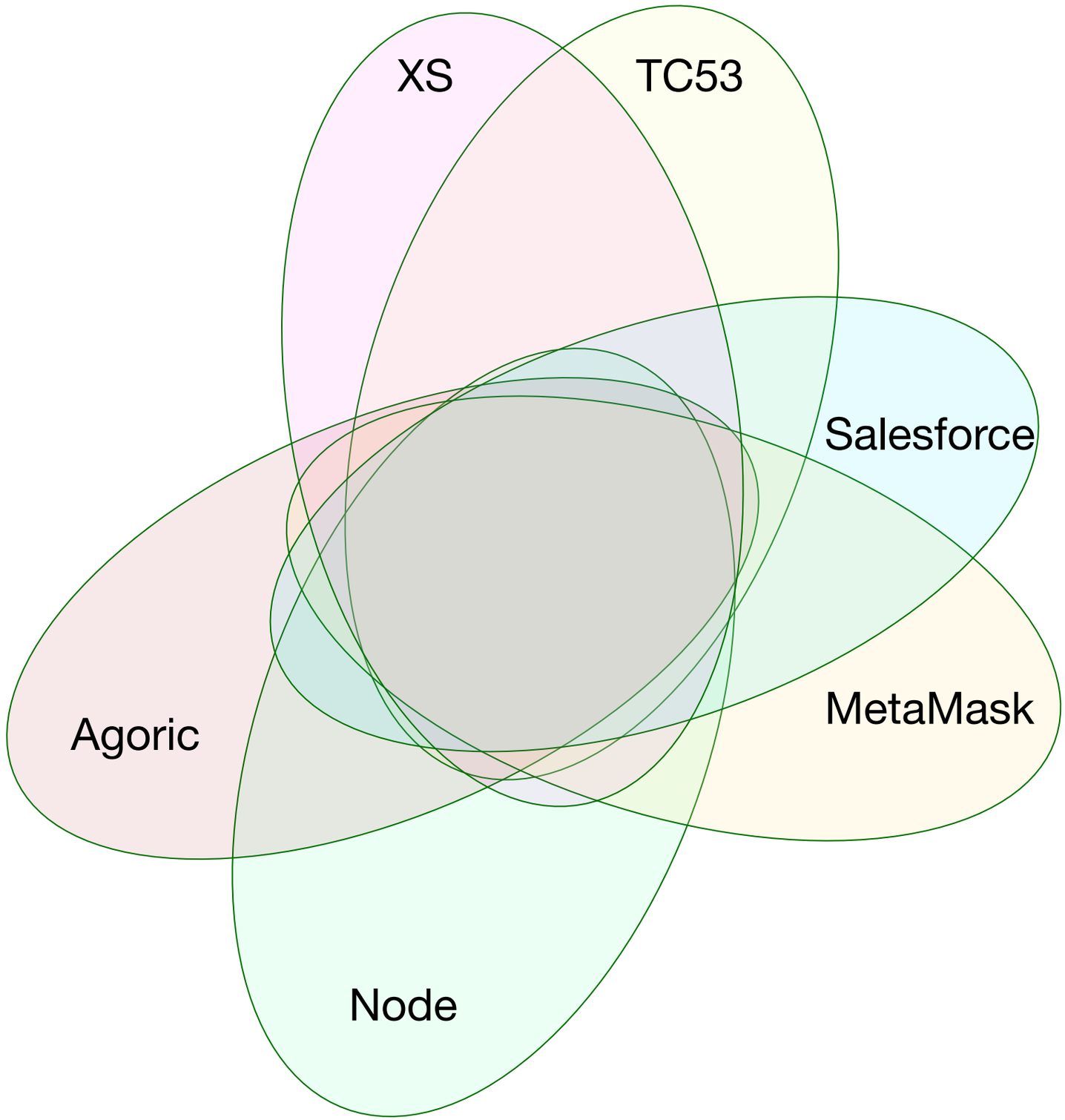
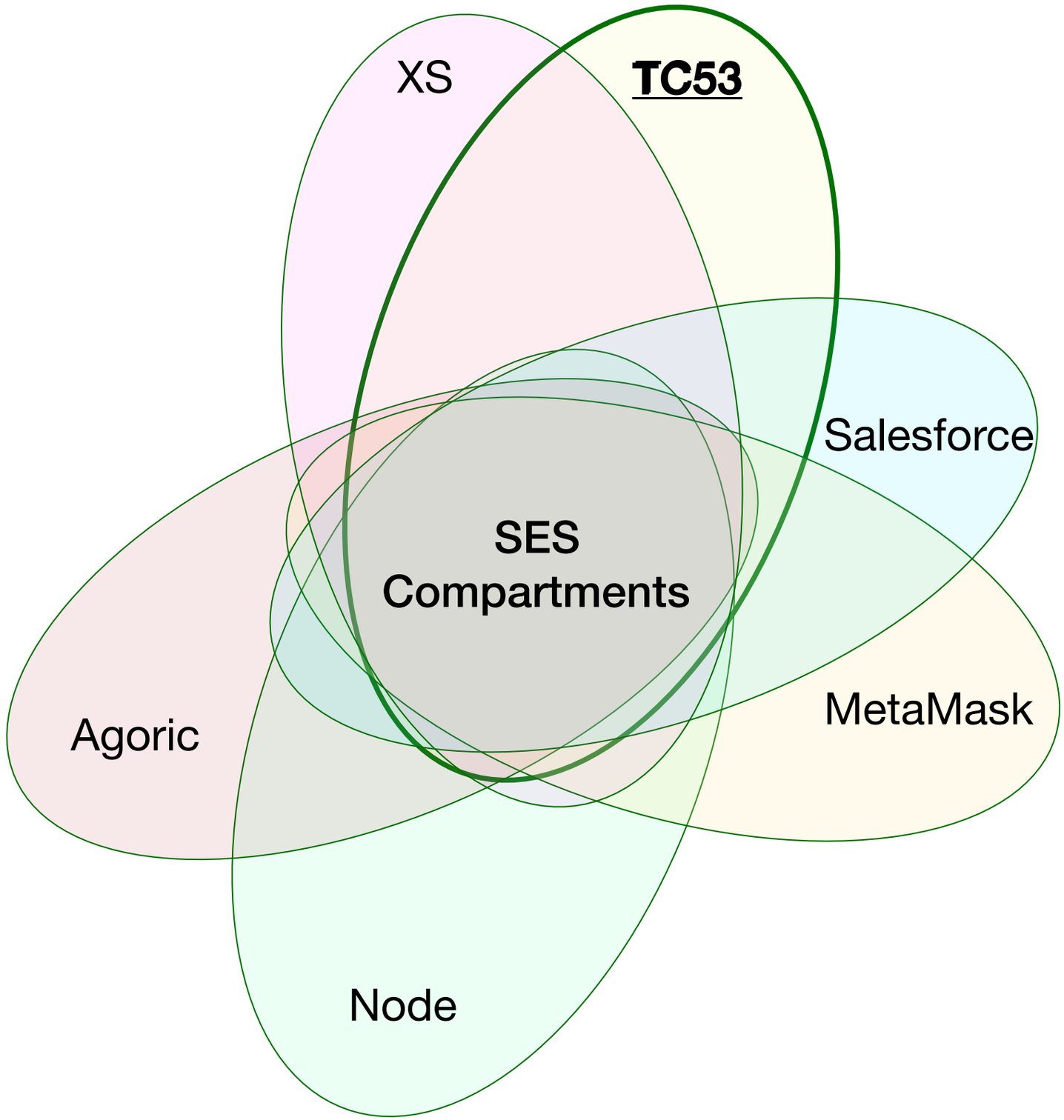
