Institutions: DYNE.org
Authors Names: Mihaela Tudorache (DYNE), Adam Burns (DYNE)
Publication Date: 30/03/2023

BUILDING THE DIGITAL INFRASTRUCTURE FOR FAB CITIES, REGIONS AND NATIONS

D3.2b Technology Report

INTERFACER

Funded by

EUROPÄISCHE UNION
Europäischer Fonds
für regionale Entwicklung

Hamburg
Behörde für Wirtschaft
und Innovation

Consortium Partners

HELMUT SCHMIDT UNIVERSITÄT
Universität der Bundeswehr Hamburg

FABCITY Hamburg

HIWW

dyne.org

interfacerproject.eu

Table of Contents

1. Introduction and scope of the document

This report is made to outline the information regarding the software stack developed by the different partners during the Interfacer project.

With the mission to reorganise globalised value creation and production systems and to establish sustainable and synergetic production processes worldwide, we concentrated our efforts into developing an initial application and distribution of an open source software system, modular and highly customizable, that each Fab City, Fab Lab and like-minded organisations, can and adopt, activate and configure based on their own needs.

The tackled challenges:

- accelerating the growth of the network by facilitating the onboarding process
- ensuring compatibility, coherence and scalability by promoting the same data exchange standards, protocols, principles and frameworks
- reducing the needed technical knowledge
- offering community support and further development along the way

Following, the scope, the technical stack, use cases and integration with other components (when available), the challenges, the public repositories, deployment and playground (when available), for each technical component.

2. Technical components

2.1 Fab City OS Installer by FCHH

Scope

As the starting point component, the Installer assists in managing each software component deployment lifecycle. It incorporates a wizard to ease (de)installation and deployment management of components and provides a substrate to test interactions such as multi-city scenarios, backup and restore scenarios, and backwards compatibility.

Technology stack

The development is based on Kubernetes platform[1], an industry standard for cloud native applications. It can run on a cloud hosted environment or on-premises, as a cluster and even on a single board, low resource demanding, Raspberry Pi.
In order to facilitate the handling of the Kubernetes resources, the installer uses the Helm charts as a packaging format. A chart is a collection of files that describe a related set of resources, and in this case has a Letsencrypt Certificate Authority and a Proxy/Load-Balancer included.

The included modules

Fab City OS Core, Fab City OS Suite, Gitea as a self-hosted distributed version control system, Matrix as a communication channel tool and Pretix as an event ticket manager.

The challenges

The component has a high complexity, having to deal with different modules and development teams and at the same time, provide a generic version potentially used by new institutions. Furthermore, in order to function properly, it needs to stay aligned with all the environment parameters and updates, and therefore it needs to integrate the stable versions of the included modules. For these reasons, given that the prototyping phase lasted throughout the entire period of the Interfecer project, the components were continually updated, making the synchronisation with the Installer hard and causing a great impact on the testing and feedback of this component.

Another challenge was serving the network of Fab Cities, which are very different entities from each other and most of them still in the definition phase [FabCity Network]. At the moment, the movement is concentrating mostly on common action plans and communication strategy,

[1] https://kubernetes.io/

therefore further effort will be needed in order to identify the technical requirements and to support such a dynamic context.

Public repositories

- https://gitlab.fabcity.hamburg/software/interfacer-swkit
- https://gitlab.fabcity.hamburg/software/fcoscore-chart

2.2 Fab City OS Core by Dyne.org

Scope

The component refers to a collection of tools designed to promote, assist and incentivize distributed and collaborative practices, offering an implementation of fair and equitable compensation schemes, throughout the whole design and manufacturing lifecycle.
It leverages crypto technology to empower participants with privacy, transparency and data ownership. Most importantly, the lifecycle of each asset can be expressed in a blockchain-anchored Digital Product Passport, enabling the verification of the asset's history on the platform or by third parties.

Technology stack

Zenflows is a modular server that leverages commons-based peer production by documenting and monitoring the life cycle of products. Its function is to enable a federated network of organisations to bundle, systematise and share data, including information and knowledge about designs, services and physical artefacts.
Most validations are provable (and sealed by means of cryptography) using Zenroom as the core crypto component, including support for advanced Digital Product Passport schemes like REFLOW Portable Crypto Functions.
The Valueflows vocabulary is adopted by Zenflows to govern a Resource Event Agent accounting model and facilitate the development and execution of business logics built on top of it.
The PosgreSQL database backend is used as battle-tested data-storage solution well compatible with established enterprise standards in system administration, on top of it a GraphQL interface is developed based on the well performant Elixir language framework.

W3C DID DID:DYNE, the World Wide Web Consortium (W3C) Distributed Identifiers (DIDs) is a standard that enables verifiable, decentralised digital identities for any subject (e.g., a person, organisation, thing, or data model). The Dyne.org Foundation has registered a specific schema with the W3C standards body and implemented its software controller to support the Interfacer project.

The W3C DID implementation provides a core building block for federated systems by allowing for interoperability and seamless data migration between components, including digital product passports of digital twins. The system offers a way to identify and authenticate any entity in the FabCity federated network, regardless of where they operate within the federated system or even outside of it, by using decentralised and distributed identifiers. It allows for each component's independent and autonomous operation while enabling communication and data sharing.

Interfacer gateway is a public Internet gateway service that acts as a proxy to route information flows to each of the back-end services, enabling a secure, modular, and scalable deployment Architecture.

Zenswarm-storage is a distributed caching and sharding service offering high availability of data: its nodes are used by applications to store arbitrarily large data objects pegged to a blockchain without overloading ledger or blockchain nodes.

Zenflows-inbox is a modular service that provides incoming and outgoing messages to allow updates to be processed from a distributed and federated network. It supports the W3C ActivityPub standard for decentralisation.

Zenflows-proxy is an HTTP proxy, serving the front end of the project with unified interactions

with the (micro) services implemented, like Zenflows back-end instance, Zenflows-inbox instance, zenflows-wallet instance, Location Auto-completion and Location Lookup provided by the external Here API.

