# Clojure in Real Life

# Clojure in Real Life

- Rick Hall
- Clojure Developer at RentPath
- Atlanta Clojure Meetup
- rickhall2000@hotmail.com
- @onbeyondlambda
- onbeyondlambda.blogspot.com

# What’s in this talk

- What is Clojure?
- Why do developers check it out?
- Why do they stay?
- What do they have to put up with?

# What is Clojure?

- Functional Programming
- Modern Lisp
- Hosted on the JVM (or JavaScript, or CLR)
- Emphasizes Immutability

# What Draws People

- Macros
- Concurrency Primitives
- Philosophy

# Macros = super powers?

- “But with Lisp our development cycle was so fast that we could sometimes duplicate a new feature within a day or two of a competitor announcing it in a press release.” - Paul Graham “Beating the Averages”

# Macros - what are they

- Code is written as data structures
- Macros are functions that manipulate data structures at compile time
- Manipulate the structure of the program in the same way you manipulate any other data

# Macros - in Practice

- Higher Order Functions provide so much power
- Clojure core already has a lot of the macros you would write

# Concurrency Primitives

- Identity vs State - “Are We There Yet?”
- Refs - Software Transactional Memory
- Atoms - Compare and Swap
- Agents - Asynchronous Updates

# Concurrency in Practice

- Discouraging mutation is the win for concurrency, not CAS or STM
- core.async - Communicating Sequential Processes

# Philosophy

- “Simple Made Easy”
- Simple
- Pragmatic

# Why Developers Get Hooked

- REPL Driven Development
- Simple, terse syntax
- Collections and Sequence Abstractions
- Polymorphism
- Interop
- ClojureScript
- Macros mean amazing libraries

# REPL Driven Development

# REPL Driven Development

- Read Eval Print Loop
- Interact with a running system
- Experiment with immediate feedback to find what works

# Simple, Terse Syntax

- Rule 1: Code as written as lists with the operation followed by the arguments.
- Tempting to say there is no rule 2, but features do require syntax

# Collections and Sequence Abstractions

- Lists - Singly Linked Lists - Code is written as lists. Not used often for data
- Vectors - Sequential Data. Lookup by index
- Maps - Key/Value pairs
- Sets - Enforce uniqueness

# Polymorphism

- Multimethods - dispatch based on the value of any arbitrary function
- Protocols - dispatch based on the type of the first argument

# Interop

- Use Java libraries from Clojure
- Many Clojure libraries to wrap Java libraries and give them a Clojure feel
- Inherit and extend Java classes
- Implement Java Interfaces

# ClojureScript

- Same language in the browser and on the server
- core.async enables asynchronous code without callbacks
- ClojureScript libraries that wrap React.js have quickly become a popular way to build user interfaces

# Macros give library developers super powers

- core.async - Adding async functionality usually means a compiler change or a whole new language. In Clojure it is a library

- core.logic - logic programming language implemented in Clojure

- DSLs for SQL, web routing, HTML generation and parsing etc.

