

Building Start-ups in Elixir

#CodeBEAMSTO

Stockholm, 17th of May 2019



Michael Schaefermeyer

mschae

Zweitag GmbH

Head of Business Development

- > Co-founder inverse.com
- > Interims-CTO for lab25 ventures
- > Elixir since 2014 (at B/R)
- > Father
- > Diver

1 Drawbacks of Elixir

Auth*

- › No authorisation library that fits my requirements
 - › Most built on JWT
 - › List, expire sessions
- › No go-to authorisation library
 - › Think pundit in Ruby
- › 3rd party services are an option
 - › Giving up a crucial part of your app
 - › You'll have to make compromises



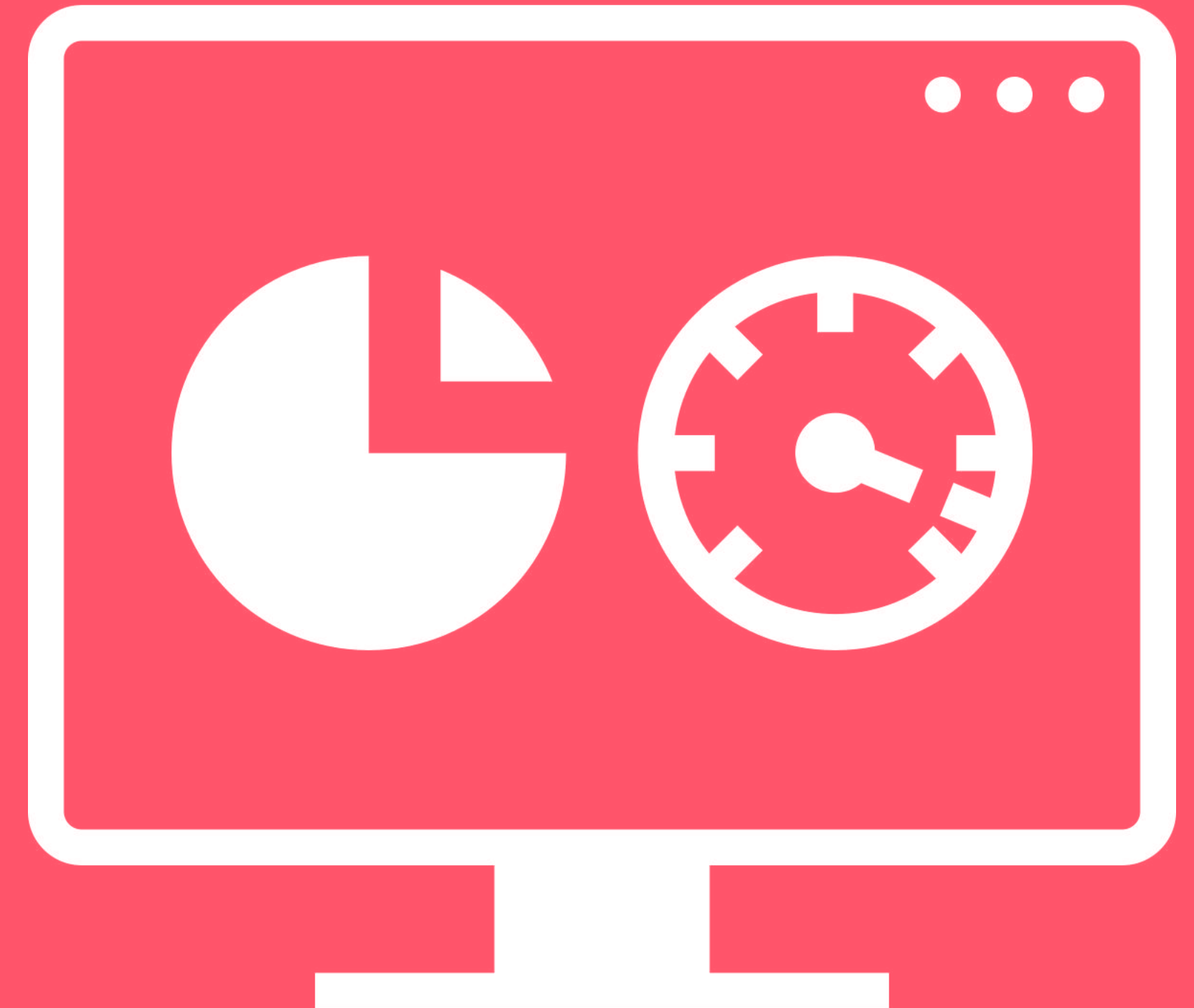
Admin Backends

- › Crucial for quick development
 - › Needs minimal configuration
 - › Needs searching, sorting, filtering, ...
- › Ruby: ActiveAdmin
- › Python: Django (Admin built in)