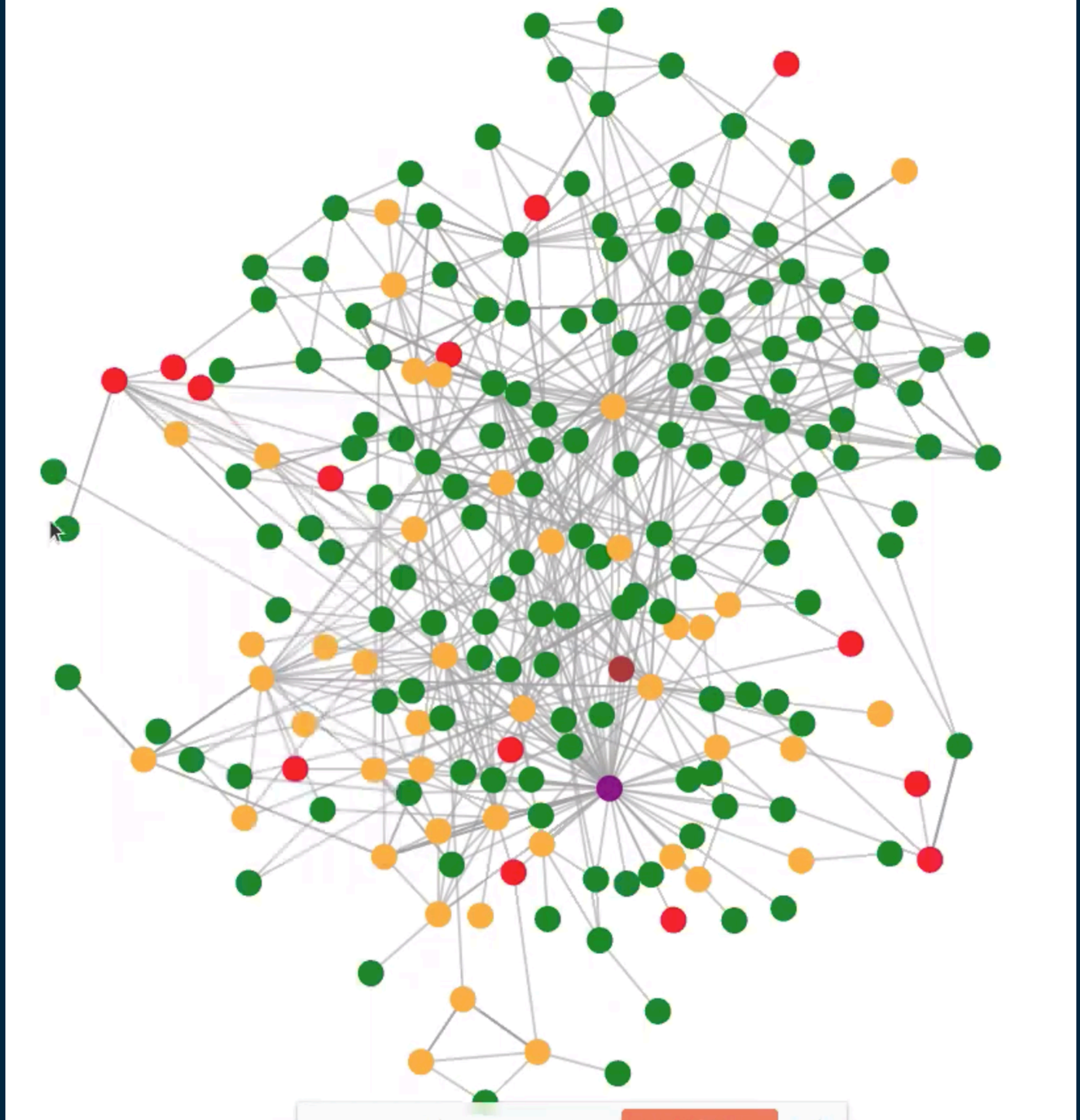
SES Compartments

