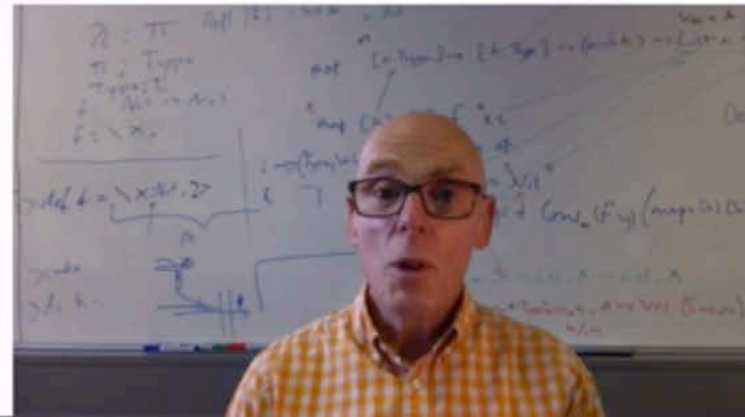

University of
Kent

University of Kent

And for the future, ...



University of Kent

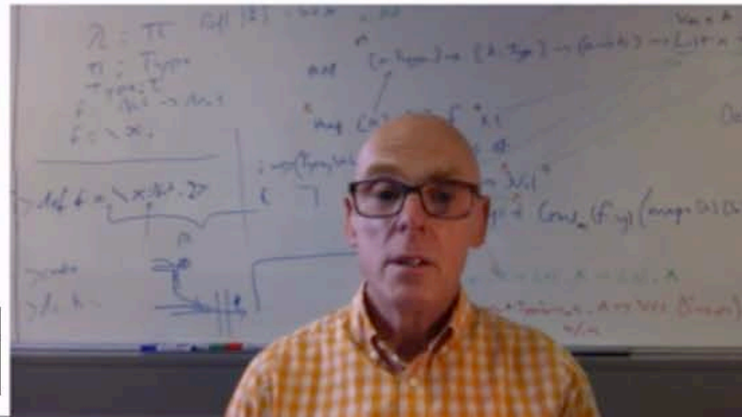
Key message #1: functional at the core

Computation by evaluating expressions … not side-effects.

Pattern matching + recursion.

Immutable data structures.

Higher-order functions: functions as data.



SIMON THOMPSON: What next?

University of Kent

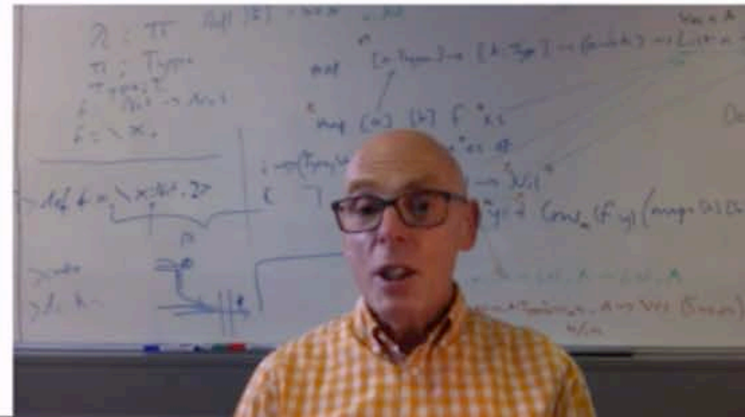
Key message #2: share-nothing processes

Concurrent processes that don't share state.

All interaction through message passing.

Messages passed asynchronously to a mailbox …

… and processed selectively by pattern matching.



University of Kent

Key message #3: let it fail!

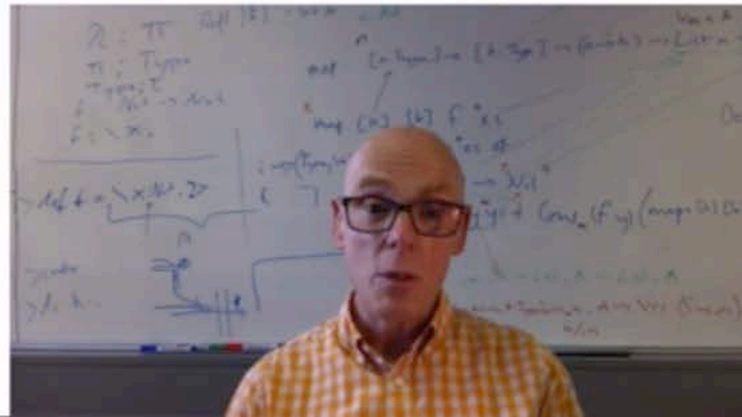
Structure processes for the usual case …

… and fail if you hit any other situation.

