08 2026

AI SUMMER PROGRAM

PLAYBOOK

Increasing public sector productivity with AI tools

Aalto-yliopisto x SITRA x Lovable

TABLE OF CONTENTS

WAYS THROUGH THIS PLAYBOOK

IF YOU'RE A CASE OWNER

You hosted a builder this summer and want to see what came out of it

IF YOU'RE A BUILDER

You want to see how peers approached similar problems, and reuse a play

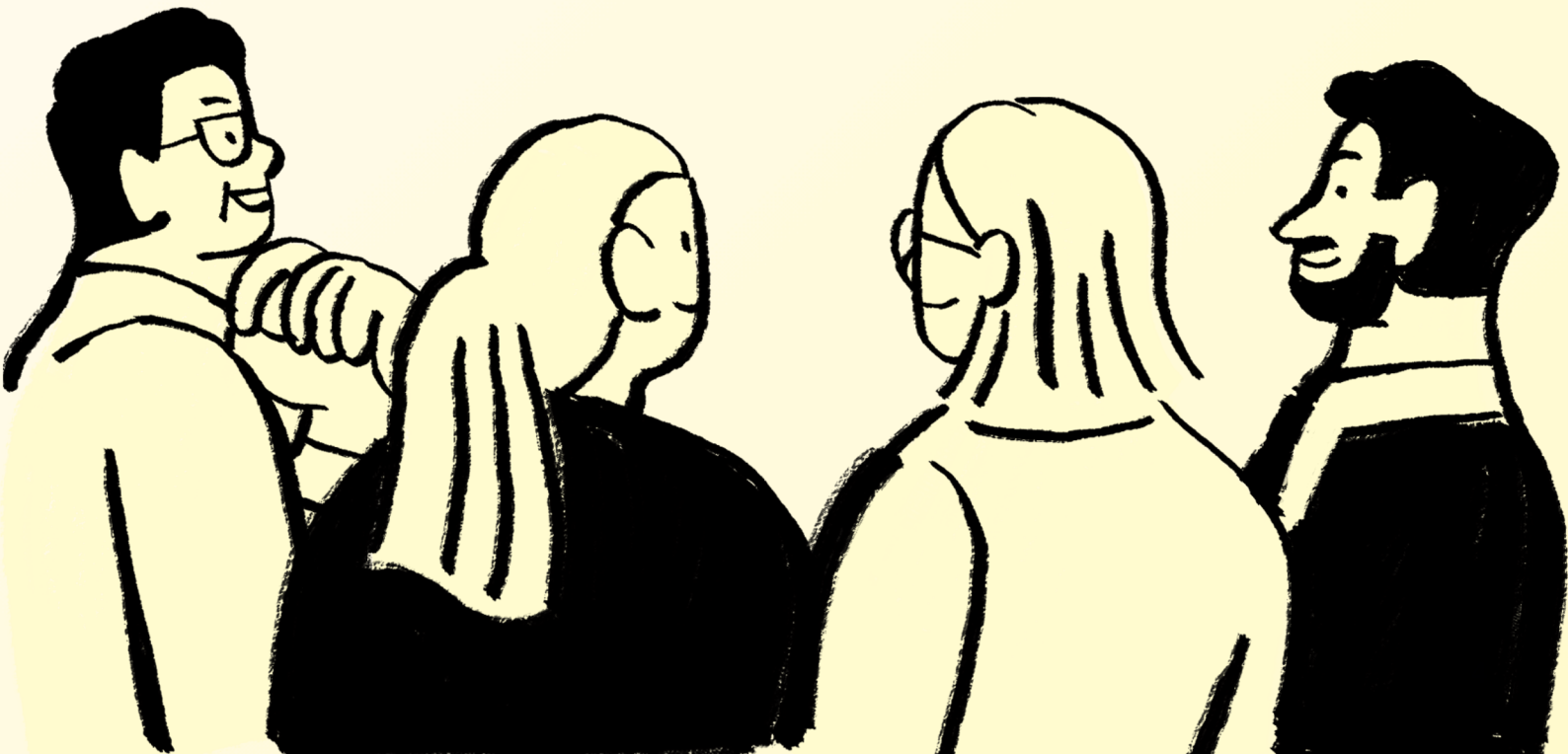
IF YOU'RE RUNNING THIS NEXT

You're organizing a future cohort, class, or pilot and want the full model

A QUICK OVERVIEW OF IT ALL

PREFACE

LETTER FROM THE ORGANIZERS



LETTER FROM AALTO

The Finnish public sector is in urgent need of productivity improvements.

At Aalto University we have some of the top talent in the country, and Lovable is leading with new tools that make it possible for everyone to build software. So it was quite natural for us at the Department of Industrial Engineering at Aalto University to team up with Sitra and Lovable to set up the AI summer program: Ten selected students using AI tools to develop solutions to public sector challenges over two months, funded from Sitra's Public Sector Productivity Programme.

It's been a great success. I'm really happy and proud of what the students have been able to build in just two months, and most of them without deep technical skills. This wouldn't have been even remotely possible a few years ago. And they've put their skills to good use, serving public sector organizations, their employees and their customers with new, innovative digital services and tools.

The next step is to share
and scale the learnings.
This is what this
playbook is for.

Jens Schmidt,
Professor, Head of the Department of Industrial Engineering and Management (TUTA), Aalto University

LETTER FROM SITRA

YOUNG TALENT AS A DRIVER OF THE PUBLIC SECTOR PRODUCTIVITY LEAP

This program was never only about the ten solutions produced in the end. It was a broader experiment in how Finland can unlock new sources of growth, productivity and renewal. Students have up-to-date, interdisciplinary expertise that Finland urgently needs, yet too many struggle to find opportunities to put it to work. The public sector can offer a powerful platform for this: meaningful challenges, better services and attractive career paths for a new generation.

The program also showed why collaboration between education, research, business technology and the public sector must be accelerated. As low-code tools democratise solution-building, more people can learn to design and develop useful services.

Warm thanks to the students, the Department of Industrial Engineering at Aalto University, Lovable and our public-sector partners. This is only the beginning of a new way to combine young talent, research, technology and public-sector needs for the benefit of Finland.

Kirsi Hanhisalo,
Senior Lead, Public Sector Productivity Program,
Sitra

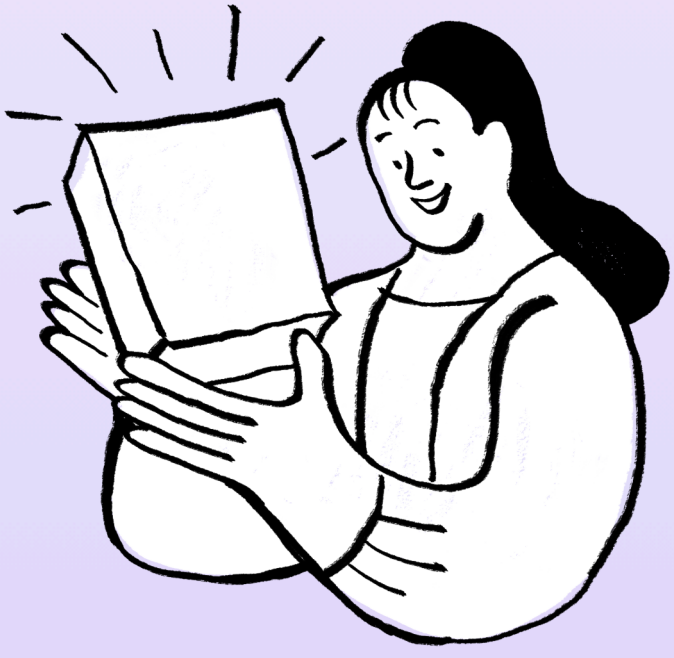
CHAPTER 1

INTRODUCTION

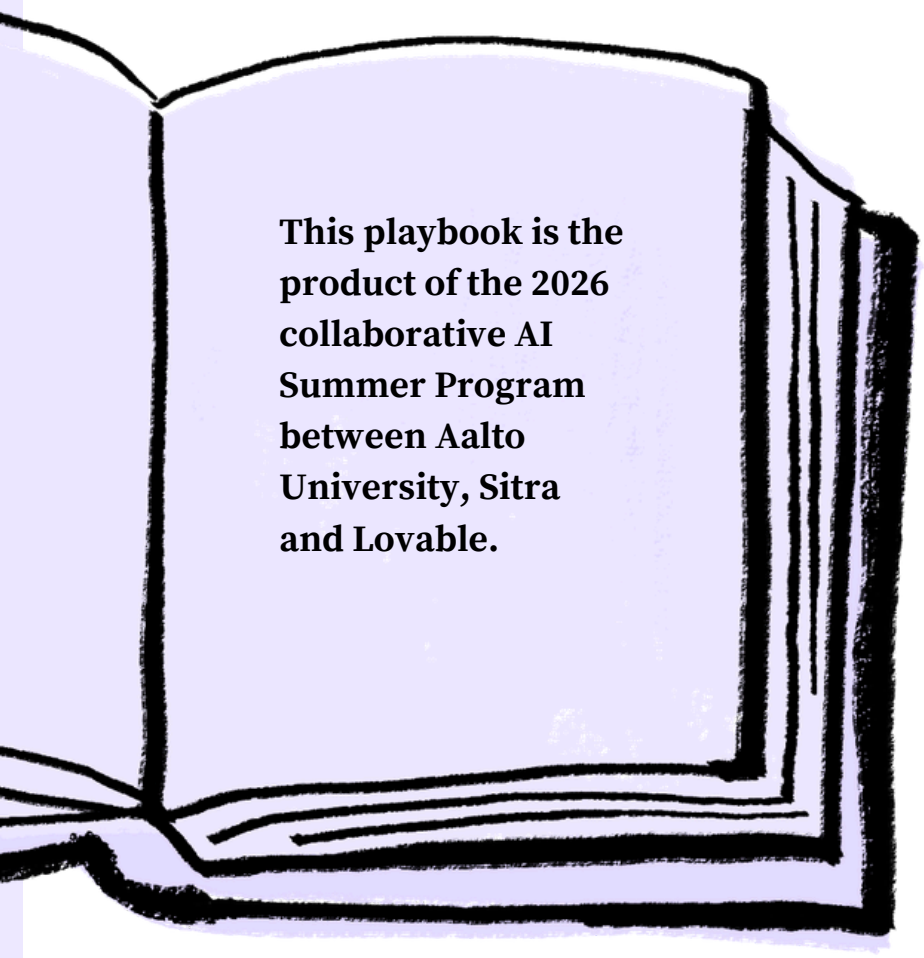
WHY THIS PLAYBOOK EXISTS,
THE PROGRAMME AT GLANCE,
WHAT ACTUALLY HAPPENED

CHAPTER 1, SECTION 1

WHY THIS PLAYBOOK EXISTS



HOW WE WROTE THIS PLAYBOOK



This program brought together ten Aalto University students and eight public sector organizations, who, between them, contributed ten project challenges of varying scopes. The students came from mixed backgrounds spanning economics, engineering, and arts and design.

The projects differed a lot: some arrived tightly pre-scoped as software engineering tasks, while others were broad design problems in which defining the challenge was the first piece of work.

The program paired the public sector's need for fast, low-cost innovation with students eager to experiment with new AI tools by solving real problems.

The goal of this playbook is to demonstrate how easy and efficient AI has made small-scale experimentation. The playbook showcases what is possible with no-code AI tools, where these tools fall short, and the difference between organizational and technological barriers. Additionally, it documents how the program and its individual projects were run, articulating both the characteristics of a good challenge, and how this kind of collaborative design and development work can be structured.

The playbook is written for public organizations wanting to experiment with the use of AI tools to rapidly ideate solutions for their real-world problems, and for anyone running a similar collaborative program.

CHAPTER 1, SECTION 2

THE PROGRAMME AT A GLANCE

WHO WAS INVOLVED

After Lovable, Sitra and Aalto had agreed on the shape of the summer research project, the search for the students began.

A job ad open to all Aalto University students was posted and advertised. In a two-week application period over 200 students applied for the positions. Of them, after a two-step application process, 10 students were selected.

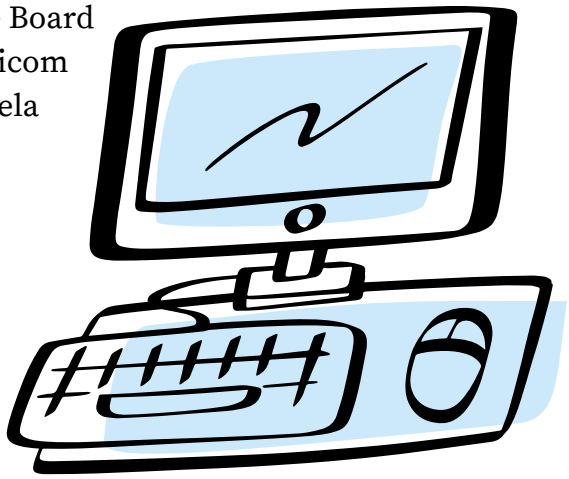
PARTNER	ROLE
Sitra	Funded the project, set the goals, connected the organisations
Aalto University	Recruited and employed the students, provided supervision and facilities
Lovable	As the technology partner, provided the building tools, credits and technical support
Client organizations	Each brought a real challenge, a named contact person, and access to users and feedback. Participating: Finnish Police Board, Kela, THL, Migri, Traficom, Finnish National Agency for Education (OPH), the Finnish heritage agency, GAINS research project
Students	Ten students from Aalto, each owning one challenge end to end

WHAT EXISTED AT THE END

Using no-code AI tools, each student spent 8 weeks building a solution for their assigned service challenge; at the end of the program, **ten user-tested prototypes had been built:**

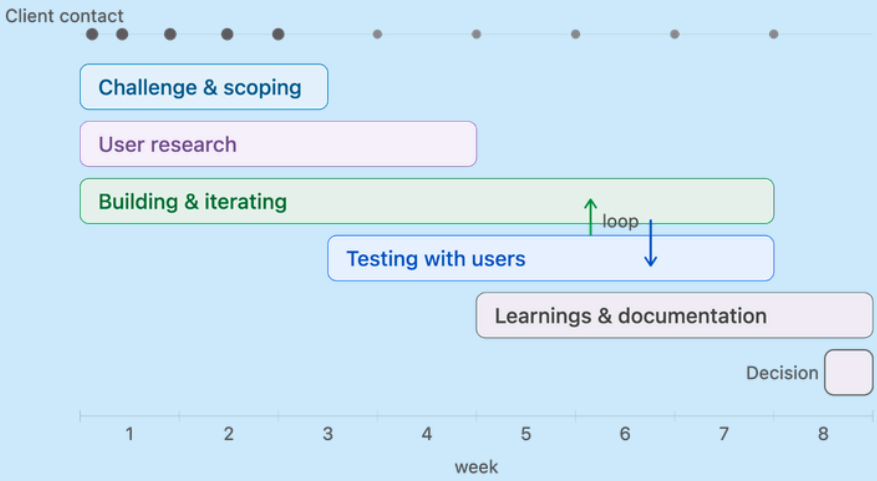
- Availability scrapers and a dashboard for the Kela65 pilot (Kela & THL)
- A decision-support dashboard for Aalto Economic Institute's GAINS research project, which researches wellbeing counties use of digital services.
- Three iterations of a redesigned Varda data-entry interface, improving key workflows for early childhood education staff
- An AI-assisted project management platform for Finnish Heritage Agencys's experts.
- A digital residence permit card concept for Migri developed for construction industry context
- A document-redaction assistant for the Police Board
- A semantic search and reporting tool for Traficom
- A research-funding monitoring concept for Kela
- A youth-facing service prototype for OPH
- An internal comparison tool for OPH

The full story of each project is in the case snapshots (Chapter 2)



THE MODEL

The program ran from 15.6 to 16.8 and began with a bootcamp week where students were introduced to the tools and ways of working. This was followed by roughly 8 weeks of sprinting (contextualizing the challenge, researching users, building, testing, and iterating), a cohort retro meeting every week, and a final project demo day on 20.8.



