



Organizing Erlang Deployments

or just thinking about stuff aloud

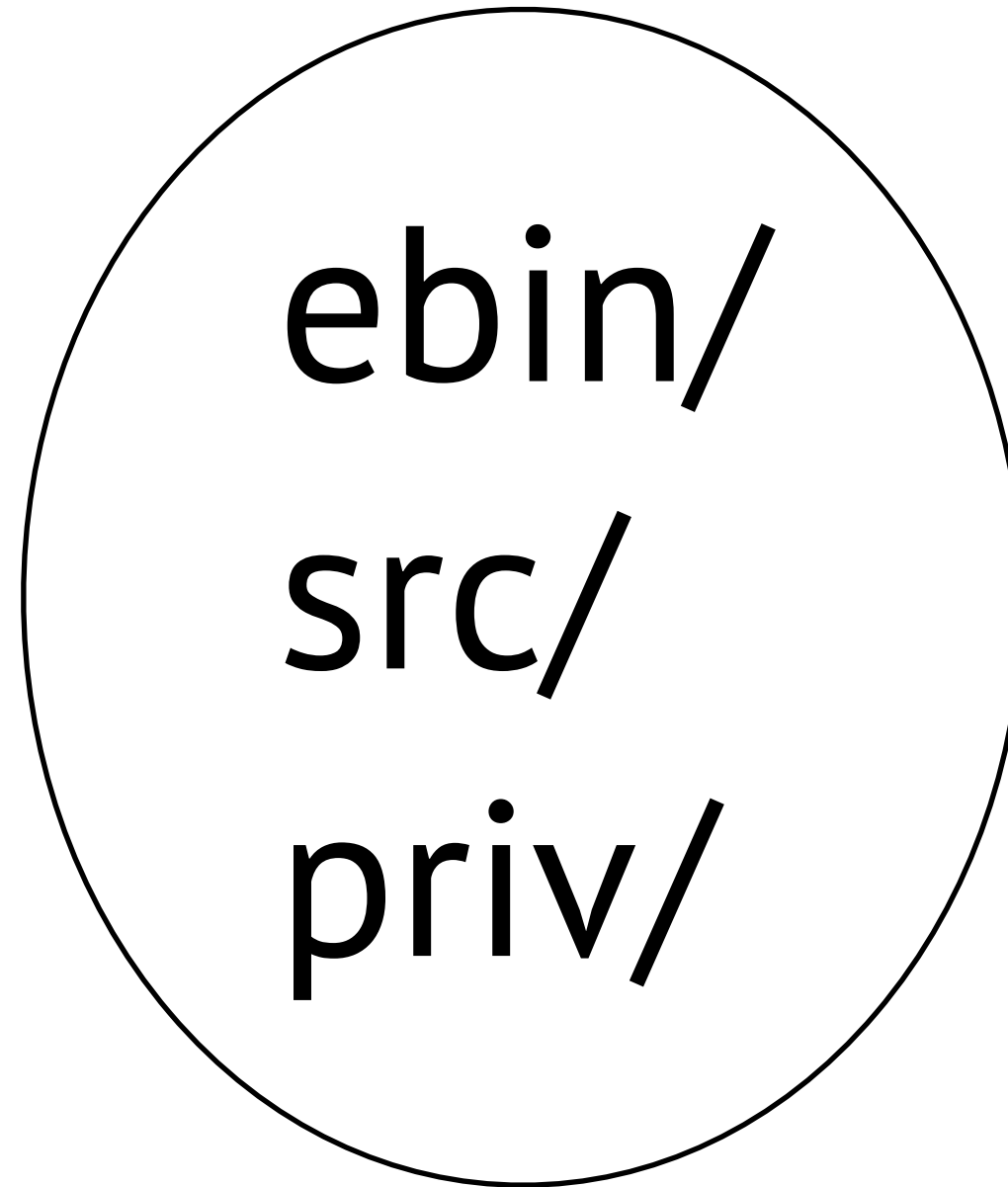
Yurii Rashkovskii
EUC 2011

Good stories don't
start at the end...

