

AUTONOMOUS WEB APPLICATIONS WITH



Dan Gebhardt
@dgeb

Wicked Good Ember 2014

Cerebris



AUTONOMOUS WEB APPLICATIONS WITH



Dan Gebhardt
@dgeb

Wicked Good Ember 2014

Cerebris





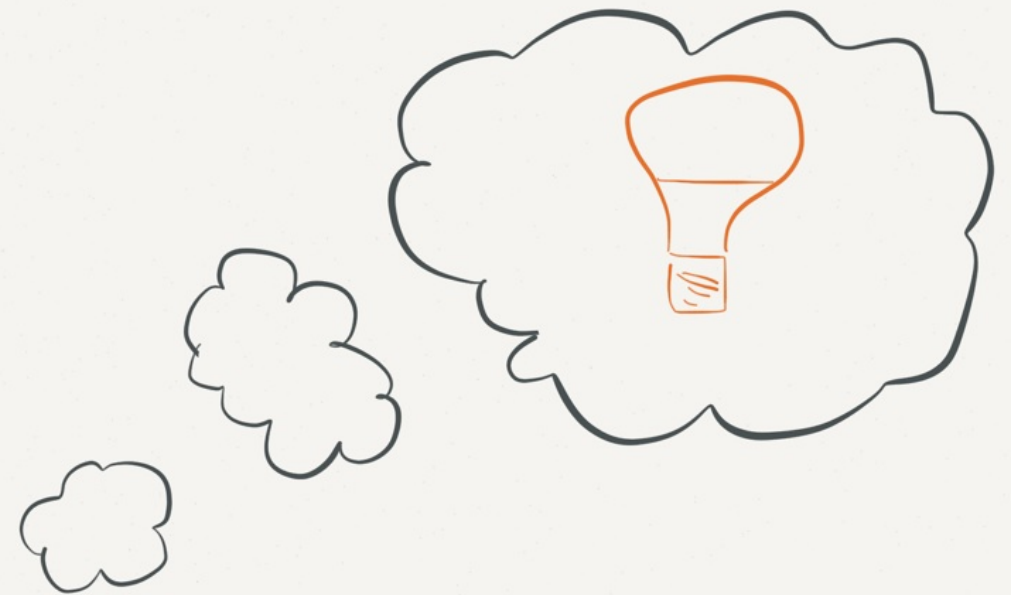
A framework for creating
ambitious web applications

WHAT'S **NEXT** FOR **AMBITIOUS** WEB APPLICATIONS?



AUTONOMY

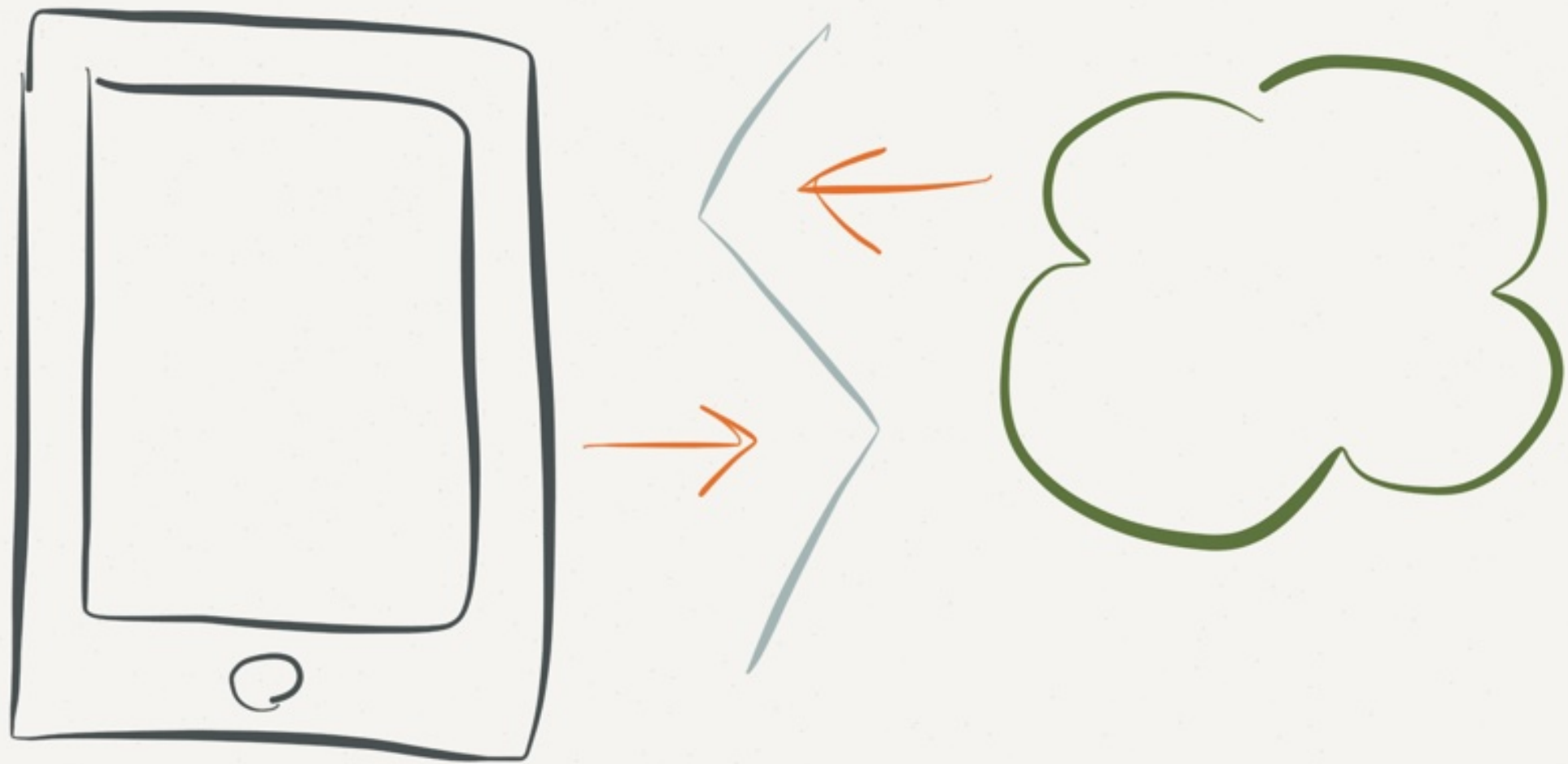
AUTONOMY ?



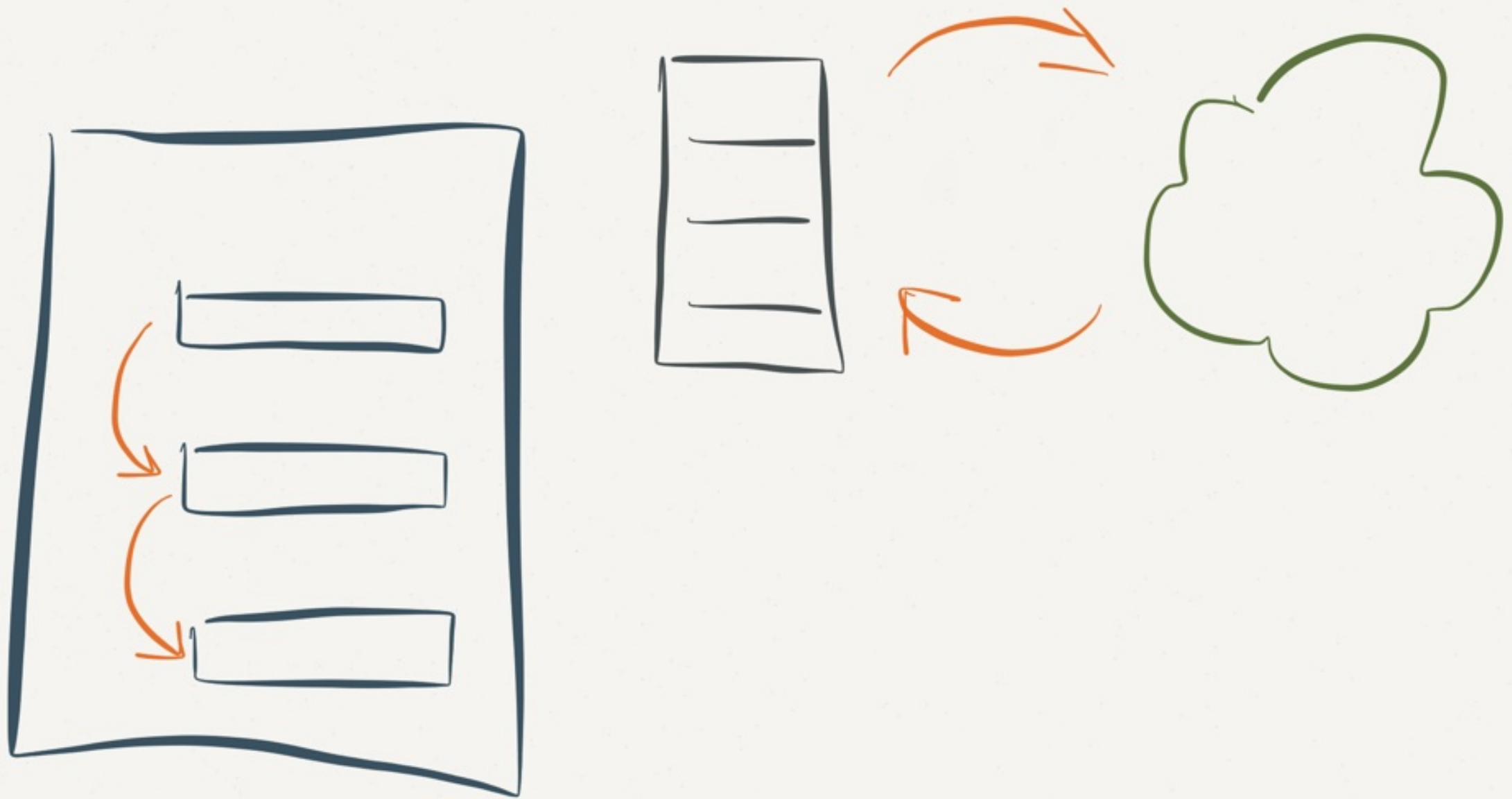
AUTONOMY?



