



Circular and Dynamic Manufacturing Supply Chain Orchestration and Optimisation

D3.2 Supply Chain Orchestrator M24			
Report Identifier:	D3.2		
Work-package:	WP3	Task:	T3.1
Responsible Partner:	ED	Version Number:	1.0
Due Date	31/08/2025	Document Date:	12/12/2025
Distribution Security:	PUB	Deliverable Type:	DEM
Keywords:	Supply chain, synthesis, Business Process Modeling		
Project website: https://circuloos.eu/			

Document History

Version	Content & Changes	Issue Date
0.1	Document created	01/ 08/25
0.2	Document sent for review	05/ 12/25
0.3	Reviews are combined	09/ 12/25
0.4	Sent for Quality Assurance	12/ 12/25
1.0	Quality Assurance and Submission	12/12/25

Quality Control

	Organisation	Date
Editor	ED	01/ 08/25
Peer review 1	INCL	02/ 12 /25
Peer review 2	MWCB	09/ 12 /25
Authorised by (Technical Coordinator)	ED	12/ 12/25
Authorised by (Quality Manager)	ED	12/ 12/25
Submitted by (Project Coordinator)	ED	12/ 12/25

Legal Disclaimer

CIRCULOOS is an EU project funded by the Horizon Europe (HORIZON) research and innovation programme under grant agreement No. 101092295. The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. The information in this document is provided “as is” and no guarantee or warranty is given that the information is fit for any specific purpose. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. The CIRCULOOS Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

Copyright notice

© Copyright by the CIRCULOOS Consortium

This document contains information that is protected by copyright. All Rights Reserved. No part of this work covered by copyright hereon may be reproduced or used in any form or by any means without the permission of the copyright holders.

Table of Contents

1	Introduction.....	8
1.1	Project Introduction.....	8
1.2	What is the Process Orchestrator.....	9
1.3	Deliverable Purpose.....	9
2	Concept and value of the Process Orchestrator.....	10
2.1	Added value and features.....	10
2.2	Implementation guide.....	11
2.2.1	Process elements.....	11
2.2.2	Process design.....	12
2.2.3	Creating executable process models.....	13
2.2.4	Running environment.....	14
2.3	Integration with other CIRCULOOS components.....	15
2.4	Adopting the Process Orchestrator.....	15
3	Demonstrators.....	17
3.1	Demo workflow with external task.....	17
3.1.1	Requirements.....	20
3.1.2	Running.....	20
3.1.3	Accessing services.....	21
3.1.4	Configuration.....	21
3.1.5	Deploying the BPMN Process.....	21
3.1.6	Starting a process instance.....	21
3.1.7	Viewing emails.....	21
3.1.8	Stopping services.....	21
3.1.9	Directory structure.....	22
3.1.10	Troubleshooting.....	23
3.2	CIRCULOOS pilots.....	23
3.2.1	Description of the Process Orchestration in the pilots.....	23
3.2.2	Making the model executable.....	28
4	Conclusion.....	32

List of Figures

Figure 1: Main process model elements.....	11
Figure 2: BPMN 2.0 Cheat Sheet.....	12
Figure 3: Process design main steps.....	12
Figure 4: Creating executable process models main steps.....	13
Figure 5: Task List for human tasks.....	14
Figure 6: Process monitoring.....	15
Figure 7: Overall operation of the demo.....	17
Figure 8: Demo process that incorporates authentication, exchanging data with the CIRCULOOS Data Platform, Interaction with user via UI and email notifications.....	18
Figure 9: Claiming a task in the tasklist.....	19
Figure 10: Completing a human task.....	19
Figure 11: Email notification to complete a task.....	20
Figure 12: BPMN model for the Wood Pilot.....	24
Figure 13 BPMN model for the Plastic Pilot.....	26
Figure 14: BPMN model for the Leather Pilot.....	27
Figure 15: Executable model for the plastics pilot.....	28
Figure 16: Connecting a task to script (a) above, (b) below.....	29
Figure 17: JSON post from the Process Orchestrator to the CIRCULOOS Data Platform.....	30
Figure 18: JSON response that is fed to the Process Orchestrator by the CIRCULOOS Data Platform.....	30
Figure 19: Form for receiving material.....	31
Figure 20: Input form for rejecting a part.....	31

Abbreviations

BPM	Business Process Model
BPMN	Business Process Model Notation
CM	Circular Manufacturing
EXD	Experiment Demonstrator (from the Open Calls)
LCA	LifeCycle Assessment
NGSI	Next Generation Service Interfaces
RAMP	Robotics and Automation MarketPlace
REST	Representational State Transfer
SME	Small and Medium Enterprise

Executive Summary

This report presents the Process Orchestrator implemented within the CIRCULOOS project as one of the key enablers of circular and cooperative manufacturing supply chains. The Process Orchestrator is designed to model, execute and monitor processes both within a production process and between multiple stakeholders by using BPMN as its foundation. It acts as a core coordinating function for workflows, data synchronization and ensuring sustainability-oriented decision-making.

The deliverable includes the conceptual foundation, technical implementation and actual application of the Process Orchestrator. It begins with its introduction and value addition, particularly for manufacturing SMEs that are usually plagued by problems limited automation and non-interoperability across different stakeholders. The Process Orchestrator addresses these problems by providing a unified, scalable and transparent process execution environment supporting automated and human-in-the-loop operations. It enables supply chain partners to collaborate in near real time to coordinate tasks, decisions and data exchange synchronously and traceably.

A step-by-step implementation guide is provided, from process element modeling and BPMN workflow design to executable model construction and runtime setup. Integration with the Data Platform of CIRCULOOS is also highlighted, so as to enable data sharing and facilitate dynamic sustainability-driven decisions. Hence, environmental performance and lifecycle data are captured and communicated across the supply chain, reinforcing the circular economy's objectives of CIRCULOOS.

The report also points out the Process Orchestrator's functionality in real pilot cases, where it has been applied to handle processes in wood, plastic and leather manufacturing contexts. The pilots show how the Process Orchestrator enables material reuse and circular production patterns.

Briefly, Process Orchestrator is a driver of digital transformation and circularity for production. It allows manufacturing SMEs to move beyond single-site operations and become genuine circular manufacturing players within dynamic, data-driven and sustainable supply chains.

1 Introduction

1.1 Project Introduction

The overall vision of CIRCULOOS is to deliver the tools to enable MSMEs become full members of the Circular Manufacturing value chain. These tools orchestrate and continuously optimise the supply-chain end-to-end and integrate planning and execution monitoring to enable transparent and on-time communication. Combining these with direct calculation of the product sustainability and circularity profile, for both internal and external partners, this environment will enable them to configure and execute disruptive circular manufacturing processes for sustainable production that covers the entire life cycle of products; either by recovering the value of product that ended-up as waste or from recycled and remanufactured products.

To achieve this objective the project aims to deploy:

- Circular end-to-end supply chain orchestration for collaborative workflows which incorporates planning and execution metrics and integrates advanced and multimodal visualisation and analytics. The visualisation is delivered by comprehensive Digital Twins of the supply chains formulated, the factory processes and product design phases.
- Supply Chain Optimisation that monitors the global (across the supply chain) and local (within the factory) processes and execution, inputs and outputs and configuration parameters, to enable data-driven AI decision making, this way supporting continuous optimisation of targeted and measured performance and sustainability parameters.
- Dynamic Sustainability Assessment functionalities that investigate alternative supply-chain scenarios (varying in terms of materials used, processing technologies, suppliers involved and/or activated circular economy practices) in place of the existing schemes, quickly measuring their performance in terms of environmental sustainability and circular economy profile.
- Supply Chain Data Spaces for seamless, multi-level data flow across the supply chain partners, supporting the reuse of materials in novel products, the extension of the life-cycle of finished products (remanufacturing) and data-driven decisions for collaboration of parties offering matching services in the most dynamic and efficient way.
- Cybersecure and trustworthy data sharing across the supply chain by employing a distributed, trusted and efficient Identity and Access management system, that together with the associated trust framework will coordinate the identities of all IoT objects and ensure trustworthy data sharing among its members, aligned with the trust framework that is being implemented in EBSI.
- CM specific tools for the automatic recognition of recyclable parts by modern Machine Vision tools and Advanced Robotics, to enable optimised flows in the selection process.
- Novel circular business processes will be demonstrated supporting reusing, reducing and recycling material in production and consumption systems. The new collaborative production models will provide quantifiable results on the sustainability increase across the supply chain, in terms of efficient use of raw materials, of by-products, of waste and energy and of emissions reduction. CIRCULOOS leverages the above with the RAMP integrated innovation IOT platform and the European network around it to deliver a CM ecosystem and platform for Manufacturing SMEs.
- Skills upskilling and reskilling will be provided in RAMP and through online courses, webinars and best practice guides and success stories based on the pilots and Experiments for Demonstration (EXDs).

1.2 What is the Process Orchestrator

The Process Orchestrator aims to help manufacturing SMEs to achieve seamless coordination of complex processes among different parties in manufacturing supply chains. When circular economy's methodologies are utilized, linear processes are no longer sufficient to deal with sustainability and efficiency goals anymore. Instead, companies must possess dynamic integrated systems that can handle tasks, decisions and streams accurately in real time. The Process Orchestrator fulfills this need by providing full-scale modeling, execution and monitoring platform for processes modeled with BPMN. Through the integration of automation, human collaboration and external systems, the Process Orchestrator facilitates supply chains to collaborate in end-to-end ecosystems rather than silos. This is particularly beneficial for manufacturing SMEs, which are usually resource-constrained and fragmented in their digital infrastructures. With orchestration, those businesses are able to increase transparency, traceability and responsiveness and pave the way to actual circular and sustainable production models.

1.3 Deliverable Purpose

The purpose of this deliverable is to provide a general overview of the Process Orchestrator within the CIRCULOOS platform. It defines its concept, added value and technical implementation, detailing how it facilitates circular manufacturing and collaborative supply chain processes. The report explains the process of designing, deploying and executing orchestrated processes using BPMN models and the process execution engine. In addition, it specifies integration points with the rest of the CIRCULOOS components, such as the Data Platform and depicts real-world deployments through the pilot scenarios. Lastly, it aims to assist manufacturing SMEs with the information needed to deploy the Process Orchestrator to be able to implement it effectively and leverage its potential in order to realize digital transformation and sustainability.

