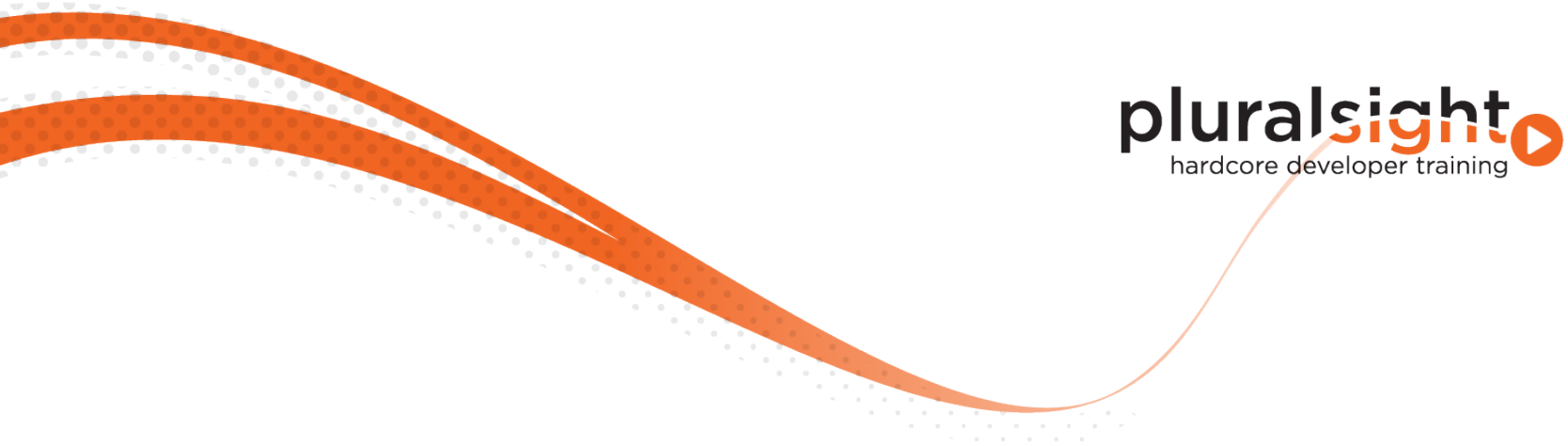
Testing Clientside JavaScript
SinonJS

Joe Eames

@josepheames

www.testdrivenjs.com

www.pluralsight.com



Sinon js

- Created by Christian Johansen
- http://tddjs.com/
- Comprehensive mocking library
- http://sinonjs.org/

Sinon spies

- **Purposes**
  - Provide a test double for a single function
  - Watch an existing method of an object
- **Created with sinon.spy method**

Sinon spy API

- called
- calledOnce
- calledTwice
- calledThrice
- firstCall
- secondCall
- thirdCall
- lastCall
- calledBefore(spy)
- calledAfter(spy)

- calledOn(obj)
- alwaysCalledOn(obj)
- calledWith(args…)
- alwaysCalledWith(args…)
- calledWithExactly(args…)
- alwaysCalledWithExactly(args…)
- notCalledWith(args…)
- neverCalledWith(args…)
- calledWithMatch(args…)
- alwaysCalledWithMatch(args…)
- notCalledWithMatch(args…)
- neverCalledWithMatch(args…)

Sinon spy API Part 2

- calledWithNew
- threw
- threw(“string”)
- threw(obj)
- alwaysThrew
- alwaysThrew(“string”)
- alwaysThrew(obj)
- returned(obj)
- alwaysReturned(obj)

Sinon spy API Part 3

- getCall(n)
- thisValues
- args
- exceptions
- returnValues
- reset
- printf

Sinon call API

- calledOn(obj)
- calledWith(args…)
- calledWithExactly(args…)
- calledWithMatch(matches…)
- notCalledWith(args…)
- notCalledWithMatch(args…)
- threw
- threw(“string”)
- threw(obj)
- thisValue
- args
- exception
- returnValue

Sinon Assertions

- called
- notCalled
- calledOnce
- calledTwice
- calledThrice
- callCount(spy, num)
- callOrder(spy1, spy2…)
- calledOn(spy, obj)
- alwaysCalledOn(spy, obj)
- calledWith(spy, args…)
- alwaysCalledWith(spy, args…)
- neverCalledWith(spy, args…)
- calledWithExactly(spy, args…)
- calledWithMatch(spy, matches...)
- alwaysCalledWithMatch(spy, args...)
- neverCalledWithMatch(spy, args...)
- threw(spy)
- threw(spy, exception)
- alwaysThrew(spy, exception)