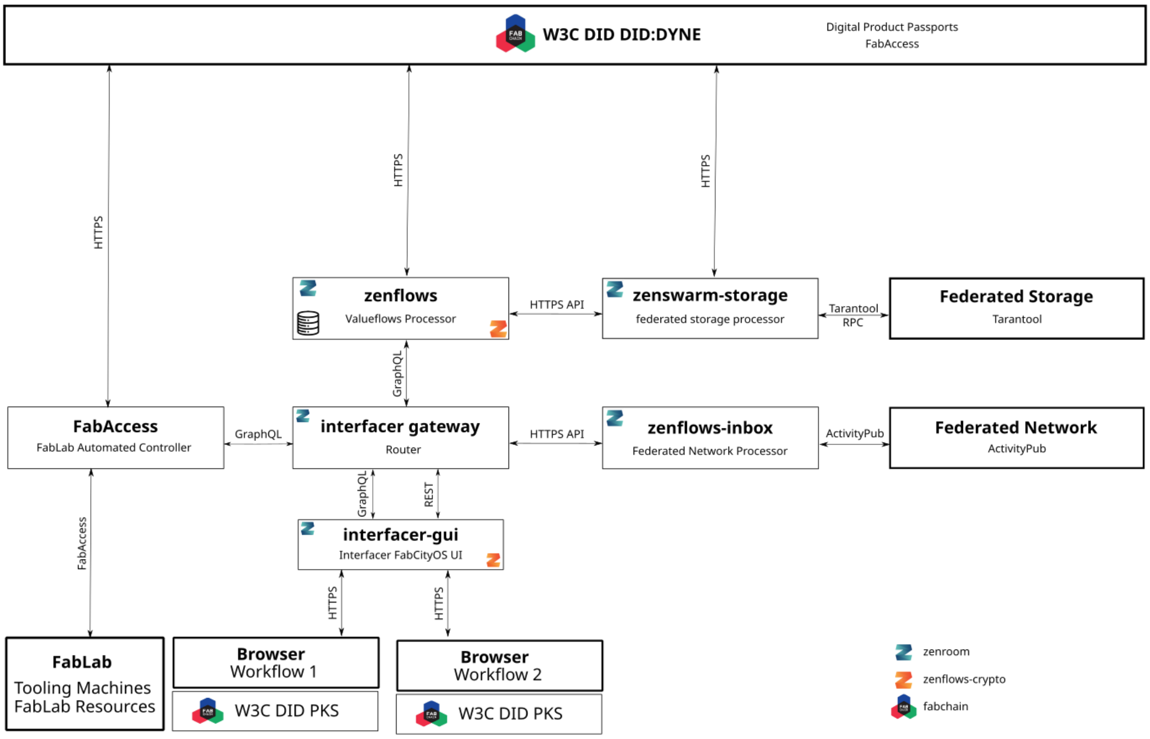


Figure 1. Architecture of Fab City OS Core

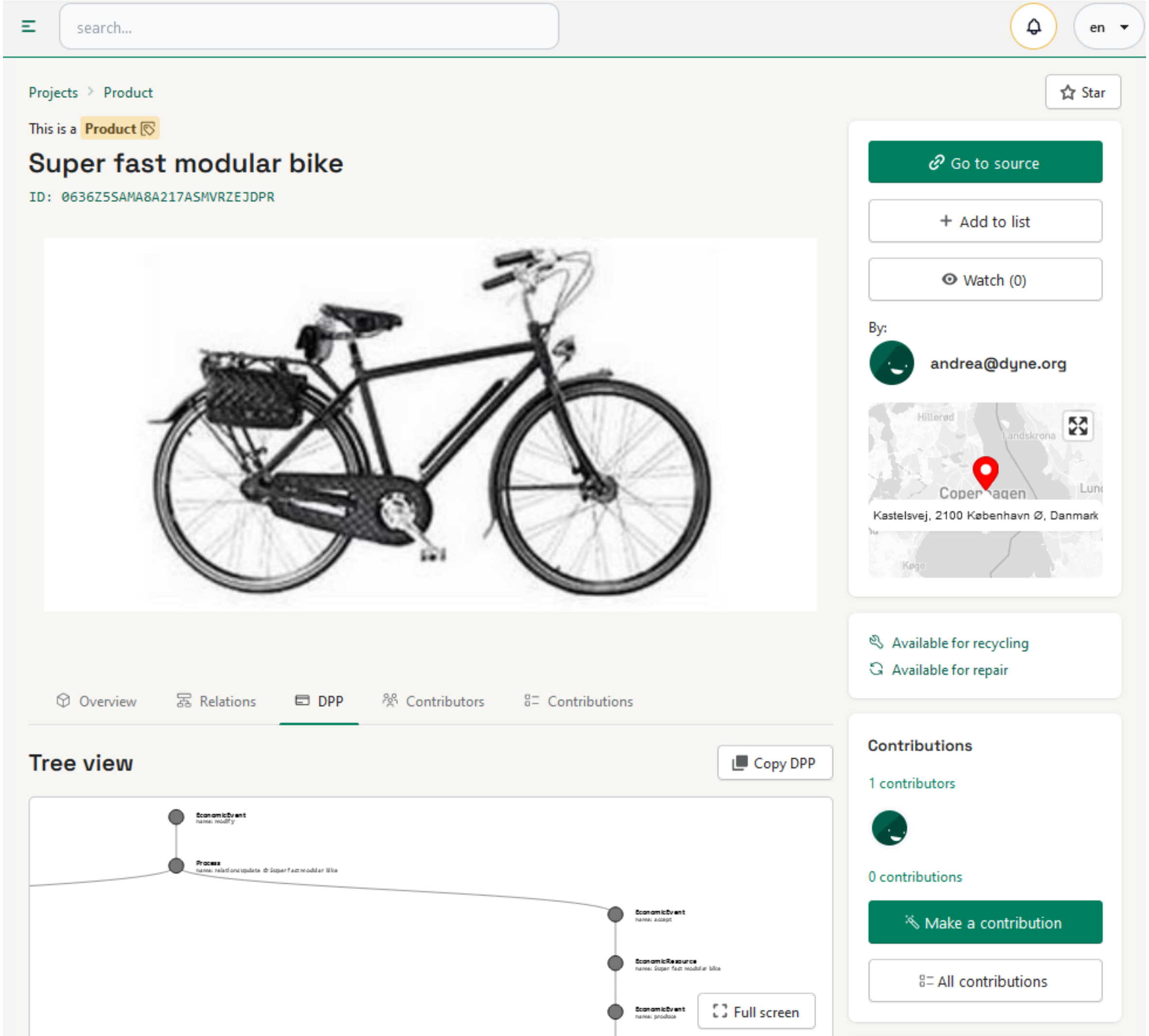
Interfacer-gui is a Progressive Web Application (PWA) that is run in-browser and acts as Graphical User Interface (GUI) and a crypto-wallet (W3C DID PKS or Personal Key Store).
The platform is the central point where all services are bundled together, enabling designers, makers and innovators to find and share open source hardware projects, collaborate with others and discover new products and services.

The two main pillars that move the collaboration are the **Digital Product Passport (DPP**) and the economic model proposal, called **Creative Flows**.

The **DPP** is the cryptographically signed trace of project activity, containing all the resources, locations and contributions through the whole supply-chain.

Mapping the supply chain enables a series of opportunities in terms of user reputation, sustainability indicators, data collection, quality assurance and green practices reinforcement, like recycling and repairing.

Furthermore, the DPP is blockchain-anchored, enabling the verification of the data history on the platform or by third parties, enabling users to easily share information about their project with others.



