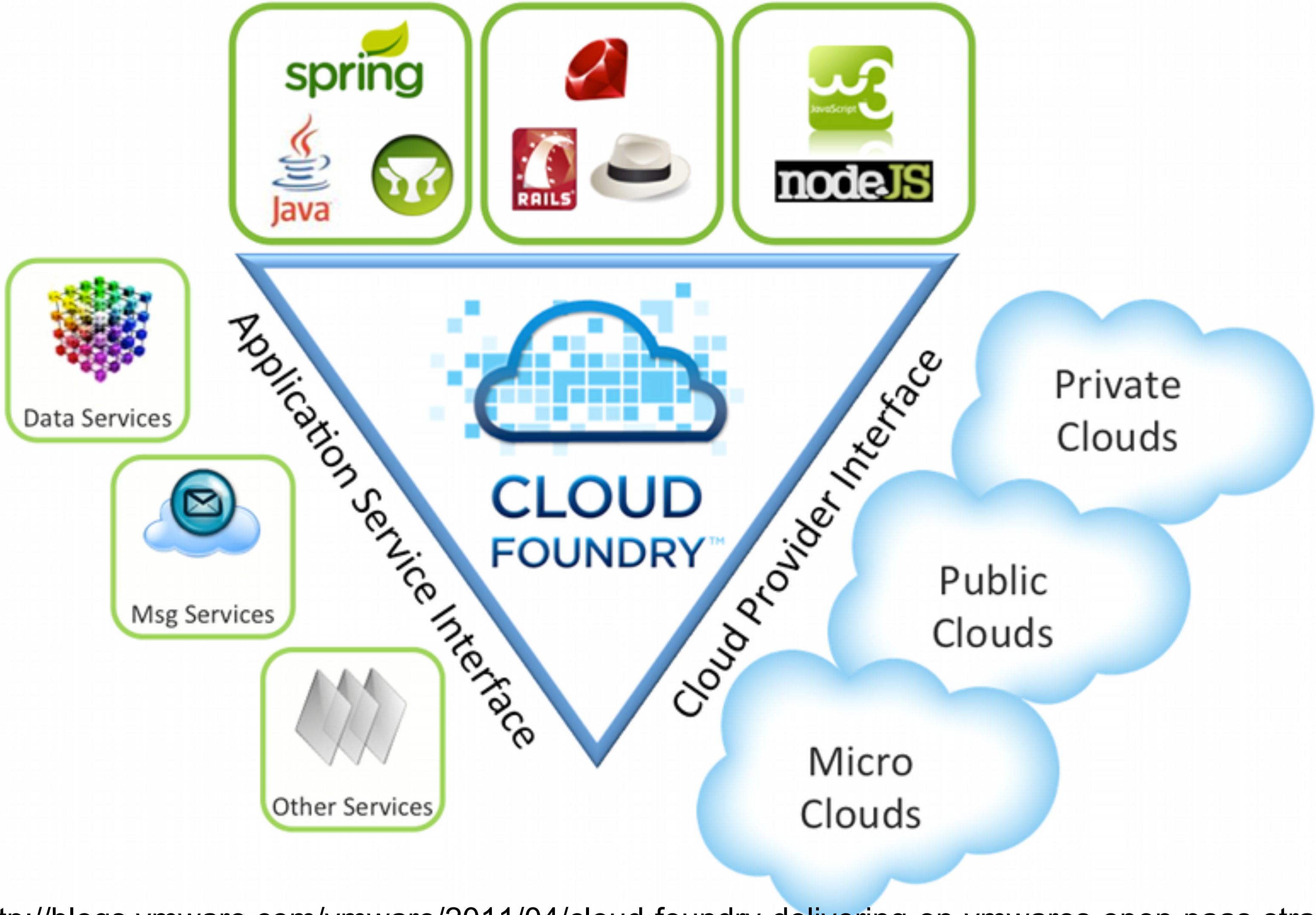
Go testing libraries

Stephan Hagemann
Pivotal Labs, Boulder

https://github.com/shageman/gotestit

