Making JavaScript Safe and Secure

AGORIC

moddable

METAMASK

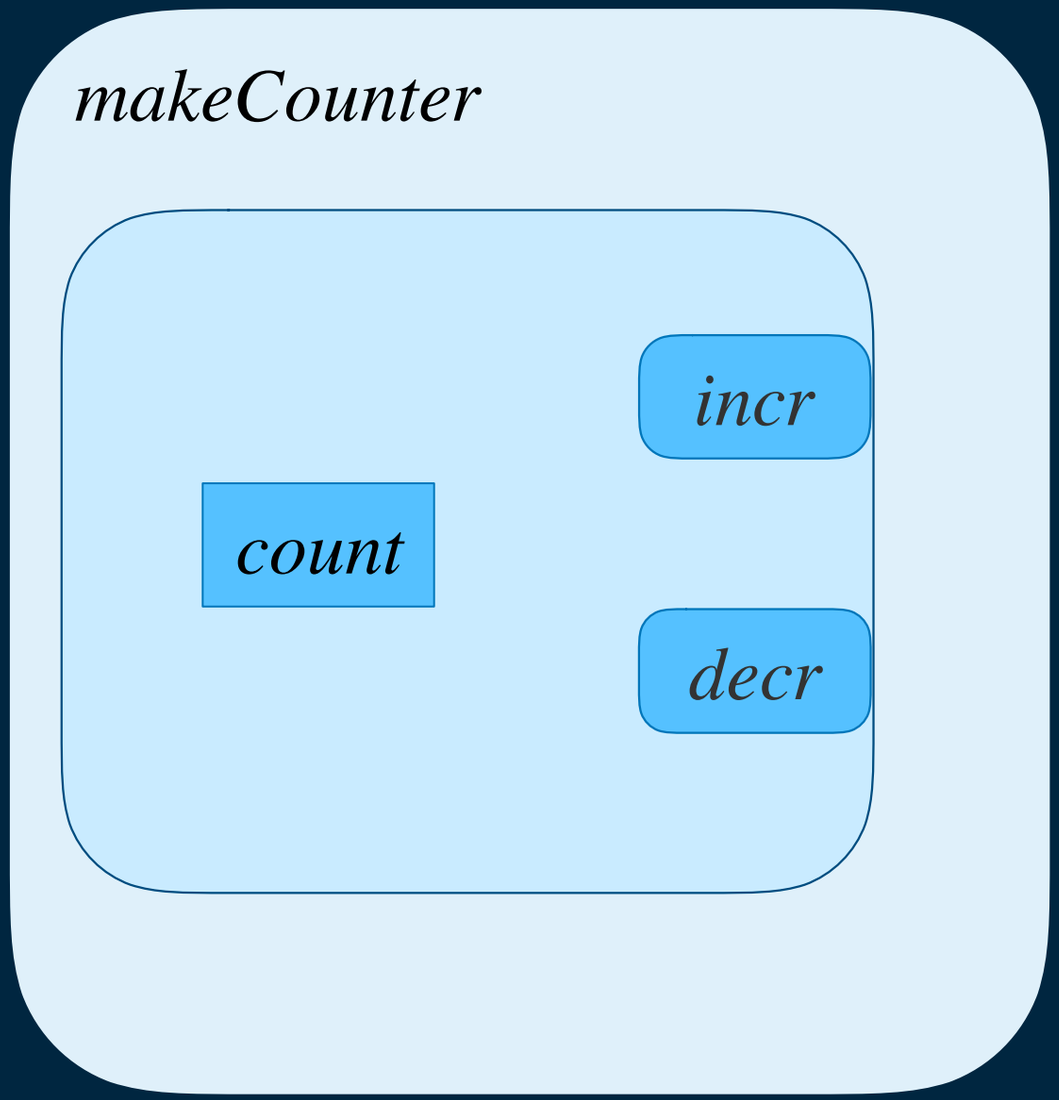
5th Dec | 6pm | 540 Howard Street, SF

Counter Example

