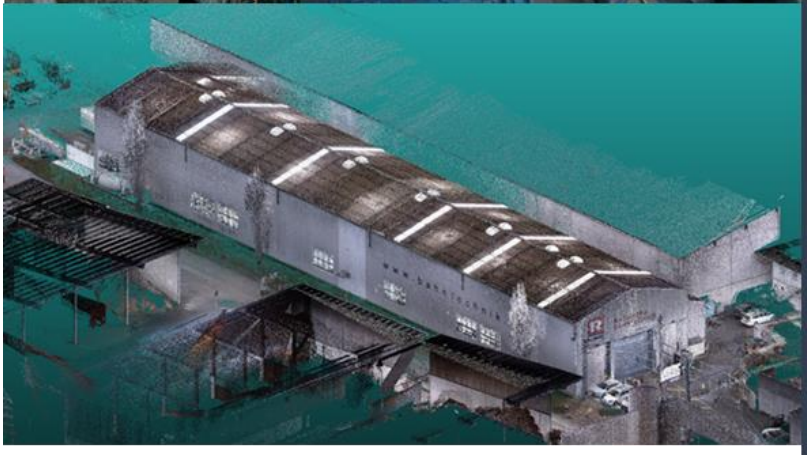


COGITO

CONSTRUCTION PHASE DIGITAL TWIN MODEL

cogito-project.eu

D6.5 – Adaptive Workflow Management and Automation Tool v1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310

D6.5 – Adaptive Workflow Management and Automation Tool v1

Dissemination Level:	Public
Deliverable Type:	Demonstrator
Lead Partner:	NT
Contributing Partners:	UEDIN, BOC-AG, QUE
Due date:	31-05-2022
Actual submission date:	19-08-2022

Authors

Name	Beneficiary	Email
Ján Varga	NT	varga@novitechgroup.sk
Martin Straka	NT	straka@novitechgroup.sk
Bohuš Belej	NT	belej@novitechgroup.sk

Reviewers

Name	Beneficiary	Email
Raúl García-Castro	UPM	rgarcia@fi.upm.es
Agnieszka Mikołajczyk	ASM	a.mikolajczyk@asmresearch.pl
Giorgos Giannakis	Hypertech	g.giannakis@hypertech.gr
Frédéric Bosché	UEDIN	f.bosche@ed.ac.uk

Version History

Version	Editors	Date	Comment
0.1	NT	01.04.2022	ToC and initial version
0.4	NT	28.07.2022	Draft version ready for internal review
0.5	UPM, ASM	29.07.2022	First draft internal review
0.5	NT	03.08.2022	Comments partially addressed
0.6	Hypertech	04.08.2022	Final internal review
0.8	NT	14.08.2022	Additional comments addressed
0.9	NT, Hypertech, UEDIN	19.08.2022	Final version
1.0	NT, Hypertech	19.08.2022	Submission to the EC portal

Disclaimer

Executive Summary

The document describes the first version of the Adaptive Workflow Management and Automation Tool, which will orchestrate the construction process tasks and facilitate information exchange between them. It will further enable stakeholders to keep track of the entire process in real time as well as automated reporting and adaptation of the workflow in case of planned or unplanned developments during the construction phase.

The provided tool is a standalone toolset based on the I3D platform adjusted for the needs of COGITO, which provides the set of tools covering the whole life cycle of the management and utilisation of the know-how. In this version, the following main functionalities are provided:

- Importing the workflow created in the Process Modelling and Simulation tool (PMS);
- Management of the workflows;
- Creation of Work Orders that are issued when executing the workflows;
- Assignment of workorders and tasks to specific human resources;
- Monitoring and management of workorders and workflow tasks; and
- Exporting data of workorders to the Digital Twin Platform.

This first version focuses on the definition of internal data structure and adjusting the existing I3D ecosystem to meet the COGITO requirements. The second release will implement all the functionalities required for interfacing with other COGITO tools as defined in the use cases "UC1.1-Efficient and detailed project workflow planning using the project's construction schedule and as-planned BIM model" and "UC1.2-Systematic and secure execution, monitoring and updating of the project workflow".

The Adaptive Workflow Management and Automation tool development has been driven by the stakeholder's requirements identified in "D2.1-Stakeholder requirements for the COGITO system" and the tool specifications that have been documented in "D2.5-COGITO system architecture v2".

Table of contents

List of Figures

List of Tables

List of Acronyms

Term	Description
BCSC	BlockChain network Smart Contracts
BIM	Building Information Model
BPMN	Business Process Model and Notation
COGITO	Construction Phase diGItal Twin mOdel
DTP	Digital Twin Platform
I3D	I3D industrial services
IoT	Internet of Things
KPI	Key Performance Indicator
PMS	Process Modelling and Simulation
RAMS	Risk Assessment (and) Method Statement
SaaS	Software as a Service
SLAM	Service Level Agreements Manager
WO	Work Order
WODM	Workflow Management and Automation
WOEA	Work Order Execution Assistance

1 Introduction

1.1 Scope and Objectives of the Deliverable

The purpose of this deliverable is to document the first release of an Adaptive Workflow Management tool, named hereafter Work Order Definition and Management tool (WODM), that orchestrates the construction process tasks. It enables stakeholders to keep track of the entire process in real time as well as timely report and adapt the workflow in case of planned or unforeseen changes during the construction phase. The first activity of this task is the definition and implementation of the concrete, executable workflow based on the stakeholders' requirements collected and documented in the deliverable D2.1 (Stakeholder requirements for the COGITO system) as well as the outcomes of T6.2 (Adaptive Processes/Workflow Modelling and Simulation-based Optimization) in terms of construction process analysis.

Built upon an existing solution that is briefly described in the following sections, extensions and adjustments are being implemented to meet the COGITO requirements for workorder modelling and monitoring, and interactions with other components of the COGITO ecosystem. In parallel, interfaces to applications developed in T6.4 are also being modified to enable an automated monitoring and reporting of the construction works. In this version automatic data exchange is not implemented, however, it provides the possibility of manual data upload for testing purposes.

1.2 Relation to other Tasks and Deliverables

This deliverable is closely related to other WP6 tasks, namely "T6.1-Blockchain & Smart Contracts on the Workflow Modelling and Management", "T6.2-Adaptive Processes/Workflow Modelling and Simulation-based Optimization" and "T6.4-Personalized On-site Works Support and Relevant Apps Development" and their main outcomes (corresponding deliverables). It is also related to task "T7.1-Digital Twin Platform Design & Interface Specification" and its deliverables. The end-user requirements for WODM were gathered and described in "D2.1-Stakeholder requirements for the COGITO system". Furthermore, the specifications, the functional and non-functional requirements, as well as the interactions of WOEA (which is a frontend of WODM used on mobile devices by on-field stakeholders like workers, inspectors etc.) with other components of the COGITO ecosystem are presented in "D2.5-System Architecture v2".