Sinon Stubs

- **Spies with pre-programmed behavior**
- **Support the spy API**
- **Contain additional methods to control behavior**
- **Can be anonymous or wrap existing methods**
- **Original functions are not called**
- **Purposes**
  - Control behavior
  - Suppress existing behavior
  - Behavior verification

Creating Sinon Stubs

```
var stub = sinon.stub();
```

```
var stub = sinon.stub(object, “method”);
```

```
var stub = sinon.stub(object, “method”, func);
```

```
var stub = sinon.stub(object);
```

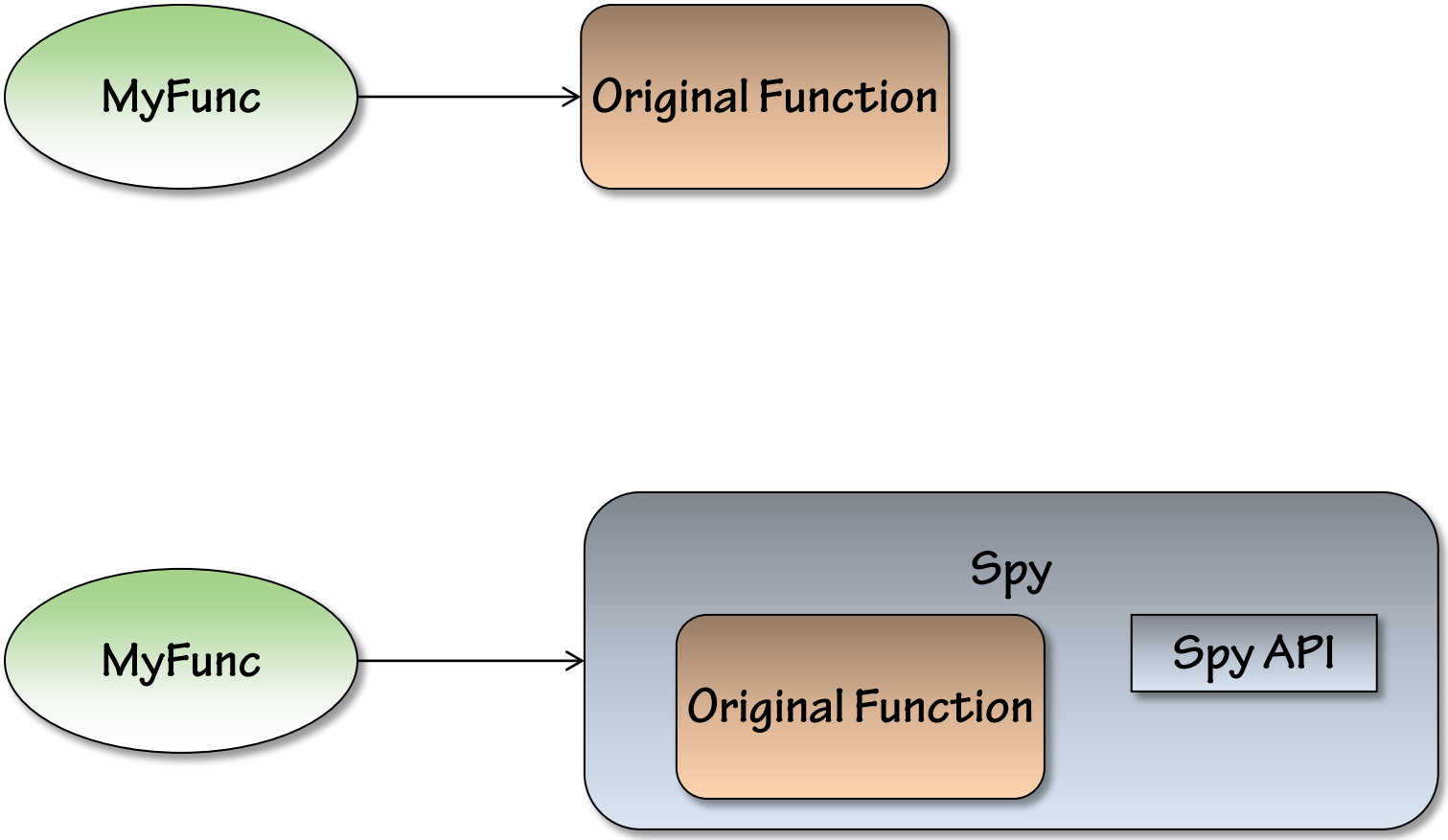
Sinon Stub API

