The story about Eksponent MØDER

Building a hardware device using my new AI superpowers



Christian Dalager · Eksponent · IDA, 17 November 2026

Agenda

01 About me

02 Enter AI

03 Personal projects

04 Eksponent M0DER

05 Diving into the work

06 Gotchas



About me

I have been building software for a long time.

- Across all the tiers of the traditional stack
- Digitalization in private companies, organizations and the public sector
- Integration-heavy, web-based architectures with backends of varying complexity

---

I know very little about hardware, though. So I stand before you with a lot of humility.

Part 02

Enter AI

The way we work in the software development pipeline has changed a lot in recent years.

We use it to build things.
And we use it in things we build.

Two entirely different things, but related through the AI technology underneath and the complications faced when applying it to solve problems.

Everyone is scrambling for footholds: individuals, companies big and small, frontier labs

How do we best use AI to build high quality software at scale?

The realization is that we don’t know yet.

While we wait, we extract value from what we have right now

Use the models

The current models are insanely good.

Borrow best practices

Pick the prevalent ones from people more clever than us and companies more resourceful than us.

Build our own tools

Around our own processes, to improve quality and feedback cycle times, augmenting existing proven practices.

Part 03

Personal projects

When I don’t work

I like to do something COMPLETELY different.

But apart from reading and a little guitar playing, my primary hobby is also doing stuff with computers. And I dont think I am the only tinkering builder in this room 🧐

MY PROJECTS FOLDER

Things I tried, failed with, learned with, often forgot and occasionally shipped.

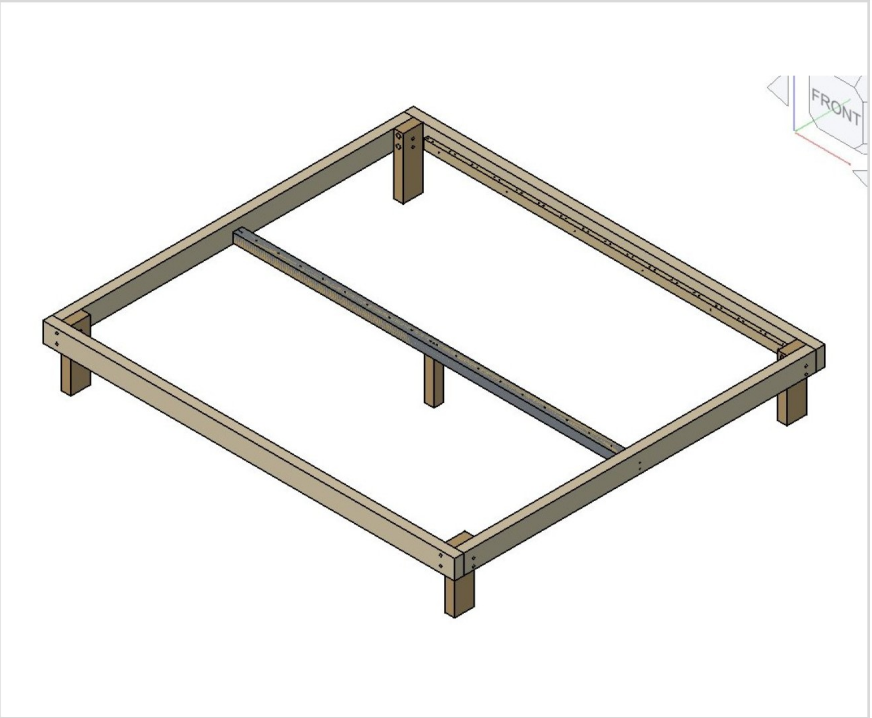
AI tools have increased my reach into areas where I previously had limited success, due to education, experience and attention span.

How far will this take me? Where are the limits?

One of the exciting things to reach into is The World of The Real: the actual, tangible, touchable things.

