Lisp-Flavoured Erlang

or How to write OTP apps without learning Erlang

WTF?

- Courtesy of Robert Virding
- It is not a Common Lisp, Scheme or Clojure implementation.
- It is a Lisp-2 that runs on the Erlang VM.
- It interoperates with OTP/Erlang.
- It has a single flat namespace (like Erlang).

Why would I use it?

- Use OTP supervision and distribution facilities with Lisp.
- Makes Erlang records tolerable. HA!
- You’re addicted to pattern matching.
- You’re just bored.

Gotchas

- Most stuff is compile-time only (so far).
- Functions have names and arity.
- You don’t have to learn Erlang, but you will have to learn about its types.
- You’ll also have to be comfortable with module and behaviour definitions.

A simple OTP FaaS app

- We’ll use **rebar** for build and dependency management.
- We’ll create 3 modules: **`lfe_example_app.erl`** (application)
- **`lfe_example_server.erl`** (gen_server)
- **`lfe_example_sup.erl`** (supervisor)

For more info:

- Github repos
  - LFE: https://github.com/rvirding/lfe
  - rebar: https://github.com/basho/rebar
  - this example: https://github.com/seansawyer/lfe-example
- Nice PDF from Virding:
- http://www.erlang-factory.com/upload/presentations/325/EFLA10-LispFlavouredErlang.pdf