makeCounter

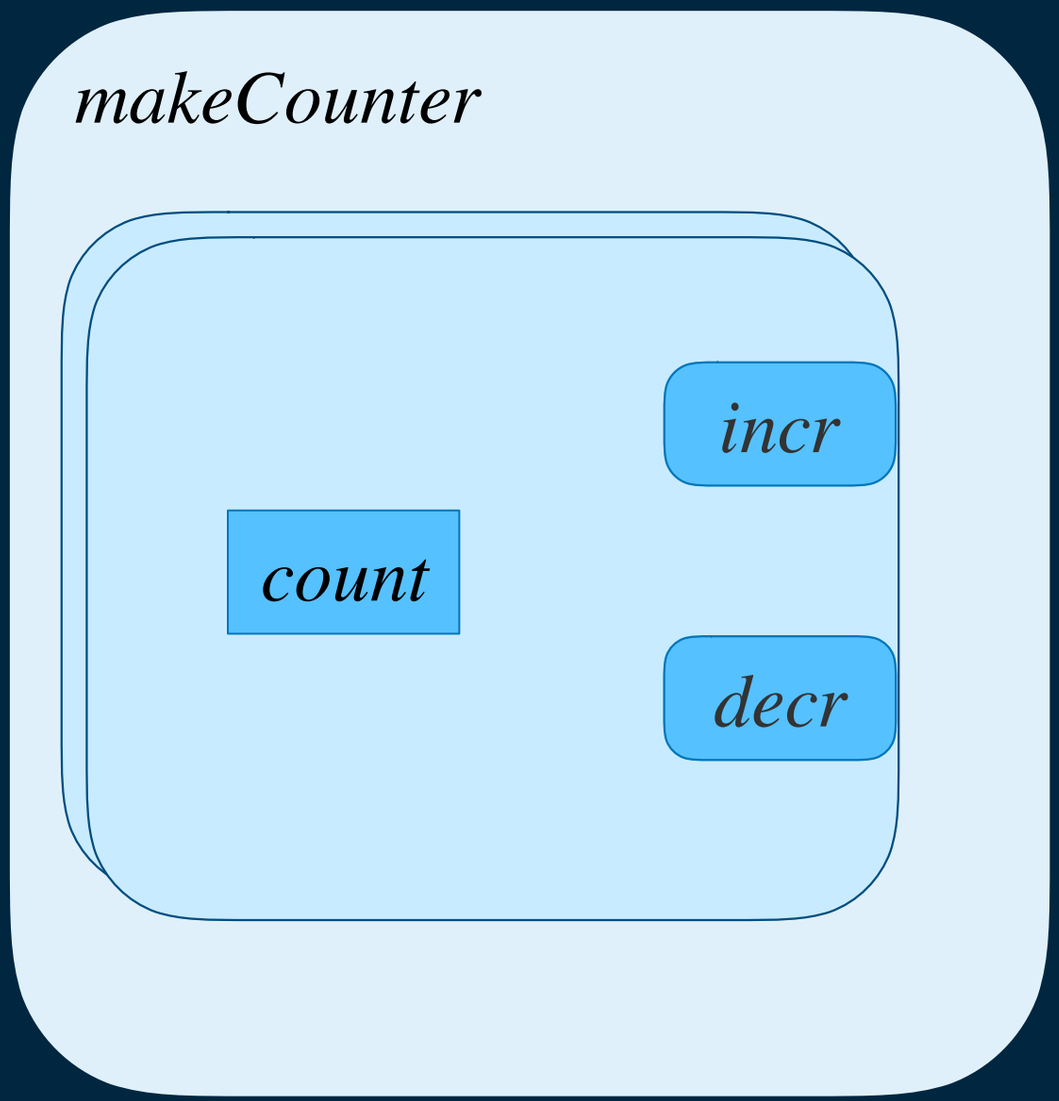
```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example



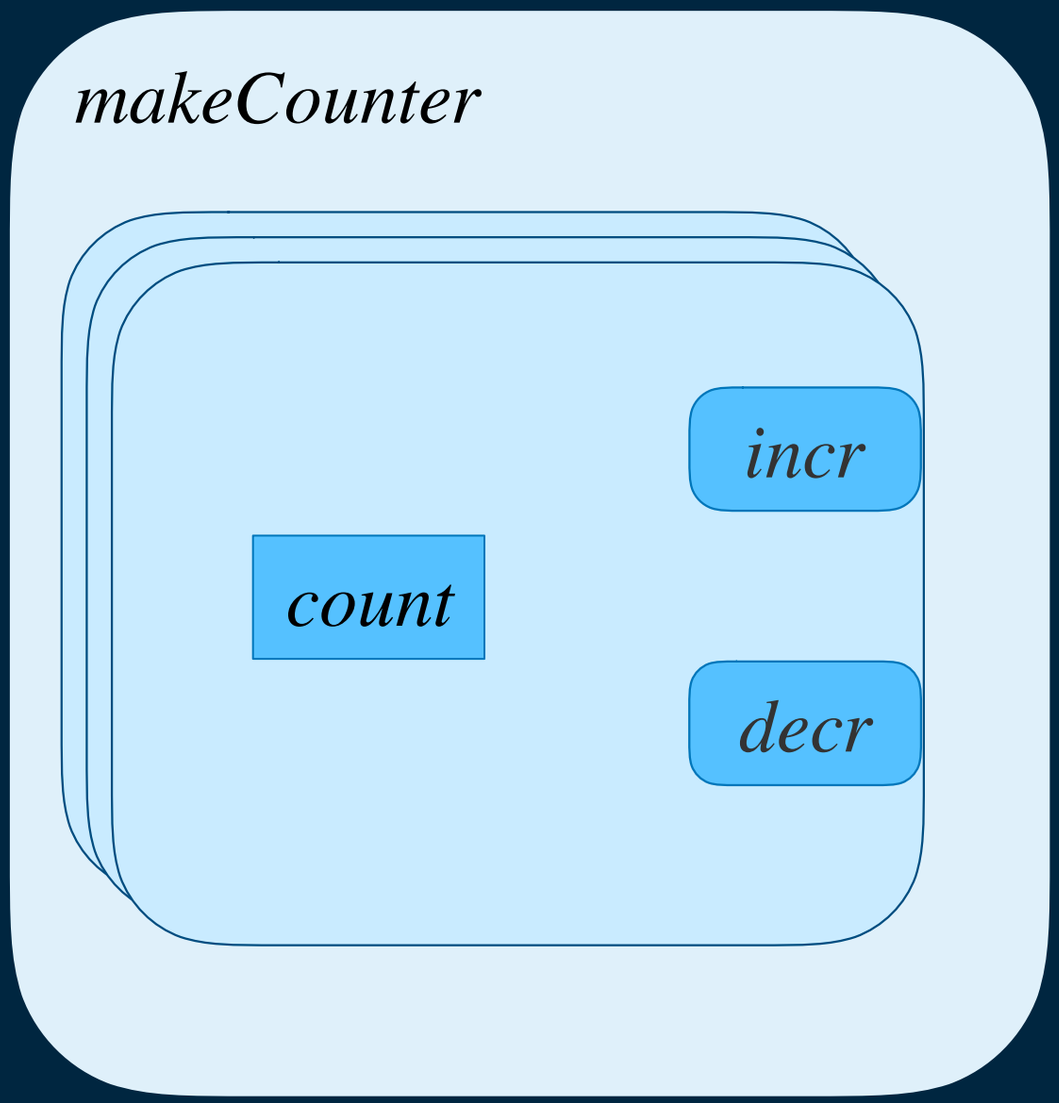