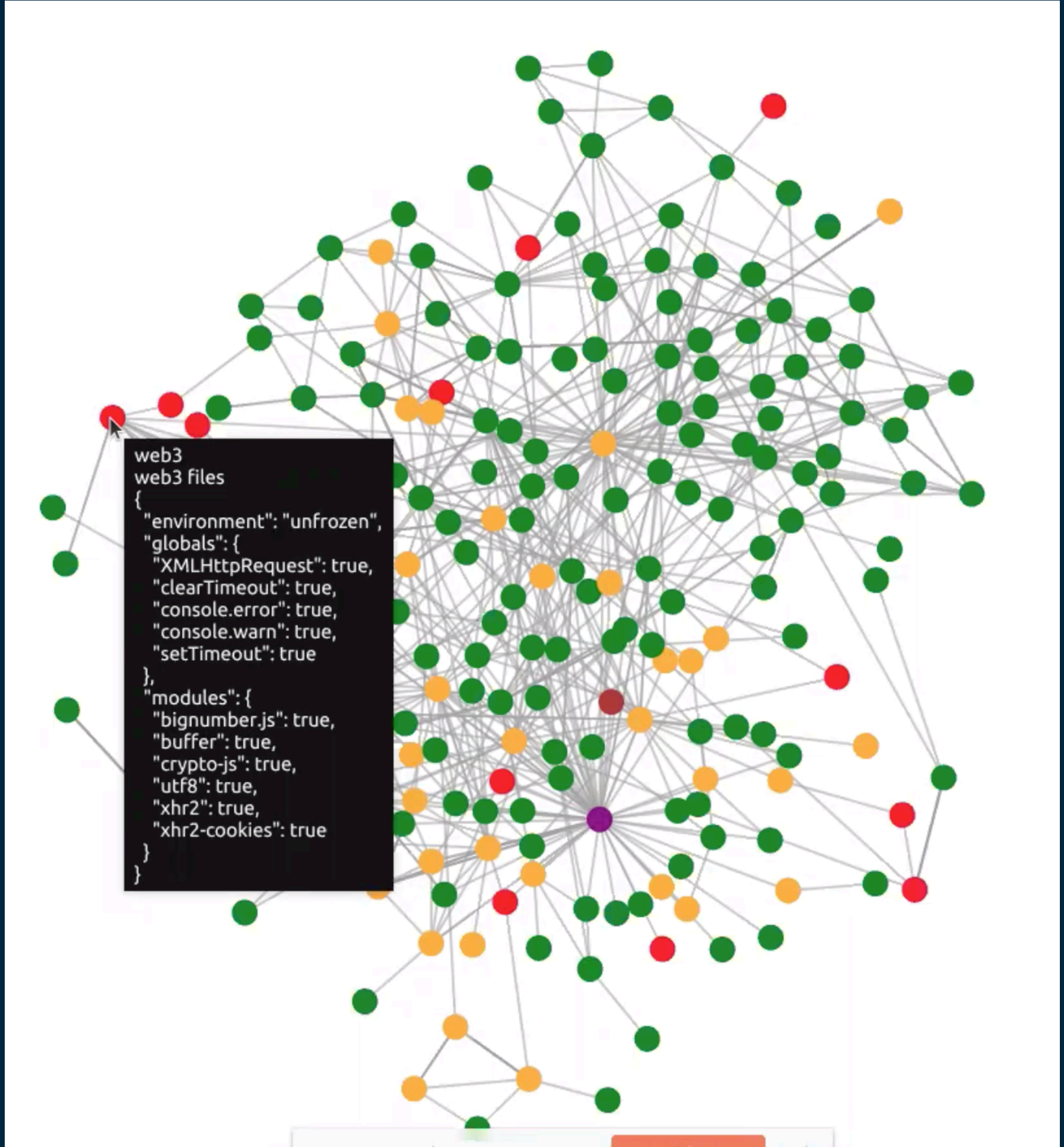
Mark S. Miller, Agoric
JF Paradis, Agoric
Caridy Patiño, Salesforce
Patrick Soquet, Moddable
Bradley Farias, GoDaddy, Node

