SIMPLIFICATION OF ELECTION PROCEDURES

Vlaamse overheid

Proudly presented by

Introduction

In any given election, certain procedures must be followed to ensure adherence to democratic principles. These procedures come with a vast amount of paperwork for all actors involved. One of the steps in the election process is the declaration of funds raised and expenses made during the campaigns.

For osoc20, the Flemish Agency for Local and Provincial Government (Agentschap Binnenlands Bestuur, ABB) decided that the entire process should be simplified, digitized, and open source. This is what our team has worked on for the past four weeks. By introducing Linked Data & Solid Pods as technological solutions for the declaration of funds, this project aimed to reduce the amount of paperwork required to be completed by the candidates, simultaneously simplifying the process of fact checking the data. Our goal was to provide a proof of concept that ABB could take to other institutions and highlight the benefits of switching to digital methods.

Within four weeks, we built an online application that offers three main features:

1. The candidates can declare their expenses via an online platform. Their personal information is then saved, meaning the next time they have to use the platform, there will be no need to re-submit their personal data;

2. A database solution via Solid, so that the candidates are in full control of their own data;

3. A platform that can be used by citizens and legal institutions to have an overview of the funding and expenses of their representatives.

Context

Target Groups

Our project was primarily concerned with digitizing the declaration of election expenses, thereby focusing on the experience of individuals running for political positions in the Flemish government.

However, as the product is available to use for anyone, we have defined three additional beneficiaries of our project.

The government.
The introduction of a digital tool for the process allows government officials to analyze and check whether the forms have been filed, their sources, and their contents.

Courthouses and the legal system.
The current election expense declaration process requires that the forms be available to the judicial courts for a minimum of 4 months. Following that period, the records are destroyed as to not take up physical space at the courthouses. With the introduction of digitized forms, we are effectively solving the issue of storage, meaning that the data could potentially be stored online indefinitely, thus creating the possibility to do research on the (anonymized if needed) data over different elections.

General public.
Public scrutiny over declaration and the way that politicians decide to expense their funds, possibility of viewing the gifts that politicians receive, and whether they stick to the limits set by the government might reinforce adherence to the regulations faced by the politicians. In addition, political sponsors will be able to review said data, and decide whether they are satisfied with the way it has been expensed.

Similarly to the case of the judicial system, the public is only able to review the forms during a 9 day period. The introduction of a digital solution has the ability to simplify the process of viewing and analyzing the data from the perspective of the citizen.

We should note that as the data is stored online, and is technically not destroyed, a distinct separation of access between the public and the legal and government system must be defined, as to not violate the GDPR.

User stories

The main purpose of user stories is to generate an understanding of how a software feature is viewed from the end user's perspective. By describing a user type, we aim to understand what the user wants and why. We created user stories from the perspectives of two citizens and a politician that has participated in the local elections.

Peter Janssens

Audience segment: Citizen

Age: 34
Marital status: Dating
Children: 1
Undergraduate study: Industrial Engineer
Occupation: Engineer
Location: Antwerp
Income: Enough

"I believe transparency in government is key to restoring our nation's faith in its elected leaders."

Bio

Peter is an engineer that focuses on his health. He follows politics very closely. He strongly believes in democracy and that a vote can be very powerful.

Goals / Needs

He wants to find the data (income and expenses) of politicians so that he knows his government is transparent.

Ideal experience

Peter finds the data easily, without making much of an effort. The data is clear, simple and easily accessible.

Pain points

Peter doesn't trust the political system. He's very critical towards the people that are responsible for running our country.

Technology & Information Sources

Peter uses his laptop and smartphone daily for work. He also has a tablet but his daughter uses it to play games.

Lorenzo Pasolini

Audience segment: Politician

Age: 43
Marital status: Married
Children: 3
Undergraduate study: History
Occupation: Elected representative
Location: Tongeren
Income: Enough

"As a representative I am the voice of the people. Before I make a decisions I want to be up-to-date about everything."

Bio

Lorenzo, son of Italian immigrants, has founded his own family in Limburg. He received his Masters degree in History at the UAntwerpen. Since the elections of 2018, he is an elected representative within the Flemish Parliament for the CD&V.

Lorenzo has always been interested in politics, especially since his parents were part of the working class.

Goals / Needs

He wants to find the data (income and expenses) of politicians so that he knows his government is transparent.

Ideal experience

Not spending a lot of time on paper work is something he keeps on dreaming about. Being able to input and check his own data in a simple way and maybe being able to look at the expenses of his competitors would make his life more easy.

Pain points

Lorenzo doesn't have a lot of time on his hands, especially not for writing/signing documents on paper all the time.

Technology & Information Sources

Lorenzo is not a fan of change. He is skeptical about digitizing and automating processes. He has a laptop and a smartphone he uses for work.

Laura Opdebeek

Audience segment: Commercial Company

Age: 36
Marital status: Divorced
Children: 1
Undergraduate study: Pharma
Occupation: Business Owner
Location: Gentbrugge
Income: Enough

"I think women should get more involved in politics, so more diverse views are represented. That's why I decided to support my friend in our municipality when she ran for becoming mayor."

Bio

Being a female business owner, Laura is able to spot potential and opportunities. She sponsors causes she believes in. That's why she decided to sponsor a candidate of her own municipality.

Goals / Needs

She wants to check if the candidate that she has sponsored has properly registered the money (or something 'in kind') correctly.

Ideal experience

Quick overview on how her investment is being used.

Pain points

Laura works hard for her money. She wants to know where her money is going and what it's being used for.

Technology & Information Sources

Laura is a multitasker. She's writing mails on her laptop, calling on her smartphone and taking notes on her tablet all the time!

Customer Journey

How were the customers dealing with the goals they wanted to achieve before?

During the first client meeting and briefing, it became clear that the users of our product (the politicians or representatives) want a simplified online procedure without paperwork.

The current expense declaration system is analog. There are four forms each political party and politician has to fill out to successfully declare their election expenses. These are the A105, A106, G103 and G104.

One of our interviewees admitted that they experienced the process as "cumbersome and comparable to filling in a tax return". According to him, many people don't understand the different boxes and codes. To simplify the process, the user needs a simplified declaration system, whilst still remaining responsible for their own input.

What were their problems?

The process is complex, takes a lot of time and might be confusing for some candidates. Especially those who do not really compete to win but rather do it as a hobby, or those who lost the election, and thus lost their motivation to keep up with the declaration of their expenses.

Some of the respondents that filled in our question form referred to the current process as "cumbersome", "a lot of paper per candidate on which hardly anything was filled in", and "complex".

What did we build?

Our team built an online application that offers three main features. First of all, we made an online platform where candidates declare their details of the campaign. We focused on saving personal data so they do not need to repeat this process every election period. Furthermore, their data is then stored in a Solid Pod, so that the candidates remain in control of their own data. Lastly, for the front-end part of our platform, we created a way to have an overview of the representatives' campaign expense details, in addition being able to see whether the candidates have completed the declaration or not. That platform can be used by citizens and legal institutions alike.

