

The promise and the hack

*Some names and identifying details have been changed
to protect the privacy of individuals.*

“Pour bien apprendre quelque chose, il faut en abuser”

– *@boblito*

Promises

Promises

```
call1( params, function( res0 ){  
  call2( res0, function( res1 ){  
    //.... THIS IS HELL!!  
  } );  
} );
```

Promises

```
call1(  
  
} );
```

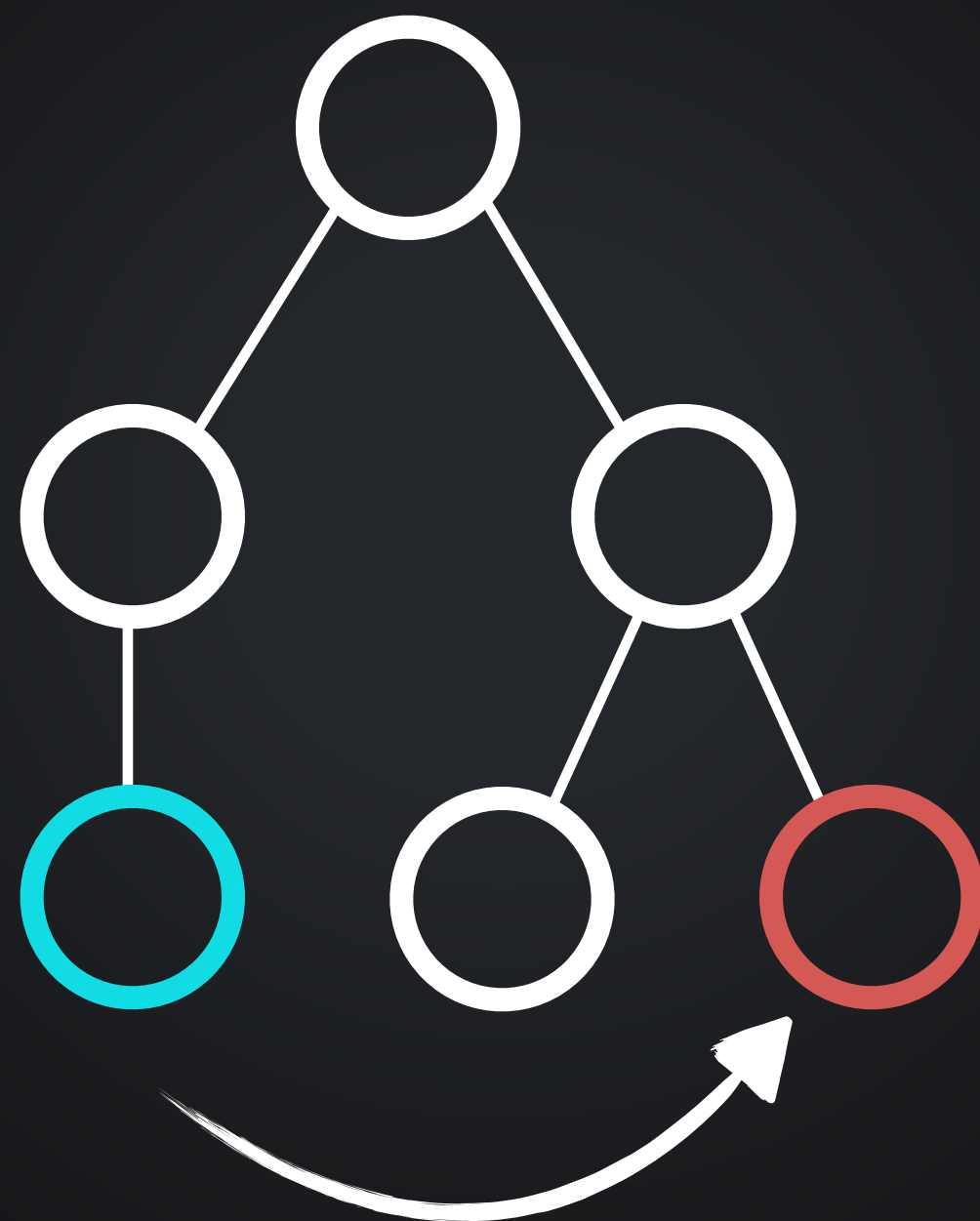
```
call1( params ).then( call2 )  
    .then( function peace(){  
        // At last  
    } );
```

