



Co-Creating Circular
Resource Flows in Cities

constRuctive mEtabolic processes For material fLOWs in
urban and peri-urban environments across Europe

Deliverable 2.5

Reflow OS Graphical Interfaces

Due date of deliverable: 31/05/2021
Actual submission date: 14/09/2021
Start date of project: 01/06/2019 Duration (36 Months)
Dissemination Level: Public ✓



This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement number 820937.

DELIVERABLE

Work Package	WP2 IT Infrastructure and Tools
Deliverable	D2.5 Graphical Interfaces for Mobile Clients and Dashboards
Task(s)	Task 2.5 Graphical Interface Development [M6-M24]
Document Name	Reflow OS Graphical Interfaces
Due Date	M24: 31.05.2021
Submission Date	M28: 14.09.2021
Dissemination Level	<input checked="" type="checkbox"/> P – Public <input type="checkbox"/> CO – Confidential
Deliverable Lead	IAAC



Author(s)	Guillem Camprodon, Viktor Smári, Óscar González (IAAC)
Point of Contact	Guillem Camprodon, guillem@iaac.net
Reviewers	Adam Burns (DYNE), Stefano Bocconi (WAAG)
Status	<input type="checkbox"/> Plan <input type="checkbox"/> Draft <input type="checkbox"/> Working <input checked="" type="checkbox"/> Final <input type="checkbox"/> Approved
Abstract (for public dissemination only)	<p>This document documents Reflow OS user interfaces. It aims to support Reflow pilot cities to build their custom software applications to incentivize circular practices in local ecosystems through monitoring and optimization of urban metabolic processes. It describes the effort undertaken within the context of Task 2.5 Graphical Interface Development dedicated to the development of ergonomic, intuitive and customizable graphical interfaces based on the requirements provided in other tasks.</p>
Keywords	Reflow OS, User Interface, User Stories, Software Development, Pilots customization

Statement of Originality	This deliverable contains original unpublished work except for references already made available previously via the WeLoop and ZenPub software repositories in Github. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.
---------------------------------	--

Revision history

Revision	Date	Author(s)	Organization	Description
D2.5 v0.1	03 May 2021	Guillem Camprodon,	IAAC	Preliminary structure
D2.5 v0.2	18 June 2021	Viktor Smári, Óscar González	IAAC	Integration of tech development
D2.5 v0.3	09 July 2021	Guillem Camprodon, Viktor Smári	IAAC	Requirements and WeLoop structure sections
D2.5 v0.4	26 July 2021	Guillem Camprodon	IAAC	Document consolidation and formatting
D2.5 v0.5	10 September 2021	Guillem Camprodon, Óscar González	IAAC	Integration of suggestions and remarks after internal review
D2.5 v0.5.2	23 September 2021	Guillem Camprodon, Óscar González	IAAC	Fixes source code repository end-point mistake in Development environment section

The information, documentation and figures in this deliverable are written by the REFLOW project consortium under EC grant agreement number 820937 and do not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.



Table of Contents

List of abbreviations and keywords	8
Introduction	11
Pilot Requirements	12
Material exchange use case	12
Backtracking of supply chains scenario	14
Functional Requirements.....	15
Architecture Summary	16
WeLoop	17
Main views	18
Secondary pages	27
Customization and reusability	28
Reflow UI Style guide	28
Reflow OS UI Library.....	29
Main modules	31
Development environment.....	33
Remote monitoring.....	34
Issues reporting and contributions	34
Production deployment.....	35
Licensing	35
Conclusions.....	36



Table of Figures

Figure 1: Resource Event Agent model from the Value Flows Vocabulary used to record different types of economic activities over time. (source https://valueflo.ws/introduction/flows.html).....	13
Figure 2: Reflow OS Components simplified for WeLoop users	16
Figure 3: WeLoop - Log In page	17
Figure 4: WeLoop - Home page view	18
Figure 5: WeLoop - Community page view.....	19
Figure 6: WeLoop - Inventory page view	19
Figure 7: WeLoop - Material Passport view	20
Figure 8: We Loop - Material Passport Full View	21
Figure 9: We Loop - New Resource View	22
Figure 10: We Loop - Location Input Form	23
Figure 11: WeLoop - New Event view.....	23
Figure 12: WeLoop - New Intent view.....	24
Figure 13: WeLoop - Community Intent view.....	24
Figure 14: WeLoop - Community Intent create Resource throughout Event	25
Figure 15: WeLoop - User Page view	25
Figure 16: WeLoop - Collection's view	26
Figure 17: WeLoop - Discover view	26
Figure 18: Reflow OS Style Guide sample.....	28
Figure 19: Reflow OS UI Library - website.....	29
Figure 20: Reflow OS UI Library - multiple views.....	30



Index of Tables

Table 1: Functional requirements implemented following D2.1 Use Case Analysis and Requirement	15
Table 2: WeLoop - Secondary pages.....	27
Table 3: WeLoop - Main modules	31



List of abbreviations and keywords

ActivityPub	ActivityPub is a standard for the Internet in the Social Web Networking Group of the World Wide Web Consortium (W3C).
API	Application programming interface
Back-end	Data access layer of a piece of software, mostly a web application in contrast to the presentation layer or front-end.
Business process	A business process is a collection of structured activities that achieve a specific goal.
Circular Economy (CE)	A Circular Economy is an alternative to a dominant linear industrial model of design, produce, purchase, consume and dispose. A circular model aims to redefine growth and a positive societal impact. For its development a systemic approach and a deep transformation of habits and behaviour are needed. It entails a transition from using finite energy resources, to using renewable ones (designing the concept of-waste out of the system), while building economic, natural and social impact. Although starting from different materials in REFLOW, the focus of the circular economy gradually extends beyond these issues related to material management and covers other aspects such as the social impact, technological aspects and the evolution of urban governance structures. Since Circular Economy is purposive, but isn't just about sustainability environmentally and economically, the social components are fundamental for this transition to happen. This happens in mindset and cultural rituals: from the way we design, produce, consume, purchase and dispose, all the way to how we see and value existing, or even abundant, resources present in the local environments.
Co-creation	Co-creation is a product or service design process that integrates users.
Core requirement	Core requirements are requirements describing the behaviour of the economic foundation. These are the requirements for functionalities that ReflowOS should offer and will be available for each pilot.
Data dashboard	A data dashboard is an information management tool that tracks, analyses and displays key data points.



Ecosystem	A (business) ecosystem is a network of organizations involved in the delivery of a product or service through competition and cooperation.
Economic Event	An observed economic flow in the Value Flows vocabulary. This could reflect a change in the quantity of an economic resource. It is also defined by its behaviour in relation to the economic resource.
Economic Resource	A resource that is useful to people or the ecosystem in the Value Flows vocabulary.
Front-end	Front-end is the client-side or web design in the web industry
Functional requirements	Functional requirements define the required behaviour of the system to be built. They define the capabilities that a product must provide to its users. Functional requirements are based on system objectives and respond to the critical task of ensuring the right implementation of the expected functionality in the final software.
GUI	Graphical user interface
GraphQL	An open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data.
Intents	In Value Flows vocabulary, intents describe potential future events which have not been agreed to by other agents, such as offers and requests. Intents are often used for discovering another agent to participate in the desired event.
Metadata	Metadata is structured data that contains information on characteristics of other data.
Minimum viable product (MVP)	A minimum viable product is a version of a product with just enough features to attract early users and validate a product idea early in the product development cycle.
Open data	Open data is the idea of data that can be used and redistributed by everyone for every purpose. Restrictions are only made to clarify the source or to secure the openness of the data.
Persona	A persona is a fictional character presenting a user type that might use the

			<p>product. It is created based on research and helps to understand the users' needs, experiences, behaviours and goals</p>
Pilot requirement			<p>Pilot requirements are pilot specific requirements that go beyond the economic foundation.</p>
Requirement's elicitation			<p>In requirements engineering, requirements elicitation is the process of collecting information to research and discover the requirements of a system.</p>
Reflow Open Data Dashboard			<p>A component of the Reflow architecture to provide access to the data generated, integrated and used in the context of the project.</p>
UI			<p>User interface</p>
Urban metabolism			<p>Urban metabolism can be understood as the collection of complex socio-technical and socio-ecological processes by which flows of materials, energy, people, and information shape the city, service the needs of its populace, and impact the surrounding hinterland.</p>
Use case			<p>A use case gathers possible scenarios of interactions between an actor and a system to achieve a business goal.</p>
User requirement			<p>User requirements describe what needs a user has when using a system, e.g., which activities the user must be able to perform.</p>
User stories			<p>A user story is a tool often used in Agile software development, describing a feature of the software from the end user's perspective. As user stories clearly state the different types of users, what they desire and why they can be seen as a very high-level definition of requirements.</p>
Value Flows			<p>A set of common vocabularies to describe flows of economic resources of all kinds within distributed economic ecosystems. The core is based on the REA (Resource, Event, Agent) ontology. The purpose is to enable internetworking among many different software projects for resource planning and accounting within fractal networks of people and groups.</p>

Introduction

“How do we imagine a system that allows us to plan economic activities in a distributed network, safely and without being dependent on any central authority? How could such a system stimulate the urban metabolism and enhance circular economies: minimising waste production, increasing the inclusiveness of the various actors present in the urban area and encouraging new practices, managing to make economically convenient what is naturally more sustainable?”¹

WeLoop is an example web application developed on top of Reflow OS. It is as well as a modular front-end framework to help Pilots develop their own Reflow OS interfaces. Reflow OS built on top of three main concepts: social networks, economic value flows and peer to peer systems.