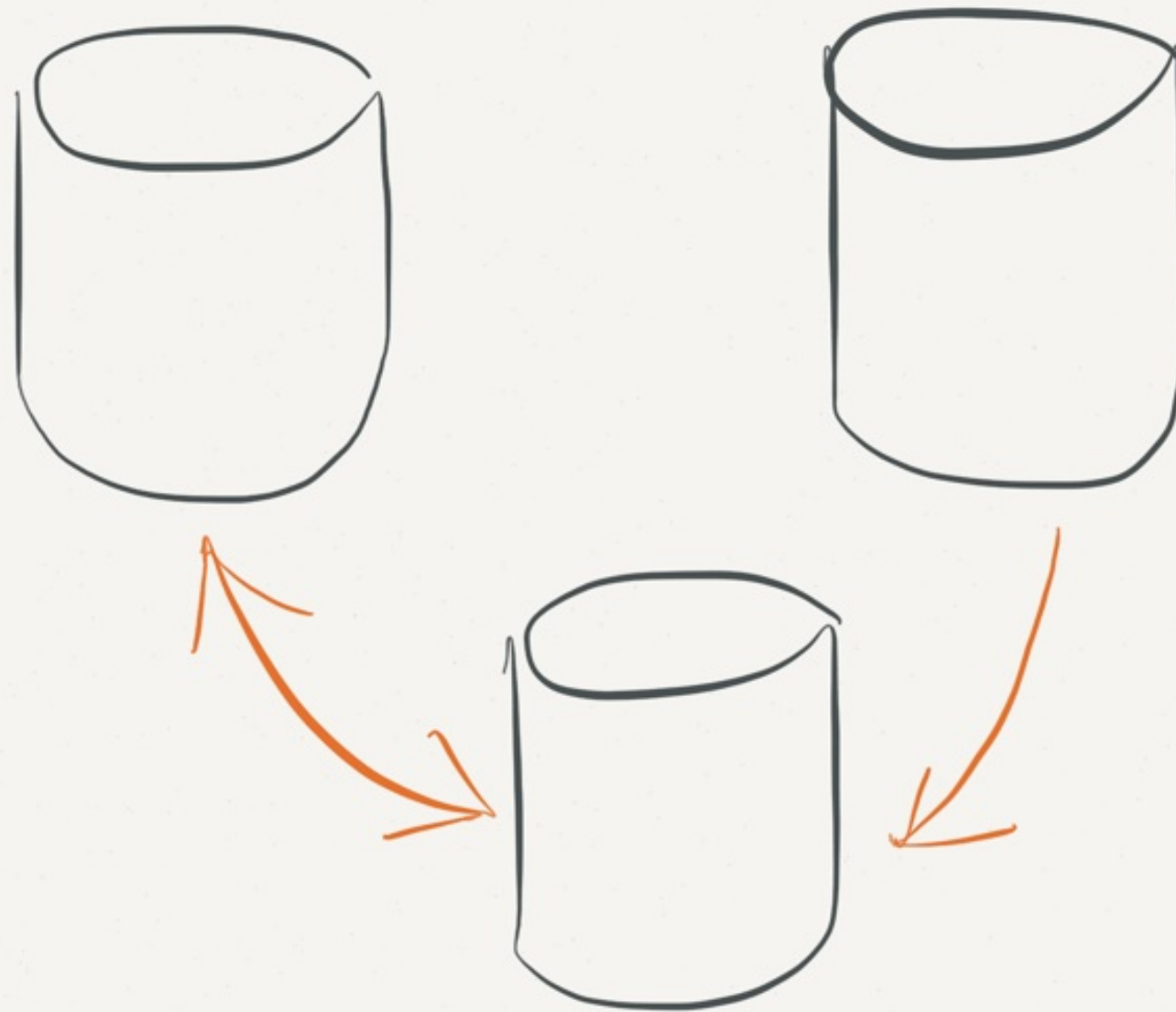
AUTONOMY



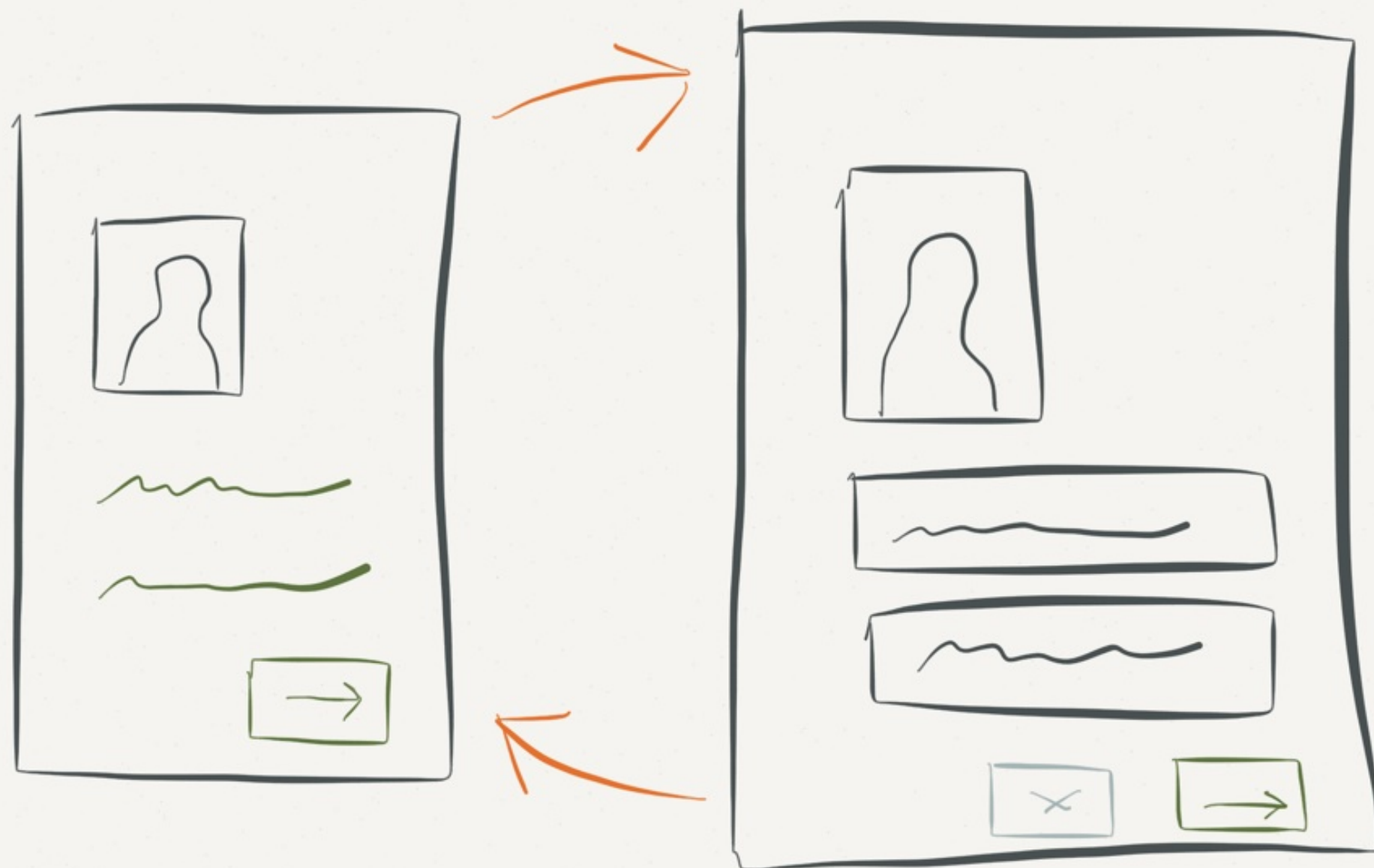
OFFLINE SUPPORT



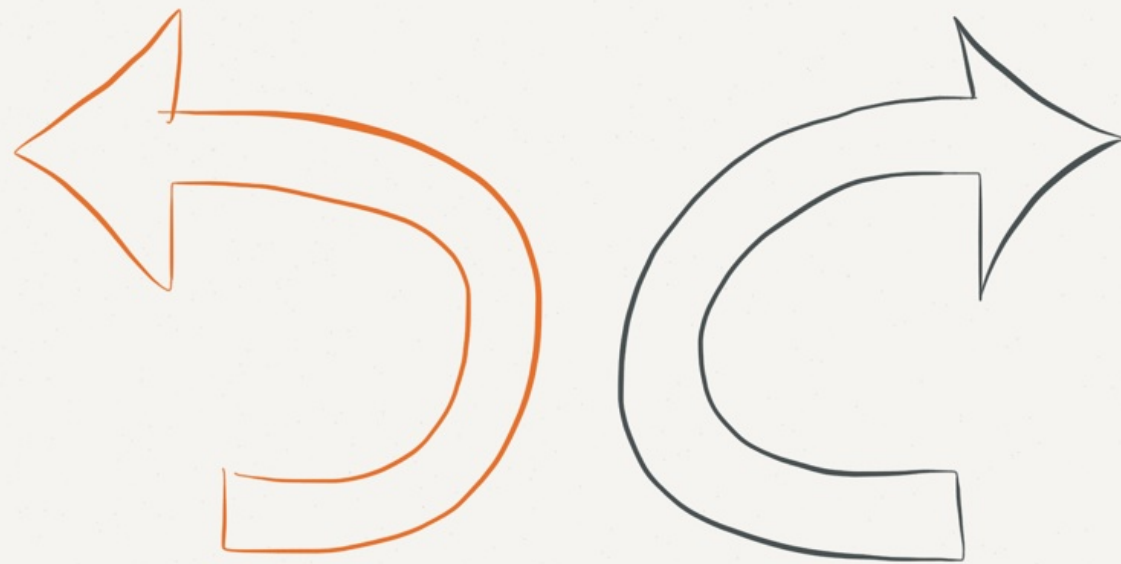
NON-BLOCKING INTERFACES



DATA SYNCHRONIZATION



TRANSACTIONAL CONTEXTS



UNDO / REDO





==

PLAIN OLE'