2 Concept and value of the Process Orchestrator

2.1 Added value and features

The Process Orchestrator is designed to coordinate and automate end-to-end processes both within a manufacturer's production process as well as across multiple actors in a supply chain. It enables collaborative workflows, ensuring that tasks, decisions and data exchanges occur seamlessly and in real time. By leveraging Business Process Model and Notation (BPMN), the Process Orchestrator provides a standardized and visual representation of processes, making them transparent, traceable and understandable to stakeholders of different background expertise.

Its main features include:

- **Centralized Orchestration:** Manages complex workflows that may span across multiple organizations, ensuring that each actor's tasks are executed in the correct sequence and under the right conditions.
- **Decision Support:** Automates decision-making where possible and prompts human intervention when necessary, reducing delays and errors.
- **Data Consistency:** Ensures that all process-related data is synchronized, utilising the CIRCULOOS Data Platform.
- **Scalability:** Supports both local factory processes and cross-company supply chain processes, making it adaptable to different levels of granularity.

Process Orchestrator's main capabilities include:

- **BPMN-Based Modeling:** Uses BPMN for process design, enabling clear visualization of tasks, gateways, events and message flows.
- **Human and Automated Task Management:** Handles both automated service tasks (e.g., database updates, API calls) and user tasks requiring human input via integrated web user interface.
- **Integration Capabilities:** Integrates with the CIRCULOOS Data Platform where appropriate, for real-time data acquisition, data sharing and lifecycle tracking. It can also communicate with external systems and devices via service tasks and delegate code.
- **Dynamic Execution:** Executes processes through BPMN execution engine, which interprets BPMN models and manages task lifecycles. It supports event-driven orchestration, including time-based alerts and message-based triggers.
- **Monitoring and Control:** It provides dashboards for real-time process monitoring and allows human agents to interact with tasks through a task list application or custom graphical interfaces.
- **Exception Handling:** Incorporates event subprocesses and boundary events to manage errors, delays and alternative flows without disrupting the entire process.

The Process Orchestrator delivers significant value to manufacturing SMEs by simplifying the complexity of managing interconnected processes across multiple actors. SMEs often operate with limited resources and fragmented digital systems, which makes coordination and data sharing challenging. By providing a unified orchestration layer, the Process Orchestrator enables seamless integration of tasks, systems and decision points. This reduces operational inefficiencies, enhances collaboration among partners and ensures real-time visibility of workflows. As a result, SMEs can respond more quickly to changes in demand, material availability, while maintaining cost efficiency and improving overall competitiveness.

Beyond operational efficiency, the Process Orchestrator plays a pivotal role in supporting circular manufacturing. It enables the design and execution of processes that prioritize reuse, recycling and

remanufacturing, ensuring that secondary materials are reintegrated into production cycles. It enables the design of circular material and waste flows, helping companies to measure and optimize their environmental performance.

2.2 Implementation guide

The implementation of the Process Orchestrator involves translating business processes into executable workflows that can be managed, monitored and optimized. This section provides a structured approach to designing and deploying these processes, from defining their scope and modeling them in BPMN to creating executable models and configuring the runtime environment. By following these steps, organizations can ensure that their orchestration layer is both technically robust and aligned with the operational needs of the own production and the circular supply chains.

2.2.1 Process elements

The Process Orchestrator relies on BPMN as the standard for modeling and executing processes. BPMN provides a visual and semantic framework that ensures clarity, interoperability and automation readiness. Within this framework, several key elements are used to represent the logic, flow and responsibilities of the process.

Activities/Tasks are the fundamental building blocks of the process model. They represent units of work that can be automated, executed by a system, or performed by a human user. Tasks can include service tasks for database integration, script tasks for logic execution and user tasks for human interaction. **Gateways** define decision points and control the divergence or convergence of process flows, enabling conditional and parallel paths. **Events** capture occurrences that affect the process, such as start and end events, intermediate events for message reception and optional boundary events for timeouts or alerts.

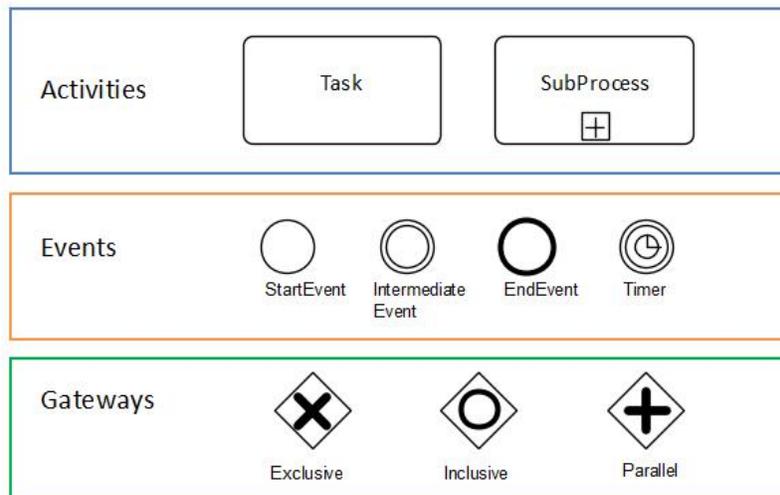


Figure 1: Main process model elements

Swimlanes and **Pools** are used to organize tasks by roles or actors, ensuring clear responsibility assignment across different participants in the supply chain. This is particularly important in CIRCULOOS, where multiple SMEs collaborate within a single orchestrated process. **Message Flows** complement this structure by representing the exchange of information between different pools, capturing the interactions between organizations or systems.

To support advanced orchestration, the Process Orchestrator also incorporates **execution listeners** and **task listeners**, which allow the injection of custom logic at specific points in the process lifecycle. These listeners can trigger notifications, update external systems, or enforce business rules dynamically. Additionally, **event subprocesses** can be embedded to handle exceptions or asynchronous messages without interrupting the main process flow, ensuring resilience and flexibility in real-world scenarios.

Figure 2 provides a consolidated cheat sheet to design process models with BPMN.

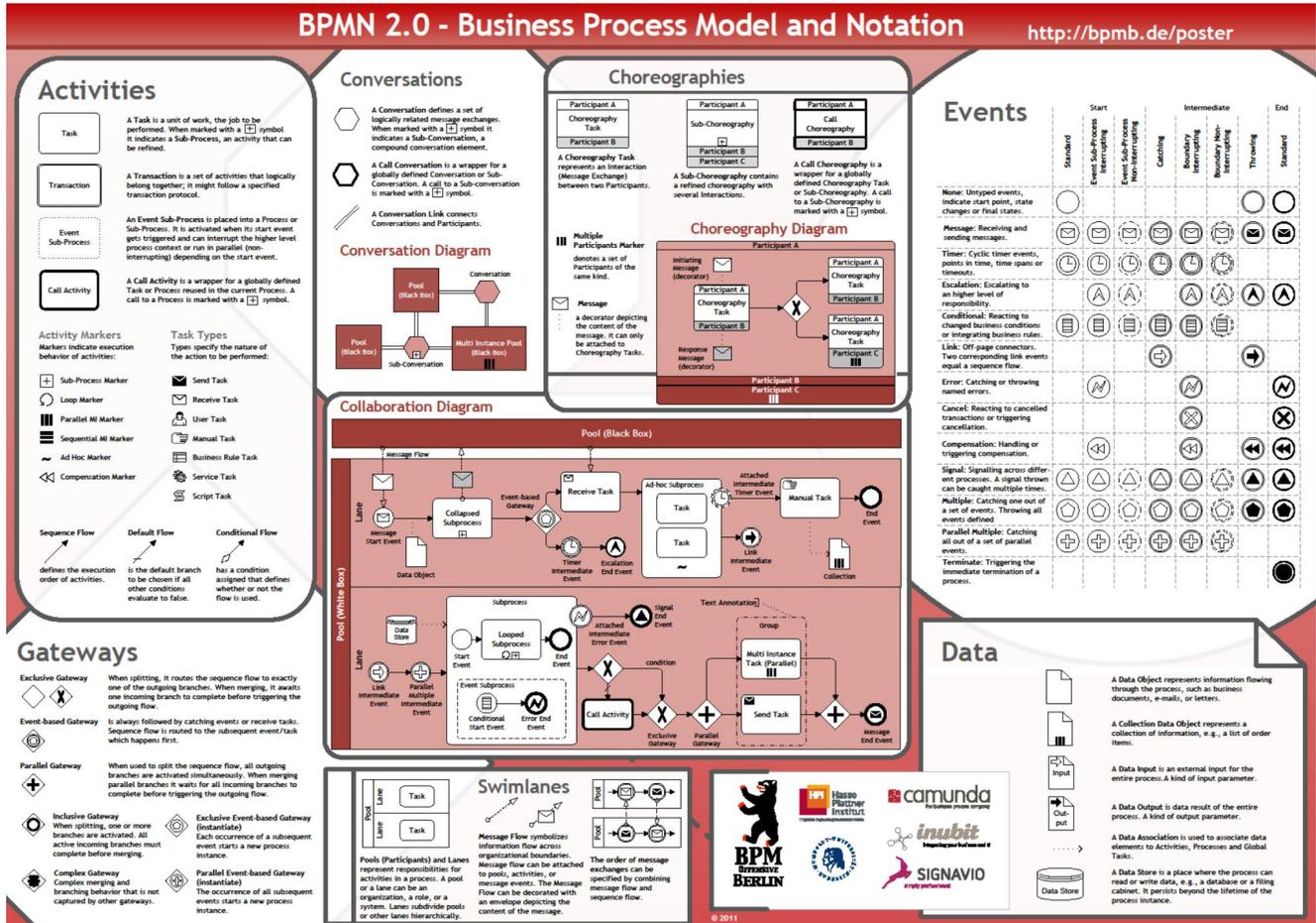


Figure 2: BPMN 2.0 Cheat Sheet

2.2.2 Process design

Figure 3 gives an overview of the main steps for designing a process model.