Is there any competition?

Because ABB is responsible for everything that has to do with local elections in Flanders, we cannot speak of competition. It's a governmental branch that is the sole player, therefore we cannot say whether our product is doing better than a competitor, but we aimed for our project to be an improvement upon the current procedure.

TIP

If we take a look at other countries, and how they handle the process, we see that the United Kingdom and the United States also have a digitized process. It might be handy to take a look at their platforms via **https://www.fec.gov/data/filings/?data_type=processed** for the United States and via **https://www.electoralcommission.org.uk/2019-candidate-spending** for the United Kingdom.

The Product

Here are some useful links that will take you directly to the results of our project.

Website for forms to be completed by candidates
Website where you can see the forms filled in by the candidates

Github repositories

oSoc20/solid-elections-submission: Site allowing candidates to declare their expenses and income
oSoc20/solid-elections-api: Python API providing endpoints for the declaration and front-end websites using the mandaten/lblod data
oSoc20/solid-elections-wrapper: Wrapper around Solid to allow the creation of an application directory, create files and fetch the data from forms
oSoc20/solid-elections-frontend: Website allowing users to see the expenses and income of the different candidates
Github Page of the project

To run these projects on a local development server, you will need Node/npm and docker-compose.
For a production server, you need a Docker Swarm instance for the API and a webserver to host the frontend on (we are using GitHub pages)

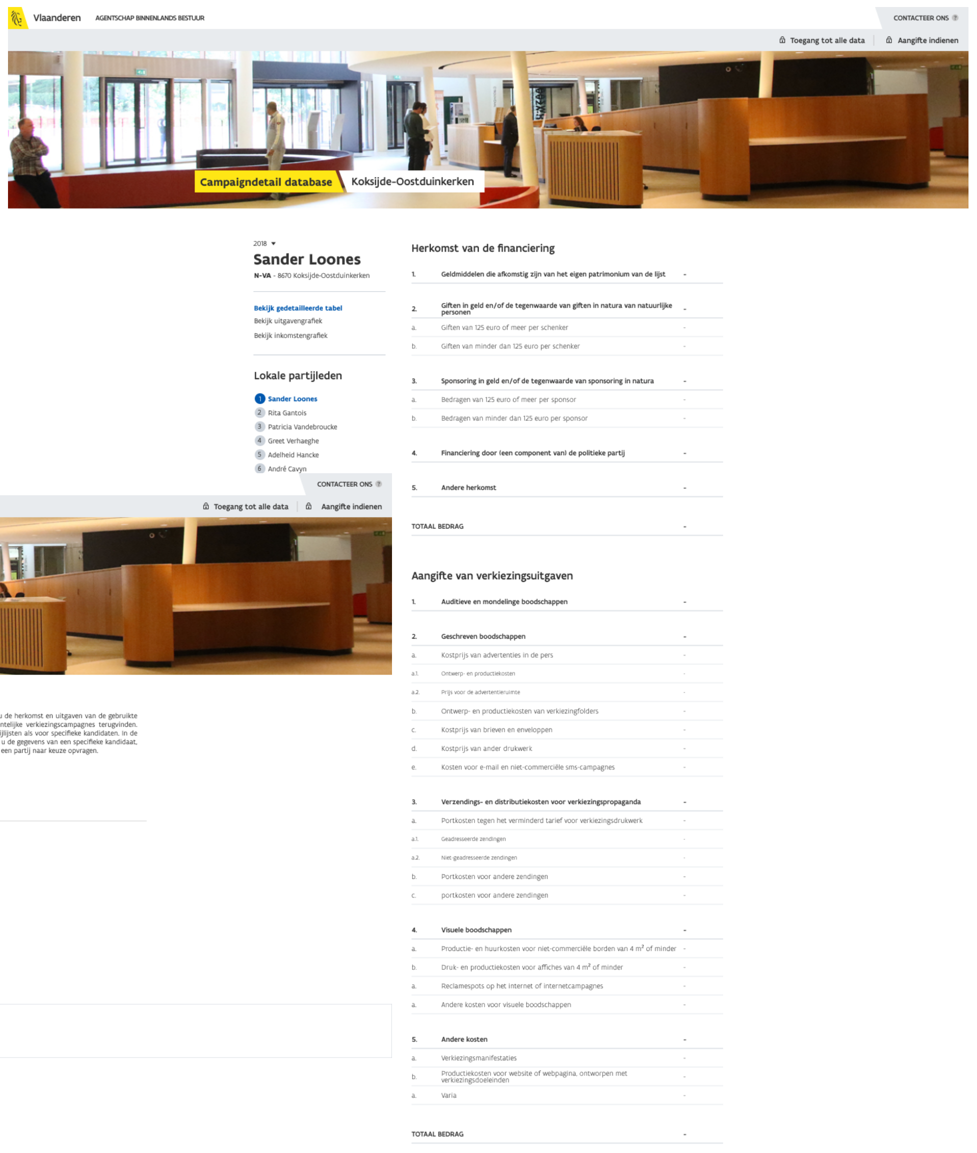
Creating the product

Branding

The Flemish government has developed its own style guide. This guide includes the “Flanders Arts” font, custom icons, and guidelines on images to name a few. They provided us with their “web universe”, this includes pre-made components that allowed us to rapidly build our application. By using these guides and elements, we were able to stay within their branding.

Sketches, wireframes & mock-ups

With this remote edition, our design teams started wireframing using [Figma](). By using Figma, we could work together on our design ideas, starting with the design of our crest, to the final mockups of the website. In the following visuals, you can see the progress of our work.



