

From iOS- to Vapor developer



New programming language



June 2014

Swift announced
at WWDC 

Dec 3, 2015

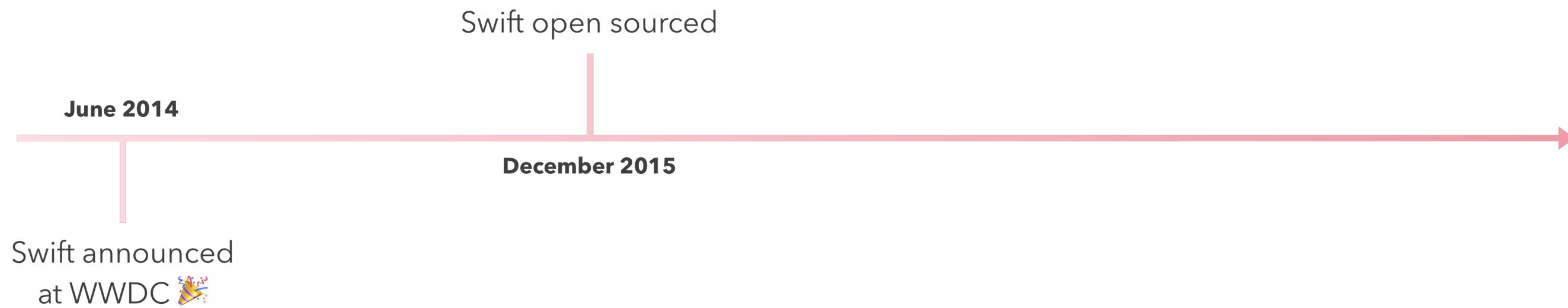
Swift is Open Source

Swift is now open source. Today Apple launched the open source Swift community, as well as amazing new tools and resources including:

- Swift.org – a site dedicated to the open source Swift community
- Public source code repositories at github.com/apple
- A new Swift package manager project for easily sharing and building code
- A Swift-native core libraries project with higher-level functionality above the standard library
- Platform support for all Apple platforms as well as Linux

Now anyone can download the code and in-development builds to see what the team is up to. More advanced developers interested in contributing to the project can file bugs, participate in the community, and contribute their own fixes and enhancements to make Swift even better. For production App Store development you should always use the stable releases of Swift included in Xcode, and this remains a requirement for app submission.

Swift ❤️ Linux



Kitura

Perfect

Vapor

Zewo

Pros

- **Swift**

Pros

- Swift
- Fast

Pros

- Swift
- Fast
- **Code share**

Pros

- Swift
- Fast
- Code share
- **More fullstack developers**

Pros

- Swift
- Fast
- Code share
- More fullstack developers
- **Xcode**

Pros

- Swift
- Fast
- Code share
- More fullstack developers
- Xcode
- **Big community**

Cons

- **New language**

Cons

- New language
- **New package manager**

Cons

- New language
- New package manager
- **New framework**

Cons

- New language
- New package manager
- New framework
- **Xcode**

Cons

- New language
- New package manager
- New framework
- Xcode
- **Hosting**

Cons

- New language
- New package manager
- New framework
- Xcode
- Hosting
- **Lack of libraries**

Programming languages in the making

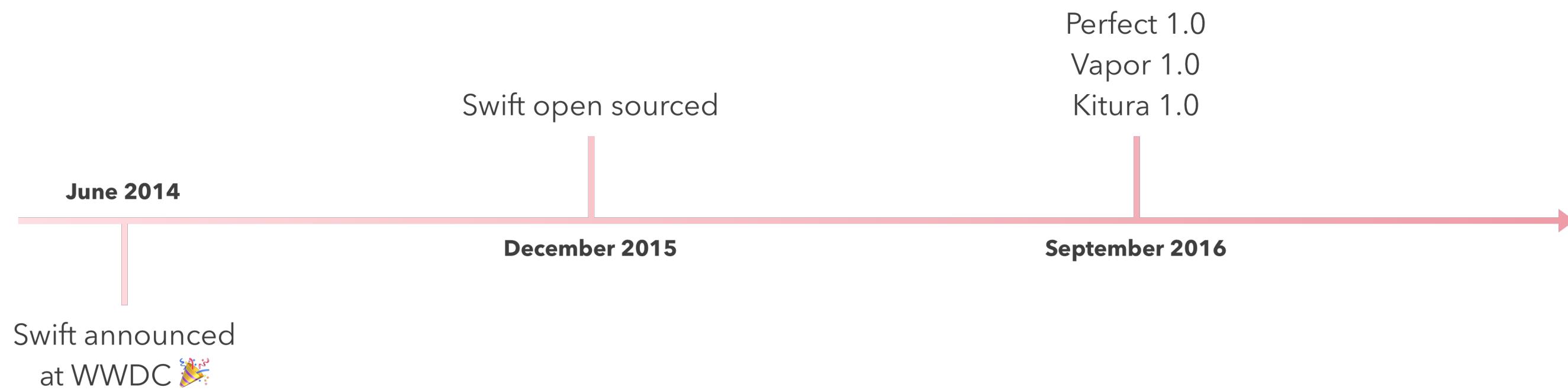
- JavaScript (1995): 22 years
- Java (1996): 21 years
- Ruby (1996): 21 years
- **Swift (2014): 3 years**