1.3 Structure of the Deliverable

This deliverable contains following sections:

- prototype overview;
- technology stack and implementation tools used;
- API documentation;
- licensing information;
- installation instructions;
- development and integration status overview;
- requirements coverage; and
- assumptions and restrictions.

2 WODM – Work Order Definition and Management tool

The development and delivery of WODM has been based on an existing workflow execution engine, called I3D Platform, provided by NT. The following key sub-components of the I3D Platform are kept, refined and repurposed to meet the tool requirements that have been detailed in D2.1 and D2.5:

- services to define and manage the templates of work processes, enriched with an Extract Transform Load (ETL) tool to import work processes in Business Process Model and Notation (BPMN) format and automatically translate its content to populate the WODM's data model; this update will streamline the interactions and data exchange with the Process Modelling and Simulation tool (PMS).
- services to issue executable work processes – work orders based on templates. Once the process template is defined, it can be used to generate running instances.
- services for the execution of work processes in semi-automatic way with the options to enable user interactions with running workorders and tasks.

The I3D Platform functionalities and services that are not relevant to the scope of COGITO have been removed, i.e., location manager and editor. Other functionalities are considered as temporary solutions, for instance the internal I3D Identity Management, and will be removed as they will be managed by different COGITO tools.

2.1 Prototype Overview

The data structure of the I3D platform is depicted in Figure 1. A workflow, as well as its executable instance, i.e. workorder, consists of a set of steps. Every step consists of several actions (typically working instructions). Step are connected to a workspace. The predefined way of execution of a work process is sequential execution of every action, step by step. This sequence can be adjusted with events and preconditions, which allow to skip the execution of any of the actions.

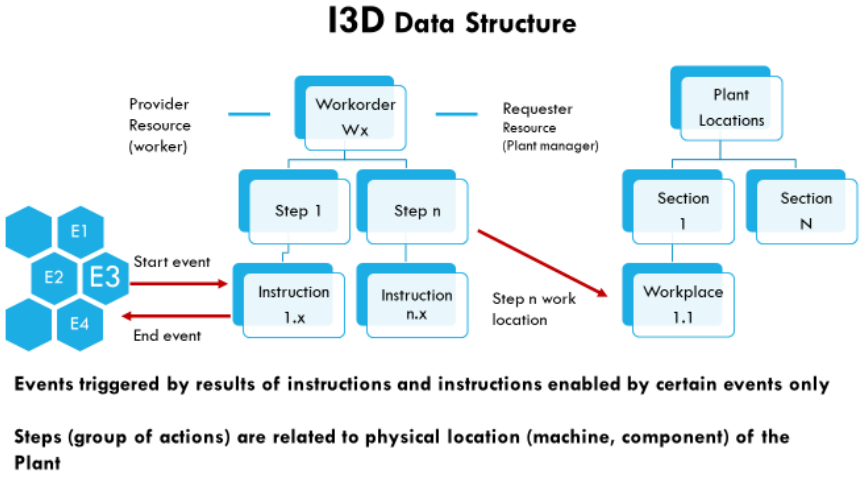


Figure 1 – Data structure of the I3D platform

Since the WODM tool constitutes a revised and updated version of I3D platform, it follows its own terminology, not fully aligned with the COGITO's terminology. Therefore, an abridged description of some key terms (workflow, workorder, step, action) is given below.

In I3D, the term **Workflow** is used for a work process definition, a template, which contains a step-by-step definition of a work process defining the "where, how and what" that needs to be done and "with what" resources. In terms of COGITO, this is the process model generated by the PMS. Workflows (with prefix W), and their relations with **Steps** (with prefix S) and **Actions** (with prefix A), as they visualised in the WODM UI, are illustrated in Figure 2. Tasks defined in COGITO are related with Steps in I3D platform.

