What It's Like to Build An Open, Repairable Laptop

Daniel Schaefer
Framework Computer

Ubuntu Summit 2023
Riga, Latvia

Who am I

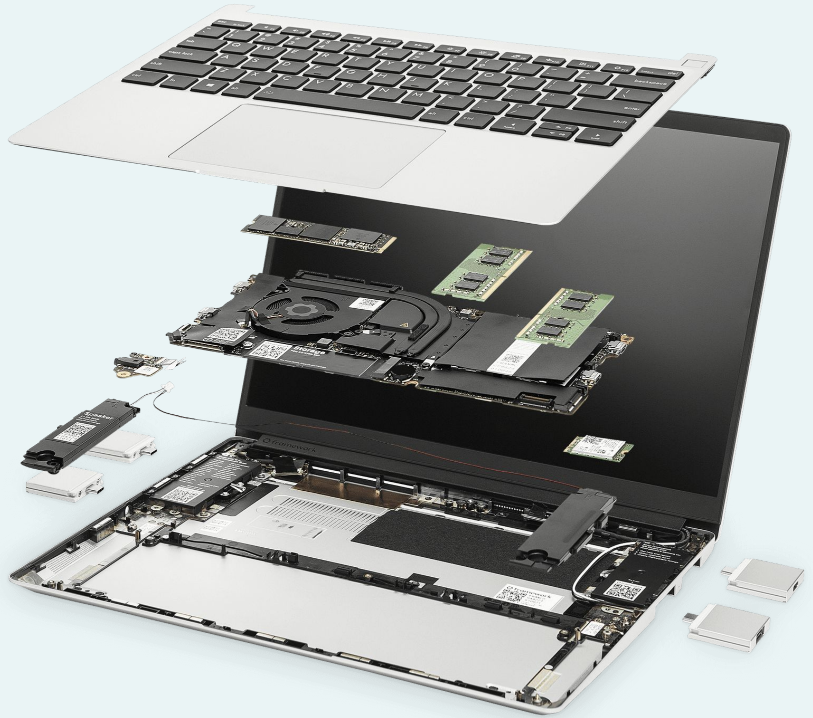
- Daniel Schaefer (Schäfer)
- System Software Engineer at Framework
- German, living and working in Taiwan 🇹🇼

What is Framework

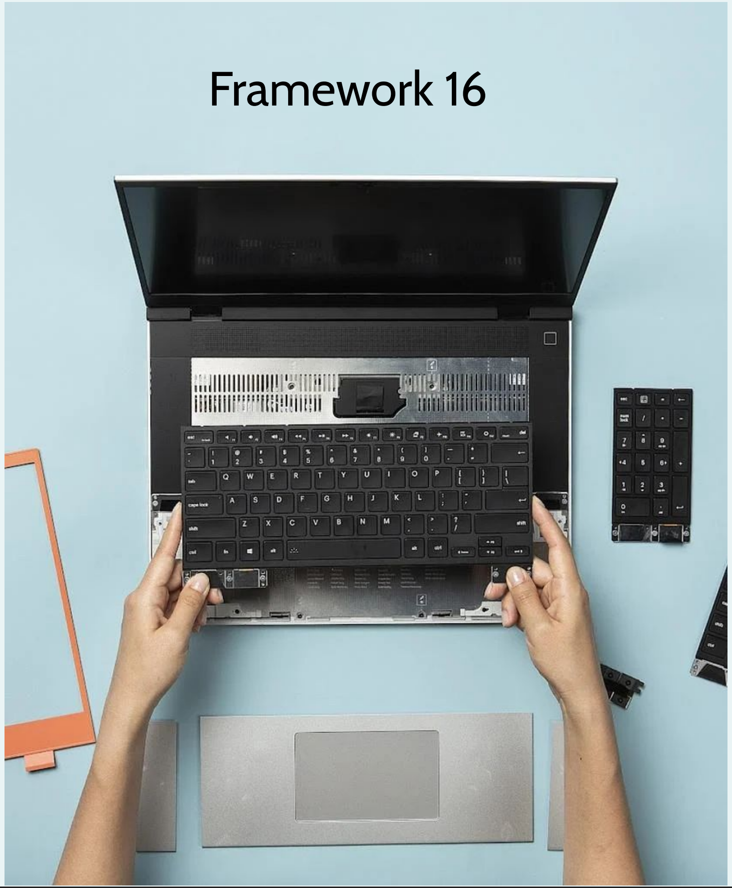
- New computer company, barely 4 years old
- Team of about 40 people in USA, Taiwan, Canada
- Trying to change the industry
- More repairable, open, sustainable, user-owned, customizable

What is Framework

Framework 13



Framework 16



Repairable - What, Why?