```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example



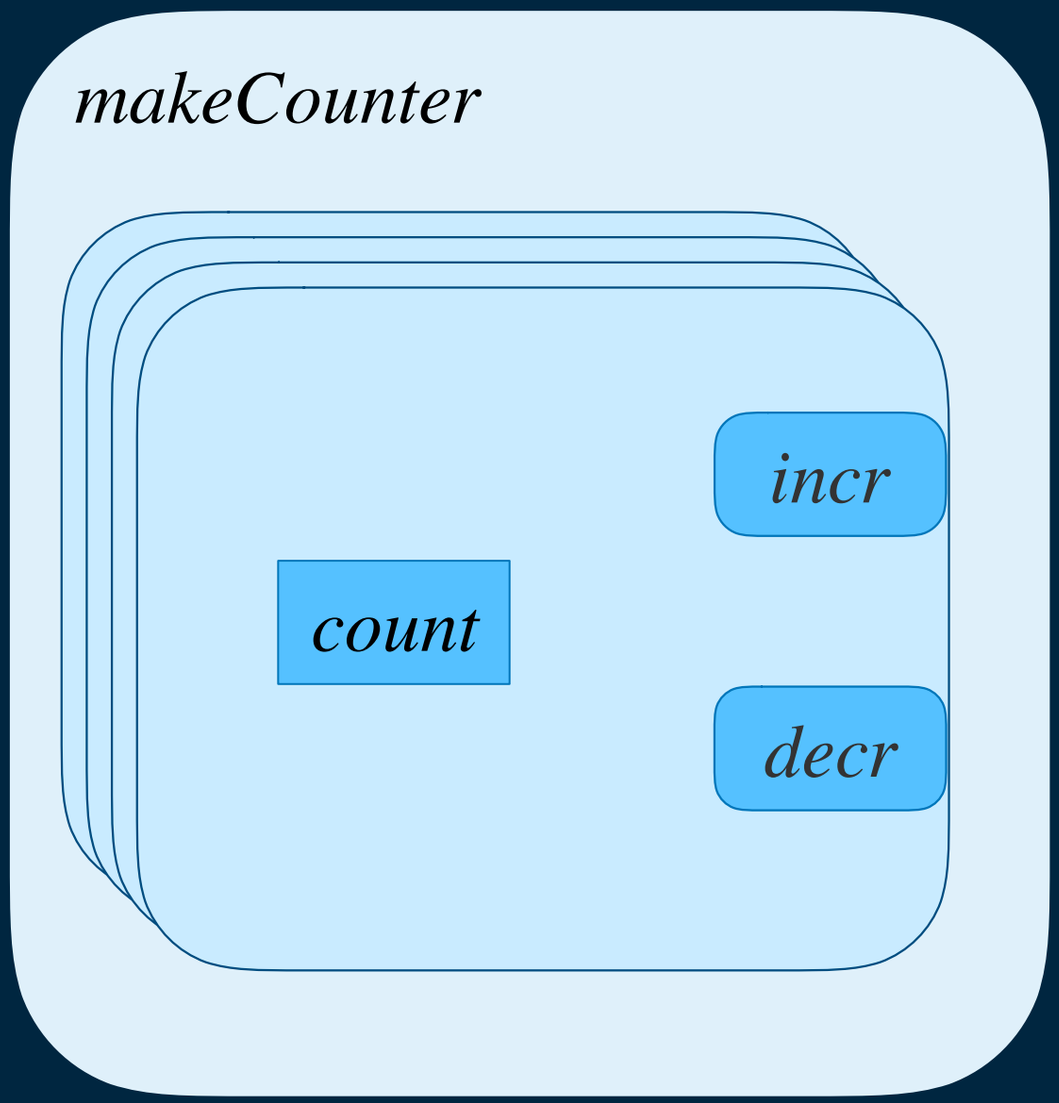

```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example



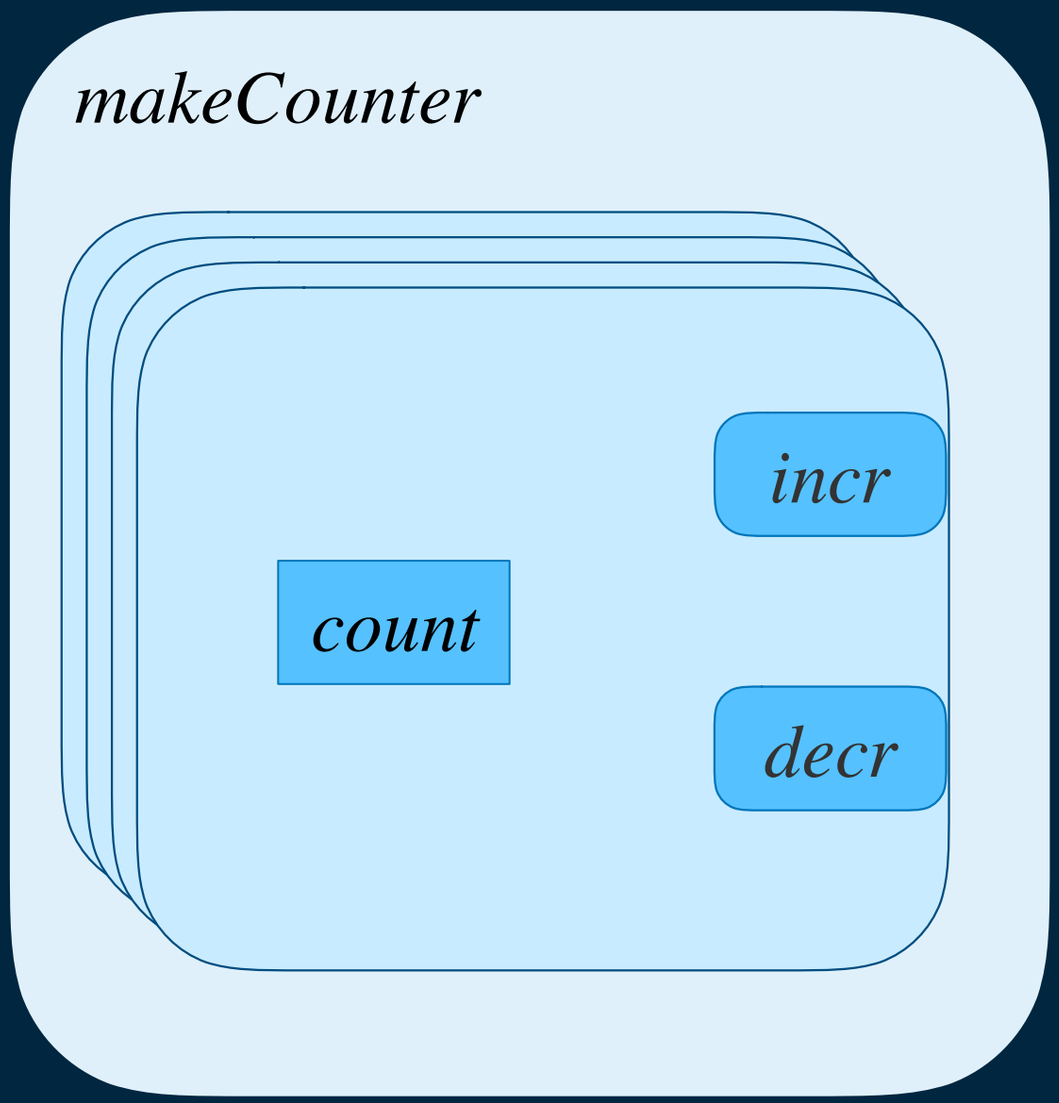