Sander Loones

N-VA - 8670 Koksijde-Oostduinkerken

Bekijk gedetailleerde tabel
Bekijk uitgavengrafiek
Bekijk inkomstengrafiek

Lokale partijleden

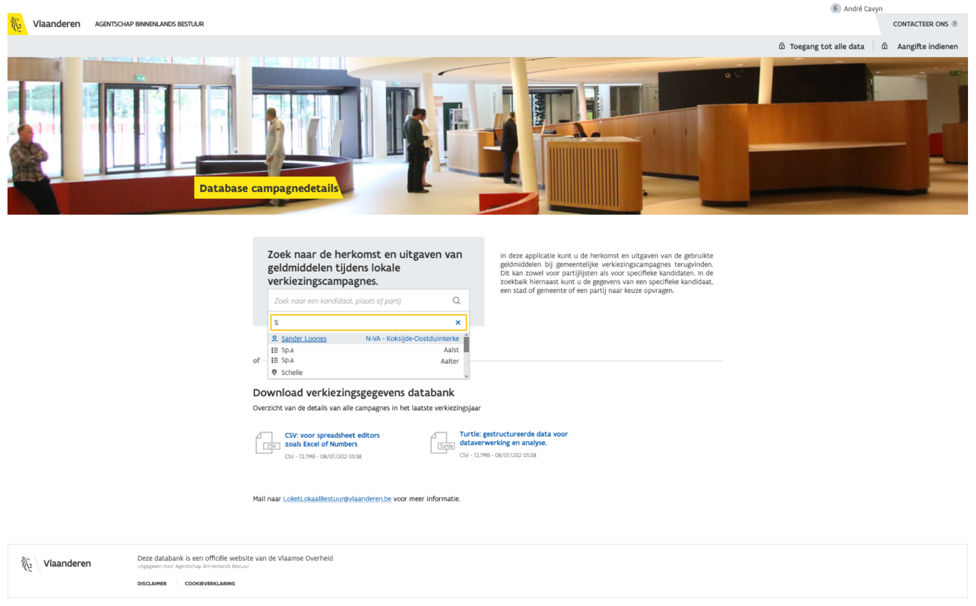
1 Sander Loones
2 Rita Gantois
3 Patricia Vandebroucke
4 Greet Verhaeghe
5 Adelheid Hancke
6 André Cavyn

Herkomst van de financiering

1.	**Geldmiddelen die afkomstig zijn van het eigen patrimonium van de lijst**	-
2.	**Giften in geld en/of de tegenwaarde van giften in natura van natuurlijke personen**	-
a.	Giften van 125 euro of meer per schenker	-
b.	Giften van minder dan 125 euro per schenker	-
3.	**Sponsoring in geld en/of de tegenwaarde van sponsoring in natura**	-
a.	Bedragen van 125 euro of meer per sponsor	-
b.	Bedragen van minder dan 125 euro per sponsor	-
4.	**Financiering door (een component van) de politieke partij**	-
5.	**Andere herkomst**	-
TOTAAL BEDRAG		-

Aangifte van verkiezingsuitgaven

1.	**Auditieve en mondelinge boodschappen**	-
2.	**Geschreven boodschappen**	-
a.	Kostprijs van advertenties in de pers	-
a.1.	Ontwerp- en productiekosten	-
a.2.	Prijs voor de advertentieruimte	-
b.	Ontwerp- en productiekosten van verkiezingfolders	-
c.	Kostprijs van brieven en enveloppen	-
d.	Kostprijs van ander drukwerk	-
e.	Kosten voor e-mail en niet-commerciële sms-campagnes	-
3.	**Verzendings- en distributiekosten voor verkiezingspropaganda**	-
a.	Portkosten tegen het verminderd tarief voor verkiezingsdrukwerk	-
a.1.	Geadresseerde zendingen	-
a.2.	Niet-geadresseerde zendingen	-
b.	Portkosten voor andere zendingen	-
c.	portkosten voor andere zendingen	-
4.	**Visuele boodschappen**	-
a.	Productie- en huurkosten voor niet-commerciële borden van 4 m² of minder	-
b.	Druk- en productiekosten voor affiches van 4 m² of minder	-
a.	Reclamespots op het internet of internetcampagnes	-
a.	Andere kosten voor visuele boodschappen	-
5.	**Andere kosten**	-
a.	Verkiezingsmanifestaties	-
b.	Productiekosten voor website of webpagina, ontworpen met verkiezingsdoeleinden	-
a.	Varia	-
TOTAAL BEDRAG		-



Zoek naar de herkomst en uitgaven van geldmiddelen tijdens lokale verkiezingscampagnes.

In deze applicatie kunt u de herkomst en uitgaven van de gebruikte geldmiddelen bij gemeentelijke verkiezingscampagnes terugvinden. Dit kan zowel voor partijlijsten als voor specifieke kandidaten. In de zoekbalk hiernaast kunt u de gegevens van een specifieke kandidaat, een stad of gemeente of een partij naar keuze opvragen.

Download verkiezingsgegevens databank

Overzicht van de details van alle campagnes in het laatste verkiezingsjaar

CSV: voor spreadsheet editors zoals Excel of Numbers
CSV - 72.7MB - 08/07/202 05:58

Turtle: gestructureerde data voor dataverwerking en analyse.
CSV - 72.7MB - 08/07/202 05:58

Mail naar LoketLokaalBestuur@vlaanderen.be voor meer informatie.

Vlaanderen
Deze databank is een officiële website van de Vlaamse Overheid
uitgegeven door Agentschap Binnenlands Bestuur
DISCLAIMER
COOKIEVERKLARING

Forms and Interactions with Solid

The first component of this project was to replace the current paper forms (A105, A106, G103, and G104) with digital ones that must be filled in by politicians and political parties when running for office. Since the goal of this project was to simplify the expense declaration process, we started by brainstorming about ways to improve this process. For example, one of our improvements involves fetching personal data (e.g. name, address, etc.) from the "Mandatendatabank" and "Data Vlaanderen", so we can remove these fields from all forms. Similarly, list name and tracking number can be removed in the same manner. We also added checks throughout the form that warn users if they need to fill out other forms, ensuring a complete declaration in a manner that meets any legal requirements.

One of the challenges of digitizing the process is confirming user identity and data validity. After some research on the subject, we have chosen to use an integrated login system of Solid, which is based on WebIDs. Solid is a place where data is online and decentralized. It also provides a system where the user can manage permissions for their data and see which data is used. These qualities are what differentiated Solid, deeming it the ideal ecosystem to use in our project. The user can currently log in with their Solid login data. In future renditions of this project, users could use their e-ID to log in. For example, you can create a Solid account, and then ask the user to connect their ID card. After these steps, we can get some of their data, and store their WebID (generated from solid) and the rijksregister (civil registry) together (currently we use WebID and LBLOD ID). This process ensures the user's identity, as they are required to use their ID card.

When a user logs in, they connect to their Solid pod, and we temporarily store their WebID. If this is the first time the user has logged into the website, we create a folder inside their pod called 'solidelections' in the AppContainer directory. We then prompt the user to fill out their profile. Once that is done, a file called 'me.ttl' is stored on the Solid pod containing all of their personal data. If, however, this file already exists, the user can continue to make declarations.
The user can still access this profile and edit it directly from the app. In fact, we fetch the me.ttl file on the candidate's solid pod and fill in the profile automatically - the user only has to edit what they want, and save it again. We do ask the user to write their name and last name, but it could be fetched with the rijksregister or e-ID. However, if you would like to prefill the profile, it is possible to ask the user to login with their e-ID, fetch their data, fill in their profile, and only then ask for the data that is missing.

