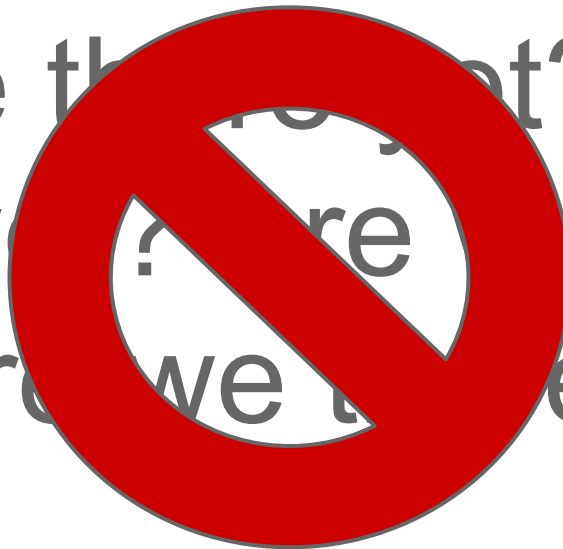


Promise Pattern

for asynchronous
JavaScript

What is asynchronous programming?

Are we there yet? Are we
there yet? Are we there
yet? Are we there yet? ...



**When do you use
asynchronous
programming?**

Performance

User input, network I/O
(AJAX), computational
processing, rendering, ...

KISS

(keep it simple, stupid!)

Separation of concerns,

Less boilerplate,

Focus on business logic

Asynchronous Patterns

Callbacks,
Events,
Promises

Callbacks

```
$.get('foobar.html', function(data) {  
    $('.result').html(data);  
    alert('Load was performed.');
```

Problem?

- Only one callback at a time
- Nested callbacks: turtles all the way down

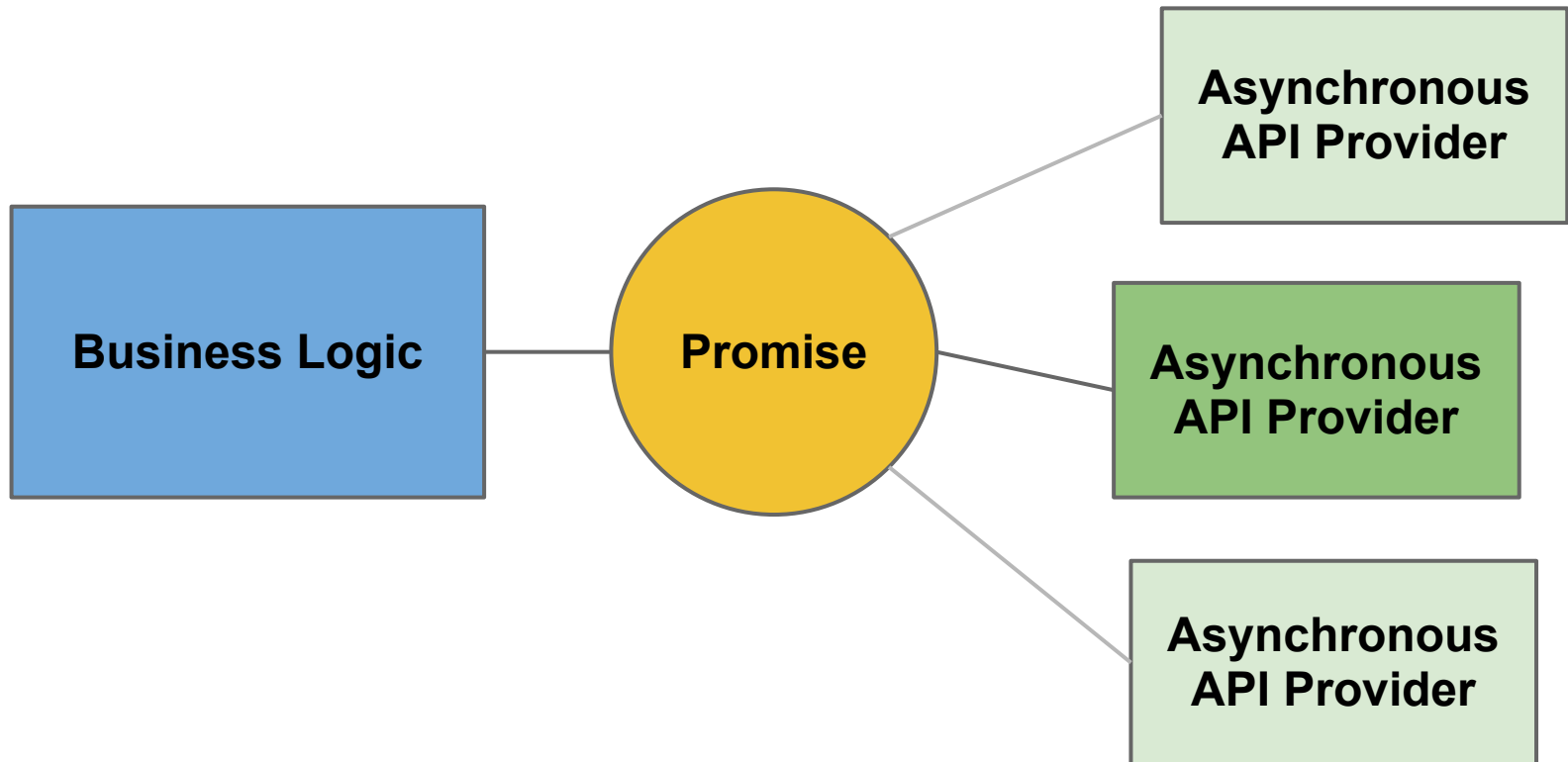
Events

```
$('a.button')  
  .on('click', function(event) {  
    // do something...  
  })  
  .on('click', function(event) {  
    // ... and do something else.  
  });
```

Problem?

- Hard to synchronise multiple events
- No standard APIs

Promises



Business Logic

```
(new Promise)  
  .when(load_asset('maps/lobby.bsp'))  
  .when(load_asset('music/intro.ogg'))  
  .when(choose_name())  
  .then(function() {  
    // everything loaded, start the game...  
  });
```

API Providers

```
function load_asset(path) {  
    return new Promise(function(deferred) {  
        $.get(path, deferred.done);  
    });  
}
```

```
function choose_name() {  
    var valid = /^[a-zA-Z0-9_]+$/  
    return new Promise(function(deferred) {  
        $('form').submit(function() {  
            if(valid.test($('name').val()))  
                deferred.done();  
        });  
    });  
}
```

CommonJS Promise/*

<http://wiki.commonjs.org/wiki/Promises>

The Basics

Business Logic:

- when
- then

API Provider:

- deferred.done

The Tricks

```
.when().when() // parallel chaining  
.then().then() // multiple listeners  
.asap() // Race condition
```

```
// Edge cases
```

```
.deferred.failure(reason)  
.then(success, error)
```

```
// Progress tracking
```

```
.deferred.progress(amount)  
.then(success, error, progress)
```