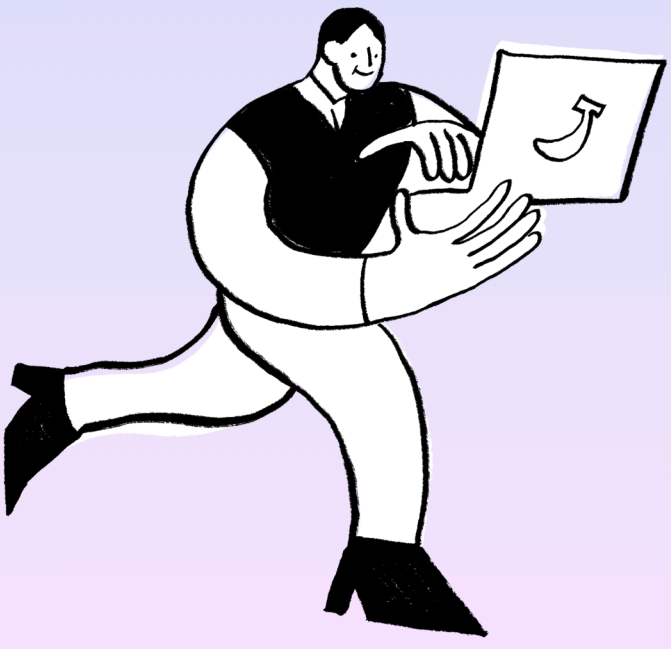
Each phase exists in every project — what varies is how much of the summer each one takes, and how many times the build–test loop runs.

- **Challenge & scoping:** The organization provides a real problem and a project contact; together with the student the problem is narrowed to a slice that is buildable in eight weeks. Scoping largely determines the shape of everything that follows — see Section 3.1 for what makes a challenge suitable.
- **User research:** Students use differing techniques to understand the people and workflows behind their challenge. In this program, research rarely finished before building started; it ran alongside it. See how the cohort found and involved users in Chapter 2 case cards.
- **Building & iterating:** Prototyping starts in week one because the AI tools make building so rapid it can be used throughout planning and iteration, rather than as a final step. To learn more about getting started with no-code AI tools see Section 3.2.
- **Testing with users:** Putting the artifact in front of real users, early and rough. Feedback here caused the only true pivots in the cohort. For methods and lessons see Chapter 2 case cards.
- **Learnings & documentation:** To see how each project captured what worked, what didn't and what production use would require, read through Chapter 2 case cards.
- **Decision**. Every project ends with an explicit call: stop, develop further, pilot, or move toward production — with responsible-AI questions checked before anything advances (see Section 3.2).

CHAPTER 1, SECTION 3

WHAT ACTUALLY HAPPENED

WHAT THE PROGRAM LOOKED LIKE IN PRACTICE



WHAT THE PROGRAM LOOKED LIKE IN PRACTICE

Data- and tool-heavy challenges with clear scopes produced build-dominant projects, while ambiguous, **concept-stage challenges** required more research and scoping. In other words, the model adapts—and its shape is largely determined during scoping, making challenge selection critical (Chapter 3).

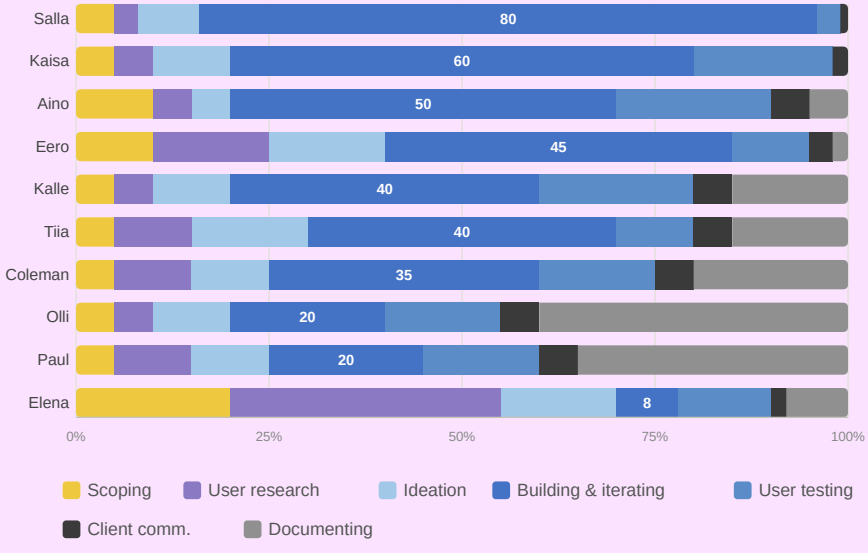
The traditional "research first, build later" sequence was often inverted. Several students used early prototypes to scope the problem and communicate with clients, ensuring they were solving the right thing. Rather than running sequentially, research, scoping, and building happened concurrently around a living prototype.

This enabled early user testing: five of nine reached users within the first two weeks, and all by week five. Early versions were often rough screenshots or partial demos, but still redirected entire projects. Cheap iterations made three to four major versions – and making frequent pivots – normal and affordable. Clearly scoped projects progressed more linearly, while ambiguous ones looped repeatedly.

A key planning consideration in Finland is the July holiday season, when users are difficult to reach. Schedule research-heavy phases before July or secure substitute contacts well in advance.

We asked every student how they divided their working time across the different phases, totaling 100%.

On average, 40% of the time was spent building and iterating, 14% user testing, 14% documenting, 11% ideating, 10% on user research, 8% scoping and 4% on client communication. However, the spread was wide, with building ranging from 8% to 80% across projects.



CHAPTER 2

CASE CARDS

CASE + PLAY

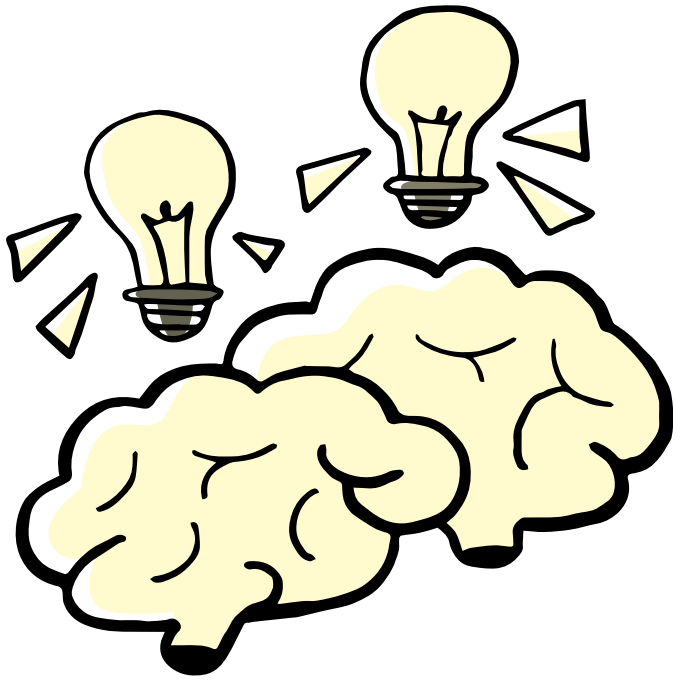
10 CASES + 10 PLAYS

As public-sector reports continue to provide generalized recommendations and benchmarks in the ever-changing AI environment, we sought to present the individual stories behind our own experiments, case by case.

The format is simple: We each have two pages to explain our challenge, the design process(es) we undertook, and the solution(s) that arose. Each case section concludes with a play: a specific learning or response to friction grounded in our project.

These plays are narrated and made credible by our own experiences, detailing not just what happened but how and why. They are not intended to be prescriptive, but adaptive to diverse contexts. In this regard, future builders might recognize when they need to adjust course and invite the complexity and nuances of each case in, not falling into the frustrating trap of expecting all the pieces to line up in an ideal way each time. As such, these plays help each of us articulate the effectiveness of our decisions while recognizing that judgement, human or AI-assisted, holds us accountable and responsive to the playing field.

This section starts with the 10 cases + plays, followed by the 10 plays we have all developed in one place. Feel free to skip around as you may find relevant.

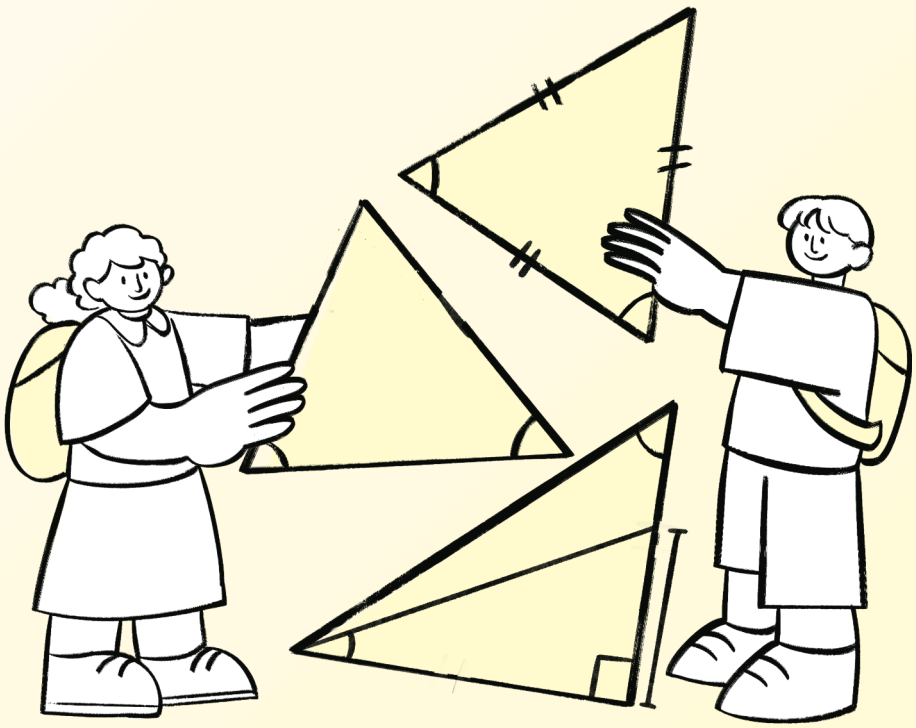


CHAPTER 2

OSAAMIS-POLKU

Finnish National Agency For Education

MY PATH: AN EASY-TO-USE TOOL FOR MAPPING FUTURE OPPORTUNITIES



THE PROBLEM

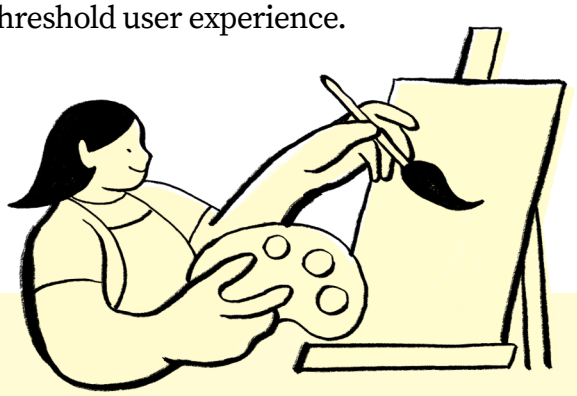
The Finnish National Board of Education presented a challenge that gave me agency and creativity. OPH recently (in Jan 2026) released a new digital online service designed to support career planning, skills development, educational choices and employment.

It helps users identify their own strengths and find suitable career paths. The problem, however, is that the service is lacking appeal and guidance for young users, which are perhaps the most important user group. The desired solution would offer a clear, motivating and continuous service journey that strengthens the young person's agency.

THE SOLUTION

I designed a solution and a feature on top of the existing Competency path service which would include a clear user journey, a justified role for AI, a meaningful Osaamispolku–Opintopolku connection, and a measurable value for the young person's agency.

The solution was a Competency Path feature called My Path, where the young person swipes questions related to their competences and interests and receives 2-3 suggested directions based on a semantic search. The solution focused a lot on a low-threshold user experience.



Things that went well

- Developed a low-threshold, easy-to-use tool for mapping future opportunities.
- Playful and pleasant user interface — not too difficult or tedious for the intended audience.
- Clear communication with the case owner and an understanding of their needs.

Challenges

- The case was open to different directions and possibilities, so I had to take time to make sure I didn't build anything unnecessary and that I stayed on the right track.
- The existing Competency path is designed to produce very specific results, while the new My Path tool aims to give compressed results based on classifications. This meant that the API´s signals had to be adjusted for my purposes, which required solving several technical problems.
- Scorecard showed very inaccurate results for a long time. Adjusting semantic search to be the primary recommendation engine required work, but ultimately produced the best results.

PROCESS DESCRIPTION

The process began with an in-depth mapping of the problem, which emphasized understanding the youth's perspective and the current UI / UX of the existing competency path. The beginning of the process involved brainstorming, refining the problem, and scoping so that the prototype could be implemented during an eight-week sprint. The chosen direction was important at this point since OPH had given agency for this choice.

After deciding the perspective from which I would build the prototype, I built a quick mock-up of the idea, which I showed to the customer. After approval, I developed the prototype based on the current competency path APIs. The process required many intermediate stages, important user testing, and iterating based on feedback. In the final version, semantic search complements the ability of the competency path tools to produce meaningful results.

FINAL STATE

A working prototype that only needs slight development to be integrated to the Osaamispolku environment. A proof-of-concept prototype has been tested with the real user group and the feedback on the prototype was positive.

