

Beating the No Win Scenario

Joe DeVivo
@joedevivo



Beating the No Win Scenario

Joe DeVivo
@joedevivo

Maybe you've heard of me



Things to know about Joe

Things to know about Joe

- I've never done this before

Things to know about Joe

- I've never done this before
- Riak User since 2010

Things to know about Joe

- I've never done this before
- Riak User since 2010
- Started in Basho Professional Services

Things to know about Joe

- I've never done this before
- Riak User since 2010
- Started in Basho Professional Services
- Moved to Basho Engineering

Things to know about Joe

- I've never done this before
- Riak User since 2010
- Started in Basho Professional Services
- Moved to Basho Engineering
- OTP Express Alumni (last year!)

A man in a white tuxedo jacket and a woman in a blue and brown patterned dress are looking up at a large, ornate chandelier hanging from the ceiling. The room has dark wood paneling. The man is holding a small device in his hand.

Why are you here?



HANDBOOK FOR NEW EMPLOYEES

A fearless adventure
in knowing what to do
when no one's there
telling you what to do

FIRST EDITION
2012





VALVE®

HANDBOOK FOR NEW EMPLOYEES

A fearless adventure
in knowing what to do
when no one's there
telling you what to do

FIRST EDITION
2012





I can do what I
want



A Call to Adventure!

A scene from Star Trek: The Motion Picture. Captain Kirk (William Shatner) is on the left, shown in profile facing right. He has dark hair and is wearing a red and white striped shirt. Mr. Spock (Leonard Nimoy) is on the right, also facing right. He has dark hair in a flat-top style and is wearing a red uniform with a white collar. They are standing in a hallway with blue walls and doors in the background.

The needs of the many,
bro



It was a painful
process

Python, Virtual Machines, and Bears

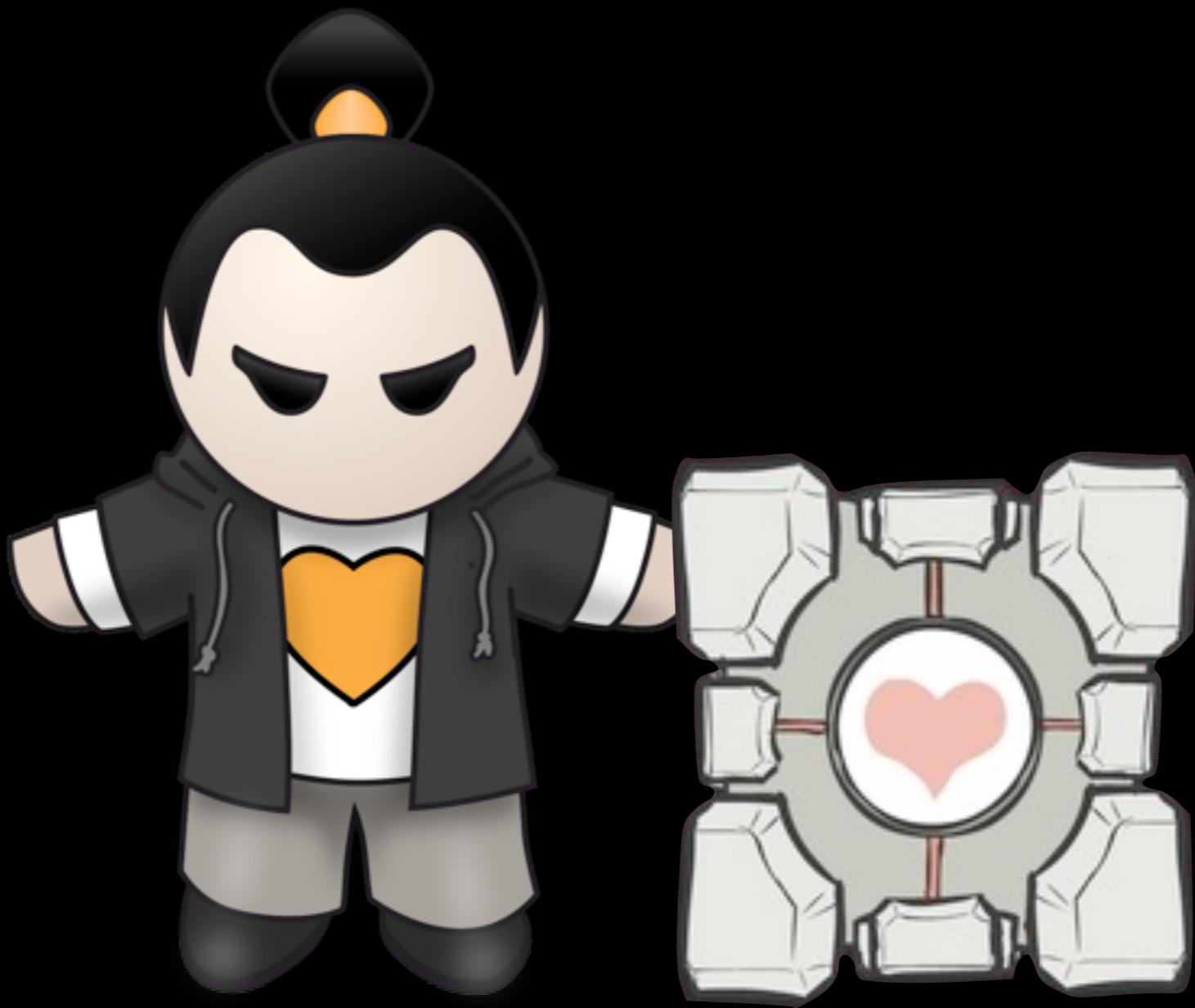




Ooooh My



I needed this pain
there are no hi-res ST:V images



Choose your own
adventure

A close-up photograph of two men in flight suits. The man on the left is looking down at something in his hands, while the man on the right looks on. They are wearing orange flight suits with white stripes on the shoulders and white patches on the sleeves.

The needs of the few,
bro

I felt like I couldn't
win

A close-up photograph of a man with dark hair and a mustache. He has a serious, focused expression, looking slightly upwards and to the side. He is wearing a brown leather jacket over a white t-shirt. In his right hand, he holds a small, rectangular object, possibly a piece of paper or a small device. The background is blurred, showing what appears to be a forest or outdoor setting.

I don't like to lose

OMG That's Complicated!



Testing A Distributed System is Hard



Jared
 @_jared

Follow

lmao if you think testing distributed systems
is any easier than writing distributed
systems.

Reply

Retweet

Favorite



More

7
RETWEETS

10
FAVORITES



9:15 AM - Feb 19, 2013

Testing A
Distributed
System is Hard
impossible

A photograph from a movie scene showing five men in a library. In the foreground, a man with dark hair and glasses, wearing a dark suit and a patterned tie, looks towards the right. Behind him, four other men are visible, all looking in the same direction. The background shows bookshelves filled with books.

Were you in the
shit?



Is it really a no-
win scenario?

Y So Hard?

Y So Hard?

- Multiple Machines!

Y So Hard?

- Multiple Machines!
- Networks!

Y So Hard?

- Multiple Machines!
- Networks!
- Nodes!

Y So Hard?

- Multiple Machines!
- Networks!
- Nodes!
- Timing!

Y So Hard?

- Multiple Machines!
- Networks!
- Nodes!
- Timing!
- Load!

A close-up of Captain Kirk from Star Trek: Into Darkness. He has a serious expression, looking slightly off-camera to his left. He is wearing his iconic brown leather flight jacket over a white shirt. A phaser is held in his right hand, pointing towards the bottom left of the frame. His left hand is partially visible, resting near his chest. The background is blurred, showing what appears to be a forest or jungle setting.

What would Cap'n Kirk
Do?

Change the
Conditions of the
Test

Change the Conditions of the Test

- We can run a ‘devrel’ to simulate a distributed system locally

Change the Conditions of the Test

- We can run a ‘devrel’ to simulate a distributed system locally
- Don’t focus on what’s right, focus on what works

A black and white photograph of a man from the chest up. He is wearing a dark tuxedo jacket over a white dress shirt with a red striped bow tie. He has short, light-colored hair and is wearing round-rimmed glasses. He is looking directly at the camera with a neutral to slightly stern expression. The background is dark and out of focus.

I see you favor
riak test



tonight I do

riak_test

riak_test

- A simple abstraction of riak cluster functions

riak_test

- A simple abstraction of riak cluster functions
- Based on a QuickCheck test

riak_test

- A simple abstraction of riak cluster functions
- Based on a QuickCheck test
- Erlang!

riak_test

- A simple abstraction of riak cluster functions
- Based on a QuickCheck test
- Erlang!
- A Machine! Your Machine!

Embracing the Most Important Feature

A User Base

riak_test is a
ghetto

riak_test is a ghetto

- Not a traditional OTP Application

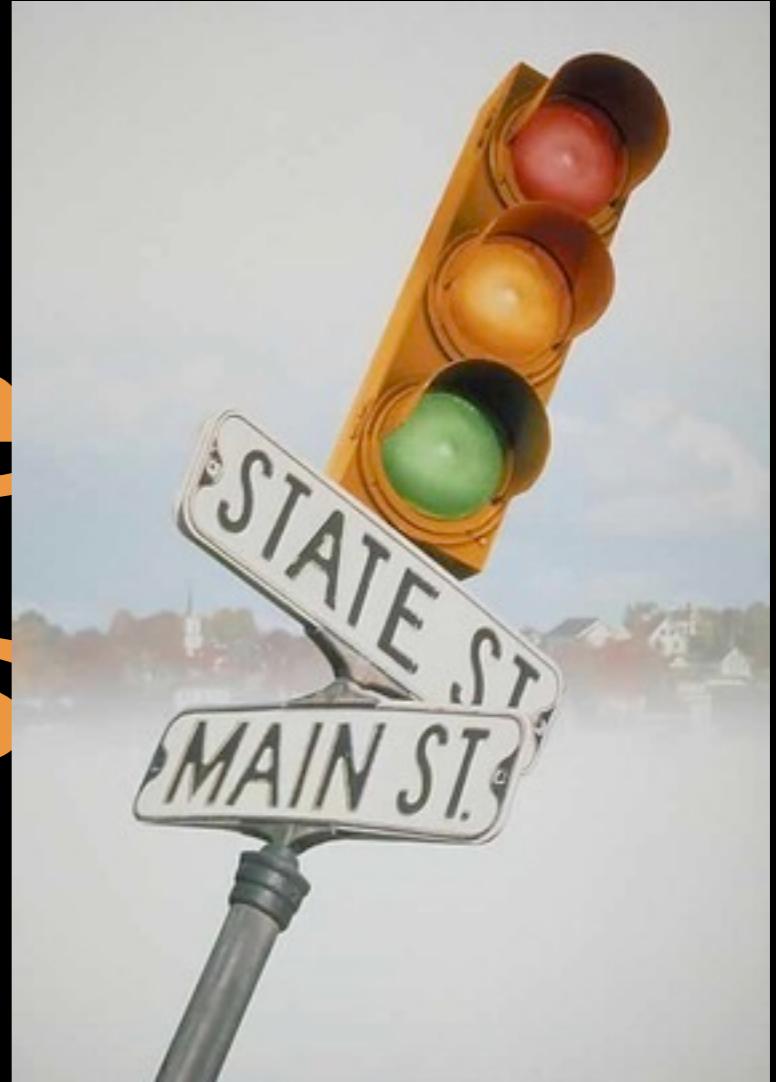
riak_test is a ghetto

- Not a traditional OTP Application
- uses rebar and app.src (I'm not an animal!)

riak_test is a ghetto

- Not a traditional OTP Application
- uses rebar and app.src (I'm not an animal!)
- Procedural

State and the Main Process

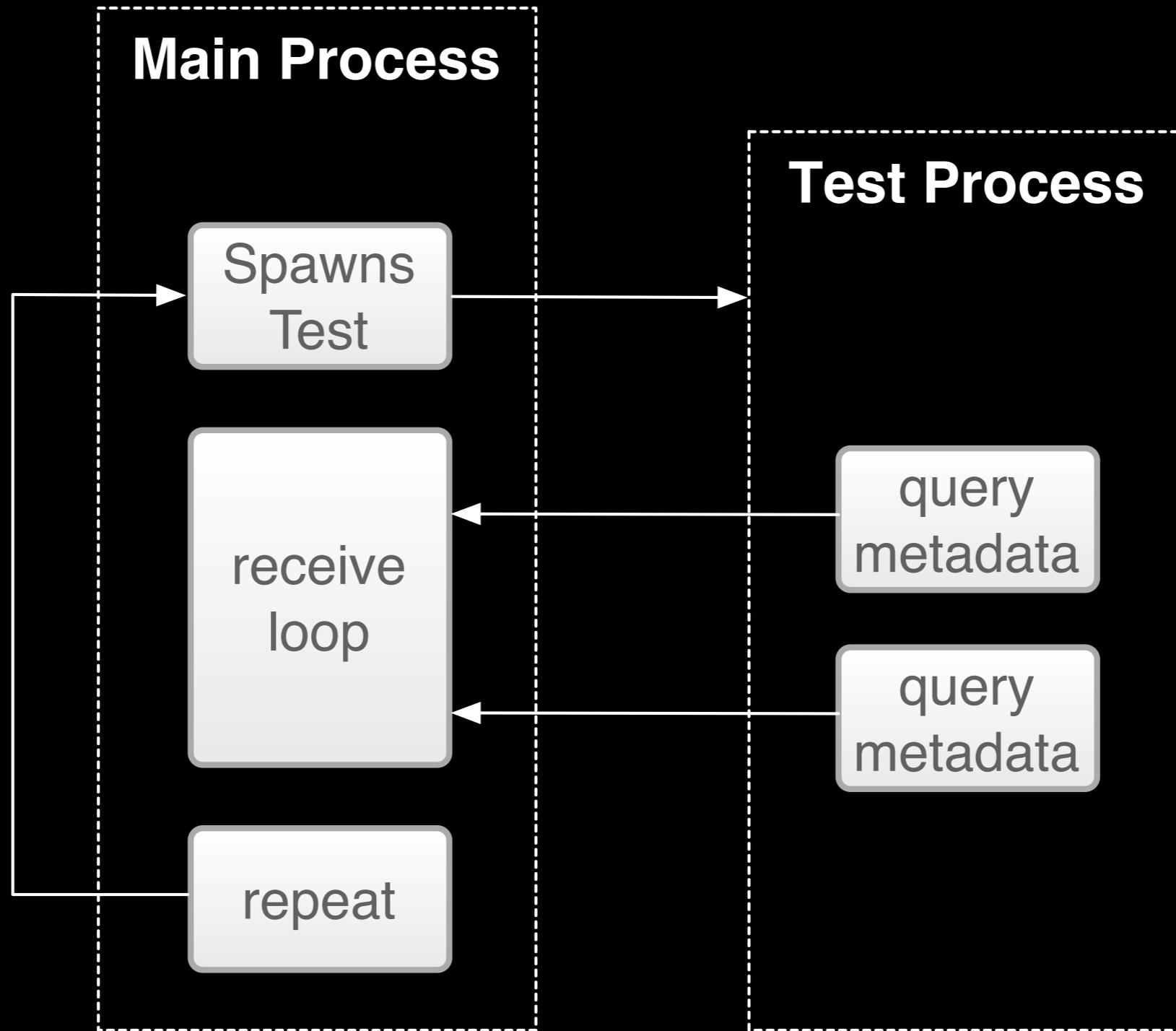


A photograph of a man and a woman looking at a motorcycle handlebar. The man, wearing a brown leather jacket, is speaking. The woman, wearing a denim jacket, is listening. They are outdoors, with a building visible in the background.

