



D4.3 Technology Support Report v1

WP4 - Technology Support

Deliverable Lead: CIS

Dissemination Level: Public

Deliverable due date: 31/01/2019

Actual submission date: 15/02/2019

Version 1.0



Document Control Page	
Title	D4.3 Technology Support Report v1
Creator	Pascal Dihé (CIS)
Description	This deliverable will accompany the WP4 software tools and report on the work performed in WP4 since the project start and provide links to access the models and tools resulting from this work. It will also provide an updated plan for the WP4 work until the project end, taking into account the user's feedback from the first version of the demonstrators.
Publisher	CLARITY Consortium
Contributors	Pascal Dihé (CIS), Bernhard Skarbal (AIT)
Creation date	09/10/2018
Type	Text
Language	en-GB
Rights	copyright "CLARITY Consortium"
Audience	<input checked="" type="checkbox"/> Public <input type="checkbox"/> Confidential <input type="checkbox"/> Classified
Status	<input type="checkbox"/> In Progress <input type="checkbox"/> For Review <input type="checkbox"/> For Approval <input checked="" type="checkbox"/> Approved

Disclaimer

Disclaimer

The text, figures and tables in this report can be reused under a provision of the Creative Commons Attribution 4.0 International License. Logos and other trademarks are not covered by this license.

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the CLARITY consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the CLARITY Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the CLARITY Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Table of contents

Table of contents.....	4
List of figures	6
CLARITY project overview.....	8
Executive summary.....	9
1 Introduction	10
1.1 Purpose of this document	10
1.2 Relation to other deliverables	10
1.3 Intended audience.....	11
1.4 Structure of the document	11
2 Reference documents	13
2.1 Abbreviations and Glossary	13
3 Realisation and implementation	15
3.1 Transition layer	15
3.2 Emergent architecture.....	15
3.2.1 Technology support plan	17
3.2.2 Mock-Ups.....	18
3.2.3 Architecture diagrams	19
3.2.4 CSIS product backlog.....	19
3.2.5 Code repositories and implementation issues.....	20
4 Catalogue of Elements at Risk and Adaptation Options (T4.2)	22
4.1 Catalogue of Elements at Risk and Adaptation Options.....	22
4.1.1 Software service for the calculation of impact and risk.....	23
4.1.2 Catalogue/inventory of elements at risk and their vulnerability classes.....	23
4.1.3 Catalogue/inventory of damage level probability functions	24
4.1.4 Interface between EMIKAT and Drupal	24
4.2 Catalogue of Data Sources and Simulation Models.....	25
4.3 Table Component	26
5 Scenario Management (T4.3).....	29
5.1 Data Dashboard	29
5.2 Data Package Export and Import Tool	31
5.3 Scenario Management.....	33
6 Scenario Transferability (T4.4)	36
6.1 Map Component.....	36
6.2 Scenario Transferability Component	37
7 Scenario Analysis, Decision Support and Report Generation (T4.5).....	40

7.1	Multi Criteria Decision Analysis Tool	40
7.2	Report Generation	42
8	Integration and Infrastructure (T4.1 and T1.4)	44
8.1	Container Engine and Cloud Infrastructure	44
8.2	UI Integration Platform	45
8.3	Integration and Development Platform	47
8.4	Integration RDMBS	48
8.5	Data Repository	50
9	Conclusion	53
	References	54
	Annex 1: Description of background from previous projects	55
1.	AIT EMIKAT (environmental catalogue).....	55
2.	SUDPLAN Scenario Management System	56
3.	CRISMA Multi Criteria Decision Analysis Tool.....	56
4.	Climate Twins	57

List of figures

Figure 1: CLARITY CSIS architecture layers and perspectives.....	11
Figure 2: Realisation and implementation	15
Figure 3: CLARITY GitHub repositories (excerpt).....	16
Figure 4: CLARITY co-creation teams.....	16
Figure 5: Climate service co-creation (excerpt).....	17
Figure 6: Technology support diagrams repository.....	18
Figure 7: Mock-up example (excerpt)	18
Figure 8: “Reference modelling workflow” (excerpt).....	19
Figure 9: CSIS product backlog (excerpt).....	20
Figure 10: CLARITY task-level project boards (excerpt)	20
Figure 11: T1.3 “Climate services co-creation” project board (excerpt).....	21
Figure 12: First 5 rows from the Result of an Impact Calculation carried out by EMIKAT (excerpt)	23
Figure 13: Aggregated impact results for the (excerpt)	23
Figure 14: EMIKAT table showing some Element at Risk Types.....	23
Figure 15: EMIKAT table showing some vulnerability classes (excerpt)	23
Figure 16: CSIS table “Exposure Elements” (EU-GL step “Evaluate Exposure”)	24
Figure 17: Damage level probability function example for heat wave	24
Figure 18: Damage level probability functions as stored in EMIKAT.....	24
Figure 19: EMIKAT REST-API functionalities	25
Figure 20: Catalogue of Data Sources and Simulation Models	26
Figure 21: Table Component mock-up (hazard indexes)	27
Figure 22: Table Component Technology Support.....	27
Figure 23: Table Component (impact).....	28
Figure 24: Updated Data Dashboard technology support	30
Figure 25: Data Dashboard prototype.....	30
Figure 26: Updated Data Package Export and Import Tool technology support	32
Figure 27: Data Package Export and Import Tool (excerpt)	32
Figure 28: Private CSIS configuration GitLab repository	34
Figure 29: CSIS Helpers module GitHub repository.....	35
Figure 30: Map Component.....	37
Figure 31: Update Scenario Transferability Component technology support	38
Figure 32: Scenario Transferability Component: Geofield Map example	39
Figure 33: Updated Multi Criteria Decision Analysis Tool technology support.....	41

Figure 34: Multi Criteria Decision Analysis Tool (excerpt)	42
Figure 35: Updated Report Generation technology support	43
Figure 36: T1.4 project board (excerpt).....	44
Figure 37: Container Engine and Cloud Infrastructure.....	45
Figure 38: UI Integration Platform iFrame integration.....	46
Figure 39: Drupal UI Integration Platform.....	47
Figure 40: Integration and Development Platform	48
Figure 41: PostgreSQL Docker container in Container Infrastructure.....	49
Figure 42: Integration RDMBS	49
Figure 43: RDBMS Backup	50
Figure 44: GeoServer Data Repository	51
Figure 45: sFTP Data Repository.....	52

CLARITY project overview

Urban areas and traffic infrastructures that are linking such areas are highly vulnerable to climate change. Smart use of existing climate intelligence can increase urban resilience and generate benefits for businesses and society at large. Based on the results of FP7 climate change, future internet and crisis preparedness projects (SUDPLAN, ENVIROFI, CRISMA) with an average TRL of 4-5. and following an agile and user-centred design process, end-users, purveyors and providers of climate intelligence co-create an integrated climate services Information System (CSIS) to systematically plan the integration of resilience measures into urban infrastructure.

As a result, CLARITY will provide an operational eco-system of cloud-based climate services to calculate and present the expected effects of cc-induced and -amplified hazards at the level of risk, vulnerability and impact functions. CLARITY will offer what-if decision support functions to investigate the effects of adaptation measures and risk reduction options in the specific project context and allow the comparison of alternative strategies. Four demonstration cases will showcase CLARITY climate services in different climatic, regional, infrastructure and hazard contexts in Italy, Sweden, Austria and Spain; focusing on the planning and implementation of urban infrastructure development projects.

CLARITY will provide the practical means to include the effects of CC hazards and possible adaptation and risk management strategies into planning and implementation of such projects, focusing on increasing CC resilience. Decision makers involved in these projects will be empowered to perform climate proof and adaptive planning of adaptation and risk reduction options.

Executive summary

This report is the third deliverable of WP4 “Technology Support” of the CLARITY project, funded by the EU’s Horizon 2020 Programme under Grant Agreement number 730355. WP4 provides the technological backbone of the CLARITY climate service Information System (CSIS) by tailoring the technological background to project needs. For this, WP4 will integrate and adapt existing (background) tools and services that are necessary for realisation of the CLARITY reference scenarios (demonstration cases) and the implementation of the EU-GL [1] into the CLARITY climate services.

It is an accompanying report to the technologies and software components that are adapted, extended, customised and deployed by WP4 and configured, assembled and integrated by WP1 to implement the CLARITY CSIS. It reports on the work performed in WP4 since the project start, provides links to access the software, tools and documentation resulting from this work and summaries the continuously updated work plan that takes emerging requirements from the agile co-creation process and feedback from the end users of the CSIS and the demonstration cases into account.

1 Introduction

The introduction chapter defines the purpose and scope of the technology support report as well as its relation to other deliverables and briefly explains the structure of the document.

1.1 Purpose of this document

This document is an accompanying report to the technologies and software components that are adapted, extended, customised and deployed by WP4 and configured, assembled and integrated by WP1 to implement the CLARITY CSIS. It reports on the work performed in WP4 since the project start, provides links to access the software, tools and documentation resulting from this work and summaries the continuously updated work plan that takes emerging requirements from the agile co-creation process and feedback from the end users of the CSIS and the demonstration cases into account. Thus, it serves also to document the emergent architecture introduced in D4.2 “CLARITY CSIS Architecture” [2].

Besides the document at hand, the actual nature of this deliverable is software (source code and/or binaries) and their technical documentation made available in online repositories, (micro)-services deployed in CLARITY’s containerised service infrastructure and the online CSIS product backlog¹ linked with kanban-style² task boards representing a dynamic work plan in support of CLARITY’s agile project management approach.

1.2 Relation to other deliverables

The technology support report heavily builds upon the results of the deliverables D4.1 “Technology Support Plan” [3] and D4.2 “CLARITY CSIS Architecture” [2], which are required readings to understand the underlying assumptions, concepts and design decisions. To avoid repetitions and to keep this report focused and concise, it makes extensive use of references to these documents.

D4.2 “CLARITY CSIS Architecture” establishes the shared understanding among all CLARITY stakeholders about the overall goals of the CSIS Architecture and the essential design decisions and architectural principles to realise these goals. The architecture is structured according to the four-layered MCRI - Mission, Concepts, realisation and implementation scheme (Figure 1), whereby the deliverable D4.3 “Technology Support Report” represents mainly the implementation layer. The architecture is furthermore organised to provide two overlapping perspectives: An explicit architecture and an emergent architecture. The main reason for making this distinction between Explicit- and emergent architecture is to be able to manage complexity and changeability in the agile and evolving product design and development, where many technical and implementation details cannot be specified upfront. The explicit architecture thereby defines the essence of the system, that is the, the design decisions with the “highest impact and cost of change” [4]. All non-critical decisions and technology choices can then be deferred to the emergent architecture that iteratively evolves during the agile co-creation process. An additional transition layer between these architectural perspectives anticipates the expected changes as opportunity to generate value while preserving the invariant essence of the system. In this sense, the technology support report represents the emergent architecture and serves as a means to document the implementation of the CSIS following a continuously updated (technology support) plan.

¹ “A list of all things that needs to be done within the agile product development cycle.” [8]

² “Kanban is a popular framework used to implement agile software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.” <https://www.atlassian.com/agile/kanban>

D4.1 “Technology Support Plan” identifies and specifies the different building blocks and software components of the CSIS. A building block is a generic, composable, adaptable and domain- and location-independent unit of functionality (component) that meets the identified business and user requirements by implementing a set of related functional requirements. A software component on the other hand is a concrete IT service, tool, system or model that is suitable for the realisation of a building block. While the technology support plan covers mainly the realisation Layer of the Architecture, the technology choices for realising a certain building blocks are part of the transition layer and subject to change.

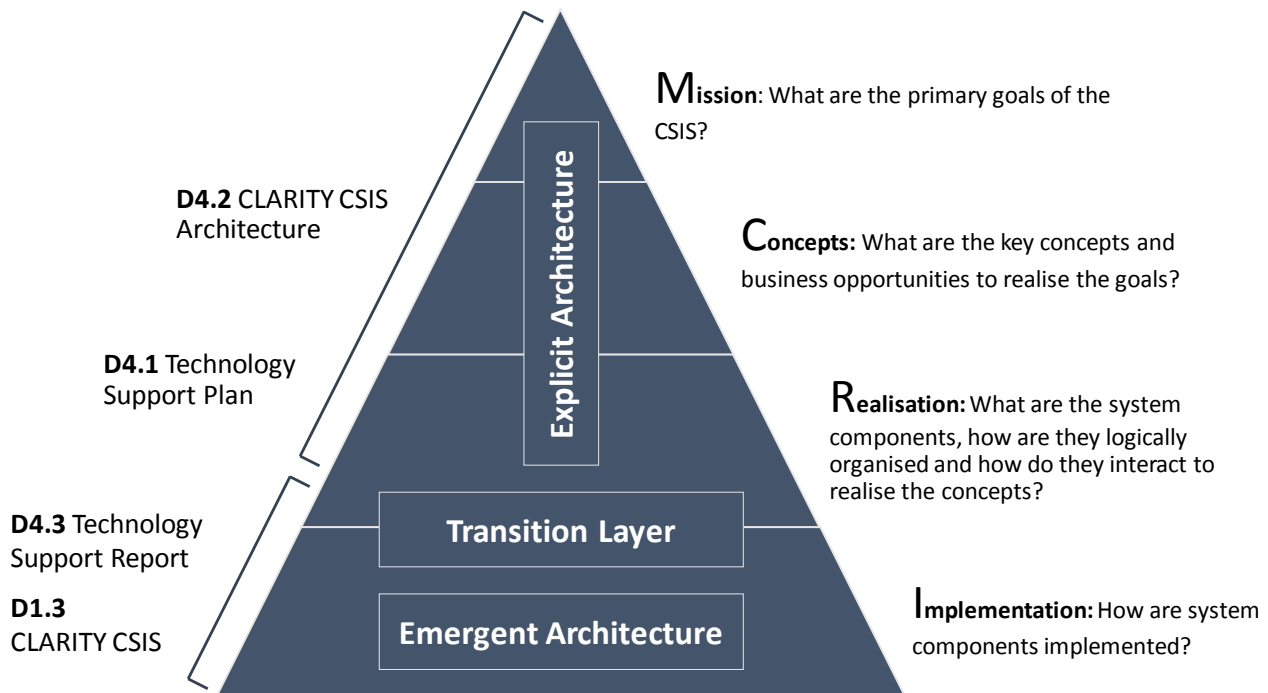


Figure 1: CLARITY CSIS architecture layers and perspectives

In this sense, the technology support report serves also for documenting the expected changes by updating the initial technology support plan. In particular, the related technology choices and the prioritisation of User Stories and derived functional requirements are explained.

While the technology support report mainly addresses independent components (building blocks), **D1.3 "CLARITY CSIS"** represents an integrated software system composed of those interacting components. Accordingly, the accompanying report to D1.3 describes that part of the emergent architecture and the implementation, respectively that covers the integrated CSIS as a whole.

1.3 Intended audience

The target readers of this document are all members of the CLARITY consortium as they cover all categories of stakeholders (end users, service suppliers, developers, etc.) of the CSIS.

1.4 Structure of the document

The structure of the document and the relationships between the different chapters is as follows:

Chapter 1 (this chapter) introduces the document and explains the overall purpose of this document and its relation to other work packages and deliverables.

Chapter 2 lists the documents that were used or referenced in the development of this report and provides a document-specific list of abbreviations.

Chapter 3 briefly summarises the current realisation and implementation parts of the overall CSIS Architecture and thus the transition layer and emergent architecture, respectively.

Chapter 4 - 8 describe the updates of the building blocks and the technology choices in of this document in relation to WP1 and WP4 tasks.

Chapter 9 provides the conclusions and a summary on follow-up activities in other work packages.

Annex 1 reports on the actual usage of tools and technologies inherited from previous projects.

2 Reference documents

The following documents were used or referenced in the development of this report:

- D1.3 “CLARITY C SIS Architecture v1”
- D2.2 “Catalogue of local data sources and sample datasets”
- D4.1 “Technology Support Plan v1”
- D4.2 “CLARITY CSIS Architecture”
- D5.1 “Exploitation Requirements and Innovation Design v1”
- D5.7 “MyClimateServices.eu Marketplace”
- D7.9 “Data Management Plan v2”

2.1 Abbreviations and Glossary

A common glossary of terms for all CLARITY deliverables, as well as a list of abbreviations, can be found in the public document “CLARITY Glossary” available at <http://cat.clarity-h2020.eu/glossary/main>.

Abbreviation/Acronym	Definition
AJAX	Asynchronous JavaScript and XML
BB	Building Block
CC	Climate Change
CCA	Climate Change Adaptation
CCCA	Climate Change Centre Austria
CKAN	Comprehensive Kerbal Archive Network
CLARITY	Integrated Climate Adaptation Service Tools for Improving Resilience Measure
CRISMA	Modelling crisis management for improved action and preparedness
CS	climate service
CSIS	CLARITY climate services Information System
CSW	Catalogue Service for the Web
CTA	Constructive Technology Assessment
DC	Demonstration Case
DC	Dublin Core
DoA	Description of Action (Annex 1 to the Grant Agreement)
DRR	Disaster Risk Reduction
EC	European Commission
ERDDAP	Environmental Research Division's Data Access Program
EU-GL	Non-paper Guidelines for Project Managers: Making vulnerable investments climate resilient (Document)
GeoJSON	geographical JavaScript Object Notation
GeoTIFF	Geographic Tagged Image File format

HTML5	Hypertext Markup Language, version 5
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
IPCC	Intergovernmental Panel on Climate Change
JSON	JavaScript Object Notation
MCDA	Multi-Criteria Decision Analysis
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
RDBMS	Relational Database Management System
REST	Representational State Transfer
RIA	Rich Internet Application
SMS	Scenario Management System
SPA	Single Page Application
SQL	Structured Query Language
SUDPLAN	Sustainable Urban Development Planner for Climate Change Adaptation
TRL	Technology Readiness Level
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WP	Work Package

3 Realisation and implementation

This chapter briefly summarises the current realisation and implementation parts of the overall CSIS Architecture and thus the transition layer and emergent architecture, respectively (Figure 2).

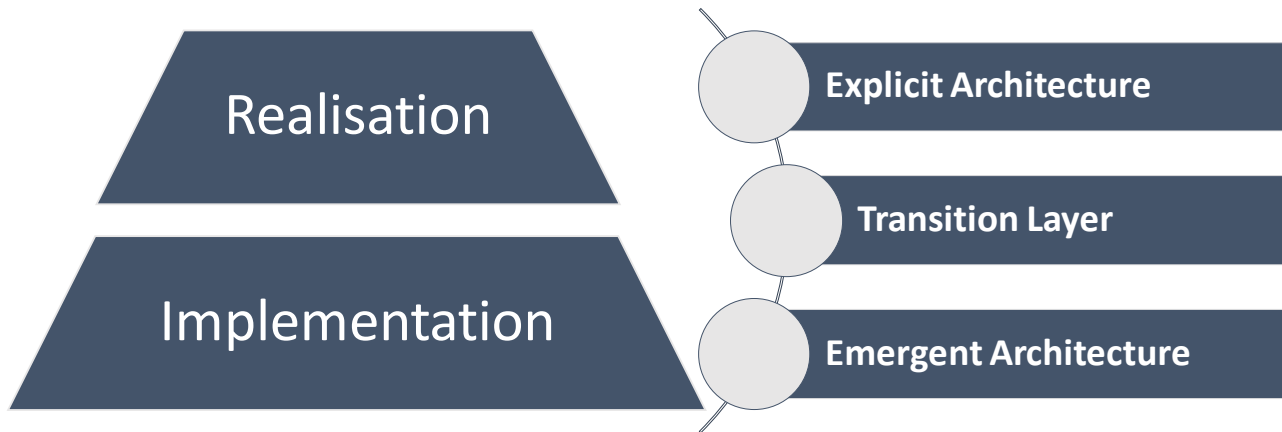


Figure 2: Realisation and implementation

3.1 Transition layer

The transition layer has been introduced to defer architecturally important decisions that could not be fully taken in advance or that were prone to changes. Located between the realisation and implementation part of the architecture (Figure 2), it covers mainly the selection of software components and Key Technologies used for realising the building blocks of the CSIS. In addition, the building blocks as such and their role in the integrated CSIS might be subject to changes. Both the updates of the building blocks and the technology choices are described in chapters 4 to 8 of this document in relation to WP1 and WP4 tasks. Thereby, only changes compared to the initial decisions that have been made in the technology support plan are included.

3.2 Emergent architecture

For (coordinating) the development of the CSIS and the co-creation of related climate services, CLARITY uses GitHub³ as central code repository and co-creation environment. Different repositories for software components implementing specific building blocks or supplemental and supporting tools and services have been created (Figure 3).

