

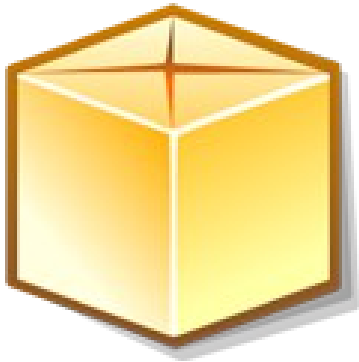
# LuaRocks

fostering an ecosystem  
of Lua modules

Hisham Muhammad  
[h@hisham.hm](mailto:h@hisham.hm)  
[@hisham\\_hm](https://twitter.com/hisham_hm)

FOSDEM 2015  
Brussels, Belgium

# What is LuaRocks



## Package manager...

like dpkg (apt-get), RPM, etc.

like RubyGems, Python eggs,  
npm, CPAN, etc.



## ...for Lua extension modules

written in Lua (.lua files)

or binary modules (.so/.dll files)

# Fostering an ecosystem of Lua modules

## Some unique challenges

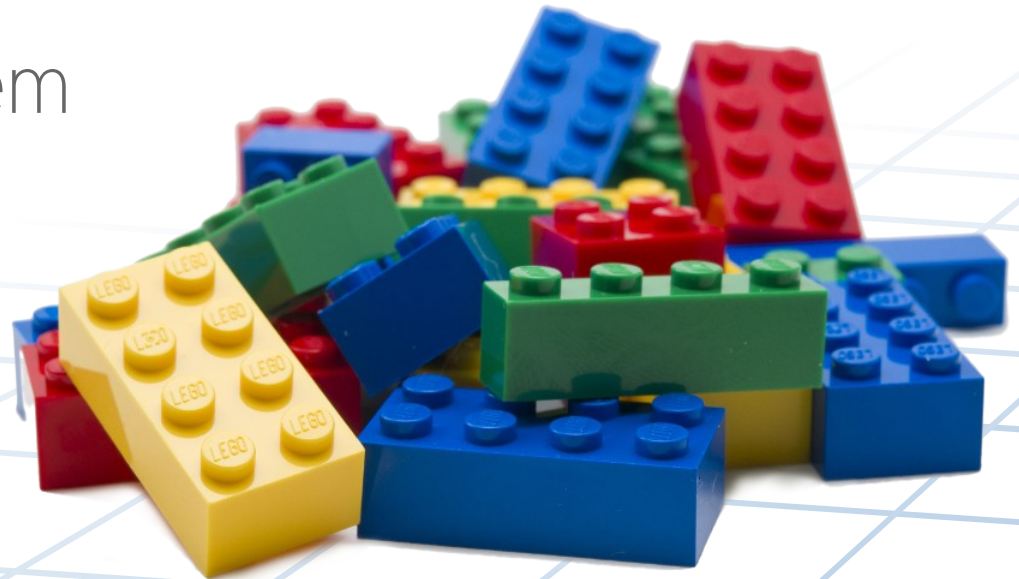
...related to Lua as a language

...related to Lua as a community



## Lessons learned

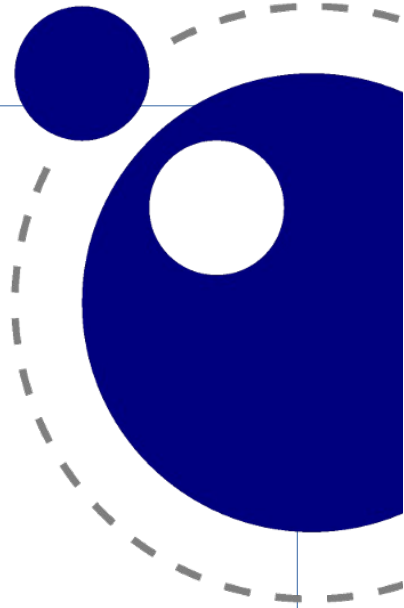
...on building an ecosystem



# A one-slide introduction to Lua

```
function process(filename, fn, ...)
  local f = io.open(filename)
  local rets = {}
  for line in f:lines() do
    rets[#rets+1] = { fn(line, ...) }
  end
  f:close()
  return rets
end

matches = process("file.txt", string.find, "foo")
for i, match in ipairs(matches) do
  print(i, table.concat(match, ", "))
end
```