Supervisors start, stop and restart processes.

Uniform architecture to deal with all kinds of failure.

Links and trap_exit articulate signals and message passing.



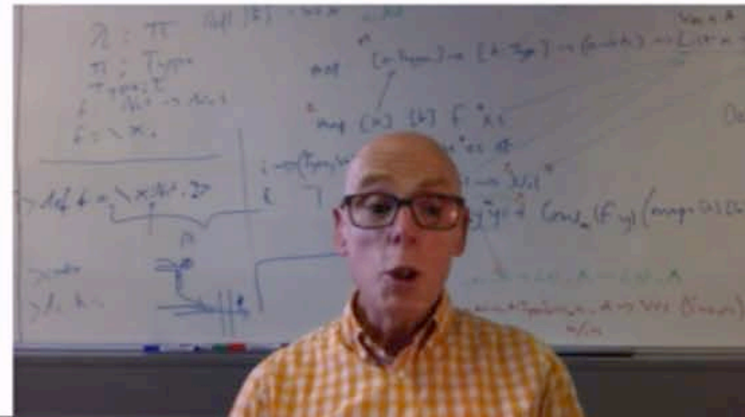
University of Kent

Key message #4: scale it up

Multicore BEAM gives scalability "out of the [black] box".

Processes and messages map directly to distribution model.

OTP gives behaviours and patterns for common components.