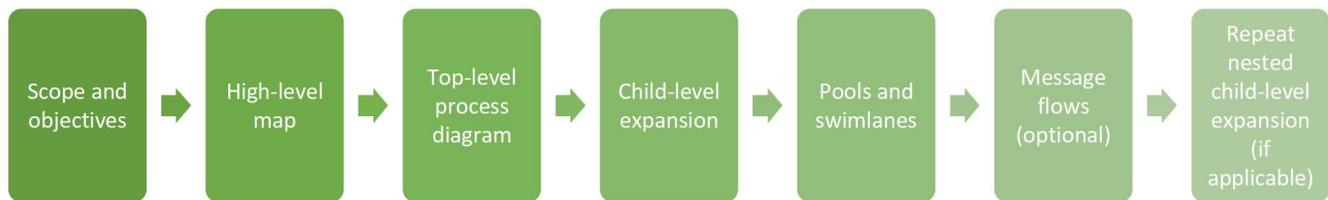


Figure 3: Process design main steps

The design begins with defining the **scope and objectives** of the model. This includes determining when the process starts, when it ends and what business case it represents. This includes identifying all possible end states and aligning the process with the physical and organizational hierarchy of the supply chain. Establishing a clear scope ensures that the model captures the entire end-to-end workflow rather than focusing only on isolated tasks.

Once the scope is defined, the next step is to create a **high-level process map**. This map should enumerate the major activities—typically no more than ten—that represent the core stages of the process. These activities are then arranged into a **top-level BPMN diagram**, where each activity is modeled as a sub-process. At this stage, gateways are introduced to represent conditional or parallel paths and pools and swimlanes are used to distinguish roles and responsibilities across different actors. The design begins with the “happy flow,” which represents the ideal execution path and then incorporates exceptional paths to handle deviations and errors.

After the top-level diagram is complete, each sub-process is expanded into **child-level diagrams** to capture detailed steps. These diagrams should maintain consistency with the parent-level end states and include all necessary user tasks, service tasks and message flows. **Message flows** are particularly important in multi-actor scenarios, as they represent the exchange of information between organizations or systems. The design process is iterative: additional **nested levels** can be added as needed and refinements are made to ensure that the lowest-level sub-processes contain executable tasks. This structured approach ensures that the final model is both comprehensive and automation-ready, providing a solid foundation for implementation in the Process Orchestrator.

2.2.3 Creating executable process models

Once the conceptual process design is complete, the next step is to transform it into an executable model that can run within the Process Orchestrator execution engine. This involves refining the BPMN diagrams to a level of granularity suitable for automation and specifying the technical details required by the execution engine. Figure 4 gives an overview of the main steps for creating executable process models.

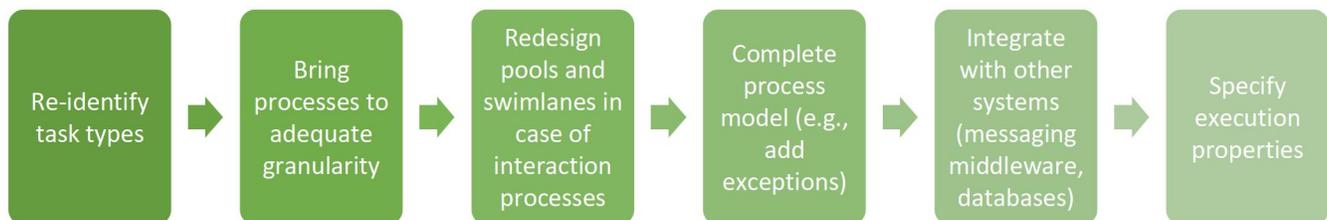


Figure 4: Creating executable process models main steps

The first activity is to **re-identify task types** and classify them as automated, user, or manual tasks. Automated tasks typically include service tasks for database integration, script tasks for business logic and call activities for invoking sub-processes or external services. User tasks are configured for human interaction, such as approvals or data entry and are linked to task lists or custom interfaces. Manual tasks, while not automated, are documented to maintain process completeness and traceability.

Next, the process must be brought to an **adequate level of granularity**. This means breaking down high-level activities into smaller, executable tasks that can be implemented within the orchestration engine. At this stage, **pools and swimlanes may need to be redesigned**, especially in processes involving multiple actors, to accurately represent roles, responsibilities and interaction points across organizations.

Once the structure is refined, the **process model is completed** by adding exceptions and alternative paths. This includes handling error conditions, timeouts and compensating actions to ensure robustness in real-world execution. After the logical flow is finalized, the model is **integrated with external systems** such as messaging middleware, databases and in general external platforms, enabling seamless data exchange and interoperability.

Finally, the executable model is **enriched with execution properties** that are not visible in the BPMN diagram but are essential for runtime behavior. These include implementation details such as delegate code references, input/output mappings and configuration of connectors for system integration.

By following these steps, the process model evolves from a conceptual representation into a fully executable workflow, ready to be deployed in the Process Orchestrator for real-time execution and monitoring.

2.2.4 Running environment

At runtime, the Process Orchestrator enacts the BPMN models created during the design phase. Each process instance is executed according to the defined logic, with automated tasks invoking delegate code for database operations, external system integration, or device communication. Human tasks are presented in a Task List through a web-based UI (Figure 5), enabling users to interact with the process by providing approvals, entering data, or resolving exceptions.

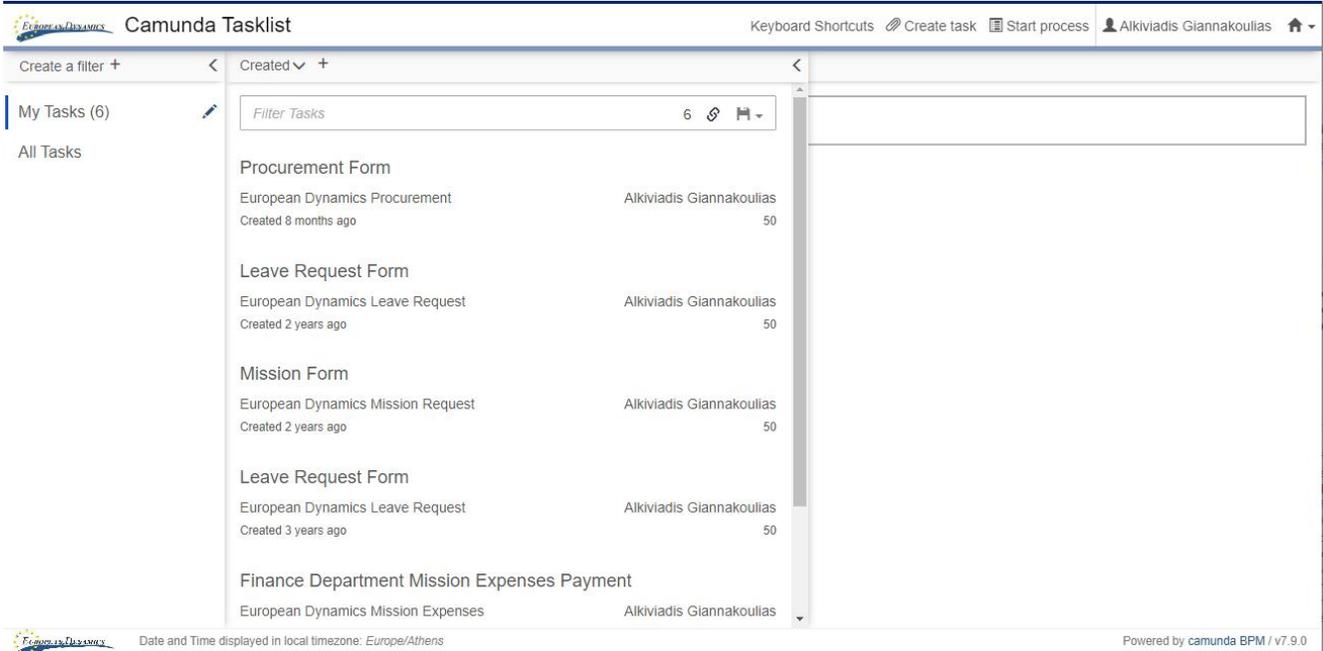


Figure 5: Task List for human tasks

For **monitoring**, the UI offers real-time visibility into process execution (Figure 6), allowing project managers to track progress, analyze performance and handle incidents.

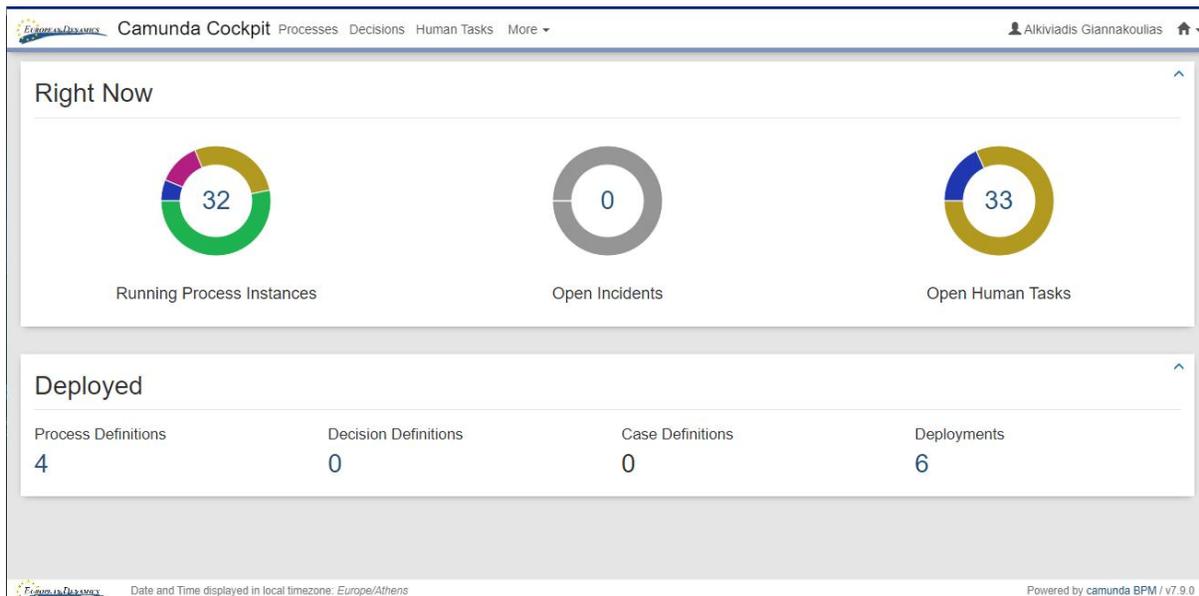


Figure 6: Process monitoring

2.3 Integration with other CIRCULOOS components

The Process Orchestrator is not limited to managing isolated factory operations. Its primary strength lies in coordinating end-to-end workflows across multiple actors in a supply chain. In the context of CIRCULOOS, this capability is essential for enabling collaborative circular manufacturing models, where different manufacturing SMEs contribute to a common value chain. By orchestrating tasks, decisions and data exchange across organizations, the Process Orchestrator ensures that processes remain synchronized, transparent and adaptable to real-time conditions.