- returns(obj)
- throws
- throws("type")
- throws(obj)
- withArgs(args...)
- returnsArg(index)
- callsArg(index)
- callsArgOn(index, context)
- callsArgWith(index, args...)
- callsArgOnWith(index, context, args...)

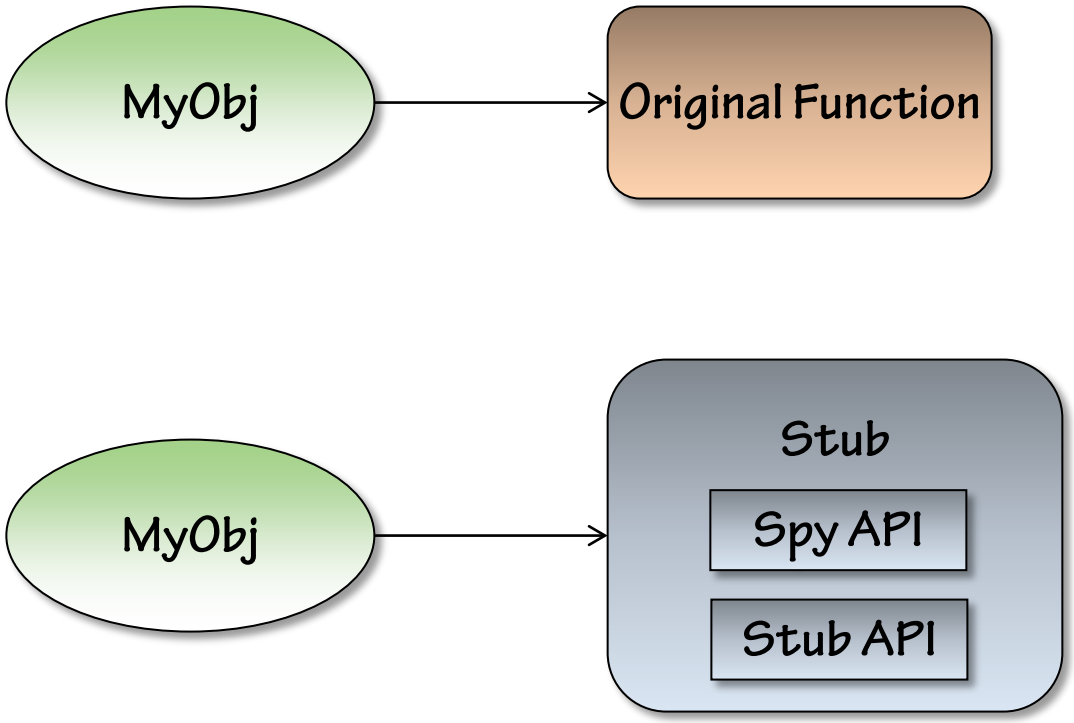
Sinon Mocks

- **Contain pre-programmed expectations**
- **Fail tests if expectations are not met**
- **Like stubs except assertions are stated before instead of after**
- **Guidelines**
  - Only use one mock object
  - Minimize use of expectations
  - Minimize use of assertions external to the mock

