Solar [ ] Space

Defiant, private communication to protest with.

evan kenny

Design against Tech Fascism

The new battleground for democratic rights is on the internet, and so far we the people are losing. The fascist "tech bro" class has come to control much of what we perceive as the internet, through monopolisation, political lobbying and manipulation. As a result, they have turned the internet from an incredible tool for communication into a devastating machine for surveillance and control.

Everything may seem fine at first glance, but often those who have the most to lose are the ones not seen or heard. Already, marginalised groups are retreating from the common digital realm due to harassment and fear of prosecution – or worse, torture and death. Similarly, political activists are trying – and failing – to evade prosecution for their private communications.

We can no longer fight this on the internet, private communication tools like Signal that are a safe have for activists are slowly being torn apart to allow surveillance, we need a new approach. Solar Space explores how we can build decentralised, private digital spaces to use in protest, anarchism and community to fight against big tech and authoritarianism for the future.

Disavowal of Artificial Intelligence

Solar Space as a concept entirely rejects so-called "Artificial Intelligence" tools. The technology industry has long been complicit in violating our privacy to scrape and sell our data, data which is now being scraped harder than ever to fuel an AI investing frenzy. Similarly they have long been complicit in the violation and systemic destruction of human rights globally. These companies are now selling and using their AI tools to commit racial profiling and ethnic cleansing at a horrifying scale. To have used Artificial Intelligence to make this project easier would have been an insult to all those that have become victims of AI, and would have completely discredited the aims of my work.

Free Palestine, Trans Rights, Black Lives Matter.

Table of Contents

01 The Situation

"First they came for the Communists

And I did not speak out

Because I was not a Communist"

"First They Came" - Martin Niemöller, 1946

The End of the World?

It's hard not to describe the current state of the world as dire. For the past ten years we've seen consistent regressions in politics, privacy, wealth divide, living conditions, the global climate, peace and all-round care for one another. With each passing year the situation seems to only get worse, the division grows deeper and the future seems a little less hopeful

It would easy to wallow in the doom and gloom, or even worse embrace a lack of care for the state of the world. But what if we want to do something about it? Is it possible to make a difference, protect our communities and empower each other against a giant, looming evil?

Through Solar Space, I wish to ask these questions. The internet has been one of the most revolutionary tools we have ever created, and yet in recent years it has devolved into a battleground for control and manipulation, and the violation of our right to privacy. Can we reimagine the internet into something that can still remain useful without trampling on our rights or destroying our planet's future?

"On Fire" webcomic by KC Green.

The Situation

ICE pictured in New York

ICE-EU

To understand the scope of the risk of inaction, it's best to take a look to both the United States of America and the European Union.

Donald Trump's second term as the US President has seen a blatant attempt to establish an authoritarian regime in America as a whole. In his first year he has assembled a private police force in the ICE Border Patrol, constructed numerous "detention centres" which are functionally no different from early concentration camps in Nazi Germany, and all the while exacting a war on American soil against it's own people.

And this has taken place without any real opposition from those who hold power in the states. Democrats and Republicans alike have appeared to fall in line, every possible kind of government department has been gutted and replaced with fascist cronies. The only people to truly challenge this regime have been the people, protesting, sabotaging and uniting together. While unlikely to be able to make a major change on their own, their methods clearly have had enough impact to be deemed a threat. "Antifa" – a shortened version of Anti-Fascist – has been deemed a terrorist organisation by Trump, this has included labelling groups that support transgender people as terrorists.

In this opposition against the encroachment of ICE agents into local communities, and their abductions and disappearing of largely non-white citizens, digital communication has been a crucial tool. For a few months at the beginning, it was common to find Facebook groups or apps that tracked and organised against government agents. However, as time went on these were took down or became liabilities. Instead, people flocked to Signal and similar private, encrypted chat applications.

These apps have been used to subversively organise: people assign themselves roles, share ICE agent sightings and call others to action for protests or proaction. The collective organisation has proven effective at combatting random ICE patrols and strikes, with people often ready to respond at a moments notice.

While private communication apps like Signal have proven to be the most effective form of organisation and communication for these communities, it is alarming to see that across the Atlantic Ocean the European Union have spent the past several years attempting to pass legislation that would make these kinds of private chat illegal.

Chat Control

The poster child for digital rights violations in Europe, the controversial "Chat Control" bill would see private, digital communication to cease existence in the EU. Under the bill, all chat platforms would be required to provide police access to messages, which would in turn be scanned by "AI" for "illegal content". To no one's surprise, this bill has been led by the far-right – especially Denmark's representatives – but has seen support across many countries politicians, including Ireland.

Chat Control is advertised by these groups as being a law exclusively for preventing the spread of child sex abuse material. It's the classic "think of the children" line so many conservatives have used over the decades as a justification for eroding democratic rights. Similarly here, the right to privacy is enshrined within the the European Union constitution and for good reason. As the United Nations have said, these privacy laws exist to protect people – including children – which Chat Control would ultimately make more unsafe.

The biggest danger is the ability to define illegality, like the States it would trivial for the EU to declare transgender people or "Antifa" illegal and arrest all who text anything remotely related. While the fight against Chat Control has been holding it back from passing, the opposition have slowly been losing ground. It's perhaps time to look at alternative for private, digital communication.

"Then they came for the Socialists

And I did not speak out

Because I was not a Socialist"

"First They Came" - Martin Niemöller, 1946

"Then they came for the Jews

And I did not speak out

Because I was not a Jew"

"First They Came" - Martin Niemöller, 1946

"Clubbed Him to the Earth" by Charles Raymond Macauley

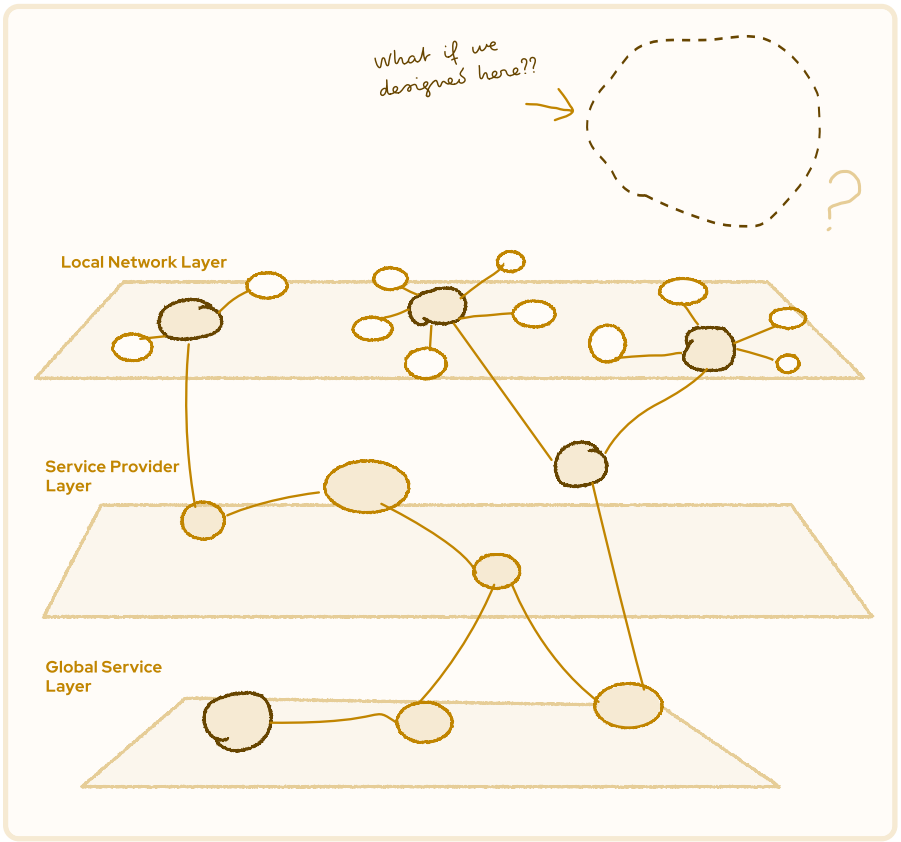
What Stands in the Way

Establishing an alternative outside the grips of government or those with vested interests is not simple though. Calling it impossible is perhaps more fitting. Our society as it stands right now relies on datacenters for survival, a consequence of the cloud computing era, no matter where you look today you will a system crucial for our survival that would crumble without the internet.

And yet the datacenters powering this vital internet infrastructure are being held to less and less scrutiny over time. The recent AI bubble has seen a massive increase in datacenter builds, with every larger footprints, bigger environmental impacts and energy consumption. For example, in 2024 datacenters made up 31% of metered power electricity consumption in Ireland, and that was before the AI building boom. An investigation by The Journal found that between 2019 and 2024, these datacenters produced over 135,000 tonnes of CO2 using fossil-fuel powered generators after their consumption surpassed what the electricity grid could handle.

What's crucial to point here though is that these datacenters are not just for hosting website or storing data, the very fabric that makes the internet function uses them. INEX, Ireland's primary internet exchange that connects us to the rest of the world, is powered by three datacenters across the country. Last year alone they handled 2.3 exabytes of traffic. While INEX is an industry-owned association, it is still a monolithic bottleneck in Ireland's access to the internet. Things like censorship or surveillance become trivial at even the local Internet Service Provider (ISP) level, let alone the national level.

Efforts to circumvent environmental, censorship and surveillance concerns largely come face-to-face with the reality of how the internet functions at it's core. It does consume energy, potentially considerable amounts, to transmit data across the world, and doing so opens us up to censorship and surveillance. Is it really necessary for us to use all this infrastructure to just text someone nearby? Is there anything that can be done to create resilient, sustainable networked connections?



A diagram of the internet as a whole.

" The decentralised networks and the communities they serve have never been more vulnerable. The peer-to-peer community is dangerously unprepared for a crisis-fuelled future that has very suddenly arrived at their door. "

"This if fine: optimism and emergency in the p2p network"
- Cade Diehm

A User-Centred Internet

The internet since it's conception has been designed to share information freely, many times though this has brought about clashes with both capital-centric industries and authoritarian governments. This has led to the creation of numerous different internet protocols, the ways in which our computers talk to each other, that sought to circumvent things like copyright laws and censorship.