Designed a bed frame with Claude and FreeCAD. And built it.

github.com/dalager/diy-ai-futon



01 The frame, modelled



02 With slats

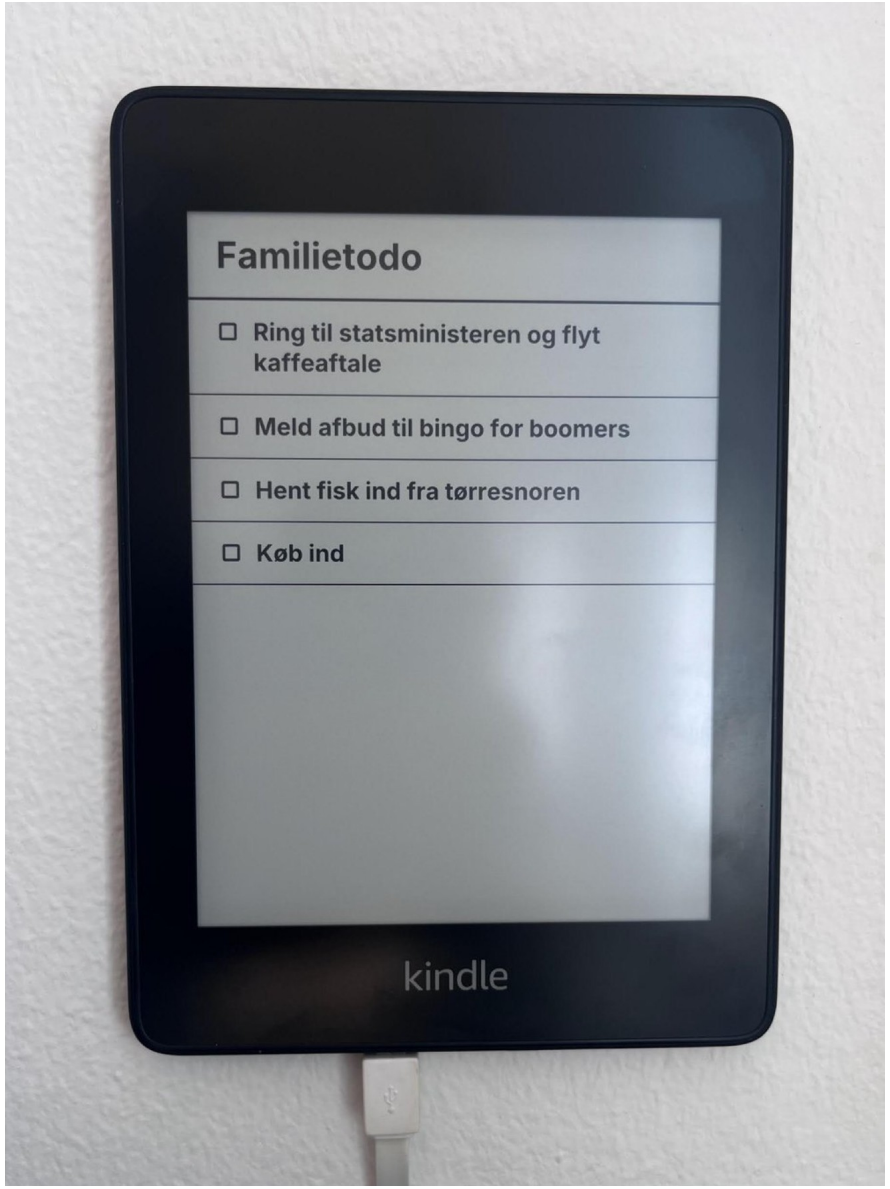


03 In the real world

Turned my old Kindle into a wall-mounted todo list.

An e-reader that now renders the family todo list on the wall.

github.com/dalager/kindle-todo



AND THEN, ON IMPULSE

I bought an ESP32 module with a touch display and a rotary knob.

At AliExpress, after watching a YouTube short about someone putting a small game on it.

SIDENOTE

I have been working on a transcription engine…

It is used in physical therapy clinics, where the physicians need to be very hands-on in client sessions.

My role

Danish transcriptions in the best possible quality, at the lowest possible cost per session.

The scale

Around 1000 hours of transcription every day.

So I already had a hot transcription endpoint running that I could throw things at…

TRANSCRIPTION DEVICE · FIRST IMPLEMENTATION

The “Puck” as a session recorder.

In front of the existing transcription pipeline. But for various reasons this wasn’t practical, and overall a bad fit for the device.



First prototype: the Hudpin Pill

So I did what all startups do at some point. After looking at a coworker's dictation device, I thought:

We can do this, but better!

To be clear: this is not a real product. It is not in use, and it will not be.

I wanted to see how far I could get with some domain insight, quite a bit of software experience and new AI superpowers.

Part 04

Eksponent MØDER

A meeting recorder. The idea is simple.



The idea is simple

01 · Device

Record

A hardware device records and stores WAV files locally.

02 · Device

Send

It sends the files to a backend.

03 · Backend

Transcribe

The backend transcribes the audio to text.