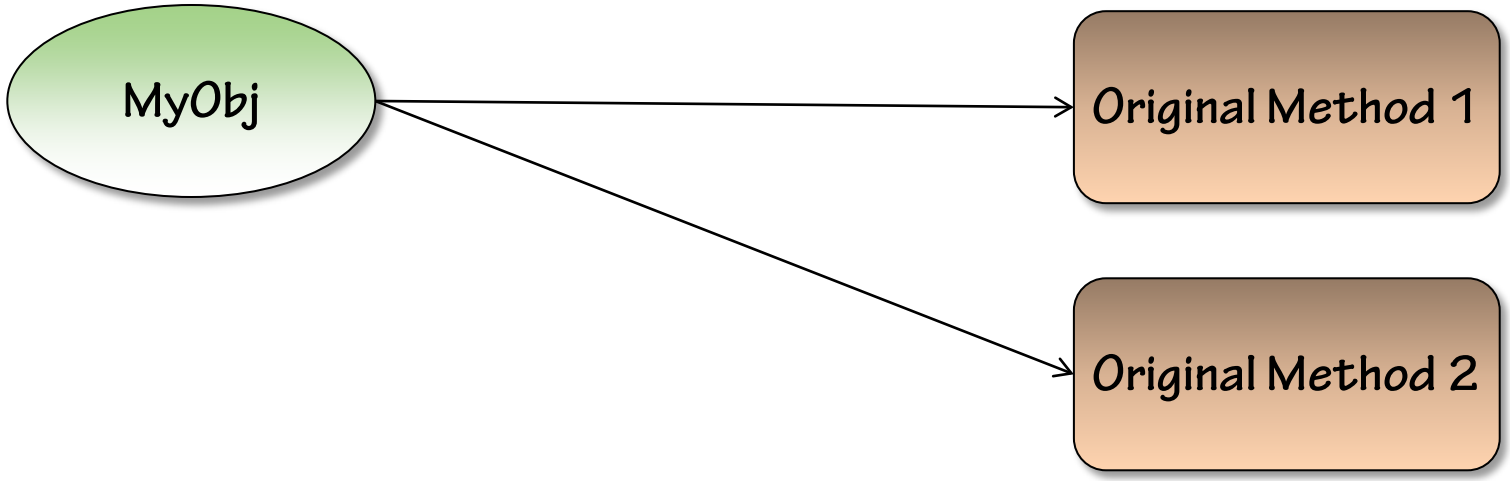
Sinon Spy Architecture



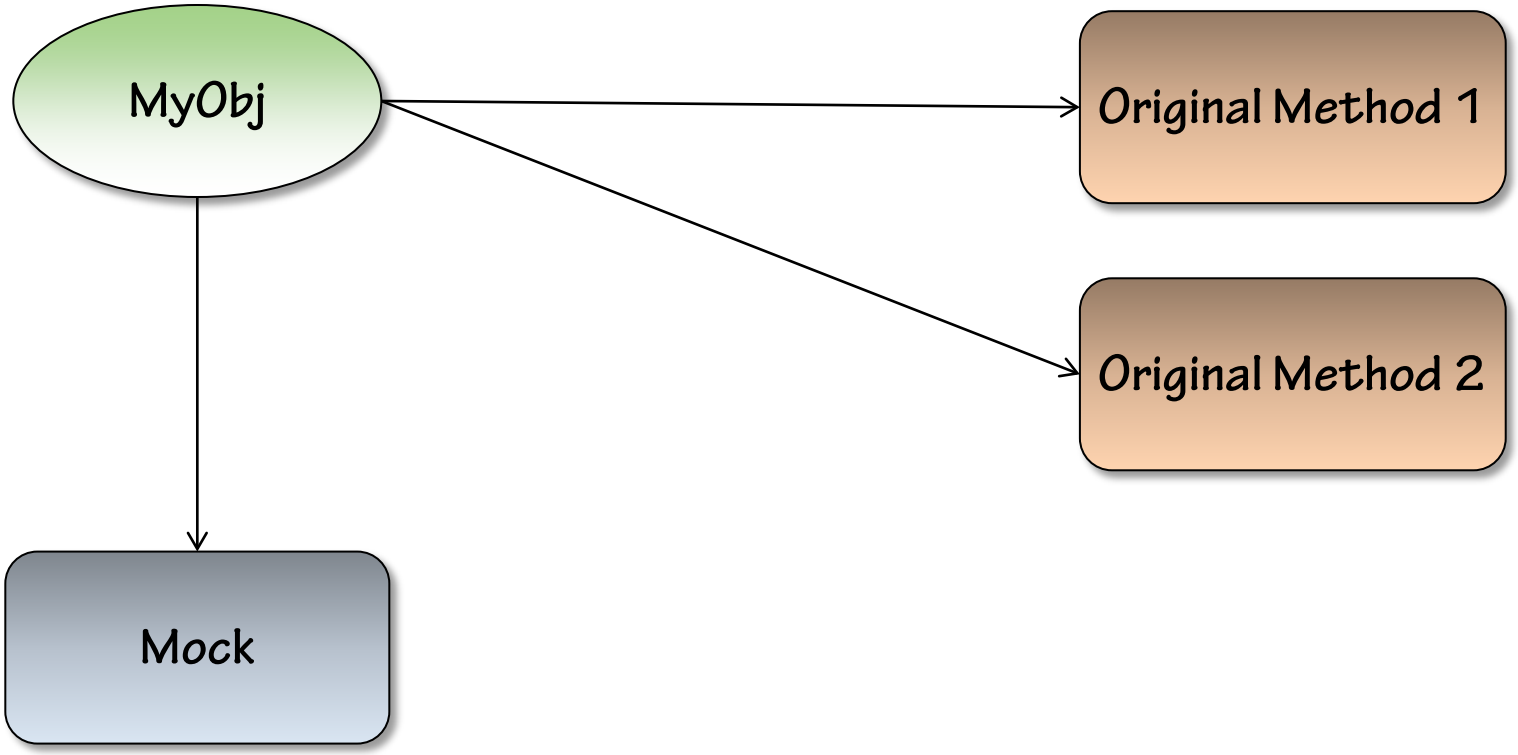
Sinon Stub Architecture



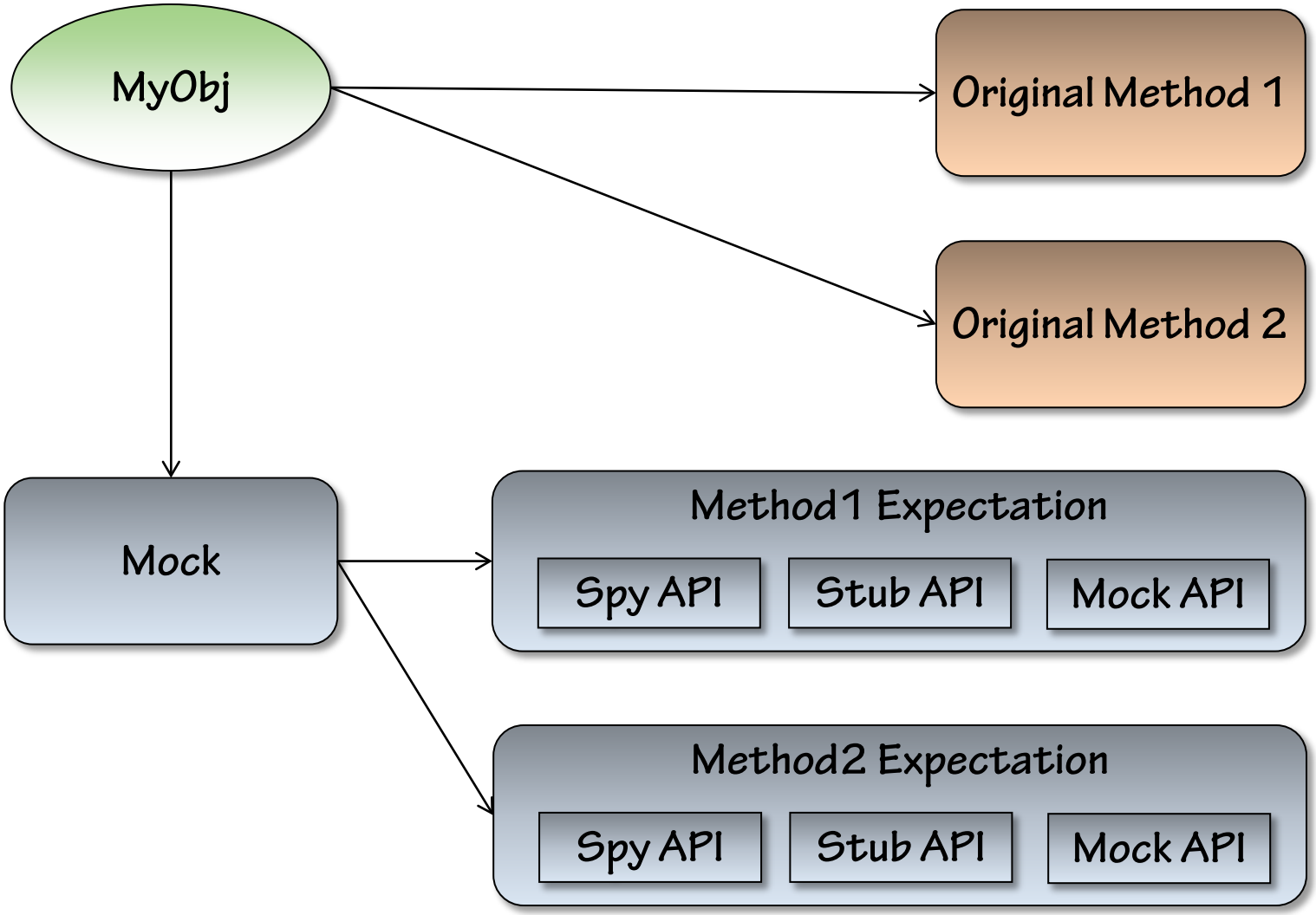
Sinon Mock Architecture



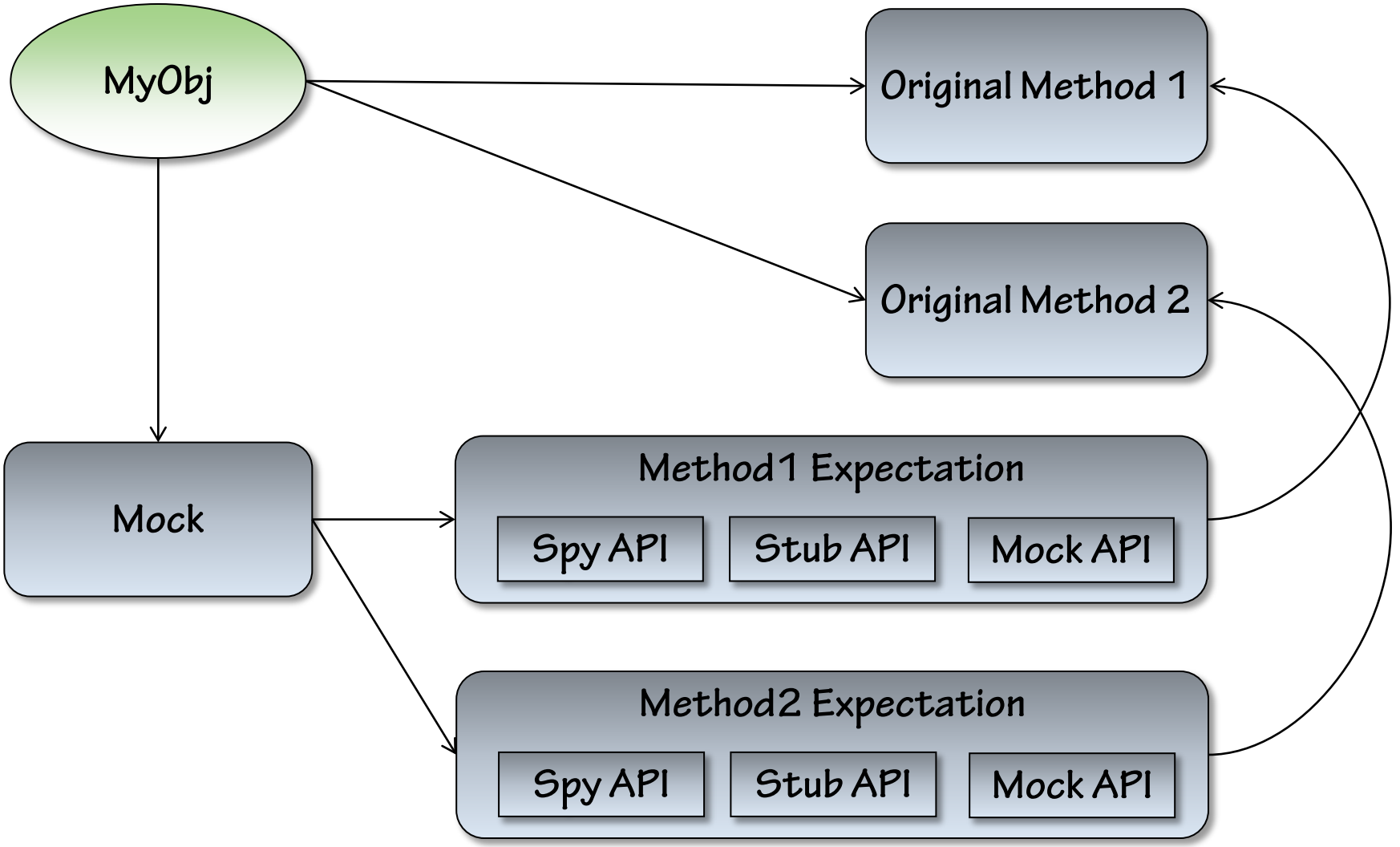
Sinon Mock Architecture



Sinon Mock Architecture



Sinon Mock Architecture



Sinon Mocks Usage

```
var myMock = sinon.mock(object);
var myExpectation1 = myMock.expects(“method1”);
// set expectations on myExpectation1
```

Sinon Expectation API

- expectation.atLeast(number)
- expectation.atMost(number)
- expectation.never
- expectation.once
- expectation.twice
- expectation.thrice
- expectation.exactly(number)
- expectation.withArgs(args...)
- expectation.withExactArgs(args...)
- expectation.on(obj)
- expectation.verify

Sinon Matchers

- Used for fuzzy matching of arguments

Using Sinon Matchers

```
var spy = sinon.spy();
spy(3);
spy.calledWithMatch(sinon.match.number);