Cade Diehm, a researcher and author who founded the independent research group the New Design Congress, coined the term "protocol activism" to describe the movement behind of these politically-charged internet protocols in his essay "This is Fine: Optimism and emergency in the P2P network".

These movements can be broadly described as attempts to build a truly user-centered internet. For example, a key selling point of the Dat P2P network was the ability for users to create and share a website without ever needing to deal with servers or domains. Similarly, the pirating era of P2P seen with the likes of Napster and BitTorrent orginanlly aimed to allow users to share their files with others around the world to spread information.

The Situation

However, the user-centric internet these protocol activists have continually pushed for over 25 years has never really manifested. For a time, torrenting held a sizable portion of internet traffic but ultimately it fell like all the rest to law enforcement and shortcomings in each network's designs. Torrenting for example is notorious for broadcasting each user's IP address (effectively exposing the exact location of a user). The risks of this behaviour was never communicated to end users, and many found themselves at the end of copyright violation lawsuits for hundreds of thousands of dollars.

Diehm points out several other examples, like the Dat network users being exposed to similar risks, or Secure Scuttlebutt being incapable of removing anything from the record. Ultimately most – if not all – user-centred, defiant internet protocols over the past two decades have failed to truly protect their users from authoritarian overreach within the "big tech" paradigm.

But can we actually design user-centred networks that protect it's users from prosecution for who they are? The internet itself brings many layers of abstraction to the equation, server owners, internet service providers (ISP) and government agencies have the potential to intercept communications at any point. Morals may be holding this back from being abused, but as we have seen in the states – and many other countries – it does not take much for obstacles to be removed.

So what if we scaled back? What if we removed the abstraction, the ISPs and so forth and built a network from scratch? How could we design to empower people to communicate freely through digital means on their own terms?

"This is Not Fine" by KC Green

The Situation

"Then they came for me

And there was no one left

To speak out for me"

"First They Came" - Martin Niemöller, 1946

Sources

"Counterterrorism" Now Officially Means Targeting Trans People - Sophie Hurwitz @ Mother Jones

Apple pulls US immigration official tracking apps - Liv McMahon et al. @ BBC News

This is fine: Optimism and Emergency in the P2P Network - Cade Diehm @ New Design Congress

Data Centres Metered Electricity Consumption 2024 - Irish Central Statistics Office

Ireland's data centres turning to fossil fuels after maxing out country's electricity grid - Conor O'Carrol @ The Journal

Internet Neutral Exchange Association website

02 Defining Solar Space

"I'm within three blocks of George Floyd Square. During that time, we had created a neighborhood chat, like Signal messenger, with three blocks by three blocks' worth of people, because there was a lot of activity."

Andrew Fahlstrom in interview with Samantha Michaels of Mother Jones

Protecting Communities

It's not much of a surprise that much of the issues we see in the world today are linked to one another. The rise of authoritarian and fascist politics being aided by technology giants through manipulation of users on their social media platforms that are powered by huge data centres destroying the environment. The people I am designing for are the communities being impacted by this vicious cabal, community not just in the sense of a local area but groups of relations or like-minded individuals.

How do we design to protect these groups from the manipulation, surveillance and abuse awaiting them through conventional, popular social media? What methods, tools and solutions can we use to ensure a level of privacy that can't be achieved right now?

"The Death Blow" by Thomas Rowlandson

The Problems

The research has highlighted three issues that I find to be at the core of what I wish to address with Solar Space.

01 Internet Surveillance

This decade has seen some of the biggest regressions in privacy protections, especially in the digital realm. The internet has too much complexity in it's structure to insure your messages aren't being intercepted. Surveillance online, especially by governments, chill free speech substantially.

02 Oppression of Marginalised Groups

Immigrant, transgender and queer-identifying groups have been coming under increased oppression and attack. Digital spaces they have used to communicate and share information are under increasing threat. The destruction of anonymity online is especially dangerous for these groups.

03 Over-reliance on Datacentres

Our society has a major over-reliance on Datacentres for our digital services. These structures are having a bigger and bigger impact on our environment with each year. More often than not, they are owned and ran by just a handful of companies that exact control over the content that exists on their servers.


FIGHT TODAY
FOR A BETTER
TOMORROW

" We need to lay aside our delusions that decentralisation grants us immunity – any ground ceded to the commons will be met with amplified resistance from those who already own these spaces. "

"This if fine: optimism and emergency in the p2p network" - Cade Diehm

Design Requirements

The role of design in this project is ultimately one of asking questions about the users. Who is this for? How will they use it? How best can they be supported? What risks to them can be avoided? The work in Solar Space is largely a programming venture, one in which the final outcome is a functional, contextual application. Many of the programming decisions were made through the lens of design. A number of key design requirements were identified early on that influenced these decisions:

01 Empowering through Communication

Any solution must directly empower a community through digital communication.

02 Be Inherently Private by Design

Privacy is crucial given the problems at hand, it should be essential to all outcomes.

03 Protect Groups at Risk

Marginalised groups are the most in danger when trusting any solutions. Their safety must be of the utmost importance at all times.

04 Be as Sustainable as Possible

Everything must be kept sustainable, this includes both the outputs and the work that goes into them.

Sources

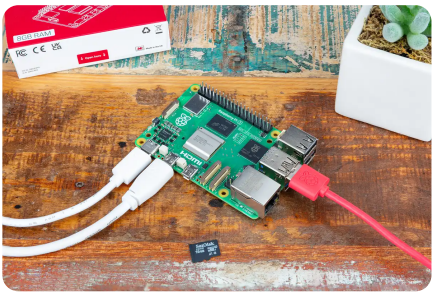
Meet the Minneapolis Neighbors Standing Up to ICE by the Thousands - Samantha Michaels @ Mother Jones

03 Technical Specification

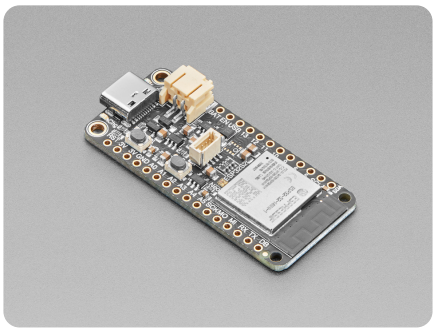
Defining the Electronics

Choosing what parts to use for the electronics of the project was decided largely by the needs of solar power. Since availability would be unreliable in Ireland, I needed a board that could last on a small battery for a long time and deal with power cuts gracefully.

Initial research ruled out many popular options like the Raspberry Pi and similar small, linux-powered computers. The projects requirements called for a much cheaper and low power solution. I narrowed the options down to the Pi Pico W and ESP32 microcontrollers. Both can be found for as low as a few euro, have sleep modes to reduce power consumption and have wifi functionality.



A Raspberry Pi 5 Computer



Adafruit's Feather version of the ESP32 S2.

After thouroghly comparing both boards, I landed on the ESP32 S2 variant as the projects core. Compared to the Pi Pico W, the ESP ecosystem had far documentation, especially on low-power modes, and has a much more vibrant community with guides, tutorials and projects to learn from. I settled on purchasing an Adafruit feather version of the S2 after getting advice from a tutor on the unreliability of cheaper, no-brand options.

A suitably capable solar panel was chosen from Adafruit's catalog, as well as a charger capable of handling the irregular nature of the incoming power. The charging board was chosen mostly from the requirements of the battery. As I intended this project to be outdoors, I felt using a Lithium Phosphate battery would be better than the Lithium-ion or Lithium-polymer batteries that are most commonly used. I concluded from my research that LifePo4 batteries were far safer and more ethical to use, lacking the cobalt of other battery types that is commonly mined by child slave labour. The bq25185 is the only breakout board for microcontrollers available in hobbyist stores that can take solar power and charge LifePo4 batteries.

Parts were sourced as locally as possible, the batteries and their holders as well as the SD card breakout were sourced from within Ireland. All other parts were ordered from the UK and shipped over.

Part	Price
Adafruit ESP32 S2 Feather	€19.40
Adafruit bq25185 Solar Charger	€7.75
Voltaic P105 - 6V 5W Solar Panel	€38.87
Generic 18650 Battery Holder	€1.23
1800mAh 18650 LiFePO4 Battery	€15.10
Generic Mini SD Breakout	€2.15

Technical Specifications

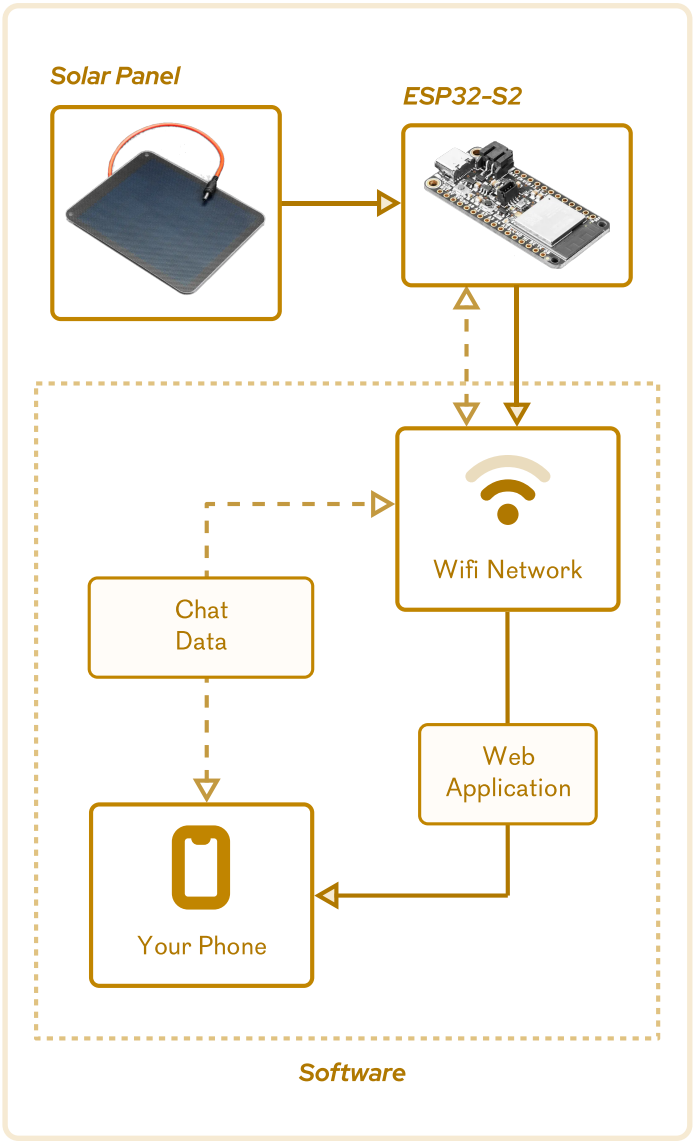


Diagram of how the system functions.

