Erlang Solutions

Concurrency Oriented Programming

DevLin2025, Linköping
November 5, 2025

Erik Schön, Managing Director
Erlang Solutions Nordic

WHY Concurrency Oriented Programming?

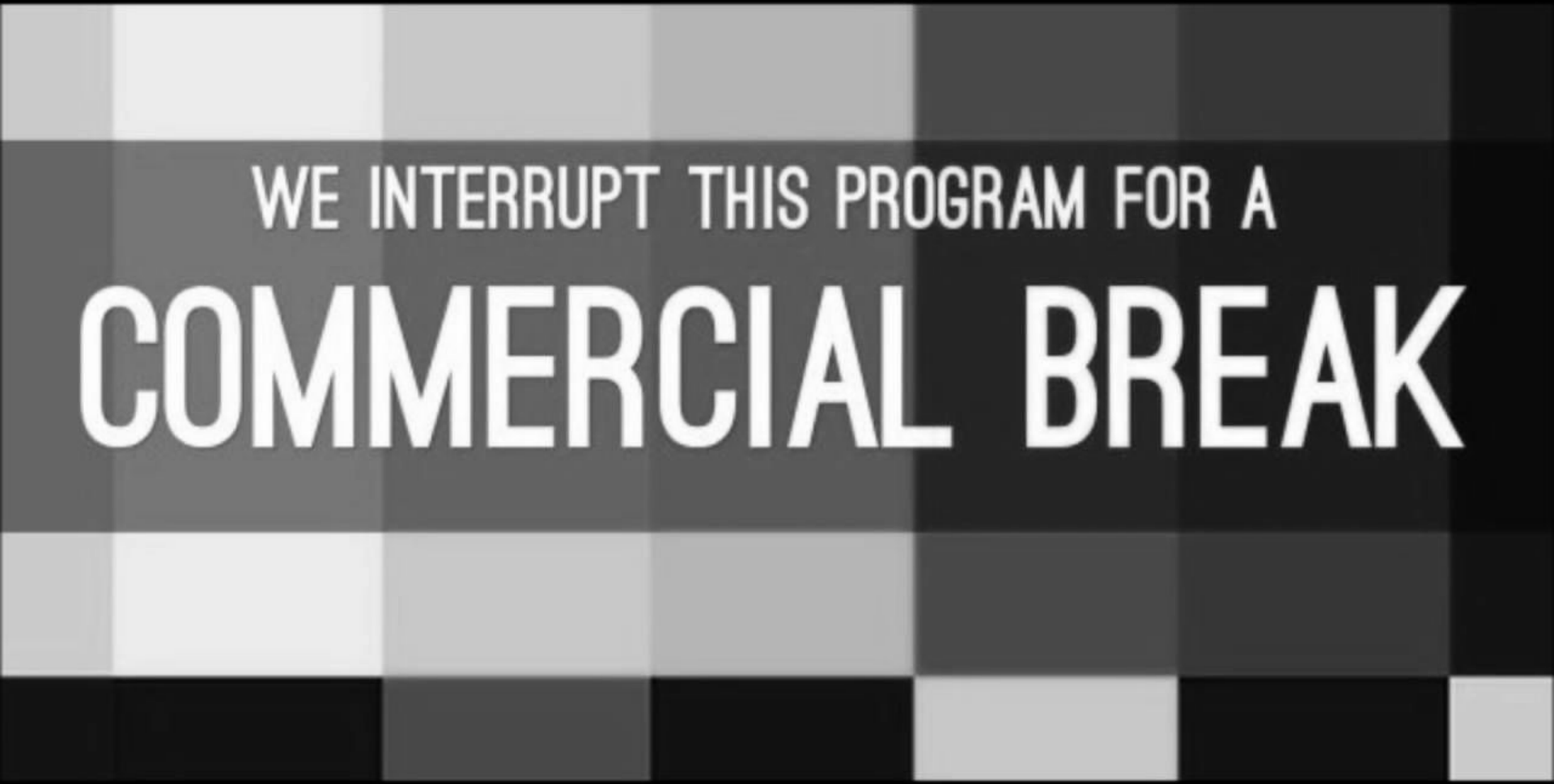
2x FASTER development of new services

10x BETTER services, down-time less than 5 minutes/year

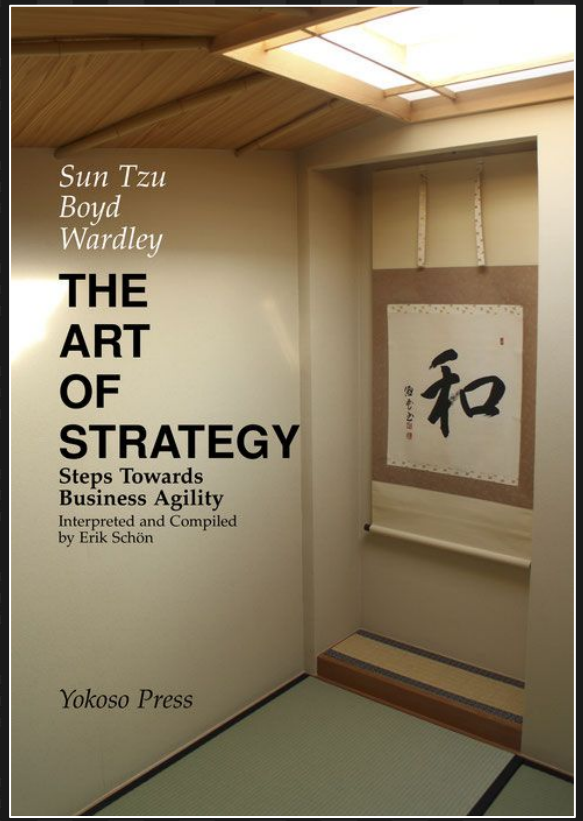
10x SAFER services that are very hard to hack or crash

10x MORE users and transactions – within milliseconds using
10x LESS costs and energy

References: Cesarini CVE


WE INTERRUPT THIS PROGRAM FOR A
COMMERCIAL BREAK

ERIK SCHÖN



Hacker
Software Researcher
System Engineer
Product Manager
Inventor
Manager
Leader
Executive
Speaker
Writer
Erlang Fan Boy

ERICSSON

NETENT

iramfab

Erlang Solutions

WHY Concurrency Oriented Programming?

2x FASTER development of new services

10x BETTER services, down-time less than 5 minutes/year

10x SAFER services that are very hard to hack or crash

10x MORE users and transactions – within milliseconds using
10x LESS costs and energy

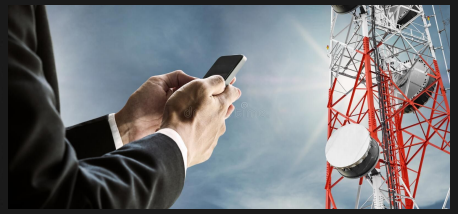
References: Cesarini CVE