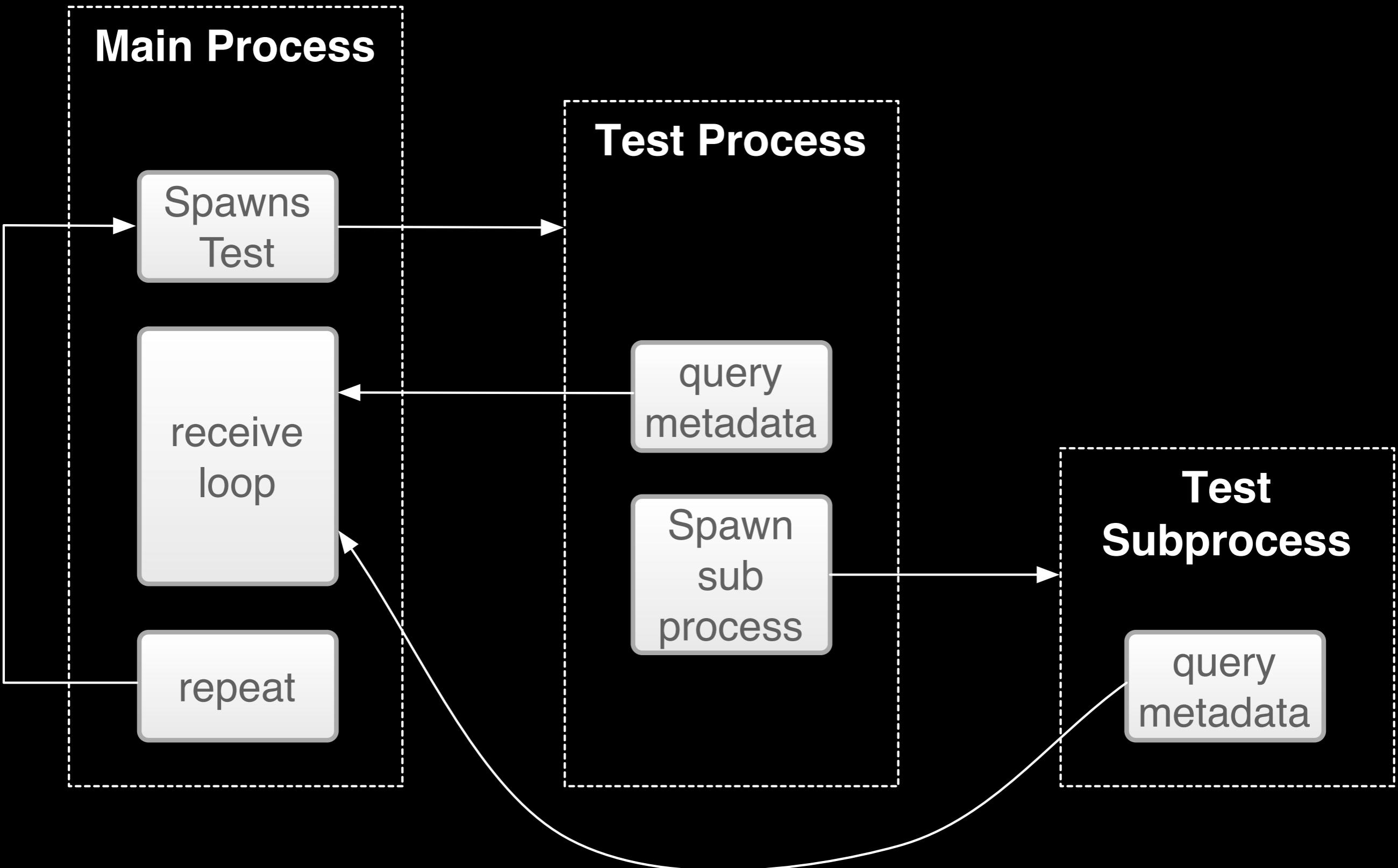
**WHOSE DATA IS
THIS?**

Test Metadata

- Each test is spawned into its own process
- `riak_test` enters a receive loop
- `riak_test` answers metadata questions

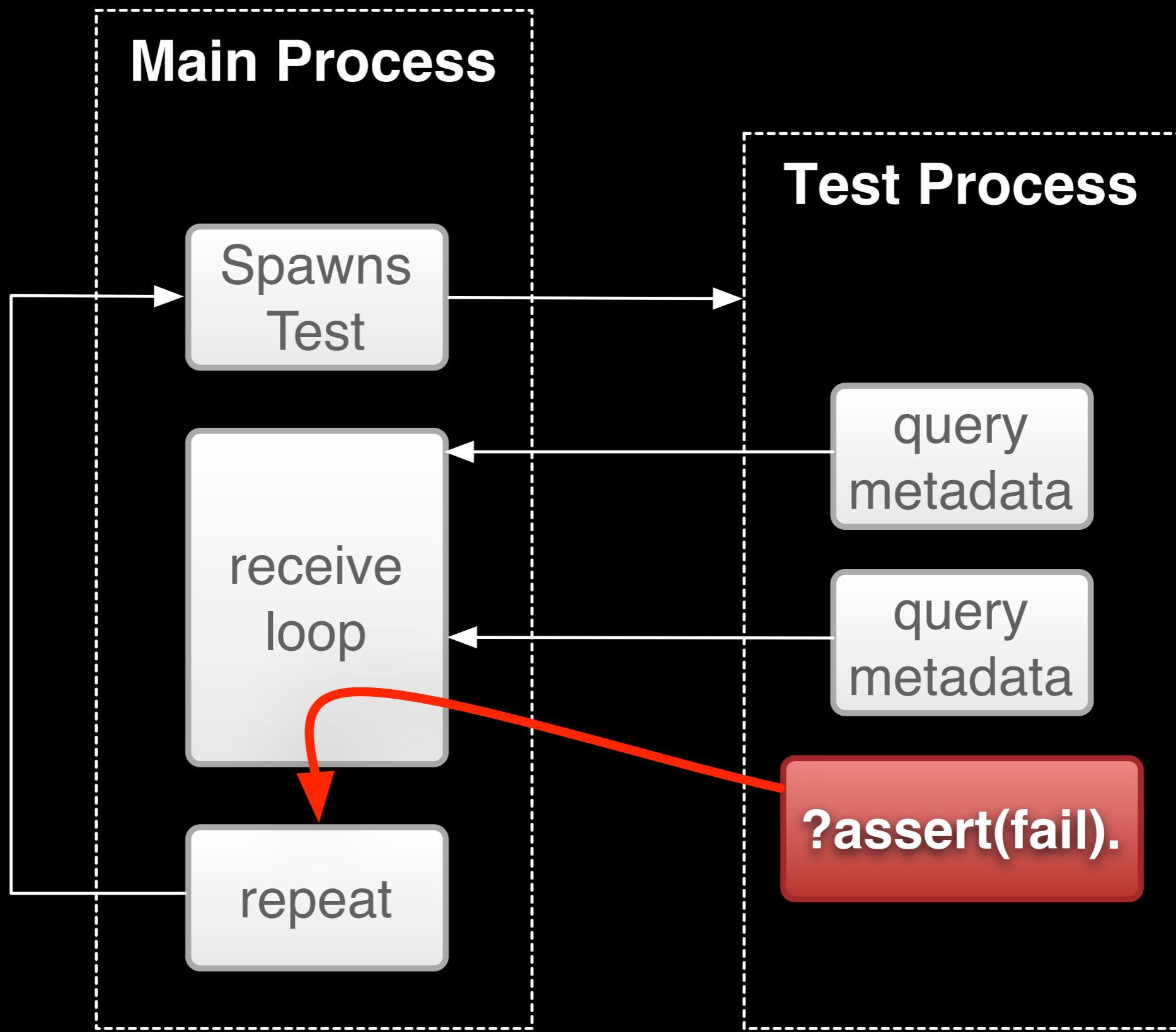


```
rec_loop(Pid, TestModule, TestMetaData) ->
receive
    metadata ->
        Pid ! {metadata, TestMetaData},
        rec_loop(Pid, TestModule, TestMetaData)
end.
```



```
rec_loop(Pid, TestModule, TestMetaDatum) ->
receive
    metadata ->
        Pid ! {metadata, TestMetaDatum},
        rec_loop(Pid, TestModule, TestMetaDatum);
    {metadata, P} ->
        P ! {metadata, TestMetaDatum},
        rec_loop(Pid, TestModule, TestMetaDatum)
end.
```