PLAY 1: Design (for) Your User

SUMMARIZED:

Build scope around the user group, and communicate it clearly to your AI tools. Design based on observed needs, challenges and usage situations of your audience, not assumptions.

To differentiate from the original OPH platform, I started by researching my user. This wasn't technical but rather an exploratory inquiry. I set up low-fidelity public testings and interacted with my user group's popular media, documenting the needs of today's youth.

During testing, I paid careful attention to my assumptions, which I updated weekly. Then, I could pivot as needed, which is a part of the creative and iterative process. In addition, an LLM's assumptions may seem confident, but you must know how to question them too. Don't settle for one prompt!

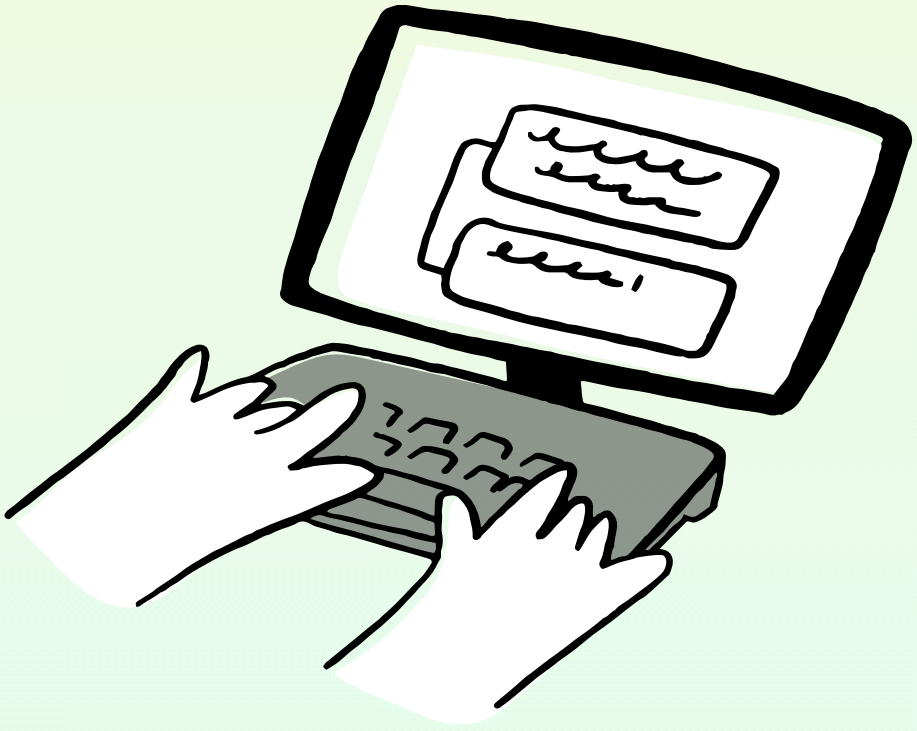
Best insights: test in easy-going situations. I conducted testing in a shopping mall, where I created a very low-threshold and natural testing situation with a small compensation. The testing situation became a potential usage case: quick and relaxed. Don't be a study counselor. Be hip like them!

CHAPTER 2

VARDA

Early Childhood Education Data Repository

IMPROVING KEY WORKFLOWS IN VARDA FOR FOR PRIVATE ECEC PROVIDERS



THE PROBLEM

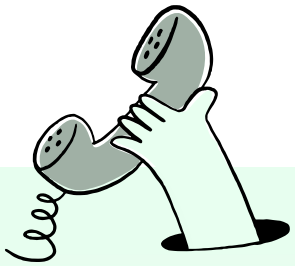
Private Early Childhood Education Care (ECEC) providers struggle with Varda's interface, data entry is confusing, errors only surface after submission, and there's no clear way to know what went wrong. The original brief pointed toward building an external compliance-checking tool, but user research redirected the focus to the existing interface itself.

THE SOLUTION

An improved UI across three versions. Version 1 focuses on incremental improvements within the existing system: clearer error messages, inline help text, improved form structure for adding children and employees, better visual hierarchy, and more consistent use of colour to distinguish errors, warnings, and informational messages.

Version 2 introduces structural changes, moving the add child and add employee flows from modals to standalone full-page views with a progress panel, relocating the reports view to its own navigation tab, and adding the ability to save a draft, set an email reminder, and receive a confirmation prompt when leaving an unfinished form.

Version 3 takes a step back from the existing system entirely, rethinking the core flows without the constraints of the current Varda data model or UI conventions.



Things that went well

- Calling providers directly was the most effective recruitment method. Despite the summer timing, kindergarten managers were generally responsive.
- Recreating Varda in Lovable from screenshots worked well as a starting point; having a recognizable base made it easier to get concrete feedback.
- A design background made it easier to identify usability issues without a formal audit process.
- Having direct access to the Varda team through Slack meant questions about the existing system could be answered quickly without blocking progress.

Challenges

- Varda has to serve a very wide range of users — from small family daycare providers (perhepäivähoitajat) to large daycare chains — within the same interface.
- The system is built to handle complex edge cases, but most users only ever need the simple version.
- Many providers were on summer vacation, which meant that recruitment for research and testing required significant outreach.

PROCESS DESCRIPTION

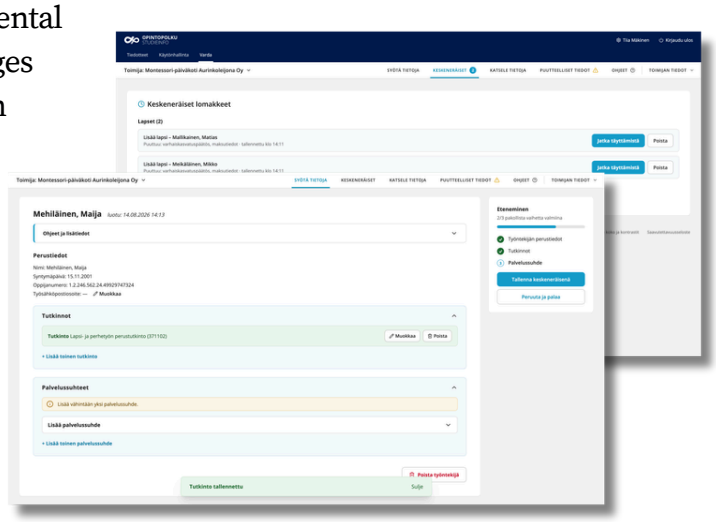
I started by reading documentation on Varda and using Claude to do a thematic analysis of the interviews I had conducted. I also interviewed one public ECEC provider to understand why the problem doesn't exist for providers using direct system integrations with Varda. From there, I took screenshots of Varda's demo version and recreated it with Lovable. Once I had a working copy, I remixed it and iterated to improve any usability issues.

I conducted 1-hour user testing sessions with people I had interviewed earlier as well as with a few new participants. Testing focused on three specific flows — adding a child, adding an employee, and the reports view — and participants filled in a System Usability Survey (SUS) on both the current Varda and the new prototype, allowing for direct comparison.

FINAL STATE

Three working prototypes: version 1 with incremental UI improvements, version 2 with structural changes to the flows, and version 3 as a rethought redesign without the constraints of the existing system.

The prototype and findings were handed over to Varda's development team at the end of the program. Whether they are implemented depends on the team's priorities, as well as whether improving the UI or building better integrations would better address the underlying problem.



PLAY 2: Screenshot to Prototype

2

SUMMARIZED:

Don't start from a blank canvas when there's already a system in place. Recreate the existing interface and visual language in Lovable using screenshots, then iterate on top of it.

Public sector redesign often means improving an existing system, not building from scratch. Abstract mockups are hard for stakeholders to react to — but an improved version of something they use daily is immediately legible.

The approach: input screenshots of the existing system to Lovable, and ask it to recreate the interface. Once you have a replica, prompt improvements. Because the starting point is recognizable, stakeholders can give concrete feedback instead of reacting to something unfamiliar.

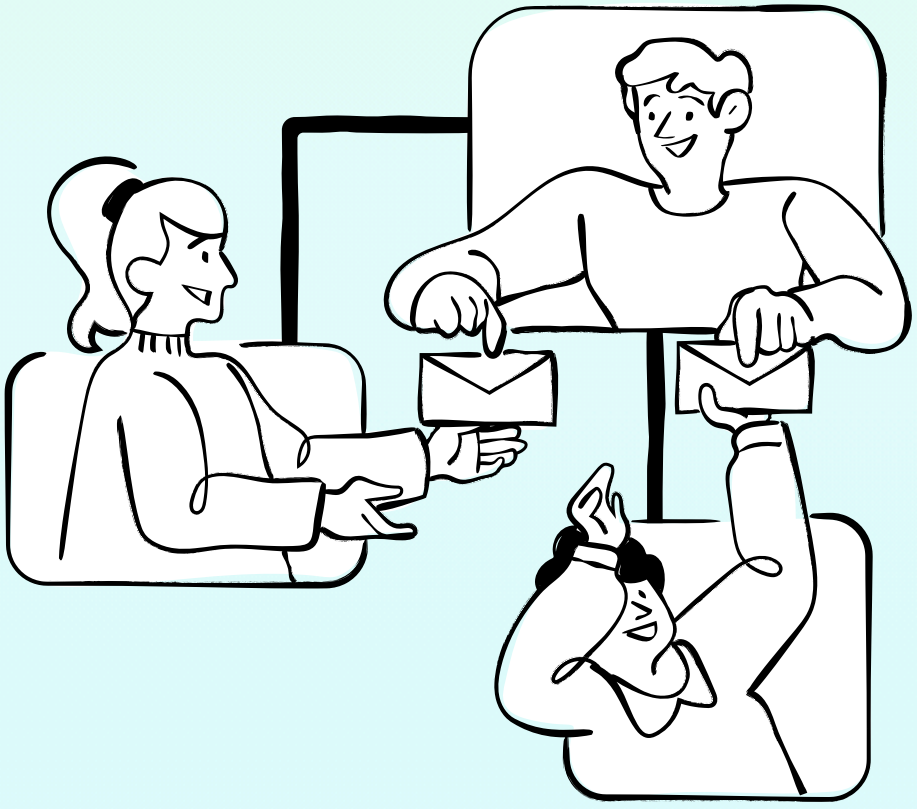
Use Claude alongside to think through design decisions and write precise prompts. For structured testing, Useberry lets you set up click-through tasks, and a SUS survey before and after gives you a quantitative baseline to compare versions.

CHAPTER 2

MIGRI

Finnish Immigration Service

ENABLING REAL-TIME FLOW OF RIGHT TO WORK INFORMATION



THE PROBLEM

In the construction industry, the main contractor is legally responsible for ensuring that every foreign worker on the site has the right to work, but cannot obtain that information officially from the Finnish Immigration Service.

The challenge was to work out how up-to-date right-to-work information could reach a main contractor from Migri. The problem lay primarily with main contractors, meaning that the actual users of the solution are external to Migri.

THE SOLUTION

A prototype that puts a digital residence permit and its right-to-work status inside the EU Digital Identity Wallet (Suomi.fi Wallet in Finland, rolling out in 2026), letting a worker share verified, official permit data directly with a main contractor from their phone.

The user group is twofold where main contractors hold the underlying need: access to reliable right-to-work information. However, workers are the actual end users who operate the service to share the information.

Things that went well

- User research paid off the most. Understanding the problem space proved more valuable than building quickly, especially given that the challenge arrived unscoped
- Pivoting is a result, not a failure. The findings that forced a change of direction were the most valuable output of the process

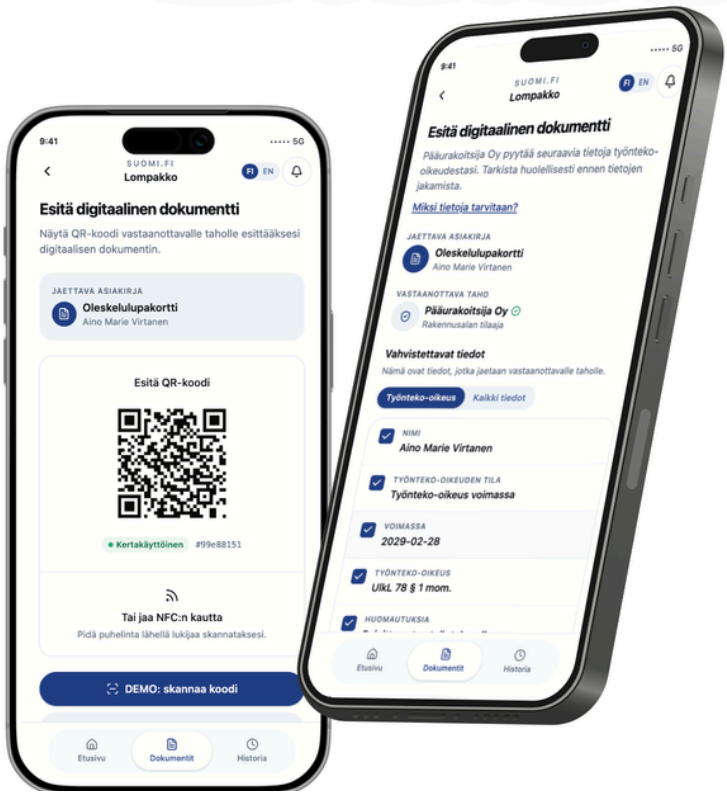
Challenges

- Much of the project went into identifying the actors, the legal architecture and the right users before a solution could even be specified
- No access to real data or Migri's systems. Building something shippable was not realistic here and the prototype had to rest on data and assumptions drawn from interviews.
- Finding users to test with was hard. The end users sit outside the client organisation, so there was no established channel to them.

PROCESS DESCRIPTION

I followed a user-centred design approach, cycling through understanding the context, defining requirements, prototyping and testing. The phases overlapped constantly and the process was far from systematic. I started by interviewing people from Migri and the construction industry. That scoped the problem down to construction as a case study.

I identified the relevant users, and prototyped a first solution: a separate portal for checking right-to-work status, and testing showed it wouldn't work. I pivoted to a digital residence permit card concept and continued testing with that. Schedules were hard to stick to, so I mostly freestyled day by day and used interviews and testing sessions as self-imposed deadlines.



FINAL STATE

A proof-of-concept prototype that has been tested with real users and a report introducing the process and the findings, handed over to Migri as a basis for deciding whether to take the work forward.

PLAY 3: Look Beyond an Application

3

SUMMARIZED:

If the initial scope is wide and involves stakeholder politics and legislative restrictions, the solution often lies beyond the design of one AI-powered app. Consider the cultures that the prototype embeds itself within, and identify what needs to change for successful implementation.

My initial challenge touched on multiple things I had almost no knowledge of: work-based residence permits, how Migri handles them, and who is legally entitled to receive the info. There were several users to consider: Migri caseworkers, employees, employers and main contractors. On top of that came unfamiliar construction industry practices and processes.