Any declarations about expenses made will be stored onto the user's Solid pod, and the apiwill be notified of this declaration. This ensures that no data is stored twice, and the public website is up-to-date.However, some personal data is stored on the Solid pod (LBLOD ID for now, could be changed to the rijksregister). This information is used to fetch data from the database of data. vlaanderen.be. If we store their rijksregister, we canfetch the candidate's last name, first name, and other personal information. We also store the form on the Solid pod, but it could be possible to give an ID to this form, and store the data with the ID of a chosen database. The information of the expenses is therefore stored on the personal solid pod of a candidate, leaving that person in control of the data. This has some consequences concerning the adjustment of the data after submission. A candidate can change their data after submission and there is not much the government can do about it when using solid technology. One suggestion was to store a hash of the expenses in the database. In this way, it can be checked if the data was changed after submission. However, if it is important to keep control of the adjustment of the data, solid might not be the ideal way to go.

In order to accomplish this, we have several dependencies within our code:

Libraries:
Bootstrap v4.3.1 (partially replaced by Vlaanderen Webcomponents)
Use to design the app (https://getbootstrap.com/)

Popper v1.14.7
Bootstrap's dependency (https://popper.js.org/)

JQuery
Bootstrap's dependency (https://jquery.com/)

Vlaanderen Webcomponents v3
Use to design and make the app responsive

Sass
Write CSS easier (https://sass-lang.com/)

Typescript
Write JavaScript easier and compiled to JavaScript at the end (https://www.typescriptlang.org/)

Languages:
JavaScript

Frameworks:
ReactJS

Detailed description

At the top of the file hierarchy lies App.js. Here we return the navigation bar component to universally place it at the top right corner of the website. This navigation bar uses RouteLinks to switch between pages. It also allows users to log in and out, and has a playful greeting on the top right. Next, a footer component is added to end the page with useful information for the users. This footer is taken directly from the Vlaanderen Webcomponents library, and thus is of similar design as the footers found in official Vlaanderen governmental websites.

Next we have the index.tsx, the default page of the website. Visually, this page simply notifies the user whether they are properly logged in, but a lot of other processes happen in the background. If the user is logged in, index.tsx will use the useEffect method to define the AppContainer. As mentioned before, this is a representation of the user's Solid pod, and will be used whenever we need to make changes to the user's data. After that, the file will render different components, depending on the current Route. For example, the profile page '/profile' will show the CandidateDataForm component. This form is used to gather personal data from the candidate.

In order for us to make the interactions with Solid easier, we have created a wrapper called 'SolidWrapper.ts' to provide a few helper functions. These functions entail, but are not limited to, creating files, folders and documents, storing data, gathering data, etc.

We have also taken measures to ensure proper loading of components. componentDidMount is invoked immediately after a component is mounted (inserted into the tree). componentDidUpdate is called whenever any data is updated, and will update the appContainer. We also ensured that the Solid pod is properly prepared to receive data. If it is not, the proper files and folders are created asynchronously.

Another important component is the G103 form. When it is loaded, several checks happen to ensure proper functioning (e.g. catching errors and ensuring a profile exists). If everything is in order, the form loads. This form is a replica of the current G103 form, but with all duplicate data removed. Therefore it is significantly shorter and easier to fill out. When the form is submitted, all fields are checked for validity, the data is stored to the Solid pod and the API is notified.

Data path: From the form to the frontend

In this section, we describe the path taken by the data collected in the form, and how the backend handles that data. First, the candidate needs to login into their Solid Pod and complete their profile by providing their LBLOD ID. When this step is done, an HTTP request is made to our API which receives the URI to the Solid Pod and the LBLOD ID of the candidate and stores it into a PostgreSQL database.

We decided to use a PostgreSQL database, but ideally only a linked data store should be used to correctly link the URI of the candidate to their LBLOD ID.

As said in the previous section, the form filled in by a candidate is directly stored in their own Solid Pod. Therefore when the front end needs the form of a candidate the API only sends the URI to the Solid Pod and the front end is in charge to fetch the data inside the Solid Pod. Our API also stores the data from the "Mandatendatabank" and other data provided by ABB regarding candidates.

API

For our API, we use a Python framework called Sanic. We chose Sanic because of our previous experience with Python and because it is faster and more lightweight than a more "complete" framework such as Django. In usage, it is quite similar to Flask, but it is roughly six times as fast and offers support for asynchronous processing of requests.

For our database, as stated before, we use PostgreSQL, with which we interact in our API with an object-relational-mapping library called peewee. This just makes it easier to work with the database as we can interact model objects instead of sending raw SQL queries (which might also expose us to security problems if we're not careful)

We are also running an Openlink Virtuoso instance, to easily and efficiently query the dataset we received with information on the running candidates using SPARQL.

The first idea we had was to provide a simple API that was just used to store and provide the URI to Solid Pod of candidates who had already filled in a form and set up an Openlink Virtuoso instance alongside so the front end could query easily both. However, since none of the team members were familiar with Linked Data at the beginning of this project, it was decided that only two or three developers would focus on how the Linked Data technology works. Therefore, we left the design and front end specialists to focus on building websites while we created a few easy-to-use endpoints for them. These endpoints are provided by the API, to remove the complexity of learning SPARQL for a more design-oriented person.

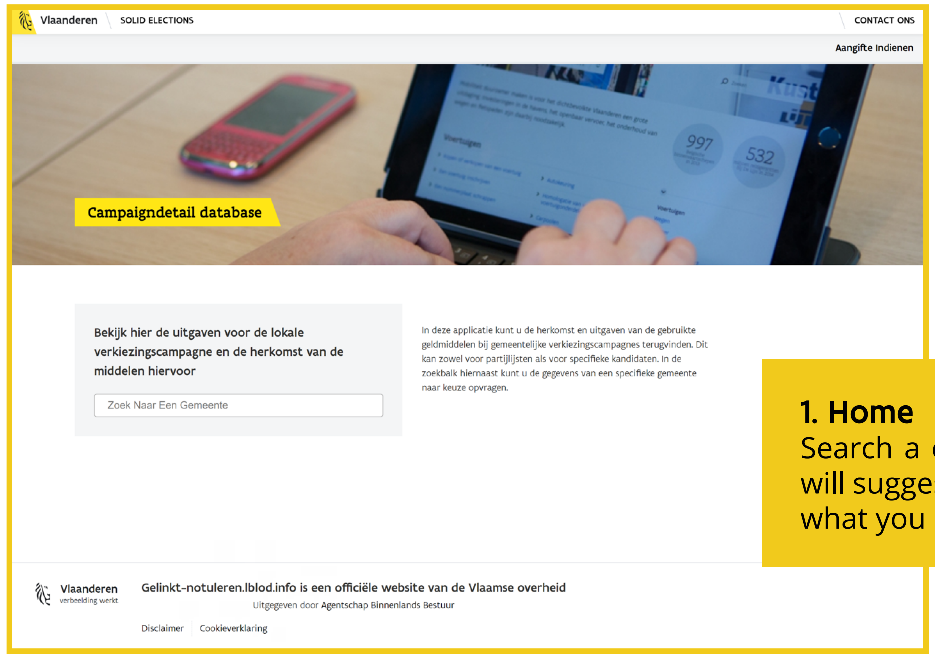
This whole stack is put inside of a Docker container, which is then deployed to a single-server Docker Swarm instance.