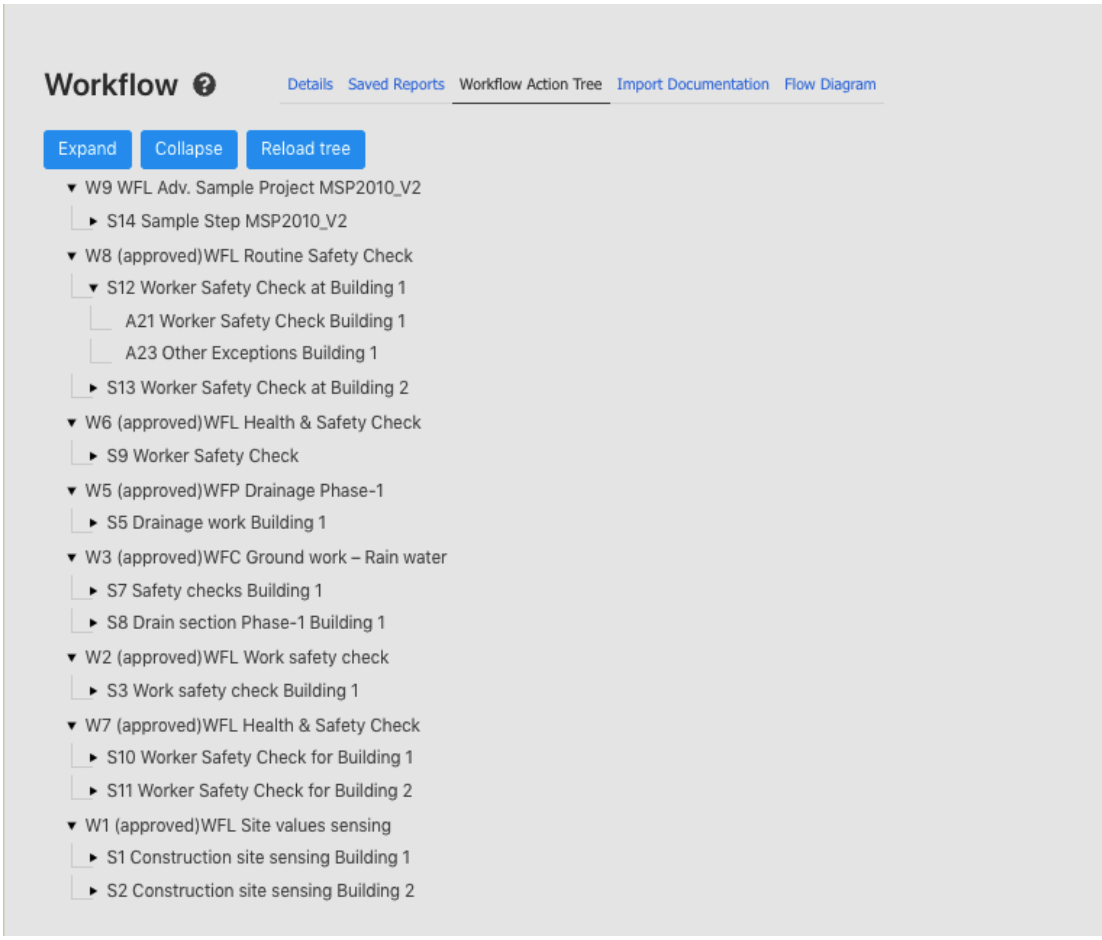


Figure 2 – Workflow tree example

The status of a workflow can be draft, approved or expired. A workflow is closed only if its status is set to "approved", meaning that it is protected against any unauthorised changes. Only an approved workflow can be used to generate a workorder. Expiration of workflow is useful when external optimisations of a process is ongoing. In that case the old workflow is expired and replaced by a new, optimized instance.

Every workflow has one or more **Steps**. Each step is linked to a specific location. Every step has its own sequence number which defines sequence of the steps inside workflow. Where needed, steps can be used to group actions that are executed at the same place.

Every step consists of one or more **Actions** (instructions). An action represents the exact work that must be done. An example of an action is depicted in Figure 3. The action has a short name and long description, and an expected duration which expresses the time required to complete the action. Specific human and equipment **Resources** are assigned to each action that will be used to execute it. Furthermore, each action can have events. **Events** are rules that are defined and assigned to an action to evaluate its results.

Workorder is the executable instance of workflow. In other words, it it's the actual implementation (or instantiation) of a plan of work that includes information about the actions that need to be performed by specific workers within the predefined timeframe. The aim of the workorders management in COGITO is to have exact information of their statuses and continuously optimising all workorders according to changing conditions.

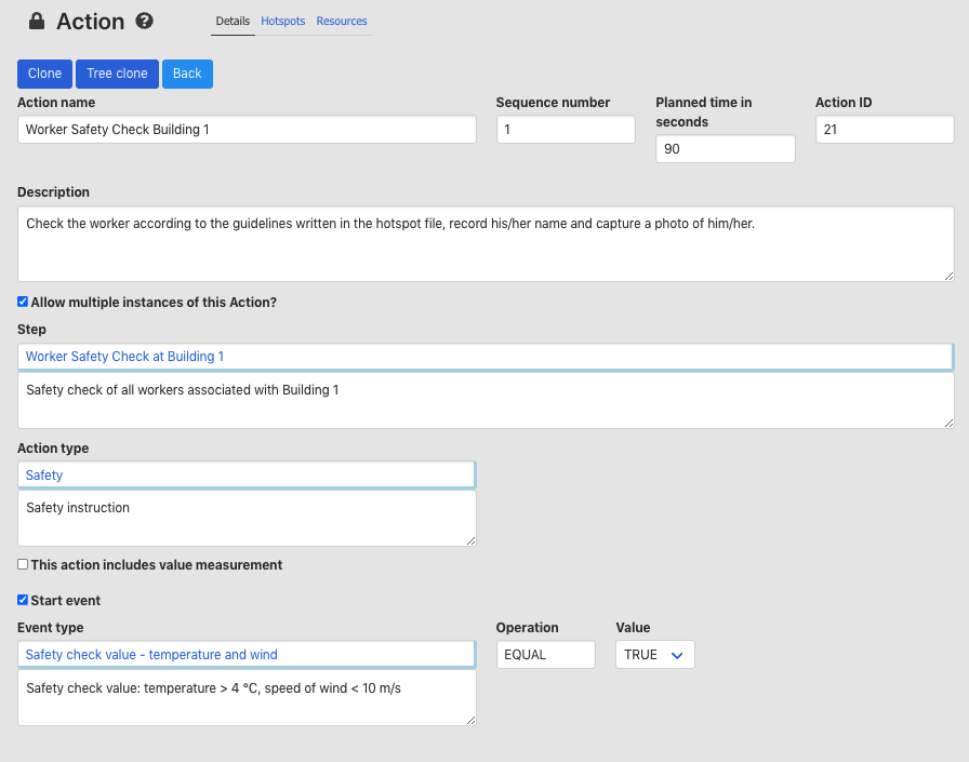


Figure 3 – An Action instance

2.2 Technology Stack and Implementation tools

WODM is a backend solution assisted by two frontend components: (1) the WOEA application, documented in "D6.7-Personalized On-site Works Support App v1"; and (2) the WODM UI which is described in "D6.9-Workflow User Interface for Project Managers v1". The WODM's deployment architecture has been based on state-of-the-art technologies, as Figure 4 depicts. The data storage layer utilises PostgreSQL to store project specific data provided by other COGITO components or generated by WODM backend. As for the business logic layer, to host all the ETL tools developed to interface with other COGITO components, a JAVA application server and a JAVA web server are deployed. HIBERNATE is used to achieve the connection with the Data Storage Layer. Finally, a webservice enables the data exchange with wearable devices, where WOEA is installed.

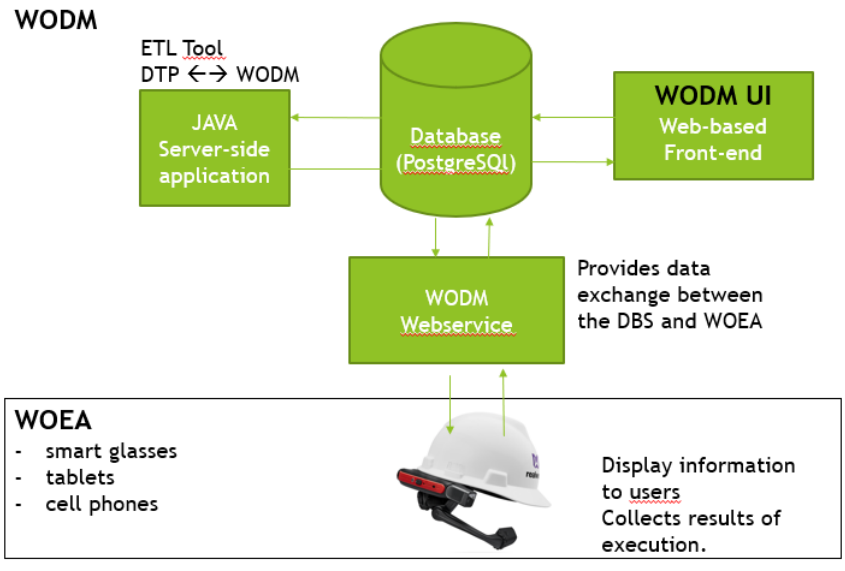


Figure 4 – WODM deployment architecture

All the libraries and frameworks used for developing WODM are summarised in Table 1.