YESTERDAY

WHAT WAS THE JOB TO BE DONE?

The Telecom World

Global voice calls



Mobile voice calls

Smartphones and mobile apps

1960 1970 1980 1990 2000 2010 2020

The Telecom World: Original Needs

Prototyping: it is not enough to have ideas, you need to be sure they work
Mike Williams
Co-creator of Erlang

Real-time, secure and scalable
Bjarne Däcker
Manager of the co-creators of Erlang

Reliable distributed systems in the presence of software errors
Joe Armstrong
Co-creator of Erlang

References: Armstrong (2003), Däcker (2000), Däcker (2009)

The (Telecom) World: First Principles

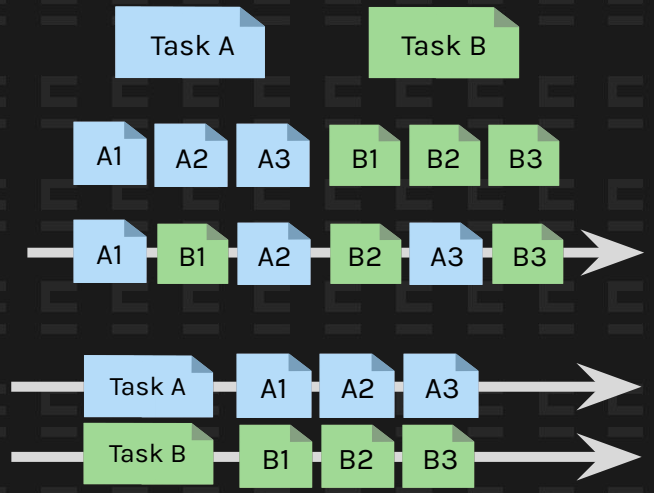
1. The world is concurrent
2. Things in the world don’t share data
3. Things communicate with messages
4. Things fail




Joe Armstrong, co-creator of Erlang

Principle 1: The World is Concurrent

- **Concurrency**
  Run many tasks on one unit
- **Parallelism**
  Run many tasks on several units
- **Distribution**
  Run many tasks on several units ***wherever***
- **Key Insight**
  “Automagically” handle concurrency, parallelism and distribution!