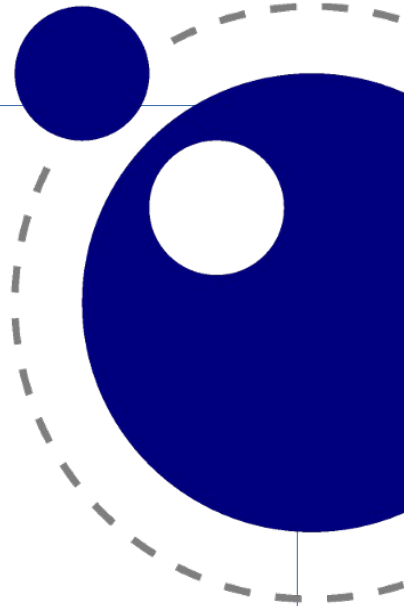


# A one-slide introduction to Lua

```
function process(filename, fn, ...)
  local f = io.open(filename)
  local rets = {}
  for line in f:lines() do
    rets[#rets+1] = { fn(line, ...) }
  end
  f:close()
  return rets
end
```

objects

```
matches = process("file.txt", string.find, "foo")
for i, match in ipairs(matches) do
  print(i, table.concat(match, ", "))
end
```

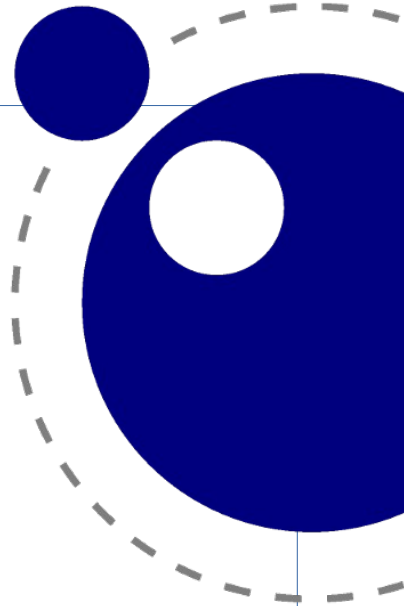


# A one-slide introduction to Lua

```
function process(filename, fn, ...)
  local f = io.open(filename)
  local rets = {}
  for line in f:lines() do
    rets[#rets+1] = {fn(line, ...)}
  end
  f:close()
  return rets
end
```

first-class functions

```
matches = process("file.txt", string.find, "foo")
for i, match in ipairs(matches) do
  print(i, table.concat(match, ", "))
end
```

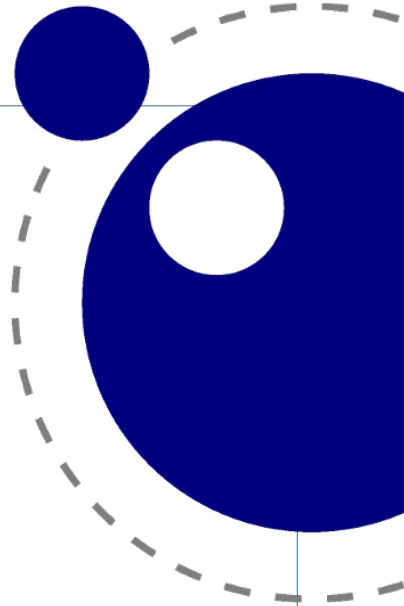


# A one-slide introduction to Lua

```
function process(filename, fn, ...)
  local f = io.open(filename)
  local rets = {}
  for line in f:lines() do
    rets[#rets+1] = { fn(line, ...) }
  end
  f:close()
  return rets
end
```

```
matches = process("file.txt", string.find, "foo")
for i, match in ipairs(matches) do
  print(i, table.concat(match, ", "))
end
```