Detailing Software

A core part of the requirements for the Software was to facilitate real time communication between users. Initially, I explored the posibility of using a peer-to-peer framework however the effort required to realise it was far outside my expertise. I looked instead towards building a HTTP web server and building the software as a web application.

While a simple approach was first attempted for serving HTML files, I quickly discovered that an asynchronius approach – one in which the server handles requests in the background – was the only viable approach for serving files from the relatively weak ESP32.

In order to realise a functional messaging system, I looked to web APIs for real time communication. While the aptly named RTC API seemed ideal at first, it's limitation to secure connections and lack of existing libraries for the Arduino IDE made it difficult to work with. On the other hand there was the mature websockets protocol, which maintains a constant connection between the client and server. It's ease of use and existing support in libraries I was using for the server

In order to maintain a small file size of the application for efficient power consumption, I used only basic Javascript without frameworks since they are typically large and bulky. This at times made programming a little challenging, but it remained manageable with research and understanding.

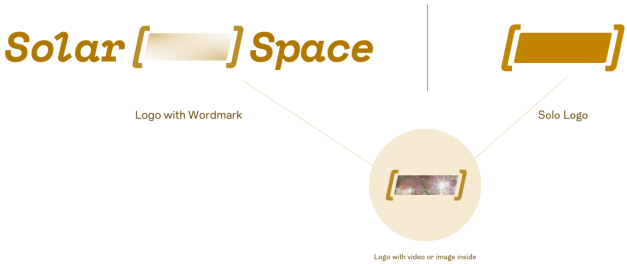
Comforting Design Language

One of the first things I worked on for the project was a design language. I had noticed that many similar projects, especially those focusing on sustainability, often had quite harsh design languages. My aim was to create a warm and welcoming environment around the Solar Space, and so I chose colours with a warm temperature alongside soft, friendly fonts.

All of the fonts used in the project are open source. The logo and certain UI elements are set in Space Mono, and character-ful typeface that creates the core identity. Body text is the friendly Cotham Sans, while headers can be either Red Hat Display or Rumeur depending on how sharp they need to be.

The logo itself represents the Solar Space the project creates, and mixing point. It has the added functionality of providing a frame for imagery when used in certain contexts.

The design language continues into the User Interface design of the applications, where shapes with rounded edges are used often and there is plenty of space to breath provided.

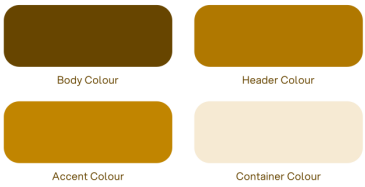


Rumeur — Header 1

Red Hat Display 800 — Header 2

Space Mono Bold Italic — Subtext

Cotham Sans — Body



Type for Print

Lorem Ipsum etc — 32pt

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut. — 13pt

Type for Digital

Lorem Ipsum etc — 28px

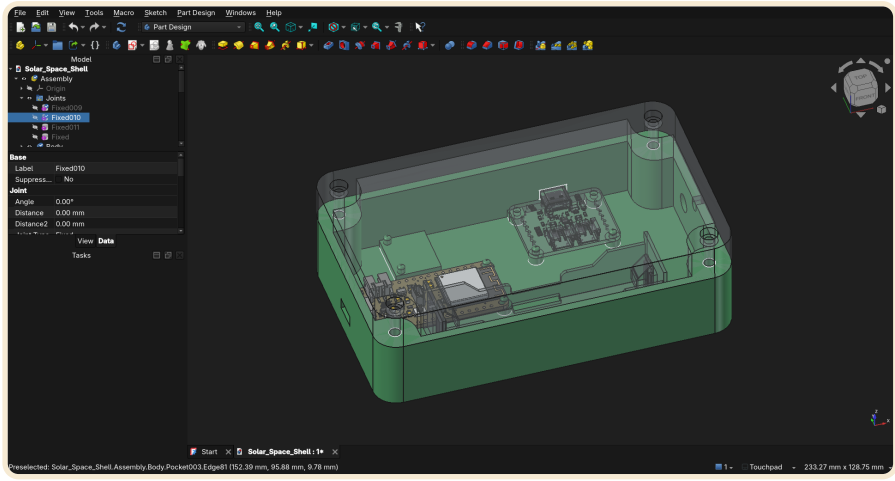
Lorem Ipsum etc — 22px

Lorem Ipsum etc — 21px

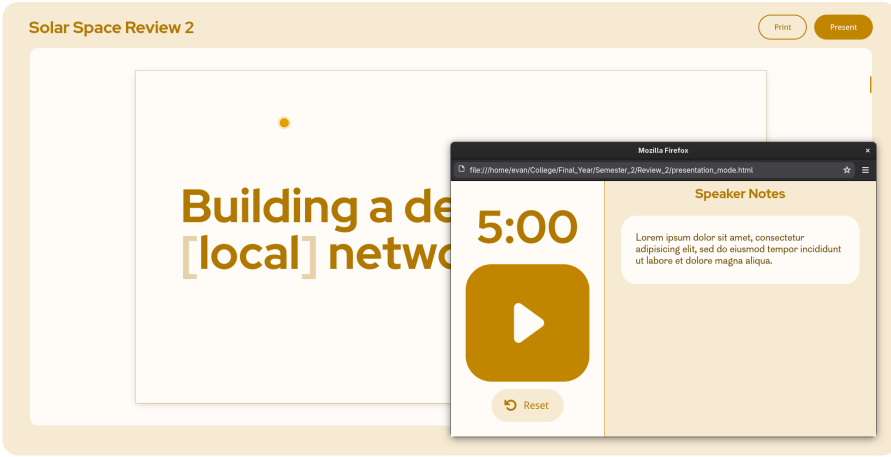
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut. — 18px

Style Guide document.

Technical Specifications



FreeCAD in use, with the shell assembly.



Straw, the templating presentation slides tool.

Open Source Only

Since this project was intended to be an open source, shared to the world free to use and modify, I decided to only use open source tools to create it. For the most part this was simple enough as I already used an open source operating system with Linux, and tools like Inkscape were already a part of my regular workflow.

Some challenge came with 3D modelling parts for 3D printing. My CAD experience is almost exclusively in SolidWorks, and while I had tried to use alternatives in the past I struggled hard. However, picking up FreeCAD again after it's recent improvements I found myself modelling with ease almost immediately.

In the circumstance that a tool did not have a sufficient open-source alternative, I resorted to building my own. I ended up building two tools for this project, a template-centric presentation tool and a similarly template-centric book designer. Both were built using HTML and CSS and follow a philosophy of designing easy to read HTML writing systems using custom tags.

Hardware Design

As part of the project's exploration, a number of different hardware form factors were designed. Each one imagines a different way to use and deploy Solar Space in the real world. Open source tools were used to create these, all consisting of laser cut frames combined with a 3D printed electronics case.

01 Window Mount Frame

This frame allows you to mount a Solar Space module in a window. Suctions cups are used to keep it attached. There are also some slots with which you can attach rope, zipties or other sorts for alternative mounting.

02 Pole Mount Frame

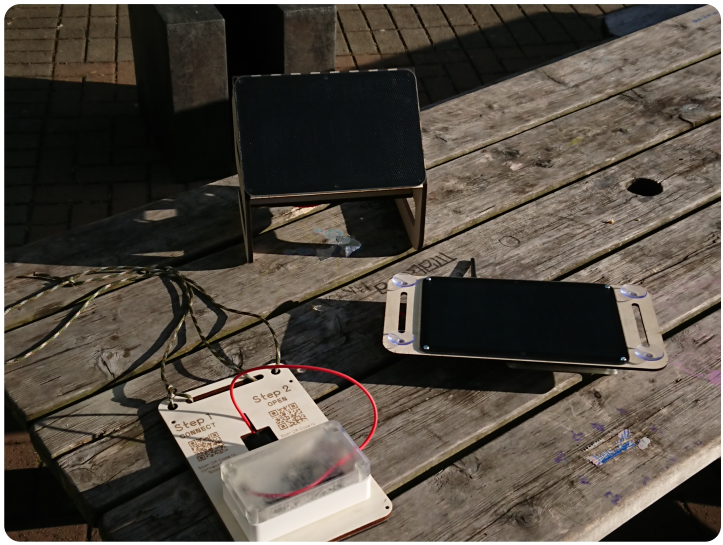
This frame is designed for out-door use. It features a sloped roof design to disperse rain and several options for zip ties or similar attachment methods on the back.

03 Ledge Mount Frame

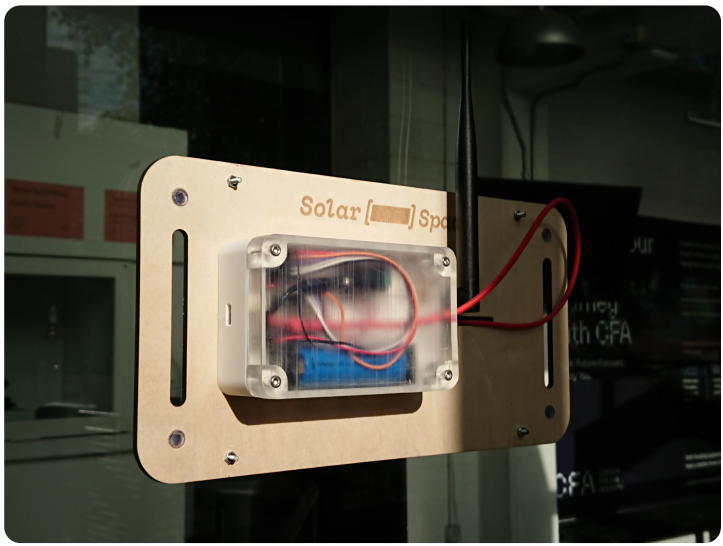
This frame is similar in shape to the Pole Mount but features a hook shape design for hanging on ledges like those found on balconies. It can also simply be placed on a flat surface.