³ “GitHub Inc. is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.” (<https://en.wikipedia.org/wiki/GitHub>)

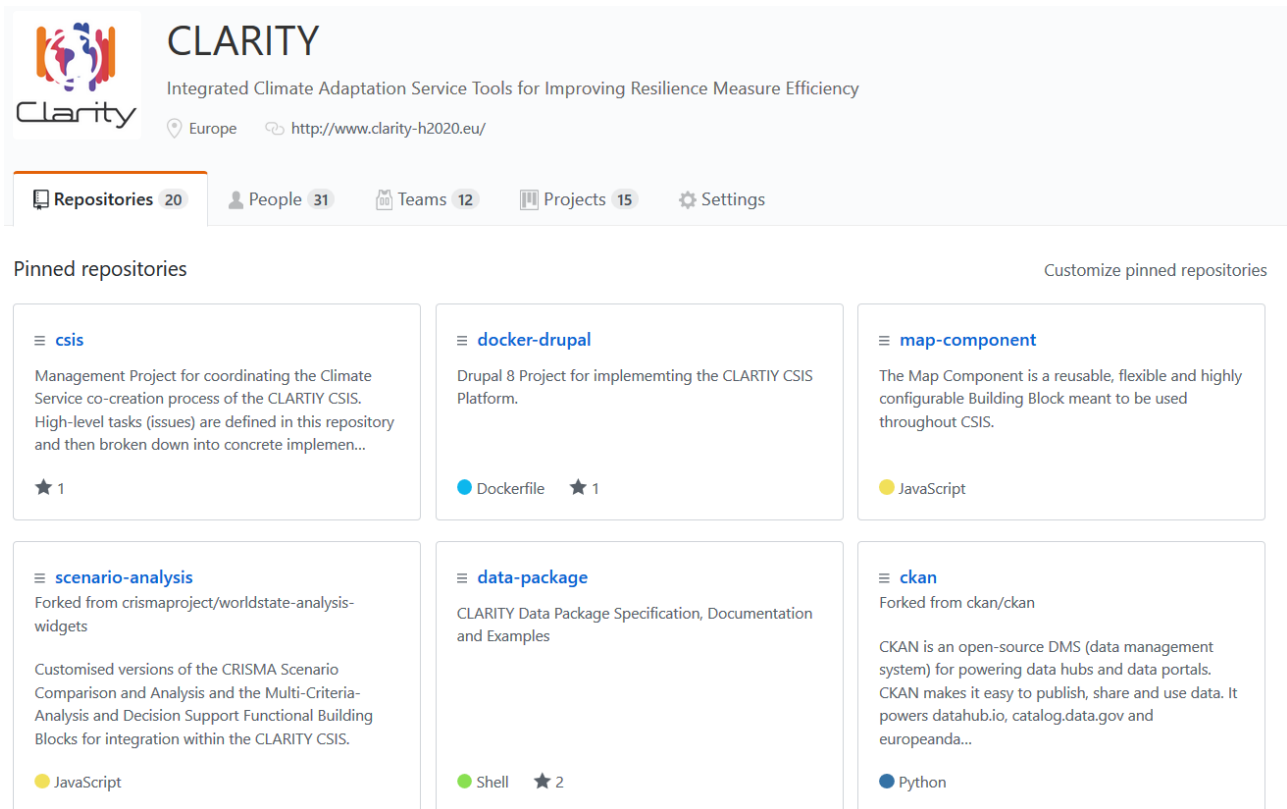



Figure 3: CLARITY GitHub repositories (excerpt)

Besides mere git-based source code management functionality, GitHub offers also a set of project management tools that are used by agile teams to discuss, coordinate and steer the entire product development life-cycle. While those tools have initially been employed to support agile software development, CLARITY uses them also for co-creation of climate services and ultimately the implementation of the CSIS. The overall CLARITY Team (Figure 4) on GitHub therefore is not dedicated to software developers only, but all stakeholders of the CSIS and its adjoining climate services.


<input type="checkbox"/> CSIS End Users CSIS End Users		12 members	5 teams
<input type="checkbox"/> Data Processing Team Experts involved in data processing and transformation, e.g. calculation of h...		9 members	0 teams
<input type="checkbox"/> Dissemination Team effective project communication and dissemination of its results		3 members	0 teams
<input type="checkbox"/> Mathematical Models Implementati... Develpers & Modellers involved in the specification and implementation Mathem...		5 members	0 teams
<input type="checkbox"/> Science Support Team (Climate) Scientits that support the developers of the CSIS		7 members	0 teams
<input type="checkbox"/> User Interface Team Team Members expected to work on the Drupal Site and the CSIS User Interface ...		4 members	0 teams

Figure 4: CLARITY co-creation teams

As example of a transparent climate service co-creation process, an excerpt of a typical conversation between different stakeholders (developers, data providers and climate scientists) of the CSIS is depicted in Figure 5.



1st Data Package: 01 HC Layers #8
 p-a-s-c-a-l opened this Issue on 3 Dec 2018 · 19 comments



@ghilbrae Could you please transform the ncf files (sftp://5.79.69.49/clarityftp/europe/hazard_indices/heat/heat-wave-duration/Index_Tx75p/) to the normalised grid and make it available in your geoserver instance?

RobAndGo commented on 5 Dec 2018

Member + 😊 ...

IMHO we don't need the additional hazard maps with 'normalised' indexes (1 to 3, low, medium, high), but just the thresholds (e.g. x = 8 and y = 14)

Yes, that is true. It would also save some work from my side. :-)

@clarity-h2020/data-processing-team One point to add is that the thresholds I used weren't fixed values, but rather, values relative to the baseline (1971-2000). So, if at one place (e.g. Vienna) the baseline maximum heatwave length was 10 days and at another place (Naples) the baseline maximum heatwave length was 15 days, then a "high" hazard level for those places in the future would be >20 days and >30 days, respectively (i.e. increase >100% as I had defined it). This gets around the problem of latitudinal variations in the climate.

However, in keeping with **@p-a-s-c-a-l**'s idea, in this case one would just need to define the appropriate percentage values for low/medium/high and then have the software calculate e.g.
 $100 \times \frac{[(\text{future layer}) - (\text{baseline layer})]}{(\text{baseline layer})}$
 Then find the points with values e.g. > 100% which would correspond to the high hazard level.
 Where the baseline layer contains zeros and the future layer positive values, one could set those points to high - otherwise keep it at low.

The percentage values one uses to define low/medium/high would be different according to the index (e.g. for the flood hazard, the 5-day precipitation index would have 10% and 20% as a better choices for the different hazard levels).

Figure 5: Climate service co-creation (excerpt)

The main artefacts that serve the documentation of the emergent architecture are summarised in the next chapters.

3.2.1 Technology support plan

It describes on a rather general level how building blocks should be implemented on the basis of software components and key technologies. It is summarised in deliverables D4.1 "Technology Support Plan" and the current deliverable document, which provides an update of the initial plan (chapters 4 to 8). Additionally, the specifications of building blocks and the related information on technology support including technology support diagrams (example: Figure 22) are maintained in a GitHub repository at <https://github.com/clarity-h2020/csis-architecture/> (Figure 6) and available from <https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/index>.

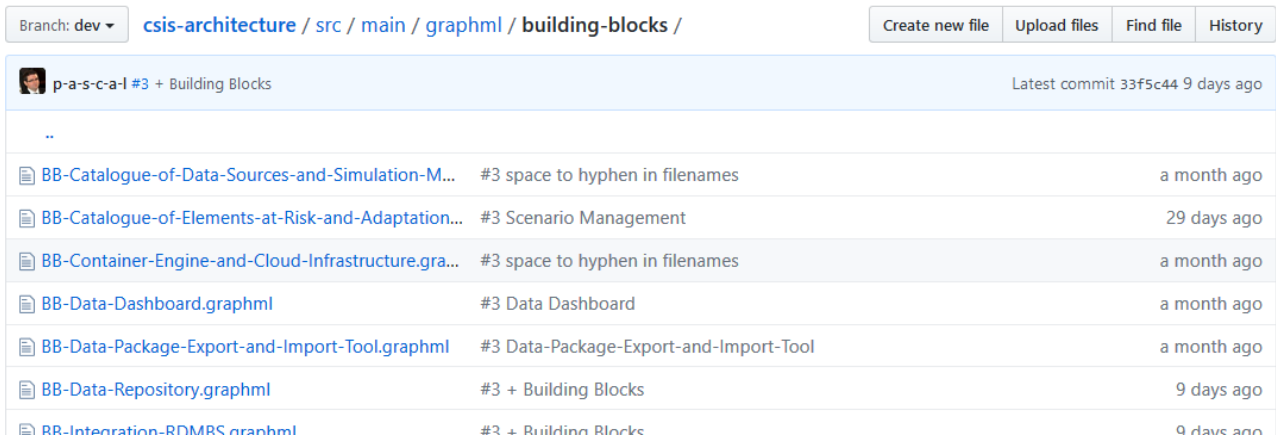


Figure 6: Technology support diagrams repository

3.2.2 Mock-Ups

Mock-ups (Figure 7) represent a visual preview of the envisaged products and services for end users and at the same time serve as "user interface blueprint" for developers.

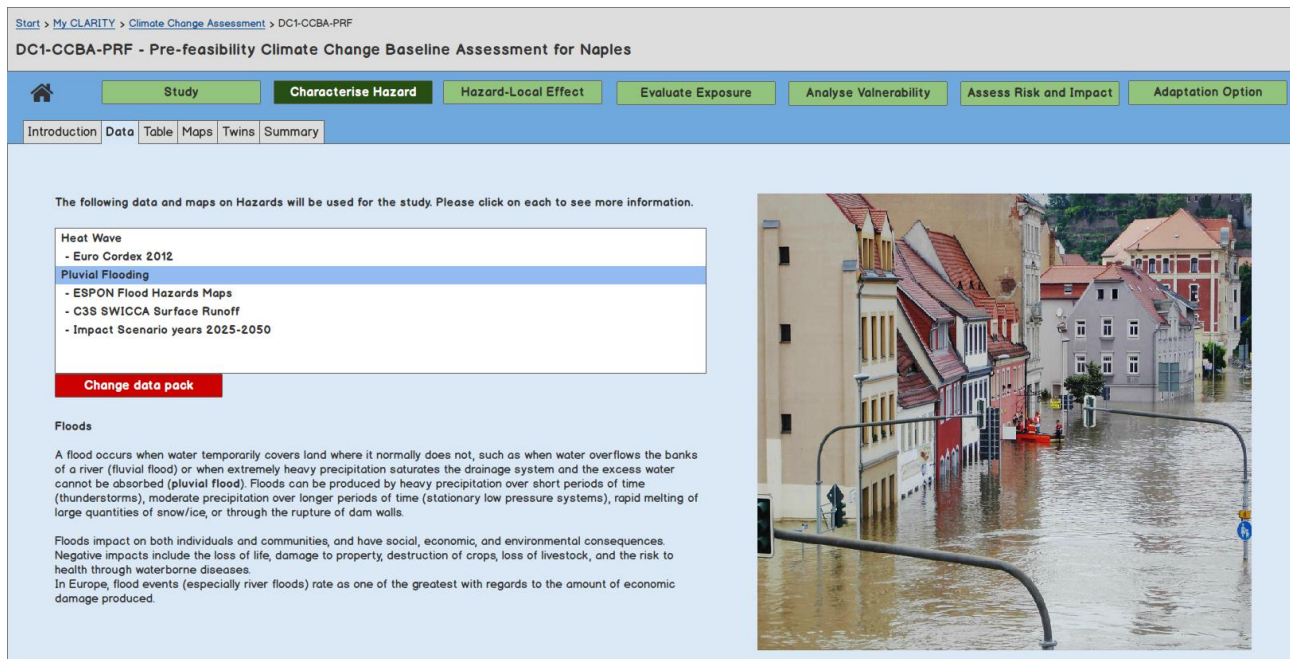


Figure 7: Mock-up example (excerpt)

CSIS mock-ups are currently maintained in a project internal ownCloud⁴ repository and will eventually be made available on GitHub. Deliverable D1.3 "CLARITY CSIS" provides an overview and a description of the current CSIS mock-ups.

⁴ "ownCloud is open source file sync and share software for everyone from individuals, to large enterprises and service providers. ownCloud provides a safe, secure, and compliant file synchronization and sharing solution on servers that you control." (https://doc.owncloud.org/server/10.1/user_manual/)

3.2.3 Architecture diagrams

Usually, rather simple and lightweight architecture diagrams are used to document and communicate the initial draft of an emergent software architecture. However, since the common understanding about the “Reference Modelling Workflow” is already pretty advanced, a more detailed technical diagram of the CLARITY CSIS architecture can be created (Figure 8). It covers mainly those components that are required for the implementation of the EU-GL process model and goes well beyond “black box” building blocks.”

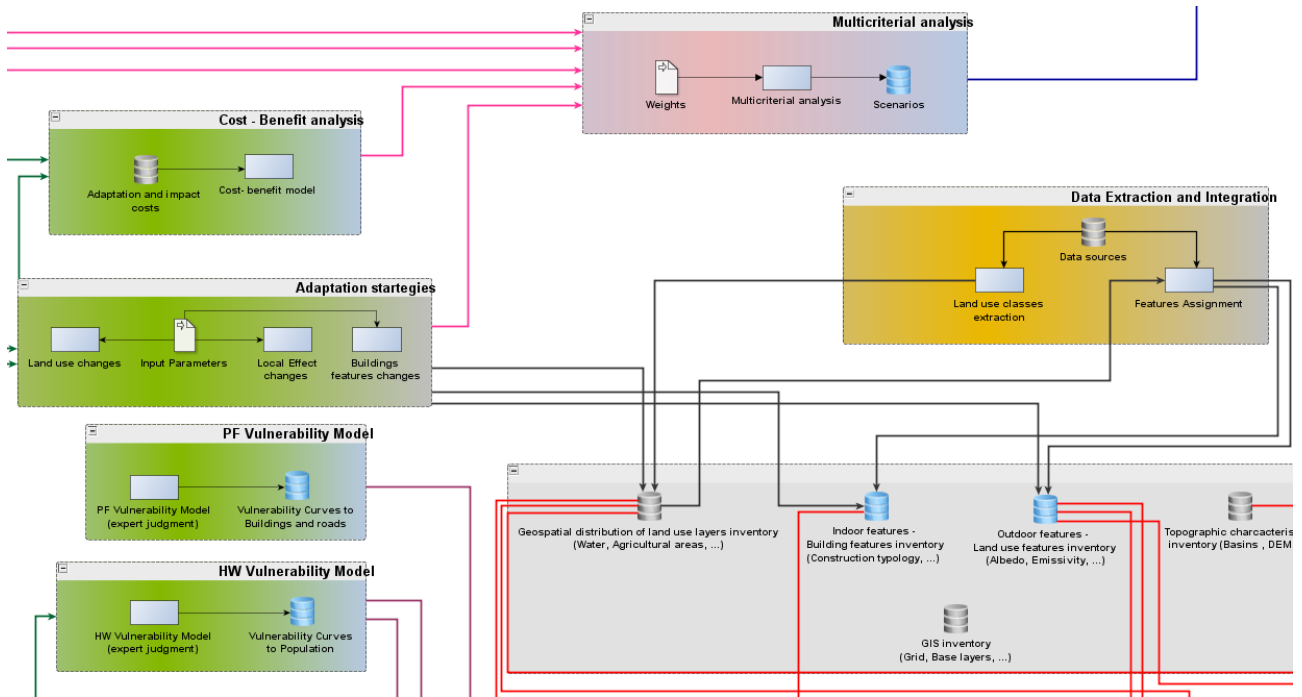


Figure 8: “Reference modelling workflow” (excerpt)

The source code of the diagrams is available at <https://github.com/clarity-h2020/csis-architecture/tree/dev/src/main/graphml> and a brief description is given in deliverable D1.3 “CSIS Architecture”.

3.2.4 CSIS product backlog

The CSIS product backlog at <https://github.com/clarity-h2020/csis/issues> (Figure 9) represents all identified highly-level tasks (“issues”⁵) that need to be accomplished in order to advance the development of the CSIS. The backlog is used to organize, manage, and track the entire CSIS development process. The level of technical and implementation details is kept low intentionally, since the entire co-creation team, which not only consists of software developers but also of end users, scientist and data providers (Figure 4), is engaged in the product backlog. A breakdown of the backlog issues into detailed subtasks (“implementable issues”) is performed at the level of dedicated repositories (Figure 3).

⁵ “Issues are a great way to keep track of tasks, enhancements, and bugs for your projects. They’re kind of like email - except they can be shared and discussed with the rest of your team.” (<https://guides.github.com/features/issues/>)

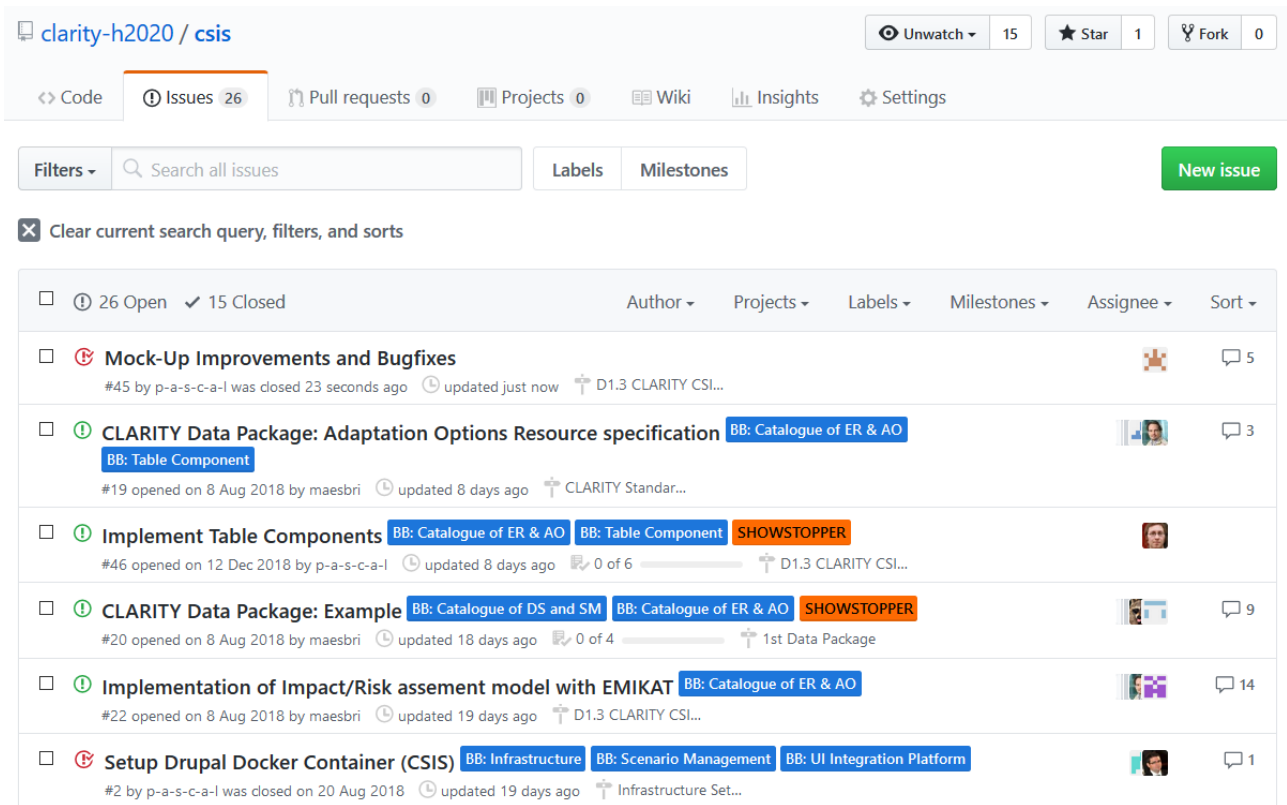


Figure 9: CSIS product backlog (excerpt)

3.2.5 Code repositories and implementation issues

The source code of adapted, newly developed components and the corresponding deployment and runtime configurations for CLARITY’s Container Engine and Cloud Infrastructure (chapter 8.1) is made available in the source code repositories (Figure 3) of the “CLARITY-H2020” GitHub Organisation at <https://github.com/clarity-h2020>.

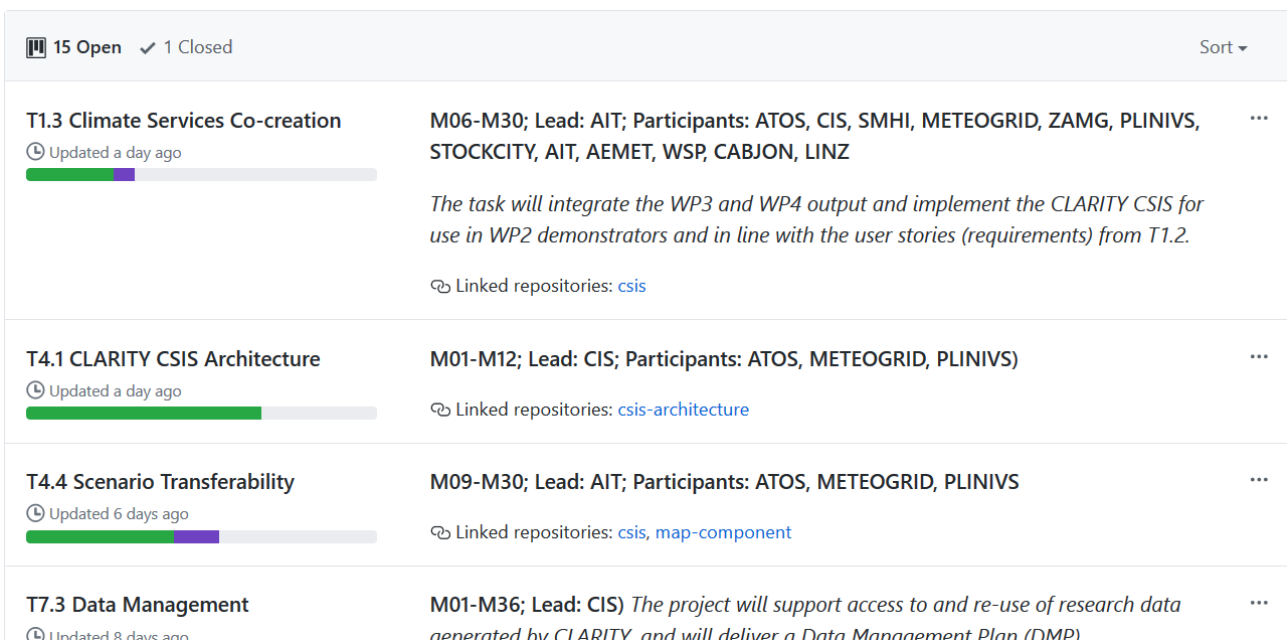
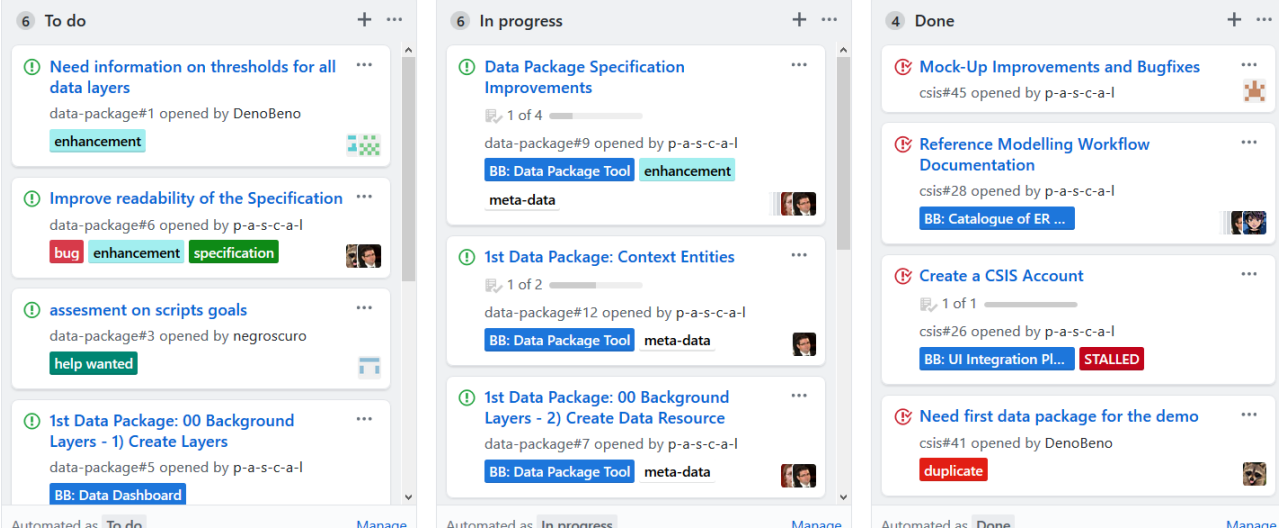


Figure 10: CLARITY task-level project boards (excerpt)

The product backlog (Figure 9) is broken down into “implementable issues” at the level of the distinct repositories and building blocks. Additionally, issues are assigned to several task-level project boards (Figure 10). Additionally, task-level project boards are used by the CLARITY CO-Creation Team to visualize and prioritise development tasks that are planned or in-progress either in a specific repository or across the complete “CLARITY-H2020” Organisation. Figure 11 shows an example of project board for T1.3 “Climate Services Co-creation”.