iterators

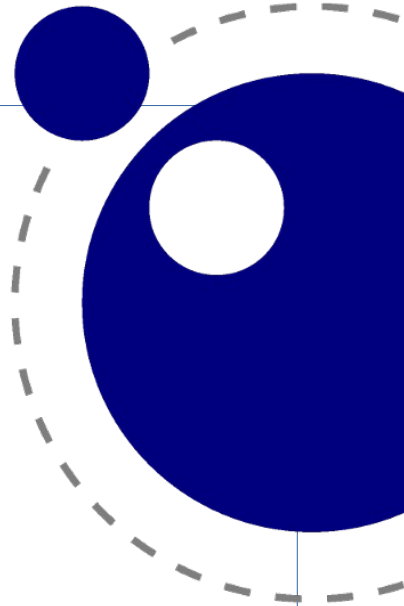


# A one-slide introduction to Lua

```
function process(filename, fn, ...)
  local f = io.open(filename)
  local rets = {}
  for line in f:lines() do
    rets[#rets+1] = { fn(line, ...) }
  end
  f:close()
  return rets
end
```

tables, tables everywhere!

```
matches = process("file.txt", string.find, "foo")
for i, match in ipairs(matches) do
  print(i, table.concat(match, ", "))
end
```





# Lua: a peculiar free software project



**Lua:** created at PUC-Rio in Brazil, developed by university professors

20 year history

MIT/X11 licensed

"open source, closed development"

# Lua: a peculiar community



Successful as an embedded language

LuaTeX, Wikipedia, etc.

very custom environments

Popular in the game industry

everywhere from Angry Birds to WoW

one-shot projects, mostly proprietary

# Lua: a peculiar language

## Super lightweight

Designed for embedding  
(.tar.gz is 272Kib)