Projects > Product

This is a Product

Super fast modular bike

ID: 0636Z5SAMA8A217ASMVRZEJDPR

By: andrea@dyne.org

Kastelsvej, 2100 København Ø, Danmark

Available for recycling

Available for repair

Contributions

1 contributors

0 contributions

Tree view

EconomicEvent
name: modify

Process
name: relationsupdate @ Super fast modular bike

EconomicEvent
name: accept

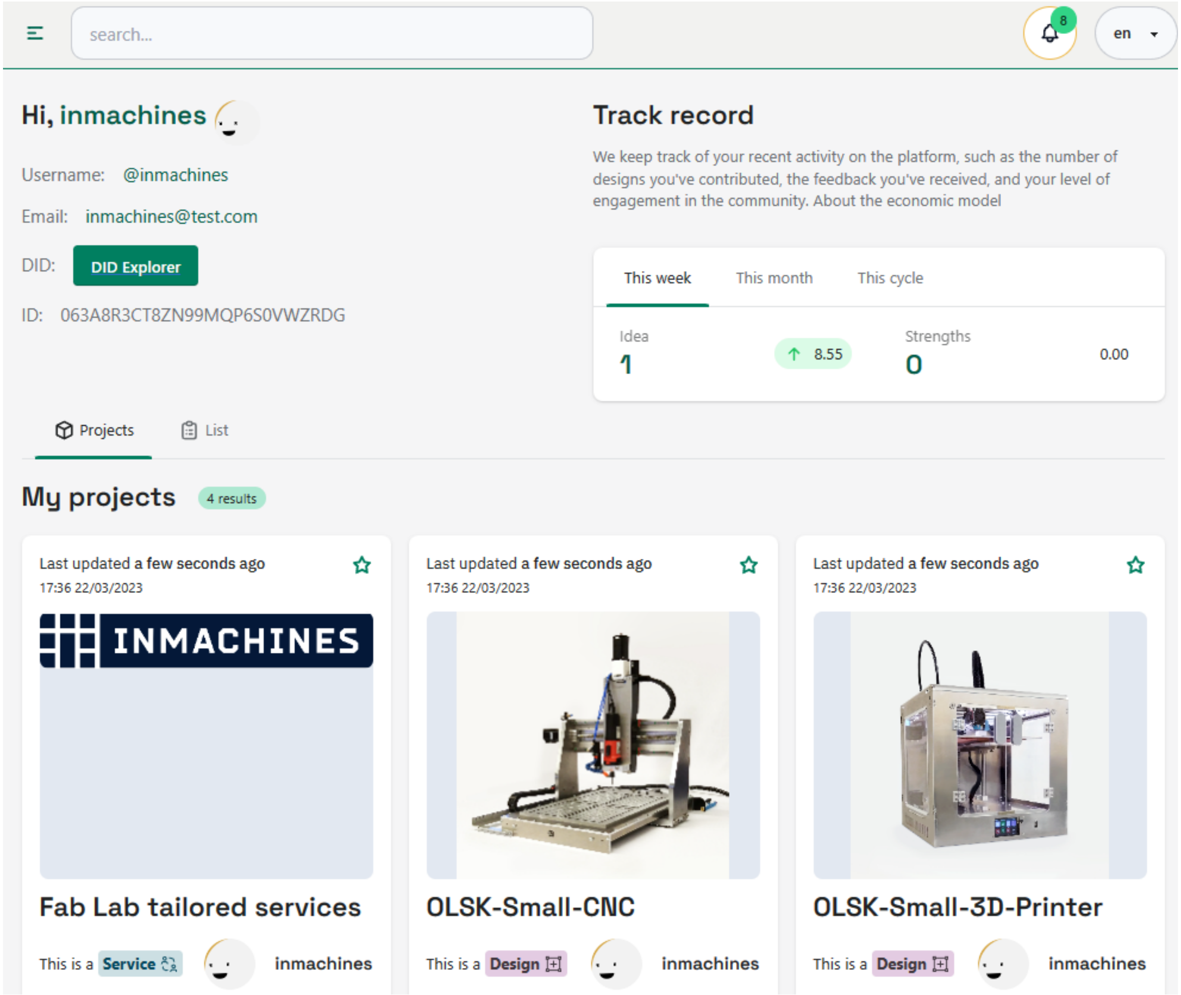
EconomicResource
name: Super fast modular bike

EconomicEvent
name: produce

Figure 2. DPP on the Interfacer GUI

Creative flows, promotes fair agreements directly between users, enabling communities to discover, remix, and improve designs and technologies, facilitating fair revenue distribution based on contributions and efforts.

The Valueflows vocabulary is used as one possible semantic organisation of interaction data. By doing so, we use “points” to the interactions between Agents, for instance, the agreed offer and provision of contributions to projects. These points map two different dimensions in Keynesian economics 💡 Idea points as Debt and 💪 Strength points as Credit.
This system does not represent its liquidity by points: they are not the money that pays for the participant’s time and fuels its economic growth. We may define 💪 strength and 💡 idea points in cryptonomics terms as “non-fungible tokens” that are “burned” using a “peg” linking them to an amount of money (or fungible tokens) at every new cycle. The pegged value will vary depending on the liquidity available to each collective.



search...

8 en

Hi, inmachines

Username: @inmachines

Email: inmachines@test.com

DID: DID Explorer

ID: 063A8R3CT8ZN99MQP6S0VWZRDG

Track record

We keep track of your recent activity on the platform, such as the number of designs you've contributed, the feedback you've received, and your level of engagement in the community. About the economic model

This week This month This cycle

Idea
1
↑ 8.55

Strengths
0
0.00

Projects List

My projects 4 results

Last updated a few seconds ago
17:36 22/03/2023
Fab Lab tailored services
This is a Service inmachines

Last updated a few seconds ago
17:36 22/03/2023
OLSK-Small-CNC
This is a Design inmachines

Last updated a few seconds ago
17:36 22/03/2023
OLSK-Small-3D-Printer
This is a Design inmachines

Figure 3. Track record of Ideas and Strength points on the Interfacer GUI

DPP visualisation tool is a web-based interactive environment we developed, having as a core component Jupyter Notebook. The tool supports users in Valueflows objects modelling and visualisation and also facilitates testing and verification of the relationships among assets. At the same time, it helps with the identification of the most relevant information. This enables possibilities for reducing complexity and separating the displayed level of detail, based on the user proficiency.
The tool also assists in translating the OSH assembly instructions into Valueflow events and with the validation process, supporting the very first steps towards automatization of this task.

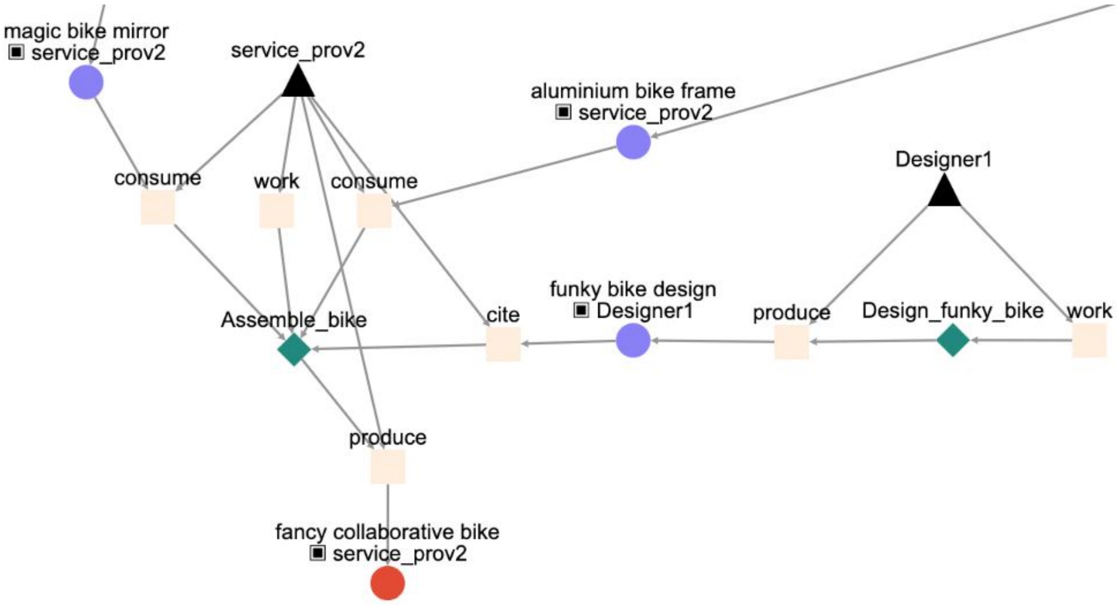


Figure 4. Trace (fragment) of "fancy collaborative bike" (at the bottom). Squares represent events, circles resources and triangles agents. The graph represents Processes (group of events) using diamond glyphs

Fab City OS Core and FabAccess integration via DID

FabAccess provides a resource access control system, enabling fab labs to deny or allow access to the machines and tools in the labs only to authorised persons. For further details about FabAccess see the dedicated section.

