Picking your (tech) poison wisely

Boris Martinović

Hi, I'm Boris.

- Software Engineer @ Euroherc Insurance
- Lecturer (Frontend) @ Algebra
- Google Developers Group Zagreb

Today on the menu

How to choose new tech
Why and **when** should you consider something new
How (not) to pick your poison

How to choose new tech when starting from scratch?

When starting from scratch

- Figure out what you want to build
- Think of the problem you want to solve
- Find out about best technologies to do it

When starting from scratch

- **Just do it!**
- If your initial choice doesn’t work for you, something else will

Why would you even consider something new?

When you should consider a new technology?

- Legacy technology
- Changing of paradigm (desktop -> web)
- Current technology stack doesn’t work with new needs

When you should consider a new technology?

- Developers not satisfied with the workflow
- Just want to try out something new and see how it goes
- …

Do you have to change everything?

Always

Sometimes.

Uhm, it...depends.

It depends on...

- What we pay for at the moment
- What our developers are familiar with, and want to use
- How expensive it is to transition

It depends on *whether*...

- Our stack is modular
- We have our all needs fulfilled by the current offering
- We want it to be simpler now

How to plan the switch?

How to plan the switch?

1. Establish project requirements, size, and scope
2. Understand what needs to be changed
3. Reevaluate the business logic
4. Define the current budget
5. Think about maintenance

How to plan the switch?

- There is no “one size fits all” solution
- Learning from (preferably others’) mistakes is what’s best
- There is always room for improvement
- Look for simplest solution that fits your bill

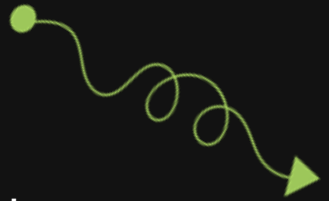
How (not) to pick your poison?

Don’t just seek complete solutions

- Viable, but not always
- Gotcha-s when you scratch underneath the surface
- “Jack of all trades and master of none”

Don’t just seek complete solutions

- Viable, but not always
- Gotcha-s when you scratch underneath the surface
- “Jack of all trades and master of none”



Instead consider what you really need

- Look for what you need according to your future requirements
- Use stacks (PERN, T3...) as a jumping off point
- Why not write a function instead of downloading a whole library

Don’t copy others

- Copying stack from Facebook, Twitter etc. might not work for everyone
- Stacks are created at different circumstances, needs, and times
- Work to **your** team’s strengths, not someone else’s

Don’t copy others

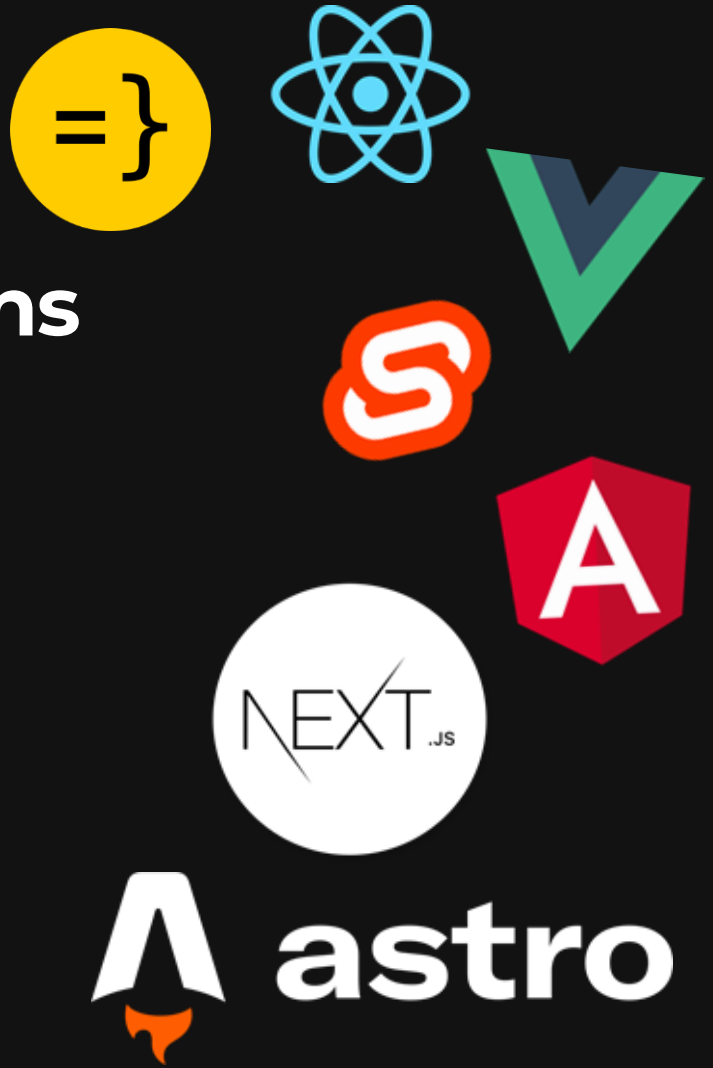
- Copying stack from Facebook, Twitter etc. might not work for everyone
- Stacks are created at different circumstances, needs, and times
- Work to **your** team’s strengths, not someone else’s

Instead take what you need from them

- Don’t blindly follow a company/author
- If something can give you a great idea - use it
- Taking what’s best from multiple sources is optimal solution

Don’t look for comparisons immediately

- Decision paralysis is real
- Too many choices can make you overlook the right one



Instead compare when you narrow it down

- Look for nuances and use cases
- Choose 2-3 that best fit your case - it’s easier to compare objectively

You can listen to others ...

That framework sucks!

I hated it so much I started from the scratch

Trust me bro

You can listen to others ...

~~That framework sucks!~~

~~I hated it so much I started from the scratch~~

~~Trust me bro~~

... but you should always have multiple sources of information and data

It's rare that something is that bad but widely used at the same time

Find perspectives from other people that use that technology

Why don't try for yourself?

Don’t go for the “next big thing”

- Technologies come and go - good ones stay
- Find ways to improve current technology
- All that glitters is not gold

Don’t go for the “next big thing”

- Technologies come and go - good ones stay
- Find ways to improve current technology
- All that glitters is not gold

Instead take what is production-ready

- XY amount of time won’t mean maturity - something can be good immediately
- Check Git repositories to see usages
- Talk to people - if it’s newer or smaller open source library you might get to authors directly

Don’t take your current way of doing things as gospel

- Processes in one platform might not translate well to other
- Use this as an opportunity to rethink current processes

Instead take your time and look for multiple solutions

- Rethink and improve your processes
- As app grows, so do its problems
- Try out different ways of solving problems that were solved in a way that’s not ideal

Don’t focus only on the benefits

- It’s very easy to find the benefits
- It’s easy to get hyped up prematurely
- Don’t get too comfortable - something that’s lacking will pop up eventually

Instead focus on finding pitfalls

- Look for weak spots that can hurt you in the future
- Figure out what your new framework is missing

Don’t be afraid to go back and try again

Use the opportunity as a learning experience

So...how to pick your poison?

- If you want **THE** universal answer, it’s the obvious one
- Don’t copy others but do take inspiration from them
- Consider many things - but don’t overthink

So...how to pick your poison?

- Make mistakes - **that's how we all learn**
- Take your time (if you can)
- Try to look at all sides of the coin and be objective

Thank you!