Co-Creating Circular Resource Flows in Cities

constRuctive mEtabolic processes For materiaL flOWs in urban and peri-urban environments across Europe

Dyne.org foundation

REFLOW Architecture and Manual for Distributed Network Setup and Maintenance

Authors: Ivan Minutillo, Denis Roio, Adam Burns

Start date of project: 01/06/2019

Project 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.

REFLOW is an EU Horizon 2020 research project running from 2019-2022, which aims to enable the transition of European cities towards circular and regenerative practices. More specifically, REFLOW uses Fab Labs and maker-spaces as catalysts of a systemic change in urban and peri-urban environments, which enable, visualize and regulate "four freedoms": free movement of materials, people, (technological) knowledge and commons, in order to reduce material consumption; maximize multifunctional use of (public) spaces; and envisage regenerative practices. The project will provide best practices aligning market and government needs in order to create favourable conditions for the public and private sector to adopt circular economy (CE) practices. REFLOW is creating new CE business models within six pilot cities: Amsterdam, Berlin, Cluj-Napoca, Milan, Paris and Vejle and assess their social, environmental and economic impact, by enabling active citizen involvement and systemic change to re-think the current approach to material flows in cities.

This document and the free and open-source software referenced sits at the core of the innovative developments in REFLOW technical work-package and implements an on-line backend and frontend stack for various use-cases observed in pilots. This document is also an operating manual for the implementation of new clients, includes a set of GraphQL examples for the main functions and an installation and deployment manual for system administrators.

The latest version of this document is available on https://github.com/dyne/reflow-docs.

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

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

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

Introduction

ReflowOS is an on-line and on-site set of operational tools for communities who want to create federated and secure economic networks. The main goal of ReflowOS is to facilitate the coordination, organization and documentation of distributed value chains. It implements the following features:

- Enhance transparency and integrity of the information stored
- Connect multiple instances in a federated context where is possible to import and export balanced values

What is an economic network?

An economic network is a set of independent agents, which could be individual persons or organizations, who regularly collaborate to create and exchange goods and services. Examples of economic networks include supply chains, joint ventures, municipalities, and bio-regions, but also economic ecosystems surrounding a base resource like the Android OS.

Each economic network has its own rules for accessing and participating in economic activities. These rules, and how they are practiced, constitute the governance of the network. Within the REFLOW context, the goal is to create an economic network that encourages the development of a municipal circular economy in order to:

- Promote the inclusion of people and/or organizations (that inhabit the urban area) in the production processes and in the creation and distribution of the generated value

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

- Trace the flow of resources in the urban and peri-urban area to reduce waste and increase reuse and recycling practices
- Plan production processes based on availability of resources, vehicles and people within the network, to encourage the growth of the local economy and its participants
- Certify the supply chain of a resource produced locally, to create incentives and narratives between producers and consumers
- Collect data from the territory to promote the creation of new policies, which can scale from a municipal to a national level

Our work is to build and test a set of possible answers to the question posed by Kate Raworth in her book Doughnut Economics: **can we produce for human needs, without exceeding planetary boundaries?**

Aware of the fact that technology influences economy and politic, we believe that this kind of infrastructures can help those fearless communities/municipalities that want to experiment with economic networks to plan economies that can become a real alternative to the ideology of the free market, a prevailing yet no longer sustainable.

Network

The OS is the entry point to interact with such network, and it can be private and permissioned - leveraging on TOR to anonymize all the data flows - or open and public, giving the possibility to any user to start a node and participate through the web.

Economic

It is **economic** in a way that goes far beyond the merely trading and exchange of resources, but allows nodes to discover each other needs and offers, follow resources transformations around the network, trace back the history of a resource, manage the inventory and coordinate activities together to reach mutual goals.

To think as Terrestrials

Such economic networks allow a wide range of use cases to be designed and developed. it may require a slightly change in perspective for participants about how to look at "business" and relationship with economic partners, how to define and follow governances that are not driven only by profit and that can allow the participations of a diverse range of actors, and how to address ecological and political challenges without losing economic sustainability. We need to think as terrestrial, quoting Bruno Latour.

They're open challenges and without a unique answer, and none of them are solved by the lone adoption of a technology.

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

Use cases

In this section we illustrate the design thinking prompted by the observation in use-cases in Reflow and leading to technical choices made in realizing the ReflowOS.

More than maps and exchanges

Through the definition of personas and the collection of user requirements, reported in the D2.1 deliverable Use Case Analysis and Requirements, we had the opportunity to get an overview of characteristics and needs of a wide range of actors who will plausibly be part of the economic networks. Although the needs differed from userbase to userbase, a recurring theme emerged during the several sessions that led to the Use Case Analysis and Requirements document, concerning the lack of coordination between actors within the urban area.

This lack of coordination led to several issues:

- Not having enough information on materials / skills available to be exchanged/traded over the territory
- Not knowing actors that have a complementary role in the same supply chain and then not being able to engage in partnerships with them
- Not being able to guarantee the traceability of the origin and manufacturing of locally produced goods

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

- Failing to encourage the inclusion of new stakeholders (citizens and / or organizations) in the city's circular economy processes
- Failing to compete with the economic convenience imposed by the capitalist system

These problems led to the ideation of a first series of solutions:

- Creating a map showing the flow of resources within a network
- Being able to offer and request resources and skills
- Being able to describe the available resources in different levels of detail, in order to create an open inventory distributed throughout the territory
- Being able to trace the different types of economic activity that affect a product in the various stages of its production, in order to create a "material passport" that verifies the value of products that are produced in a virtuous and sustainable way
- Being able to plan and coordinate supply chains within the urban/peri-urban area, in order to include different actors in different phases of the production of goods
- Distributing the value generated from the production of goods, based on each stakeholder's contributions to the production phases

The use cases presented in this section are intended to be food for thoughts of what can be achieved with ReflowOS.

Defining the network sentiment is a practice that allows participants to choose different lenses to observe a network and the data produced, and which constraints, shades, angles, aspects to highlight. This helps building more solid and data-driven assumptions.

These data-driven assumptions, far from being "the reality", can be used as a map to navigate the tangled mess of interconnected data that are produced by a network, looking for valuable insights that can empower human discussions and decisions.

"The map is not the territory" (Alfred Korzybski, 1932)

Taking actions - backed by such data-driven analysis - may have a meaningful impact upon complex networks such as a municipality, where usually is hard to connect and intertwine data together in reliable ways, and where most of the information are gathered mostly through questionnaires or direct involvements/experiences.

How it works

With ReflowOS each participant can record different types of economic activities that are performed over time.

One such economic activity could represent the production of a new resource (eg. a cake).

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

Productions usually are preceded by a series of activities: transformations, transfers, exchanges, some work done, tools used - any step that contributed to produce the resulting good.