The screenshot displays a Jira project board with three columns: 'To do', 'In progress', and 'Done'. Each column contains several task cards with details such as titles, descriptions, assignees, and labels.

Column	Task Title	Description	Assignee	Labels
To do (6)	Need information on thresholds for all data layers	data-package#1 opened by DenoBeno		enhancement
	Improve readability of the Specification	data-package#6 opened by p-a-s-c-a-l		bug, enhancement, specification
	assessment on scripts goals	data-package#3 opened by negroscurio		help wanted
	1st Data Package: 00 Background Layers - 1) Create Layers	data-package#5 opened by p-a-s-c-a-l		BB: Data Dashboard
In progress (6)	Data Package Specification Improvements	data-package#9 opened by p-a-s-c-a-l		BB: Data Package Tool, enhancement, meta-data
	1st Data Package: Context Entities	data-package#12 opened by p-a-s-c-a-l		BB: Data Package Tool, meta-data
	1st Data Package: 00 Background Layers - 2) Create Data Resource	data-package#7 opened by p-a-s-c-a-l		BB: Data Package Tool, meta-data
Done (4)	Mock-Up Improvements and Bugfixes	csis#45 opened by p-a-s-c-a-l		
	Reference Modelling Workflow Documentation	csis#28 opened by p-a-s-c-a-l		BB: Catalogue of ER ...
	Create a CSIS Account	csis#26 opened by p-a-s-c-a-l		BB: UI Integration Pl..., STALLED
	Need first data package for the demo	csis#41 opened by DenoBeno		duplicate

Figure 11: T1.3 “Climate services co-creation” project board (excerpt)

4 Catalogue of Elements at Risk and Adaptation Options (T4.2)

The project board of T4.2 is available at <https://github.com/orgs/clarity-h2020/projects/7>.

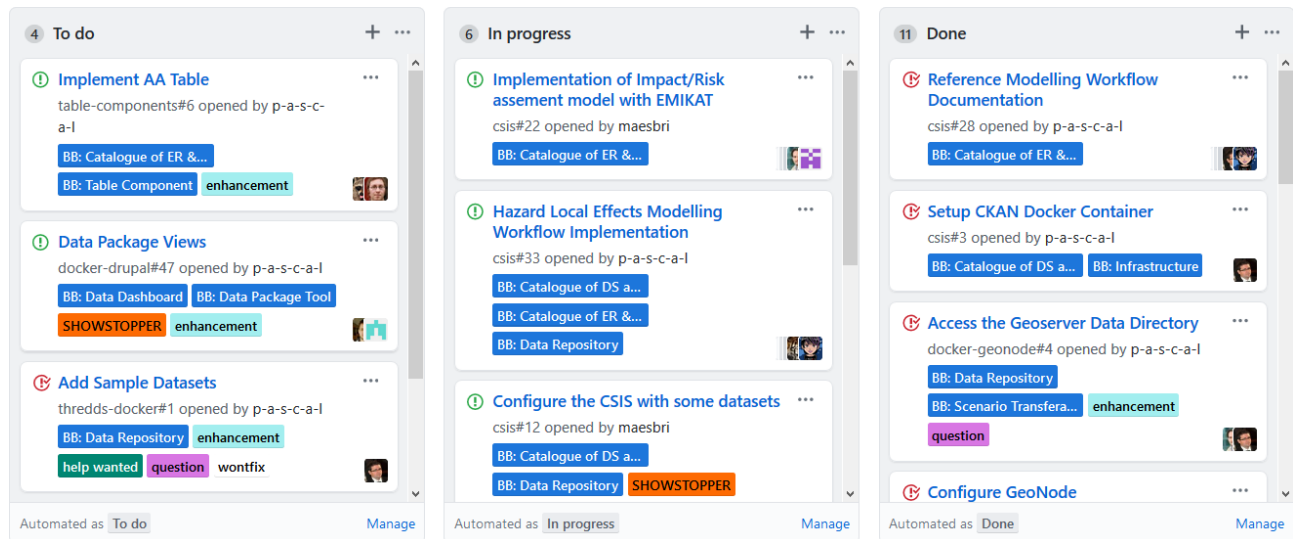


Figure: T4.2 project board (excerpt)

4.1 Catalogue of Elements at Risk and Adaptation Options

“The Catalogue of Elements at Risk and Adaptation Options is strongly linked to the EU-GL modules/steps "Characterise Hazard", "Evaluate Exposure", "Vulnerability Analysis", "Assess Risks and Impact" and "Identify Adaptation Options" as the actions to be carried out in these steps (except for "Characterise Hazard") are based on the respective “elements at risk types” or “elements at risk inventories”. The catalogue is capable of handling geo-data (e.g. points, lines, grids, political areas, etc.) which is especially relevant for the handling of the elements at risk (e.g. points for building locations, lines for roads/transport networks, grids for population densities, etc.” [2]

Up-to-date building block description and requirements:

<https://github.com/search?q=org%3Aclarity-h2020+la-bel%3A%22BB%3A+Catalogue+of+ER+%26+AO%22+sort%3Acreated-asc++sort%3Acreated-asc&type=issues>

Continuously updated (technology support) plan:

<https://github.com/clarity-h2020/csis/issues?q=la-bel%3A%22BB%3A+Catalogue+of+ER+%26+AO%22+sort%3Acreated-asc>

Source code and configuration files repository:

<https://github.com/clarity-h2020/emikat>

Deployed instance (access protected):

Deployed on AIT Servers.

As foreseen in the initial technology support plan (D4.1), the back-end of this building block is realised by AIT’s proprietary EMIKAT software system. The service instance is deployed on AIT servers. The front-end part is implemented as Drupal 8 Views in the Scenario Management building block (chapter 5). The relevant data, maps and other information is exchanged via REST-API and GeoServer.

The work in this task was extended to also include the calculation of impact and risk. It mainly focuses on the implementation of the following functionalities and services:

4.1.1 Software service for the calculation of impact and risk

For the calculation of impact and risk data from different maps/layers and parameters (“Hazard-Local Effects” layers, exposure layers) have to be combined by the impact calculation model for the respective project area of the user. Some rows (5 out of 15.900) of the result of an impact calculation using exemplary data is shown in Figure 12.

ImpactScenario_ID	Impact Scenario Description	VulnerabilityClass_ID	Vulnerability Class Name	Cell_ID	ExposedQuantity	DamageLevel1Probability	DamageLevel2
1	Population Mortality heatwave, 2041-2070, model 1	21	Age>65	500mE46615N19870	0,225	0,000000000156	0,0000000112
1	Population Mortality heatwave, 2041-2070, model 1	21	Age>65	500mE46615N19875	0,225	0,000000000156	0,0000000112
1	Population Mortality heatwave, 2041-2070, model 1	21	Age>65	500mE46615N19880	24,6	0,000000000156	0,0000000112
1	Population Mortality heatwave, 2041-2070, model 1	21	Age>65	500mE46615N19885	24,6	0,000000000156	0,0000000112
1	Population Mortality heatwave, 2041-2070, model 1	21	Age>65	500mE46615N19890	56,02	0,000000000156	0,0000000112

Figure 12: First 5 rows from the Result of an Impact Calculation carried out by EMIKAT (excerpt)

To create a human-readable form the detailed results of the calculation are aggregated by EMIKAT.

ImpactSc...	ImpactC...	HazardSc...	HazardInd...	HazardIndexType...	EaRTyp...	VulnerabilityClass_ID	VulnerabilityClassName	SUM_EXPOSEDQUANTITY	SUM_DL1...	SUM_DL2...	SU...
2	1	2	1	heat-wave:consecutive-summer-days	people	26	Age 15-65	1.249.549	0,3311	32,49	1...
1	1	1	1	heat-wave:consecutive-summer-days	people	26	Age 15-65	1.249.549	0,3311	32,49	1...
1	1	1	1	heat-wave:consecutive-summer-days	people	25	Age<14	499.820	0,01015	1,699	11...
2	1	2	1	heat-wave:consecutive-summer-days	people	25	Age<14	499.820	0,01015	1,699	11...
1	1	1	1	heat-wave:consecutive-summer-days	people	21	Age>65	749.730	0,0000117	0,008397	2...
2	1	2	1	heat-wave:consecutive-summer-days	people	21	Age>65	749.730	0,0000117	0,008397	2...

Figure 13: Aggregated impact results for the (excerpt)

4.1.2 Catalogue/inventory of elements at risk and their vulnerability classes

The actual data to be used for the impact calculation is part of the exposure layers of the respective data package. For the purpose of showing meaningful results to users, element at risk types are implemented.

EARTy...	Name	Acronym	QuantityUnit	Quantity unit (short)
1	people	POP	number of people	number
2	buildings	BUILD	number of buildings	number
3	buildings	BUILD	m² of buildings	m² of
4	critical infrastructure	INFR	number of critical infrastructure	number
5	transport infrastructure	ROAD	road length km	road l

Figure 14: EMIKAT table showing some Element at Risk Types.

VulnerabilityClass_ID	ImpactCase_ID	Impact Case Name	Name	Acronym	Description	...
26	1	people - heat-wave 36°	Age 15-65	C_p	Age group 15-...	
25	1	people - heat-wave 36°	Age<14	B_p	Age group 0-14	
21	1	people - heat-wave 36°	Age>65	A_p	Age group >65	
24	4	people - flood:extreme-precipitation:rx5day	Age>65	A_p	Age group >65	
22	5	people - extreme-heat:summer-days	Age>65	A_p	Age group >65	
23	6	people - extreme-heat:tropical-nights	Age>65	A_n	Age group >65	

Figure 15: EMIKAT table showing some vulnerability classes (excerpt)

The Drupal front-end manages the element at risk types and their vulnerability classes (Figure 16).

Exposure elements

Hazards	Element at risk	Classes	values	Unit
▼ HW,PF,FL (1)	Population			
	▼ Population (3)	Age group 0-14, Age group 15-64, >65	, ,	pop/km2, pop/km2, po...
		Age group 0-14		pop/km2
		Age group 15-64		pop/km2
		>65		pop/km2
▼ PF,FL (2)	Buildings, Infrastructure			
	▼ Buildings (4)	Continuous Residential (S.L. > 80%), Med-Hi Density Discon...	, , ,	m3/m2, m3/m2, m3/m...
		Continuous Residential (S.L. > 80%)		m3/m2
		Med-Hi Density Discontinuous Res. (30% < S.L. < 80%)		m3/m2
		Low Density Discontinuous Res. (S.L. < 30%)		m3/m2
		Non Residential		m3/m2
	▼ Infrastructure (2)	Roads, Railways	,	ml, ml
		Roads		ml
	Railways		ml	

Figure 16: CSIS table “Exposure Elements” (EU-GL step “Evaluate Exposure”)

4.1.3 Catalogue/inventory of damage level probability functions

The damage level probability functions are part of and described by the data package and are used by EMIKAT to calculate impact for each grid cell and damage level.

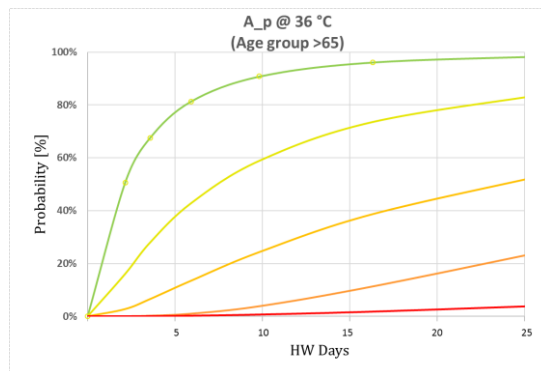


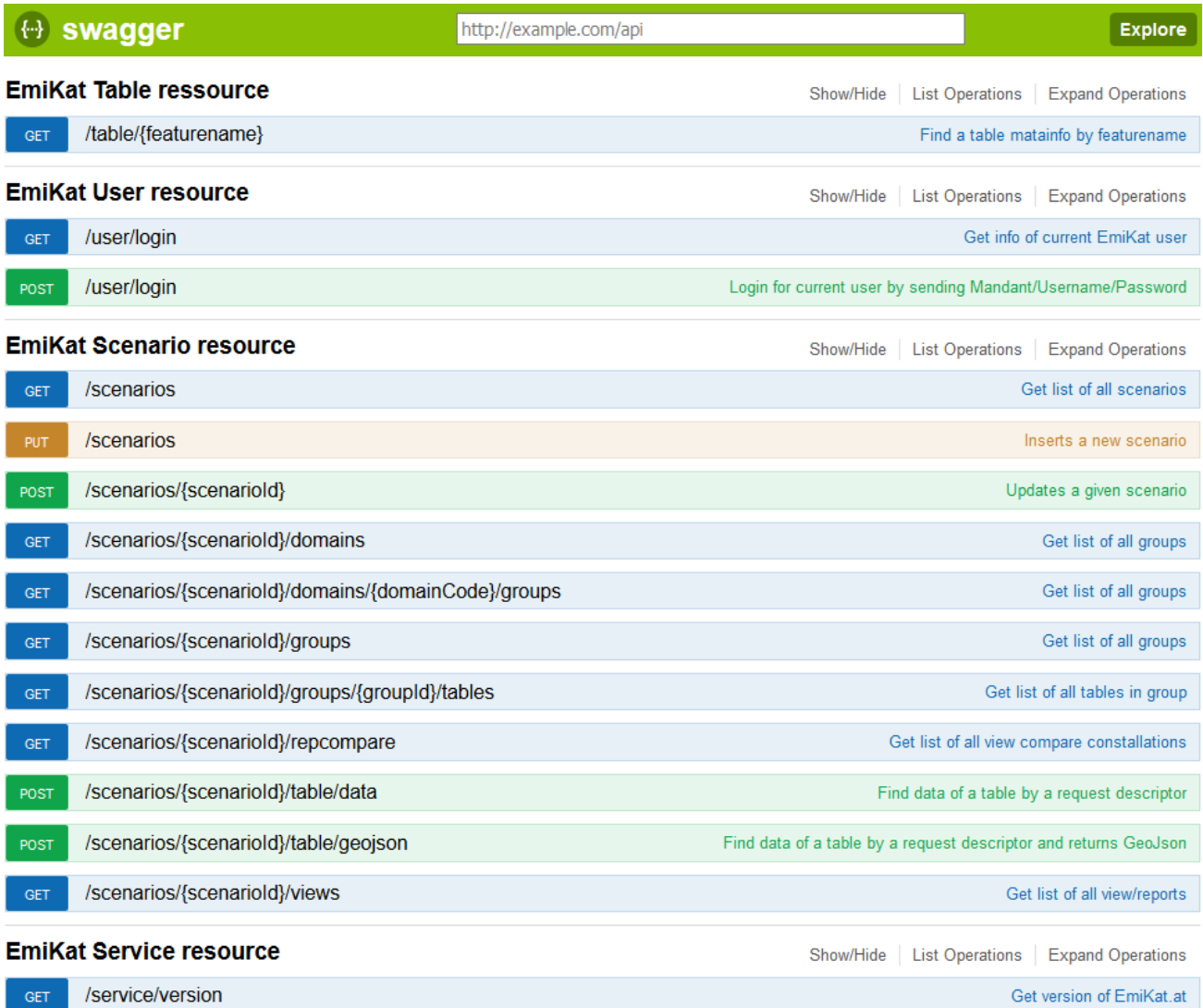
Figure 17: Damage level probability function example for heat wave

Function_ID	ImpactCase_ID	Impact Case Name	VulnerabilityClass_ID	Vulnerability Class Name	Description	Damage Level 1	Damage Level 2	Damage L
13	1	people - heat-wave ...	26	Age 15-65	people - heat-wave 36° - Age 15-65	5 0.8230896 0.70038170.56146880...	5 0.1634086 0...	5 0.01297...
12	1	people - heat-wave ...	25	Age<14	people - heat-wave 36° - Age<14	5 0.7413716 0.5614687 0.3682748 ...	5 0.2286326 0...	5 0.02820...
14	1	people - heat-wave ...	21	Age>65	people - heat-wave 36° - Age>65	5 0.6639696 0.4166817 0.2022538 ...	5 0.2833546 0	5 0.04837...

Figure 18: Damage level probability functions as stored in EMIKAT

4.1.4 Interface between EMIKAT and Drupal

When impact/risk is calculated, the CSIS front-end sends a request to the Impact/Risk Model Service (EMIKAT + REST interface) with the URI of the current object of study. The Impact/Risk Model Service needs to authenticate and retrieves the study object JSON representation. Figure 19 shows some of the functionalities of the EMIKAT REST-API. Swagger provides documentation and testing possibilities of the REST-API.



The screenshot displays the Swagger UI for the EmiKat REST-API. At the top, there is a green header with the Swagger logo, a text input field containing 'http://example.com/api', and an 'Explore' button. Below the header, the API is organized into four resource sections, each with a title, 'Show/Hide', 'List Operations', and 'Expand Operations' links, and a list of endpoints with their methods and descriptions.

- EmiKat Table resource:**
 - GET /table/{featurename} Find a table matainfo by featurename
- EmiKat User resource:**
 - GET /user/login Get info of current EmiKat user
 - POST /user/login Login for current user by sending Mandant/Username/Password
- EmiKat Scenario resource:**
 - GET /scenarios Get list of all scenarios
 - PUT /scenarios Inserts a new scenario
 - POST /scenarios/{scenariold} Updates a given scenario
 - GET /scenarios/{scenariold}/domains Get list of all groups
 - GET /scenarios/{scenariold}/domains/{domainCode}/groups Get list of all groups
 - GET /scenarios/{scenariold}/groups Get list of all groups
 - GET /scenarios/{scenariold}/groups/{groupId}/tables Get list of all tables in group
 - GET /scenarios/{scenariold}/repcompare Get list of all view compare constallations
 - POST /scenarios/{scenariold}/table/data Find data of a table by a request descriptor
 - POST /scenarios/{scenariold}/table/geojson Find data of a table by a request descriptor and returns GeoJSON
 - GET /scenarios/{scenariold}/views Get list of all view/reports
- EmiKat Service resource:**
 - GET /service/version Get version of EmiKat.at

Figure 19: EMIKAT REST-API functionalities

Since EMIKAT is one of the tools inherited from previous projects, more information using and/or extending this software component is given in Annex 1.

4.2 Catalogue of Data Sources and Simulation Models

“The Catalogue of Data Sources and Simulation Models is a meta-data catalogue that makes climate-related information accessible by providing functionalities to streamline publishing, sharing, finding and using data and models. The catalogue can be used for data discovery and meta-data storage by different climate services and building blocks, respectively.” [2] Additionally, this catalogue is used for the implementation of the deliverables D2.2 “Catalogue of local data sources and sample datasets” and D7.9 “Data Management Plan v2”.

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Catalogue-of-Data-Sources-and-Simulation-Models>

Continuously updated (technology support) plan:

<https://github.com/clarity-h2020/ckan/issues?q=sort%3Acreated-asc>

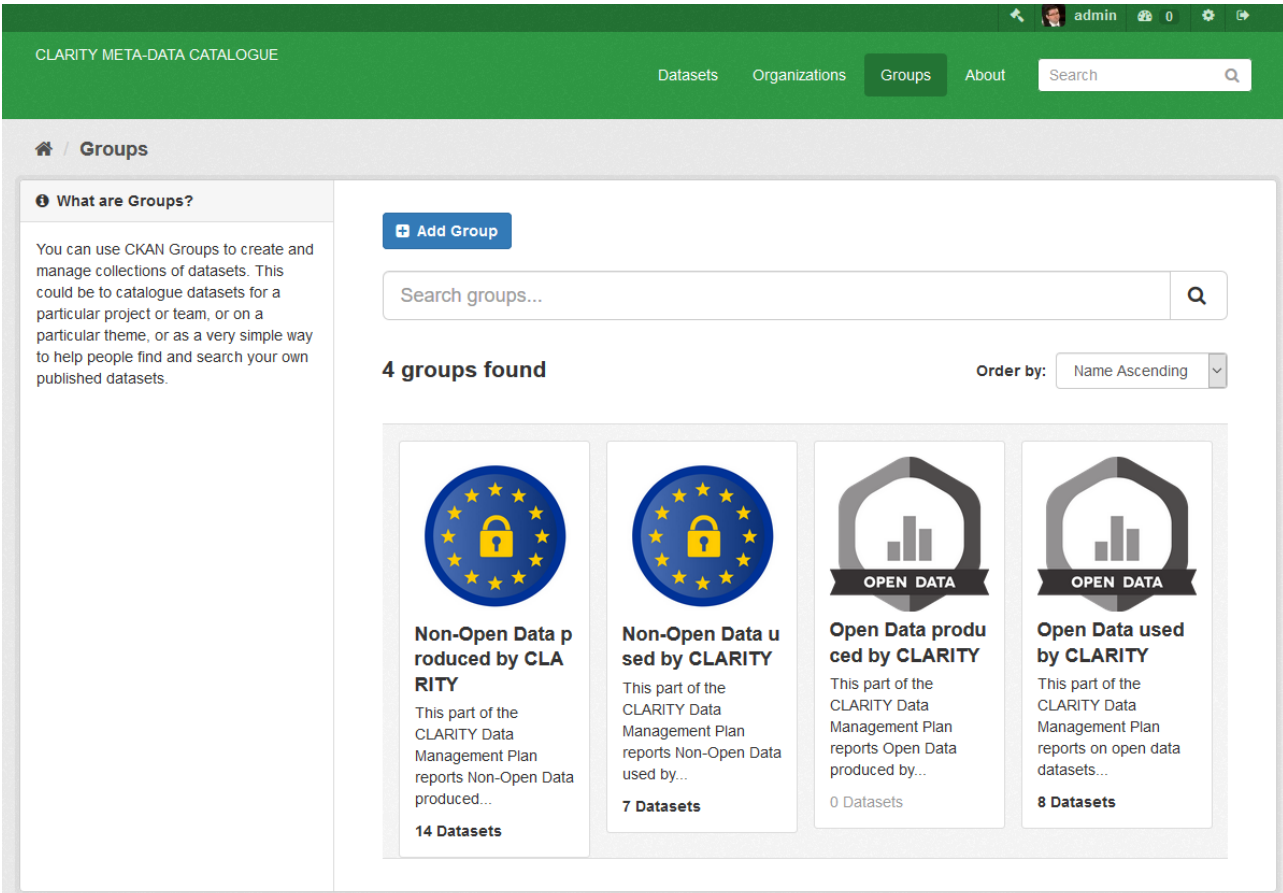
Source code and configuration files repository:

<https://github.com/clarity-h2020/ckan/>

Deployed instance:

<https://ckan.myclimateservice.eu/>

As foreseen in the initial technology support plan, this building block is realised by the open-source cataloguing software CKAN. To satisfy the functional requirements of this building block, WP4 created a CLARITY-specific configuration (branding, meta-data model, categories, organisations, documentation, etc.) and installed configured the respective plugins and add-ons. These configurations are available as open-source in a public GitHub repository that was forked from the original ckan repository. No direct changes to the source code of CKAN where necessary. The actual service instance (Figure 20) is deployed as Docker container in CLARITY's cloud infrastructure.



