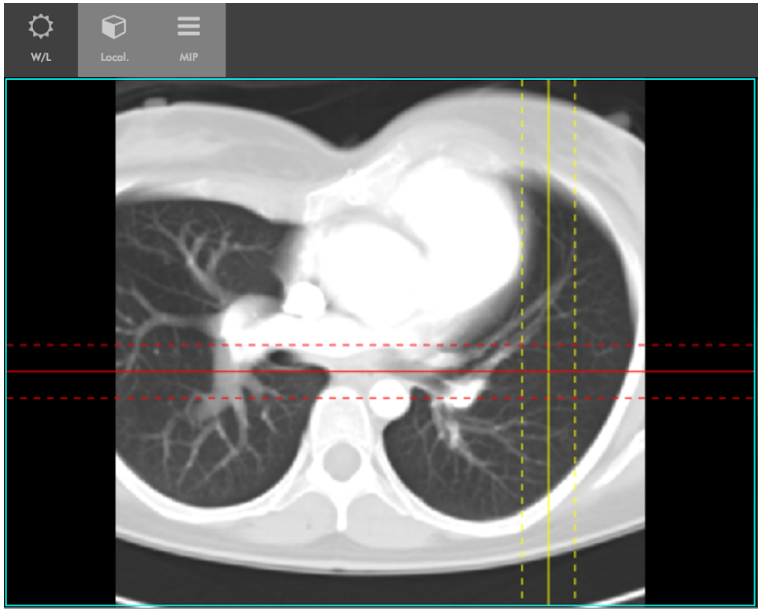
Haskell at DICOM Grid

An Experience Report

What we make

Software for radiologists and patients to

- view
- share
- manage



their medical data

Introducing Haskell

- Already using Java, Scala, Groovy, C#, JavaScript, TypeScript, Perl, Python…
- Try to use “the best tool for the job”
- Very autonomous developer workflow
- A need for correctness
- Microservices architecture

… Easy to test and demonstrate Haskell

Introducing Haskell

- Tools
  - typescript-docs
  - purescript
- First big project: xds-server
  - Interop with external SOAP services
  - Running in production for 9 months
  - Some great ways to apply (and showcase) Haskell

Current Projects

- metadata

  Manages patient metadata, (en/de)cryption, long term storage
- permissions

  Handles auth requests from Nginx based on routes
- xds-service

  Integrates with an external 3rd-party SOAP service
- messaging

  A websocket server for pushing service events to the browser

Haskell in the Real World

Need to

- Read and write databases/files/sockets
- Keep logs
- Track performance
- Think concurrently

Taught Haskell different from real world code in many ways

Haskell in the Real World

But the same techniques apply:

- Follow the types!
- Separate pure code from impure code as much as possible

Haskell in the Real World

Several excellent libraries for solving real-world problems:

- ekg
- hslogger
- scotty
- hedis
- wreq
- amqp
- servant
- groundhog
- stm
- …

How we use Haskell

- Try to separate out pure code:
  - Lots of small DSLs
  - Free monads
- Simple type tricks go a long way:
  - Newtypes
  - Sum types
- Techniques to break up concurrent code:
  - Transactional channels
- “Haskell” on the client
  - PureScript w/ React (see my talk!)

Testing

- QuickCheck
  Property-based tests
- HSpec
  Specification-based component testing
- test-framework
  To arrange test results

Packaging

- Cabal sandboxes
  - cabal freeze
- Hudson-based CI build
  - Ship statically-linked binaries (simple!)
- Haven’t hit problems of scale yet
  - Nix?
  - Local Hackage?

Haskell's Strengths

- Parsing
  - Binary and text formats
- DSLs
  - Web routing
  - Data templating (XML/JSON/text/other)
- Testing
- Concurrency
- Lots of open source libraries

Hiring Haskellers

- We hire slowly (deliberately)
- Haskell acts as a good filter even though we don’t hire directly for Haskell positions.
- Also a good way to retain existing employees!
- Three employees learning Haskell currently.

Talk to me if you want to write real-world Haskell

(and Scala etc.)

Issues

- Debugging can be tricky at times.
  But a REPL enables new styles of debugging.
- Important to keep detailed logs and profiling data.
- Code reviews can take longer
  But code is generally self-explanatory.

Open Source

- Current libraries on Hackage:
  - tinytemplate
  - dicom
  - typescript-docs
- Also:
  - text-xml-qq

Questions?