- Users can diagnose and fix issues
- Avoid E-Waste
- Keep it for a long time and maintain it
- Great for organizations with IT
- Save money and the planet

How to be Repairable

- To diagnose and solve issues
  - Lots of guides
  - QR Codes on every part, links to the guide
  - Good Support
  - Community Forum for discussions
- **All** parts are user-serviceable
- All parts can be bought from us
  - Big things: Mainboard, Display, Keyboard, ...
  - Little things: Screws, Hinges, eDP cable, ...
- Full schematic available to repair shops

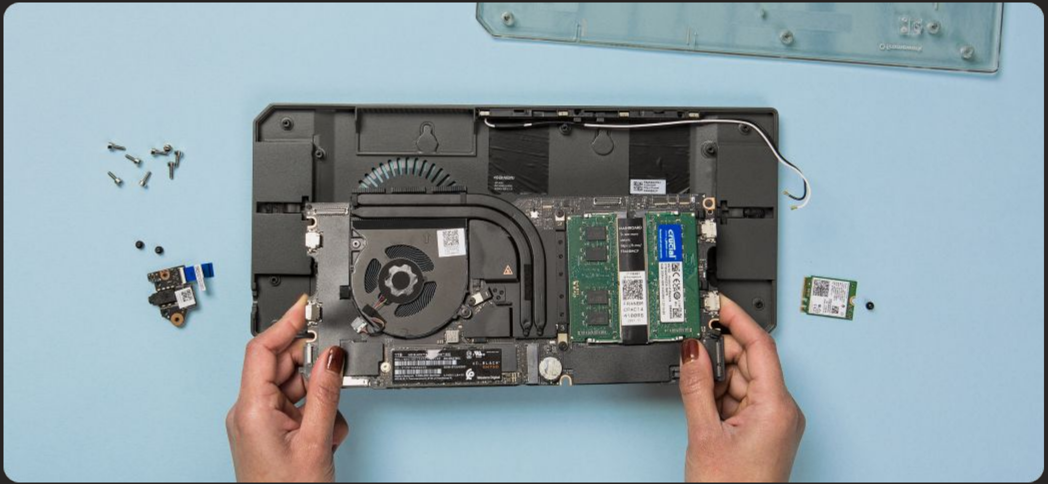
Repairable Design

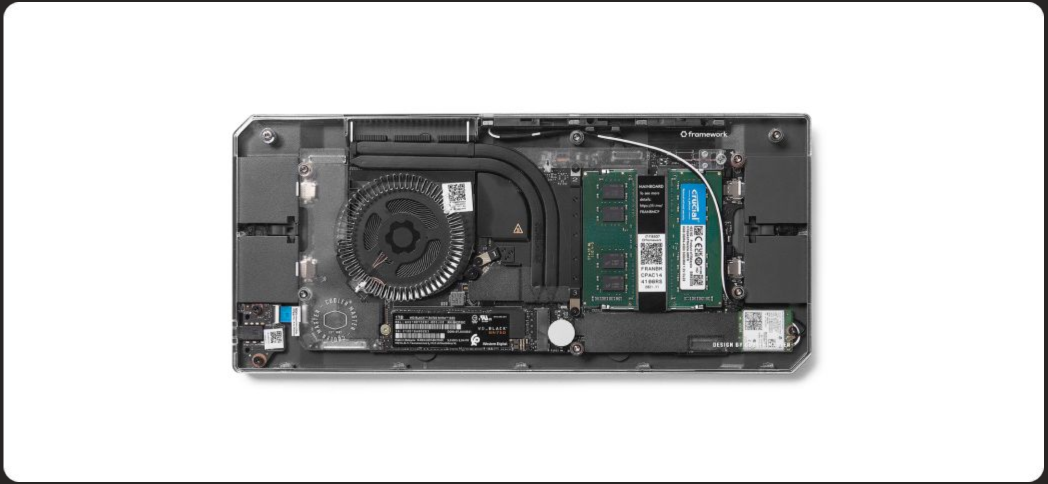
- No adhesive (except for antenna)
- Screws for everything
    - Screwdriver in the box, Just two types of screw heads
    - Extra screws in the case - always with you
    - Captive screws
    - Screwdriver included in the box
- Lots of magnets
- Sturdy connectors
- Break out into individual parts
    - Audio Board, Expansion Cards
- Socketed parts: WiFi/BL, RAM, SSD

Repairable Design - Reusability

- What if you upgrade a part? - Re-use it!
- 3D printed or CoolerMaster mainboard case
- Power button on the mainboard
- Community
    - Many, many mainboard enclosures (AIO, Tablet, Cyberdeck, …)
    - eDP to DP adapter
    - Keyboard to USB adapter

Mainboard Case





Repairable - Troubles

- Magnets can cause audible noise -> tweak design and firmware to avoid
- Must maintain forward and backwards compatibility
  - Same connectors
  - Fit mechanically
- Logistic challenge to sell replacement parts
- Great for enthusiasts but scares average consumer
- Firmware and EE changes to enable standalone mainboard
  - “Dead-battery” mode
  - PD firmware update when PD powers the system?

