providers within the data platform. A data provider can be a registered service that utilises QF data platforms APIs for storing data within the platform. As it is illustrated in Figure x it is expected that a data provider will be able to import, update and remove datasets according to the access rights that is assigned with. During dataset import a data translation process needs to be realised to ensure data harmonisation with the use of the QSCM.
- Data consumer: This actor represents entities that have the right to perform queries to the database and retrieve selected dataset. According to the issued request parameters the response can be routed either to the local data store or to remote data repositories.

5.2 QuantiFarm Data platform functional architecture

Figure 11 illustrates the QuantiFarm data platform in a diagram. The individual components are described below, but broadly speaking, we rely on a *central data storage* where most of the relevant data for the tools is stored. *Mappers* are components that can retrieve specific data from external APIs that are either too big to be loaded into the central data storage entirely, or have the requirement to be available in real-time. The *tool services* (corresponding to the tools defined in the project proposal) retrieve data from the central data storage and mappers to fulfil their data needs. The *user interface* gives a human friendly interface with all tool functionality in a single consistent interface.

Design principles:

- Large datasets—such as granular weather data—are not stored in our triple store, as it would be impractical to synchronise them.
- External datasets that need to be available for the tools in realtime are not stored in our triple store, but disclosed via a web service that translates data from the external data format to the QuantiFarm Common Semantic Model (QCSM).
- When data is ingested into the QuantiFarm Data Platform, it should be mapped into the QCSM at the moment of ingestion.



D3.1 Data Requirements and Interoperability – initial version

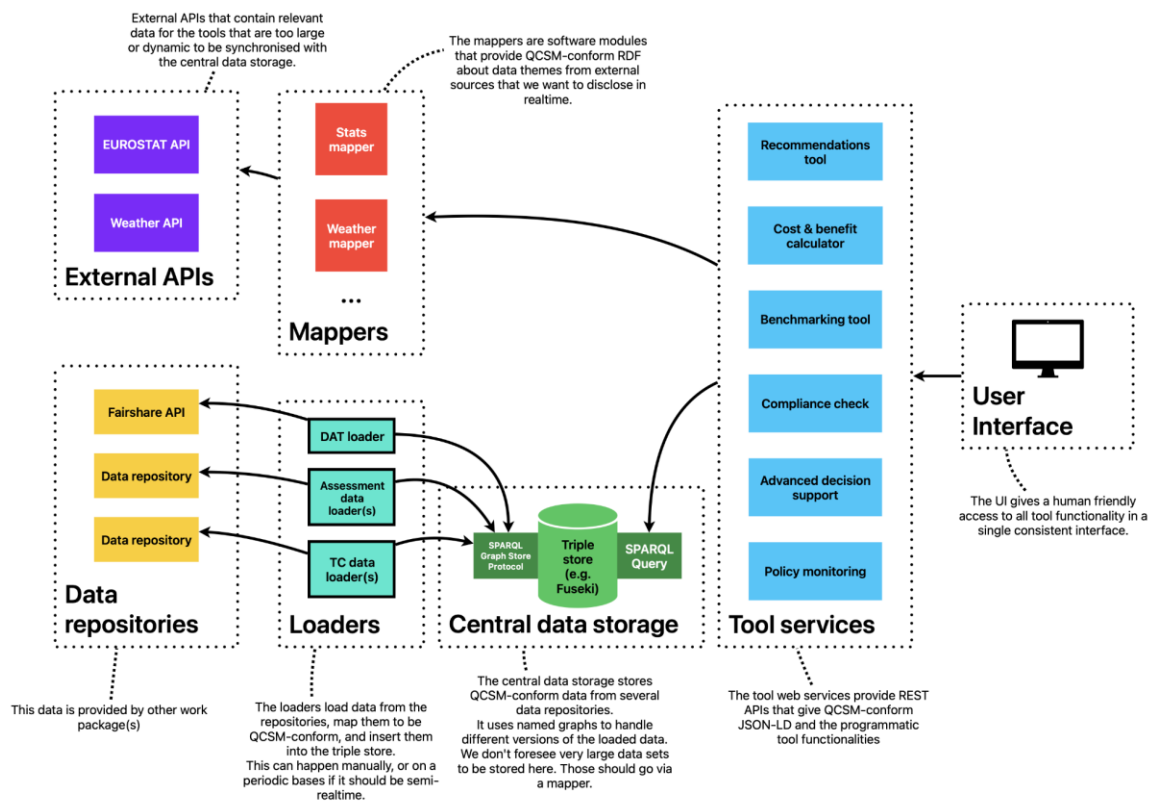


Figure 11 Functional architecture of QuantiFarm data platform.

Central data storage

The *central data storage* (green components in the bottom middle of the diagram) is responsible for storing RDF data that need to be available for the tool services. All data that is ingested into the data storage should be conformant with the ontologies in the QCSM.

For the tool services to retrieve the necessary data, SPARQL queries can be used, which is a flexible and standardised query language for RDF data.

In the current prototype, the [Apache 2.0-licensed](#) triple store [Apache Jena Fuseki](#) is used. It provides a multitude of ways to load and retrieve data from the triple store, using standardised protocols such as SPARQL.

Loaders

The *loaders* (cyan components in the bottom of the diagram) load data from several data sources in an *extract, transform load (ETL)* manner. The data concerned does not involve real-time use cases, so it can be loaded into the data storage whenever it is deemed necessary.

The task of a loader is to access its corresponding data repository, map it into QCSM-conform RDF, and load it into the central data storage. In our data platform, this concerns the repository of DATs in FAIRshare, KPIs of farms that participate in test cases (and which DATs they use, if any), among others.

An important requirement is that the loader can be executed again after a while (e.g., when the data changed, or when the loader is improved), and then it should replace its previous data. This is



D3.1 Data Requirements and Interoperability – initial version

implemented with the SPARQL Graph Store HTTP protocol, specifically with the HTTP PUT method³ on a named graph for the loader.

To produce RDF, the well known rdflib library in Python is used. It provides utilities for generating statements in a graph, and serialising the graph into formats that the central data storage understands.

Mappers

Some external data sources are simply too big to be loaded into the central data storage, or they need to be made available in a real-time fashion instead of ETL. For these data sources, we use special *mappers* which are software modules that can retrieve specific data from these sources, and map them to QCSM-conform RDF for the tool services to use.

These mappers were originally envisioned to be separate microservices, but it is more practical to simply implement them as reusable code modules that the tool services can import.

Data repositories

There are several *data repositories* that we want to leverage in the data sharing platform. In general, these are not available in a homogeneous way, and should thus be transformed into QCSM-conform RDF before they can be conveniently consumed by the tools.

The FAIRshare project has produced a list of DATS (Digital Advisory Tools and Services) which roughly translate into what we call DATs (Digital Technologies in Agriculture) in QuantiFarm . These are enriched with relevant metadata about the technology so that it can be transformed into RDF statements about the technology.

WP2 produced KPIs and metadata about farms participating in the test cases in the form of Excel sheets. The data from these sheets are loaded and transformed by the loaders using the pandas tool in Python. Subsequently, they are loaded into the central data storage.

External APIs

We envision several *external APIs* to be consumed directly as they are needed, and so bypass the central data storage because they're either too big to be loaded entirely, or are needed in real-time. Examples include (historical) weather APIs, CAP indicators and other [EU level data stores](#).

User Interface

The *user interface* is the frontend that the user interacts with. There will be a single consistent user interface that gives personalised access to all the tools using a single thematic style. We have a non-functional prototype, but other than that, this is planned work, and more will be written about it in planned deliverables (D3.6).

The user interface uses the tool services to populate its views with the content and functionality that are required.

Tool services

Each tool is implemented as an independent *tool service*, which comprises one or more web services with well defined APIs. With these APIs, the user interface can retrieve the data that it needs to present the content and provide the functionality.

³ <https://www.w3.org/TR/sparql11-http-rdf-update/#http-put>



D3.1 Data Requirements and Interoperability – initial version

5.3 Implementation

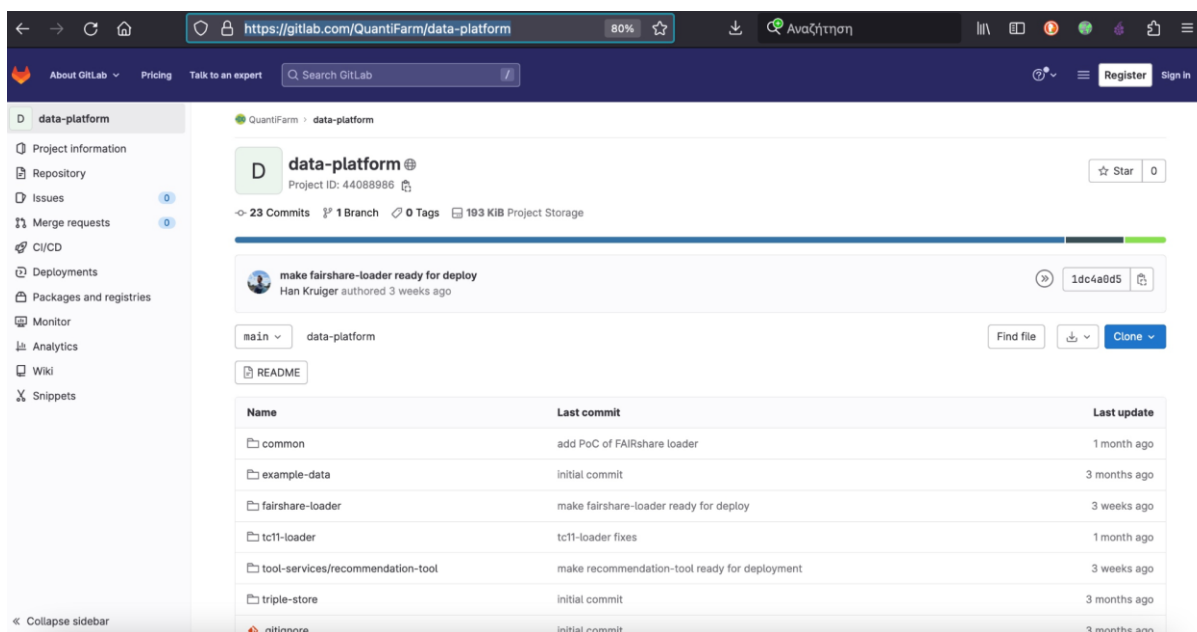
The first prototype of QuantiFarm’s data platform is already implemented and an end-to-end working example of data collection from an external source, data harmonisation with the use of QCSM, data storage to Database and data retrieval through the webservice is demonstrated.

The implementation process and code development follows open source principles even from an early stage and hence a public gitlab platform is available for code development in a collaborative manner.

<https://gitlab.com/QuantiFarm/data-platform>

Directions on how to deploy the software components and to initiate the working examples are available in the README file (<https://gitlab.com/QuantiFarm/data-platform/-/blob/main/README.md>) and also available in the Annex of this Deliverable.

The source code of the data sharing prototype is available as a Docker Compose project.



The screenshot shows the GitLab web interface for the 'data-platform' project. The browser address bar displays 'https://gitlab.com/QuantiFarm/data-platform'. The project page includes a sidebar with navigation options like 'Project information', 'Repository', 'Issues', 'Merge requests', 'CI/CD', 'Deployments', 'Packages and registries', 'Monitor', 'Analytics', 'Wiki', and 'Snippets'. The main content area shows the project name 'data-platform', its ID '44088986', and statistics: '23 Commits', '1 Branch', '0 Tags', and '193 KIB Project Storage'. A recent commit by 'Han Kruijer' is highlighted with the message 'make fairshare-loader ready for deploy'. Below this, there are buttons for 'Find file', 'Clone', and 'README'. A table lists the project's directory structure and the last commit for each:

Name	Last commit	Last update
common	add PoC of FAIRshare loader	1 month ago
example-data	initial commit	3 months ago
fairshare-loader	make fairshare-loader ready for deploy	3 weeks ago
tc11-loader	tc11-loader fixes	1 month ago
tool-services/recommendation-tool	make recommendation-tool ready for deployment	3 weeks ago
triple-store	initial commit	3 months ago
.gitignore	initial commit	3 months ago

Figure 12 Screenshot of QuantiFarm’s gitlab project.



6 Conclusions

In this deliverable, a first version of the QuantiFarm Common Semantic Model (QCSM) and the QuantiFarm Data Platform are described. This model and data platform is being used to collect data from various sources that is needed for the QuantiFarm toolkit for farmers, advisors and policy makers. The collected data is stored in a central datastore and made accessible in terms of the QCSM. The data requirements for the model and platform are collected from various angles, such as the QuantiFarm assessment framework from work package 2 that is used to assess the 30 QuantiFarm test cases and the behavioural intentions framework from work package 1 that describes motivations for using DATs. In addition, existing sources of DAT information, such as FairShare and SmartAKIS, are being reused.

As the test cases, the assessment framework and behavioural framework are still in progress and might change over time, also the QCSM and data platform should be able to adapt to that. Therefore, the design and implementation will further develop during the remainder of the QuantiFarm project.



References

- [1] Brickley, Dan; [Guha, Ramanathan V.](#); Layman, Andrew, eds. (1998-04-09). "[Resource Description Framework \(RDF\) Schemas](#)". W3C. W3C Working Draft. RDF Schema Working Group. Retrieved 2021-04-23.
- [2] Bikakis N.; Tsinaraki C.; Gioldasis N.; Stavrakantonakis I.; Christodoulakis S., eds. (2012-03-21). "[XML and Semantic Web W3C Standards Timeline-History](#)" (PDF). Archived from [the original](#) (PDF) on 2013-04-24. Retrieved 2021-04-23.
- [3] Brickley, Dan; [Guha, Ramanathan V.](#), eds. (2014-02-25). "[RDF Schema 1.1](#)". W3C. 1.1. RDF Working Group. Retrieved 2021-04-23.
- [4] Knublauch, Holger; Oberle, Daniel; Tetlow, Phil; Wallace, Evan (9 March 2006). "[A Semantic Web Primer for Object-Oriented Software Developers](#)". [W3C](#). Retrieved 19 November 2017.
- [5] "[W3C Standard Facilitates Data Management and Integration](#)". W3.org. 27 October 2009. Retrieved 15 October 2013.
- [6] S. Rajbhandari and J. Keizer, The AGROVOC Concept Scheme – A Walkthrough, Journal of Integrative Agriculture 11(5) (2012), 694–699. 35 doi:10.1016/S2095-3119(12)60058-6. <https://www.sciencedirect.com/science/article/pii/S2095311912600586>.
- [7] T. Baker, B. Whitehead, R. Musker and J. Keizer, Global Agricultural Concept Space: Lightweight Semantics for Pragmatic Interoperability, npj Science of Food 3(1) (2019), 16. doi:10.1038/s41538-019-0048-6. <https://www.nature.com/articles/s41538-019-0048-6>.
- [8] E. Arnaud, M.-A. Laporte, S. Kim, C. Aubert, S. Leonelli, B. Miro, L. Cooper, P. Jaiswal, G. Kruseman, R. Shrestha, P.L. Buttigieg, C.J. Mungall, J. Pietragalla, A. Agbona, J. Muliro, J. Detras, V. Hualla, A. Rathore, R.R. Das, I. Dieng, G. Bauchet, N. Menda, C. Pommier, F. Shaw, D. Lyon, L. Mwanzia, H. Juarez, E. Bonaiuti, B. Chiputwa, O. Obileye, S. Auzoux, E.D. Yeumo, L.A. Mueller, K. Silverstein, A. Lafargue, E. Antezana, M. Devare and B. King, The Ontologies Community of Practice: A CGIAR Initiative for Big Data 41 in Agrifood Systems, Patterns 1(7) (2020), 100105. doi:10.1016/j.patter.2020.100105. <https://www.sciencedirect.com/science/article/pii/S2666389920301392>.
- [9] EFSA, The Food Classification and Description System FoodEx2 (Revision 2), Technical Report, European Food Standards Agency, 2015. <https://efsa.onlinelibrary.wiley.com/doi/epdf/10.2903/sp.efsa.2015.EN-804>.
- [10] D.M. Dooley, E.J. Griffiths, G.S. Gosal, P.L. Buttigieg, R. Hoehndorf, M.C. Lange, L.M. Schriml, F.S.L. Brinkman and W.W.L. Hsiao, FoodOn: A Harmonized Food Ontology to Increase Global Food Traceability, Quality Control and Data Integration, npj Science of Food 2(1) (2018), 23. doi:10.1038/s41538-018-0032-6.
- [11] L. Magliulo, L. Genovese, V. Peretti and N. Murru, Application of Ontologies to Traceability in the Dairy Supply Chain, Agricultural Sciences 04(05) (2013), 41–45. doi:10.4236/as.2013.45B008.
- [12] J.P.C. Verhoosel and J. Spek, Applying Ontologies in the Dairy Farming Domain for Big Data Analysis, in: Proceedings of the Joint Proceedings of the 3rd Stream Reasoning (SR 2016) and the 1st Semantic Web Technologies for the Internet of Thing (SWIT 2016) Workshops (SR+SWIT), Kobe, Japan, 2016, pp. 91–100. <http://ceur-ws.org/Vol-1783/paper-09.pdf>.
- [13] J. Verhoosel, M. Van Bekkum and T. Verwaart, Semantic Interoperability for Data Analysis in the Food Supply Chain, International Journal on Food System Dynamics Vol 9 (2018), 101–111 Pages. doi:10.18461/IJFSD.V9I1.917. <http://centmapress.ilb.uni-bonn.de/ojs/index.php/fsd/article/view/917>



