Expressing Security Constraints using capabilities

Mark S. Miller and the Cajadores

Overview

This talk

The *What* and *Why* of object-capabilities (ocaps)

My “Securing EcmaScript 5” talk tomorrow

The *How* of doing ocaps in JavaScript

Patterns of Safe Cooperation

In Secure EcmaScript (SES)

Distributed Cryptographic Capabilities

In Distributed Resilient Secure EcmaScript (Dr. SES)

Security as Extreme Modularity

Modularity: Avoid needless dependencies

Security: Avoid needless vulnerabilities

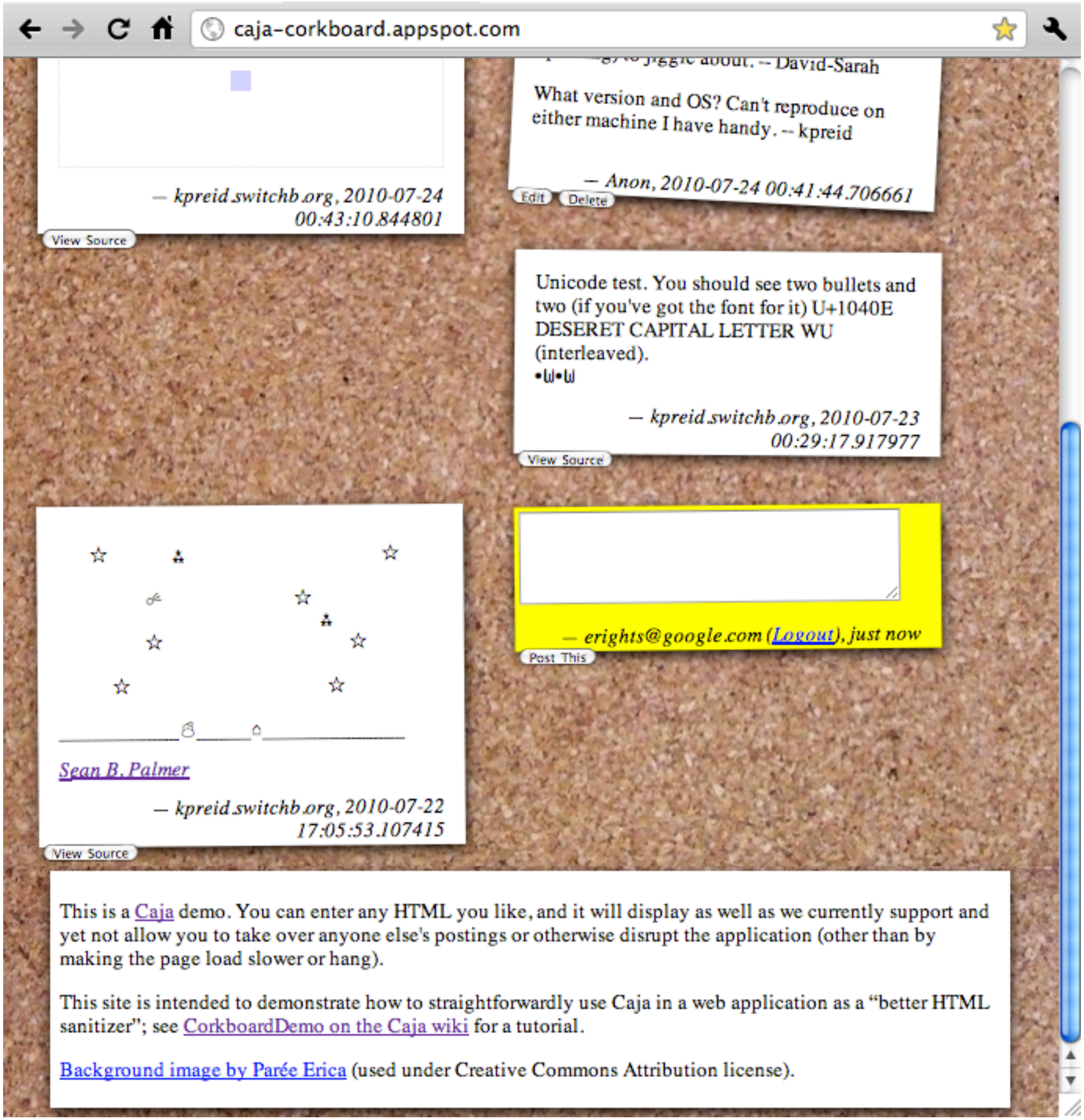
Vulnerability is a form of dependency

Mod: Principle of info hiding - need to know.

Sec: Principle of least authority - need to do.

The Mashup problem: Code as Media

```
<html> <head> <title>Basic Mashup</title> <script>
  function animate(id) {
    var element = document.getElementById(id);
    var textNode = element.childNodes[0];
    var text = textNode.data;
    var reverse = false;
    element.onclick = function() { reverse = !reverse; };
    setInterval(function() {
      textNode.data = text = reverse ? text.substring(1) + text[0]
                : text[text.length-1] + text.substring(0, text.length-1);
    }, 100);
  }
</script> </head> <body onload="animate('target')">
  <pre id="target">Hello Programmable World!  </pre>
</body> </html>
```



— kpreid.switchb.org, 2010-07-24 00:43:10.844801

...ggle about. -- David-Sarah

What version and OS? Can't reproduce on either machine I have handy. -- kpreid

— Anon, 2010-07-24 00:41:44.706661

Unicode test. You should see two bullets and two (if you've got the font for it) U+1040E DESERET CAPITAL LETTER WU (interleaved).
•𐐎•𐐎

— kpreid.switchb.org, 2010-07-23 00:29:17.917977

Sean B. Palmer

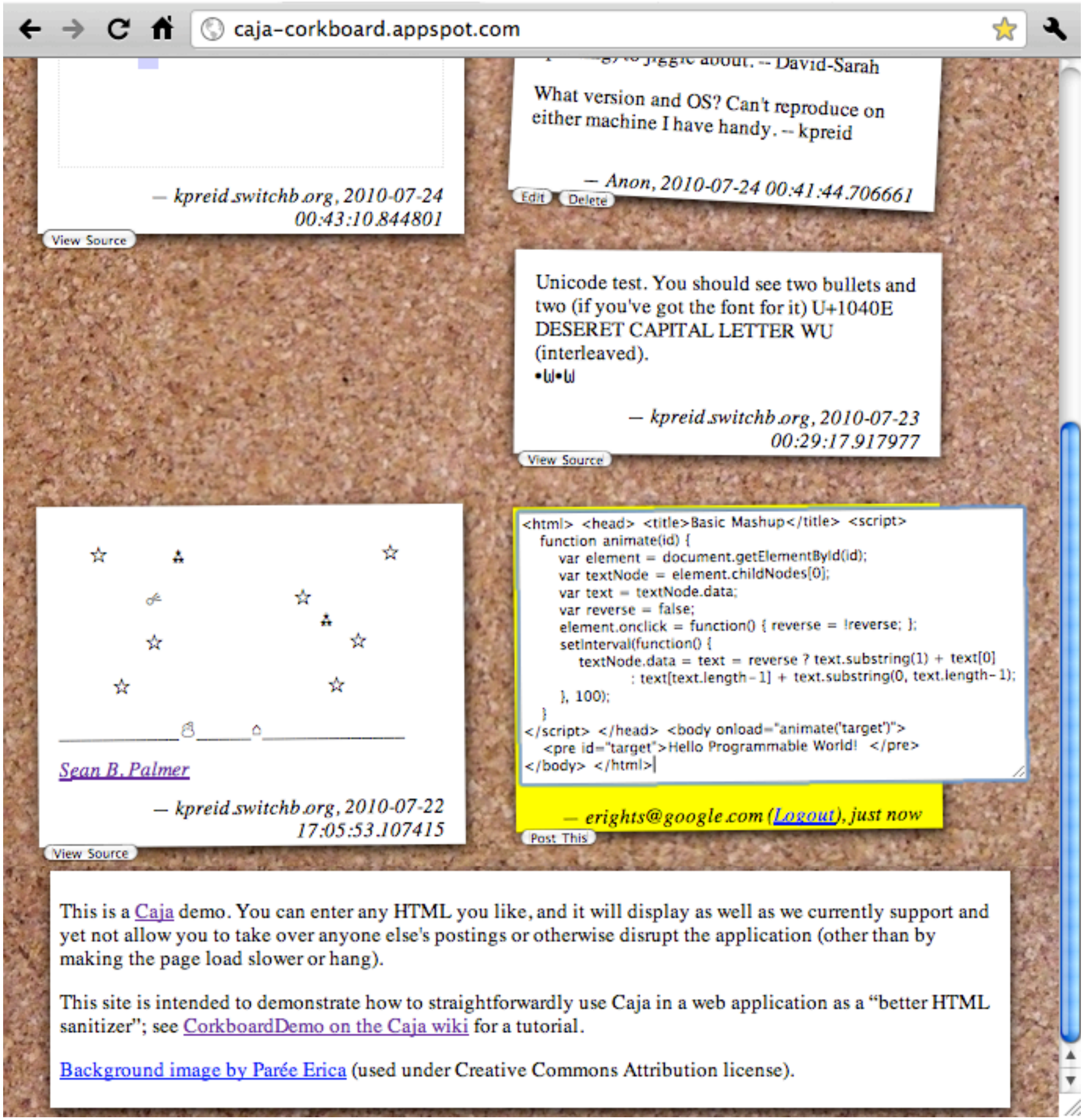
— kpreid.switchb.org, 2010-07-22 17:05:53.107415

— erights@google.com (Logout), just now

This is a Caja demo. You can enter any HTML you like, and it will display as well as we currently support and yet not allow you to take over anyone else's postings or otherwise disrupt the application (other than by making the page load slower or hang).

This site is intended to demonstrate how to straightforwardly use Caja in a web application as a “better HTML sanitizer”; see CorkboardDemo on the Caja wiki for a tutorial.

Background image by Parée Erica (used under Creative Commons Attribution license).



— kpreid.switchb.org, 2010-07-24 00:43:10.844801

jiggle about. -- David-Sarah

What version and OS? Can't reproduce on either machine I have handy. -- kpreid

— Anon, 2010-07-24 00:41:44.706661

Unicode test. You should see two bullets and two (if you've got the font for it) U+1040E DESERET CAPITAL LETTER WU (interleaved).
•𐐎•𐐎

— kpreid.switchb.org, 2010-07-23 00:29:17.917977

Sean B. Palmer

— kpreid.switchb.org, 2010-07-22 17:05:53.107415

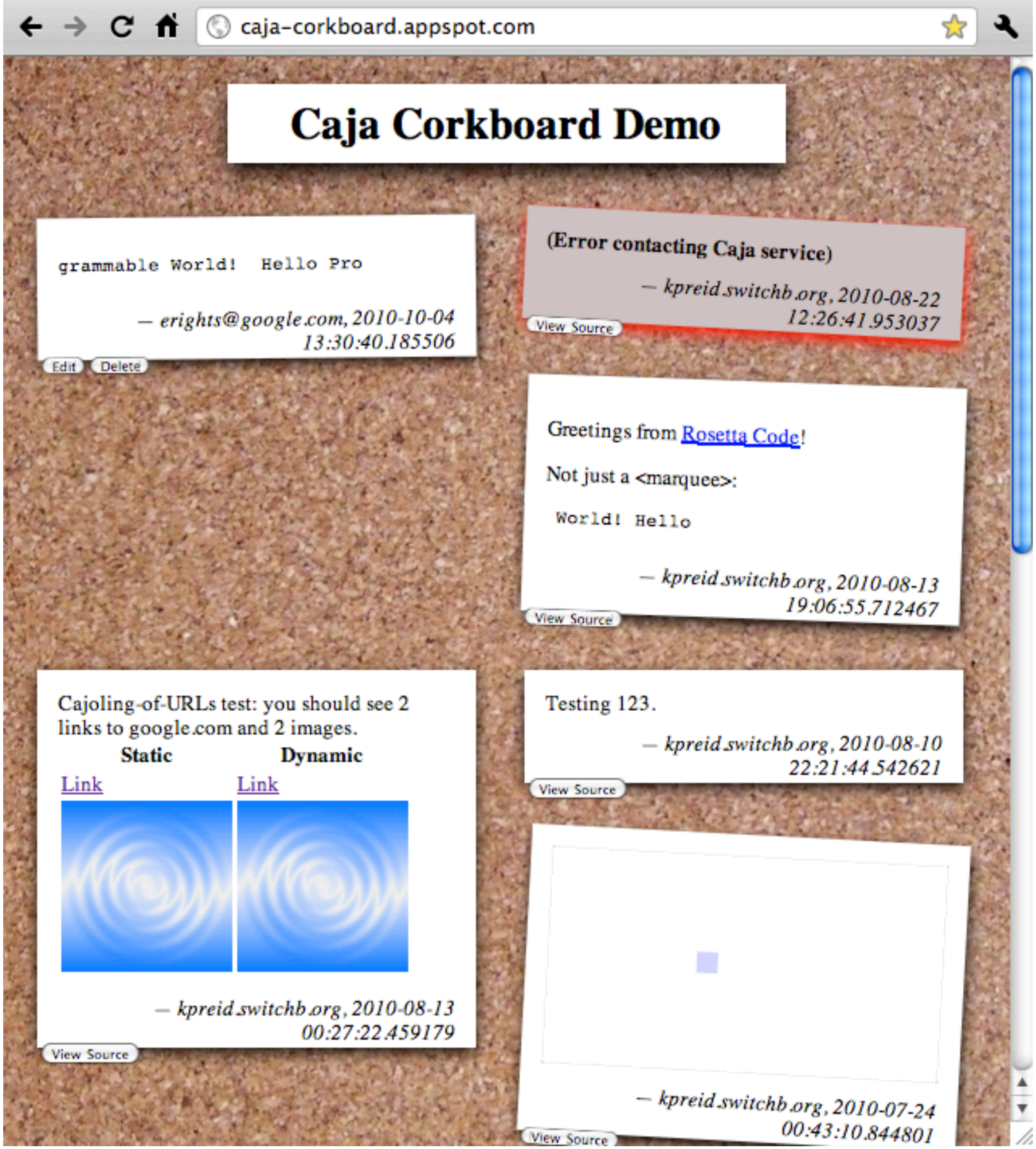
```
<html> <head> <title>Basic Mashup</title> <script>
  function animate(id) {
    var element = document.getElementById(id);
    var textNode = element.childNodes[0];
    var text = textNode.data;
    var reverse = false;
    element.onclick = function() { reverse = !reverse; };
    setInterval(function() {
      textNode.data = text = reverse ? text.substring(1) + text[0]
                 : text[text.length-1] + text.substring(0, text.length-1);
    }, 100);
  }
</script> </head> <body onload="animate('target')">
  <pre id="target">Hello Programmable World!  </pre>
</body> </html>
```

— erights@google.com (Logout), just now

This is a Caja demo. You can enter any HTML you like, and it will display as well as we currently support and yet not allow you to take over anyone else's postings or otherwise disrupt the application (other than by making the page load slower or hang).

This site is intended to demonstrate how to straightforwardly use Caja in a web application as a “better HTML sanitizer”; see CorkboardDemo on the Caja wiki for a tutorial.

Background image by Parée Erica (used under Creative Commons Attribution license).



Caja Corkboard Demo

```
grammable World!  Hello Pro
```

— *erights@google.com, 2010-10-04 13:30:40.185506*

Cajoling-of-URLs test: you should see 2 links to google.com and 2 images.

Static	**Dynamic**
Link	Link

— *kpreid.switchb.org, 2010-08-13 00:27:22.459179*

(Error contacting Caja service)

— *kpreid.switchb.org, 2010-08-22 12:26:41.953037*

Greetings from Rosetta Code!

Not just a <marquee>:

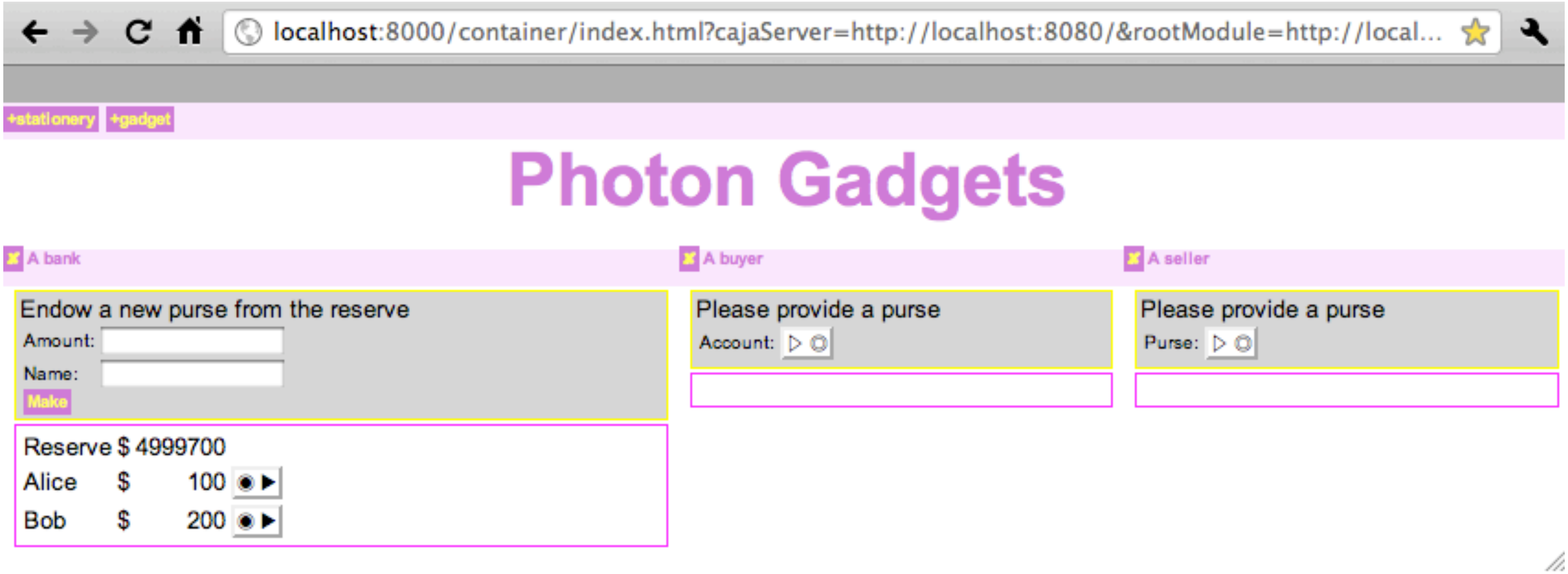
```
World! Hello
```

— *kpreid.switchb.org, 2010-08-13 19:06:55.712467*

Testing 123.

— *kpreid.switchb.org, 2010-08-10 22:21:44.542621*

— *kpreid.switchb.org, 2010-07-24 00:43:10.844801*



localhost:8000/container/index.html?cajaServer=http://localhost:8080/&rootModule=http://local...