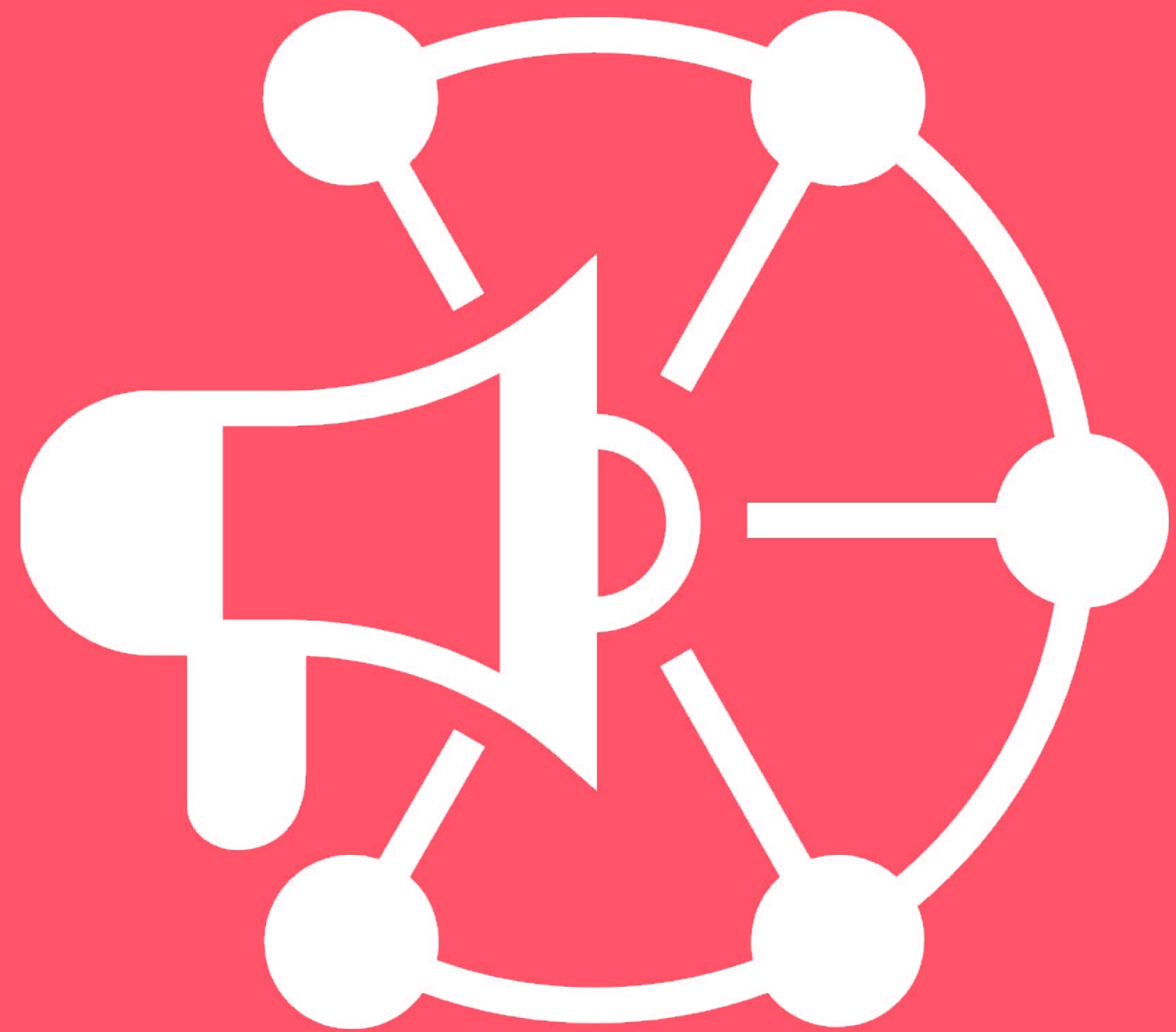


Monitoring

- › Used to be very hands-on
 - › Exometer, StatsD, ...
 - › Elixometer never really took off?
- › Positive development: Telemetry
 - › Not yet included everywhere but great start
 - › Still need to report metrics somewhere
 - › Needs more tutorials



Adoption



- > Not as widely used as other popular start-up languages (ruby, go, python, ...)
 - > (Official) Libraries might not exist
 - > No community-wide libraries for some topics (see above)
- > Experienced engineers are hard to find
 - > A start-up might have to teach while building

2

Benefits of Elixir

Functional Programming

- › Function features → clean code → less technical debt down the road
 - › Pattern matching
 - › No in-block breaks
 - › Strict division of data and logic
- › Explicitness makes it easy to understand what's going on
- › No side-effects