D3.1 Data Requirements and Interoperability – initial version

- [14] D. Project, D2.3 Common Data Models and Semantic Interoperability Mechanisms - Release 2, Deliverable, Deliverable D2.3, Demeter Project, 2021.
- [15] F.K. van Evert, S. Fountas, D. Jakovetic, V. Crnojevic, I. Travlos and C. Kempenaar, Big Data for Weed Control and Crop Protection, *Weed Research* 57(4) (2017), 218–233. doi:10.1111/wre.12255.
- [16] J. Verhoosel, M. van Bekkum and T. Verwaart, Semantic Interoperability for Data Analysis in the Food Supply Chain, *International Journal on Food System Dynamics* 9(1) (2018), 101–111. doi:10.18461/ijfsd.v9i1.917. <http://centmapress.ilb.uni-bonn.de/ojs/index.php/fsd/article/view/917>.
- [17] M. Solanki and C. Brewster, Enhancing Visibility in EPCIS Governing Agri-Food Supply Chains via Linked Pedigrees:, *International Journal on Semantic Web and Information Systems* 10(3) (2014), 45–73. doi:10.4018/IJSWIS.2014070102. http://cbrewster.com/papers/Solanki_IJSWIS15.pdf.
- [18] M. Solanki and C. Brewster, OntoPedigree: Modelling Pedigrees for Traceability in Supply Chains, *Semantic Web* 7(5) (2016), 483–491. doi:10.3233/SW-150179. <http://www.semantic-web-journal.net/system/files/swj980.pdf>.
- [19] A.Y. Halevy, A. Rajaraman and J.J. Ordille, Data Integration: The Teenage Years, in: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, September 12-15, 2006, U. Dayal, K.-Y. Whang, D.B. Lomet, G. Alonso, G.M. Lohman, M.L. Kersten, S.K. Cha and Y.-K. Kim, eds, ACM, 2006, pp. 9–16. <http://dl.acm.org/citation.cfm?id=1164130>.
- [20] I. Horrocks, M. Giese, E. Kharlamov and A. Waaler, Using Semantic Technology to Tame the Data Variety Challenge, *IEEE Internet Computing* 20(6) (2016), 62–66. doi:10.1109/MIC.2016.121. <http://ieeexplore.ieee.org/document/7781544/>.
- [21] M. Klusch, P. Kapahnke, S. Schulte, F. Lecue and A. Bernstein, Semantic Web Service Search: A Brief Survey, *KI - Künstliche Intelligenz* 30(2) (2016), 139–147. doi:10.1007/s13218-015-0415-7.
- [22] A.L. Lemos, F. Daniel and B. Benatallah, Web Service Composition: A Survey of Techniques and Tools, *ACM Computing Surveys* 48(3) 20 (2015), 33:1–33:41. doi:10.1145/2831270.
- [23] J. Domingue, D. Roman and M. Stollberg, Web Service Modeling Ontology (WSMO): An Ontology for Semantic Web Services, in: *W3C Workshop on Frameworks for Semantics in Web Services*, Innsbruck, Austria, 2005. https://www.w3.org/2005/04/FSWS/Submissions/1/wsmo_position_paper.html.
- [24] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan and K. Sycara, Bringing Semantics to Web Services: The OWL-S Approach, in: *Semantic Web Services and Web Process Composition*, J. Cardoso and A. Sheth, eds, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2005, pp. 26–42. ISBN 978-3-540-30581-1.
- [25] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal and K. Sivashanmugam, WSDL-S: Adding Semantics to WSDL, *Kno.e.sis Publications* (2005). <https://corescholar.libraries.wright.edu/knoesis/199>.
- [26] S.A. McIlraith and T.C. Son, Adapting Golog for Composition of Semantic Web Services, in: *Proceedings of the Eight International Conference on Principles of Knowledge Representation and Reasoning, KR'02*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 482–496. ISBN 978-1-55860-554-1.



D3.1 Data Requirements and Interoperability – initial version

- [27] D. Redavid, L. Iannone, T.R. Payne and G. Semeraro, OWL-S Atomic Services Composition with SWRL Rules, in: Foundations of Intelligent Systems, 17th International Symposium, ISMIS 2008, Toronto, Canada, May 20-23, 2008, Proceedings, A. An, S. Matwin, Z.W. Ras and D. Slezak, eds, Lecture Notes in Computer Science, Vol. 4994, Springer, 2008, pp. 605–611. <https://eprints.soton.ac.uk/265132/1/ISMIS08.pdf>.
- [28] E. Sirin and B. Parsia, Planning for Semantic Web Services, in: Proceedings of the ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications, SWS@ISWC 2004, Hiroshima, Japan, November 8, 2004, D. Martin, R. Lara and T. Yamaguchi, eds, CEUR Workshop Proceedings, Vol. 119, CEUR-WS.org, 2004. <http://ceur-ws.org/Vol-119/paper1.pdf>.
- [29] D. Androćec, M. Novak and D. Oreški, Using Semantic Web for Internet of Things Interoperability: A Systematic Review, International Journal on Semantic Web and Information Systems (IJSWIS) 14(4) (2018), 147–171. doi:10.4018/IJSWIS.2018100108. <https://www.igi-global.com/article/using-semantic-web-for-internet-of-things-interoperability/210657>.
- [30] M. Noura, M. Atiquzzaman and M. Gaedke, Interoperability in Internet of Things: Taxonomies and Open Challenges, Mobile Networks and Applications 24(3) (2019), 796–809. doi:10.1007/s11036-018-1089-9.
- [31] M. Bauer, E. Kovacs, A. Schülke, N. Ito, C. Criminisi, L.-W. Goix and M. Valla, The Context API in the OMA Next Generation Service Interface, in: 2010 14th International Conference on Intelligence in Next Generation Networks, 2010, pp. 1–5. doi:10.1109/ICIN.2010.5640931.
- [32] C. Brewster, N. Kalatzis, B. Nouwt, H. Kruijer, J. Verhoosel, Data sharing in agricultural supply chains: Using semantics to enable sustainable food systems, Semantic Web Journal, IOS Press, 2023, DOI: 10.3233/SW-233287



Annex A

This annex presents the QuantiFarm Common Semantic Model and the Saref4Agri and Ploutos Common Semantic Model ontologies that it reuses and extends.

```
# baseURI: https://quantifarm.eu/ontology/
# imports: http://purl.obolibrary.org/obo/envo.owl
# imports: http://purl.obolibrary.org/obo/foodon.owl
# imports: http://www.ontology-of-units-of-measure.org/resource/om-2/
# imports: http://www.w3.org/2002/07/owl
# imports: http://www.w3.org/2006/time
# imports: http://www.w3.org/ns/org.ttl
# imports: http://www.w3.org/ns/sosa/
# imports: http://www.w3.org/ns/ssn/
# imports: https://bimerr.iot.linkeddata.es/def/weather
# imports: https://gs1.org/voc/
# prefix: qcsm

@prefix : <https://quantifarm.eu/ontology/> .
@prefix MathM: <http://www.w3.org/1998/Math/MathML> .
@prefix cybele: <https://w3id.org/cybele/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix envo: <http://purl.obolibrary.org/obo/envo.owl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix foodie: <http://foodie-cloud.com/model/foodie#> .
@prefix foodon: <http://purl.obolibrary.org/obo/foodon.owl#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix gs1: <https://gs1.org/voc/> .
@prefix gs1-new: <https://gs1.org/voc-new/> .
@prefix obo: <http://purl.obolibrary.org/obo/> .
@prefix om: <http://www.ontology-of-units-of-measure.org/resource/om-2/> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix ploutos: <https://www.tno.nl/agrifood/ontology/ploutos/common#> .
@prefix ploutosData: <https://www.tno.nl/agrifood/data/ploutos/common#> .
@prefix qcsm: <https://quantifarm.eu/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix s4a: <https://saref.etsi.org/saref4agri/v1.1.2/> .
@prefix s4agri: <https://saref.etsi.org/saref4agri/> .
@prefix s4agri-new: <https://saref.etsi.org/saref4agri-new/> .
@prefix saref: <https://saref.etsi.org/core/> .
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix spin: <http://spinrdf.org/spin#> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix sw: <http://www.w3.org/2003/06/sw-vocab-status/ns#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix weather: <https://bimerr.iot.linkeddata.es/def/weather#> .
@prefix weather-new: <https://bimerr.iot.linkeddata.es/def/weather-new#> .
@prefix wgs84: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<http://dbpedia.org/resource/DBM>
```

```
rdf:type owl:NamedIndividual ;
rdf:type saref:UnitOfMeasure ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:label "decibel-milliwatts"@en ;
.
foodie:Intervention
  rdf:type owl:Class ;
  owl:equivalentClass ploutos:SoilManagementOperation ;
.
foodie:Plot
  rdf:type owl:Class ;
.
obo:ENVO_00000078
  rdf:type owl:Class ;
.
obo:ENVO_00001998
  rdf:type owl:Class ;
.
obo:ENVO_01000267
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty weather-new:hasWeatherProperty ;
    owl:someValuesFrom weather:WeatherProperty ;
  ] ;
.
obo:FOODON_03411266
  rdf:type owl:Class ;
  rdfs:label "FOODON 03411266" ;
  rdfs:subClassOf obo:NCBITaxon_89151 ;
.
obo:FOODON_03411439
  rdf:type owl:Class ;
  rdfs:label "FOODON 03411439" ;
  rdfs:subClassOf obo:NCBITaxon_89151 ;
.
obo:FOODON_03412072
  rdf:type owl:Class ;
  rdfs:label "FOODON 03412072" ;
  rdfs:subClassOf obo:NCBITaxon_3716 ;
.
obo:FOODON_03413355
  rdf:type owl:Class ;
  rdfs:label "FOODON 03413355" ;
  rdfs:subClassOf obo:NCBITaxon_3716 ;
.
obo:PATO_0000033
  rdfs:subClassOf sosa:ObservableProperty ;
.
obo:TAXRANK_0000000
  rdf:type owl:Class ;
  rdfs:label "Taxon rank"@en ;
.
obo:TAXRANK_1000000
  rdf:type owl:ObjectProperty ;
  rdfs:label "has taxon rank"@en ;
.
dcterms:contributor
  rdf:type owl:AnnotationProperty ;
.
dcterms:created
  rdf:type owl:AnnotationProperty ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
dcterms:creator
  rdf:type owl:AnnotationProperty ;
.
dcterms:description
  rdf:type owl:AnnotationProperty ;
.
dcterms:issued
  rdf:type owl:AnnotationProperty ;
.
dcterms:license
  rdf:type owl:AnnotationProperty ;
.
dcterms:modified
  rdf:type owl:AnnotationProperty ;
.
dcterms:publisher
  rdf:type owl:AnnotationProperty ;
.
dcterms:source
  rdf:type owl:AnnotationProperty ;
.
dcterms:title
  rdf:type owl:AnnotationProperty ;
.
vann:preferredNamespacePrefix
  rdf:type owl:AnnotationProperty ;
.
vann:preferredNamespaceUri
  rdf:type owl:AnnotationProperty ;
.
om:degreeCelsius
  rdf:type owl:NamedIndividual ;
  rdf:type saref:UnitOfMeasure ;
  rdfs:label "degree Celsius"@en ;
.
om:millibar
  rdf:type owl:NamedIndividual ;
  rdf:type saref:UnitOfMeasure ;
  rdfs:label "millibar"@en ;
.
om:millimetre
  rdf:type owl:NamedIndividual ;
  rdf:type saref:UnitOfMeasure ;
  rdfs:label "millimetre"@en ;
.
om:millivolt
  rdf:type owl:NamedIndividual ;
  rdf:type saref:UnitOfMeasure ;
  rdfs:label "millivolt"@en ;
.
geo:Feature
  rdf:type owl:Class ;
  rdfs:comment "This class represents the top-level feature type. This class is equivalent to GFI_Feature defined
in ISO 19156:2011, and it is superclass of all feature types."@en ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/geosparql> ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/gml> ;
  rdfs:label "Feature"@en ;
  rdfs:subClassOf geo:SpatialObject ;
  rdfs:subClassOf [
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Restriction ;
    owl:allValuesFrom geo:Geometry ;
    owl:onProperty geo:hasGeometry ;
  ] ;
owl:disjointWith geo:Geometry ;
.

geo:Geometry
  rdf:type owl:Class ;
  rdfs:comment "The class represents the top-level geometry type. This class is equivalent to the UML class GM_Object defined in ISO 19107, and it is superclass of all geometry types."@en ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/geosparql> ;
  rdfs:label "Geometry"@en ;
  rdfs:subClassOf geo:SpatialObject ;
.

geo:SpatialObject
  rdf:type owl:Class ;
  rdfs:comment "The class spatial-object represents everything that can have a spatial representation. It is superclass of feature and geometry."@en ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/geosparql> ;
  rdfs:label "SpatialObject"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom geo:Geometry ;
    owl:onProperty geo:hasGeometry ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom geo:SpatialObject ;
    owl:onProperty geo:sfContains ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom geo:SpatialObject ;
    owl:onProperty geo:sfWithin ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom wgs84:Point ;
    owl:onProperty wgs84:location ;
  ] ;
.

geo:hasGeometry
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A spatial representation for a given feature."@en ;
  rdfs:domain geo:Feature ;
  rdfs:isDefinedBy geo: ;
  rdfs:label "has geometry"@en ;
  rdfs:range geo:Geometry ;
.

geo:sfContains
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Exists if the subject SpatialObject spatially contains the object SpatialObject. DE-9IM: T*****FF**"@en ;
  rdfs:domain geo:SpatialObject ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/geosparql> ;
  rdfs:isDefinedBy <http://www.opengis.net/spec/geosparql/1.0> ;
  rdfs:label "contains"@en ;
  rdfs:range geo:SpatialObject ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
geo:sfWithin
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Exists if the subject SpatialObject is spatially within the object SpatialObject. DE-9IM:
T**F**F****"@en ;
  rdfs:domain geo:SpatialObject ;
  rdfs:isDefinedBy <http://www.opengis.net/ont/geosparql> ;
  rdfs:isDefinedBy <http://www.opengis.net/spec/geosparql/1.0> ;
  rdfs:label "within"@en ;
  rdfs:range geo:SpatialObject ;
.
xsd:date
  rdf:type rdfs:Datatype ;
.
xsd:duration
  rdf:type rdfs:Datatype ;
.
wgs84:Point
  rdf:type owl:Class ;
  rdfs:comment ""
Uniquely identified by lat/long/alt. i.e.

spaciallyIntersects(P1, P2) :- lat(P1, LAT), long(P1, LONG), alt(P1, ALT),
  lat(P2, LAT), long(P2, LONG), alt(P2, ALT).

sameThing(P1, P2) :- type(P1, Point), type(P2, Point), spaciallyIntersects(P1, P2).
"" ;
  rdfs:comment "A point, typically described using a coordinate system relative to Earth, such as WGS84."@en ;
  rdfs:isDefinedBy wgs84: ;
  rdfs:label "Point"@en ;
  rdfs:subClassOf geo:Geometry ;
.
wgs84:SpatialThing
  rdf:type owl:Class ;
.
wgs84:alt
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "The WGS84 altitude of a SpatialThing (decimal meters above the local reference ellipsoid)."@en
;
  rdfs:domain wgs84:SpatialThing ;
  rdfs:label "altitude"@en ;
.
wgs84:lat
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "The WGS84 latitude of a SpatialThing (decimal degrees)."@en ;
  rdfs:domain wgs84:SpatialThing ;
  rdfs:label "latitude"@en ;
.
wgs84:location
  rdf:type owl:ObjectProperty ;
  rdfs:comment ""
"The relation between something and the point,
or other geometrical thing in space, where it is. For example, the relationship between
a radio tower and a Point with a given lat and long.
Or a relationship between a park and its outline as a closed arc of points, or a road and
its location as a arc (a sequence of points).
Clearly in practice there will be limit to the accuracy of any such statement, but one would expect
an accuracy appropriate for the size of the object and uses such as mapping .
""@en ;
  rdfs:label "location"@en ;
  rdfs:subPropertyOf foaf:based_near ;
```



