

Full-Stack Haskell: from Prototype to Production

Ryan Trinkle, Obsidian Systems
YOW! Lambda Jam, Sydney 2018

Obsidian Systems—March 2015

- Ryan Trinkle
- Ali Abrar
- Two laptops
- GHCJS
- Reflex

Obsidian Systems

Hypothesis: We can build apps **better, faster,**
and **cheaper** using **full-stack** functional
programming

Obsidian Systems — 3 Years Later



The Good

Full-Stack Development

- Less to communicate or negotiate
 - But still collaborative
- Clearer understanding
- Ship each feature when it's ready
- More rewarding
- Shared types
- Reusable pure code

Don't Work Around—Fix

- Cheaper
- More robust
- Meet lots of developers
- Contribute to the community

No Prototypes

- More honest
- More predictable
- Less stressful
- Better quality product
- Quicker delivery to production

The Bad

Perfectionism

- Not all strong typing is free
- Don't speculate - reduce assumptions
- Start with thin, end-to-end solution
- Refactoring is easy

Shiny New Things Syndrome

- It's more impressive to take out a tank with a BB gun than with a nuke
- Cognitive overhead is real, especially on teams
- Most great software is built without even Haskell 98's features

The Absent

Managing GHCJS

- Problem: complex and lengthy build
- Solution: Nix
 - Purity means everyone can build it - no "works on my machine"
 - Binary cache avoids wasting time building
 - Fine-grained expressions allow incremental, pure rebuilds - not like docker
- Upstreamed to Nixpkgs

Mobile Apps

- Problem: different codebase/team per platform
- Solution: cross-compile Haskell
 - Native code, visuals rendered with DOM
 - No need to re-design user interface
 - Full access to native APIs
 - Full native performance
- Upstreamed to GHC, Nixpkgs, Reflex Platform

Obelisk

- Problem: too much to learn upfront
- Solution: sensible defaults for the whole stack
 - Project setup
 - Development
 - Deployment
- Open Source:

<https://github.com/obsidiansystems/obelisk>

Obelisk

- ob init
- ob run
- ob deploy
- Mobile

Coming Up

- GHCJS 8.2/8.4
- More docs
 - Common tasks
 - How things work
- You decide!

Thanks

- Haskell Community
 - Luite Stegeman
 - Hamish Mackenzie
 - Will Fancher
 - Doug Beardsley
 - David Laing & QFPL
 - Ben Gamari & Well-Typed
 - and many others

Thanks

- Obsidian Team
 - Luigy Leon
 - Ishaq Sloan
 - Dan Haraj
 - Elliot Cameron
 - Alexandre Esteves
 - Sridhar Ratnakumar
 - Tom Smalley

Full-Stack Haskell: from Prototype to Production