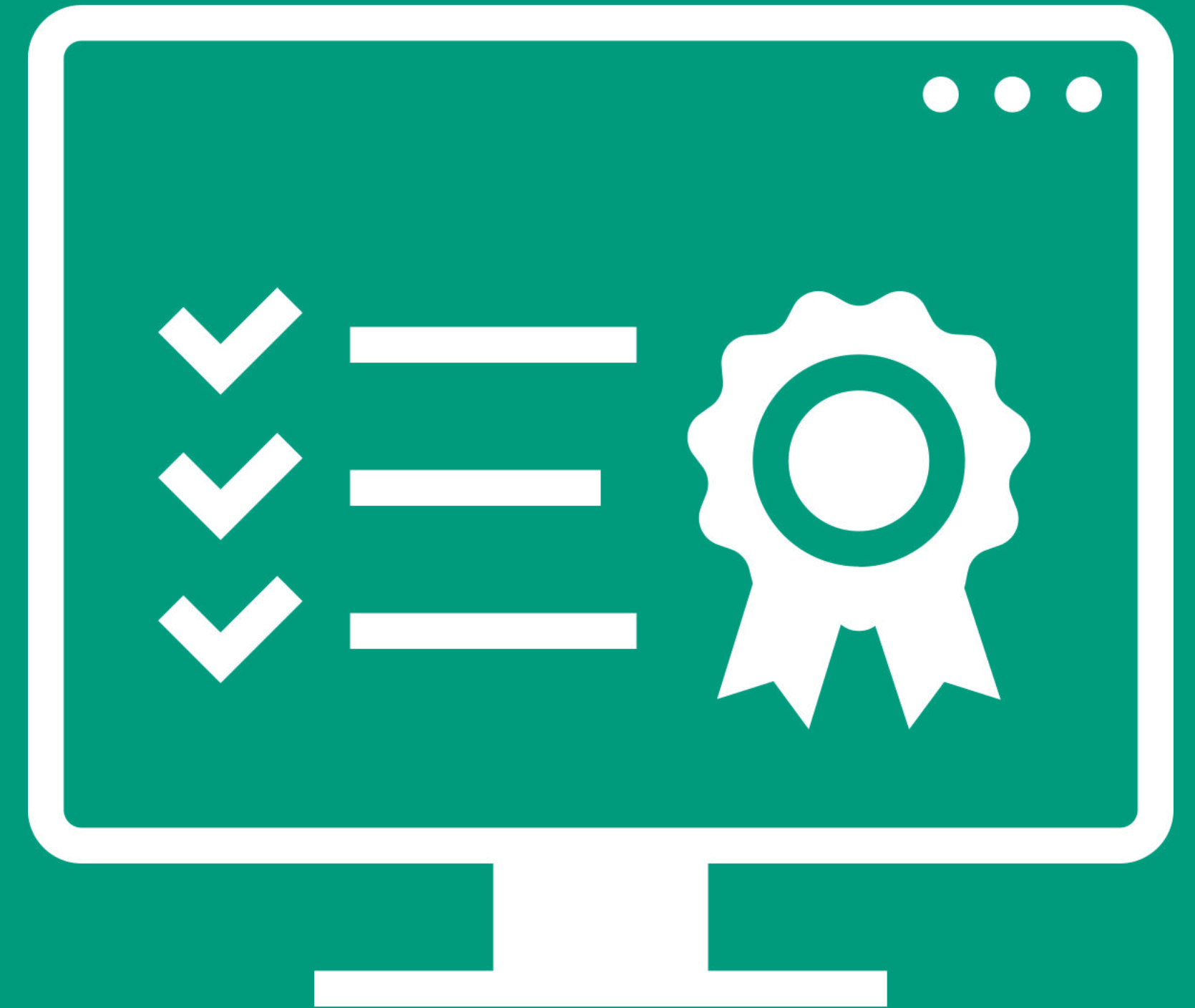
Documentation

- > Document is intimate part of the language
- > Overall one of the best documented languages I've ever seen
 - > 3rd party libraries are very well documented
 - > Crucial parts of the documentation are educational
- > Documentation is beautiful



Testing

- › Testing is an integral part of the language
- › Functional programming makes it easy to test
 - › Limited set-up
 - › Clear scopes
- › Allows fast-moving teams and constant refactoring without testing being too much “in the way”



Community



- › Inclusive, approachable community
 - › Makes it easy to contribute
 - › Allows junior engineers to go the next step
- › Multiple channels to get help
 - › Slack, ICQ, elixirforum.com
 - › Easy access to the right people
- › Friendliness and respect is key

