Building Erlang Apps with Rebar

Dave Smith (dizzyd@basho.com)

Goals

Introduce the major features and functionality that rebar provides to Erlang developers.

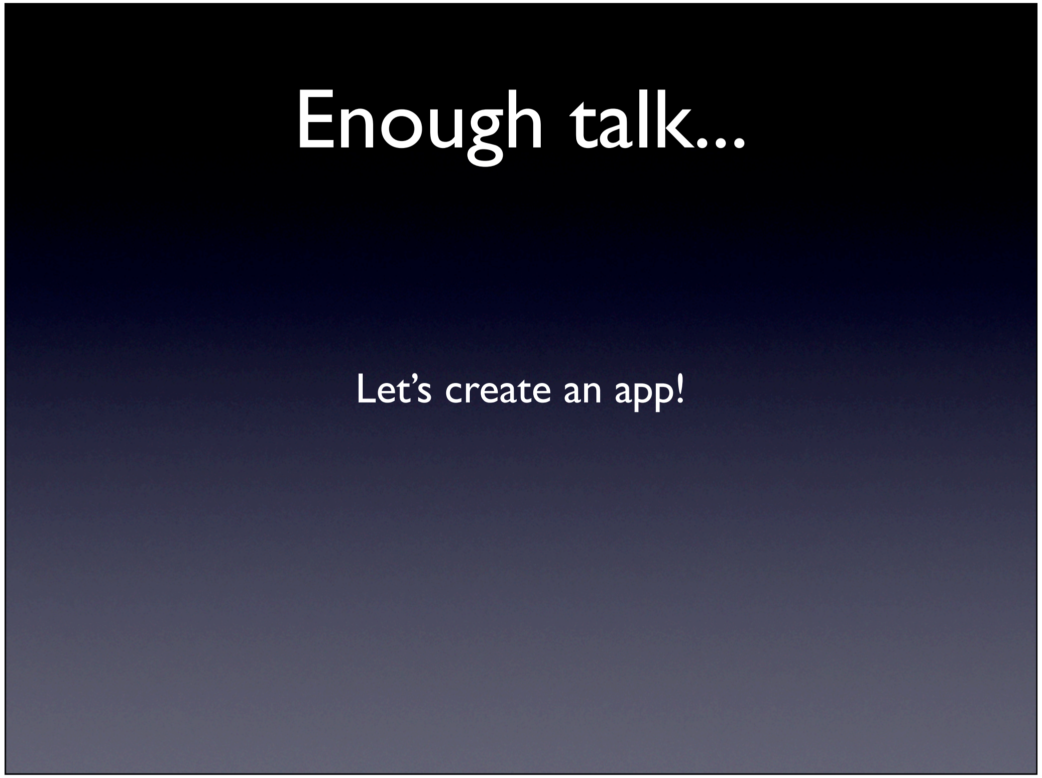
Examine the architecture of rebar and discuss how it can be extended.

Background

- Makefiles : lots of duplication
- Scons : hard to install
- erlbox (Rakefiles) : hard to maintain