University of Kent

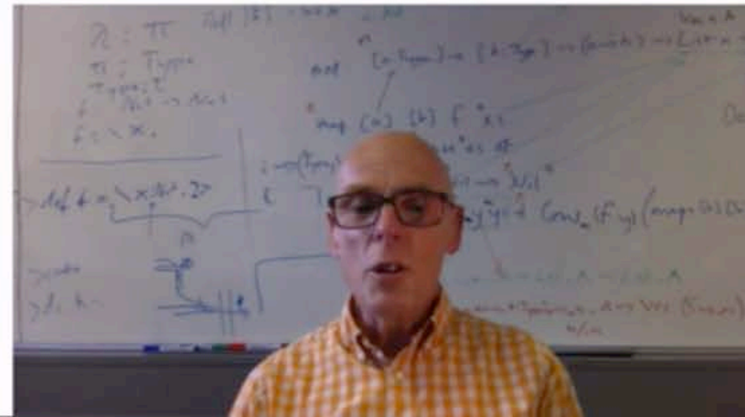
A beautiful design

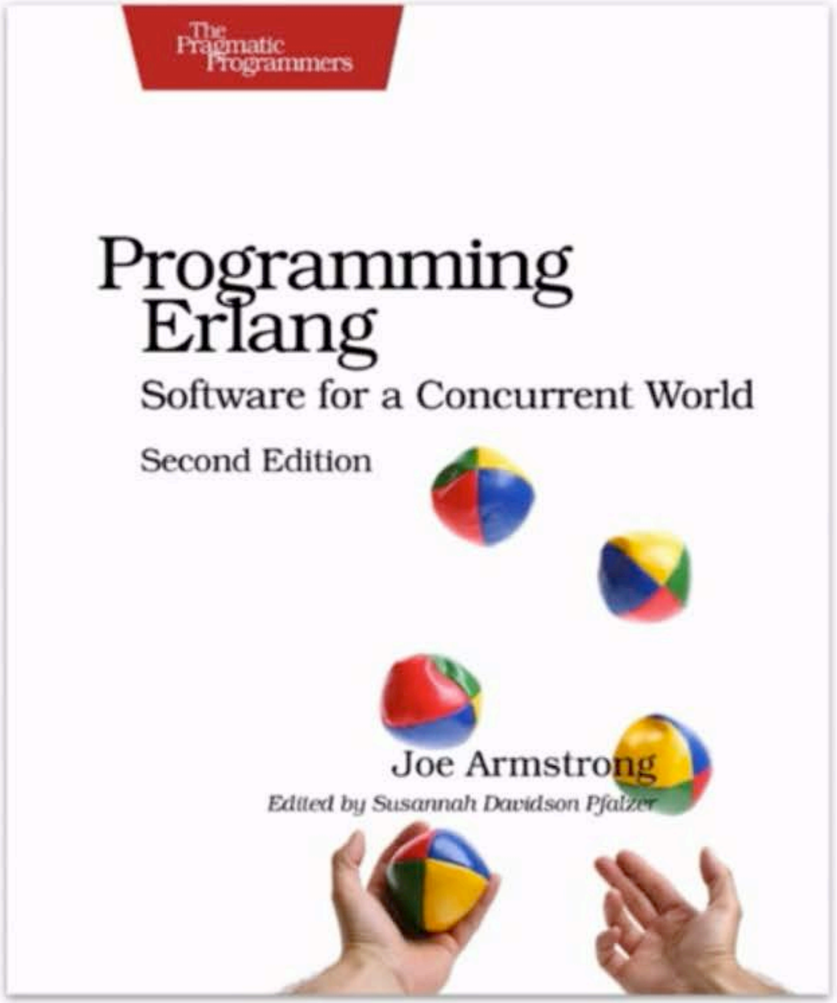
Mailboxes and message handling

Process errors and trapping exits

OTP generics

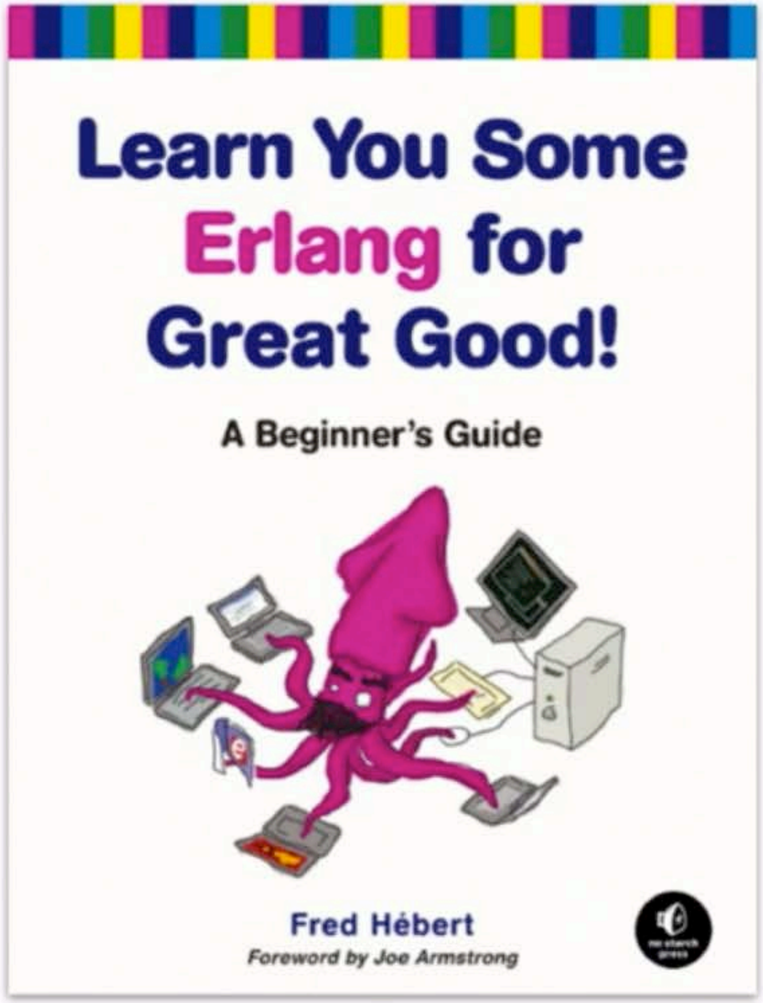
Concurrency and distribution





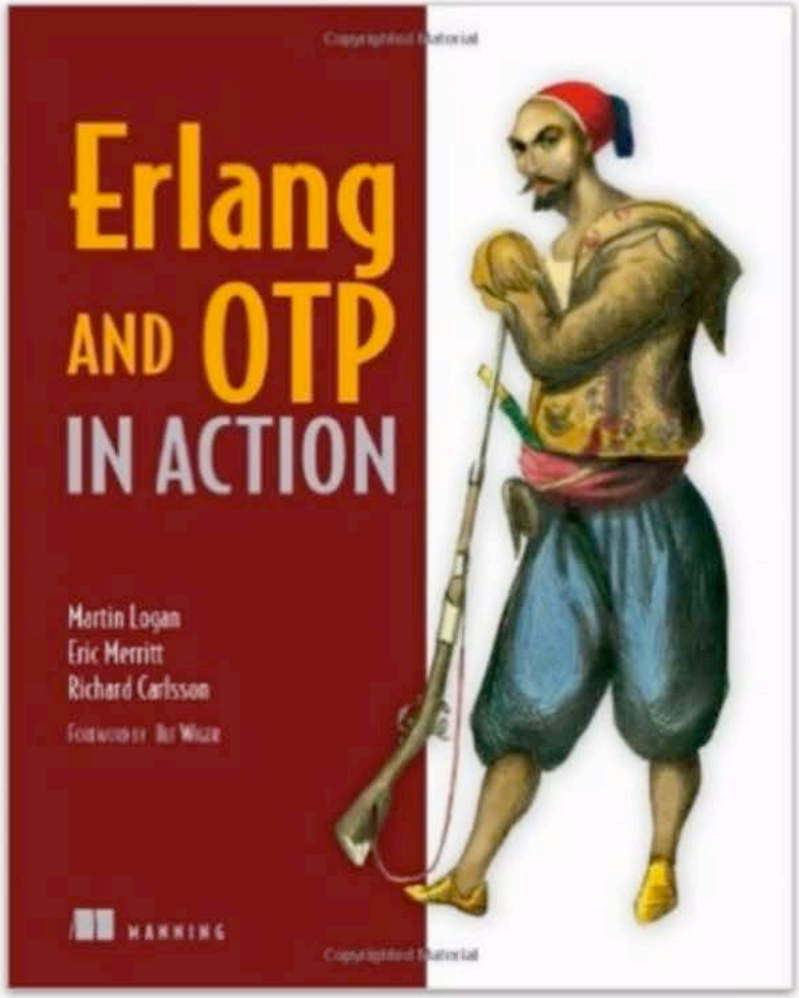
