

Navigating Tech Decisions

A Startup Survival Guide



PHPUGFFM · 2026-05-21

Hi, I'm Toby! You might know me from ...



2004

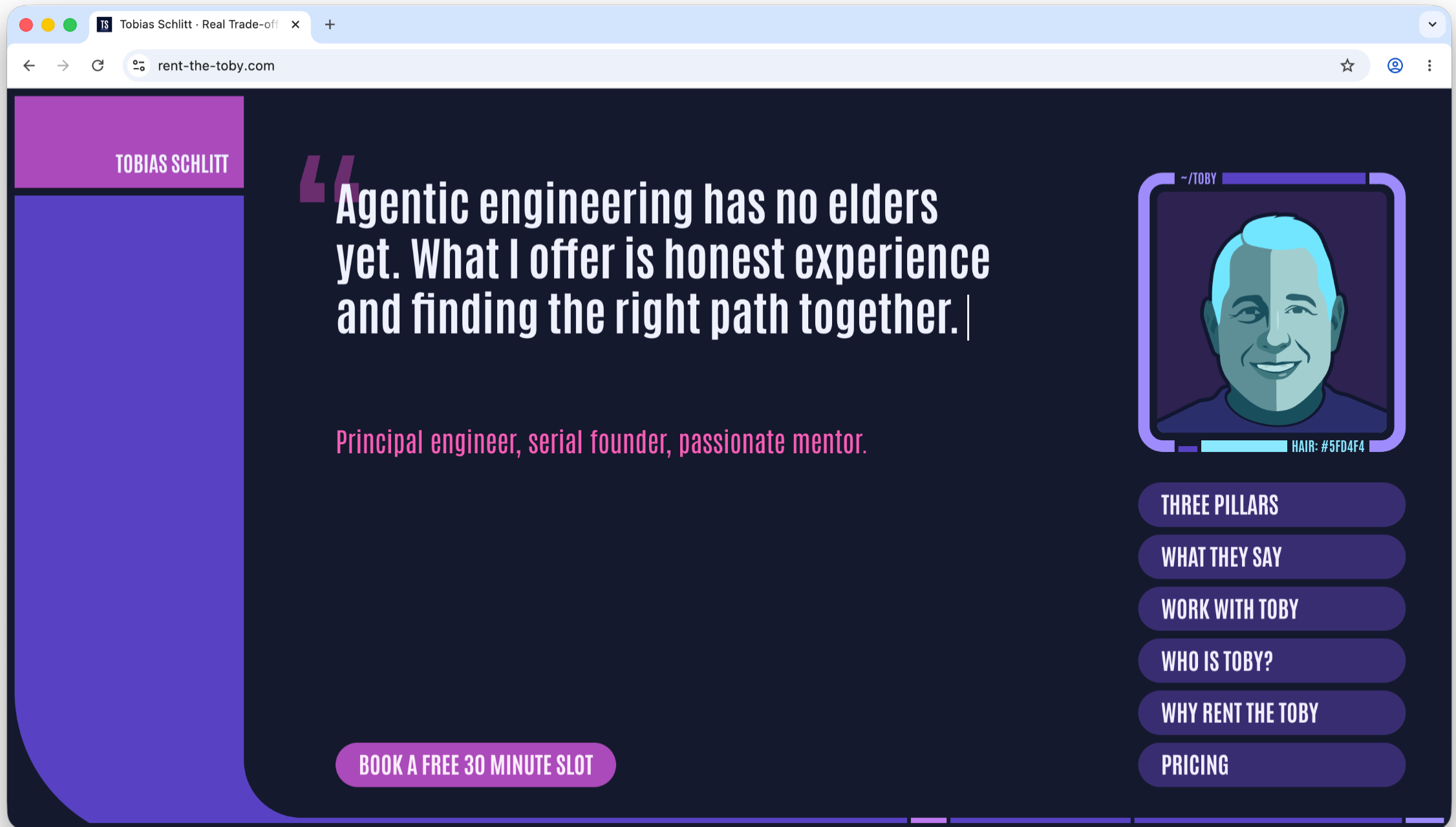


2014



2019

- Web since **1996**, PHP since **2000**
- Deutsche Bank, PEAR, eZ Systems, **Qafoo**, **Tideways**^{*}, **Frontastic**, commercetools, ...



This talk contains anecdotal knowledge. Do not take it as scientific evidence!