04 Paracord Frame

Lastly, the paracord frame allows you to wear a Solar Space module, attach it a bag or simply hang it wherever. It's similar in design to the window mount frame but with options only on one end.



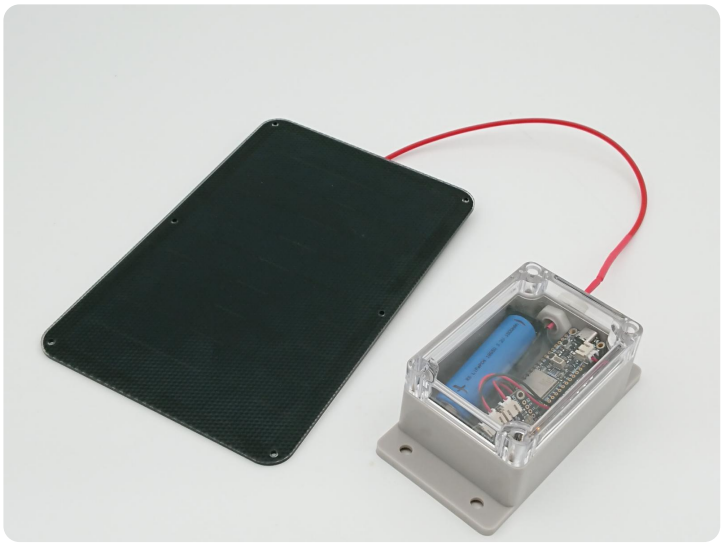
Three of the hardware frames.



Back of the Window Mount Frame.

04 Iteration One

The first prototype hosting a simple website over Wifi.



Solar circuit being charged for the first time in a light box.

Viability Prototyping

In order to confirm the viability of my initial approach, I first set out to build the basis of a HTTP web server on the ESP32. This iteration created a Wifi access point and served the web files as a captive portal once a user attempted to connect. The LITTLEFS filesystem was used to store website files on the ESP32's flash memory, however the serving of each file was done manually by intercepting requests and responding with the desired file for each request needed to render the page correctly.

This prototype ultimately served it's purpose of communicating the technical basics of Solar Space well. It did however highlight a number of issues with the approach, notably the captive portal failed to work on certain kinds of devices. This appears to be a known issues with how captive portals work as there is currently no standardised way to do such things. Another issue was the manual nature of serving files, which was prone to errors and took time to setup especially for larger files. However for the core purpose, it was considered a success and these issues were noted.

Sources

ESP32 Web Server using LittleFS Filesystem by Sara Santos @ Random Nerd Tutorials

Captive_Portal_LittleFS_HTML.ino by yash-sanghvi on Github

05 Iteration Two

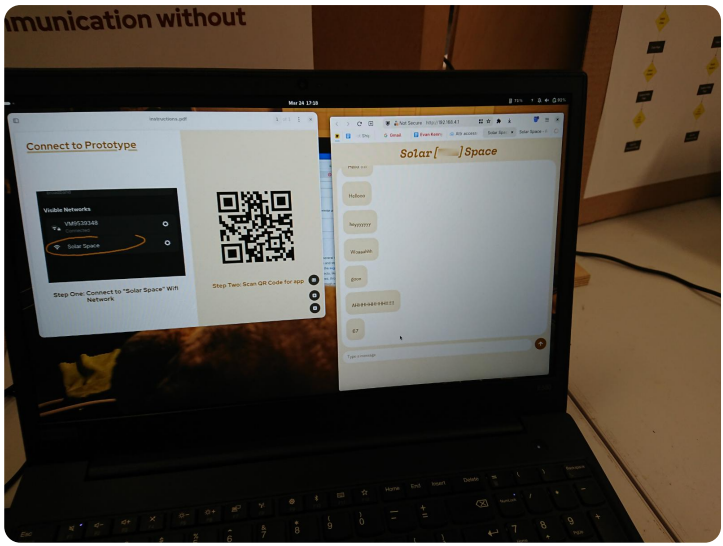
Experiential Prototype

My second prototype aimed to create a basic chat application in order to test the software and get feedback on the concept. The prototype involved a more complex rewrite of the initial server, this time serving HTML from a C++ Header file. Chat messages were sent as POSTs from the client which were added to a variable in the firmware and appended to the HTML when it was being sent to a user. It had a basic user interface with messages in a familiar display to most modern chat applications.

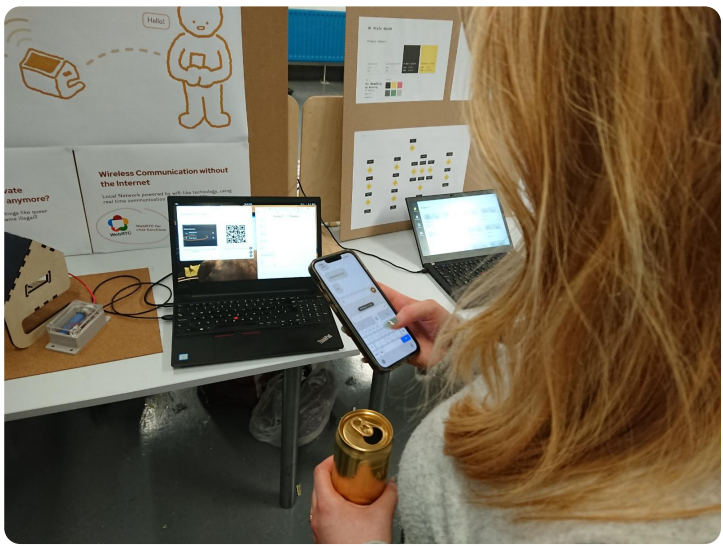
Much of the code for this prototype was based on the guestbook code from the brilliant server charms workshop by feminist computing group actimony. The idea of attaching received messages onto the bottom of the page after a GET request in particular made this experience possible. However it strayed from their work in regards to using captive portals, from my experience with the initial viability prototype I explored using the IP address to the server in a QR code to access the website.

I gained valuable insights from observations and conversations with people using this prototype. While things like usernames and live refresh didn't work, the ability to send messages and see them received on each others devices was enough of an experience to inspire people. The privacy and activism aspect of the project had a strong resonance with many and I received a number of stories from others. A group of friends enjoyed people able to post memes and "shit-posts" in the chat, however there were some surprised reactions from one or two people not in said group of the content of the messages.

This prototype also proved to be helpful in diagnosing some key issues of the approach made in firmware. Due to the method in which the web server was written, the ESP32 could only handle 3 connections before it would start rejecting further attempts to connect. Many users did not disconnect after leaving a message and ended up rendering the prototype inaccessible for periods of time. Research afterwards suggested using an Async web server would prevent this issue.



Connection Instructions beside the chat feed.



A person connecting to the chat and leaving a message.

Sources

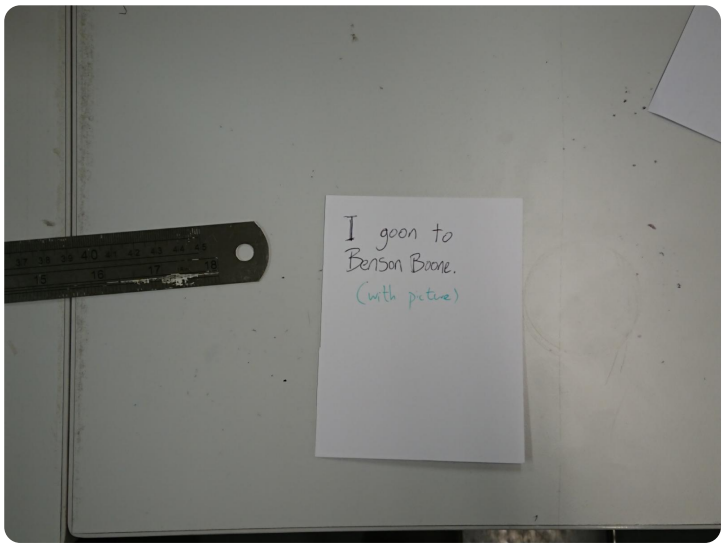
Server Charms Workshop by Actimony on Codeberg

06 Iteration Three

Slow-cial Media Prototype

For the third prototype, my goal was to create the foundation of the final approach. This involved rewriting the the web server and changing the storage method to use SD cards. As part of this prototype, I organised a small group of people who were friends to simulate the function of Solar Space as a digital social space. Since I did not yet have a method of allowing users to post, I gave each person a card and instructed them to write whatever they wanted to share on it with the option to leave their name or not. The following day I inserted the messages into a webpage, set up the prototype in a window and gave each participant instructions on how to connect and view the messages.

Since this was a group of friends the messages were loose and casual, spread evenly between posts of in-jokes, ranting and more normal messages. There were quite positive reactions from all participants, though one expressed anxiety about the privacy of their post and had questions around the possibility of outsiders being able to view it.



A written message before being put on the app.

Developing Firmware

Writing the server code for this iteration, I wanted to design a generic system that could allow more rapid prototyping which the ease to quickly change the user interface and structure of content. I settled on the ESPAsyncWebServer library for the Arduino IDE, which would solve the user bottleneck issue while hopefully providing an easier way of sharing files.