Photon Gadgets

A bank

Endow a new purse from the reserve

Amount:

Name:

Reserve $ 4999700

Alice	$	100	● ▶
Bob	$	200	● ▶

A buyer

Please provide a purse

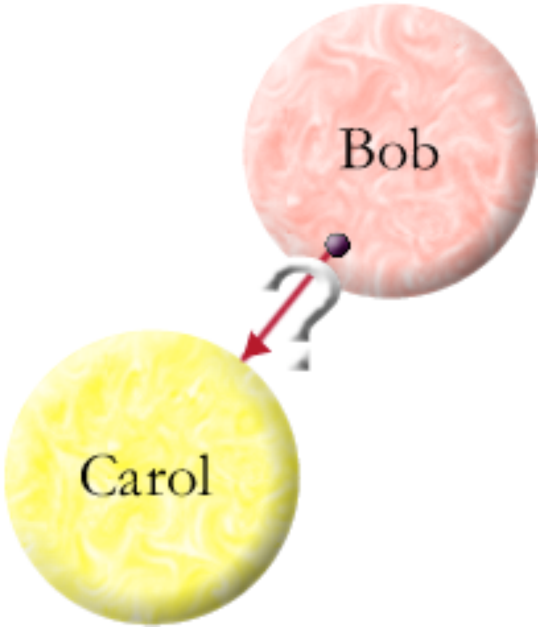
Account: ▷ ○

A seller

Please provide a purse

Purse: ▷ ○

How do I designate thee?

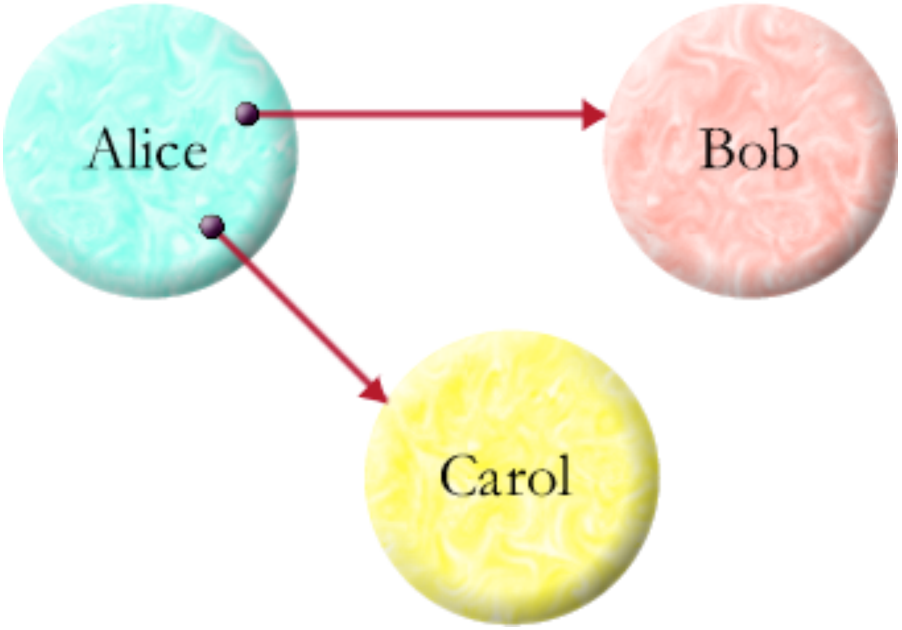


- by Introduction
  - ref to Carol
  - ref to Bob
  - decides to share
- by Parenthood
- by Endowment
- by Initial Conditions

How might object Bob come to know of object Carol?

How do I designate thee?

Alice says: bob.foo(carol)



by Introduction

- ref to Carol
- ref to Bob
- decides to share

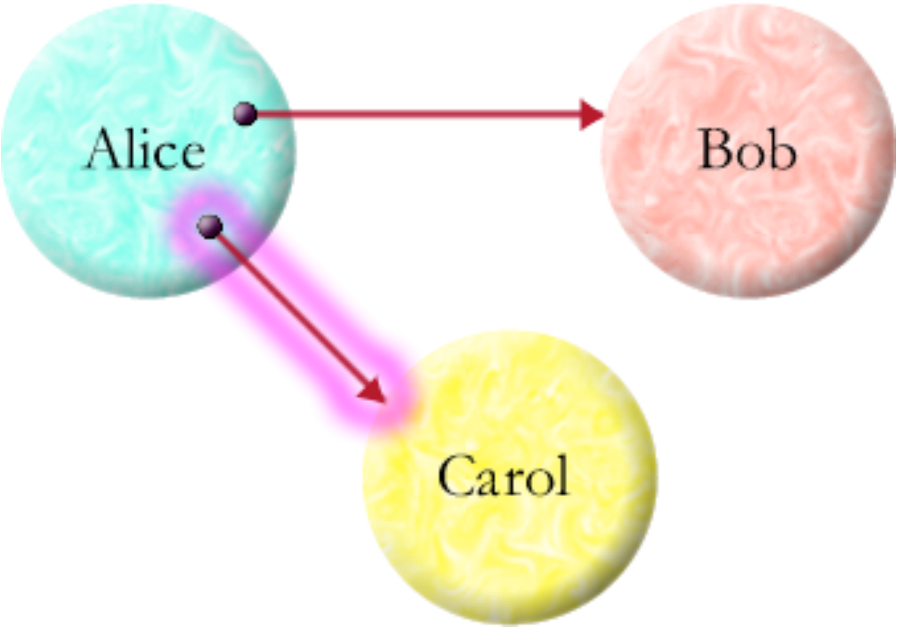
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Alice says: bob.foo(carol)



by Introduction

- ***ref to Carol***
- ref to Bob
- decides to share

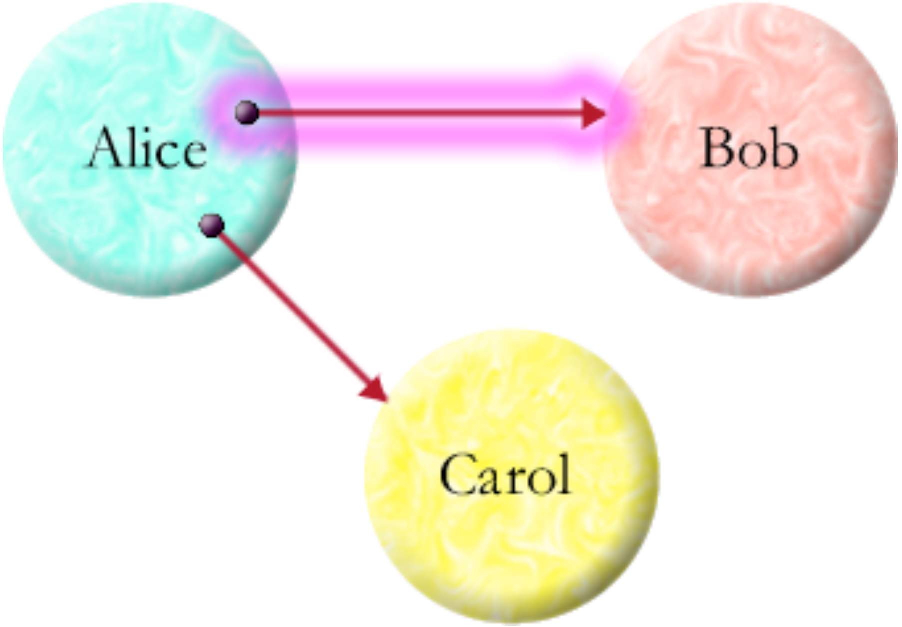
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Alice says: bob.foo(carol)



by Introduction

ref to Carol

ref to Bob

decides to share

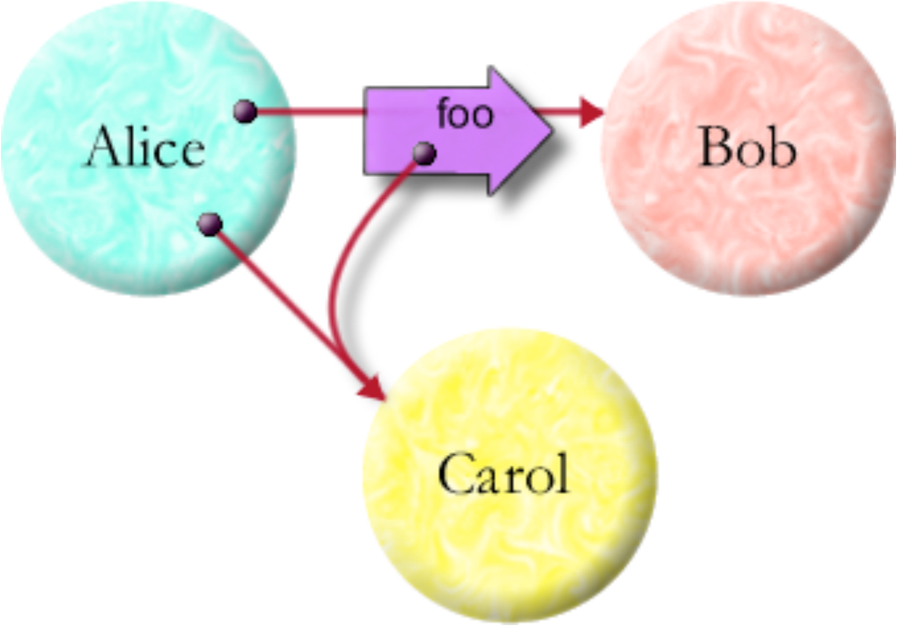
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Alice says: bob.foo(carol)



by Introduction

ref to Carol

ref to Bob

decides to share

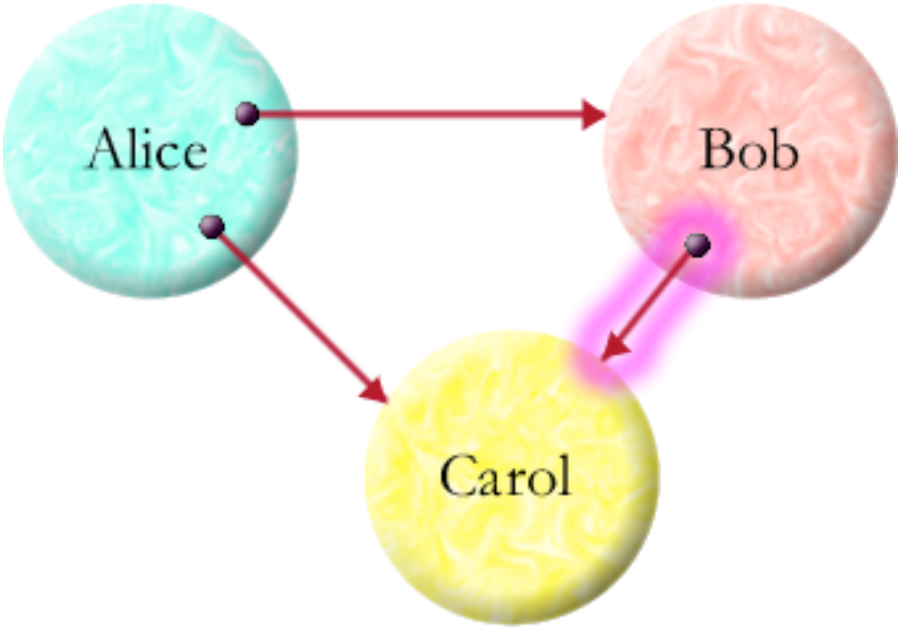
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Alice says: bob.foo(carol)



by Introduction

- *ref to Carol*
- *ref to Bob*
- *decides to share*

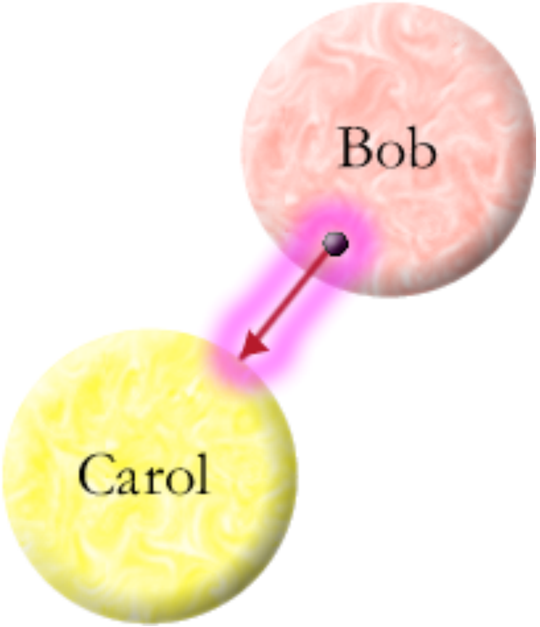
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Bob says: var *carol* = { ... };



by Introduction

- ref to Carol
- ref to Bob
- decides to share

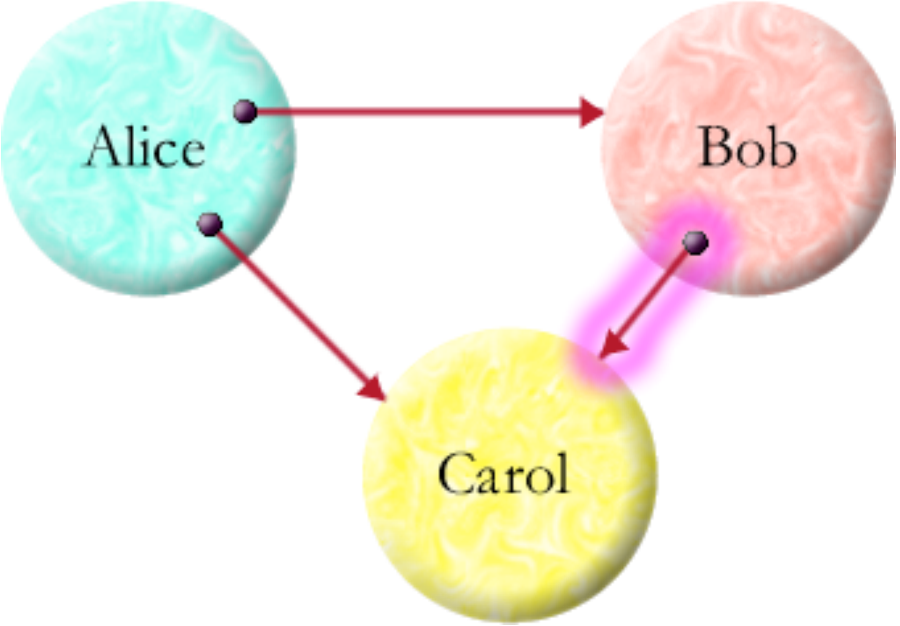
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

Alice says: var *bob* = { … carol … };



by Introduction

- ref to Carol
- ref to Bob
- decides to share

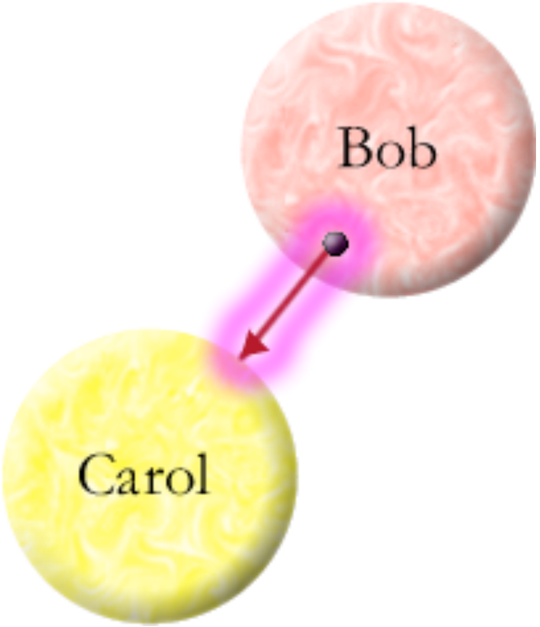
by Parenthood

by Endowment

by Initial Conditions

How do I designate thee?

At t_0:



by Introduction

- ref to Carol
- ref to Bob
- decides to share

by Parenthood

by Endowment

by Initial Conditions

OCaps: Small step from pure objects

Memory safety and encapsulation

+ Effects ***only*** by using held references

+ No powerful references by default

OCaps: Small step from pure objects

Memory safety and encapsulation

+ Effects ***only*** by using held references

+ No powerful references by default

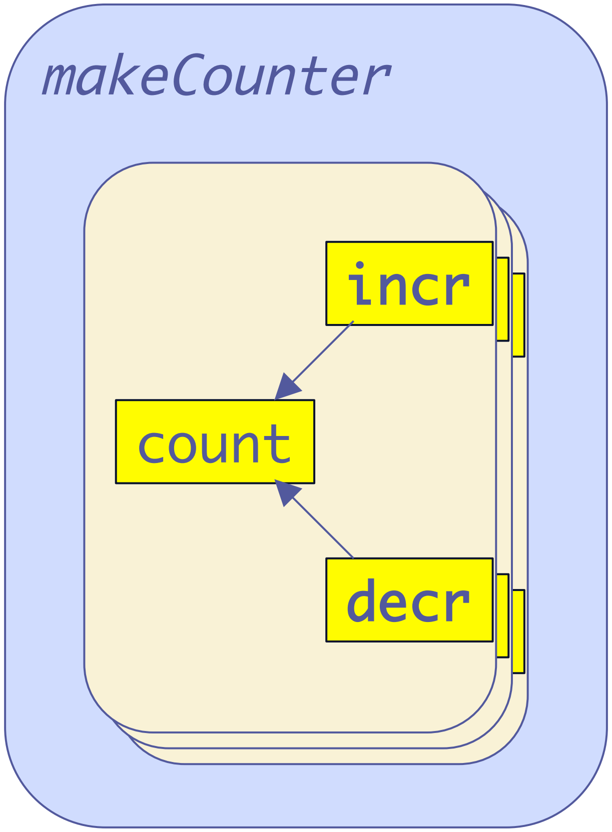
Reference graph ≡ Access graph

Only connectivity begets connectivity

Natural *Least Authority*

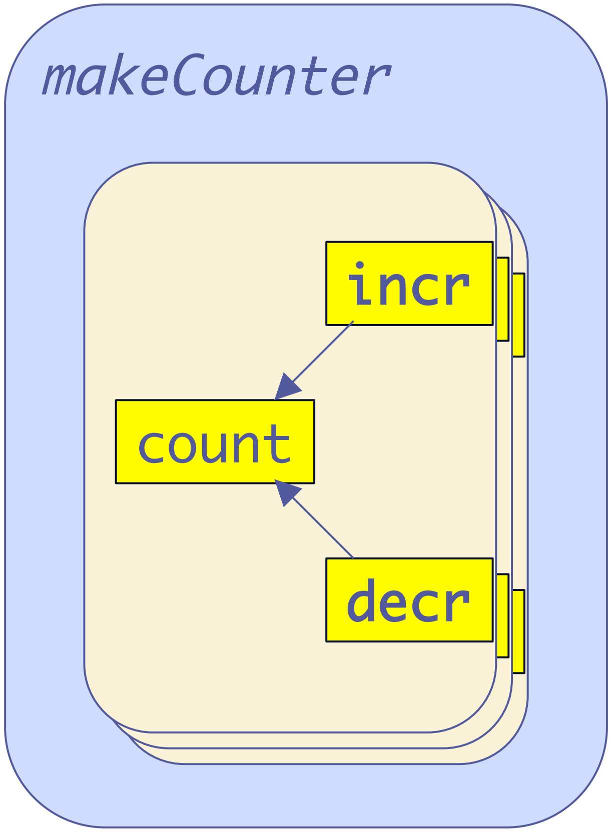
OO expressiveness for security patterns

Objects as Closures



```
function makeCounter() {
    var count = 0;
    return def({
        incr: function() { return ++count; },
        decr: function() { return –count; }
    });
}
```