Agenda

- Background
- Startup life
- Tech stack
- Slicing
- Make or buy?
- Technical debt

Background

Background: Frontastic

- Founded **2017**
- Sold to **commercetools** in 2022
- Integrated into commercetools until 2025
- Continues as **commercetools Frontend**

My roles in Frontastic

- Co-Founder
- Implementer
- Co-CTO
- Principal Engineer
- Sales, Pre-Sales, Support, Marketing, Mental Health Advocate, Mentor, Mother, ...

Startup life

How a startup works ...



... how it actually works



Product market fit

Existing customers

Adapt product



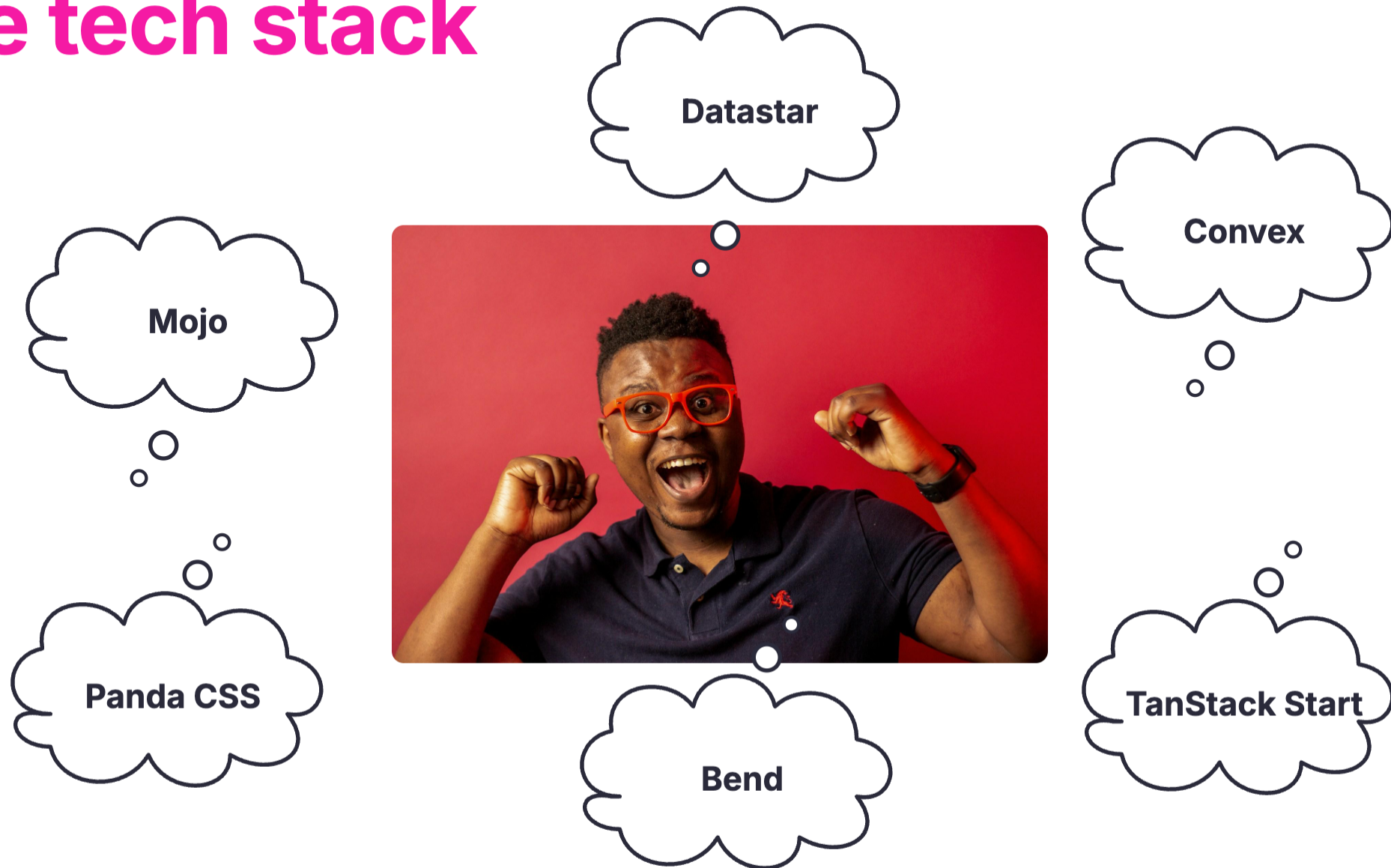
New customers

Investors

Tech stack



The tech stack

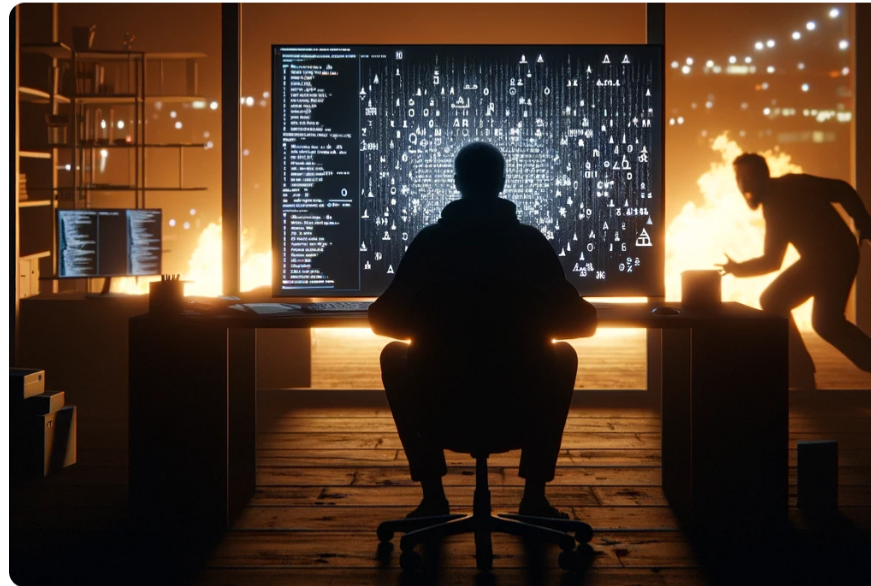


The tech stack

Datastar

Convex

Mojo



Panda CSS

Bend

TanStack Start

Tech maintenance startup reality

Imagine

- You did 2 investor pitches today
- You had an on-site workshop with a potential customer
- You hunted a Heisenbug in between
- You went for dinner with the team
- At **3am** you get called because a **customer is down**

→ this is when you need to maintain your tech stack.

**The fancy technology that you use
does not count. It's the fancy
technology that you create that
matters!**

Embrace boredom

- The most boring technology is where you **excel**
- Boring technology is where you can **juggle fast**
- The product you imagine is not the product you will have next week
- Consider: **"Innovation tokens"**¹

¹ mcfunley.com/choose-boring-technology

Slicing



Micro-services

Authentication

Product catalog

User profile

Authorization

Recommendation

Teapot



But wait ...

What are the actual reasons to have micro-services?

- Small, focussed teams
- Scale services independently