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 an exchange in the local market, which person was involved in the production of the cake and how much time it took, and the quantity of resources consumed.

Having such data available only for one cake is not particularly meaningful, but how would things be different if such data was recorded for example for all the cakes sold in the municipal markets?

The network sentiment analysis could help us find insights about which ingredients are the most used in the niche we're exploring. Where such ingredients were retrieved, if there are particular activities that monopolize the area and what are the logistics involved in all the steps of production.

Such observations, when available openly, can stimulate discussions and actions: participants in that area can find ways to coordinate their activities in order to optimize the logistic, with the end goal of reducing CO2 emissions, or trying to grow local producer contribution to the market, among other things.

Example

Network sentiment can act as a sentinel for taking specific actions over a network or as a way to ease the planning of specific activities, having the possibility to check actions against the data produced.

The material passport (MP) gives us meaningful insights about the genealogy of a specific product.

Rather than merely focusing upon the qualities of the end product, it provides information about the whole set of actors, tools, collaborations, agreements, efforts and energy involved in its production, transportation and disposal.

Such perspective empowers actors to look at a more inclusive story of a product, according to its specific and unique flow of transformations: the value chain.

Empower citizens

The MP is a powerful way for citizens to discover and understand - in a critical way - the use value and exchange value of a product, untied from its marketing and branding.

It can promote the flourishing of a more conscious and municipally driven business, where products with MPs that highlight waste reduction, low energy consumption, local production, or even specific producers or resources involved in the production which are preferred.

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

It helps to understand the urban metabolism the citizen is part of how resources flow in, where they are stored and who are the actors actively involved in their transformations and transportation.

How can such actors collaborate together and eventually how can citizens fit in such value chains?

Empower producers

The MP helps producers to improve the product value chain, for example reducing the waste and efforts involved in each step of a product life cycle. It can strengthen relationships between producers and suggest new opportunities for new or improved supply chains.

Some of the relevant data showed in a material passport includes:

- Design: any relevant design that has been used for the manufacturing of the product
- Resources inventory: the set of resources that have been used and are part of the end product
- Transformations: events that affected the product, step by step in chronological order
- The places, actors and tools involved in the product supply chain

Conclusion

The material passport is an essential tool to empower different stakeholders of the city’s social and economic life to have meaningful and productive discussions together and to produce new narratives or boost existing ones, in order to switch toward a more circular and fair economy.

With ReflowOS, every participant manages their personal inventory, which contains all their available resources, correlated with a set of meaningful data to help describe and categorize them.

Resources in ReflowOS can be produced or consumed, offered or requested, transformed or used, traded, and even just cited in other processes.

Every activity performed on a resource can affect it in different ways, therefore it is essential that each resource is updated accordingly after every transformation, and showed with the updated information.

Such updates are reflected in each participant’s inventory, that will allow the user to search, filter and view all the resources.

The user can also enter see details for each resource: like dimensions, quantities, categories and the related material passport for each of them.

Each participant only has access to their own inventory. The only way other participants can discover available resources is by looking at published proposals (offers).

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

Reflow is a decentralized network

The Reflow network is based on open protocols and standards to form a decentralized collection of independent *instances* that send, receive, and store data.

Components of a Reflow instance

A Reflow instance comprises:

- the Reflow server (running on a Elixir-capable web server)
- a Reflow client (typically static JS/HTML/CSS/images which can be served by any web server)
- a database managed by PostgreSQL
- a file store for uploaded files
- an optional search index (powered by Meili)

Zooming in on one instance

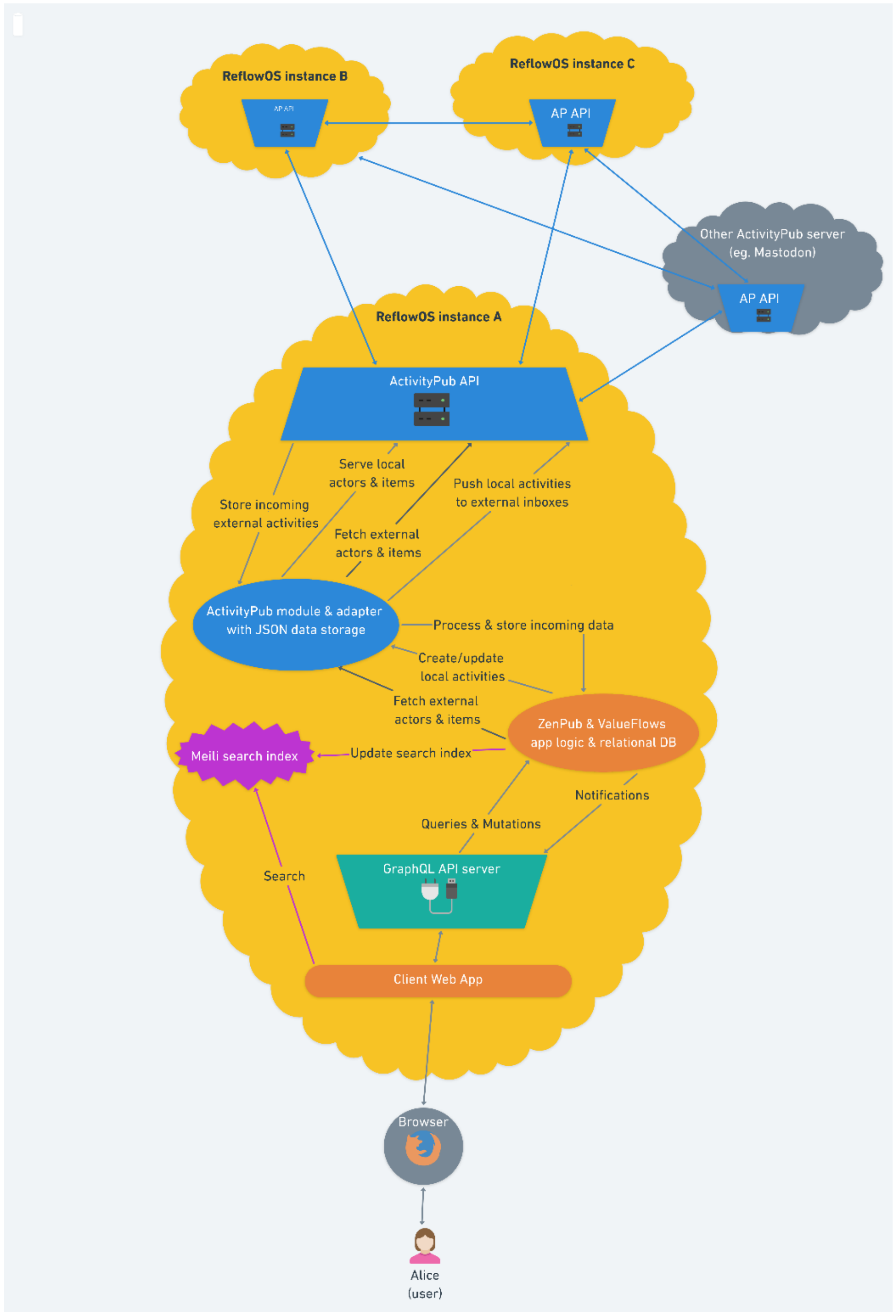
All of these components can run on a single server; or they can be separated with many load-balanced web-servers, a database cluster, and a file-server; or anywhere between those extremes.

