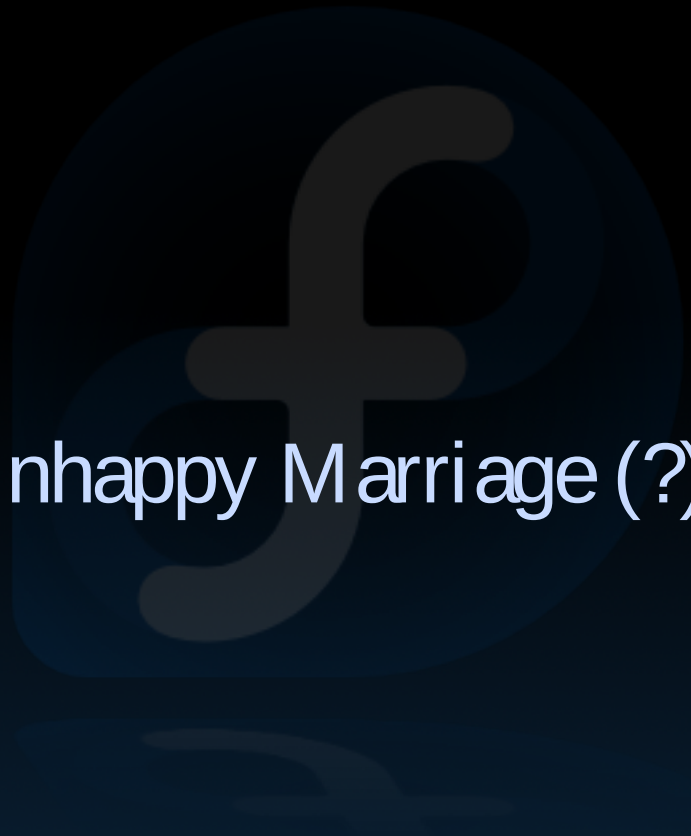
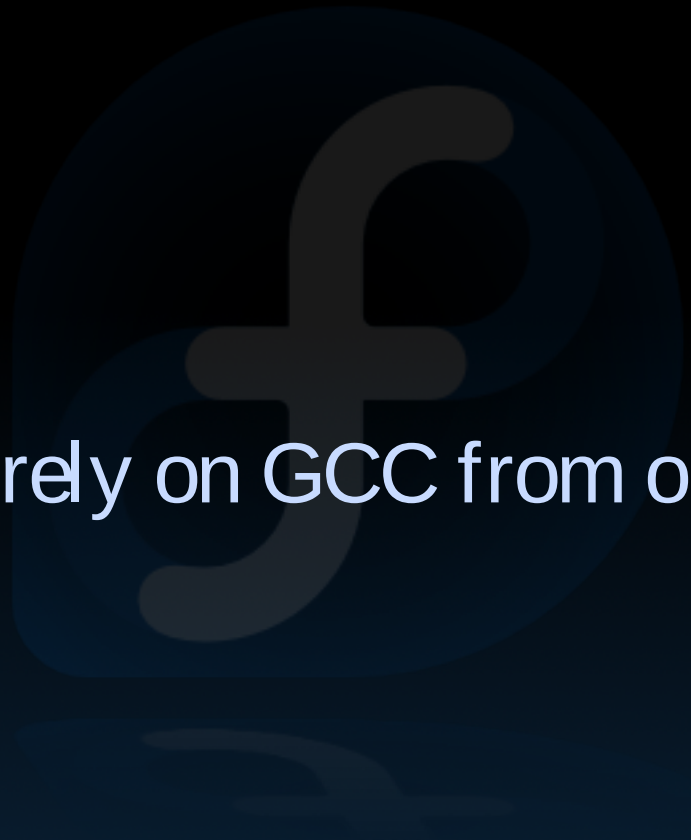
Erlang & distributions



Unhappy Marriage (?)

Intro



- Why we rely on GCC from our distro?

Difference between GCC and Erlang

- Young (immature) project
- Functions appear and disappear
- Backwards incompatibility