Adapts itself to your project's API

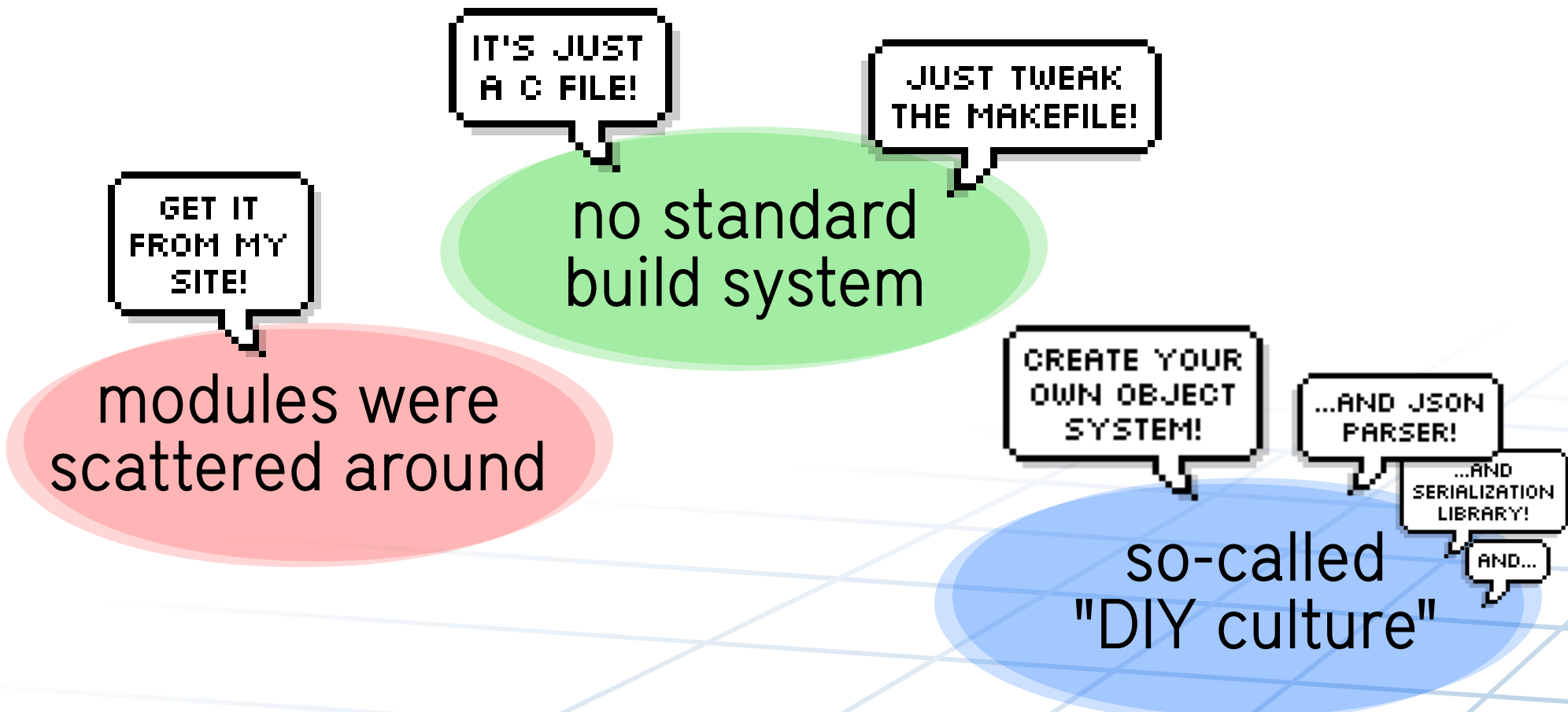
Doesn't provide many APIs of its own

Modules fill this gap!



# But where are the modules?

Side-effects of a **minimalistic** language:



(Lua only got modules in version 5.0, after 10 years of existence!)

# Attempting to broaden Lua's horizons

Kepler Project, 2006 onwards

research project to develop a platform for web dev



WSAPI

Xavante

Orbit

Sputnik

Copas

Cosmo

Coxpcall

LFS

Rings

LuaProfiler

RemDebug

LuaExpat

LuaSQL

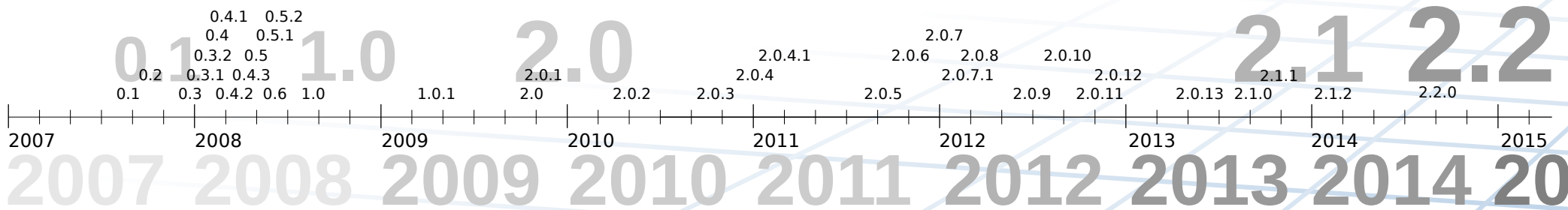
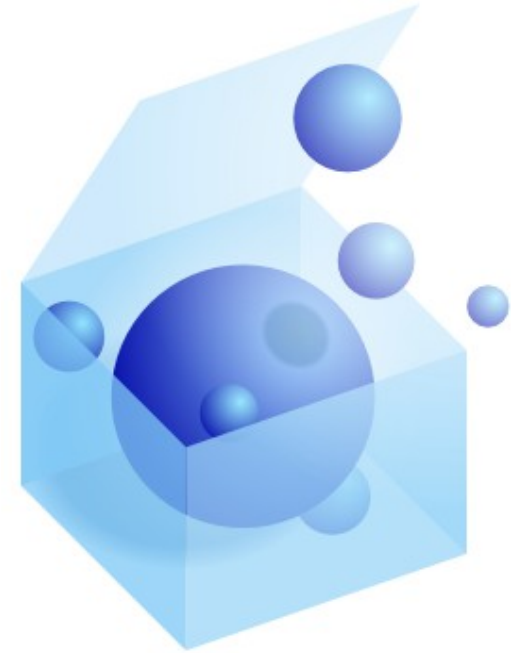
CGILua

...

# Enter LuaRocks

The missing link:  
**a package manager!**

Initial goal: build/deploy Kepler  
modules on Unix and Windows  
...and it grew from there



# Up and running in one slide!

```
~$ wget http://lua.org/ftp/lua-5.3.0.tar.gz
~$ tar xzpf lua-5.3.0.tar.gz
~$ cd lua-5.3.0
~$ make linux; sudo make install; cd ..
~$ wget http://luarocks.org/releases/luarocks-2.2.0.tar.gz
~$ tar xzpf luarocks-2.2.0.tar.gz
~$ cd luarocks-2.2.0
~$ ./configure; sudo make bootstrap; cd ..
~$ sudo luarocks install luasocket
~$ lua
Lua 5.3.0 Copyright (C) 1994-2015 Lua.org, PUC-Rio
> require "socket"
```