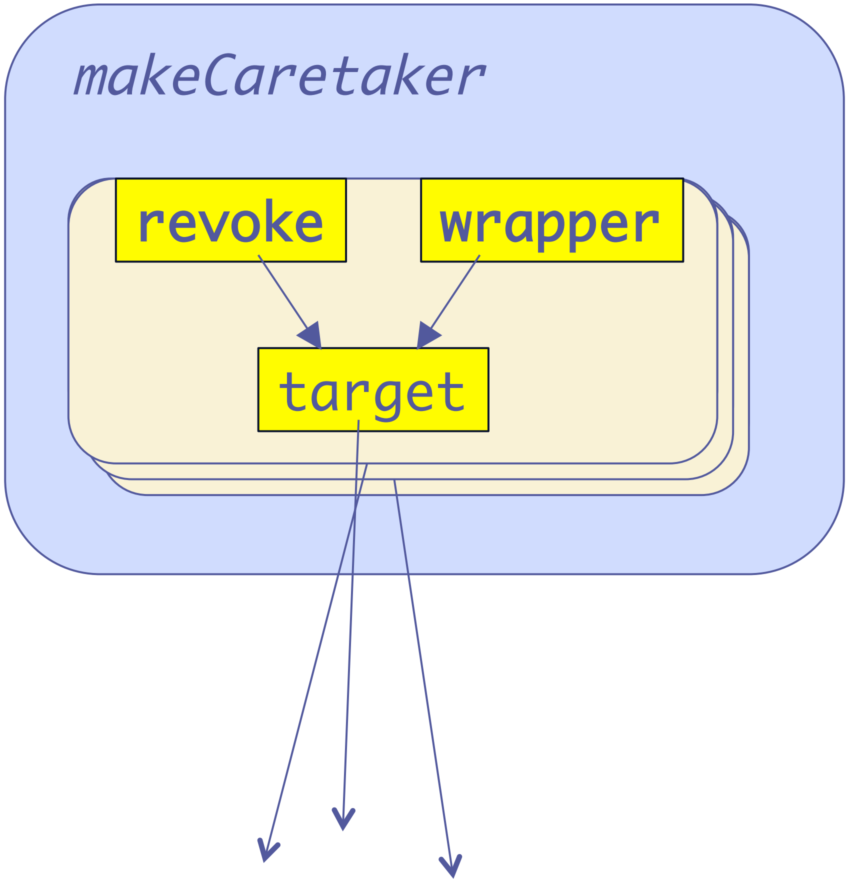
Objects as Closures



```
function makeCounter() {
    var count = 0;
    return def({
        incr: function() { return ++count; },
        decr: function() { return –count; }
    });
}
```

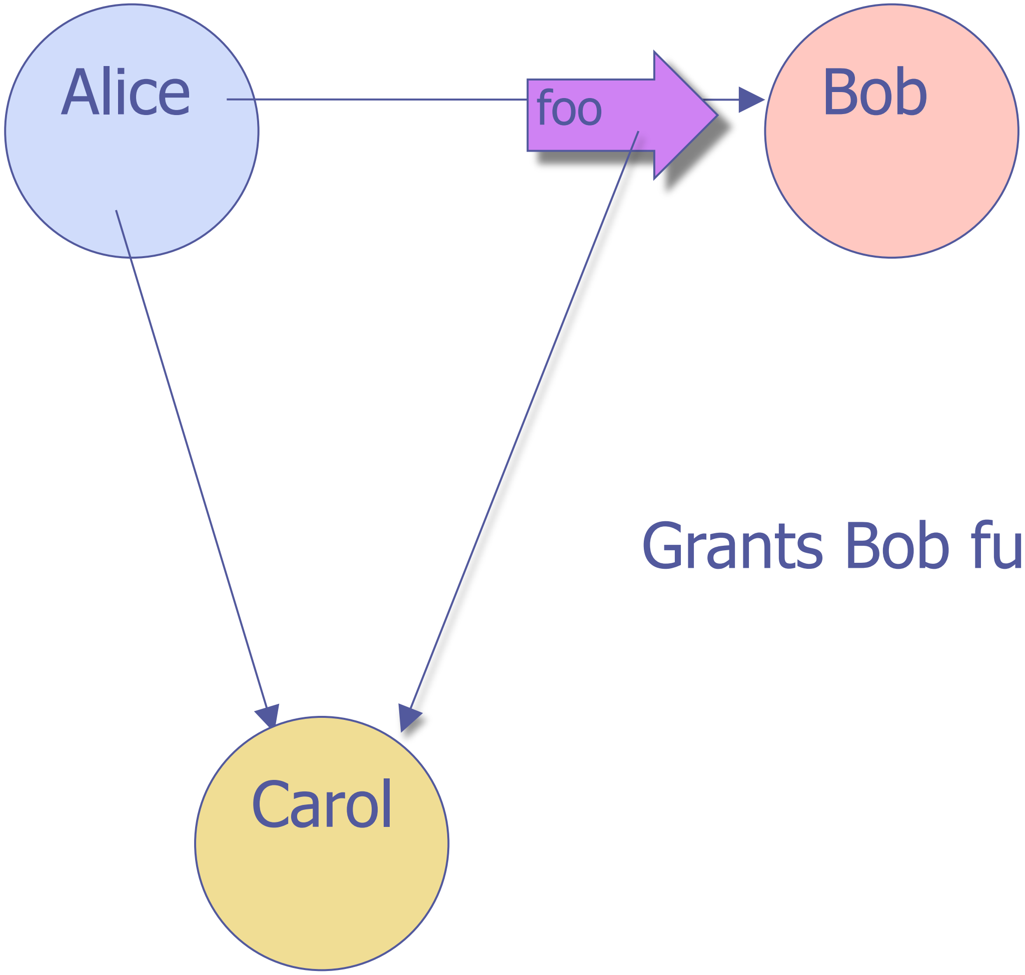
A *record* of *closures* hiding *state*
is a fine representation of an
object of *methods* hiding *instance vars*

Revocable Function Forwarder



```
function makeFnCaretaker(target) {
    return def({
        wrapper: function(…args) {
            return target(…args);
        },
        revoke: function() { target = null; }
    });
}
```

Unconditional Access

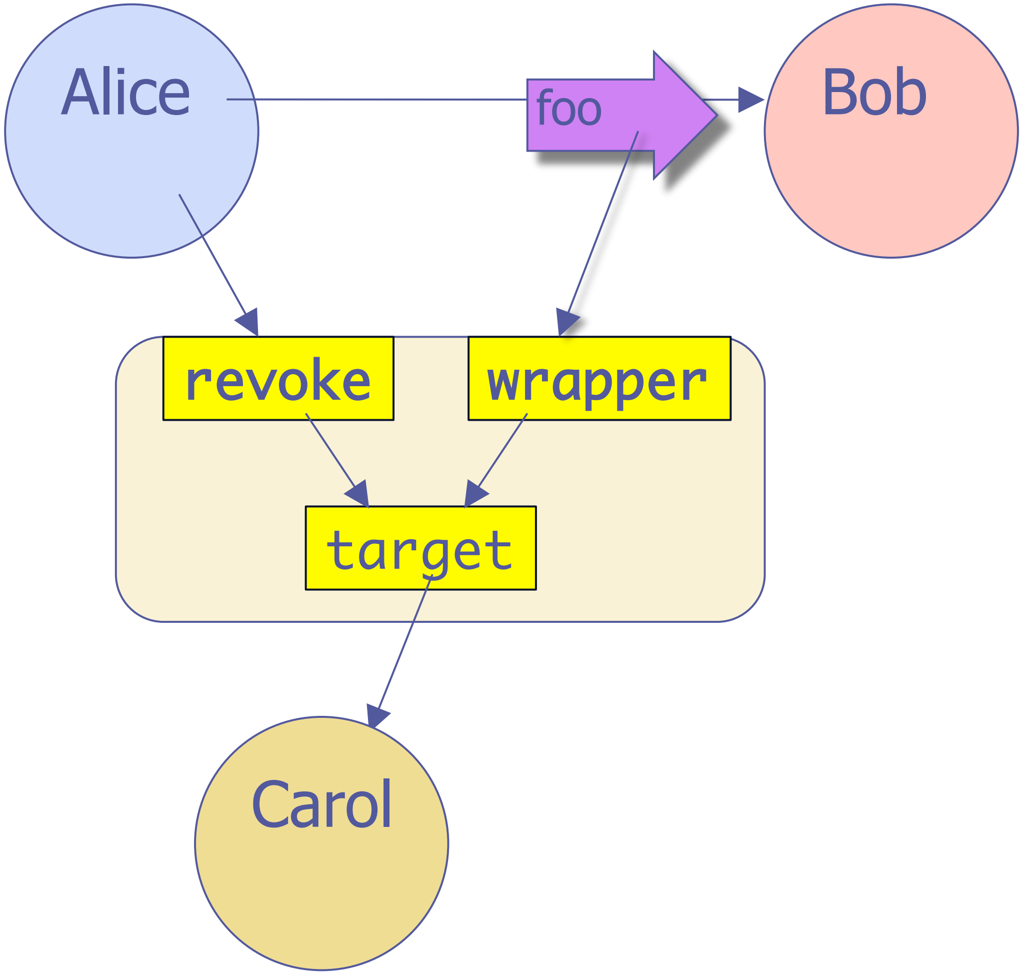


Alice says:

bob.foo(carol);

Grants Bob full access to Carol forever

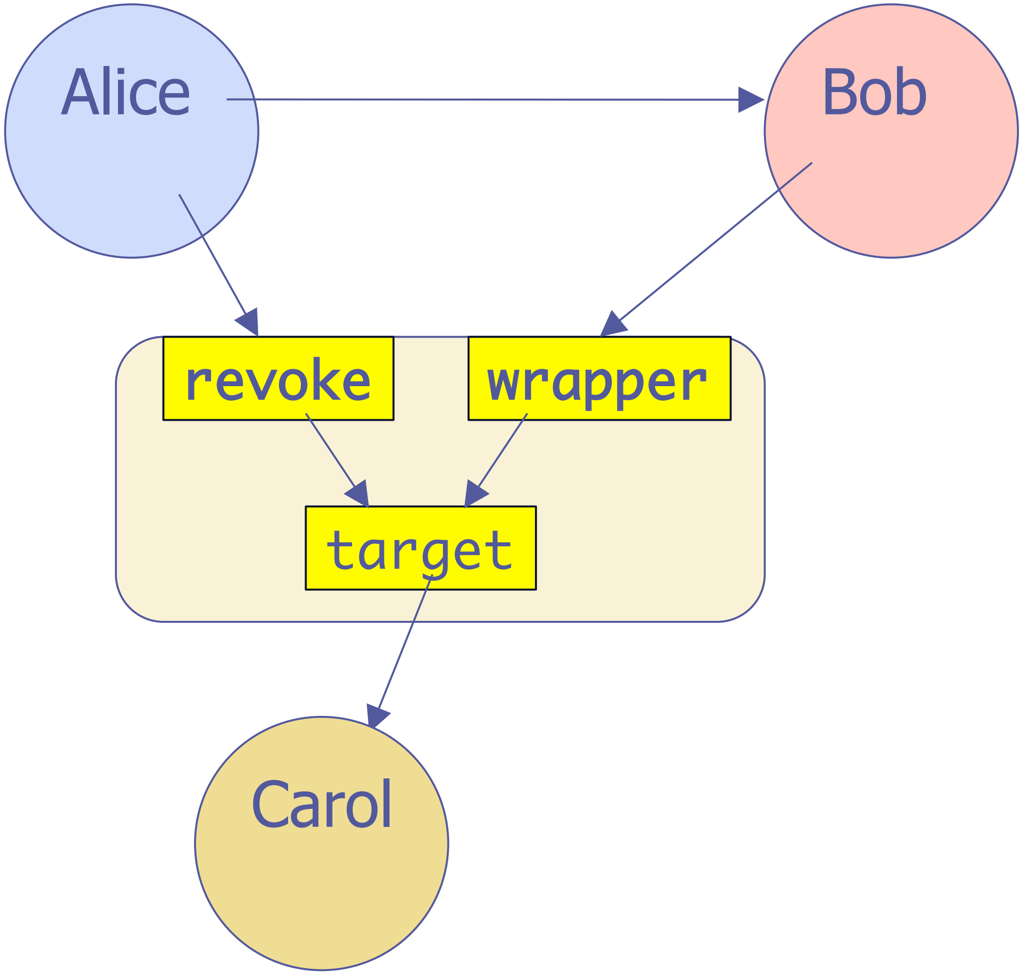
Revocability ≡ Temporal attenuation



Alice says:

```
var ct = makeCaretaker(carol);
bob.foo(ct.wrapper);
```

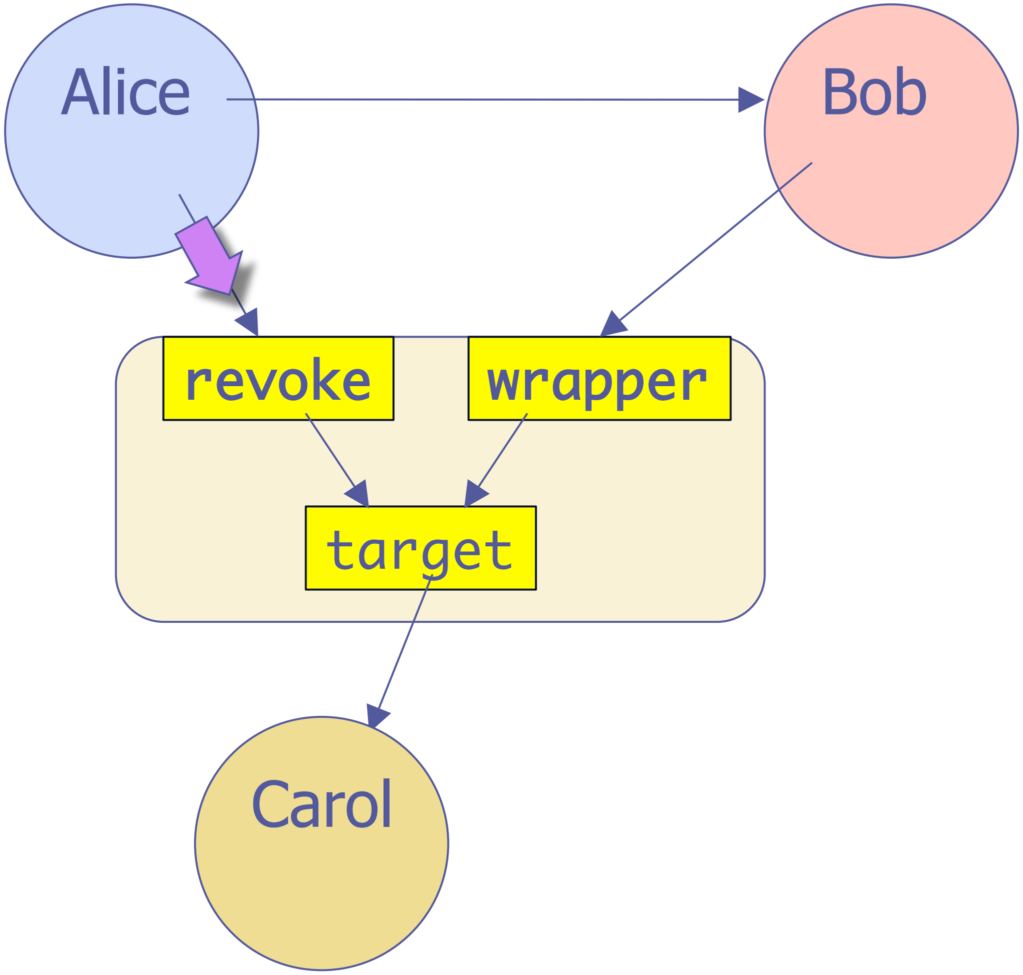
Revocability ≡ Temporal attenuation



Alice says:

```
var ct = makeCaretaker(carol);
bob.foo(ct.wrapper);
//…
```

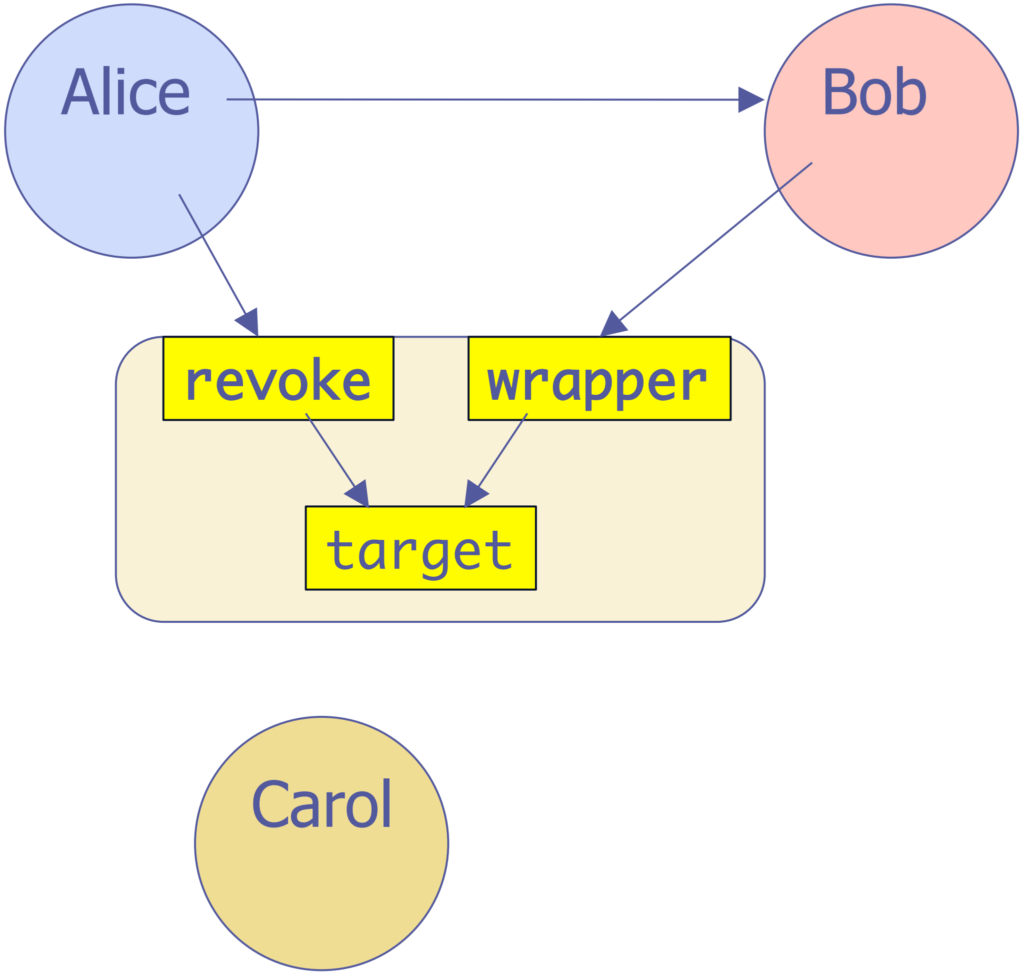
Revocability ≡ Temporal attenuation



Alice says:

```
var ct = makeCaretaker(carol);
bob.foo(ct.wrapper);
//…
ct.revoke();
```

