START SWISH!!!!

Night before:

Set presso and swish to autostart

disable skype, workrave, etc.

Distributed SWI-Prolog Development

Anne Ogborn

SWI Prolog

Pengines

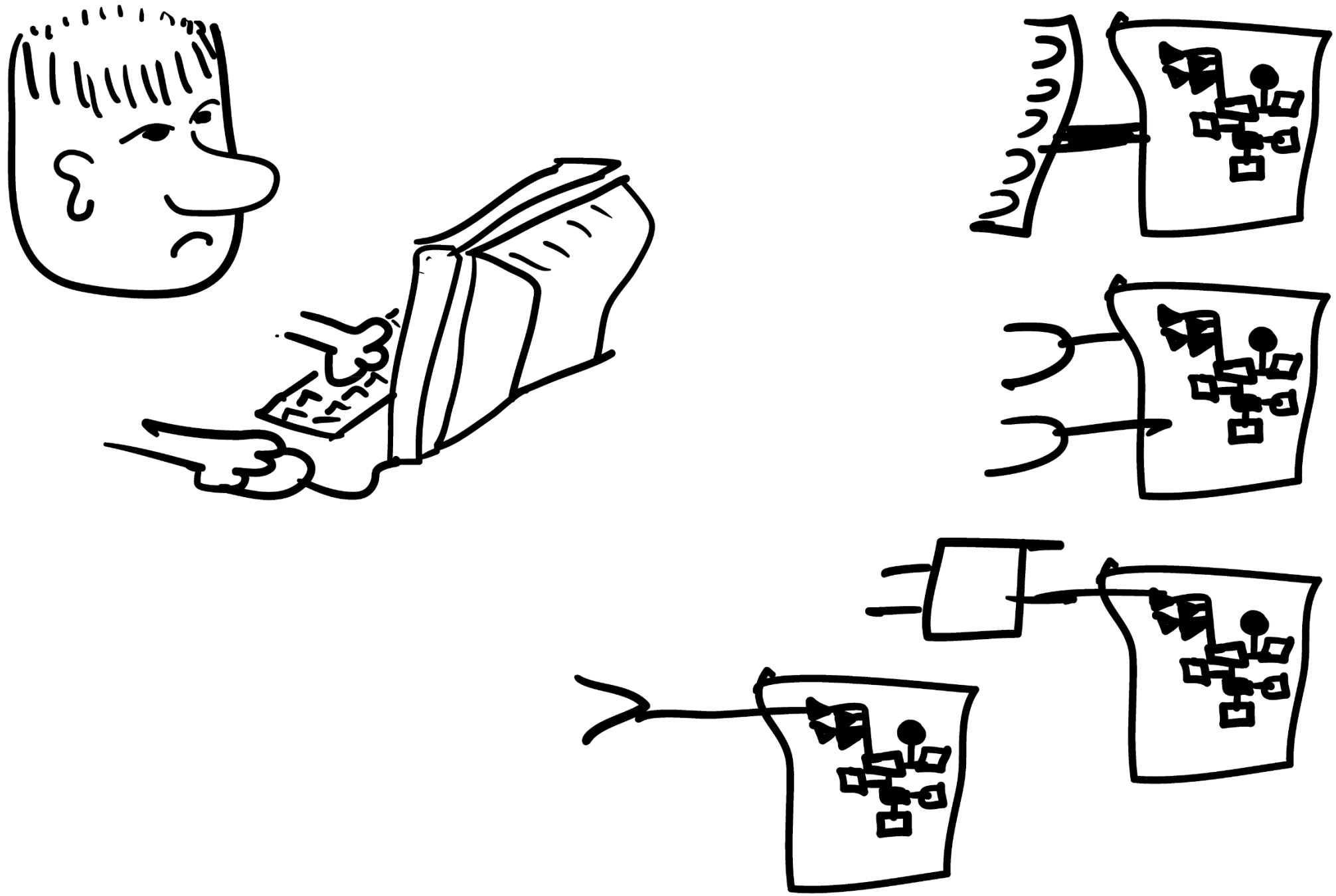
- Torbjörn Lager
- University of Gothenborg

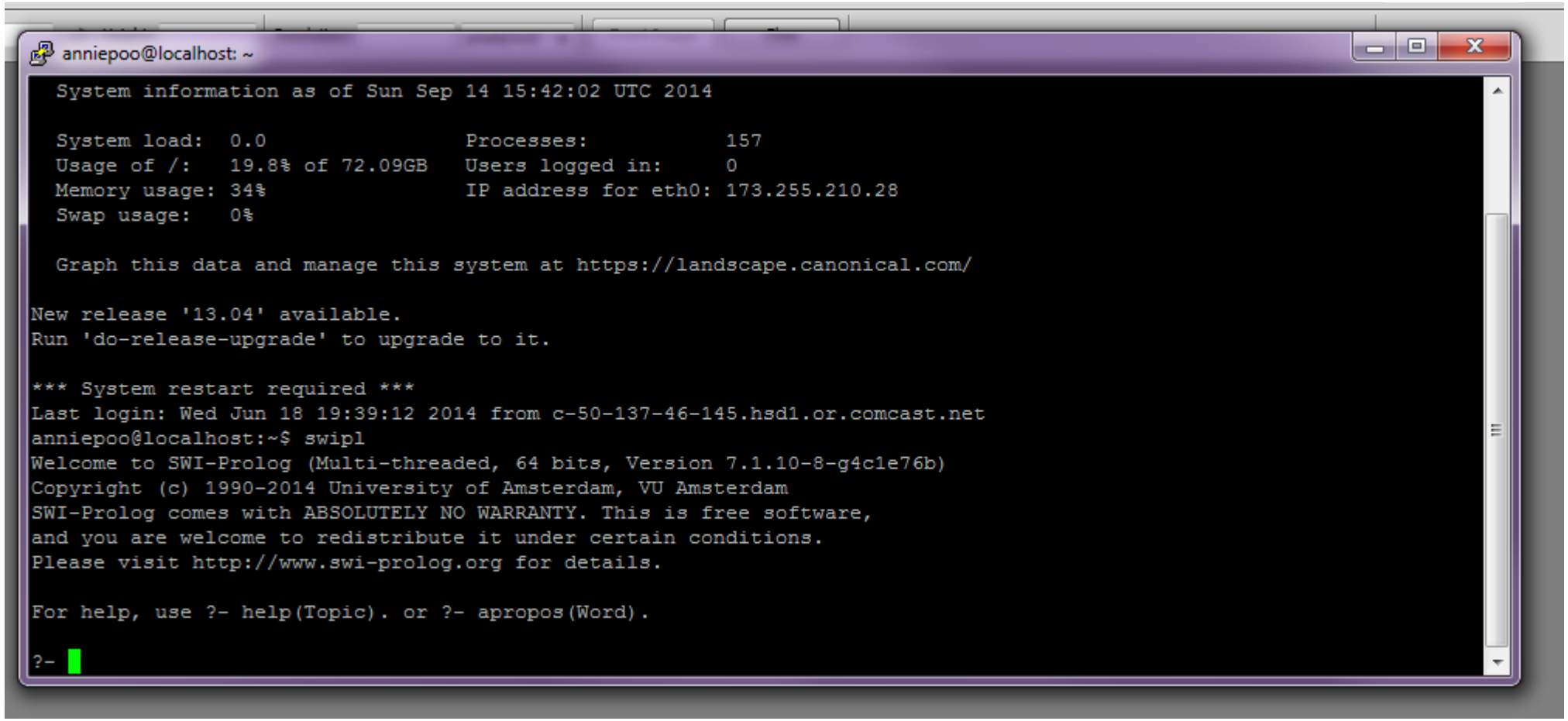
- Jan Wielemaker
- Free Univ. of the Netherlands