Deployment starts on
a developer's machine.

So I'll skip the whole
deb/rpm thing for now.

Application



Project

ebin/
src/
priv/

ebin/
src/
priv/

ebin/
src/
priv/

ebin/
src/
priv/

apps

ebin/
src/
priv/

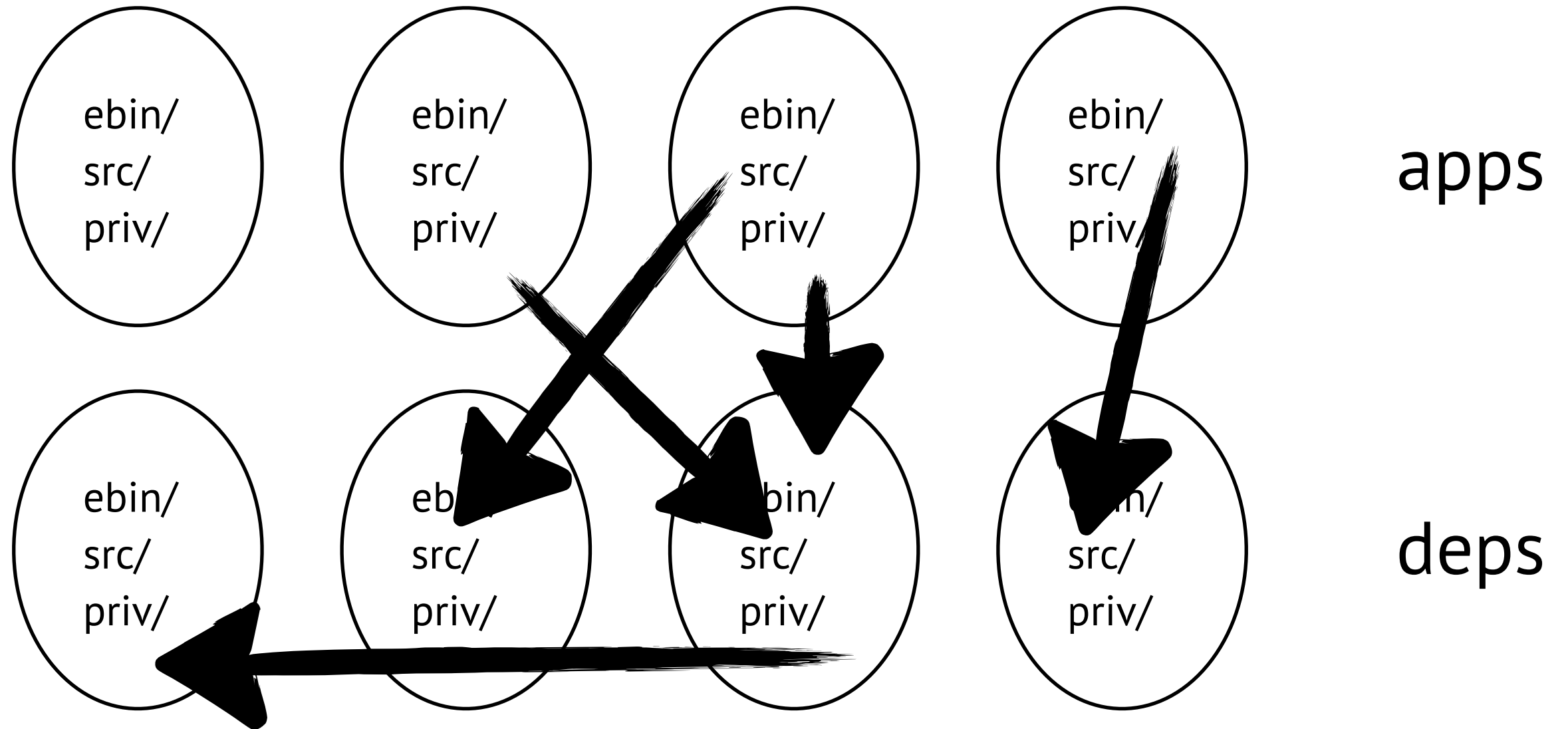
ebin/
src/
priv/

ebin/
src/
priv/

ebin/
src/
priv/

deps

Dependencies



How do we start all
this stuff?