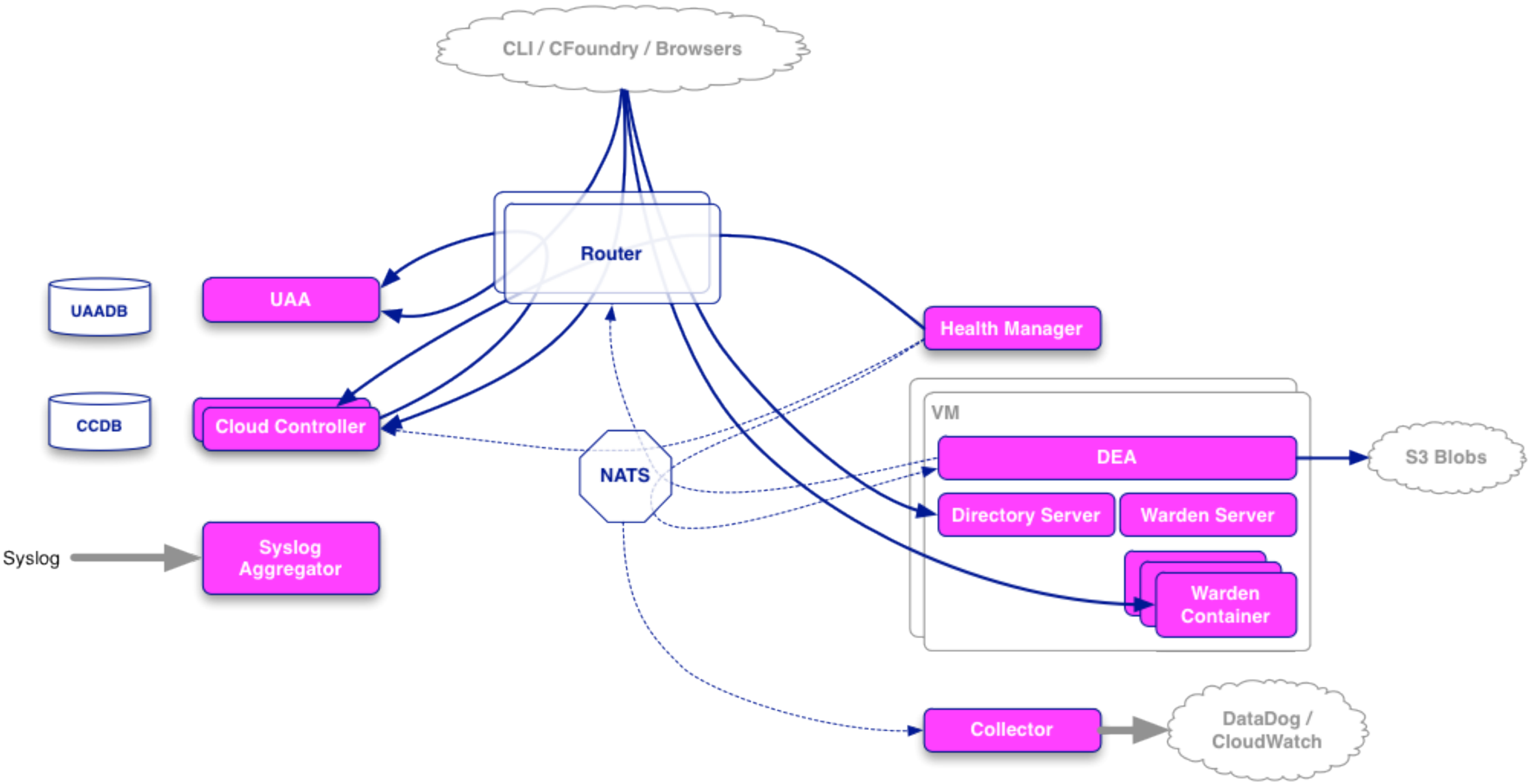
Background

CloudFoundry



http://blogs.vmware.com/vmware/2011/04/cloud-foundry-delivering-on-vmwares-open-paas-strategy.html