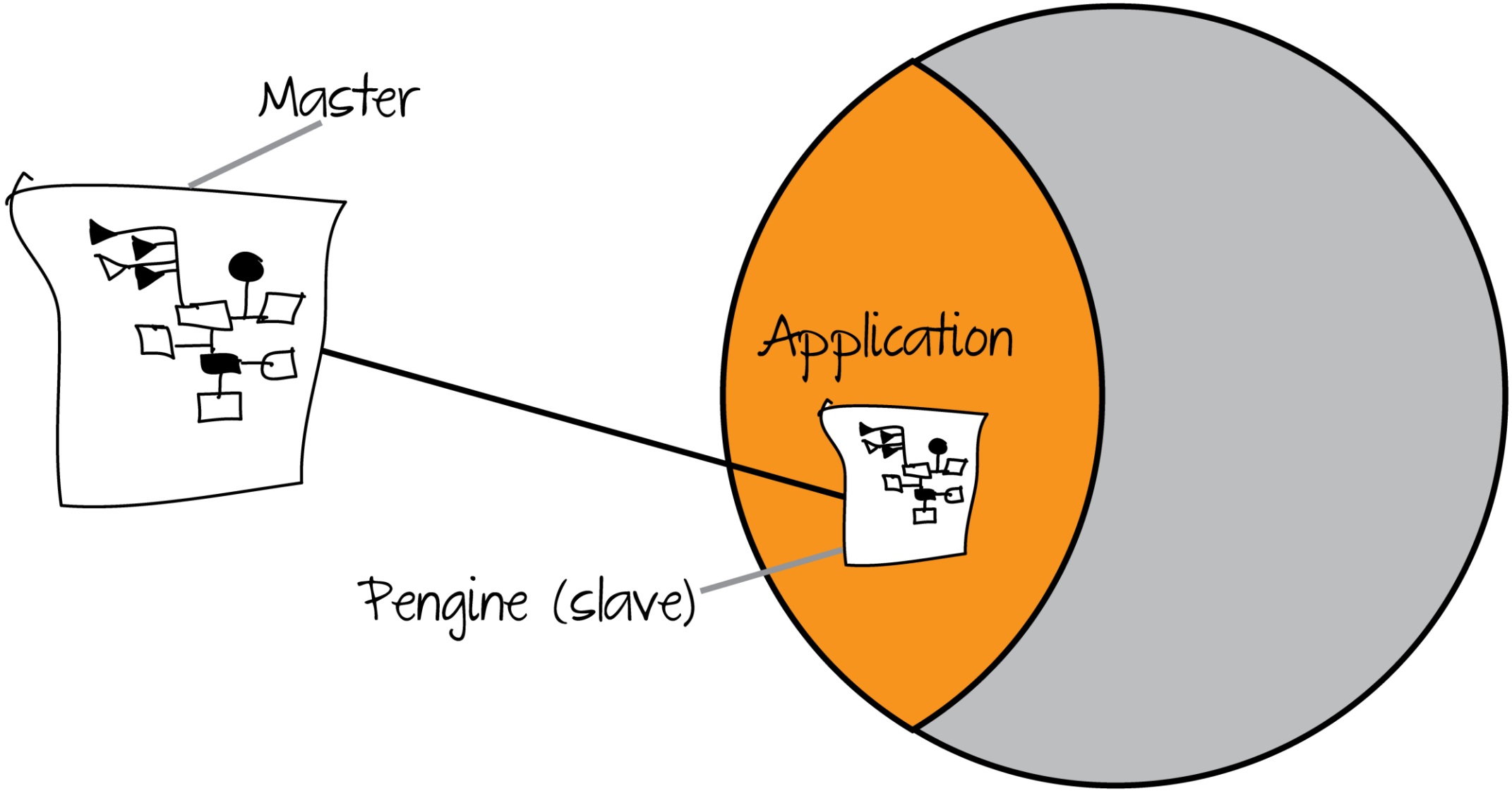
Prolog?

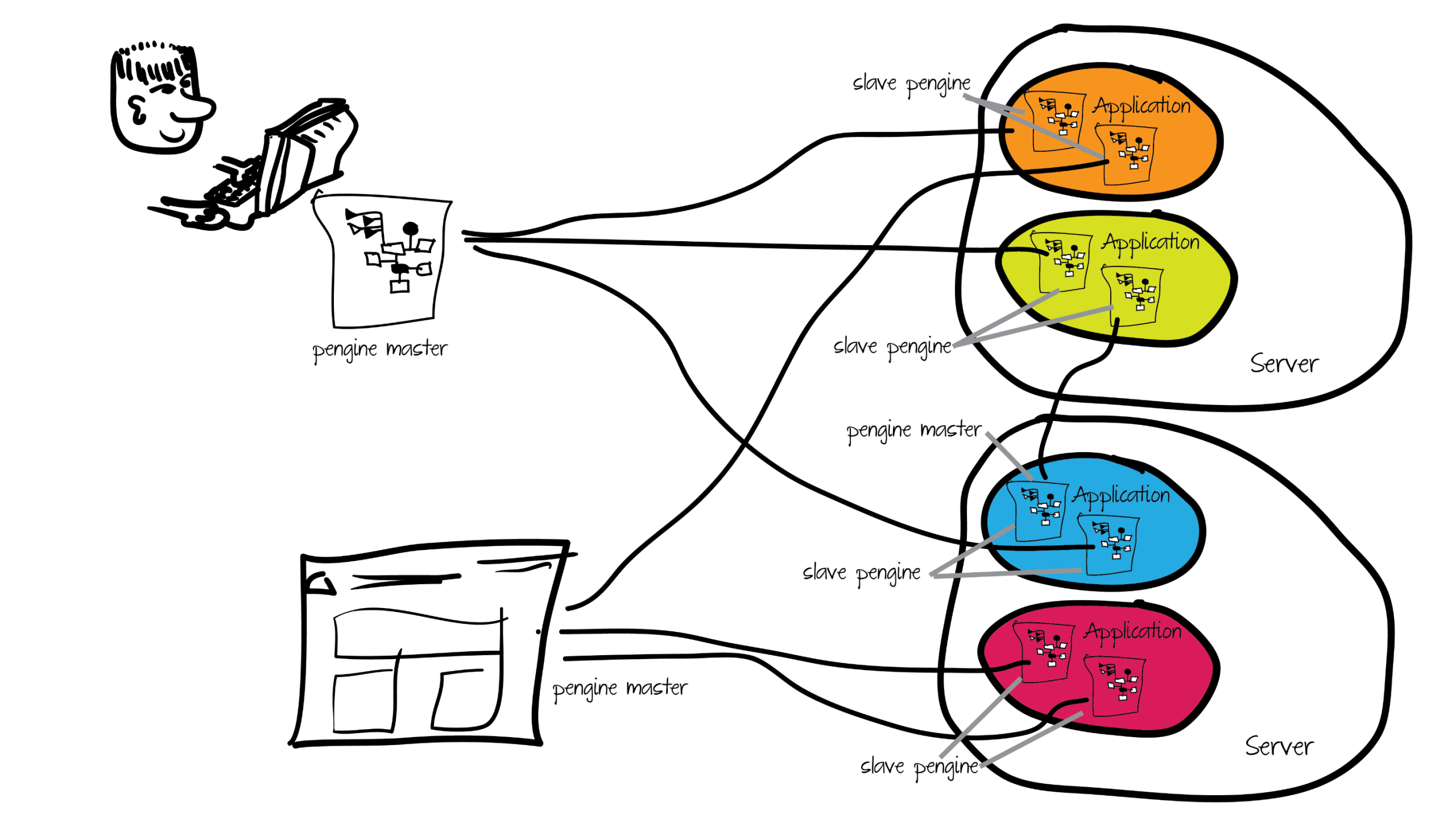
Semantic Web?