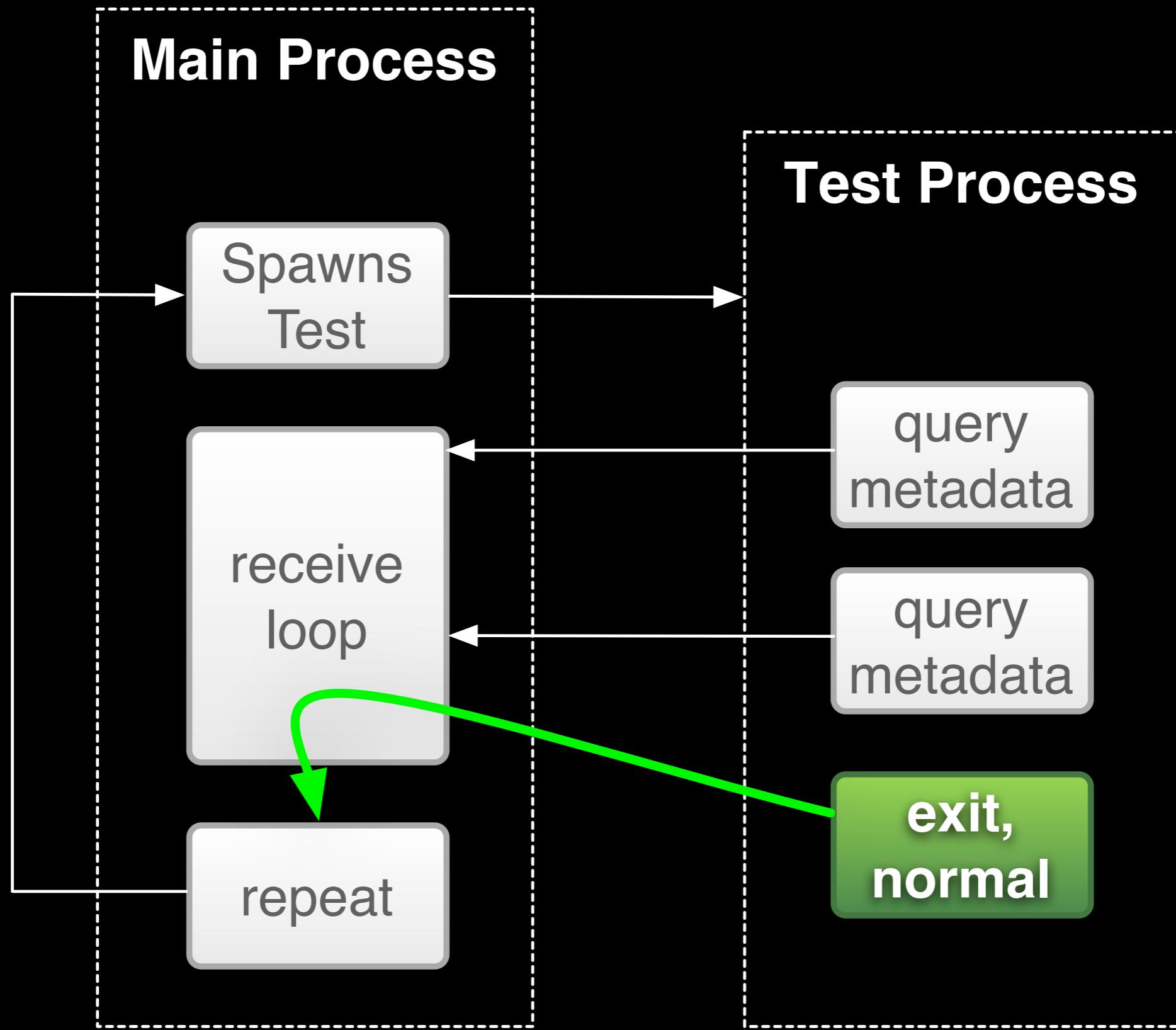
?assert



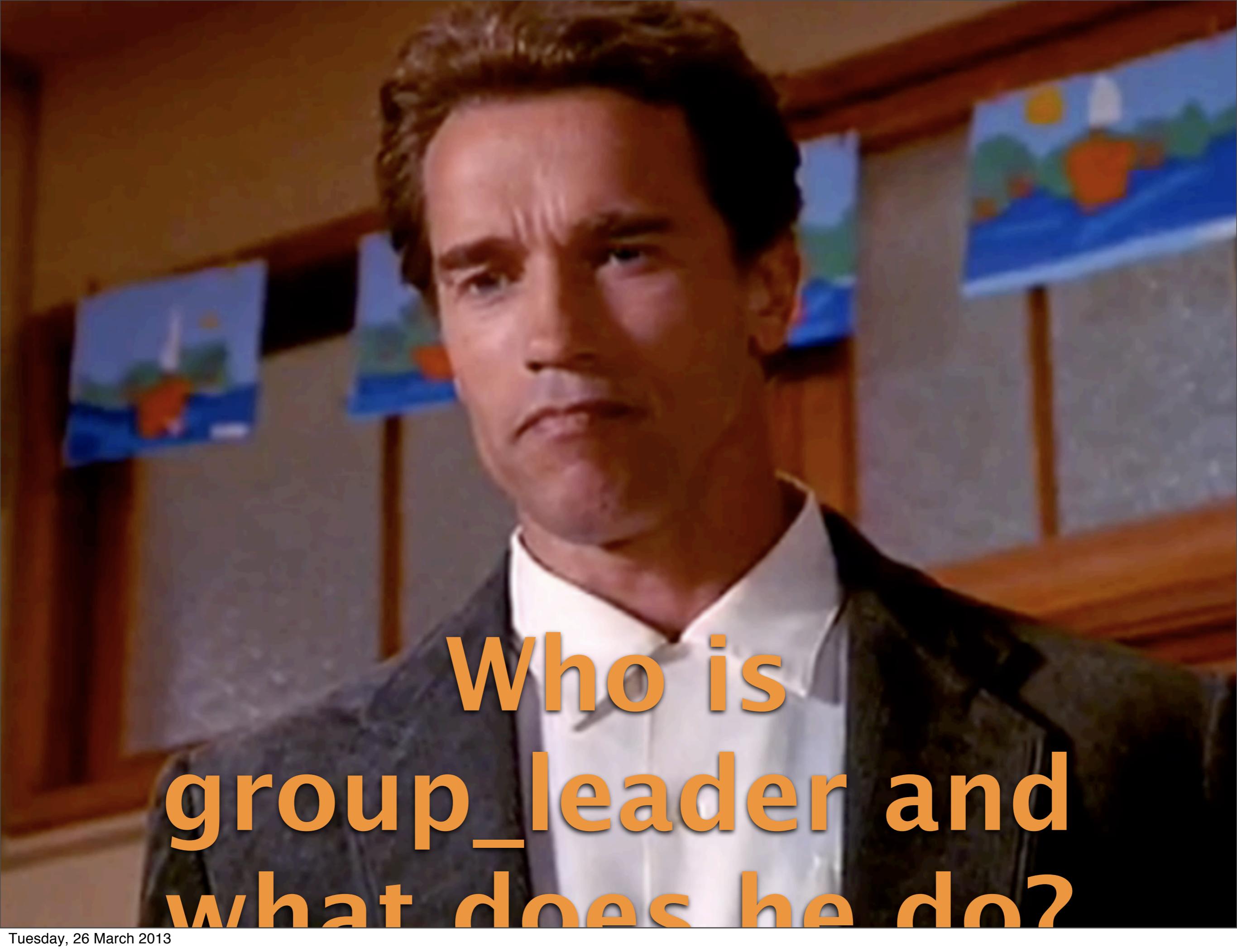
```
rec_loop(Pid, TestModule, TestMetaData) ->
receive
    metadata ->
        Pid ! {metadata, TestMetaData},
        rec_loop(Pid, TestModule, TestMetaData);
    {metadata, P} ->
        P ! {metadata, TestMetaData},
        rec_loop(Pid, TestModule, TestMetaData);
    {'EXIT', Pid, normal} ->
        {pass, undefined};
    {'EXIT', Pid, Error} ->
        lager:warning("~-s failed: ~p",
                      [TestModule, Error]),
        {fail, Error}
end.
```



even if I fail, I don't
fail



Collecting Output

A close-up photograph of a man with dark hair, wearing a dark suit jacket over a white shirt. He has a serious, contemplative expression, looking slightly to his left. The background is blurred, showing what appears to be an indoor setting with colorful artwork on the walls.

who is
group_leader and
what does he do?

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),
{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),

{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),

{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),
{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),

{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

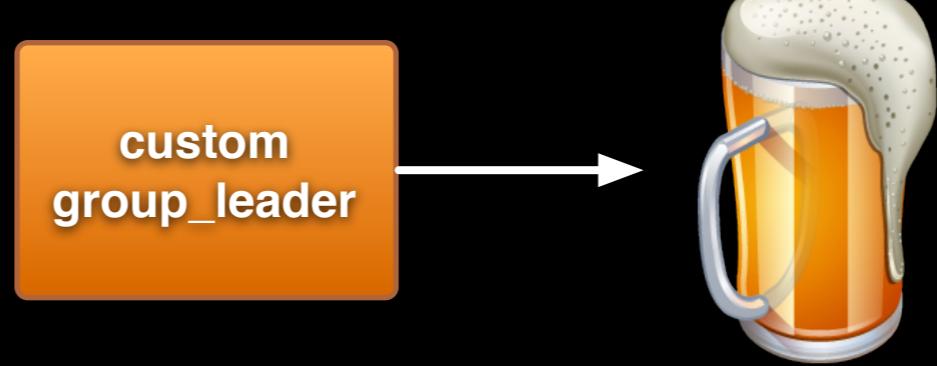
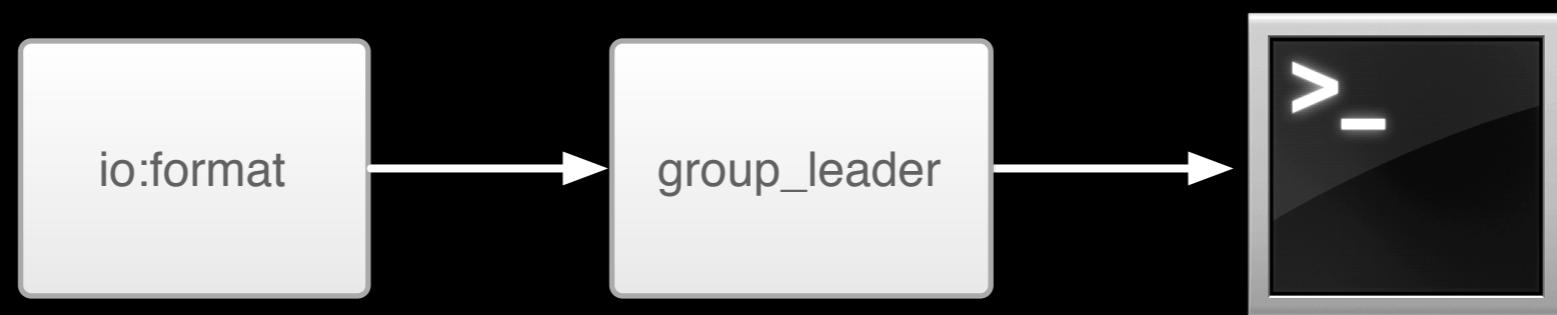
```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

Pid = spawn_link(TestModule, confirm, []),

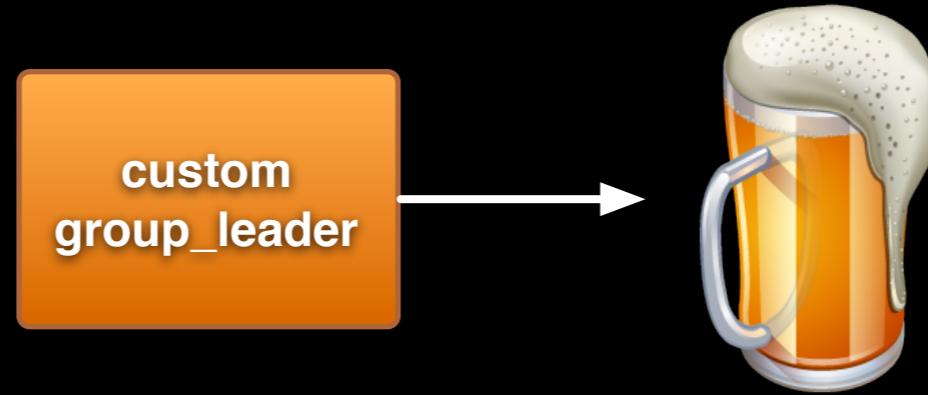
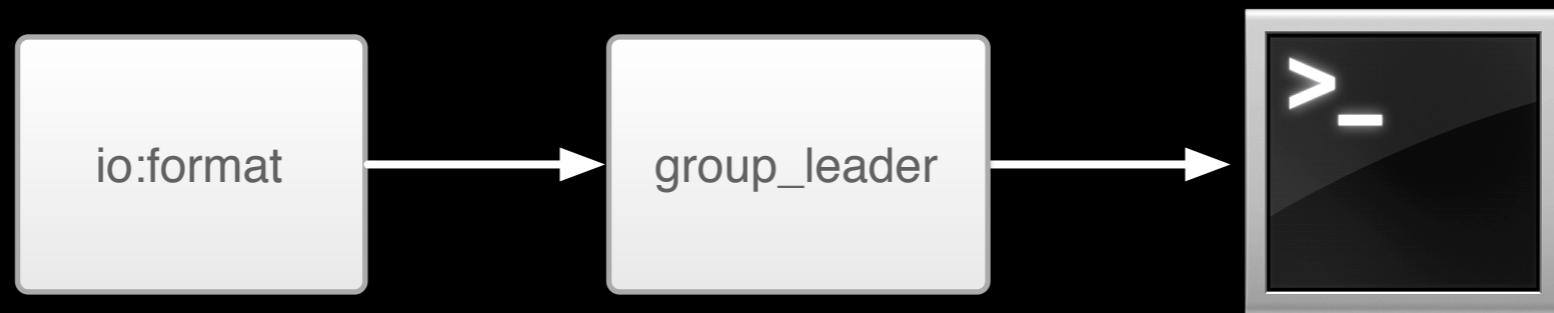
{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

```
OldGroupLeader = group_leader(),
NewGroupLeader =
    riak_test_group_leader:new_group_leader(self()),
group_leader(NewGroupLeader, self()),