Releases

Releases

- Used to be really hard to do.
- 2009: reltool and
- 2010-2011: rebar generate.

Almost no excuses to
not use them.

Yeah, still kind of on a
slow side.

Well, there's ~~an app~~
plugin for that.

Quickrel

(essentially, a Quickhack)

releases/myrel

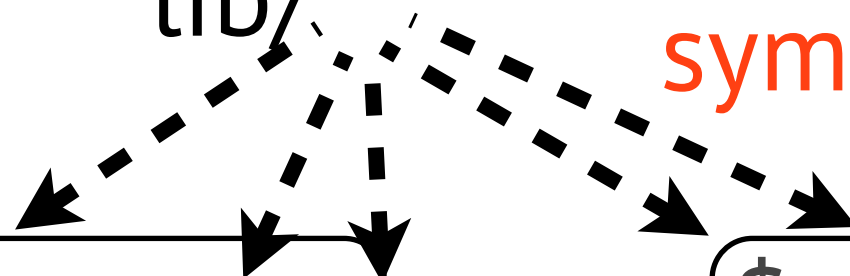
releases/1.0/myrel.boot

lib/

symlinks!

myrepo/apps/app1
myrepo/apps/app2
myrepo/deps/dep1
myrepo/deps/dep2

\$otp/lib/kernel-x.xx
\$otp/lib/stdlib-y.yy
\$otp/lib/sasl-z.zz
...



Quickrel

(essentially, a Quickhack)

How does it know what should go into the release?

Your reltool.config you already have

Quickrel

(essentially, a Quickhack)

<https://github.com/spawngrid/quickrel>

Not a replacement for
real releases

Not really for hot code
reload maniacs

But gives you a quick
solution when other
methods don't

Okay, fine.

How do we get this stuff
out to the servers?

You can learn some
Ruby and use Chef.

You can learn some
Puppet and use Puppet.

But hey!

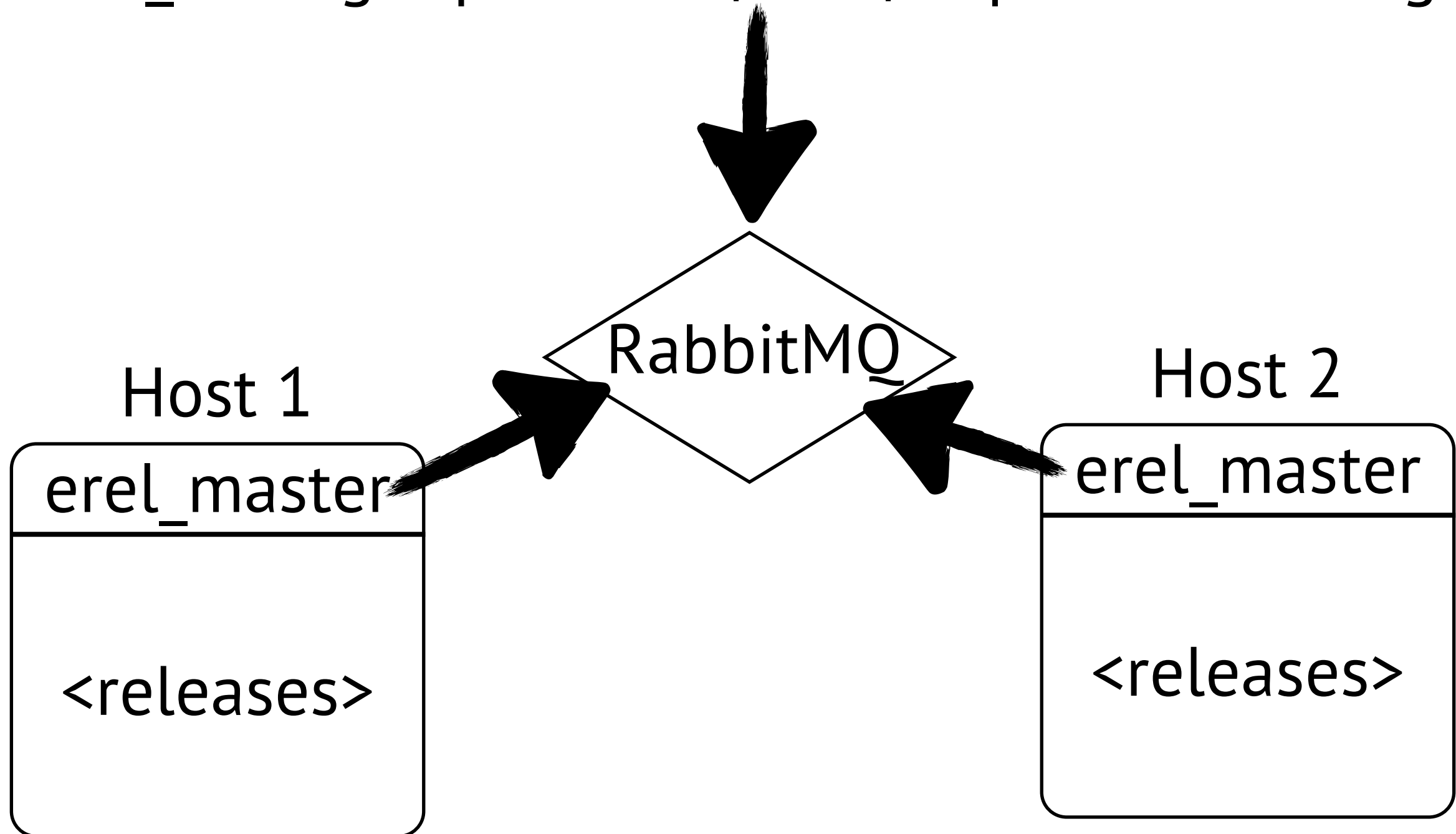
How about using Erlang
stuff? I mean, seriously.