CloudFoundry Architecture



http://docs.cloudfoundry.com/docs/running/architecture/

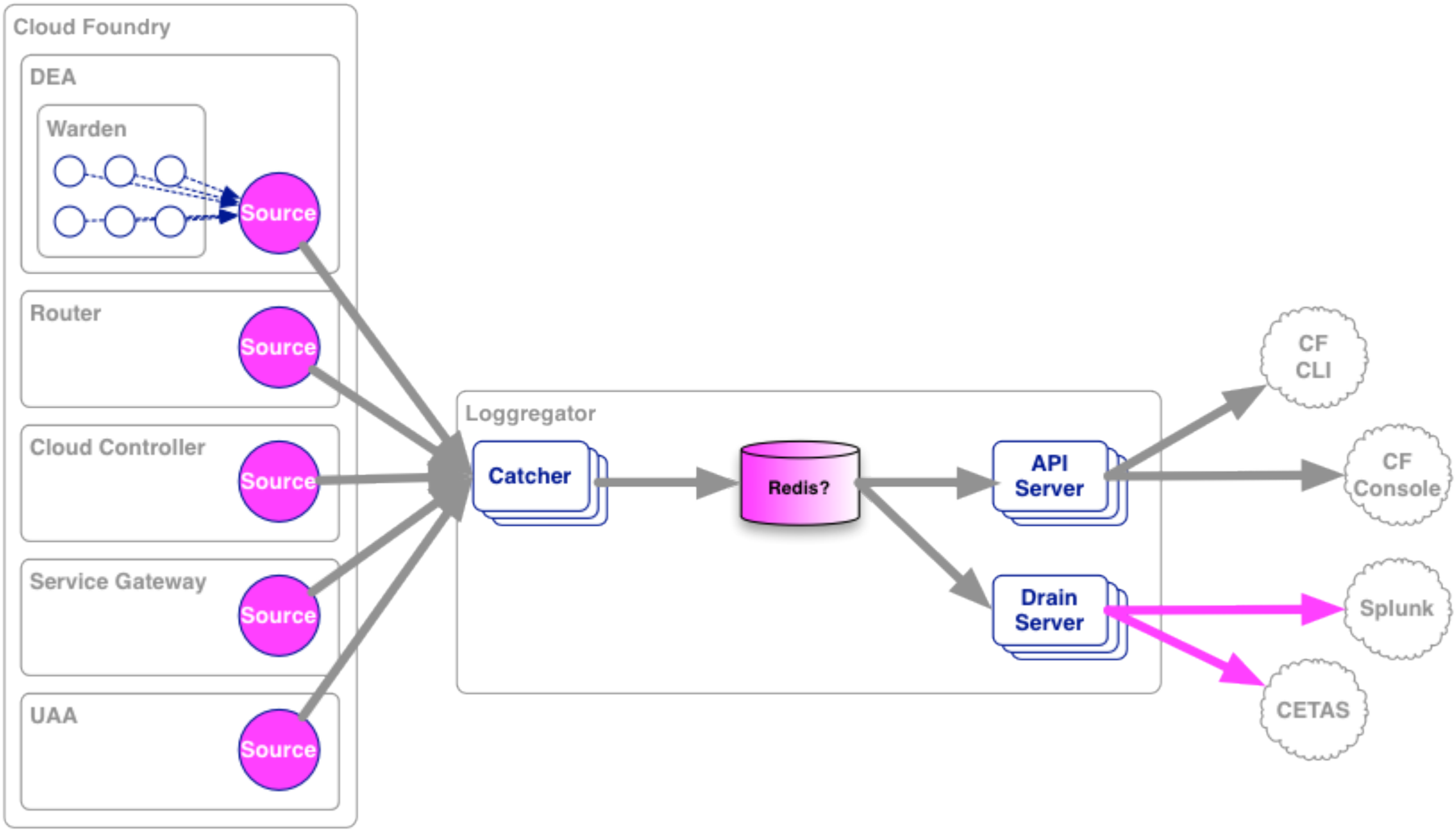
The Loggregator project

Loggregator is the user application logging subsystem for Cloud Foundry.