Table 1 – Libraries and Frameworks used in WODM

Library/Technology Name	Version	License
Angular 7.2	7.2	MIT licence
Ng2-charts	4.0.0	ISC licence
Angular-tree-component	8.2.0	MIT licence
@ngx-translate/i18-polyfill	1.0.0	MIT licence
Ng2-cytoscape	0.4.0	MIT licence
Angular2-query-builder	0.5.0	MIT licence
jquery	3.6.0	MIT licence
PHPMailer	6.0.7	LGPL 2.1

Angular is a development platform, built on TypeScript. As a platform, Angular includes: (1) a component-based framework for building scalable web applications, (2) a collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more; and (3) a suite of developer tools to help develop, build, test, and update the code

Ng2-charts is a library to create charts based on Chart.js

Angular-tree-component allows users to represent hierarchical data in a tree-view structure, maintaining parent-child relationships, as well as to define static tree-view structure without a corresponding data mode.

@ngx-translate/i18-polyfill is an extraction tool and service to add support for code translations in Angular.

Ng2-cytoscape is an Angular wrapper around the CytoscapeJS 3 module. This library enables to add Cytoscape-based graph visualizations into Angular application, with values supplied by instance variables.

Angular2-query-builder - A modernized Angular 4+ query builder based on jQuery QueryBuilder. It provides support for heavy customization with Angular components and a flexible way to handle custom data types.

Jquery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

PHPMailer is a code library to send emails safely and easily via PHP code from a web server.

2.3 Input, Output and API Documentation

WODM, as an orchestrator of the workflow management and execution, interacts with all relevant services or tools. Namely they are:

- DTP (Digital Twin Platform): DTP interacts with WODM as Identity Provider and As-planned resources provider. WODM is sending task progress and information of allocated resources with associated SLAs.
- PMS (Process Model and Simulation): PMS is providing Process Model and WODM sends Task Progress update to PMS.
- SLAM (Service Level Agreement Manager): SLAM is providing KPI and SLA definition, WODM sends workorders with associated SLAs and KPIs.
- BCSC (Blockchain - Smart Contract): BCSC sends SLAs performance and WODM sends updated Work Orders with associated SLAs and KPIs.
- WOEA (Work Order Execution Assistance): This is internal communication within WODM tool, on-field device with WOEA tool receiving assigned task to be executed by user and sends task status to WODM.

These interactions, introduced in "D2.5-COGITO System Architecture v2", are depicted in Figure 5. In this section, examples of the payload formats that have been defined and agreed among the involved partners for each data interaction with WODM before its first release, are provided. The examples will be further elaborated in the next version of WODM's documentation (D6.6), planned to be released in October 2022.

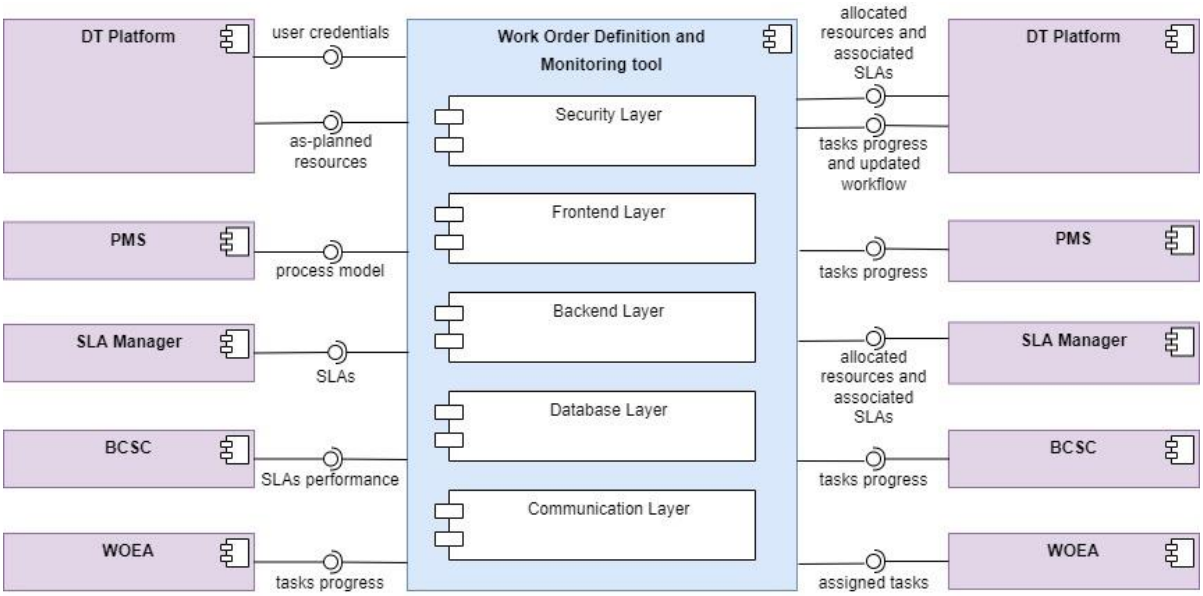


Figure 5 – WODM component diagram

The sequence diagrams of UC 1.1 and UC1.2, thoroughly described in "D2.5-System Architecture v2", are illustrated in Figure 6 and Figure 7, respectively. Each arrow represents a data flow (exchange) and has a number. Those numbers will be referenced in the following data exchange descriptions.