Process Orchestrator is integrated with the CIRCULOOS Data Platform, which acts as the hub for all process data, material life cycle data and sustainability KPIs. Integration ensures that every action carried out within the Process Orchestrator is supported by accurate and timely data exchange across the supply chain. Every step of the process, such as material procurement, production, or shipping, having completed by the system, performs updates to the Data Platform for an up-to-date record of the material state, properties and environmental impact. Real-time synchronization enables transparency and traceability required for simultaneous workflows between several Manufacturing SMEs.

2.4 Adopting the Process Orchestrator

Adopting the Process Orchestrator within a manufacturing or supply chain context involves a structured approach that bridges business process modeling with technical implementation. The first step is to **model the physical hierarchy** and define the **end-to-end process scope**, ensuring that all relevant actors, systems and decision points are captured. This includes identifying and defining the tasks and agents involved, as well as designing the product data model, which represents the business-related information that will flow through the process.

Once the conceptual model is established, the next phase is to **create BPMN process models**. The process begins with high-level business models and progressively evolves into detailed executable models. These models incorporate user tasks, service tasks and message flows, ensuring that both automated and human interactions are represented. After modeling, the implementation phase involves developing process application projects in an integrated development environment (IDE), where delegate code is written to handle business logic, system integration and data exchange.

The final steps include **building and deploying** the process applications within the orchestration environment. The Process Orchestrator executes the BPMN models and manages runtime interactions. Human agents can perform assigned tasks through a task list UI or custom interfaces, while project managers monitor execution and performance. This setup ensures real-time visibility, exception handling and continuous improvement of processes.

For the CIRCULOOS context, additional considerations apply. The focus extends beyond local production processes to the entire supply chain, requiring orchestration across multiple organizations. Therefore, the level of detail in process design should emphasize critical decision points and data collection steps that impact sustainability and circularity. By following this adoption pathway, companies can leverage the Process Orchestrator to achieve seamless integration, enhanced collaboration and measurable progress toward circular manufacturing goals.

3 Demonstrators

3.1 Demo workflow with external task

This demo showcases the integration of the Process Orchestrator (Camunda 7), Spring Boot External Task Worker, Keycloak (for authentication and authorization), CIRCULOOS Data Platform and Mailpit for an event-driven human-workflow process. More specifically it incorporates:

- Secure token-based authentication (Keycloak)
- External task execution (Spring Boot)
- Data enrichment from CIRCULOOS Data Platform
- User task completion via Process Orchestrator Tasklist
- Email-based human approval loop (Mailpit)

The overall operation is depicted in Figure 7.

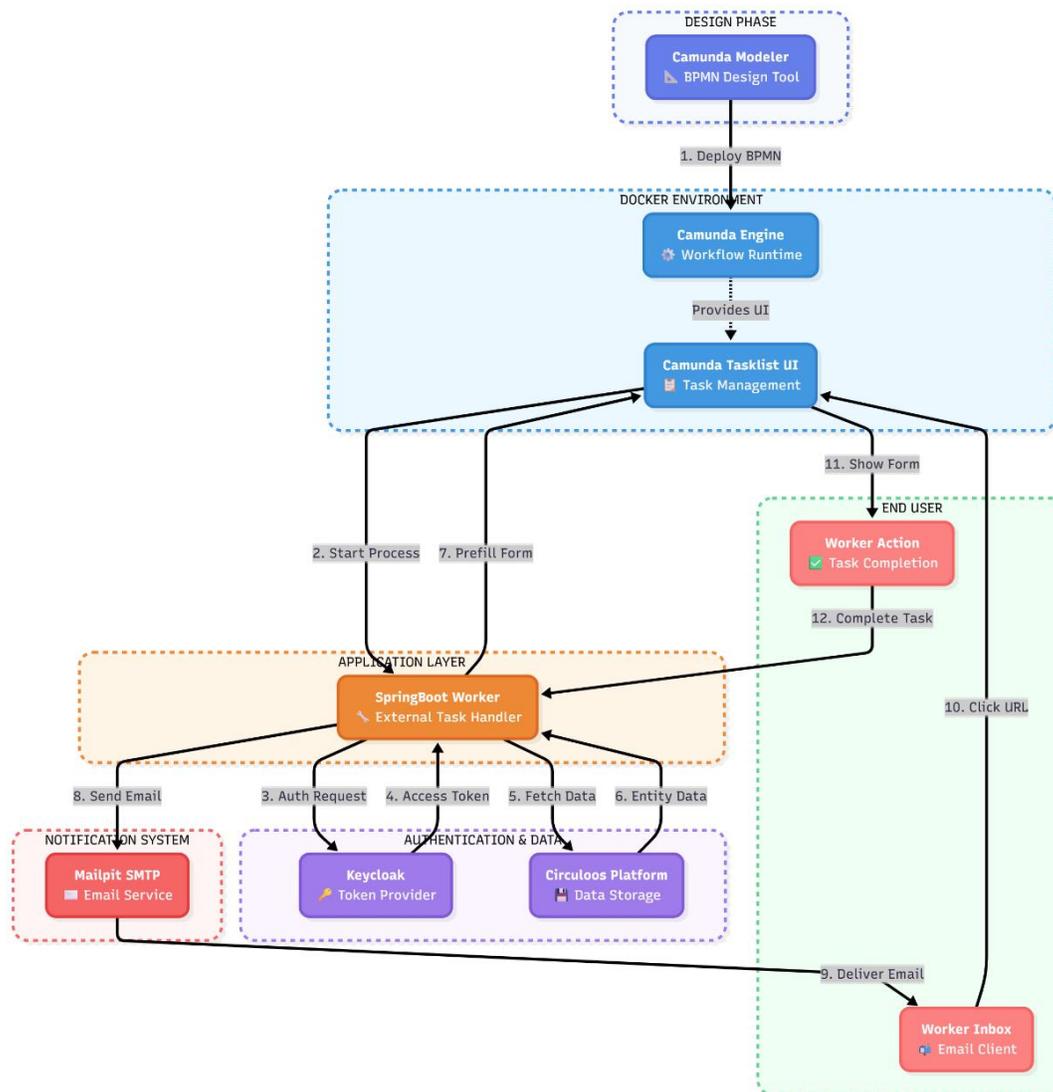


Figure 7: Overall operation of the demo

In summary:

1. The BPMN is deployed via Camunda Modeler (Process Orchestrator)
2. The Camunda Engine (Process Orchestrator) executes the workflow, generating the Tasklist UI
3. The Spring Boot worker fetches OAuth token from Keycloak
4. The Worker queries the CIRCULOOS Data Platform
5. The task form is pre-filled in the Tasklist
6. Email notifications are sent via Mailpit to the user
7. The user clicks link and completes the task
8. Workflow continues in Camunda (Process Orchestrator)

The following subcomponents are utilised:

Table 1: Subcomponents utilised in the demo

Layer	Component	Purpose
BPMN Design	Camunda Modeler	Process authoring
Workflow Engine	Camunda 7	BPMN runtime
User Task UI	Camunda Tasklist	Form interaction
Worker	Spring Boot External Task Worker	Executes service tasks
Auth	Keycloak	OAuth2 token provider
Data Platform	Circuloos Data Platform	Entity data retrieval
Notification	Mailpit SMTP	Local email delivery
User	Email client + Task UI	Approves/handles tasks

The demo process is illustrated in Figure 8.

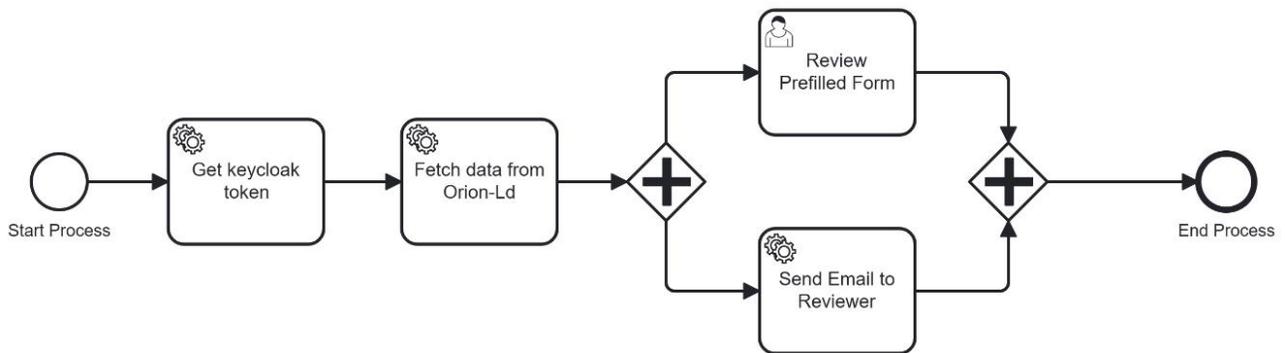


Figure 8: Demo process that incorporates authentication, exchanging data with the CIRCULOOS Data Platform, Interaction with user via UI and email notifications

Figure 9 and Figure 10 show how the tasklist is formatted to claim and complete a task.