Value Flow is an economic value flows vocabulary to facilitate groups experimenting with innovative collaborative models, small business ecosystem, commons-based peer production as well as any transitional economies. Utilizing a common language has the benefit to ensure interoperability across multiple federated instances of the Reflow OS and facilitate connectivity with other emergent economic software projects.

When multiple economic value flows are linked together it creates an economic network. That is a set of independent agents, which could be individual persons or organisations, who regularly collaborate to create and exchange goods and services. Examples of economic networks include supply chains, municipalities or future bioregions. Each economic network has its own rules for accessing and participating in economic activities.

As an application willing to map and facilitate relationships across agents, primarily persons, WeLoop inherits multiple functionalities from traditional social networks such as activity streams, a list of recent activities performed by an individual. Moreover, the use of that paradigm might contribute to reducing friction by providing users with a familiar interface already found in the world's most popular websites. However, traditional social media sites follow a centralized governance and ownership model that differs from the project goals. For that reason, the software takes the approach of a federated social network. That is a system that is decentralised, developed and operated by distinct providers but fully interoperable thanks to the use standards.

“Overall, it means experimenting with mixing social and economic activities together in a federated social network to unlock a whole new range of unexplored possibilities offers and needs listings with geolocation; creating, discussing and voting on proposals; coordinating labour and distributing outcomes among participants; time banking or mutual credit communities; collaborative budgets; as well as crowdfunding campaigns.”²

The following report unfolds how these foundational concepts are implemented into the Reflow OS ecosystem. It covers the end-user interface design and implementation of WeLoop. First, as an introduction for users willing to use it and second as a guide for Pilots to develop their custom applications on top of it. The deliverable is submitted later than planned. However, Reflow pilots have been experimenting with WeLoop for months. Their insights were fundamental to the report.

¹ Dev diary: our journey towards the reflowos, Ivan Minutillo <https://reflowproject.eu/blog/testing/>

² Introduction to the Bonfire project, a core component of Reflow OS <https://bonfirenetworks.org/>



Pilot Requirements

Reflow OS UI requirements built upon the work of Task 2.1 Use Case Analysis and Requirements to define use cases and requirements for the technical tools that support the pilot cities in Reflow. They are documented in *D2.1 Use Case Analysis and Requirements*.

A use case is an abstraction that describes all possible scenarios involving the described functionality. A scenario is an instance of a use case describing a concrete set of actions. Use cases are used to describe all possible cases; their focus is on completeness.³

Use cases for WeLoop were created in workshops between the technical partners and the pilots. In March 2020 a workshop⁴ was conducted to identify the user stories with Amsterdam, Paris, Vejle around the Material Exchange pilot case to further define a common scenario.

Material exchange use case

Mapping material exchanges is the foundational use case of Reflow OS. Its goal is to provide tools for pilots and participants to record different types of economic activities that are performed over time (see Figure 1). One such economic activity could represent the production of a new resource (e.g., a cake). Productions usually are preceded by a series of activities: transformations, transfers, exchanges, some work done, tools used - any step that contributed to producing the resulting goods. In the case of the cake, someone can record the ingredients involved and how they acquired them: harvesting trees (let's say it's an apple pie) or performing exchange in the local market, which person was involved in the production of the cake and how much time it took, the number of resources consumed, and the ones left. Such observations, when available openly, can stimulate discussions and actions to become more sustainable by incentivising circular practices in local ecosystems through monitoring and optimization of urban metabolic processes (see D2.2 Architecture and Manual for Distributed Network Setup and Maintenance pg. 8)

³ Object-Oriented Software Engineering Using UML, Patterns, and Java (Bruegge, Dutoit, 2010) Section 2.4.1

⁴ Read more about the workshop results in projects deliverable **D2.1 Use Case Analysis and Requirements** (pg. 16)

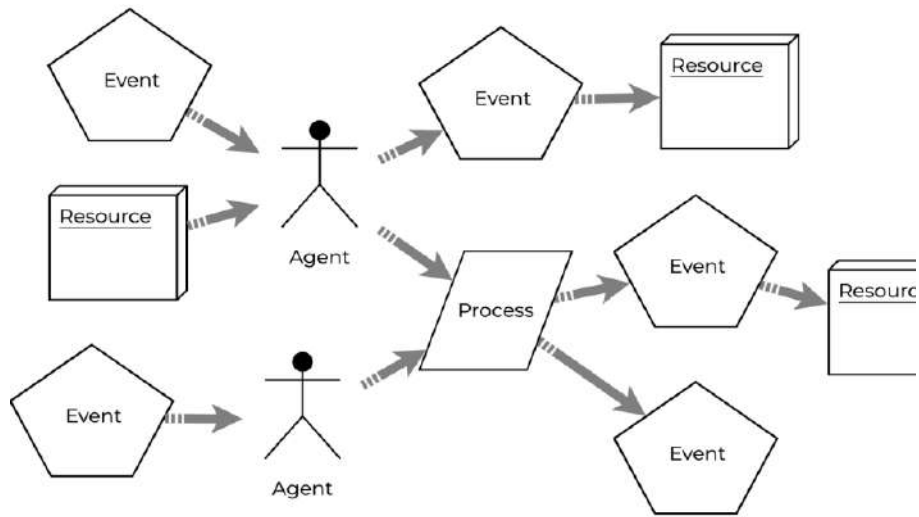


Figure 1: Resource Event Agent model from the Value Flows Vocabulary used to record

Backtracking of supply chains scenario

The scenario implements a particular material exchange use case and defines the core features of the WeLoop app. It supports two main features: shared inventories and material passports.

Pilot participants can register materials by filling out a form providing information such as name, description, images or location about the economic resource. Materials are listed as part of the user publicly shared inventory.

Pilot participants can log the transfers of material from one place to another or any kind of economic activities that have been performed using the Value Flows vocabulary. Here are some of the actions a user can perform on a material:

- Consume, decrements the quantity of an existing resource provided as an input. For example, an ingredient or component composed into the output, after the process the ingredient is gone
- Produce, increment the quantity of an existing resource and provide it as an output. For example, a new resource created in that process, or an existing stock resource added to
- Transfer gives full rights and responsibilities plus physical custody to another person or organization of the total or a part of a resource. For that reason, a second party is needed to have a receiver and a provider.

Based on each pilot policy, other stakeholders can trace back the journey of that material by viewing the material passport. Each material can be physically identified with a QR code to facilitate access to the online inventory and material passport.

Engaging participants to contribute additional metadata on resource consumption or material transformations can provide information to the pilot specific supply chain narratives. Transfers can occur within an organization or between different stakeholders, Reflow OS allows to record those interactions safely and securely benefiting from the Reflow OS federated architecture. In the future, such economic activities may be facilitated by digital agreements (see *D2.3 Implementation of Portable Cryptographic Functions*).