Each instance stores the full data for local users, groups, and economic resources and activities (including personal information and private messages).

Due to federation, the instance also will contain partial data that has been federated from external users across the network. Each item in the database and search index always refers back to the originating instance for the full data (e.g.: memberships, roles and permissions; current status, quantity or balance).

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

Data flows


ReflowOS instance B
AP API
ReflowOS instance C
AP API
Other ActivityPub server (eg. Mastodon)
AP API
ReflowOS instance A
ActivityPub API
Serve local actors & items
Store incoming external activities
Push local activities to external inboxes
Fetch external actors & items
ActivityPub module & adapter with JSON data storage
Process & store incoming data
Create/update local activities
Fetch external actors & items
ZenPub & ValueFlows app logic & relational DB
Meili search index
Update search index
Notifications
Queries & Mutations
Search
GraphQL API server
Client Web App
Browser
Alice (user)

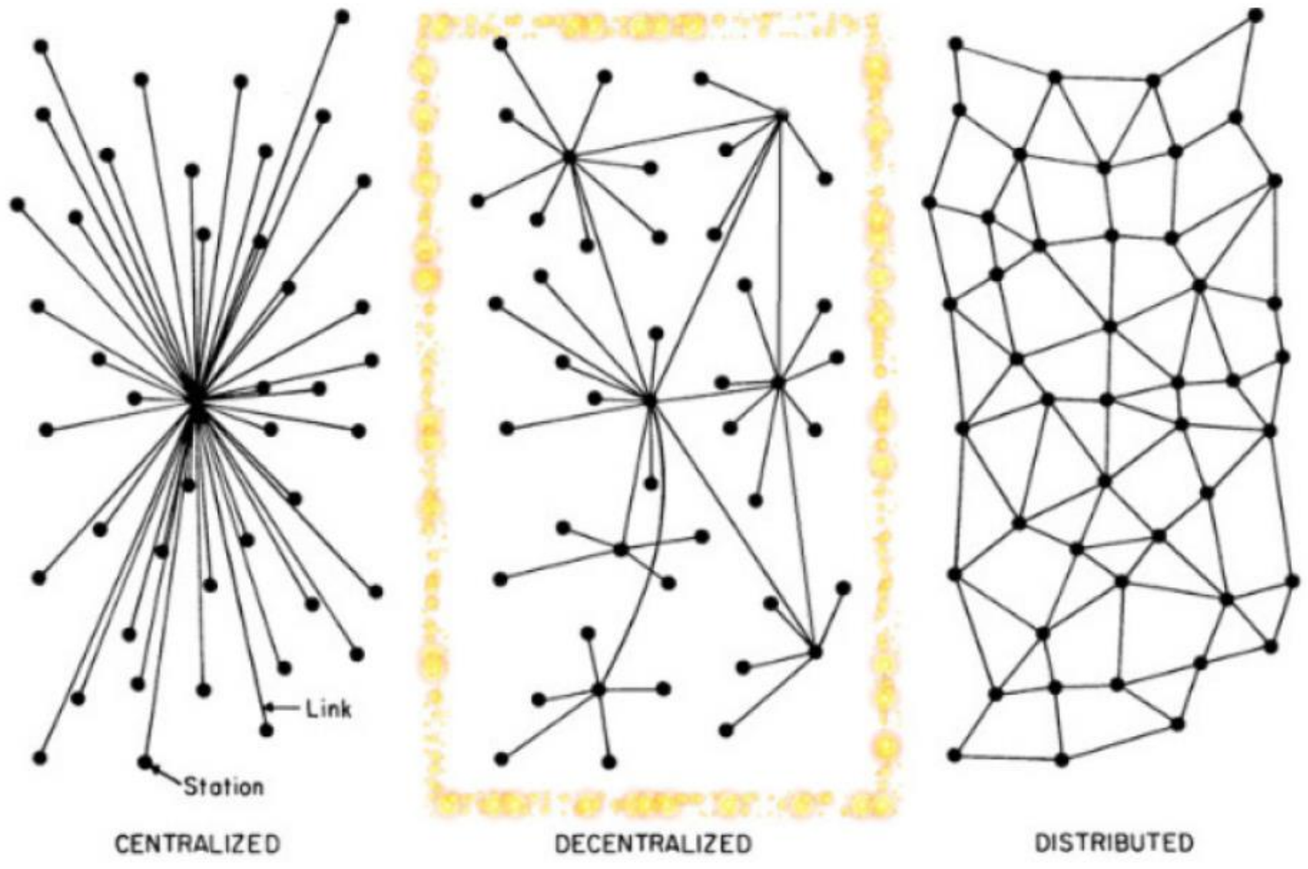
This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement number 820937.

Federation

A **federated** social network is a system that is decentralized, developed, and operated by distinct providers (something like email but for social networks). It consists of multiple social apps and websites, where users of each site can communicate with users of any of the other compatible sites.

From a societal perspective, one may compare this concept to that of social media being a public utility, rather than a centralized service that a company tries to monopolize by locking users in.