Pengine Roles

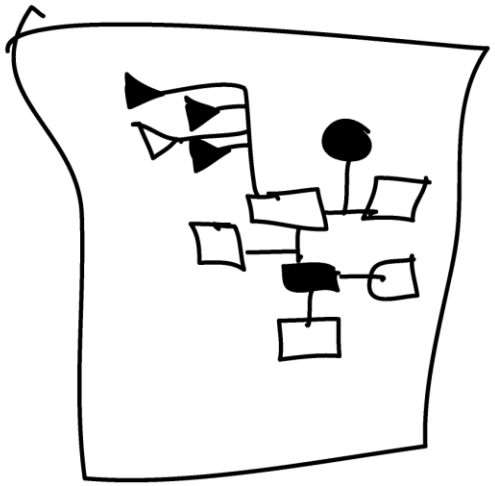



anniepoo@localhost: ~
System information as of Sun Sep 14 15:42:02 UTC 2014
System load: 0.0 Processes: 157
Usage of /: 19.8% of 72.09GB Users logged in: 0
Memory usage: 34% IP address for eth0: 173.255.210.28
Swap usage: 0%
Graph this data and manage this system at https://landscape.canonical.com/
New release '13.04' available.
Run 'do-release-upgrade' to upgrade to it.
*** System restart required ***
Last login: Wed Jun 18 19:39:12 2014 from c-50-137-46-145.hsd1.or.comcast.net
anniepoo@localhost:~$ swipl
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.1.10-8-g4c1e76b)
Copyright (c) 1990-2014 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
For help, use ?- help(Topic). or ?- apropos(Word).
?-

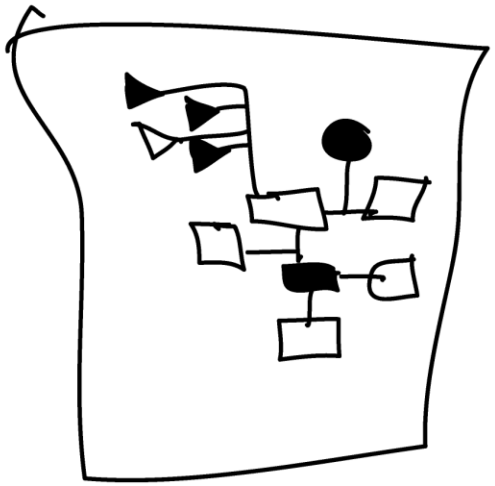

Master
Application
Pengine (slave)

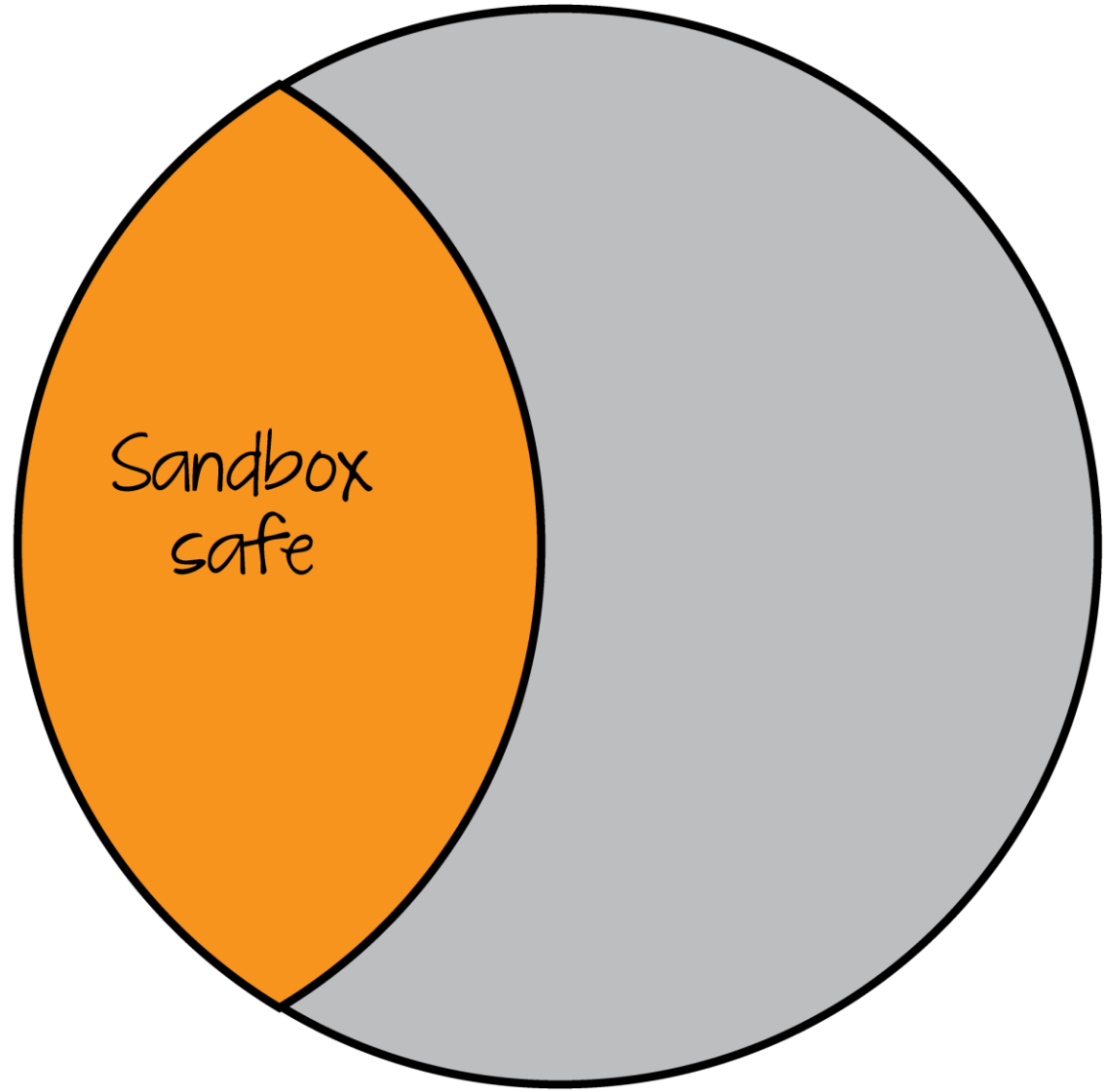

slave pengine
Application
Application
pengine master
slave pengine
Server
pengine master
Application
slave pengine
Application
pengine master
slave pengine
Server