```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example



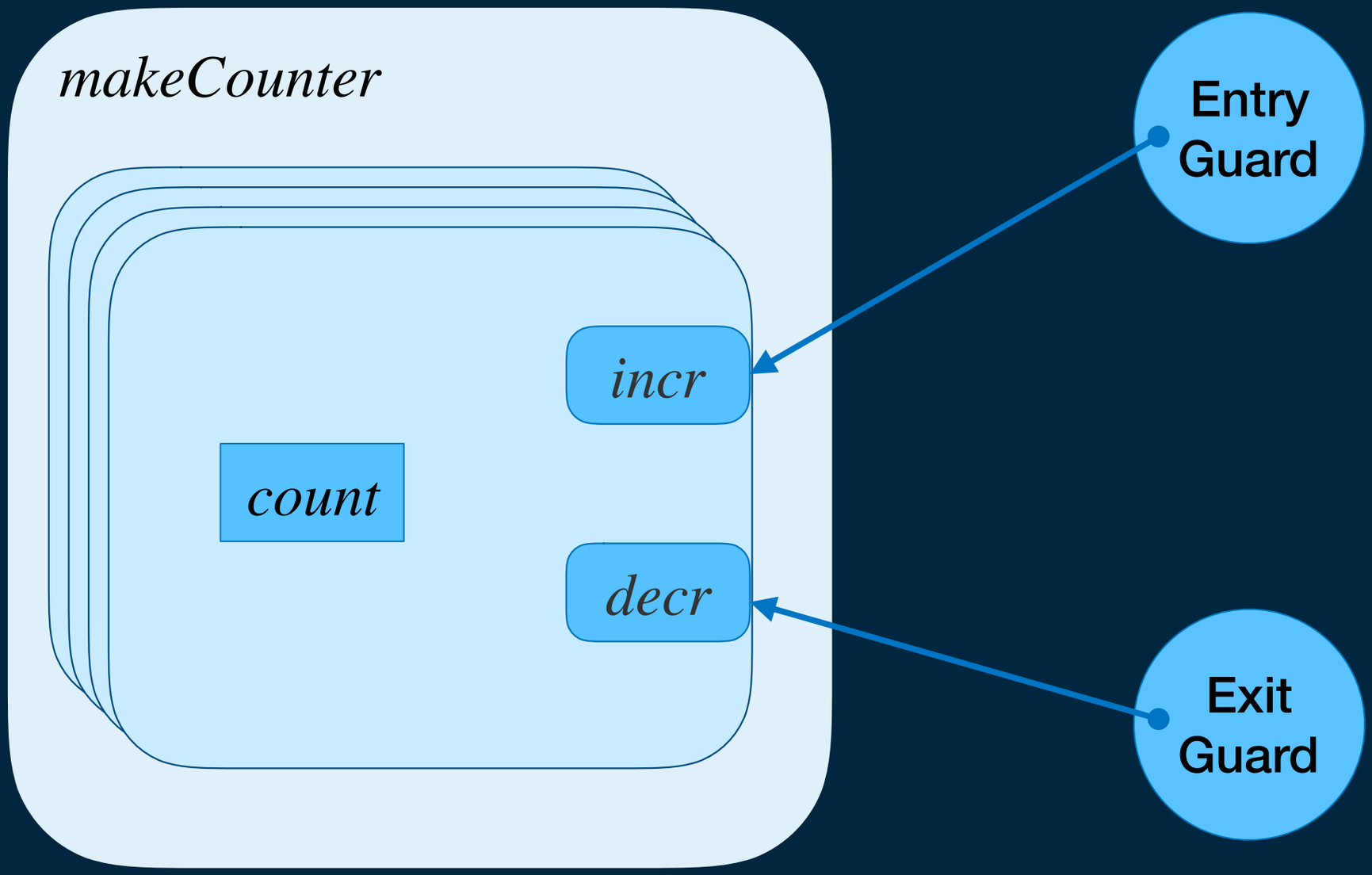