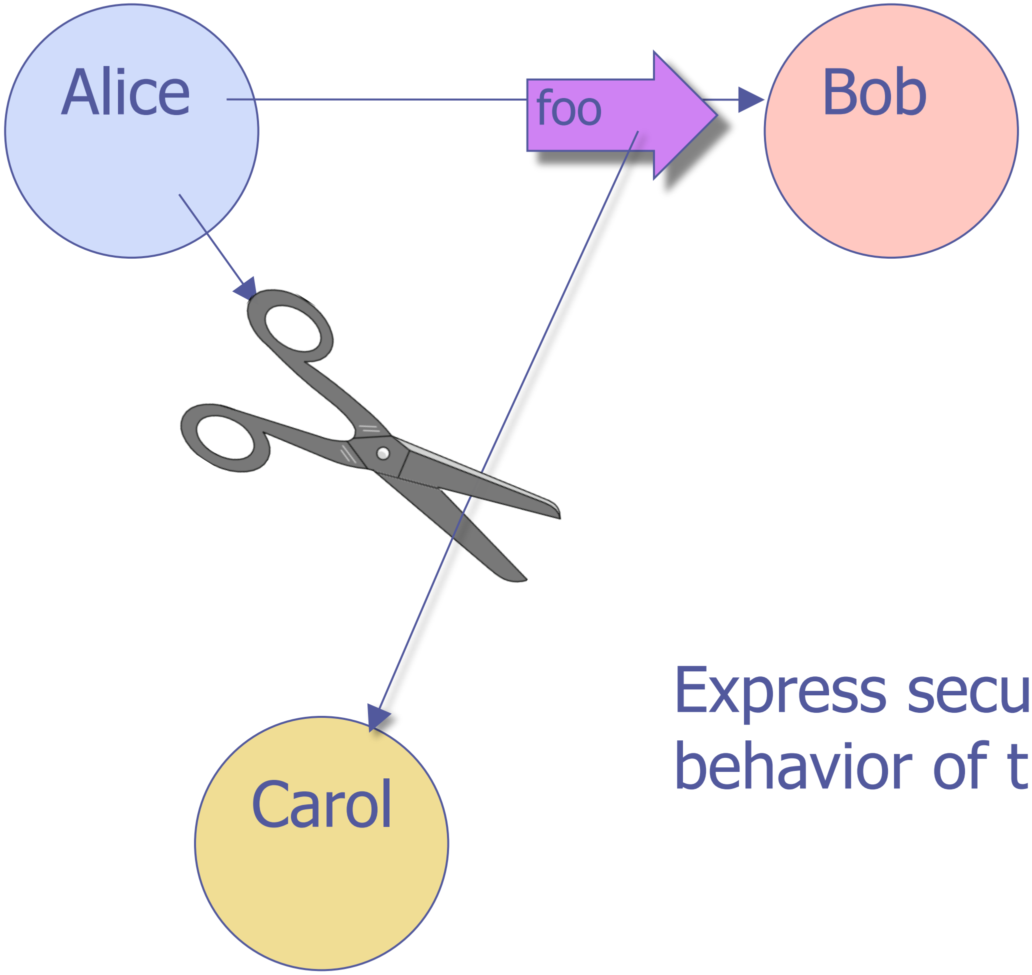
Revocability ≡ Temporal attenuation



Alice says:

```
var ct = makeCaretaker(carol);
bob.foo(ct.wrapper);
//…
ct.revoke();
```

Attenuators ≡ Access Abstractions

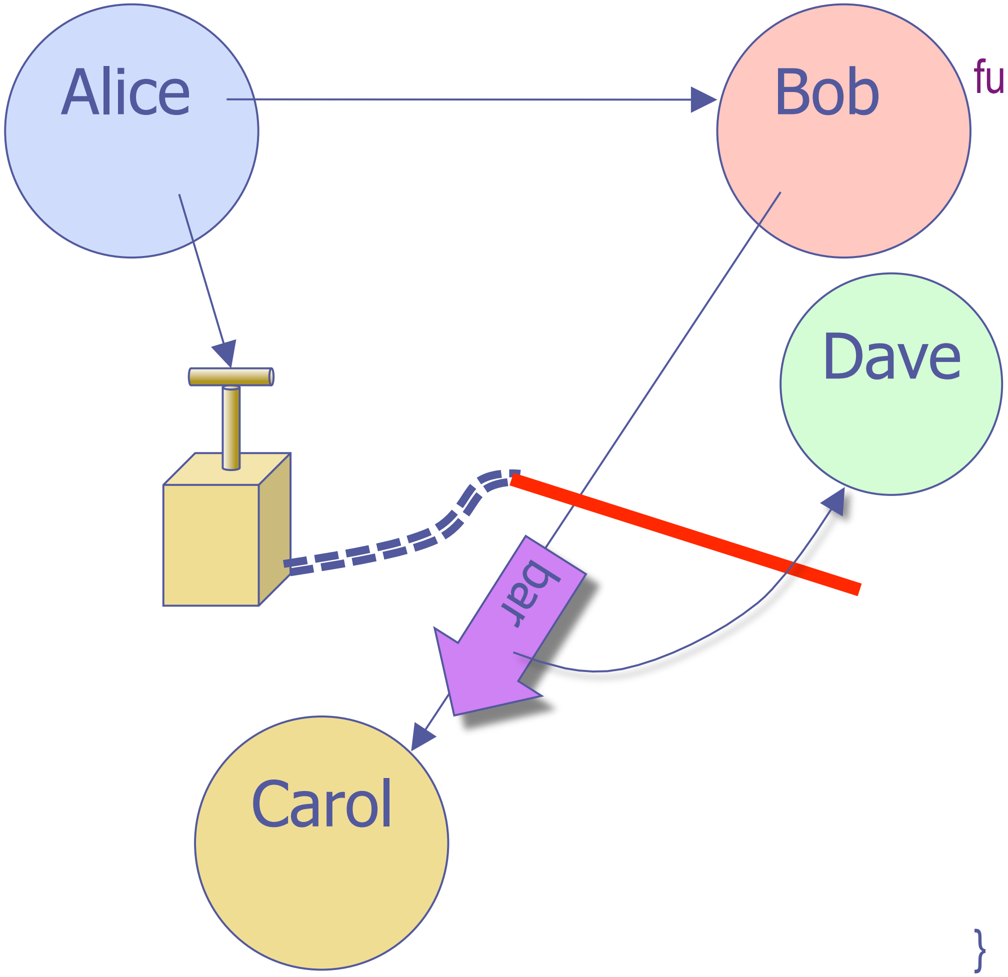


Alice says:

```
var ct = makeCaretaker(carol);
bob.foo(ct.wrapper);
```

Express security policy by the behavior of the objects you provide

Membranes: Transitive Interposition

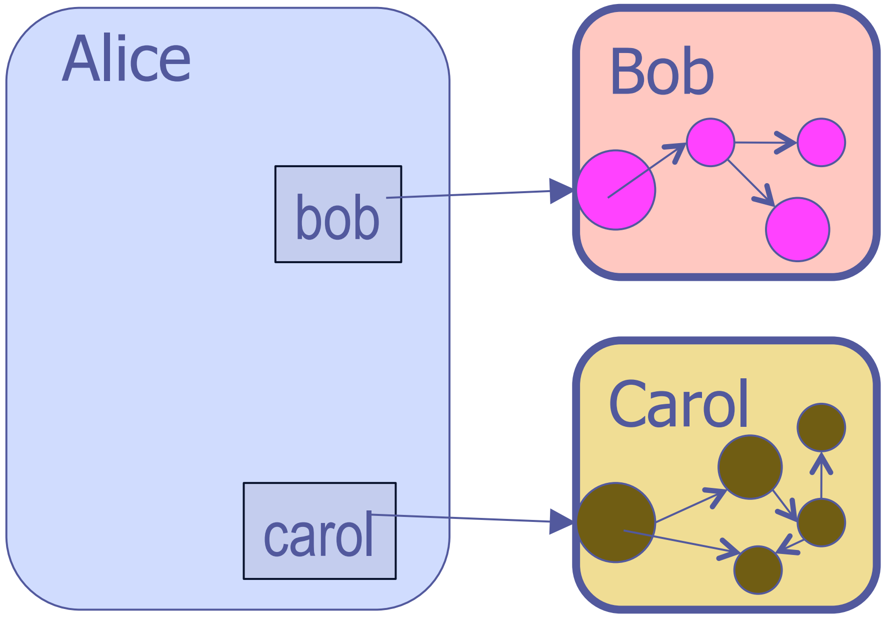


```
function makeFnMembrane(target) {
  var enabled = true;
  function wrap(wrapped) {
    if (wrapped !== Object(wrapped)) {
      return wrapped;
    }
    return function(…args) {
      if (!enabled) { throw new Error(“revoked”); }
      return wrap(wrapped(…args.map(wrap)));
  } }
  return def({
    wrapper: wrap(target),
    revoke: function() { target = null; }
  });
}
```

Attenuators Compose

```
function makeROFile(file) {
    return def({
        read: file.read,
        getLength: file.getLength
    });
}
var rorFile = makeROFile(revocableFile);
```

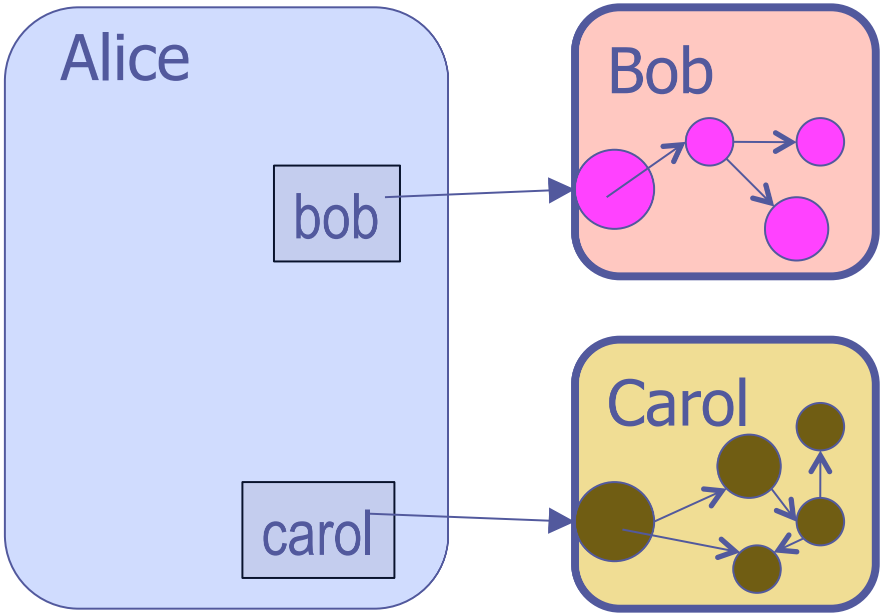
No powerful references by default



Alice says:

```
var bobSrc = //site B
var carolSrc = //site C
var bob = safeEval(bobSrc);
var carol = safeEval(carolSrc);
```

No powerful references by default



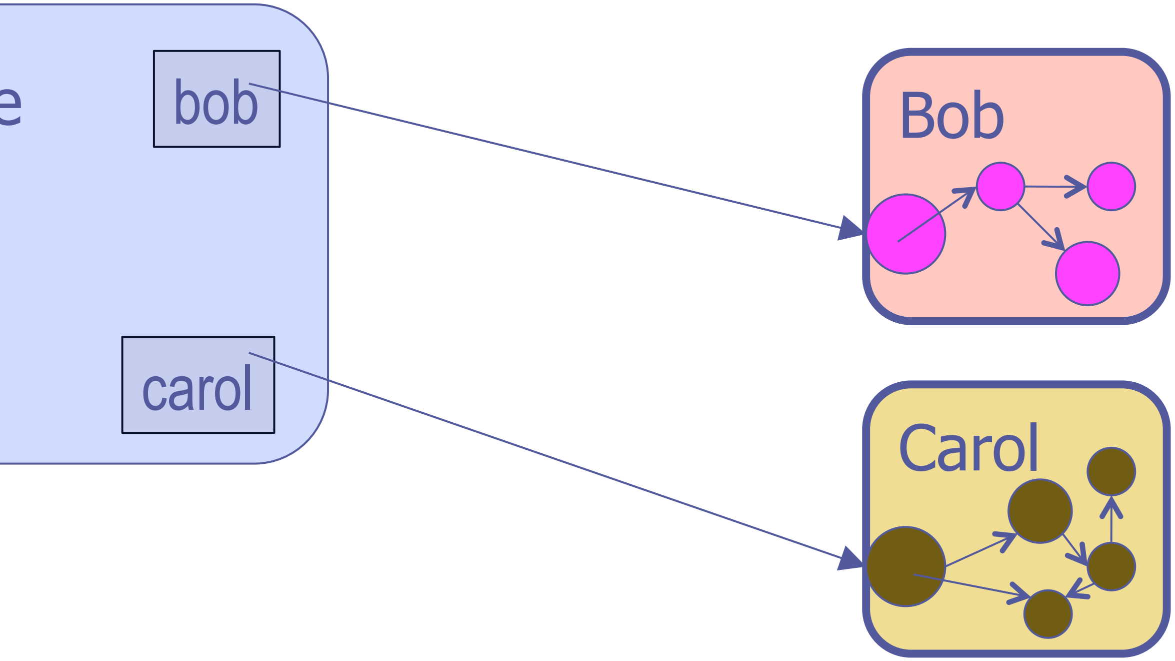
Alice says:

```
var bobSrc = //site B
var carolSrc = //site C
var bob = safeEval(bobSrc);
var carol = safeEval(carolSrc);
```

Bob and Carol are ***confined***.

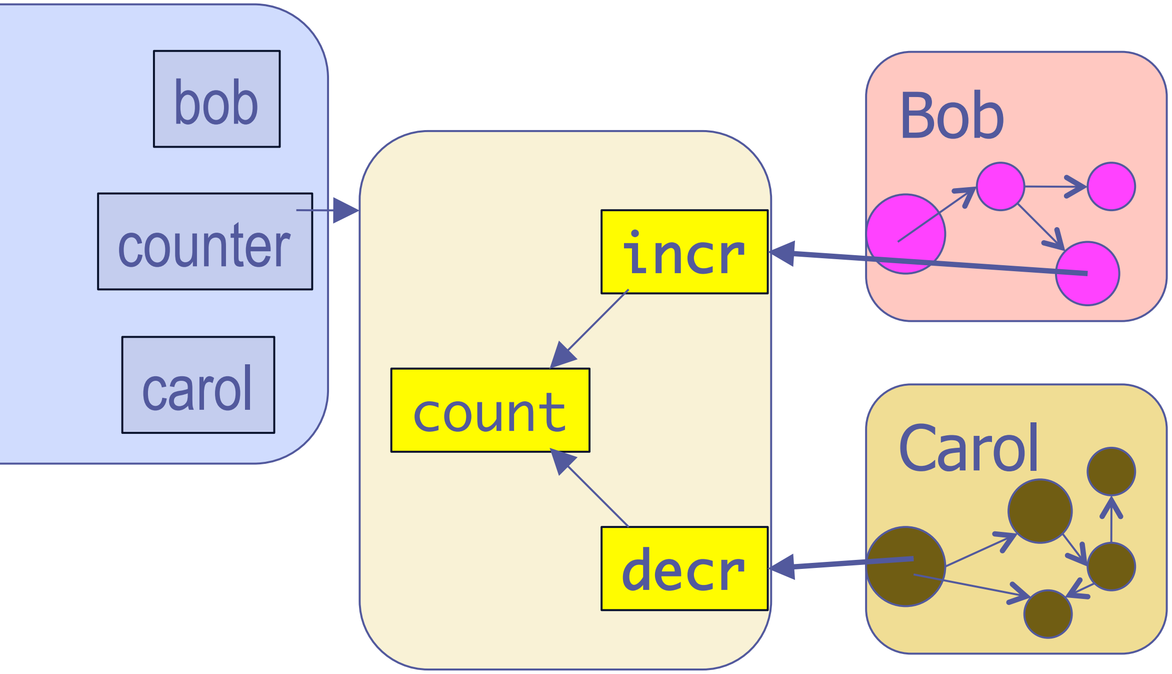
Only Alice controls how they can interact or get more connected.

No powerful references by default



Alice says:

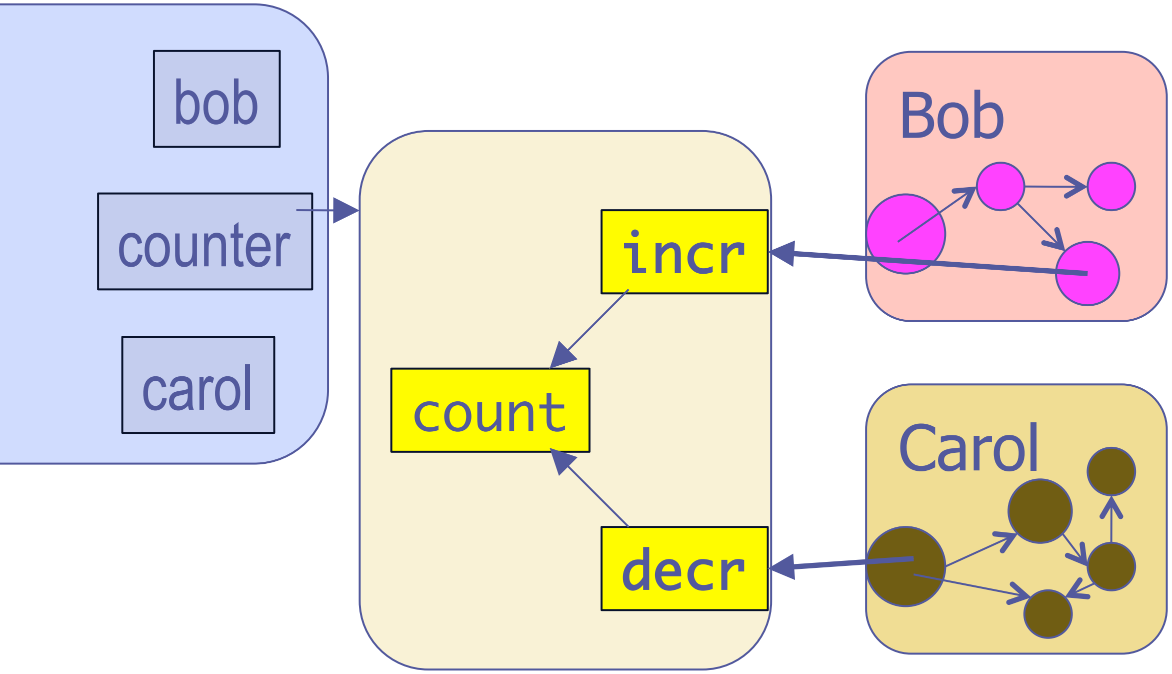
Only connectivity begets connectivity



Alice says:

```
var counter = makeCounter();
bob(counter.incr);
carol(counter.decr);
bob = carol = null;
```

Only connectivity begets connectivity



Alice says:

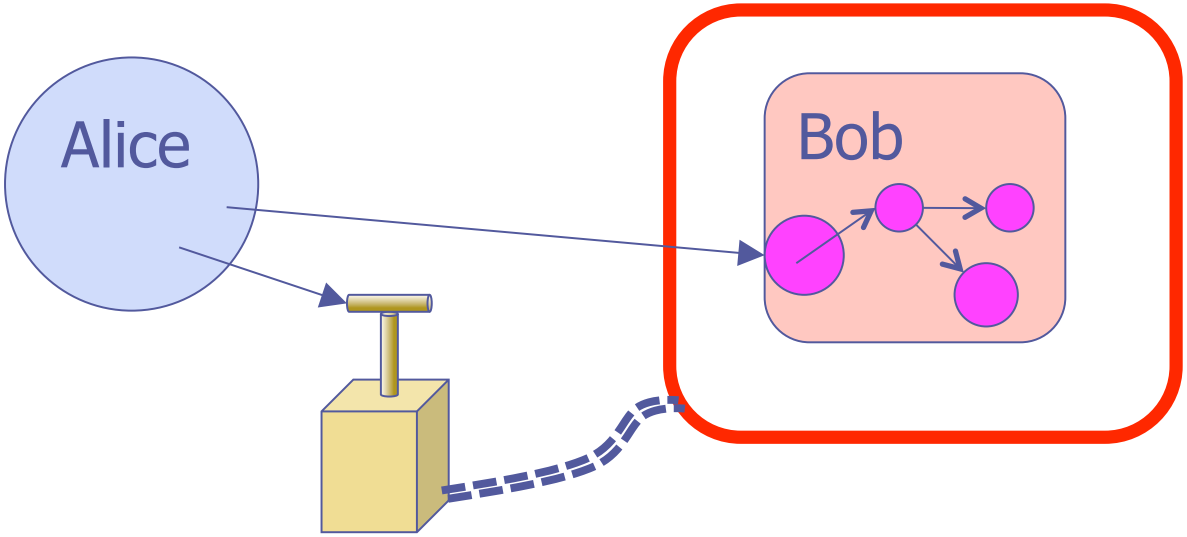
```
var counter = makeCounter();
bob(counter.incr);
carol(counter.decr);
bob = carol = null;
```

Bob can only count up and see result. Carol only down.