# Rocks and rockspecs



A **rock** file contains sources or binaries...

```
luafilesystem-1.5.0-2.src.rock
```

```
lpeg-0.10-2.win32-x86.rock
```

```
lxsh-0.8.6-1.all.rock
```

...plus a **rockspec**, which is a specification file:



```
package = "midialsa"; version = "1.21-0"
source = {
  url = "http://www.pjb.com.au/comp/luamidialsa-1.21.tar.gz",
  md5 = "072844348e66c04cee42a5b489784453"
}
description = {
  summary = "Provides access to the ALSA sequencer",
  detailed = "...",
  homepage = "http://www.pjb.com.au/comp/luamidialsa.html",
  license = "MIT/X11"
}
dependencies = { "lua >= 5.1" }
external_dependencies = {
  ALSA = { header = "alsa/asoundlib.h", library = "asound" }
}
```



# Addressing the lack of standards

IT'S JUST  
A C FILE!

Build systems aplenty:

JUST TWEAK  
THE MAKEFILE!

nothing at all – cmake – autoconf – make – etc.

Solution?

Be like Lua: **adapt** to the developers' environment

# Build back-ends

```
build = {  
  type = "make",  
  variables = {  
    PREFIX = "$(PREFIX)",  
    LUALIB = "$(LUA_LIBDIR)",  
  },  
}
```

```
build = {  
  type = "builtin",  
  modules = {  
    ["midialsa"] = "midialsa.lua",  
    ["C-midialsa"] = {  
      sources = { "C-midialsa.c" },  
      libraries = { "asound" },  
      incdirs = { "$(ALSA_INCDIR)" },  
      libdirs = { "$(ALSA_LIBDIR)" },  
    },  
  },  
  copy_directories = { "doc", "test" }  
}
```

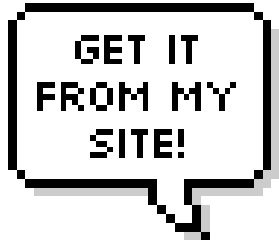
LuaRocks can use **different back-ends** for different modules:  
make, cmake, autoconf...

We also added a **built-in build tool**

**75%** of all rocks use it!

# The module repository

Moving to a "repository" model affects:



**developers**

it changes the workflow for releasing code

and

**users**

it changes how users setup their environment

our users are developers too!