JS



PRIMITIVES

EVENTS



PRIMITIVES

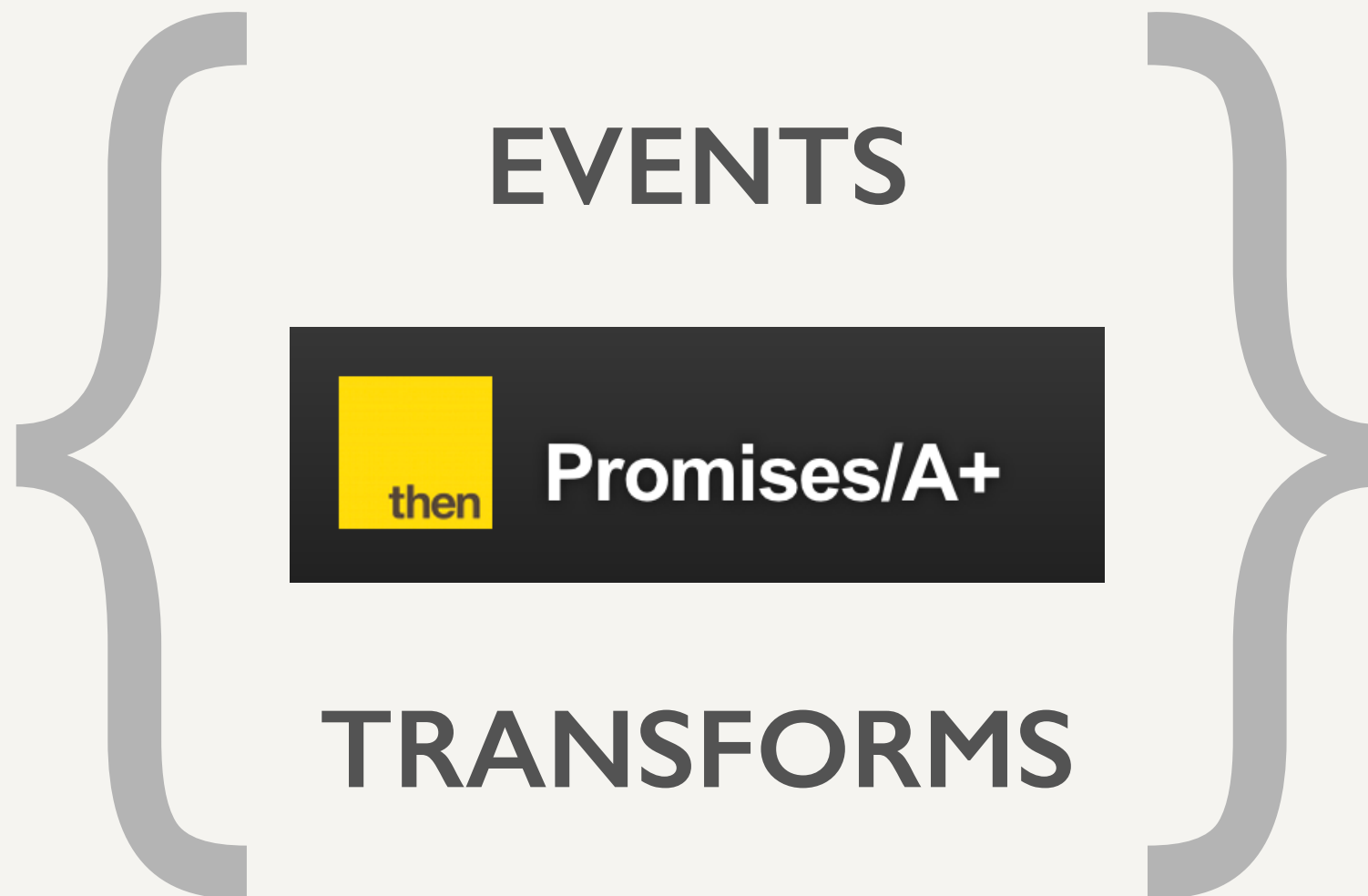
EVENTS



Promises/A+

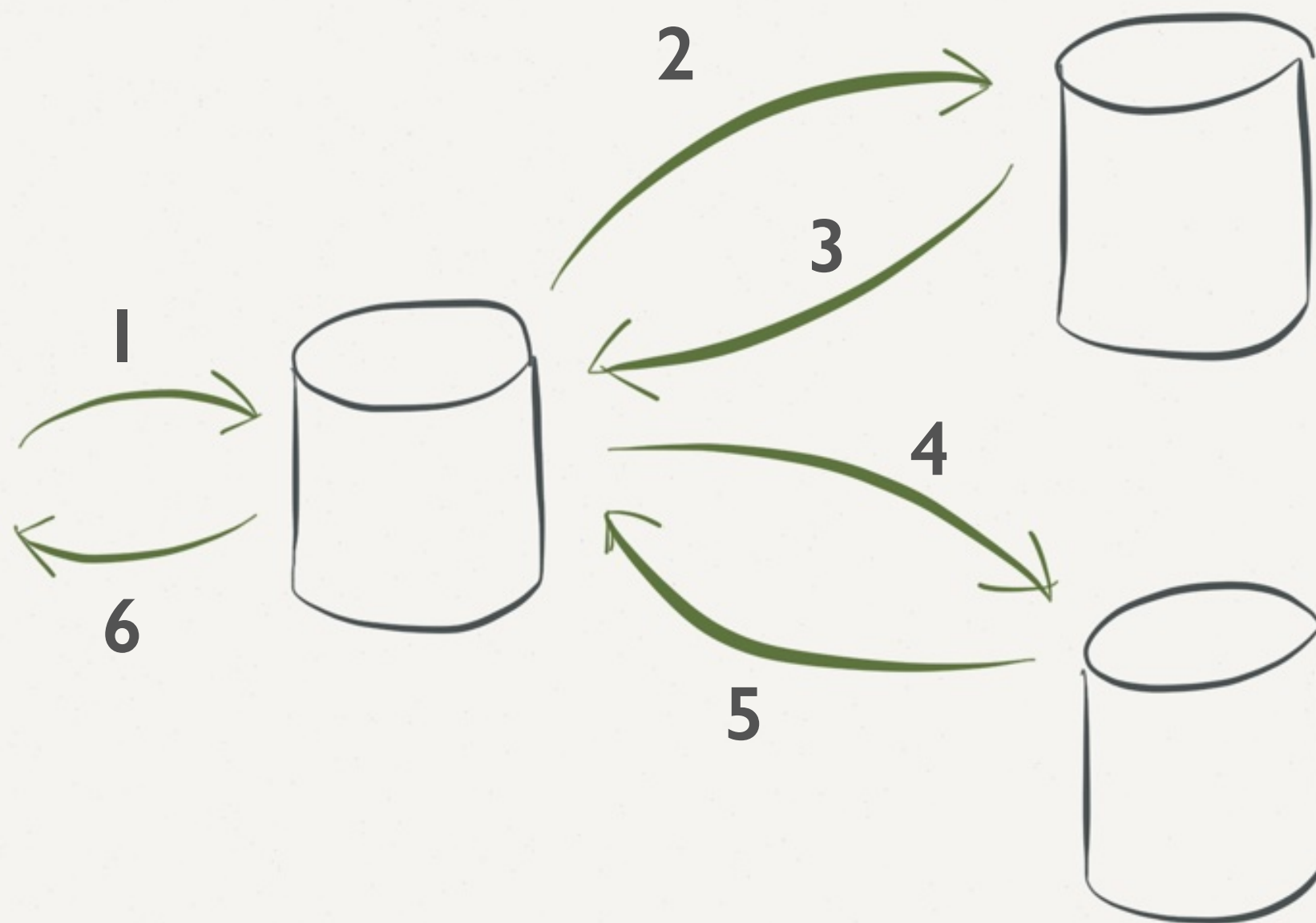


PRIMITIVES



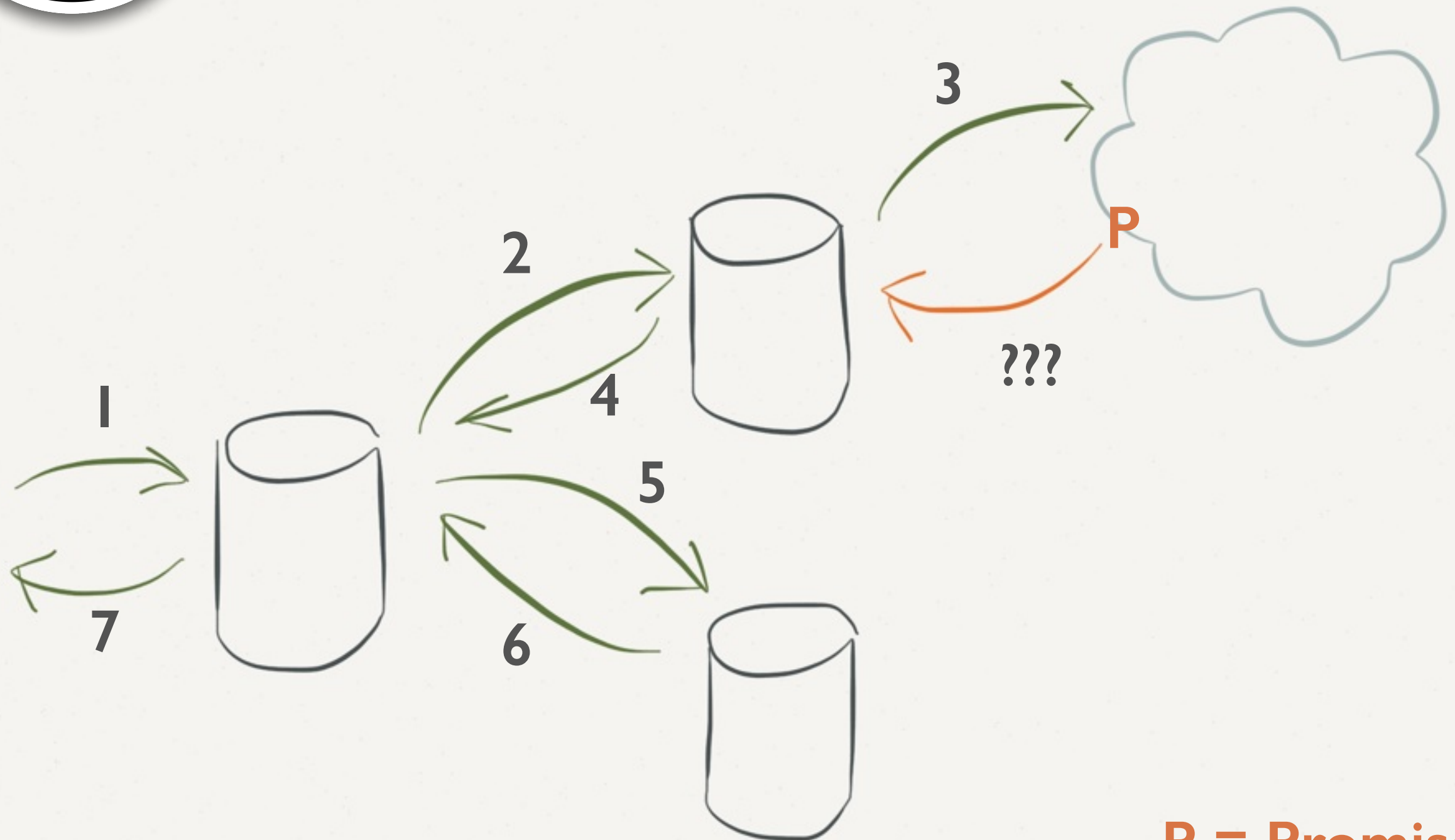


SYNCHRONOUS EVENT HANDLING





SYNCHRONOUS EVENT HANDLING

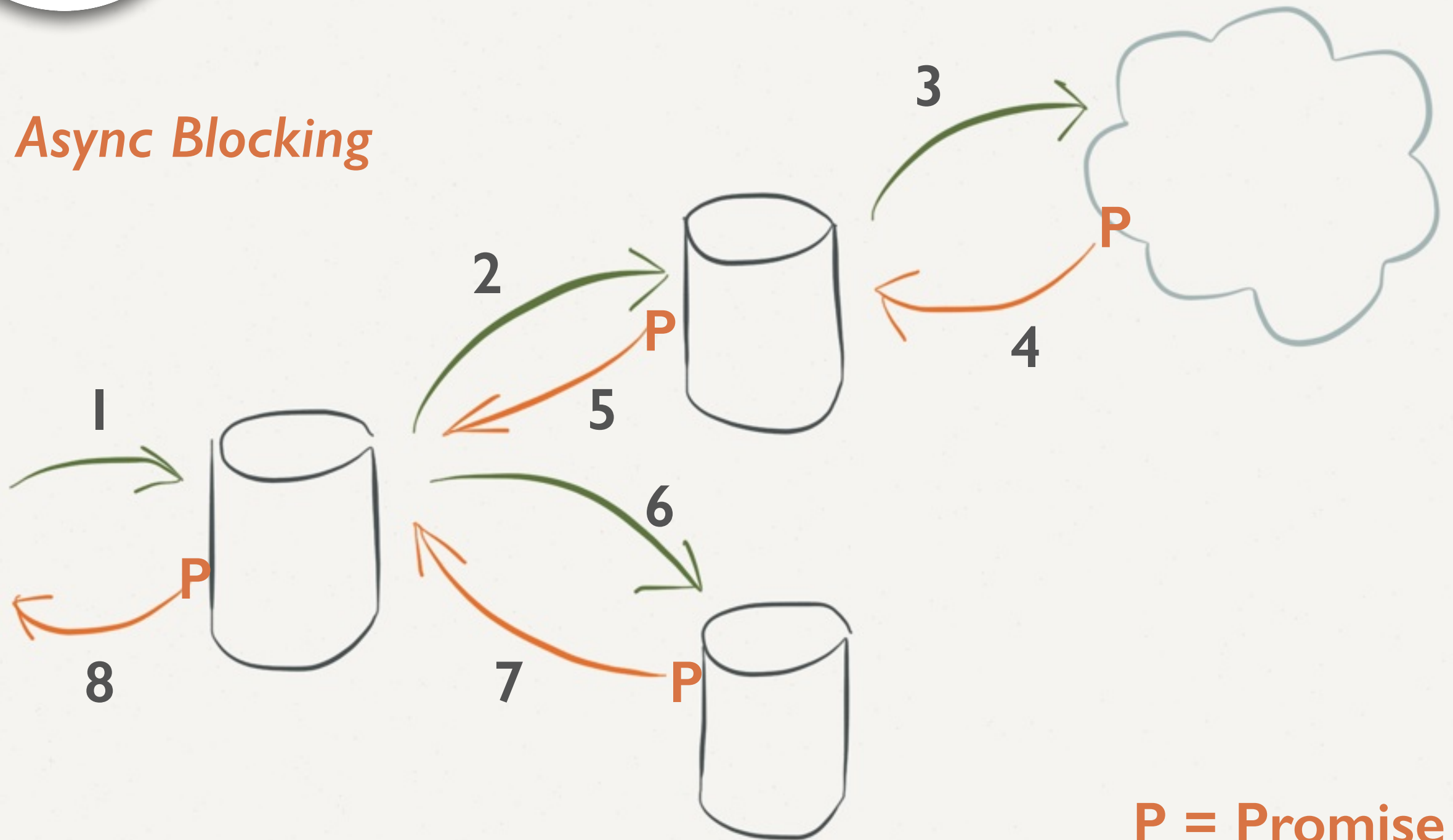


P = Promise



PROMISE-AWARE EVENTS

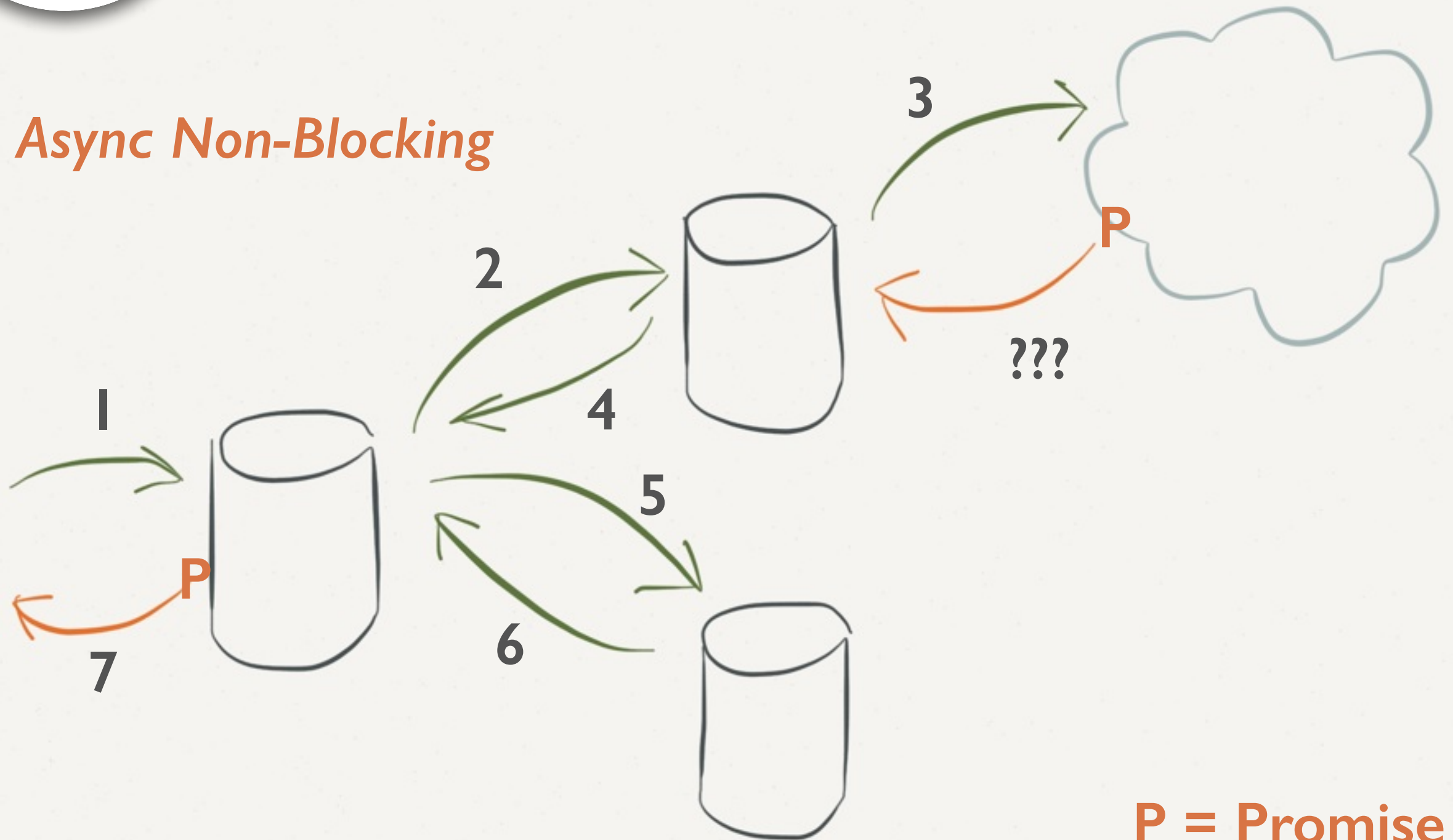
Async Blocking





PROMISE-AWARE EVENTS

Async Non-Blocking



P = Promise

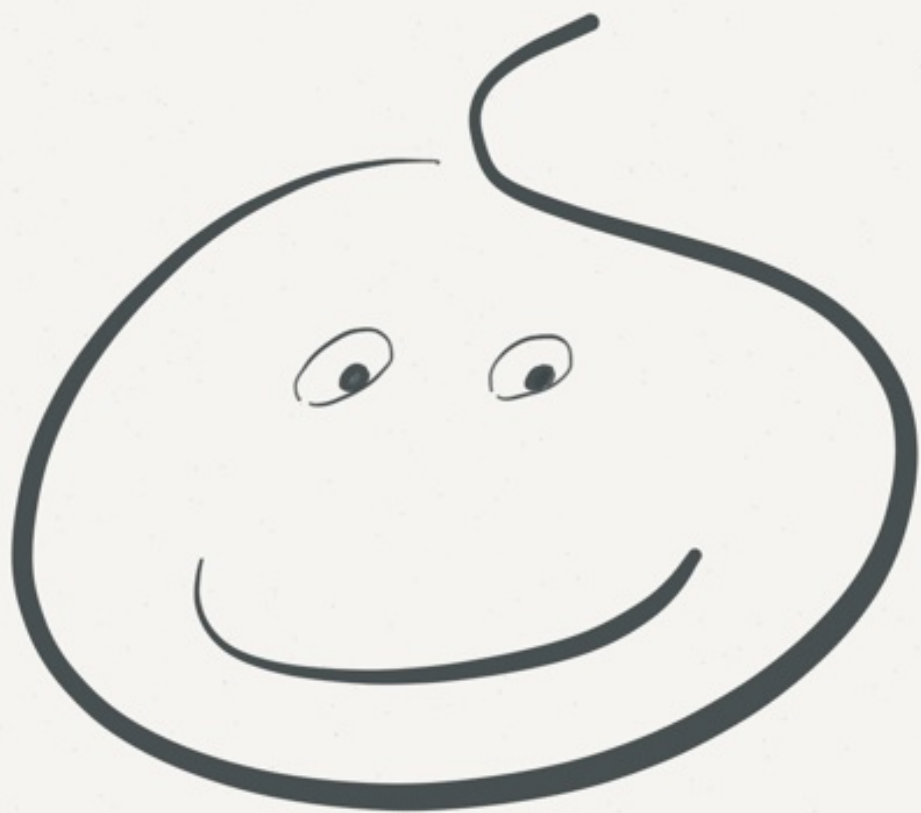


PRIMARY INTERFACES





REQUESTABLE



find

add

remove

update

patch

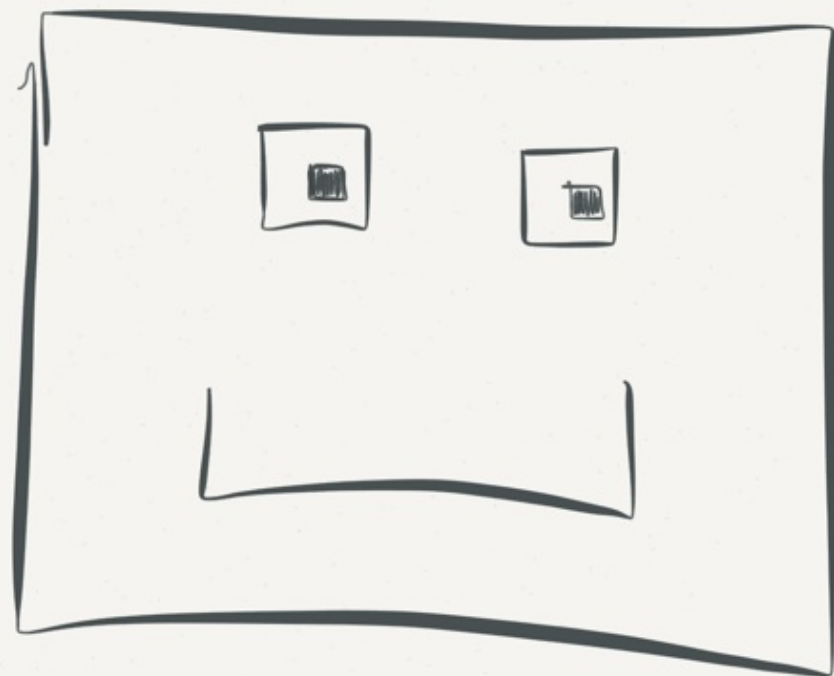
findLink

addLink

removeLink



TRANSFORMABLE



transform(operation)



REQUEST FLOW

```
source.find("planet", "1")
```



```
<< assistFind >>
```

```
  _find()
```

```
    << rescueFind >>
```

```
    << didFind /didNotFind >>
```