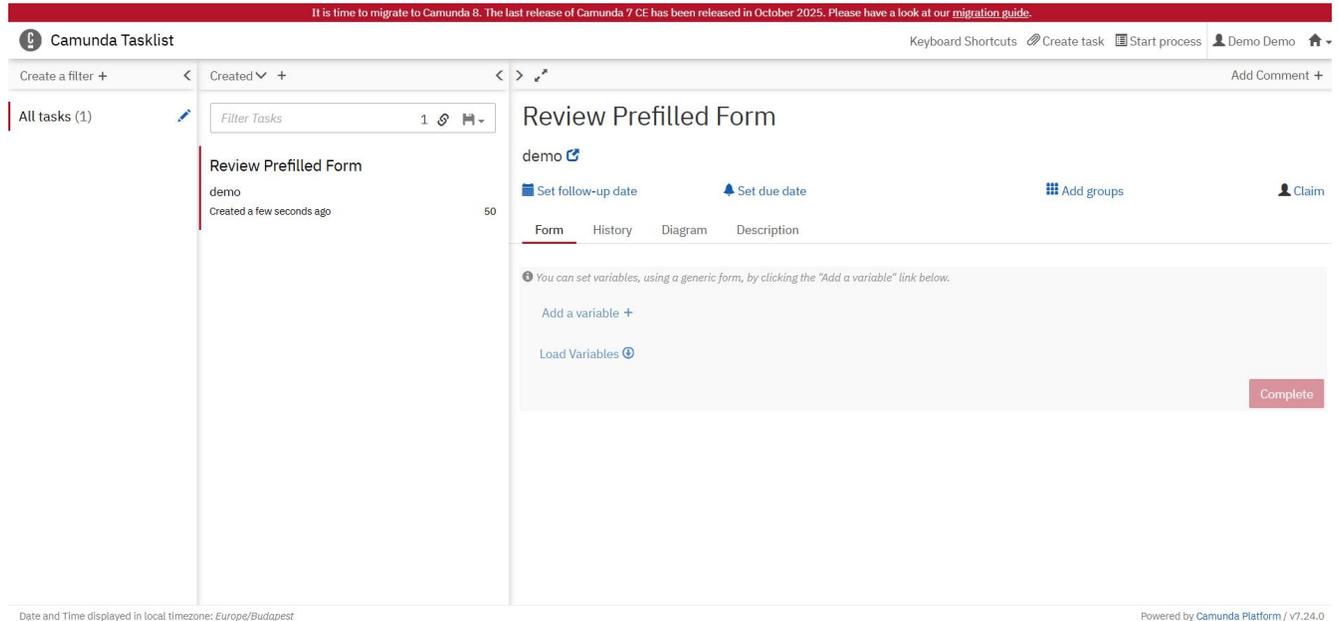


Figure 9: Claiming a task in the tasklist

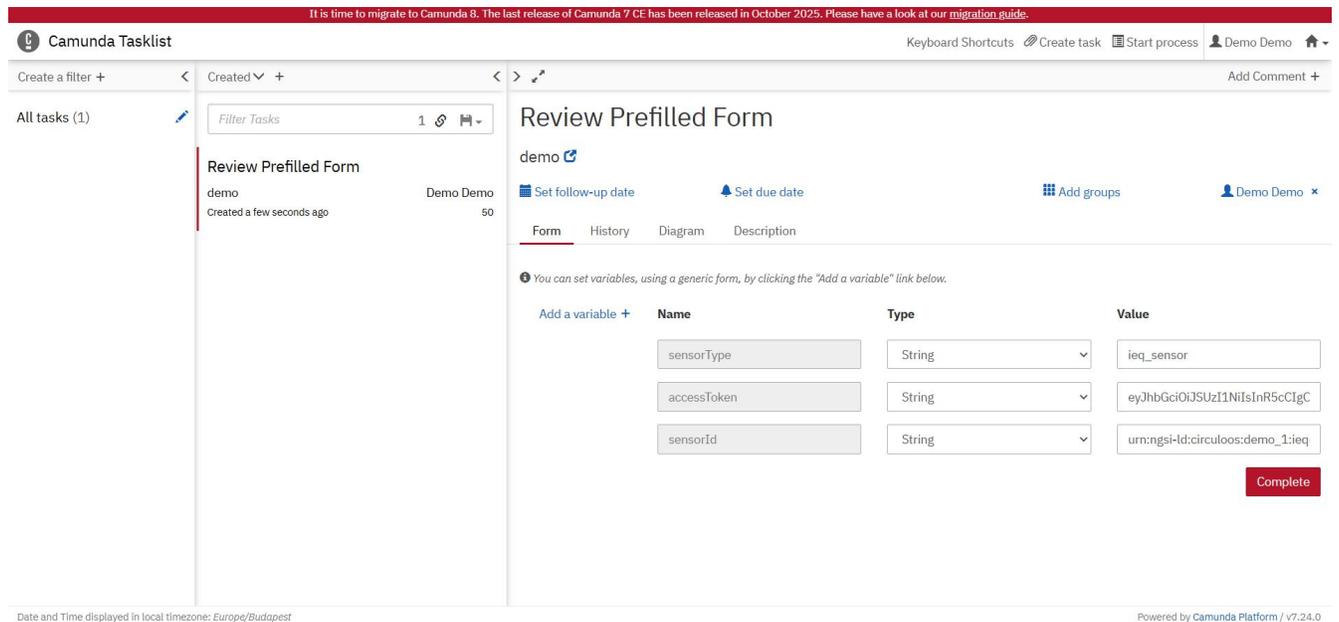


Figure 10: Completing a human task

Figure 11 shows the email notification to a user to complete a task.

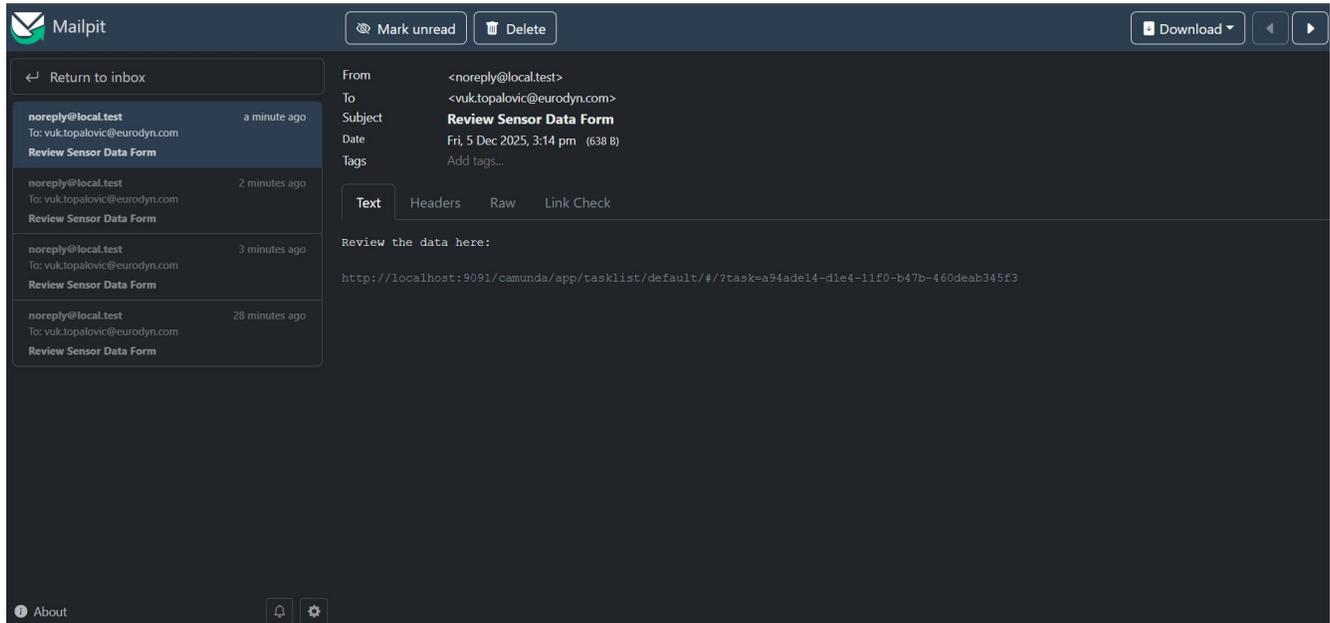


Figure 11: Email notification to complete a task

3.1.1 Requirements

- Docker & Docker Compose
- Java 17+
- Maven or Gradle

3.1.2 Running

Clone the demo's repository

```
git clone https://github.com/vukeurodyn/camunda-integration-app.git
cd camunda-integration-app
```

Start all services using Docker Compose

```
docker compose up --build -d
```

3.1.3 Accessing services

Once the services are running, they can be accessed as follows:

Table 2: URLs and credential for accessing services of the demo

Service	URL	Credentials
Cockpit	http://localhost:9091/camunda	demo / demo
Tasklist	http://localhost:9091/camunda/app/tasklist	demo / demo
Mailpit web UI	http://localhost:8025	-

3.1.4 Configuration

The application expects the following environment variables (or application.yml configuration):

```
# Camunda
camunda.bpm.client.base-url: http://localhost:9091/engine-rest

# Mail (Mailpit SMTP)
spring.mail.host: localhost
spring.mail.port: 1025
spring.mail.username:
spring.mail.password:
```

3.1.5 Deploying the BPMN Process

To deploy the BPMN process:

1. Open Camunda Modeler
2. Load src/main/resources/bpmn/demo.bpmn
3. Click Deploy

3.1.6 Starting a process instance

Go to Camunda Tasklist at <http://localhost:9091/camunda/app/tasklist> and start an instance named "demo".

3.1.7 Viewing emails

All emails sent during the workflow can be viewed at: <http://localhost:8025>

3.1.8 Stopping services

To stop all services:

```
docker compose down
```

To remove all data:

```
docker compose down -v
```

3.1.9 Directory structure

```
├─ src/
│   └─ main/
│       └─ java/
│           └─ com/camunda_integration/
│               ├── DemoApplication.java
│               ├── CamundaRestClient.java
│               ├── ProcessStarterService.java
│               └─ api/
│                   ├── ProcessController.java
│                   └─ StartRequest.record
│                       └─ vars/
│                           ├── CamundaVars.java
│                           └─ workers/
│                               └─ ReviewEmailWorker.java
│
├─ resources/
│   ├── application.yml
│   ├── bpmn/
│   │   ├── demo.bpmn
│   │   └─ review.form
│   └─ docs/
│       └─ architecture_camunda.md
├─ docker-compose.yml
├─ Dockerfile
├─ pom.xml
└─ README.md
```

3.1.10 Troubleshooting

Camunda not accessible

- Verify container is running:

```
docker ps | grep camunda
```

- Check logs:

```
docker logs camunda
```

Keycloak authentication fails

- Ensure realm and client are configured correctly
- Verify client secret matches configuration

Emails not sent

- Check Mailpit is running:

```
docker ps | grep mailpit
```

- Verify SMTP configuration points to port 1025

Worker not fetching tasks

- Check worker is connected: Look for external task subscription logs
- Verify Camunda REST API is accessible from worker

3.2 CIRCULOOS pilots

3.2.1 Description of the Process Orchestration in the pilots

Figure 12 shows the representation of the TO-BE scenario through the BPMN notation concerning the wood pilot, highlighting the main processes and information flows. Process tasks colored yellow indicate activity on the RAMP.eu web interface, tasks colored blue indicate data exchange with the CIRCULOOS Data Platform and tasks colored green indicate 'shipping' tasks where the environmental impact is variable and depends every time on the specific supply chain that is formed.