Functional Requirements

Almost the totality of the high priority core requirements defined in *D2.1 Use Case Analysis and Requirements* are implemented as shown in the table below (see Table 1):

Table 1: Functional requirements implemented following *D2.1 Use Case Analysis and Requirement*

ID	Description	References	Priority	Status
FR06	ReflowOS should be able to display the processing chain of a respective material or resource from the point the material entered the Reflow system	US: A01, US: A02, US:A05, US:A06, US:A08, US:A09, US:A12, US:A19, US:P03, US:P08, US:P13, UC:ROS13	High	Ready
FR07	ReflowOS allows user to track the consumption of a new resource	UC:ROS16	High	Ready
FR08	ReflowOS allows users to track the production of a new resource	UC:ROS16	High	Ready
FR09	ReflowOS allows users to track the transfer of a resource to another agent	UC:ROS15	High	Ready
FR10	ReflowOS allows users to track the usage of a resource	UC:ROS16	High	Ready
FR13	ReflowOS allows users to follow and unfollow other users	UC:ROS04, UC:ROS05	High	Ready
FR14	ReflowOS should allow the discovery of other users that are part of the network	UC:ROS03	High	Ready
FR15	ReflowOS allows users to edit their profile information	UC:ROS02	High	Ready
FR21	ReflowOS should implement end-to-end encryption		High	Pending
FR27	ReflowOS should allow the creation of different users/groups and access rights for authorized system user		High	Pending
FR28	ReflowOS allows users to create custom taxonomies	UC:ROS06	High	Pending
FR30	ReflowOS allows users to subscribe to taxonomies	UC:ROS10	High	Pending

Architecture Summary

Reflow OS is a collection of integrated software tools to support the Reflow pilots (see Figure 2). Its software architecture is a combination of multiple open-source components. Despite the report focusing only on one of the components, it's important to understand the architecture, primarily the core backend behind it. Reflow OS has a core backend that works as a central API for Pilots to build tools that programmatically interface Reflow OS such as WeLoop.

Zen Pub is the core backend component and builds on Commons Pub, a federated social network built on the Elixir programming language originally inspired by the Pleroma project. Zen Pub enhances the existing federated social network functionalities by providing a rich set of interfaces to map and facilitate economic value flows, based on the Value Flows standard. All the functionalities are accessible using a single GraphQL API. For a much in-depth reading see deliverable *D2.2 Architecture and Manual for Distributed Network, Setup and Maintenance* or the Reflow OS online documentation for the latest version⁵.

To ensure Reflow OS long term maintenance and support, Zen Pub is evolving into a much modular architecture called Bonfire. The new software continues to be built on Elixir and features the same GraphQL API for backward compatibility, but it follows a modular approach. For example, support for Value Flows or federation via the ActivityPub web standard can be maintained as two separate modules. Multiple modules can be combined to create new Bonfire apps to serve a specific purpose. Bonfire bundling inside Reflow OS means Pilots transition from Zen Pub will happen transparently in the future.

WeLoop is not the only application being built on top of the Zen Pub / Bonfire backend. Currently, integration with other components such as the Reflow Open Data Dashboard is being defined as part of WP2 work. A definition of the overall interfaces is available in *D2.4 Design and Development of REST API Application and Pub/Sub Interfaces*.

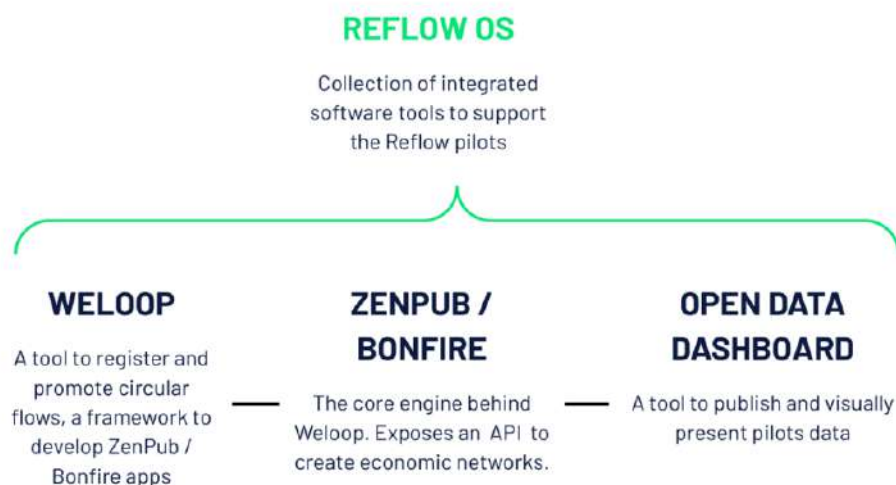


Figure 2: Reflow OS Components simplified for WeLoop users

⁵ Reflow OS online documentation <https://reflowos.dyne.org/>

WeLoop

WeLoop is the first client developed on top of Reflow OS as well as a modular framework to help others develop their own custom interfaces; it connects to the backend using the Reflow OS API⁶. WeLoop is fully open source and is available in Github⁷. A staging instance for Pilots to register and explore the latest updates is available at weloop.reflowproject.eu.

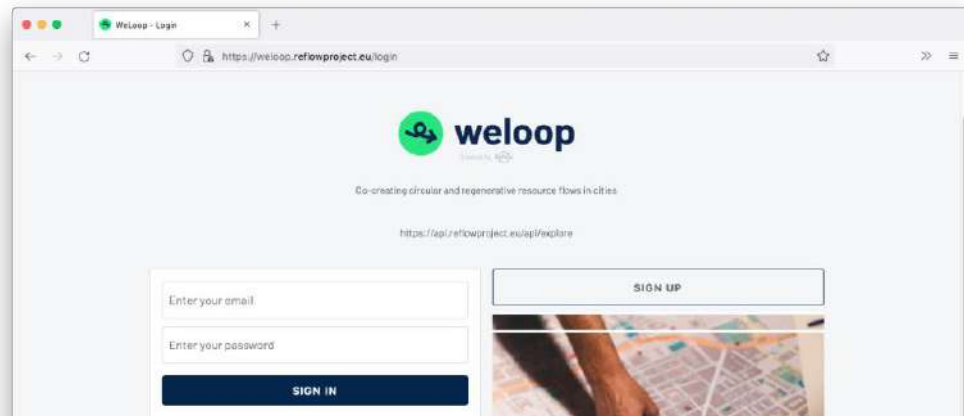


Figure 3: WeLoop - Log In page

WeLoop is a web-based (see Figure 3) client built using React JS⁸, a free and open-source front-end JavaScript library for building user interfaces or UI components. It uses Apollo Client⁹, a GraphQL library to perform queries and mutation over the Reflow OS backend remote instance using the GraphQL API. It uses Rebase and Styled Components¹⁰ to build reusable UI modules as well as Storybook, an open-source tool for building UI components and pages in isolation (see Reflow OS UI Library section). It also supports internationalization using i18n via Lingui¹¹.

⁶ Learn more about the API in <https://reflowos.dyne.org/> and <http://api.reflowproject.eu/api/explore>

⁷ WeLoop Repository (See also Licensing section further below) <https://github.com/reflow-project/weloop>

⁸ React JS online documentation <https://reactjs.org/>

⁹ Apollo Client online documentation <https://www.apollographql.com/docs/react/>

¹⁰ Rebase online documentation <https://rebassjs.org/> and Styled Components online documentation <https://styled-components.com/>

¹¹ Lingui, Javascript internationalization framework documentation <https://lingui.js.org/>

The original WeLoop codebase was forked from MoodleNet¹², Moodle's new social network to share and curate open educational resources with educators worldwide, which was originally forked from Pleroma federated social network¹³.

Main views

The following section introduces WeLoop core features by describing the graphical user interface main views or pages. That way it ensures features are presented in the user context rather than as abstract description of each implementation.