REQUEST FLOW

```
source.find("planet", "1")
```



```
<< assistFind >>
```

```
_find()
```

```
<< rescueFind >>
```

```
<< didFind /didNotFind >>
```



REQUEST FLOW

```
source.find("planet", "1")
```



```
<< assistFind >>
```

```
_find()
```

```
<< rescueFind >>
```

```
<< didFind /didNotFind >>
```



REQUEST FLOW

```
source.find("planet", "1")
```



```
<< assistFind >>
```

```
_find()
```

```
<< rescueFind >>
```

```
<< didFind /didNotFind >>
```



TRANSFORM FLOW

source.transform(op)



_transform()

<< didTransform >>



TRANSFORM FLOW

source.transform(op)




_transform()

<< didTransform >>



Orbit.Document

Complete implementation of
JSON Patch (RFC 6902):



add
remove
replace
move
copy
test

transform
retrieve





TRANSFORMATIONS

JSON Patch transformations:

```
{ "op": "remove", "path": [ "planet", "1389295329110.3" ] }
{ "op": "add", "path": [ "planet", "1389295895882.1" ], "value":
{ "name": "Mercury", "id": "1389295895882.1" } }
{ "op": "remove", "path": [ "planet", "1389295895882.1" ] }
{ "op": "add", "path": [ "planet", "1389296188090.2" ], "value":
{ "name": "Mercury", "id": "1389296188090.2" } }
{ "op": "add", "path": [ "planet", "1389296196274.3" ], "value":
{ "name": "Venus", "id": "1389296196274.3" } }
{ "op": "add", "path": [ "planet", "1389296197897.4" ], "value":
{ "name": "Earth", "id": "1389296197897.4" } }
{ "op": "add", "path": [ "planet", "1389296199041.5" ], "value":
{ "name": "Mars", "id": "1389296199041.5" } }
```



CONNECTORS

- **TransformConnector** for connecting a source and target.
 - Two connectors are needed for bi-directional syncs.
 - Connectors can be blocking or not
- **RequestConnector** for coordinating requests to data