Alice can do both.

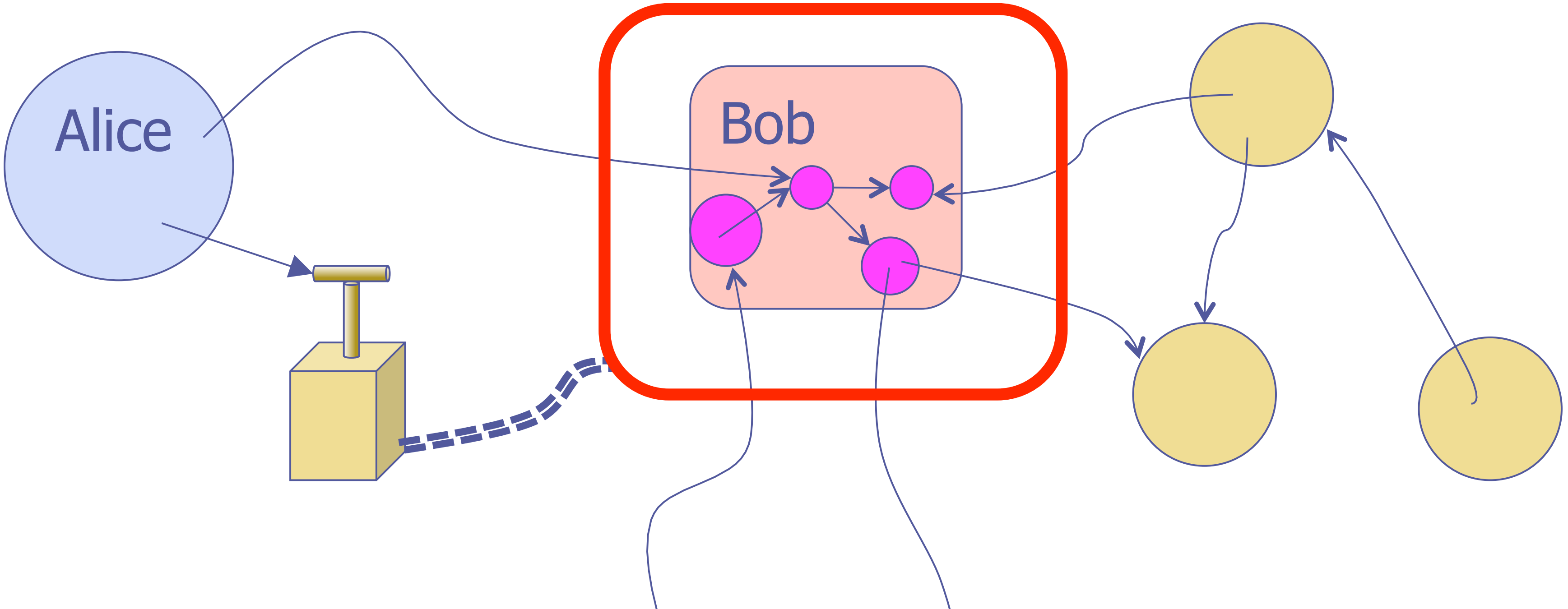
Membrane safeEval → compartment

```
var compartment = makeMembrane(safeEval);
var vbob = compartment.wrapper(bobSrc);
```



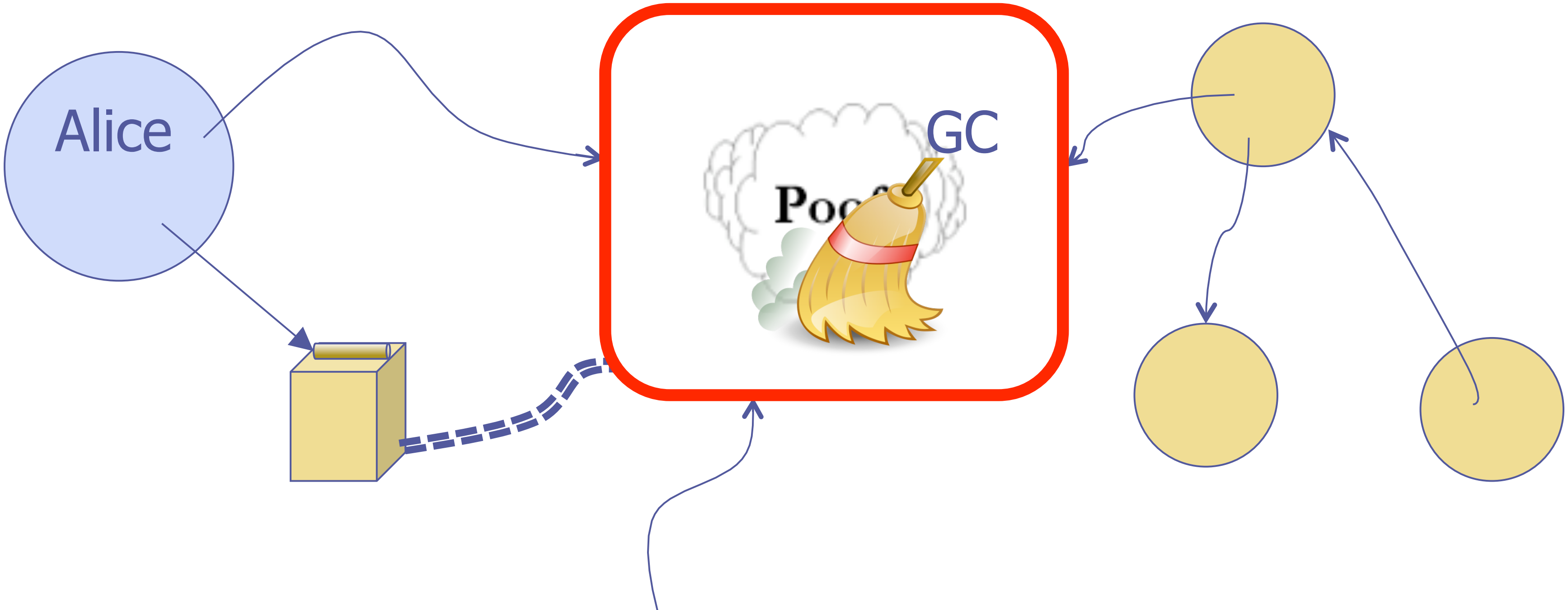
Membrane safeEval → compartment

```
var compartment = makeMembrane(safeEval);
var vbob = compartment.wrapper(bobSrc);
//…
```

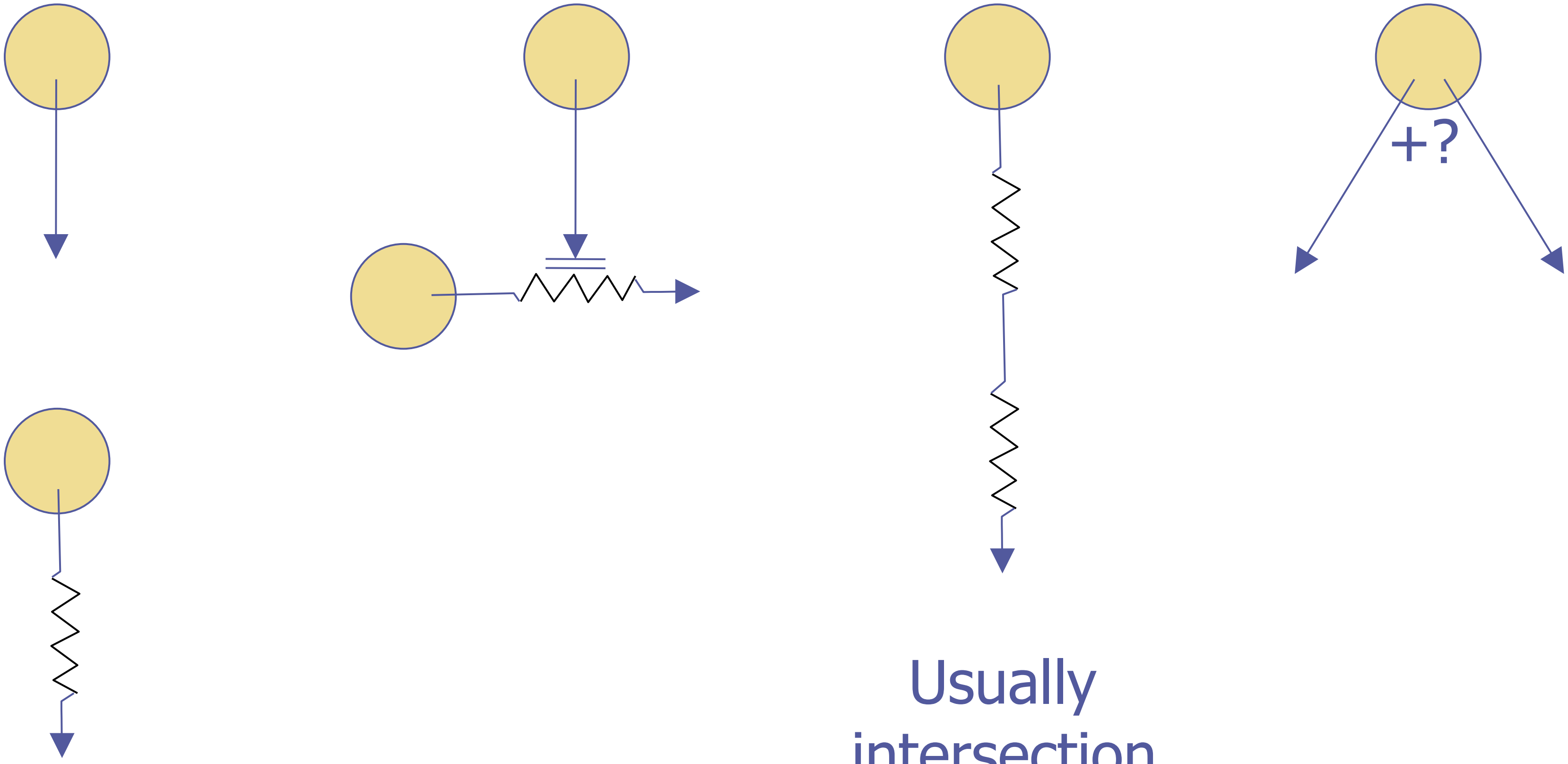


Membrane safeEval → compartment

```
var compartment = makeMembrane(safeEval);
var vbob = compartment.wrapper(bobSrc);
//…
compartment.revoke();
```

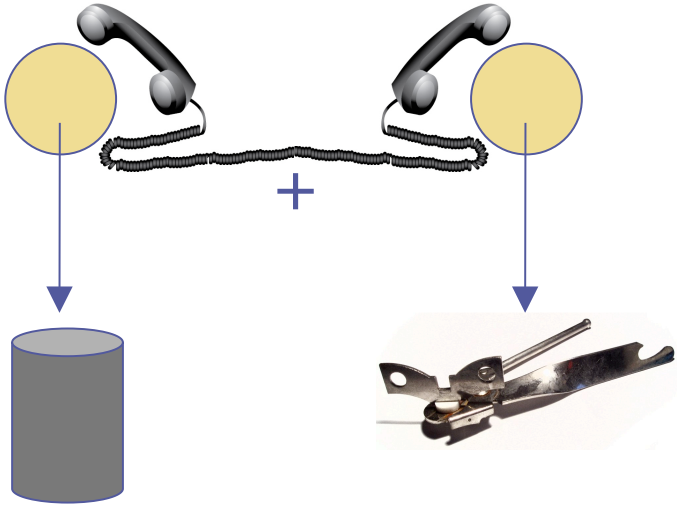


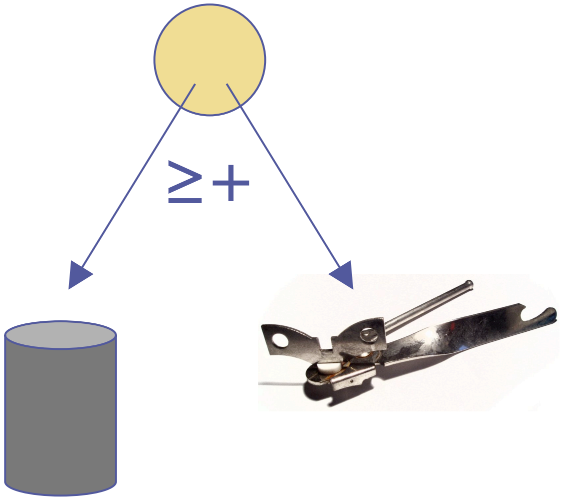
Composing Authority



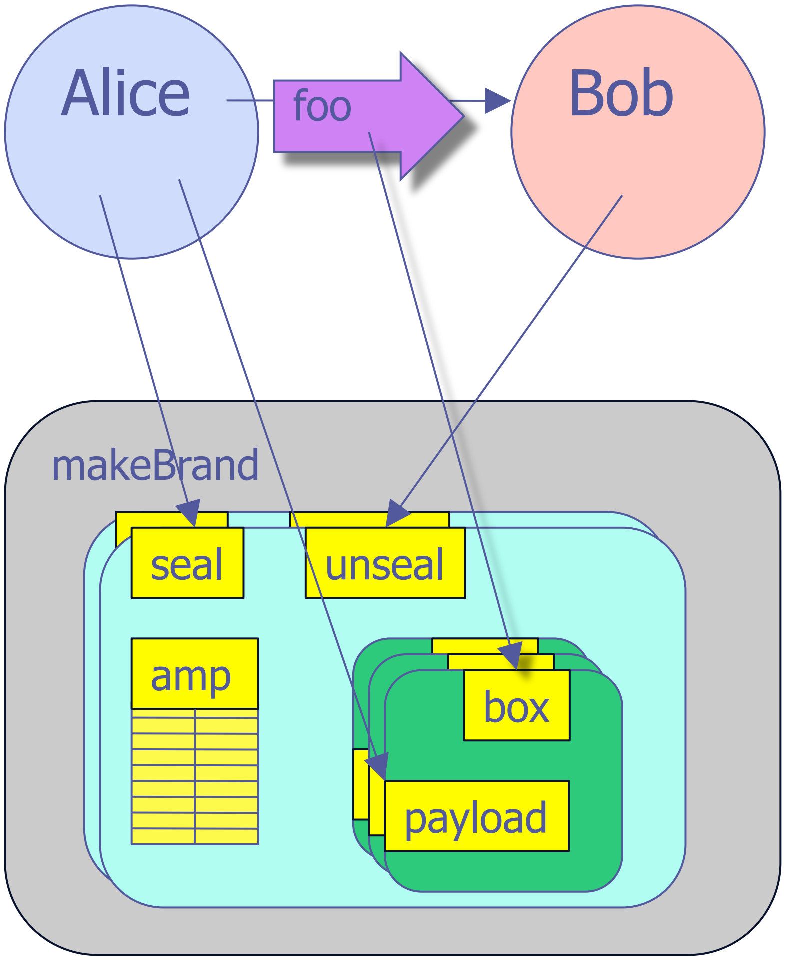
Usually intersection

Rights Amplification





Rights Amplification



```
function makeBrand() {
    var amp = WeakMap();
    function seal(payload) {
        var box = def({});
        amp.set(box, payload);
        return box;
    }
    function unseal(box) {
        return amp.get(box);
    }
    return def({seal: seal, unseal: unseal});
}
```

Dr. SES
Distributed Resilient Secure EcmaScript

Most suspicion is not within an address space

Stretch reference graph between machines

Preserve distributed "memory safety"

Dr. SES
Distributed Resilient Secure EcmaScript

	Shared State	**Message Passing**
Blocking	C++/pthreads Java, C#, Mozart/Oz JoCAML, Polyphonic C#	*Blocking receive* CSP, Occam, CCS Erlang, Scala, Go
Non-blocking	*Soft Transactional Mem* Argus, Fortress, X10	*Comm Event Loops* Actors, AmbientTalk E, Waterken **Ajax**

Dr. SES
Distributed Resilient Secure EcmaScript

	Shared State	**Message Passing**
Blocking	C++/pthreads Java, C#, Mozart/Oz JoCAML, Polyphonic C#	*Blocking receive* CSP, Occam, CCS Erlang, Scala, Go
Non-blocking	*Soft Transactional Mem* Argus, Fortress, X10	*Comm Event Loops* Actors, AmbientTalk E, Waterken **Ajax**

No conventional deadlocks or memory races

Dr. SES
Distributed Resilient Secure EcmaScript

	Shared State	**Message Passing**
Blocking	C++/pthreads Java, C#, Mozart/Oz JoCAML, Polyphonic C#	*Blocking receive* CSP, Occam, CCS Erlang, Scala, Go
Non-blocking	*Soft Transactional Mem* Argus, Fortress, X10	*Comm Event Loops* Actors, AmbientTalk E, Waterken **Ajax, Dr. SES**

No conventional deadlocks or memory races

```
var result = bob.foo(carol);        // do it immediately

var resultP = bobP ! foo(carol);    // do it eventually
```