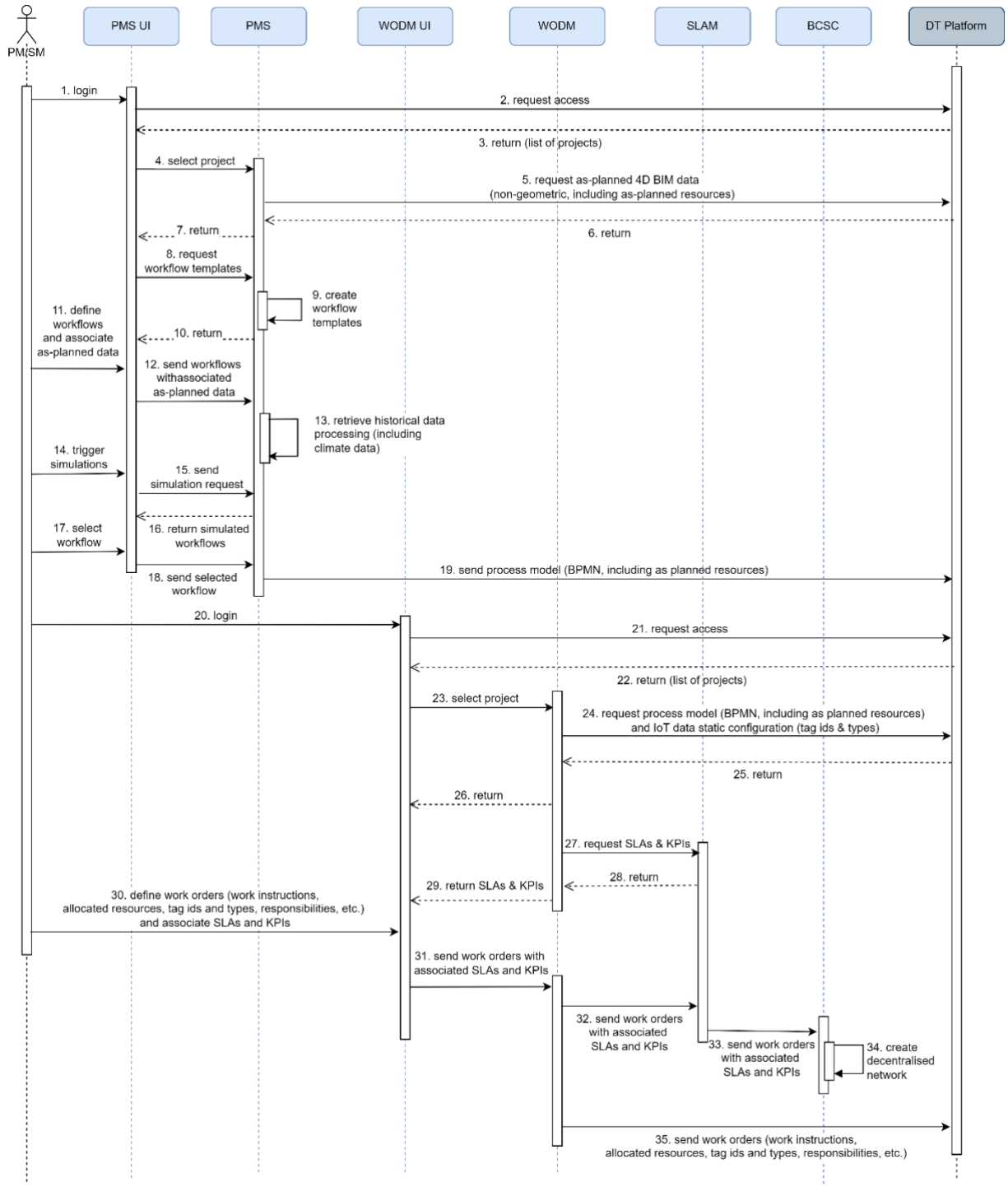


Figure 6 – Sequence diagram of UC-1.1

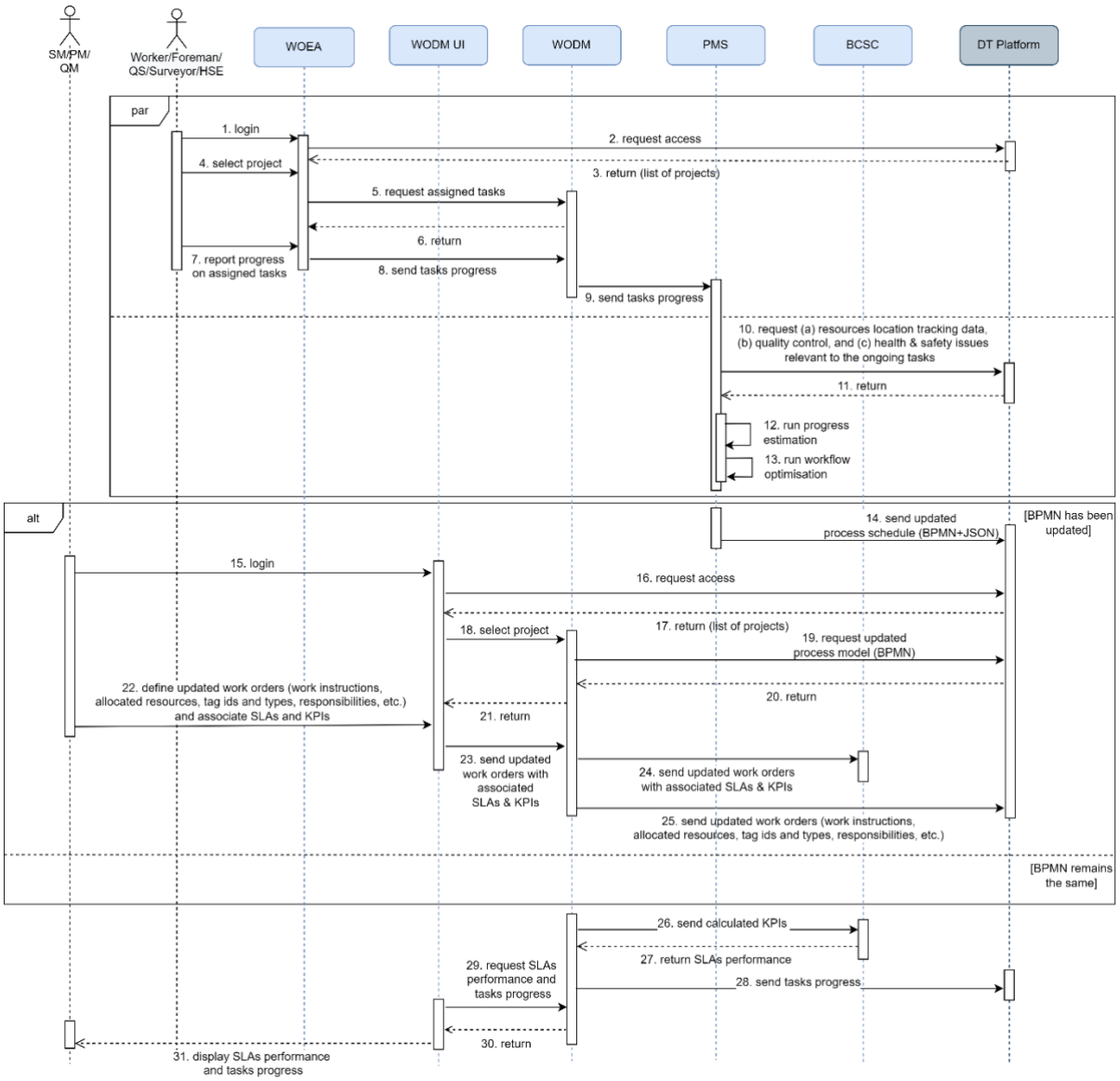


Figure 7 – Sequence diagram of UC-1.2

From DTP to WODM – planning phase (UC1.1 exchange 24, 25): The main input for WODM is the process model designed in the PMS. In the existing solution, we use an ETL service for importing various BPMN files. Since DTP acts as the central repository of COGITO, the involved partners aggreged that the PMS-to-WODM interaction should be realised through DTP. The JSON format that will be used for that data exchange is shown in Figure 8. In the next version of the WODM, a similar ETL service will be implemented to automatically process JSONs of that structure.

```
{
    "project_id": "project_id_1",

    "tasks": {
        "task_id_1": {
            "name": "Task name",
            "start_time": "",
            "end_time": "",
            "previous_task_list": ["task_id_3", "task_id_4
            "parent_task": "",
            "sub_task_list": ["", ""],
            "zone": "zone_id_1",
            "resource_list": [{
              "resource_id": "resource_id_1",
              "quantity_needed": 1
            }, {
              "resource_id": "resource_id_2",
              "quantity_needed": 2
            }]
        }
    },
    "resources": {
        "resource_id_1": {
            "name": "",
            "type": "human/equipment",
            "quantity": 0,
            "cost_per_hour": 0
        }
    },
    "tags": {
        "tag_1": {
            "group": "group_id_1"
        }
    },
    "tag_groups": {
        "group_id_1": {
            "type": "human/equipment",
            "name": "Concrete mixer truck"
        }
    },
    "users": {
        "user_1": {
            "email": "",
            "first_name": "",
            "last_name": "",
            "roles": ["keycloak_role_id_1", "keycloak_role_id_2
        },
        "user_2": {
            "email": "",
            "first_name": "",
            "last_name": "",
            "roles": ["keycloak_role_id_1"]
        }
    },
    "user_roles": {
        "keycloak_role_id_1": {
            "name": "role_name"
        },
        "keycloak_role_id_2": {
            "name": "role_name"
        }
    }
}
```


Figure 8 – DTP to WODM JSON: planning phase

From SLAM to WODM (UC1.1 exchange 27,28): SLAM provides the WODM with the KPIs definition. These KPIs represent measurable performance of construction tasks. This exchange is not implemented yet, but manual input has been tested. In the next version of the tool automatic exchange will be provided. The structure of the JSON file is presented in Figure 9.


```
{
  "KPI": [
    "Percentage of worked hours",
    "Percentage of completed work",
    "Percentage of labor downtime",
    "Percentage of equipment downtime",
    "Percentage of available Workers",
    "Percentage of available equipment",
    "Percentage of received materials",
    "Percentage of used materials",
    "Percentage of passed site inspections"
  ],
  "target_value": "target value for KPI",
  "rule": [
    "more than",
    "less than",
    "equal to"
  ]
}
```


Figure 9 – SLAM to WODM JSON

From WODM to SLAM (UC1.1 exchange 32): This data exchange provides the SLAM with the WODM's workorders with their directly associated KPIs and SLAs. The associations are made on the side of WODM after it receives the list of KPIs and SLAs from the SLAM sooner (*UC1.1 exchange 27,28)*. Associations between the workorders and the received KPIs and SLAs on the side of WODM will be implemented in next release. The structure of the JSON file is presented in Figure 10.

```
{
    "slaID":"SLA ID",
    "parties":
        [{
            "partyId": "default partyId",
            "partyRole": "default partyRole"