Besides, there's an app
for that.

erel

- Group-based deployment.
- Lightweight, simple config.
- Works with your reltool+rebar releases
- Uses RabbitMQ (moar Erlang!)
 - (can be extended to use other pubsubs)

\$ erel_manager provision|start|stop cluster.config



config file

```
{group, "Main", ["host1", "host2"]}
```

```
{release, "myrel", "path/to/myrel"}
```

```
{deploy, "myrel", ["Main"]}
```

(it will become a bit
more flexible soon,
though)

Caveat: no pipes!

- Can't attach to a deployed node's console
- But can **-remsh!**

Caveat: no pipes!

- Some would argue that console is sometimes necessary!
- Like, for example, for **rb:list()**.

Caveat: no pipes!

- Well, you don't really need a pipe for that
- Group leaders!

mynode@server

group_leader() <0.810.0>

sasl_sup

rb_server

remsh@ops

group_leader() <0.24.0>

Change rb_server's group leader to remsh's group leader (and don't forget to switch it back!)

mynode@server

group_leader() <0.810.0>

sasl_sup

rb_server

remsh@ops

group_leader() <0.24.0>

Or simply start rb_server outside of sasl_sup supervision tree.

erel

- A very young tool (read: almost ready)
 - 1st release expected by the end of Nov
- Lots of improvements already planned.
- I am sure there are nasty bugs.
 - (would that stop a true Erlanger?)

Lots of improvements planned

- Upgrades (didn't have enough time!)
- Cookie management
- Primitive release monitoring
- Not only “all or none” quorums
- Better quorum recovery (master crashes)
- Something awesome (proper_statem?)

erel

<https://github.com/spawngrid/erel>

But some of us still
need rpm/debs!

Primary motivation:
no mind control over
sysadmins!

Primary motivation:
sysadmins are more
comfortable with
them.

That's okay.
Understandable.

But packaging every
app individually is just
wrong

Version clashes,
=:= ERTS versions,
mixed use of 32/64
VMs...

But there's (almost?)
nothing wrong with
packaging a release
into a deb/rpm.

Or erel_master.

I guess I am done.

Questions?

Thanks.