tc53 February 2020, Palo Alto


XS
TC53
Salesforce
Agoric
MetaMask
Node


XS
TC53
Salesforce
SES
Compartments
Agoric
MetaMask
Node




web3
web3 files
{
"environment": "unfrozen",
"globals": {
"XMLHttpRequest": true,
"clearTimeout": true,
"console.error": true,
"console.warn": true,
"setTimeout": true
},
"modules": {
"bignumber.js": true,
"buffer": true,
"crypto-js": true,
"utf8": true,
"xhr2": true,
"xhr2-cookies": true
}
}

EcmaScript

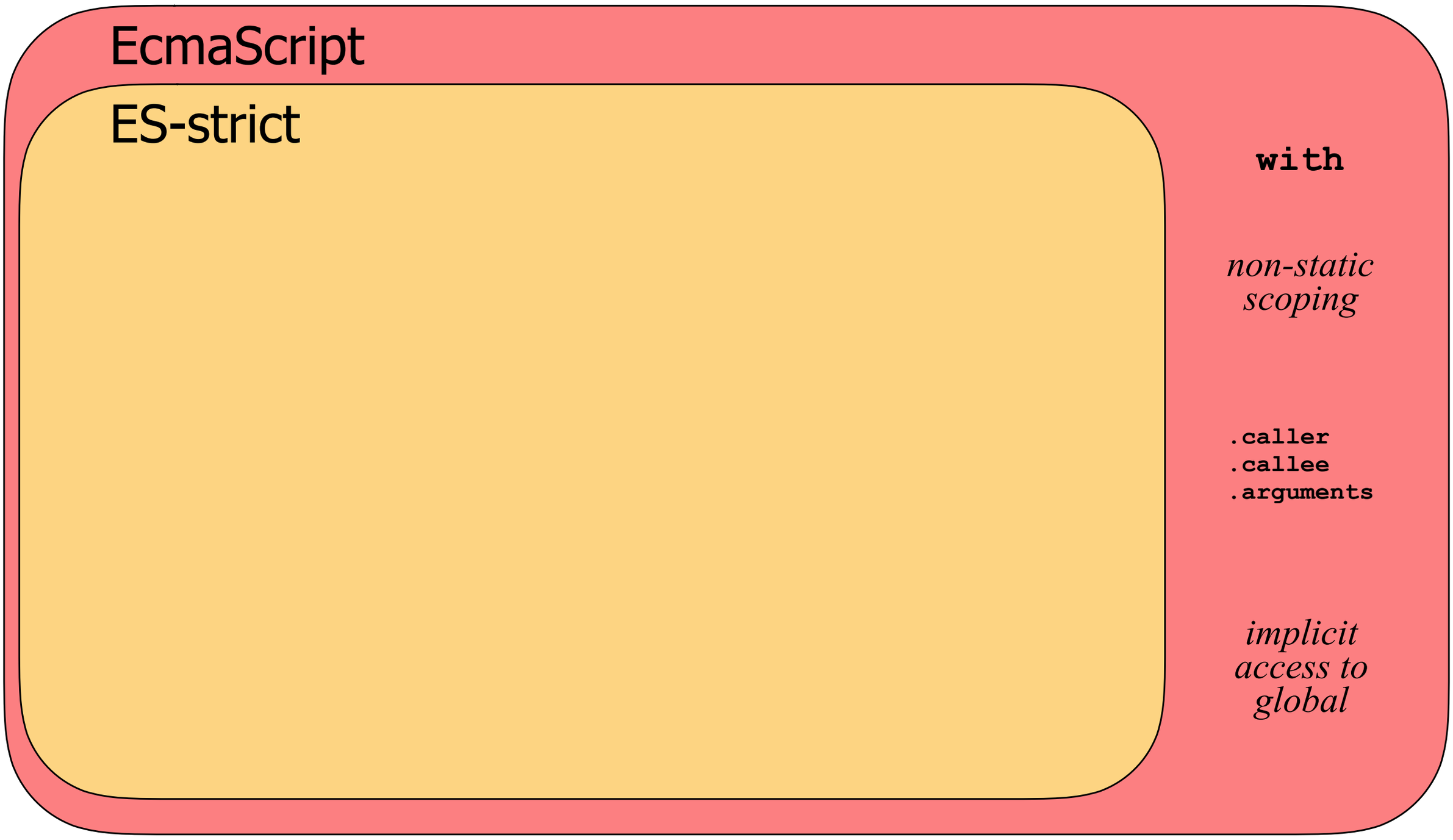
`with`

non-static scoping

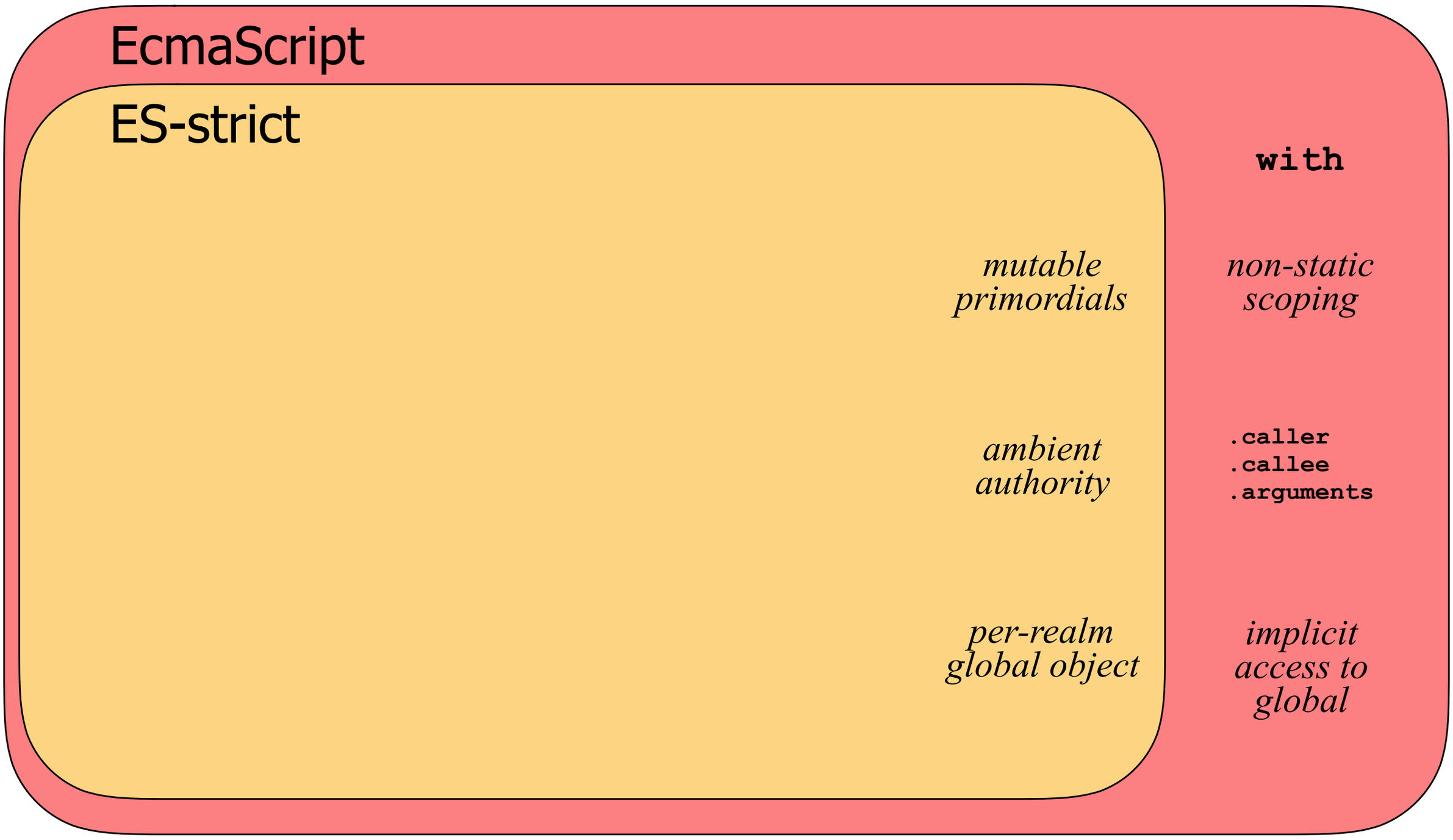
`.caller`
`.callee`
`.arguments`

implicit access to global

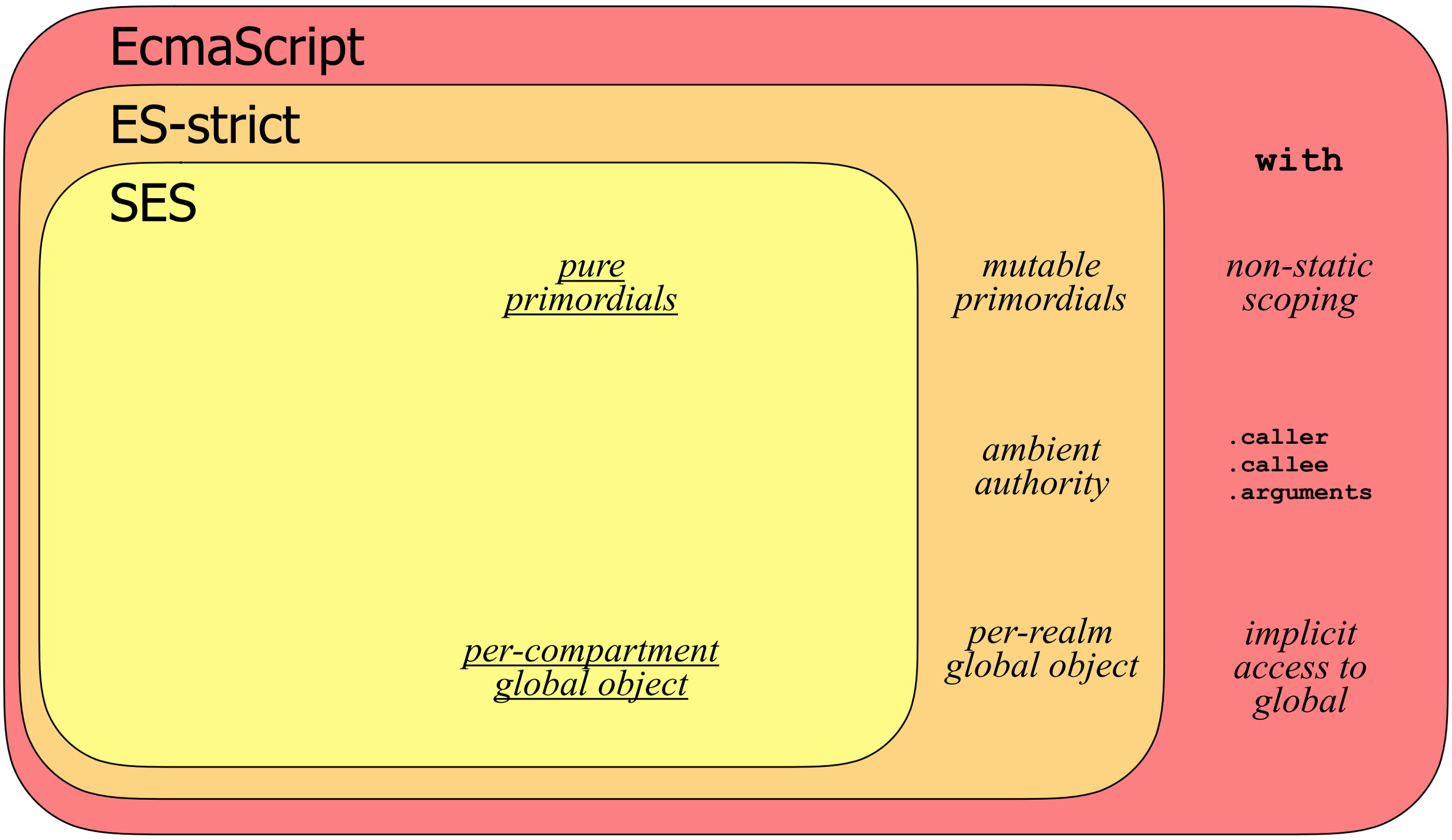
Don’t add security. Remove insecurity.