Async object ops as JSON/REST ops

Object operations

```
var resultP = bob ! foo;

var resultP = bob ! foo(carol);


Q.when(resultP, function(result) {
  …result…
}, function (ex) {
  …ex…
});
```

https: JSON/RESTful operations

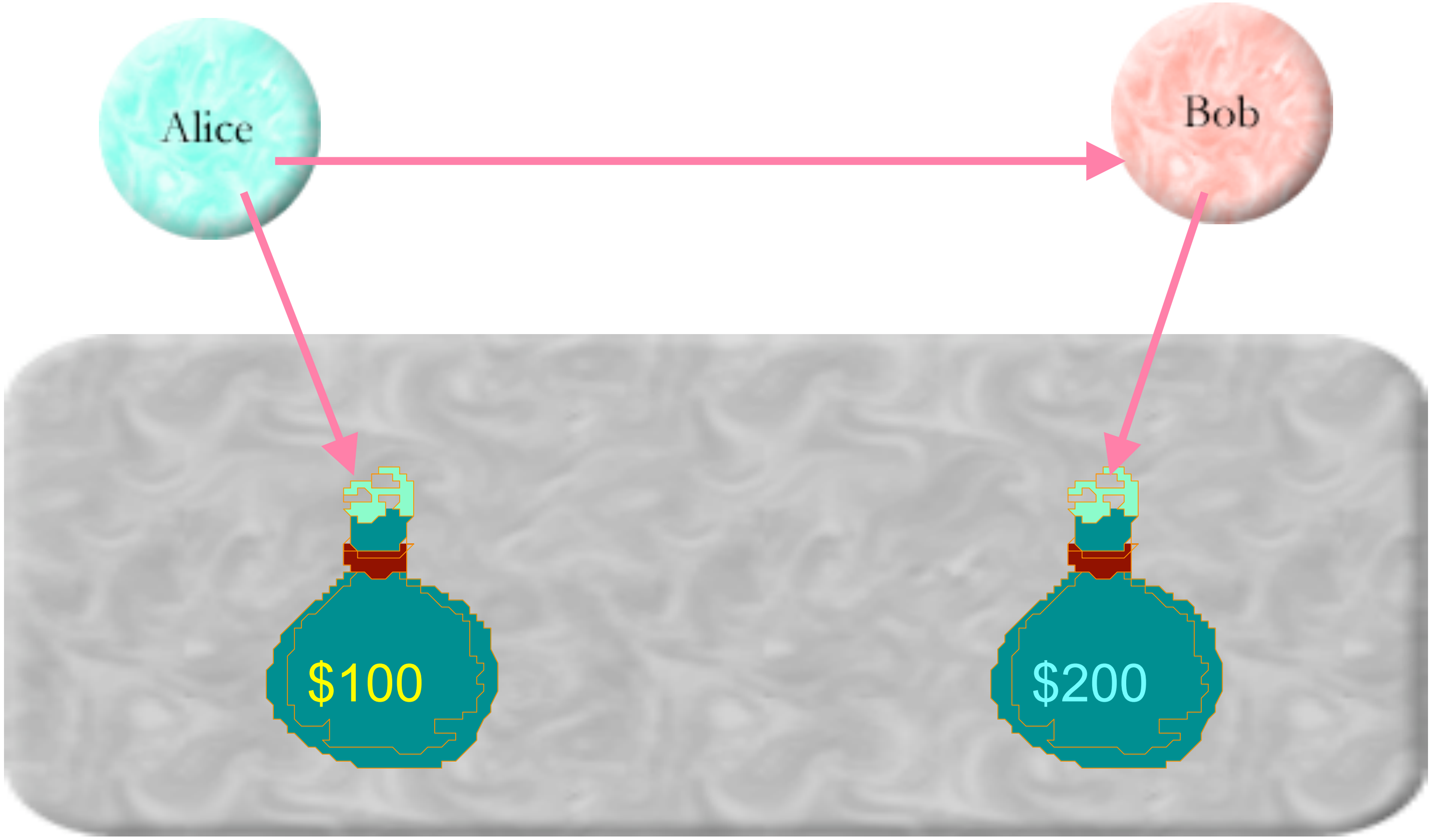
```
GET https://…q=foo

POST https://…q=foo {…}
```

Register for notification using

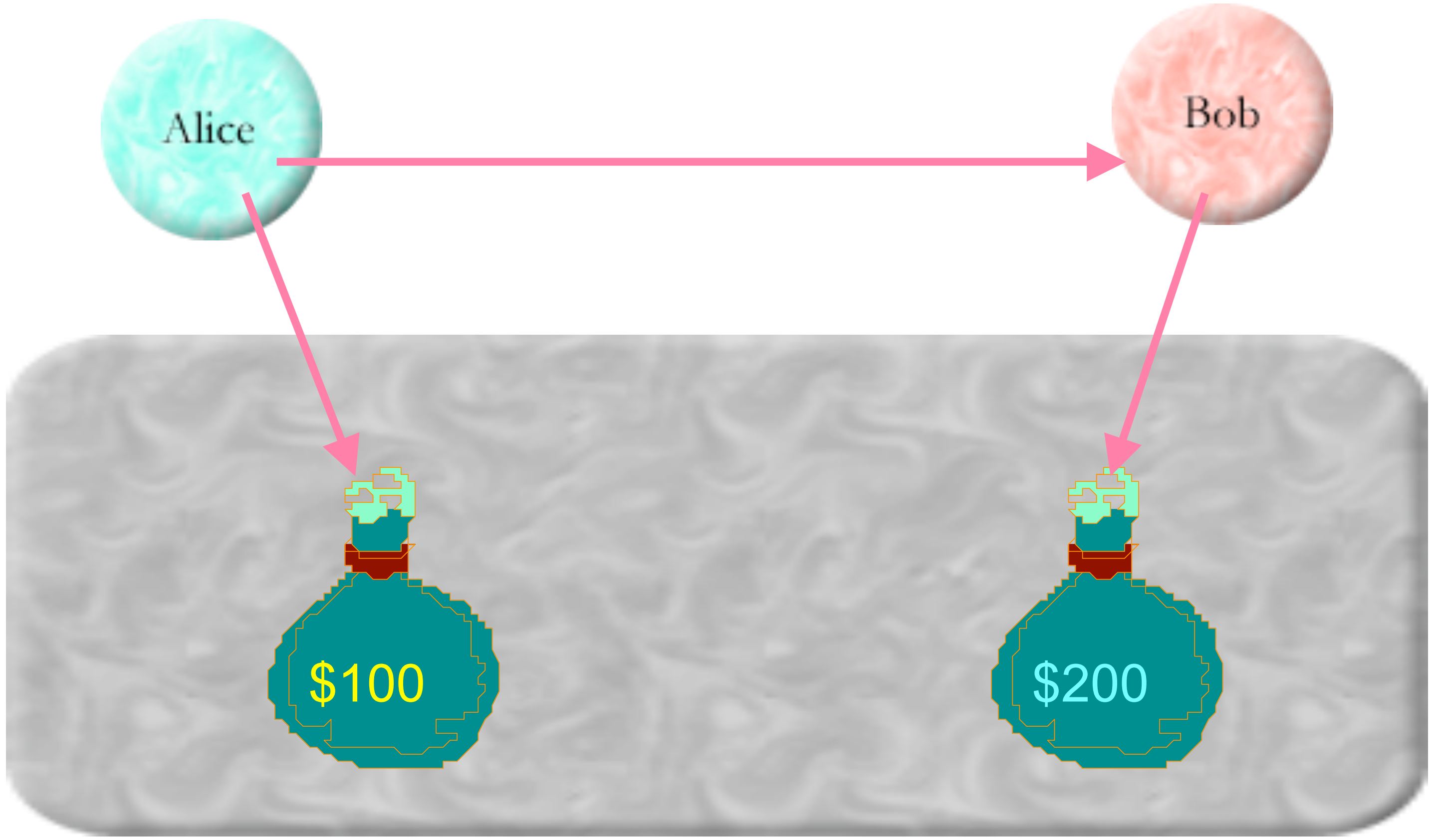
```
xhr.onreadystatechange = …
```

Distributed Secure Currency



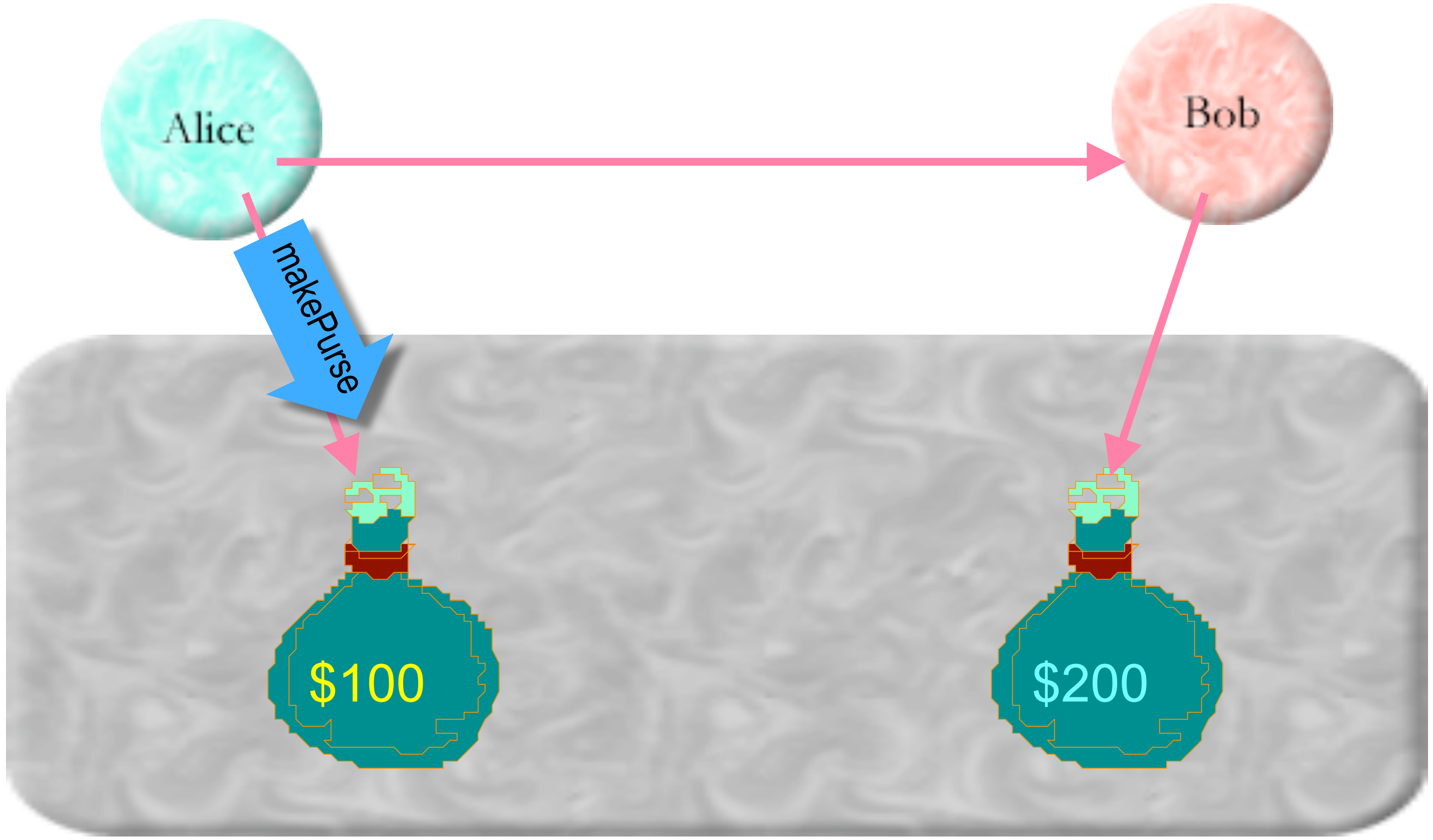
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
```



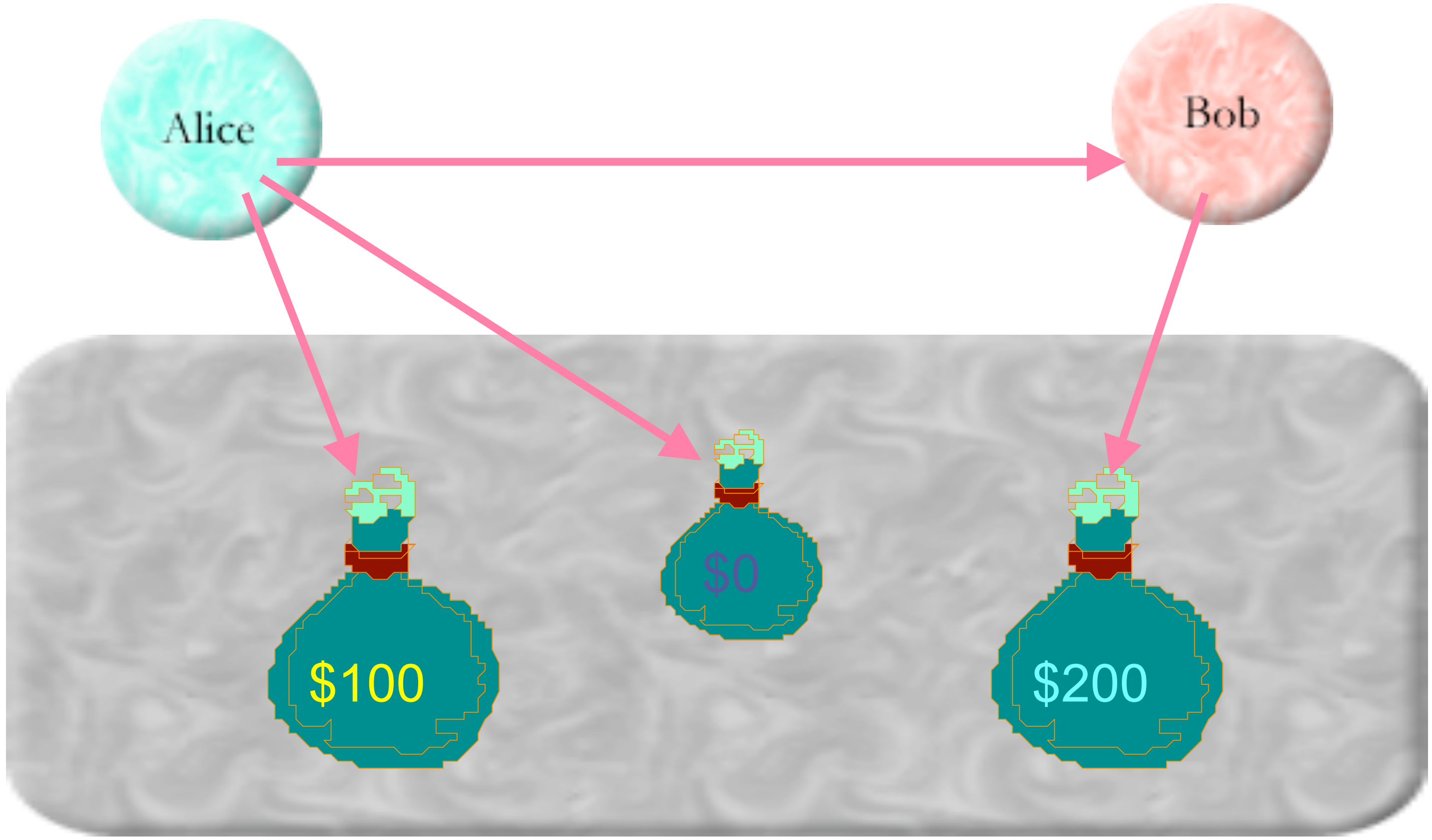
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
```



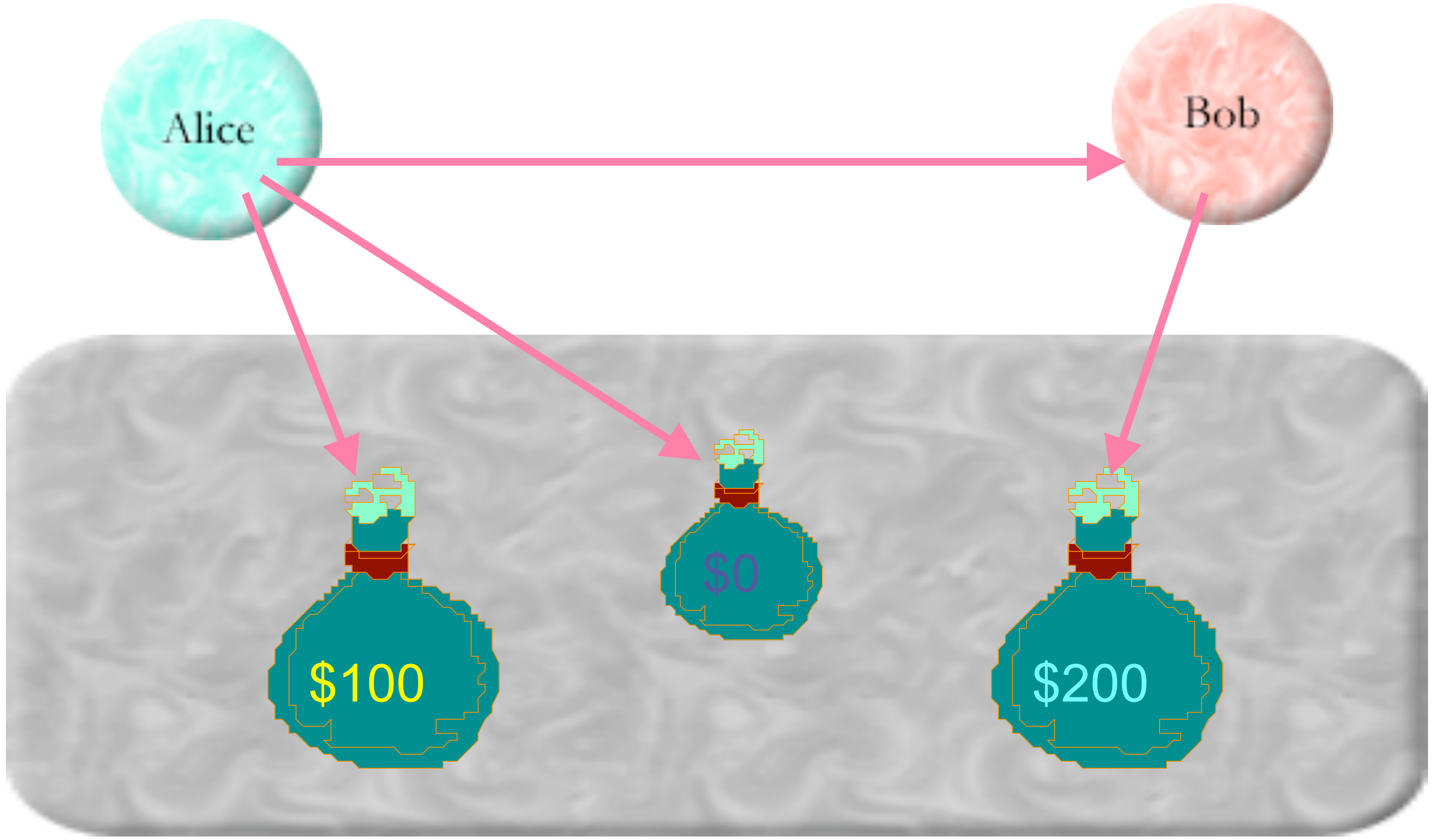
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
```



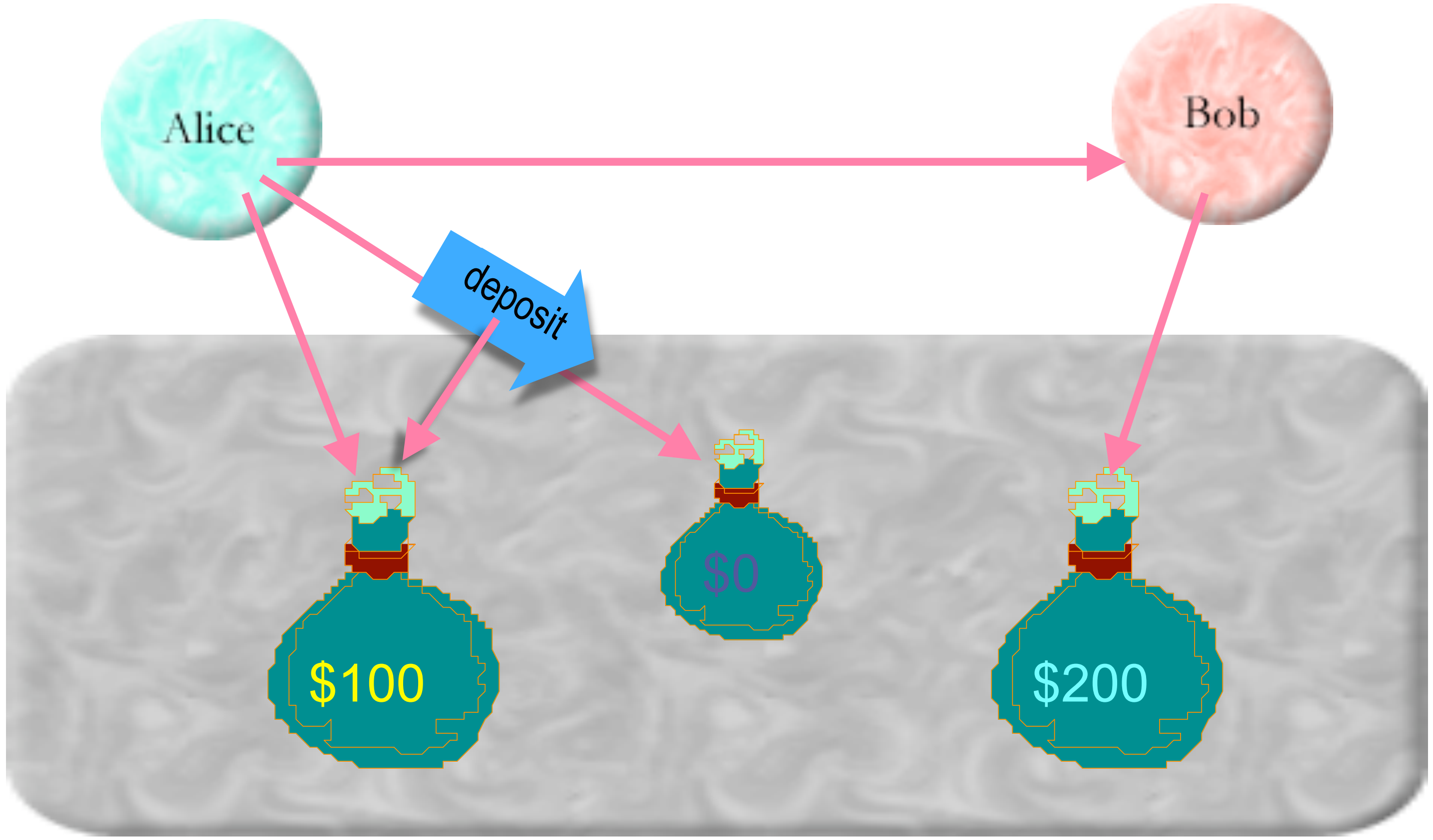
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
```