I had to conduct user research to get an understanding of the context, and to decide whose perspective the solution should take. The amount of information I gathered was overwhelming, and it surfaced multiple subproblems that all pointed in the same direction: this problem isn't solvable by a single solution. Not one app, not one service. It involves legislation, it is structural, and it is the sum of multiple things.

CHAPTER 2

GAINS

A Government-Funded Multidisciplinary Research Initiative

DASHBOARD TO SUPPORT DECISION-MAKING FOR FINNISH WELLBEING COUNTIES

THE PROBLEM

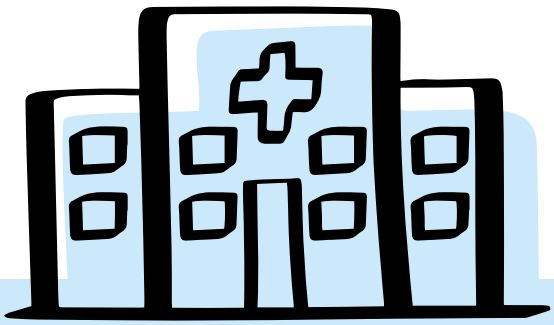
Finland's healthcare is split into 22 independent wellbeing counties. Each one buys and tests AI in silos, so lessons rarely spread. There is no national overview of what AI is being used, or if it actually has positive impacts on access, cost-effectiveness, or system performance. Current data is self-reported and quickly outdated, risking counties to waste money repeating their peers' dead ends.

I built this solution for external users, such as county digital directors, procurement managers, and national coordinators, to finally give them a clear picture of the AI landscape in Finnish healthcare.

THE SOLUTION

I developed a dashboard that provides a fully sourced overview of healthcare AI across Finland.

An automated pipeline scrapes public sources—like EU and national procurement notices, county decision documents, medical device registers, and news—and collects information about AI products deployed in each wellbeing county. The info from the database then gets turned into a live catalog of AI products and evidence.



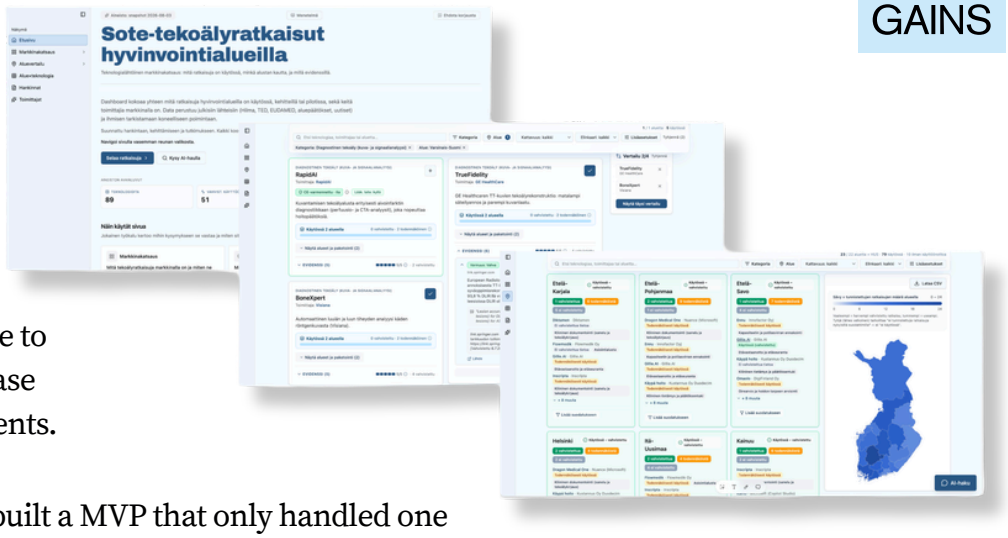
Things that went well

- Quickly sharing a rough draft was the fastest way to clarify the client's vision.
- Rebuilding in week three made everything after it significantly faster.
- Everything links to a source, making it easy to verify that the AI isn't fabricating any data.

Challenges

- Finding user testers during the July summer holidays was difficult.
- Information buried in PDFs, PowerBIs, and incompatible portals made the data collection difficult to automate and host.
- Broad searches caused false positives, and raw sources rarely distinguish between pilot programs and full deployments.
- Counties documenting AI projects without naming suppliers creates a structural blind spot.

PROCESS DESCRIPTION



I started with making a quick prototype in Lovable to give the client something concrete to react to, which helped tease out the project requirements.

By the third week, I had built a MVP that only handled one category type. However, the data pipeline wasn't scalable, and had become a tangled mess of workflows. Instead of patching it, I scrapped it and rebuilt the backend using the knowledge I gained from the previous build. Because I now understood the domain and constraints better, the rebuild only took a day.

The new architecture is clean: A GitHub-hosted backend parses the primary sources and writes its findings to a Supabase database. Lovable then reads from Supabase to visualize that data on the dashboard. With a stable system in place, I spent the following weeks expanding the catalog as well as focusing more on usability and tweaking the UI.

FINAL STATE

The system is self-refreshing, and in production end-to-end. The backend and dashboard work together, resulting in the broadest overlook of the Finnish healthcare AI solutions on a single website.

PLAY 4: Build to Learn, Rebuild to Scale

4

SUMMARIZED:

The first thing you build is not the product. It is the first step in finding out what the product should be. Treat version one as a probe, and choose a date where you ask honestly whether it is a foundation or a rehearsal.

When starting complex projects, accept that you can't know everything on day one. Instead of rigid planning, embrace improvisation and treat your first version as a probe, then test it on real, messy data to uncover actual constraints.

To avoid getting trapped by your own rough draft, set a date in the project to sit down and answer the question: is this a scalable foundation or a rehearsal to learn the rules? Without this review, you might end up compounding on decisions you made when you understood the problem least.

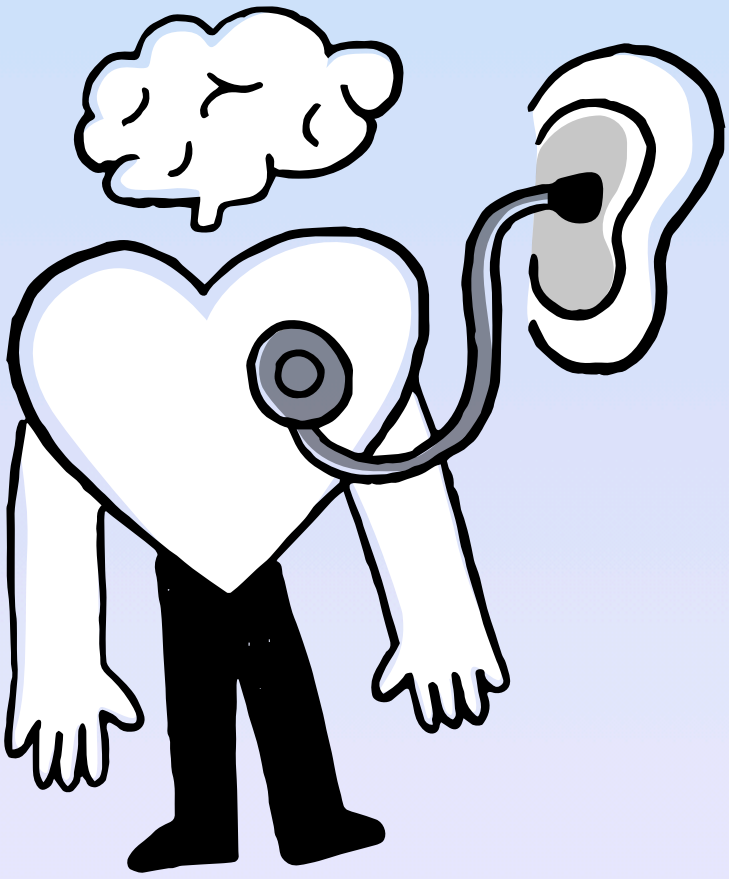
For example, my first data pipeline collapsed after three weeks of messy reality. But because I treated it as a rehearsal, I was able to rebuild the backend in a single day based on my learnings, allowing me to later focus on catalogue expansion rather than patching a messy backend.

CHAPTER 2

THL

Finnish Institute for Health and Welfare

TRACKING GP APPOINTMENT AVAILABILITY FOR THE FREEDOM-OF-CHOICE PILOT



THE PROBLEM

In September 2025, Kela and THL launched a pilot allowing people over 65 to book doctor's appointments with private healthcare providers, aiming to ease pressure on public healthcare queues. THL and Kela's researchers are tracking how well the pilot is working, but they're missing a key data: how many GP appointments private providers are actually making available under the pilot.

Combined with Kela's records of used appointments, this availability data would let researchers see how effective the pilot really is. Not just how much it's being used, but whether the appointments are there to use in the first place.

THE SOLUTION

I built a system that checks the booking pages of the three biggest private healthcare providers automatically every night and collects the available GP appointments into one properly organised database. On top of that sits a dashboard that gives researchers a quick overview of the collected data and confirms that the nightly checks are actually running correctly.

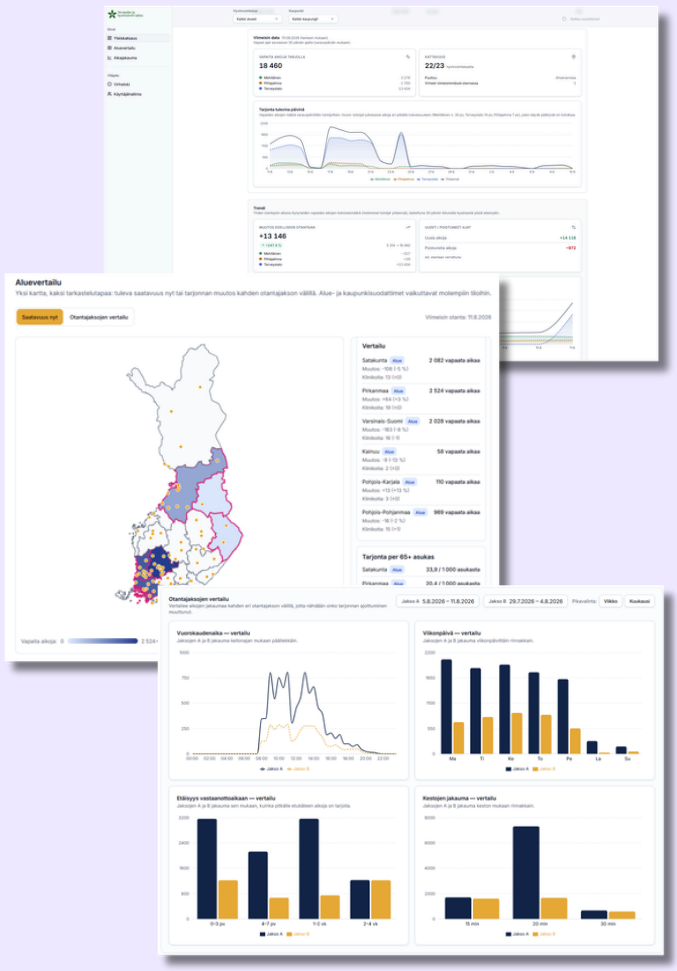
Researchers doing deeper analysis can work directly with the database, while the dashboard exists for fast comparisons by location and time, being particularly useful for the more managerial roles who need a quick overview rather than the full dataset.

Things that went well

- An organized process that is well documented
- I developed a tool that went straight into actual use
- Quality data suitable for research use

Challenges

- Lacking the access to systems the data came from: Having to negotiate with each healthcare provider directly during summer holidays was a slow process
- Running out of AI-tool credits: backend heavy project required lots of tokens, which slowed down the process. This forced me to be strategic with my tool usage.
- Time is challenging concept to visualize in an understandable way when you have multiple snap-shots for the same time period. This got my brain fried at times



PROCESS DESCRIPTION

I worked from a fairly structured plan that I frequently updated throughout the project. I started by digging into the problem and then held four interviews with THL and Kela representatives to pin down scope and requirements. In parallel, I reached out to each of the three private healthcare providers directly, since I needed their agreement on how their booking sites could be checked and what limits I had to respect. This required some real negotiation at times. Once that groundwork was in place, I built the data-collection side first, then the database, and only then the dashboard on top of it. I used Claude to plan the technical architecture and build the parts that needed to run reliably in the background, and Lovable specifically for the researcher-facing dashboard. I kept the two clearly separated, which made it easier to get both right.

FINAL STATE

The system runs automatically and collects appointment data every night from three private healthcare providers, feeding directly into THL and Kela's research. The dashboard lets researchers accurately track how the dataset grows over time and compare availability across providers and regions. Ownership of the tools, the database, and the code was handed over to THL's staff to keep up and develop further, supported by documentation and guides covering how the system works and is maintained.

PLAY 5: Learn Together with the Tools

5

SUMMARIZED:

Build both your own and the AI-tools understanding of the project in tandem. Sometimes AI-tools try to rush you into solutions. Don't hesitate to take a step back and plan the big picture before starting to build a new feature.

AI tools can encourage quick decisions and fast results, sometimes leading to solutions that are not fully thought through. While AI can build an understanding of a problem faster than a human, it is often worth stepping back to consider the bigger picture: How do the features work together? What is the user flow? What will the backend architecture look like?

Creating a plan provides clarity for both the developer and the AI, giving the tools the right direction before prompting. Build your own understanding alongside the AI's. These tools can support project management by tracking tasks and suggesting next steps. Take time to understand where your project stands, keep the tools updated on what you learn and decide, and use this shared knowledge to support your work and communicate plans clearly with stakeholders.

CHAPTER 2

POLIISI-HALLITUS

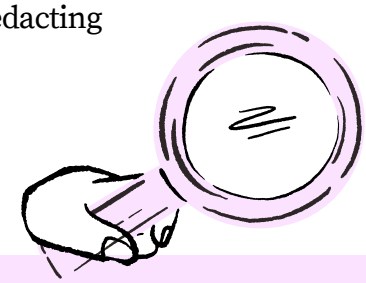
National Police Board

AI-BASED REDACTION TOOL FOR PUBLIC SECTOR DOCUMENTS

THE PROBLEM

Finnish authorities need to release documents on request while withholding confidential information under the Act on the Openness of Government Activities (JulkL 621/1999, § 24). This is a time-consuming manual process, as excessive redaction withholds information the public is entitled to, while insufficient redaction risks disclosing confidential information.

The challenge is identifying all sensitive information and judging whether a piece of information, at a given stage of a case, falls under a specific legal subsection. Case experts and public sector employees spend a lot of time manually processing, marking, and redacting millions of pages annually.



THE SOLUTION

I built an AI-based redaction tool that processes a PDF document, suggests what information is sensitive while citing the subsection of JulkL 24 § justifying each marking proposal, allows the user to review each marking, and lastly adds the correct confidentiality stamps to each page.