Front-End

The front-end has the general public as its target audience. As a citizen, they can view all declarations of each candidate in the candidate list.

Tech
React.js
Web Universum
App Universum
SASS

Deployment
Step 1: download files from github Step 2: at root of this folder: type in terminal `yarn install`
Step 3: type in terminal: `yarn build` or `yarn start` to test the website
Step 4: copy the build folder and put it on your static server



Bekijk hier de uitgaven voor de lokale verkiezingscampagne en de herkomst van de middelen hiervoor

Zoek Naar Een Gemeente

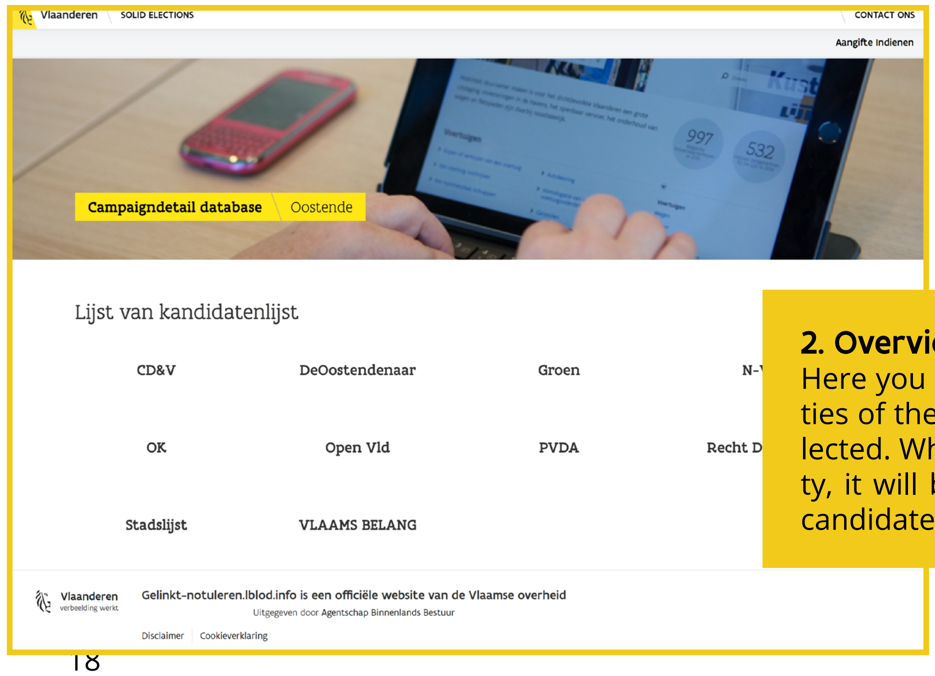
In deze applicatie kunt u de herkomst en uitgaven van de gebruikte geldmiddelen bij gemeentelijke verkiezingscampagnes terugvinden. Dit kan zowel voor partijlijsten als voor specifieke kandidaten. In de zoekbalk hiernaast kunt u de gegevens van een specifieke gemeente naar keuze opvragen.

Vlaanderen verbeelding werkt

Gelinkt-notuleren.lblod.info is een officiële website van de Vlaamse overheid
Uitgegeven door Agentschap Binnenlands Bestuur

Disclaimer | Cookieverklaring

1. Home
Search a city, start typing and it will suggest a few cities based on what you are typing.



Lijst van kandidatenlijst

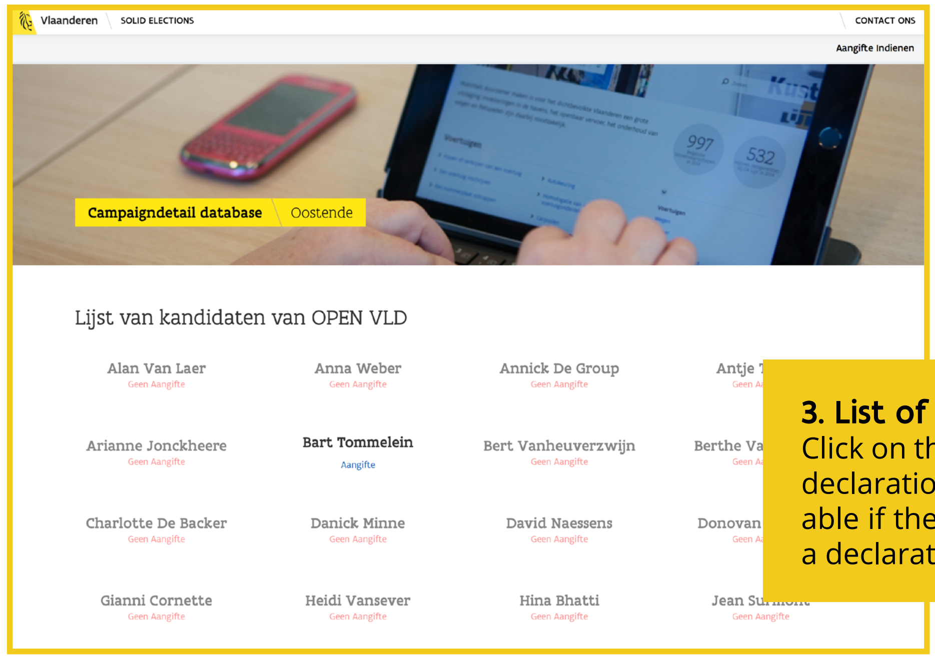
CD&V	DeOostendenaar	Groen	N-V
OK	Open Vld	PVDA	Recht D
Stadslijst	VLAAMS BELANG		

Vlaanderen verbeelding werkt

Gelinkt-notuleren.lblod.info is een officiële website van de Vlaamse overheid
Uitgegeven door Agentschap Binnenlands Bestuur