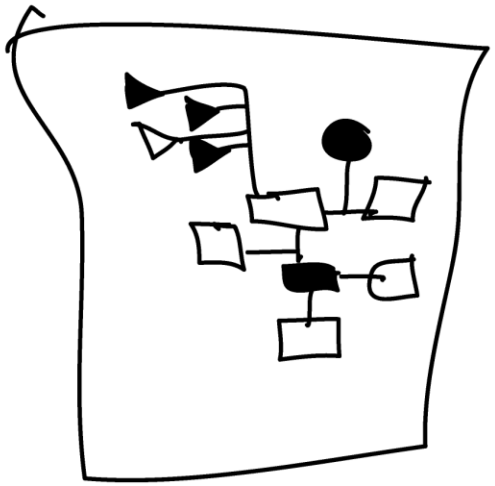
The Pengine Knowledgebase

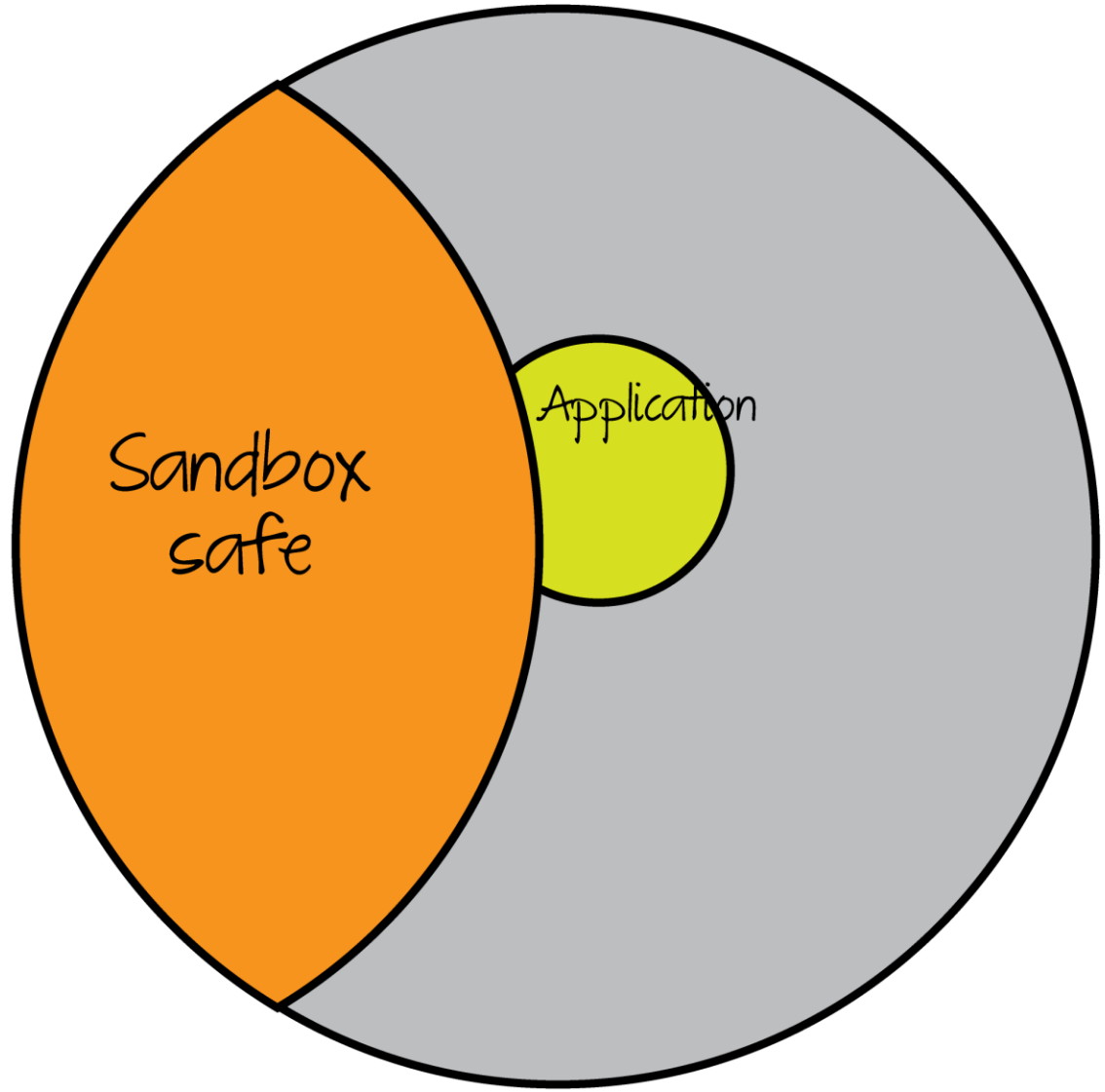


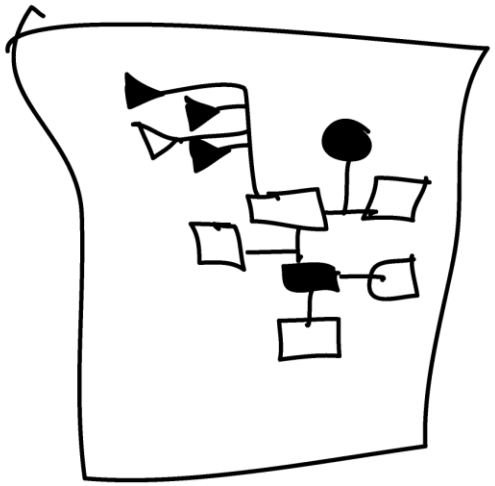
Prolog
Knowledge Base

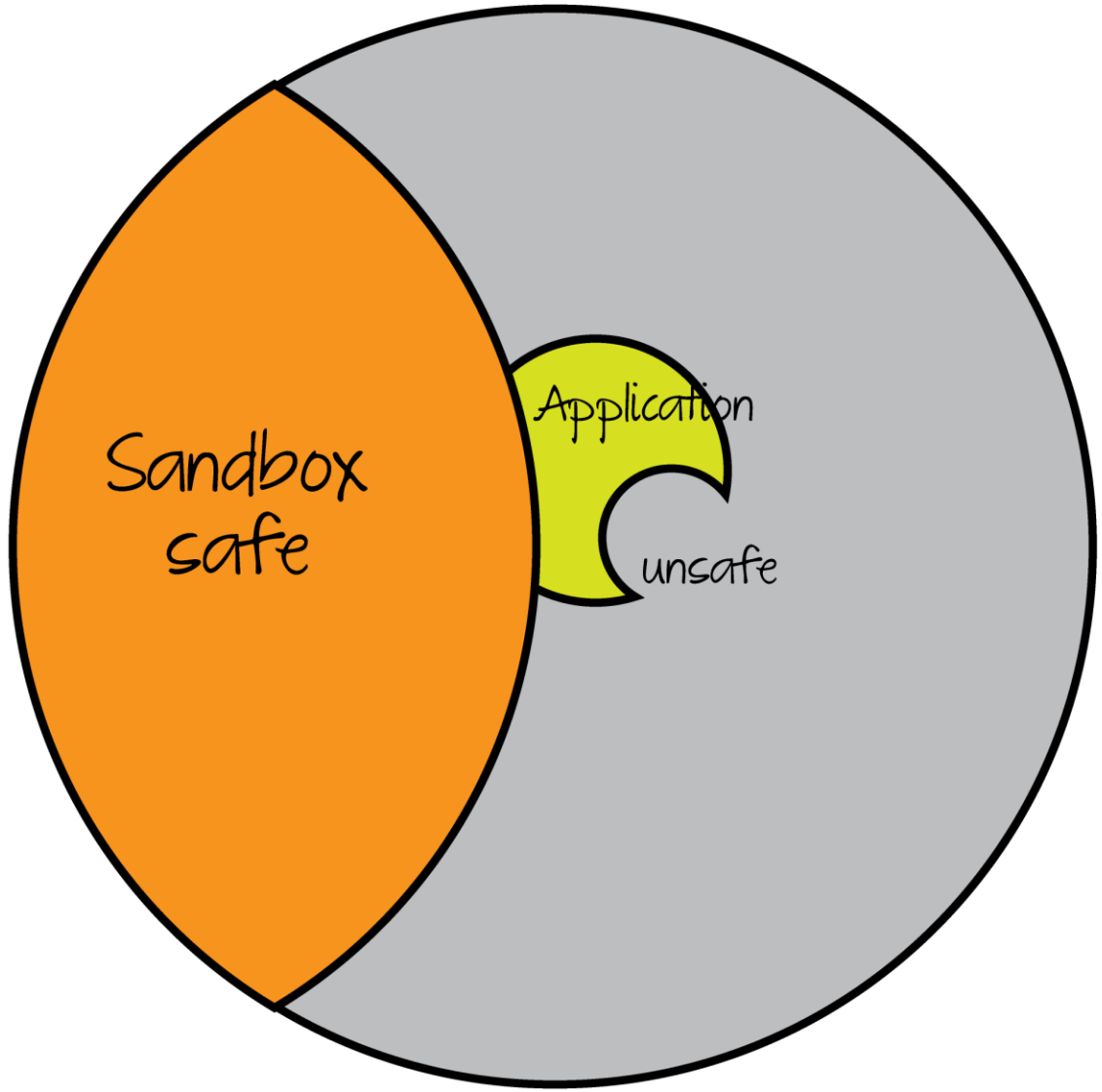


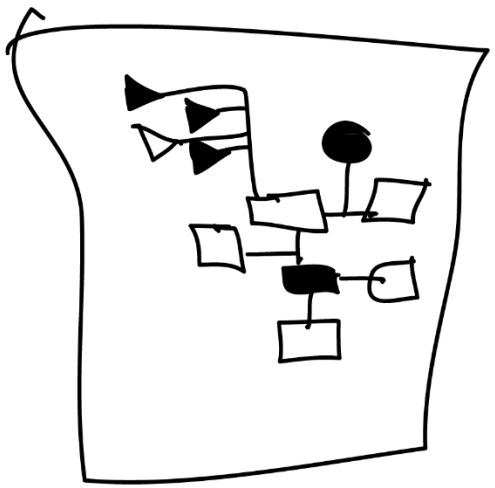

Sandbox
safe

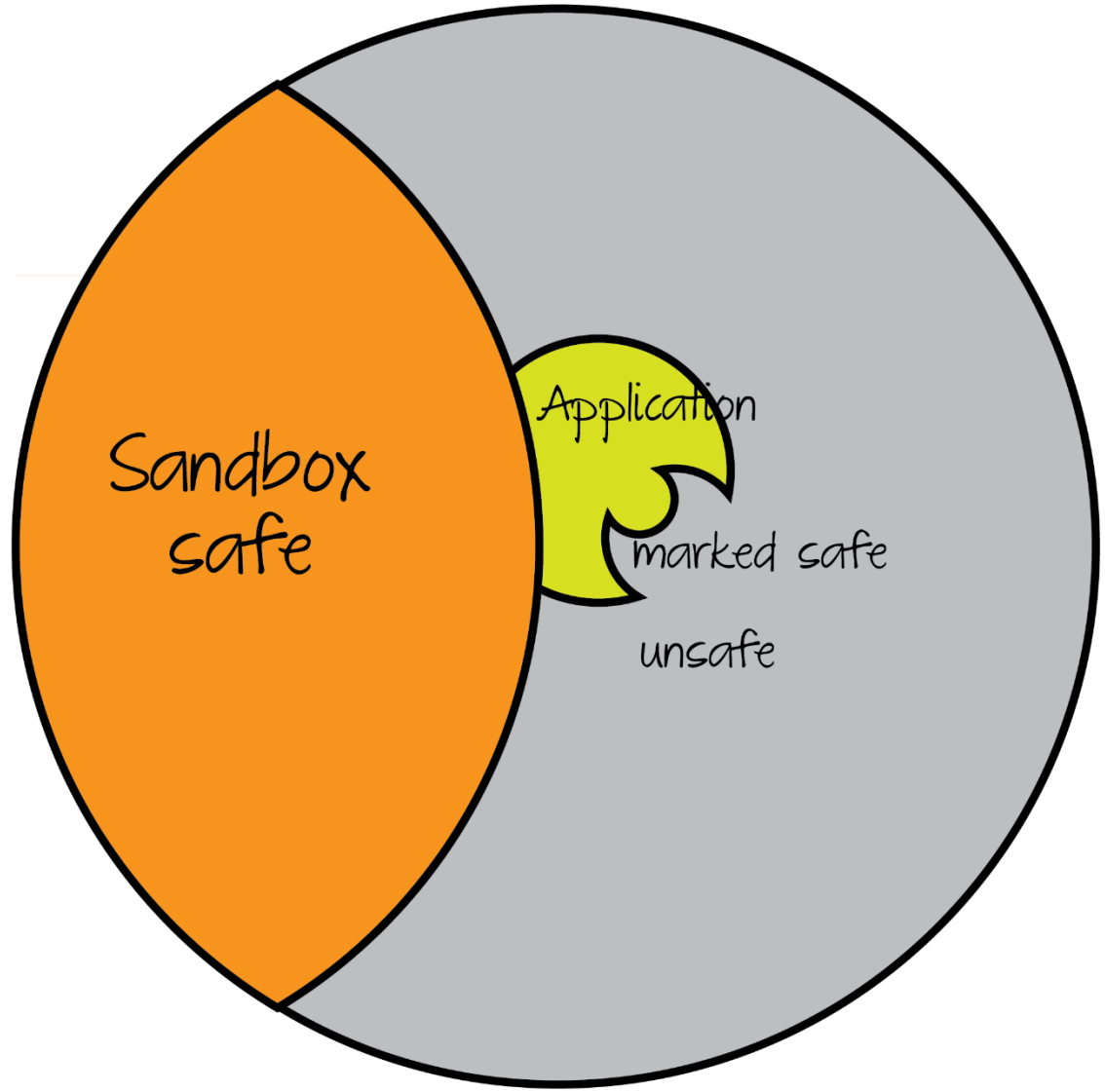


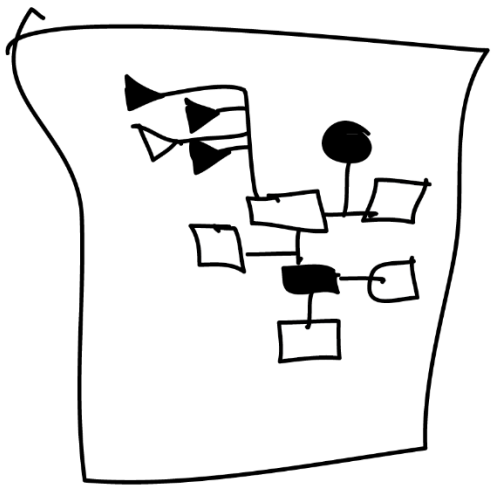

Sandbox
safe
Application

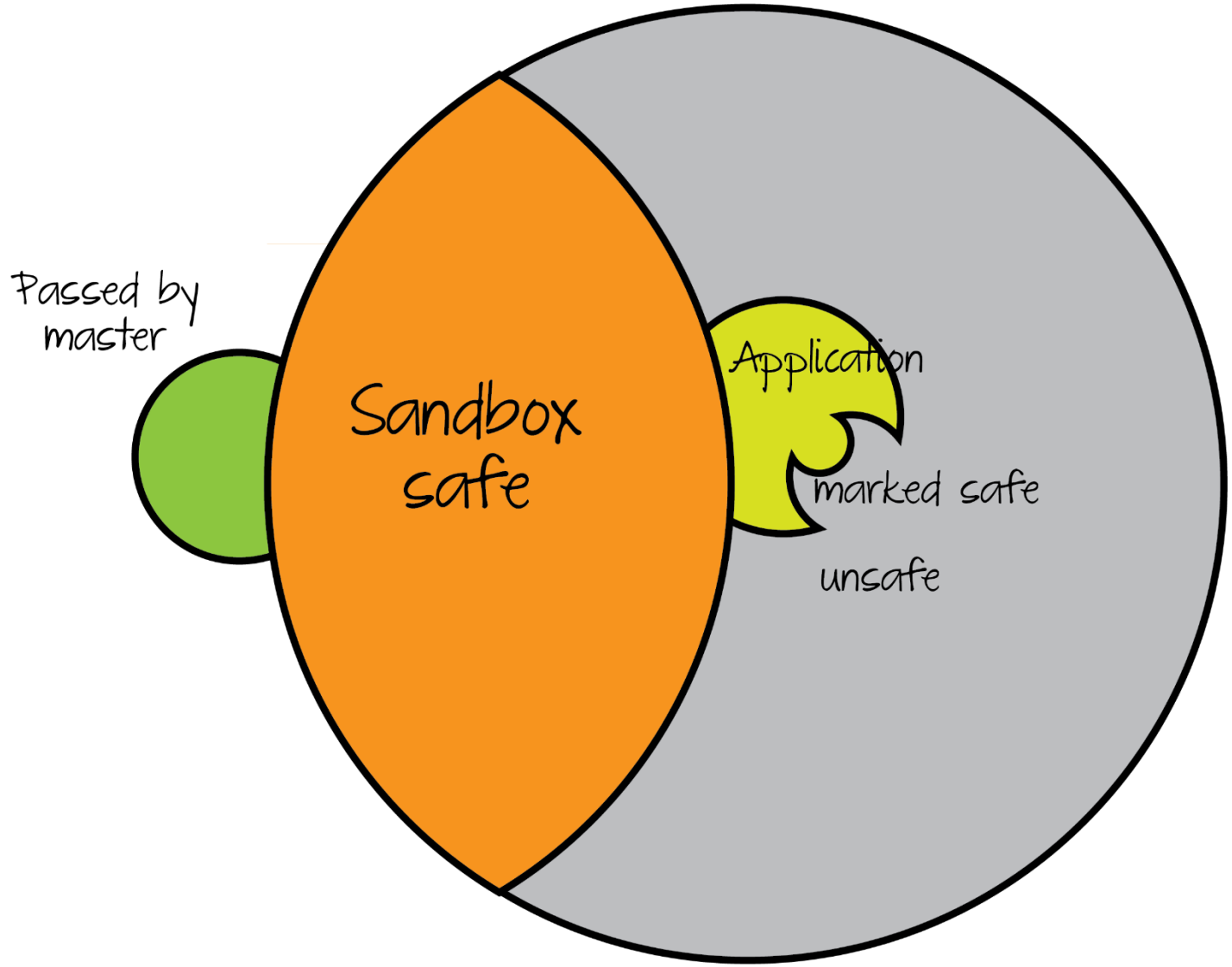


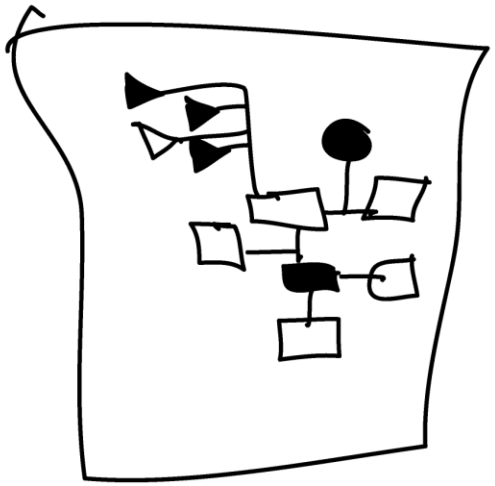

Sandbox
safe
Application
unsafe