D3.1 Data Requirements and Interoperability – initial version

```
.
wgs84:long
  rdf:type owl:ObjectProperty ;
  rdfs:comment "The WGS84 longitude of a SpatialThing (decimal degrees)."@en ;
  rdfs:domain wgs84:SpatialThing ;
  rdfs:label "longitude"@en ;
.
<http://www.w3.org/2004/02/skos/core#definition>
  rdf:type owl:AnnotationProperty ;
.
<http://www.w3.org/2004/02/skos/core#prefLabel>
  rdf:type owl:AnnotationProperty ;
.
time:Instant
  rdf:type owl:Class ;
  rdfs:comment "A temporal entity with zero extent or duration"@en ;
  rdfs:isDefinedBy time: ;
  rdfs:label "Instant"@en ;
  rdfs:subClassOf time:TemporalEntity ;
.
time:Interval
  rdf:type owl:Class ;
  rdfs:comment "A temporal entity with an extent or duration"@en ;
  rdfs:isDefinedBy time: ;
  rdfs:label "Time interval"@en ;
  rdfs:subClassOf time:TemporalEntity ;
.
time:TemporalEntity
  rdf:type owl:Class ;
  rdfs:comment "A temporal interval or instant."@en ;
  rdfs:isDefinedBy time: ;
  rdfs:label "Temporal entity"@en ;
.
<http://www.w3.org/ns/prov#hadPrimarySource>
  rdf:type owl:AnnotationProperty ;
.
sosa:Platform
  rdf:type owl:Class ;
  rdfs:comment "A Platform is an entity that hosts other entities, particularly Sensors, Actuators, Samplers, and
other Platforms (definition taken from http://www.w3.org/ns/sosa/Platform)"@en ;
  rdfs:isDefinedBy sosa: ;
  rdfs:label "Platform"@en ;
.
sosa:hosts
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation between a platform and the systems it hosts."@en ;
  rdfs:isDefinedBy sosa: ;
  rdfs:label "host"@en ;
  owl:inverseOf sosa:isHostedBy ;
.
sosa:isHostedBy
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation between a system and its hosting platform."@en ;
  rdfs:isDefinedBy sosa: ;
  rdfs:label "is hosted by"@en ;
.
ssn:Deployment
  rdf:type owl:Class ;
  rdfs:isDefinedBy ssn: ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:label "Deployment"@en ;
.
ssn:System
  rdf:type owl:Class ;
  rdfs:comment "System is a unit of abstraction for pieces of infrastructure that implement Procedures. A System may have components, its subsystems, which are other Systems. (Definition taken from http://www.w3.org/ns/ssn/System)"@en ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "System"@en ;
.
ssn:deployedOnPlatform
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation between a deployment and the platform(s) involved in the deployment."@en ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "deployed on platform"@en ;
  owl:inverseOf ssn:inDeployment ;
.
ssn:deployedSystem
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation between a deployment and the system deployed."@en ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "deployed system"@en ;
  owl:inverseOf ssn:hasDeployment ;
.
ssn:hasDeployment
  rdf:type owl:ObjectProperty ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "has deployment"@en ;
.
ssn:hasInput
  rdfs:subPropertyOf owl:topObjectProperty ;
.
ssn:hasOutput
  rdfs:subPropertyOf owl:topObjectProperty ;
.
ssn:hasSubSystem
  rdf:type owl:ObjectProperty ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "has subsystem"@en ;
.
ssn:inDeployment
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation between a platform and the deployment(s) that takes place in the platform."@en ;
  rdfs:isDefinedBy ssn: ;
  rdfs:label "in deployment"@en ;
.
foaf:Agent
  rdf:type owl:Class ;
  rdfs:label "Agent"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty foaf:member ;
    owl:someValuesFrom foaf:Agent ;
  ] ;
.
foaf:Person
  rdf:type owl:Class ;
  rdfs:label "Person"@en ;
  rdfs:subClassOf foaf:Agent ;
```



D3.1 Data Requirements and Interoperability – initial version

```
.
foaf:based_near
  rdf:type owl:ObjectProperty ;
.
foaf:member
  rdf:type owl:ObjectProperty ;
  rdfs:label "member"@en ;
.
weather-new:hasWeatherProperty
  rdf:type owl:ObjectProperty ;
  rdfs:comment "This property indicates that the atmosphere can have different weather-related properties, such
as temperature, wind, humidity, etc." ;
  rdfs:domain obo:ENVO_01000267 ;
  rdfs:label "has weather property" ;
  rdfs:range weather:WeatherProperty ;
.
gs1-new:certificateConformityAssessmentOrganization
  rdf:type rdf:Property ;
  rdf:type owl:ObjectProperty ;
  rdfs:comment "The organisation that has assessed the conformity of the object (e.g. product, asset, container),
party or location to the certification standard."@en ;
  rdfs:domain gs1:CertificationDetails ;
  rdfs:isDefinedBy gs1: ;
  rdfs:label "Certificate Conformity Assessment Organization"@en ;
  rdfs:range ploutos:ConformityAssessmentOrganization ;
  sw:term_status "proposed" ;
.
gs1-new:productPurpose
  rdf:type rdf:Property ;
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "1: This property expresses the purpose for which a product that is cultivated is meant, e.g.
industrial processing or private consumption"@en ;
  rdfs:comment "2: For now, this is modelled for simplicity as a string that contains the purpose of the product. In
the future, this can be replaced by a class ProductPurpose with more properties than just the name."@en ;
  rdfs:domain gs1:Product ;
  rdfs:isDefinedBy gs1: ;
  rdfs:label "Product Purpose"@en ;
  rdfs:range rdf:langString ;
  sw:term_status "proposed" ;
.
gs1:AllergenTypeCode
  rdf:type owl:Class ;
  rdfs:label "Allergen type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:AnatomicalFormCode
  rdf:type owl:Class ;
  rdfs:label "Anatomical form code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:CertificationDetails
  rdfs:comment """"gs1-new: The certification details contains a number of properties. See:
https://www.gs1.org/voc/CertificationDetails. We propose a few changes to these properties:
- property gs1:certificationAgency: replace the range of this property from a string to the class of certificate
standard owners. For now, we have added the class ploutos:CertificateStandardOwner to the range.
- property gs1:certificationIdentification: gs1-new: replace the range of this property from a string to the class of
certificate identifiers. For now, we have added the class ploutos:AuthoritySpecificCertificateIdentifier to the
range.
```



D3.1 Data Requirements and Interoperability – initial version

- property `gs1:certificationAgencyURL`: change the range to a string instead of an organization. The `gs1:certificationAgency` needs to have the range of the class of organizations that own a certification standard. This property only needs to point to a string containing the URL of the organization.

- add a property named `gs1:certificateConformityAssessmentOrganization` that contains as range the organization that performed the conformity assessment checks for the certificate, this organization includes the specific certificate auditor/reviewer/issuer roles."@en ;

`rdfs:comment "gs1-new: We propose to rename this concept to gs1:Certificate as this more intuitively represents the class of certifications issues for an organization, place or product."@en ;`

`gs1:CertificationStatus`

`rdf:type owl:Class ;`
`rdfs:label "Certification status" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:CheeseFirmnessCode`

`rdf:type owl:Class ;`
`rdfs:label "Cheese firmness code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:ColourCodeListCode`

`rdf:type owl:Class ;`
`rdfs:label "Colour code list code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:ConsumerLifestageCode`

`rdf:type owl:Class ;`
`rdfs:label "Consumer lifestage code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:ConsumerSalesConditionsCode`

`rdf:type owl:Class ;`
`rdfs:label "Consumer sales conditions code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:DietTypeCode`

`rdf:type owl:Class ;`
`rdfs:label "Diet type code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:DiscountTypeCode`

`rdf:type owl:Class ;`
`rdfs:label "Discount type code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:FoodBeverageRefrigerationClaimCode`

`rdf:type owl:Class ;`
`rdfs:label "Food beverage refrigeration claim code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:FoodBeverageTargetUseCode`

`rdf:type owl:Class ;`
`rdfs:label "Food beverage target use code" ;`
`rdfs:subClassOf gs1:TypeCode ;`

`gs1:FootwearFasteningTypeCode`

`rdf:type owl:Class ;`
`rdfs:label "Footwear fastening type code" ;`
`rdfs:subClassOf gs1:TypeCode ;`



D3.1 Data Requirements and Interoperability – initial version

```
gs1:FreshOrSeawaterFarmedCode
  rdf:type owl:Class ;
  rdfs:label "Fresh or seawater farmed code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:GrowingMethodCode
  rdf:type owl:Class ;
  rdfs:label "Growing method code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:LevelOfContainmentCode
  rdf:type owl:Class ;
  rdfs:label "Level of containment code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:MaturationMethodCode
  rdf:type owl:Class ;
  rdfs:label "Maturation method code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:MeasurementPrecisionCode
  rdf:type owl:Class ;
  rdfs:label "Measurement precision code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:MeasurementType
  rdf:type owl:Class ;
  rdfs:label "Measurement type" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:NonbinaryLogicCode
  rdf:type owl:Class ;
  rdfs:label "Nonbinary logic code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:NutrientBasisQuantityCode
  rdf:type owl:Class ;
  rdfs:label "Nutrient basis quantity code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:NutritionalClaimTypeCode
  rdf:type owl:Class ;
  rdfs:label "Nutritional claim type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:OfferRedemptionTypeCode
  rdf:type owl:Class ;
  rdfs:label "Offer redemption type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:OrganicClaimAgencyCode
  rdf:type owl:Class ;
  rdfs:label "Organic claim agency code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:Organization
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ploutos:hasAuthoritySpecificOrganisationIdentifier ;
```



D3.1 Data Requirements and Interoperability – initial version

```
    owl:someValuesFrom ploutos:AuthoritySpecificOrganisationIdentifier ;
  ] ;
.
gs1:OrganizationRoleType
  rdf:type owl:Class ;
  rdfs:label "Organization role type" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingFeatureCode
  rdf:type owl:Class ;
  rdfs:label "Packaging feature code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingFunctionCode
  rdf:type owl:Class ;
  rdfs:label "Packaging function code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingMarkedDietAllergenCode
  rdf:type owl:Class ;
  rdfs:label "Packaging marked diet allergen code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingMarkedFreeFromCode
  rdf:type owl:Class ;
  rdfs:label "Packaging marked free from code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingMarkedLabelAccreditationCode
  rdf:type owl:Class ;
  rdfs:label "Packaging marked label accreditation code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingMaterialTypeCode
  rdf:type owl:Class ;
  rdfs:label "Packaging material type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingRecyclingProcessTypeCode
  rdf:type owl:Class ;
  rdfs:label "Packaging recycling process type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingRecyclingSchemeCode
  rdf:type owl:Class ;
  rdfs:label "Packaging recycling scheme code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PackagingShapeCode
  rdf:type owl:Class ;
  rdfs:label "Packaging shape code" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PaymentMethod
  rdf:type owl:Class ;
  rdfs:label "Payment method" ;
  rdfs:subClassOf gs1:TypeCode ;
.
gs1:PreparationTypeCode
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Class ;
    rdfs:label "Preparation type code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:PreservationTechniqueCode
    rdf:type owl:Class ;
    rdfs:label "Preservation technique code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:ProductYieldTypeCode
    rdf:type owl:Class ;
    rdfs:label "Product yield type code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:ReferencedFileTypeCode
    rdf:type owl:Class ;
    rdfs:label "Referenced file type code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SeasonParameterCode
    rdf:type owl:Class ;
    rdfs:label "Season parameter code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SecurityMarking
    rdf:type owl:Class ;
    rdfs:label "Security marking" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SensorAlertType
    rdf:type owl:Class ;
    rdfs:label "Sensor alert type" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SharpnessOfCheeseCode
    rdf:type owl:Class ;
    rdfs:label "Sharpness of cheese code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SizeGroupCode
    rdf:type owl:Class ;
    rdfs:label "Size group code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SizeSystemCode
    rdf:type owl:Class ;
    rdfs:label "Size system code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SizeTypeCode
    rdf:type owl:Class ;
    rdfs:label "Size type code" ;
    rdfs:subClassOf gs1:TypeCode ;
.

gs1:SourceAnimalCode
    rdf:type owl:Class ;
    rdfs:label "Source animal code" ;
    rdfs:subClassOf gs1:TypeCode ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
gs1:SportingActivityTypeCode
  rdf:type owl:Class ;
  rdfs:label "Sporting activity type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.

gs1:TargetConsumerGenderCode
  rdf:type owl:Class ;
  rdfs:label "Target consumer gender code" ;
  rdfs:subClassOf gs1:TypeCode ;
.

gs1:TradeChannelCode
  rdf:type owl:Class ;
  rdfs:label "Trade channel code" ;
  rdfs:subClassOf gs1:TypeCode ;
.

gs1:UpperTypeCode
  rdf:type owl:Class ;
  rdfs:label "Upper type code" ;
  rdfs:subClassOf gs1:TypeCode ;
.

gs1:certificationAgency
  rdfs:comment "gs1-new: We propose to replace the range of this property from a string to the class of certificate
standard owners. For now, we have added the class ploutos:CertificateStandardOwner to the range."@en ;
  rdfs:range ploutos:CertificateStandardOwner ;
.

gs1:certificationAgencyURL
  rdfs:comment "gs1-new: We propose to change the range to a string instead of an organization. The
gs1:certificationAgency needs to have the range of the class of organizations that own a certification standard.
This property only needs to point to a string containing the URL of the organization. For now, we added xsd:string
to the range."@en ;
  rdfs:range xsd:string ;
.

gs1:certificationIdentification
  rdfs:comment "gs1-new: We propose to replace the range of this property from a string to the class of certificate
identifiers. For now, we have added the class ploutos:AuthoritySpecificCertificateIdentifier to the range."@en ;
  rdfs:range ploutos:AuthoritySpecificCertificateIdentifier ;
.

gs1:certificationStandard
  rdfs:comment "gs1-new: We assume that this property relates to the specific certification scheme that was used
to check whether the certificate could be issued, e.g., GlobalGAP General Regulations, GlobalGAP Integrated
Farm Assurance Control Points & Compliance Criteria version 5.2 for Crops - Fruit & Vegetables"@en ;
.

gs1:certificationSubject
  rdfs:comment "gs1-new: We assume that this property relates to the object that was the target for certification,
i.e. a system, process or product. In the food chain, this is usually a product for which all the production systems,
processes and procedures have to be checked according to the certification scheme" ;
.

qesm:
  rdf:type owl:Ontology ;
  dcterms:title "QuantiFarm Common Semantic Model" ;
  owl:imports <http://purl.obolibrary.org/obo/envo.owl> ;
  owl:imports <http://purl.obolibrary.org/obo/foodon.owl> ;
  owl:imports om ;
  owl:imports <http://www.w3.org/2002/07/owl> ;
  owl:imports <http://www.w3.org/2006/time> ;
  owl:imports <http://www.w3.org/ns/org.ttl> ;
  owl:imports sosa ;
  owl:imports ssn ;
  owl:imports <https://bimerr.iot.linkeddata.es/def/weather> ;
```