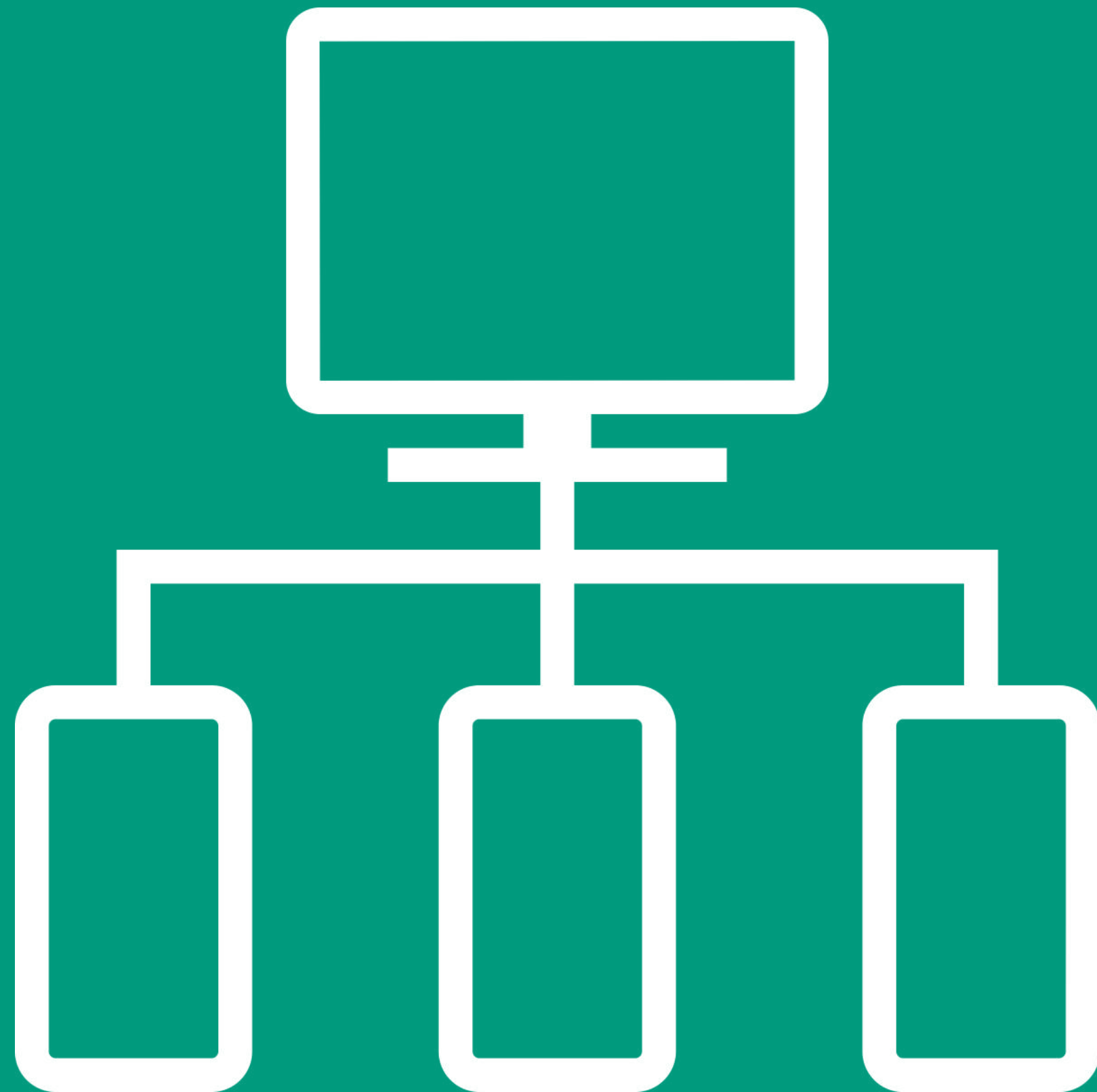
Recruiting

- › Elixir is great to teach, has a very manageable learning curve
 - › Thanks to its explicitness, Phoenix isn't magic
 - › Sufficient to understand parts of the system
 - › OTP not a must from the get-go
- › Senior developers like new challenges
 - › Elixir offers extremely high developer happiness
 - › Elixir just „makes sense“



Operations

- > Releases + Docker = ❤️
 - > Super small images
 - > Very clean containers
 - > Elixir v1.9 will be huge for operations
- > Extreme stable platform, can handle traffic
 - > “\$1,000 instead of \$3,000” for our production stack
 - > No CDN - no problem



3

Learnings/Recommendations

Make it simple

- › Use frameworks
- › Consider server-side rendering (+ Phoenix LiveView!)
- › Build a monolith!



"MAKE IT WORK, THEN MAKE
IT BEAUTIFUL, THEN IF YOU
REALLY, REALLY HAVE TO,
MAKE IT FAST.

90% OF THE TIME,
IF YOU MAKE IT
BEAUTIFUL, IT WILL
ALREADY BE FAST.
SO REALLY, JUST MAKE IT
BEAUTIFUL."

- JOE ARMSTRONG

4

Summary

Elixir allows you to scale

- › Team
- › Code base
- › Infrastructure

Investors don't mind Elixir

- › Proven, known foundation
- › Used by known start-ups

Use what works for you and your team

- › You'll have to be lean
- › Elixir can be a great choice

Thank you

@mschae

michael.schaefermeyer@zweitag.de

www.zweitag.de