Programming Erlang, Second Edition, Joe Armstrong, Pragmatic Bookshelf, 2013

Erlang Programming, Francesco Cesarini and Simon Thompson, O'Reilly, 2009



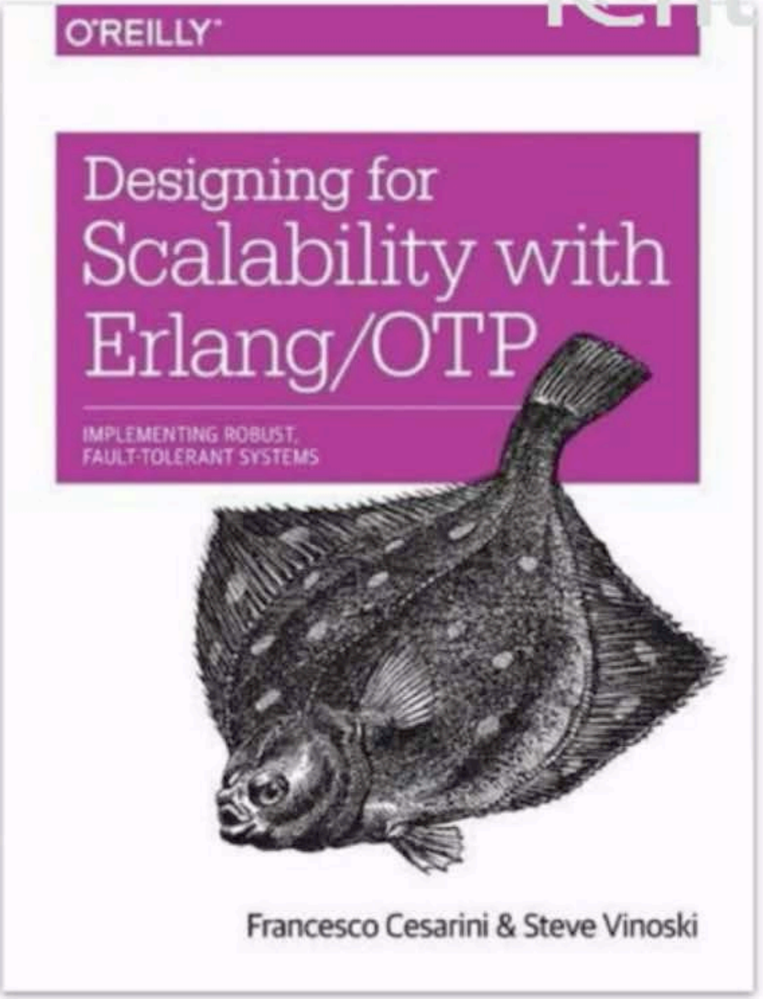
Learn You Some Erlang for Great Good, Fred Hébert, No Starch, 2013, *learnyousomeerlang.com*

