



VALEO: programs robustifier

by Younès HAFRI

younes.hafri@aleph-archives.com

talk to the outside?



Who Am I ?

Aleph Archives' CTO & Founder

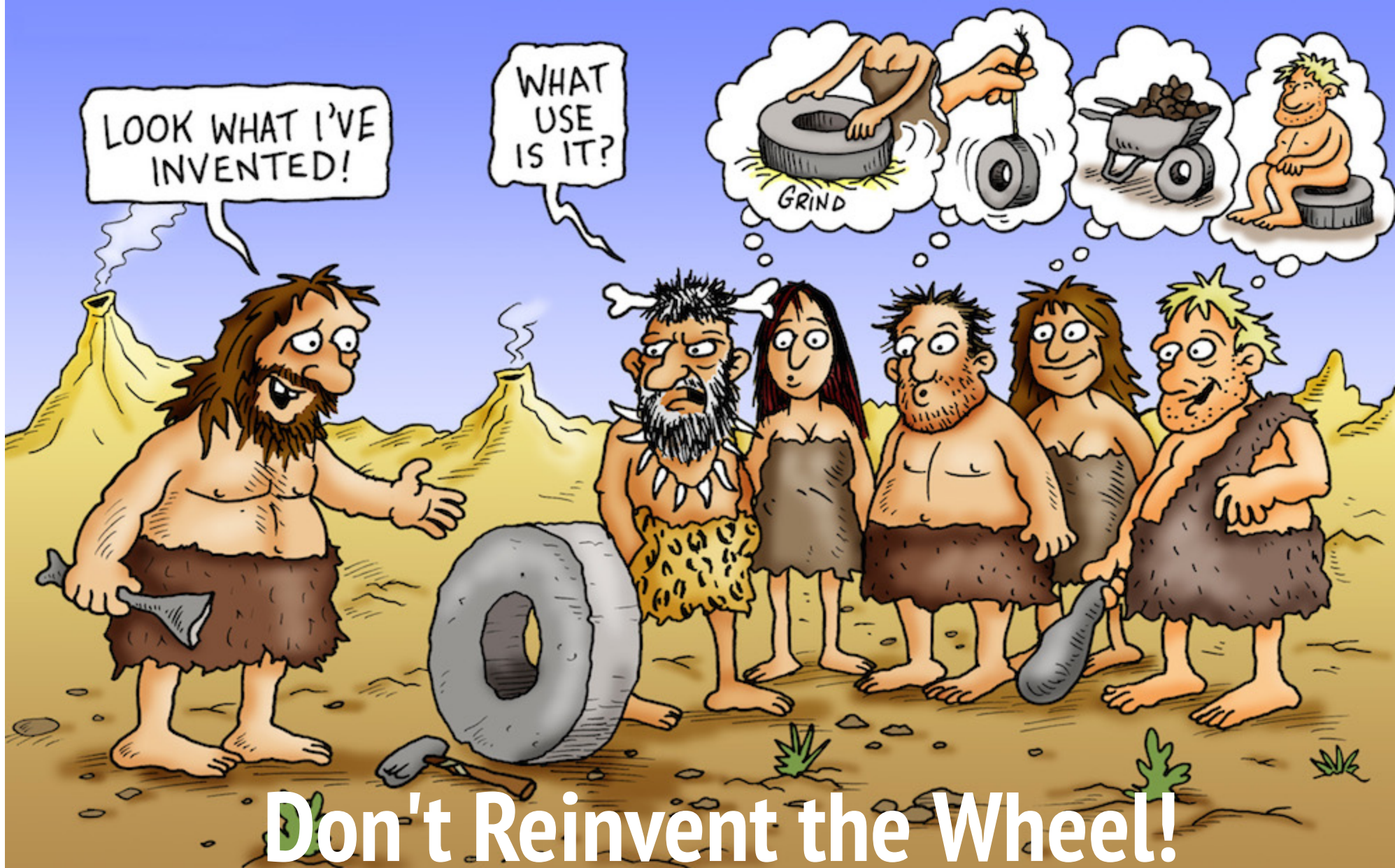
- love **coding**
- hate **complexity**
- love ***un*-optimizing** programs
- hate **cooking**
- love **data structures**

ARCHIVE MONITOR PLAYBACK



Outline

1. Erlang: patterns to talk to the outside
2. Mashup everywhere: `open_port`
3. The (Jackson) **5** use cases
4. Transition to `valeo`
5. Is it really useful ?