        }],
    "sla_startDate": "default start date",
        "sla_endDate": "default end date",
        "sla_description": "default description",
    "task": [{
                "taskId": "default task ID",
                "taskDescription": "default task description"
    }],
    "serviceLevelObjectives": [
        {"KPI":"selected kpi",
        "target_value":"selected target value",
        "rule": "more than / less than / equal to",
        "KPI_value":"current value",
        "taskId": "default task ID",
        "weighted factor":1}],
    "equipment":[{
          "equipmentID":"equipment ID",
          "name": "resource name"
        }]
      }
}
```


Figure 10 – WODM to SLAM JSON

From WODM to DTP – planning phase (UC1.1 exchange 35): In this data exchange, the WODM sends to the DTP all workorders and associated data, including: the instructions, all allocated resources, linked tag ids and types, all worker responsibilities, and further details. These would usually be either automatically linked to the workorder, or manually set by the site manager (through WODM UI). Some of the details are already implemented and are assigned automatically or manually in the WODM, the rest still needs to be integrated in the next version of the WODM tool. The structure of the JSON file is presented in Figure 11.


```
{
  "project_id": "project_id_1",
  "workorders": {
    "wodm_workorder_id_1": {
          "main_provider": "human_instance_id_1",
          "task_list": [],
      "start_time": "",
      "end_time": ""
    }
  },
  "tasks": {
    "task_id_1": {
      "name": "Task name",
      "start_time": "",
      "end_time": "",
          "parent_task": "task_id_9",
          "zone": "zone_id_1",
          "previous_task_list": ["task_id_3","task_id_4"],
          "sub_task_list": ["task_id_6","task_id_7"],
          "equipment_list": ["wodm_equipment_id_1","wodm_equipment_id_2"],
          "human_list": ["wodm_human_instance_id_1"]
    }
  },
  "equipment_instances":{
        "wodm_equipment_id_1":{
              "name":"resource name",
              "tag":"tag_id_1"
        }
  },
  "human_instances":{
        "wodm_human_instance_id_1":{
              "email": "user@email.com",
              "tag": "tag_id_2"
        }
  }
}
```


Figure 11 – WODM to DTP JSON: planning phase

From BCSC to WODM (UC-1.2 exchange 27): This data exchange provides SLA performance for WODM. The BCSC to WODM data exchange has not been implemented yet; Manual testing has been performed. In the next version automatic import will be adopted. The structure of the JSON file is presented in Figure 12.

```
{
    "slaID":"SLA ID",
    "parties": [{
        "partyId": "default partyId",
        "partyRole": "default partyRole"
     }],
    "sla_startDate": "default start date",
        "sla_endDate": "default end date",
        "sla_description": "default description",
    "task": {
                "taskId": "default task ID",
                "taskDescription": "default task description"
    },
    "serviceLevelObjectives": [{
        "KPI":"selected kpi",
        "target_value":"selected target value",
        "rule": "more than / less than / equal to",
        "KPI_value":"current value",
        "taskId": "default task ID",
        "weighted factor":1
        }],
    "equipment":[{
        "equipmentID":"equipment ID",
        "name": "resource name"
          }],
      "slaPerformance": "default performance"
}
```


Figure 12 – BCSC to WODM JSON

From WODM to WOEA (UC1.2 exchange 5,6): The internal communication between WODM and WOEA aims to provide the workers with the workorders and their data. The WOEA receives all relevant workorders assigned to the specific worker who has logged into the application.