Loggregator allows users to:

- Tail their application logs.
- Dump a recent set of application logs (where recent is on the order of an hour).
- Continually drain their application logs to 3rd party log archive and analysis services.

Loggregator Architecture



https://github.com/cloudfoundry/loggregator

Libraries

Tests vs Specs

```
class TestMeme < Minitest::Test
  def setup
    @meme = Meme.new
  end

  def test_that_kitty_can_eat
    assert_equal "OHAI!", @meme.i_can_has_cheezburger?
  end

  def test_that_it_will_not_blend
    refute_match /^no/i, @meme.will_it_blend?
  end

  def test_that_will_be_skipped
    skip "test this later"
  end
end
```

```
describe Meme do
  before do
    @meme = Meme.new
  end

  describe "when asked about cheeseburgers" do
    it "must respond positively" do
      @meme.i_can_has_cheezburger?.must_equal "OHAI!"
    end
  end

  describe "when asked about blending possibilities" do
    it "won't say no" do
      @meme.will_it_blend?.wont_match /^no/i
    end
  end
end
```

https://github.com/seattlerb/minitest

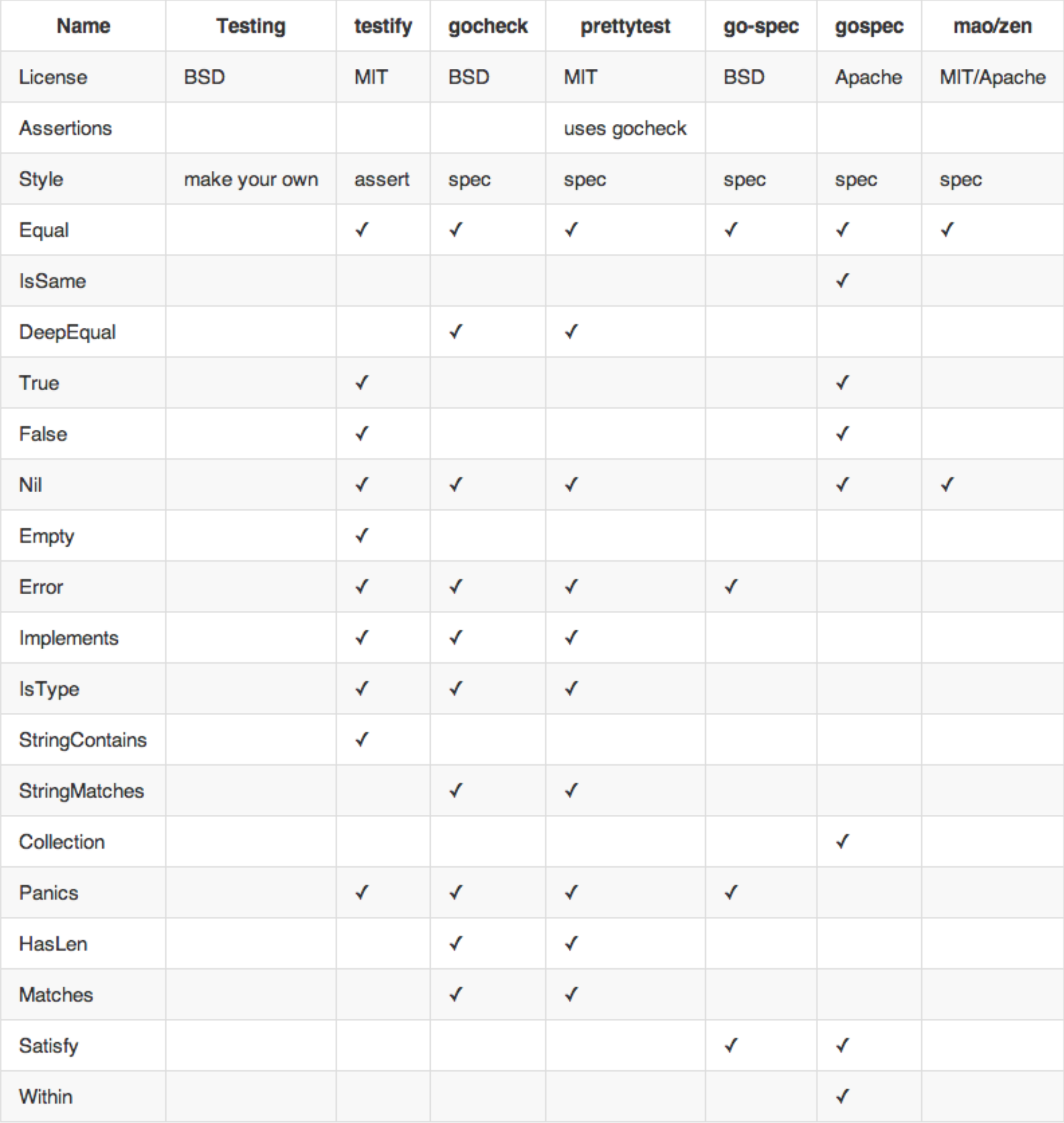
Go testing libraries

testing
testify
gocheck
prettytest
go-spec
gospec
mao
zen

Licences

testing	BSD
testify	MIT
gocheck	BSD
prettytest	MIT
go-spec	BSD
gospec	Apache
mao	MIT
zen	Apache

Matchers



Name	Testing	testify	gocheck	prettytest	go-spec	gospec	mao/zen
License	BSD	MIT	BSD	MIT	BSD	Apache	MIT/Apache
Assertions				uses gocheck			
Style	make your own	assert	spec	spec	spec	spec	spec