COMMON LIBRARY



ORBIT COMMON LIB

SOURCES

MEMORY



ORBIT COMMON LIB

SOURCES





ORBIT COMMON LIB

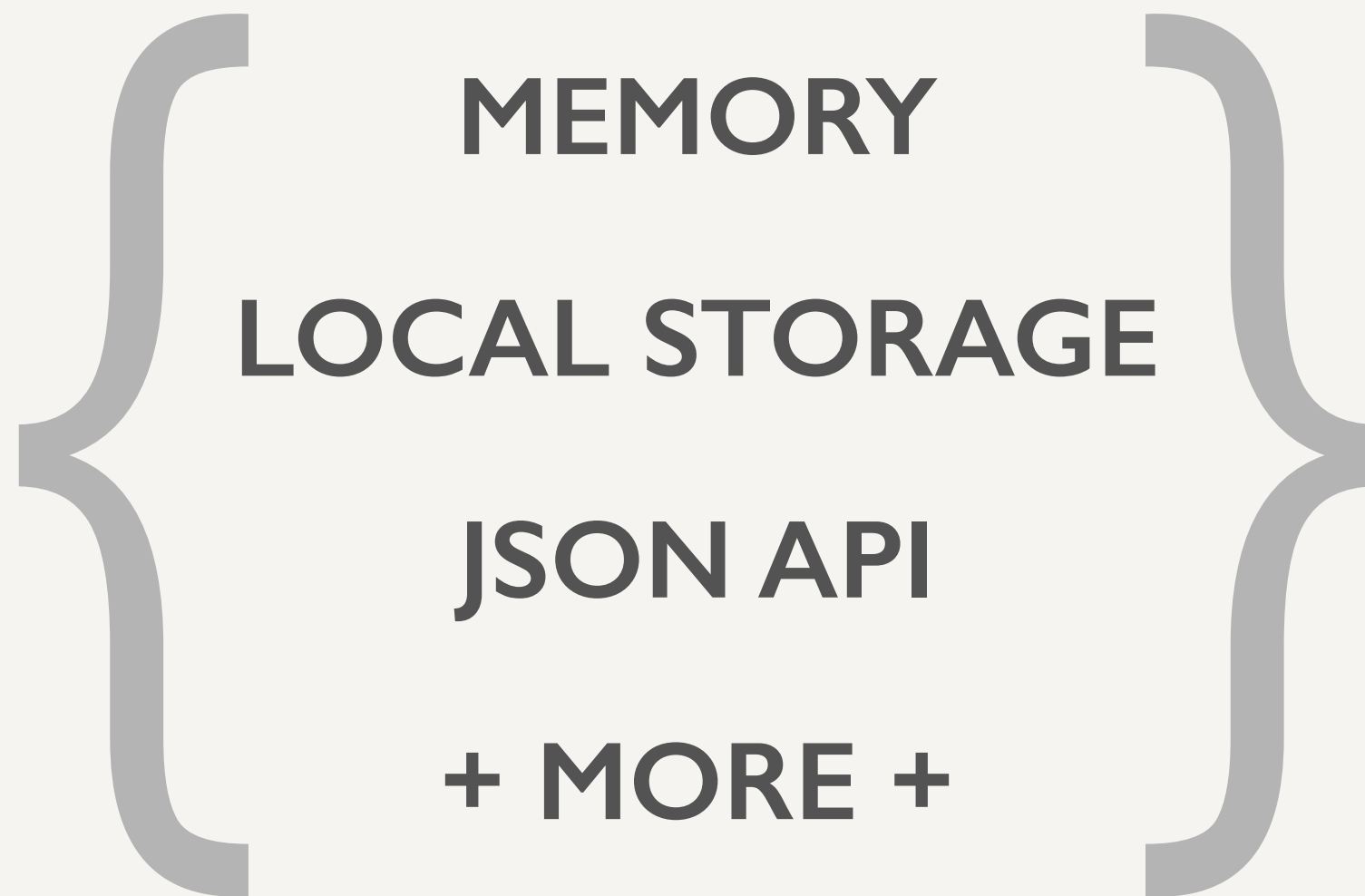
SOURCES





ORBIT COMMON LIB

SOURCES





ORBIT COMMON LIB

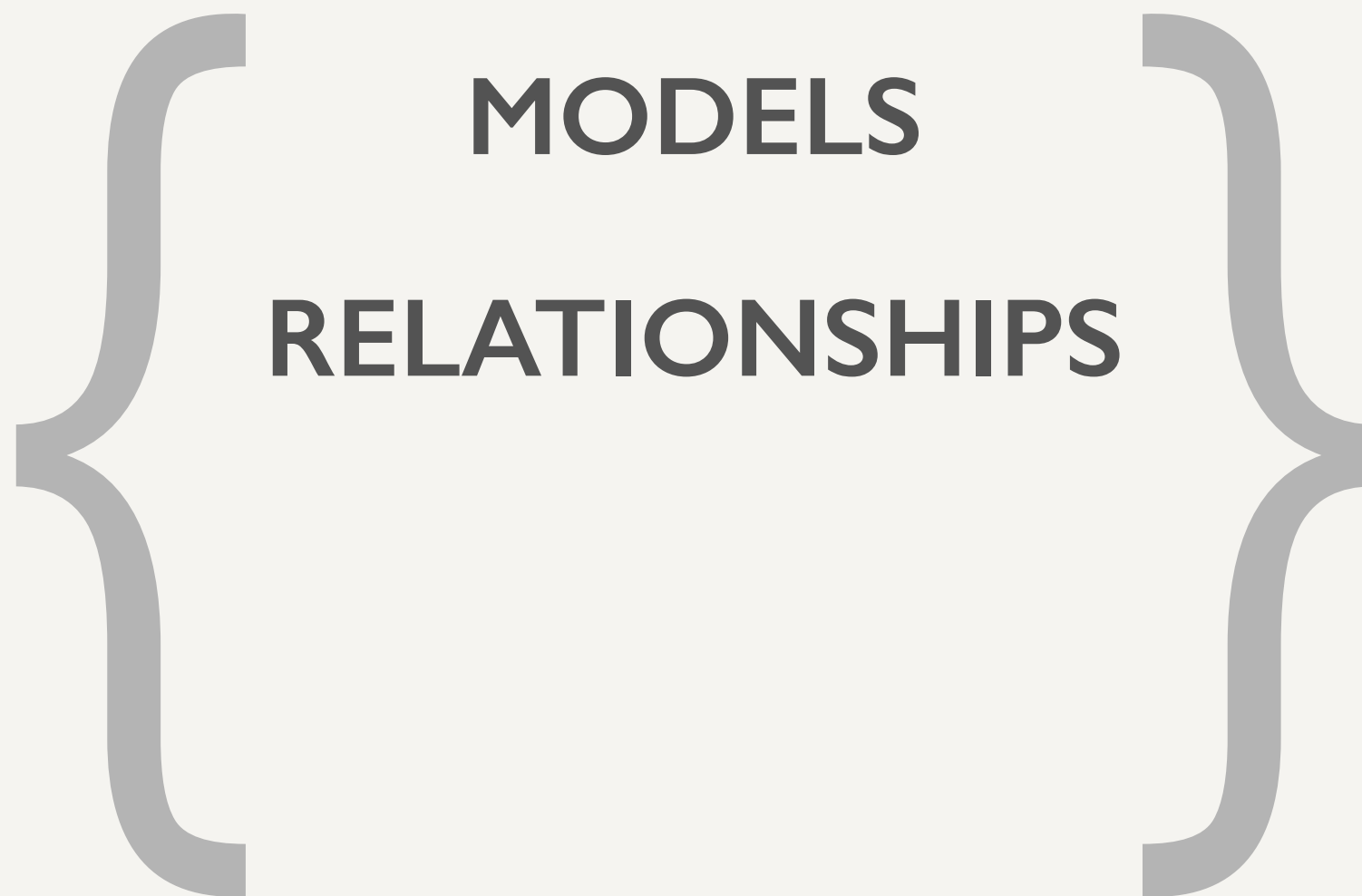
SCHEMA

MODELS



ORBIT COMMON LIB

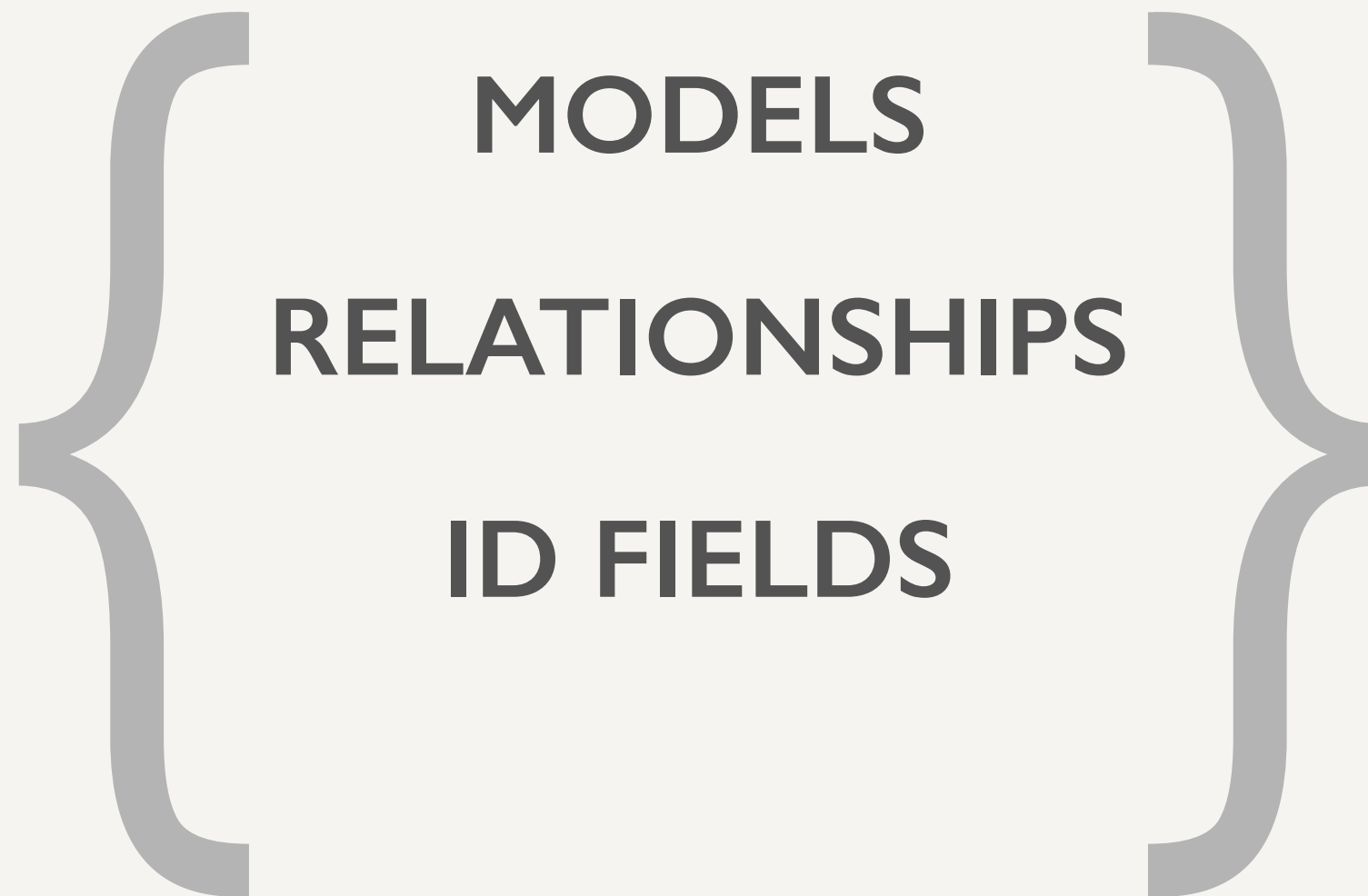
SCHEMA





ORBIT COMMON LIB

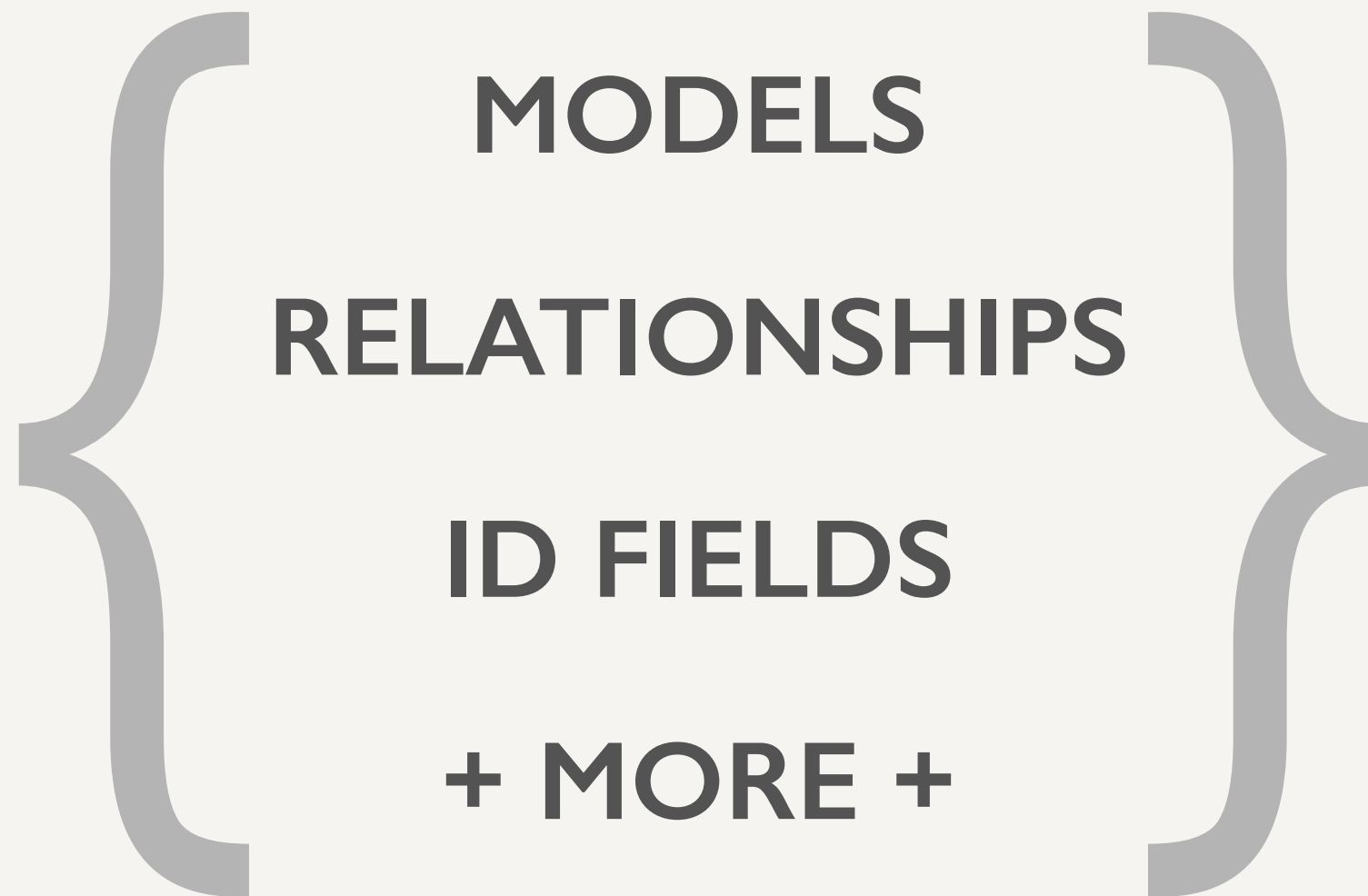
SCHEMA





ORBIT COMMON LIB

SCHEMA





+





EO.Source

Provides a loose wrapper around an **OC.Source**:

```
App.LocalStorageSource = EO.Source.extend({  
  orbitSourceClass: OC.LocalStorageSource,  
  orbitSourceOptions: {  
    namespace: "myApp" // n.s. for localStorage  
  }  
});
```



EO.Store

- Extends **EO.Source**
- Maintains an identity map of model instances.
- Familiar interfaces to access and request data.



EO.Store

Synchronous

{
all
filter
retrieve
}

Asynchronous

{
find
add
remove
patch
findLink
addLink
removeLink
}



EO.Model

- Defines schema with attributes and relationships
- Backed directly by data in the source
- Familiar interfaces to access and request data