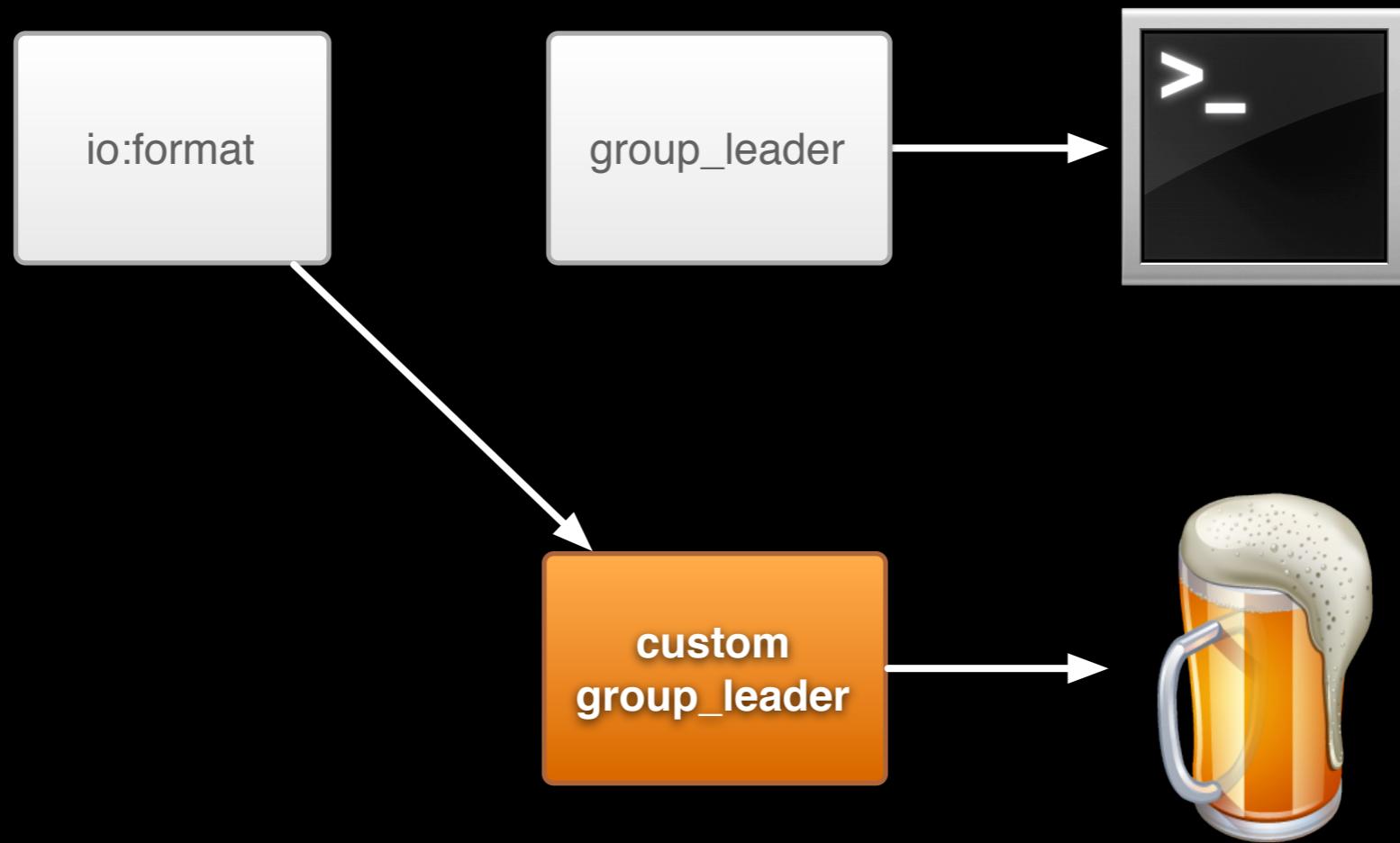
Pid = spawn_link(TestModule, confirm, []),
{Status, Reason} =
    rec_loop(Pid, TestModule, TestMetaDatum),
group_leader(OldGroupLeader, self()),
```

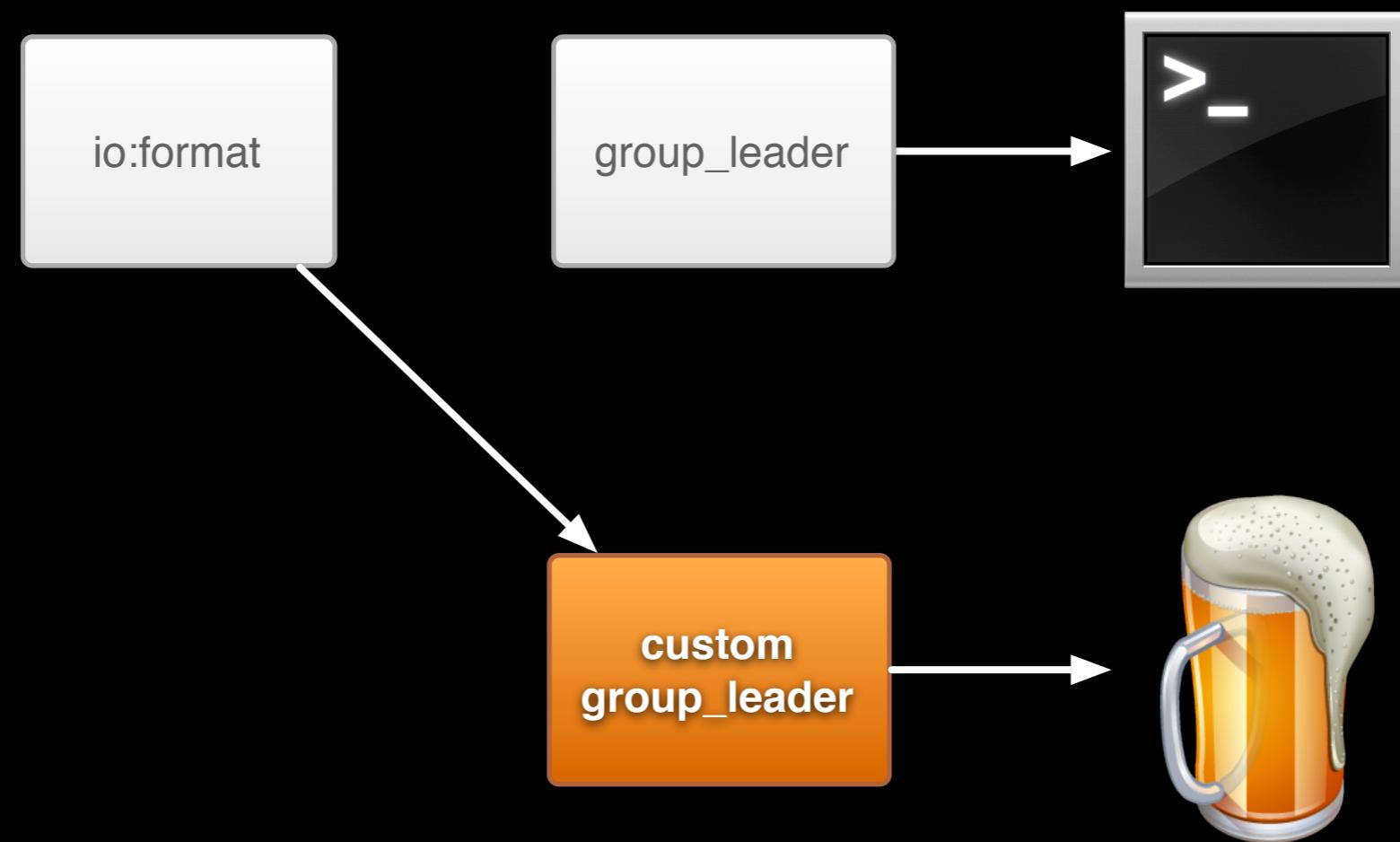


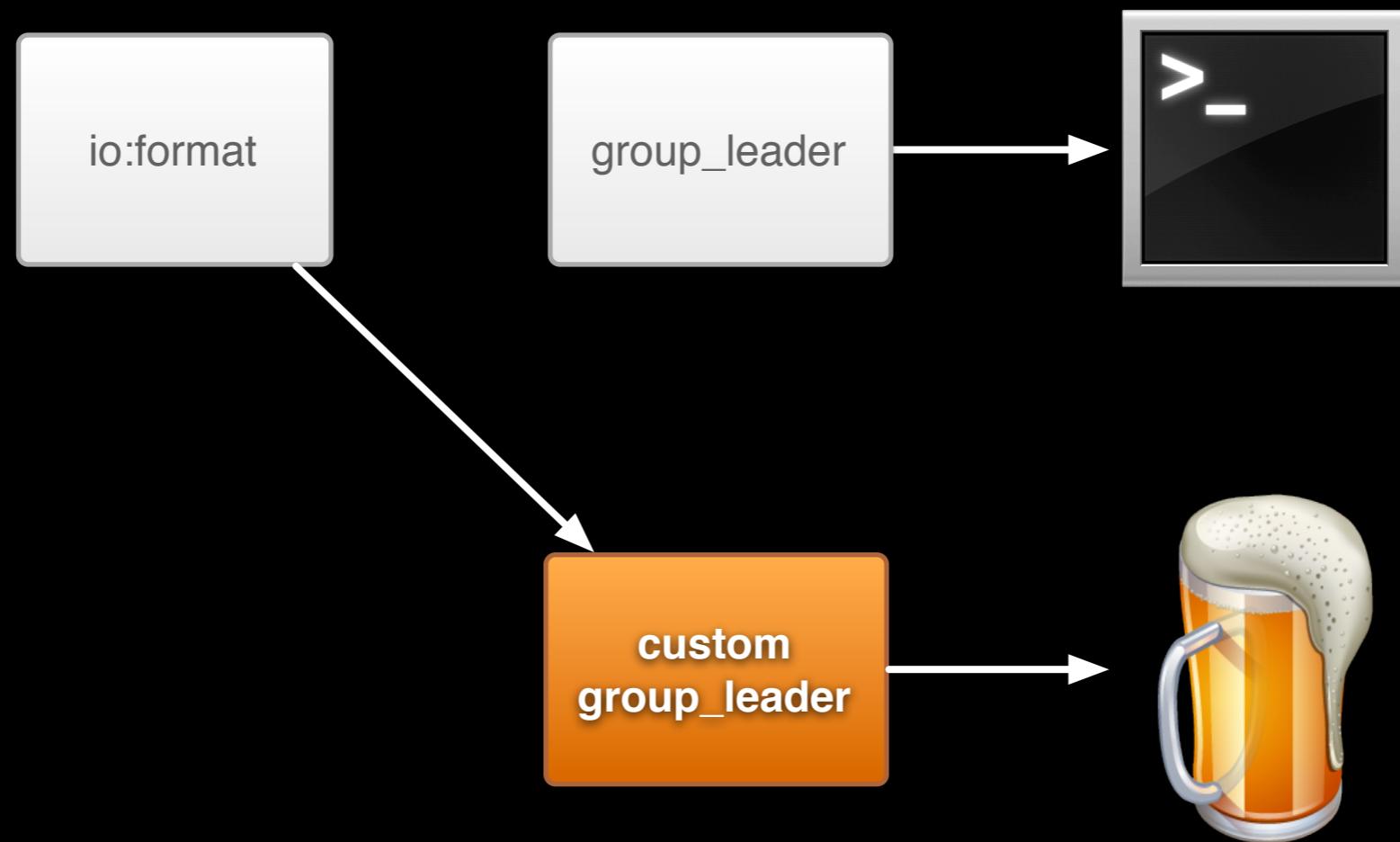
```
group_leader(NewGroupLeader, self()),
```



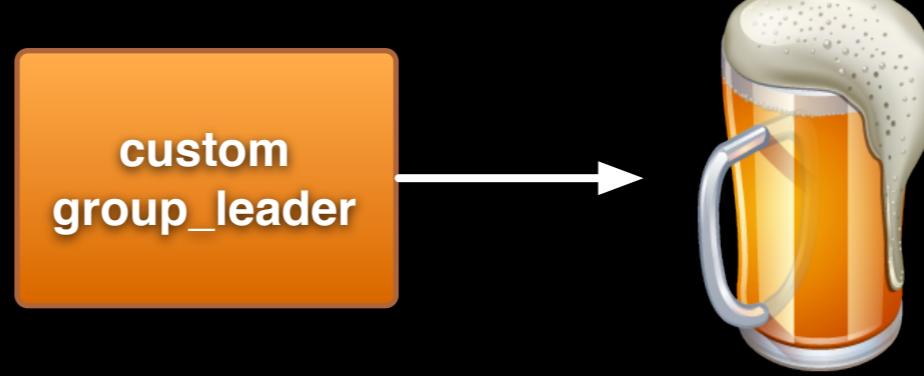
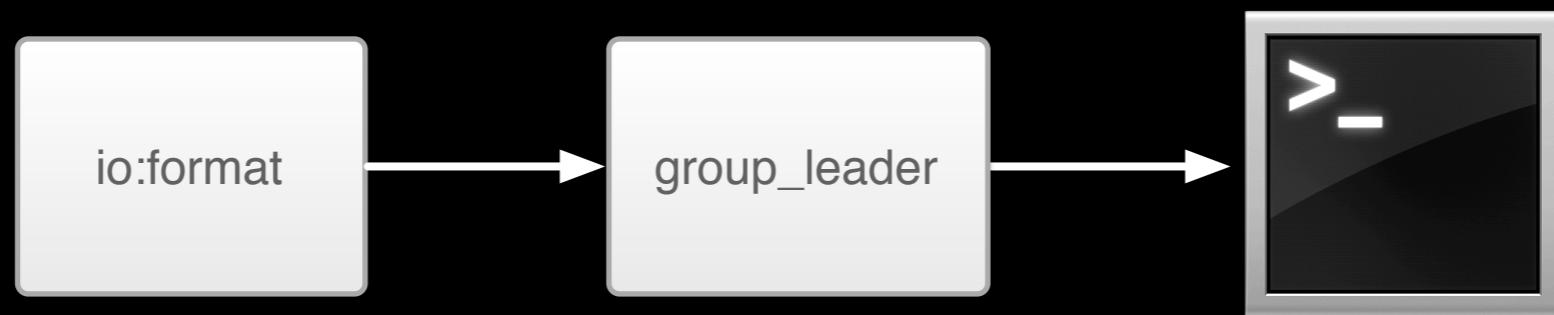
```
group_leader(NewGroupLeader, self()),
```







`group_leader(OldGroupLeader, self()),`



`group_leader(OldGroupLeader, self()),`

console interaction



who is
port_command and
what does he do?

```
Port = open_port({spawn, Cmd},  
[ stream, use_stdio, exit_status,  
binary, stderr_to_stdout]),  
  
port_command(Port, Msg),  
  
receive  
{Port, {data, Data}} ->  
%% do something  
end
```

```
Port = open_port({spawn, Cmd},  
[ stream, use_stdio, exit_status,  
binary, stderr_to_stdout]),  
  
port_command(Port, Msg),  
  
receive  
 {Port, {data, Data}} ->  
 %% do something  
end
```

```
Port = open_port( {spawn, Cmd} ,  
[ stream, use_stdio, exit_status,  
binary, stderr_to_stdout] ),  
  
port_command(Port, Msg),  
  
receive  
    {Port, {data, Data}} ->  
        %% do something  
end
```

```
Port = open_port( {spawn, Cmd} ,  
[ stream, use_stdio, exit_status,  
binary, stderr_to_stdout] ),  
  
port_command(Port, Msg),  
  
receive  
{Port, {data, Data}} ->  
%% do something  
end
```

```
Port = open_port({spawn, Cmd},  
[ stream, use_stdio, exit_status,  
binary, stderr_to_stdout]),  
  
port_command(Port, Msg),  
  
receive  
{Port, {data, Data}} ->  
%% do something  
end
```

```
rt:attach(Node, CommandExpectations),
```

```
rt:attach(Node, CommandExpectations),  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

```
rt:attach(Node, CommandExpectations),  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

```
rt:attach(Node, CommandExpectations),  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

```
rt:attach(Node, CommandExpectations),  
  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

```
rt:attach(Node, CommandExpectations),  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

```
rt:attach(Node, CommandExpectations),  
[{expect, "\(^D to exit\)"},  
{send,  
 "riak_core_ring_manager:get_my_ring()."},  
{expect, "dict,"},  
{send, [4]}] %% 4 = ^D
```

commands



sean carey
@densoneold



Following

commands

Reply

Retweeted

Favorited

More

7
RETWEETS

9
FAVORITES



2:39 PM - Feb 8, 2008

13:14:17.861 [info] Cmd: java -Dcom.basho.riak.host=127.0.0.1 -Dcom.basho.riak.http.port=10018 -Dcom.basho.riak.pbc.port=10017 -cp /tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-jar-with-dependencies-and-tests.jar:/tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-tests.jar org.junit.runner.JUnitCore
com.basho.riak.client.AllTests

13:16:54.580 [info]

JUnit version 4.4

I

I

Time: 153.93

OK (583 tests)

13:14:17.861 [info] Cmd: java -Dcom.basho.riak.host=127.0.0.1 -Dcom.basho.riak.http.port=10018 -Dcom.basho.riak.pbc.port=10017 -cp /tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-jar-with-dependencies-and-tests.jar:/tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-tests.jar org.junit.runner.JUnitCore
com.basho.riak.client.AllTests

13:16:54.580 [info]

JUnit version 4.4

.

.

Time: 153.93

OK (583 tests)

11:29:38.463 [info] Cmd: java -Dcom.basho.riak.host=127.0.0.1 -Dcom.basho.riak.http.port=10018 -Dcom.basho.riak.pbc.port=10017 -cp /tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-jar-with-dependencies-and-tests.jar:/tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-tests.jar org.junit.runner.JUnitCore
com.basho.riak.client.AllTests

11:29:38.865 [info]

11:29:38.865 [info] JUnit version 4.4

11:29:39.301 [info]

11:29:40.234 [info]

11:29:41.049 [info]

11:29:42.158 [info] .

11:29:43.303 [info] .

11:29:45.009 [info]

11:29:46.328 [info]

11:29:47.286 [info]

11:29:49.044 [info] ...I.

11:29:50.333 [info]

11:29:51.190 [info] ..

11:29:52.070 [info]

11:29:53.845 [info]I.

11:29:54.939 [info]

11:29:55.160 [info] .

11:29:56.939 [info]

11:29:38.463 [info] Cmd: java -Dcom.basho.riak.host=127.0.0.1 -Dcom.basho.riak.http.port=10018 -Dcom.basho.riak.pbc.port=10017 -cp /tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-jar-with-dependencies-and-tests.jar:/tmp/riak_test_scratch/riak-client-1.1.0-SNAPSHOT-tests.jar org.junit.runner.JUnitCore
com.basho.riak.client.AllTests

11:29:38.865 [info]

11:29:38.865 [info] JUnit version 4.4

11:29:39.301 [info]

11:29:40.234 [info]

11:29:41.049 [info]

11:29:42.158 [info] .

11:29:43.303 [info] .

11:29:45.009 [info]

11:29:46.328 [info]

11:29:47.286 [info]

11:29:49.044 [info] ...I.

11:29:50.333 [info]

11:29:51.190 [info] ..

11:29:52.070 [info]

11:29:53.845 [info]I.

11:29:54.939 [info]

11:29:55.160 [info] .