```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example

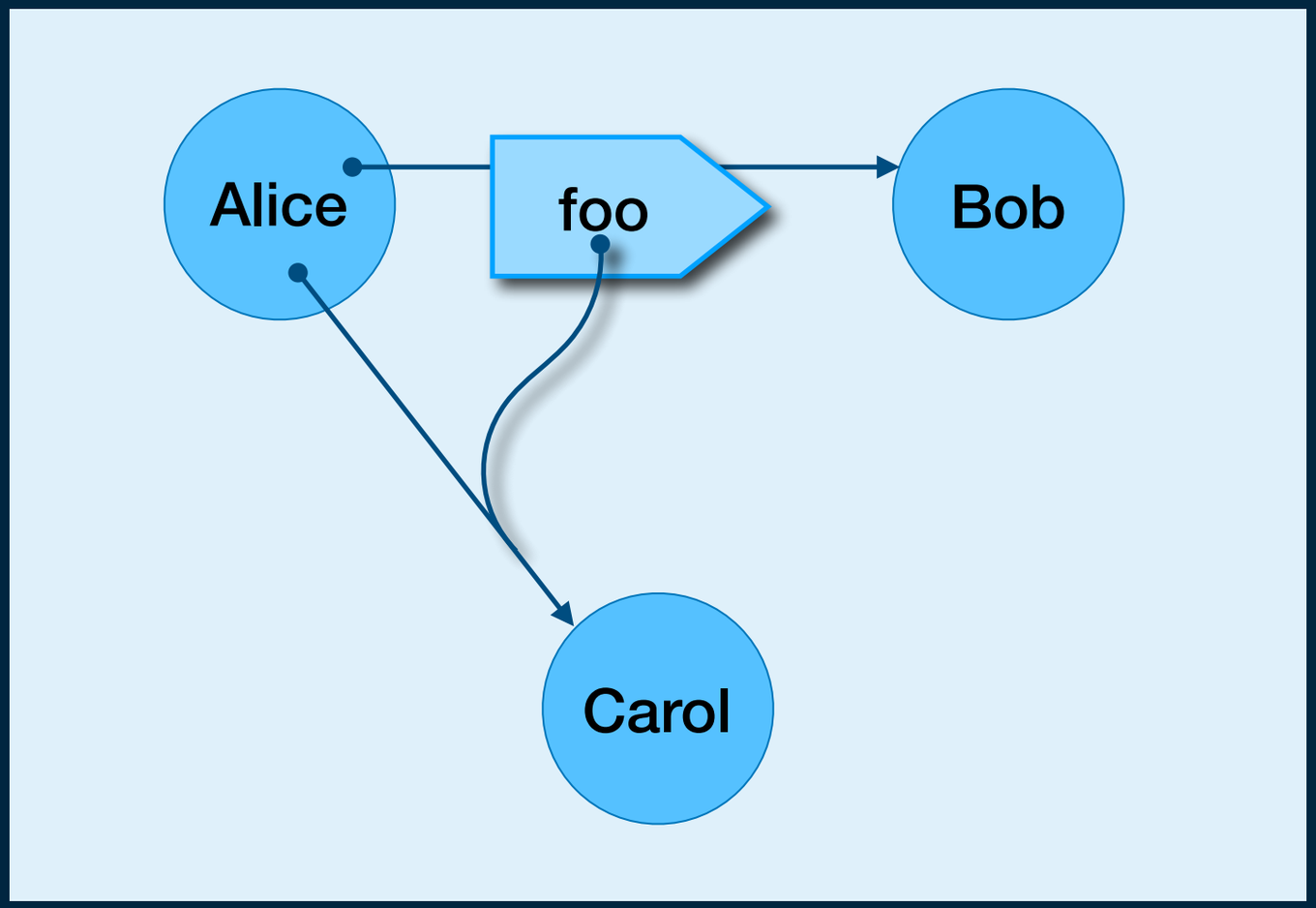