# Bootstrapping a repository

## Phase 1: Curated repository

submissions through mailing list

help out early adopters

leverage LuaRocks via Kepler modules

## Phase 2: Public repository

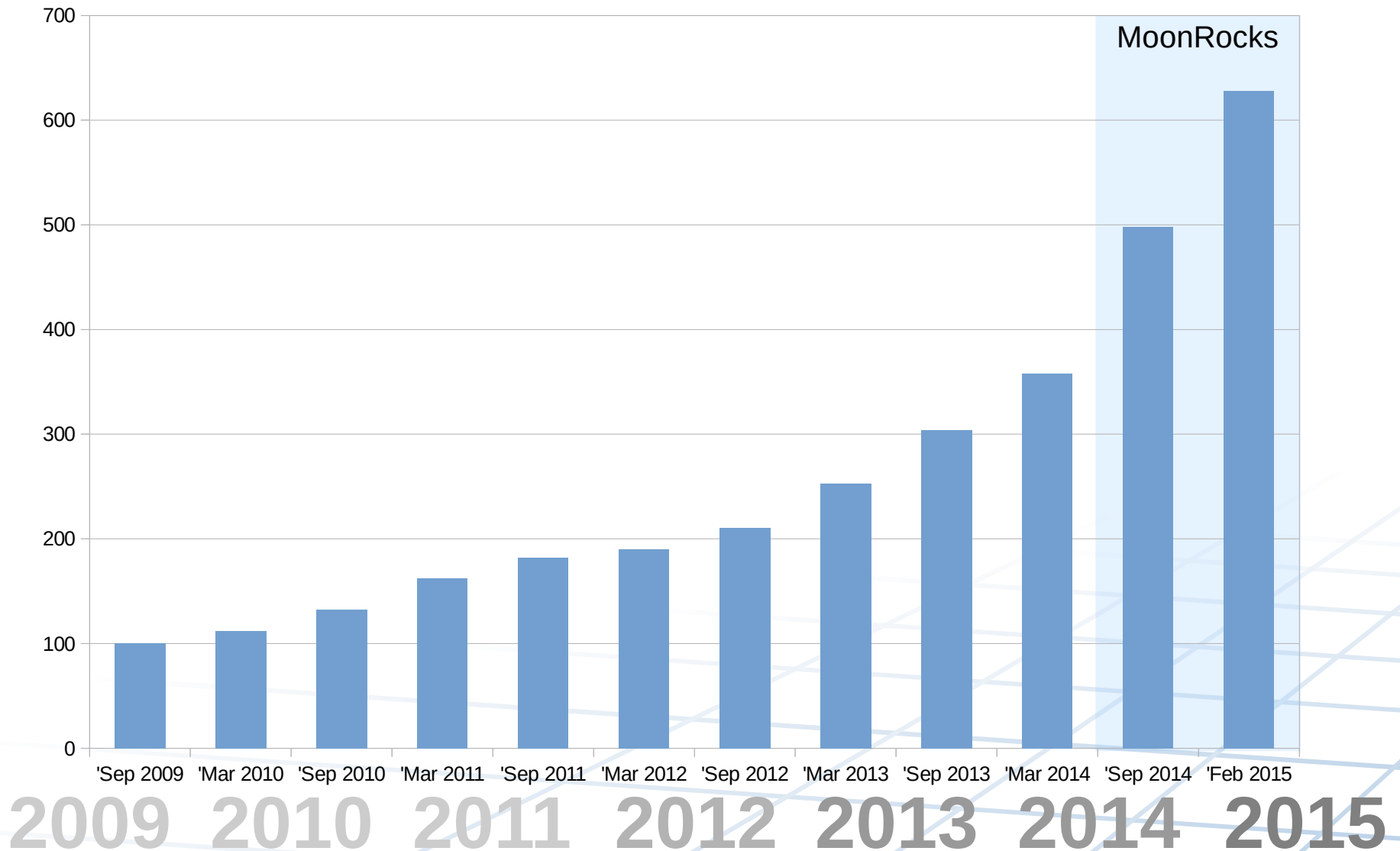
MoonRocks

<http://rocks.moonscript.org>

greater developer autonomy



# Recent growth of the repository



# A virtuous circle

CREATE YOUR OWN OBJECT SYSTEM!

...AND JSON PARSER!

...AND SERIALIZATION LIBRARY!

AND...

Breaking the DIY (NIH?) culture

Developers are also users!

Less reinvention of the wheel

- luahue
  - ↳ penlight
    - ↳ luafilesystem
  - ↳ luasocket
  - ↳ luajson
  - ↳ lunit
  - ↳ lpeg

- sailor
  - ↳ datafile
  - ↳ luafilesystem
  - ↳ valua
  - ↳ lbase64
  - ↳ cgilua
  - ↳ xavante
    - ↳ luasocket
  - ↳ copas
    - ↳ coxpcall
  - ↳ wsapi-xavante
    - ↳ wsapi
  - ↳ rings

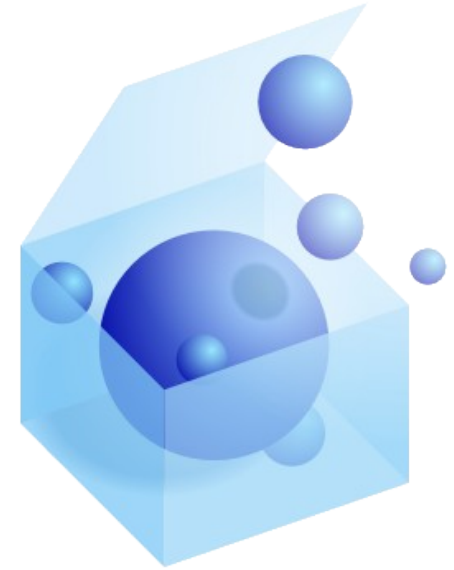
- lua-jet
  - ↳ lua-cjson
  - ↳ lua-websockets
    - ↳ luabitop
  - ↳ copas
    - ↳ coxpcall
  - ↳ luasocket
  - ↳ lua-ev
  - ↳ lpack

# Latest 10 [ANN] from lua-l

- [ANN] LuaExec
- [ANN] Compat-5.3 0.1 "luarocks install compat53"
- [ANN] lua-fcgi v1.0 "You can install it with luarocks"
- [ANN] lua-arc4random 1.0 "it's on both luarocks and Github!"
- [ANN] lua-symmetric "not comfortable to release it as 1.0 yet, hence it is not on luarocks"
- [ANN] luaposix 33.2.1 "Install it with LuaRocks"
- [ANN] StackTracePlus 0.1.2 "You can install it simply by doing: luarocks install stacktraceplus"
- [ANN] luacheck 0.8.0 "It can be installed using LuaRocks"
- [ANN] Firestr 0.5
- [ANN] lua-lluv "Install current master: luarocks install lluv..."