D3.2 Supply Chain Orchestrator M24

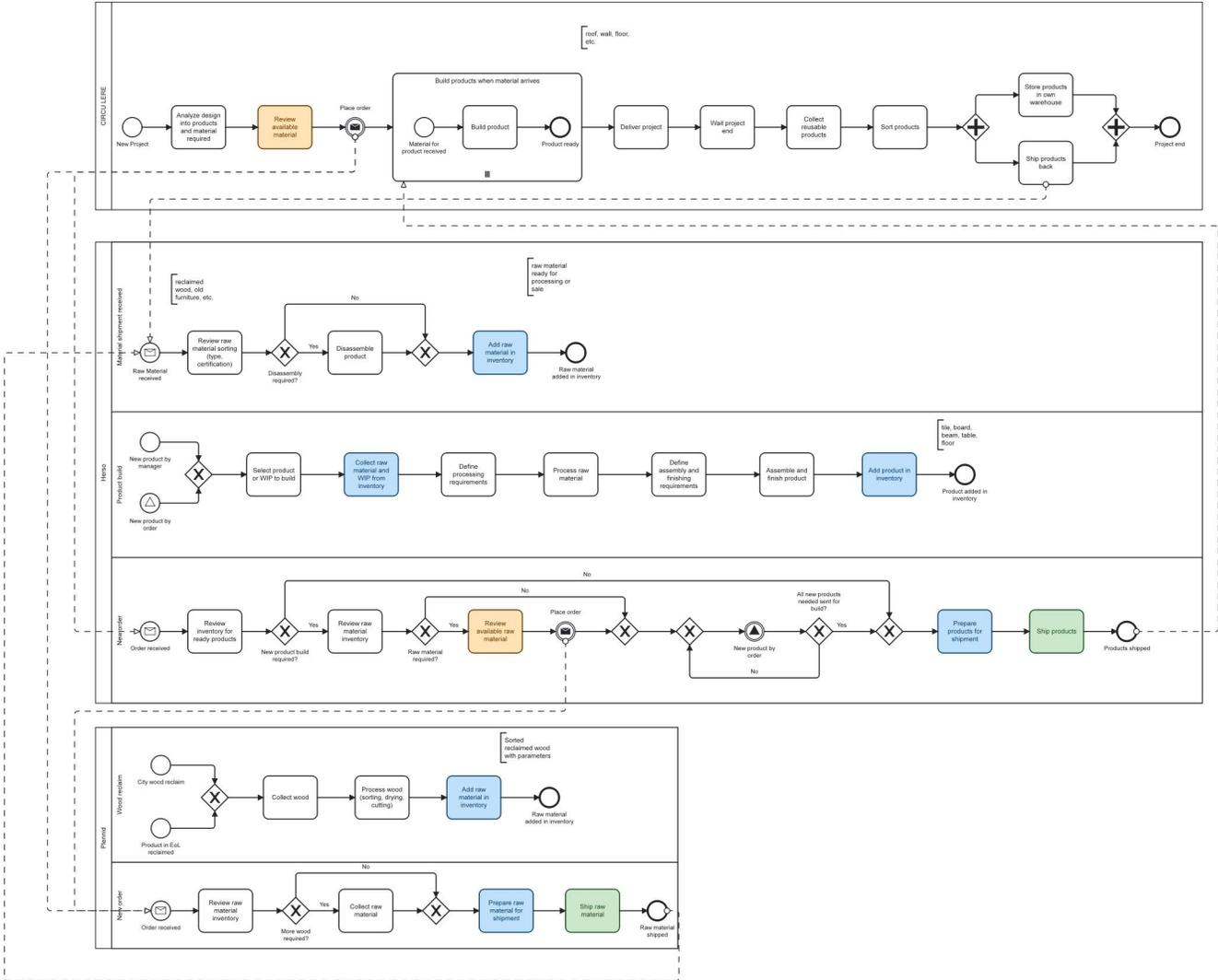


Figure 12: BPMN model for the Wood Pilot

The Process Orchestrator manages the process of each actor (indicated as a ‘swim lane’ in the model), taking decisions were appropriate, gathering user input (either for indicate the fulfillment of a task, or to gather data) and communicating with external systems as needed. More detail on how the process is executed within the Process Orchestrator is given in deliverable D3.2. Each actor’s internal processes are explained below, and are connected to the other actors as indicated in the model:

- CIRCULERE: CIRCULERE’s main process starts when a new project is launched. CIRCULERE analyses the requirements of the project and the material that is needed and proceeds to review the available material for sale on RAMP.eu. Once the provider is selected, an order is placed to buy the material (in the pilot case, this is Herso). Then the material is received by the provider and the products needed for the project are built. The project is delivered and when the project ends, CIRCULERE recollects and sorts reusable products. These products are either stored in their own warehouse or are sent back to the provider, and the overall CIRCULERE process is finished.
- HERSO: Herso has 3 main internal parallel processes: 1) Collecting and sorting raw material, and making it ready for producing products or selling it (hence, this process ends with adding the

material to RAMP.eu), 2) Producing a product, that can be launched either by the manager or because of an order, which ends in adding the product to the inventory (either for sale or for a specific order), and 3) Executing an order, explained in more detail: When an order is received, inventory is checked for available products in the inventory. If available products are sufficient for the order, the order is prepared and shipped. If products need to be produced, the available raw material is checked. If available raw material is sufficient, the production of products is triggered, and once the products are ready, they are shipped. If raw material is not sufficient, available material for sale is checked on RAMP.eu and an order is placed to the selected provider (in the pilot case, this is Plennid). Once raw material is received, the production of the order products and shipping follows, as above.

- PLENNID: Plennid has 2 main internal parallel processes: 1) Collecting and sorting raw material, for city wood in EoI, processing it and making it available for sale, and 2) Executing an order, by preparing the available material for shipping, or collecting more if required.

Figure 13 shows the representation of the TO-BE scenario through the BPMN notation concerning the plastic pilot, highlighting the main processes and flows. Process tasks colored yellow indicate activity on the RAMP. In the EU web interface, tasks coloured blue indicate data exchange with the CIRCULOOS Data Platform, while tasks coloured green indicate 'shipping' tasks where the environmental impact varies depending on the specific supply chain formed.

One main process is included, triggered by a new order. The first decision is, based on the order, whether virgin raw material is required or if material from other sources, like recycled plastic, can be used. When material from other sources may be used, the available material (own inventory and data on the Data Platform) is reviewed. If material in the own repository is sufficient, it is fetched and forwarded to the production line material reception; else an order is placed to the selected supplier. The raw material is checked (receipt and control), where the material is once again inspected. If material is rejected, it is labelled and according to its quality, it is either returned to the supplier or stored within its own inventory for another order. If it is accepted, it is forwarded for production. Along the production and up to packaging, there are quality controls. When a part (in progress or final) is considered

D3.2 Supply Chain Orchestrator M24

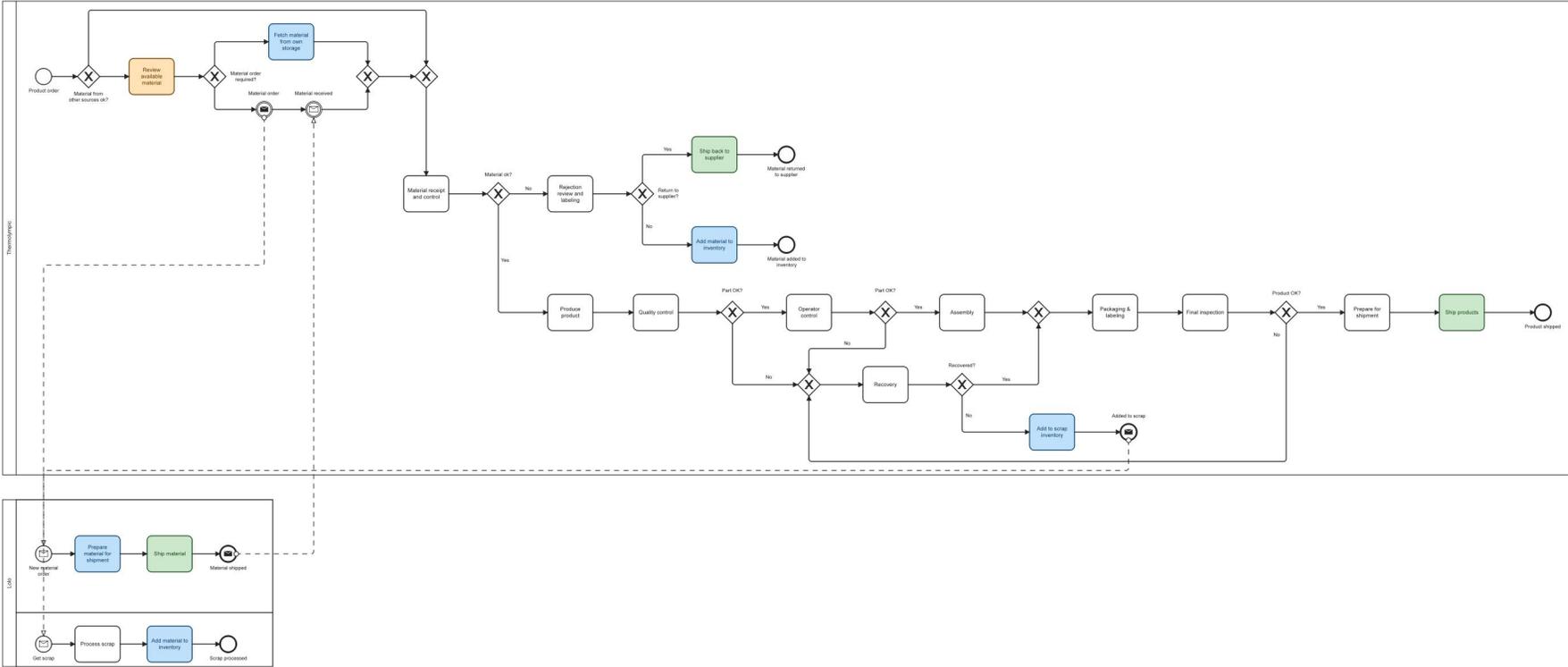


Figure 13 BPMN model for the Plastic Pilot

Figure 14 shows the representation of the TO-BE scenario through the BPMN notation concerning the leather pilot, highlighting the main processes and information flows. Process tasks colored yellow indicate activity on the RAMP.eu web interface, tasks colored blue indicate data exchange with the CIRCULOOS Data Platform and tasks colored green indicate ‘shipping’ tasks where the environmental impact is variable and depends every time on the specific supply chain that is formed.