Web frameworks in the making¹

- Spring MVC (2002): 15 years
- Ruby on Rails (2004): 13 years
- NodeJS (2009): 8 years
- **Kitura/Perfect/Vapor (2016): 1 year**

¹ Since their version 1.0 release.

The future is bright for server-side
Swift



Kitura

Perfect

Vapor

Kitura

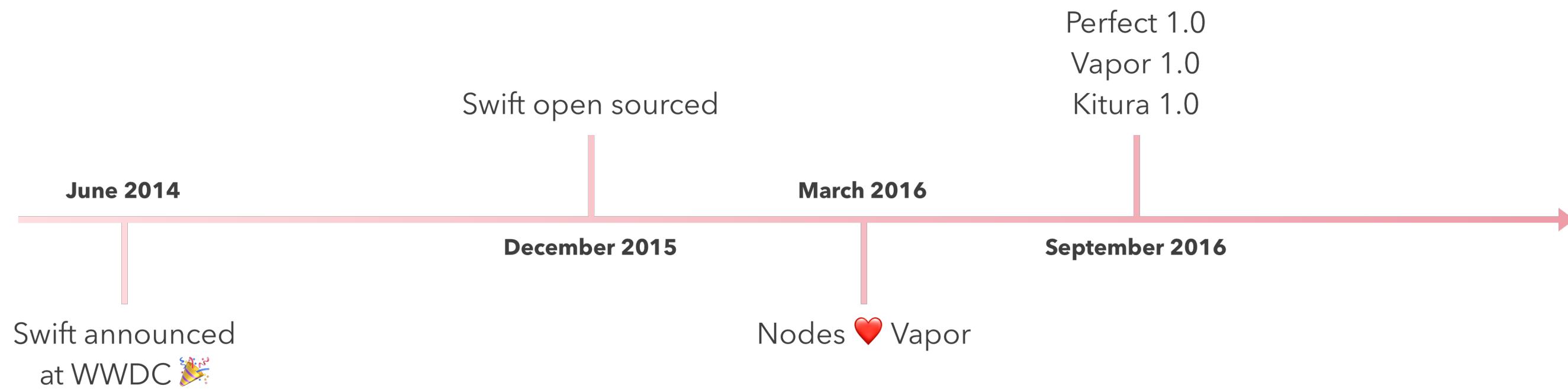
Perfect

Vapor