11:29:56.939 [info]

Prerequisites

```
-prereq("java").  
-prereq("curl").  
  
check_prereqs(Module) ->  
    Prereqs = proplists:get_all_values(prereq,  
        Module:module_info(attributes)),  
    P2 = [ {Prereq, rt:which(Prereq)} || Prereq <- Prereqs],  
    lager:info("~s prereqs: ~p", [Module, P2]),  
    [  
        lager:warning(~s prereq '~s' not installed.,  
            [Module, P])  
    || {P, false} <- P2],  
    GoodToGo = lists:all(fun({_, Present}) -> Present end, P2),  
    ?assertEqual({all_prereqs_present, true},  
        {all_prereqs_present, GoodToGo}).
```

```
-prereq("java").  
-prereq("curl").  
  
check_prereqs(Module) ->  
    Prereqs = proplists:get_all_values(prereq,  
        Module:module_info(attributes)),  
    P2 = [ {Prereq, rt:which(Prereq)} || Prereq <- Prereqs],  
    lager:info("~s prereqs: ~p", [Module, P2]),  
    [  
        lager:warning("~s prereq '~s' not installed.",  
            [Module, P])  
    || {P, false} <- P2],  
    GoodToGo = lists:all(fun({_, Present}) -> Present end, P2),  
    ?assertEqual({all_prereqs_present, true},  
        {all_prereqs_present, GoodToGo}).
```

```
-prereq("java").  
-prereq("curl").  
  
check_prereqs(Module) ->  
    Prereqs = proplists:get_all_values(prereq,  
        Module:module_info(attributes)),  
    P2 = [ {Prereq, rt:which(Prereq)} || Prereq <- Prereqs],  
    lager:info("~s prereqs: ~p", [Module, P2]),  
    [  
        lager:warning(~s prereq '~s' not installed.,  
            [Module, P])  
    || {P, false} <- P2],  
    GoodToGo = lists:all(fun({_, Present}) -> Present end, P2),  
    ?assertEqual({all_prereqs_present, true},  
        {all_prereqs_present, GoodToGo}).
```

```
-prereq("java").  
-prereq("curl").  
  
check_prereqs(Module) ->  
    Prereqs = proplists:get_all_values(prereq,  
        Module:module_info(attributes)),  
    P2 = [ {Prereq, rt:which(Prereq)} || Prereq <- Prereqs],  
    lager:info("~s prereqs: ~p", [Module, P2]),  
    [  
        lager:warning(~s prereq '~s' not installed.,  
            [Module, P])  
    || {P, false} <- P2],  
    GoodToGo = lists:all(fun({_, Present}) -> Present end, P2),  
    ?assertEqual({all_prereqs_present, true},  
        {all_prereqs_present, GoodToGo}).
```

```
-prereq("java").  
-prereq("curl").  
  
check_prereqs(Module) ->  
    Prereqs = proplists:get_all_values(prereq,  
        Module:module_info(attributes)),  
    P2 = [ {Prereq, rt:which(Prereq)} || Prereq <- Prereqs],  
    lager:info("~s prereqs: ~p", [Module, P2]),  
    [  
        lager:warning(~s prereq '~s' not installed.,  
            [Module, P])  
    || {P, false} <- P2],  
    GoodToGo = lists:all(fun({_, Present}) -> Present end, P2),  
    ?assertEqual({all_prereqs_present, true},  
        {all_prereqs_present, GoodToGo}).
```

```
14:27:32.712 [info] Checking for presence of java
14:27:32.721 [info] Checking for presence of curl
14:27:32.727 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}]
14:27:32.728 [notice] Running Test client_java_verify
```

```
14:27:32.712 [info] Checking for presence of java
14:27:32.721 [info] Checking for presence of curl
14:27:32.727 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}]
14:27:32.728 [notice] Running Test client_java_verify
```

```
14:27:32.712 [info] Checking for presence of java
14:27:32.721 [info] Checking for presence of curl
14:27:32.727 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}]
14:27:32.728 [notice] Running Test client_java_verify
```

```
14:27:32.712 [info] Checking for presence of java
14:27:32.721 [info] Checking for presence of curl
14:27:32.727 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}]
14:27:32.728 [notice] Running Test client_java_verify
```

```
14:27:32.712 [info] Checking for presence of java
14:27:32.721 [info] Checking for presence of curl
14:27:32.727 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}]
14:27:32.728 [notice] Running Test client_java_verify
```

```
-prereq("java").  
-prereq("curl").  
-prereq("missing_prereq").
```

```
-prereq("java").  
-prereq("curl").  
-prereq("missing_prereq").
```

```
10:59:31.402 [info] Checking for presence of java
10:59:31.414 [info] Checking for presence of curl
10:59:31.423 [info] Checking for presence of missing_prereq
10:59:31.435 [warning] `missing_prereq` is not installed
10:59:31.436 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}, {"missing_prereq":false}]
10:59:31.437 [warning] client_java_verify prereq 'missing_prereq' not
installed.
escript: exception error: {assertEqual_failed,
  [{module,riak_test_runner},
   {line,122},
   {expression,"{ all_prereqs_present , GoodToGo }"},{expected,{all_prereqs_present,true}}, {value,{all_prereqs_present,false}}]}}
```

```
10:59:31.402 [info] Checking for presence of java
10:59:31.414 [info] Checking for presence of curl
10:59:31.423 [info] Checking for presence of missing_prereq
10:59:31.435 [warning] `missing_prereq` is not installed
10:59:31.436 [info] client_java_verify prereqs:
  [{"java":true}, {"curl":true}, {"missing_prereq":false}]
10:59:31.437 [warning] client_java_verify prereq 'missing_prereq' not
installed.
escript: exception error: {assertEqual_failed,
  [{module,riak_test_runner},
   {line,122},
   {expression,"{ all_prereqs_present , GoodToGo }"},{expected,{all_prereqs_present,true}}, {value,{all_prereqs_present,false}}]}]
```

```
?assertEqual(true, GoodToGo).
```

```
?assertEqual(true, GoodToGo).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"GoodToGo"},  
{expected,true},  
{value,false}]}
```

```
?assertEqual(true, GoodToGo).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"GoodToGo"},  
{expected,true},  
{value,false}]}
```

```
?assertEqual(  
    {all_prereqs_present, true},  
    {all_prereqs_present, GoodToGo}).
```

```
?assertEqual(  
    {all_prereqs_present, true},  
    {all_prereqs_present, GoodToGo}).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"{ all_prereqs_present , GoodToGo }"},  
{expected,{all_prereqs_present,true}},  
{value,{all_prereqs_present,false}}]}
```

```
?assertEqual(  
    {all_prereqs_present, true},  
    {all_prereqs_present, GoodToGo}).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"{ all_prereqs_present , GoodToGo }"},  
{expected,{all_prereqs_present,true}},  
{value,{all_prereqs_present,false}}]}
```

```
?assertEqual(  
    {all_prereqs_present, true},  
    {all_prereqs_present, GoodToGo}).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"{ all_prereqs_present , GoodToGo }"},  
{expected,{all_prereqs_present,true}},  
{value,{all_prereqs_present,false}}]}
```

```
?assertEqual(  
    {all_prereqs_present, true},  
    {all_prereqs_present, GoodToGo}).
```

```
escript: exception error: {assertEqual_failed,  
[{module,riak_test_runner},  
{line,122},  
{expression,"{ all_prereqs_present , GoodToGo }"},  
{expected,{all_prereqs_present,true}},  
{value,{all_prereqs_present,false}}]}
```

Reporting

1.3.0

1.3.0rc4

1.3.0rc3

1.3.0rc2

1.3.0rc1

1.3.0pre3

1.3.0pre2

1.3.0pre1

1.2.1

1.2.0

1.2.1rc2

	centos-5-64	centos-6-64	fedora-17-64	freebsd-9-64	osx-64	smartos-64	solaris-10u9-64	ubuntu-1004-32	ubuntu-1004-64	ubuntu-1104-64
--	-------------	-------------	--------------	--------------	--------	------------	-----------------	----------------	----------------	----------------

basic_command_line

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

client_java_verify

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

client_python_verify

L	L	L	L	L	L	L	L	L	L
---	---	---	---	---	---	---	---	---	---

client_ruby_verify

M	M	M	M	M	M	M	M	M	M
---	---	---	---	---	---	---	---	---	---

gh_riak_core_154

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

gh_riak_core_155

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

gh_riak_core_176

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

loaded_upgrade

B-2	B-2	B-2	B-1	B-2	B-2	B-2	B-2	B-2	B-2
B-1	B-1	B-1	B-1	L-1	B-1	B-1	B-1	B-1	B-1
L-2	L-2	L-2	L-2	L-1	L-2	L-2	L-2	L-1	L-2
L-1	L-1	L-1	L-1	L-1	L-1	L-1	L-1	L-1	L-1

mapred_verify_rt

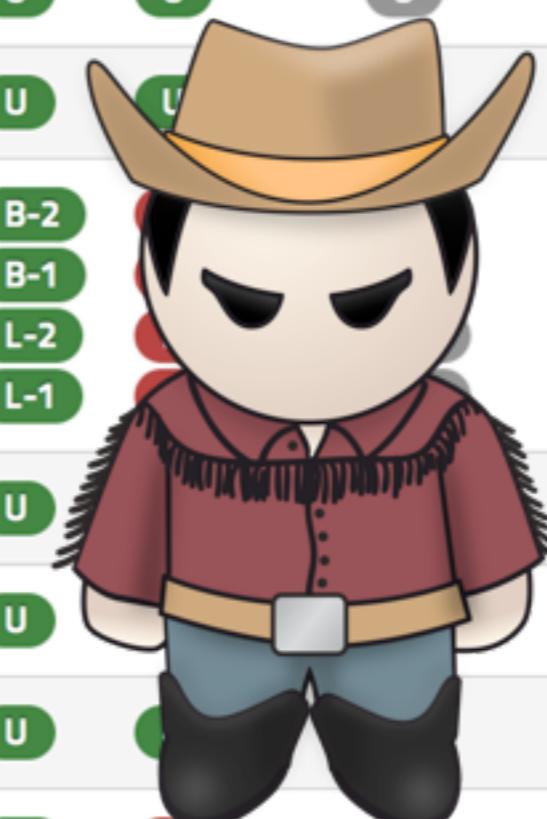
U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

partition_repair

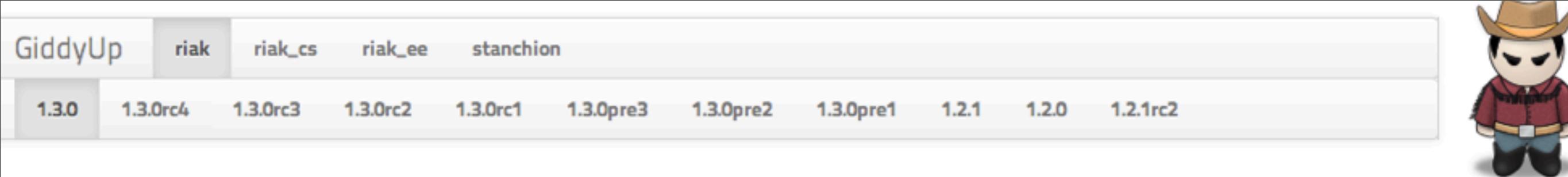
U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---

riaknostic_rt

U	U	U	U	U	U	U	U	U	U
---	---	---	---	---	---	---	---	---	---



riak_test_lager
backend



GiddyUp

riak

riak_cs

riak_ee

stanchion

1.3.0

1.3.0rc4

1.3.0rc3

1.3.0rc2

1.3.0rc1

1.3.0pre3

1.3.0pre2

1.3.0pre1

1.2.1

1.2.0

1.2.1rc2

loaded_upgrade / centos-5-64 / bitcask / previous

x Hide results

Success
1.3.0-1.3]15 days ago [riak-
1.3.0-1.3]Success
1.3.0-1.3]16 days ago [riak-
1.3.0-1.3]Failure
1.3.0-1.3]21 days ago [riak-
1.3.0-1.3]

```
ed,undefined,[{"key":undefined,"value":{}},60000}],60000]}]} in context child_terminated
13:49:34.216 [debug] Supervisor loaded_upgrade_worker_sup started loaded_upgrade_worker_sup:test
er_start_link(search_tester, 'dev3@127.0.0.1', previous) at pid <0.17445.0>
13:52:07.528 [error] CRASH REPORT Process <0.17416.0> with 0 neighbours exited with reason: {ass
ertEqual_failed,[{module,rt},{line,589},{expression,"wait_until ( Node , F )"},{expected,ok},{va
lue,fail}]} in rt:'-wait_for_service/2-fun-2-/3 line 589 in gen_server:init_it/6 line 328
13:52:07.536 [warning] loaded_upgrade failed: shutdown
13:52:07.539 [error]
```

[open log in new window](#)

	centos-5-64	centos-6-64	fedora-17-64	freebsd-9-64	osx-64	smartos-64	solaris-10u9-64	ubuntu-1004-32	ubuntu-1004-64	ubuntu-1104-64	ubuntu-1204-64
basic_command_line	U	U	U	U	U	U	U	U	U	U	U
client_java_verify	U	U	U	U	U	U	U	U	U	U	U
client_python_verify	L	L	L	L	L	L	L	L	L	L	L
client_ruby_verify	M	M	M	M	M	M	M	M	M	M	M
gh_riak_core_154	U	U	U	U	U	U	U	U	U	U	U
gh_riak_core_155	U	U	U	U	U	U	U	U	U	U	U
gh_riak_core_176	U	U	U	U	U	U	U	U	U	U	U
loaded_upgrade	B-2	B-2	B-2	B-1	B-2	B-2	B-2	B-2	B-2	B-2	B-2

Test Suites

1.3.0

1.3.0rc4

1.3.0rc3

1.3.0rc2

1.3.0rc1

1.3.0pre3

1.3.0pre2

1.3.0pre1

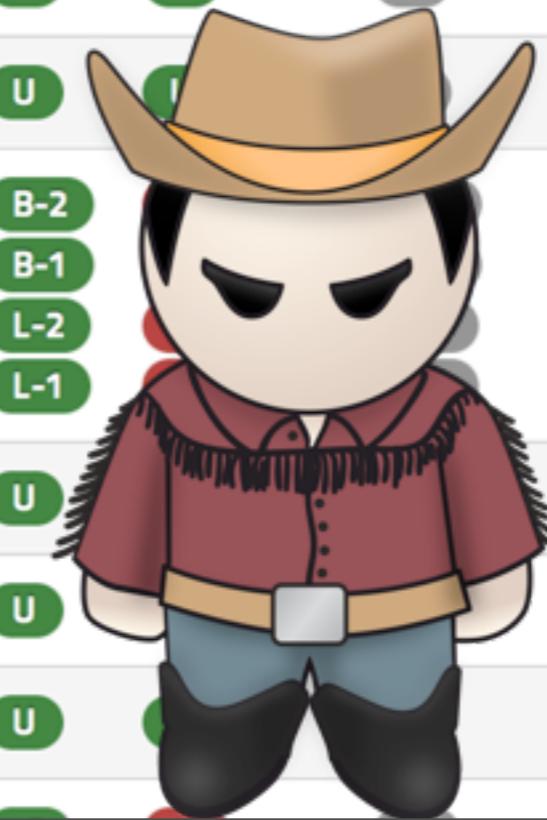
1.2.1

1.2.0

1.2.1rc2

	centos-5-64	centos-6-64	fedora-17-64	freebsd-9-64	osx-64	smartos-64	solaris-10u9-64	ubuntu-1004-32	ubuntu-1004-64	ubuntu-1104-64
--	-------------	-------------	--------------	--------------	--------	------------	-----------------	----------------	----------------	----------------

basic_command_line	U	U	U	U	U	U	U	U	U	U
client_java_verify	U	U	U	U	U	U	U	U	U	U
client_python_verify	L	L	L	L	L	L	L	L	L	L
client_ruby_verify	M	M	M	M	M	M	M	M	M	M
gh_riak_core_154	U	U	U	U	U	U	U	U	U	U
gh_riak_core_155	U	U	U	U	U	U	U	U	U	U
gh_riak_core_176	U	U	U	U	U	U	U	U	U	U
loaded_upgrade	B-2 B-1 L-2 L-1	B-2 B-1 L-2 L-1	B-2 B-1 L-2 L-1	B-1 L-1	B-2 B-1 L-2 L-1		B-2 B-1 L-2 L-1	B-2 B-1 L-2 L-1	B-2 B-1 L-2 L-1	B-2 B-1 L-2 L-1
mapred_verify_rt	U	U	U	U	U		U	U	U	U
partition_repair	U	U	U	U	U		U	U	U	U
riaknostic_rt	U	U	U	U	U		U	U	U	U



Returning to a Pristine State

```
10:59:27.995 [info] Resetting nodes to fresh  
state  
10:59:27.995 [debug] Running: git --git-  
dir "~/riak_test/.git" --work-tree "~/  
riak_test/" reset HEAD --hard  
10:59:29.662 [debug] Running: git --git-  
dir "~/riak_test/.git" --work-tree "~/  
riak_test/" clean -fd
```

```
10:59:27.995 [info] Resetting nodes to fresh  
state  
10:59:27.995 [debug] Running: git --git-  
dir "~/riak_test/.git" --work-tree "~/  
riak_test/" reset HEAD --hard  
10:59:29.662 [debug] Running: git --git-  
dir "~/riak_test/.git" --work-tree "~/  
riak_test/" clean -fd
```

Race Conditions?