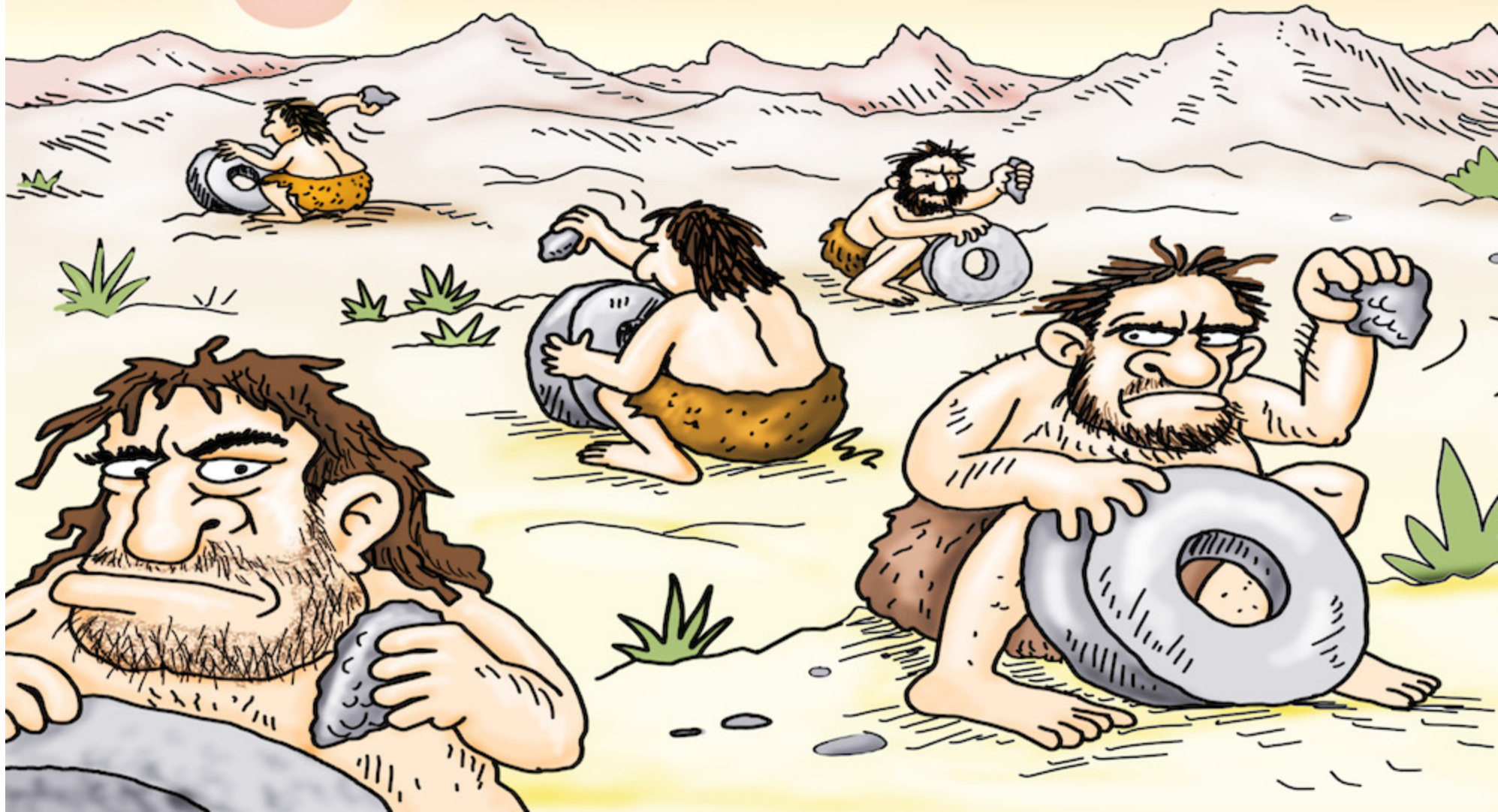
LOOK WHAT I'VE INVENTED!

WHAT USE IS IT?



Don't Reinvent the Wheel!

Should You?



Unless You Plan on Learning More About Wheels



VALEO ?

VALEO is an Erlang **behaviour** that allows you **monitor** and **control** external programs on Windows and UNIX-like operating systems [..]

👉 Erlang controls a Command



≡ Control Pattern (skeleton)

```
1 -module(skel_ctrl).
2
3 -compile(export_all).
4
5 start() ->
6     %% magic files: erts/emulator/beam/{erl_bif_port,io}.c
7     Port = open_port({spawn, "external command/script"},
8                     [exit_status]),
9     receive
10         {Port, {exit_status, _}} ->
11             start(); %% always restart me dude
12         Msg ->
13             throw(Msg)
14     end.
```

E1. Clamav Antivirus



(2010)

Erlang Clamd Server

```
03/31/14 09:32:33 clamav.erl
```

```
1  -module(clamav).
2
3  -compile(export_all).
4
5  start() ->
6      Port = open_port({spawn, "/sbin/clamd &>/dev/null"},
7                      [exit_status]),
8      receive
9          {Port, {exit_status, _}} ->
10             ok = timer:sleep(3000),
11             start(); %% always restart
12     Msg ->
13         throw(Msg)
14     end.
```