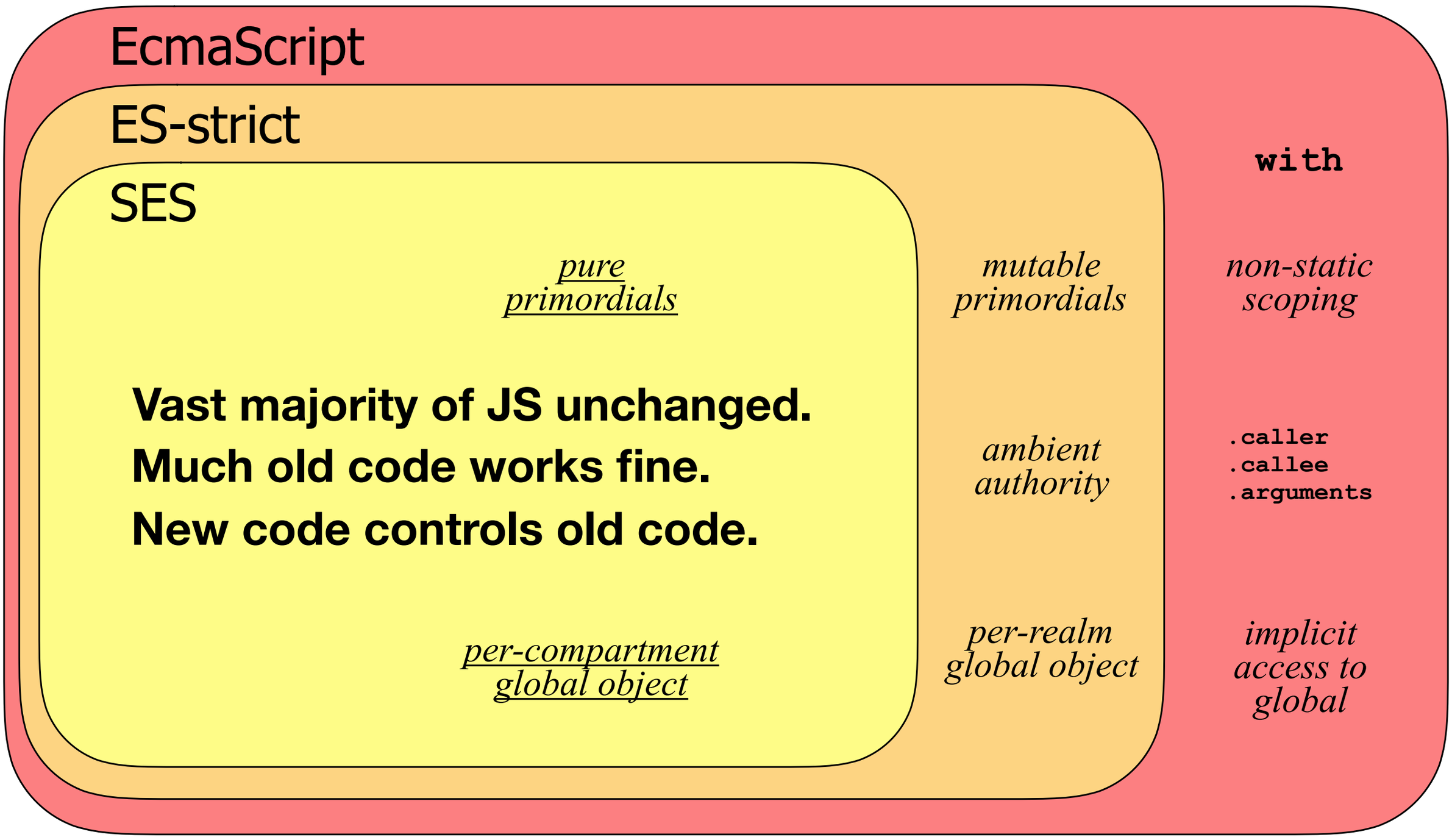
Don’t add security. Remove insecurity.



Don’t add security. Remove insecurity.



Don’t add security. Remove insecurity.



SES

pure primordials

Vast majority of JS unchanged.