# core.async

# Transducers

# Pain points and how to address them

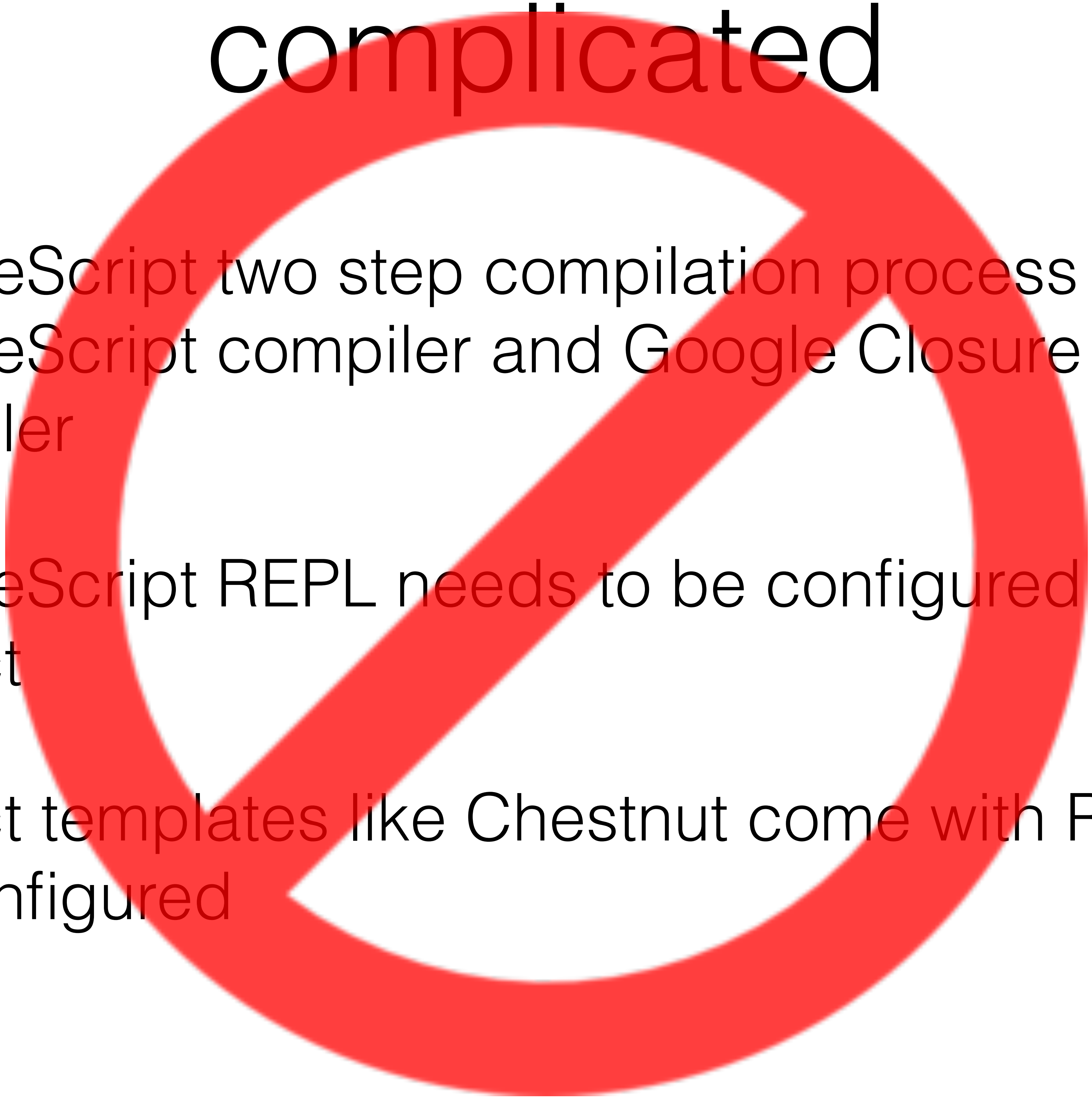
# Dynamic Typing

- Good or bad?
- core.typed provides compile time type checking
- Prismatic schema allows run time checking and declarative coercion

# Difficult Setup

- Install Java
- Install Leiningen
- Install Editor (LightTable, Emacs, IntelliJ)

# ClojureScript setup complicated

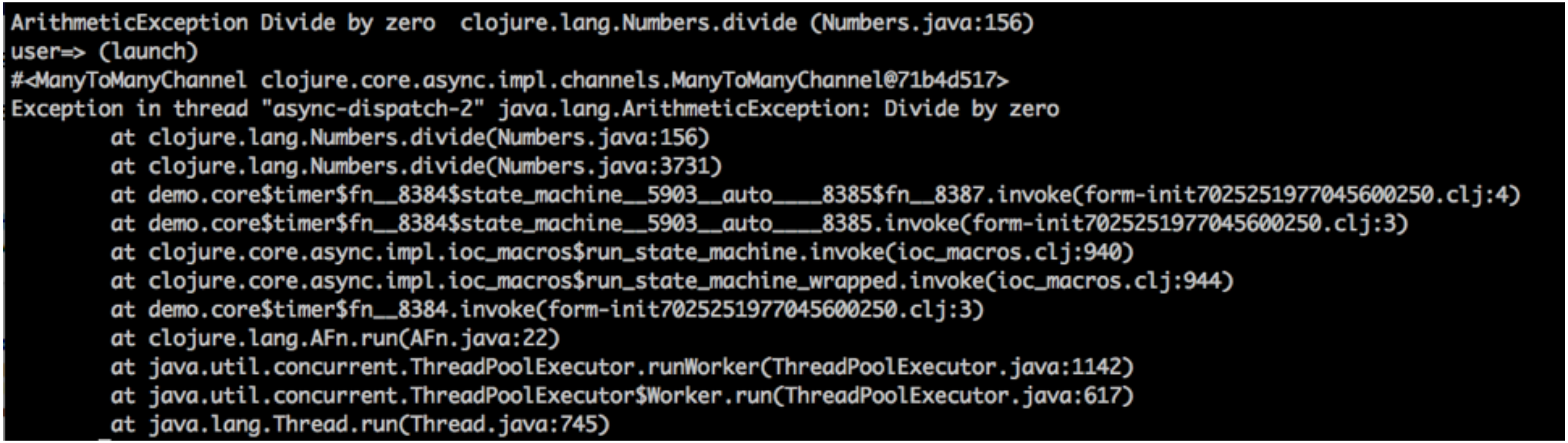


- ClojureScript two step compilation process - ClojureScript compiler and Google Closure compiler

- ClojureScript REPL needs to be configured in each project

- Project templates like Chestnut come with REPL preconfigured

# Stack Traces



```
ArithmeticException Divide by zero  clojure.lang.Numbers.divide (Numbers.java:156)
user=> (launch)
#<ManyToManyChannel clojure.core.async.impl.channels.ManyToManyChannel@71b4d517>
Exception in thread "async-dispatch-2" java.lang.ArithmeticException: Divide by zero
        at clojure.lang.Numbers.divide(Numbers.java:156)
        at clojure.lang.Numbers.divide(Numbers.java:3731)
        at demo.core$timer$fn__8384$state_machine__5903__auto____8385$fn__8387.invoke(form-init7025251977045600250.clj:4)
        at demo.core$timer$fn__8384$state_machine__5903__auto____8385.invoke(form-init7025251977045600250.clj:3)
        at clojure.core.async.impl.ioc_macros$run_state_machine.invoke(ioc_macros.clj:940)
        at clojure.core.async.impl.ioc_macros$run_state_machine_wrapped.invoke(ioc_macros.clj:944)
        at demo.core$timer$fn__8384.invoke(form-init7025251977045600250.clj:3)
        at clojure.lang.AFn.run(AFn.java:22)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
```

# Where to learn more

- Mailing List
- FreeNode
- Meetups

# Online Books

- Clojure For the Brave and True
- Clojure From the Ground Up

# Recommended Books

- Living Clojure - Carin Meier - O’Reilly Early Release - Tutorial and 7 week training plan

- Clojure Programming - Emerick, Carper, Grand - Lots of material clearly explained

# Clojure Mindset

- The Joy of Clojure - Fogus and Houser
- Any talk by Rich Hickey
- ClojureTV channel on YouTube

# Questions?

- rickhall2000@hotmail.com
- https://github.com/rickhall2000/dev-nexus-talk