From Eicar to Stuxnet

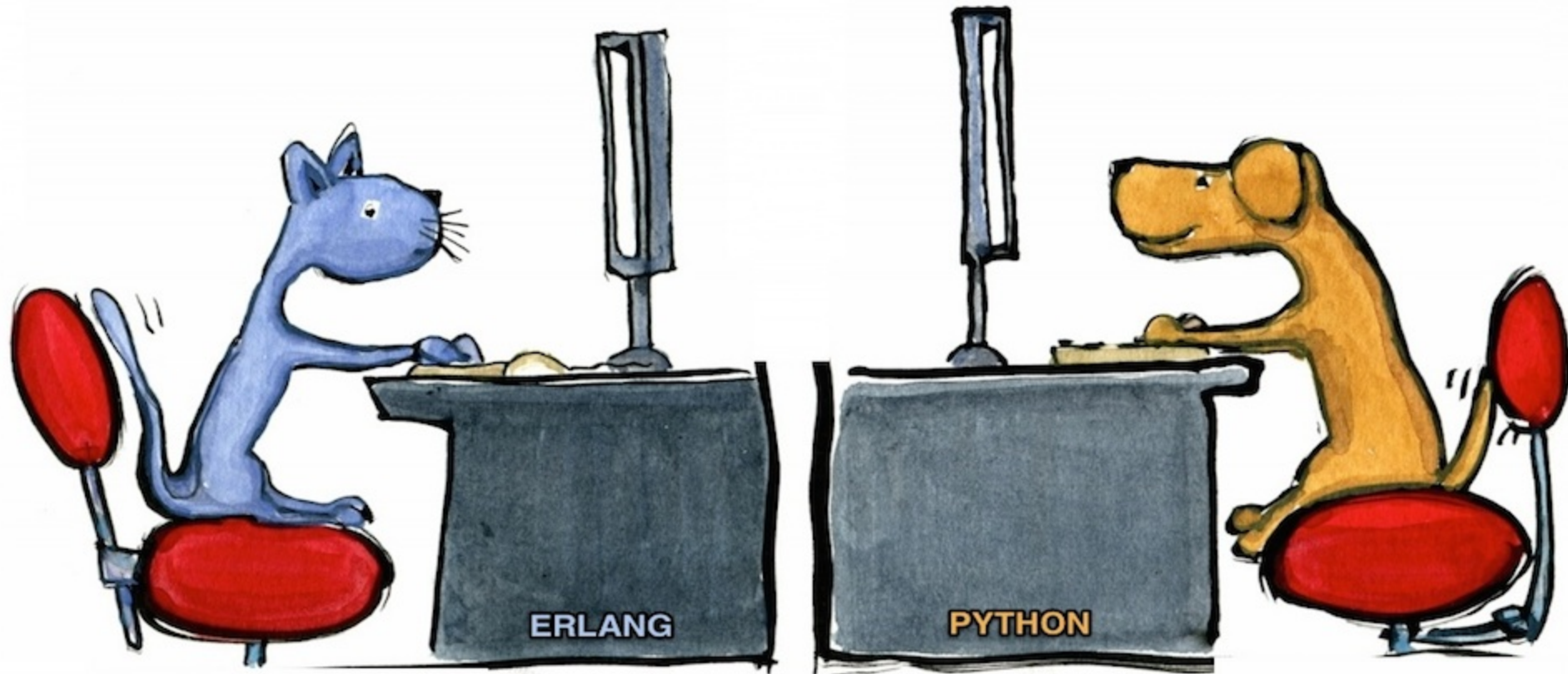
```
%% Project "Erlang-Clamd":  
%% https://github.com/athoune/Erlang-Clamd  
  
C:\> erl -pa ebin -pa deps/*/ebin  
1> ok = application:start(clamd).  
  
%% ensure "Clamd" server is up & running  
2> {ok, _} = clamd:ping().  
  
%% check Stuxnet from file  
3> {ok, virus, _} =  
    clamd:stream(clamd:file_wrapper("MRxNet.exe")).  
  
%% check EICAR from string (stream interface)  
4> clamd:transaction(fun(Worker) ->  
    clamd:start_stream(Worker),  
    clamd:chunk_stream(Worker, "X50!P%@AP[4\\PZX54(P^)7CC)7}"),  
    clamd:chunk_stream(Worker, "$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!
```

~280m Checks / Day / Host

(Web Archiving)



? Erlang **talks** to a Program



≡ Communication Pattern (skeleton)

```
1 -module(skel).
2 -compile(export_all).
3
4 start() ->
5     open_port({spawn, "external command/script"},
6               [{packet,4}, binary, exit_status]).
7
8 run(Port, Input) ->
9     true = port_command(Port, Input),
10    receive
11        {Port, {data, Reply}} ->
12            Reply;
13        Err ->
14            throw(Err)
15    end.
```

erlang side

```
1 ## load dependencies
2 import module
3 ...
4
5 object run ( input ) {
6     return module.run( input )
7 }
8
9 while(1) {
10     ## read 4-bytes length header (network byte-order)
11     bytes = read(0, 4)
12     sz = unpack('network byte-order', bytes)
13
14     ## read the data payload
15     input = read(0, sz)
16
17     ## process the data
18     reply = run(input);
19
20     ## encode the reply before sending it back
21     write(1, pack('network byte-order', len(reply)))
22     write(1, reply) ## non-blocking write calls !!!
23 }
```

external program side

Communication Pattern (erlang skeleton)

```
1 -module(skel).
2 -compile(export_all).
3
4 start() ->
5     open_port({spawn, "external command/script"},
6               [{packet, 4}, binary, exit_status]).
7
8 run(Port, Input) ->
9     true = port_command(Port, Input),
10    receive
11        {Port, {data, Reply}} ->
12            Reply;
13        Err ->
14            throw(Err)
15    end.
```


E2. Markdown Format

```
## Sweet Screenshot Markdown

*Markdown Here* is a Google Chrome, Firefox, and Thunderbird extension that lets you write Markdown in the browser.

* Markdown is cool. Put code inline.
* Link to stuff, like [this project](https://github.com/markdown-here/markdown-here)
* Ordered, you say? And a sub-list?
  1. Done.
  2. And done.

Actual review quote:
> Does exactly what I need it to do. Markdown support in email without getting in the way.

### Syntax highlighting: super rad

Some JavaScript:


```
javascript
alert('Hello Markdown');

```



How about an unusual example -- a diff:


```
diff --git a/src/common/markdown-here.js b/src/common/markdown-here.js
index 24dcb97..20c36a1 100644
--- a/src/common/markdown-here.js
+++ b/src/common/markdown-here.js
@@ -201,12 +179,14 @@ function makeStylesExplicit(wrapperElem, css) {
- applyStyleToElement(rule.style, selectorMatches[j]);
+ styleAttr = selectorMatches[j].getAttribute('style') || '';
+ styleAttr += rule.style.cssText;
+ selectorMatches[j].setAttribute('style', styleAttr);

```


```

Sweet Screenshot Markdown

Markdown Here is a Google Chrome, Firefox, and Thunderbird extension that lets you write Markdown in the browser.