Forward and Backwards Compatibility

- Two base chassis: 13.5 inch and 16 inch
- 5 different mainboards with full compatibility
  - Only Chromebook is slightly different
- Laptop from 2021, supports all upgrades

Forward and Backwards Compatibility

Forward and Backwards Compatibility

- Two base chassis: 13.5 inch and 16 inch
- 6 different laptops with full compatibility
  - Only Chromebook is slightly different
- Bought Laptop in 2021, supports all upgrades
  - Matte screen
  - Stiffer hinges
  - Mainboard/CPU Upgrade
  - Battery with more capacity
  - Louder speakers
  - New bezel colors
  - Different keyboard language
  - New Expansion Cards: Ethernet, Audio, HD/DP Power Savings
  - Many other **potential** upgrades: display, camera, touchpad, RGB keyboard, KB nub, touchscreen

Open

- Every sense of the word - your hardware, your choice
- Run any OS you like
    - Official Support for Linux and W...
- Open documentation and source code
- Tinkering and custom modules encouraged

Open Source

- Mainboard and Module Interface Pinouts
- 3D-Print modules (Mainboard, Expansion Cards, Display, Hinges)
- Example electrical designs
- Firmware
  - Embedded Controller (based on Chromebook's)
  - Framework 16 - Input Modules (QMK, baremetal Rust)
  - Check out my talk at the *Open Source Firmware Conference 2023*

Open - Troubles

- Supporting Linux and Windows takes a lot of effort
- Partners and Vendors often only reluctantly support Linux
- Expansion Card power draw
  - Audio Card - Completely powered off until connector inserted
  - DisplayPort - Firmware Update
  - HDMI - New HW rev (or rework) and firmware update
  - Custom PD controller behavior
  - Others - PD firmware changes, improved Expansion Cards
- Modularity on Framework 16 - mechanical challenge
- Internal USB keyboard on Framework 16

Linux Compatibility

- Choose parts that are already compatible (WiFi, Fingerprint, ...)
- Work with silicon vendor to fix bugs
- Hardware workarounds: Linux I2C HID bug
- BIOS Option for Linux audio tuning
- Release firmware on LVFS
- Port devices to fwupd/LVFS
- Write our own software
- User Choice
  - Support Ubuntu,Fedora and others instead of a custom distro
- Dedicated Linux Support
  - Support, Forum, Social Media (Mastodon)
  - Installation Guides for many distros
  - Internal Linux QA

Why all that?

- We genuinely care
  - About users
  - About the environment
  - About technology
- It’s fun and makes us proud
- To incentivise the rest of the industry to follow our example
- If people can repair and upgrade, they keep buying from us

How can we do better?

- Make sure we keep compatibility in future models
- More open source firmware
- Better software to control the system
- Open up the *Marketplace*
- Should we pre-install Linux?
- Let us know!

Summary

- It’s challenging to build great hardware
- But it’s worth it

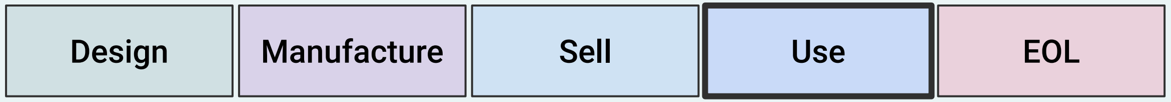
Thanks for listening!

Talk to me or check out https://frame.work

and https://github.com/FrameworkComputer

Repairable

Device makers
& Consumers



Three ways of getting to this

Regulation

Governments can enforce requirements around device longevity and interoperability. (e.g. repairability index reporting)

Demand

Consumers and businesses can demand that devices makers build longer-lasting products and vote with their wallets.

Competition

Related to demand, companies can enable healthier economics with a better business model.

Leveraging network effects

Resale markets

Repairable products are easier to re-sell, and your business can help foster the market.

Marketplaces

Upgradeable and customizable products enable a path for third party businesses to participate.

Software and services

A long-lived, loyal base of customers on the hardware makes it easier to build software for.

Three ways to align incentives around “use”

Counter-positioning

Capturing market share

Re-engagement

Increasing LTV

Network effects

Building a moat

framework