Home

Displays a timeline with past actions from users a user follows like other social media sites. (see Figure 4)

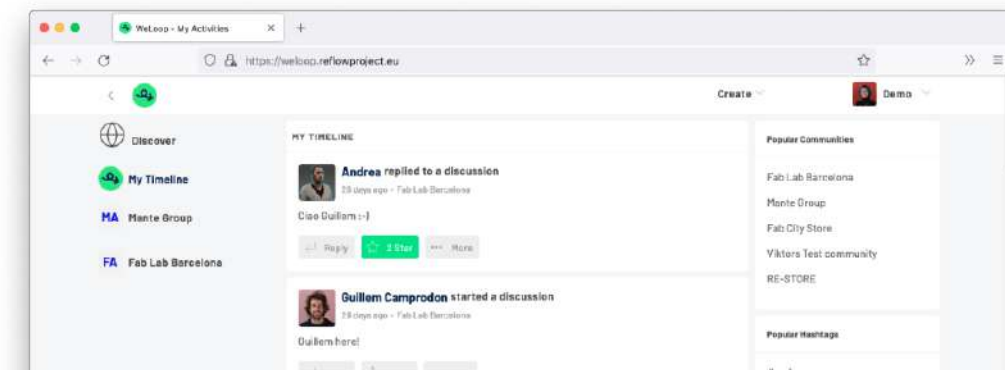


Figure 4: WeLoop - Home page view

Communities

Communities can be created by users to define a particular scope to enable users' conversations and exchanges around a particular topic (see Figure 5). Each community contains discussions, intents and members. Users can perform the following actions:

- Create new communities and join an existing one.
- Assign intents, a potential future event such as offers and requests to a community.
- Start and reply to discussions on community.
- Browse the full list of communities and join new ones.
- See the latest activity of a community.

¹² MoodleNet project online documentation <https://docs.moodle.org/dev/MoodleNet>

¹³ Pleroma project official homepage <https://pleroma.social/>

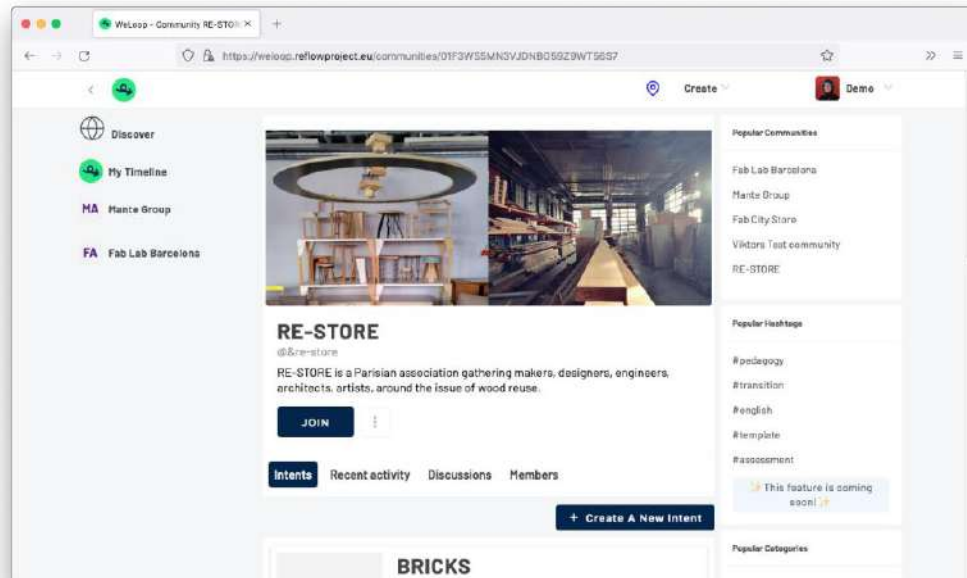


Figure 5: WeLoop - Community page view

Inventory

The inventory page lists all the available (economic) resources a user owns or derived from resources it owns. It is accessible from the user profile page and displays basic information about each resource such as name, description, quantity and image (see Figure 6). From it, we can access the full information of a resource also known as the Material Passport. As the number of resources increases, filters might be added to support finding and exploring resources.

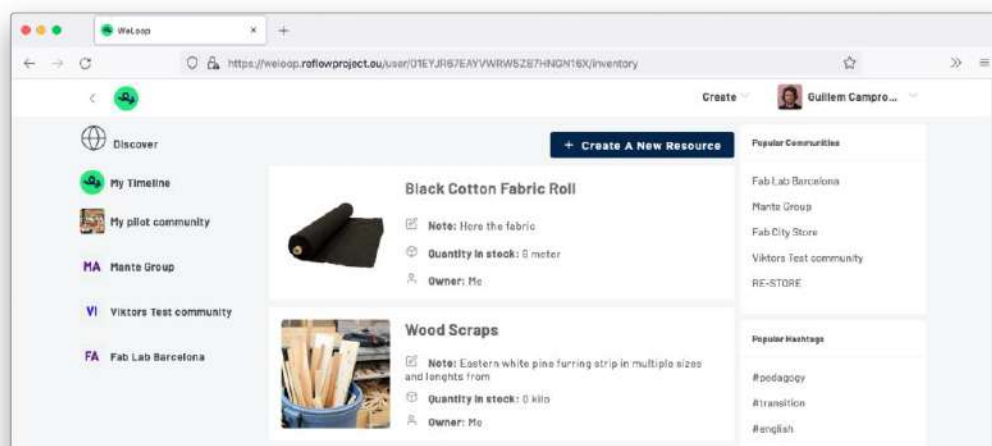


Figure 6: WeLoop - Inventory page view

Material Passport

The material passport page includes all the available information about a particular economic resource with a special emphasis on the resource history (see Figure 7). It is accessible from the inventory page as well as by scanning the provided QR code containing the economic resource unique URL. It includes the following information (see Figure 8):

- Main resource data such as name, description, quantity
- Resource history includes a list of all the economic events that resulted in that economic resource. It includes information such as provider, receiver, quantity and the action (transfer, consume, produce...). Following the links, we can navigate through the parent and child resources that originated a particular resource.
- Resource location displayed when available on a zoomable map.
- Resource image displayed when available.
- QR code to facilitate access to the online inventory URL. It can be printed on stickers to provide a convenient way to display materials' lifecycle.
- Owner actions: when a user is the owner or primary accountable for an economic resource it can also edit and delete the resource as well as perform an economic event (i.e., transfer) on top of it.