The reviewer can ask the model for reasoning behind redaction judgements, confirm or modify each redaction suggestion as necessary, and finalize the document for download. The output documents include 1) a redacted PDF with confidential information permanently removed, 2) a marked PDF copy for archives, and 3) an internal log of all markings. The final tool supports experts in finding the correct information faster with the corresponding legal section citations and stamps.

Things that went well

- Developing an early version of the product allowed for rapid learning and iteration.
- Testing frequently with different users and IT specialists supported rapid development.
- Interviewing users early on enabled a more comprehensive solution, e.g., by broadening the scope to include the legal section and stamping functionalities.

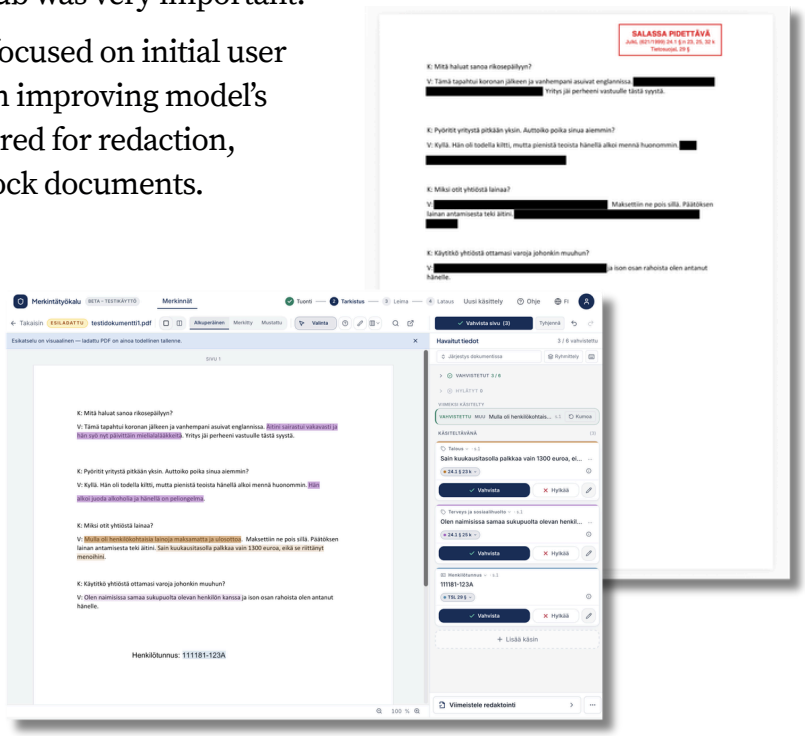
Challenges

- Legal interpretation was surprisingly challenging to translate to language-based rules as interpretation varied by case, and domain understanding was required for full context.
- Developing a local model was not possible via Lovable and testing sufficiently large local models proved challenging.
- Processing time scales with the number of findings, and testing documents of hundreds of pages was difficult due to token limits.

PROCESS DESCRIPTION

I built a working interface in Lovable within the first week, allowing me to discuss the MVP with the partners quickly. Lovable and Claude handled frontend and backend respectively, and connecting them early on through GitHub was very important.

The second phase of development was focused on initial user testing and focusing the development on improving model's understanding of the legal aspects required for redaction, which was tested using AI-generated mock documents.



After that I primarily focused on architecture, security and scaling, while continuing UI development and user testing throughout the summer. Lovable and Claude Code made the building substantially faster, which let me concentrate on what to build, testing with users, interpreting the legislation, and verifying what the AI produced.

FINAL STATE

1) Analysis, review and document processing with corresponding legal citations implemented
2) UI developed together with users ensuring the functionalities support their full use cases
3) A locally-hosted model has been tested as a route to closed-network deployment

PLAY 6: Leveraging Mock Data

SUMMARIZED:

AI-generated test documents can fill in for real data in restricted projects, enabling model learning without confidentiality concerns. Generate diverse mock data to prevent training biased models on limited test sets

In public sector projects you might not have access to any data, and the redaction tool for the Police Board was built using only public information and AI-generated mock documents. A broad range of invented case files with AI-generated personal information, stories, and mock confidential information embedded within the document were used to teach the model what should be redacted based on the different legal subsections.

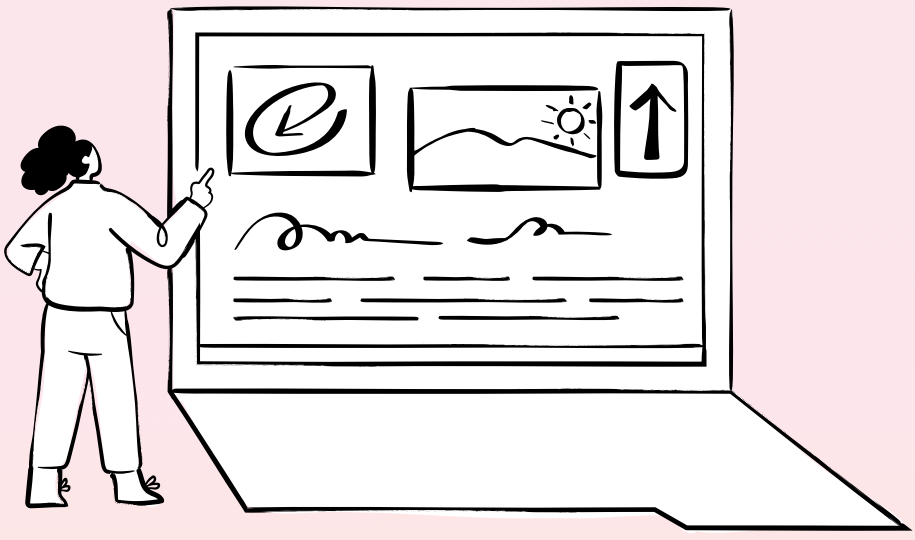
It is important to test and learn from a wide range of document and attachment types, so prompting a diverse set of training documents was valuable. While generated mock documents do not necessarily provide full coverage or a perfect substitute to testing with real data, it was a helpful starting point that allowed the development of a prototype that can be piloted and further adapted to user feedback.

CHAPTER 2

MUSEO-VIRASTO

Finnish Heritage Agency

INTERNAL, EXPERT-FACING PROJECT MANAGEMENT PLATFORM



THE PROBLEM

Finna is home to millions of items of cultural and scientific material in Finland and partners with hundreds of organizations, including Museovirasto, to both maintain their individual archives and expand Finna's.

Finna's public facing inquiry link allows visitors to send record specific questions to the Museovirasto team. Currently all of these public inquiries are routed to a singular Outlook inbox shared by all MV experts across all archival collections.

At over 1,000 inquires a year, filtering and managing the inbox is an unnecessary and taxing labor to defer to the experts, especially since they must each read every message to determine which they are responsible for.

THE SOLUTION

I created an expert-facing, AI-assisted project management platform. While the organization wide visibility made familiar by the outlook workflow is retained, experts now get their own profile where inquiries are routed to them specifically.

Core to my solution is an AI assistant built to minimize extraneous work. It has both the authority to manage a selection of low-stakes cases, and the ability to conduct preliminary research on expert facing cases.

Included alongside the unadulterated public inquiry, is an overview of the initial AI conducted research where the sources used and a synopsis of its findings contextualize the editable response draft provided.

Things that went well

- With a background in project management, experience with AI tools, and as a student studying industrial design I felt well matched for this project
- I provided Claude an overview of who I am, how I work and what my experience looks like, as well as an overview of the program, my organization's problem and their desired solution
- I had Claude create a running MD file called to-do; any time I thought of something, I'd ask for it to be added to the list
- I'd reference my to-do list against the problem, desired solution, progress and timeline weekly, this helped me stay on track and ensured valuable ideas were never forgotten

Challenges

- The initial scope was clear and open-ended enough that I had room to propose my own thoughts. That said, frequent holidays at MV made it hard to move on the implementation of those ideas, as I still needed approval before I could build them
- I learned that the better way to fix features with Claude is to describe the problem, not the solution. I often got tunnel visioned trying to explain my desired fix, this caused more friction than simply ideating changes to Claude's fix after it was built
- I focused too much on the details early on. I would have saved time by waiting until after I'd run the broad sweeping audits at the end of the build, as they caught and fixed these well

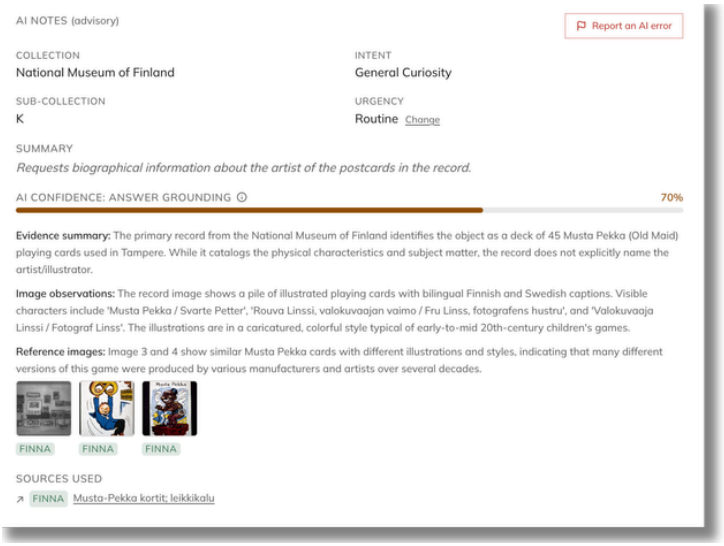
PROCESS DESCRIPTION

I began by creating a rapid low-fidelity prototype in week one, giving me something tangible to scaffold my proposed solution around in week two. This helped clear up ambiguity, reveal misunderstandings early, and validate my direction before I had invested too much time.

I then sent my MV contact weekly batches of questions, allowing me to tackle major build tasks while progressing on other work as I waited. In week four, I conducted my first user test and kept the prototype published, using a Google Form to gather asynchronous feedback. In week six, testing with the platform's target experts validated the work, exposed assumptions, uncovered new feature requests, and answered outstanding questions. The final phase focused on refinement and handoff.

FINAL STATE

After correcting some UX / UI and misalignment with accessibility requirements I've been building the Finnish version of my platform. Other than MV dependent items, the handoff preparation is all that remains. That said, I've left things as close to plug and play as I can in hopes that the handoff document and their ability to adopt this platform (if they wish to) should be easy.



PLAY 7: Befriend Your Claude and Trust It

7

SUMMARIZED:

Plan first, then trust the build. Co-design a list of design and collaboration rules for your AI to follow, and update it as your context evolves. Use AI to stress-test, clean, and audit the build, and challenge its decisions and assumptions against these rules.

Start by introducing yourself, your experience, your work style, the context of the program, your organization, their problem + desired solution, and the aesthetic you want to build in. Then prompt Claude to use this info to make: a master list of rules, a build-tracking document, and a to-do list file which you'll add to across your chats. A good foundational rule is "before you make a design assumption ask for my preference so we can turn it into a design rule."

Then trust Claude to make and push large batches of decisions at a time. Prompt Claude often to audit sections of your platform, use external Claude skills to audit your build as well (ex. impeccable). These audits are incredible at discovering and fixing bugs and inconsistencies you wouldn't see, and oftentimes solve errors that you would have otherwise wasted time trying to fix in isolation.

CHAPTER 2

KELA

The Social Insurance Institution of Finland

DEVELOPING FINANCIAL MONITORING FOR RESEARCH PROJECTS

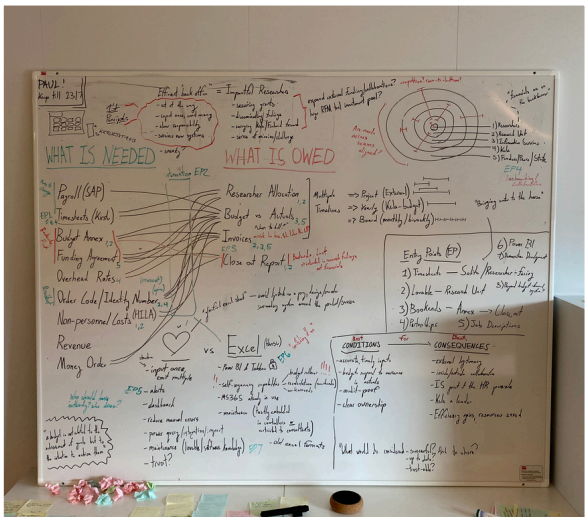
THE PROBLEM

Research project accounting information comes from disparate financial systems and highly varied funding agreements. Controllers manually consolidate this information into spreadsheets, a repetitive and risky practice where small mistakes can cost hundreds of thousands of Euros.

Moreover, controllers rely on inputs from other actors, which determine the accuracy and timeliness of their reporting. In this case, the problem of "garbage in, garbage out" is pertinent.

THE SOLUTION

The solution resembles a glorified spreadsheet. It reduces repetitive entries by deterministically parsing data from the documents Kela already produces, ensuring that every figure can be traced to the cell it came from. From there, a project's budget can be followed by both its controllers and researchers, and a dashboard shows the financial distribution and researcher allocation across multiple projects, priority areas, and funders to inform decisions within the unit. Lastly, an export function compiles the invoices sent to funders.



Things that went well

- Became really close to my partners, who were responsive even during holidays
- One of my partner's assumptions — that they're not the only organization facing this challenge — gave me the freedom to learn from other teams within Kela and explore other organizations
- Always having a **whiteboard** and **sketchbook** handy — there's a human behind this work!
- Trusting my intuition and testing it quickly

Challenges

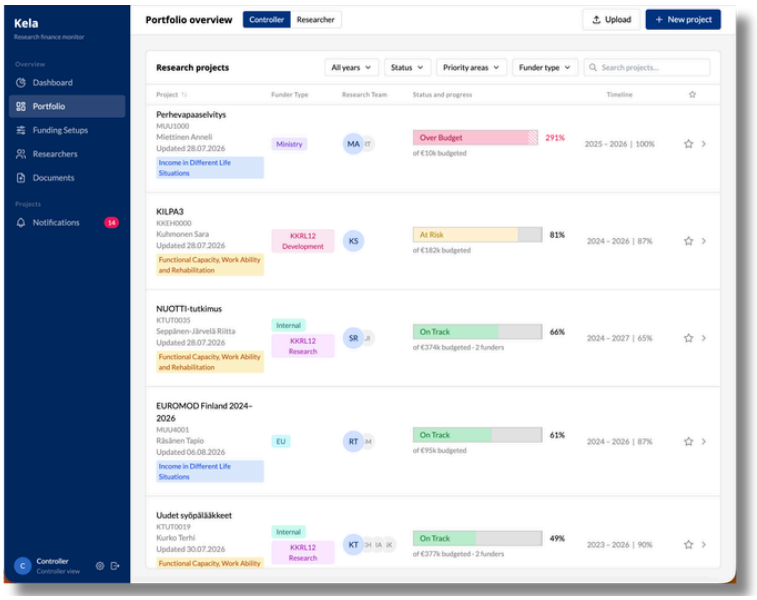
- Language barrier (non-Finnish speaker): Relied heavily on Claude to make sense of materials
- Security issues: working with fabricated materials due to sensitive data like researchers' payroll; not having a Kela organizational email; and no API access to integrate directly with the prototype.