- Markdown is **cool**. Put `code inline`.
- Link to stuff, like [this project](#).
- Ordered, you say? **And** a sub-list?
 1. Done.
 2. And done.

Actual review quote:

> Does exactly what I need it to do. Markdown support in email without getting in the way.

Syntax highlighting: super rad


Some JavaScript

```
alert('Hello Markdown');
```

How about an unusual example -- a diff:

```
diff --git a/src/common/markdown-here.js b/src/common/markdown-here.js
index 24dcb97..20c36a1 100644
--- a/src/common/markdown-here.js
+++ b/src/common/markdown-here.js
@@ -201,12 +179,14 @@ function makeStylesExplicit(wrapperElem, css) {
-   applyStyleToElement(rule.style, selectorMatches[j]);
+   styleAttr = selectorMatches[j].getAttribute('style') || '';
+   styleAttr += rule.style.cssText;
+   selectorMatches[j].setAttribute('style', styleAttr);

```



(2004)

Erlang Markdown

03/31/14 09:59:21 md.erl

```
1 -module(md).
2 -compile(export_all).
3
4 start() ->
5     open_port({spawn, "perl ./md.pl"},
6               [{packet, 4}, binary, exit_status]).
7
8 run(Port, Markdown) ->
9     true = port_command(Port, Markdown),
10    receive
11        {Port, {data, Reply}} ->
12            io:format("~s~n", [Reply]);
13        Err ->
14            throw(Err)
15    end.
```

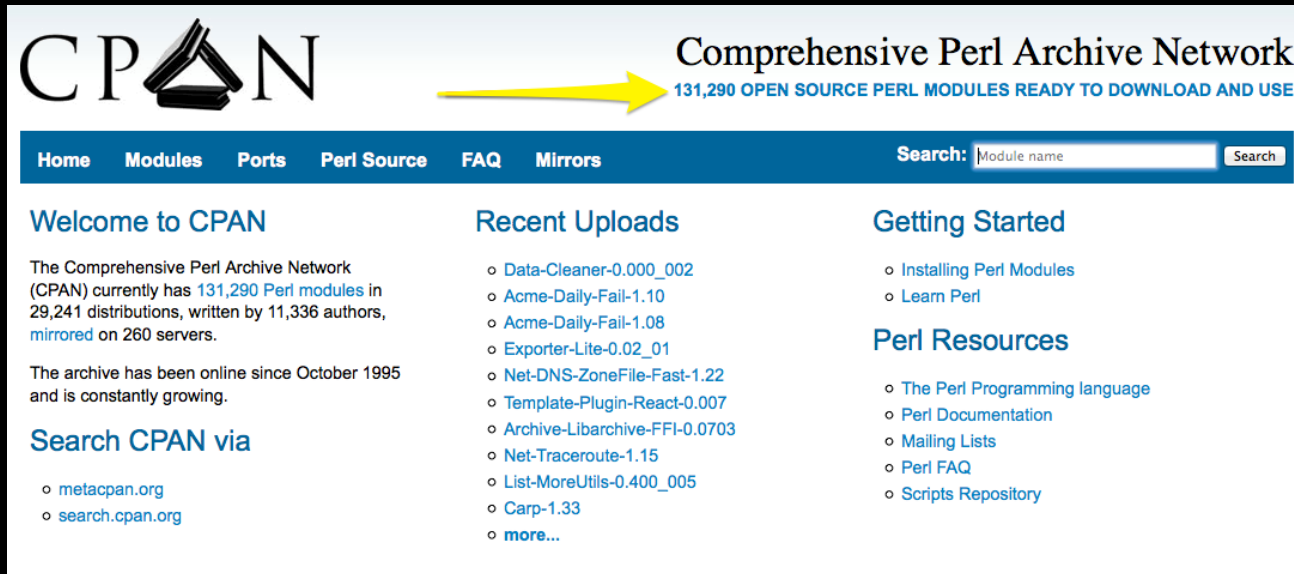
Perl Markdown Parser

```
04/02/14 08:33:08 md.pl
```

```
1  #!/usr/bin/env perl -w
2
3  use Text::Markdown::Discount;
4
5  sub run {
6      return Text::Markdown::Discount::markdown( $_[0] );
7  }
8
9  # do not buffer STDOUT (auto-flushing)
10 my $stdout = select (STDOUT);
11 $| = 1;
12 select ($stdout);
13
14 until(eof(STDIN)) {
15     my ($szBytes, $len, $data, $reply);
16
```

~130,000 Perl Modules

(march 2014)



CPAN Comprehensive Perl Archive Network
131,290 OPEN SOURCE PERL MODULES READY TO DOWNLOAD AND USE

Home Modules Ports Perl Source FAQ Mirrors Search: Search

Welcome to CPAN

The Comprehensive Perl Archive Network (CPAN) currently has [131,290 Perl modules](#) in 29,241 distributions, written by 11,336 authors, [mirrored](#) on 260 servers.

The archive has been online since October 1995 and is constantly growing.