Equal		✓	✓	✓	✓	✓	✓
IsSame						✓	
DeepEqual			✓	✓			
True		✓				✓	
False		✓				✓	
Nil		✓	✓	✓		✓	✓
Empty		✓					
Error		✓	✓	✓	✓		
Implements		✓	✓	✓			
IsType		✓	✓	✓			
StringContains		✓					
StringMatches			✓	✓			
Collection						✓	
Panics		✓	✓	✓	✓		
HasLen			✓	✓			
Matches			✓	✓			
Satisfy					✓	✓	
Within						✓	

Sample tests

https://github.com/shageman/gotestit

Summary

testing	really basic, direct use tedious
testify	
gocheck	
prettytest	
~~go-spec~~	
gospec	
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	
prettytest	
~~go-spec~~	
gospec	
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	
~~go-spec~~	
gospec	
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	suites, pretty
~~go-spec~~	
gospec	
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	suites, pretty
~~go-spec~~	doesn't work
gospec	
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	suites, pretty
~~go-spec~~	doesn't work
gospec	Describe, expect, have to state all tests
mao	
zen	

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	suites, pretty
~~go-spec~~	doesn't work
gospec	Describe, expect, have to state all tests
mao	-
zen	desc, it, expect, basic assertions

Summary

testing	really basic, direct use tedious
testify	lots of useful assertions
gocheck	suites, before, after, lots of assertions
prettytest	suites, pretty
~~go-spec~~	doesn't work
gospec	Describe, expect, have to state all tests
mao	-
zen	desc, it, expect, basic assertions

Thanks!

travis

```
go test -- race
```

shared state

parallel