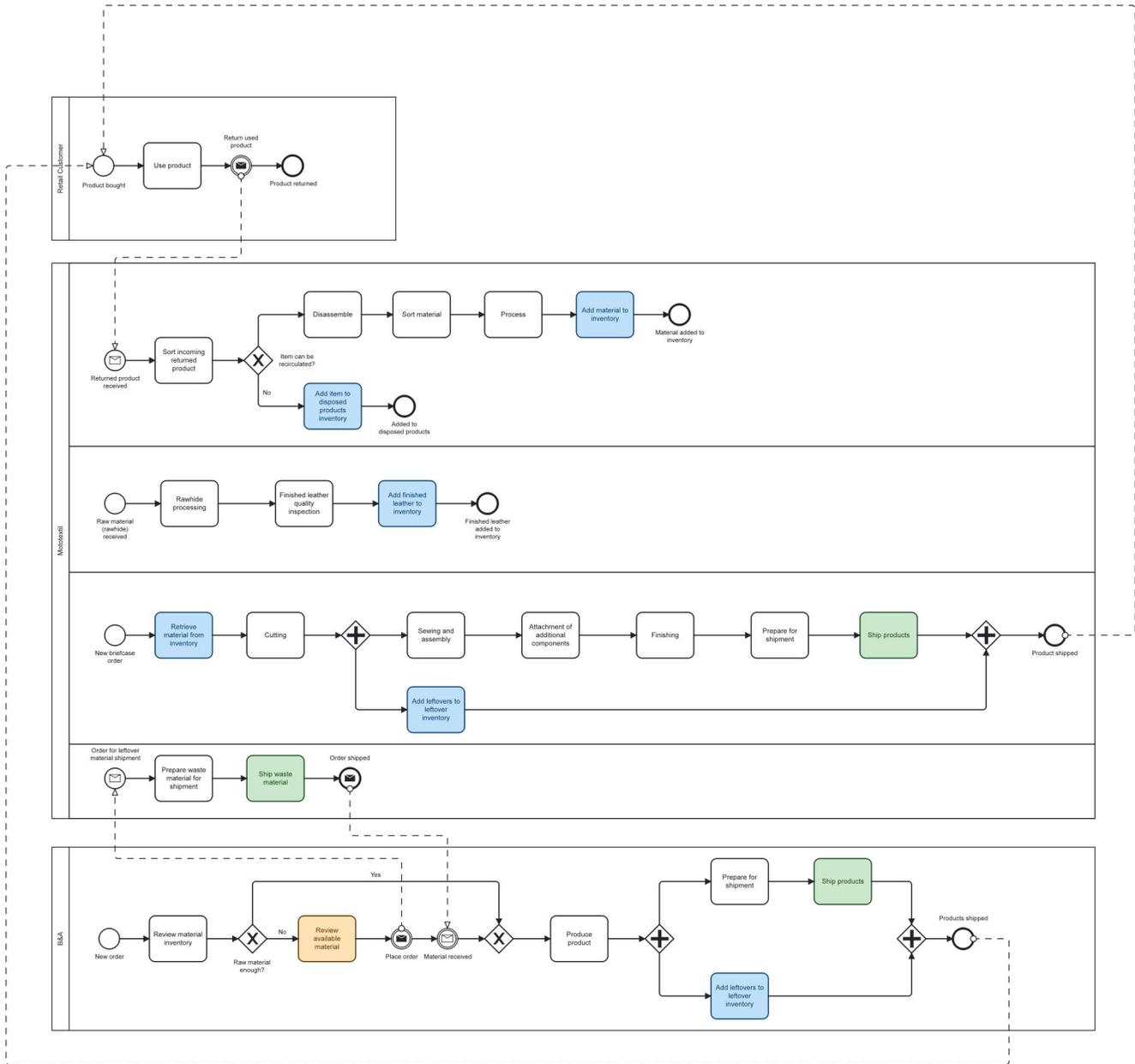


Figure 14: BPMN model for the Leather Pilot

The overall process includes 3 actors:

- RETAIL CUSTOMER: The retail customer opens and closes the circular overall process. It buys the product and, after using it, returns it to Mototextil.
- MOTOTEXTIL: Mototextil has 4 parallel processes: 1) One is for the reception of used bags, where they are sorted, processed, and material is either added to the inventory for reuse, when possible, or they are put in the disposed products inventory. 2) Another process is for receiving and processing rawhide and then adding it to the inventory. 3) The main production process includes the receipt of material from the inventory and the production of the bag. During this process, the leftover material (leather) is produced and added to the leftovers inventory. 4) One last process is included, related to the preparation and shipping of the leftovers.
- B&A: B&A has one main production process. It starts by reviewing the available material when an order is received. When it is not enough to fulfil the order, the leftovers of Mototextil are ordered and used for production. Also, during this production, further leftover leather is produced and added to the inventory. That means that even when using the leftovers of Mototextil, during the B&A production, leftover of these leftovers is generated and made available for other manufacturers.

3.2.2 Making the model executable

The pilots are available in the same github repository that was indicated for the demo (<https://github.com/vukeurodyn/camunda-integration-app.git>).

In this paragraph, the executable of the plastics pilot is further explained. As indicated in the guide, the model needs to be further refined. This leads to a modified, more technical model for the plastics pilot, depicted in Figure 15.

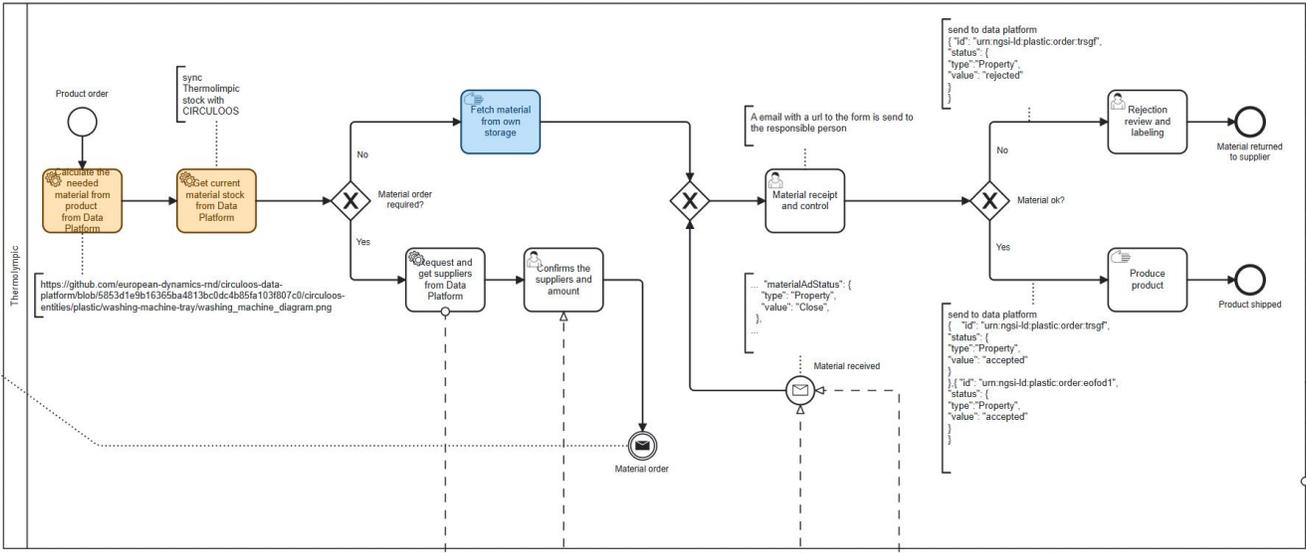


Figure 15: Executable model for the plastics pilot

As previously explained, service tasks are linked to code and scripts that are developed for the specific task. For example, Figure 16 represents a service task that mimics a post request to the camunda-

D3.2 Supply Chain Orchestrator M24

integration-app which talks to the data platform and where from it takes information about machine, components that one consists of and materials as well as their stock level which is calculated and got back with calculation id. Figure 17 and Figure 18 represents a service task that gets the data from camunda-integration-app and

result is stock adequate to requested units of material which is needed for required amount of units of machine that need to be produced. Later it comes with "sufficient" field which is checked on the next gateway to determine way of the flow. Figure 19 and Figure 20 depict forms where user input and decision-making is required along the process.