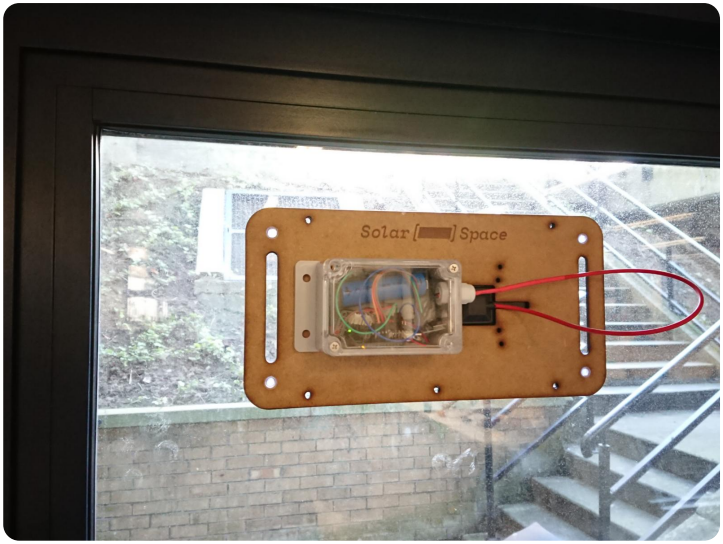
Initially I read guides like those from Sara Santos of "Random Nerd Tutorials", which recommended a method of manually intercepting requests for files like what my viability prototype used. However after spending some time reading ESPAsyncWebServer's documentation, I discovered it had a single line method for started a web server and serving files from a filesystem.

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(LittleFS, "/index.html", String(), false, processor);
});
```

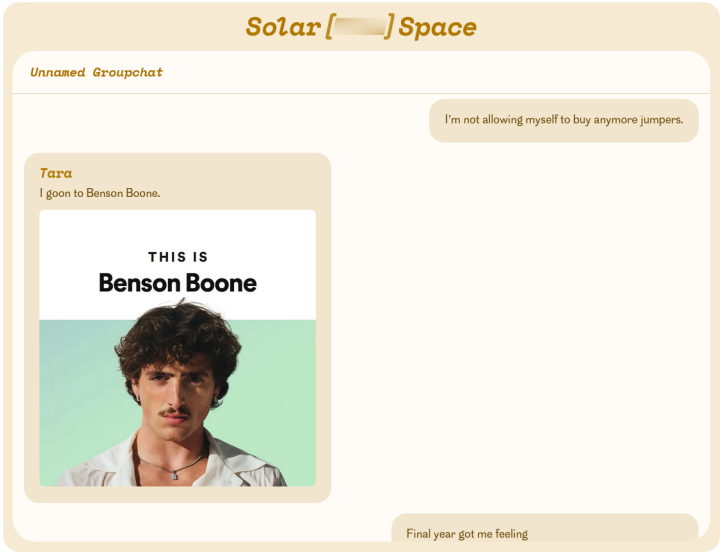
```
server.serveStatic("/", LittleFS, "/").setDefaultFile("index.html");
```

Top: Code Before. Bottom: Code After.

This greatly simplified the serving of files, automatically returning requested files without having to manually declare each. Then to hook up the SD card as a filesystem I simply had to initialise the card over SPI and then swap out "LITTLEFS" for "SD" in the web file serving code.



The Solar Space module charging in a window.



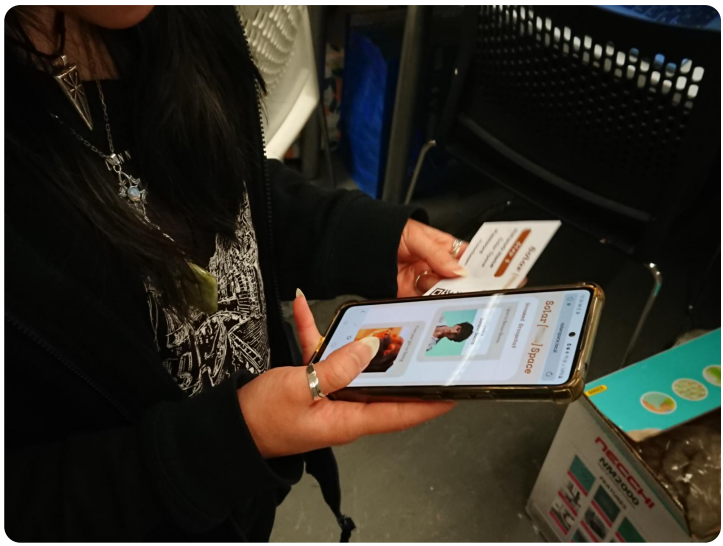
A screenshot of the message feed.

Participant Feedback

I gained good insight into the onboarding process here since moving away from the captive portal, connecting now had two steps, connecting to the Wifi and opening the web page. Instructions provided to participants had the Wifi access point name and password in text, with a QR code for the website. The vast majority of people went to scan the QR code and ignored the Wifi instructions. I questioned a user about this and discovered a lack of description for each process and difficulty understanding the order of the steps.

In an interview afterwards, a user stated that the anonymity of the posters had no impact on the experience for them or detracted from the purpose of the content. However they continued that if they were to use the chat to organise plans, they would prefer to know who each person was. There was an instance of one participant using the name of another participant in their message, however this was a common joke in the group which the impersonated person was in on. Another interviewee found the messaging style layout confusing, with messages being displayed on the left and right they said it seemed at first that the messages on the right were theirs.

This iteration provided a good early evaluation for the design of the firmware and external communication. The was a clear lack of confidence and understanding in how private the network was, but this was contrasted by an enthusiasm to use it as an alternative to the groupchats the participants already had.



Scanning and accessing the chat.

Participant Writing a message.

Sources

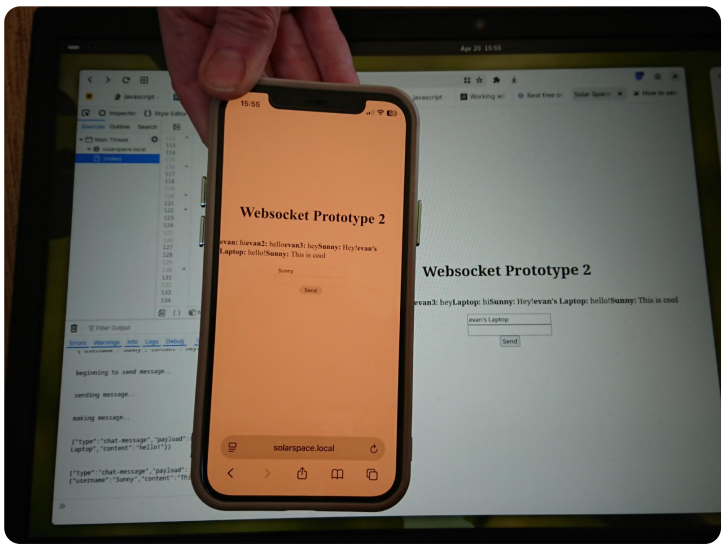
Serving Static Files - ESPAsyncWebServer documentation

07 Iteration Four

Minimum Viable Prototype

The next iteration of Solar Space exclusively focused on understand and establishing a functioning websockets prototype. The websockets protocol allows for a constant connection between two devices with data transfer possible in both directions. Further research of the ESPAsyncWebServer library documentation led me to it's plugin that appeared to provide a simple server-side setup.

Building the client side application proved to be a lot more difficult, the ease-of-use libraries I researched all required bulkier javascript frameworks that I wanted to avoid for the sake of the server's battery life. An introduction to websockets on Beej's "Bit Bucket" blog proved to be incredibly helpful learning how to structure the javascript for websockets. The first messages were simple strings of HTML that got sent between clients and appended to the body once received.



The first messages sent through the websocket API.

Understanding JSON

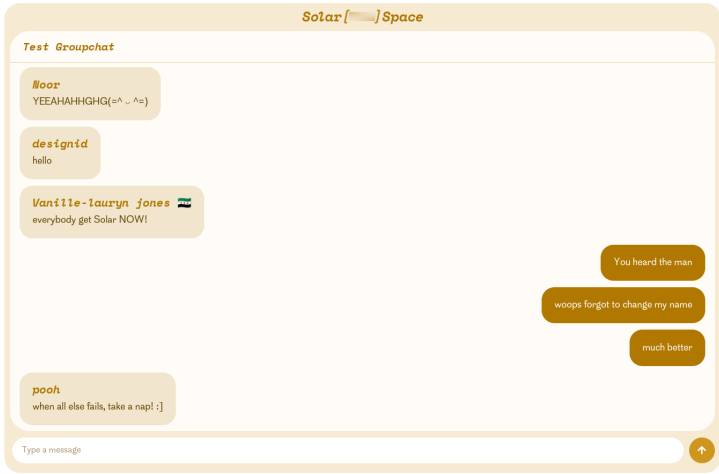
To expand my initial websocket prototype into a more featured chat app, I knew I would need to encode more data into each message that was sent. At first I struggled as I had no previous experience working with complex data, however after researching for a number of days I came to the conclusion that Javascript Object Notation (JSON) was the best approach for what I needed.

Understanding how to use JSON proved to be quite challenging, it took me some time to understand how to properly structure a JSON string. Similarly figuring out how to parse a JSON string took a long time. Early on assumed that the use of double and single quotes could be used interchangeably in a string much like it can in programming languages I am used to, however this was not the cause and caused a lot of confusion until I eventually did deeper debugging and discovered my mistake.

Throughout this learning period, the Mozilla Developers Network (MDN) documentation pages were unbelievably helpful in understanding and working with JSON. Eventually leading to my first prototype using websockets and JSON, a modified version of the early experience prototype application.