Much old code works fine.
New code controls old code.

per-compartment global object

Min memory. Omit needless evaluation

SES

Typical tc53

pure primordials

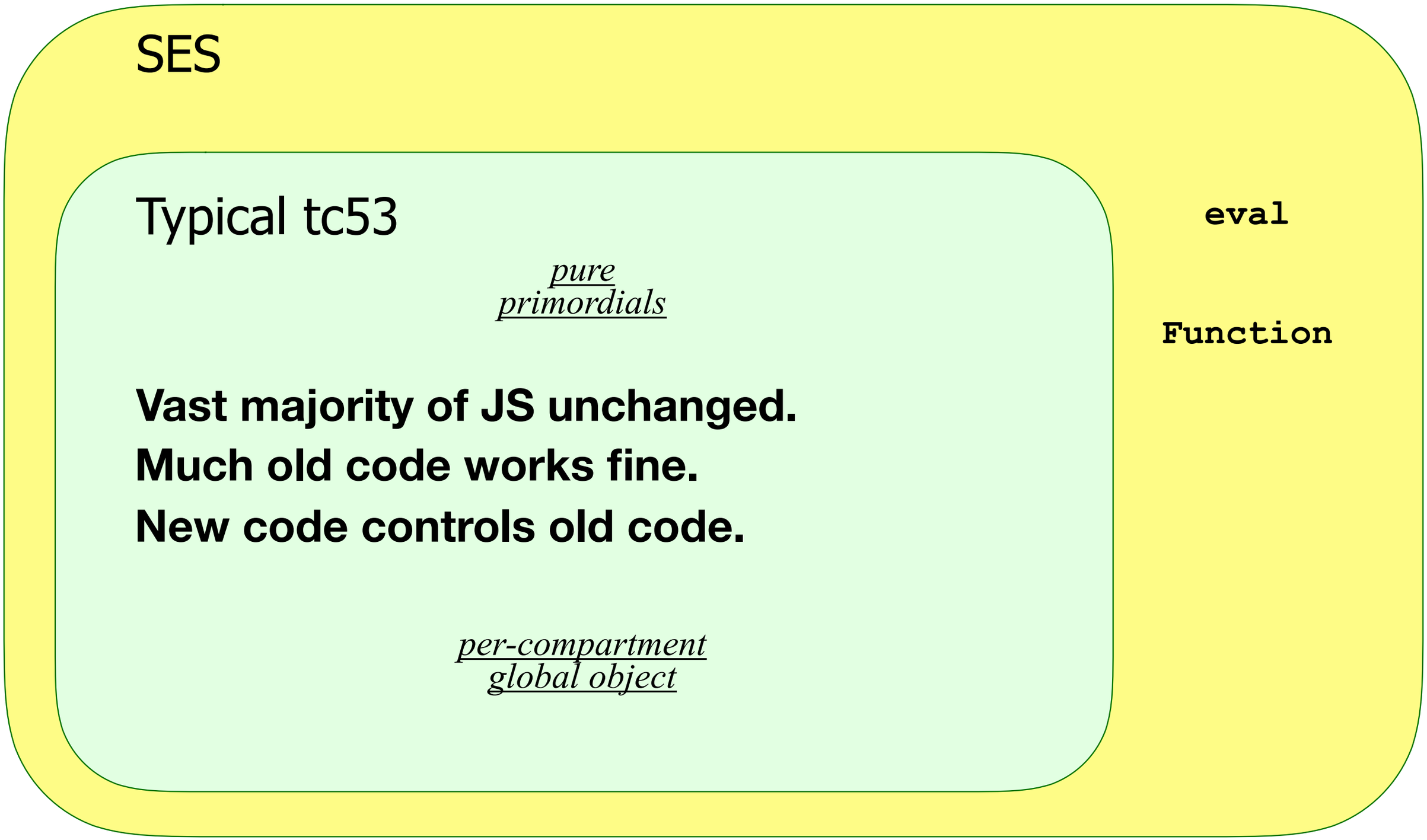
Vast majority of JS unchanged.