It's all wibbly wobbly
timey whimey stuff



Race Conditions?

It's all wibbly wobbly
timey whimey stuff



```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
  {_, true} ->
    ok;
  {0, _} ->
    TimeoutFun(Node);
  _ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
          Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
  {_, true} ->
    ok;
  {0, _} ->
    TimeoutFun(Node);
  _ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
          Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
  {_, true} ->
    ok;
  {0, _} ->
    TimeoutFun(Node);
  _ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
  {_, true} ->
    ok;
  {0, _} ->
    TimeoutFun(Node);
  _ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

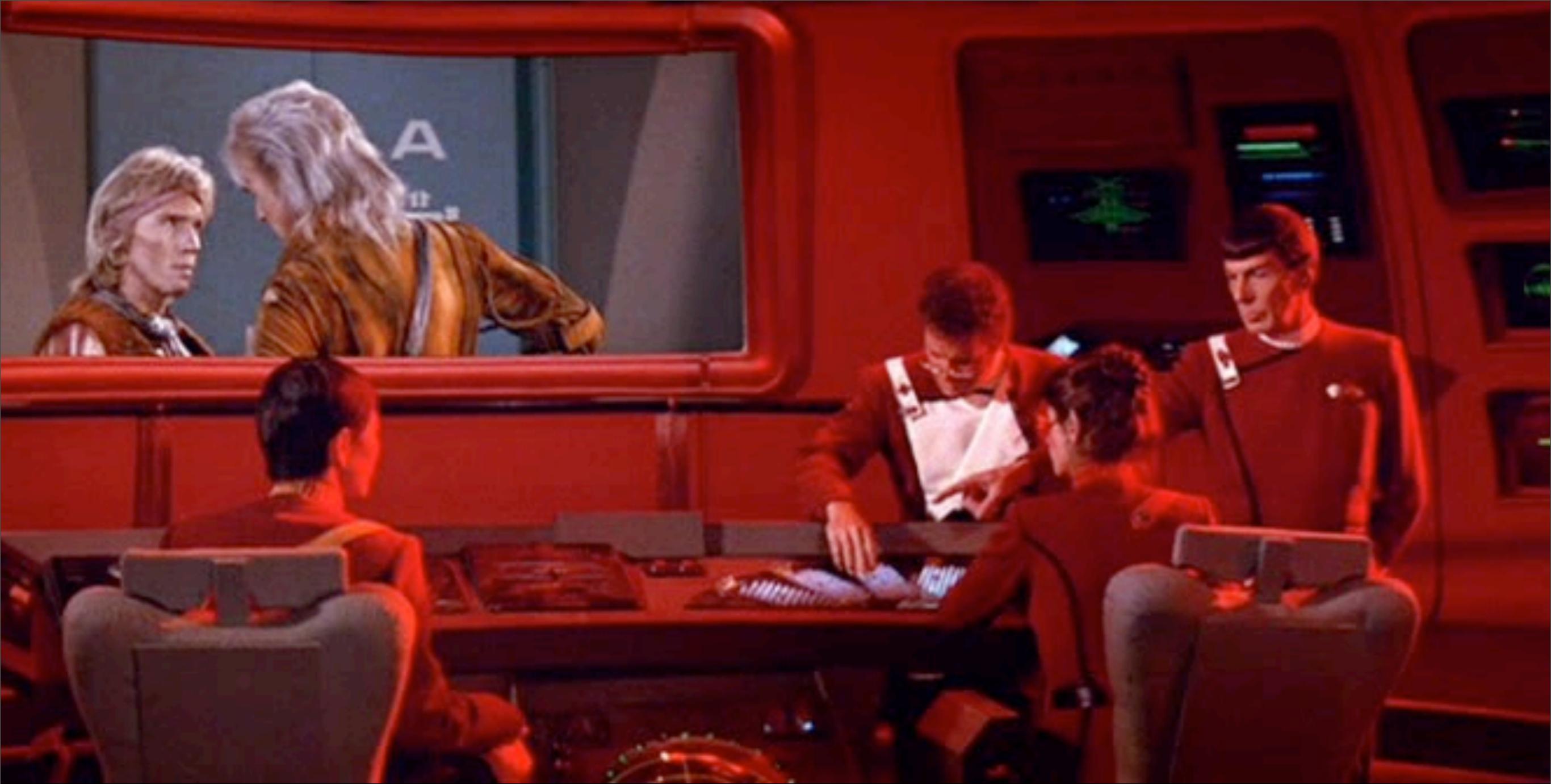
```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
{_, true} ->
    ok;
{0, _} ->
    TimeoutFun(Node);
_ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
{_, true} ->
    ok;
{0, _} ->
    TimeoutFun(Node);
_ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
{_, true} ->
    ok;
{0, _} ->
    TimeoutFun(Node);
_ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

```
wait_until(Node, Fun, Retry,
           Delay, TimeoutFun) ->
Pass = Fun(Node),
case {Retry, Pass} of
  {_, true} ->
    ok;
  {0, _} ->
    TimeoutFun(Node);
  _ ->
    timer:sleep(Delay),
    wait_until(Node, Fun, Retry-1,
               Delay, TimeoutFun)
end.
```

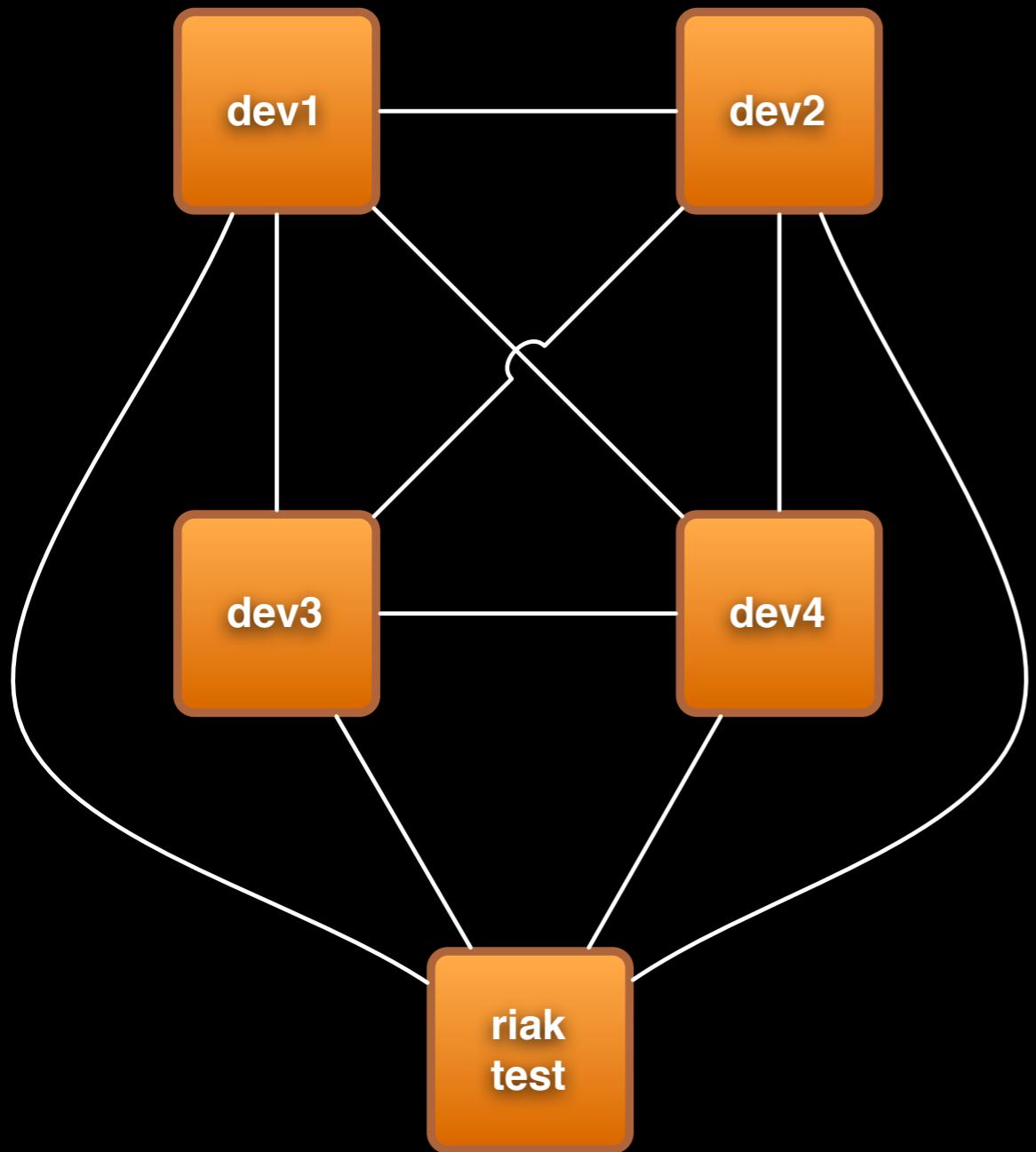
Network Partitions

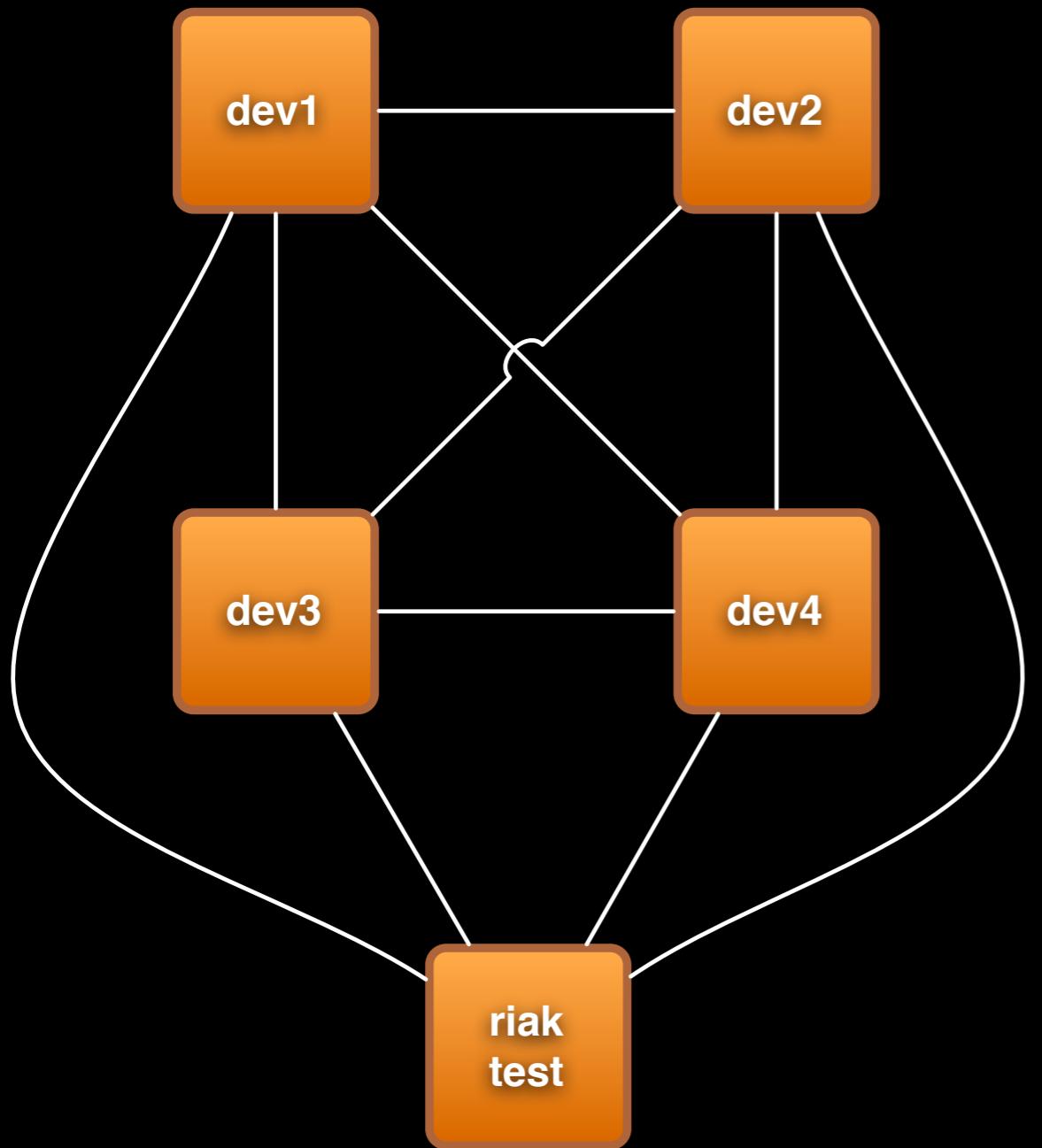


The Prefix Code?