```
{"type": "chat-message","payload": {"username": "Solar
Space","userid":"1234567890","content": "Welcome to Solar Space!"}}
```

What a typical message JSON string looks like.



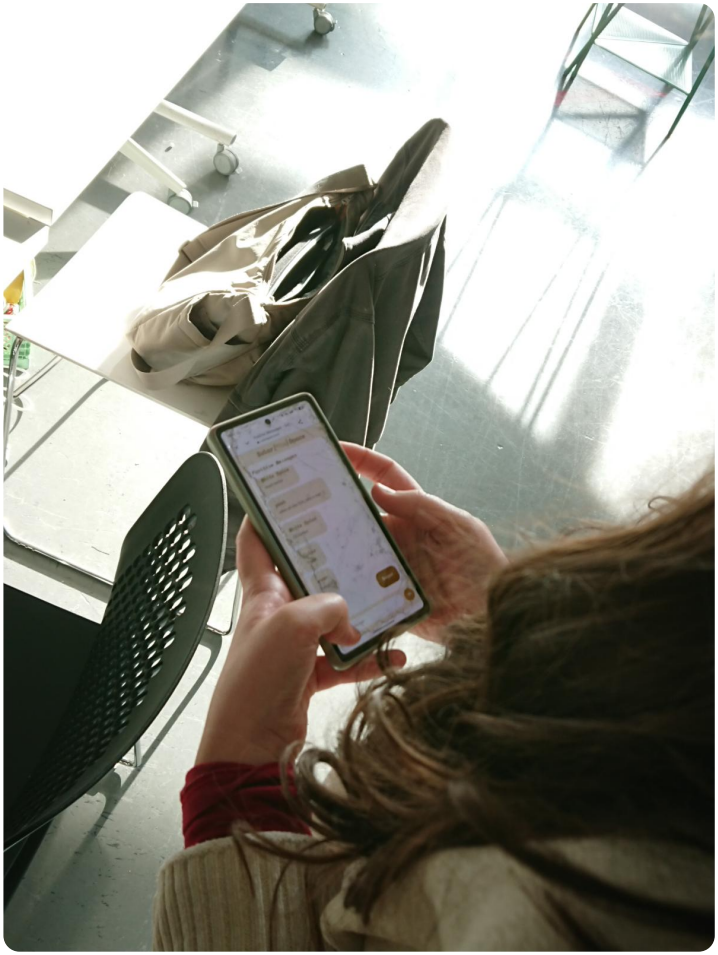
The MVP in action.

Building Better Chat

While websockets and JSON created functioning message transportation between users, when they first log on or return to the application there would not be a history of messages. From short interviews with 3 people of a mixed ages, I found chat history to be an important feature for this application. To achieve a functional log I return to another Sara Santos' post exploring the filesystem functions for reading and writing on the ESP32. I borrowed their text-appending code to create a simple chatlog.txt file which stored all messages in a long JSON array. When the application was loaded, it would request the file from the server and parse it into the chat as regular messages. This proved both simple and effective.

In order to create distinction between messages sent by the user and those sent by other users I needed to generate IDs. When a user creates a username, a random 10 digit ID is created and stored too. When a message is sent this is attached within the JSON and whenever a message is received the client checks it's ID to see if it matches with the users ID. If it does, then a "my-message" class is attached and the message gets rendered on the right side without a username.

Iteration Four

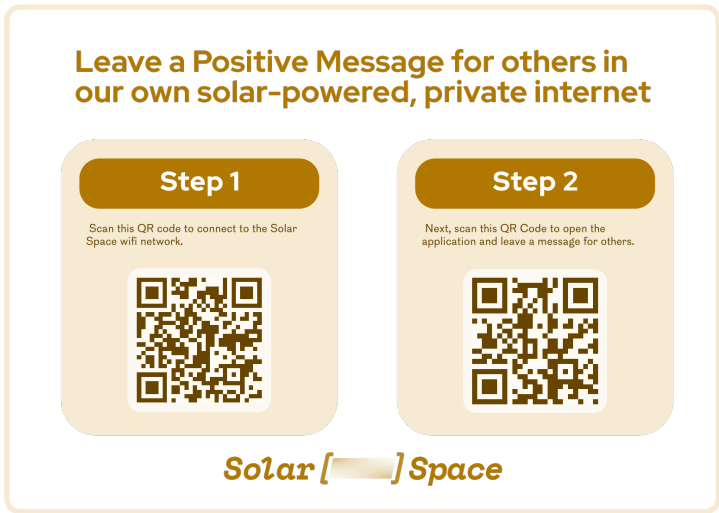


A person connecting to the chat.

Testing the Iteration

I initially took a much broader and loose approach to evaluating this prototype. My goal was to gauge more reactions to the concept while also stress testing the software systems. To do this I went up to people I knew around the college and got them to test it in small groups at a time. This created some fun interactions and insightful conversations around privacy and offline technology. Following up on this, up created some posters to stick up near a Solar Space module to invite people to interact and leave a positive message.

While one group did interact from it, this poster approach ultimately failed to generate much engagement with the digital space. After some observations and a discussion with a Graphic Designer, I concluded that the poster was too neutral and did not stand out. My takeaway from this small rollout was that my communication to potential users needed to be much stronger in printed material. From a technical standpoint, the testing was incredibly useful. The prototype functioned smoothly, with the only minor drawback discovered being the connection range for the Wifi network.



The poster design.

Sources

Intro to WebSockets - Beej @ Beej's Bit Bucket

ESP32 WebSocket Server: Control Outputs - Sara Santos @ Random Nerd Tutorials

Writing WebSocket client applications - Mozzila Developer Network

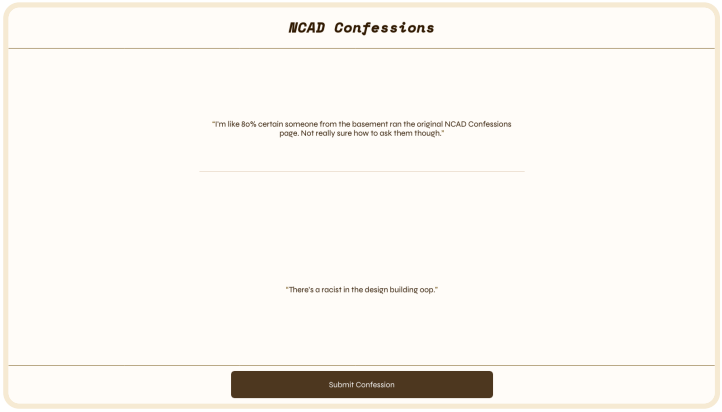
JSON - Mozilla Developer Network

08 Iteration Five

Going Provocative

Wanting to explore the communication aspect of the project more after the previous poster failure, I decided to build a version of the previous websocket prototype with a redesigned interface for a more provocative campaign for engagement. While it would be contributing towards my understanding of the project, this iteration was seen more as an experiment than a legitimate direction from the start.

I drew inspiration from the numerous anonymous "confession" pages that have been popular in the university for the past several years, and created a similar experience through the Solar Space module. This version did not store usernames or IDs, instead going for an almost exaggerated approach to private social media. I also changed up the visual style to be me more stark and simple, removing any of the Solar Space branding.



The only two messages sent.

The poster placed around the campus.

Poster Engagement

Ultimately the application itself saw no submissions or any discernable engagement that I was aware of. However the experiment itself proved successful in other ways. I designed a large, bold poster with the goal of drawing attention easily. Seeing a lot of people with disdain for MetaU's social media platforms, and often directing that ire at it's CEO Mark Zuckerberg, I gave the poster an obscene title directed at Zuckerberg and Instagram. This was highly effective at catching eyes, I was told that a Graphic Design tutor brought it up in a tutorial as a good example for drawing attention, and generated the most conversation so far.

Much of the engagement however came either from conversations with people who knew me or was passed on to me by those who overheard people talking about the posters. Those I spoke to about it found it interesting and agreed with it's anti-Instagram message, many of whom were unaware that it was part of the Solar Space project. A number of people said they couldn't get it to work, and while impossible to know the cause for sure, I deduced the reasons to be either lack of power due to shade, incorrect order of QR code scanning on the poster, or operating system quirks when connecting to offline Wifi connections.

On the other hand, much of the overheard conversations around the poster were less in support of the approach. It appears some people felt anxious about the total anonymity of the application, highly valid concerns given the lack of moderation controls and several thousand students of the college having access to the space.

"It's a cool idea. I'm kind of sick of using Instagram for things but I have friends that use it for messaging who I don't want to lose contact with."

Workshop participant

A Stronger Perspective

While the lack of any submissions made it difficult to provide a hard metric for success, the lessons learned from this experimental iteration were incredibly useful. I gained a much deeper insight into how people felt about access to this kind of private social media in public settings, certainly the anonymity made some uncomfortable engaging at all while others found it freeing. The successful poster design also further motivated me to invest time in the graphic design end of the project in the future. I sought some feedback from Graphic Designers in the college on my initial design and gained some helpful feedback which I carried forward by making the next iterations more characterful.