EO.Model

```
Star = EO.Model.extend({  
  name: attr('string'),  
  planets: hasMany('planet', {inverse: 'sun'})  
});
```

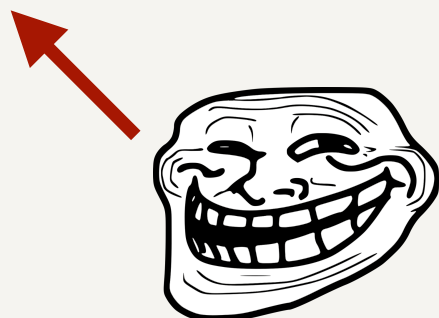
```
Planet = EO.Model.extend({  
  name: attr('string'),  
  classification: attr('string'),  
  sun: hasOne('star', {inverse: 'planets'})  
});
```



EO.Model

```
Star = EO.Model.extend({  
  name: attr('string'),  
  planets: hasMany('planet', {inverse: 'sun'})  
});
```

```
Planet = EO.Model.extend({  
  name: attr('string'),  
  classification: attr('string'),  
  sun: hasOne('star', {inverse: 'planets'})  
});
```





EO.Model

Synchronous

{
get
set
}

Asynchronous

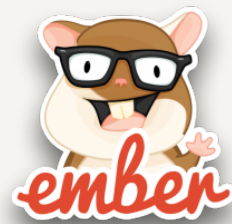
{
remove
patch
findLink
addLink
removeLink
}



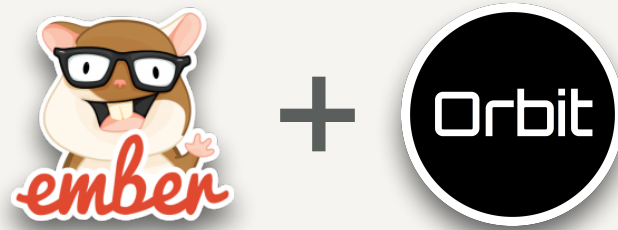
EO.Model

```
store.add("planet", {name: "Jupiter"}).then(  
  function(jupiter) {  
    jupiter.get("moons").pushObject(io);  
    jupiter.set("classification", "gas giant");  
  }  
);
```

```
store.then(function() { // all requests resolve  
  console.log(io.get("planet.name")); // Jupiter  
});
```