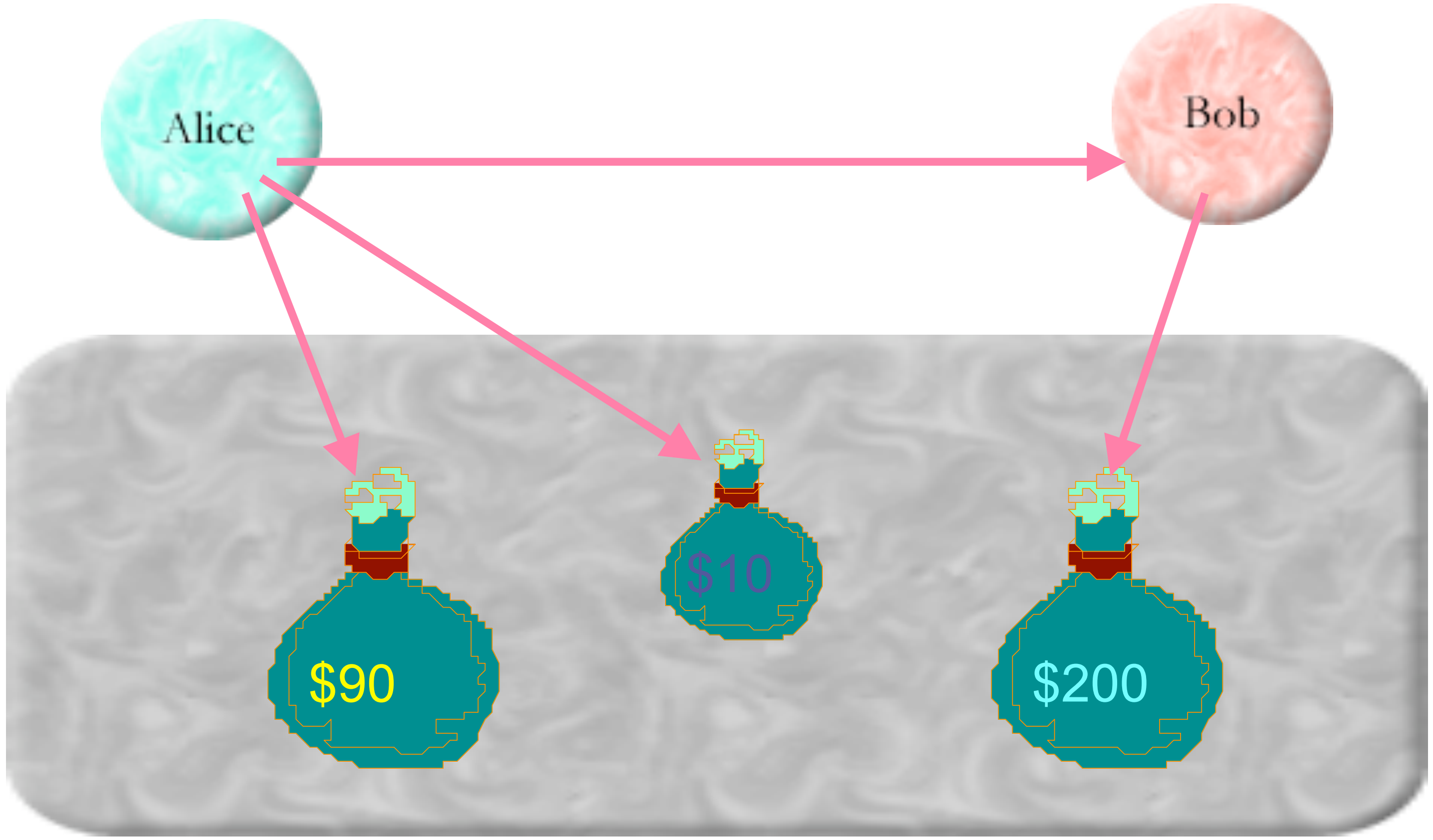
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
```



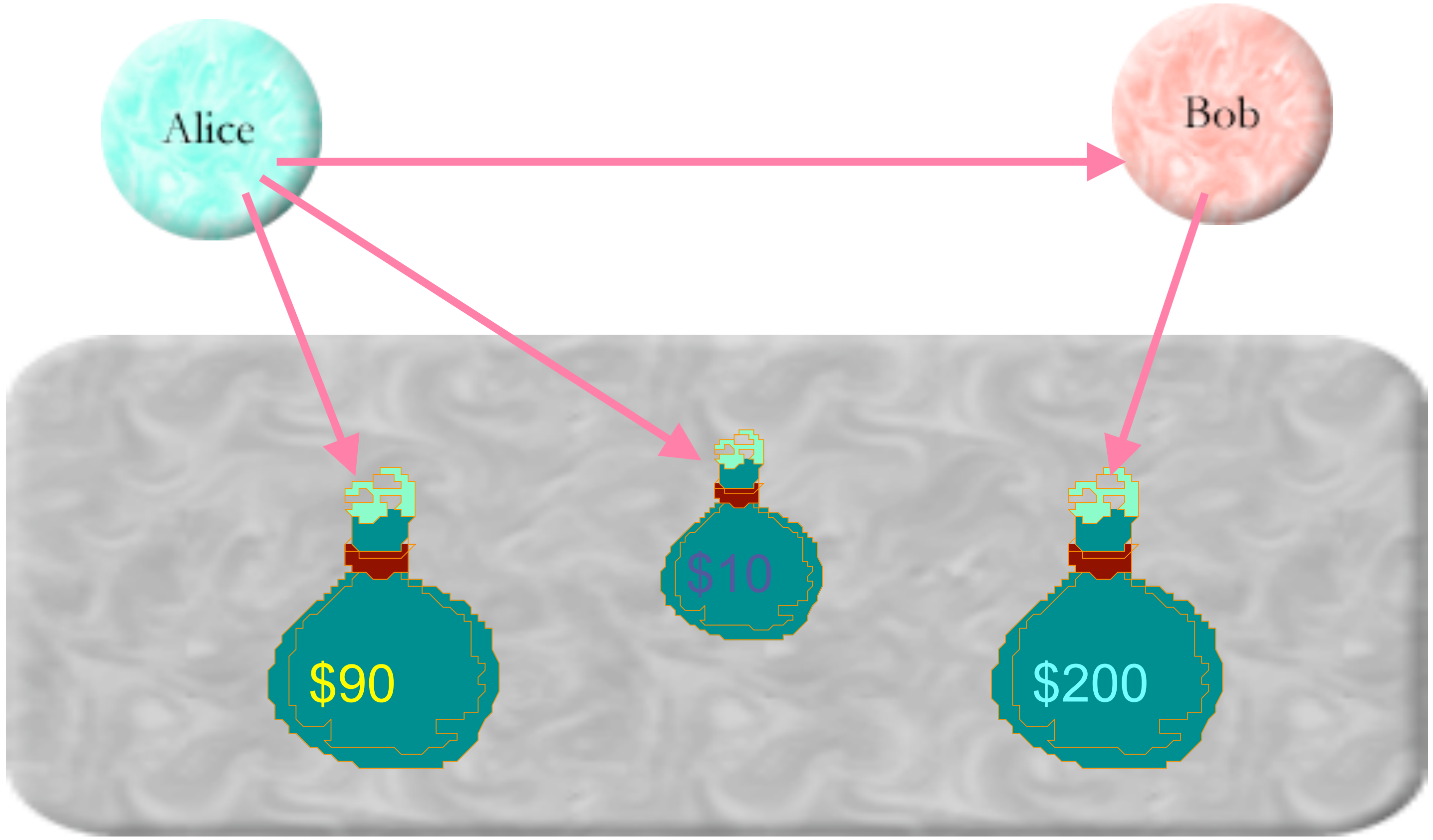
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
```



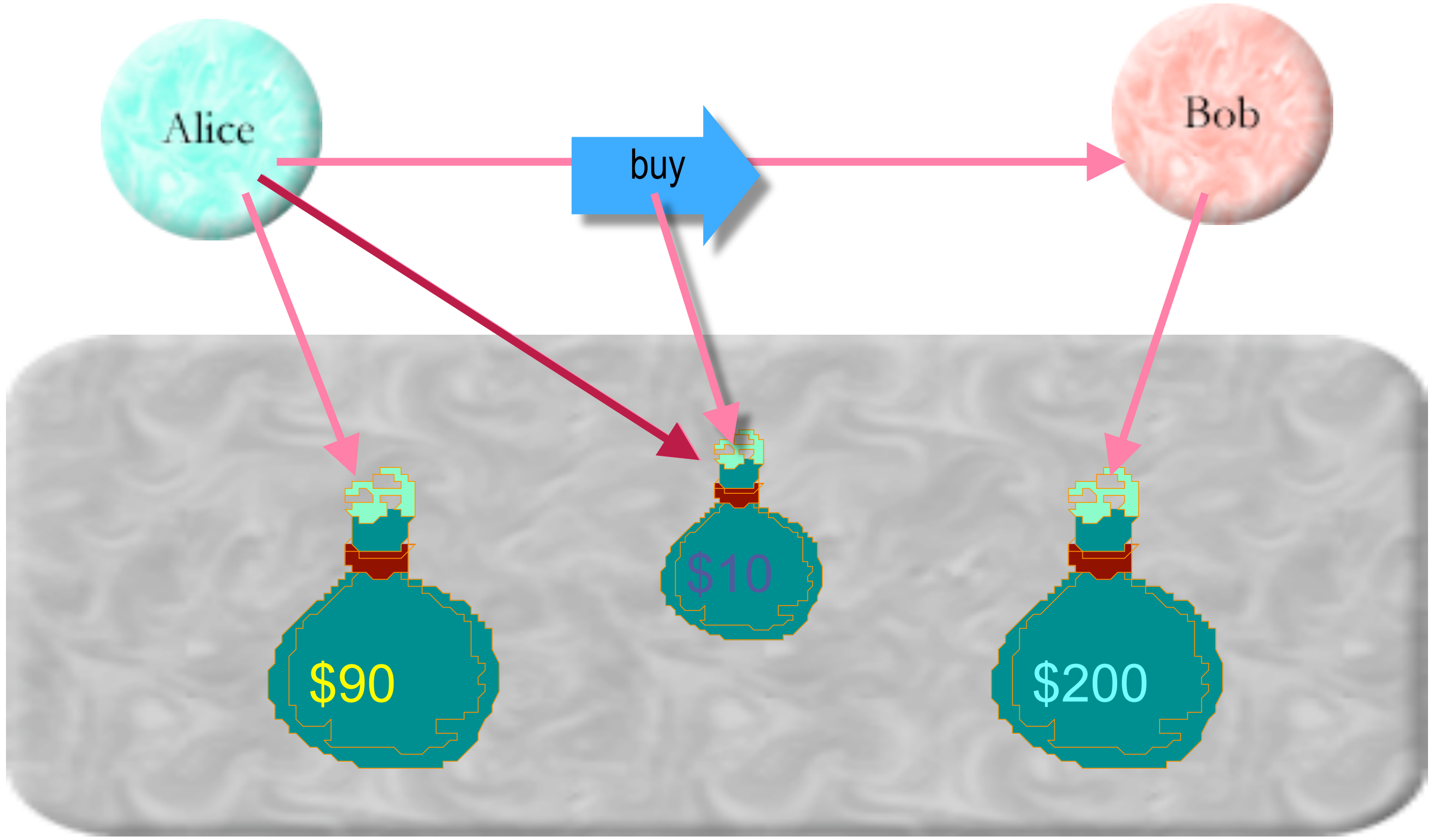
Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```