A "Big Brother" poster someone placed beside the Confessions poster.

09 Final Concept

"I'd probably start using it with people I know.. Familiarity first and then expand out to other people.. Build the community with it."

Eoin Murphy - Workshop participant

Tools for Community

This final concept is more a proposal than a completed physical iteration like the previous work, this is as a result of the work required to realise it taking far more expertise I currently have. Nonetheless, I have created a mock-up application with much the same experience one would have if it was fully functional

I undertook two short workshops in which I introduced the project, running participants through the functioning chat prototype and discussed topics around privacy, socialising and government overreach.

In these workshops and interviews I asked people to envision the project being used by a group, or how they would use it themselves. There was an overwhelming theme of care and empowering others, which inspired me to expand the set of tools to provide more specific ways of sharing information and communicating.

One of the workshops.

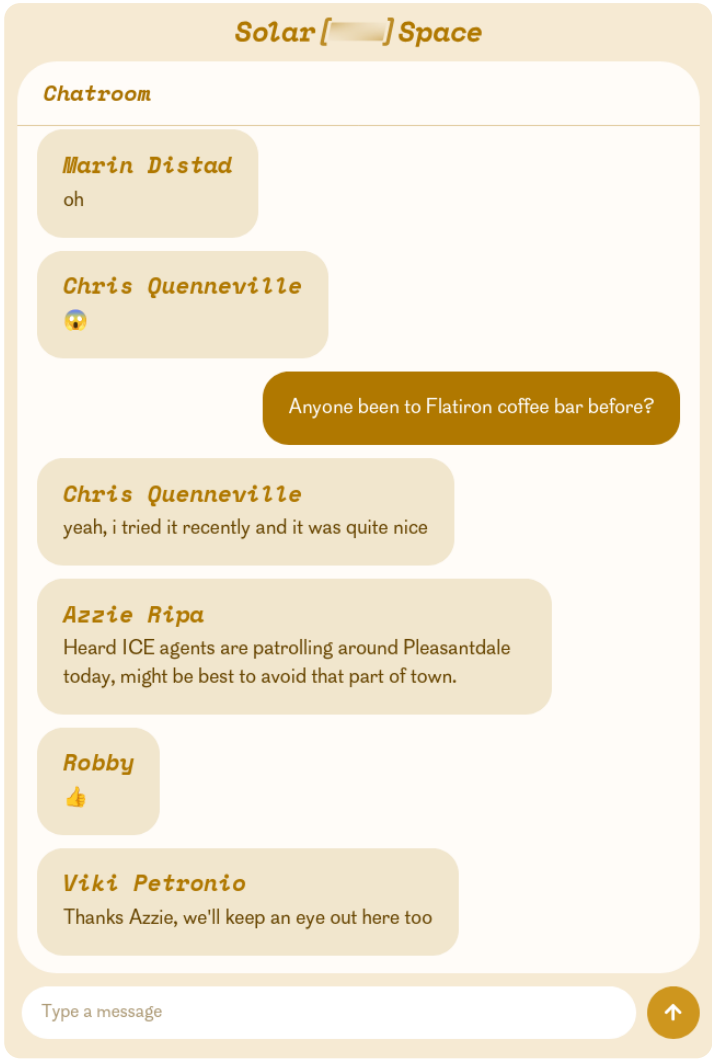
Live Chat Room

The first function to cover is the live chat application developed in the previous iterations. For the most part it remains familiar with some additions to improve it's functionality and safety.

A major feature lacking so far has been the ability to moderate the content. The potential for abuse of anonymity in the service was highlighted through interviews, and the need for moderator or administration roles to delete messages or remove users is a necessary for a community tool. To achieve this would require the addition of user authentication, especially in a manner which could not be hijacked, something which is outside of my abilities at this current time.

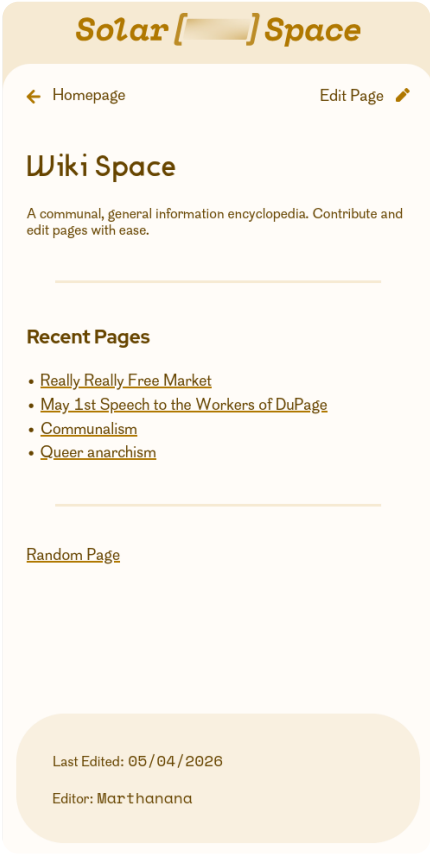
Another important feature that would benefit from authentication is a more mature user profile. Currently users cannot change their name, especially across all messages previously sent, which is necessary to prevent dead-naming or to remove identifying info from messages. This has been a core need mentioned in similar project, like that of the Willow protocol which attempts to succeed the Secure Scuttlebutt protocol which cannot rename users historically.

Lastly, the ability to send more kinds of media like images, audio or even video was identified early on as important. Currently the Solar Space websockets implementation can only handle messages that fit within one TCP frame. While this is still several kilobytes worth of data, to send images of any readable kind would require handling messages with multiple frames. The effort to do such again outside of my abilities, however I do believe it to be possible.

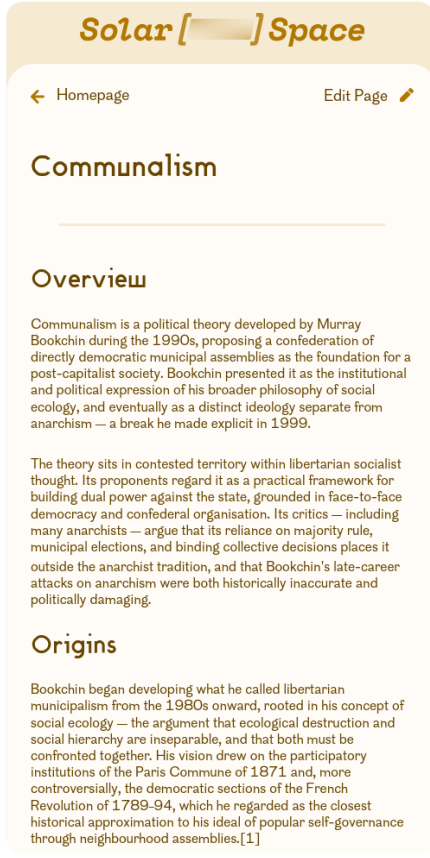


The Live Chat Application.

Final Concept



Wiki Space homepage.



Page about Communalism on Wiki Space.

Simple Wiki Tool

I was inspired to add a wiki tool to Solar Space after seeing so many unique communities online that ran their own with information specific to their group. This could host copies of wikipedia articles (or similar), or be host to information users of the server find important to share.

For readers, it would be a familiar experience with basic navigation between pages of content starting with a homepage listing recent edits and categories. However users can also edit any page or add a new one, this would open a simple text editor UI that would then send the edits over websockets before being processed.

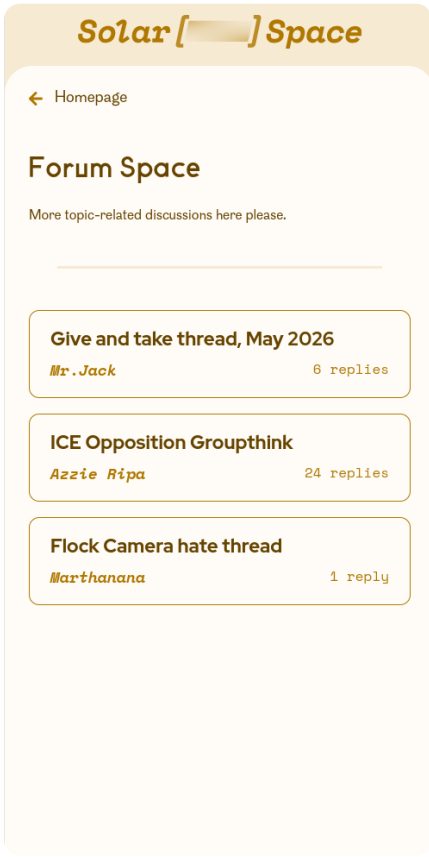
The most difficult part in realising this tool would be the creation of new pages. As of now the server code is very simple, but would require a rewrite in order to be able to parse incoming data and perform different functions depending on what tool it came from and what action was needed. I believe some form of JSON parsing may be possible, but this would require a lot more researching and learning to be achieved.

The final prototype (pictured left) includes a mocked up version of the wiki without page editing.

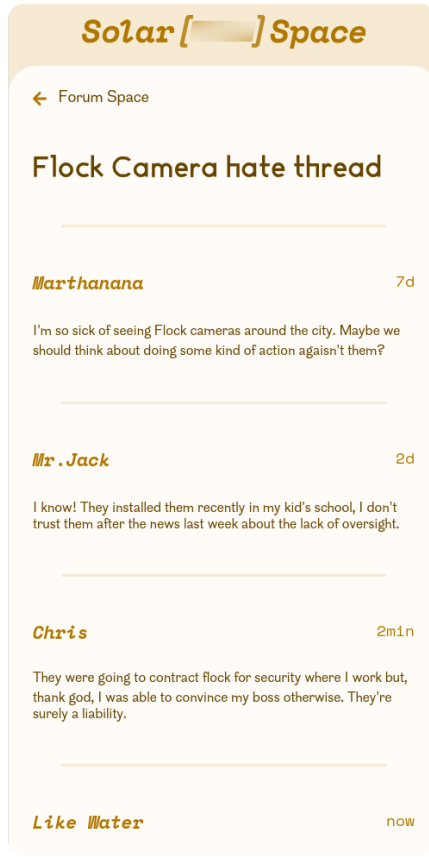
Forum Messaging Board

Another social space, the messaging board would be a place for questions, sharing and general discussion. Much of it's technical function would be similar to the Wiki tool, individual posts would be HTML pages created by server while comments would be appended to the page.

Inspiration for this came from seeing many online communities sharing items between each other that they no longer need. While I initially pictured creating a specific app for listing items you want to give away, I realised a more general forum would offer far more benefits.

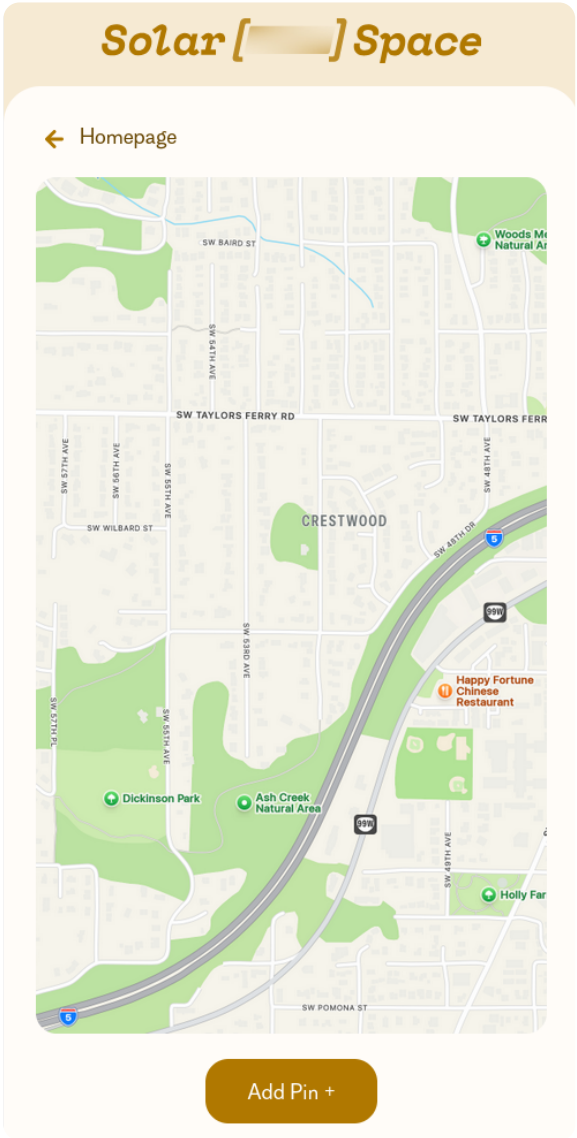


Forum Space homepage.



Forum Post discussing Flock Cameras.

Final Concept



Local Map page.

Local Points of Interest

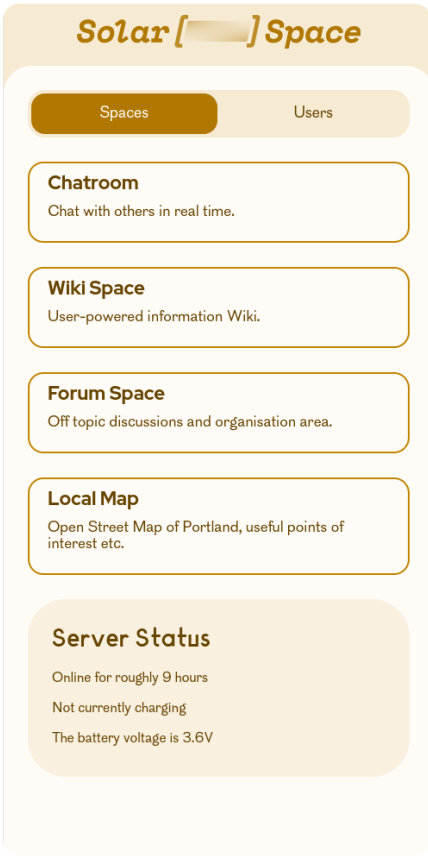
Lastly, the local map tool would provide a copy of the Open Street Map data for the local area. Users would be able to create pins for points of interest, perhaps a place where mutual aid has been left or a spot where ICE agents were recently spotted.

While downloading and displaying Open Street Map data is easy to do with SVGs, adding the ability to create and edit pins on the other hand would be much trickier. My research uncovered a number of possible solutions, like the fairly lightweight library Leaflet.js for rendering maps and pins. However I am unsure of how this would integrate with Solar Space's websocket editing system.

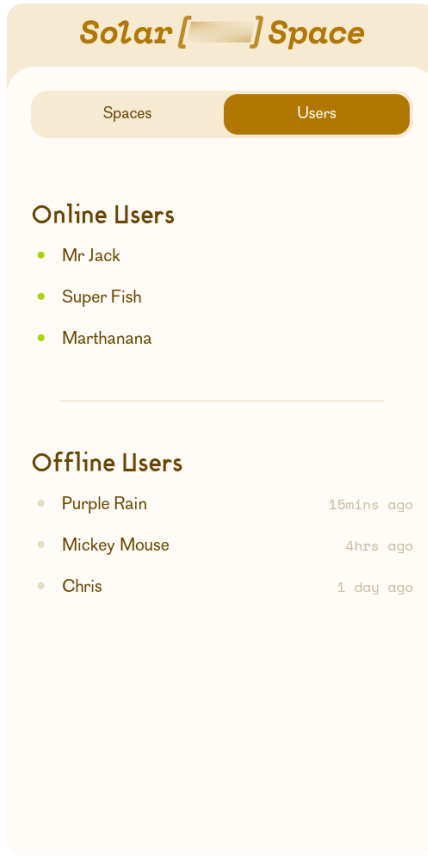
Onboarding Experience and Homepage

For the final concept I created an onboarding flow for new users that explains the general concept and prompts them to create a username. The first page is intended to be modified to describe the community running the server.