Acknowledgements

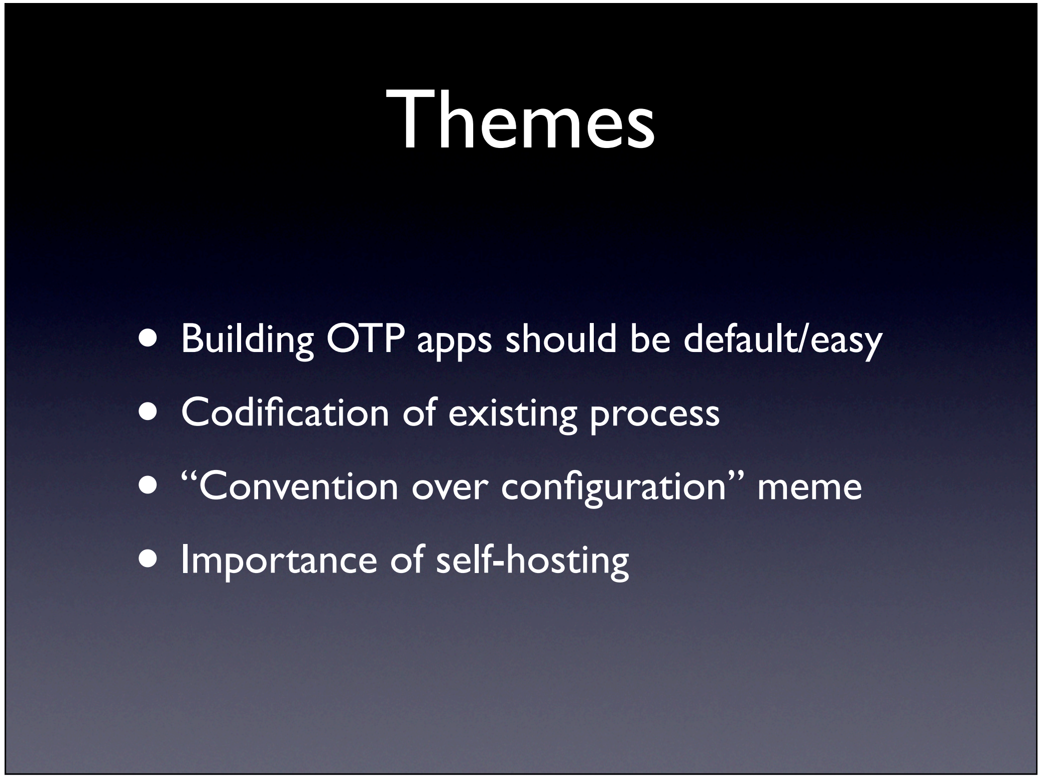
- Basho Crew
- Tuncer Ayaz, Joe Williams, Bob Ippolito
- 47 contributors in total!



Enough talk...

Let’s create an app!

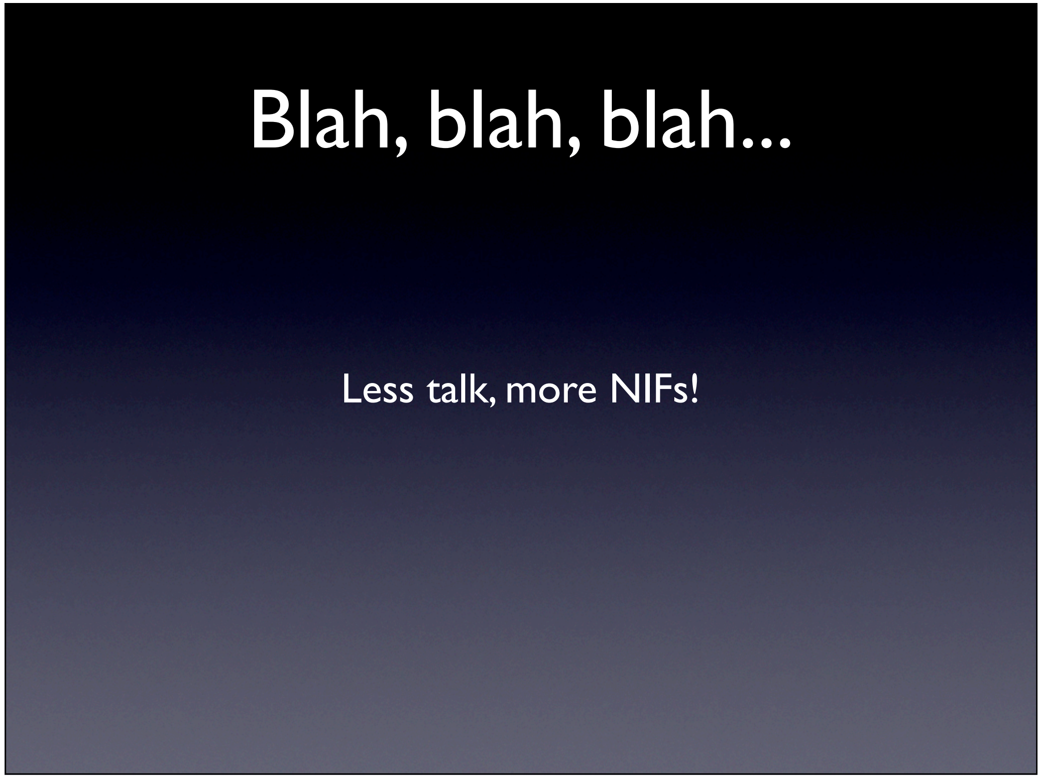
```
mkdir helloapp
cd helloapp
rebar list-templates
rebar create template=simpleapp appid=helloapp
rebar compile
rebar create template=simplesrv srvid=helloapp_server
* Add entry to supervisor ?CHILD(helloapp_server, worker)
* Define hello/1 that io:formats + returns formatted string
rebar compile
* erl -pa `pwd`/ebin
* application:start(helloapp).
* helloapp_server:hello(foobar).
```