04 · Backend

Summarize

With a fixed prompt, an existing template selected from the text, or an autogenerated prompt.

05 · Device + web

Sync back

Both outputs get synced back to the device. A web page shows the transcriptions too.

On the device



Record

Timer, pause and stop



Recordings

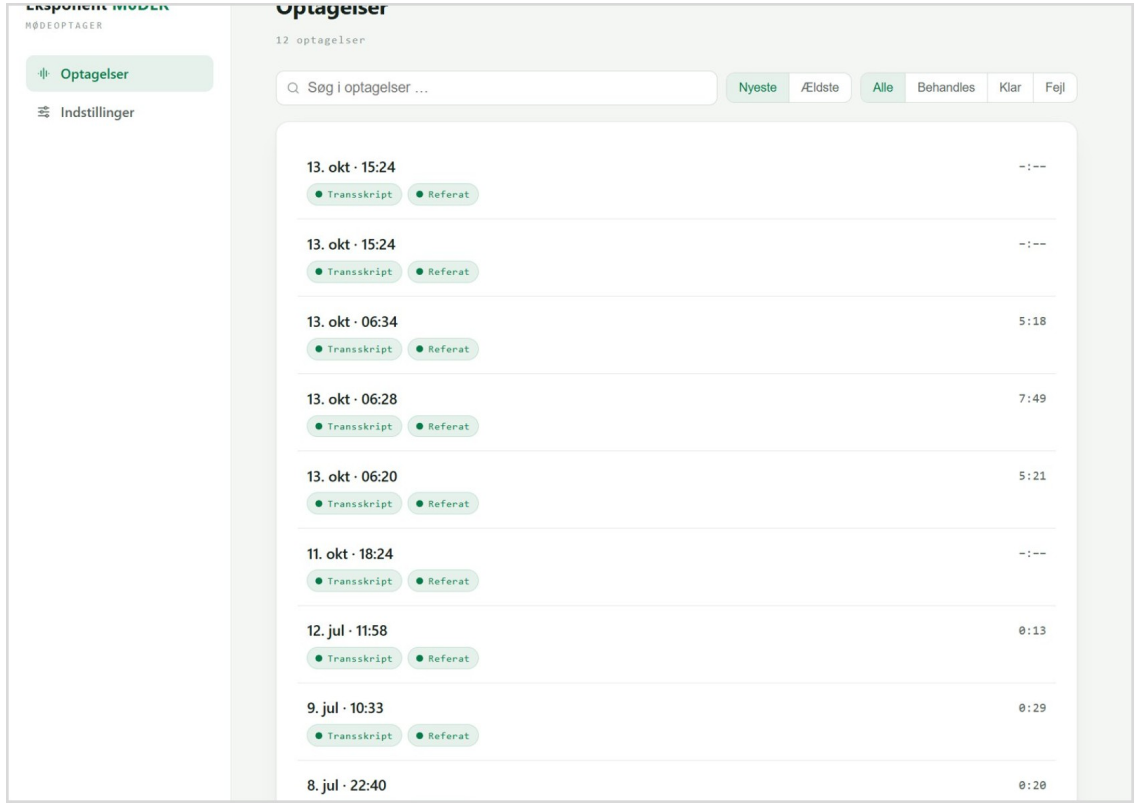
Stored locally, synced to the backend