The homepage features a tabbed design with one tab including links to the various tools as well as the system status, while the other tab features the list of registered users. The system status gives information to users on the battery voltage, total time since last offline and if the battery is charging or not. The user list shows what users are currently online, as well as how long it has been since an offline user was seen last. Tapping on an online users name sends them a notification of a wave emoji, this feature I felt was necessary to make the service feel more alive to users.



Solar Space Homepage.



List of Online and Offline users.

Evaluation and Feedback

I took a broad approach to evaluating the final concept of Solar Space. Want to get general opinions from people within my target communities, I reached out to people I know or am acquainted with, held 2 small workshops to explain the project and discuss with people what they thought, and lastly posted on online forums.

The online forums received little attention in the places I was able to successfully post, and mostly regarded feedback on the technological design rather than the conceptual. The workshops were more successful and I received much more insightful feedback on how people would imagine it being used, what they felt were possible drawbacks or flaws and their feelings on privacy violations.

Most successful was reaching out to acquaintances and connections. I contacted two activists I knew from my time on Erasmus, with which I got great critique from and some resources to follow up on. I also had Roel Roscam Abbing, one of the architects of the Solar Low Tech Magazine, review my project with a scheduled call. Lastly I also contacted Durian, a pseudonymous digital activist formerly at DuckDuckGo now working on a small, communal digital services server.

Feedback was mostly positive of the aims, with many identifying with the activist tool aspect of Solar Space. There was also good critique of the overall system, notably communicating that it was not a "Mesh" device, or interconnected with other Solar Space modules.

"naah that's insane ! i love the idea ! ... my fellow protester would love it."

Tomi La Segue - Student from France

"Didn't look through your git too hard, though skimmed through and looks like a well thought-out project!"

stupiddogmademelook - reddit user on r/ask_privacy

"I love the idea of the Wiki.. I would use that for my communist businesses 😌"

Anna Alberti - Italian Communist Activist

10 Conclusions

Reflections

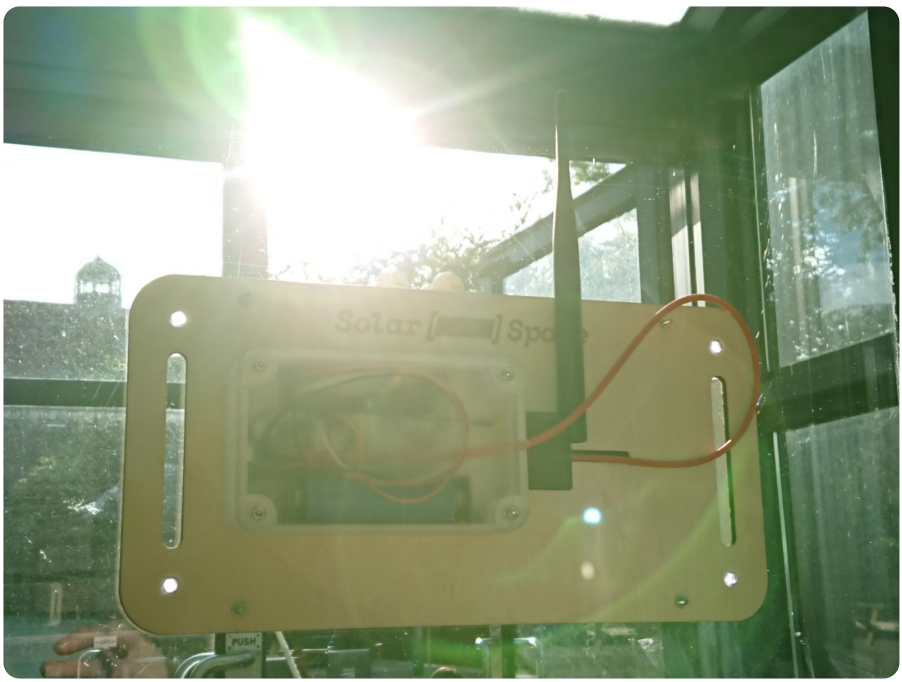
This project challenged me in many ways that I could not have expected. Starting as an idea of defying data centres and surveillance, the project grew into a much more defined conversation around how we can empower our communities in proaction. Communication is the most powerful tool we have, and Solar Space has become a way in which we can build user-powered, democratic local networks of digital communication.

Resolving myself to build a functional prototype of this project ended up being a difficult challenge with many roadblocks, retries and long nights. In an era of AI programming, doing it myself was a time consuming process, yet proved fruitful in achieving my goal of building a realistic solution for my identified problems.

While I believe my outcomes fit my original goals, I feel my process lacked clear routes to evaluation, often relying on hastily assembled groups of students. Towards the end I was able to contact clear experts in the field but I could have given myself more room to breathe with their feedback. My self-evaluation, drawn from years of research and immersion within these fields of protest, activism and sustainable computing proved vital however.

Throughout my time in college I have failed to deliver a project I am truly satisfied with, however Solar Space is a project that I believe has the ability to make an impact be it with activist groups, marginalised communities or generating a conversation within the broader public.

Conclusions



Where to?

Solar Space as a project wont end here. There is a lot of work and learning to do on the programming and hardware, the software framework requires a few more things to be capable of the many different features in the final concept. As a whole the system requires more long term testing of the hardware and software to be truly confident and honest about it's capabilities.

Similarly however, I would like to expand the range of the project. Currently each Solar Space module is limited to a short throw distance, a maximum of 25-30 meters. During my research I discovered the Wifi HaLow specification, a version of wifi that can reach 3-5km. Current market-available HaLow boards are illegal in Europe, but the next generation makes amendments to the spec to allow their use here. The addition of Wifi HaLow to Solar Space could evolve it into a much larger scale social network for activism.

Lastly, and most importantly I'd like to get Solar Space out in the real world with groups to use for activism and organisation. Getting people for testing was the hardest part of the project, but I want to keep pushing to get feedback and improve Solar Space as time goes on.

Conclusions

Many Thanks

There are many to thank for their help, not only with Solar Space but over the past three years.

I'd like to thank Marcus Hanratty for putting up with me, and more importantly supporting in navigating this project. Further thanks to Sam Russel, Jennifer Groarke, Brian Gough and Tara Whelan for their help over the years. Alan, Gerry and Konrad for their feedback and advice on making the physical aspects of the project.

Thanks to those who took time out of their days to give feedback, especially Roel, Durian, Wakest, Anna and Tomi. Thanks Kristine for keeping me sane. Thank you Conor, Eoin and Chris for helping out with the video. Thanks to everyone in the basement for being amazing company throughout the past few years.

- Evan