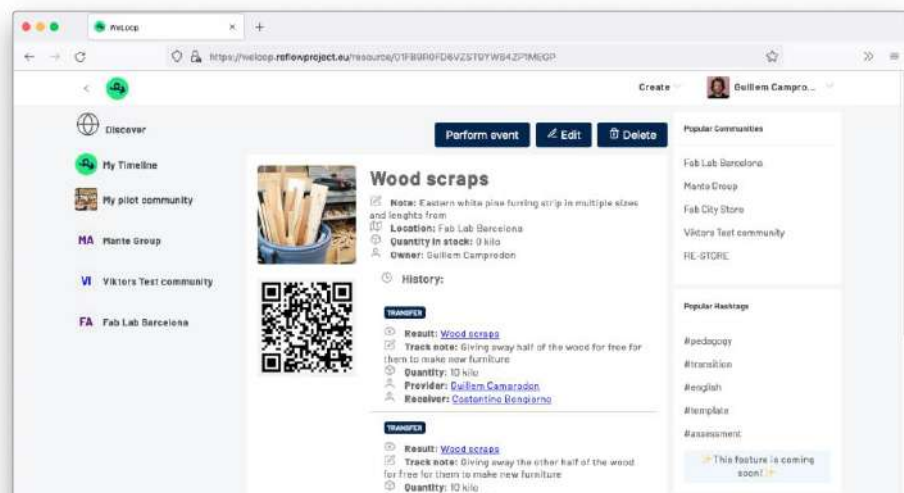


Figure 7: WeLoop – Material Passport view

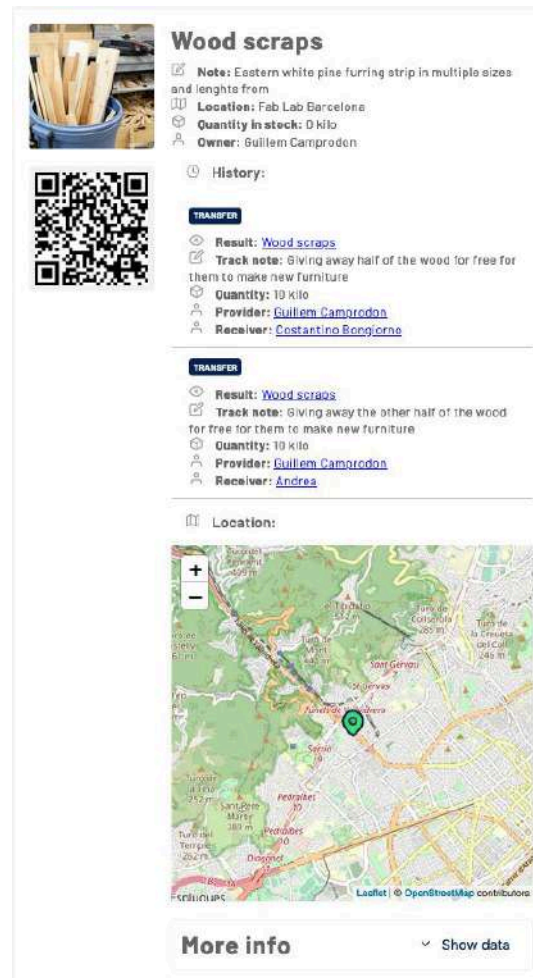


Figure 8: We Loop - Material Passport Full View

New Resources and New Event

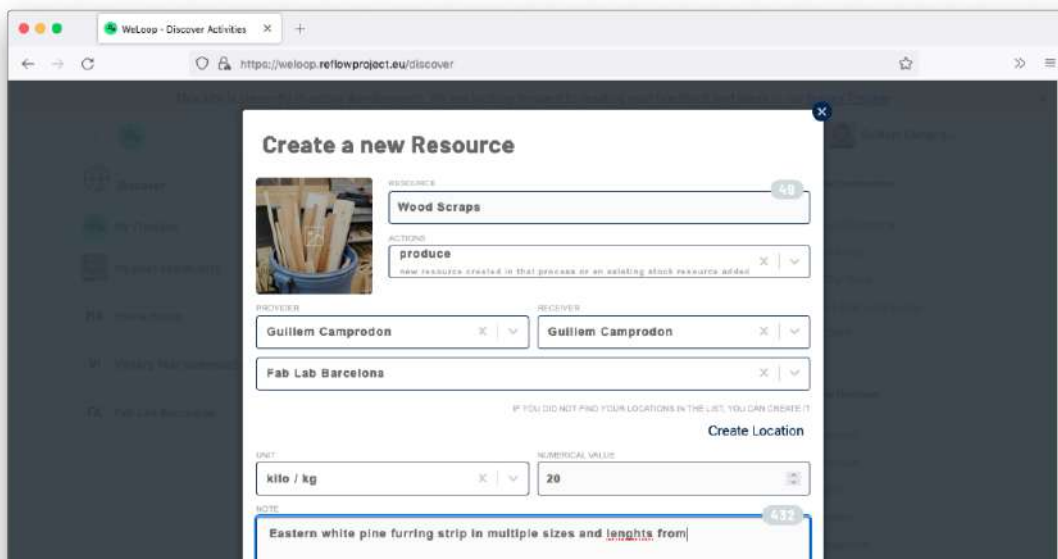
The New Resources and New Event page enable key functionalities to manage the users' Inventory and update the Material Passport(s) accordingly. The two pop-up windows "Create a new Resource" (see Figure 9) and "Create a new event on" (see Figure 11) can be accessed by clicking on the "Create a new Resource" and "Perform event" buttons, from the inventory page and from the resource page respectively.

Resources represent any economic resource we want to track within WeLoop. It is mostly designed to track physical materials., but is not limited to this type only

Following the Value Flows paradigm, a resource (economic resource) is always created by an event (economic event). That means to register any new resource we will always need to specify the action that creates it, even if the action comes from a resource or action that was not registered in the application previously.

Currently, the following information can be provided when creating a new resource:

- Name: Free text field to name the resource
- Action: Predefined list of actions to designate what the flow is doing and how it will affect an economic resource (or not) following the Value Flows convention. Current implementation mostly supports consume, transfer and produce (for an explanation of these actions see: Backtracking of supply chains scenario section in pg. 13)
- Provider and Receiver: Existing user involved in the action, defaults to the current user
- Location: Allows to select from existing locations or create a new one. New locations are created filling the location name or address (see Figure 10). Google Maps geo-search provides autocomplete and automatically translated the location to coordinates.
- Unit and quantity: Quantity available is measured using a predefined list (but not limited) of units (litre, kg, meter)
- Note: Free text field to annotate observations
- Image: Allows to upload an image of the resource



The screenshot shows a web browser window with the URL <https://weloop.reflowproject.eu/discover>. The main content is a form titled "Create a new Resource".

The form fields are as follows:

- RESOURCE:** Wood Scraps
- ACTION:** produce (with a sub-note: "new resource created in that process or an existing stock resource added")
- PROVIDER:** Guillem Campron
- RECEIVER:** Guillem Campron
- LOCATION:** Fab Lab Barcelons
- UNIT:** kilo / kg
- NUMERICAL VALUE:** 20
- NOTE:** Eastern white pine furring strip in multiple sizes and lengths from

There is also a small image of wood scraps in a blue bucket next to the resource name. A "Create Location" link is visible below the location field.

Figure 9: We Loop - New Resource View

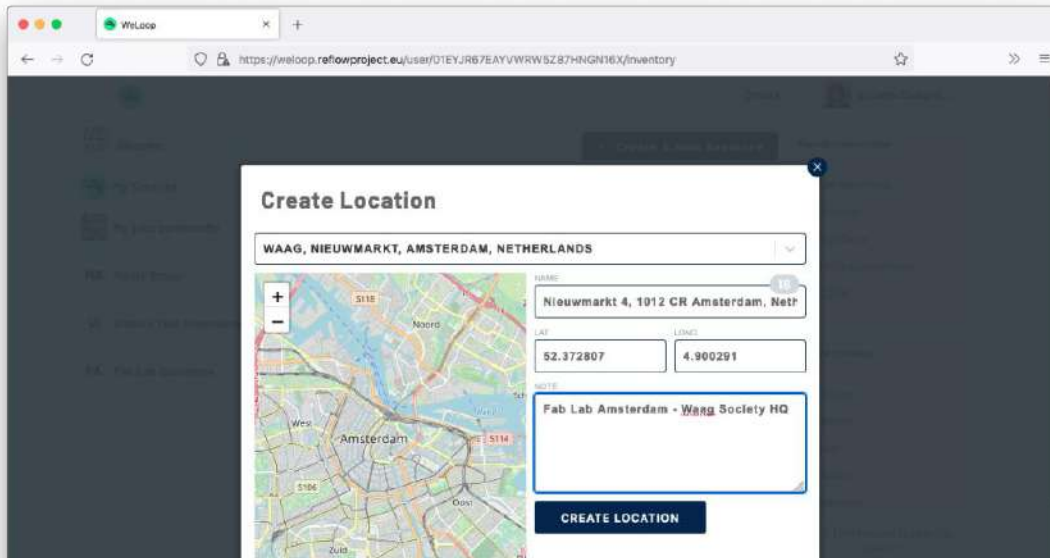


Figure 10: We Loop - Location Input Form

Once a resource is created, we can apply events on top to map changes in the quantity (i.e., decrease quantity), the rights (i.e., who owns it) or new resources that result from it (i.e. we consume one resource to make another one)

- Action: Predefined list of actions to designate what the flow is doing and how it will affect an economic resource (or not) following the Value Flows convention. Current implementation mostly supports consume, transfer and produce (see: Backtracking of supply chains scenario, section)
- Provider and Receiver: User performing the action and receiver of the action, defaults to the current user
- Quantity: Quantity used by the action measured using the resource defined units.
- Note: Free text field to annotate observation.

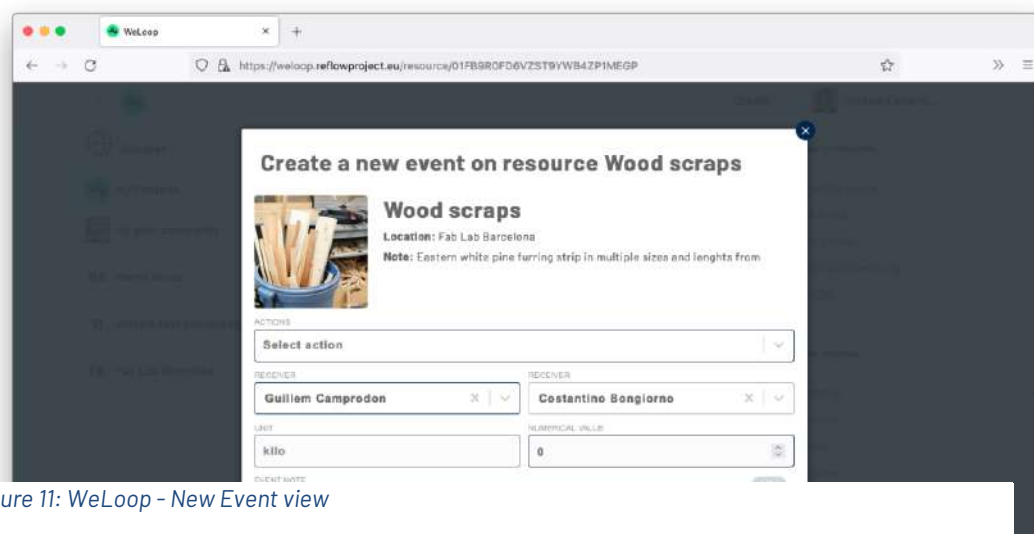


Figure 11: WeLoop - New Event view

Intents

The intent page is a preliminary feature for users to describe potential future events which have not been agreed to by other users, such as offers and requests. Intents are often used for discovering another agent to participate in the desired event. In WeLoop intents are always associated with a particular Community, to allow other members of the community to discover them(see Figure 13). At the time of this report writing, functionalities to match intents and resources are not yet fully functional. However, a preliminary version can be already used as part of the application(see Figure 14). Currently, the intents implementation is limited to the attributes below (see Figure 12).