Search CPAN via

- [metacpan.org](#)
- [search.cpan.org](#)

Recent Uploads

- [Data-Cleaner-0.000_002](#)
- [Acme-Daily-Fail-1.10](#)
- [Acme-Daily-Fail-1.08](#)
- [Exporter-Lite-0.02_01](#)
- [Net-DNS-ZoneFile-Fast-1.22](#)
- [Template-Plugin-React-0.007](#)
- [Archive-Libarchive-FFI-0.0703](#)
- [Net-Traceroute-1.15](#)
- [List-MoreUtils-0.400_005](#)
- [Carp-1.33](#)
- [more...](#)

Getting Started

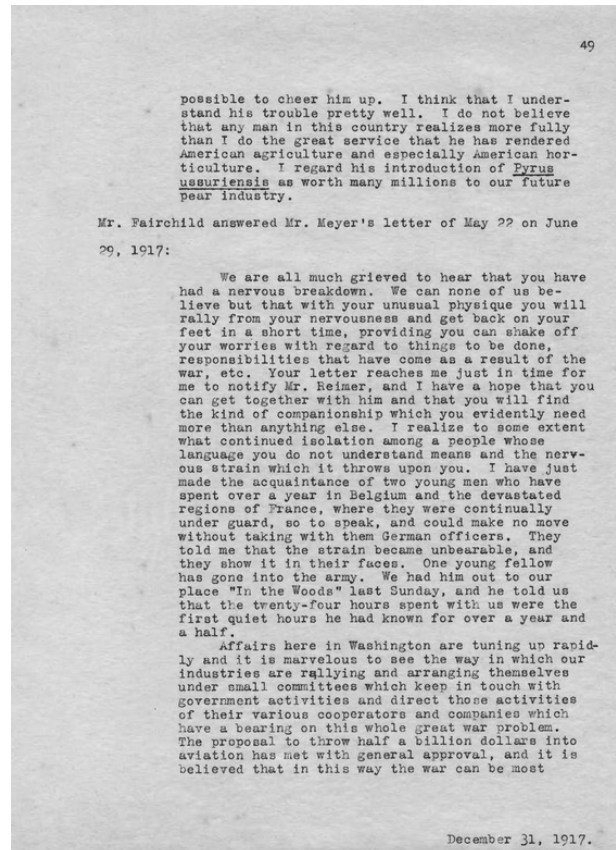
- [Installing Perl Modules](#)
- [Learn Perl](#)

Perl Resources

- [The Perl Programming language](#)
- [Perl Documentation](#)
- [Mailing Lists](#)
- [Perl FAQ](#)
- [Scripts Repository](#)

E3. Optical Character Recognition

(2012)



Erlang OCR

```
03/31/14 10:00:54 ocr.erl
```

```
1 -module(ocr).
2 -compile(export_all).
3
4 start() ->
5     open_port({spawn, "python -u ./ocr.py"},
6               [{packet,4}, binary, exit_status]).
7
8 run(Port, ImgFile) ->
9     true = port_command(Port, ImgFile),
10    receive
11        {Port, {data, Reply}} ->
12            io:format("~s~n", [Reply]);
13        Err ->
14            throw(Err)
15    end.
```

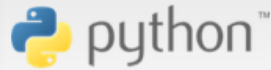
Python OCR Driver

```
03/31/14 10:04:42 ocr.py
```

```
1  #!/usr/bin/env python -u
2
3  import os
4  import sys
5  import struct
6  import Image
7
8  # https://github.com/madmaze/python-tesseract
9  from pytesseract import image_to_string
10
11 def run ( infile ):
12     return image_to_string(Image.open(infile))
13
14 while 1:
15     # 4 bytes length header (network byte-order)
16     szBytes = os.read(0, 4)
```

~40,000 Python Modules

(march 2014)



» Package Index

PyPI - the Python Package Index

The Python Package Index is a repository of software for the Python programming language. There are currently **41592** packages here. To contact the PyPI admins, please use the [Support](#) or [Bug reports](#) links.

Package documentation is now at pythonhosted.org

Package documentation is hosted on its own domain, pythonhosted.org (it was at packages.python.org and that domain will still work and automatically redirect to the new documentation home.)

E4. Search Engine - US Const.

(2006)



Erlang Search Engine

```
03/31/14 10:06:16 se.erl
```

```
1 -module(se).
2 -compile(export_all).
3
4 -define(EXT, <<".txt">>).
5 -define(DATA, <<"us_constitution">>).
6 -define(IDX, <<?DATA/binary, ".idx">>).
7
8 start() ->
9     open_port({spawn, "ruby ./se.rb"},
10             [{packet, 4}, binary, exit_status]).
11
12 index(Port) ->
13     cmd(Port, {index, ?DATA, ?IDX, ?EXT}).
14
15 search(Port, Query) when is_binary(Query) ->
16     cmd(Port, {search, ?IDX, Query}).
```

Ruby Search Engine Driver

```
04/02/14 08:40:58 se.rb
```

```
1  #!/usr/bin/env ruby
2
3  require 'rubygems'
4
5  $LOAD_PATH << '.'
6  require 'se_core.rb'
7
8  def run( cmd )
9    SE.command(cmd)
10 end
11
12 # do not buffer STDOUT (auto-flushing)
13 $STDOUT.sync = true
14
15 while 1
16   # 4 bytes length header (network byte-order)
```