Distributed Secure Currency

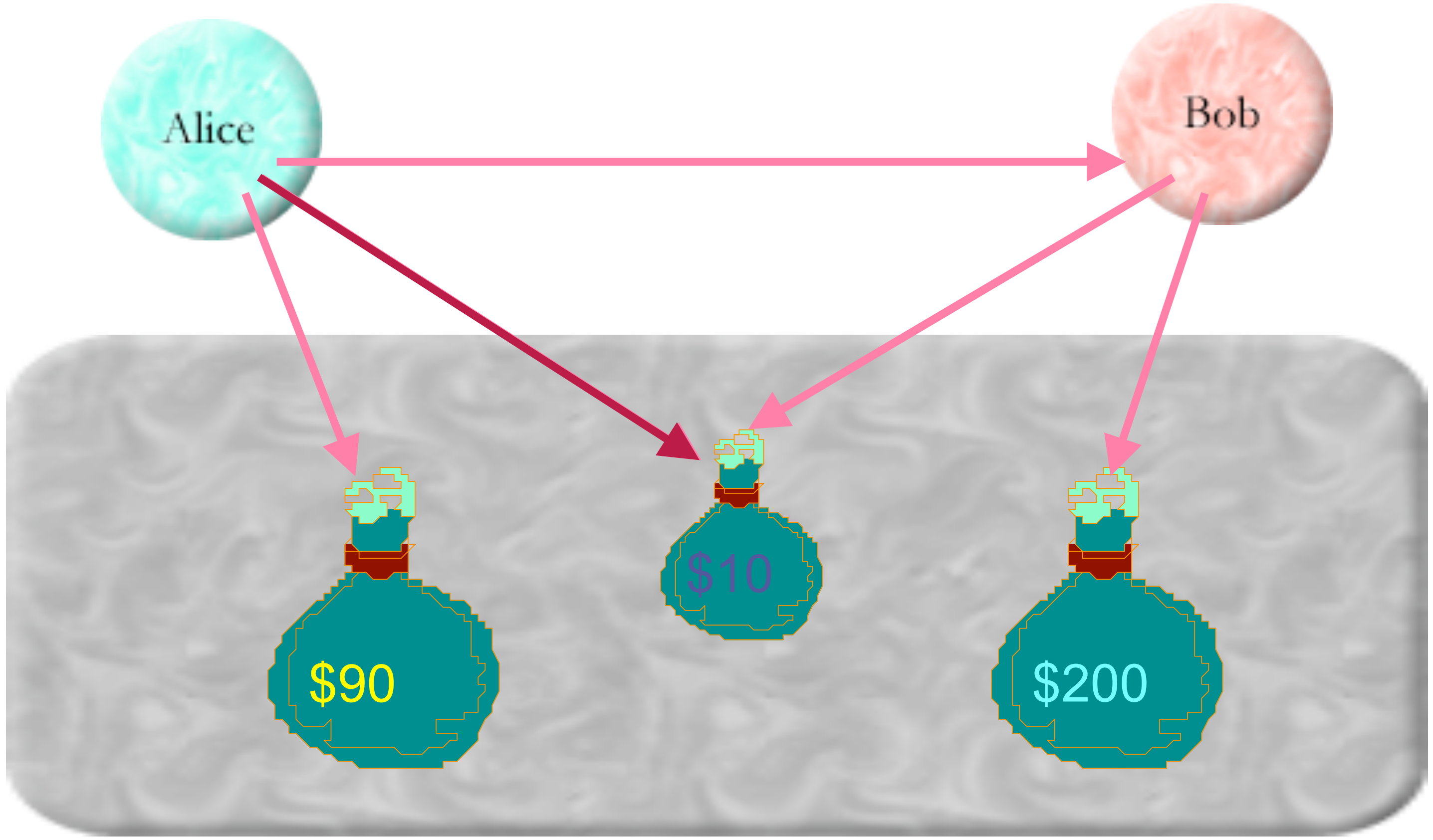
```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```



Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```

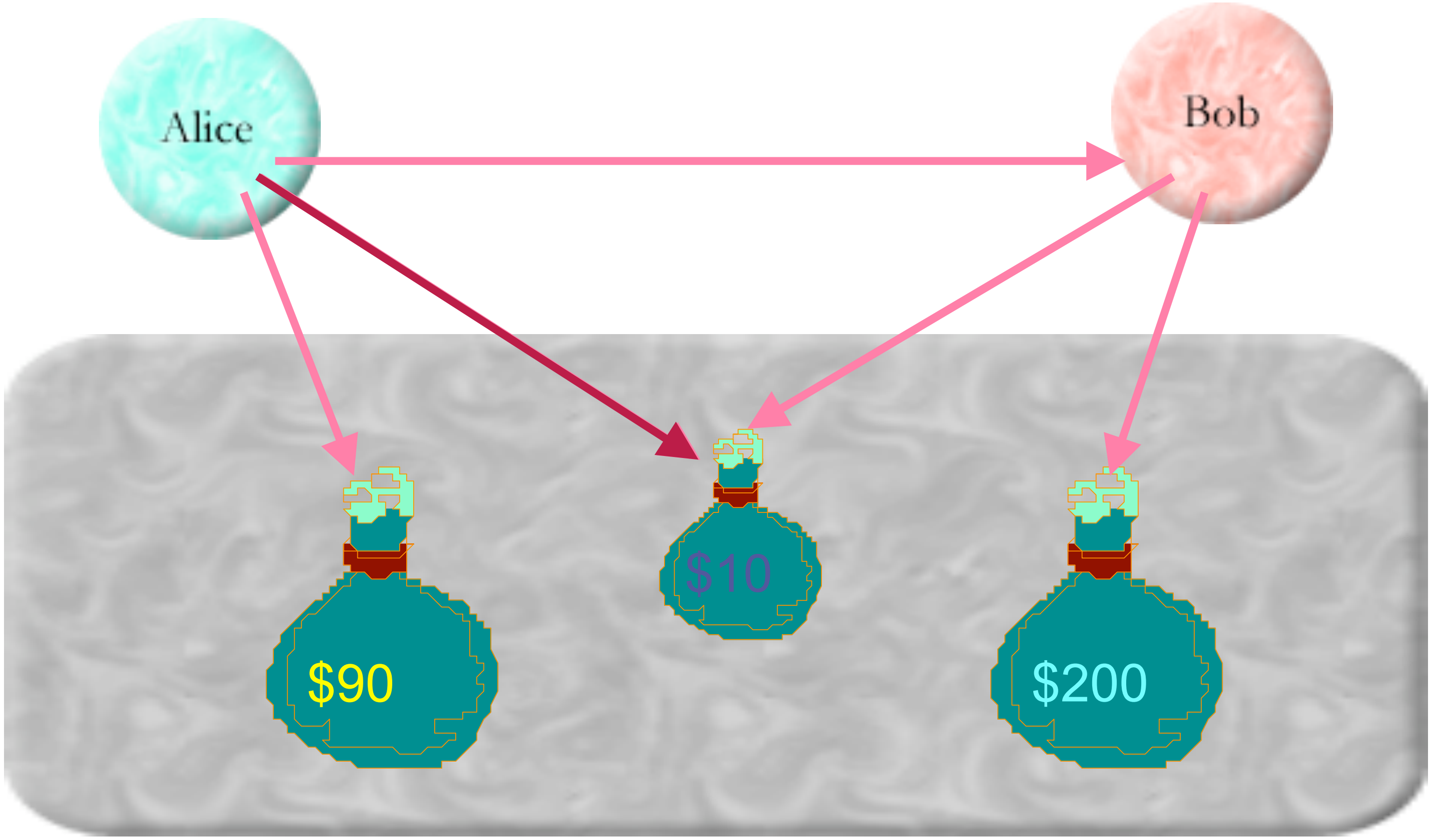
```
return Q.when(paymentP, function(p) {
```



Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```

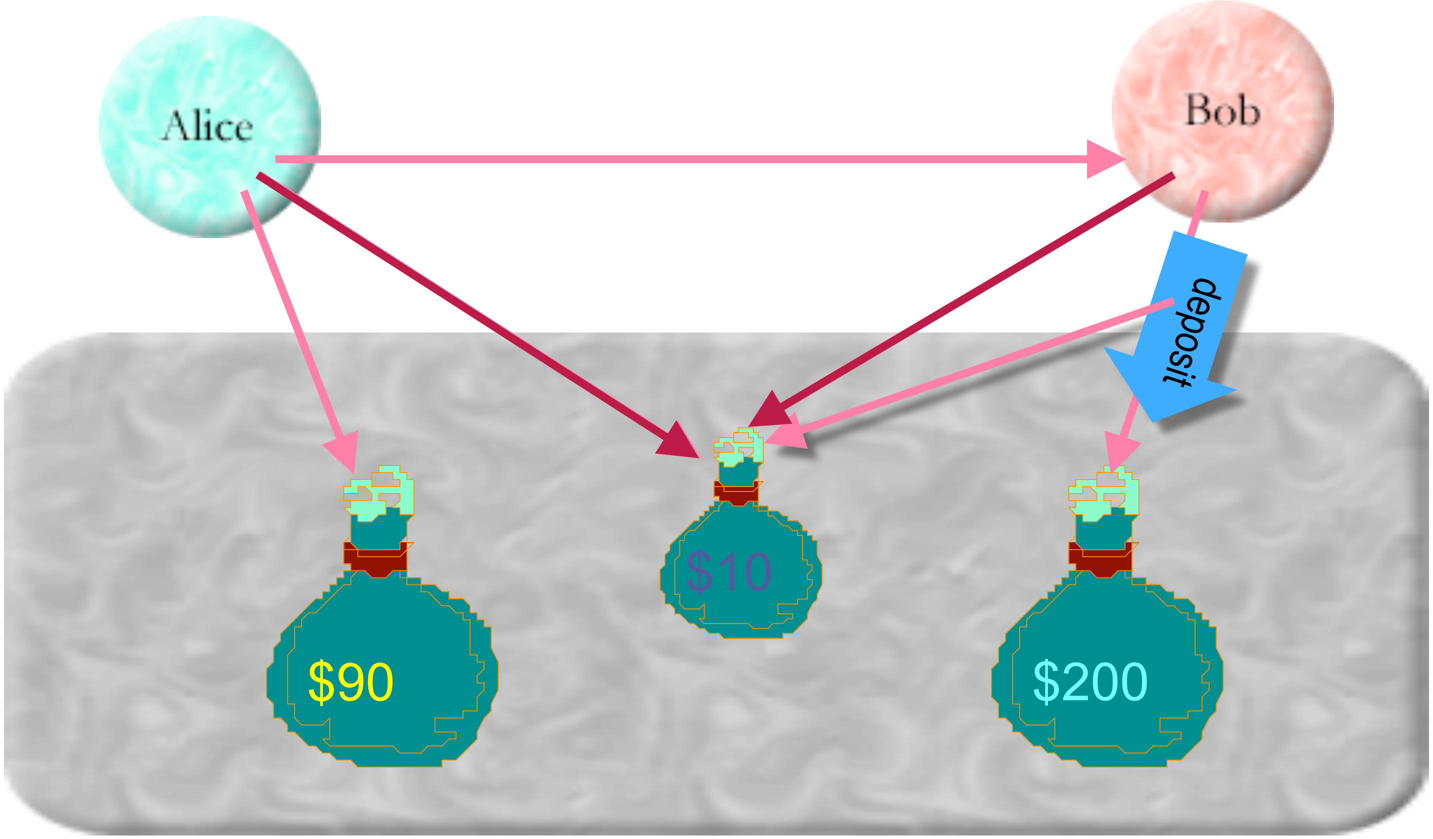
```
return Q.when(paymentP, function(p) {
    return Q.when(myPurse ! deposit(10, p), function(_) {
```



Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```

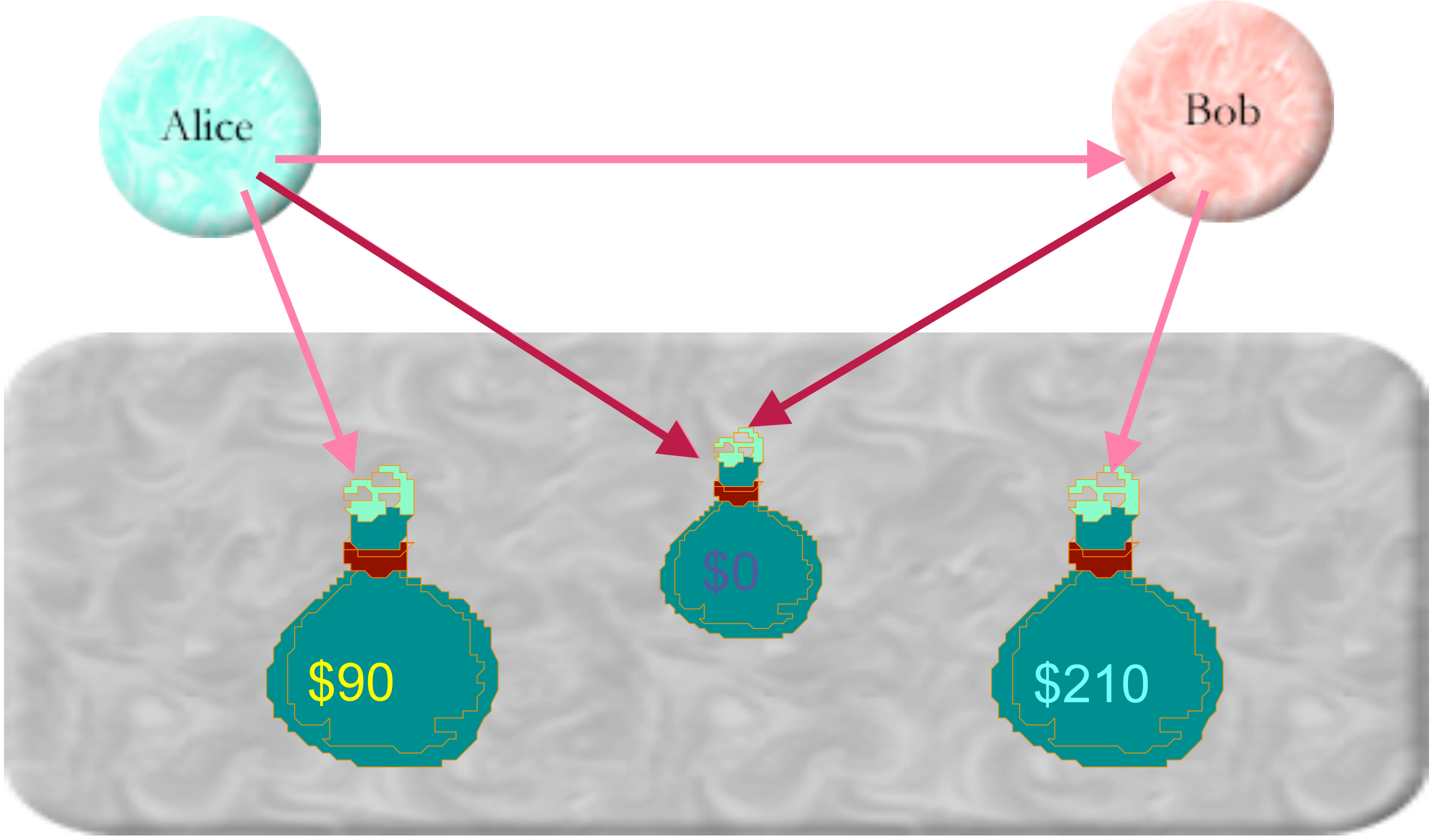
```
return Q.when(paymentP, function(p) {
    return Q.when(myPurse ! deposit(10, p), function(_) {
```



Distributed Secure Currency

```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```

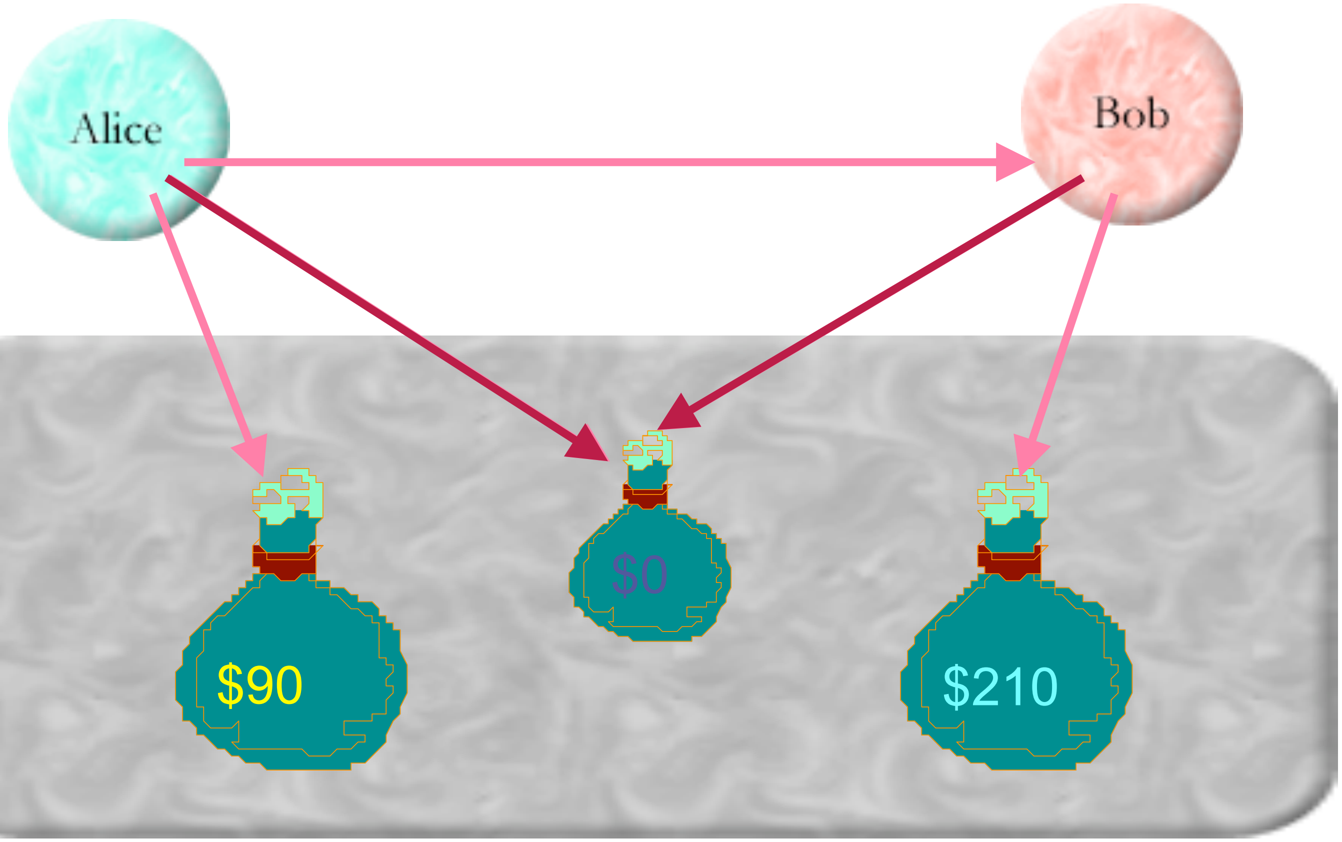
```
return Q.when(paymentP, function(p) {
   return Q.when(myPurse ! deposit(10, p), function(_) {
```



Distributed Secure Currency

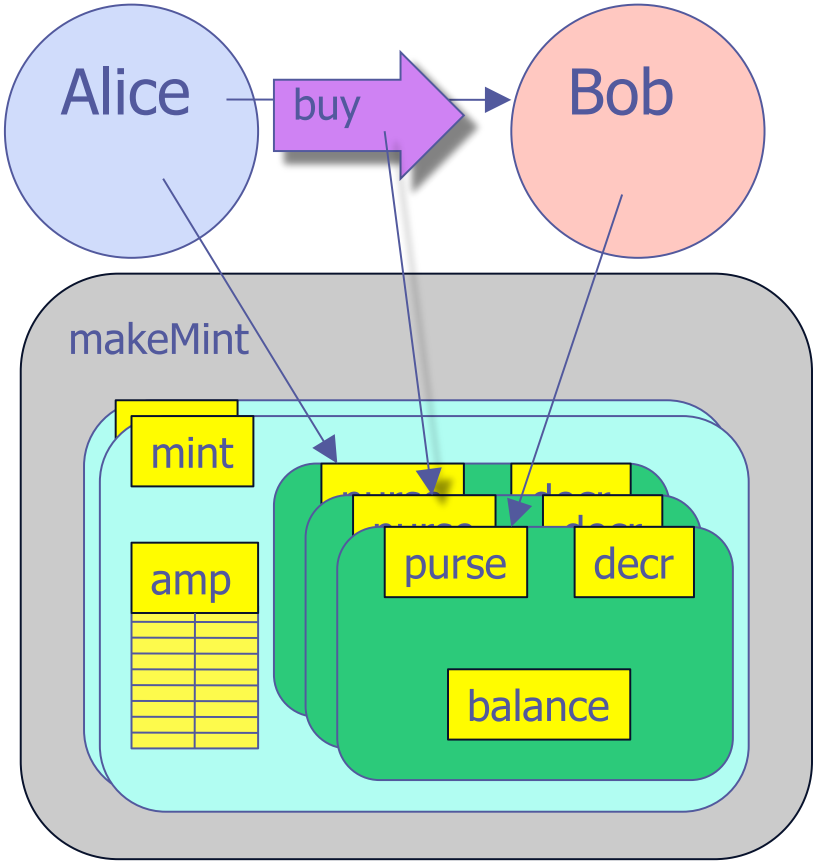
```
var paymentP = myPurse ! makePurse();
paymentP ! deposit(10, myPurse);
var goodP = bobP ! buy(desc, paymentP);
```

```
return Q.when(paymentP, function(p) {
   return Q.when(myPurse ! deposit(10, p), function(_) {
      return good; }, …
```



Money as "factorial" of secure coding

No explicit crypto



```
function makeMint() {
  var amp = WeakMap();
  return function mint(balance) {
    var purse = def({
      getBalance: function() { return balance; },
      makePurse: function() { return mint(0); },
      deposit: function(amount, src) {
        var newBal = Nat(balance + amount);
        amp.get(src)(Nat(amount));
        balance = newBal;
    } });
    function decr(amount) {
      balance = Nat(balance – amount);
    }
    amp.set(purse, decr);
    return purse;
  }
}
```

The other half of the object revolution

Protect object from world	Protect world from object
Responsibility driven design	Authority driven design
Avoid needless coupling	Avoid needless vulnerability
Information hiding	Principle of Least Authority
Avoid global variables	Forbid mutable static state
Procedural, data, control, …	…, and access abstractions
Patterns and frameworks	Patterns of safe cooperation
Say what you mean	Mean only what you say

Questions?

“def” is for defining defended objects

```
var defended = WeakMap();
function def(root) {
    var defending = WeakMap(), defendingList = [];
    function recur(val) {
        if (val !== Object(val) || defended.get(val) || defending.get(val)) { return; }
        defending.set(val, true); defendingList.push(val);
        Object.freeze(val);
        recur(Object.getPrototypeOf(val));
        Object.getOwnPropertyNames(val).forEach(function(p) {
            var desc = Object.getOwnPropertyDescriptor(val, p);
            recur(desc.value); recur(desc.get); recur(desc.set);
        });
    }
    recur(root);
    defendingList.forEach(function(obj) {
        defended.set(obj, true);
    });
    return root;
}
```

“**Nat**” validates its arg is a UInt32

```
function Nat(arg) {
    if (arg === arg >>> 0) { return arg; }
    throw new TypeError(‘Not a UInt32: ’ + arg);
}
```

“makeCaretaker” for defended targets

```
function makeCaretaker(target) {
    var wrapper = (typeof target !== 'function') ? {} : function(var_args) {
        return target.apply(this, arguments);
    };
    Object.getOwnPropertyNames(target).forEach(function(p) {
        var desc = Object.getOwnPropertyDescriptor(target, p);
        Object.defineProperty(wrapper, p, desc);
    });
    return def({
        wrapper: wrapper,
        revoke: function() { target = null; }
    });
}
```

"makeMembrane" for defended targets

```
function makeMembrane(target) {
    var enabled = true;
    function wrap(wrapped) {
        if (wrapped !== Object(wrapped)) { return wrapped; }
        var wrapper = (typeof wrapped !== 'function') ? {} : function(var_args) {
            return wrap(wrapped.apply(wrap(this), Array.slice(arguments, 0).map(wrap)));
        };
        Object.getOwnPropertyNames(wrapped).forEach(function(p) {
            var desc = Object.getOwnPropertyDescriptor(wrapped, p);
            Object.defineProperty(wrapper, p, desc);
        });
        return wrapper;
    }
    return def({
        wrapper: wrap(target),
        revoke: function() { enabled = false; }
    });
}
```