Themes

- Building OTP apps should be default/easy
- Codification of existing process
- “Convention over configuration” meme
- Importance of self-hosting

* Out of the box, rebar should satisfy 80% of functionality a dev needs
* DRY
* Best changes to any dev process is incremental; big bang should be avoided
* Self-hosting + using existing practices == enables minimal commitment

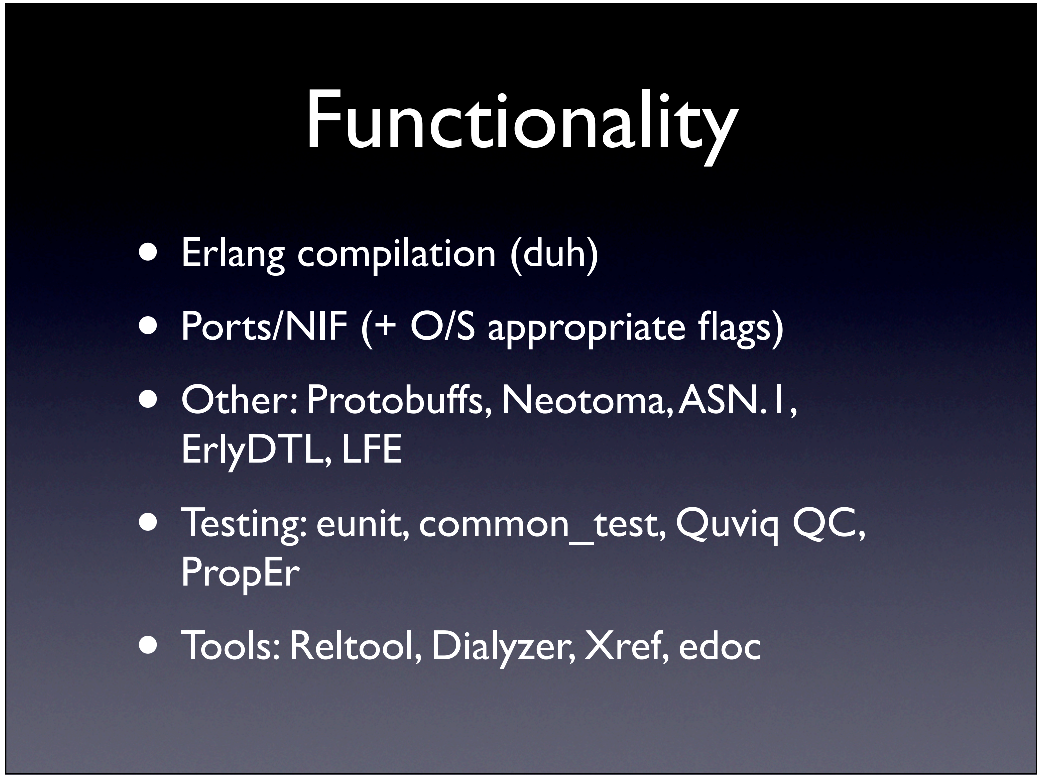


Blah, blah, blah...