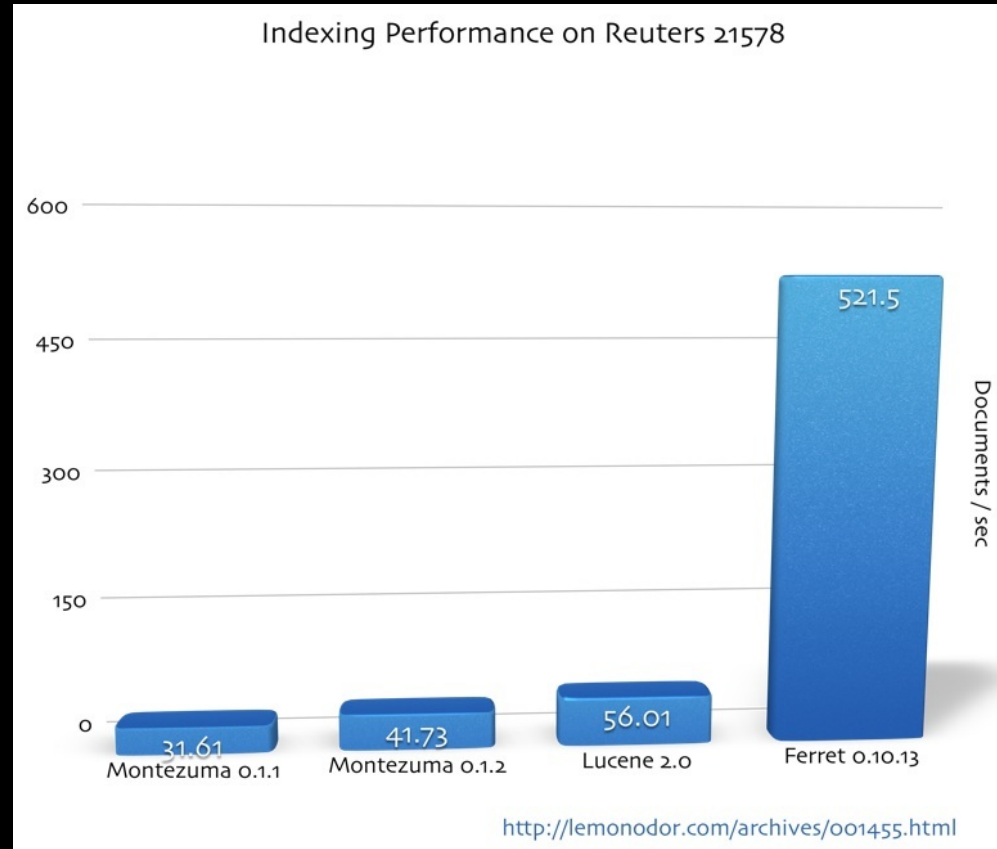
Core Search Engine

04/07/14 12:48:07 se_core.rb

```
1  #!/usr/bin/env ruby
2
3  require 'rubygems'
4  require 'fileutils'
5  require 'bert'      # http://bert-rpc.org/
6  require 'ferret'   # http://rubygems.org/gems/ferret
7
8  include Ferret
9  include Ferret::Index
10
11 module SE
12   ## command
13   def SE.command( cmd )
14     msg = BERT.decode(cmd)
15     if msg.size == 3 || msg.size == 4
16       if msg[0] == :search
```

How Fast Is It?

Montezuma (lisp) vs Lucene (java) vs Ferret (c + ruby)



~70,000 Ruby Modules

(march 2014)

RubyGems.org
your community gem host

all gems sign in sign up

Search gems...

2,858,145,311 downloads
of 72,805 gems cut since July 2009

Welcome to your community RubyGem host.
Find your gems easier, publish them faster, and have fun doing it.

LEARN

- Install RubyGems 2.2.2**
Ruby's premier packaging system
- Browse the Guides**
In depth explanations, tutorials, and references
- Gem Specification**
Your gem's interface to the world

SHARE

- `gem update --system`
Update to the latest RubyGems version
- `gem build foo.gemspec`
Build your gem
- `gem push foo-1.0.0.gem`
Deploy your gem instantly

RubyGems.org is the Ruby community's gem hosting service. **Instantly publish** your gems and install them. **Use the API** to interact and find out more information about available gems. **Become a contributor** and enhance the site with your own changes.

E5. The Unbreakable Erlang VM



Erlang VM Robustifier

```
03/31/14 10:09:54 vm.erl
```

```
1 -module(vm).
2 -compile(export_all).
3
4 -define(CMD, "erl -sname foo -setcookie secret -noshell").
5
6 start() ->
7     Port = open_port({spawn, ?CMD}, [exit_status]),
8     receive
9         {Port, {exit_status, _}} ->
10            ok = timer:sleep(3000),
11            start() %% always restart
12     end.
```

Transition: `open_port` \rightsquigarrow `valeo`

```
03/31/14 10:23:28 vm2.erl
```

```
1  -module(vm2).
2  -export([
3      start/0,
4      valeo_init/1,           %% behaviour callbacks
5      valeo_handle_info/2,
6      valeo_terminate/1
7  ]).
8
9  -behavior(valeo).
10
11 start() -> %% run an Erlang VM under VALE0 supervision
12     ProgArgs = [
13         {cmd_name, "erl"},
14         {cmd_params, ["-sname foo -setcookie secret -noshell"]},
15         {cmd_init_state, []}
16     ].
```