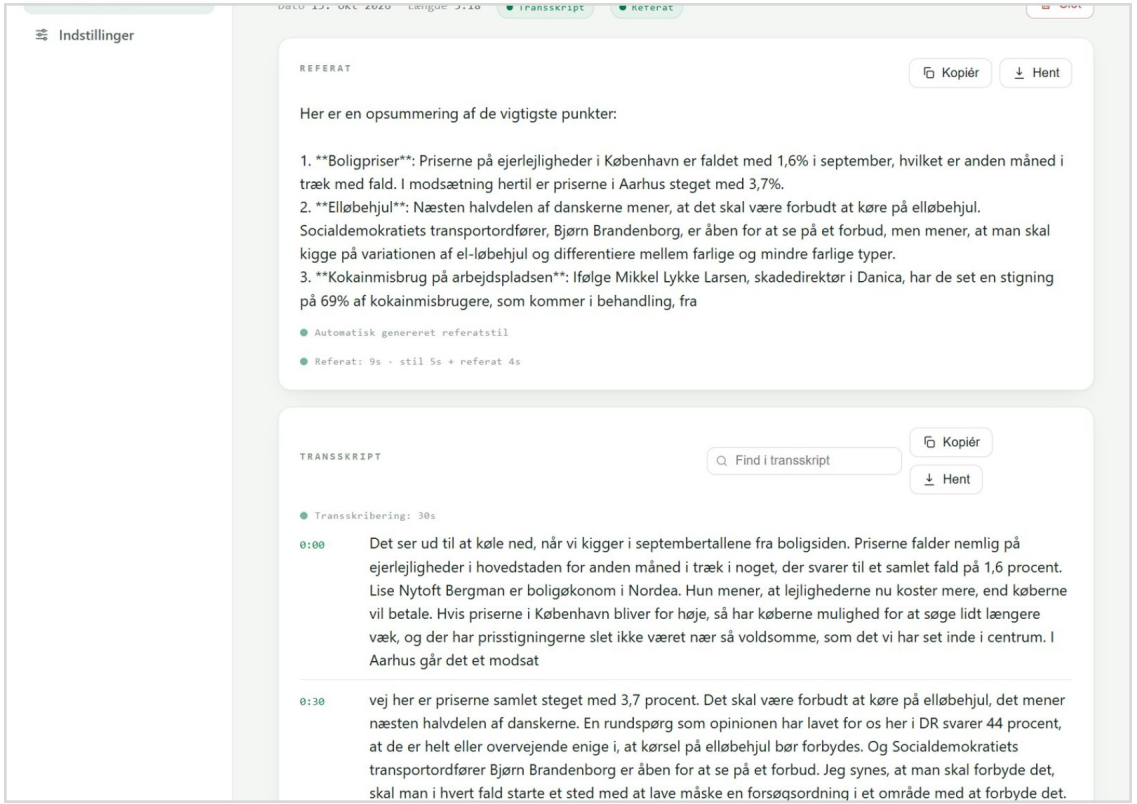
Settings

Wi-Fi, brightness, PIN lock, account

And on the web



Recordings Every session, with transcript and summary status



One recording The summary, then the timestamped transcript

Three ways to summarize

01 A fixed prompt

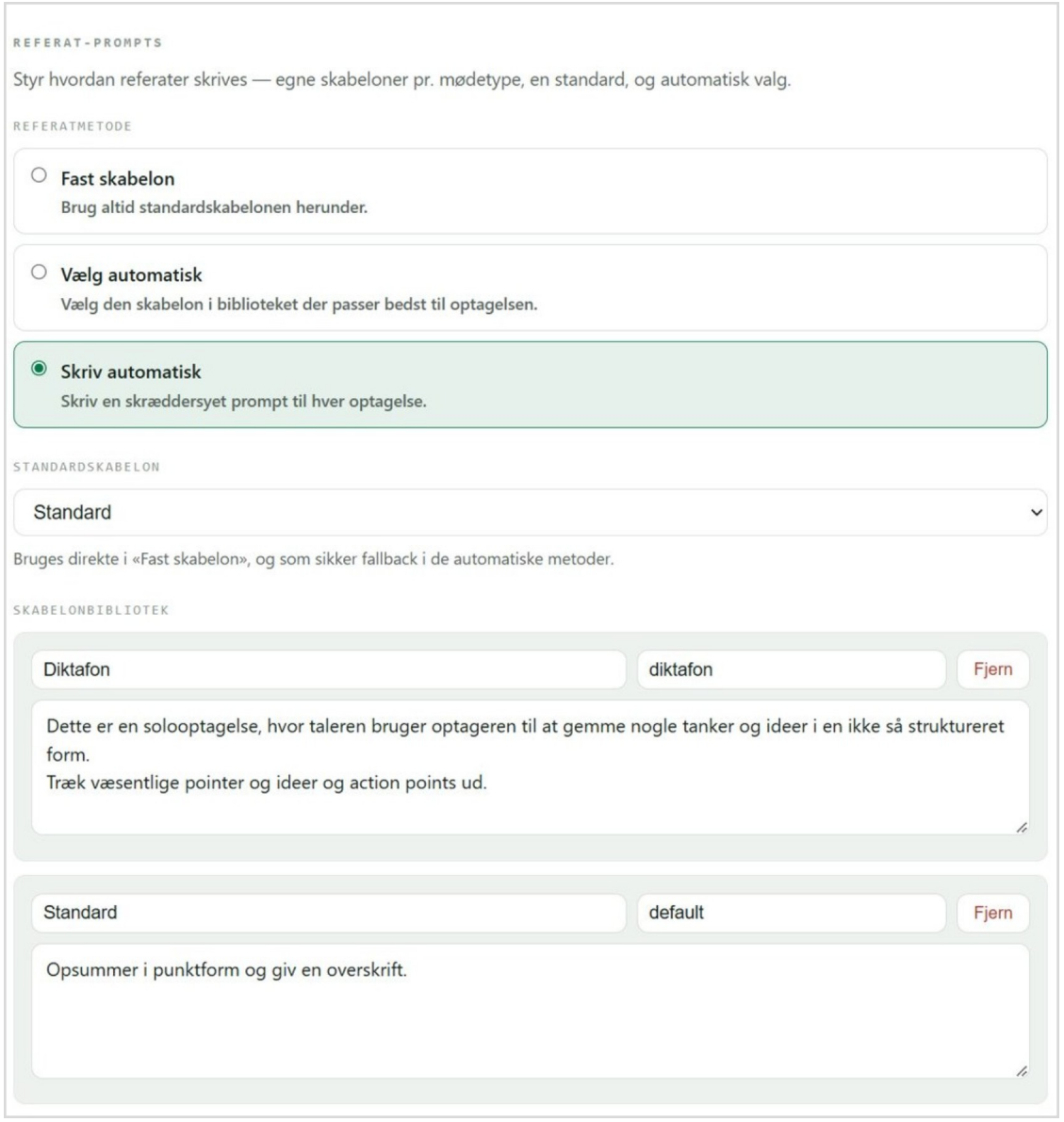
Always the same summarization template.