- Name: Free text field to name the potential future resource
- Community Name: Allows to select an existing community where the intent will be published.
- Location: Allows to select from existing locations or provide one using Google Maps API geo-search. The location name or address is automatically translated to coordinates.
- Unit and quantity: Quantity available is measured using a predefined list (but not limited) of units (litre, kg, meter)
- Note: Free test field to annotate observations

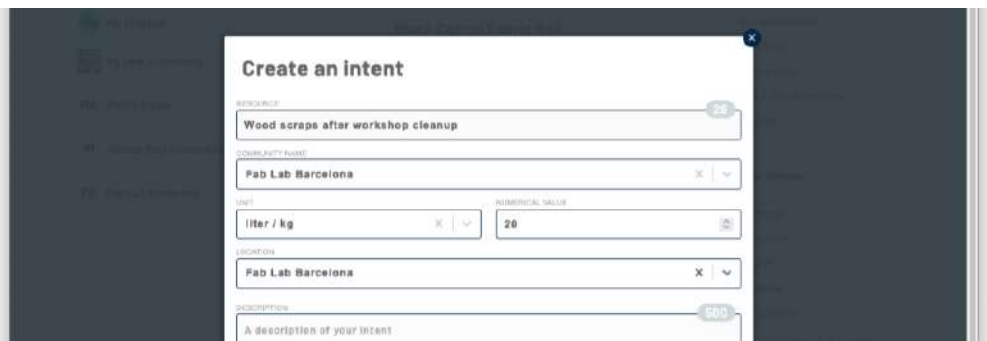


Figure 12: WeLoop - New Intent view

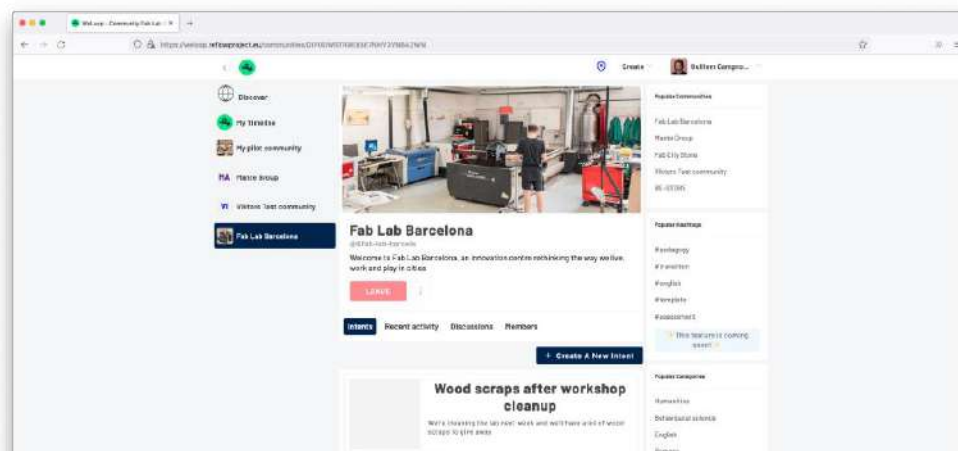


Figure 13: WeLoop - Community Intent view



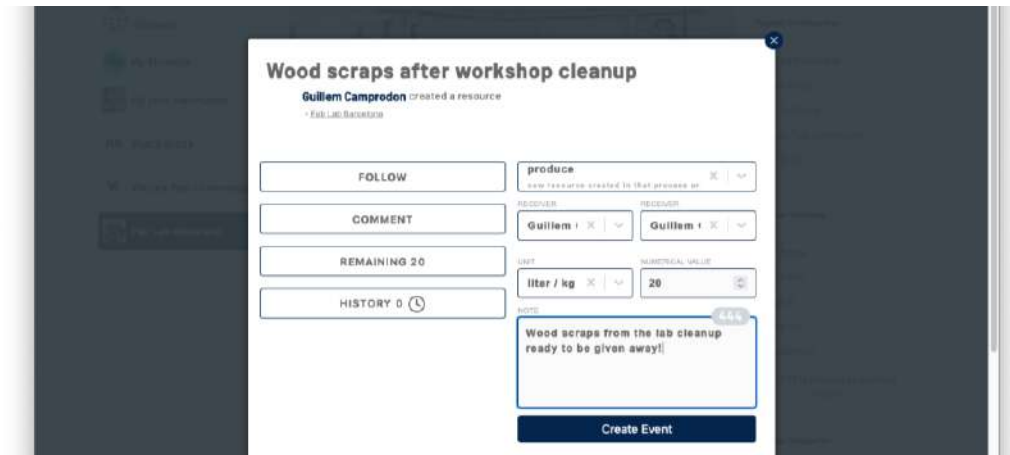


Figure 14: WeLoop - Community Intent create Resource throughout Event

Users

The users page displays the user profile and allows users to follow each other (see Figure 15). The primary sections are:

- Recent activity: Displays the latest user actions: publications, comments, economic events (currently not fully operational)
- Starred: Conversation and publications starred by the user
- Communities: Communities followed by the user
- Inventory: List of available resources owned or resulted from owned resources

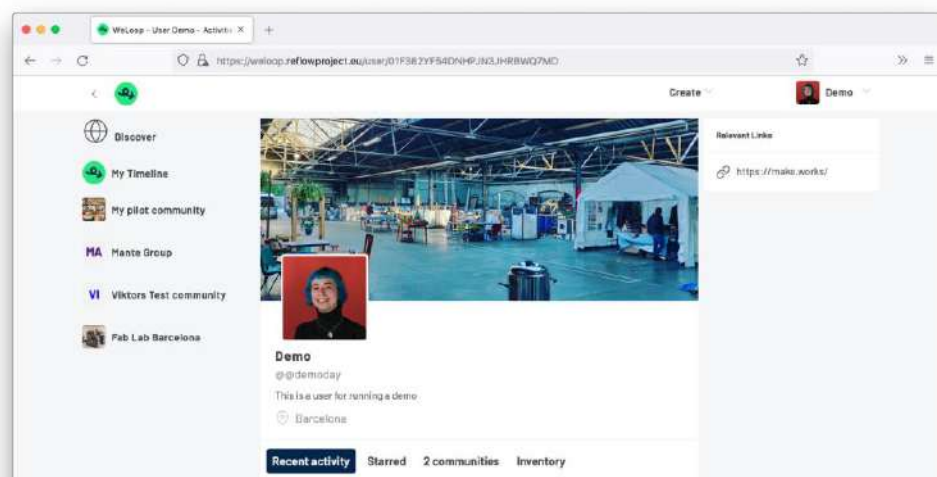


Figure 15: WeLoop - User Page view



Collection

Like communities, collections allow users to group resources under the same scope to access them later. Users can follow the activity of a collection by selecting one from the collections list (see Figure 16). At the time of this report writing, that functionality is disabled because it is still to be fully finished.