CLARITY META-DATA CATALOGUE

Datasets Organizations **Groups** About Search

Home / Groups

What are Groups?

You can use CKAN Groups to create and manage collections of datasets. This could be to catalogue datasets for a particular project or team, or on a particular theme, or as a very simple way to help people find and search your own published datasets.

Add Group

Search groups... Q

4 groups found Order by: Name Ascending

Group Name	Description	Dataset Count
Non-Open Data produced by CLARITY	This part of the CLARITY Data Management Plan reports Non-Open Data produced...	14 Datasets
Non-Open Data used by CLARITY	This part of the CLARITY Data Management Plan reports Non-Open Data used by...	7 Datasets
Open Data produced by CLARITY	This part of the CLARITY Data Management Plan reports Open Data produced by...	0 Datasets
Open Data used by CLARITY	This part of the CLARITY Data Management Plan reports on open data datasets...	8 Datasets

Figure 20: Catalogue of Data Sources and Simulation Models

4.3 Table Component

The Table Component is a reusable and configurable building block for the tabular visualisation of hazard (Figure 21), exposure, vulnerability and impact (Figure 23) Data. The table shows aggregated and normalised values for respective hazards, elements at risk, etc. relative to the selected study area. Aggregation and normalisation of data is not performed by the Table Component itself, but by separate web services that transform the original data to standardised JSON format supported by the Table Component.

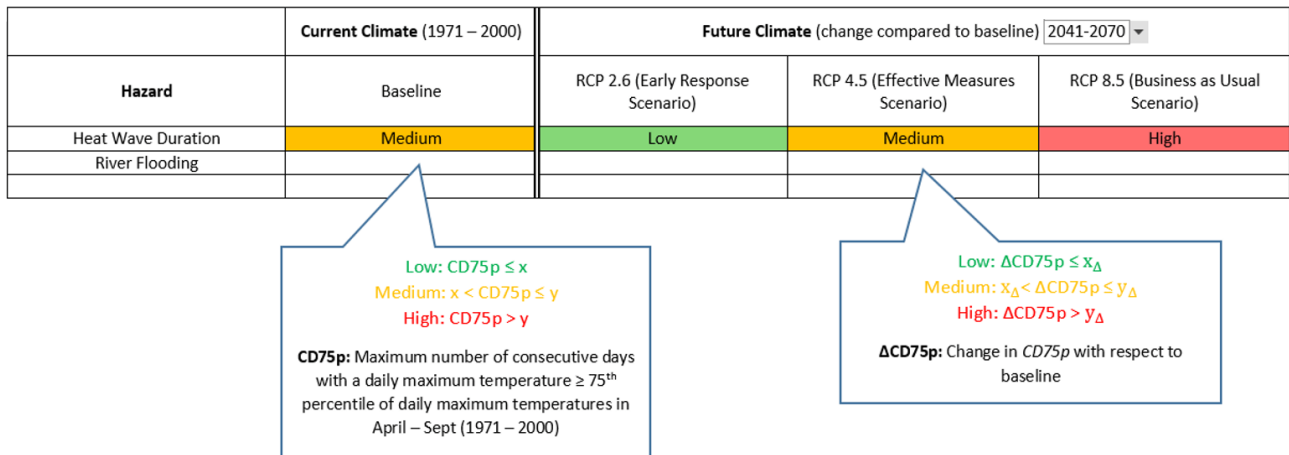


Figure 21: Table Component mock-up (hazard indexes)

Up-to-date building block description and requirements:
<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Table-Component>

Continuously updated (technology support) plan:
<https://github.com/clarity-h2020/table-components/issues?q=%20>

Source code and configuration files repository:
<https://github.com/clarity-h2020/table-components>

Deployed instance:
<https://ckan.myclimateservice.eu/>

The Table Component is a new building block that was not considered in D4.1 “Technology Support Plan” [3]. Therefore, the plan for implementing this building block (Figure 22) is briefly summaries in this document

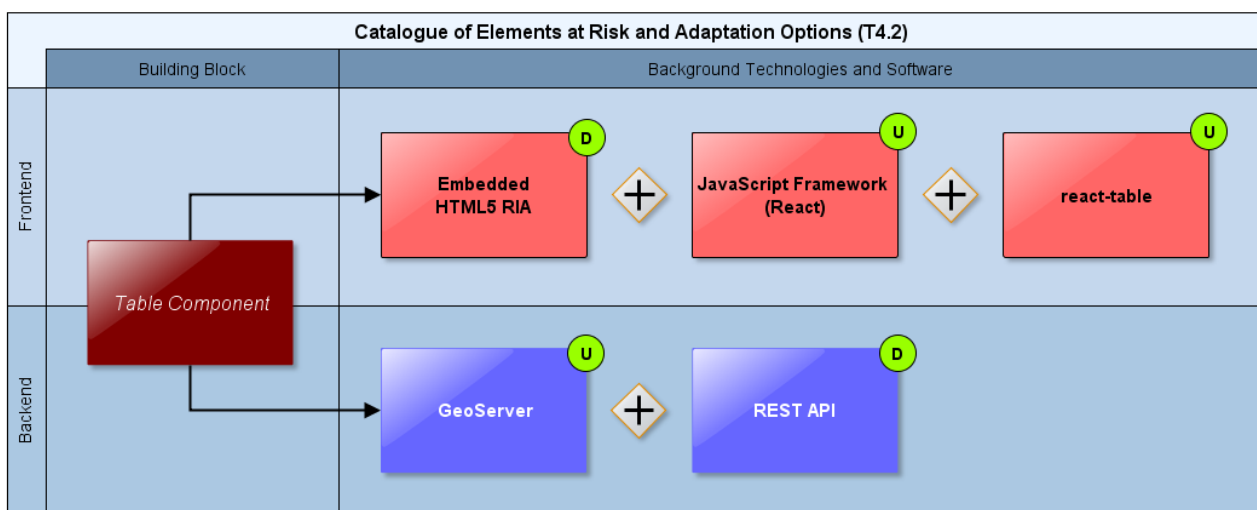


Figure 22: Table Component Technology Support

The high interactivity and flexibility expected from this component requires an approach based on responsive and highly adaptable technologies. This approach can be achieved using client-side rendering along with libraries and tools that have already proved their usability and popularity, meaning that a big and active community is supporting their development and use. To ensure this high interactivity approach of this web application a good approach would be to use React, React allows rich site interactions, fast website rendering

after the initial load, and a good selection of JavaScript libraries. It is also designed to build encapsulated components that can be composed to make complex UIs. In consequence, the Table Component is developed as independent HTML5/AJAX RIA that is loosely embedded with help of the React Mount Node Drupal module (https://www.drupal.org/project/reactjs_mount) in the UI Integration Platform (Drupal 8) and relies as backend on Data repository and various OGC services, respectively. In terms of an open-source solution for the table itself, react-table (<https://react-table.js.org/>), a lightweight, fast and extendable data grid built for React, is used for the implementation.

Hazards	Element at risk (...)	Vulnerability clas...	Unit	Damage Classes			
				D1	D2	D3	D4
HW	Population	Age group 0-14	pop/km2	1	3	2	3
HW	Population	Age group 15-64	pop/km2	1	3	2	3
HW	Population	>65	pop/km2	1	3	2	3
FL	Population	Age group 0-14	pop/km2	1	3	2	3
FL	Population	Age group 15-64	pop/km2	1	3	2	3

Previous Page 1 of 3 5 rows Next

Figure 23: Table Component (impact)

The open source code of this building block is maintained in a dedicated GitHub repository. Since it is directly integrated into the CSIS Drupal Platform as explain in UI Integration Concept (chapter of D4.2 “CLARITY CSIS Architecture” [2]) it is deployed together with the CSIS Docker container in CLARITY’s cloud infrastructure.

5 Scenario Management (T4.3)

The project board of T4.3 is available at <https://github.com/orgs/clarity-h2020/projects/8>.

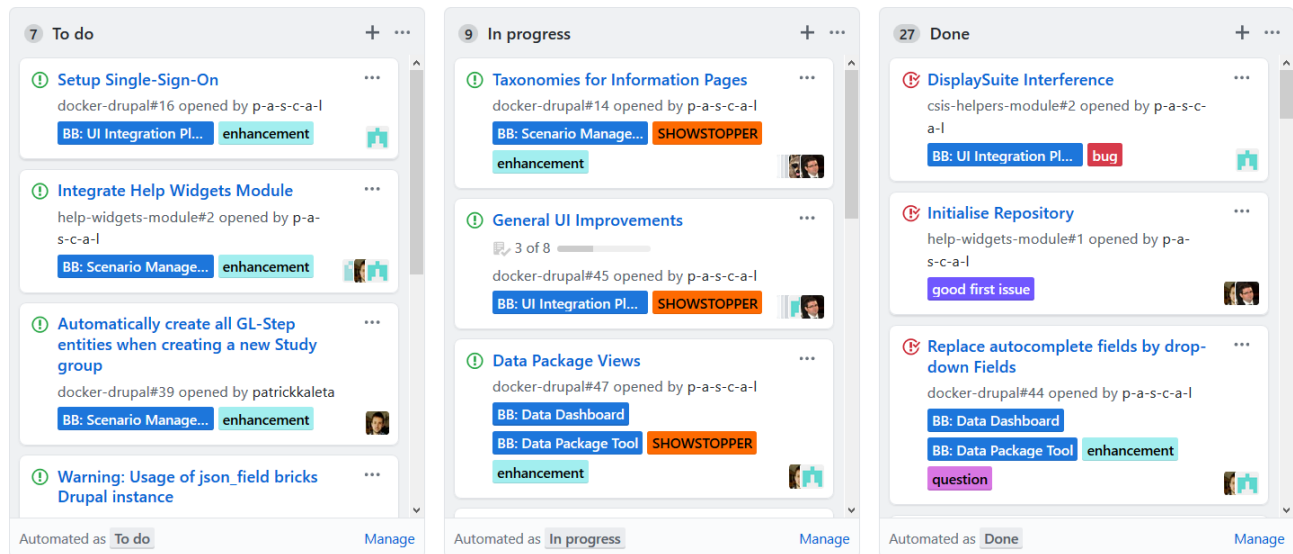


Figure: T4.3 project board (excerpt)

5.1 Data Dashboard

The Data Dashboard building block provides an overview of all the different datasets that are used, produced, ordered, collected, requested, exchanged etc. by an end user (e.g. project planner or climate resilience manager) during an assessment (planning session). Datasets are organised according to their relation to the modules of the EU-GL (e.g. hazard maps, impact scenario results, elements of risk inventory). [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Data-Dashboard>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+sort%3Acreated-asc+label%3A%22BB%3A+Data+Dashboard%22&type=issues>

Drupal configuration repository (private):

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

Deployed instance:

<https://csis.myclimateservice.eu/>

The initial technology support plan mentioned two possibilities for implementing the frontend part of this building block: As independent HTML5/AJAX RIA that is loosely embedded into the CSIS main site (Drupal) following user interface integration concept described in chapter 4.2.5 of D4.2 “CLARITY CSIS Architecture”, or as dedicated Drupal form that is tightly coupled to the UI Integration Platform. Since the Data Dashboard is closely related to the Data Package Export and Import Tool (chapter 5.2) and thus to the Data Package specification (<https://github.com/clarity-h2020/data-package>), the WP4 team decided to use the same frontend technologies for both building block. The technology support plan (Figure 4) has been updated accordingly.

Since both building blocks are realised as Drupal forms (Figure 25), they benefit not only from the rich functionality offered to a Drupal form and the powerful Drupal API, but are also able to reuse the respective entity types and view mode configurations in both building blocks. These configurations are stored in a common private git repository offered by Atos Research & Innovation department. Since these configurations are part of the overall CSIS Drupal configuration, they are deployed as part of the CSIS Drupal Docker container in CLARITY’s cloud infrastructure.

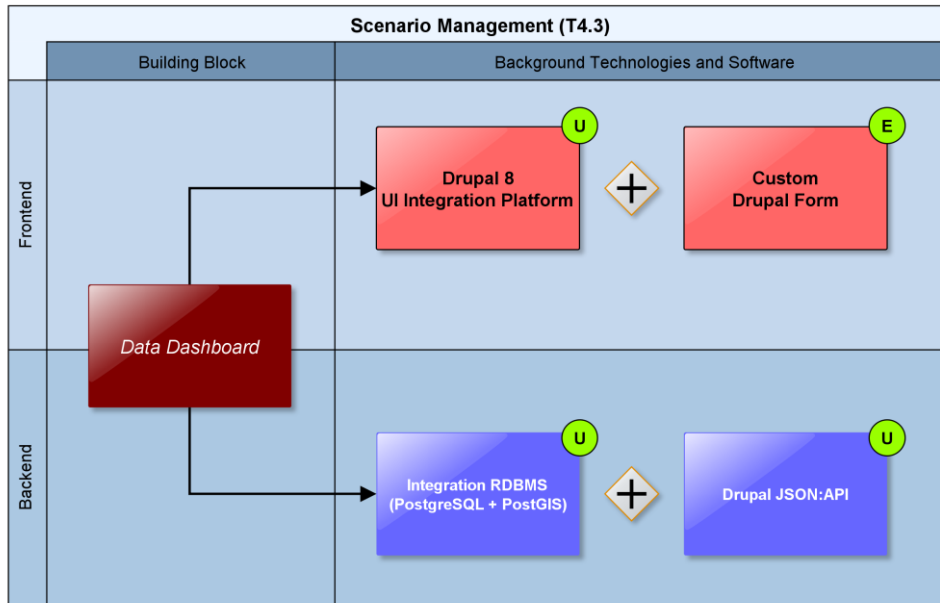


Figure 24: Updated Data Dashboard technology support

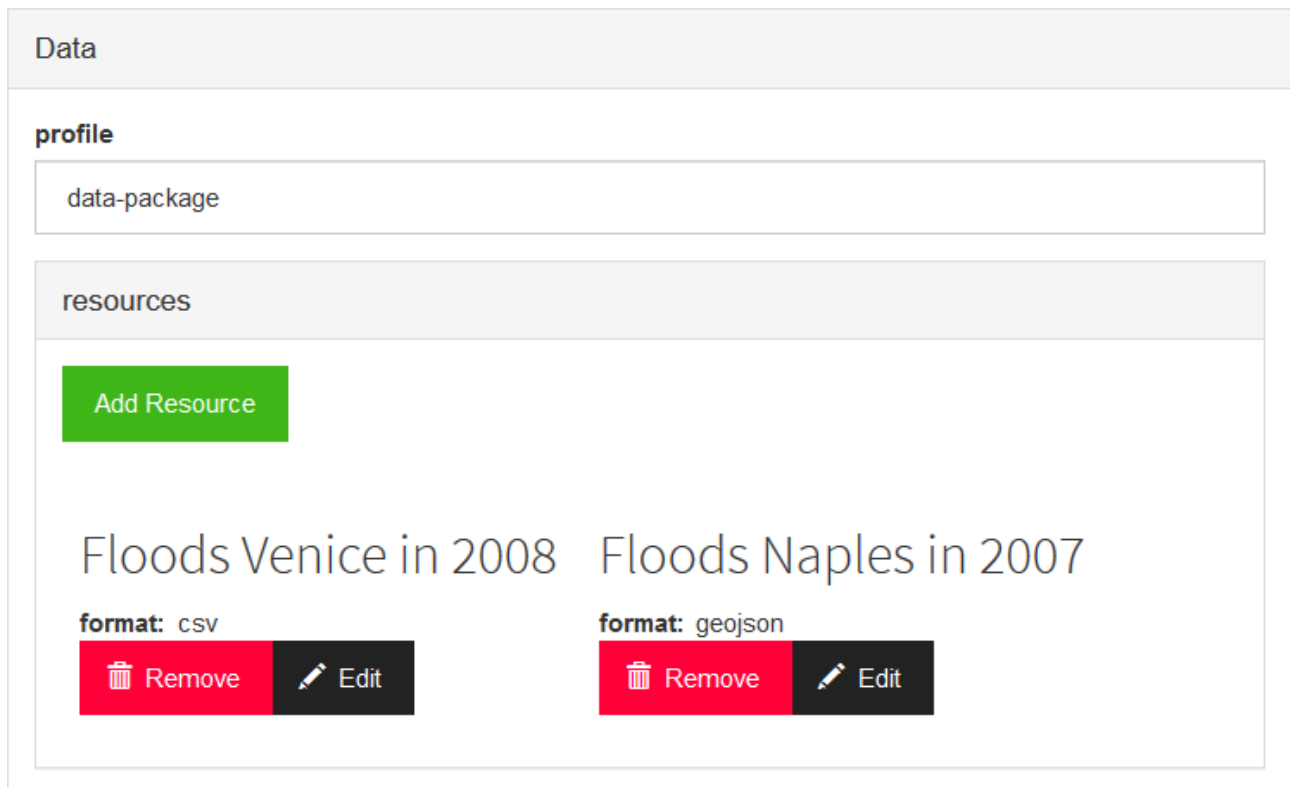


Figure 25: Data Dashboard prototype

5.2 Data Package Export and Import Tool

“The Data Package Export and Import building block is a tool that can be used at any stage of the adaptation planning process to export (download) any data that is directly available in the CSIS in standardised format, the CLARITY Data Package. It can furthermore be used to import additional Data Packages prepared by experts and available from the Marketplace.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Data-Package-Export-and-Import-Tool>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+sort%3Acreated-asc+label%3A%22BB%3A+Data+Package+Tool%22&type=issues>

Data Package Specification repository:

<https://github.com/clarity-h2020/data-package>

Drupal configuration repository (private):

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

Experimental QGIS 3 plugin repository:

<https://github.com/clarity-h2020/data-package-builder>

Deployed instance:

<https://csis.myclimateservice.eu/>

As explained in chapter 5.1, both this building block and the Data Dashboard are realised as Drupal forms. However, the Data Package Export and Import Tool building blocks follow thereby a hybrid approach. Although the management of a Data Package’s meta-data can easily be achieved using the facilities of the Drupal 8 platform, deriving such meta-data that follows CLARITY’s Data Package standards from the original data sources is still a complex task. To support providers of Data Packages in the creation of such and to lower the entry hurdles for the myclimateservices.eu marketplace, WP4 is exploring the possibility to develop a qGIS plugin for this purpose. The technology support plan (Figure 26) has been updated accordingly.

Apart from this, the Drupal-specific configurations of this building block are stored in a common private git repository offered by Atos Research & Innovation department. Since these configurations are part of the overall CSIS Drupal configuration, they are deployed as part of the CSIS Drupal Docker container in CLARITY’s cloud infrastructure. The specification of the Data Package standard is made available as open specification in a public GitHub repository. The same is true for the envisaged qGIS plugin.