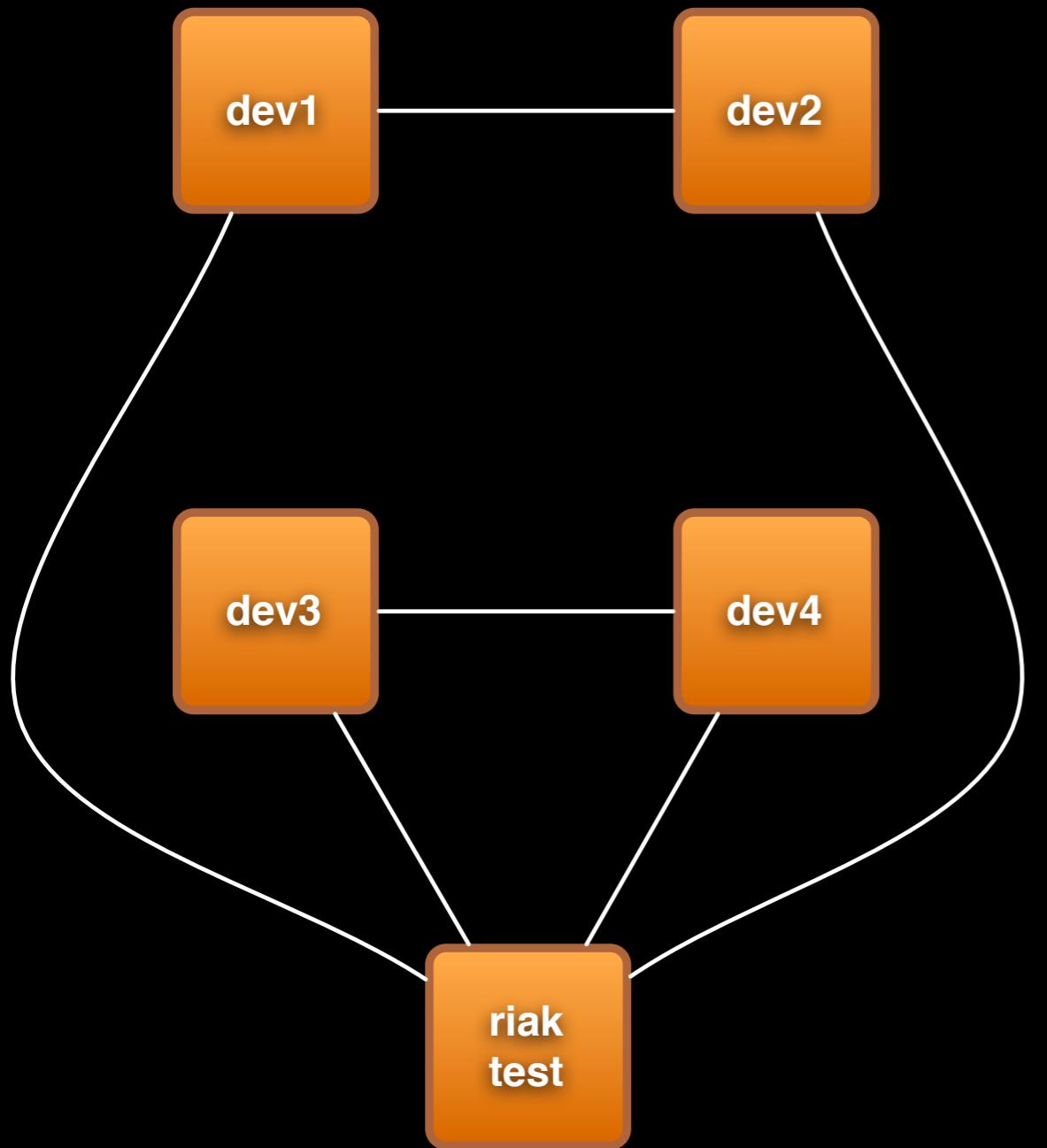


The Cookie!



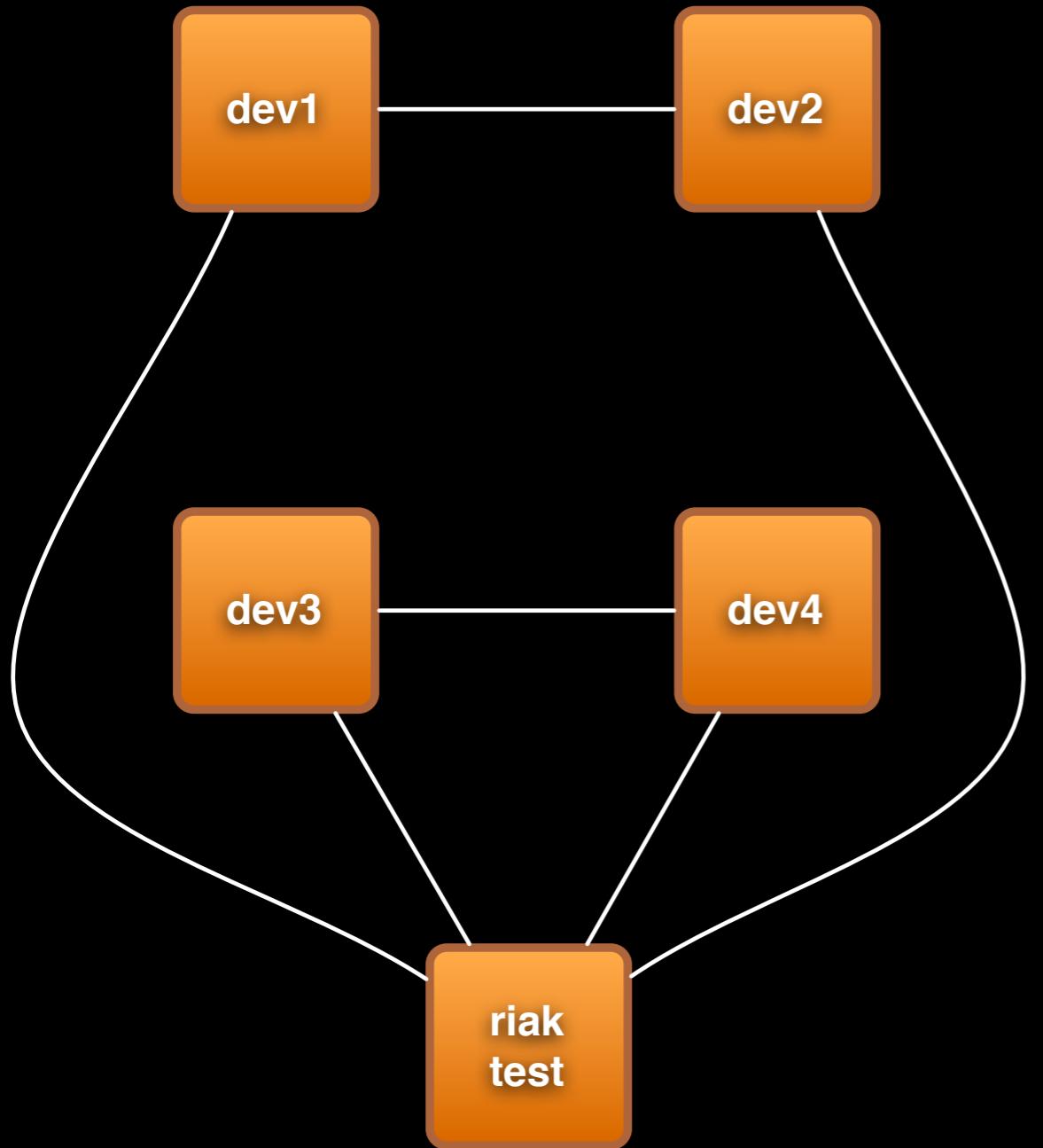


reverse the cookie on
dev3 and dev4



reverse the cookie on
dev3 and dev4

disconnect dev3 and dev4
from dev1 and dev2



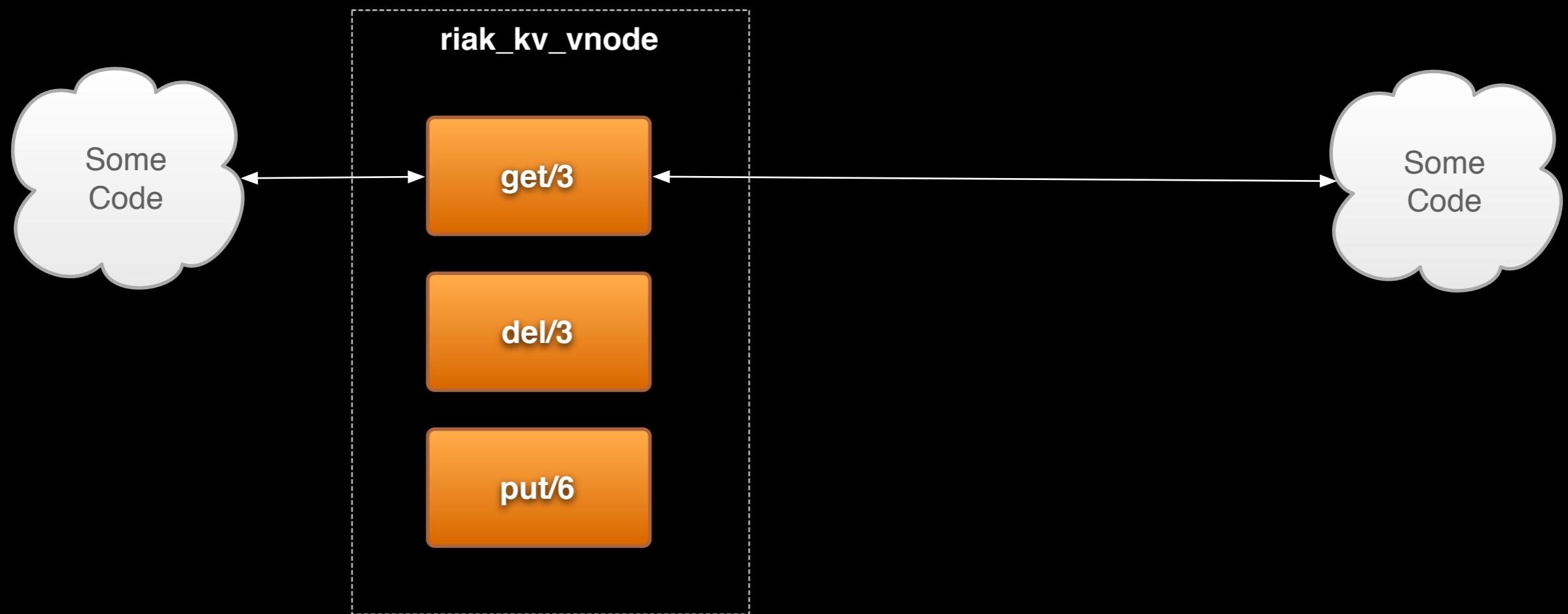
reverse the cookie on
dev3 and dev4

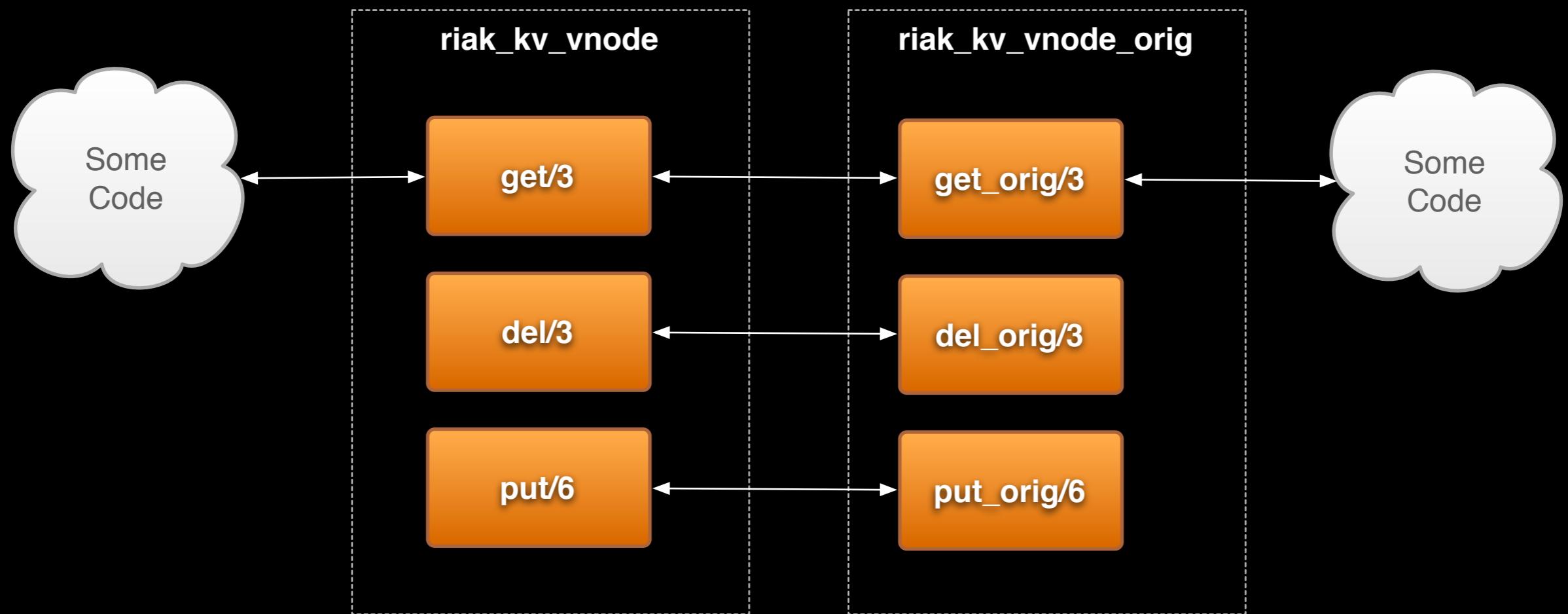
disconnect dev3 and dev4
from dev1 and dev2