D3.1 Data Requirements and Interoperability – initial version

```
owl:imports gs1 ;
.
qcm:AgriFoodChainActor_1
  rdf:type ploutos:AgriFoodChainActor ;
  rdfs:label "AgriFood chain actor 1" ;
.
qcm:DigitalAgriculturalTechnology
  rdf:type owl:Class ;
  rdfs:comment "A Digital Agricultural Technology (DAT) is a digital technology that is used mainly in agriculture." ;
  rdfs:comment ""We need to extend this with:
- targetAudience
- deliveryType
- benefitsRealised
- challengesTackled"" ;
  rdfs:label "Digital Agricultural Technology" ;
.
qcm:EconomicSustainability
  rdf:type owl:Class ;
  rdfs:label "Economic sustainability" ;
.
qcm:EnvironmentalSustainability
  rdf:type owl:Class ;
  rdfs:label "Environmental sustainability" ;
.
qcm:SocialSustainability
  rdf:type owl:Class ;
  rdfs:label "Social sustainability" ;
.
qcm:TechnologyProvider
  rdf:type owl:Class ;
  rdfs:comment "The class represents all companies that provide technology for agriFood chain actors, especially focused on digital agricultural technologies" ;
  rdfs:label "Technology provider" ;
  rdfs:subClassOf ploutos:AgriFoodChainActor ;
.
qcm:hasDescription
  rdf:type owl:DatatypeProperty ;
  rdfs:domain qcm:DigitalAgriculturalTechnology ;
  rdfs:label "has description" ;
  rdfs:range xsd:string ;
.
qcm:hasFunctionality
  rdf:type owl:DatatypeProperty ;
  rdfs:domain qcm:DigitalAgriculturalTechnology ;
  rdfs:label "has functionality" ;
  rdfs:range xsd:string ;
.
qcm:hasInitialInvestmentCosts
  rdf:type owl:DatatypeProperty ;
  rdfs:domain qcm:DigitalAgriculturalTechnology ;
  rdfs:label "has initial investment costs" ;
  rdfs:range xsd:decimal ;
.
qcm:hasKeyword
  rdf:type owl:DatatypeProperty ;
  rdfs:domain qcm:DigitalAgriculturalTechnology ;
  rdfs:label "has keyword" ;
  rdfs:range xsd:string ;
```



D3.1 Data Requirements and Interoperability – initial version

.
qesm:hasLaunchYear
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has launch year" ;
rdfs:range xsd:date ;

.
qesm:hasMaintenanceCosts
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has maintenance costs" ;
rdfs:range xsd:decimal ;

.
qesm:hasProvider
rdf:type owl:ObjectProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has provider" ;
rdfs:range qesm:TechnologyProvider ;

.
qesm:hasSupportedLanguage
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has supported language" ;
rdfs:range xsd:string ;

.
qesm:hasTitle
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has title" ;
rdfs:range xsd:string ;

.
qesm:hasUpdateYear
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "has update year" ;
rdfs:range xsd:date ;

.
qesm:isAppliedByFarm
rdf:type owl:ObjectProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "is applied by farm" ;
rdfs:range s4agri:Farm ;

.
qesm:isTargetedTowardsCountry
rdf:type owl:ObjectProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "is targeted towards country" ;
rdfs:range gsl:Country ;

.
qesm:isTargetedTowardsAgriculturalSector
rdf:type owl:DatatypeProperty ;
rdfs:domain qesm:DigitalAgriculturalTechnology ;
rdfs:label "is targeted towards agricultural sector" ;
rdfs:range xsd:string ;

.
saref:Actuator
rdf:type owl:Class ;
rdfs:isDefinedBy saref: ;
rdfs:label "Actuator"@en ;



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:subClassOf saref:Device ;
.
saref:Command
  rdf:type owl:Class ;
  rdfs:comment "A directive that a device must support to perform a certain function. A command may act upon
a state, but does not necessarily act upon a state. For example, the ON command acts upon the ON/OFF state, but
the GET command does not act upon any state, it simply gives a directive to retrieve a certain value. We propose
here a list of commands that are relevant for the purpose of SAREF, but this list can be extended."@en ;
  rdfs:label "Command"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:State ;
    owl:onProperty saref:actsUpon ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty saref:hasDescription ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass saref:Function ;
    owl:onProperty saref:isCommandOf ;
  ] ;
.
saref:Device
  rdf:type owl:Class ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "Device"@en ;
  rdfs:subClassOf ssn:System ;
.
saref:FeatureOfInterest
  rdf:type owl:Class ;
  rdfs:comment saref: ;
  rdfs:label "Feature of interest"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Measurement ;
    owl:onProperty saref:hasMeasurement ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Property ;
    owl:onProperty saref:hasProperty ;
  ] ;
.
saref:Function
  rdf:type owl:Class ;
  rdfs:comment "The functionality necessary to accomplish the task for which a Device is designed. A device can
be designed to perform more than one function. Functions can be structured in categories (subclasses) that reflect
different points of view, for example, considering the specific application area for which a function can be used
(e.g., light, temperature, motion, heat, power, etc.), or the capability that a function can support (e.g., receive,
reply, notify, etc.), and so forth."@en ;
  rdfs:label "Function"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass saref:Command ;
```



D3.1 Data Requirements and Interoperability – initial version

```
    owl:onProperty saref:hasCommand ;
  ] ;
.
saref:Humidity
  rdf:type owl:Class ;
  rdfs:comment "A saref:Property related to some measurements that are characterized by a certain value that is
measured in a humidity unit"@en ;
  rdfs:label "Humidity"@en ;
  rdfs:label "Humidiy"@en ;
  rdfs:subClassOf saref:Property ;
.
saref:Measurement
  rdf:type owl:Class ;
  rdfs:comment "Represents the measured value made over a property. It is also linked to the unit of measure in
which the value is expressed and the timestamp of the measurement."@en ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "Measurement"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom xsd:dateTime ;
    owl:onProperty saref:hasTimestamp ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:FeatureOfInterest ;
    owl:onProperty saref:isMeasurementOf ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Property ;
    owl:onProperty saref:relatesToProperty ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:UnitOfMeasure ;
    owl:onProperty saref:isMeasuredIn ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty saref:hasValue ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass saref:Property ;
    owl:onProperty saref:relatesToProperty ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass saref:UnitOfMeasure ;
    owl:onProperty saref:isMeasuredIn ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:float ;
    owl:onProperty saref:hasValue ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
```



D3.1 Data Requirements and Interoperability – initial version

```
];
owl:disjointWith saref:Property ;
owl:disjointWith saref:UnitOfMeasure ;
.
saref:Property
  rdf:type owl:Class ;
  rdfs:comment "An aspect of an entity that can be observable by a sensor."@en ;
  rdfs:comment "Anything that can be sensed, measured or controlled in households, common public buildings or offices. We propose here a list of properties that are relevant for the purpose of SAREF, but this list can be extended."@en ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "Property"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Device ;
    owl:onProperty saref:isControlledByDevice ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Device ;
    owl:onProperty saref:isMeasuredByDevice ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:FeatureOfInterest ;
    owl:onProperty saref:isPropertyOf ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom saref:Measurement ;
    owl:onProperty saref:relatesToMeasurement ;
  ] ;
  owl:disjointWith saref:UnitOfMeasure ;
.
saref:Sensor
  rdf:type owl:Class ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "Sensor"@en ;
  rdfs:subClassOf saref:Device ;
.
saref:State
  rdf:type owl:Class ;
  rdfs:comment "The state in which a device can be found, e.g, ON/OFF/STANDBY, or ONLINE/OFFLINE. We propose here a list of states that are relevant for the purpose of SAREF, but this list can be extended."@en ;
  rdfs:label "State"@en ;
.
saref:Task
  rdf:type owl:Class ;
  rdfs:comment "The goal for which a device is designed (from a user perspective). For example, a washing machine is designed for the task of washing. We propose here a list of tasks that are relevant for the purpose of SAREF, but this list can be extended."@en ;
  rdfs:label "Task"@en ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass saref:Device ;
    owl:onProperty saref:isAccomplishedBy ;
  ] ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
saref:Temperature
  rdf:type owl:Class ;
  rdfs:comment "A saref:Property related to some measurements that are characterized by a certain value that is
measured in a temperature unit (degree_Celsius, degree_Fahrenheit, or degree_kelvin)"@en ;
  rdfs:label "Temperature"@en ;
  rdfs:subClassOf saref:Property ;
.

saref:Time
  rdf:type owl:Class ;
  rdfs:comment "A saref:Property that allows to specify the time concept in terms of instants or intervals according
to the imported W3C Time ontology."@en ;
  rdfs:label "Time"@en ;
  rdfs:subClassOf saref:Property ;
.

saref:UnitOfMeasure
  rdf:type owl:Class ;
  rdfs:comment "The unit of measure is a standard for measurement of a quantity, such as a Property. For example,
Power is a property and Watt is a unit of power that represents a definite predetermined power: when we say 10
Watt, we actually mean 10 times the definite predetermined power called \"watt\". Our definition is based on the
definition of unit of measure in the Ontology of units of Measure (OM). We propose here a list of some units of
measure that are relevant for the purpose of SAREF, but this list can be extended, also using some other ontologies
rather than the Ontology of units of Measure (OM)."@en ;
  rdfs:comment "The unit of measure is a standard for measurement of a quantity, such as a Property. For example,
Power is a property and Watt is a unit of power that represents a definite predetermined power: when we say 10
Watt, we actually mean 10 times the definite predetermined power called \"watt\". Our definition is based on the
definition of unit of measure in the Ontology of units of Measure (OM). We propose here a list of some units of
measure that are relevant for the purpose of the Smart Appliances ontology, but this list can be extended."@en ;
  rdfs:isDefinedBy <http://www.wurvoc.org/vocabularies/om-1.6/Unit_of_measure> ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "Unit of measure"@en ;
.

saref:accomplishes
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relationship between a certain entity (e.g., a device) and the task it accomplishes"@en ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "accomplishes"@en ;
  owl:inverseOf saref:isAccomplishedBy ;
.

saref:actsUpon
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relationship between a command and a state"@en ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "acts upon"@en ;
.

saref:controlsProperty
  rdf:type owl:ObjectProperty ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "controls property"@en ;
.

saref:hasCommand
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relationship between an entity (such as a function) and a command"@en ;
  rdfs:isDefinedBy saref: ;
  rdfs:label "has command"@en ;
  owl:inverseOf saref:isCommandOf ;
.

saref:hasDescription
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "A relationship providing a description of an entity (e.g., device)"@en ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:isDefinedBy saref: ;
rdfs:label "has description"@en ;
rdfs:range xsd:string ;
.
saref:hasFunction
rdf:type owl:ObjectProperty ;
rdfs:comment "A relationship identifying the type of function of a device"@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "has function"@en ;
.
saref:hasManufacturer
rdf:type owl:DatatypeProperty ;
rdfs:comment "A relationship identifying the manufacturer of an entity (e.g., device)"@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "has manufacturer"@en ;
rdfs:range xsd:string ;
.
saref:hasMeasurement
rdf:type owl:ObjectProperty ;
rdfs:comment "Relation between a feature of interest and the measurments that are about it."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "is feature of interest of"@en ;
.
saref:hasModel
rdf:type owl:DatatypeProperty ;
rdfs:comment "A relationship identifying the model of an entity (e.g., device)"@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "has model"@en ;
rdfs:range xsd:string ;
.
saref:hasProperty
rdf:type owl:ObjectProperty ;
rdfs:comment "Relation between a feature of interest and its properties."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "has property"@en ;
owl:inverseOf saref:isPropertyOf ;
.
saref:hasState
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "has state"@en ;
.
saref:hasTimestamp
rdf:type owl:DatatypeProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "has timestamp"@en ;
.
saref:hasValue
rdf:type owl:DatatypeProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "has value"@en ;
.
saref:isAccomplishedBy
rdf:type owl:ObjectProperty ;
rdfs:comment "A relationship indentifying the task accomplished by a certain entity (e.g., a device)"@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "is accomplished by"@en ;
.
saref:isCommandOf
```



D3.1 Data Requirements and Interoperability – initial version

```

rdf:type owl:ObjectProperty ;
rdfs:comment "A relationship between a command and a function."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "is command of"@en ;
.
saref:isControlledByDevice
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "is controlled by device"@en ;
.
saref:isMeasuredByDevice
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "is measured by device"@en ;
.
saref:isMeasuredIn
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "is measured in"@en ;
.
saref:isMeasurementOf
rdf:type owl:ObjectProperty ;
rdfs:comment "Relation between a measurement and the feature of interest being measured."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "has feature of interest"@en ;
owl:inverseOf saref:hasMeasurement ;
.
saref:isPropertyOf
rdf:type owl:ObjectProperty ;
rdfs:comment "Relation between a property and a feature of interest in which such property appears."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "is property of"@en ;
.
saref:makesMeasurement
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "makes measurement"@en ;
owl:inverseOf saref:measurementMadeBy ;
.
saref:measurementMadeBy
rdf:type owl:ObjectProperty ;
rdfs:comment "Relation between a measurement and the device that made it."@en ;
rdfs:isDefinedBy saref: ;
rdfs:label "measurement made by"@en ;
.
saref:measuresProperty
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "measures property"@en ;
.
saref:relatesToMeasurement
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "relates to measurement"@en ;
.
saref:relatesToProperty
rdf:type owl:ObjectProperty ;
rdfs:isDefinedBy saref: ;
rdfs:label "relates to property"@en ;
```



D3.1 Data Requirements and Interoperability – initial version

```
.
s4agri-new:AgriculturalMachine
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of all agricultural machines that are used on the farm"@en ;
  rdfs:label "Agricultural machine"@en ;
  rdfs:subClassOf owl:Thing ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri-new:hasGreenhouseGasEmission ;
    owl:someValuesFrom sosa:Observation ;
  ] ;
.
s4agri-new:CropCycle
  rdf:type owl:Class ;
  rdfs:comment "1: This class represents the set of crop cycles. A crop cycle is a single cycle of cultivating a crop from sowing till harvesting. For a multiyear plant or crop, a crop cycle is the period between first leaving until harvesting." ;
  rdfs:comment "2: A crop cycle has various attributes, such as the parcel on which the crop is being cultivated as well as various dates, such as first leaves date, blossoming date and fruiting date." ;
  rdfs:comment "3: A s4agri:Crop is equivalent to the class obo:PO_0000003, which contains all or most plant varieties." ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasBlossomingDate ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasEndDatetime ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasFirstLeavesDate ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasFruitingDate ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasStartDatetime ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasToponym ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass s4agri:Crop ;
    owl:onProperty s4agri:contains ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
.
s4agri-new:FarmAssociation
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Class ;
    rdfs:comment "This class represents the set of organizations to which a farm and its farmer can be associated. A
farm/farmer might belong to a larger conglomerate of cooperating farms." ;
    rdfs:subClassOf ploutos:AgriFoodChainActor ;
.

s4agri-new:FarmWorker
    rdf:type owl:Class ;
    rdfs:comment "A class to describe a saref4agri farm worker as subclass of a foaf:Person"@en ;
    rdfs:label "Farmworker"@en ;
    rdfs:subClassOf foaf:Person ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:allValuesFrom xsd:string ;
        owl:onProperty s4agri:hasName ;
    ] ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:onProperty s4agri-new:gotHurtByInjury ;
        owl:someValuesFrom s4agri-new:Injury ;
    ] ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:onProperty s4agri-new:isTypeOfWorkers ;
        owl:someValuesFrom xsd:string ;
    ] ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:onProperty s4agri-new:worksAtFarm ;
        owl:someValuesFrom s4agri:Farm ;
    ] ;
.

s4agri-new:Injury
    rdf:type owl:Class ;
    rdfs:comment "A class to describe possible injuries that a saref4agri farm worker can experience"@en ;
    rdfs:label "Injury"@en ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:allValuesFrom xsd:string ;
        owl:onProperty s4agri:hasName ;
    ] ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:onProperty s4agri-new:occuredDuringOperation ;
        owl:someValuesFrom ploutos:Operation ;
    ] ;
.