Valeo in Action

















```
c:\> erl
Erlang R16B03-1 (erts-5.10.4) [source] [64-bit] [smp:2:2] [async-threads:0]
Eshell V5.10.4 (abort with ^G)

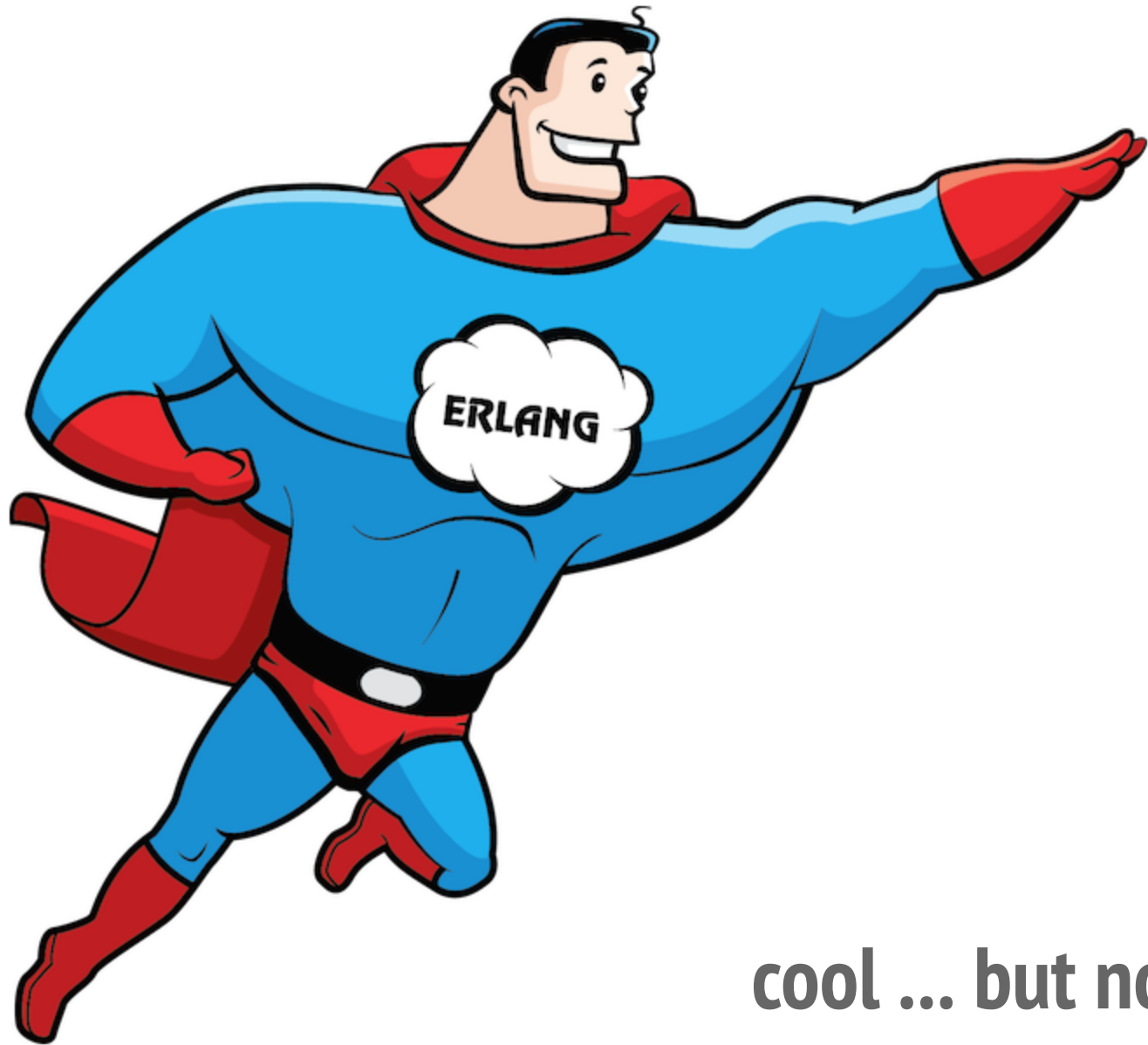
1> {ok, Port} = vm2:start().

2> valeo:proc_id(Port). %% returns the OS process ID
5056

3> valeo:stop(Port).
```

WinProc. Manager

Processus		Performance		Historique des applications		Démarrage		Utilisateurs		Détails		Services	
Nom	PID	Statut	Nom d'util...	Pro...	Mémoire (...)	Description							
 AdminService.exe	1792	En cours d'exé...	Système	00	784 Ko	AdminService Application							
 agent.exe	1836	En cours d'exé...	Système	00	3 948 Ko	Immunet Protect Agent							
 Ath_CoexAgent.exe	1236	En cours d'exé...	Système	00	1 128 Ko	Atheros Coex Service Application							
 BtTray.exe	4760	En cours d'exé...		00	7 328 Ko	BtTray							
 BtvStack.exe	4964	En cours d'exé...		00	4 640 Ko	Extension Core							
 CLMLSvc_P2G8.exe	1156	En cours d'exé...		00	352 Ko	CyberLink MediaLibray Service							
 cmd.exe	5812	En cours d'exé...		00	128 Ko	Interpréteur de commandes Windows							
 cmd.exe	5464	En cours d'exé...		00	392 Ko	Interpréteur de commandes Windows							
 conhost.exe	3748	En cours d'exé...		00	368 Ko	Hôte de la fenêtre de la console							
 conhost.exe	8040	En cours d'exé...		00	684 Ko	Hôte de la fenêtre de la console							
 csrss.exe	724	En cours d'exé...	Système	00	804 Ko	Processus d'exécution client-serveur							
 csrss.exe	816	En cours d'exé...	Système	00	1 508 Ko	Processus d'exécution client-serveur							
 dllhost.exe	2712	En cours d'exé...	SERVICE L...	00	600 Ko	COM Surrogate							
 dllhost.exe	1116	En cours d'exé...	SERVICE L...	00	1 368 Ko	COM Surrogate							
 dwm.exe	1032	En cours d'exé...	DWM-1	00	12 884 Ko	Gestionnaire de fenêtres du Bureau							
 erl.exe	5056	En cours d'exé...		00	8 376 Ko	erl							



cool ... but not yet **sex**

the more you have
the better



☯ Getting Stronger: valeo + pg2

```
03/31/14 10:24:45 md2.erl
```

```
1  -module(md2).
2
3  -export([
4      start/0,
5      run/1,
6      valeo_init/1, %% behaviour callbacks
7      valeo_handle_info/2,
8      valeo_terminate/1
9  ]).
10
11 -behavior(valeo).
12
13 -define(GROUP, 'markdown').
14 -define(PICK_ONE, pg2:get_closest_pid(?GROUP)).
15
16 start() ->
```