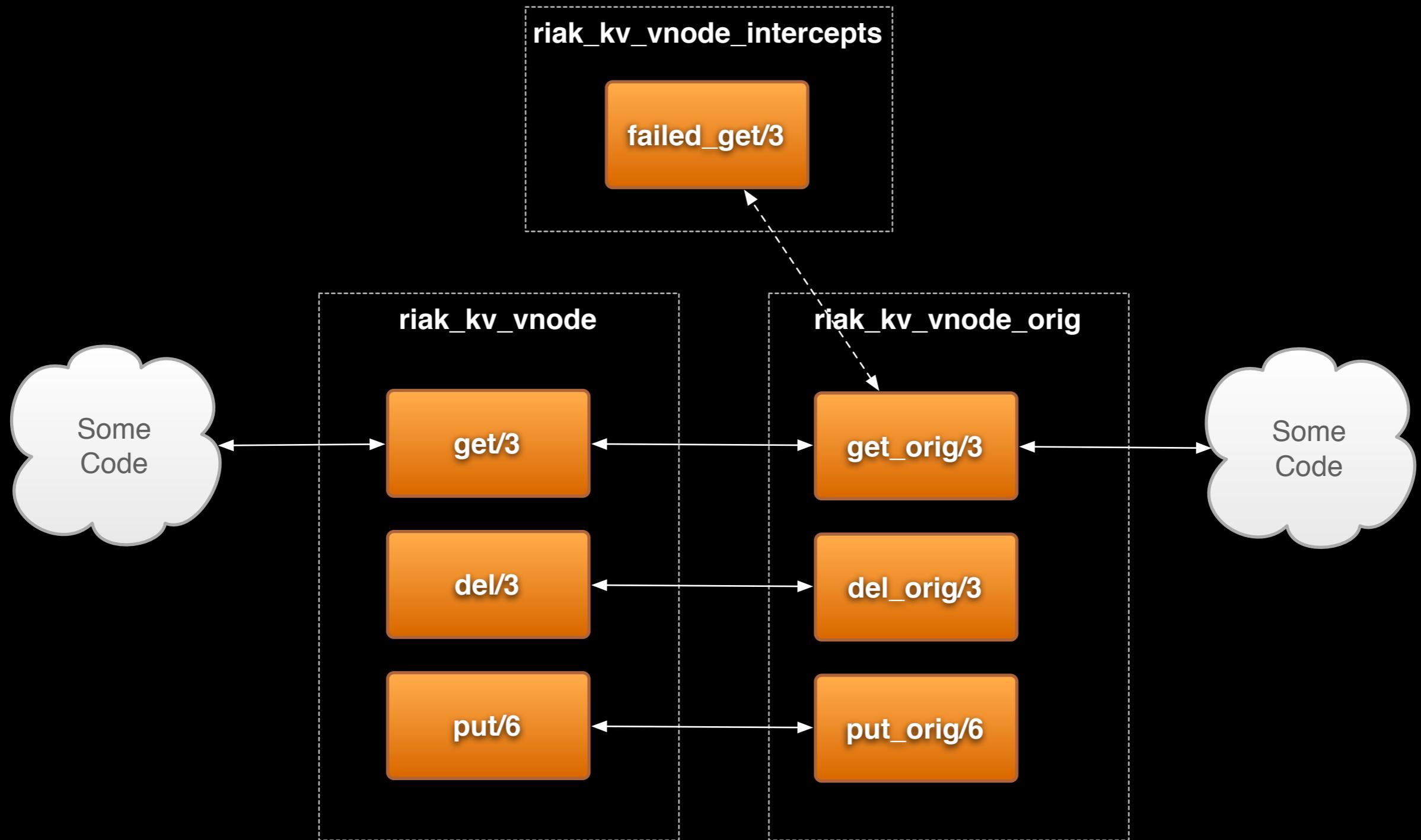
leave riak_test node
connected to all 4

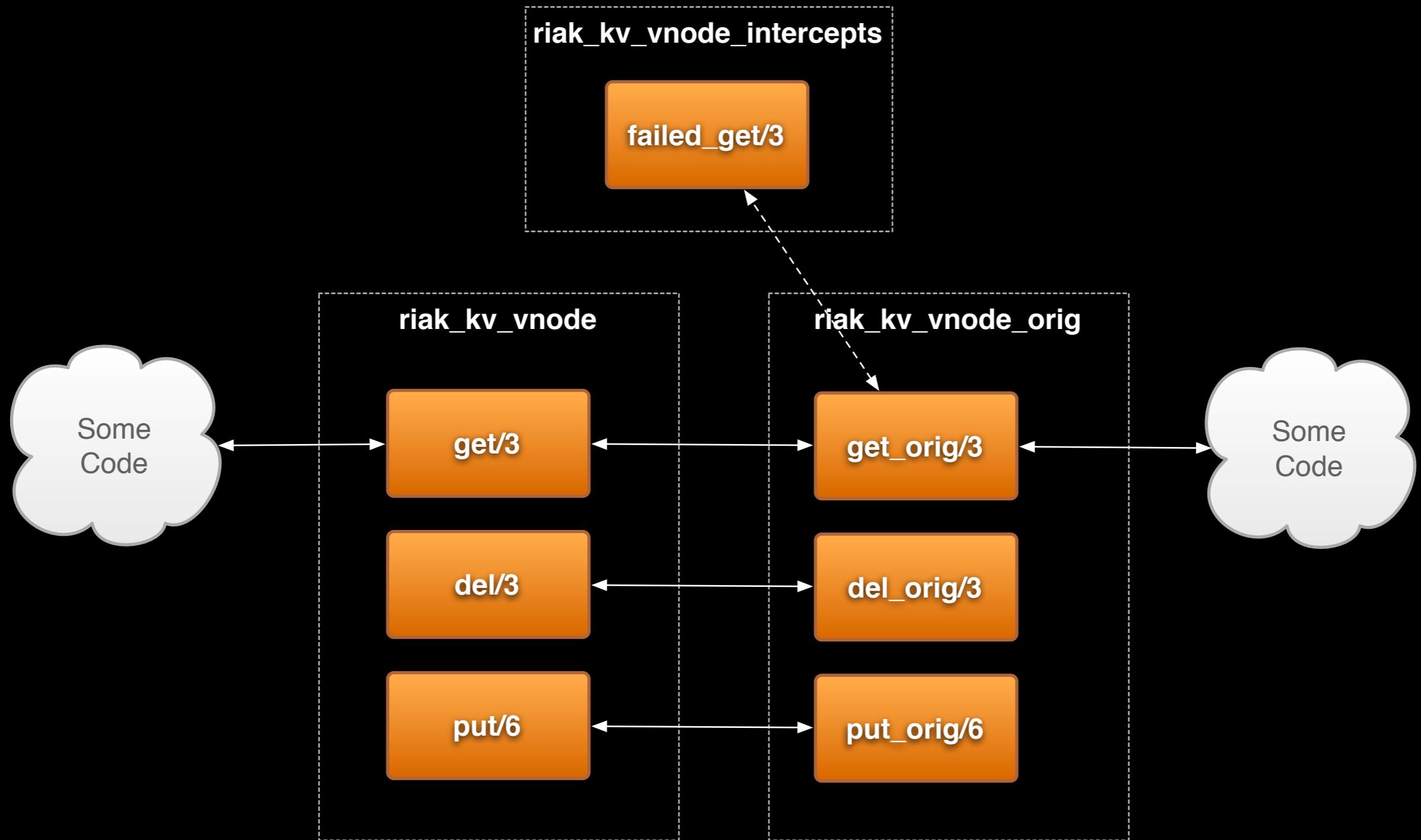
Intercepts



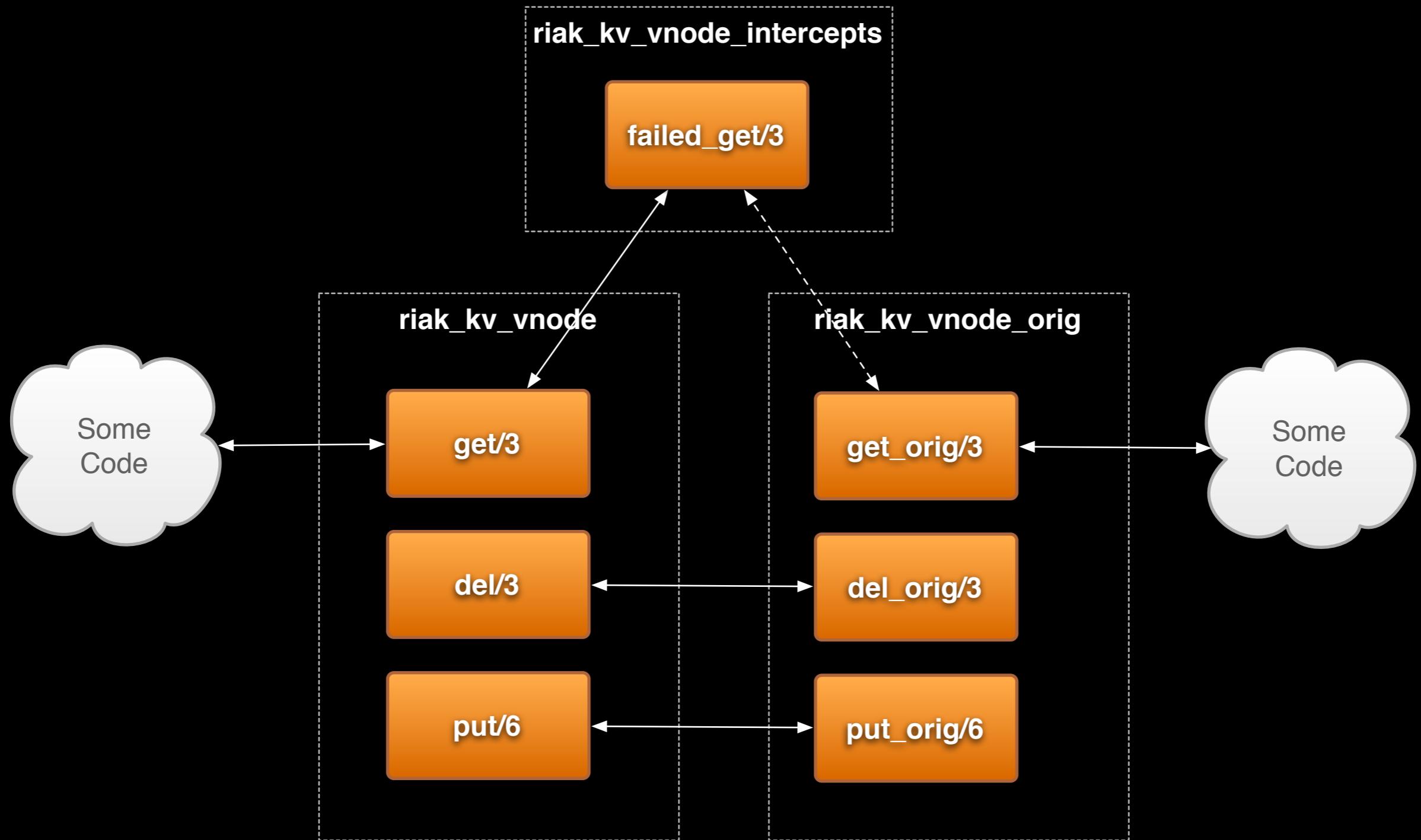








```
rt_intercept:add(Node,  
  {riak_kv_vnode, [{get,3}, failed_get]})
```

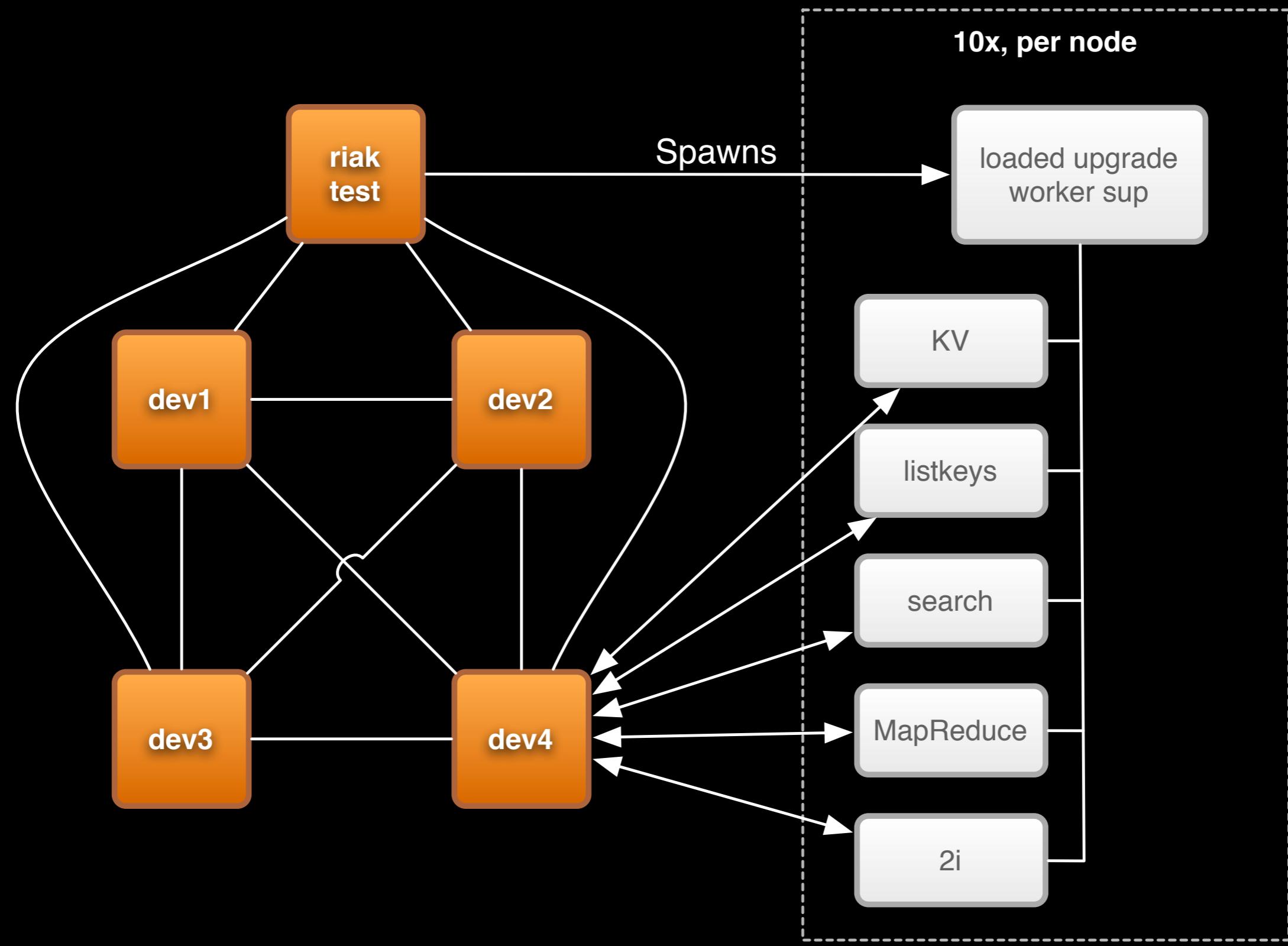


`rt_intercept:add(Node,
 {riak_kv_vnode, [{get,3}, failed_get]})`



ENOTENUFPIXELS

My Melty-Faced Spock Moment





Joe DeVivo

@joedevivo

Come to my talk and I'll explain why I was opening more file descriptors than Mac OS X's erlang VM could handle: erlang-factory.com/conference/SFB...

[Reply](#) [Delete](#) [Favorite](#) [More](#)

4

RETWEETS

2

FAVORITES



6:03 PM - 4 Feb 13

**Where do we
go from here?**

Successes!

Successes!

- less people spend less hours on release validation

Successes!

- less people spend less hours on release validation
- For Erlangers by Erlangers

Successes!

- less people spend less hours on release validation
- For Erlangers by Erlangers
- Features can be developed with integration testing

Failure was an

Failure was an

- Release Validation took a long calendar time

Failure was an

- Release Validation took a long calendar time
- A process problem Nightlies can't solve

Failure was an

- Release Validation took a long calendar time
- A process problem Nightlies can't solve
- Performance Testing still Manual

Failure was an

- Release Validation took a long calendar time
- A process problem Nightlies can't solve
- Performance Testing still Manual
- Test Environment can cause issues with results, giving us a “giddyup who cried wolf”

**tl;dr best release
yet; will only get
better**

Thanks!

- Joe Blomstedt
- Sean Cribbs
- Chris Meiklejohn
- Jared Morrow
- John Newman
- Dave “Chest Burster” Parfitt
- Engel Sanchez
- Ryan Zezeski

And many more!

QUESTIONS?



QUESTIONS?



WHO'S TAKING

A photograph of a woman with curly hair, wearing a shiny, sequined, multi-colored dress (blue, green, yellow) and a matching headband. She is singing into a black microphone. The background is dark with some stage lights.

CLAP IF YOU LOVE
ME



CLAP IF YOU LOVE
JOE DEVIVO