```
function makeCounter() {
  let count = 0n;
  return harden({
    incr() { return ++count; },
    decr() { return --count; }
  });
}

const counter = makeCounter();

entryGuard~.use(counter.incr);
exitGuard~.use(counter.decr);
```

Counter Example



Alice says: bob.foo(carol)



Object-capabilities (ocaps)

Only refs carry causality

Reference graph === Access graph

Principle of Least Authority (POLA)

LavaMoat

XS

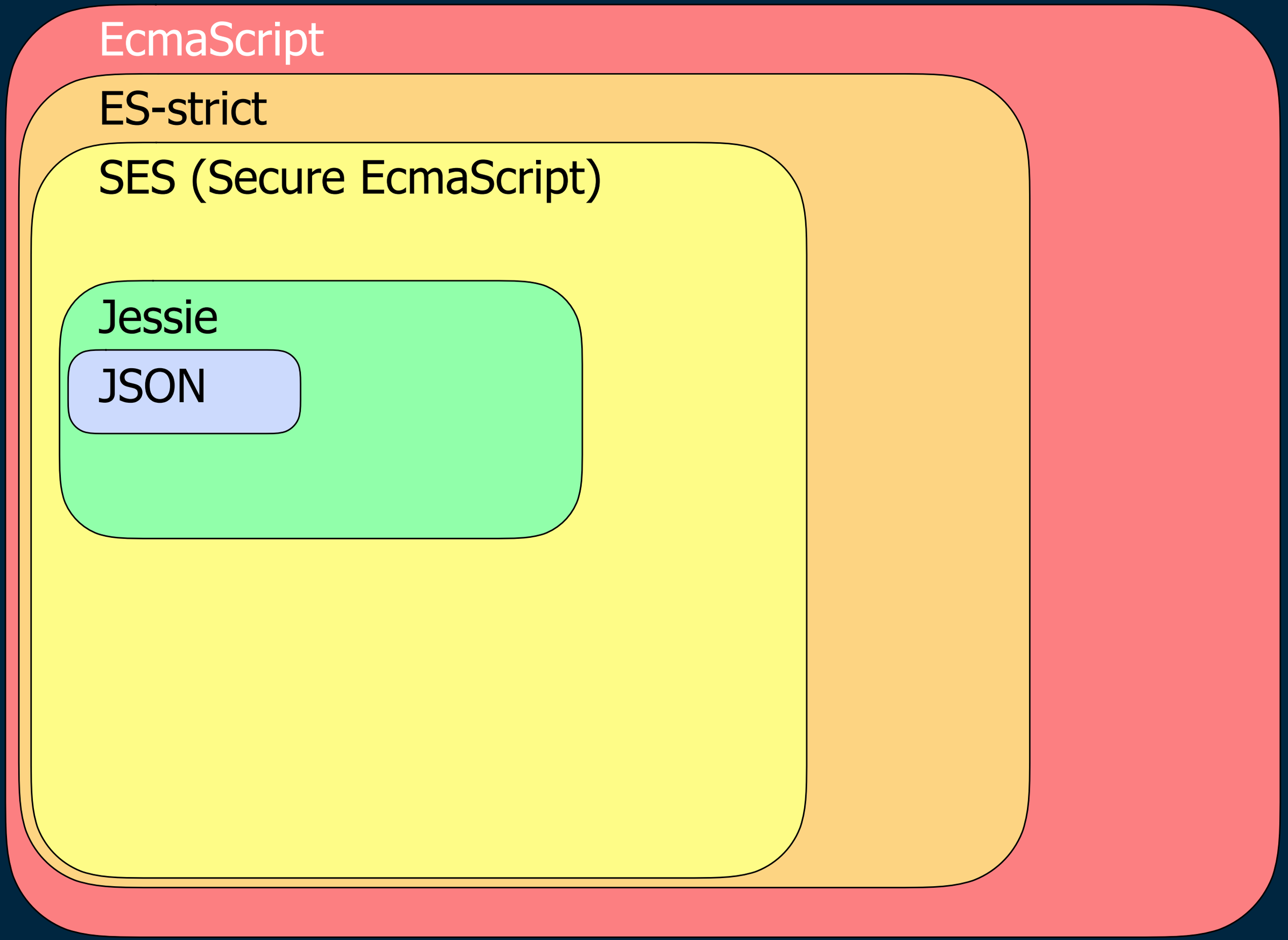
Distributed SES

Lightning Locker

Core

ECMAScript Modules for Embedded Systems

`"use strict";`	
`Object.freeze`	
`Promise`	
Tagged template literals	
`Proxy, WeakMap`	
`Realm`	
`SES, Compartment`	
`WeakRef`	
`HandledPromise, ~.`	


EcmaScript
ES-strict
SES (Secure EcmaScript)
Jessie
JSON

EcmaScript

ES-strict

SES (Secure EcmaScript)

Jessie

JSON

non-static scoping

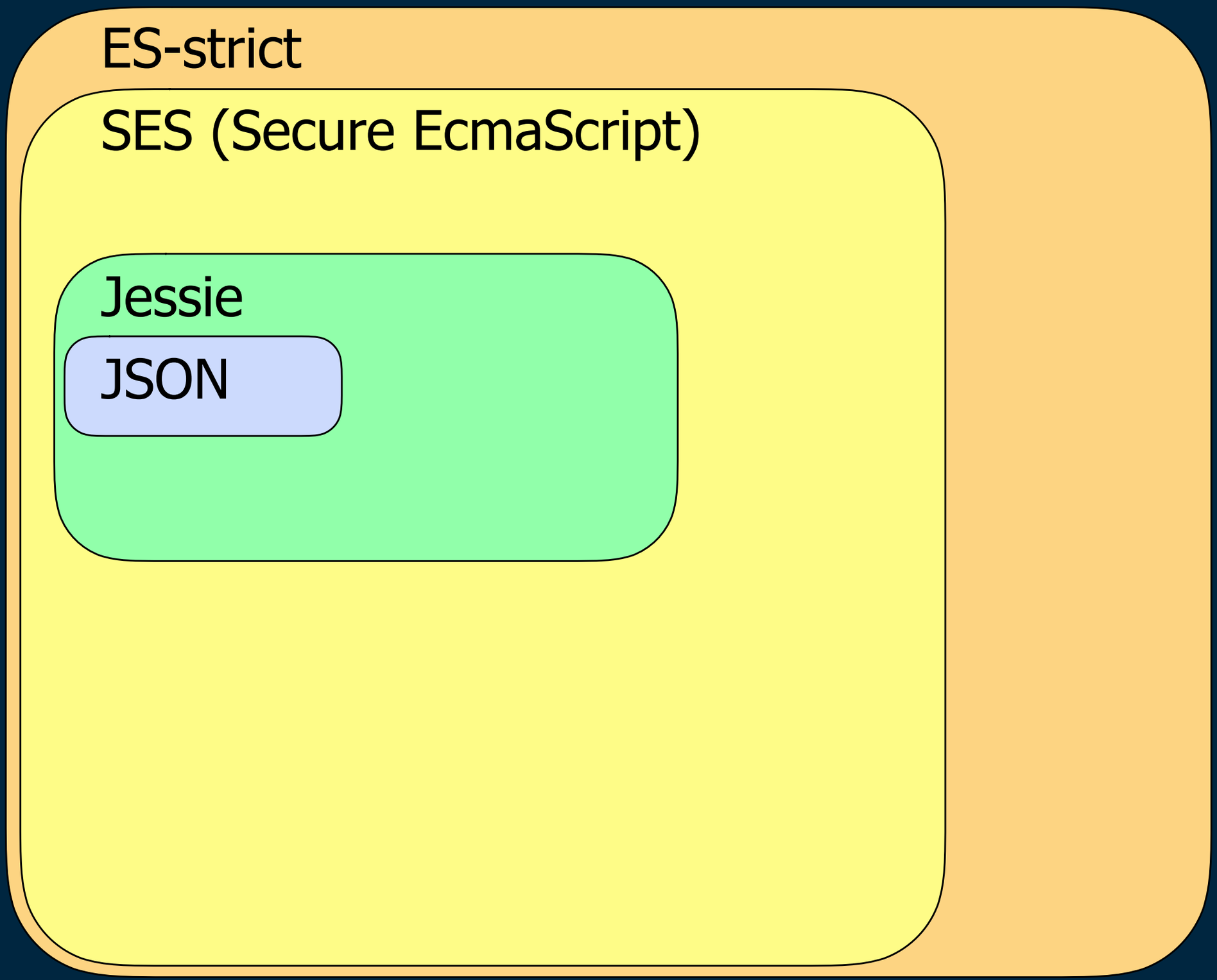