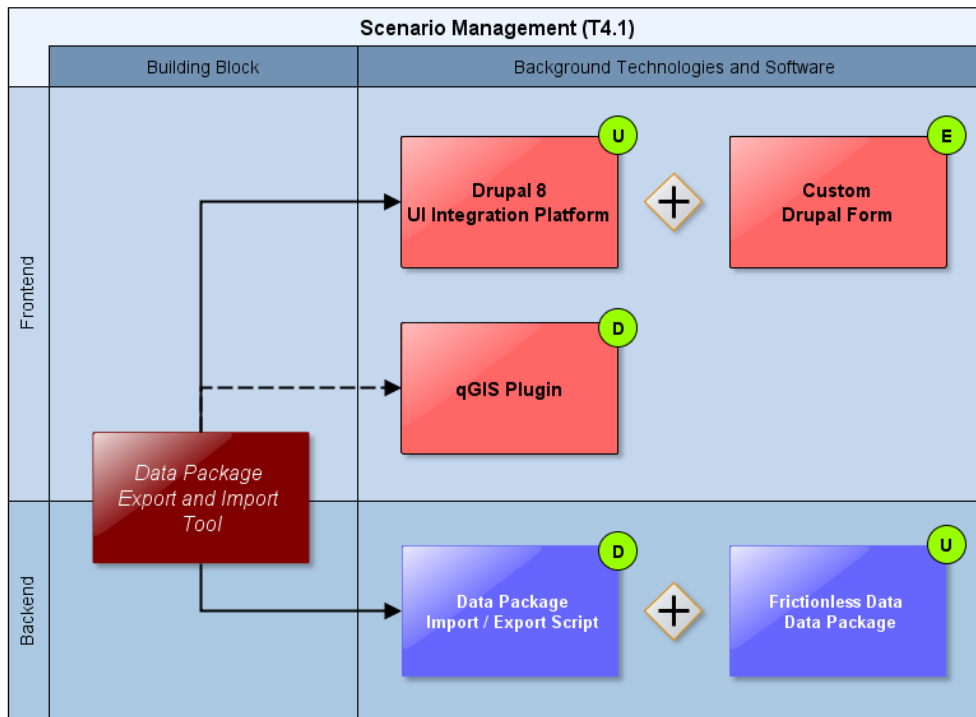


Figure 26: Updated Data Package Export and Import Tool technology support

The screenshot shows a web interface titled "Edit entity Floods in Naples and Venice during 2007 and 2008". The form contains the following fields and instructions:

- id*:** A text input field. Below it, instructions state: "Use a URL as a means for ensuring global uniqueness of the data package id. The following examples would be valid data package identifiers :
 - A URL that points directly to the datapackage.json file: <http://data.myclimateservice.eu/datapackages/clarity-dc4.json>
 - A URL that points directly to the datapackage: <http://github.com/clarity-h2020/datapackages/clarity-dc4>
 - A GitHub URL: <http://github.com/clarity-h2020/datapackages/clarity-dc4>
- name*:** A text input field. Below it, instructions state: "A short url-usable (and preferably human-readable) name of the package. This MUST be lower-case and contain only alphanumeric characters along with ".", "_", or "-" characters. It will function as a unique identifier and therefore SHOULD be unique in relation to any registry in which this package will be deposited (and preferably globally unique)."
- version*:** A text input field containing "1.0.0". Below it, instructions state: "A version string identifying the version of the package. It should conform to the Semantic Versioning requirements (<http://semver.org/>) and should follow the Data Package Version pattern (<https://frictionlessdata.io/specs/patterns/#data-package-version>): MAJOR.MINOR.PATCH (e.g., 1.0.0)".
- title*:** A text input field.
- description*:** A text input field.

A green "Save" button with a checkmark is located at the bottom right of the form.

Figure 27: Data Package Export and Import Tool (excerpt)

5.3 Scenario Management

“The Scenario Management building block supports and enforces first and foremost the standardised workflows of the EU-GL [1] for each of the distinct planning steps and provides respective user interfaces that guide the user through the process of co-creating a climate adaptation study. Basically, the end user is presented with the list of recommended and required steps for performing a complete climate adaptation study for the respective (infrastructure) project under assessment and is asked to provide the information that is needed to complete the current step and advance to the next step.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Scenario-Management>

Continuously updated (technology support) plan:

<https://github.com/search?o=desc&q=org%3Aclarity-h2020+label%3A%22BB%3A+Scenario+Management%22&type=issues>

Drupal configuration repository (private):

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

CSIS Helpers Drupal module repository:

<https://github.com/clarity-h2020/csis-helpers-module>

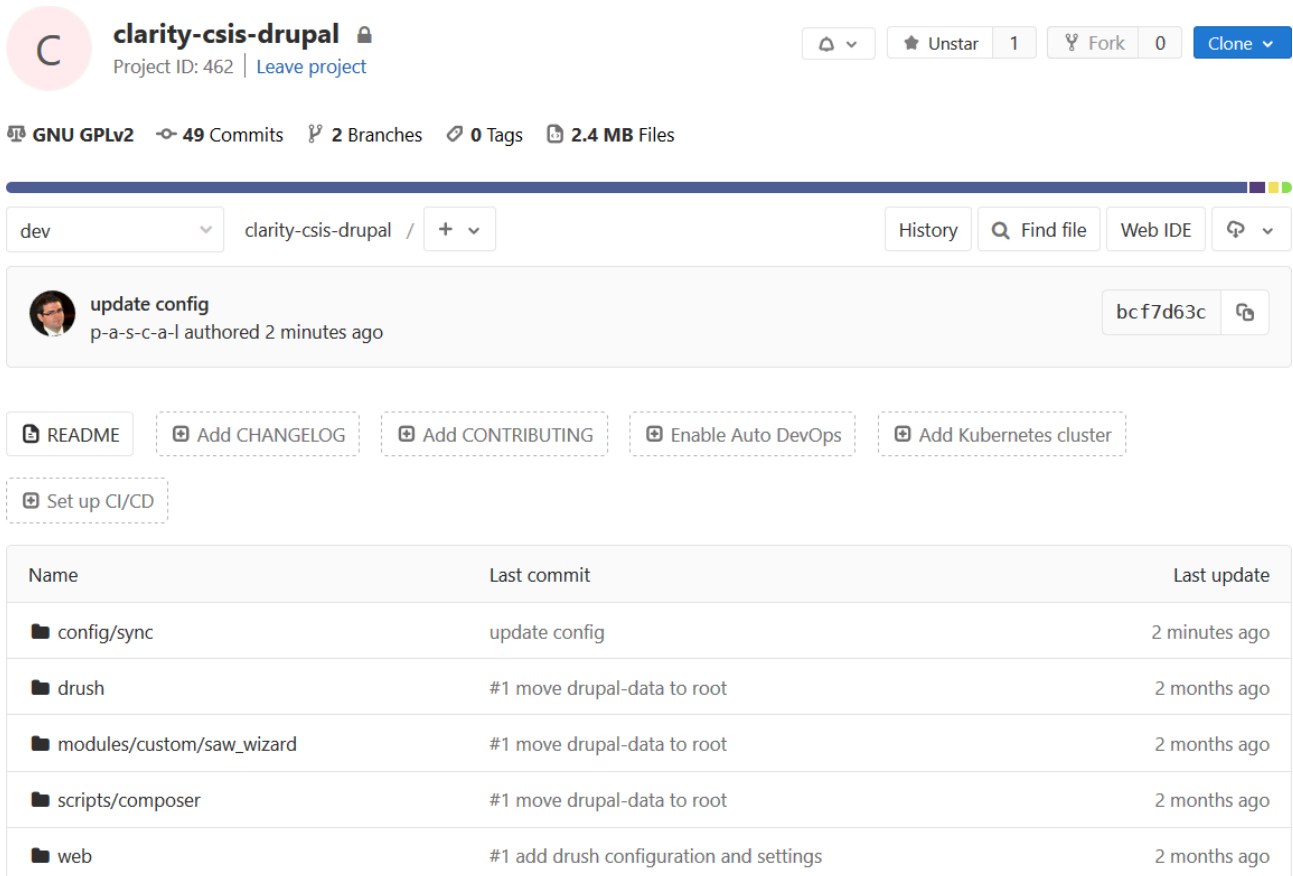
Help Widget Drupal module repository:

<https://github.com/clarity-h2020/csis-helpers-module>

Deployed instance:

<https://csis.myclimateservice.eu/>

The Scenario Management building block represents the overall workflow tool that guides the user through the different steps of the EU-GL methodology. Thus, it is tightly coupled with the overall UI Integration Platform (7.5) and needs to integrate and interact with nearly all other building blocks of the CSIS. As foreseen in the initial technology support plan, the fronted part of this building block is therefore realised by Drupal forms and custom Drupal modules. An example of such user interface that communicates with the EMIKAT REST API (Figure 19) is shown in chapter 4.1 in Figure 16.



The screenshot shows a GitLab repository page for 'clarity-csis-drupal'. The repository is public and has 1 star, 0 forks, and 49 commits. The current branch is 'dev'. A recent commit titled 'update config' by user 'p-a-s-c-a-l' is shown, committed 2 minutes ago. Below the commit, there are several suggested actions: 'Add README', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Enable Auto DevOps', 'Add Kubernetes cluster', and 'Set up CI/CD'. A table lists the repository's structure and the last commit for each directory.

Name	Last commit	Last update
config/sync	update config	2 minutes ago
drush	#1 move drupal-data to root	2 months ago
modules/custom/saw_wizard	#1 move drupal-data to root	2 months ago
scripts/composer	#1 move drupal-data to root	2 months ago
web	#1 add drush configuration and settings	2 months ago

Figure 28: Private CSIS configuration GitLab repository

The creation of the respective view forms, editor forms and the related entity types in Drupal can be considered an administrative task that does not involve actual software development. Nevertheless, the resulting configuration is rather complex and exhaustive and thus managed in a decided version controlled code repository (Figure 28). This applies also to the other building blocks like the Data Dashboard (chapter 5.1) that are realised as Drupal components. Thus, a common git repository operated by Atos Research & Innovation department is used for the management of the overall CSIS Drupal Configuration and the configurations of all related Drupal-based building blocks. Since these configurations may include also sensitive data like API keys and credentials for accessing the connected data repositories and services, the repository is private and only accessible by authorised users.

CSIS Helpers Drupal Module Edit

building-block new-component drupal-8 drupal-module Manage topics

9 commits 2 branches 1 release 2 contributors GPL-3.0

Branch: dev New pull request Create new file Upload files Find file Clone or download

File	Commit Message	Time Ago
js	creation of drupoalSettings for entity info moved from hook preproces...	a day ago
src/Plugin/Field	added new custom field Snapshot	11 days ago
.gitignore	added entity browser helpers	2 months ago
LICENSE	added entity browser helpers	2 months ago
README.md	added entity browser helpers	2 months ago
csis_helpers.info.yml	added entity browser helpers	2 months ago
csis_helpers.libraries.yml	creation of drupoalSettings for entity info moved from hook preproces...	a day ago
csis_helpers.module	creation of drupoalSettings for entity info moved from hook preproces...	a day ago

Figure 29: CSIS Helpers module GitHub repository

While a large part of the functionality offered by the Drupal 8 platform meets functional requirements (refer to chapter 4.3.1 of D4.1 “Technology Support Plan” [3] for more details) of this building block, there is still the need to extend the Drupal platform. As foreseen in the initial technology support plan, so called Drupal modules are developed for this purpose. Currently, WP4 is providing the “CSIS Helpers” and “Help Widget” modules, which are released as open source software and maintained in dedicated GitHub repositories (Figure 29). Both the configurations and the modules are deployed as part of the CSIS Drupal Docker container in CLARITY’s cloud infrastructure.

6 Scenario Transferability (T4.4)

The project board of T4.4 is available at <https://github.com/orgs/clarity-h2020/projects/9>.

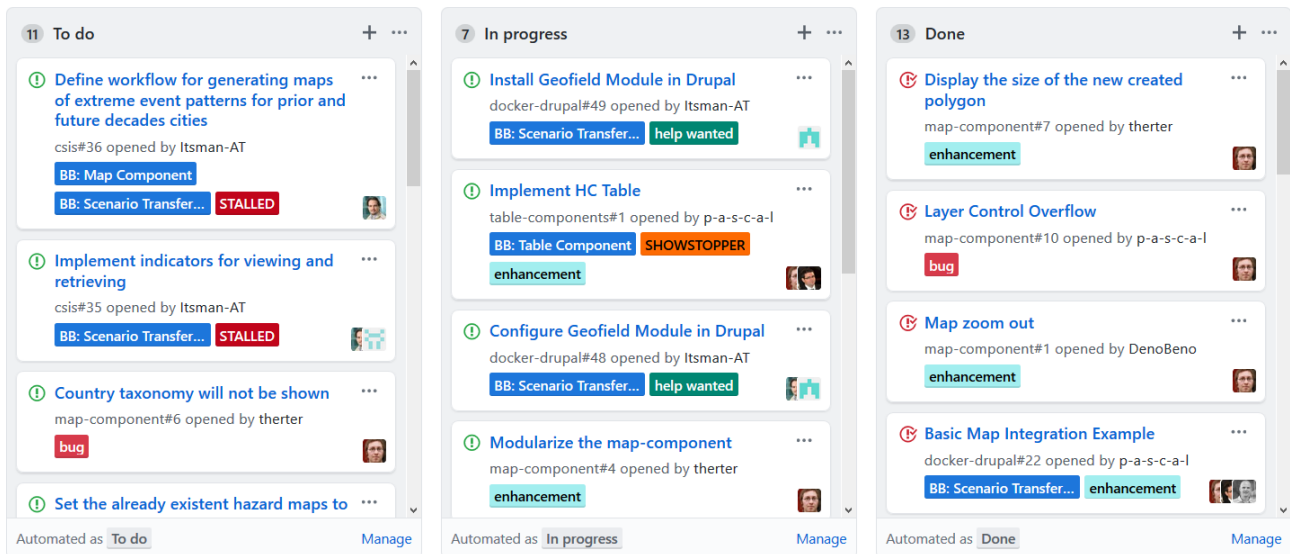


Figure: T4.3 project board (excerpt)

6.1 Map Component

“The Map Component is understood as a reusable, flexible and highly configurable building block meant to be used throughout CSIS. It is envisioned as an embeddable component that can be easily adapted to different parts of the common CSIS UI. The core functionalities of this component must be a clear and easy visualization of different maps and layers. It is also a key feature of the map component to allow for a degree of interactivity meant to enable users to better define locations, elements at risk, hazards, scenario results, etc.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Map-Component>

Continuously updated (technology support) plan:

<https://github.com/clarity-h2020/map-component/issues?q=%20>

Source code and configuration files repository:

<https://github.com/clarity-h2020/map-component>

Deployed instance:

<https://csis.myclimateservice.eu/>

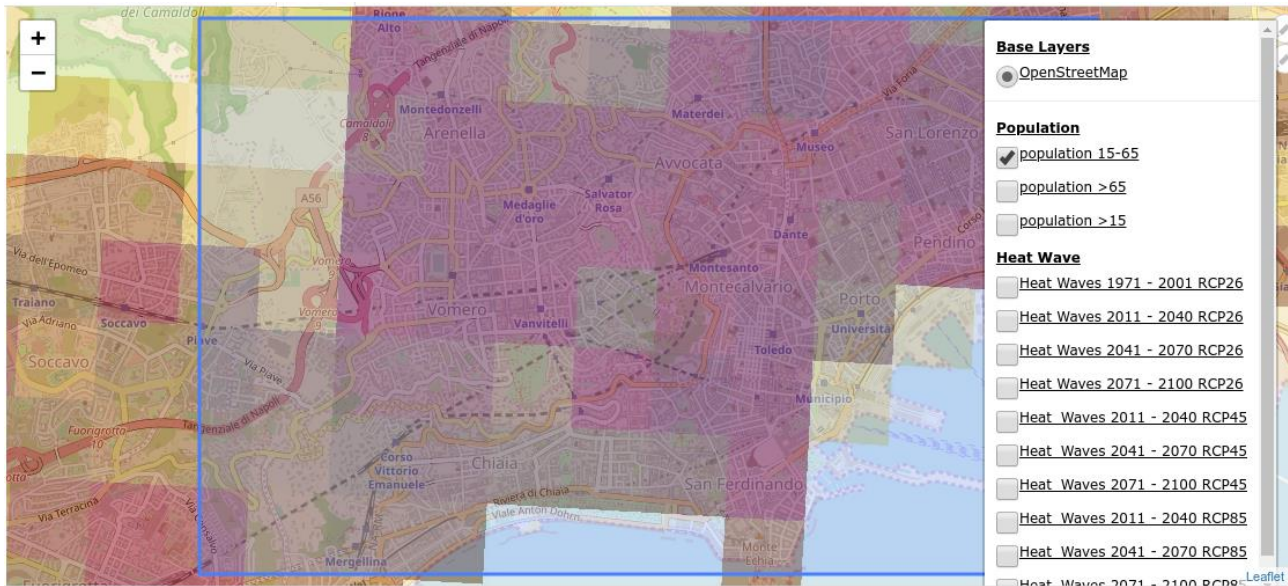


Figure 30: Map Component

Following the initial technology support plan, the Map Component is implemented as RIA on basis of the ReactJS JavaScript framework and the LeafletJS map visualisation library. Integration into the CSIS Drupal 8 system is performed in the same manner as for the Table Component (chapter 4.3). Accordingly, it is deployed together with the CSIS Docker container in CLARITY's cloud infrastructure. The source code of this building block is released under an open source license and maintained in a dedicated GitHub repository.

6.2 Scenario Transferability Component

The Scenario Transferability Component can be used for discovery and matchmaking of related entities like scenarios, projects, elements at risk, adaptation options, etc. For example, applied to the Catalogue of Elements at Risk and Adaptation Options (chapter 4.1), infrastructure projects being assessed by end users can be matched to other projects that share the same elements at risk (covering a variety of sectors). It can also be used for (visual) scenario analysis and comparison. Thereby, it allows the side-by-side comparison not only of different climate scenarios (Climate Twins Concepts), but also the comparison of alternate adaptation scenarios resulting from Impact Scenarios.

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Scenario-Transferability-Component>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+label%3A%22BB%3A+Scenario+Transferability%22+sort%3Acreated-asc&type=issues>

Source code and configuration files repository⁶:

n/a

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

⁶ For the Geofield Module no source code will be available since the Scenario Transferability Component will be configured within the CSIS platform in a first step

Deployed instance:

<https://csis.myclimateservice.eu/>

From the two usage scenarios, “Matchmaking” and “Scenario Analysis and Comparison” in the initial technology support plan the “Matchmaking” scenario has been evaluated and is currently under development. In a first step, it will be using the Drupal module “Geofield Map” (Figure 31) which is using Leaflet as map component and is capable of storing MULTIPOINT information.

In a next step, it is planned to fork the Map Component (chapter 6.1) and host the fork code in a dedicated GitHub repository.

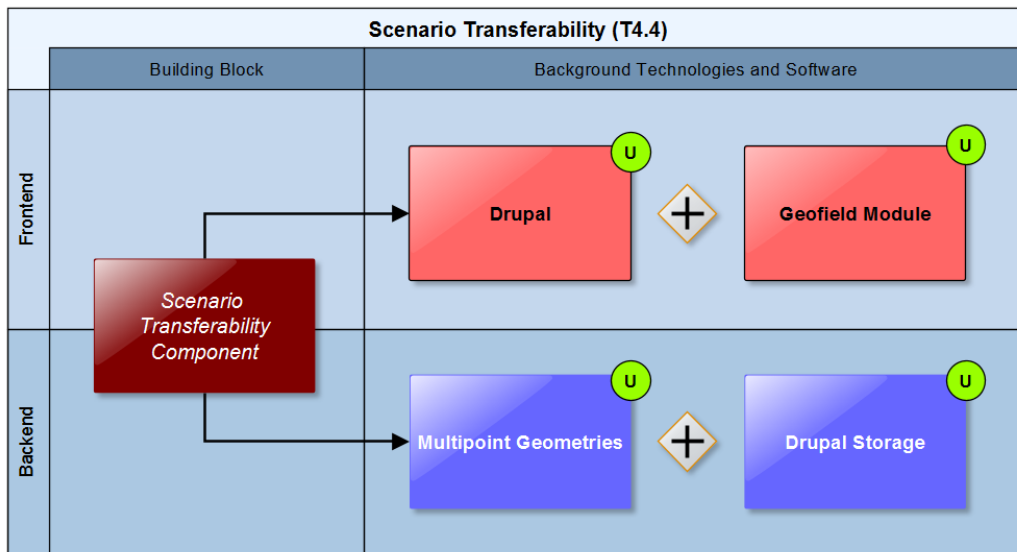


Figure 31: Update Scenario Transferability Component technology support

Since the Scenario Transferability Component is one of the tools and technologies inherited from previous projects, more information using and/or extending this software component is given in Annex 1.

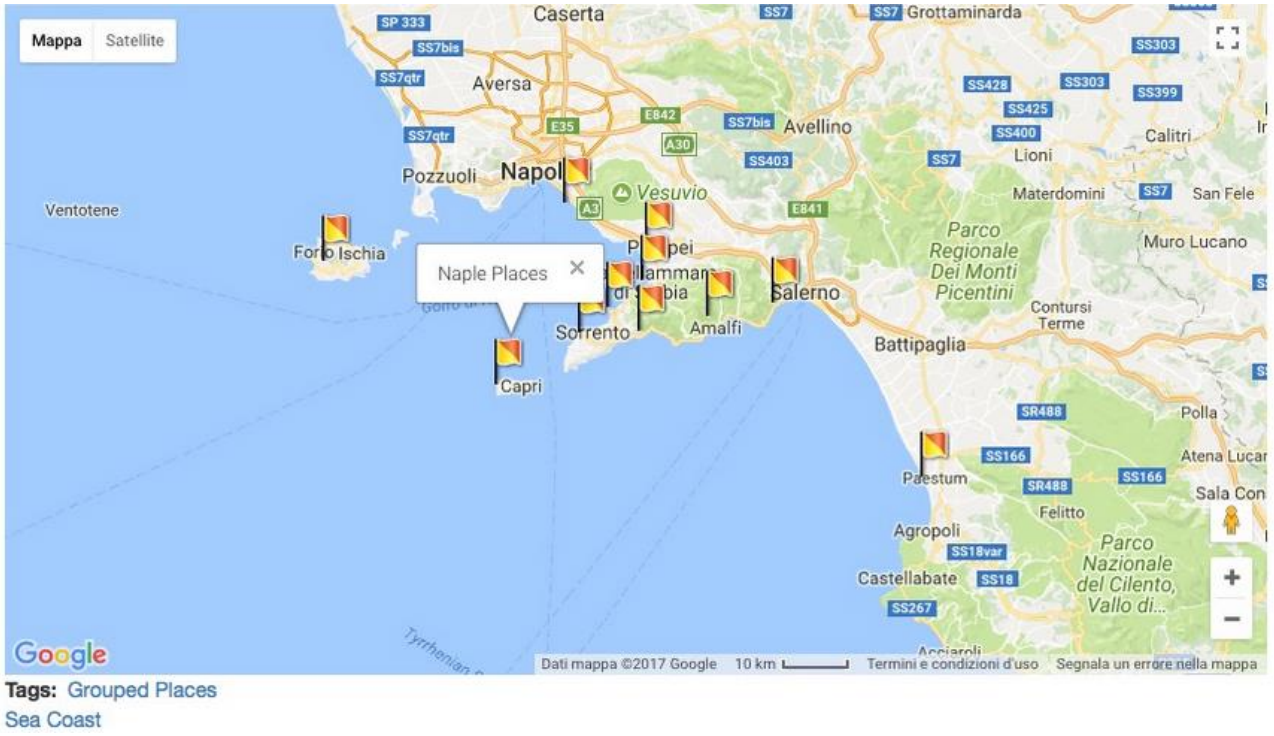


Figure 32: Scenario Transferability Component: Geofield Map example

7 Scenario Analysis, Decision Support and Report Generation (T4.5)

The project board of T4.5 is available at <https://github.com/orgs/clarity-h2020/projects/9>.