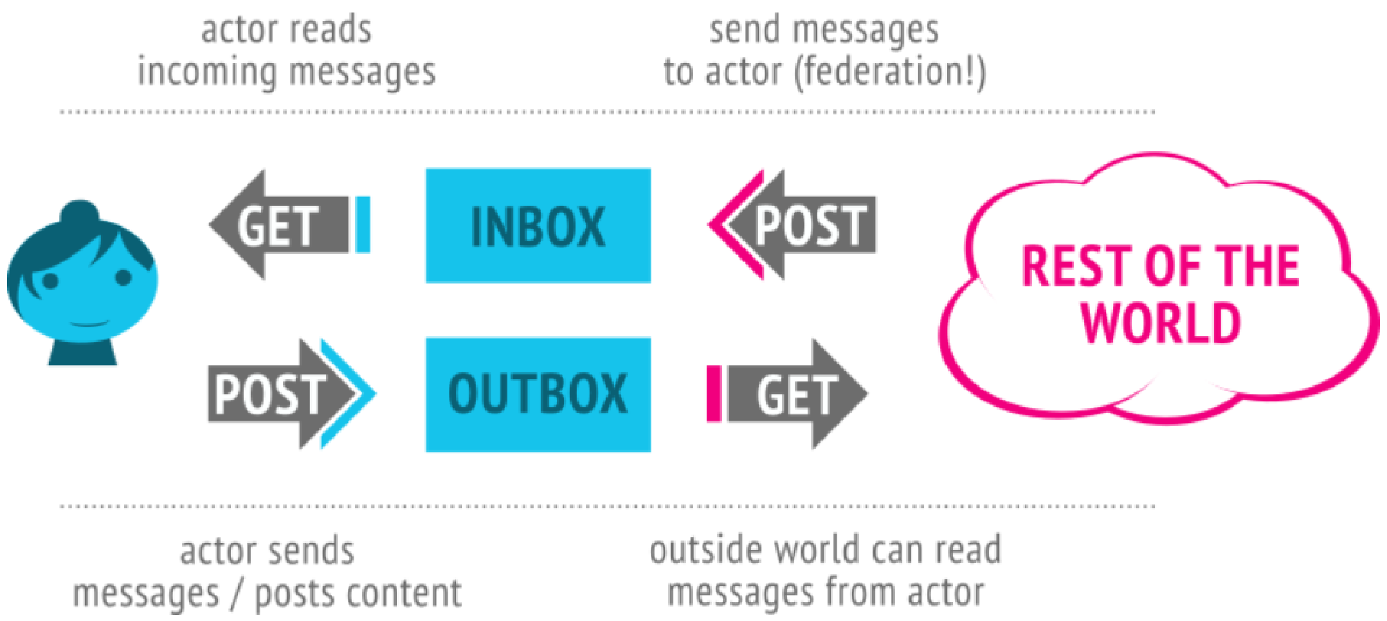
An application or website participating in a federated social network is interoperable with the other sites (known as *instances*) and federates (fetching and especially pushing data) with them based on their users' interactions. Communication among the social websites is conducted through *social networking protocols*.



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

Federation protocols

ActivityPub is an open, decentralized *social networking protocol*, which defines a standard federated server-to-server API for delivering content and notifications.


actor reads incoming messages
send messages to actor (federation!)
GET
INBOX
POST
REST OF THE WORLD
POST
OUTBOX
GET
actor sends messages / posts content
outside world can read messages from actor

ActivityStreams is an open format specification [comprised of a JSON-based syntax and an extensible vocabulary] used to syndicate user activities in social web applications and services. It provides a general way to represent activities, using a simple actor+verb+object structure, for instance "*Alyssa followed Ben*" would be represented as *actor:alyssa, verb:follow, object:ben* or "*Ben added Paris to his list of places to visit*" would be represented as *actor:ben, verb:add, object:paris, target:places_to_visit*.

Both ActivityPub and ActivityStreams are web standards defined by the W3C (World Wide Web Consortium) and are used by many open source federated applications.

Through RSS, a user can subscribe to almost any news/blog/content feed in the world from any number of independently developed feed readers. ActivityPub is having the same impact on social network interactions. It enables a decentralized social web, where a network of servers talk to each other on behalf of individual users, very much like email service in the sense that it is made up of different apps operated by different providers but with all users being able to communicate regardless.

Advantages of a federated network:

- Robust and resilient

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

- Scales horizontally
- Private data stays on each user's home instance
- Each instance is responsible for its content (moderation of discussions, etc)
- Standards-based but also extensible
- Open

Economic federation

The above protocols and standard vocabularies are made for social activities, but what about conducting economic activities?

Fortunately, these standards were designed to be extensible, and even more fortunate, some fine folks have been working on something that fits the gap perfectly: ValueFlows is a set of common vocabularies to describe flows of economic resources of all kinds within distributed economic ecosystems.

Reflow has been developed as an implementation of a subset of ValueFlows vocabularies, the Reflow client API is based on ValueFlow's GraphQL API, and all economic data can federate over ActivityPub, by making ValueFlows JSON-encoded data available as an ActivityPub *actor* (for people, organizations, and groups), *activity* (eg. economic events) or *object* (eg. economic resources, proposals, processes, etc).

What does it mean for Reflow?

We have built upon these standards so that any interaction one may have with an actor (person, organization, group), or object (eg. a proposal or an economic resource) within the system (eg. posting a need in a neighborhood group, or recording the use of a 3D printer in a Fablab of a city one is visiting) can happen seamlessly even if the other people/groups involved are not users of the same instance, for example if one signed up to the Amsterdam Fablab of Reflow and then is interacting with resources and recording events on othe premises of the Barcelona Fablab, or another example can when there is more than one Fablab in the same city and resources and events by each of them can be networked and made available through every federated instance.

This can foster more connection and cooperation between participants in Reflow, but thanks to the heavy use of interoperable standards, it also means that future adaptations of Reflow as well as entirely new projects will be able to integrate with Reflow as well.

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

Valueflows

A vocabulary for the distributed economic networks of the next economy

Value Flows is a set of common vocabularies to describe flows of economic resources of all kinds within distributed economic ecosystems.

The purpose is to enable internetworking among many different software projects for resource planning and accounting within fractal networks of people and groups.

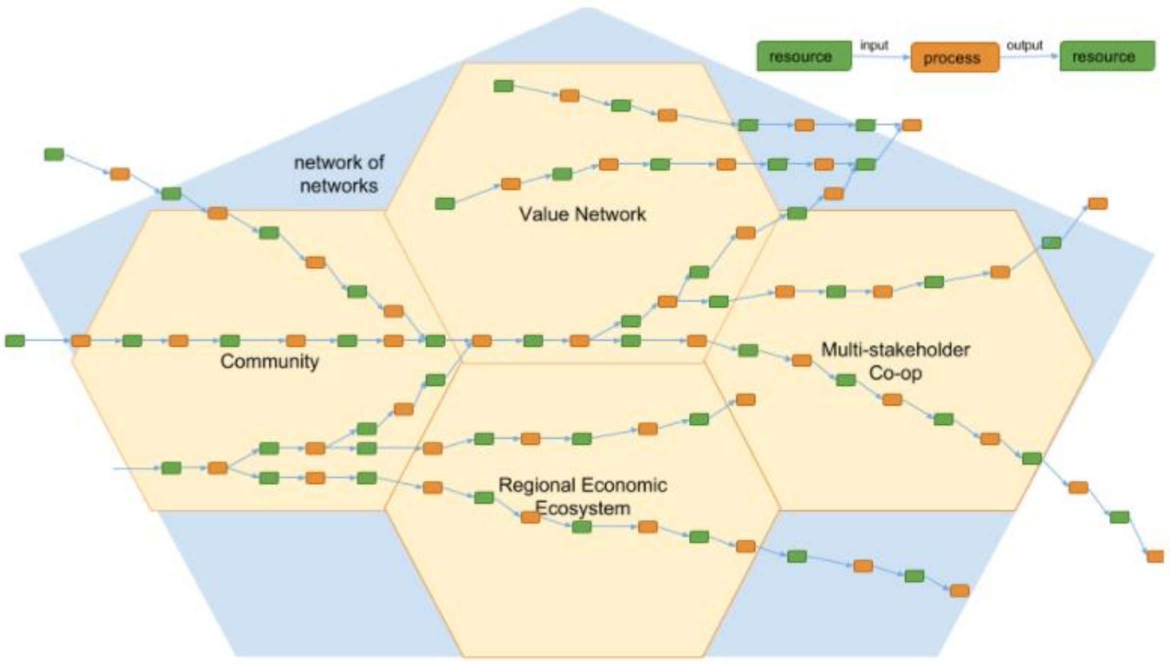
The vocabulary will work for any kind of economic activity, but the focus is to facilitate groups experimenting with solidarity / cooperative / collaborative / small business ecosystem / commons based peer production / any transitional economies.