Kitura

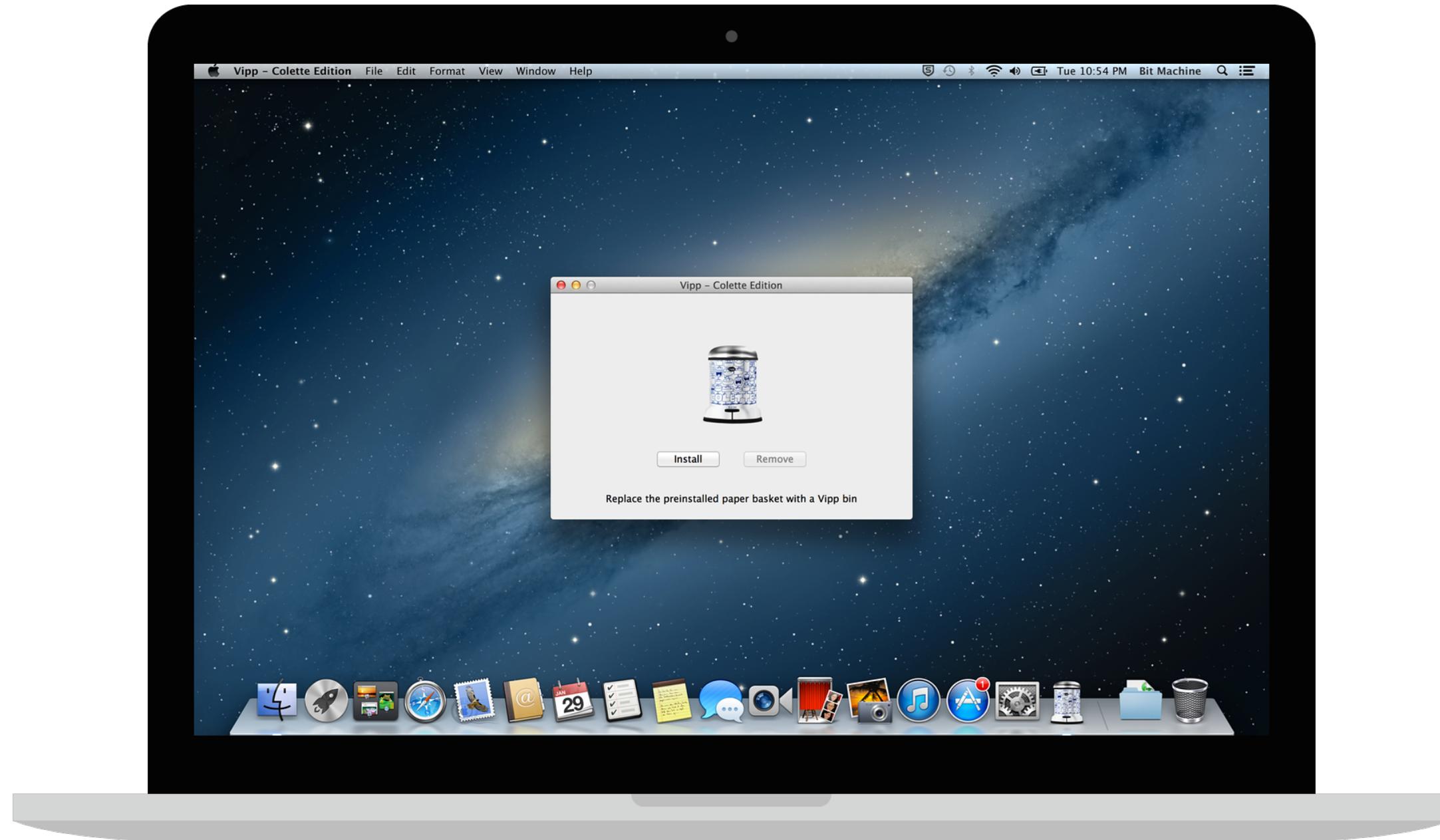
Perfect

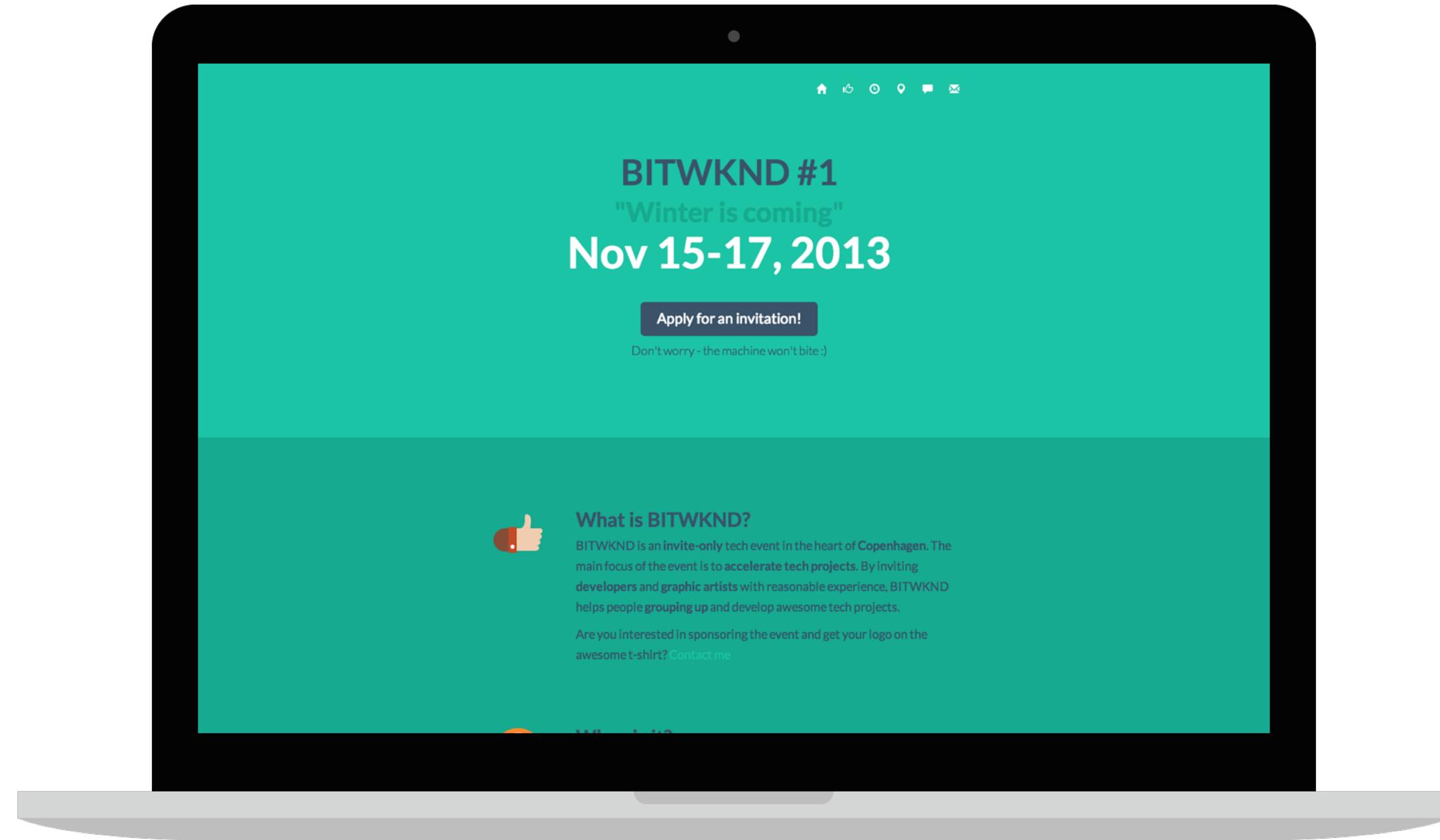
Vapor



Hi, my name is
Steffen D. Sommer











Steffen ❤️ iOS

Steffen ❤️ ~~iOS~~ Swift

Swift as a language, not iOS as a
platform



steffendsommer commented on 5 Mar 2016

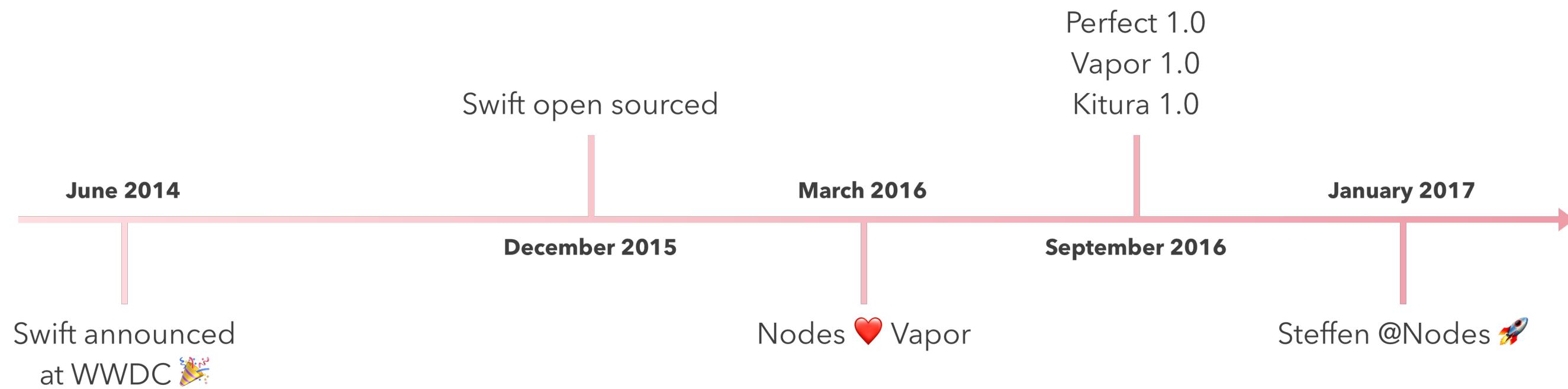
Member



First of all, I just want to say that I think Vapor is a great library that I have enjoyed using.

One of the benefits, in my opinion, with the rising number of web frameworks written in Swift is that we as iOS developers are able to reuse code between the frontend and the backend. An obvious candidate for sharing is model objects.