implicit primitive wrapping

`.caller`
`.callee`
`.arguments`

silent failed assignment

implicit access to global

`with`


ES-strict
SES (Secure EcmaScript)
Jessie
JSON

ES-strict

SES (Secure EcmaScript)

Jessie

JSON

mutable primordials

ambient authority

per-realm global object

SES (Secure EcmaScript)

Jessie

JSON

SES (Secure EcmaScript)

Jessie

JSON

hardened objects

pure modules

objects as closures

mutable properties

inheritance

pure primordials

`Reflect`
`RegExp`
`Date`
`Symbol`
`Function`
`Proxy`
`Realm`

`==` `!=`

`/.*/`

`this`
`new`
`instanceof`
`class`
`extends`
`super`

`@def` *defensible classes*

`var`
`for/in`
`in`
`delete`

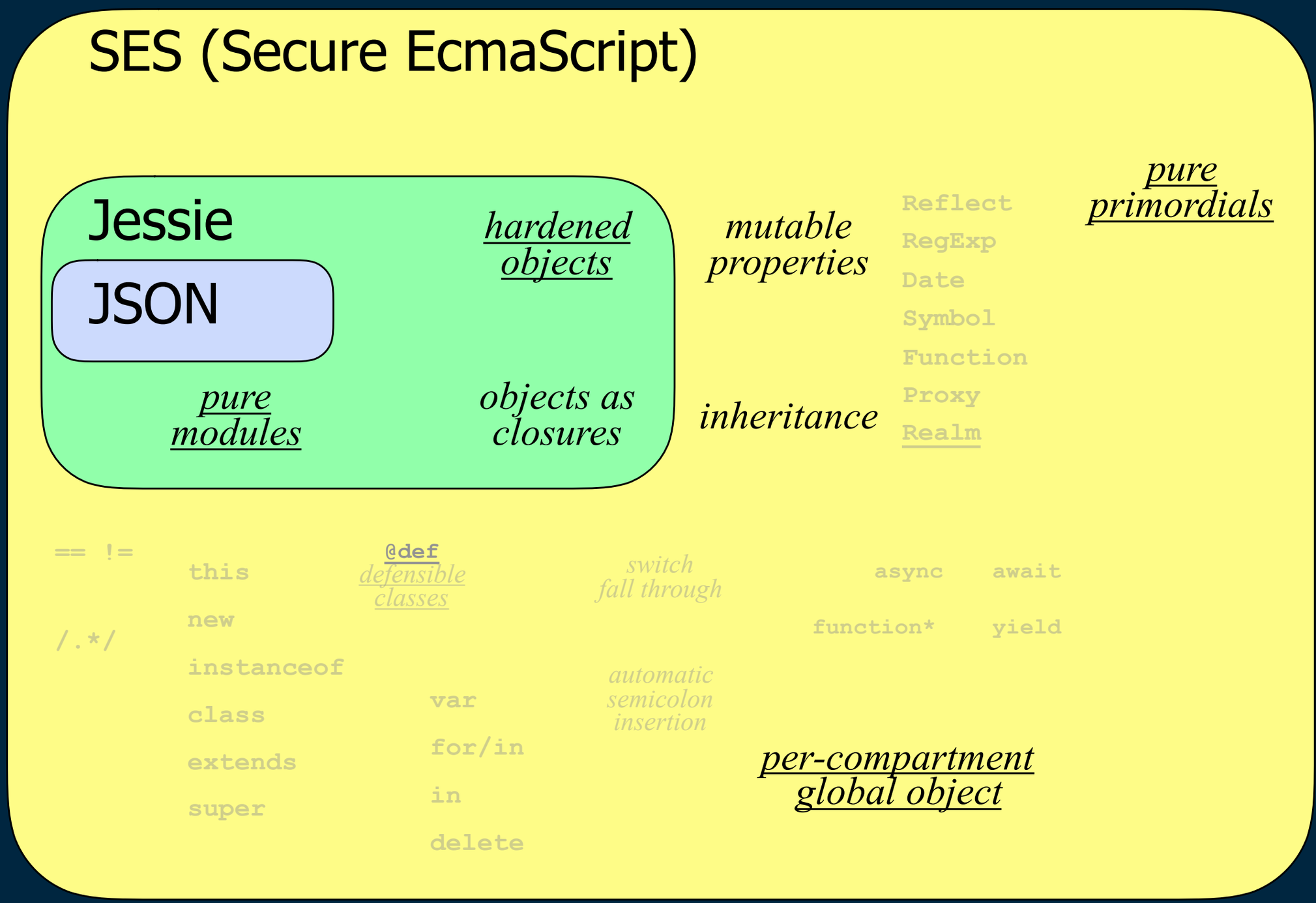
switch fall through

automatic semicolon insertion

`async` `await`
`function*` `yield`

per-compartment global object

Don’t add security. Remove insecurity.