During the project, we developed an adapter between the Fab City OS Core platform and the FabAccess server, using the python bindings that verify and enable the user via the Dyne DID on both systems.
The adapter wraps the FabAccess API to enable it to work with the HTTP API used by FabCity OS. By proxying the commands to activate machines through the adapter, it ensures that commands are checked for signing and executed correctly on the FabAccess Backend.

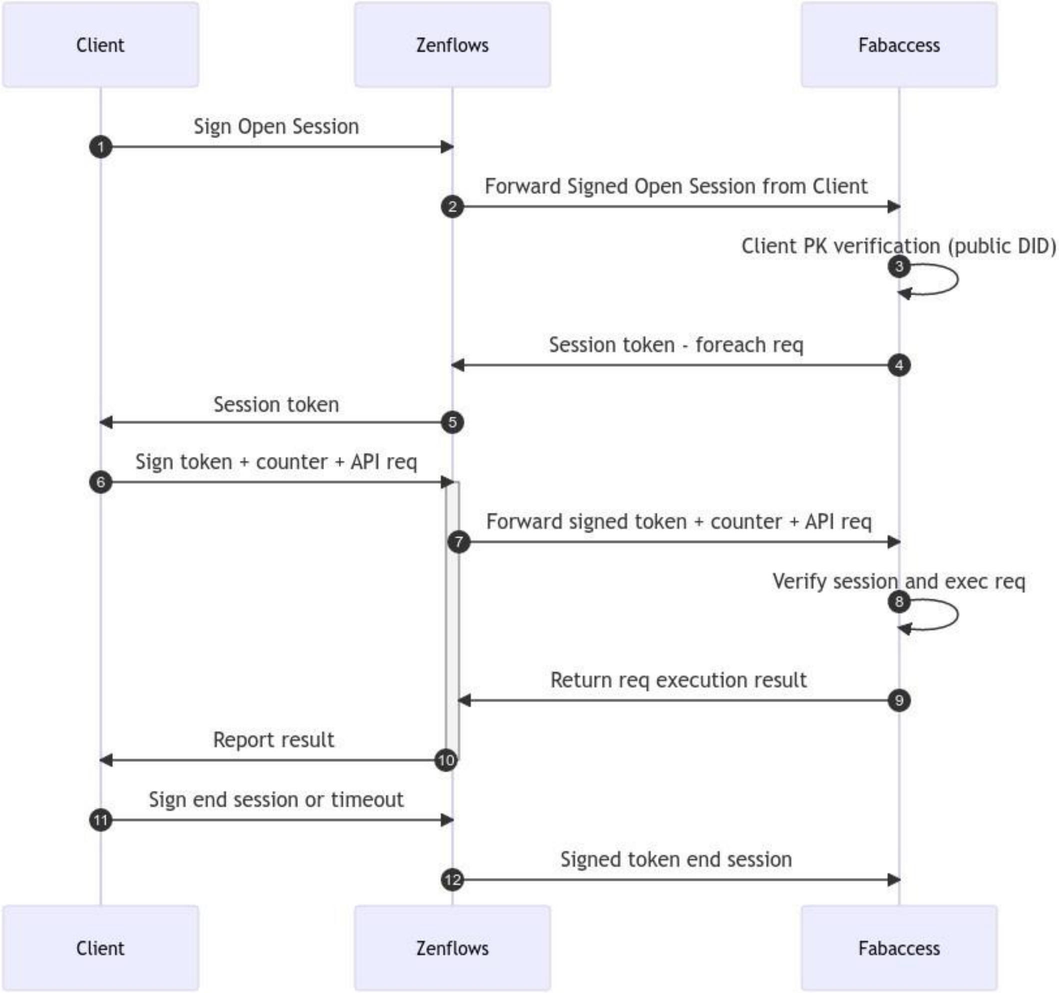

Client
Zenflows
Fabaccess
1
Sign Open Session
2
Forward Signed Open Session from Client
3
Client PK verification (public DID)
4
Session token - foreach req
5
Session token
6
Sign token + counter + API req
7
Forward signed token + counter + API req
8
Verify session and exec req
9
Return req execution result
10
Report result
11
Sign end session or timeout
12
Signed token end session
Client
Zenflows
Fabaccess

Figure 5. Fab City OS GUI & FabAccess workflow diagram

The challenges

The project involved a certain amount of innovation and therefore research and test. As we progressed with these activities, we aligned the different stakeholders, we clarified initial requirements, we involved makers and designers in the co-design process, and we identified potential frames and technical components to be used.

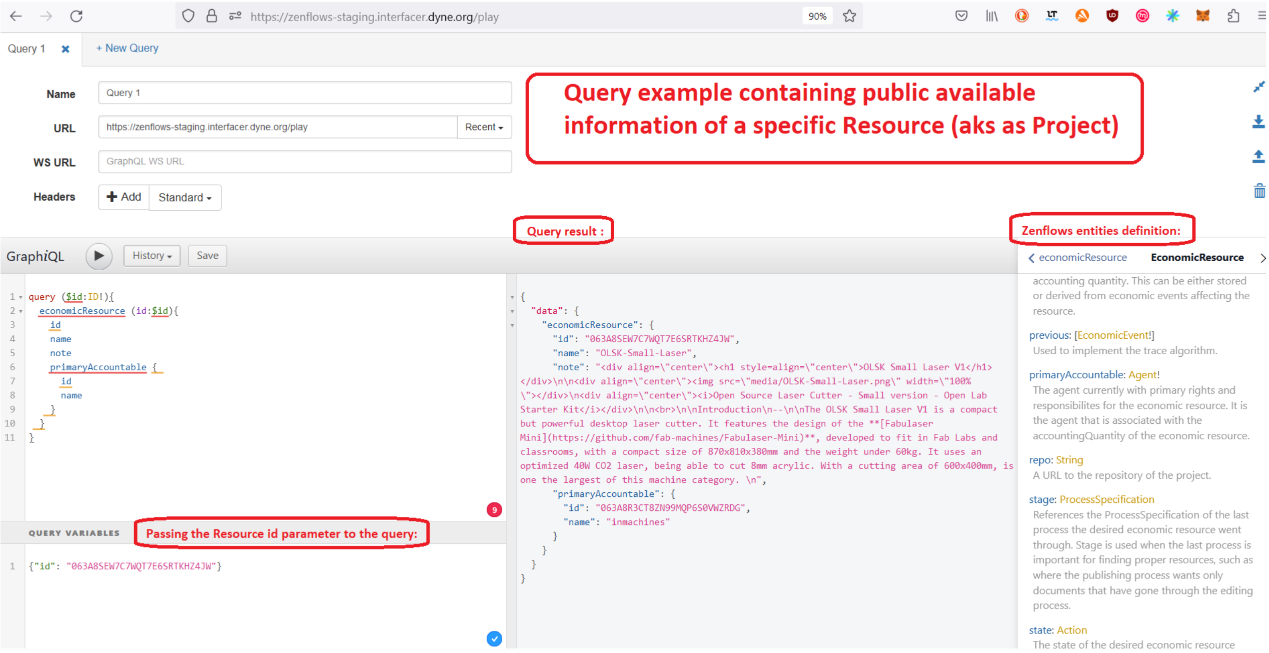
During the development process, we cyclically assessed and incorporated the feedback and our tests and based on our findings, we adjusted the development plan. For this reason, parts of the initial plan changed, this was the case of the e-commerce shop. The component had dependencies with modules under research and development like the innovative cryptographic stack and the futuristic economic model, that had to reach a stable version, before integration with other services. With the beta release, the software reaches a good level of stability and is ready for the integration with already existing e-commerce services.