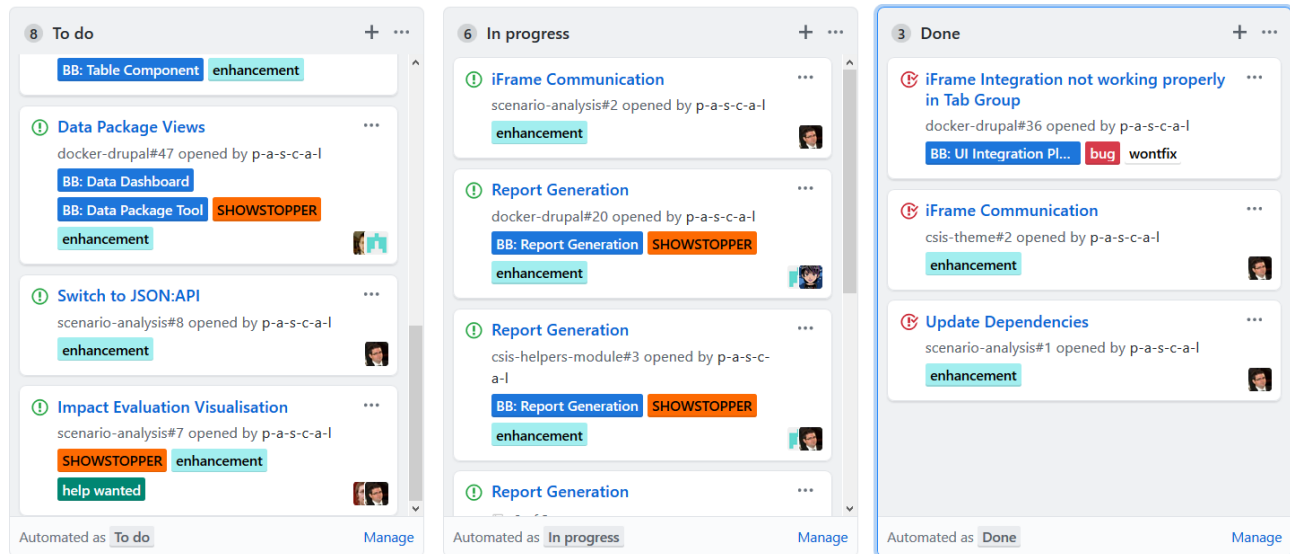


Figure: T4.5 project board (excerpt)

7.1 Multi Criteria Decision Analysis Tool

“The Multi Criteria Decision Analysis Tool supports the analysis and comparison of (adaptation) scenarios regarding performance indicators that can be defined by the end user and thus leverages what-if decision support to investigate the effects of adaptation measures and risk reduction options in the specific project context, and allows the comparison of alternative strategies. Thereby the tool provides multi-criteria ranking functions to compare and rank different scenarios and corresponding adaptation plans according to different criteria and their relative weight and level of importance.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Multi-Criteria-Decision-Analysis-Tool>

Continuously updated (technology support) plan:

<https://github.com/clarity-h2020/scenario-analysis/issues?q=%20>

Source code and configuration files repository:

<https://github.com/clarity-h2020/scenario-analysis>

Deployed instance:

<https://csis.myclimateservice.eu/apps/scenario-analysis/app/index.html>

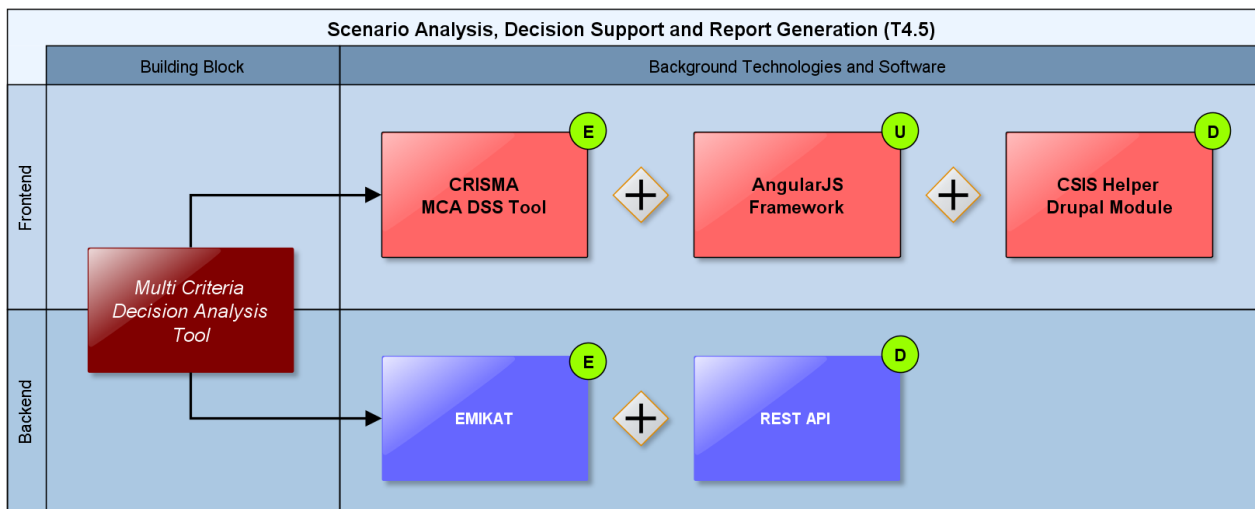


Figure 33: Updated Multi Criteria Decision Analysis Tool technology support

As initially foreseen in the technology support plan, this building block is being developed on basis of the CRISMA Multi Criteria Decision Analysis (MCDA) Tools described in Annex 1 of this document. To satisfy the functional requirements of this building block, some extension of the original of the original software have to be made (Figure 33). This concerns mainly the support for report generation (see chapter 7.2) and the communication with the back-end services.

Although the CRISMA tools have been implemented on basis of the AngularJS 1.0 JavaScript framework, which has been superseded by Angular 5.0 in 2017, the existing tools could successfully be integrated into CSIS Drupal platform (Figure 34). For supporting the seamless communication between Drupal’s JavaScript API and the MCDA tools, which are embedded as iFrame, WP4 developed a dedicated Drupal module. This module offers common helper functions and is used by other building blocks that are realised by independent RIAs and integrated into the Drupal 8 platform following the CLARITY’s user interface integration concept. Regarding the backend of the tools, technology support plan has been updated (Figure 33) to reflect the fact that Risk & Impact is calculated by AIT EMIKAT as described in chapter 4.1. A dedicated REST API for aggregation impact data is being developed.

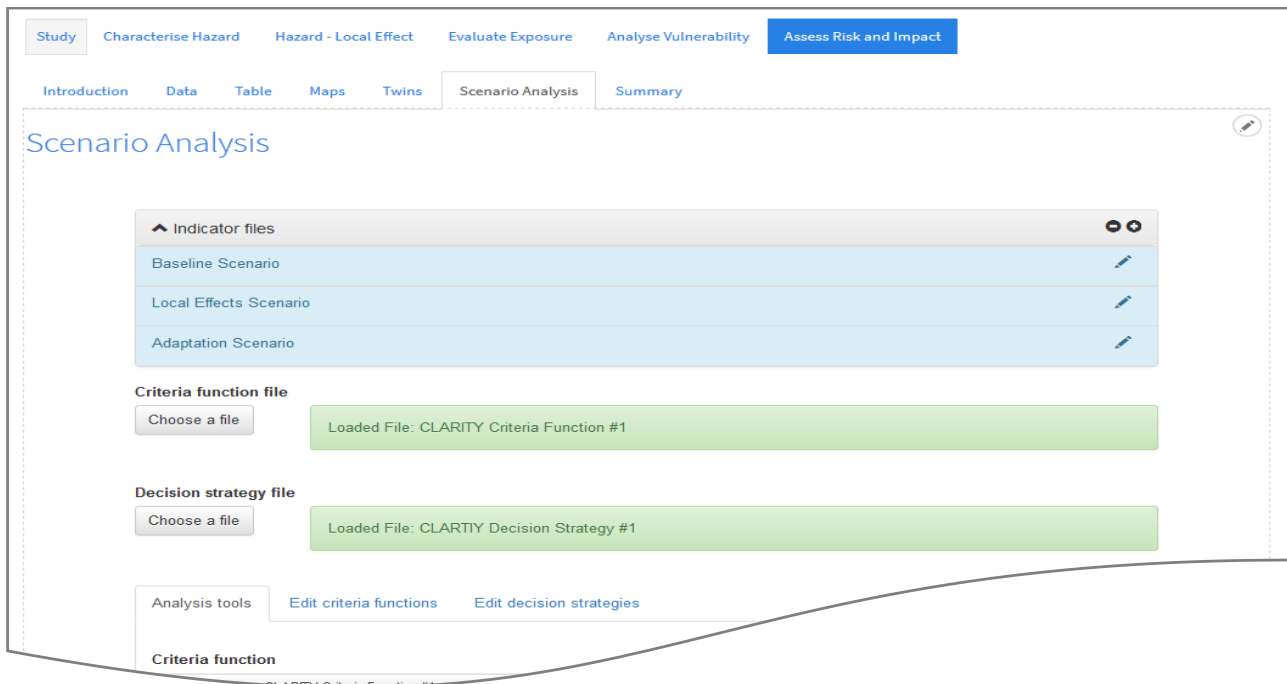


Figure 34: Multi Criteria Decision Analysis Tool (excerpt)

The open source code of this building block is maintained in a dedicated GitHub repository that was forked from the original CRISMA repository. Since it is directly integrated into the CSIS Drupal Platform, it is deployed together with the CSIS Docker container in CLARITY's cloud infrastructure.

7.2 Report Generation

“The result of a climate adaptation study is a report that could be (semi-)automatically generated. Report Generation should enable the user to easily access and download draft and final reports packages at the end of the project assessment process. Such report packages should include automatically generated documentation (with embedded supporting tables, graphs and maps of the area under study).” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Report-Generation>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+label%3A%22BB%3A+Report+Generation%22+sort%3Acreated-asc&type=issues>

Drupal configuration repository (private):

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

CSIS Helpers Drupal module repository:

<https://github.com/clarity-h2020/csis-helpers-module>

Deployed instance:

<https://csis.myclimateservice.eu/>

Unless proposed in the technology support report, there is currently no need to develop separate user interface component for report generation. The functional requirements of the building blocks can be satisfied by Drupal 8 platform and the html2canvas and pdf.js JavaScript Libraries (Figure 35). Support for report generation will therefore be implemented in the CSIS Helper Drupal module and thus will be available to any other component that intends to add graphs, maps, tables etc. to a PDF Report. Report Generation functionality will allow to take screenshots of embedded RIAs like the Map Component (chapter 6.1) or the MCDA Tools (chapter 7.1) and to save them to the respective study entity, which represents the user’s scenario. With help of the pdf.js library, the study entity can then be exported and visualised as PDF document.

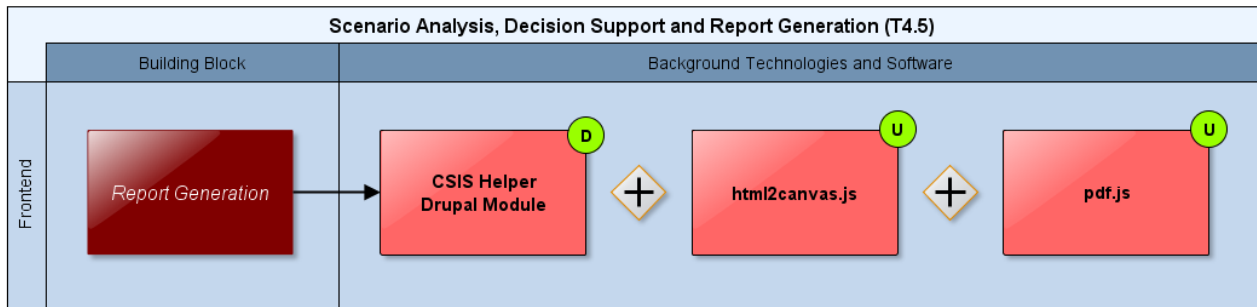


Figure 35: Updated Report Generation technology support

Apart from the CSIS Helper Drupal module there is not dedicated source code repository needed for this building block. As the module is added to the CSIS Drupal Platform, it is deployed together with the CSIS Docker container in CLARITY’s cloud infrastructure.

8 Integration and Infrastructure (T4.1 and T1.4)

The project board of T4.5 is available at <https://github.com/orgs/clarity-h2020/projects/3>.

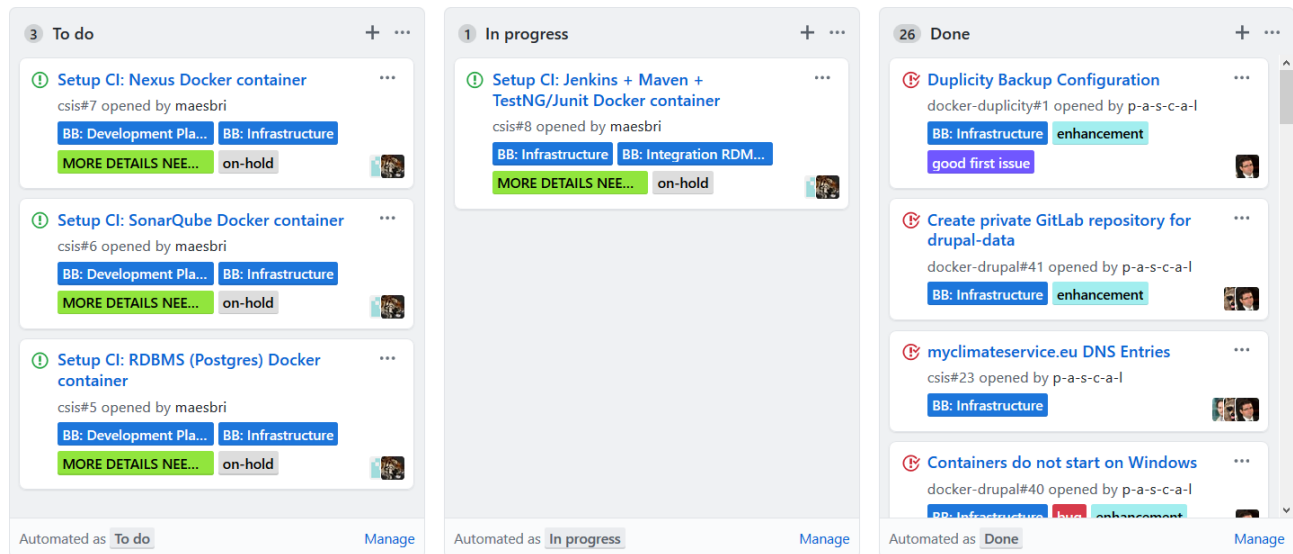


Figure 36: T1.4 project board (excerpt)

8.1 Container Engine and Cloud Infrastructure

“The CSIS is envisioned to be composed of a set of (micro) services that can independently be deployed as isolated containers either on a self-hosted physical server that provides its own container engine or in a virtualized environment offered by a cloud-hosting provider. For this purpose, a Container Engine and Cloud Infrastructure building block must be part of the CSIS Architecture.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Container-Engine-and-Cloud-Infrastructure>

Continuously updated (technology support) plan:

<https://github.com/search?q=desc&q=org%3Aclarity-h2020+label%3A%22BB%3A+Infrastructure%22+sort%3Acreated-asc&s=created&type=issues>

Docker configuration files repositories:

<https://github.com/clarity-h2020/docker-pgadmin>

<https://github.com/clarity-h2020/docker-duplicity>

<https://github.com/clarity-h2020/docker-compose-letsencrypt-nginx-proxy-companion>

<https://github.com/clarity-h2020/docker-hoster>

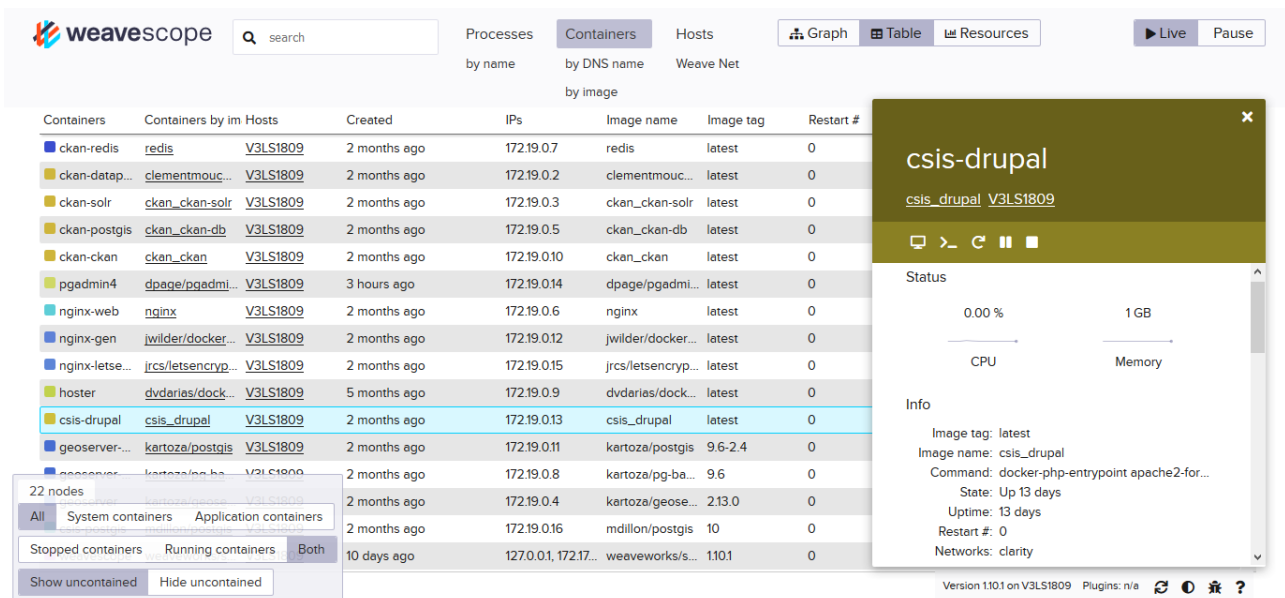
Deployed instances:

Dedicated Server V3LS1809

In accordance to the initial technology support plan, CLARITY partners AIT provided a dedicated development and deployment server (64 GB RAM, 8 cores, 8 TB hard drive) where the Docker container engine (Figure 37) is made available for deploying the CSIS components’ containers. Since such Docker containers can be installed on premise as well as in cloud-based environments, this option can be used unless project development requires a more powerful infrastructure.

The Docker containerisation technology is able to encapsulate lightweight runtime environments and provides good portability, performance, ease of replication, environment isolation, high availability and scalability for micro services. Additionally, so called compose files offer a lightweight automation technology to coordinate, monitor and manage the different containerised micro services.

For all micro-service based CSIS components, source code repositories for hosting the respective Docker compose files and auxiliary configuration files have been created at GitHub. This includes also several infrastructure (micro) services like proxies, load balancers, name services, etc.



Containers	Containers by im	Hosts	Created	IPs	Image name	Image tag	Restart #
ckan-redis	redis	V3LS1809	2 months ago	172.19.0.7	redis	latest	0
ckan-datap...	clementmouc...	V3LS1809	2 months ago	172.19.0.2	clementmouc...	latest	0
ckan-solr	ckan_ckan-solr	V3LS1809	2 months ago	172.19.0.3	ckan_ckan-solr	latest	0
ckan-postgis	ckan_ckan-db	V3LS1809	2 months ago	172.19.0.5	ckan_ckan-db	latest	0
ckan-ckan	ckan_ckan	V3LS1809	2 months ago	172.19.0.10	ckan_ckan	latest	0
pgadmin4	dpape/pgadmi...	V3LS1809	3 hours ago	172.19.0.14	dpape/pgadmi...	latest	0
nginx-web	nginx	V3LS1809	2 months ago	172.19.0.6	nginx	latest	0
nginx-gen	jwilder/docker...	V3LS1809	2 months ago	172.19.0.12	jwilder/docker...	latest	0
nginx-letse...	jrcs/letsencryp...	V3LS1809	2 months ago	172.19.0.15	jrcs/letsencryp...	latest	0
hoster	dvda/ries/dock...	V3LS1809	5 months ago	172.19.0.9	dvda/ries/dock...	latest	0
csis-drupal	csis_drupal	V3LS1809	2 months ago	172.19.0.13	csis_drupal	latest	0
geoserver...	kartoza/postgis	V3LS1809	2 months ago	172.19.0.11	kartoza/postgis	9.6-2.4	0
geoserver...	kartoza/pg-ba...	V3LS1809	2 months ago	172.19.0.8	kartoza/pg-ba...	9.6	0
geoserver...	kartoza/geose...	V3LS1809	2 months ago	172.19.0.4	kartoza/geose...	2.13.0	0
geoserver...	mdillon/postgis	V3LS1809	2 months ago	172.19.0.16	mdillon/postgis	10	0
geoserver...	weaveworks/s...	V3LS1809	10 days ago	127.0.0.1, 172.17...	weaveworks/s...	1.10.1	0

Figure 37: Container Engine and Cloud Infrastructure

8.2 UI Integration Platform

“CLARITY's common User Interface Integration Platform is the unified entry point to the CLARITY ecosystem. It integrates the different frontends (user interfaces) of CLARITY building blocks and ICT climate services, respectively, with the CLARITY Marketplace and the CSIS.” [2] It is therefore the central component for integrating all building blocks of the CSIS Architecture.

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/UI-Integration-Platform>

Continuously updated (technology support) plan:

<https://github.com/clarity-h2020/docker-drupal/issues?q=%20>

Source code and configuration files repository:

<https://gitlab.atosresearch.eu/ari/clarity-csis-drupal>

<https://github.com/clarity-h2020/docker-drupal/>

Deployed instance:

<https://csis.myclimateservice.eu/>

The initial technology support report proposed three different approaches for realising the CLARITY UI Integration Platform: A portal-based approach, a CMS-based approach and a JavaScript Framework-based approach. With the selection of the Drupal 8 CMS and the possibility to integrate arbitrary RIAs directly or via iFrame (Figure 38), WP4 technology support team implemented the CMS-based approach. Additional user-interface centred building blocks like the Map Component (see chapter 6.1) and the MCDA Tools (see chapter 7.1) are then independently developed following the JavaScript Framework-based approach.