Or, with less buzzwords, "let's help a lot of alternative economic software projects that are solving different pieces of the same puzzle be able to work together".

Another purpose is **to support resource flows** connecting many software applications. These flows may be oriented around processes that make things, exchanges that trade things, or combinations of both.

The vocabulary is presented in several ways, as *Linked Open Data* using the Resource Description Framework (RDF) family of languages, as well as a *GraphQL* reference, *json-schema* reference, and a *UML* model. We want to support RDF based and non-RDF based use of the vocabulary, basically any way that people want to use software and data on the internet to help create economic networks.

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



Useful links

[ValueFlows Website](https://valueflo.ws) https://valueflo.ws

[ValueFlows Code](https://lab.allmende.io/valueflows) https://lab.allmende.io/valueflows

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

Network metabolism

Any economic event stored in the OS can be analysed in relation to previous or following events, the more activities are performed and stored, the more data will be available for shaping and studying emergent flows that help defining the network metabolism.

Collecting offers and needs

Imagine you have just joined a network, and you may know some of the participants, but you are still quite alien to the overall dynamics and existing interactions.

The network allows you to publish to other participants what your **needs** are, and what you have available to **offer** (your skill, tools, services, resources…).

It also exposes the overall needs and offers present in the network that are not yet satisfied.

This activity can have a great impact over network participants, including:

- Being able to analyze the scarcity and abundance of resources and skills available in the network
- Based on the analysis, stakeholders can find ways to overcome scarcity or adjust their economic activities in relation with the network sentiment
- Engage in new economic relationships and discover new economic partners

But this is still far from generating an effective model for representing the network metabolism.

To enable a proper network metabolism model, participants need to input any activity that affects a resource, to keep track of its whole journey.

Such recorded activities can include the transfer of a material from one place to another or the consumption of a material involved in the production of a new one.

Another useful activity to record is the usage of a resource, *eg. each time a 3d printer is used*.

Such activities aren't only about the raw event that happened to a resource, but they can be expanded with a set of metadata to cover additional essential information, like a description of the event, the resource, dates/times, a set of tags to facilitate the aggregation and filtering of data or a reference to a commitment or a process, and so on.

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

The network metabolism is not only about **intents**, namely the combination of offers and needs expressed by participants but is *"the sum total of the technical and socio-economic process that occur in cities, resulting in growth, production of energy and elimination of waste."*[1].

Track and tracing of economic resources

To enable a proper network metabolism model, participants need to store any activity that affects a resource, to keep track of its whole journey.

Such set of activities can include the transfer of a material from one place to another or the consumption of a material involved in the production of a new one. Another useful activity record is the usage of a resource, *eg. each time a 3d printer is used*.

Such activities will not be only about the raw event that happened to a resource, but they can be expanded with a set of metadata to cover additional essential info. Such metadata can be a description of the event, the resource, datetimes, a set of hashtags to facilitate the aggregation and filtering of data or a reference to a commitment or a process, and so on.

Perform events

Imagine you've spent some time digging into the network and you're acquainted with its dynamics and sentiment, you now have a clear idea of what are the urgences and what the network can offer to you, now it's time to pass to action! Means, participants may start perform economic activities among them, requesting available resources and coordinate activities that involves different stakeholders. Activities will be linked together by the affected resources. Such links can then show relevant information about the whole value chain of a resource, as it is explained in this section. In our historical period, it is easy to confuse or reduce an economic network for a usual marketplace. The main differences are that a marketplace is focused **only** on exchanges, meanwhile an economic network allows users to perform economic activities (work, consume, produce, cite, use, transfer). Another fundamental difference is that marketplaces do not allow to keep track of resources but they only store the exchange between 2 participants, meanwhile in an economic network one of the core aspects is being able to create the flow that materials and activities create together in a specific period of time.

1 Kennedy, C., Cuddihy, J., & Engel-Yan, J. (2007). The changing metabolism of cities. Journal of Industrial Ecology, 11(2), 43-59.

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

Client API

Here is a series of sample queries and mutations for the Reflow client API (which uses GraphQL) that achieves the use cases proposed in this manual.

The API is flexible enough to satisfy a wider range of use cases and scenario, according to the stakeholders needs. Think of the following ones as a starting point to help shape your apps and ideas.

Authentication

Register a new user

```
mutation {
  createUser(
    icon: {url: String},
    image: {url: String},
    user: {
      email: String!,
      extraInfo: Json,
      location: String,
      name: String!,
      password: String!,
      preferredUsername: String!,
      summary: String,
      wantsEmailDigest: Boolean!,
      wantsNotifications: Boolean!,
      website: String
    }) {
      email
      isConfirmed
      user {
        preferredUsername
        displayUsername
        canonicalUrl
        ...
      }
  }
}
```

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

Confirm email

```
mutation {
  confirmEmail(token: String) {
    me {
      isConfirmed
    }
    token
  }
}
```

Login

```
mutation {
    createSession(email: String!, password: String!) {
      token # Use the token to activate a session with the server
    }
}
```

Get the current user details

The current logged in account and user. Contains different fields (including private information) than querying for myAgent.

```
{me {
  email
  isConfirmed
  isInstanceAdmin
  user {
    icon
    image
    name
    preferredUsername
  }
}}
```

Display currently logged in agent profile

```
 {myAgent {
  id
```

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

```
    name
    displayUsername
    canonicalUrl
    image
    primaryLocation {
      name
      lat
      long
    }
    intents {
      name
    }
    proposals {
      name
    }
    inventoriedEconomicResources {
      name
    }
    economic_events {
      name
    }
  }}
```