I wanted to ask if anyone have any guidance or practical tips on how to manage a repository consisting both an iOS app and a backend using Vapor, with some classes being shared across the two. Tips on e.g. file structure, Xcode project setup and how to manage the deployment without uploading unneeded iOS specific files to the server is highly appreciated.



iOS != Vapor

iOS != Vapor

Number of users

iOS

- 1

Vapor

- Lots of users

iOS != Vapor

Performance

iOS

- Not an issue

Vapor

- **Users do not want to wait for requests to return**

Vapor

- Users do not want to wait for requests to return
- **More performance = more money**

iOS != Vapor

Crash handling

iOS

- "**We'll fix it in the next app update**"

Vapor

- Fix it now

iOS != Vapor

Updates

iOS

- 2 week release cycles

Vapor

- Whenever we want

iOS != Vapor

Platforms

iOS

- One platform to rule them all

Vapor

- macOS + Linux

iOS != Vapor

Foundation

iOS



Vapor



iOS != Vapor

Breaking changes in the API

iOS

- Good to go

Vapor

- Handle different versions depending on the client

iOS != Vapor

The Xcode project file

iOS

- Merge madness

Vapor

- .. what file?

iOS != Vapor

Third party frameworks

iOS

- Which one to pick?

Vapor

- Roll out your own

iOS != Vapor

User Interfaces

iOS

- Storyboards and auto layout

Vapor

- **HTML, CSS and JavaScript**

iOS != Vapor

Persistence

iOS

- Used to some extent

Vapor

- Used heavily

iOS != Vapor

Security

iOS

- Mostly handled by the OS

Vapor

- Needs to be a first class citizen in your project

iOS != Vapor

Consumers

iOS

- **None**

Vapor

- Frontend

iOS != Vapor

Releasing

iOS

- iTunes Connect

Vapor

- Deploying

Tips to overcome these challenges

Tip #1

Start reporting

Tip #2

Setup continuous integration

Tip #3

Write tests

Tip #4

Write documentation

Tip #5

Learn about databases

Tip #6

Consider when to use Foundation

Tip #7

Use Vapor Cloud

Tip #8

Use security packages

Tip #9

Use a template language

Tip #10

Get involved

My experiences with Vapor in
production

My experiences with Vapor in production

- **Lots of moving targets (Swift, Vapor, Vapor Cloud)**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- **Overhead in creating and maintaining open source packages**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- Overhead in creating and maintaining open source packages
- **Lack of documentation**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- Overhead in creating and maintaining open source packages
- Lack of documentation
- **Outdated resources**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- Overhead in creating and maintaining open source packages
- Lack of documentation
- Outdated resources
- **Few experts**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- Overhead in creating and maintaining open source packages
- Lack of documentation
- Outdated resources
- Few experts
- **Lots of boilerplate**

My experiences with Vapor in production

- Lots of moving targets (Swift, Vapor, Vapor Cloud)
- Overhead in creating and maintaining open source packages
- Lack of documentation
- Outdated resources
- Few experts
- Lots of boilerplate
- **Linux quirkiness**

How things have improved

- **Updates are slowing down (Swift and Vapor)**

How things have improved

- Updates are slowing down (Swift and Vapor)
- **Vapor Cloud reached 1.0 and is stable**

How things have improved

- Updates are slowing down (Swift and Vapor)
- Vapor Cloud reached 1.0 and is stable
- **We don't create as many open source packages anymore**

How things have improved

- Updates are slowing down (Swift and Vapor)
- Vapor Cloud reached 1.0 and is stable
- We don't create as many open source packages anymore
- **Documentation is improving**

How things have improved

- Updates are slowing down (Swift and Vapor)
- Vapor Cloud reached 1.0 and is stable
- We don't create as many open source packages anymore
- Documentation is improving
- **More and more resources**

How things have improved

- Updates are slowing down (Swift and Vapor)
- Vapor Cloud reached 1.0 and is stable
- We don't create as many open source packages anymore
- Documentation is improving
- More and more resources
- **Bigger community - more experts**

How things have improved

- Updates are slowing down (Swift and Vapor)
- Vapor Cloud reached 1.0 and is stable
- We don't create as many open source packages anymore
- Documentation is improving
- More and more resources
- Bigger community - more experts
- **Less boilerplate**

How things have improved

...

- **Swift on Linux is more mature**

What it means to us

- **25+ customer projects running in production using Vapor and Vapor Cloud**

What it means to us

- 25+ customer projects running in production using Vapor and Vapor Cloud
- **30+ open source packages created**

What it means to us

- 25+ customer projects running in production using Vapor and Vapor Cloud
- 30+ open source packages created
- **Hosting fees are going down**

What it means to us

- 25+ customer projects running in production using Vapor and Vapor Cloud
- 30+ open source packages created
- Hosting fees are going down
- **More and more of our iOS developers starts to catch up**

What it means to us

- 25+ customer projects running in production using Vapor and Vapor Cloud
- 30+ open source packages created
- Hosting fees are going down
- More and more of our iOS developers starts to catch up
- **More rapid development**

No major issues since we made the
switch