Public repositories

https://github.com/interfacerproject

Beta release & API documentation & playground

- https://beta.interfacer.dyne.org/
- https://zenflows-staging.interfacer.dyne.org/play



Query example containing public available information of a specific Resource (aks as Project)

Name: Query 1

URL: https://zenflows-staging.interfacer.dyne.org/play

```
query ($id:ID!){
  economicResource (id:$id){
    id
    name
    note
    primaryAccountable {
      id
      name
    }
  }
}
```

Passing the Resource id parameter to the query:

```
{"id": "063A8SEW7C7WQT7E6SRTKHZ4JW"}
```

Query result :

```
{
  "data": {
    "economicResource": {
      "id": "063A8SEW7C7WQT7E6SRTKHZ4JW",
      "name": "OLSK-Small-Laser",
      "note": "<div align=\"center\"><h1 style=align=\"center\">OLSK Small Laser V1</h1>
</div>\n\n<div align=\"center\"><img src=\"media/OLSK-Small-Laser.png\" width=\"100%
\"></div>\n<div align=\"center\"><i>Open Source Laser Cutter - Small version - Open Lab
Starter Kit</i></div>\n\n<br>\n\nIntroduction\n--\n\nThe OLSK Small Laser V1 is a compact
but powerful desktop laser cutter. It features the design of the **[Fabulaser
Mini](https://github.com/fab-machines/Fabulaser-Mini)**, developed to fit in Fab Labs and
classrooms, with a compact size of 870x810x380mm and the weight under 60kg. It uses an
optimized 40W CO2 laser, being able to cut 8mm acrylic. With a cutting area of 600x400mm, is
one the largest of this machine category. \n",
      "primaryAccountable": {
        "id": "063A8R3CT8ZN99MQP6S0VWZRDG",
        "name": "inmachines"
      }
    }
  }
}
```

Zenflows entities definition:

economicResource | EconomicResource

accounting quantity. This can be either stored or derived from economic events affecting the resource.

previous: [EconomicEvent!]
Used to implement the trace algorithm.

primaryAccountable: Agent!
The agent currently with primary rights and responsibilites for the economic resource. It is the agent that is associated with the accountingQuantity of the economic resource.

repo: String
A URL to the repository of the project.

stage: ProcessSpecification
References the ProcessSpecification of the last process the desired economic resource went through. Stage is used when the last process is important for finding proper resources, such as where the publishing process wants only documents that have gone through the editing process.

state: Action
The state of the desired economic resource

Figure 6. Zenflows documentation screenshot and query example

2.3 Fab City OS Suite by FCHH

Scope

This component aims to provide the users (both new Fab Cities and also members/labs within a Fab City) with a simple system to set up their own website using established open source workflows and using a given, consistent Fab City design template. The solution will not lock them in and will allow them to easily move on to a different/more complex system in the future.

Technology stack

Fcos-suite-astro is based on the Astro static site generation and enables Fab Cities, Makerspaces and other institutions to get their website up and running quickly while supporting Open Source workflows. It adds to the features list a number of custom elements that take into consideration specific needs a Fab City may have, like a knowledge management hub component for publishing and sharing information and integration with a ticket event handler.

Fcos-suite-map this project contains a Map component that can be used to display relevant locations of a Fab City. It is a npm package that contains a React component, which can be embedded into other web projects. It uses Mapbox GL as an interactive mapping framework and TailwindCSS for styling. There is no backend and no admin interface – this is a simple component that displays POIs that are passed in via a single prop.

Fcos-suite-ui contains reusable UI components based on React and TailwindCSS.