Getting Stronger: revisited

```
03/31/14 10:25:50 md3.erl
```

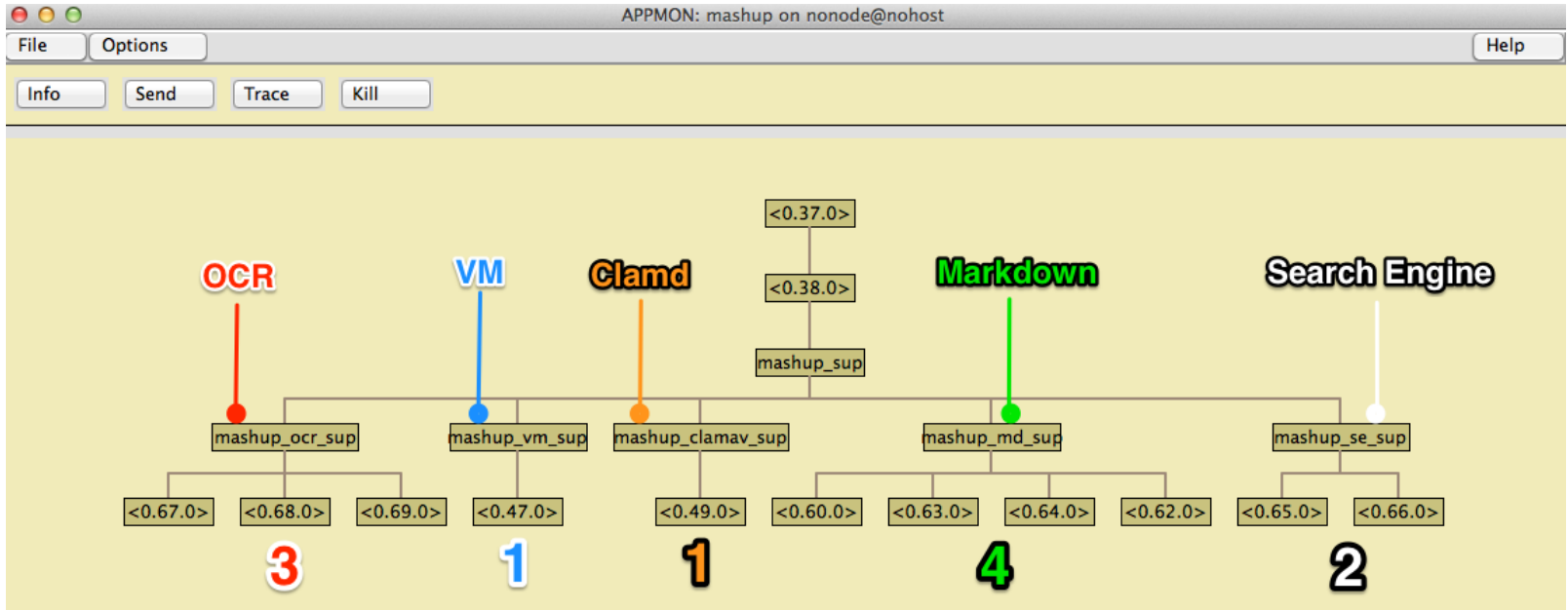
```
1  -module(md3).
2
3  -export([
4      start/0,
5      run/1,
6      valeo_init/1, %% behaviour callbacks
7      valeo_handle_info/2,
8      valeo_terminate/1
9  ]).
10
11 -behavior(valeo).
12
13 -define(GROUP, 'markdown').
14 -define(PICK_ONE, pg2:get_closest_pid(?GROUP)).
15
16 start() ->
```

Mashup Supervisor(s)

```
04/01/14 11:18:16 mashup.erl
```

```
1  -module(mashup).
2
3  -export([
4      start/0,
5      stop/0
6  ]).
7
8  %% Supervisor with restart strategy "simple_one_for_one".
9  %% Dynamically add N childs to SUP.
10 -define(RUN(N, SUP),
11     [{ok, _} = supervisor:start_child(SUP, []) ||
12      _ <- lists:seq(1, N), N >= 1]).
13
14 start() ->
15     ok = application:start(?MODULE),
16
```

Mashup Application



Comparison

	open_port	valeo
flexible	✓	✓
safe	✓	✓
exit	✗	✓
integration	~	✓

don't forget that "valeo" extends "open_port" behind the scenes.

Small is Beautiful

```
→ ls c_src/*.c
c_src/valeo.c

→ cloc c_src/valeo.c src/valeo.erl
  2 text file.
  2 unique file.
  0 files ignored.
```

<http://cloc.sourceforge.net> v 1.60

Language	files	blank	comment	code
Erlang	1	185	359	369
C	1	87	109	234
SUM:	2	272	468	603

VALEO's Key Features

1. Reliable control of **external programs** under supervision
2. React to program's events (ex. unexpected crashes)
3. Ensure program **termination**
4. No messages mess between processes
5. Simple behaviour "à la" OTP
6. Production-ready
7. Cross-platform (32/64-bit): Windows, OSX, **Linux** *

* should also work on FreeBSD, SmartOS, etc.

[webarchivingbucket.com/
valeo/mashup.zip](http://webarchivingbucket.com/valeo/mashup.zip)

QUESTIONS?

 @y_hafri

younes.hafri@aleph-archives.com

Find Executable Paths

```
%% First, check inside the ?MODULE "priv" directory.
%% If not found, try the "$PATH" environment variable.
case valedo:exec_find("rsync", ?MODULE) of
  false ->
    throw(no_rsync);
  ExecPath ->
    %% "rsync" exec path
    ExecPath
end,
...

```