s4agri-new:IrrigationSystem
    rdf:type owl:Class ;
    rdfs:comment "This class represents all kind of irrigation systems that are used to irrigate the soil of a parcel on
a farm." ;
    rdfs:label "Irrigation system" ;
    rdfs:subClassOf ssn:System ;
    rdfs:subClassOf [
        rdf:type owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onDataRange xsd:float ;
        owl:onProperty s4agri-new:hasSprinklerFlow ;
    ] ;
    rdfs:subClassOf [
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onDataRange xsd:integer ;
    owl:onProperty s4agri-new:hasNumberOfSprinklers ;
  ] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onDataRange xsd:integer ;
  owl:onProperty s4agri-new:hasNumberOfSprinklersPerPlant ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onDataRange xsd:string ;
  owl:onProperty s4agri-new:hasIrrigationType ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
.
s4agri-new:gotHurtByInjury
  rdf:type owl:ObjectProperty ;
  rdfs:comment "got hurt by injury"@en ;
  rdfs:domain s4agri-new:FarmWorker ;
  rdfs:label "got hurt by injury"@en ;
  rdfs:range s4agri-new:Injury ;
.
s4agri-new:hasAdministrator
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "1: This property expresses the administrator of the parcel, which can be someone else than the farmer that owns the parcel." ;
  rdfs:comment "2: For now, this is modelled for simplicity as a string that contains the name of the administrator. In the future, this can be replaced by a class person with more properties than just the name." ;
  rdfs:label "has administrator" ;
  rdfs:range xsd:string ;
.
s4agri-new:hasAppliedAmount
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Here we make the link to the Ontology of units of Measure and enable classes to have a measure with a numerical value and a unit" ;
  rdfs:label "has applied amount" ;
  rdfs:range om:Measure ;
.
s4agri-new:hasArea
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "This property expresses the area of a flat object such as a parcel or field on the land" ;
  rdfs:label "has area" ;
  rdfs:range xsd:float ;
.
s4agri-new:hasBlossomingDate
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has blossoming date" ;
  rdfs:range xsd:dateTime ;
.
s4agri-new:hasCost
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Here we make the link to the Ontology of units of Measure and enable classes to have a measure with a numerical value and a unit" ;
  rdfs:label "has cost" ;
  rdfs:range om:Measure ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
s4agri-new:hasCropVariety
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has crop variety" ;
  rdfs:range xsd:string ;
.
s4agri-new:hasCultivator
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "1: This property expresses the cultivator of the parcel, which can be someone else than the farmer
that owns the parcel, and takes care of the cultivation of the crop from seeding to harvesting." ;
  rdfs:comment "2: For now, this is modelled for simplicity as a string that contains the name of the cultivator. In
the future, this can be replaced by a class person with more properties than just the name." ;
  rdfs:label "has cultivator" ;
  rdfs:range xsd:string ;
.
s4agri-new:hasCycle
  rdf:type owl:ObjectProperty ;
  rdfs:comment "This is the property that expresses the relationship between a parcel and multiple crop cycles." ;
  rdfs:label "has cycle" ;
.
s4agri-new:hasEndDatetime
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has end datetime" ;
  rdfs:range xsd:dateTime ;
.
s4agri-new:hasFirstLeavesDate
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has first leaves date" ;
  rdfs:range xsd:dateTime ;
.
s4agri-new:hasFruitingDate
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has fruiting date" ;
  rdfs:range xsd:dateTime ;
.
s4agri-new:hasGreenhouseGasEmission
  rdf:type owl:ObjectProperty ;
  rdfs:comment "This property indicates that a farm or a machine can have an estimation of the total greenhouse
gas emission during a time period of usually a year. The emission is expressed in terms of the amount of kilograms
of carbon equivalent per ton of product produced by the farm." ;
  rdfs:domain [
    rdf:type owl:Class ;
    owl:unionOf (
      s4agri:Farm
      s4agri-new:AgriculturalMachine
    ) ;
  ] ;
  rdfs:label "has greenhouse gas emission" ;
  rdfs:range sosa:Observation ;
.
s4agri-new:hasGreenhouseGasEmissionEstimationFormula
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "This property indicates the formula that is used to estimate the total greenhouse gas emission of
a farm during a time period of usually a year. The emission is expressed in terms of the amount of kilograms of
carbon equivalent per ton of product produced by the farm." ;
  rdfs:domain sosa:Procedure ;
  rdfs:label "has greenhouse gas emission estimation formula" ;
  rdfs:range xsd:string ;
.
s4agri-new:hasIrrigationType
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:DatatypeProperty ;
    rdfs:domain s4agri-new:IrrigationSystem ;
    rdfs:label "has irrigation type" ;
    rdfs:range xsd:string ;
.

s4agri-new:hasNumberOfSprinklers
    rdf:type owl:DatatypeProperty ;
    rdfs:domain s4agri-new:IrrigationSystem ;
    rdfs:label "has number of sprinklers" ;
    rdfs:range xsd:integer ;
.

s4agri-new:hasNumberOfSprinklersPerPlant
    rdf:type owl:DatatypeProperty ;
    rdfs:domain s4agri-new:IrrigationSystem ;
    rdfs:label "has number of sprinklers per plant" ;
    rdfs:range xsd:integer ;
.

s4agri-new:hasNutrientConcentration
    rdf:type owl:ObjectProperty ;
    rdfs:label "has nutrient concentration" ;
.

s4agri-new:hasSeedRate
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "This property represents the quantity of seed of a crop that is required per hectare of land to sow
for optimum crop production" ;
    rdfs:domain obo:PO_0000003 ;
    rdfs:label "has seed rate" ;
    rdfs:range xsd:integer ;
.

s4agri-new:hasSprinklerFlow
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "this property indicates the flow of water through the sprinklers of an irrigation system in cubic
meters per hour (m3/h)" ;
    rdfs:domain s4agri-new:IrrigationSystem ;
    rdfs:label "has sprinkler flow" ;
    rdfs:range xsd:float ;
.

s4agri-new:hasStartDatetime
    rdf:type owl:DatatypeProperty ;
    rdfs:label "has start datetime" ;
    rdfs:range xsd:dateTime ;
.

s4agri-new:hasToponym
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "This property expresses the toponym of a parcel, meaning the name of the parcel as it is used by
the locals" ;
    rdfs:label "has toponym" ;
    rdfs:range xsd:string ;
.

s4agri-new:hasTreatedArea
    rdf:type owl:ObjectProperty ;
    rdfs:comment "Here we make the link to the Ontology of units of Measure and enable classes to have a measure
with a numerical value and a unit" ;
    rdfs:label "has treated area" ;
    rdfs:range om:Measure ;
.

s4agri-new:inRegion
    rdf:type owl:DatatypeProperty ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:comment "1: This property expresses the region in which a parcel is located, meaning the county or province of the parcel." ;
rdfs:comment "2: For now, this is modelled as a string for simplicity. In the future, this can be replaced by a class of region with more properties than just the name." ;
rdfs:label "in region" ;
rdfs:range xsd:string ;
.
s4agri-new:isAppliedPer
rdf:type owl:ObjectProperty ;
rdfs:label "is applied per" ;
rdfs:range om:Area ;
.
s4agri-new:isExecutedBy
rdf:type owl:ObjectProperty ;
rdfs:comment "is executed by"@en ;
rdfs:domain ploutos:Operation ;
rdfs:label "is executed by"@en ;
rdfs:range s4agri-new:FarmWorker ;
.
s4agri-new:isPricePer
rdf:type owl:ObjectProperty ;
rdfs:comment "Here we make the link to the Ontology of units of Measure and enable classes to have a measure with a numerical value and a unit" ;
rdfs:label "is price per" ;
rdfs:range om:Unit ;
.
s4agri-new:isTypeOfWorker
rdf:type owl:DatatypeProperty ;
rdfs:comment "is type of worker"@en ;
rdfs:comment "type of worker can vary from full time employee to seasonal worker"@en ;
rdfs:label "is type of worker"@en ;
.
s4agri-new:occuredDuringOperation
rdf:type owl:ObjectProperty ;
rdfs:comment "occurred during operation"@en ;
rdfs:domain s4agri-new:Injury ;
rdfs:label "occured during operation"@en ;
rdfs:range ploutos:Operation ;
.
s4agri-new:usesAgriculturalMachinery
rdf:type owl:ObjectProperty ;
rdfs:comment "This property indicates the kind of agricultural machine that is used for a specific operation" ;
rdfs:domain ploutos:Operation ;
rdfs:label "uses agricultural machinery" ;
rdfs:range s4agri-new:AgriculturalMachine ;
.
s4agri-new:usesIrrigationSystem
rdf:type owl:ObjectProperty ;
rdfs:label "uses irrigation system" ;
rdfs:range s4agri-new:IrrigationSystem ;
.
s4agri-new:worksAtFarm
rdf:type owl:ObjectProperty ;
rdfs:comment "works at farm"@en ;
rdfs:label "works at farm"@en ;
.
s4agri:
dcterms:contributor <https://www.researchgate.net/profile/Jack\_Verhoosel> ;
dcterms:created "2019-04-15"^^xsd:date ;
```



D3.1 Data Requirements and Interoperability – initial version

```
dcterms:creator <http://w3id.org/people/mpoveda> ;
dcterms:creator <http://www.garcia-castro.com/foaf.rdf#me> ;
dcterms:creator <https://www.linkedin.com/in/lauradaniele> ;
dcterms:creator <https://www.linkedin.com/in/mderoode> ;
dcterms:description "SAREF4AGRI is an extension of SAREF for the agriculture and food domain. SAREF4AGRI is specified and published by ETSI in the TS 103 410-6 associated to this ontology file. The list of use cases, standards and requirements that guided the creation of SAREF4AGRI are described in the associated ETSI TR 103 511."@en ;
dcterms:description "This ontology extends the SAREF ontology for the Agricultural domain. This work has been developed in the context of the STF 534 (https://portal.etsi.org/STF/STFs/STFHomePages/STF534.aspx), which was established with the goal to create three SAREF extensions, one of them for the Agricultural domain."@en ;
dcterms:issued "2019-04-30"^^xsd:date ;
dcterms:license <https://forge.etsi.org/etsi-software-license> ;
dcterms:modified "2020-06-05"^^xsd:date ;
dcterms:publisher <https://www.etsi.org/> ;
dcterms:source <https://saref.etsi.org/sources/saref4agri/> ;
dcterms:title "SAREF4AGRI: an extension of SAREF for the agriculture and food domain"@en ;
vann:preferredNamespacePrefix "s4agri" ;
vann:preferredNamespaceUri "https://saref.etsi.org/saref4agri/" ;
rdfs:seeAlso
<https://www.etsi.org/deliver/etsi_ts/103400_103499/10341006/01.01.02_60/ts_10341006v010102p.pdf> ;
owl:versionIRI s4a ;
owl:versionInfo "v1.1.2" ;
.
s4agri:AirTemperature
rdf:type owl:NamedIndividual ;
rdf:type saref:Temperature ;
rdfs:comment "The degree or intensity of heat present in the air."@en ;
rdfs:label "Air temperature"@en ;
.
s4agri:AmbientHumidity
rdf:type owl:NamedIndividual ;
rdf:type saref:Humidity ;
rdfs:comment "The amount of water vapour in the air."@en ;
rdfs:label "Ambient humidity"@en ;
.
s4agri:Animal
rdf:type owl:Class ;
rdfs:comment "An individual and identifiable living organism that feeds on organic matter, typically having specialized sense organs and nervous system and able to respond rapidly to stimuli."@en ;
rdfs:label "Animal"@en ;
rdfs:subClassOf saref:FeatureOfInterest ;
rdfs:subClassOf [
rdf:type owl:Restriction ;
owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
owl:onClass s4agri:ID ;
owl:onProperty s4agri:hasID ;
] ;
rdfs:subClassOf [
rdf:type owl:Restriction ;
owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
owl:onDataRange xsd:dateTime ;
owl:onProperty s4agri:hasBirthDate ;
] ;
rdfs:subClassOf [
rdf:type owl:Restriction ;
owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
owl:onDataRange xsd:dateTime ;
```



D3.1 Data Requirements and Interoperability – initial version

```
    owl:onProperty s4agri:hasDeathDate ;
  ];
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onDataRange xsd:string ;
  owl:onProperty s4agri:hasName ;
];
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty obo:TAXRANK_1000000 ;
  owl:someValuesFrom obo:TAXRANK_0000000 ;
];
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty s4agri:isLocatedIn ;
  owl:someValuesFrom geo:Feature ;
];
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty s4agri:isMemberOf ;
  owl:someValuesFrom s4agri:AnimalGroup ;
];
.
s4agri:AnimalGroup
  rdf:type owl:Class ;
  rdfs:comment "A collection of one or multiple animals."@en ;
  rdfs:label "Animal Group"@en ;
  rdfs:subClassOf saref:FeatureOfInterest ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onDataRange xsd:string ;
    owl:onProperty s4agri:hasName ;
  ];
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:hasMember ;
    owl:someValuesFrom s4agri:Animal ;
  ];
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:isLocatedIn ;
    owl:someValuesFrom geo:Feature ;
  ];
.
s4agri:Building
  rdf:type owl:Class ;
  rdfs:comment "A building represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, storey, and space)."@en ;
  rdfs:label "Building"@en ;
  rdfs:subClassOf geo:Feature ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri:hasName ;
  ];
  rdfs:subClassOf [
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:contains ;
    owl:someValuesFrom s4agri:BuildingSpace ;
  ] ;
.
s4agri:BuildingSpace
  rdf:type owl:Class ;
  rdfs:comment "An entity used to define the physical spaces of the building. A building space contains devices or building objects."@en ;
  rdfs:label "Building space"@en ;
  rdfs:subClassOf geo:Feature ;
.
s4agri:Crop
  rdf:type owl:Class ;
  rdfs:comment "A collection of cultivated plants that is grown on a large scale commercially, especially a cereal, fruit, or vegetable."@en ;
  rdfs:comment "s4agri-new: This class represents all kinds of cultivated plants that can be grown on a parcel of a farm or more generically on any type of substrate in any kind of environment. It is thus a superclass of a wide variety of different species."@en ;
  rdfs:comment "s4agri-new: We have defined this class to be equivalent to obo:PO_0000003 which is the concept of a \"whole plant\", in the PO classification, which is reused by the FOODON ontology. Thereby, the ploutos:Crop concept automatically inherits all the subclasses defined in the FOODON ontology under PO_0000003."@en ;
  rdfs:label "Crop"@en ;
  rdfs:subClassOf saref:FeatureOfInterest ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasSeedRate ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onDataRange xsd:string ;
    owl:onProperty s4agri-new:hasCropVariety ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:string ;
    owl:onProperty s4agri:hasName ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty obo:TAXRANK_1000000 ;
    owl:someValuesFrom obo:TAXRANK_0000000 ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:hasHarvestDate ;
    owl:someValuesFrom xsd:dateTime ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:hasPlantDate ;
    owl:someValuesFrom xsd:dateTime ;
  ] ;
  owl:equivalentClass obo:PO_0000003 ;
.
.
```



D3.1 Data Requirements and Interoperability – initial version

```
s4agri:Deployment
  rdf:type owl:Class ;
  rdfs:comment "Deployment"@en ;
  rdfs:comment "The scenario, including temporal and spatial information, in which the systems and devices are
installed."@en ;
  rdfs:label "Deployment"@en ;
  rdfs:subClassOf ssn:Deployment ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ssn:deployedOnPlatform ;
    owl:someValuesFrom sosa:Platform ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ssn:deployedSystem ;
    owl:someValuesFrom ssn:System ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:hasDeploymentPeriod ;
    owl:someValuesFrom time:TemporalEntity ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:isDeployedAtSpace ;
    owl:someValuesFrom geo:SpatialObject ;
  ] ;
.
```

```
s4agri:EatingActivitySensor
  rdf:type owl:Class ;
  rdfs:comment "Eating activity sensor"@en ;
  rdfs:label "Eating activity sensor"@en ;
  rdfs:subClassOf saref:Sensor ;
.
```

```
s4agri:Farm
  rdf:type owl:Class ;
  rdfs:comment "A plot of land used for the scope of farming which can contain buildings and parcels."@en ;
  rdfs:label "Farm"@en ;
  rdfs:subClassOf geo:Feature ;
  rdfs:subClassOf plontos:AgrifoodChainActor ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri:hasName ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri-new:hasGreenhouseGasEmission ;
    owl:someValuesFrom sosa:Observation ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:contains ;
    owl:someValuesFrom s4agri:Building ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:contains ;
    owl:someValuesFrom s4agri:Parcel ;
  ] ;
```



D3.1 Data Requirements and Interoperability – initial version