The Docker-based container configuration (e.g. compose file) as well as the Drupal-specific configurations are stored in dedicated git repositories. The respective Drupal Docker container is deployed in CLARITY's container engine (see chapter 8.1). As backed storage of the Drupal platform, a dedicated PostgreSQL with PostGIS extension container is used (see chapter 8.4). External applications and building blocks like for example AIT EMIKAT (see chapter 4.1) can communicate with the Drupal Platform via a built-in REST API.

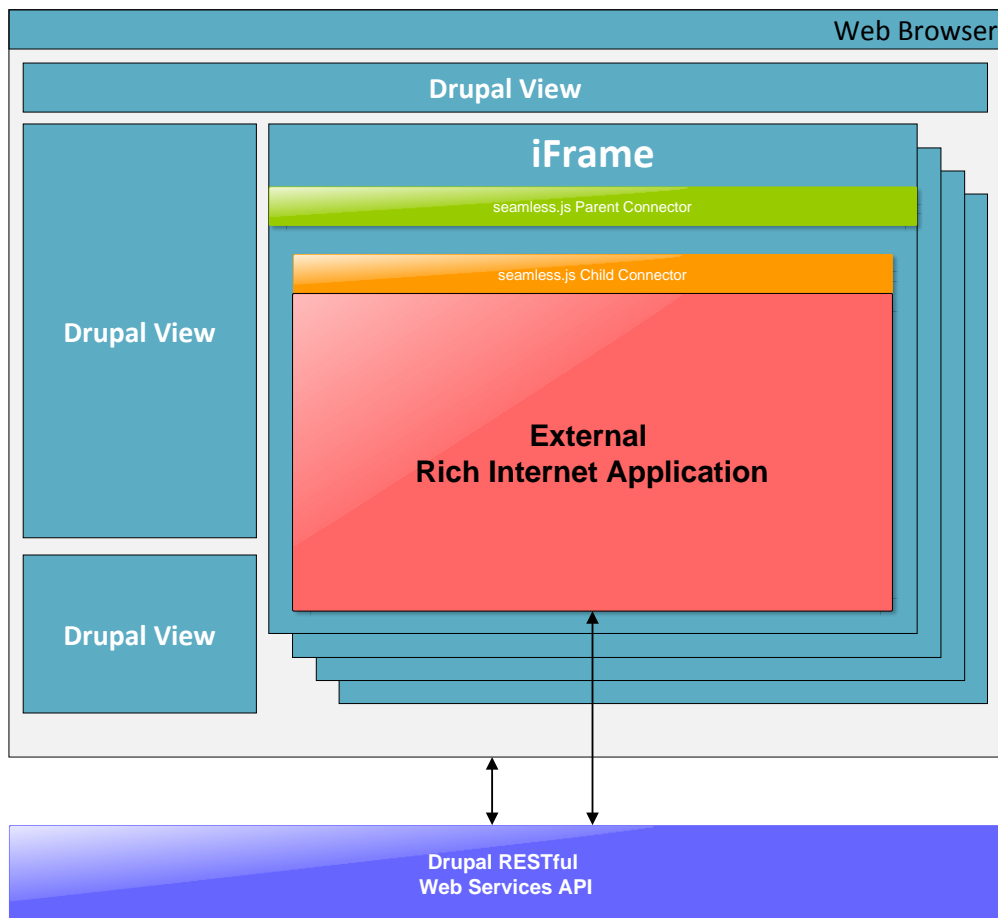


Figure 38: UI Integration Platform iFrame integration

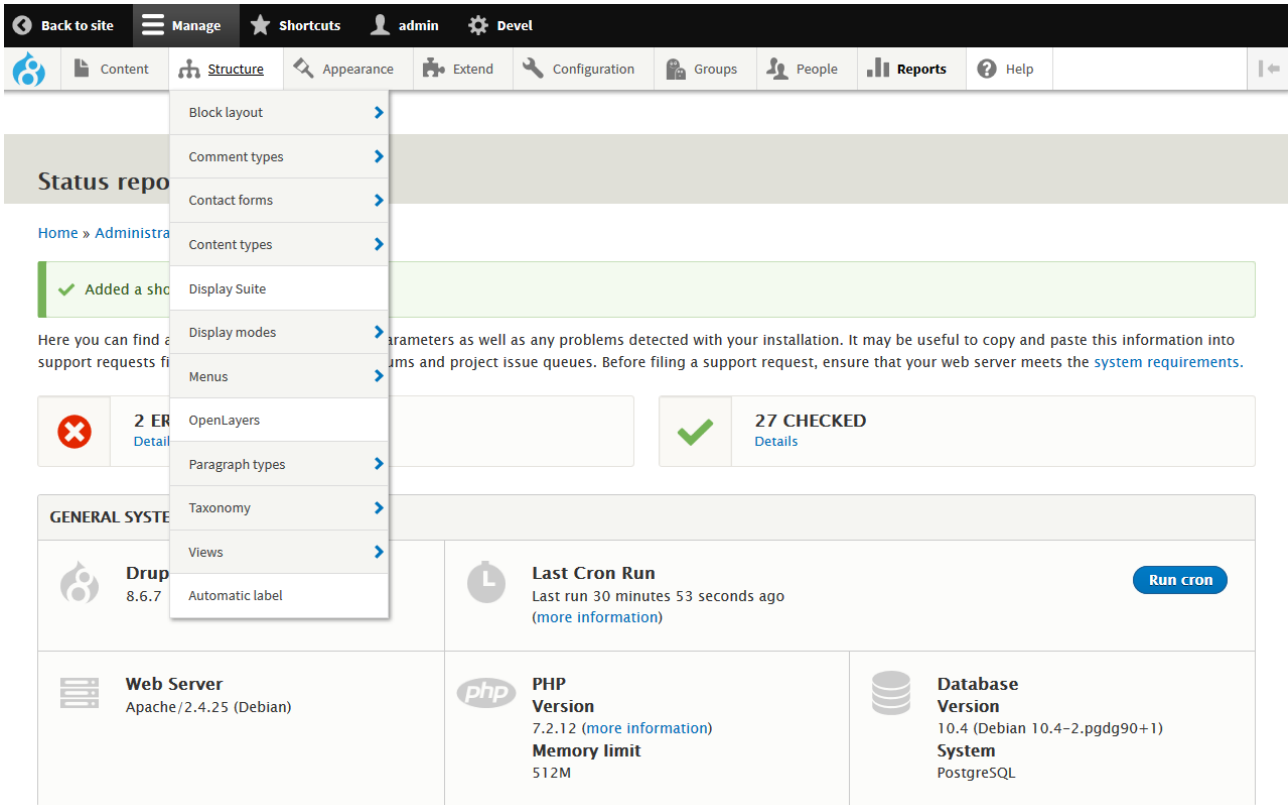


Figure 39: Drupal UI Integration Platform

8.3 Integration and Development Platform

“The purpose of this building block is to provide a continuous integration platform allowing every consortium partner to be equipped with the tools and measures for best practices in software engineering. One of the most important factors on a successful IT project is to develop high quality software. Thereby, an appropriate development infrastructure and best practices are crucial in development in a distributed environment.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Integration-and-Development-Platform>

Continuously updated (technology support) plan:

<https://github.com/search?o=desc&q=org%3Aclarity-h2020+label%3A%22bb%3A+infrastructure%22+sort%3Acreated-asc&s=&type=issues>

Source code and configuration files repository:

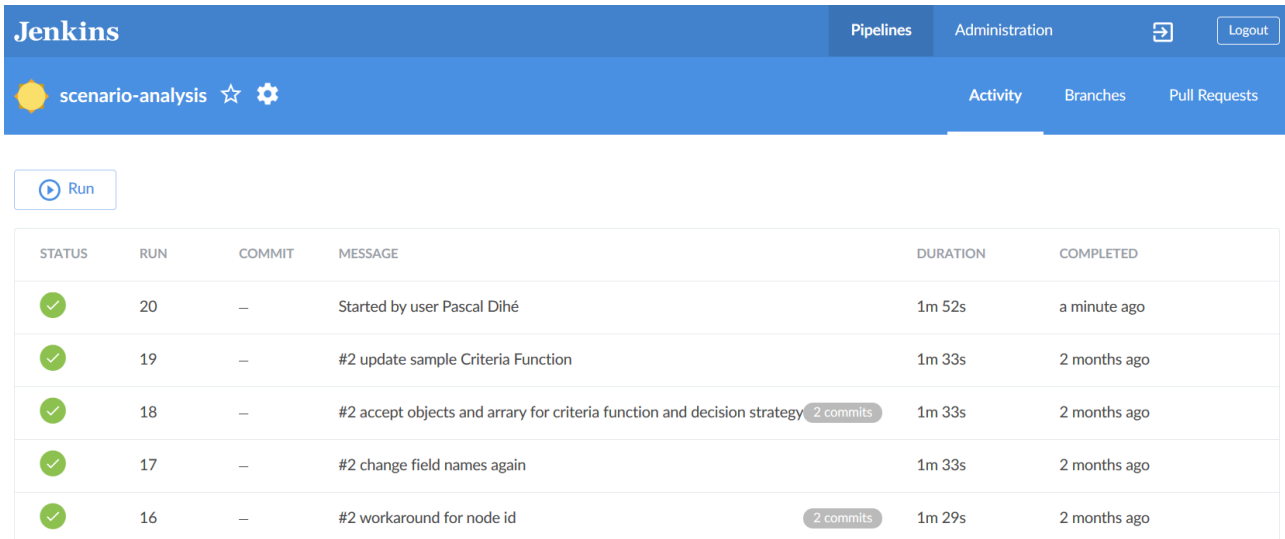
n/a

Deployed instance (protected):

<https://ci.cismet.de>

According to the initial technology support report and the specifications in D1.1 “Initial workshops and the CLARITY development environment” [8], the technology stack of the Integration and Development Platform is composed of GitHub, Jenkins, Sonatype Nexus, SonarQube, Selenium-Grid, Gulp and Apache Maven.

While no changes to the initial continuous integration tools selection has been made, most tools haven't been deployed yet since there was no need to. Moreover, since no developments in the Java Programming Language are planned (see also chapter 2 in Annex 1), the Java-related tools (Gulp and Apache Maven) are not needed at all. A private Jenkins CI installation hosted at cismet servers is currently used for their developments (Figure 40). This instance will eventually be moved to the Container Engine and Cloud Infrastructure (chapter 8.1), as soon as demand from other partners arises.



STATUS	RUN	COMMIT	MESSAGE	DURATION	COMPLETED
✓	20	–	Started by user Pascal Dihé	1m 52s	a minute ago
✓	19	–	#2 update sample Criteria Function	1m 33s	2 months ago
✓	18	–	#2 accept objects and array for criteria function and decision strategy 2 commits	1m 33s	2 months ago
✓	17	–	#2 change field names again	1m 33s	2 months ago
✓	16	–	#2 workaround for node id 2 commits	1m 29s	2 months ago

Figure 40: Integration and Development Platform

8.4 Integration RDMBS

“The Integration RDMBS is the central relational database management system for management and integration of common and shared information stored as relations (in tabular form). It stores, among others, the individual infrastructure project configurations and the associated assessment and adaptation planning information created by end users. Thereby, it is important to highlight, that the actual datasets generated during the EU-GL/CLARITY adaptation planning process (hazards maps, model outputs, etc.) are not stored in this Integration RDMBS but in general in a separate Data repository” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Integration-RDMBS>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+label%3A%22BB%3A+Integration+RDMBS%22&type=Issues>

Source code and configuration files repository:

<https://github.com/clarity-h2020/docker-pgadmin>

<https://github.com/clarity-h2020/docker-duplicity>

<https://github.com/clarity-h2020/docker-drupal/https://github.com/clarity-h2020/map-component>

Deployed instance:

<https://ckan.myclimateservice.eu/>

As defined in the initial technology support plan, the integration RDBMS is realised as PostgreSQL + PostGIS database container that is used by all Drupal 8 based components as backend storage. It is deployed together with the overall Drupal Instance in CLARITY Container Infrastructure (Figure 41).

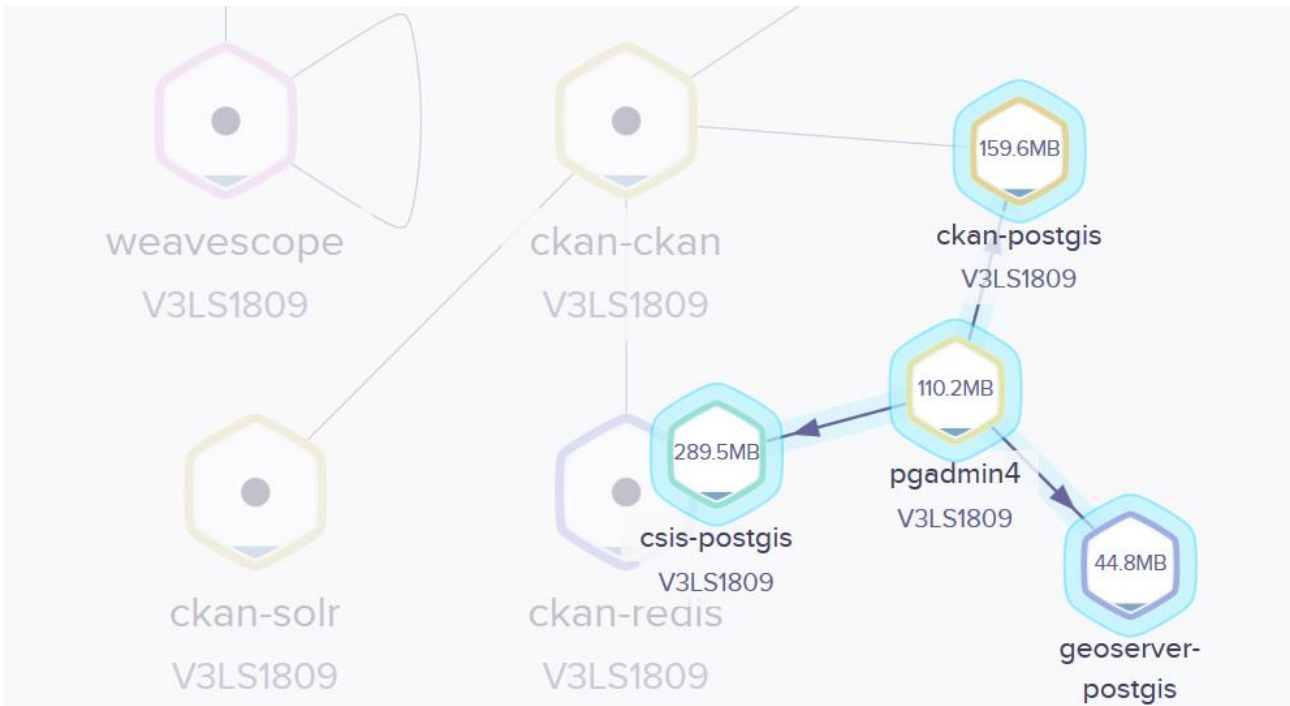


Figure 41: PostgreSQL Docker container in Container Infrastructure

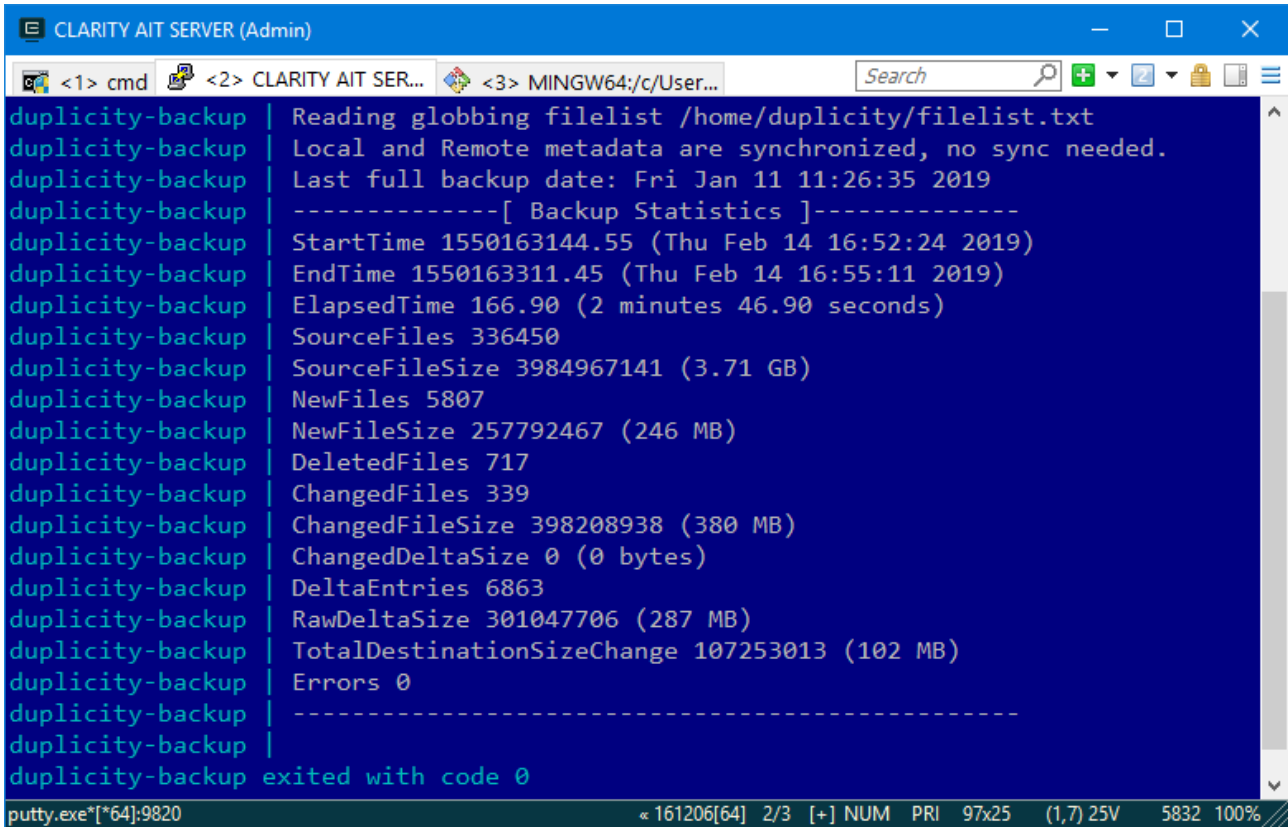
The deployment configuration (Docker Compose) is therefore shared with the UI Integration Platform (chapter 8.2). Thus, actually no development is required for this building block and the activities are limited to installation, configuration and maintenance (e.g. backups).

The screenshot shows the pgAdmin 4 interface with the 'Statistics' tab selected for the 'drupal' database. The left sidebar shows a tree view of servers and databases. The main panel displays a table of statistics.

Statistics	Value
Backends	1
Xact committed	5792174
Xact rolled back	1277
Blocks read	316030
Blocks hit	222193928
Tuples returned	355220402
Tuples fetched	110510170
Tuples inserted	762685
Tuples updated	109912
Tuples deleted	626863
Last statistics reset	2019-01-11 12:12:51.592456+00
Tablespace conflicts	0
Lock conflicts	0
Snapshot conflicts	0
Bufferpin conflicts	0
Deadlock conflicts	0
Temporary files	0

Figure 42: Integration RDMBS

Management tasks are carried out via the PG-Admin Web Interface (Figure 42). Incremental database backups with help of the duplicity tool (Figure 43) in addition to complete server backups performed with “Veeam Backup & Replication”. Both management tools are deployed as separate containers and their runtime configuration is maintained in dedicated repositories.



```

CLARITY AIT SERVER (Admin)
<1> cmd <2> CLARITY AIT SER... <3> MINGW64;/c/User... Search
duplicity-backup | Reading globbing filelist /home/duplicity/filelist.txt
duplicity-backup | Local and Remote metadata are synchronized, no sync needed.
duplicity-backup | Last full backup date: Fri Jan 11 11:26:35 2019
duplicity-backup | -----[ Backup Statistics ]-----
duplicity-backup | StartTime 1550163144.55 (Thu Feb 14 16:52:24 2019)
duplicity-backup | EndTime 1550163311.45 (Thu Feb 14 16:55:11 2019)
duplicity-backup | ElapsedTime 166.90 (2 minutes 46.90 seconds)
duplicity-backup | SourceFiles 336450
duplicity-backup | SourceFileSize 3984967141 (3.71 GB)
duplicity-backup | NewFiles 5807
duplicity-backup | NewFileSize 257792467 (246 MB)
duplicity-backup | DeletedFiles 717
duplicity-backup | ChangedFiles 339
duplicity-backup | ChangedFileSize 398208938 (380 MB)
duplicity-backup | ChangedDeltaSize 0 (0 bytes)
duplicity-backup | DeltaEntries 6863
duplicity-backup | RawDeltaSize 301047706 (287 MB)
duplicity-backup | TotalDestinationSizeChange 107253013 (102 MB)
duplicity-backup | Errors 0
duplicity-backup | -----
duplicity-backup | exited with code 0
putty.exe[*64]:9820

```

Figure 43: RDBMS Backup

8.5 Data Repository

“This Building Block represents a set of generic data repositories that can be used to store, manage, and retrieve different types of (file-based) vector and raster datasets. Among others, this building block is used to facilitate the sharing of datasets between users and providers of Climate Services.” [2]

Up-to-date building block description and requirements:

<https://clarity-h2020.github.io/csis-architecture/docs/building-blocks/Integration-RDMBS>

Continuously updated (technology support) plan:

<https://github.com/search?q=org%3Aclarity-h2020+label%3A%22BB%3A+Data+Repository%22&type=Issues>

Source code and configuration files repository:

<https://github.com/clarity-h2020/docker-geoserver>

<https://github.com/clarity-h2020/docker-geonode>

<https://github.com/clarity-h2020/thredds-docker>

<https://github.com/clarity-h2020/thredds-docker>

Deployed instances

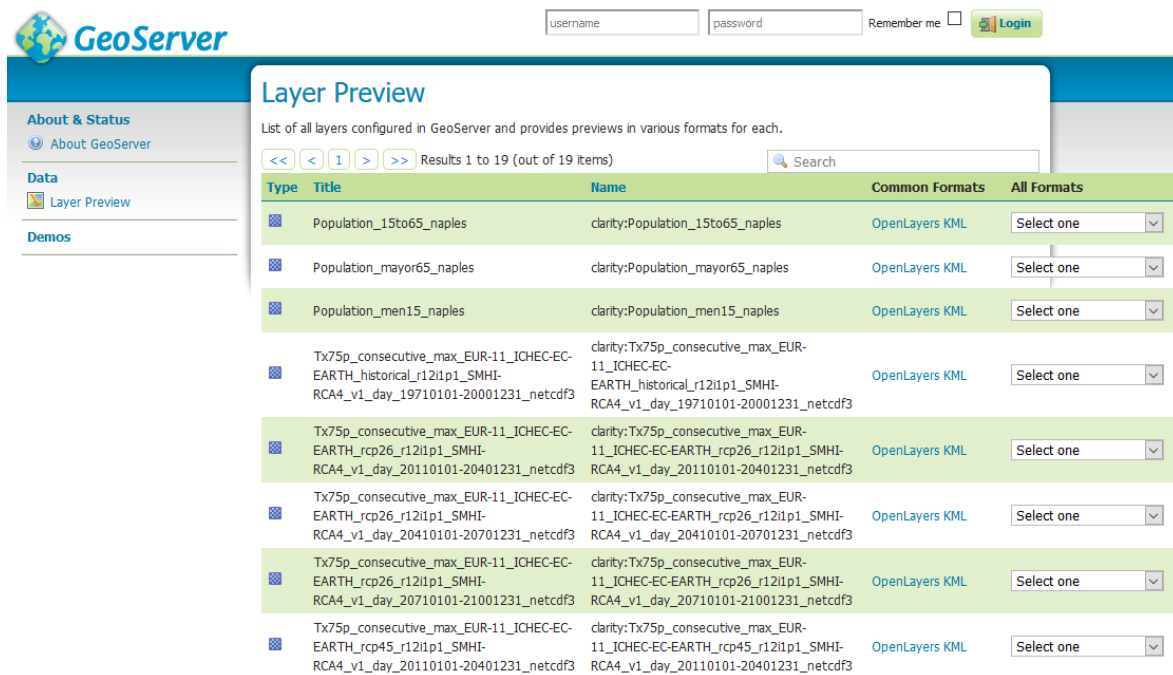
:

<https://geoserver.myclimateservice.eu/geoserver/web/>

<https://service.emikat.at/geoserver/web/>

<https://clarity.meteogrid.com/geoserver/web/>
<https://service.clarity-h2020.eu:8080/geoserver/web>

The initial technology support plan mentioned different Data Repositories, e.g. ERDDAP and THREDDs for gridded data in NetCDF format. During assessment installation and test of these repositories, it turned out, that the best choice for managing geospatial directly used in CSIS is GeoServer (Figure 44). This is also owed to the fact, that thanks to the Data Package Specification, the number of different data formats needed / used throughout the CSIS could be reduced. GeoServer's Web Coverage Service for gridded data is mainly used by EMIKAT (chapter 4.1) and the Table Component (chapter 4.3) while the Web Map Service is mainly used by the Map Component (chapter 6.1) and the Scenario Transferability Component (chapter 6.2).