PROCESS DESCRIPTION

My role shifted between software developer and strategy consultant, between the product-specific details and the surrounding system. In other words, I was using Lovable to both *probe* and *build*. As a *probe*, Lovable helped me position the architecture of both the problem and the organization. Much of this work involved making sense of the context, designing entry points — among them the anticipated redesign of Kela's timesheet system — for others to act on, and making the case for those moves in a report. As a software development platform, Lovable helped me *build* a prototype in close collaboration with my partners. All things said and done, this process involved: 20+ interviews, 6 site visits to the Alvar Aalto-designed Kela headquarters, a lot of sketching, and loose yet applied exploration of analogous contexts.

FINAL STATE

Working tool with gaps — e.g., no integration into existing Kela APIs because of security considerations. The project files were transferred to an IT team, who will verify the platform's security and guide its implementation. Lastly, I created a report to support my partners in carrying the tool and learnings forward.



PLAY 8: Democratize Building

8

SUMMARIZED:

Building quickly is great, but watching your partners build quickly might sometimes be even better. Help them use Lovable to articulate (and address) challenges in their work, and pay close attention to what they make and why.

On my first day, I ran a Lovable build session with my partners. They used this opportunity to develop a tool that reminds researchers to submit their hours — a choice that became a finding in itself. This set a principle the prototype has followed ever since: reduce admin load on researchers where possible. Building together with Lovable also develops competency, trust, and skin in the game, which makes maintenance and ownership easier after handoff.

Still, building should not be left entirely in partners' hands. They might know best the associated risks of not acting, but are also least likely to notice if they are falling into the trap of a local maximum. Thus, builders from outside the context can help reframe the challenge and suggest more productive alternatives to the dominant way of working.

CHAPTER 2

TRAFICOM

Transport and Communications Agency of Finland

IMPROVING SITUATIONAL AWARENESS DATA FINDABILITY AND REPORTING WITH AI

THE PROBLEM

Traficom's initial challenge was to build an AI-assisted analytics service that integrates and interprets up-to-date information to produce impactful analysis. Scoped to the width of this project, the challenge was to build a tool that can automate the reporting made by Traficom professionals.

Additionally, an external problem space was discovered during the first week of the project. The external problems consisted of the difficulty of using Traficom's statistics website. Hence, the problem and the upcoming solution had two distinct target groups: external and internal users.

THE SOLUTION

The solution itself consists of a search engine along with better UI for the Traficom's statistic websites, which mainly supports the external user group. The improved UI includes pre-calculated direct links and reports for communal professionals who need to fetch certain datasets annually.

The solution also consists of a report building tool, which supports both groups, but is targeted mainly toward internal use. The reporting tool allows users to make new statistic and situational analysis based on the sources provided to it.

Things that went well

- Splitting numeric reporting into deterministic sandboxed code runs from the LLM based reporting solved the number-tokenization
- The work assessing AI usage in data analytics became valuable additional information for the client organization.

Challenges

- Having only one initial point of contact to the client organization made reaching the right people slow and painful.
- Being neither an employee at the client organization nor a full Aalto researcher made it hard to secure proper computing resources for prototyping.

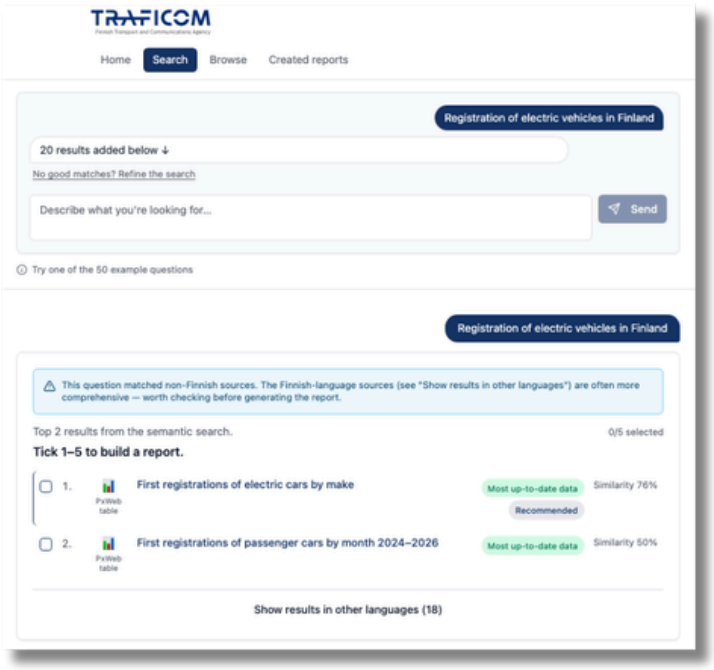
PROCESS DESCRIPTION

My process followed a rapid prototyping approach, where I began by sending email inquiries to external users and having a couple meets with Traficom representatives, but starting to build a first prototype and solidifying the first solution approach on the first week already. After a cycle of this build-test approach, I refined the scope to include more of the original internal solution, and after the pivot I have incrementally improved the solution in accordance with the approach planned during the pivot. The later user tests have confirmed that the solution meets the expectations.

FINAL STATE

A shippable product that could technically be directly implemented to the existing websites, but still needs to go through an auditing process.

In addition to the product, I have a documentation of the important findings during the project, including the information on how to utilize AI securely with numbers.



PLAY 9: Identify the Stakeholders at the Start

9

SUMMARIZED:

It is helpful for client organizations to assemble a small cross-functional team before the project starts, as without it, outsiders will spend the first half of the project discovering stakeholders by accident rather than design; if this hasn't been done for you, advocate for it at the start

When an outside consultant or a student is sourced from outside to solve a problem, they rarely know the internal structure of the organization. Asking around for the right stakeholders burns time and blurs the initial scope.

For most of my project, I got access to people inside the client organization based on the employees recommending random people they know as new people to approach. I had no clue if I had talked to the right stakeholders.

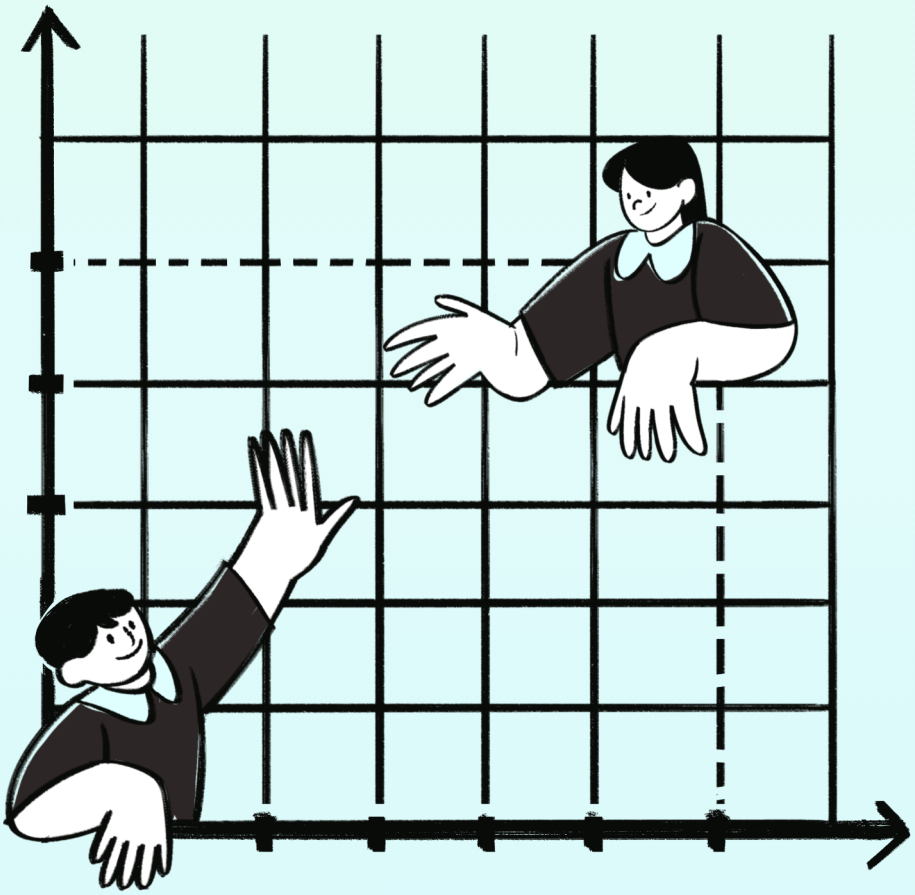
Therefore, client organizations should assemble a small cross-functional team before day one: IT, legal, communications and whoever is a relevant stakeholder. If this is not done before the project begins, ask for the list and introductions to these people immidiately.

CHAPTER 2

OPH

Finnish National Agency for Education

DEGREE UNIT COMPARISON TOOL



THE PROBLEM

Two problems. Firstly, there are 3500 active or upcoming degree units, some of which handle similar themes like sales, leadership or customer service. Currently, there is no way of knowing which units have overlap unless someone does manual comparisons. Reducing the overlap would cut down the manual work of updating units going forward.

Secondly, the smallest individual unit ePerusteet currently allows is a full qualification. However, there is a need to create micro-credentials for education providers due to the current changing working environment.

THE SOLUTION

Two tools. The first one is a comparison tool for OPH's internal use. It lets the user search for units, select them for comparison and then it automatically forms pairs and groups out of the selected bunch. The user can then inspect them further, all the way down to word or letter based differences.

The second tool is a Proof of Concept of how micro-credentials could be formed, if there weren't these legacy system boundaries. The primary users for this tool would be education providers.

Things that went well

- Building on Lovable and testing out the capabilities of the software was fun
- Testing revealed usage patterns, which would have gone unnoticed otherwise
- Simplifying and reducing steps turned out to be essential for usability

Challenges

- Limited user research and testing
- No opportunity to validate the findings against other users.

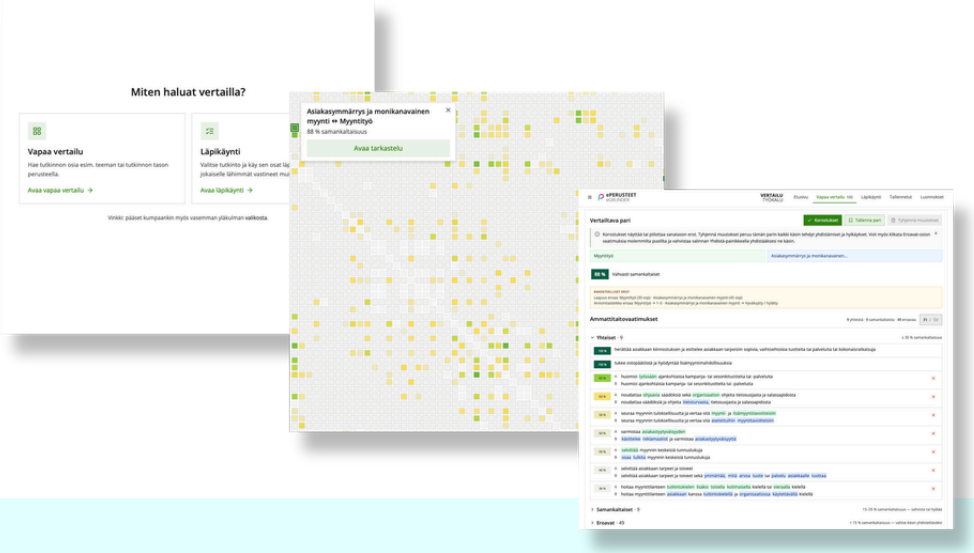
PROCESS DESCRIPTION

Instead of doing a ton of research at the start, I approached the challenge build first. I quickly build something, so that I could confirm the client and I are aligned. It was a good strategy for my case, as there were two separate problems that needed separate solutions.

Unfortunately, I underestimated how valuable user research would have been from the start, so I went a long while with only minimal user input. If I would do the project again, I would do user research right at the beginning to learn how the tool would sit in the current workflows of the users and what do the users actually need. Luckily, there was no harm done and user testing later showed that I was on the right track. Testing with users was highly valuable and helped me to not over engineer the tool but focus more on the usability and the features the users actually need.

FINAL STATE

The comparison tool is ready and usable for the client immediately. The second micro-credential builder tool is a Proof of Concept, and not an implementable prototype.



PLAY 10: Use the Cohort to Your Advantage

10

SUMMARIZED:

A mixed cohort is a resource. Even when you are the only person working on your project, the people around you carry expertise you don't have. Ask for it early and often, and treat the cohort as your team.

In our cohort we had students from multiple different fields: 4 designers, 5 engineers and 1 business student. I used the varied backgrounds and knowledge of the cohort to my advantage multiple times. I got help in the beginning to understand machine learning and embedding. I got help with user testing. I got advice on choosing the colours of my UI. And countless times I tested with the others and feedback and new ideas each time.

If you end up in a similar program, I would suggest to not be afraid to ask for help and use the cohort to make the most out of your project. I know my project wouldn't be where it is now without the help. Even if you are working on your project alone, you have a team.

PLAYBOOK

1

DESIGN (FOR) YOUR USER

Build scope around the user group, and communicate it clearly to your AI tools. Design based on observed needs, challenges, and usage situations of your audience, not assumptions.

SCREENSHOT TO PROTOTYPE

Don't start from a blank canvas when there's already a system to improve. Recreate the existing interface and visual language in Lovable using screenshots, then iterate on top of it.

2

3

LOOK BEYOND AN APPLICATION

If initial scope is wide and involves stakeholder politics and legislative restrictions, the solution often lies beyond the design of one AI-powered app. Consider the cultures that the prototype embeds itself within, and identify what needs to change for successful implementation.

BUILD TO LEARN, REBUILD TO SCALE

The first thing you build is not the product. It is the first step in finding out what the product should be. Treat version one as a probe, and put a date in the calendar where you ask honestly whether what you have is a foundation or a rehearsal.

4

5

LEARN TOGETHER WITH THE TOOLS

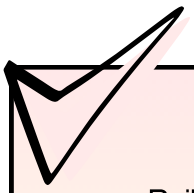
Build your own and the AI-tools understanding of the project in tandem. Sometimes AI-tools try to rush you into solutions. Don't hesitate to take a step back and plan the bigger picture before starting to build a new feature.