Disclaimer | Cookieverklaring

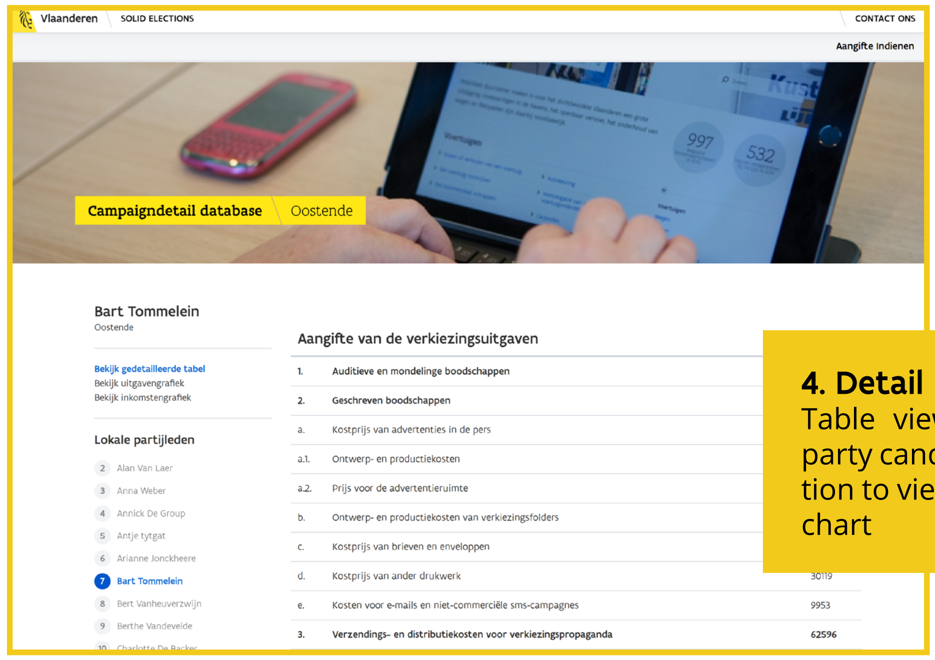
2. Overview of candidatelists
Here you can view all of the parties of the location you have selected. When clicked on one party, it will bring you to the list of candidates of this party.



Lijst van kandidaten van OPEN VLD

Alan Van Laer Geen Aangifte	Anna Weber Geen Aangifte	Annick De Group Geen Aangifte	Antje T Geen A
Arianne Jonckheere Geen Aangifte	Bart Tommelein Aangifte	Bert Vanheuverzwijn Geen Aangifte	Berthe Va Geen A
Charlotte De Backer Geen Aangifte	Danick Minne Geen Aangifte	David Naessens Geen Aangifte	Donovan Geen A
Gianni Cornette Geen Aangifte	Heidi Vansever Geen Aangifte	Hina Bhatti Geen Aangifte	Jean Su Geen Aangifte

3. List of candidates
Click on the available name go to declaration detail, It's only clickable if the person has submitted a declaration.



Bart Tommelein
Oostende

Bekijk gedetailleerde tabel
Bekijk uitgavengrafiek
Bekijk inkomstengrafiek

Lokale partijleden
2 Alan Van Laer
3 Anna Weber
4 Annick De Group
5 Antje tytgat
6 Arianne Jonckheere
7 Bart Tommelein
8 Bert Vanheuverzwijn
9 Berthe Vandevelde
10 Charlotte De Backer

Aangifte van de verkiezingsuitgaven

1.	Auditieve en mondelinge boodschappen	
2.	Geschreven boodschappen	
a.	Kostprijs van advertenties in de pers	
a.1.	Ontwerp- en productiekosten	
a.2.	Prijs voor de advertentieruimte	
b.	Ontwerp- en productiekosten van verkiezingsfolders	
c.	Kostprijs van brieven en enveloppen	
d.	Kostprijs van ander drukwerk	30119
e.	Kosten voor e-mails en niet-commerciële sms-campagnes	9953
3.	Verzendings- en distributiekosten voor verkiezingspropaganda	62596

4. Detail page decleration
Table view of declaration with party candidates sidebar and option to view income and expense chart

Communication plan

We have two main groups to focus on. The first one being the politicians that participated in the elections and need to declare their expenses. Many can be reached on the platform vlaanderenkiest.be, a website of the Flemish government that shares information on the upcoming elections including how to file expenses.

Another way of sharing information to the candidates is via the party's central management and the communication channels of the parties. If the Flemish government shares the new information with those institutions, the communication departments can use a trickle-down process to give all the representatives or even all the candidates the necessary information.

Focussing on the second group, the general public, the Flemish government can use its own press releases to communicate the new digitized way of working and accessing the data. Additionally, it may put a brief statement in its newsletters. It is also an option to put down flyers at the old access points. In this way, the people who used to come to retrieve the data can get a physical reminder that the process is now entirely online.

Feedback

User Testing

We worked very closely with Veronique Volders, one of the client's representatives. She can also be seen as part of our target audience. Therefore, we did not engage in other user tests. The real-time feedback made it possible to be nimble in developing the project.

If our client wants to perform a user test in a later stage, we can only recommend that. It is always helpful to receive valuable feedback from the end user. This way, details that the users value can come to light when working out the further phases and changes can be made.

User testing - set up

In order to test the application, we have created an account on the Inrupt Solid Pod. This section explains how you can test the application.

NOTE
If you face any problems during the testing process, do not hesitate to send us an email, and we will try to resolve it as soon as possible.

During this test, you will take on the role of Maxime Callaert, an N-VA candidate from Sint-Niklaas.

Below you will find step-by-step instructions for this user test:

1. Go to https://osoc20.github.io/solid-elections-submission/
2. Use the 'Login' button at the top right corner.
3. Select 'Inrupt' and insert these credentials:
 username: abbaccount1
 password: ABB_account1
4. Go to the 'profiel' page where you can create the profile.
5. In the LBLOD ID field you need to insert the LBLOD ID of Maxime Callaert: http://data.lblod.info/id/personen/cb69bf6f1987d7aaa80db8037725526e98ab1521ebca57d772d53a2c798308bf
6. Once your profile is completed, you can go to the 'declareer' page, and complete the form.
7. When all the above steps are done, you can go to https://osoc20.github.io/solid-elections-frontend/
8. Search for 'Sint-Niklaas'.
9. Click on the N-VA text.
10. Search for Maxime Callaert, and provided everything was completed correctly, you should be able to see the content of the form.

The future

Added value for the open source & open data community

Introducing a digital solution to the process of election expense declaration ensures a greater exposure to a larger population of potential actors that will now be able to view and interact with the data. The current process of the general public gaining access to this data involves going to the courthouses during the 9 day period (during which the forms are made available to be accessed by the general public), and physically sifting through the individual forms of each candidate. We are hoping that by digitizing the entire process, and therefore reducing most of the friction preventing individuals from reviewing the data would inspire political involvement, and ex-post increase the politicians' adherence to campaign expense laws.

To be implemented

The remainder of the forms - G104, A105, and A106 - must be implemented. We implemented G103 directly into the code itself which is a bad practice as putting the content of a form directly into the code does not allow the extensibility of the code to other forms, but due to the time constraints we have not had the time to do it in a better way. One great solution for the form implementation would be to save the form using Linked Data directly. This solution makes it possible to link candidates' answers directly to the forms and to keep everything in a Linked Data format.

Reflection on Solid

One of the problems we ran into when using Solid is that it is very slow when dealing with a lot of data from different users. This is an inherent weakness in Solid's design. Because data is stored in a decentralized manner, the client has to make a request to every Solid pod separately instead of sending one request to a central database. There are ways to work around this by caching the data centrally or creating a network of Solid pods that cache each other's content, but this is a very hard thing to get right.