# What we achieved

The entry barrier for Lua as an application development language is now much lower:



LUAROCKS BUILD LPEG

a simple, popular  
build system

LUAROCKS INSTALL  
LUASOCKET

a one-stop shop  
repository

LUAROCKS SEARCH  
JSON

a culture of  
code reuse

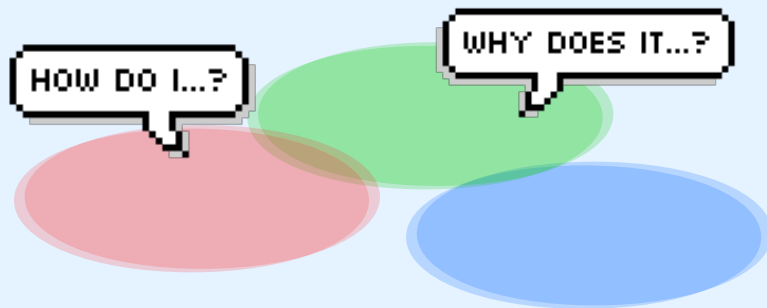
(And those who don't want it can still do things the old hard way!)



# Fostering an ecosystem: lessons learned

It takes **time**

You may need to "**bootstrap**" it to get it off the ground

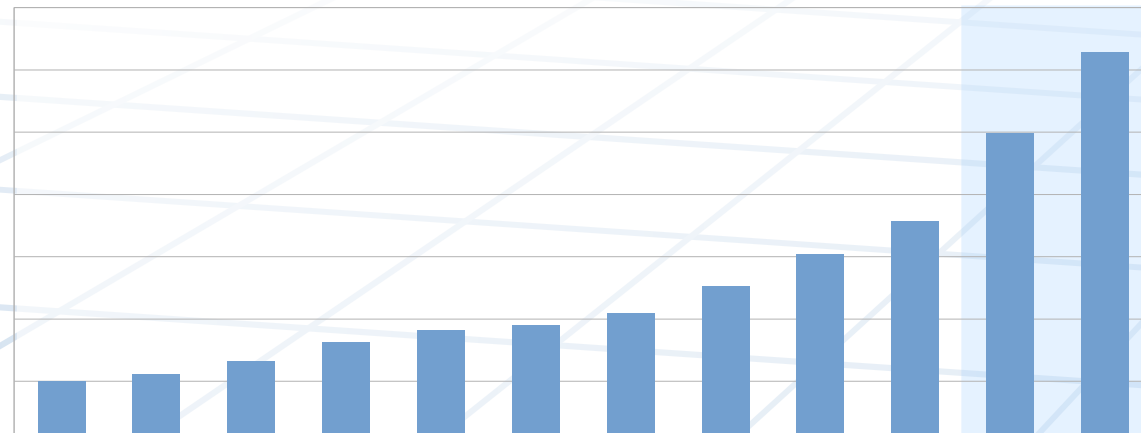


Identify your **audience**

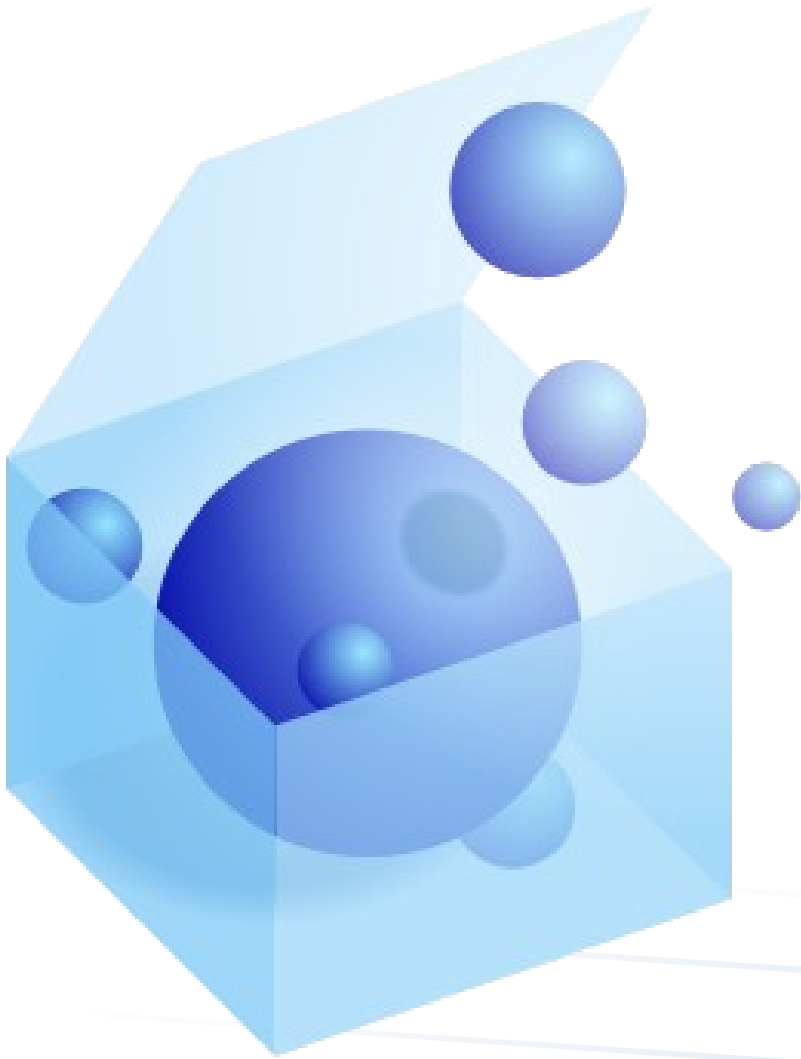
They are those who are listening to you, **listen** back!

Be bold and **share** control!

It's riskier, but it's the only way to build a **team** in FOSS



# Thank you!



<http://luarocks.org>

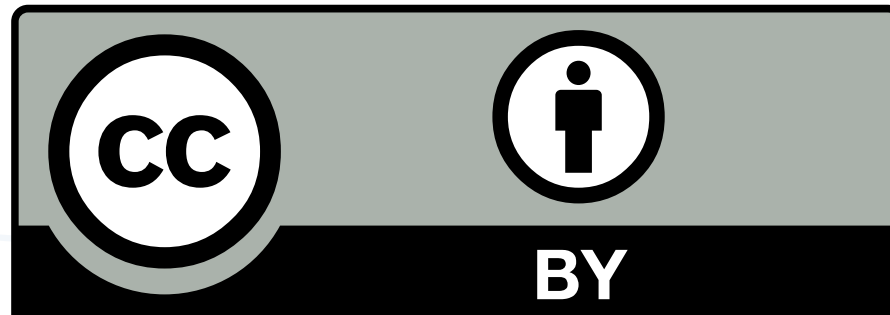
Contact:  
<http://hisham.hm/>  
h@hisham.hm  
@hisham\_hm

# About these slides

Feel free to share this presentation  
and to use parts of it in your own material

Licensed under the Creative Commons CC BY 4.0:

<https://creativecommons.org/licenses/by/4.0/>



# Bonus slide - future ideas (#luaworkshop14)

Extensible rockspecs: `luarocks.addon.youraddon`

Using: `using={"foo"}` loads `luarocks.addon.foo`

Lots of details to decide, but now LuaRocks can upgrade itself so we can evolve the `rockspec_format`

The future could look like this:

```
rockspec_format = "3.0"
using = {
  "build_dependencies", -- adds support for build-only dependencies
  "busted",             -- ensures Busted is installed, runs tests
  "ldoc",               -- generates docs using LDoc
  "build.ext >= 2.0",  -- example build type extending builtin
}
build_dependencies = { "bin2c >= 1.2" }
build = {
  type = "ext",
  modules = { ["foo"] = { language = "c99", sources = "foo.c" } },
}
doc = { --[[ ldoc specific flags ]]
```