Deferred objects

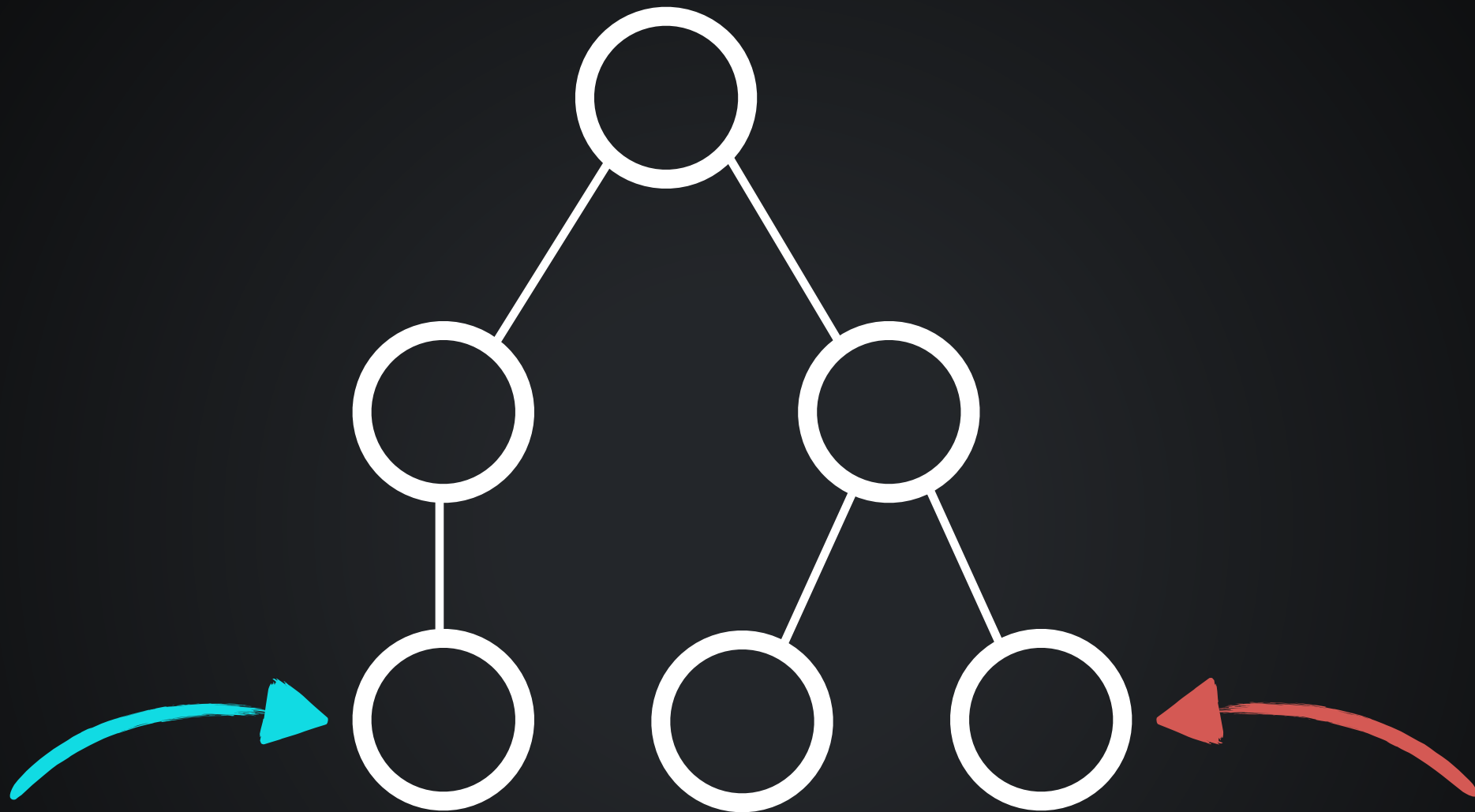
- The origin of all promises
- Remote controls of the promises

Deferred objects

- Get a promise from it
- Set the state of the promise state with reject (error etc) or resolve (value)




```
var actions = {  
  colorIt : Q.defer()  
}
```



```
var self = this;  
actions.colorIt.then( function( colorItF ){  
  self.colorOtherComponent = colorItF;  
} );
```

```
actions.colorIt.fCall( theNewColor );
```

```
actions.colorIt.resolve(  
  function colorBgComponentA( color ){  
    //... PROFIT!  
  }  
)
```

Benefits

- No order required during components initialisation
- Documentation of all the messages that can be exchange in the app
- The root can check if all the actions are resolved

Thanks!

- [@bobyrito](#) (follow me ;))
- Partners in crime : [@42loops](#) && [@greweb](#)
- Check [zanimio](#), [qAjax](#), [qAnimationFrame](#)