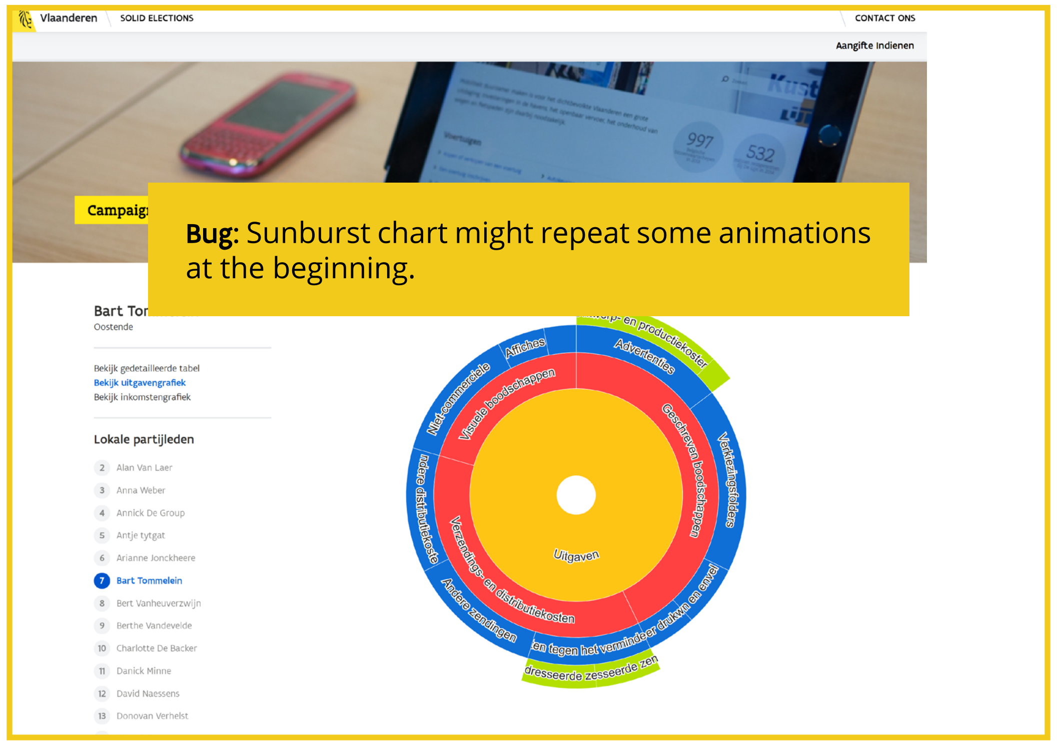
Even though Solid technology has some brilliant ideas, it is still very young. At the time of writing, the documentation is incomplete, and most of the solutions to simple problems can only be found by searching through their forum. Therefore, using this technology in production can be a significant challenge.

Another inherent weakness of Solid is that it is very hard to prove a user is actually who they are claiming to be. Anyone can create a Solid pod and set any name in their WebID. This means that a malicious user could register a Solid pod for a certain candidate and upload false expense data. One option we have considered is using the existing e-ID solution, but this is simply too complex for the scope of this project.

Coincidentally, another group at this year's Open Summer of Code is working on an issue similar to this one. The Bit of Trust team's goal is to express a human trust relationship digitally, using either public-key cryptography or software tokens, neither of which are perfect solutions to the problem they are trying to tackle.

Another potential improvement in the interaction with Solid would be to use query-ldflex.

Fixes & Bugs



Bug: Sunburst chart might repeat some animations at the beginning.

Future features and ideas

Although we did our very best to deliver the best product we can, there are always improvements that can be made. For example, currently it is not possible to create an overview of politicians that have submitted their data. In Solid, you can only fetch data from one Pod at a time. You can do that, by going through every single SolidPod, but it takes a lot of time to load that data. Saving the submission data in a separate database might solve this problem.

Our client also has given us some additional assignments. Since our team focussed on getting the form G103 fully-working, these additional challenges have still to be completed.

Payment of travel allowances and attendance fees

Citizens, who have to sit in a polling or counting station, and voters can apply for relocation allowances and attendance fees. The client would like to see these applications go online too. By digitising these forms as well, the payment of the travel allowances and attendance fees can be made much smoother, faster and automated. The payments will also be more correct compared to the current process.

Individual expenditure limit over multiple lists

Some politicians are listed multiple times, and therefore have a greater budget to spread out. It would be very insightful for politicians to automatically see whether they have respected this limit. This however is quite complicated, since the backend would need to query each list this politician is associated on, slowing down the website.

Registration of proxies

Today, the 'power of attorney process' is a process on paper in which a voter - for specific reasons - gives a power of attorney to another voter. The polling station itself checks this on the basis of generated certificates. This is both time-consuming (resulting in traffic jams at the polling station) and ineffective, as it is more difficult to establish abuse at that time.The client would like to have a solution where voters can register the proxy online (or via the municipality).

Closed section with detailed information

Every once in a while, a court or other governmental instance may have to look into someone's data. These instances usually have more access to someone's data, and thus should be able to view things like social security numbers, names of donors and sponsors, addresses, etc. It would therefore be important to create a private section of the website where these parties can view this data.

Legal aspect

Currently, all the paper forms from previous elections are destroyed because they can not archive the lot. But with the digitized forms it is another thing to keep in mind: General Data Protection Regulation (GDPR). It is still unclear for ABB how long and for which purposes the data can be requested, used and saved.

What is clear, is that the data must be saved as long the ‘courts’ need them to control it. There can be different sorts of appeals and different sorts of sanctions (criminal or administrative). Also, some data can be destroyed after the periods allowed for appeals are expired.

How long the data can be saved is another question and depends on the motivation to save them:

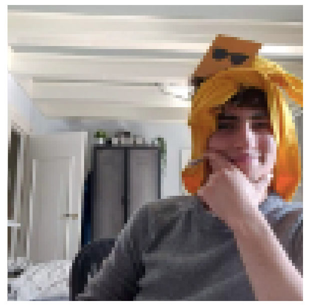
- Control of the data is the main argument (until the periods allowed for appeals are expired).
- Universities can make studies based on the data of several elections: then it would be interesting to anonymize the data

In the future, ABB could look into the possibility of ‘destroying’ data but also a possibility to store it safely and anonymous. Next to that, we recommend that ABB deploys a strategy regarding the issue of GDPR.