Display any agent profile

```
  {agent(id: ID!) {
    id
    name
    displayUsername
    canonicalUrl
    image
    primaryLocation {
      name
      lat
      long
    }
    intents {
      name
    }
```

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

```
    proposals {
      name
    }
    inventoriedEconomicResources {
      name
    }
    economic_events {
      name
    }
  }}
```

Display the activities on the local instance (paginated)

Contains all activities from the instance.

```
  {instance {
    description
    outbox(after: [id], before: [id], limit: Number) { # <- pagination
      edges {
        user {
          icon {
            url
          }
          name
          image {
            url
          }
        }
        verb
        context {
          # it can be an intent, an economic event, a social activity, etc
          __typename
          ...on Intent {
            name
            canonicalUrl
          }
           ...on Comment {
            content
          }
```

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

```
          }
        }
        totalCount
        pageInfo {
          hasNextPage
          hasPreviousPage
          startCursor
          endCursor
        }
      }
    }}
```

Display a user's activities (paginated)

```
    {user(id: String!) {
        outbox(after: [id], before: [id], limit: Number) {
          edges {
            user {
              icon
              name
              image
            }
            verb
            context {
              __typename
              ...on Intent {
              name
              canonicalUrl
            }
             ...on Comment {
              content
            }
            }
          }
          totalCount
          pageInfo
        }
      }
    }
```

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

Display current user’s feed (paginated)

Contains activities from people, agents or groups the current user is following.

```
{me {
user(id: String!) {
    inbox(after: [id], before: [id], limit: Number) {
      edges {
        user {
          icon
          name
          image
        }
        verb
        context {
          __typename
          ...on Intent {
          name
          canonicalUrl
        }
         ...on Comment {
          content
        }
        }
      }
      totalCount
      pageInfo
    }
  }
}}
```

Display all users (paginated)

```
{peoplePages(limit: 10) {
  edges {
    id
    name
    displayUsername
    image
    canonicalUrl
```

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

```
  }
}}

{organizationsPages(limit: 10) {
  edges {
    id
    name
    displayUsername
    image
    canonicalUrl
  }
}}
```

Define the network sentiment

Phase 1. Populate the network with activities

Create a location

Populate the network with a set of meaningful locations, so that they will be filterable and associated with specific events.

```
mutation {
  createSpatialThing(
    spatialThing: {
    alt: Float,
    lat: Float,
    long: Float,
    mappableAddress: String,
    name: String!,
    note: String
  }) {
    spatialThing {
      id
      name
      lat
      long
      long
      mappableAddress
```

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

```
          note
        }
      }
    }
```

Create a unit

Create the units that will be used within the network

```
    mutation {
      createUnit(
        unit: {
          label: String!,
          symbol: String!
        }
      ) {
        unit {
          id
          label
          symbol
        }
      }
    }
```

Record an offer (inventory entry)

```
    mutation {
      createOffer(intent: {
          action:String!,
          name: String,
          note: String
          atLocation: ID,
          availableQuantity: {hasUnit: ID!, hasNumericalValue: String!},
          due: DateTime,
          image: {url: String},
          tags: [ID!]
        }
      ) {
        intent {
```

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

```
        id
        name
        note
      }
    }
  }
```

Record a need (wishlist entry)

```
  mutation {
    createNeed(intent: {
      action:String!,
      name: String,
      note: String
      atLocation: ID,
      availableQuantity: {hasUnit: String!, hasNumericalValue: String!},
      due: DateTime,
      image: {url: String},
      tags: [ID!]
    }
    ) {
      intent {
        id
        name
        note
      }
    }
  }
```

Publish an offer or a need (proposal)

1. Publish a proposal

```
  mutation {
    createProposal(
      proposal: {
        hasEnd: Date,
        inScopeOf: ID,
      }
    ) {
```

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

```
      proposal {
        id
      }
    }
  }
```

2. Link the offer or need (intent) to the proposal

```
  mutation {
    proposeIntent(
          publishes: ID!  # the (Intent) to include as part of the proposal
      publishedIn: ID! # the (Proposal) to include the intent in
      reciprocal: Boolean # true if this is a reciprocal intent of this propos
  al, not primary (eg. what amount of currency is offered in exchange for the
  proposed need)
    ) {
          proposedIntent {
        id
        publishes {
          name
        }
        publishedIn {
          name
        }
      }
    }
  }
```

Publish a proposal (by itself)

```
  mutation {
    createProposal(
      proposal: {
        name: String,
        note: String,
        eligibleLocation: ID,
        hasEnd: Date,
        inScopeOf:ID
      }
    ) {
```

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

```
      proposal {
        id,
        name,
        note
      }
    }
  }
```

Create a process

An activity that changes inputs into outputs. It could transform or transport economic resource(s).

```
  mutation {
    createProcess(
          process: {
        name: String,
        note: String,
        hasEnd: DateTime,
        hasBeginning: DateTime
      }
    ) {
          process {
        id
        name
        note
      }
    }
  }
```

Conduct an economic event that results in a completely new resource

```
  mutation {
    createEconomicEvent(
          event: {
        note: String,
        action: String!, # eg. produce or raise
        provider: ID!,
        receiver: ID!,
        resourceQuantity: {hasUnit: ID!, hasNumericalValue: Float!}
```

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

```
    },
    newInventoriedResource: { # details of the new resource
      name: String
    }
  ) {
        economicEvent {
      id
      note
      receiver {
        id
        name
        note
      }
      provider {
        id
        name
        note
      }
      resourceQuantity {
        hasNumericalValue
        hasUnit {
          label
          symbol
        }
      }
      resourceInventoriedAs { # the newly created resource
        id
        name
        onhandQuantity {
          hasNumericalValue
          hasUnit {
            label
            symbol
          }
        }
        accountingQuantity {
          hasNumericalValue
          hasUnit {
```

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

```
              label
              symbol
            }
          }
        }
      }
    }
  }
```

Conduct an economic event on an existing resource

```
mutation {
  createEconomicEvent(
        event: {
      note: String,
      action: String!, # eg. consume
      resourceInventoriedAs: ID!, # what resource to act on
      provider: ID!,
      receiver: ID!,
      resourceQuantity: {hasUnit: ID!, hasNumericalValue: Float!} # how much
of the resource is affected
    }
  ) {
        economicEvent {
      id
      note
      resourceInventoriedAs { # updated details of the existing resource
        id
        name
        onhandQuantity {
          hasNumericalValue
          hasUnit {
            label
            symbol
          }
        }
        accountingQuantity {
          hasNumericalValue
          hasUnit {
```

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

```
              label
              symbol
            }
          }
        }
      }
    }
  }
```