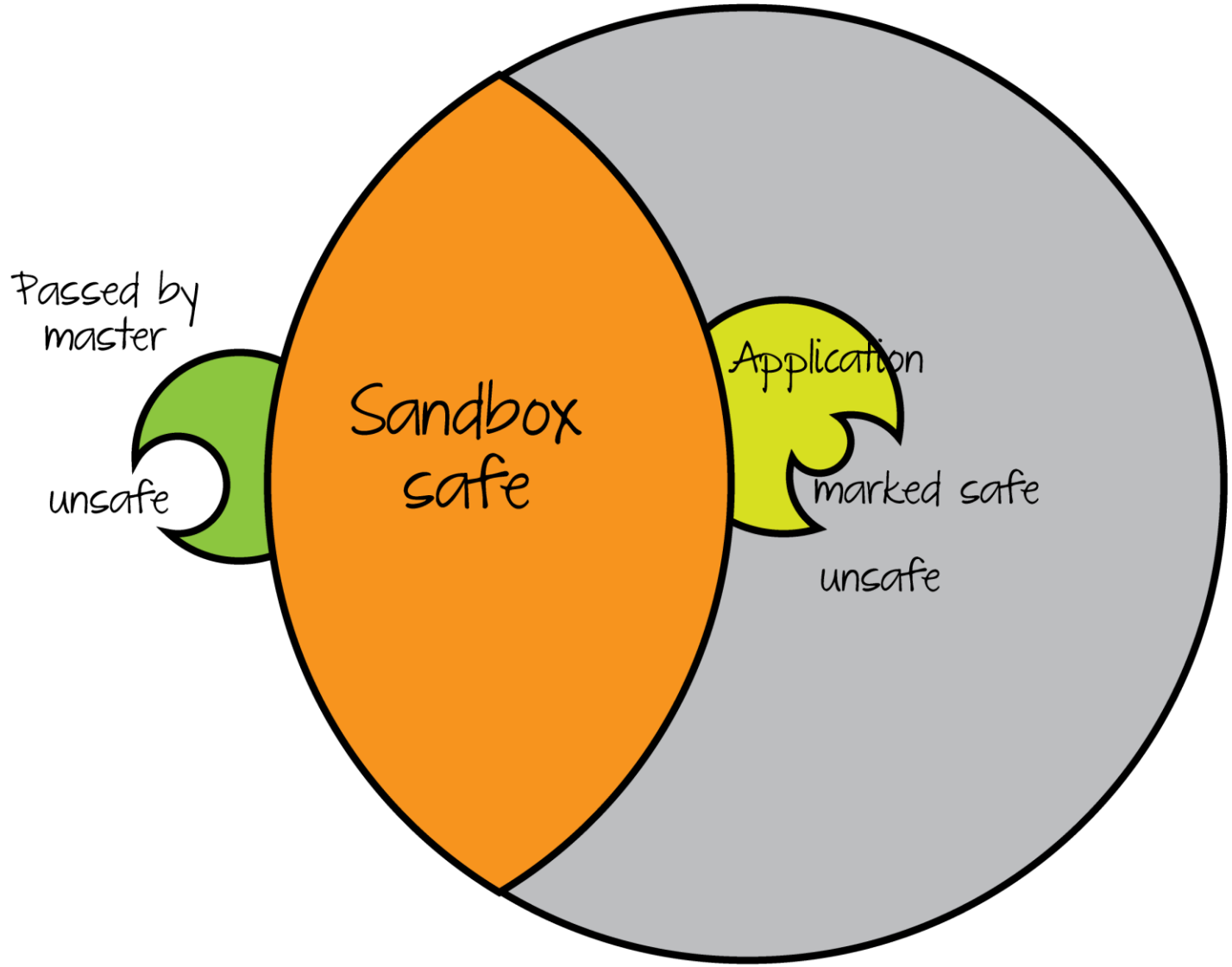


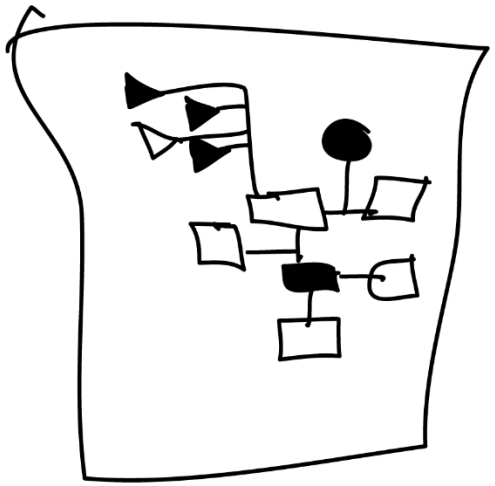

Sandbox
safe
Application
marked safe
unsafe

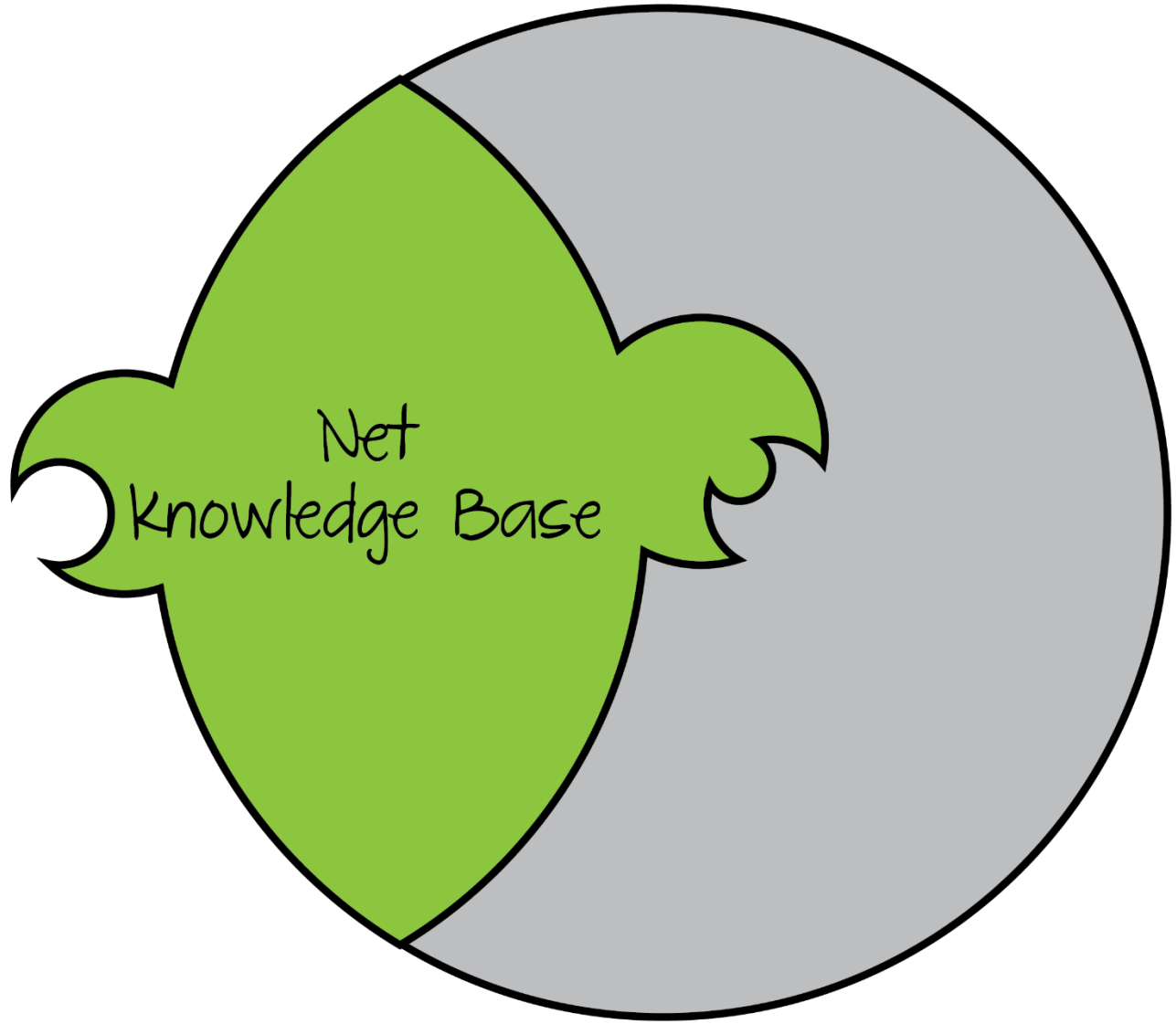



Passed by
master
Sandbox
safe
Application
marked safe
unsafe




Passed by master
unsafe
Sandbox safe
Application
marked safe
unsafe




Net
Knowledge Base

Querying The Pengine

client.pl

```
pengine_demo(Port) :-
      format(atom(URL),
'http://localhost:~d', [Port]),
      pengine_create(
          [ server(URL),
            src_text("