- Clear communication boundaries
- Technology independence

Which of these apply to the startup situation?

Drawbacks of micro-services

- Increased complexity in
 - Inter-service communication
 - Orchestration & operation
 - Testing
- Decreased flexibility
- Debugging overhead

What does that mean?

~~Monolith!~~ Macro-Services

Define services for

- Unique scaling requirements
- A dedicated team
- Inevitable unique tech stack

Make it **easy** to extract services later.

Make vs. buy

Make or buy?



Make or buy?

Make

- Your **core domain**
- What **distinguishes** you

Focus!

Buy

- Side-services
- What perfectly suits you
- What **you** can afford

Prepare to replace!

Technical debt

Technical debt

What do you associate with technical debt?



Technical debt

- Any investment in your startup is **debt**
- So your **technical** debt is an investment
- Don't fear the technical debt
- Be aware of it, make it conscious
- Make yourself aware of paying the debt

**Technical debt is a bet on the future
where you will have more information.**

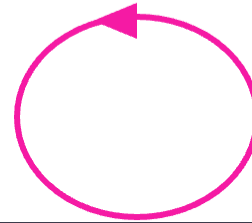
Conclusion

~~Engineering in tech startup is pure horror.~~

Engineering in tech startup is a lot of fun!

- **Excel** in your comfort zone
- Focus on proper trade-offs
- Optimize your software for **adaptability**
- Business value in the focus

... how it actually works (revisited)



Summary

- Product-market-fit is the **interesting** phase
- Embrace **boredom**
- Prepare to change your software/product
 - Frequently
 - Fast
- Accept technical debt, make it **conscious**

Thank you!

Questions?



tobias@schlitt.info
rent-the-toby.com
[linkedin.com/in/tobiasschlitt](https://www.linkedin.com/in/tobiasschlitt)