Less talk, more NIFs!

rebar create template=basicnif module=helloapp_nif
* Edit helloapp_nif.erl to correct priv_dir call and SoName
* application:start
* helloapp_nif:myfunction(helloapp_nif:new()).

rebar eunit (note new test)



Functionality

- Erlang compilation (duh)
- Ports/NIF (+ O/S appropriate flags)
- Other: Protobuffs, Neotoma, ASN.1, ErlyDTL, LFE
- Testing: eunit, common_test, Quviq QC, PropEr
- Tools: Reltool, Dialyzer, Xref, edoc

Also release/upgrade generation w/ systools!

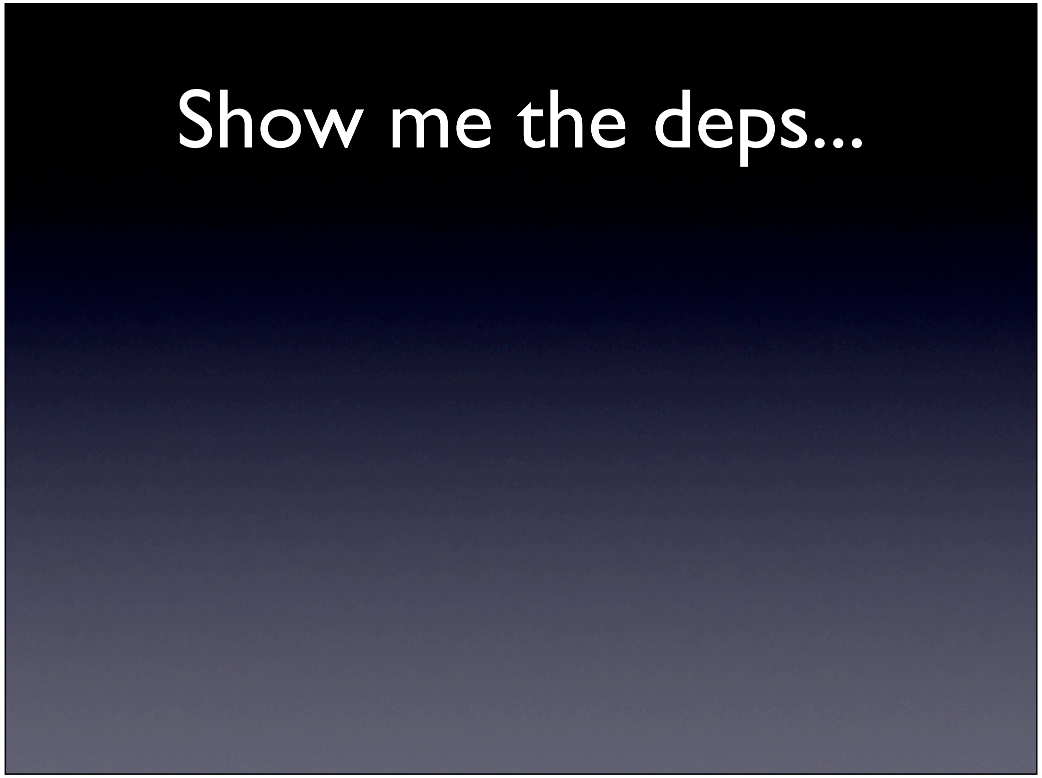
Hey, fool...

Gimme a release!

```
cd .. && mkdir hello
rebar create template=simplenode nodeid=hello
* Edit reltool.config; add lib_dirs .. and {app, helloapp, [{incl
* Edit reltool.config for   {excl_archive_filters, [".*"]},
rebar generate
* cd hello
* bin/hello console
* application:which_applications().
* helloapp_server:hello(foobar).
```

Embedded Releases

- Power of run_erl/to_erl
- Product value of embedded ERTS
- Safety of self-contained dependencies
- Tested =:= Deployed



Show me the deps...

not designed for usage with things installed in your OTP (or ERL_LIBS, for that matter). The expectation was that dependencies were _external_ locations which would automatically be downloaded and compiled.