```
];
.
s4agri:FarmHolding
  rdf:type owl:Class ;
  rdfs:comment "A class to describe a saref4agri farm holding as subclass of an org:Organization"@en ;
  rdfs:label "Farm holding"@en ;
  rdfs:subClassOf <https://schema.org/Organization> ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom xsd:string ;
    owl:onProperty s4agri:hasName ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:managesFarm ;
    owl:someValuesFrom s4agri:Farm ;
  ] ;
.
s4agri:Farmer
  rdf:type owl:Class ;
  rdfs:comment "A class to describe a saref4agri farmer as subclass of a foaf:Person"@en ;
  rdfs:label "Farmer"@en ;
  rdfs:subClassOf foaf:Person ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom xsd:string ;
    owl:onProperty s4agri:hasName ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:isMemberOf ;
    owl:someValuesFrom s4agri-new:FarmAssociation ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri:managesFarm ;
    owl:someValuesFrom s4agri:Farm ;
  ] ;
.
s4agri:ID
  rdf:type owl:Class ;
  rdfs:comment "ID"@en ;
  rdfs:label "ID"@en ;
.
s4agri:Intake
  rdf:type owl:Class ;
  rdfs:comment "The amount of food or other substance taken into the body of an animal or into the soil."@en ;
  rdfs:label "Intake"@en ;
.
s4agri:IrrigationWater
  rdf:type owl:NamedIndividual ;
  rdf:type saref:Property ;
  rdfs:comment "The amount of water falling in the soil by irrigation methods."@en ;
  rdfs:label "Irrigation water"@en ;
.
s4agri:MilkingSensor
  rdf:type owl:Class ;
  rdfs:comment "Milking sensor"@en ;
  rdfs:label "Milking sensor"@en ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:subClassOf saref:Sensor ;
.
s4agri:MovementActivitySensor
  rdf:type owl:Class ;
  rdfs:comment "Movement activity sensor"@en ;
  rdfs:label "Movement activity sensor"@en ;
  rdfs:subClassOf saref:Sensor ;
.
s4agri:Parcel
  rdf:type owl:Class ;
  rdfs:comment "1: This class represents the set of parcels. A parcel is a plot of land, usually a division of a larger area."@en ;
  rdfs:comment "2: A parcel has various attributes, such as an area, a region, a toponym and an administrator. In addition, the crop that is being cultivated on the parcel has various dates, such as first leaves date, blossoming date and fruiting date. Last but not least, various systems can be used on the parcel for cultivation, such as an irrigation system."@en ;
  rdfs:comment "An area of land, which might be used for grazing animals or planting crops. The parcel is defined as an undividable logical area of land which contains homogeneous items."@en ;
  rdfs:label "Parcel"@en ;
  rdfs:subClassOf obo:ENVO_00000114 ;
  rdfs:subClassOf geo:Feature ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasAdministrator ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:hasToponym ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:cardinality "1"^^xsd:nonNegativeInteger ;
    owl:onProperty s4agri-new:inRegion ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass s4agri-new:IrrigationSystem ;
    owl:onProperty s4agri-new:usesIrrigationSystem ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onDataRange xsd:string ;
    owl:onProperty s4agri:hasName ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:float ;
    owl:onProperty s4agri-new:hasArea ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty s4agri-new:hasCycle ;
    owl:someValuesFrom s4agri-new:CropCycle ;
  ] ;
];
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:subClassOf [  
  rdf:type owl:Restriction ;  
  owl:onProperty s4agri:contains ;  
  owl:someValuesFrom s4agri:Crop ;  
] ;  
.
s4agri:PlantGrowthStage  
  rdf:type owl:NamedIndividual ;  
  rdf:type saref:Property ;  
  rdfs:comment "The level or stage of growth of the plant."@en ;  
  rdfs:label "Plant growth stage"@en ;  
.
s4agri:Platform  
  rdf:type owl:Class ;  
  rdfs:comment "Platform"@en ;  
  rdfs:label "Platform"@en ;  
  rdfs:subClassOf sosa:Platform ;  
.
s4agri:Pluviometer  
  rdf:type owl:Class ;  
  rdfs:comment "A sensor that measures the rain fall."@en ;  
  rdfs:label "Pluviometer"@en ;  
  rdfs:subClassOf saref:Sensor ;  
.
s4agri:Precipitation  
  rdf:type owl:NamedIndividual ;  
  rdf:type saref:Property ;  
  rdfs:comment "The amount of water falling in the soil by natural process (e.g. rain)."@en ;  
  rdfs:label "Precipitation"@en ;  
.
s4agri:Soil  
  rdf:type owl:Class ;  
  rdfs:comment "Upper layer of the earth in which plants grow."@en ;  
  rdfs:comment "s4agri-new: This class represents all kinds of soil. Soil is an environmental material which is primarily composed of minerals, varying proportions of sand, silt, and clay, organic material such as humus, gases, liquids, and a broad range of resident micro- and macroorganisms."@en ;  
  rdfs:label "Soil"@en ;  
  rdfs:subClassOf obo:ENVO_00001998 ;  
  rdfs:subClassOf saref:FeatureOfInterest ;  
.
s4agri:SoilMoisture  
  rdf:type owl:NamedIndividual ;  
  rdf:type saref:Property ;  
  rdfs:comment "The amount of water or humidity contained in the soil."@en ;  
  rdfs:label "Soil moisture"@en ;  
.
s4agri:SoilTemperature  
  rdf:type owl:NamedIndividual ;  
  rdf:type saref:Temperature ;  
  rdfs:comment "The degree or intensity of heat present in the soil."@en ;  
  rdfs:label "Soil temperature"@en ;  
.
s4agri:SoilTensiometer  
  rdf:type owl:Class ;  
  rdfs:comment "A sensor that measures the soil moisture."@en ;  
  rdfs:label "Soil tensiometer"@en ;  
  rdfs:subClassOf saref:Sensor ;  
.
s4agri:Thermometer
```



D3.1 Data Requirements and Interoperability – initial version

```

rdf:type owl:Class ;
rdfs:comment "Thermometer"@en ;
rdfs:label "Thermometer"@en ;
rdfs:subClassOf saref:Sensor ;
.
s4agri:WateringGun
rdf:type owl:Class ;
rdfs:comment "An actuator to irrigate an space."@en ;
rdfs:label "Watering gun"@en ;
rdfs:subClassOf saref:Actuator ;
.
s4agri:WateringSystem
rdf:type owl:Class ;
rdfs:comment "Watering system"@en ;
rdfs:comment "s4agri-new: This concept should be renamed to s4agri:IrrigationSystem. We added the class
s4agri:IrrigationSystem, so this s4agri:WateringSystem can be removed"@en ;
rdfs:label "Watering system"@en ;
rdfs:subClassOf ssn:System ;
.
s4agri:WateringValve
rdf:type owl:Class ;
rdfs:comment "Watering valve"@en ;
rdfs:label "Watering valve"@en ;
rdfs:subClassOf saref:Actuator ;
.
s4agri:WeatherStation
rdf:type owl:Class ;
rdfs:comment "A sensor o system that measure weather conditions."@en ;
rdfs:label "Weather station"@en ;
rdfs:subClassOf ssn:System ;
rdfs:subClassOf saref:Sensor ;
.
s4agri:WeightSensor
rdf:type owl:Class ;
rdfs:comment "Weight sensor"@en ;
rdfs:label "Weight sensor"@en ;
rdfs:subClassOf saref:Sensor ;
.
s4agri:Yield
rdf:type owl:Class ;
rdfs:comment "The produced amount of food for animals or of agricultural products."@en ;
rdfs:label "Yield"@en ;
.
s4agri:contains
rdf:type owl:ObjectProperty ;
rdfs:comment "contains"@en ;
rdfs:label "contains"@en ;
rdfs:subPropertyOf geo:sfContains ;
owl:inverseOf s4agri:isContainedIn ;
.
s4agri:generates
rdf:type owl:ObjectProperty ;
rdfs:comment "A relation to represent the yield generated by a Crop."@en ;
rdfs:label "generates"@en ;
rdfs:subPropertyOf saref:hasProperty ;
.
s4agri:hasBirthDate
rdf:type owl:DatatypeProperty ;
rdfs:comment "A relation to express the birth date of an animal."@en ;
```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:label "has birth date"@en ;
rdfs:range xsd:dateTime ;
.
s4agri:hasDeathDate
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "A relation to express the death date of an animal."@en ;
  rdfs:label "has death date"@en ;
  rdfs:range xsd:dateTime ;
.
s4agri:hasDeploymentPeriod
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation to indicate the temporal duration or period of a deployment."@en ;
  rdfs:comment "has deployment period"@en ;
  rdfs:label "has deployment period"@en ;
.
s4agri:hasHarvestDate
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "A relation to express the day the crop is harvested."@en ;
  rdfs:label "has harvest date"@en ;
  rdfs:range xsd:dateTime ;
  rdfs:subPropertyOf owl:topDataProperty ;
.
s4agri:hasID
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relation to express the unique identifier of an animal."@en ;
  rdfs:label "has id"@en ;
  rdfs:range s4agri:ID ;
  rdfs:subPropertyOf owl:topObjectProperty ;
.
s4agri:hasMember
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relation to represent the members of an AnimalGroup."@en ;
  rdfs:label "has member"@en ;
  owl:inverseOf s4agri:isMemberOf ;
.
s4agri:hasName
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "A relation to express the name of an entity (e.g.,animal)."@en ;
  rdfs:label "has name"@en ;
  rdfs:range xsd:string ;
.
s4agri:hasPlantDate
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "A relation to express the day the crop is planted."@en ;
  rdfs:label "has plant date"@en ;
  rdfs:range xsd:dateTime ;
  rdfs:subPropertyOf owl:topDataProperty ;
.
s4agri:hasReceived
  rdf:type owl:ObjectProperty ;
  rdfs:comment "has received"@en ;
  rdfs:label "has received"@en ;
  owl:inverseOf s4agri:receives ;
.
s4agri:isContainedIn
  rdf:type owl:ObjectProperty ;
  rdfs:comment "is contained in"@en ;
  rdfs:label "is contained in"@en ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
s4agri:isDeployedAtSpace
  rdf:type owl:ObjectProperty ;
  rdfs:comment "Relation to indicate the geographical componet of a deployment."@en ;
  rdfs:comment "is deployed at space"@en ;
  rdfs:label "is deployed at space"@en ;
.
s4agri:isLocatedIn
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relation to express the physical location of a saref4agri entity (e.g., an animal)"@en ;
  rdfs:label "is located in"@en ;
  owl:inverseOf s4agri:isLocationOf ;
.
s4agri:isLocationOf
  rdf:type owl:ObjectProperty ;
  rdfs:comment "is location of"@en ;
  rdfs:label "is location of"@en ;
.
s4agri:isMemberOf
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relation to express that an animal can be part of groups."@en ;
  rdfs:label "is member of"@en ;
.
s4agri:managesFarm
  rdf:type owl:ObjectProperty ;
  rdfs:comment "manages farm"@en ;
  rdfs:label "manages farm"@en ;
.
s4agri:receives
  rdf:type owl:ObjectProperty ;
  rdfs:comment "A relation to represent the intake/consumption of certain substances in a Crop."@en ;
  rdfs:label "receives"@en ;
  rdfs:subPropertyOf saref:hasMeasurement ;
.
<https://schema.org/Organization>
  rdf:type owl:Class ;
  rdfs:label "Organization"@en ;
  rdfs:subClassOf foaf:Agent ;
.
plutosData:cow-meat-part-1
  rdf:type owl:NamedIndividual ;
.
plutosData:cow-meat-part-2
  rdf:type owl:NamedIndividual ;
.
plutosData:cow-meat-part-3
  rdf:type owl:NamedIndividual ;
.
plutosData:cow-meat-part-packaged-1
  rdf:type owl:NamedIndividual ;
.
plutosData:cow-meat-part-packaged-2
  rdf:type owl:NamedIndividual ;
.
plutosData:cow-meat-part-packaged-3
  rdf:type owl:NamedIndividual ;
.
<https://www.tno.nl/agrifood/ontology/plutos/common>
  spin:imports <http://topbraid.org/spin/owlrl-all> ;
.
```



D3.1 Data Requirements and Interoperability – initial version

```
ploutos:AgriFoodChainActor
  rdf:type owl:Class ;
  rdfs:comment "This class represents all organizations acting in the agrifood supply chain." ;
  rdfs:label "Agrifood chain actor" ;
  rdfs:subClassOf gs1:Organization ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:allValuesFrom xsd:string ;
    owl:onProperty s4agri:hasName ;
  ] ;
.

ploutos:AuthoritySpecificCertificateIdentifier
  rdf:type owl:Class ;
  rdfs:comment "1: An authority specific certificate identifier is an identifier that is issued by an authority, to identify a certificate." ;
  rdfs:label "Authority specific certificate identifier" ;
  rdfs:subClassOf ploutos:AuthoritySpecificIdentifier ;
.

ploutos:AuthoritySpecificIdentifier
  rdf:type owl:Class ;
  rdfs:comment "An authority specific identifier is an identifier that is issued by an authority, to identify an entity that is authorised by the authority, e.g. an organization to be certified or a certificate issued by the authority." ;
  rdfs:label "Authority specific identifier" ;
  rdfs:subClassOf owl:Thing ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass ploutos:CertificateStandardOwner ;
    owl:onProperty ploutos:isProvidedBy ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:string ;
    owl:onProperty ploutos:hasIdentifier ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
.

ploutos:AuthoritySpecificOrganisationIdentifier
  rdf:type owl:Class ;
  rdfs:comment "An authority specific identifier is an identifier that is issued by an authority, to identify an organisation. For instance, this can be a GlobalGapNumber: a GGN is a 13-digit number that uniquely identifies each producer and individual member of a producer group in the GLOBALG.A.P. Database. The GGN matches the Global Location Number specification of GS1." ;
  rdfs:label "Authority specific organisation identifier" ;
  rdfs:subClassOf ploutos:AuthoritySpecificIdentifier ;
.

ploutos:CertificateStandardOwner
  rdf:type owl:Class ;
  rdfs:comment "1: This class represents the set of organizations that own certificate standards, such as GlobalGAP or EuroCert. They are also responsible for providing access to certificates that are issued by a conformity assessment organisation after auditing and reviewing the certified object." ;
  rdfs:comment "2: This class is a subclass of ploutos:AgriFoodChainActor and thus a subclass of gs1:Organization. Consequently, it also contains the properties for a supply chain actor." ;
  rdfs:label "Certificate standard owner" ;
  rdfs:subClassOf ploutos:AgriFoodChainActor ;
.

ploutos:ChemicalControlOperation
  rdf:type owl:Class ;
```



D3.1 Data Requirements and Interoperability – initial version

rdfs:comment "This class represents the set of operations to protect crop against any kind of pest or disease. It is a specific PestInterventionOperation and uses a specific pesticide with a treatment material is meant for pest prevention, pest monitoring and pest intervention." ;

```
rdfs:label "Crop protection operation" ;
rdfs:subClassOf ploutos:PestInterventionOperation ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onClass om:Area ;
  owl:onProperty s4agri-new:isAppliedPer ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onClass om:Measure ;
  owl:onProperty s4agri-new:hasAppliedAmount ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onClass om:Measure ;
  owl:onProperty s4agri-new:hasTreatedArea ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onDataRange xsd:string ;
  owl:onProperty ploutos:isJustifiedBecause ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:usesPesticide ;
  owl:someValuesFrom ploutos:Pesticide ;
] ;
```

ploutos:ConformityAssessmentOrganization

```
rdf:type owl:Class ;
rdfs:comment "This class represents the set of conformity assessment organizations, such as ControlUnion, SKAL, QMSCert,.... They are responsible for auditing and reviewing the conditions for issuing certifications according to a specific scheme on behalf of a partner in the chain or a governmental organization that requires the specific certificate." ;
rdfs:label "Conformity assessment organization" ;
rdfs:subClassOf ploutos:AgriFoodChainActor ;
```

ploutos:CropManagementOperation

```
rdf:type owl:Class ;
rdfs:comment "1: This is the class of crop management operations in which the crop/plant is being treated to enhance the performance, e.g. a crop protection operation or pruning operation" ;
rdfs:comment ""2: The actsOnCropStage property indicates that the crop management operation acts on a crop that is in a certain crop stage. Crop stages can be:
```

- 1: germination;
- 2: leaf development;
- 3: formation of side shoots;
- 4: stem elongation;
- 5: vegetative plant parts;
- 6: inflorescence emergence;
- 7: flowering;
- 8: fruit development;
- 9: ripening;