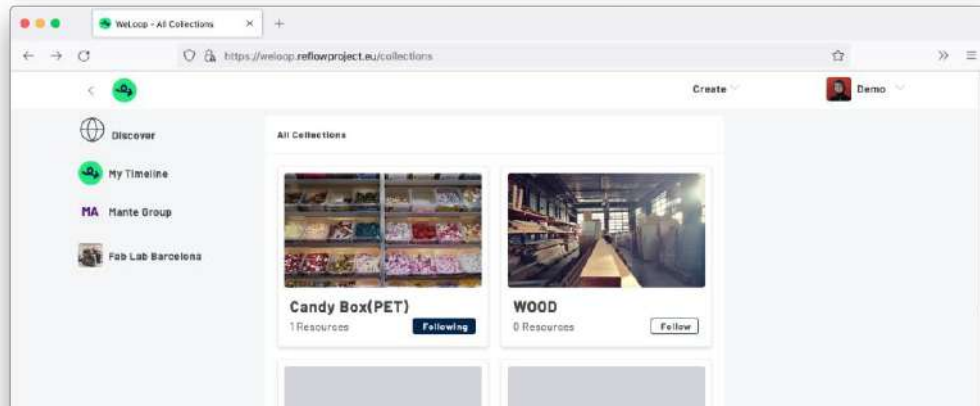


Figure 16: WeLoop - Collection's view

Discover

Allows users in a WeLoop instance to discover relevant information about the available users and communities (see Figure 17).

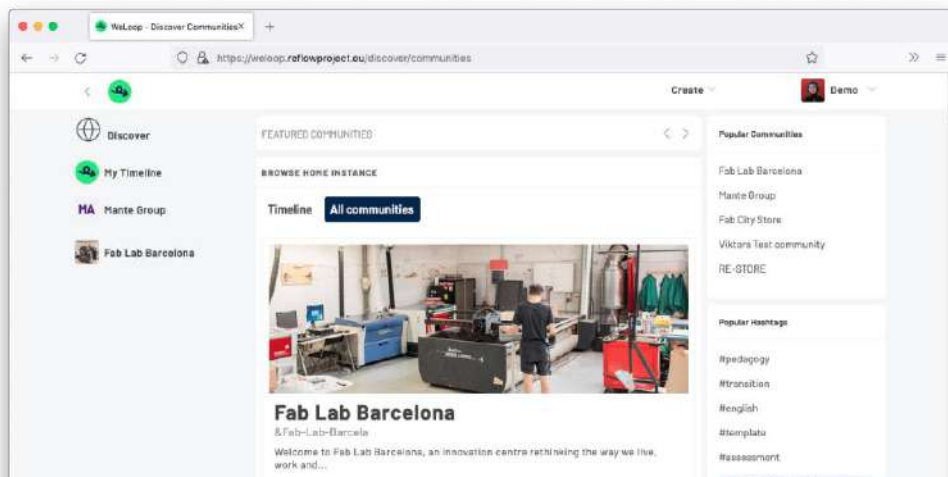


Figure 17: WeLoop - Discover view



Secondary pages

Other secondary pages (see Table 2Table 1) are available in WeLoop to perform common operation found in many similar web applications. See the *Reflow OS UI Library* section below for more information about how to use them.

Table 2: WeLoop - Secondary pages

Page	Description
Signup and Login	Standard sign up and login views to authenticate users
Create New Password, Reset Password, Confirm Email	Standard user confirmation and password retrieval views
Settings	Allows users to set settings related to their account
Not Found	Requests are redirected here when not found (404)

Customization and reusability

WeLoop goal was never to develop a solution that supports all Pilots needs but to develop a sample client on top of Reflow OS, and more importantly, a modular framework to help others build their applications.

Reflow UI Style guide

Reflow UI Style¹⁴ guide is a set of standards for the formatting and design of the project user interfaces (see Figure 18). It is not part of the WeLoop application itself. Instead, it provides a much general set of guidelines mean to be used for the development of all Reflow project user interfaces, such as the Reflow Open Data Dashboard. It establishes standard style requirements to improve communication by ensuring consistency both within an application interface and across multiple applications. The style covers best practices in usage, layout, colour schemes and typography. It also provides design recommendations for common UI elements such as buttons. Finally, it also includes a collection of icons to represent key project concepts and to help as a guideline to design further visual assets.



Figure 18: Reflow OS Style Guide sample

¹⁴ Download link <https://github.com/reflow-project/welooop/blob/main/reflow-visual-design-guidelines.pdf>

The Reflow UI Style guide provides only visual recommendations and is meant to be implemented on top of any UI software framework. However, on top of it, we provide the Reflow OS UI Library, a React JS component library design to facilitate even more the adoption of the Reflow project style guides (See Reflow OS UI Library section below).

Reflow OS UI Library

WeLoop is also the Reflow OS UI Library available at weloop.reflowproject.eu/docs/ui¹⁵ (see Figure 19). It is built using isolated components to make customization faster and easier by allowing each component to provide only a functionality at a time. That means Pilots can develop entire UIs without needing to start from scratch.

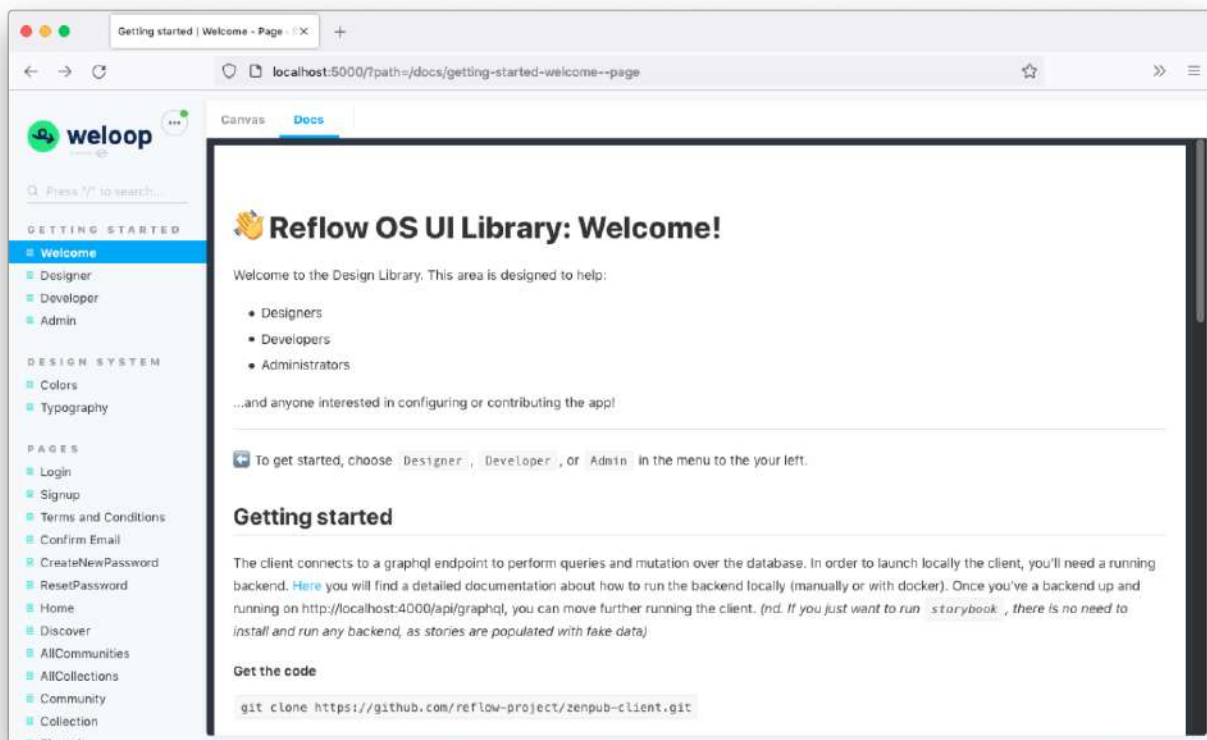


Figure 19: Reflow OS UI Library - website

¹⁵ Full Reflow OS UI Library URL <https://weloop.reflowproject.eu/docs/ui>



That is enabled by Storybook¹⁶, a component library and an open-source tool for UI development. The main objective of the storybook is to provide an isolated and independent environment, called 'Sandbox', to create components outside the context of the main application. Storybook helps you document components for reuse and automatically visually test your components to prevent bugs.

Reflow OS UI Library makes it quick and easy for pilot leaders, designers and developers to get up-to-speed on how pages are constructed and how all the functionality works (see Figure 20).