LEVERAGING MOCK DATA

AI-generated test documents can fill in for real data in restricted projects, enabling model learning and development without confidentiality concerns. Generate diverse mock data to prevent training biased models on limited test sets.

BEFRIEND CLAUDE AND TRUST IT

Plan first, then trust the build. Co-design a list of design and collaboration rules for your AI to follow, and update it as your context evolves. Use AI to stress-test, clean, and audit the build, and challenge its decisions and assumptions against these rules.



DEMOCRATIZE BUILDING

Building quickly is great, but watching your partners build quickly might sometimes be even better. Help them use Lovable to articulate (and address) challenges in their work, and pay close attention to what they make and why.

IDENTIFY THE STAKEHOLDERS AT THE START

It is helpful for client organizations to assemble a small cross-functional team before the project starts, as without it, outsiders will spend the first half of the project discovering stakeholders by accident rather than design; if this hasn't been done for you, advocate for it at the start

USE THE COHORT TO YOUR ADVANTAGE

A mixed cohort is a resource. Even when you are the only person working on your project, the people around you carry expertise you don't have. Ask for it early and often, and treat the cohort as your team.

CASE + PLAY MATRIX

Each case card names the play that defined it, but most of us leaned on more than one across the summer. This matrix marks every play each builder actually drew from, self-reported by the ten of us.

	CASES										
PLAYS		Osaamispolku	Varda	Migri	GAINS	THL	Poliisihallitus	Museovirasto	Kela	Traficom	OPH
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	10										

CHAPTER 3

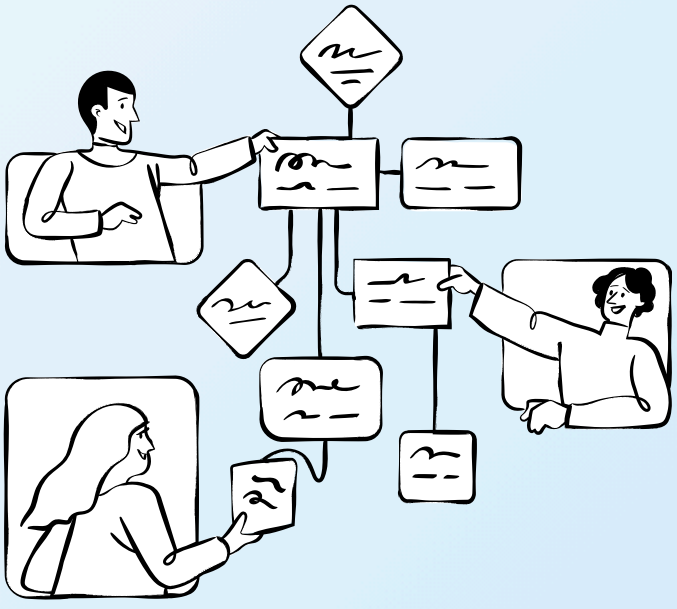
WHAT WE LEARNED

SETTING UP FOR SUCCESS
QUICK START GUIDE TO AI
REFLECTION

CHAPTER 3, SECTION 1

SETTING UP FOR SUCCESS

HOW TO PREPARE FOR THIS PROJECT
AS A PUBLIC ORGANIZATION



HOW TO PREPARE FOR THIS PROJECT AS A PUBLIC ORGANIZATION

Almost every issue that arose across the projects stemmed from things that weren't clarified before the program started, whether that was the **problem itself, the stakeholders, or access to data**. This section covers what to settle in advance, across those three areas.

Before the programme starts ensure:

- The problem is clearly defined, and the deliverable type is agreed on
- Availability of a representative to meet weekly with the student
- Users are identified and available for testing
- Data is either open, mocked, or access has been agreed on

Watch out for:

- Multiple stakeholders with conflicting views
- Classified data with no realistic access
- Holidays that haven't been planned around in advance

The problem, not the solution, should be clearly defined from the start. There should be room for ideation, creativity, and out-of-the-box thinking; offering multiple challenges rather than a single fixed brief can help with this.

The deliverable type needs to be agreed on before the program starts. If the goal is a working tool or a shippable product, the problem should be narrow and well defined. If the goal is to explore solutions, a wider problem space that the student can narrow while working is fine. Either way, expectations should be set to match. Organizational and legislative constraints should also be thought through in advance, since they shape what even is realistic to create.

Scope should be expected to move. Frequent communication between client and student is what makes that manageable. Even when scope narrows, keep sight of the original scope, both for architectural decisions and as a future recommendation once the program ends.

Even when everything has been set up well, the problem and scope can still evolve and there should be room to pivot even in a well-set-up challenge.

PEOPLE

Time for a weekly meeting between client and student should be alotted from the start. Decisions are hard to make without client feedback, and the lack of it slows the project down more than almost anything else.

Frequent communication, client guidance, and feedback are what make flexibility and progress possible during the project. If there's more than one stakeholder involved, their views on scope should be aligned before the program starts, and any other relevant stakeholders should be notified of the project and onboarded if needed.

An eight-week build assumes user needs are already established, or that users are accessible enough to establish them immediately. The user group should be clearly defined and its availability for testing confirmed across the whole programme.

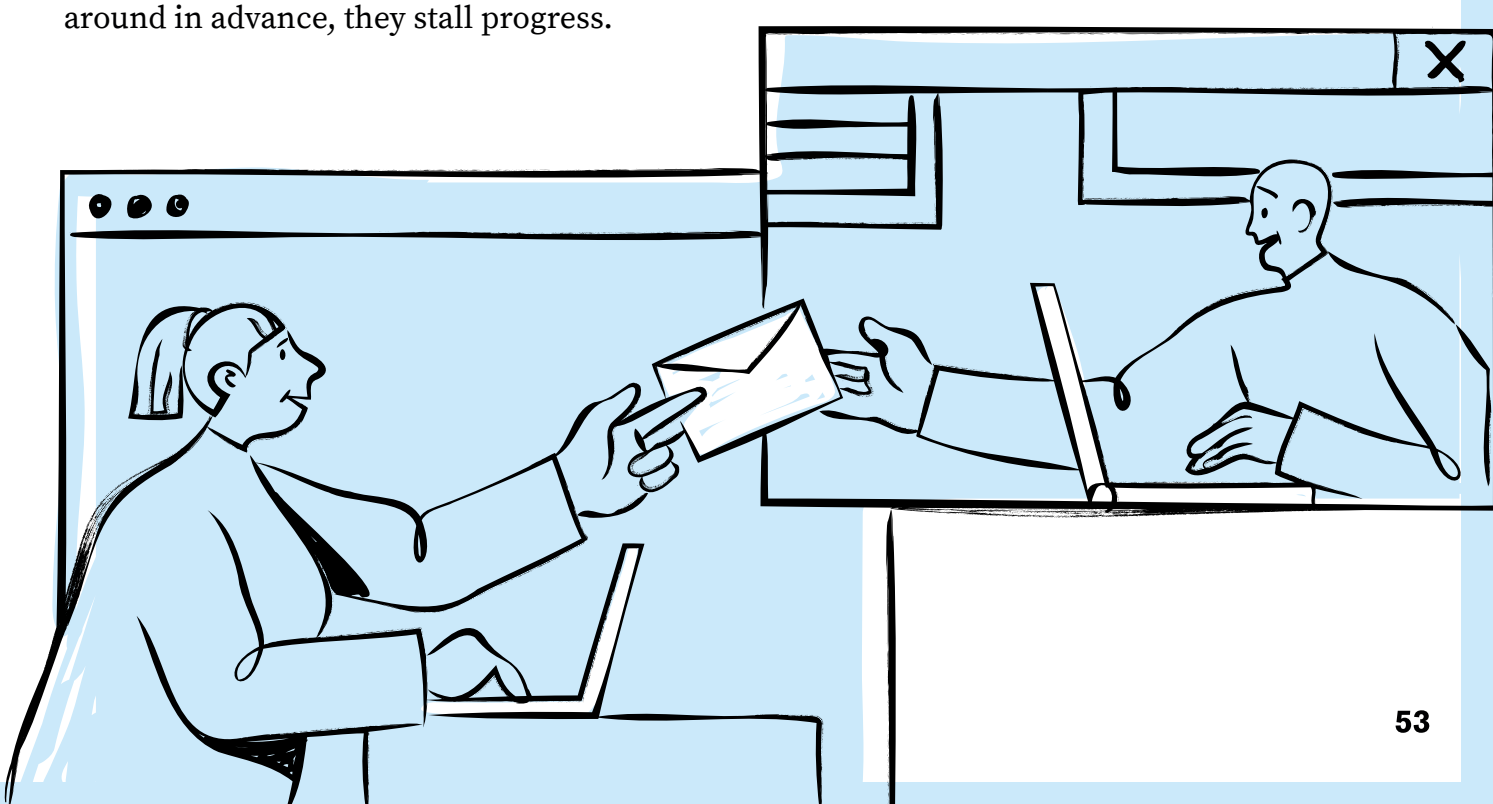
All client, stakeholder, and user holidays must be communicated before the program starts, if they're not planned around in advance, they stall progress.

ACCESS

The data, integrations and any access agreements a project will need should be mapped out before the program starts. Data should be open or externally available, if no real data is available, mock data should be prepared and provided from the start rather than becoming something the student has to chase down.

How much access is available also sets a ceiling on how finished the end product can realistically be. Without access to the real data or the systems the tool is meant to integrate with, a fully working product usually isn't possible, and the outcome is more likely to be a concept or a prototype than something ready to hand over.

If an additional techonlogies contact will be needed, ensure they’re named and their avability is confirmed.



CHAPTER 3, SECTION 2

QUICK START GUIDE TO AI

HOW TO SCOPE THE USAGE OF AI
AND MAKE GUIDELINES
FOR THE PROJECT

GETTING STARTED WITH NO-CODE AI TOOLS

No-code AI tools accelerated prototyping. Most participants optimized results by combining Lovable with Claude via GitHub. Lovable enabled rapid iteration, UI/UX design, and access to integrated databases, while Claude managed planning, back-end development, agentic building, and debugging. AI use ranged from traditional automation and semantic search to local and API-based LLMs, with each participant defining guidelines around data access, model choice, and the review of AI-generated code.

Lovable is best suited for interface-led work and projects built on public or open data, where the aim is to demonstrate a concept and to learn quickly. **Development requires more care if** work depends on a local or air-gapped model, restricted data, or on substantial back-end and data-pipeline logic. In these cases Claude Code excelled.

TIPS FOR GETTING STARTED

- **Connect Lovable ⇄ GitHub**, so you own the version history and a route to production.
- **Create a design system in Lovable** using the client's brand guide or screenshots of their website to unify the visual identity with their brand
- **Verify qualitative insights with real users**, and test a variety of use cases to fully understand how the tool handles different situations
- **Use AI to review the code** it has produced for dead code, security issues, UI/UX improvements, efficiency, organization, accessibility, etc.
- **Prepare for handoff** of the project and code early, and think about all the connections built into the tool and how they can be maintained effectively long-term

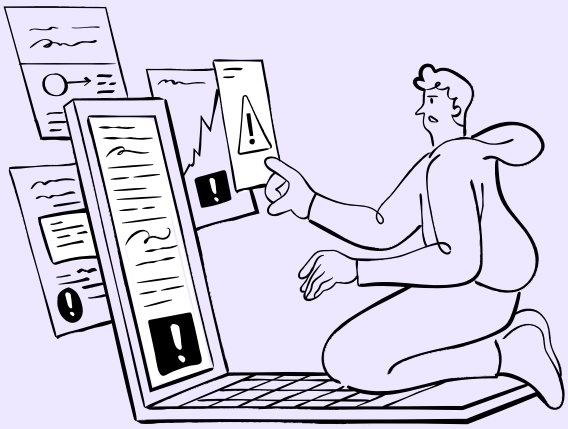
KEY TOOLS

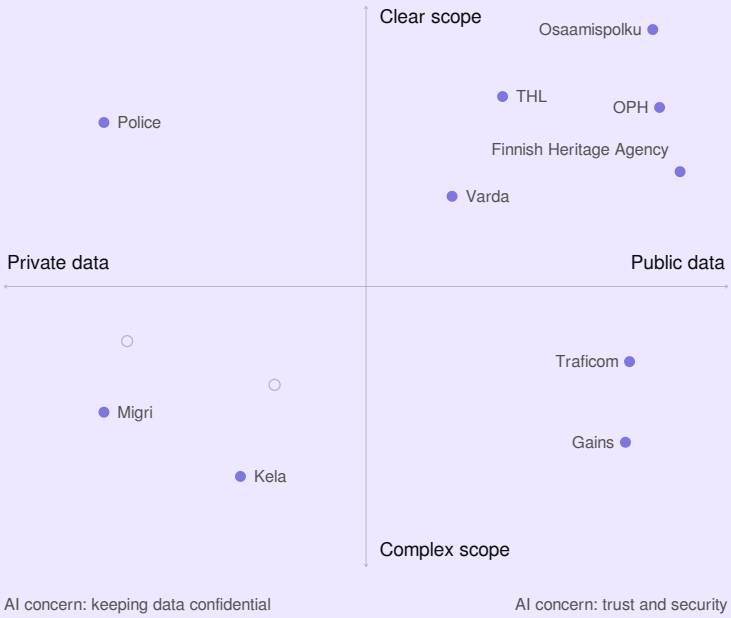
Tool	Role in the build	Working tip
Lovable	Interfaces, rapid UI/UX iteration, data-integration wiring	Build a proto early, you will be surprised by how fast you can have an MVP
Claude	Planning and architecture dialogue, back-end, scrapers, embeddings, data pipelines, and code debugging.	Do your planning in a dedicated project folder, and remember to push updated to GitHub to avoid merge conflicts
GitHub	Two-way sync for Lovable and Claude; full version history for backup	Connect it early to allow for one shared version of the code with version history
supabase	Stores and serves project data	Confirm what data can be stored in Supabase; consider keeping it read-only

RESPONSIBLE USE OF AI

As important as the projects themselves is the analysis and reflection on the responsible use of AI in public sector projects. In this section, “using AI” includes both the usage of AI tools such as Claude and Lovable as well as built AI solutions for the clients.

Our projects could be divided into four quarters of a matrix with one axis being the publicity of the data we were provided with or accessed during the project, and another axis being the clearness of the initial scope given by the client organization. Along these groups, the concerns related to AI use differed a lot. Cohorts working with private data mostly worried about keeping information confidential and out of external tools, while cohorts working with public data focused on broader trust and security concerns.





On the security side, a major concern was the accidental leakage of proprietary information, such as API keys, when using AI tools or scraping unwanted information. The source of risk in these events came from AI tools’ default behaviour. In projects like this, it is the cohort’s duty to be sceptical about giving materials and information to AI tools. On the regulatory side, only one organization ran the project through their legal team beforehand to discuss the legality of public website scraping, even though multiple cohorts scraped different websites and APIs and independently researched the legality and good practices related to scraping.