From WOEA to WODM (UC1.2 exchange 8): The WOEA periodically sends all task progress updates to the WODM, which calculates all KPIs and handles the data automatically. This data exchange is already implemented, as a part of the I3D toolkit.

From WODM to PMS UC1.2 exchange 9): After the workorder's progress is reported through WOEA, it is automatically reflected in the WODM via an internal data exchange. After this, the WODM sends the progress to the PMS. This data exchange still needs to be automated on the WODM's side and will be implemented in next release. The structure of the JSON file is presented in Figure 13


```
{
    "process_id": "",
    "process_name": "",
    "execution_status": "ENUM",
    "planned_start": "",
    "planned_finish": "",
    "actual_start": "",
    "actual_finish": "",
    "provider_id": "",
    "provider_email": "",
    "manager_id": "",
    "manager_email": "",
    "tasks": [{
            "i3d_result_id": "",
            "id": "",
            "name": "",
            "result": "",
            "i3_provider_login": "",
            "i3d_provider_email": "",
            "start": "",
            "duration": 0,
            "planned_start": "",
            "planned_end": "",
            "real_start": "",
            "real_end": ""
        }
    ]
}
```


Figure 13 – WODM to PMS JSON

From DTP to WODM (user authorisation) (UC1.2 exchange 16,17): This data exchange provides the WODM with an option to log in the users of the WODM and WOEA. Once they are successfully verified and logged in, the WODM receives the user roles and groups linked with the specific users. This data exchange will be implemented in the final version.

From DTP to WODM – construction phase (UC1.2 exchange 19,20): The WODM request the process model including as planned resources, and IoT data static configuration (tag ids & types) from the DTP, which automatically sends the data back. This data exchange has not been implemented yet, so for now it is simulated. The structure of the JSON file is presented in Figure 14.


```
{
    "project_id": "project_id_1",

    "tasks": {
        "task_id_1": {
            "name": "Task name",
            "start_time": "",
            "end_time": "",
            "previous_task_list": ["task_id_3", "task_id_4"],
            "parent_task": "",
            "sub_task_list": ["", ""],
            "zone": "zone_id_1",
            "resource_list": [{
              "resource_id": "resource_id_1",
              "quantity_needed": 1
            }, {
              "resource_id": "resource_id_2",
              "quantity_needed": 2
            }]
        }
    },
    "resources": {
        "resource_id_1": {
            "name": "",
            "type": "human/equipment",
            "quantity": 0,
            "cost_per_hour": 0
        }
    },
    "tags": {
        "tag_1": {
            "group": "group_id_1"
        }
    },
    "tag_groups": {
        "group_id_1": {
            "type": "human/equipment",
            "name": "Concrete mixer truck"
        }
    },
    "users": {
        "user_1": {
            "email": "",
            "first_name": "",
            "last_name": "",
            "roles": ["keycloak_role_id_1", "keycloak_role_id_2"]
        },
        "user_2": {
            "email": "",
            "first_name": "",
            "last_name": "",
            "roles": ["keycloak_role_id_1"]
        }
    },
    "user_roles": {
        "keycloak_role_id_1": {
            "name": "role_name"
        },
        "keycloak_role_id_2": {
            "name": "role_name"
        }
    }
}
```


Figure 14 – DTP to WODM JSON: construction phase

From WODM to BCSC (UC1.2 exchange 24): This data exchange allows the WODM to send the updated workorders along with their linked SLAs and KPIs directly to the BCSC. It is not implemented on the side of WODM yet. The structure of the JSON file is presented in Figure 15.

```
{
    "partyId": "default partyId",
        "partyRole": "default partyRole",
    "slaID":"SLA ID",
    "timestamp":"default timestamp",
    "updatedKPI":[{
        "KPI":"selected kpi",
        "KPI_value":"current value",
        "taskId": "default task ID"
        }]
}
```

Figure 15 – WODM to BCSC JSON

From WODM to DTP (task progress) – construction phase (UC1.2 exchange 26): This data exchange works in a comparable way to the one described in previous section. While the previous one was more focused on the initial data transfer during the planning phase, this one will run periodically any time there is a progress in any of the workorder's tasks. Once the previous one will be fully integrated, this data exchange will be implemented too. The structure of the JSON file is presented in Figure 16.

```
{
    "project_id": "project_id_1",
    "tasks": { /*updated leaf tasks*/
        "task_id_1": {
            "real_start": null,
            "real_end": null,
                        "result":"TEXT",
                        "progress": %
        }
    }
}
```

Figure 16 – WODM to DTP (task progress) JSON: construction phase

2.4 Licensing

WODM is a closed source component.

2.5 Installation Instructions

WODM is provided as a SaaS (Software as a Service), thus installation or downloading of any component is not required. To access WODM and test the already implemented functionalities, please navigate its main UI, accessible through the following link: https://I3D.econtent.lu/i3d2/I3D-frontend/I3D-en-cogito/.

Demo user credentials are provided below:

- **Name**: CogitoUser
- **Password**: rdh486o38qw9sf4jz

After login, choose "Construction prototype". The specific project uses as input data imported from the DTP.

The application is tested on all relevant web browsers, Chrome, Edge and Firefox. WODM UI has not been widely tested on mobile devices. Android and Apple tablets with large screens are working well.

2.6 Development and integration status

The current version of WODM implements the main functionalities related to workflow management and workorder creation and monitoring. The integration with other components of the COGITO ecosystem has been partially tested based on manual data importing and exporting tests. In the next version, the automated data exchange with other tools will be implemented. For instance, WODM will interact with the identity provider of DTP to ensure secure access to data by authorised users, based on their roles, and authenticated clients. Furthermore, the ETL service to import workflows from DTP will be delivered. Finally, all the necessary REST communications with other COGITO tools will be implemented and tested. In parallel, WODM UI will be updated to accommodate all the changes on the WODM backend.

2.7 Requirements Coverage

The WODM tool is designed as a back-end COGITO solution, but it covers several of the requirements defined in D2.1 and D2.4. Table 14 presents the Stakeholders Requirements documented in D2.1 which are relevant for WODM. COGI-CS-1, COGI-CS-4 and COGI-CS5 are covered simply by the fact that UI is web based, COGI-CS-6 has not been tested on all Android devices, but it is working well on larger tablets. All other requirements are maintained by the principle of programming language of the application.