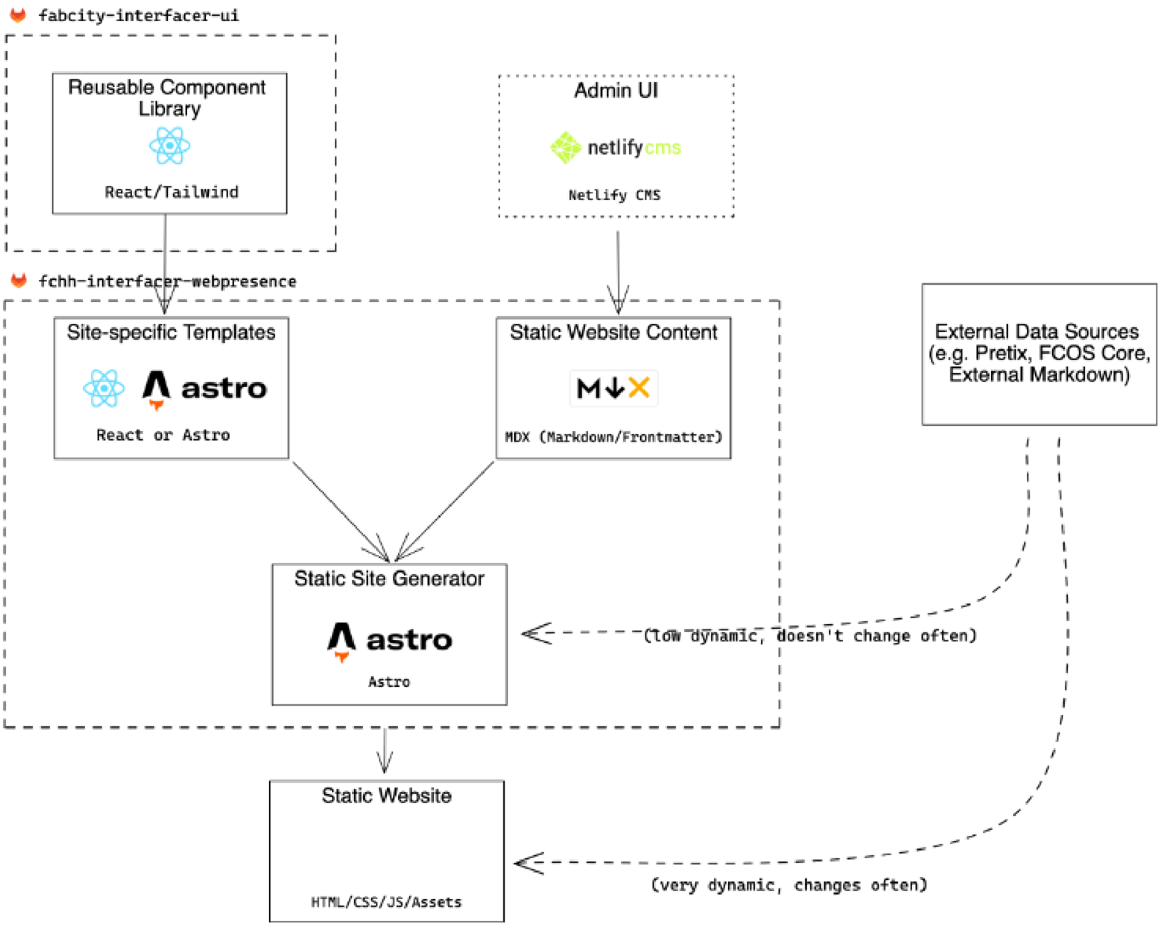
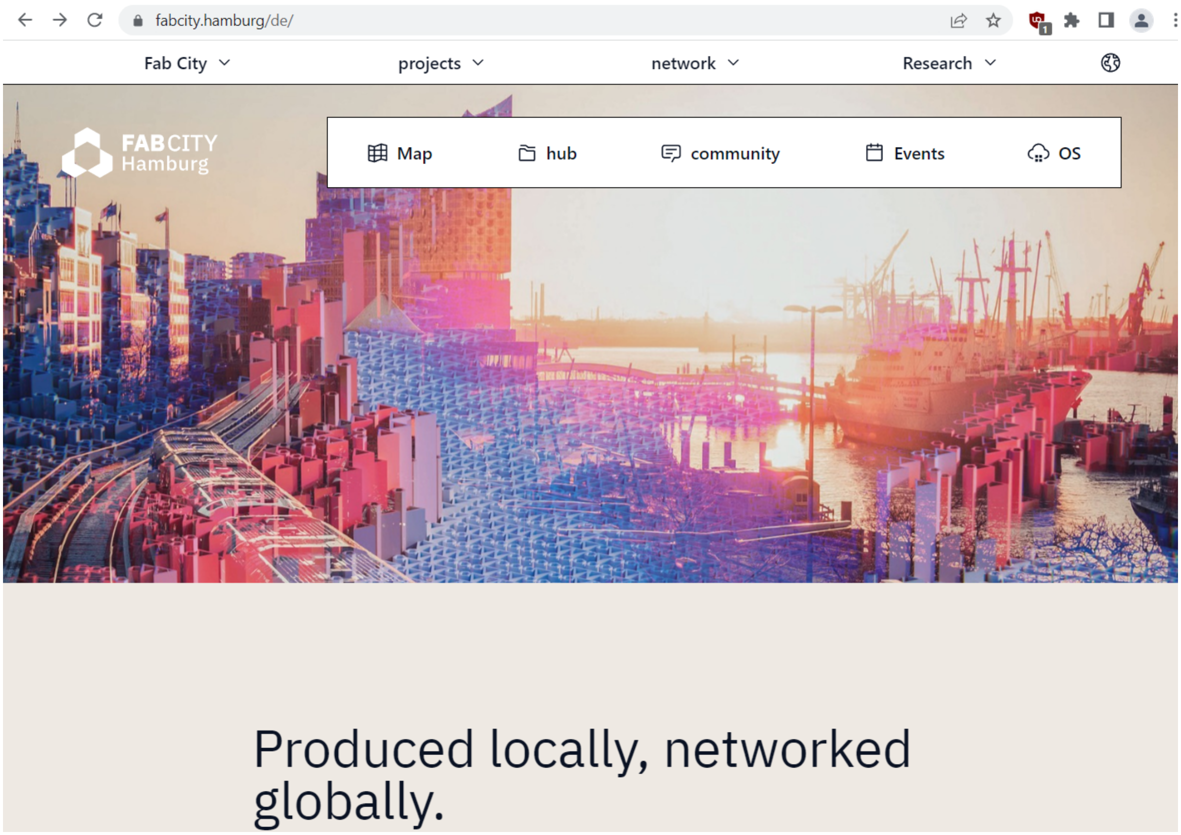

fabcity-interfacer-ui
Reusable Component Library
React/Tailwind
Admin UI
netlifycms
Netlify CMS
fchh-interfacer-webpresence
Site-specific Templates
astro
React or Astro
Static Website Content
M↓X
MDX (Markdown/Frontmatter)
External Data Sources (e.g. Pretix, FCOS Core, External Markdown)
Static Site Generator
astro
Astro
(low dynamic, doesn't change often)
Static Website
HTML/CSS/JS/Assets
(very dynamic, changes often)

Figure 7. Fab City OS Suite architecture

The Fabcity.hamburg website use-case

The Suite served as a base for the implementation and customization of the Fab City Hamburg website, providing a clear example of how it can be used as a tool for community communication both for the local members and for the global community network.
The Fab City Hamburg website has a light configuration with a CMS on top of git with Markdown (.md) text file support, is easy to maintain and to integrate with other third party open source tools.

Furthermore, it is released with an AGPL V3 licence and provides a production ready solution.



fabcity.hamburg/de/

Produced locally, networked globally.

Figure 8. FabCity Hamburg website, homepage screenshot

The challenges

In order to complete both the development of the generic template and the Hamburg website implementation in a limited amount of time, the team concentrated on putting together the whole idea at the expense of the user research. While this has proven to be a good approach for the short time period, in order for this solution to thrive, more efforts will have to be made for involving other cities as potential consumers and testers.

Public repositories

https://gitlab.fabcity.hamburg/fcos-suite

2.4 FabAccess by Fab-access.org

Scope

FabAccess is a prototype-grade software suite for managing access, mostly aimed at Makerspaces, FabLabs, and other open workshops. It is designed to allow secure access control to machines and other equipment that is dangerous or expensive to use. FabAccess makes of accessibility a core principle in terms of sw-hw components prices and the use of opened protocols and licences pushing the usage towards all other things one would like to give exclusive access to even, when they are not dangerous or expensive to use (think 3D printers, smart light bulbs, meeting rooms).

Technology stack

FabAccess uses a Client/Server architecture with:

Back-end: Diflouroborane aka BFFH, based on data interchange format and capability, based on Remote Procedure Call (RPC) system called Cap'n Proto. The solution provides authentication and access to machines via SmartCard, Audit-Log and handling of electric locks.

Client side: called Borepin, uses Xamarin platform based on the .net framework for the User Interface, facilitating portability. The solution is capable of machine identification via NTAG (as NFC Tag with Universal Compatibility, with NFC Chip), bind and unbind FabFire-Card to users and create DES-Fire Cards (as Data Encryption Standard Fast Innovative Reliable and Secure), permission handling.

Grafana implementation: for interactive data visualisation of machine states, data and power consumption.

FabAccess in the Microfactory - tfom23 show-case room

During Interfacer’s final event expo, FabAccess made the system available to all attendees by opening up the network. This allowed users to explore the system and its features by downloading the app and accessing the machines.

To make resource identification simple, all machines and lockers had a QR-Code and NFC tag that could be easily scanned using a mobile device. Moreover, to activate the machines, they installed NFC readers on each machine that could be used with a smart card.

To demonstrate the system’s use and functionality, FabAccess created three test users with different permissions. An admin user had the authority to manage all other users, a manager user had the permission to manage machines and free them if they were currently in use. Additionally, we created a maker user to demonstrate a typical user who operates the machines.