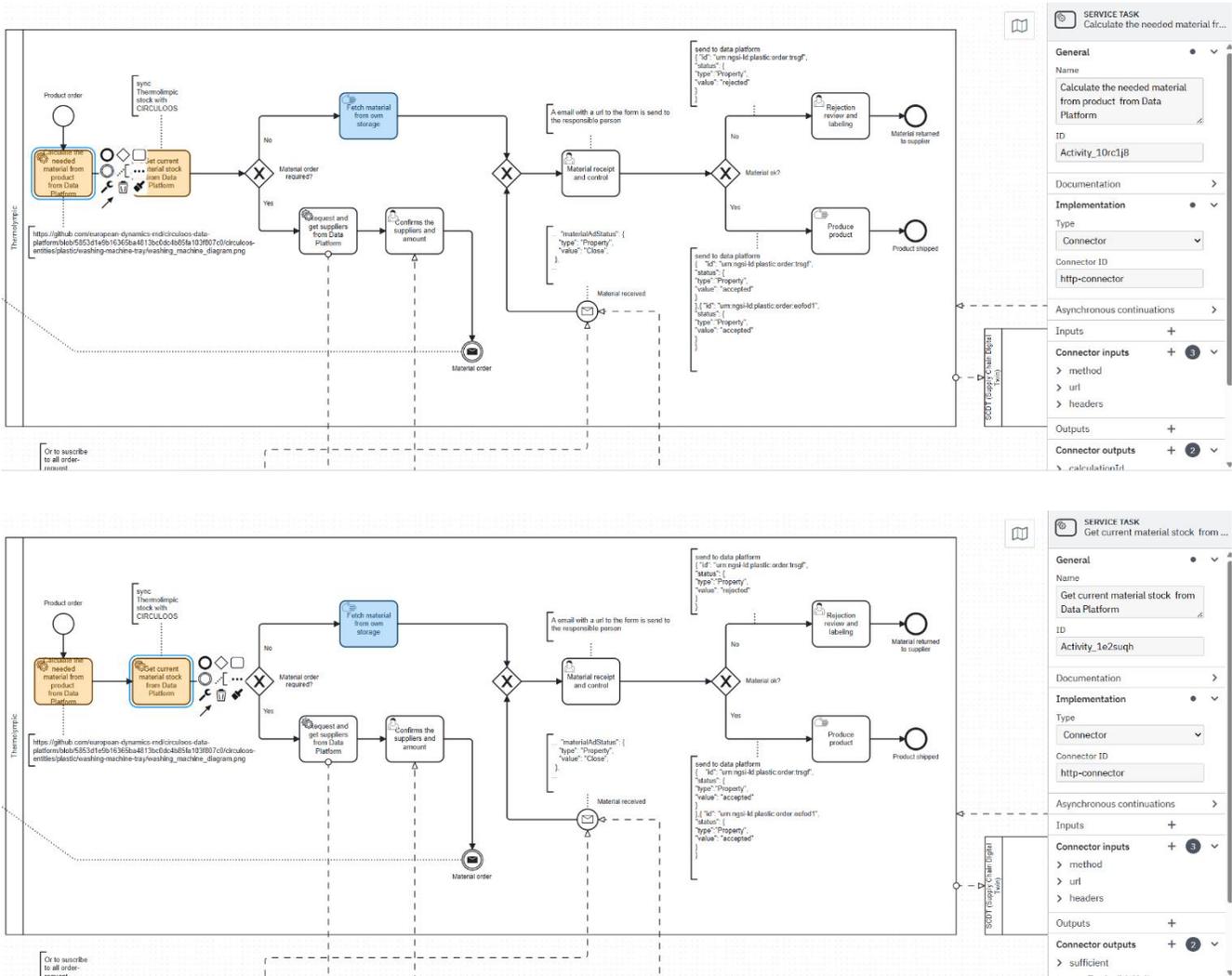


Figure 16: Connecting a task to script (a) above, (b) below

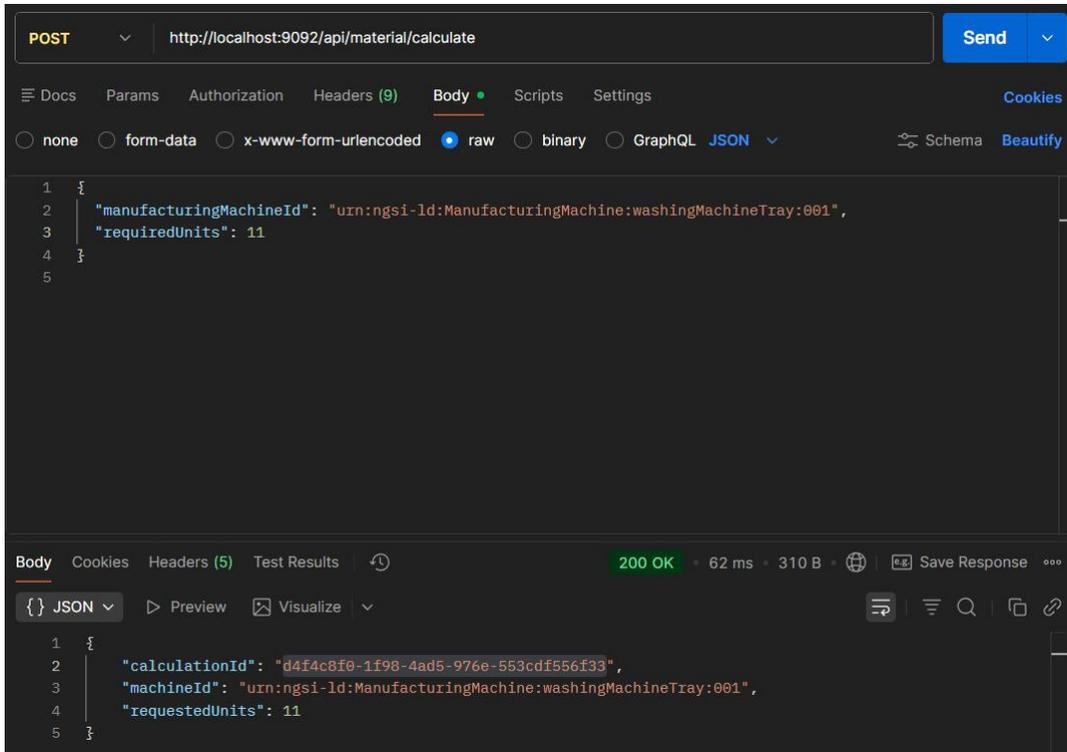


Figure 17: JSON post from the Process Orchestrator to the CIRCULOOS Data Platform

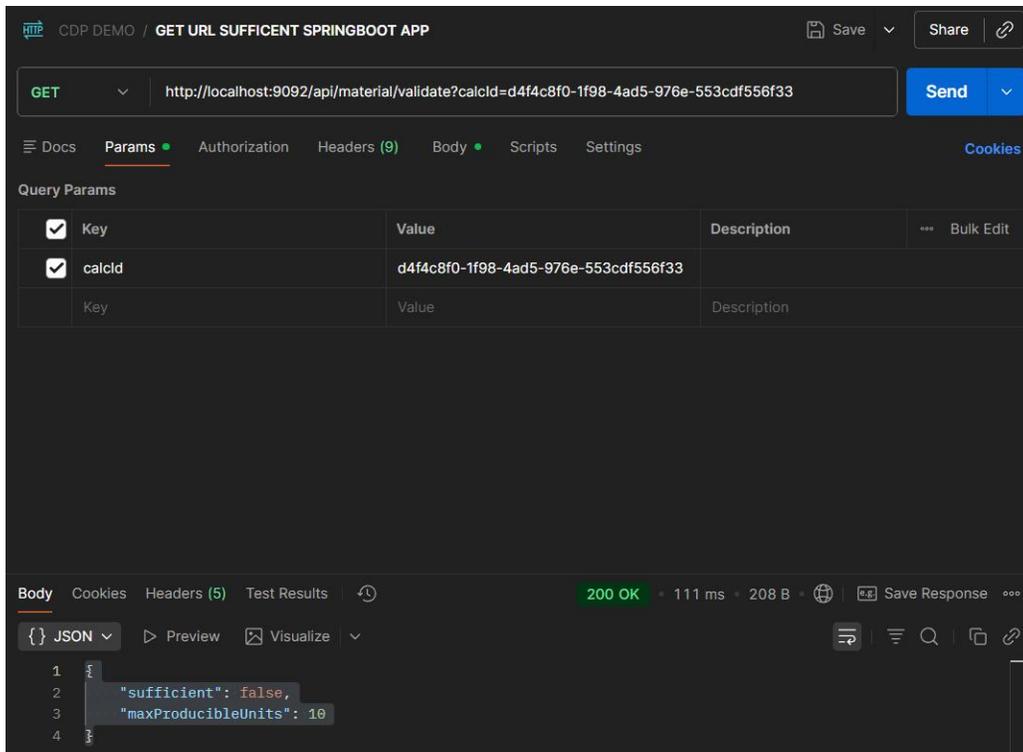


Figure 18: JSON response that is fed to the Process Orchestrator by the CIRCULOOS Data Platform

The screenshot shows a web interface for a form titled "FORM material_receipt_and_control". The "Form Preview" section on the left contains three input fields: "Material Id" (a text box), "Sufficient" (a dropdown menu with "Select" as the current value), and "Material qty" (a text box with a vertical scrollbar). Below this, the "Form Output" section displays a JSON object:

```
1 {  
2   "material_id": "",  
3   "sufficient": null,  
4   "material_qty": null  
5 }
```

 The right-hand side of the interface shows a sidebar with a "General" section containing fields for "ID" (with the value "material_receipt_and_control") and "Version tag" (which is empty).

Figure 19: Form for receiving material

The screenshot shows a web interface for a form titled "FORM rejection_and_review_labeling". The "Form Preview" section on the left contains two input fields: "Label review*" (a dropdown menu with "Select" as the current value) and "Comments on rejection" (a text box). Below this, the "Form Output" section displays a JSON object:

```
1 {  
2   "label_review": null,  
3   "comments_on_rejection": ""  
4 }
```

 The right-hand side of the interface shows a sidebar with a "General" section containing fields for "ID" (with the value "rejection_and_review_labeling") and "Version tag" (which is empty).

Figure 20: Input form for rejecting a part

4 Conclusion

The Process Orchestrator is an important component of the CIRCULOOS architecture, enabling the transition from traditional, siloed manufacturing processes to integrated, collaborative and circular supply chains. By leveraging BPMN-based modeling and a robust execution engine, it provides a unified environment where automated tasks, human interactions and system integrations coexist seamlessly. This orchestration capability ensures that complex workflows across multiple manufacturing SMEs can be executed with precision, transparency and adaptability.

Its value extends beyond process automation: the orchestrator acts as a strategic enabler of circular manufacturing. It coordinates material reuse, recycling and logistics while ensuring that sustainability data is captured and shared across the value chain. Through its integration with the CIRCULOOS Data Platform it supports real-time decision-making with a focus on environmental performance indicators. This empowers manufacturing SMEs to adopt circular economy principles without disrupting their operations, fostering resource efficiency, reducing waste and enhancing competitiveness.

Ultimately, the Process Orchestrator is not just a technical component but a business transformation tool. It equips manufacturing SMEs with the digital capabilities needed to thrive in a circular economy, enabling them to collaborate effectively, comply with sustainability targets and unlock new value streams. Its scalability and interoperability make it a future-proof solution for orchestrating complex supply chains in an increasingly dynamic and sustainability-driven industrial landscape.

The next steps include:

- Identify in which EXDs the Process Orchestrator can be tested
- Collect feedback to identify weaknesses and space for improvement
- Consolidate final documentation
- Prepare final packaging and repository

CIRCULOods



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101092295. The herewith information reflects only the author's view. The European Commission is not responsible for any use that may be made of the information herewith included.