Table 2 – Stakeholders' requirements – computing systems

ID	Solution	Priority	Status
COGI-CS-1	runs on desktop or laptop PC	Must	Achieved Via WODM UI
COGI-CS-4	runs on Windows	Must	Achieved Via WODM UI
COGI-CS-5	runs on Mac	Could	Achieved Via WODM UI
COGI-CS-6	runs on Android	Would	Achieved Via WODM UI
COGI-CS-7	allows access to the whole data in one location	Must	Achieved
COGI-CS-8	maintains communication and data security	Must	Achieved
COGI-CS-9	differentiates data and system access levels and modification rights	Must	Achieved

Stakeholders' requirements about Workflow planning, Execution and Monitoring are defined in D2.1 and presented in Table 15. COGI-WF-5 is partially achieved as material usage and cost have not been tested yet. COGI-WF-6 is partially achieved due missing cost escalation. COGI-WF-28 and 29 requirements will be achieved in final version as original functionalities of I3D allowing these functions.

Table 3 – Workflow planning, execution and monitoring requirements

ID	Solution	Priority	Status
COGI-WF-1	allows the PM and Client to share information (design data, photos, videos, schedules, design issues, cost)	Could	Achieved
COGI-WF-2	allows the PM and SM to share information (design data, photos, RAMs, design issues, schedules, work orders, work reports, materials, schedule and usage, equipment usage, costs)	Must	Achieved
COGI-WF-3	allows the PM and the QS to share information (design data, photos, RAMS, design issues, schedules, work orders, materials schedule and usage, equipment usage, costs)	Should	Achieved
COGI-WF-4	allows the PM and QM to share information (design data, photos, design issues, schedules, work orders, materials schedule and usage, costs)	Should	Achieved
COGI-WF-5	allows the SM to share information with Subcontractors, Foreman, and Workers (design data, photos, RAMs, design issues, schedules, work orders, equipment usage)	Should	Partially achieved
COGI-WF-6	allows the PM and SM to efficiently detect and prioritise delays and cost escalation elements	Should	Partially achieved
COGI-WF-7	allows the PM to extract reports about project time performance, project cost performance, costs per unit, resource consumption	Must	Achieved
COGI-WF-9	issues work orders that include detailed method statements	Should	Achieved
COGI-WF-10	allows work orders assignment to specific workers/crews.	Should	Achieved
COGI-WF-11	updates the activity status during work execution and monitoring	Should	Achieved
COGI-WF-12	allows display of activity description, planned duration and activities relationship	Should	Achieved
COGI-WF-13	defines work orders containing work description, location, start and end, construction drawings or BIM, safety measures, materials and equipment needed, quality measures, etc.	Must	Achieved
COGI-WF-14	allows work progress reports	Must	Achieved
COGI-WF-15	updates work progress weekly	Must	Achieved
COGI-WF-16	updates the project schedule at least monthly	Must	Achieved

COGI-WF-18	displays only current information or document versions, related to the project, to all stakeholders	Must	Achieved
COGI-WF-19	enables quick and easy reporting.	Must	Achieved
COGI-WF-21	facilitates resource allocation during scheduling	Should	Achieved
COGI-WF-23	incorporates health and safety planning	Should	Achieved
COGI-WF-24	incorporates quality control planning	Should	Achieved
COGI-WF-25	accesses the project execution monitoring off site	Must	Achieved
COGI-WF-26	allows efficient reporting of work completion (using sensor data or simple app interface)	Could	Achieved via WOEA
COGI-WF-27	offers simple, easy to use, and intuitive interface to avoid workforce over-burdening	Must	Achieved via WODM UI
COGI-WF-28	facilitates pre-construction training sessions (e.g., by using BIM models in augmented reality)	Would	Not implemented yet
COGI-WF-29	facilitates swift tool adoption by easily available video tutorials and other learning materials online	Would	Not implemented yet

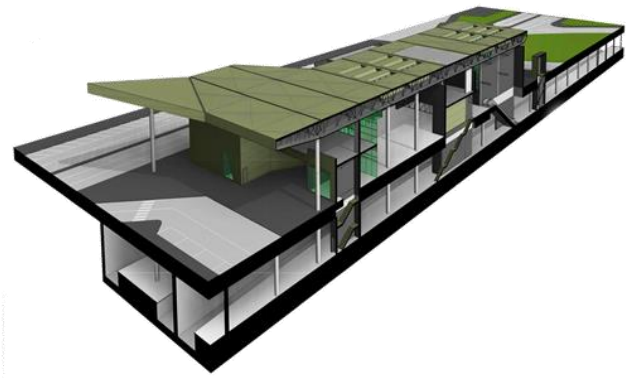
The functional and non-functional requirements, which are relevant to the WODM component, were documented in D2.5 and are presented in Table 16. Req-2.1 is achieved by WODM UI as WODM is only backend application.

Table 4 – Functional and Non-Functional Requirements coverage

ID	Description	Type	Status
Req-1.1	Ability to create work orders from workflows	Functional	Achieved
Req-1.2	Assigning multiple workers to work orders	Functional	Achieved
Req-1.3	Monitoring the work order and seeing the report	Functional	Achieved
Req-1.4	Editing the work order details during the runtime	Functional	Achieved
Req-2.1	User-friendly UI	Non-Functional	Achieved by WODM UI
Req-2.2	Scalability	Non-Functional	Achieved
Req-2.3	Stability	Non-Functional	Achieved

3 Conclusions

This demonstrator deliverable presents the current status of development of the WODM component which plays a key role in the workflow and workorder management. The component is an extended and repurposed version of an existing commercial platform, I3D, which provides functionalities tailored to COGITO requirements that have been elicited from the use-cases UC1.1 and UC1.2 sequence diagrams. In its current version, WODM's data exchange with other COGITO tools is partially covered. The next version will implement all the methods required to perform automated data exchanges with the relevant COGITO components. The second version of the WODM is expected to be released along with the corresponding deliverable "D6.6 – Adaptive Workflow Management and Automation Tool v2" in M24.



COGITO

CONSTRUCTION PHASE DIGITAL TWIN MODEL

cogito-project.eu

HYPERTECH energy labs

UCL

AARHUS UNIVERSITET

THE UNIVERSITY of EDINBURGH

BoC www.boc-group.com

iti

POLITÉCNICA

QUE TECHNOLOGIES

Novitech New Information Technologies

ASM

ferrovial construction

ΟΛΥΜΠΙΑ ΟΔΟΣ

RHOMBERG SERSA RAIL GROUP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310