Public repositories & technical documentation

- https://gitlab.com/fabinfra/fabaccess
- https://fab-access.readthedocs.io/en/v0.3/

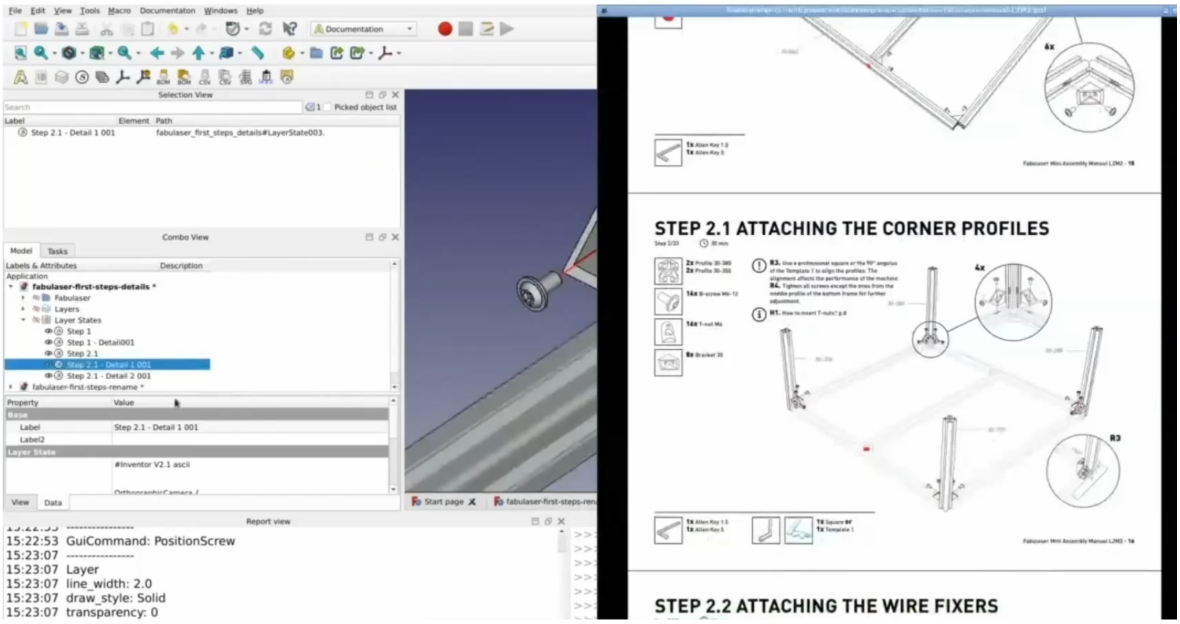
2.5 OSH Toolchain by HIWW

Scope

The OSH tools are a collection of tools that assist in two areas: the automation and standardisation of producing documentation and the creation of an initial standardised associated metadata structure required for searching and filtering Open Source Hardware projects in a consistent fashion.

The included modules

OSH Automated Documentation parses a Scribble file with a specification of an assembly manual and generates a high-quality PDF manual. This project makes use of the Base project to parse the Scribble file and provide an abstract syntax tree (AST). Within this project, the AST is further processed to finally produce a manual in the form of a PDF.



Label	Element	Path
Step 2.1 - Detail 1 001		fabulaser_first_steps_details#LayerState003.

Labels & Attributes	Description
Application	
fabulaser-first-steps-details *	
Fabulaser	
Layers	
Layer States	
Step 1	
Step 1 - Detail001	
Step 2.1	
Step 2.1 - Detail 1 001	
Step 2.1 - Detail 2 001	
fabulaser-first-steps-rename *	

Property	Value
Base	
Label	Step 2.1 - Detail 1 001
Label2	
Layer State	
	#Inventor V2.1 ascii OrthographicCamera {

```
15:22:53 GuiCommand: PositionScrew
15:23:07 ----------------
15:23:07 Layer
15:23:07 line_width: 2.0
15:23:07 draw_style: Solid
15:23:07 transparency: 0
```

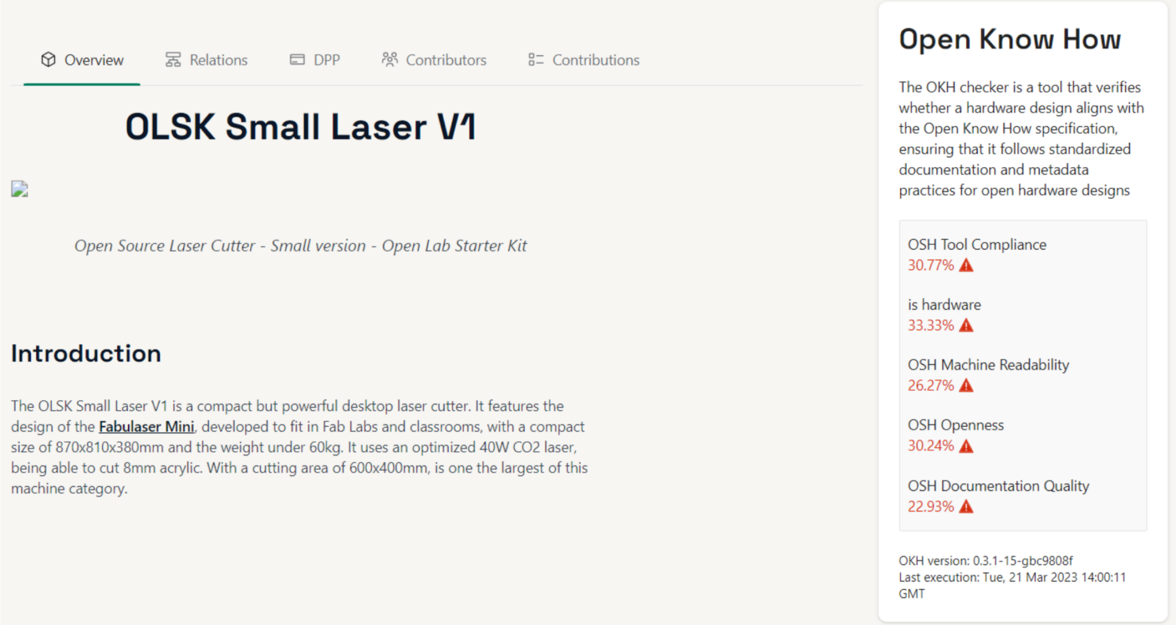
STEP 2.1 ATTACHING THE CORNER PROFILES

STEP 2.2 ATTACHING THE WIRE FIXERS

Figure 9. Automated documentation on Freecad

OSH Tool is a command line tool of Open Source Hardware (OSH) technical project linting, quality assessment based on **Open Know How** standard. This standard makes OSH compatible and comparable via meta-data, facilitating searching and filtering of already existing projects and components.

FabCity OS Core component integrates and uses the OSH tool for its Open Know How compliance report. By displaying the compliance report in a user-friendly way, and by redirecting the users to the most up-to-date guideline for further support, the platform takes the first steps towards supporting the adoption of the standard.



OLSK Small Laser V1

Open Source Laser Cutter - Small version - Open Lab Starter Kit

Introduction

The OLSK Small Laser V1 is a compact but powerful desktop laser cutter. It features the design of the Fabulaser Mini, developed to fit in Fab Labs and classrooms, with a compact size of 870x810x380mm and the weight under 60kg. It uses an optimized 40W CO2 laser, being able to cut 8mm acrylic. With a cutting area of 600x400mm, is one the largest of this machine category.