              q(X) :- p(X).
               p(a). p(b). p(c).
            ")
          ]),
      pengine_event_loop(handle,
[]).
```

```
handle(create(ID, _)) :-
      pengine_ask(ID, q(_X), []).
handle(success(ID, X, false)) :- !,
      writeln(X),
      pengine_destroy(ID).
handle(success(ID, X, true)) :-
      writeln(X),
      pengine_next(ID, []).
```

javascript example

```
<script type="text/x-prolog">
q(X) :- p(X).
   p(a).
   p(b).
   p(c).
</script>
```

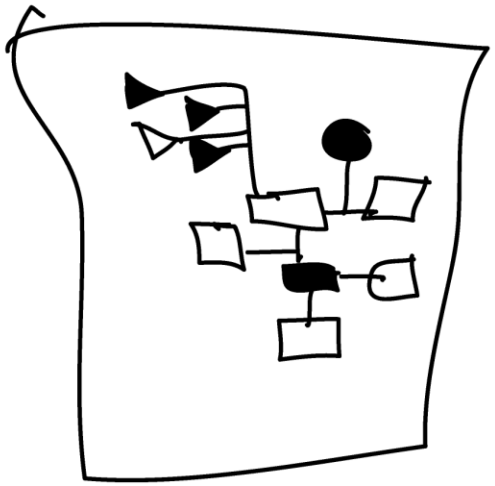