URLS DRIVE APPLICATION STATE



SOURCES DRIVE MODEL STATE

AUTONOMOUS WEB APPLICATIONS WITH





+



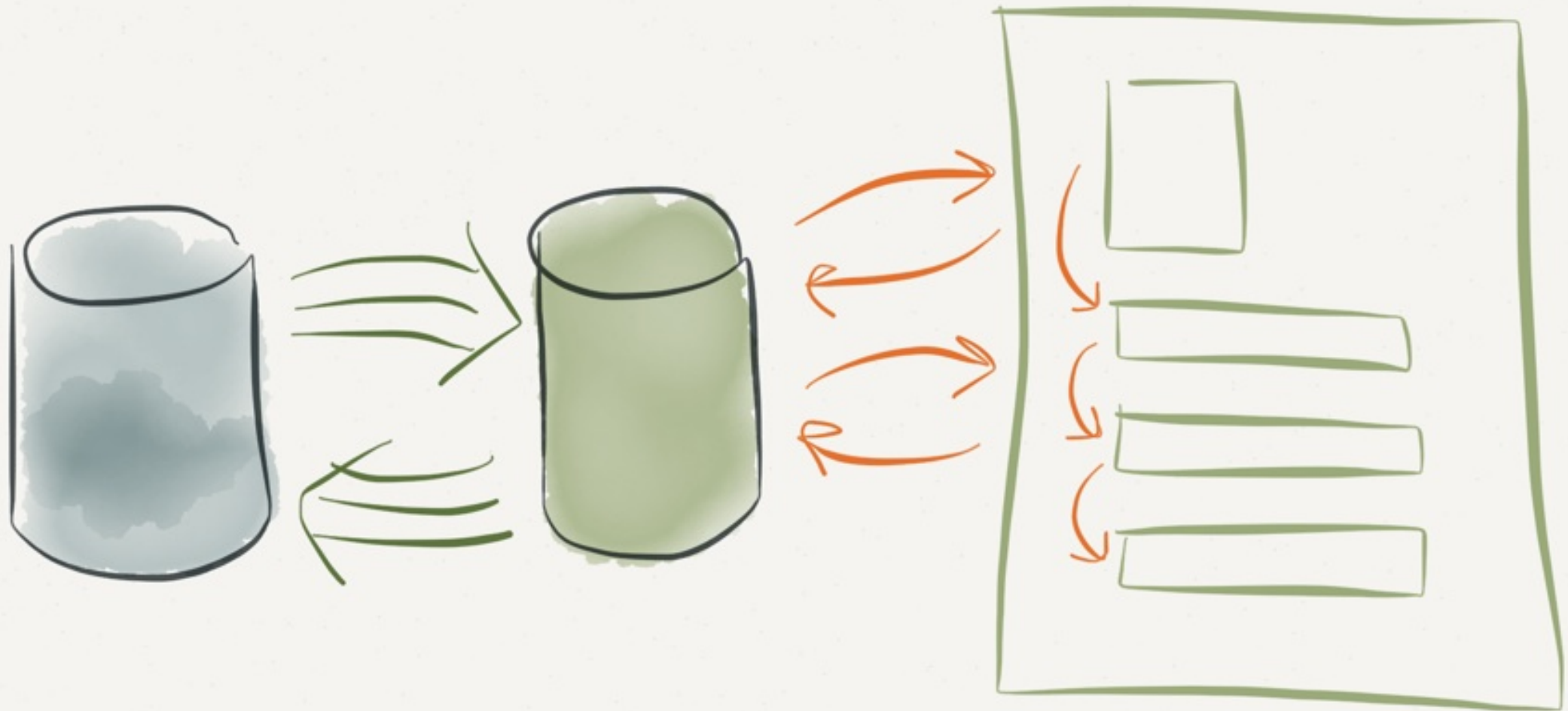
IndexedDB

LocalStorage

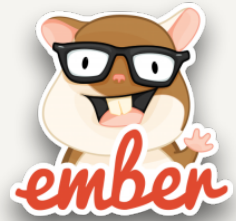
PLUGGABLE SOURCES



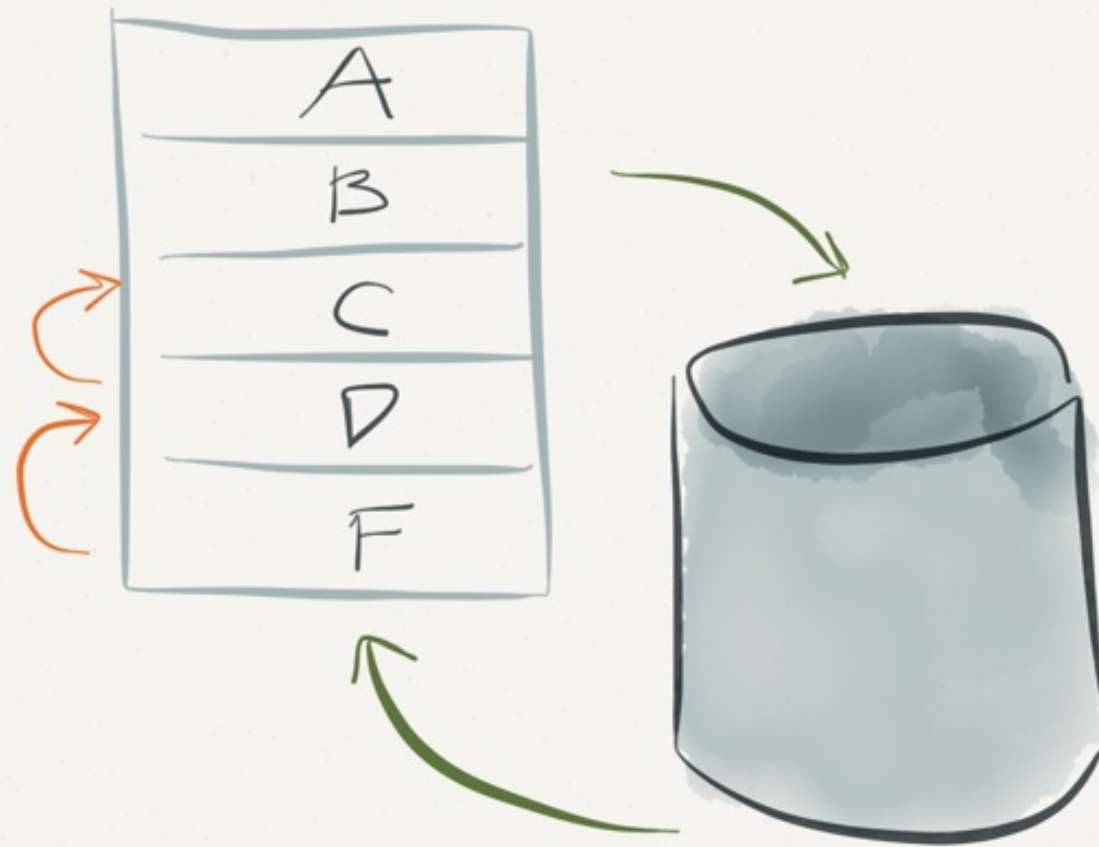
+



EDITING CONTEXTS



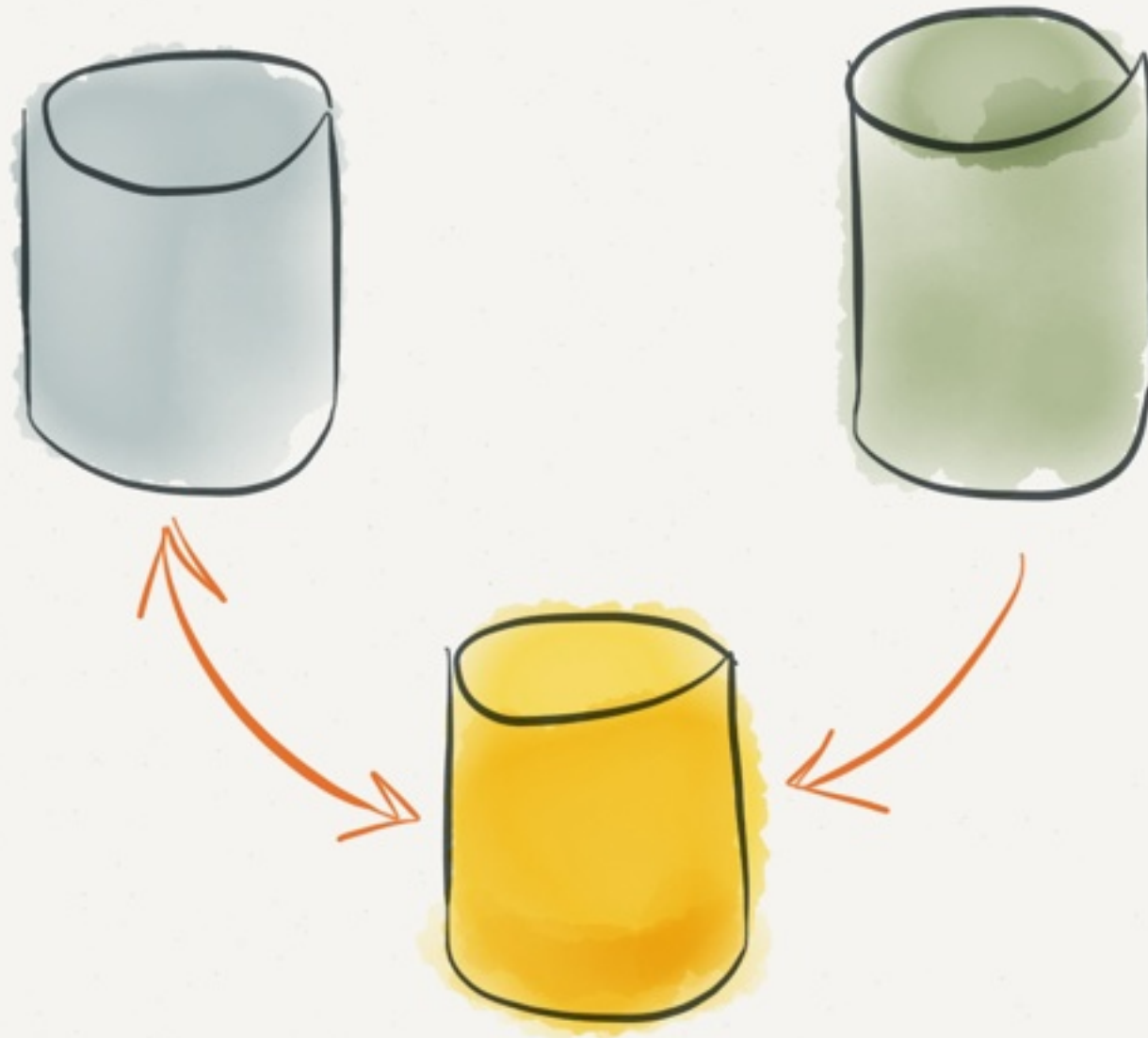
+



UNDO / REDO



+



DATA SYNCHRONIZATION



STATUS

- **orbit.js**
 - Released January 9, 2014
 - v0.2.0
- **ember-orbit**
 - Released June 17, 2014 (!!!)
 - v0.1.0



TODOS

- **orbit.js**
 - More sources
 - More synchronization strategies
- **ember-orbit**
 - Fully encapsulate Orbit's API
 - Improve ergonomics, esp. serialization & normalization



COMING SOON

orbitjs.com

API docs and guides for Orbit and Ember-Orbit

Follow progress:

Twitter: **@orbitjs**

Github: **<https://github.com/orbitjs>**

THANKS!

@dgeb

Wicked Good Ember 2014

Cerebris