The screenshot shows the GeoServer web interface. At the top, there is a login section with fields for 'username' and 'password', a 'Remember me' checkbox, and a 'Login' button. Below this is the 'Layer Preview' section, which lists all layers configured in GeoServer. The list includes columns for 'Type', 'Title', 'Name', 'Common Formats', and 'All Formats'. The layers listed are:

Type	Title	Name	Common Formats	All Formats
	Population_15to65_naples	clarity:Population_15to65_naples	OpenLayers KML	Select one
	Population_mayor65_naples	clarity:Population_mayor65_naples	OpenLayers KML	Select one
	Population_men15_naples	clarity:Population_men15_naples	OpenLayers KML	Select one
	Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_historical_r12i1p1_SMHI-RCA4_v1_day_19710101-20001231_netcdf3	clarity:Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_historical_r12i1p1_SMHI-RCA4_v1_day_19710101-20001231_netcdf3	OpenLayers KML	Select one
	Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20110101-20401231_netcdf3	clarity:Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20110101-20401231_netcdf3	OpenLayers KML	Select one
	Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20410101-20701231_netcdf3	clarity:Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20410101-20701231_netcdf3	OpenLayers KML	Select one
	Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20710101-21001231_netcdf3	clarity:Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp26_r12i1p1_SMHI-RCA4_v1_day_20710101-21001231_netcdf3	OpenLayers KML	Select one
	Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp45_r12i1p1_SMHI-RCA4_v1_day_20110101-20401231_netcdf3	clarity:Tx75p_consecutive_max_EUR-11_ICHEC-EC-EARTH_rcp45_r12i1p1_SMHI-RCA4_v1_day_20110101-20401231_netcdf3	OpenLayers KML	Select one

Figure 44: GeoServer Data Repository

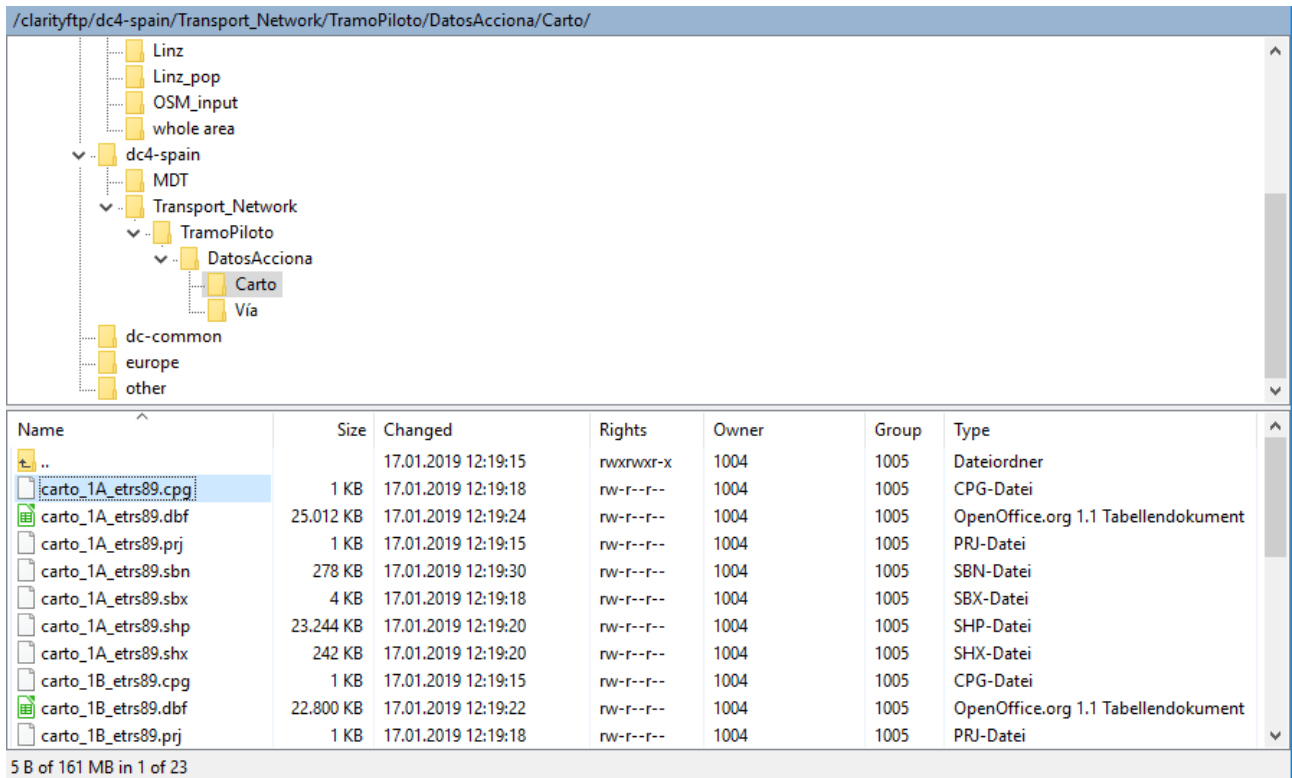
At the moment, four different GeoServer instances are deployed at AIT (for EMIKAT results), ATOS (Local Effects and Exposure Layers) and METEOGRID (for Hazard and Hazard with Local Effects layers) premises plus a common instance deployed in the CLARITY Container Infrastructure. Once data production and collection is finished, the separately managed datasets will be consolidated in the common CSIS instance.

For any *temporary* or “raw” data collected and produced in CLARITY, including for example data that serves as input for local models or intermediate output data in NetCDF Format, CLARITY provides a secure FTP Server (Figure 45). The CLARITY Data Management Plan (D7.9) reports also on such data that is not directly used within in the CSIS but may be needed for an intermediate process step and that is not meant for publication. Thus, some of the datasets used and produced in CLARITY and described in the DMP link to the CLARITY sFTP.

Such data that is temporarily stored on the CLARITY sFTP and that is eligible for publication under an open licence (open data) will, after undergoing a verification and quality control process, be made available on Zenodo research data repository (<https://zenodo.org/communities/clarity/>), the DRMKC Risk Data Hub (<https://drmkc.jrc.ec.europa.eu/risk-data-hub>) and the Climate Change Centre Austria's (CCCA) Data Server (<https://data.ccca.ac.at/organization/zamg>).

The actual live version of the CLARITY DMP is realised with help of the Catalogue of Data Sources (chapter 4.2) and available at <https://ckan.myclimateservice.eu/>. The sFTP service itself is hosted at ATOS premises.

/clarityftp/dc4-spain/Transport_Network/TramoPiloto/DatosAcciona/Carto/



Name	Size	Changed	Rights	Owner	Group	Type
..		17.01.2019 12:19:15	rw-rw-r-x	1004	1005	Dateiordner
carto_1A_etr89.cpg	1 KB	17.01.2019 12:19:18	rw-r--r--	1004	1005	CPG-Datei
carto_1A_etr89.dbf	25.012 KB	17.01.2019 12:19:24	rw-r--r--	1004	1005	OpenOffice.org 1.1 Tabellendokument
carto_1A_etr89.prj	1 KB	17.01.2019 12:19:15	rw-r--r--	1004	1005	PRJ-Datei
carto_1A_etr89.sbn	278 KB	17.01.2019 12:19:30	rw-r--r--	1004	1005	SBN-Datei
carto_1A_etr89.sbx	4 KB	17.01.2019 12:19:18	rw-r--r--	1004	1005	SBX-Datei
carto_1A_etr89.shp	23.244 KB	17.01.2019 12:19:20	rw-r--r--	1004	1005	SHP-Datei
carto_1A_etr89.shx	242 KB	17.01.2019 12:19:20	rw-r--r--	1004	1005	SHX-Datei
carto_1B_etr89.cpg	1 KB	17.01.2019 12:19:15	rw-r--r--	1004	1005	CPG-Datei
carto_1B_etr89.dbf	22.800 KB	17.01.2019 12:19:22	rw-r--r--	1004	1005	OpenOffice.org 1.1 Tabellendokument
carto_1B_etr89.prj	1 KB	17.01.2019 12:19:18	rw-r--r--	1004	1005	PRJ-Datei

5 B of 161 MB in 1 of 23

Figure 45: sFTP Data Repository

9 Conclusion

One of the main reasons for the wide acceptance of agile software methods is that they demonstrably lead to higher productivity and product quality than classical development models. Product quality is not measured solely by purely technical criteria, but rather by the objective benefit, the product offers the customer. The close cooperation between development teams and real users contributes significantly to the success of the agile process model. Direct, continuous user feedback, short development cycles and the focus on the essential aspects in each cycle mean that changes during development are not perceived as a disruptive factor but as an opportunity for product improvement.

CLARITY applies these methods not only to the implementation of isolated building blocks offered by WP4 “Technology Support” but to the whole CSIS and climate services co-creation process. Thereby, it makes intensive use of the GitHub Platform that offers a set of project management tools used by agile teams to discuss, coordinate and steer the entire product development life cycle. Accordingly, the overall CLARITY co-creation Team on GitHub consists of all stakeholders of the CSIS including software developers, scientist, data providers/purveyors and end users. CLARITY has successfully implemented a transparent and multi-disciplinary co-creation process for tailored and integrated climate services.

This document summarises the emergent architecture introduced in D4.2 “CLARITY CSIS Architecture” [2] and describes on how building blocks of the CSIS are implemented on basis of software components and key technologies. The actual nature of this deliverable is

- software (source code and/or binaries) and their technical documentation made available in online repositories at GitHub for specific building blocks or supplemental and supporting tools and services;
- (micro)-services and user interfaces deployed in CLARITY’s containerised service infrastructure;
- mock-ups which represent a visual preview of the envisaged products and services for end users and at the same time serve as "user interface blueprint" for developers;
- the online CSIS product backlog, that represents all identified highly-level tasks (“issues”) that need to be accomplished in order to advance the development of the CSIS;
- a breakdown of the backlog issues into detailed subtasks (“implementable issues” ready to be picked up by the development team) at the level of building blocks; and
- project boards, that are used by task leaders to monitor, prioritise and coordinate the development of building blocks covered by their task.

The next deliverable of WP4 (D4.4 “Technology Support Report v2”) will accompany the final release of the WP4 software and summarise the work performed in WP4 since release of this deliverable.

References

- [1] Directorate-General Climate Action, “Non-paper Guidelines for Project Managers: Making vulnerable investments climate resilient,” European Commission, 16 April 2013. [Online]. Available: <http://climate-adapt.eea.europa.eu/metadata/guidances/non-paper-guidelines-for-project-managers-making-vulnerable-investments-climate-resilient/guidelines-for-project-managers.pdf>. [Accessed 6 November 2017].
- [2] P. Dihé, “D4.2 CLARITY CSIS Architecture,” Deliverable D4.2 of the European Project H2020-730355 Integrated Climate Adaptation Service Tools for Improving Resilience Measure Efficiency (CLARITY), 2018. [Online].
- [3] P. Dihé, “D4.1 Technology Support Plan,” Deliverable D4.1 of the European Project H2020-730355 Integrated Climate Adaptation Service Tools for Improving Resilience Measure Efficiency (CLARITY), 2018. [Online].
- [4] G. Booch, “Handbook of Software Architecture,” 2017. [Online]. Available: <https://handbookofsoftwarearchitecture.com/?p=63>. [Accessed 19 April 2018].
- [5] M. Zuvela-Aloise, A. Kainz, C. Hahn, M. Leone, G. Zuccharo, D. Del Cogliano, M. Iorio and S. Schlobinski, “D3.1 Science Support Plan and Concept,” Deliverable D3.1 of the European Project H2020-730355 Integrated Climate Adaptation Service Tools for Improving Resilience Measure Efficiency (CLARITY), 2018. [Online].
- [6] P. Abrahamsson, M. Ali Babar and P. Kruchten, “Agility and Architecture: Can They Coexist?,” *IEEE SOFTWARE*, March/April 2010.
- [7] R. Duro and D. Havlik, “D1.1 Initial workshops and the CLARITY development environment,” Deliverable D1.1 of the European Project H2020-730355 Integrated Climate Adaptation Service Tools for Improving Resilience Measure Efficiency (CLARITY), 5 January 2018. [Online].
- [8] S. Schlobinski, G. Zuccaro, M. Scholl, D. Meiers, R. Denzer¹, S. Guarino, W. Engelbach, K. Taveter and S. Frysinger, “Decision Making and Strategic Planning for Disaster Preparedness with a Multi-Criteria-Analysis Decision Support System,” in *Environmental Software Systems. Infrastructures, Services and Applications*, Melbourne, 2015.
- [9] G. Zuccaro and P. Filomena, “Multi-criteria analysis in vulnerability assessment,” in *Second Egyptian Conference on Earthquake Engineering*, Aswan, Egypt, 1997.
- [10] R. R. Yager, “Quantifier guided aggregation using OWA operators,” *International Journal of Intelligent Systems*, p. 11–49, 1996.
- [11] J. Ungar, J. Peters-Anders and W. Loibl, “Climate Twins - An Attempt to Quantify Climatological Similarities,” *Environmental Software Systems. Frameworks of eEnvironment*, 2011.

Annex 1: Description of background from previous projects

According to the CLARITY workplan, both the technology support plan as well as the technology support report shall “also provide an overview of the tools that are inherited from previous projects and explain the initial plan for using and/or extending these inputs in CLARITY WP4.” This overview was already provided as Annex 1 of D4.1 “Technology Support Plan” [3], therefore this document will just provide an incremental update.

Since the CSIS Architecture follows an agile and emergent approach, technical possibilities are constantly reconsidered as the project matures. Therefore, several open source software solutions have been identified to replace or complement the software components that have originally foreseen by the CLARITY DoA in the different WP4 tasks. The assessment of available background technologies and prototype software from previous projects (e.g. ENVIROFI) revealed furthermore some discrepancies not only between required, expected and actually provided functionality and level of technological readiness (TLR) of those technologies but also regarding practical issues like outdated technology stacks (e.g. Java GUIs, Adobe Flash websites, AngularJS v1) or user interaction concepts oriented towards different target groups (model experts). These issues were already addressed in the technology support report and the CSIS Architecture, respectively.

Accordingly, Annex 1 of the initial technology support plan gave a brief overview of tools and technologies inherited from previous projects and explained the initial plan for either using and/or extending these components for the different WP4 tasks or for replacing them by more suitable open source software. The technology support report reports now on the actual usage of these components, whether and how these initial plans have been carried out and which additional (foreground) implementations, extensions and customisations were developed by the owners and maintainers of the respective software components.

1. AIT EMIKAT (environmental catalogue)

Description of the background technology or software

EMIKAT is a data management and modelling system for both non-spatial and spatially resolved data that has been developed by CLARITY partner AIT. A brief overview is provided in Annex 1 of the technology support plan [3]. EMIKAT is running on AIT-Server, more information can be found on the EMIKAT website at http://www.emikat.at/?page_id=994&lang=en

The system is fully generic and capable of handling many types of data and models with spatial or temporal properties. This was demonstrated also in the CRISMA project by using EMIKAT as a software base for the population exposure model.

Planned and actual usage in CLARITY

In CLARITY, EMIKAT is used for the implementation of the Catalogue of Elements at Risk and Adaptation Options (chapter 4.1) as described in D4.2 and additionally as calculation service for both impact and risk.

The system is capable of handling geo-data (e.g. points, lines, grids, political areas, etc.) which is especially relevant for the handling of the elements at risk (e.g. points for building locations, lines for roads/transport networks, grids for population densities, etc.) within CLARITY. EMIKAT provides some data import routines, e.g. for CSV, Excel, Shape files, etc. In CLARITY, EMIKAT’s REST-API is being expanded upon to fulfil the requirements towards the interface between the front-end and EMIKAT (e.g. exchange of relevant information from a climate adaption study/project, the layers required for the calculation of impact and risk and the returning of the result tables after the impact/risk calculation run).

During the planning phase of CLARITY, it was not entirely clear whether all relevant layers will be stored in one central location. Since the CLARITY CSIS system aims to handle data also from different external data sources and storage locations, a flexible approach can be taken that does not require all e.g. hazard/exposure

layers to be stored in one central place. The data packages enable EMIKAT to only retrieve the relevant layers within the area / bounding box of the project to carry out the necessary impact and risk calculations.

EMIKAT contributes towards the CLARITY CSIS EU-GL steps “Evaluate Exposure”, “Analyse Vulnerability”, “Assess Risk and Impact”, “Identify Adaptation Options” and “Appraise Adaptation Options”.

2. SUDPLAN Scenario Management System

Description of the background technology or software

The Scenario Management System of the European Project FP7-247708 SUDPLAN – “Sustainable Urban Development Planner for Climate Change Adaptation” (<http://www.sudplan.eu/>) is a framework for developing easy-to-use planning, prediction, decision support and training tools, for the use in an urban context, based on a what-if scenario execution environment implemented in the platform-independent Java Programming Language. A brief overview is provided in Annex 1 of the technology support plan [3].

Planned and actual usage in CLARITY

As already observed in the initial technology support plan, the SUDPLAN SMS did not meet requirements of the Scenario Management building block (chapter 5.3) and was based on discontinued technologies (Java Web Start). Following the initial plan, the SUDPLAN SMS is not used in CLARITY and the scenario management functionality of the CSIS is realised based on Drupal 8 as described in chapter 4.3.3 of the D4.1 “Technology Support Plan”.

3. CRISMA Multi Criteria Decision Analysis Tool

Description of the background technology or software

“The CRISMA FP7project (<http://www.crismaproject.eu/>) developed a set of generic Multi Criteria Decision Analysis (MCDA) Tools that can be further exploited by CLARITY. These tools can be categorised into Scenario Analysis and Comparison Tools and Multi Criteria Analysis and Decision Support Tools. While the Scenario Analysis and Comparison Tools allow a comparison of indicators and criteria for different scenarios, the Multi Criteria Analysis and Decision Support Tools allows a ranking of different scenarios with respect to a specific decision strategy.” [3]

Planned and actual usage in CLARITY

The CRISMA MCDA Tools are developed as generic HTML5/AJAX widgets (RIA) on basis of the AngularJS (<https://angularjs.org/>) JavaScript front-end which makes them directly usable as generic ICT climate services.

Tools have been implemented on basis of the AngularJS 1.0 JavaScript framework which has been superseded by Angular 5.0 in 2017 and version 1.0 reached end of life in 2016. AngularJS 1.0 is still actively maintained by the community on GitHub (<https://github.com/angular/angular.js/>). However, no new plugins and libraries are being developed for version 1.0. Since only minor the adaptations and additions must be made to the CRISMA MCDA Tools in CLARITY to satisfy the requirements, it has been decided that the development can continue on basis of the original JavaScript Framework.

4. Climate Twins

Description of the background technology or software

The assumption of Climate Twins is the following: Climate change means that the future climate in one region will become similar to the present/past climate in other regions in the world. Hence, the Climate Twins application prototype has been developed to explore the possibility of raising public awareness about future climate changes through identifying so called “twin regions”, which show today a similar climatological situation as the user’s region of interest will face in the future. Looking at these “Climate Twin regions” shall provide an impression of how the climate might look like in the future, when comparing, e.g., the temperature of a region of interest to a region in the south of Europe today.

Planned and actual usage in CLARITY

As already observed in the initial technology support plan, the Climate Twins prototype’s technology is too old to be re-used in CLARITY (see Underlying Technology section in Annex 1 of D4.1 “Technology Support Plan” [3]), CLARITY will make use of the prototype’s idea only and extend it by new indicators for comparison of elements at risk and project types.