D3.1 Data Requirements and Interoperability – initial version

```
10: senescence. "" ;
rdfs:subClassOf ploutos:Operation ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:actsOnCropStage ;
  owl:someValuesFrom xsd:string ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:isOperatedOn ;
  owl:someValuesFrom s4agri:Parcel ;
] ;
.
ploutos:CropProtectionOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to protect crop against any kind of pest or disease. It is a specific CropManagementOperation and is meant for pest prevention, pest monitoring and pest intervention." ;
  rdfs:label "Crop protection operation" ;
  rdfs:subClassOf ploutos:CropManagementOperation ;
.
ploutos:CuttingOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to for cutting a product in pieces for further processing. It is a specific ProductOperation." ;
  rdfs:subClassOf ploutos:ProductOperation ;
.
ploutos:ENVO_00001998_1
  rdf:type obo:ENVO_00001998 ;
  rdfs:label "ENVO 00001998 1" ;
.
ploutos:FertilizationOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to fertilize the soil on the land of a farm. It is a specific SoilManagementOperation and uses a fertilizer or a specific type and applies this on an area that is treated. A fertilizer is a chemical or natural substance added to soil or land to increase its fertility." ;
  rdfs:label "Fertilization operation" ;
  rdfs:subClassOf ploutos:SoilManagementOperation ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass om:Measure ;
    owl:onProperty s4agri-new:hasTreatedArea ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:string ;
    owl:onProperty ploutos:hasApplicationMethod ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onDataRange xsd:string ;
    owl:onProperty ploutos:hasFertilizationType ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ploutos:usesFertilizer ;
    owl:someValuesFrom ploutos:Fertilizer ;
  ] ;
```



D3.1 Data Requirements and Interoperability – initial version

```
];  
.br/>ploutos:Fertilizer  
  rdf:type owl:Class ;  
  rdfs:comment "1: This represents the class of fertilizers that are used in fertilization operations to give nutrients to soil, water or another substrate. The FreeDictionary defines it as: \"any of a large number of natural and synthetic materials, including manure and nitrogen, phosphorus, and potassium compounds, spread on or worked into soil to increase its capacity to support plant growth\"." ;  
  rdfs:comment "2: The property hasNutrientConcentration relates to an om:Percentage that has a value and a \"phenomenon\" that indicates for which nutrient the percentage holds. Please check the obo:CHEBI list of concepts to find the correct nutrient to be instantiated." ;  
  rdfs:label "fertilizer" ;  
  rdfs:subClassOf ploutos:TreatmentMaterial ;  
  rdfs:subClassOf [  
    rdf:type owl:Restriction ;  
    owl:onProperty s4agri-new:hasNutrientConcentration ;  
    owl:someValuesFrom om:Percentage ;  
  ] ;  
.br/>ploutos:FoodCompany  
  rdf:type owl:Class ;  
  rdfs:comment "The class represents all companies that process food or produce food products" ;  
  rdfs:label "Food company" ;  
  rdfs:subClassOf ploutos:AgriFoodChainActor ;  
.br/>ploutos:HarvestingOperation  
  rdf:type owl:Class ;  
  rdfs:comment "1: This class represents the set of operations to harvest a product from the land of a farm. " ;  
  rdfs:comment "2: It is a ProductOperation in the sense that it has as input the tree, plant or crop that has produced fruits or vegetables that are ripe for harvesting and as output the harvested fruits/vegetables" ;  
  rdfs:comment "3: It is a CropManagementOperation in the sense that it is the last operation to manage the in a crop cycle on a parcel" ;  
  rdfs:label "Harvesting operation" ;  
  rdfs:subClassOf ploutos:CropManagementOperation ;  
  rdfs:subClassOf ploutos:ProductOperation ;  
.br/>ploutos:IrrigationOperation  
  rdf:type owl:Class ;  
  rdfs:comment "This class represents the set of operations to irrigate the soil of a parcel of a farm with water. It is a specific SoilManagementOperation." ;  
  rdfs:label "Irrigation operation" ;  
  rdfs:subClassOf ploutos:SoilManagementOperation ;  
  rdfs:subClassOf [  
    rdf:type owl:Restriction ;  
    owl:cardinality "1"^^xsd:nonNegativeInteger ;  
    owl:onProperty ploutos:hasCycleDuration ;  
  ] ;  
.br/>ploutos:Operation  
  rdf:type owl:Class ;  
  rdfs:comment "An operation is a set of actions on an object performed by a responsible agent during a specific time interval" ;  
  rdfs:subClassOf [  
    rdf:type owl:Restriction ;  
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;  
    owl:onClass foaf:Agent ;  
    owl:onProperty ploutos:responsibleAgent ;  
  ] ;  
  rdfs:subClassOf [  
    owl:cardinality "1"^^xsd:nonNegativeInteger ;  
    owl:onProperty ploutos:hasCycleDuration ;  
  ] ;
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:Restriction ;
    owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
    owl:onClass s4agri-new:AgriculturalMachine ;
    owl:onProperty s4agri-new:usesAgriculturalMachinery ;
  ] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onDataRange xsd:dateTime ;
  owl:onProperty s4agri-new:hasEndDatetime ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onDataRange xsd:dateTime ;
  owl:onProperty s4agri-new:hasStartDatetime ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:hasWeatherObservation ;
  owl:someValuesFrom sosa:Observation ;
] ;
.
ploutos:PackagingOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to package a product into a larger compound. Packaging is the activity of enclosing or protecting products for distribution, storage, sale, and use. It is a specific ProductOperation." ;
  rdfs:subClassOf ploutos:ProductOperation ;
.
ploutos:PestInterventionOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to protect crop against any kind of pest or disease by intervening current pest attacks with pest control methods. It is a specific CropProtectionOperation." ;
  rdfs:label "Pest intervention operation" ;
  rdfs:subClassOf ploutos:CropProtectionOperation ;
.
ploutos:PestMonitoringOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to protect crop against any kind of pest or disease by trying to monitor the presence of possible pests. It is a specific CropProtectionOperation." ;
  rdfs:label "Pest monitoring operation" ;
  rdfs:subClassOf ploutos:CropProtectionOperation ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ploutos:isTargetedTowards ;
    owl:someValuesFrom xsd:string ;
  ] ;
.
ploutos:PestPreventionOperation
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of operations to protect crop against any kind of pest or disease by trying to prevent pest attacks. It is a specific CropProtectionOperation." ;
  rdfs:label "Pest prevention operation" ;
  rdfs:subClassOf ploutos:CropProtectionOperation ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ploutos:isTargetedTowards ;
    owl:someValuesFrom xsd:string ;
  ] ;
```



D3.1 Data Requirements and Interoperability – initial version

```
];
.
ploutos:Pesticide
  rdf:type owl:Class ;
  rdfs:comment "This class represents the set of pesticides. A pesticide is a substance that is meant to control pests. The FreeDictionary defines it as: \"A substance or agent used to kill pests, such as unwanted or harmful insects, rodents, or weeds\"." ;
  rdfs:label "pesticide" ;
  rdfs:subClassOf ploutos:TreatmentMaterial ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
    owl:onDataRange xsd:time ;
    owl:onProperty ploutos:hasPreHarvestInterval ;
  ] ;
.
ploutos:ProductOperation
  rdf:type owl:Class ;
  rdfs:comment "An operation on a product where it changes in a meaningful way, e.g., packaging,"@en ;
  rdfs:subClassOf ploutos:Operation ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ssn:hasInput ;
    owl:someValuesFrom gs1:Product ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ssn:hasOutput ;
    owl:someValuesFrom gs1:Product ;
  ] ;
.
ploutos:SoilManagementOperation
  rdf:type owl:Class ;
  rdfs:comment "1: This is the class of soil management operations in which certain material is applied to the soil an a parcel to enhance the performance, e.g. a fertilization operation, a spraying operation or an irrigation operation" ;
  rdfs:comment "2: Wikipedia says: soil management is the application of operations, practices, and treatments to protect soil and enhance its performance (such as soil fertility or soil mechanics). It includes soil conservation, soil amendment, and optimal soil health. See: https://en.wikipedia.org/wiki/Soil\_management" ;
  rdfs:subClassOf ploutos:Operation ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    owl:onClass om:Area ;
    owl:onProperty s4agri-new:isAppliedPer ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onClass om:Measure ;
    owl:onProperty s4agri-new:hasAppliedAmount ;
    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  ] ;
  rdfs:subClassOf [
    rdf:type owl:Restriction ;
    owl:onProperty ploutos:isOperatedOn ;
    owl:someValuesFrom s4agri:Parcel ;
  ] ;
.
ploutos:TreatmentMaterial
```



D3.1 Data Requirements and Interoperability – initial version

```

rdf:type owl:Class ;
rdfs:comment "This class represents all material or substances that can be used on a parcel, i.e. in the soil or on
the crop, e.g. fertilizers, pesticides, but also water" ;
rdfs:label "Treatment material" ;
rdfs:subClassOf owl:Thing ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:cardinality "1"^^xsd:nonNegativeInteger ;
  owl:onProperty ploutos:hasCommercialName ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onClass om:Measure ;
  owl:onProperty s4agri-new:hasCost ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onClass om:Unit ;
  owl:onProperty s4agri-new:isPricePer ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:hasActiveSubstance ;
  owl:someValuesFrom xsd:string ;
] ;
rdfs:subClassOf [
  rdf:type owl:Restriction ;
  owl:onProperty ploutos:isTargetedTowards ;
  owl:someValuesFrom xsd:string ;
] ;
.

ploutos:actsOnCropStage
rdf:type owl:DatatypeProperty ;
rdfs:comment ""This property indicates that a crop management operation acts on a crop that is in a certain crop
stage. Crop stages can be:
1: germination;
2: leaf development;
3: formation of side shoots;
4: stem elongation;
5: vegetative plant parts;
6: inflorescence emergence;
7: flowering;
8: fruit development;
9: ripening;
10: senescence."" ;
rdfs:domain ploutos:CropManagementOperation ;
rdfs:label "acts on crop stage" ;
rdfs:range xsd:string ;
.

ploutos:hasActiveSubstance
rdf:type owl:DatatypeProperty ;
rdfs:comment "1: This property is used by instances of class TreatmentMaterial to indicate the active substance
in the treatment material." ;
rdfs:comment "2: For now, this is modelled as a string for simplicity. In the future, this can be replaced by a
Substance class of which there can be a different types from a vocabulary" ;
rdfs:domain ploutos:TreatmentMaterial ;
rdfs:label "has active substance" ;

```



D3.1 Data Requirements and Interoperability – initial version

```
rdfs:range xsd:string ;
.
ploutos:hasApplicationMethod
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "this property defines the type of methods that can be used for crop fertilization, e.g., broadcasting,
foliar, placement, and fertigation." ;
  rdfs:label "has application method" ;
  rdfs:range xsd:string ;
.
ploutos:hasAuthoritySpecificOrganisationIdentifier
  rdf:type owl:ObjectProperty ;
  rdfs:domain gs1:Organization ;
  rdfs:label "has authority specific organization identifier" ;
  rdfs:range ploutos:AuthoritySpecificOrganisationIdentifier ;
.
ploutos:hasCommercialName
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has commercial name" ;
  rdfs:range xsd:string ;
.
ploutos:hasCycleDuration
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "This property represents the duration of a sprinkling cycle period including the soak time, i.e.,
the time needed for the water to soak into the soil." ;
  rdfs:domain ploutos:IrrigationOperation ;
  rdfs:label "has cycle duration" ;
  rdfs:range time:Duration ;
.
ploutos:hasFertilizationType
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "this property defines the type of fertilization that is used, e.g., pre-planting." ;
  rdfs:domain ploutos:FertilizationOperation ;
  rdfs:label "has fertilization type" ;
  rdfs:range xsd:string ;
.
ploutos:hasIdentifier
  rdf:type owl:DatatypeProperty ;
  rdfs:label "has identifier" ;
  rdfs:range xsd:string ;
.
ploutos:hasPreHarvestInterval
  rdf:type owl:DatatypeProperty ;
  rdfs:comment "This property indicates preharvest interval, which is the waiting time between a pesticide
application and when a crop can be harvested. During the PHI, the pesticide may be broken down in the plant, or
on its surface." ;
  rdfs:domain ploutos:Pesticide ;
  rdfs:label "has preharvest interval" ;
  rdfs:range time:Duration ;
.
ploutos:hasWeatherObservation
  rdf:type owl:ObjectProperty ;
  rdfs:comment "This property indicates that an operation can have an observation of a weather phenomenon done
during the execution of the operation. Such a weather phenomenon can have influence on the operation and the
environment in which the operation is executed." ;
  rdfs:domain ploutos:Operation ;
  rdfs:label "has weather observation" ;
  rdfs:range sosa:Observation ;
.
ploutos:isJustifiedBecause
```



D3.1 Data Requirements and Interoperability – initial version

```
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "This property indicates the reason why a chemical control operation and the use of a certain
pesticide is justified" ;
    rdfs:domain ploutos:ChemicalControlOperation ;
    rdfs:label "is justified because" ;
    rdfs:range xsd:string ;
.

ploutos:isOperatedOn
    rdf:type owl:ObjectProperty ;
    rdfs:domain ploutos:SoilManagementOperation ;
    rdfs:label "is operated on" ;
.

ploutos:isProvidedBy
    rdf:type owl:ObjectProperty ;
    rdfs:comment "This property relates a certificate to the organization that provides the certificate, e.g. GlobalGap,
EuroCert, ..." ;
    rdfs:domain ploutos:AuthoritySpecificIdentifier ;
    rdfs:label "is provided by" ;
    rdfs:range ploutos:CertificateStandardOwner ;
.

ploutos:isTargetedTowards
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "1: This property expresses the organism towards which a pest prevention operation or a treatment
material is targeted, e.g. late blight" ;
    rdfs:comment "2: For now, this is modelled for simplicity as a string that contains the name of the targeted
organism. In the future, this can be replaced by a class Organisms with more properties than just the name." ;
    rdfs:domain [
        rdf:type owl:Class ;
        owl:unionOf (
            ploutos:TreatmentMaterial
            ploutos:PestPreventionOperation
            ploutos:PestMonitoringOperation
        ) ;
    ] ;
    rdfs:label "is targeted towards" ;
    rdfs:range xsd:string ;
.

ploutos:isTechnicallyAuthorizedByDocument
    rdf:type owl:DatatypeProperty ;
    rdfs:comment "This property indicates the link to a document that technically authorizes a chemical control
operation" ;
    rdfs:domain ploutos:ChemicalControlOperation ;
    rdfs:label "is technically authorized by document" ;
    rdfs:range xsd:string ;
.

ploutos:responsibleAgent
    rdf:type owl:ObjectProperty ;
    rdfs:range foaf:Agent ;
.

ploutos:usesFertilizer
    rdf:type owl:ObjectProperty ;
    rdfs:label "uses fertilizer" ;
    rdfs:range ploutos:Fertilizer ;
.

ploutos:usesPesticide
    rdf:type owl:ObjectProperty ;
    rdfs:label "uses pesticide" ;
    rdfs:range ploutos:Pesticide ;
.
```



Annex B

Test Case and DAT data collection template. In this template questions referring to data types and data recording process were introduced that assisted the requirement specifications for the QuantiFarm data model and the QuantiFarm data platform. The relevant questions are listed here and are underlined within the template.

Technology Description

3. Referring to the DAT’s software, please indicate and/ or describe:

- a. The parameters or indicators that can be extracted from the software providing important information to the farmer.
- b. The output delivered by software analysis and how it influences the farmers decision making process.
- c. The platform and display mode used to consult the data provided by the DAT.

4. Referring to the DAT’s hardware, please indicate, and describe:

The sensors and technologies used for the collection of data on the parameters referred in the answers above

8. Regarding data collection, please indicate:

- a. How often do you collect data?
- b. How does the data collection process work?
- c. Is the data collected through the DAT enough for decision making purposes, or some other tool is used (even manually)?
- d. Is there any data that you collect periodically and have to communicate/ report to any external organisation? Please detail.

Test Case (TC) Description Form

This form will serve to better understand the Test Case (TC) details with special focus on the DAT (Digital Technologies in Agriculture) description, the cost benefit relation, and sustainability impact. With this information we will be able to develop the most suitable analysing methods for your Test Case. The objective is to coordinate efforts among all of us and plan the methodologies that will be used to process and analyse the data provided by you. Given the multilateral aspects of this project, we will be testing and analysing data from different technologies, sectors, crop/ animal, and biogeographical regions. Also, given the importance of cycles on the farm, parts of this survey, along with new questions, will be asked again during the project.

In order to facilitate this task, the analysing parameters should be set accordingly to the TC needs so we may do as little changes as possible along the project, avoiding major methodological changes and complications in the data collection process along the projects’ duration. For this reason, it is important to gather as much information as possible about the technology, the parameters it provides, how it assists in the decision-making process, and the conditions in which it is operating.