Introducing Erlang, Simon St. Laurent, O'Reilly, 2013



Erlang and OTP in Action, Logan, Merritt and Carlsson, Manning, 2010

Designing for Scalability with Erlang/OTP, Cesarini and Vinoski, O'Reilly, 2016



University of Kent

Erlang online

Official Erlang site	`https://www.erlang.org`
Erlang central site	`http://erlangcentral.org`
Package downloads	`https://www.erlang-solutions.com/resources/download.html`
Twitter hashtag	`#erlang`
`#erlang` at freenode	

University of Kent

… and then there's Elixir

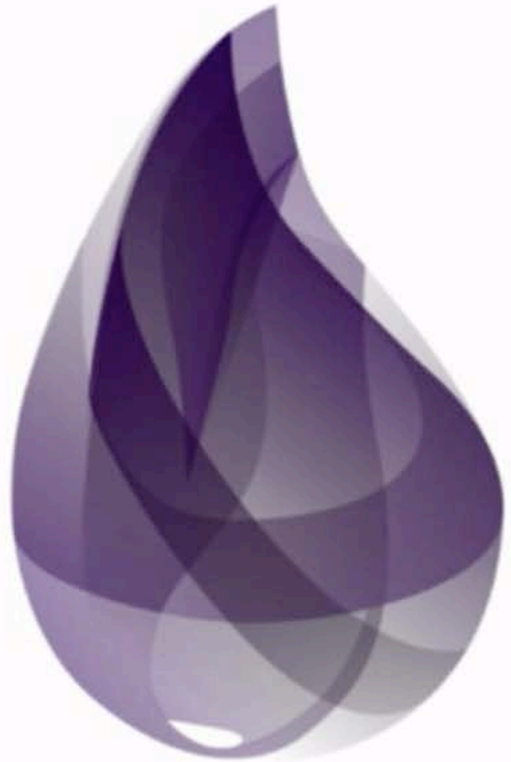
Elixir language `http://elixir-lang.org`

Runs on the Erlang VM, and is byte-code compatible.

Ruby-like syntax …

… and some changes to semantics, e.g. mutable vars :-(

Strong tool ecosystem: build, Phoenix web framework.



University of Kent

… and Akka

Akka toolkit and runtime `http://akka.io`

Runs on the JVM, with Java and Scala bindings.

"Erlang/OTP in Scala"

Processes, asynchronous communication, distribution, supervision hierarchies …

University of Kent

Simon Thompson @thompson_si 32d

Hello #erlang people on twitter: where would you recommend someone who has just learned erlang to find an open source project to work on?

University of Kent

Mariano Guerra
@warianoguerra

@thompson_si @jmiletm here's a long list by topic to look
efene.org/toolbox.html

Simon Thompson @tho
Hello #erlang people on twitter: where would you recommend someone who has just learned erlang to find an open source project to work on?

University of Kent

Mariano Guerra
@warianoguerra

@thompson_si @jmiletm here's a long list by topic to look efene.org/toolbox.html

Mark Allen @bytemeorg
@thompson_si The #erlang channel on free node is pretty good.

: where
ne who has
pen source

University of Kent

Mariano Guerra
@warianoguerra

@thompson_si @jmiletm here's a long list by topic to look efene.org/toolbox.html

Mark Allen @bytemeorg
@thompson_si The #erlang channel on free node is pretty good.

in reply to Simon Thompson

Serenity Du Fluff @SerenityFluff 23d
@thompson_si @Marutks you might want to reimplement software you've already written.
Kinda worked for me :p


University of
Kent