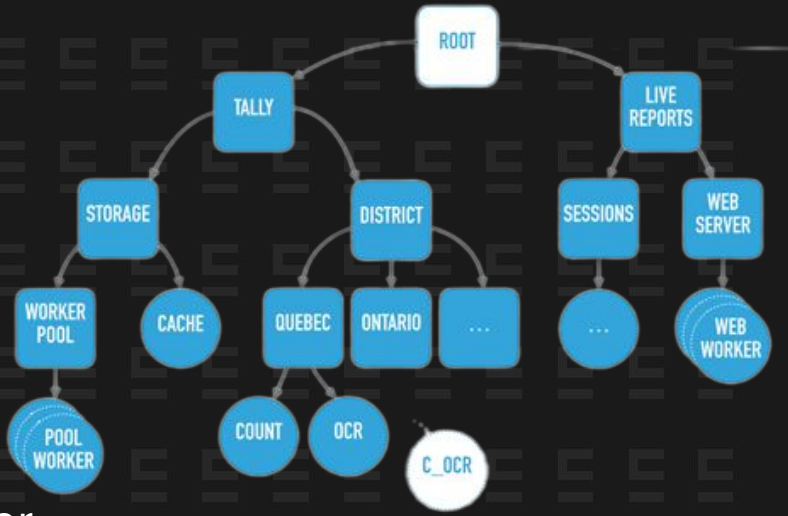


Principles 2 and 3:
Things Don’t Share Data and Communicate with Messages

Key Insight

Treat processes as immutable data structures to easily handle distribution and failures!

Principle 4: Things Fail



- **“Anything that can go wrong will go wrong”**
  Edward A. Murphy Jr.
- Supervision trees to isolate failures and recover
- Let small, isolated parts crash *responsibly* to safeguard overall system
- **Key Insight**
  ”Automagically” embrace failure *over* trying to prevent failure!

Reference: Hebert

TODAY

WHAT IS THE RIGHT TOOL FOR THE JOB?

WHY Erlang & Elixir for Concurrency Oriented Programming?

Robert Virding's (co-creator of Erlang)
First Rule of Programming

Any sufficiently complicated concurrent program
in another language contains
an ad hoc informally-specified bug-ridden
slow implementation
of half of Erlang

The Erlang Programming Language

- Functional/declarative
- Immutable data and message passing
- Concurrency and error handling primitives
- Open source – since 1998: “best European open-source since Linux”
- An amazing environment: a set of tools and help for building systems

The Elixir Programming Language

- Inherits the Erlang virtual machine and environment
- Adds influences from Ruby and Clojure
- Adds front-end focus
- Open source – since 2012
- An amazing developer experience:
  Top-2 most admired programming language
  Most admired web development framework: Phoenix LiveView Native

Reference: Stack Overflow Developer Surveys with 50,000+ participants 2022-2025

Powered by Elixir & Erlang

TOMORROW

EVEN BETTER
Gradual typing

EVEN FASTER
Full-stack development

EVEN MORE FOR LESS
Effective generative AI

Backed by
WhatsApp/Meta
Ericsson

Erlang
Ecosystem
Foundation

Concurrency Oriented Programming: Q&A

2x FASTER development of new services

10x BETTER services, down-time less than 5 minutes/year

10x SAFER services that are very hard to hack or crash

10x MORE users and transactions – within milliseconds using
10x LESS costs and energy

Perfect for digital infrastructure critical for society
fintech, digital health, media, telecom, defence, ...

motorola

ERICSSON

CVE® Common Vulnerabilities and Exposures

WhatsApp

BBC

Erlang Solutions

References: Cesarini CVE

For Further Inspiration

Armstrong, Joe. Making Reliable Distributed Systems in the Presence of Software Errors

Cathcart, Will. Improving WhatsApp's Server Efficiency by 25%

Cesarini, Francesco. Which Companies Are Using Erlang, and Why?

CVE. CVE Security Vulnerability Database

Däcker, Bjarne. Concurrent Functional Programming for Telecommunications

Däcker, Bjarne. CS Lab and All That ...

Hebert, Fred. The Zen of Erlang

Rubio, Manuel. Which Companies Are Using Elixir, and Why?

Valim, José. Nx (Numerical Elixir) is Now Publicly Available

Virding, Robert. Virding's First Rule of Programming