Conduct an economic event between an existing resource and a new resource

```
mutation {
  createEconomicEvent(
        event: {
      note: String,
      action: String!, # eg. transfer
      provider: ID!,
      receiver: ID!,
      resourceInventoriedAs: ID!, # what resource are we transfering
      resourceQuantity: {hasUnit: ID!, hasNumericalValue: Float!} # how much
of the resource to transfer
    },
    newInventoriedResource: { # details of the transfered (part of the) reso
urce
      name: String
    }
  ) {
        economicEvent {
      id
      note
      resourceInventoriedAs { # updated details of the existing resource
        id
        name
        onhandQuantity {
          hasNumericalValue
          hasUnit {
            label
            symbol
          }
```

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

```
          }
          accountingQuantity {
            hasNumericalValue
            hasUnit {
              label
              symbol
            }
          }
        }
      }
      resourceInventoriedAs { # details of the new transfered resource
          id
          name
          onhandQuantity {
            hasNumericalValue
            hasUnit {
              label
              symbol
            }
          }
          accountingQuantity {
            hasNumericalValue
            hasUnit {
              label
              symbol
            }
          }
        }
      }
    }
  }
```

Start a discussion upon an offer or a need

```
  mutation {
    createThread(
      contextId: ID # what are we commenting on (eg a proposal) or in what con
  text are we starting a discussion (eg a community)
      comment: {
```

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

```
        content: String!
      }
    }) {
      id
      creator {
        name
        id
        likerCount
      }
      name
      content
    }
  }
```

Phase 2. Search and filter aggregated data

Look for proposals at a specific place

```
  {
    proposalsFiltered(atLocation: ID) {
      id
      name
      note
      image
    }
  }
```

Look for proposals near a specific location

```
  {
    proposalsFiltered(geolocation: {
      nearAddress: String, # eg. Place de la Republique, Paris, France
      distance: {meters: 5000}
    }) {
      id
      name
      note
      image
```

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

```
  }
}
```

Look for proposals from a specific agent

```
{
  proposalsFiltered(agent: [ID]) {
    id
    name
    note
    image
  }
}
```

Look for proposals with a specific scope (eg. a community)

```
{
  proposalsFiltered(inScopeOf: [ID]) {
    id
    name
    note
    image
  }
}
```

Look for all proposals tagged with a specific category / taxonomy

```
{
  proposalsFiltered(tagIds: [ID]) {
    id
    name
    note
    image
  }
}
```

Look for all EconomicEvents with a specific scope

```
{
  economicEventsFiltered(in_scope_of: [ID!]) {
```

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

```
      id
      provider {
        id
        name
      }
      receiver {
        id
        name
      }
      action
    }
  }
```

Look for all EconomicEvents of a specific agent

```
  {
    economicEventsFiltered(agent: ID) {
      id
      provider {
        id
        name
      }
      receiver {
        id
        name
      }
      action
    }
  }
```

Generate a material passport

To generate a material passport and explore its format is possible to use directly the Zenroom crypto component as described in the implementation section of deliverable 2.3 about REFLOW cryptographic scheme.

The modeling of the underlying crypto is not final and needs to be adapted to particular use-cases with a degree of tailoring which we facilitate through the provision of an Integrated Development Environment (IDE) publicly available at Beta stage on https://apiroom.net.

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

It is possible to run locally and tweak the material passport generation and verification by downloading the software from https://zenroom.org and installing it so that the zenroom executable can be called from the system (available for all operating systems: Linux, Windows, Apple/OSX as well for ARM platforms such as Raspberry Pi 0/1/2/3/4). Then execute the following commands:

```
git clone https://github.com/dyne/zenroom

cd test/zencode_reflow

./run.sh

./material_pass.sh
```

This sequence of commands will run scripts that simulate a cryptographic generation and validation of a material passport from simple sequences of event/resource/processes all available as JSON formatted files in the same directory.

The core cryptographic material inside a material passport is the *seal* component found in these data structures, which can be generated, aggregated and verified by anyone knowing the issuer (the public key of the node) without need to know who are the actors involved. Here an example schema of seal, with length of each field expressed in bytes of binary material, encoding may vary this value (default is base64).

```
{
  "seal1": {
    "SM": 49 bytes,
    "fingerprints": (array) [
      49 bytes,
     (49 bytes) ],
    "identity": 49 bytes,
    "verifier": 192 bytes
} }
```

The seal constitutes an aggregable signature of the field *reflow_identity* which is a particular hash (hash-to-point on an elliptic curve) of the whole data structure being signed, which is aggregable to other *reflow_identity* fields connected to the object by its track-and-trace.

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

Manage your inventory

Filter resources by owner

```
{
  economicResourcesFiltered(agent:[ID]) {
    name
    note
    id
    image
    accountingQuantity {
      hasUnit {
        label
      }
      hasNumericalValue
    }
    currentLocation {
      name
      lat
      long
    }
    tags

  }
}
```

Filter resources by current location

```
{
  economicResourcesFiltered(currentLocation:[ID]) {
    name
    note
    id
    image
    accountingQuantity {
      hasUnit {
        label
      }
      hasNumericalValue
```

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

```
        }
        currentLocation {
          name
          lat
          long
        }
        tags

      }
    }
```

Filter resources by tags

```
    {
      economicResourcesFiltered(tags:[ID]) {
        name
        note
        id
        image
        accountingQuantity {
          hasUnit {
            label
          }
          hasNumericalValue
        }
        currentLocation {
          name
          lat
          long
        }
        tags
      }
    }
```

Track back from an EconomicResource

Currently is only possible to go one level back in the chain. It will always return an EconomicEvent, when tracing back from an EconomicResource.

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

```
{
  economicResource(id: "ID") {
    id
    trace {
      id
    }
  }
}
```

Track back from an EconomicEvent

Currently is only possible to go one level back in the chain. Tracing back from an EconomicEvent, it may return a Process or an EconomicResource

```
{
  economicEvent(id: "ID") {
    id
    trace {
      __typename
      ... on Process {
        id
      }
      ... on EconomicResource {
        id
      }
    }
  }
}
```

Track back from a Process

Currently is only possible to go one level back in the chain. Tracing back from a Process, it will return an EconomicEvent

```
{
  Process(id: "ID") {
    id
    trace {
      __typename
```

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

```
         id
      }
   }
}
```

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

Deployment

This section illustrates how to configure, install and run a ReflowOS instance on premises.

The main reference for the deployment of ReflowOS is the repository at https://github.com/dyne/reflow-docs.

In order to get started, clone the repository:

```
    git clone https://github.com/dyne/reflow-os
```

Then enter the directory *reflow-os* and execute all following commands from a terminal, making sure the latest version of Docker and docker-compose script are installed on the machine and executable by the current user.

Then to download the whole ReflowOS simply type:

```
    make pull
```

Components overview

Server backend

Reflow server is the back-end software for the ReflowOS.