```

Sinon Matchers

- sinon.match(number)
- sinon.match(string)
- sinon.match(regexp)
- sinon.match(object)
- sinon.match(function)
- sinon.match.any
- sinon.match.defined
- sinon.match.truthy
- sinon.match.falsy
- sinon.match.bool
- sinon.match.number
- sinon.match.string

Sinon Matchers Part 2

- sinon.match.object
- sinon.match.func
- sinon.match.array
- sinon.match.regexp
- sinon.match.date
- sinon.match.same(ref)
- sinon.match.typeOf(type)
- sinon.match.instanceOf(type)
- sinon.match.has(property[,expectation])
- sinon.match.hasOwn(property[, expectation])

Custom Sinon Matchers

```
var lessThan100 = sinon.match(function(value) {
   return value < 100;
}, “less than 100”)
```

Combining Sinon Matchers

- Use “and” and “or”

```
sinon.match.func.or(sinon.match.string)
```

Sinon Matchers Summary

- Used for fuzzy matchers of arguments.
- Very comprehensive
- Custom matchers give ultimate flexibility
- Can be combined with “and” and “or”

Faking Timers

- Allows control of setTimeout and setInterval
- When used with IE, requires sinon-ie
- Provides a clock object to control the passage of time
- Allows control over Dates, such as Date.now
- Works with jQuery animations

Faking XMLHttpRequest

- **Provides a fake implementation of XMLHttpRequest**
- **Allows each request to be inspected and responded to**
- **Requires sinon-ie when working in IE**
- **Provides two interfaces**
  - useFakeXmlHttpRequest
  - fakeServer

useFakeXmlHttpRequest API

```
var requests = [];
var xhr = sinon.useFakeXMLHttpRequest();
xhr.onCreate = function(req) {
   requests.push(req);
};
…
xhr.restore();
```

FakeXMLHttpRequest API

- request.url
- request.method
- request.requestHeaders
- request.requestBody
- request.status
- request.statusText
- request.username
- request.password
- request.responseXml
- request.responseText
- request.getResponseHeader(header)
- request.getAllResponseHeaders()

FakeXMLHttpRequest response API

- request.respond(status, headers, body)
- request.setResponseHeaders(object)
- request.setResponseBody(body)

Fake Server

```
var server = sinon.fakeServer.create();
server.respondWith(response);
// make requests
server.restore();
```

Fake Server API

- server.respondWith(response)
- server.respondWith(url, response)
- server.respondWith(method, url, response)
- server.respondWith(urlRegExp, response)
- server.respondWith(method, urlRegExp, response)
- server.respond()

Sandboxing

- **Restores global objects after test**
- **Restores:**
  - Spies
  - Stubs
  - Mocks
  - Fake Timers
  - Fake XHR
- **Implemented with**
  - sinon.sandbox.create
  - sinon.test()

Summary

- **Sinon Objects**
  - Spies
  - Stubs
  - Mocks
- **Matchers**
- **Mock Timers**
- **Mock XHR**
  - useFakeXMLHttpRequest
  - fakeServer
- **Sandboxing**