```
const x = 2;
const y = 3;
```

```
const x = 2;
const y = 3;

x + y;  // 5
```

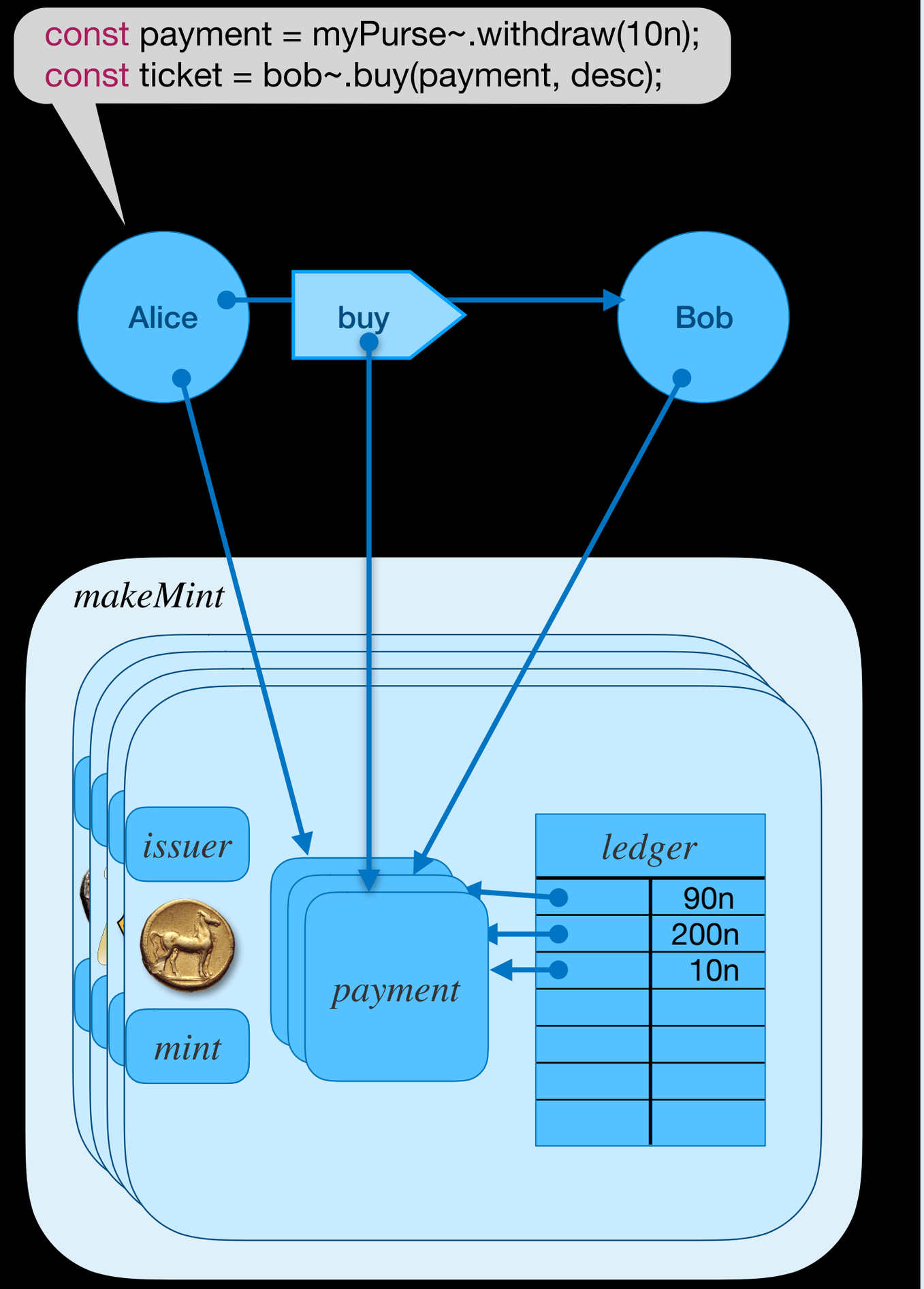
```
const x = 2;
const y = 3;

x + y;  // 5

const e = new Evaluator();
```

```
const x = 2;
const y = 3;

x + y;  // 5

const e = new Evaluator();

e.evaluateScript('x + y');  // ReferenceError: x is not defined
```

```
const x = 2;
const y = 3;

x + y;  // 5

const e = new Evaluator();

e.evaluateScript('x + y');  // ReferenceError: x is not defined
e.evaluateScript('x + y', {x: 7, y: 9});  // 16
```

```
const x = 2;
const y = 3;

x + y;  // 5

const e = new Evaluator();

e.evaluateScript('x + y');  // ReferenceError: x is not defined
e.evaluateScript('x + y', {x: 7, y: 9});  // 16

function evilFunction(x) { throw 'gotcha'; }
```

```
const x = 2;
const y = 3;

x + y;  // 5

const e = new Evaluator();

e.evaluateScript('x + y');  // ReferenceError: x is not defined
e.evaluateScript('x + y', {x: 7, y: 9});  // 16

function evilFunction(x) { throw 'gotcha'; }

Array.prototype.push = evilFunction;  // TypeError: Cannot assign…
```




```
function makeMint() {
  const ledger = new WeakMap();

  const issuer = harden({
    makeEmptyPurse() { return mint.makePurse(0n); }
  });

  const mint = harden({
    makePurse(initialBalance) {
      const purse = harden({
        getIssuer() { return issuer; },
        getBalance() { return ledger.get(purse); },

        deposit(amount, src) {
          Nat(ledger.get(purse) + Nat(amount));
          ledger.set(src, Nat(ledger.get(src) - amount));
          ledger.set(purse, ledger.get(purse) + amount);
        },
        withdraw(amount) {
          const newPurse = issuer.makeEmptyPurse();
          newPurse.deposit(amount, purse);
          return newPurse;
        }
      });
      ledger.set(purse, initialBalance);
      return purse;
    }
  });
  return mint;
};
```

Questions?