It is based on Bonfire, a federated free software app ecosystem.

The Reflow server implementation language is Elixir (running on Erlang/OTP), a language designed for building highly reliable and scalable systems. It uses the Phoenix web framework and the Absinthe GraphQL Toolkit to deliver a GraphQL API which the frontend interacts with. It uses ActivityPub to deliver server-to-server federation.

Client frontend

Reflow client is the front-end software for the Reflow project, which connects to the API of Reflow server; it is based on web technologies so you can use it with any modern browser (desktop or mobile). The client is written in Elixir.

Cloud container

The ReflowOS cloud container is designed to connect a distributed cloud of micro-services through a private and anonymous peer-to-peer network cluster, enforcing privacy by design principles and lowering the liability of network and service providers.

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

Smart contracts

The Reflow crypto smart-contracts grant the integrity and portability of authentication functions and the interchange of material passports. They provide a layer of blockchain integration for zero-knowledge authentication and are documented in depth by the Reflow D2.3 deliverable.

System requirements

Server

For best performance, stability and functionality we have documented some recommendations for running a ReflowOS server.

You can either use Docker on any server that supports it, and the docker-compose file provided will manage the OS, database, and app/web servers for you, or you can deploy manually with the following:

Platform	Options
Operating System	Debian 10+ (Buster), Alpine Linux 3+ (recommended), Ubuntu 20.04+ LTS, Red Hat Enterprise Linux 8+, SUSE Linux Enterprise Server 15+, openSUSE Leap 42.1+, or CentOS 8+
Database	Postgres 12+ with Postgis extension
App Server	Erlang / Elixir
Search index	Meili
Web proxy	Nginx or Caddyserver

Memory

Memory requirements for running a ReflowOS server are greatly variable, depending on the numbers of users, apps, files and volume of server activity.

ReflowOS runs fine with a minimum of 512MB RAM, but we recommend at least 2GB.

Web browser

For the best experience with the ReflowOS web interface, we recommend that you use the latest and supported version of a browser from this list, or one based on those:

- Microsoft Edge

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

- Mozilla Firefox
- Google Chrome/Chromium
- Apple Safari

Configuration

These instructions are for setting up ReflowOS in production.

Configure public settings

Configure the settings inside *conf/bonfire-public.env* and in particular:

```
# name of the application instance

APP_NAME=reflow

# configure the mail server to use (smtp auth)

MAIL_BACKEND=smtp

MAIL_DOMAIN=smtp.sendgrid.net

MAIL_PORT=587

# a name and tagline for your instance

INSTANCE_DESCRIPTION="Reflow node for valueflows circular economy networks"
```

Configure secret settings

Configure the settings inside *conf/bonfire-gensecrets.sh* and in particular:

```
# smtp auth configuration

MAIL_FROM=no-reply@dyne.org

MAIL_SERVER=smtp.sendgrid.net

MAIL_USER=apikey

onthology MAIL_PASSWORD=passport
```

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

Generate site configurations

Once the above settings are correctly configured (check with your email provider and system administrator) then generate the final setup for this installation by running:

```
make config
```

Beware that this command should be **run only once** in the first configuration phase and should not be run again because it will overwrite certain generated secrets and it may make the current installation unusable.

Run

To start ReflowOS in foreground (with an axtive elixir console) use the command:

```
make run
```

This command should be used on the first run, when it is necessary then to issue the console command:

```
Bonfire.Repo.ReleaseTasks.migrate
```

After this command the installation should be ready and listening on port 4000 (TCP) so it can be accessed on http://localhost:4000/.

In order to start interacting within the network, you will need to register and create an account, exactly what you would do in all the today available networks. The first account registered will also be the node administrator.

Once you are registered and logged, you will be able to set your own profile, or act on behalf of one or more organizations.

Glossary

ActivityPub - a standard decentralized social networking protocol based on ActivityStreams.

ActivityStreams - a standard format for syndicating social activities.

API - Application Programming Interface. A set of definitions, protocols, and tools for building application software, to enable communication between various components.

Canonical - the main or reference location for something (eg. canonical URL)

Data handling module - a database abstraction library, using an ORM approach, which would have added support for caching and federation (when applicable).

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

Decentralisation - the process by which the activities of a network are distributed or delegated away from a central, authoritative location or group.

Federation - the ability for decentralized systems to send data to one another in a standardised way, to prevent fragmentation of the network. Email is a great example of a federated system.

GNU AGPL - a free, copyleft license published by the Free Software Foundation, and based on version 3 of the GNU General Public License (GPL) and the Affero General Public License. This license is compatible with the GPL and is recommended for any software that will commonly be run over a network.

GNU GPL - a widely used free software license guaranteeing end users the freedom to run, study, share and modify the libre software.

Libre software - software that is distributed under terms that allow users to run the software for any purpose as well as to study, change, and distribute it (and any adapted versions).

Metadata - data that provides information about other data. For educational resources, it could describe various aspects of the material, including grade level, subject area, and content type or format.

Instance - an independently-hosted version of ReflowOS.

Node - a member of a decentralized network, which can sometimes serve as client, sometimes as server. See also Instance.

Open Source - a decentralized software-development model that encourages open collaboration, with products such as source code, blueprints, and documentation freely available to the public.

ORM - Object-relational mapping, used to create a 'virtual object database' which can then be referenced programmatically (with less code and more flexibility as to the underlying data storage).

Protocol - a defined set of rules and formats that determine how data is transmitted.

SaaS - Software as a Service. Web-based software that's centrally hosted and made available to use (free or by subscription).

Scaling horizontally - implies adding more nodes to a system to support its growth, for example installing copies of the same software on three web servers instead of just one.

Scaling vertically - implies adding resources (like CPUs or memory) to a single server in a system to support its growth.

Search index - a component of search engines which collects, parses, and stores data to facilitate fast and accurate search and information retrieval. Index design incorporates interdisciplinary concepts from linguistics, cognitive psychology, mathematics, informatics, and computer science.

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

Standard - a collection of agreed specifications, usually organised by a standards body such as the W3C.

Intent - A planned economic flow which has not been committed to, which can lead to economic events (sometimes through commitments).

Proposal - Published requests or offers, sometimes with what is expected in return.

Action - An action verb defining the kind of event, commitment, or intent.

Economic Event - An observed economic flow, as opposed to a flow planned to happen in the future. This could reflect a change in the quantity of an economic resource. It is also defined by its behavior in relation to the economic resource (see *Action*)

Economic Resource - A resource which is useful to people or the ecosystem.

Resource Specification -Specification of a kind of resource. Could define a material item, service, digital item, currency account, etc. Used instead of a classification when more information is needed, particularly for recipes.

Process - An activity that changes inputs into outputs. It could transform or transport economic resource(s).

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