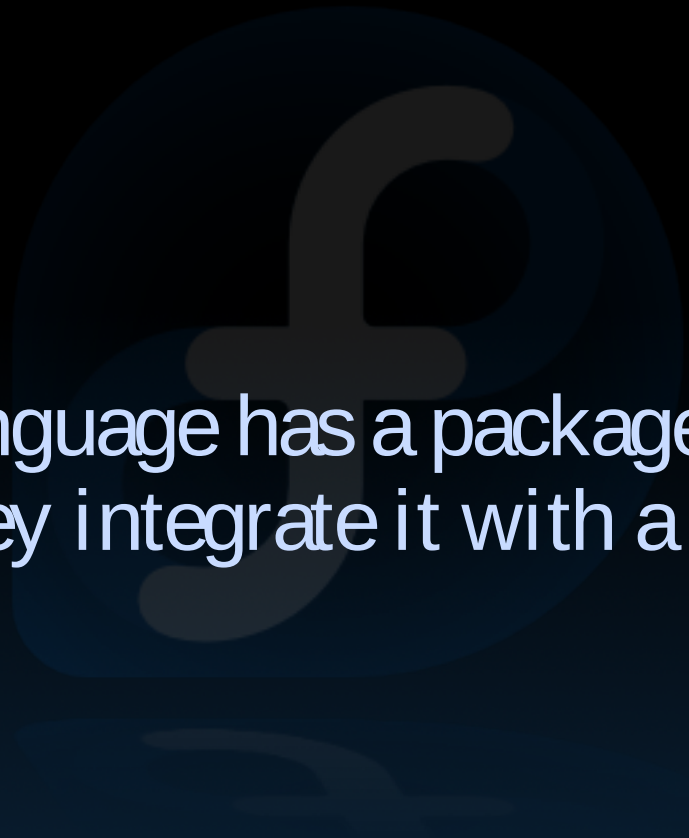
Why people don't use distro packages?

- Very few packages
- “Frozen” versions of apps and libraries
- We have rebar and that's all we need.

What developers want?

- Deliver software to the People/Target Hosts
- Ensure that it's up-to-date
- Prove that it works

A bit of history

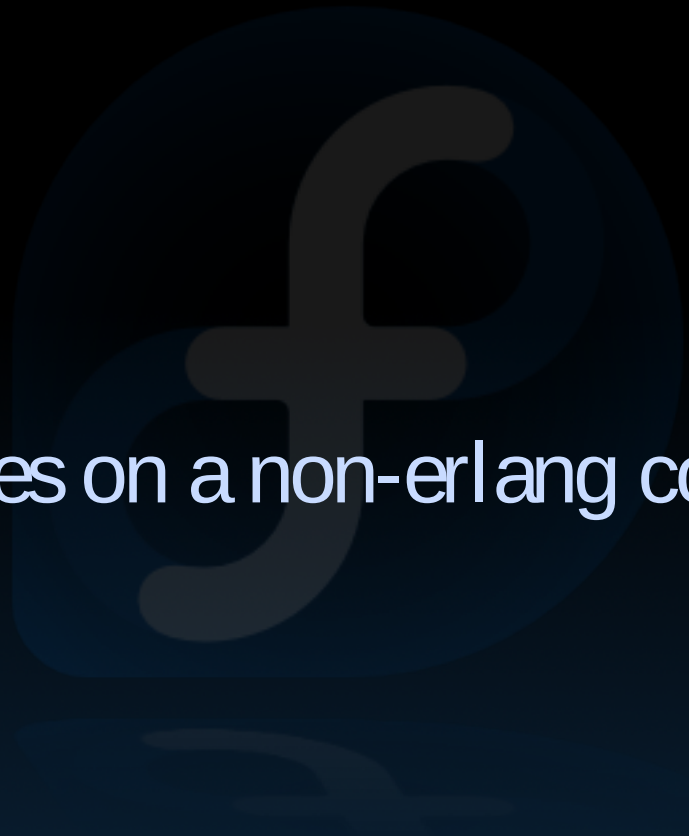


- Every language has a package manager
- Later they integrate it with a real P.M.

Updates and delivery

- Drop a zipball with a rebar-generated release.
- Drop a package built with the package builder from a rebar-generated binary data – much better!

Why package management?



Dependencies on a non-erlang components.

Why not rebar only?

- Different stages - prepare, build, pack, compute dependencies, upload, install/upgrade/remove.
  - Rebar can be used on a first three stages.

Rebar and package managers

- Rebar is a build tool.
- A package manager can compute dependencies
- A P.M. ensures that all dependencies are met.
- A P.M. prevents from breaking a dependency chain

How to create dependency chain?

- Meet beam_lib application.
- Provide hooks for post-build stage.
- Rebuild all packages with new dependency generator.

Caveats

- Unbundle bundled 3rd party libs
- Be careful with eunit

How about releases?

- Yes, it's possible.
- A dependency generator needs to be fixed (strict version dependency – as in Ocaml)
- All known popular package managers allow simultaneous installation of several versions of a package (until they don't conflict with each other).

Proposal

You should participate in packaging of an Erlang-related stuff for your favorite distro / OSX / BSD.

Why really?

- “I did because I can”
- You're already providing some kind of packaging so why not to make one step further?
- Overall quality will be better → more happy users.
- You will borrow others' expertise for free in some border cases.

Predictions

- Integration between rebar and package management systems.
- Further expanding Erlang support for Autotools.
- A simple and handy library for dealing with beam-files (written in Python).

Thanks

- Author: Peter Lemenkov, Fedora Project.
- E-mail: lemenkov@gmail.com
- G+: https://www.google.com/profiles/lemenkov