During the application stage you have provided some information about the TC. To normalise the information gathered from all TCs, we are now asking for more details about the DAT and the impact on agriculture activity. Please feel free to use the information provided at the application stage to give background on specific details asked in this document. We count on your expertise on the field and technology to provide a Test Case description as detailed as possible.

Note - Throughout the questionnaire the word “DAT” is mentioned repeatedly. In the context of QuantiFarm and this questionnaire, a DAT is defined as a particular Digital Agriculture solution that consists of hardware (e.g.: tractor, drone, computer, robot, weather station, etc.) and/or software, and that helps the farmer in



D3.1 Data Requirements and Interoperability – initial version

performing an individual production step (e.g.: fertilisation, weeding, irrigation, harvesting etc) or in managing the farm.		
Partners involved		
Partner name	Role / Value Chain Actor Type <i>(e.g.: Farmer, Cooperative, Agri-food company, Technology Provider, Contractor etc.)</i>	Involvement <i>(Full partner / Third party)</i>
Country and Biogeographical Region		
Agricultural Sector		
<ul style="list-style-type: none"> ● Please select from the following list which agricultural objects are the focus of your Test Case <ul style="list-style-type: none"> ○ Apiculture ○ Aquaculture ○ Arable ○ Livestock - Dairy ○ Horticulture - Fruit ○ Livestock - Meat ○ Horticulture -Vegetables 		
Please provide the following information about the farm and the farmers involved		
<ul style="list-style-type: none"> ● Please fill-out the following tables (according to necessity) for each of the farms. Copy the table as many times as necessary. ● Please copy paste as many tables as necessary to organise the information 		
Exact location of the farm where the DAT is used		
Number of parcels on the farm where the DAT is applied		
Size of these parcels (ha or number of animals)		
Crop cultivated on these parcels (there might be more than one)		
Total approximate revenue of the farm (please indicate an interval of values)		
Farmers Name		
Farmers contact (email)		
Exact location of the farm where the DAT is used		
Number of parcels on the farm where the DAT is applied		
Size of these parcels (ha or number of animals)		
Crop cultivated on these parcels (there might be more than one)		
Total approximate revenue of the farm (please indicate an interval of values)		



D3.1 Data Requirements and Interoperability – initial version

Farmers Name	
Farmers contact (email)	
.....	
Description of the end user of the DAT	
<ul style="list-style-type: none"> Please provide some information on who will be mostly using the DAT in daily practice and in what role (e.g.: farmer, advisor, agricultural cooperative, etc). 	
Technology Description	
To contextualise the Test Case, please provide a general description on the DAT (250 words max)	
<ul style="list-style-type: none"> What are the main goals in using the technology (e.g.: input/cost reduction, yield increase, risk reduction, environmental sustainability, animal welfare, etc.) Part of the farm operations where the technology is applied (e.g.: individual production steps or “whole-farm approach”). In which process phases are the DAT applied and how does it help (e.g.: cultivation, irrigation, sowing, harvesting, farm management, traceability)? 	
1- Referring to the technology provider:	
a. Who is/are the technology developer(s)? Please describe the organisations that have been involved in the development of the DAT and their specific roles.	
<ul style="list-style-type: none"> Please indicate the name of the organisations involved, and what roles they had (e.g.: researcher, prototype development, manufacturer, etc.) 	
b. Who is responsible for the DATs' commercialization and what is the commercialization model?	
<ul style="list-style-type: none"> Please let us know who the owner of the DAT is Did the farmer buy the solution or is it just paying for services? Did a contractor buy the solution and is using it to offer services to farmers? Who made the decision to adopt the DAT? Was there an investment needed to adopt the DAT? If yes, how much was the total investment? Who were the sponsors / public incentives / investors (if there were any)? If a different funding model was used, please describe it as detailed as possible (e.g.: annual subscription with no additional investment). 	
2- Referring to farmers:	
What type of benefits have you obtained thanks to the use of the DAT? Was the investment “worth it”?	
<ul style="list-style-type: none"> On what specific tasks is the DAT helpful? How is it helpful? (e.g.: time saving, money saving, worker comfort, friendlier impact on the environment, information management, etc.) 	
3- Referring to the DAT’s software, please indicate and/ or describe:	
a. <u>The parameters or indicators that can be extracted from the software providing important information to the farmer.</u>	
<ul style="list-style-type: none"> Some of the information provided by the software comes in the form of parameters and indicators like - (for precision irrigation) the water evapotranspiration (for animal welfare) body temperature, (arable) growth rate, (milking) amount of milk. Please let us know what is provided by your DAT. 	
b. <u>The output delivered by software analysis and how it influences the farmers decision making process.</u>	



D3.1 Data Requirements and Interoperability – initial version

<ul style="list-style-type: none"> • <i>Apart from single parameters and indicators, some software compiles that information providing direct suggestions and lines of action (e.g.: when to harvest, what section needs water, what animal has a health problem etc.). Please elaborate for your DAT.</i>
<p>c. <u>The platform and display mode used to consult the data provided by the DAT.</u></p> <ul style="list-style-type: none"> • <i>Please let us know where you check the information provided by the DAT (e.g.: phone, laptop, if there is remote access, if you must be close to the DAT)</i> • <i>Please indicate how the data is made available (e.g.: via an application programming interface, email, cloud repository, paper)</i>
<p>4- Referring to the DAT's hardware, please indicate, and describe: <u>The sensors and technologies used for the collection of data on the parameters referred in the answers above.</u></p> <ul style="list-style-type: none"> • <i>In the previous question you mentioned the important information gathered from the software. Now, we would like to ask you about the sensors and technologies (hardware) that gather the same information. (e.g.: a tractor, pivot, weather tower, processing line, drone, temperature sensor, infrared sensor, pressure sensor, humidity sensor, specific substance detector, etc.)</i>
<p>5- <u>Regarding DAT installation, please describe the process</u></p> <ul style="list-style-type: none"> • <i>Please describe, to the best of your knowledge, the installation steps.</i> • <i>Who was responsible for installing the DAT on the farm? (e.g.: DAT provider's technical team, farmer, advisor, a combination of stakeholders, etc.)</i> • <i>The installation process went according to plan, or were there adjustments needed to accommodate specific needs?</i> • <i>The DAT installation finished on time? If not, was it longer or shorter than expected?</i>
<p>6- Usually, the DATs are implemented to address a certain need on the farm and to assist in performing the tasks that are directly related with that need. But there might be cases when the DAT use enables farmers to accomplish additional tasks beyond the focus of the DAT (e.g.: use a particular system of the DAT, like a tractor, a drone, weather tower, or a specific sensor). If that is the case, please indicate and describe:</p> <p>a. What other tasks benefit from the use of the DAT?</p>
<p>b. What DATs are involved in those other tasks.</p> <ul style="list-style-type: none"> • <i>e.g.: weather station, tractor, drone, etc.</i>
<p>c. How do these systems contribute to the accomplishment of additional tasks?</p> <ul style="list-style-type: none"> • <i>e.g.: a weather station may be there to support a DSS on pest management but could also support the farmer in getting an early warning about frost, even if this was not foreseen originally by the DAT provider.</i>
<p>7- For how long this DAT has been used on the farms integrated in the TC (consider the first farms to systematically use the DAT)?</p>
<p>8- <u>Regarding data collection, please indicate:</u></p> <p>a. <u>How often do you collect data?</u></p> <ul style="list-style-type: none"> • <i>Hourly, daily, weekly, monthly, annually, etc.?</i>
<p>b. <u>How does the data collection process work?</u></p> <ul style="list-style-type: none"> • <i>Some DATs have this process automated via IoT, but there might be the case where you must dislocate to the DAT to extract the data directly from there. Please describe your situation.</i> • <i>List of data types that the Test Case is expected to generate (e.g.: weather parameters recordings, fuel consumption, scouting outcomes about pest infestation)</i> • <i>Please indicate the unit for measuring each data type (e.g.: Celsius, Litres)</i> • <i>What data model is available that describes the data that is being measured about these agricultural objects?</i>



D3.1 Data Requirements and Interoperability – initial version

c.	<u>Is the data collected through the DAT enough for decision making purposes, or some other tool is used (even manually)?</u>
	<ul style="list-style-type: none"> • Do you use excel or any other tool/app to treat the data somehow? If you do, please let us know the tools and the methods used to analyse the data. • Please tell us about the format that the Dataset is expressed (e.g.: JSON, XLS, CSV). • Is it necessary to extract data using a hard drive?
d.	Is there any data that you collect periodically and have to communicate/ report to any external organisation? Please detail.
	<ul style="list-style-type: none"> • What data is shared, and to what purpose? • Which data types are open and available to be reused in the context of <i>QuantiFarm</i> ? • To which organisation do you communicate/ report it to? • Is this communication/reporting mandatory?
9-	The TC will consist of comparing data from a parcel using the DAT to another parcel that is not using the DAT. Referring to the parcel that will not be using the DAT, please indicate if the same tasks/ processes are done. If so, please describe how the same tasks/ processes are performed.
	<ul style="list-style-type: none"> • We would like to know how the tasks were made before DAT implementation (or without DAT) and compare a “before and after” scenario. • If some of the tasks were not done before DAT implementation, please state it and explain why.
10-	Still referring to the parcel not using the DAT, please indicate in more detail:
a.	If and how the parameters/ indicators monitored by the DAT (when the DAT is available) are controlled when the DAT is not available.
	<ul style="list-style-type: none"> • Please try to be as specific as possible on how the parameters analysis was done without the DAT. If the analysis was not possible because there was no data available, please let us know the method that was used on the farm to understand how the decisions were taken (e.g.: how did you know when to water the crop, when to harvest, when to call the vet?).
b.	How many more (or less) farm workers are involved in performing the tasks without the DAT when compared to the same tasks using the DAT?
	<ul style="list-style-type: none"> • It is natural that DAT implementation allows for the reduction of the working force on performing certain tasks, but in some cases that might not happen. Please let us know if there was a reduction in the workload and to what point.
c.	How many additional hours do the farm workers spend performing the tasks without the DAT?
	<ul style="list-style-type: none"> • If there is no direct reduction on the number of workers with DAT implementation, perhaps there was a reduction in the amount of time they spent performing the tasks. Please let us know an approximation of the time saved in person-hours.
d.	Are there any tools or practices that became obsolete with DAT implementation? Does this obsolescence have any consequences to other tasks in the farm? Please describe it.
	<ul style="list-style-type: none"> • With DAT implementation, some previously used tools and common practices for the same purpose may have lost their practicality / usefulness (or part of it). Please let us know, in that case, if there are additional methods that have “suffered” or tasks that became harder to accomplish.

Table X - Template for the Test Case Description Form.



Annex C

The schema utilised by the FAIRShare platform for modelling the properties of Digital Agriculture Technologies. These properties have been incorporated within the QCSM.

```
const toolSchema = mongoose.Schema({  
  name: { type: String, required: true },           // No 1  
  countries: [{ type: String, required: true }],    // No 2  
  languages: [{ type: String, required: true }],    // No 3  
  website: {type: String, required: true},          // No 4  
  cost: {type: String, required: true},             // No 5  
  launchYear: {type: String, required: true},      // No 6  
  updateYear: {type: String, required: false},     // No 7  
  usersNo: {type: String, required: false},        // No 8  
  downloads: {type: String, required: false},      // No 9  
  providerName: {type: String, required: true},    // No 10  
  providerEmail: {type: String, required: true},   // No 11  
  providerWebsite: {type: String, required: true}, // No 12  
  category: [{type: String, required: true}],      // No 13  
  target: [{type: String, required: true}],        // No 14  
  sector: [{type: String, required: true}],        // No 15  
  applied: {type: String, required: true},         // No 16  
  organic: {type: String, required:false},        // No 17  
  languageOther: {type: String, required: false},  // No 18  
  title: { type: String, required: true },         // No 19  
  titleOther: { type: String, required: false },   // No 20  
  desc: {type: String, required: true},            // No 21  
  descOther: {type: String, required: false},     // No 22  
  keywords: {type: String, required: true},        // No 23  
  keywordsOther: {type: String, required: false},  // No 24  
  delivery: [{type: String, required: true}],      // No 25  
  conn: {type: String, required: false},          // No 26
```



D3.1 Data Requirements and Interoperability – initial version

```
source: [{type: String, required: true}], // No 27
access: {type: String, required: true}, // No 28
know: {type: String, required: true}, // No 29
train: {type: String, required: true}, // No 30
benefits: [{type: String, required: true}], // No 31
challenges: [{type: String, required: true}], // No 32
imagePath: [{type: String, required: true}], // No 33
documentPath: [{type: String, required: false}], // No 34
videos: {type: String, required: false}, // No 35
creator: {type: mongoose.Schema.Types.ObjectId, ref: 'User', required:
true }, // No 36
isDraft: {type: String, required: true} // No 37
});
```

Annex D

This Annex provides directions on how to deploy the data platform and initiate the software modules that load data to the platform. The directions are also available in the README.md file here: <https://gitlab.com/QuantiFarm/data-platform/-/blob/main/README.md>

QuantiFarm Data Platform

This is the source code repository of the QuantiFarm Data Platform.

The main components are the following:

- **triple-store**: Triple store for storing test case and assessment data.
 - This triple store will be queried by the various *tool web services*, which are not yet included in this prototype.
- **tc11-loader**: Proof-of-concept loader (ETL) of test case data that reads data from a web service, maps it to QCSM-conform RDF, and loads it into the triple store.

It should be noted that every test case loader should write triples to **its own graph**. This way, when source data is updated, it is trivial to replace any existing data using the SPARQL Graph Store Protocol.

Implementation

Currently, this is a prototype, and not yet deployed anywhere. **status**

Running

the

platform

To start the platform and run the loader(s), run the following:

```
docker compose up
```



D3.1 Data Requirements and Interoperability – initial version

Now, the services should be available at:

- <http://localhost:3030/quantifarm/query>
 - The triple store's endpoint for SPARQL 1.1 Query (with ARQ extensions)
 - To be used by the tool web services
- <http://localhost:3030/quantifarm/update>
 - The triple store's endpoint for SPARQL 1.1 Update (with ARQ extensions)
 - To be used by the loaders
- <http://localhost:3030/quantifarm/data>
 - The triple store's endpoint for SPARQL Graph Store Protocol and Quad extensions
 - To be used by the loaders and tool web services
- <http://localhost:3030/quantifarm/get>
 - The triple store's endpoint for SPARQL Graph Store Protocol and Quad extensions (read-only)
 - To be used by the tool web services

To receive data from the triple store, you can, for example, do the following:

receive and write all of the data in the TC11 graph to a Turtle file.

```
curl
localhost:3030/quantifarm/data?graph=https://quantifarm.eu/graphs/tc
11 --header "Accept: text/turtle" > tc11-graph.ttl
```

To update the data from the loaders, they can simply be run as a Docker Compose service:

this will overwrite the data that it previously wrote, if any

```
docker compose up tc11-loader
```



Annex E

The QuantiFarm data platform is bundled with the use of the Docker framework. The docker-compose.yml configuration file for initiating the platform is available in QuantiFarm 's gitlab (<https://gitlab.com/QuantiFarm/data-platform/-/blob/main/docker-compose.yml>) and also here:

```
services:
  triple-store:
    platform: linux/amd64
    image: registry.gitlab.com/quantifarm/data-platform/triple-store:0.0.2
    build:
      context: ./triple-store
    args:
      JENA_VERSION: 4.8.0
    ports:
      - "3030:3030"
    command: [ "--conf", "/fuseki/configuration/conf.ttl" ]
    volumes:
      - ./fuseki-conf.ttl:/fuseki/configuration/conf.ttl:ro
      - triple-store-logs:/fuseki/logs
      - triple-store-data:/fuseki/databases
  tc11-loader:
    build:
      context: .
    dockerfile: ./tc11-loader/Dockerfile
    environment:
      GAIASENSE_URL:
      GAIASENSE_USERNAME:
      GAIASENSE_PASSWORD:
  fairshare-loader:
    build:
      context: .
    dockerfile: ./fairshare-loader/Dockerfile
    environment:
      TRIPLE_STORE_ENDPOINT:
```



D3.1 Data Requirements and Interoperability – initial version

```
volumes:
  - ./fairshare-loader/tools-collection.json:/data/tools-
collection.json
  recommendation-tool:
    build:
    context: .
    dockerfile: ./tool-services/recommendation-tool/Dockerfile
    environment:
      SPARQL_QUERY_ENDPOINT:
    restart: unless-stopped
    ports:
      - "8000:8000"
volumes:
  triple-store-data:
  triple-store-logs:
```