Team members

Akvilė Kovalčikaitė
Studentcoach

E-mail	a.kovalcikaite@lse.ac.uk
Twitter	@kovalcia
LinkedIn	www.linkedin.com/in/kovalcikaite

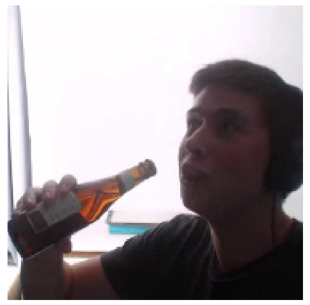


Bram Kreulen
Full stack developer

E-mail	boazkreulen.main@gmail.com
Twitter	
LinkedIn	www.linkedin.com/in/bram-kreulen-3ba60a1b3

Damien Brebion
Back-end developer in charge of submitting data

E-mail	contact@damienbrebion.com
Twitter	@BrebionDamien
LinkedIn	www.linkedin.com/in/damien-brebion



Florent Collin
Back-end developer

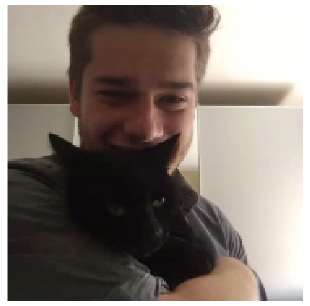
E-mail	florentcollinpro@gmail.com
Twitter	
LinkedIn	www.linkedin.com/in/florent-collin-a67354185

Gianni Van Laere
Communication specialist

E-mail	giannivanlaere@gmail.com
Twitter	@gi_nni97
LinkedIn	www.linkedin.com/in/gianni-van-laere

Inti Valderas Caro
General Designer & Student Coach

E-mail	inti@valderas.be
Twitter	@ValderasCaro
LinkedIn	www.linkedin.com/in/inti-valderas-caro



Jodi De Loof
Coach

E-mail	jodi@jodideloof.be
Twitter	@JodidLoof
LinkedIn	www.linkedin.com/in/jodideloof

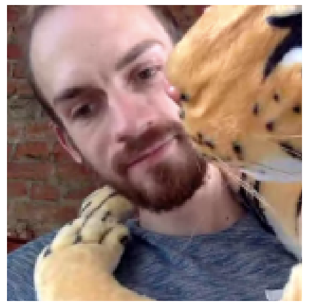
Jonas Vervloet
Back-end developer

E-mail	vervloetjonas@hotmail.com
Twitter	
LinkedIn	www.linkedin.com/in/jonas-vervloet-93567a19b

Team members

Nawang Tendar
Frontend Developer & Student Coach

E-mail nawang.tendar@gmail.com
Github github.com/nawatend
LinkedIn www.linkedin.com/in/nt23



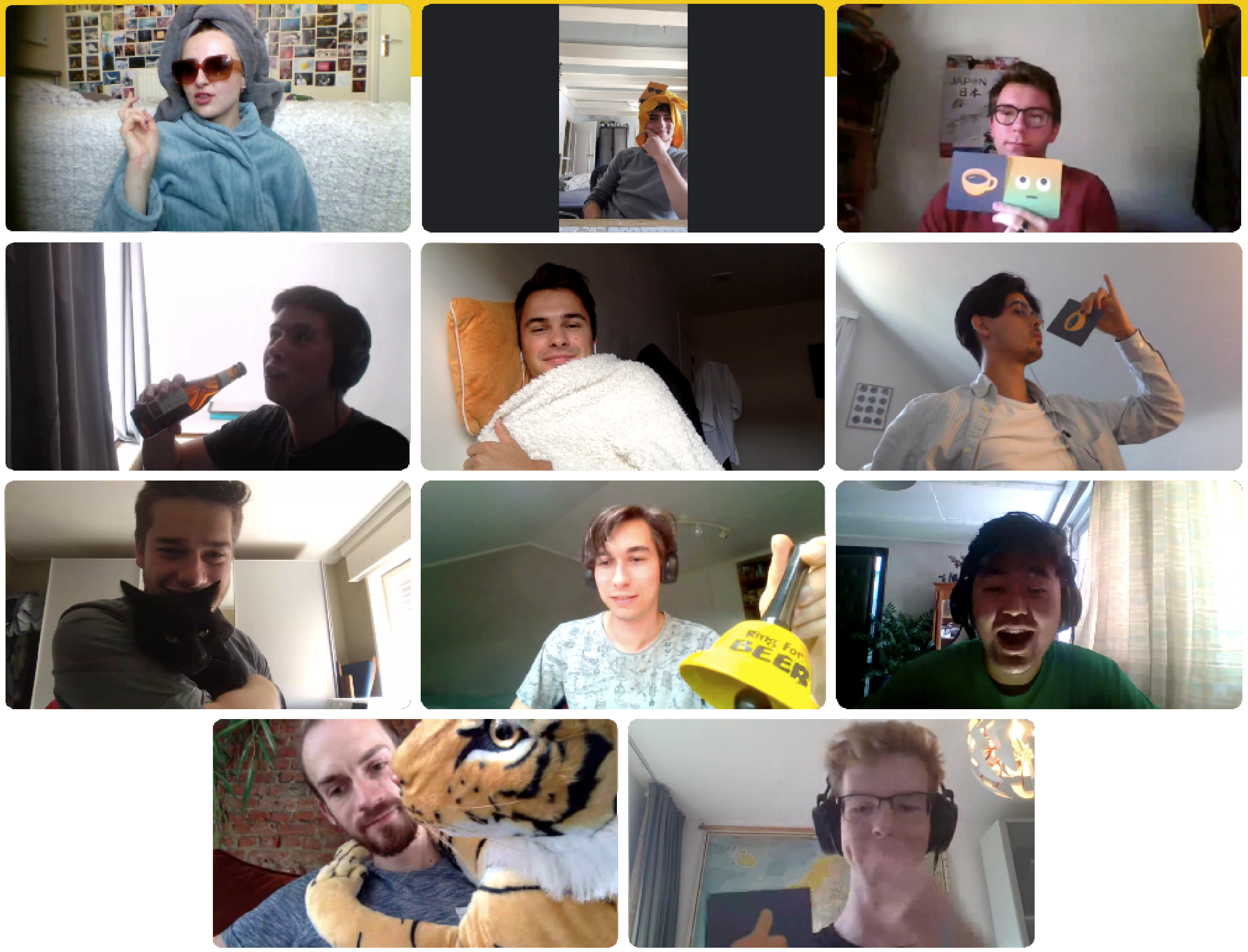
Rutger Bevers
Coach

E-mail rutger.bevers@gmail.com
Twitter @RutgerBevers
LinkedIn www.linkedin.com/in/rutgerbevers

Sam van der Kris
Backend developer & sysadmin

E-mail samvdkris@disroot.org
Twitter @samvdkris
LinkedIn www.linkedin.com/in/sam-van-der-kris-041226168

Thank you!



Our team would like to thank our partners Agentschap Binnenlands Bestuur and Informatie Vlaanderen, with a big shout out to Veronique Volders, An Van Herck, and Goedele Van der Spiegel for providing us with valuable feedback and the opportunity to work on this project.

As a team, we are also incredibly grateful for the osoc community for providing insurmountable amounts of help, workshops, and feedback on our designs. It has been an experience none of us will forget.

Made with love at **Open Summer of Code** 2020