Trust-related concerns in using AI

Most of the cohort have blindly trusted the Lovable- and Claude-created code, and few have even checked the code at the repository-structure level. Most cohort members mandated that their AI tools, such as Claude, explain the technical solutions to ensure that the built solutions are trustworthy. As an exception, most cohort members did not utilize AI for interview analysis. This was due to AI’s poor understanding of the problem space and legal concerns with putting un-anonymized interviews to Claude or Skimle.

LOCAL LLMS

Three of our projects utilized local LLMs in their solutions. These have the advantage of allowing private data to be completely auditable and implementable. Two of the three projects using local models chose them because of their additional benefits, including the improved trainability and accuracy of the LLM in a specialized context compared to API integrated commercial LLMs, and the scalability of local models without additional costs. The cost structure of local LLMs consists almost entirely of the hardware used to run them, while commercial API connected LLMs are mostly billed based on usage.

CLIENT GUIDANCE

The client organizations generally did not give guidance on using AI within the projects, nor did they define the safeguards. Where guidance did exist, it usually correlated with how clearly the project was scoped by the client instead of the privacy of the data being handled. The initial scope of most projects did not clarify the type of AI that should be used for the problem, yet almost all project descriptions mentioned AI as a solution, showing that organizations wanted to see what AI tools are capable of without understanding it as a broad category of different tools and methods.

While all projects used AI tools such as Lovable and Claude, the actual solutions differ, ranging from traditional automation solutions, web scrapes, non-LLM neural networks (such as semantic search), local open-source LLMs, self-trained projection heads, commercial embedders, general LLMs (via API), to specified LLMs (like ones creating Python code). The cohorts independently defined and managed AI usage guidelines, and some projects revolved around the responsible use of AI in a way that mandated defining what data reaches an LLM, what kind of LLM is used for the solution (local, API, not at all), and how LLM-created code should be analyzed either by humans or by other LLMs.

We suggest that in future projects like this, client organizations make clear guidelines on the use of AI beforehand. In practice, client organizations should consult their legal teams and make executive decisions that can answer these questions from the beginning:

What data – including public data – can be given to which AI tools?
How should AI created code or documentation be inspected? Including what resources the client organization can allocate for this auditing process?
What type of AI should the solution utilize: only traditional automation, local neural networks, local LLMs, commercial LLMs? Just saying “AI” is not enough.
What resources and hosting budgets can be allocated for the project?

CHAPTER 3, SECTION 3

PROGRAM MANAGER'S CHECKLIST

HOW TO RUN A PROGRAM LIKE THIS

A PROGRAM MANAGER'S CHECKLIST

This playbook is something I wish I had had before starting this program. If you are considering running something similar, read it completely. Here are my tips for specifically running a program like ours.

BEFORE THE PROGRAM

Recruit the team. Building with AI successfully requires different skills: mainly design and people skills and comfort with technical topics. Individuals aren't perfect, so hire a diverse, complementing team.

Watch out for the holidays. Several projects were bottlenecked by the July holidays. Clients, users and third parties were absent. Give extra attention to mitigate the bottlenecks.

Talk with your legal team early. Partnership terms, NDAs, and who owns the code.

Make the clients do their homework. Have them read the Setting up for Success section of this playbook. Meet them and ensure everyone's aligned.

Make a plan and set general milestones for the projects. Even if you go off track it's still better to have *a* plan and schedule. Milestone examples: MVP shown to the client, deep understanding of problem context achieved, 3x user testing sessions completed.

Check what AI tools are available to use and who pays for them. Our program benefitted from having multiple different tools for use.

STARTING THE PROGRAM

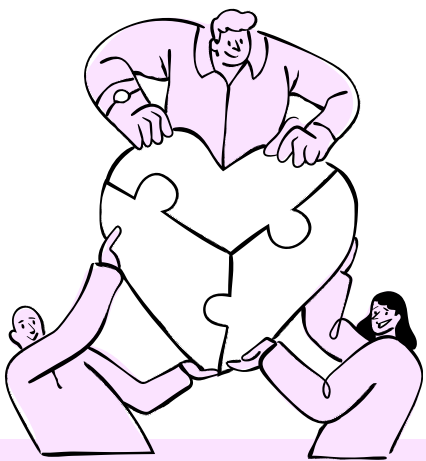
Give the first day to settling in. Handle the HR technicalities, then spend the real time on people getting to know each other's backgrounds and skills. The sooner they're comfortable, the sooner they work as a team. We under-invested, and several students said the cohort only became useful to them later.

Day 2 kickoff & joint building session. Get everyone involved in the same room. Introduce the projects and have a mini hackathon where everyone builds. Make sure the students and clients align schedules for the full program here.

Use Slack for internal communication. We used WhatsApp and it was bad.

Walk through the schedule and milestones with the participants. Communicate the big picture and ambitions of the program.

Organize some mini-lectures. We had a design, software engineering and public sector person come in. They provided motivation and perspectives on how to achieve success and avoid typical pitfalls with the projects.



RUNNING THE PROGRAM

Run a daily meeting. What I did yesterday, what I'll do today, any bottlenecks — one or two sentences each. No feedback during it; people help each other afterwards. Keep it short.

Weekly review on Thursdays. We started reviews on Fridays, but changed it so people have time to react to peer-feedback before the weekend.

Split the review into smaller groups after the beginning We had everyone reviewing everyone first, but then split into smaller groups. Then people had a team able to provide higher quality feedback to them

Hold an UI/UX workshop where students give feedback to each others products. UI & UX are things where everyone can help regardless of the solution itself.

Push towards user testing. Showing something to a real user too late was the cohort's most repeated regret.

Push for solving bottlenecks. Students might be too polite to solve bottlenecks caused by other parties. Help them with rapid communcation if needed.

Keep the big picture in view. When building starts, people may lose track of the big picture. Check that the general milestones will be achieved.

Organise afterworks, and one bigger thing mid-program. Ours was a day trip to the Lovable offiece in Stockholm in week eight, which was definitely a highlight. Effort to build the team spirit pays off!

ENDING THE PROGRAM

Set a clear boundary for the project work. Decide when the project has to be done. With AI tools you could iterate forever. We didn't have a clear end date and it showed. Having constant communication with the client helps the projects reach a finished state.

Have a debrief when everything's over. An AI building project can be very intense, so it's good to get real closure.

Coach the handover. Students know their project but not how to pass it to a development team they've never met. Go through what to document, what to demo and who to send it to. Preferably set a hand-off meeting with the client.

WHAT HAPPENS AFTER

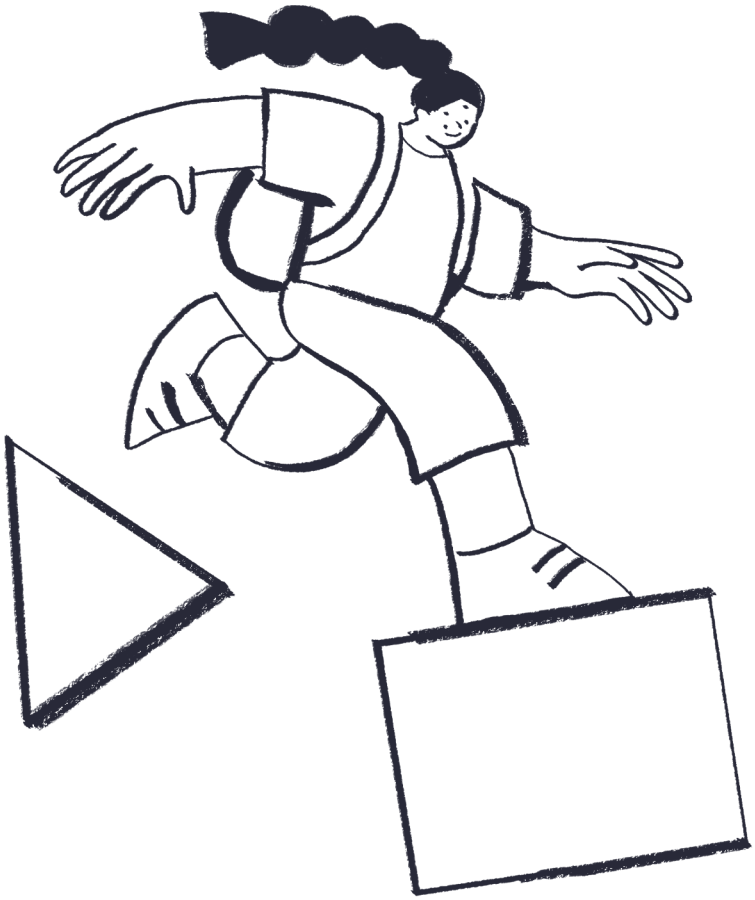
HOW TO MOVE FROM
PROTOTYPE TO PRODUCTION

TAKING IT FORWARD

To return to Sitra's broader goals at the level of a larger change, Mika Ruokonen's (2025b) manual for the public sector, "The AI Transformation Task," cites McKinsey's (2025) report finding that ~80% of companies that have adopted AI have ultimately not achieved improved operational margins. MIT's (Massachusetts Institute of Technology) widely cited NANDA study estimates an even higher level by stating that around 95% of the pilots never show up on the bottom line.

Sitra (2025a) also addresses this problem and the possibility of efficiency paradoxes. In many cases, resources are lost in fragmented experiments that never move into real implementation, let alone scaling. Our AI program could therefore be linked to Sitra's broader renewal goal to address this: **link the project KPIs directly to the real value,** lock the gains into the budget rather than measuring the pilot, and especially reuse shared tools, standards, and ways of working across many projects instead of each building their own in silos. Sitra's productivity model exists precisely because this pattern of failures is so well known in the public sector (Sitra, 2025a).

This aligns with the programs core question: not only whether ten students can build convincing prototypes in eight weeks, but also whether the underlying project model of rapid prototyping with public sector partnerships offers a scalable tool for public integration.



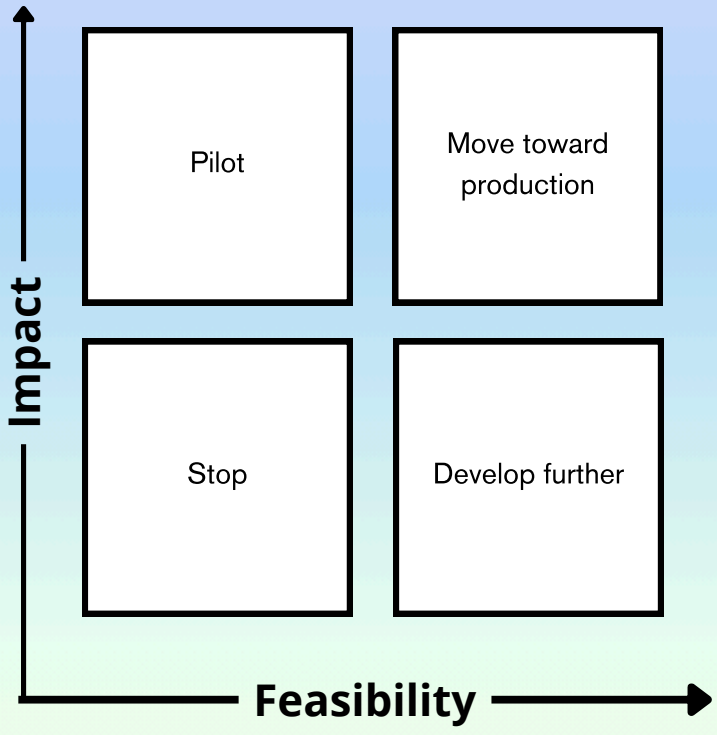
Both our and Sitra's eyes have always been on the future. Future impact dictates public sector efficiency. Without the next step, we will only depict a situation of big words without any action. It's time to walk the talk.

Integrating this pilot into the broader productivity model proposed by Sitra has not yet been tested on a large scale. The uncertainty regarding the next step stems from organizational readiness, not prototype viability. When asked what moving our tools into production would actually require, we were able to sketch this direction fairly well.

As highlighted by Sitra (2025b), **adoption requires leadership, momentum, clear ownership, and a deep pro-change culture.** The leadership of public organizations must be ready to provide functioning settings for rapid and innovative prototyping, and give people the power to act.

Sitra (2025b:81) distinguishes narrow pilots from organization-wide initiatives. Most cohort prototypes sit closer to the narrow end, even though the ambitions of many also fit the second category. The Sitra guideline is that proven functionality must be standardized and multiplied with common components to scale value (Sitra, 2025a:24).

A four-point evaluation matrix shown on the right guided our post-prototype evaluation. Most participants aimed for production, but doing so exposed a critical bottleneck: a lack of organizational ownership. As Sitra (2025a) notes, successful scaling requires a named, committed owner from the outset. When handovers lack clear authority, projects risk stagnation.



The four-point evaluation matrix

Beyond ownership, transitioning to production in the public sector requires overcoming technical and security hurdles, such as migrating off no-code platforms, auditing code, and verifying data classifications.

KEY TAKEAWAYS FOR PUBLIC ORGANIZATION LEADERSHIP:

- Ensure a decision-maker with real authority is involved from day one
- Groundwork for post-pilot transitions must be scoped before prototyping begins, rather than leaving students to force connections.

Bibliography:
McKinsey (2025). The state of AI: How organizations are rewiring to capture value. McKinsey & Company.
MIT/NANDA (2025). The GenAI Divide: State of AI in Business 2025. MIT NANDA research initiative.
Sitra (2025a). Heliskoski, J. Tuottavuuslähtöisen uudistumisen toimintamalli. Sitra, November 2025.
Sitra (2025b). Ruokonen, M. Tehtävänä tekoälymurros: Käsikirja julkisen sektorin johdolle. Sitran selvityksiä, December 2025.

THE COHORT

Aino Alkio

Coleman Piburn

Eero Kuokka

Elena Silvola

Kaarlo Mustonen

Kaisa Seppälä

Olli Särkilahti

Paul Grandchamp

Salla Siira

Tiia Mäkinen

Organizers:
Valentin Jutila - DIEM Aalto University
Jens Schmidt - DIEM Aalto University
Kirsi Hanhisalo - SITRA
Aino Bergius - Lovable

Participants:
Aino Alkio
Coleman Piburn
Eero Kuokka
Elena Silvola
Kaarlo Mustonen
Kaisa Seppälä
Olli Särkilahti
Paul Grandchamp
Salla Siira
Tiia Mäkinen

Aalto-yliopisto

x

SITRA

x

Lovable

Playbook:
Written by participants
Layout Kaisa Seppälä

"Horses don't stop they keep going"
Valentin Jutila