

# A Visual Introduction To Parrot Virtual Machine

Jonathan "Duke" Leto  
Community Manager  
Parrot Virtual Machine  
<http://parrot.org>



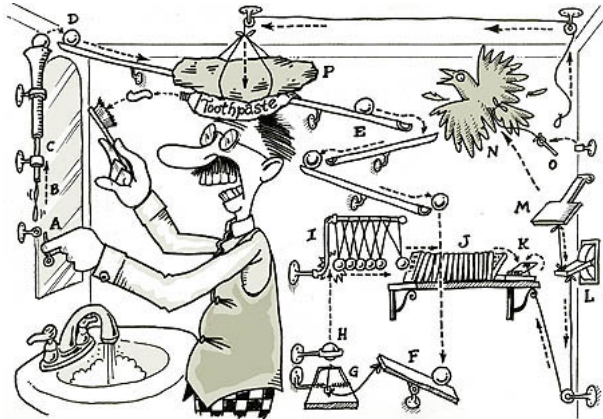
# What is Parrot, really?

Parrot is many things:

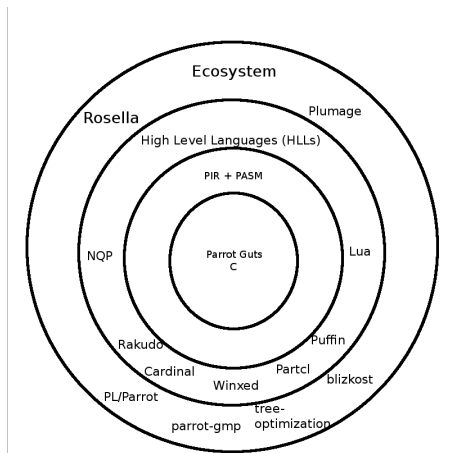
- A culture
- A collection of languages
- A virtual machine to run said languages
- A set of tools to write new languages
- A playground for compiler + language research
- A second cousin to Perl 6

Parrot is what you want it to be.

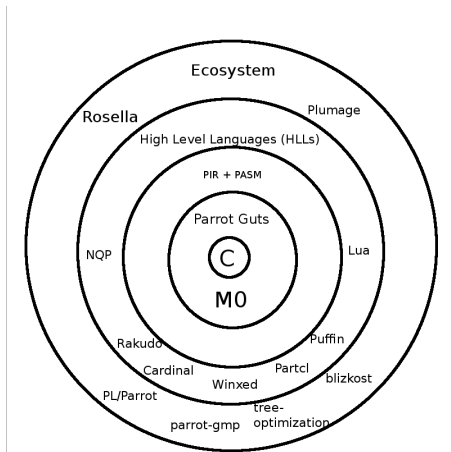
# Parrot Culture



# The Parrot Onion (NOW)



# The Future Parrot Onion



## PMC is for Cookie ...

PMC = Parrot Magic Cookie a.k.a. Objects  
Serious people prefer to call them PolyMorphic  
Containers, but not me



# PMC is for Cookie ...

Meta-recipe:

- \_ cups of \_ flour
- \_ \_ eggs
- \_ cups of \_ chocolate chips
- \_ cups of \_ sugar
- \_ sticks of \_ butter
- \_ spoons of \_ butter

# PMC is for Cookie ...

Some people like really sweet, dark chocolate chip, peanut butter cookies.

- 2 cups of white flour
- 2 organic eggs
- 1 cup of dark chocolate chips
- 3 cups of cane sugar
- 2 sticks of unsalted butter
- 3 spoons of peanut butter



## PMC is for Cookie ...

OH NOES! Some people are allergic to peanut butter, dislike processed flour and prefer milk chocolate! Also, almond butter is a delicious replacement for peanut butter.

- 2 cups of wheat flour
- 2 organic eggs
- 1 cup of milk chocolate chips
- 3 cups of cane sugar
- 2 sticks of unsalted butter
- 3 spoons of almond butter

# The meta-recipe is actually a VTABLE!

VTABLE = virtual tables

Examples of actual VTABLEs:

- `get_integer` - get integer representation of PMC
- `set_integer` - set integer representation of PMC
- `get_number` - get numeric/float representation
- `set_number` - set numeric/float representation
- `elements` - get the number of elements

Just like no recipe uses every ingredient in your fridge,  
each PMCs does not implement every VTABLE.

# Register-based

No stacks! Like Lua VM and Dalvik VM



Different set of optimizations than stack-based VMs

# Continuation Passing Style

Keep passing it along...



# Deprecation Policy

We give our APIs a bath every three months...



# Deprecation Policy

## api.yaml

```
21 -  
22   name: '":init" Sub flag'  
23   eligible: '3.4'  
24   note: 'At this point, ":init" is a no-op, and will therefore be removed.'  
25   tags:  
26     - 'PIR'  
27     - 'syntax'  
28     - 'deprecated'  
29   detection:  
30     regex:  
31       pir: '^ ".sub" .+ ":init" '  
32   ticket: 'https://trac.parrot.org/parrot/ticket/1896'  
33 -
```

tools/dev/show\_deprecated.nqp

tools/dev/resolve\_deprecated.nqp

tools/dev/show\_experimental.nqp

tools/dev/dedeprecator.nqp

# Parrot Compiler Toolkit (PCT)

We give you the dough, you cook it...



# Parrot Compiler Toolkit (PCT)

Web Interface Coming Real Soon Now

- `tools/dev/mk_language_shell.pl` - PIR build system
- `tools/dev/create_language.pl` - Perl 5 build system



# Resources

- <http://docs.parrot.org>
- [#parrot](#) on [irc.parrot.org](http://irc.parrot.org)
- [parrot-dev](#) and [parrot-users](#) mailing lists
- <https://trac.parrot.org>
- <https://github.com/parrot>
- <https://github.com/Benabik/cish>

# Thanks!



Slides at <https://github.com/leto/presentations>