02 Select a template

Pick an existing template, based on the transcription text.

03 Autogenerate the prompt

Write a summarization prompt from the transcription text, and run that.



The software is a bit more involved

Device firmware

- UI
- Wi-Fi
- Security
- Memory constraints

Backend architecture

- AI inference endpoints: transcription + summarization
- Queues
- Storage
- Security

Today we focus on the device, and how I actually built it.

Part 05

Diving into the work

I have played with Arduinos and Raspberry Pis before, but that’s about it. So bear with me.

Primary challenges, for me

Hardware

No experience with, or usable understanding of, hardware architectures

Memory

No experience working under the memory constraints of embedded systems

C

No usable experience with C (a lot of other programming languages, though)

UI

No experience with UIs that are not web-based, and especially circular UIs

Test

How do we test this?

Speed

How fast can we go?

SOLUTIONS

Hardware dev

01 **Download all the documentation**

Everything available for the module: PDFs, schematics, everything.

02 **Search GitHub**

Existing projects working with the same module.

03 **Find AI skills**

Skills that encode the experience of real embedded developers.

04 **Deep probing of the connected device**

Let the agent ask the device about its specs. A very important supplement to the docs shipped from AliExpress.

SOLUTIONS

C, UI and test

C

Trust the model.

UI

Use Claude Design to make something pretty, and export it as a design handoff

Iterate on established UX patterns for round displays

Lean heavily on the LVGL UI toolkit

Test

Unit tests

Functional tests: drive the UI and take screenshots

Outside verification with a “verify with webcam” skill

Gotchas

01 **Memory management is hard**

Tell the agent to work with a memory budget.

02 **Stack overflows are a thing again**

03 **The UI thread cannot block**

Delegate.

04 **Feedback loops are VERY slow**

Separate pure logic from hardware code and cover it in unit tests.

Use the most intelligent, most expensive agent for adversarial reviews.

05 **Fonts are generated as small as possible**

Unexpected characters will [] on you.

FINAL THOUGHTS

A lot of software work

01 Product shaping

02 Architecture

03 Coupling, cohesion and modularity reviews

04 Testing and verification

05 UX/UI design



I have not written
a single line of code.

SO MANY QUESTIONS

Do I feel Shame? Pride? Empowered?

Is this how my professional life will be now?

How do I feel about that?

What is of value here?

The personal value is very high

As a vehicle for reflection, learning and meeting new people like you.

But anyone can clone it

If I can make this with no effort, everyone can clone this with even less effort.

One thing is certain

The personal hardware appliance space is about to explode.

Everyone is putting their boutique devices online, without necessarily thinking about things like safety and cyber threats. Or freezer temperatures...

Questions?

How far will this take you?

Projects

github.com/dalager/ida-embedded-workshop

github.com/dalager/diy-ai-futon

github.com/dalager/kindle-todo

Me, elsewhere, reach out and connect 😊

linkedin.com/in/dalager

eksponent.com

github.com/dalager

christian.dalager@eksponent.com

christian@dalager.com