Open Know How

The OKH checker is a tool that verifies whether a hardware design aligns with the Open Know How specification, ensuring that it follows standardized documentation and metadata practices for open hardware designs

OSH Tool Compliance
30.77% ⚠

is hardware
33.33% ⚠

OSH Machine Readability
26.27% ⚠

OSH Openness
30.24% ⚠

OSH Documentation Quality
22.93% ⚠

OKH version: 0.3.1-15-gbc9808f
Last execution: Tue, 21 Mar 2023 14:00:11 GMT

Figure 10. Open Know How compliance report on the FabCity OS Core

Public repositories

- https://codeberg.org/osh-autodoc
- https://github.com/topics/interfacer-project-eu-wp4-3

2.6 The Building workshops by FCHH

Scope

‘Building for all’ was a three-month free workshop series on sustainability, open-source principles and production techniques. The hands-on workshops tested and explored the possibilities of local and open production in Hamburg, a wide variety of products and prototypes being manufactured and assembled in Hamburg’s open workspaces (Fab Labs) using open-source hardware documentation and guidance from workshop trainers.

The build instructions for each of the prototypes exhibited at #tfom23 were documented in detail with an open-source licence, which entitles anyone worldwide to rebuild the prototype and freely redesign it according to local requirements like taste, size or functionality.

Furthermore, the projects served as use cases mapping and simulating the manufacturing process, mapped within the Fab City OS Core, collaborative platform [BuildWorkshops EVENT].

Public repositories

- https://gitlab.fabcity.hamburg/hardware/interfacer-osh-build-workshops

2.7 Fab City OS Index by FCHH

Scope

Fab City OS Index (aks Fab City Dashboard) is a visualisation tool developed for automated data acquisition, able to monitor the circularity status of a Fab City.

Technology stack

The component is base on the open source software node-red and grafana[2] and displays the representation of OpenLab [3]on card modules, contained machine assets/commons in the OpneLab, machine status, energy and temperature parameters and material data points.

The Microfactory - tfom23 show-case room

For the final event a Microfactory, conceptualising and testing how local manufacturing can be realized in a highly modular and flexible micro factory, was installer, with 3 machines from the starter kit: the small cnc, the small laser cutter, and the small 3d printer, all alimenting the Fab City Dashboard.

[2] https://grafana.com/docs/grafana/latest/panels-visualizations/visualizations/

[3] https://www.fabcity.hamburg/de/projects/openlabstarterkit//

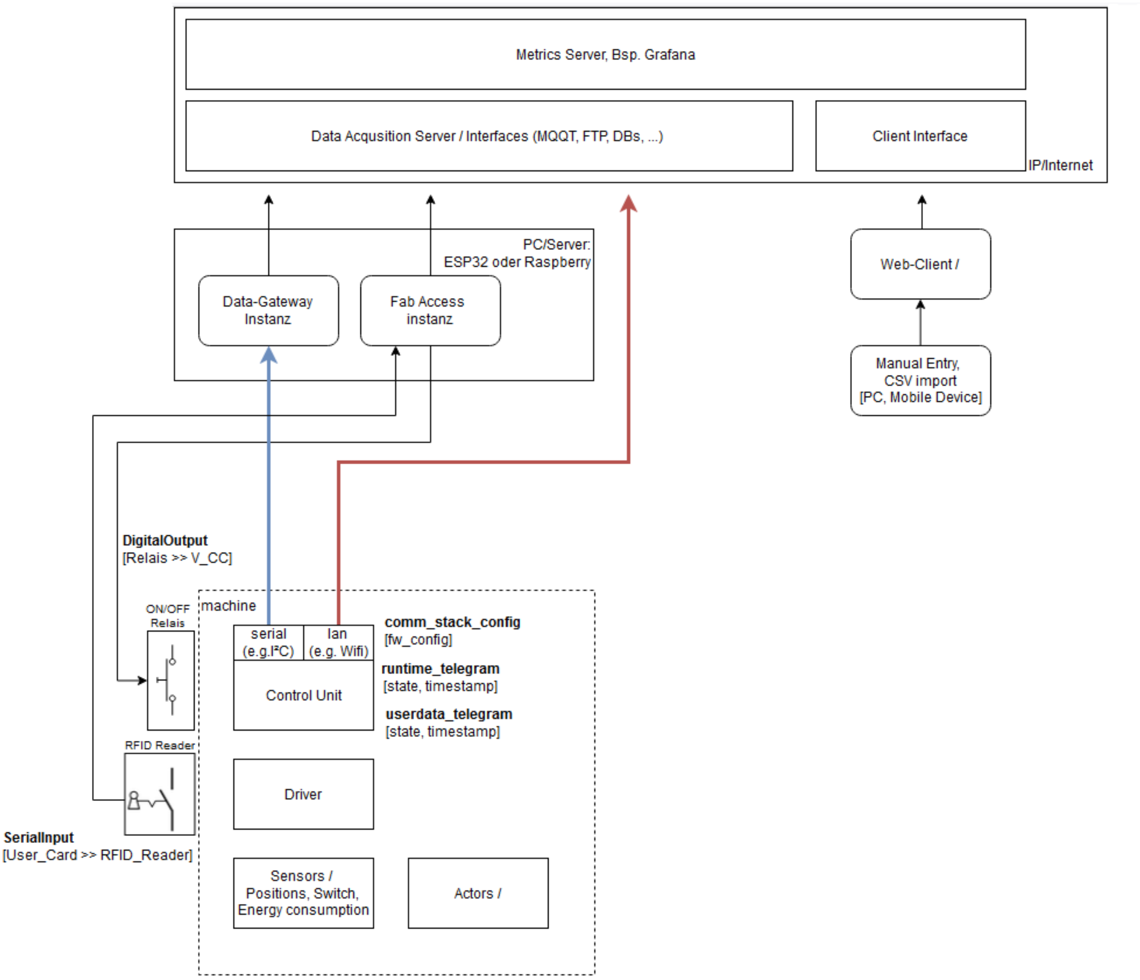


Figure 11. Microfactory - tfom23 show-case room, architecture plan by Michel Langhammer (FCHH)

Public repositories

- https://gitlab.fabcity.hamburg/software/fc-dashboard

3. Conclusion

The interfacer project had very ambitious and interdependent goals, spanning from a new economic model to a federated autonomous infrastructure and proposing a global frame, so that we can change the economic system we are living in, and try to do it in one year and a half, the project duration.

In order to rise to such a magnificent vision and to bring it closer to reality, a lot of coordination, brainstorming, collaboration, feedback and patience was needed, supported by a strong agile methodology. That enabled us to progress and accelerate with the development and to provide practical and functional components, and at the same time offered enough flexibility to integrate real user feedback (via Technical Demonstrations and Technical Alignment meetings) with evolving legislation and technical standards.

Although we developed breakthrough innovative tools, such as the federated digital identity, the Creative Flows economic proposal, the automated documentation etc, we also had the occasion to discover how much more can and should be done in order to support and scale the new DIDO model.

We are only at the beginning of a new approach towards global collaboration and further research, trials and testing will be needed in order to address tensions between local diverse multidisciplinarities and global system thinking; DIY culture and standards adoption; as well as fair value distribution and accumulation tendencies, just to name a few.

We believe that by documenting our work, knowledge and experience, more can join the process either by collaborating, adopting or by further developing the described solutions.

4. References

[FabCity Network] Fab City Foundation cities members of the network. Url: https://fab.city/network-sustainable-cities/
[BuildWorkshops EVENT] blog on the Interfacer website, Url: https://www.interfacerproject.eu/news/buildworkshops/