```
<script>
var pengine = new Pengine({
    oncreate: handleCreate,
    onsuccess: handleSuccess,
    onerror: handleSuccess
});
function handleCreate () {
    pengine.ask("q(X)", {
        template:'X'
     });
}
function handleSuccess() {
    $('#out').html(this.data);
    pengine.next();
}
</script>
```

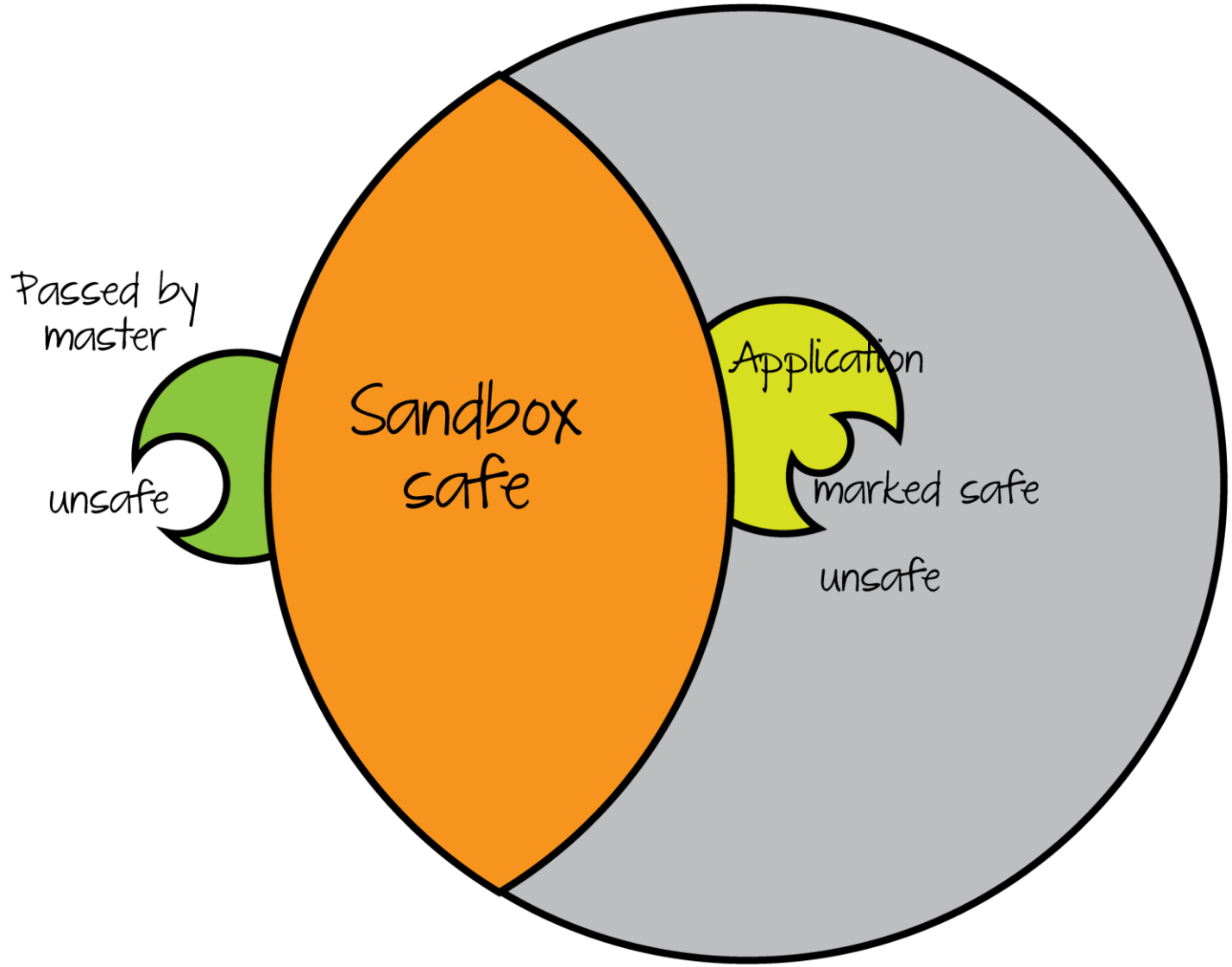
Client using pengine_rpc

```
rpc_demo(Port, X) :-
    pengine_rpc(
       'http://someserver.nl/',
        member(X,
            [aap, noot, mies])).
```