- As a pilot leader, you may be interested in configuring the interface for your context or adding new functionalities and features. Reflow OS UI Library can help to learn how to configure it for the pilot users. The fastest way to get started is to: Go through the pages listed in the menu, familiarise yourself with the various layouts and check out the components that build each page.
- As a developer, you may be interested in contributing to the development, configuring it for a particular context or adding new functionality and features. The fastest way to get started is to use Reflow OS UI Library to go through the pages in Storybook and familiarise yourself with the various layouts. Check out the components that build each page and use the Show code option at the bottom-right of a page. Find the element that you would like to change.
- As a designer, you may be interested in configuring it for your context or adding new functionalities and features.

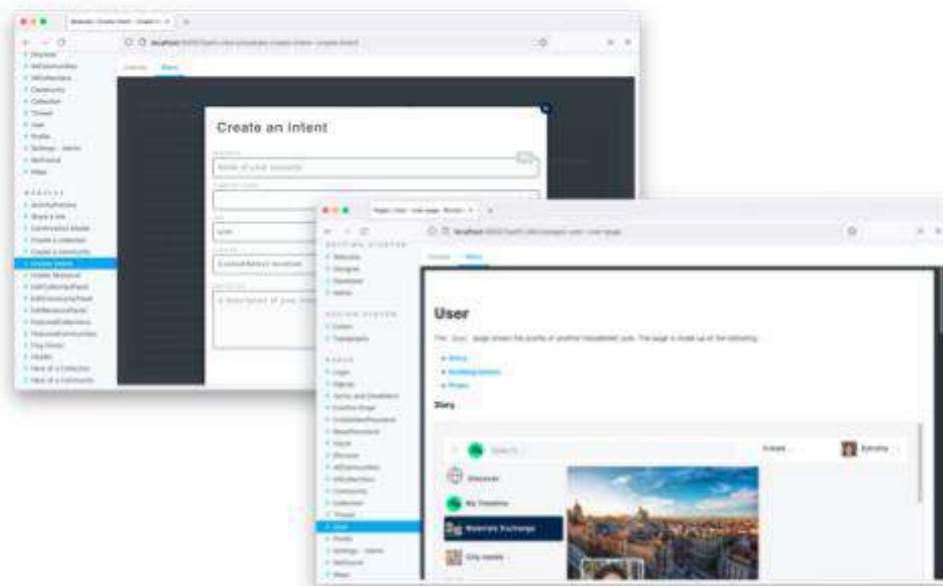


Figure 20: Reflow OS UI Library - multiple views

¹⁶ Visit the Storybook website (<https://storybook.js.org/>) for more general information and examples about it

Main modules

Below (see Table 3) are the main software modules currently available in WeLoop. They are reusable interface modules used to compose the pages described above.

Table 3: WeLoop - Main modules

Module	Description
Activity Preview	Provides ways to display users' activity commonly found in social networking sites.
Confirmation Panel	Provides a confirmation modal for user actions
Create Collection Panel	Allows to display multiple publication collections
Create Community Panel	Provides a display for communities
Create Event on Resource Panel	Provides a panel to create economic events on top of resources
Create Intent Panel	Provides a panel to create intents for future economic events
Create Resource Panel	Provides a panel to create an economic resource
Edit Collection Panel	Provides a panel to edit an existing collection
Edit Community Panel	Provides a panel to edit an existing community
Edit Resource Panel	Provides a panel to edit an existing resource
Featured Collections	Displays a collection of collections

Hero Collection	Provides a module to display a collection
Main Header	Displays the main website header
Map	Displays a map given the right markers representing one or multiple locations
Proposed Intent Panel	Displays proposed intents
Sidebar	Provides a way to display content on the side of the page such as links to other relevant sections
Update Resource Panel	Provides a way to update an existing resource

Development environment

The WeLoop client connects to a Reflow OS backend. In order to launch the client locally, you'll need a running backend either locally or remotely. You will find detailed documentation about how to run the Reflow OS backend using docker in reflow.dyne.org¹⁷. Once you have a backend up and running, you can move further to running the client.

Get the code

```
git clone https://github.com/reflow-project/weloop.git
```

Install the needed dependencies

```
cd weloop
```

```
yarn
```

Launch WeLoop

```
yarn start
```

Visit <http://localhost:3000>

Launch Storybook

```
yarn storybook
```

Visit <http://localhost:5000>

Load specific the languages

WeLoop can use linguists to provide multi-languages. Before launching the app, make sure you've added one or more translations to the app.

Currently, some components but not the complete application are translated in the following languages

```
en_GB: 'English, British',
```

```
en_US: 'English, USA',
```

```
es_MX: 'Español, Méjico',
```

```
es_ES: 'Español, España',
```

¹⁷ Reflow OS oficial online documentation <http://reflow.dyne.org/>

```
fr_FR: 'Français, France',
```

```
eu: 'Euskara',
```

```
ar_SA: 'العربية, المملكة العربية السعودية'
```

Apart from en_GB that is the default language, feel free to add the needed translations for your scope, when launching the script:

```
yarn add locale en_GB en_US es_MX es_ES fr_FR eu ar_SA
```

Customize your instance basic settings

To quickly start the project, we only need to create a .env file pointing to the REFLOW GraphQL API.

Example part of an .env file (see .env.example for the whole file):

```
BASE_URL=https://api.reflowproject.eu
```

```
REACT_APP_GRAPHQL_ENDPOINT=https://api.reflowproject.eu/api/graphql
```

For more advanced settings you can edit directly the configuration settings in ./src/mn-constants.tsx

Remote monitoring

WeLoop integrates with [Sentry](#), a cross-platform application monitoring, with a focus on error reporting. It allows you to remotely detect JavaScript performance issues before they become downtime. View the entire end-to-end distributed trace to see the exact, poor-performing API call and surface any related errors. Sentry is fully [open source](#) and instances can be locally deployed by Pilots together with the other Reflow OS tools. For preliminary test purposes, Sentry can be used as a SAAS from [sentry.io](#) without any installation required. To use it you need to provide a valid Sentry Key in your .env file under REACT_APP_SENTRY_API_KEY

Issues reporting and contributions

Found issues can be reported using the standard Github issues¹⁸ workflow. A banner in the application explicitly indicates users to report issues that way. Pilots are encouraged to fork the application and develop their own custom version. Contributions can be submitted using Pull Requests. They will be reviewed and integrated into the source code using the standard Github pull request procedure.

¹⁸ Issues Reporting Page <https://github.com/reflow-project/weloop/issues>

Production deployment

Complete deployment of the full bundle will be described as part of the overall Reflow OS installation guide.

Self-hosted

Manually use `yarn build` to generate the `./build` folder including the `index.html` file and all the static dependencies. Host the resulting website on any static web server such as NGINX or Apache Web Server. Alternatively, a fully bundled version of a full Reflow OS instance including backend and front-end (WeLoop) is being under development and will be documented as part of the main documentation at reflowos.dyne.org¹⁹.

Github Pages

WeLoop main repository uses Github Actions to automatically deploy the repository default branch as a public website using Github Pages every time changes are pushed. Anyone can fork the existing repository and deploy their own version quickly. Custom domains are defined via a `./build/CNAME` following the standard Github pages workflow. Two Github Secrets can be defined to customize the app `REACT_APP_GRAPHQL_ENDPOINT` and `REACT_APP_SENTRY_API_KEY`. Learn more about them in the *Customize your instance basic settings* and *Remote monitorings* sections.

Licensing

The present software is released under a copyleft license (*GNU Affero General Public License v3.0*) to promote use, customization and contribution inside and outside the consortium. The license conditions developers to make available the complete source code and any modifications. It fosters an open ecosystem and protects rights users by granting them access to the source code.

¹⁹ Reflow OS official online documentation <http://reflow.dyne.org/>



Conclusions

For the first time, a functional and easy to use interface showcasing the foundation of Reflow OS is available. Pilots have been experimenting with WeLoop, despite the roughness of the beta software, for quite a long time. The authors have seen how it has contributed to the understanding of the fundamental's concepts behind Value Flows.

Over the next few months, we hope to see pilots using it to unlock the full potential of WP2 tools to support the implementation of the project local actions. We hope to see pilots deploying WeLoop custom instances by making use of tools such as the Reflow OS UI Library.

At the same time, as part of the new Bonfire component, a new user interface will be released that might contribute even more to showcase the number of solutions Reflow OS supports. Thanks to the use of modular, well documented and open-source software we ensure Reflow OS evolves further than the project timeline, supporting a whole new generation of circular initiatives to make EU cities more sustainable.