IO

- pengine_input(+Prompt, -Term)
- pengine_output(+Term)




Passed by master
unsafe
Sandbox safe
Application
marked safe
unsafe

main.pl

```
:- use_module(library(pengines)).
:- use_module(library(sandbox)).
:-
use_module(pengine_sandbox:my_apis).
```

my_apis.pl

```
:- module(my_apis, [my_public/1]).

:- use_module(library(dcg/basics)).

my_public(X) :-
        dont_say_walrus(X),
        debug(pengine_example, 'my_public says
~w', [X]).

dont_say_walrus(X) :-
        atom_codes(X, XC),
        phrase(walrus, XC),
        !,fail.
dont_say_walrus(_).
walrus --> string(_) ,  "walrus", string(_).
```

main.pl

```
:- use_module(library(pengines)).
:- use_module(library(sandbox)).
:- use_module(pengine_sandbox:my_apis).
```

my_apis.pl

```
:- module(my_apis, [my_public/1, my_unsafe/1]).

:- use_module(library(dcg/basics)).

my_public(X) :-
        dont_say_walrus(X),
        debug(pengine_example, 'my_public says
~w', [X]).

my_unsafe(X) :-
        atom_length(X, Len),
        Len < 25,
        open('foo.txt', write, Stream),
        format(Stream, 'Hello Out There ~w~n',
[X]),
        close(Stream).

:- multifile sandbox:safe_primitive/1.

sandbox:safe_primitive(my_apis:my_unsafe(_)).
```

Federating Queries

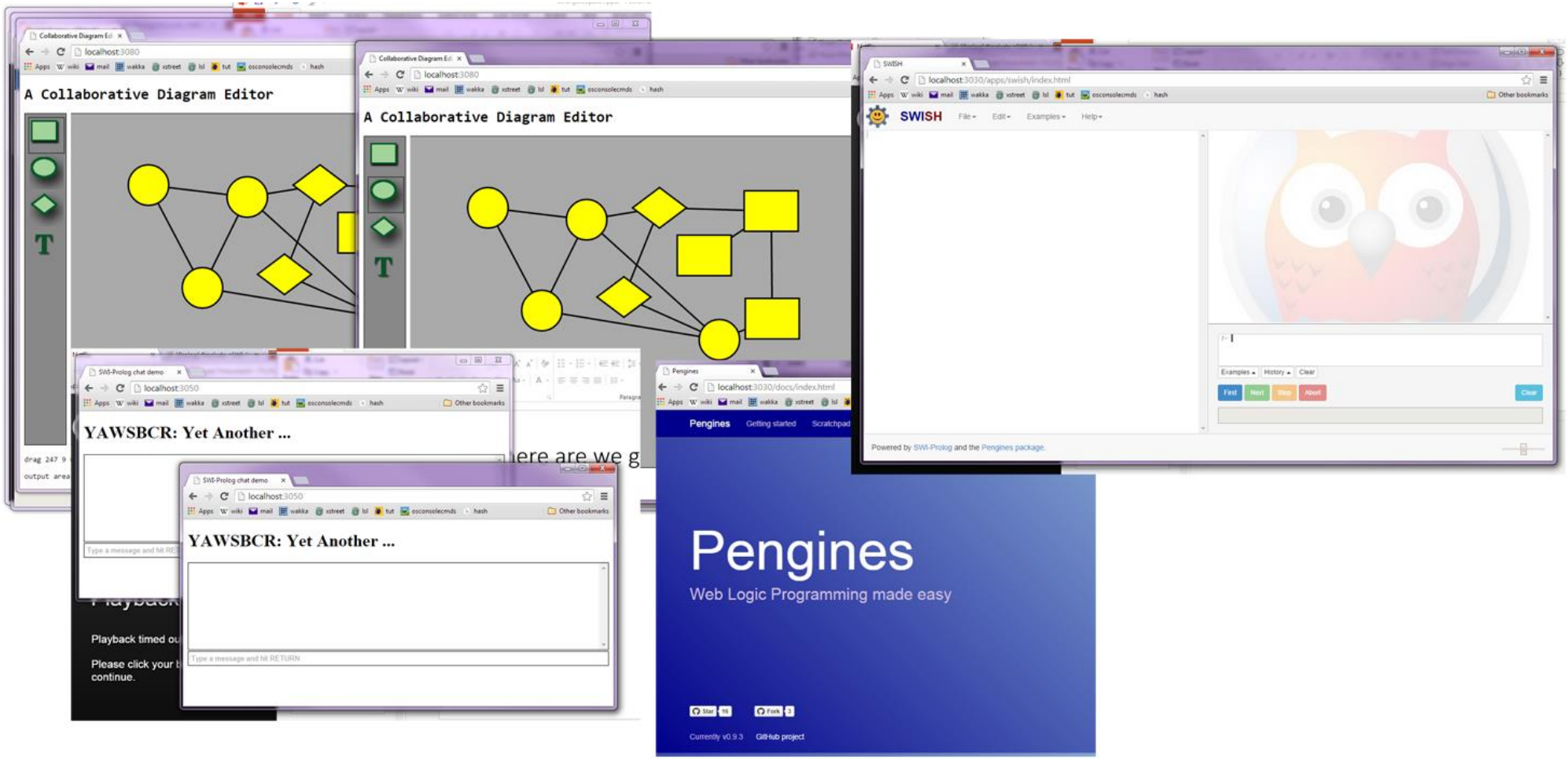
```
from_everywhere(Name, Address) :-
     pengine_rpc('http://someserver.com/',
          rdf(S, rdf:type, foaf:Agent)),
     pengine_rpc('http://someserver.com/',
          rdf(S, foaf:name, Name)),
     pengine_rpc('http://whitepages.com/pengines/',
          rdf(S2, foaf:name, Name)),
     pengine_rpc('http://whitepages.com/pengines/',
          rdf(S2, wp:address, Address)).
```

Cliopatria Whitepaper

Useful paper for understanding the relationship between SWI-Prolog and RDF

http://cliopatria.swi-prolog.org/help/whitepaper.html

Where Are We Going?



Resources

slides	https://github.com/Anniepoo/strangeloop2014
Sources/Nightlies	
SWISH 2.0	http://swish.swi-prolog.org
SWISH 1.0	http://pengines.swi-prolog.org
whiteboard	https://github.com/Anniepoo/whiteboard.git
chat	https://github.com/JanWielemaker/swi-chat
Docs	http://pengines.swi-prolog.org/docs/documentation.html