Much old code works fine.

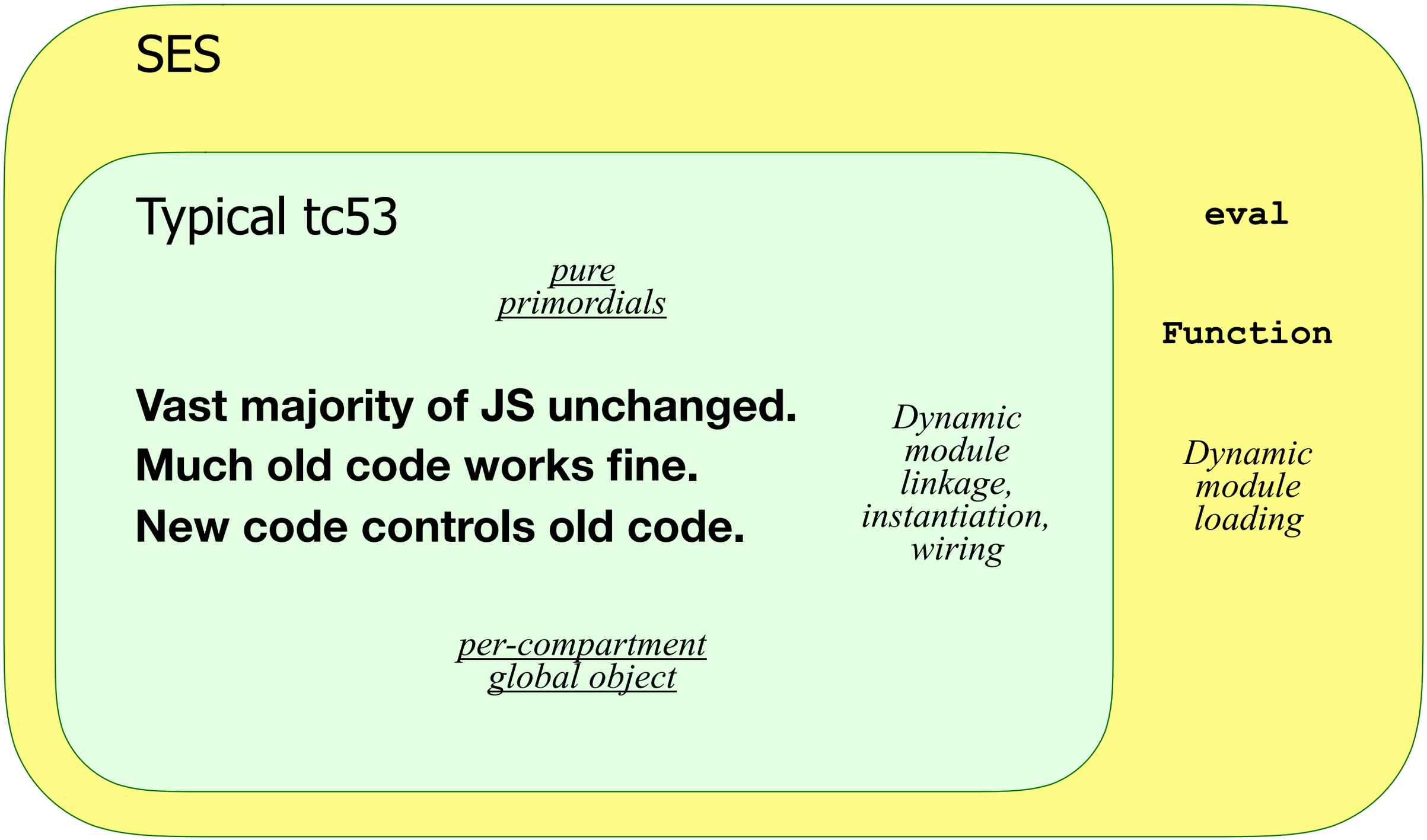
New code controls old code.

per-compartment global object

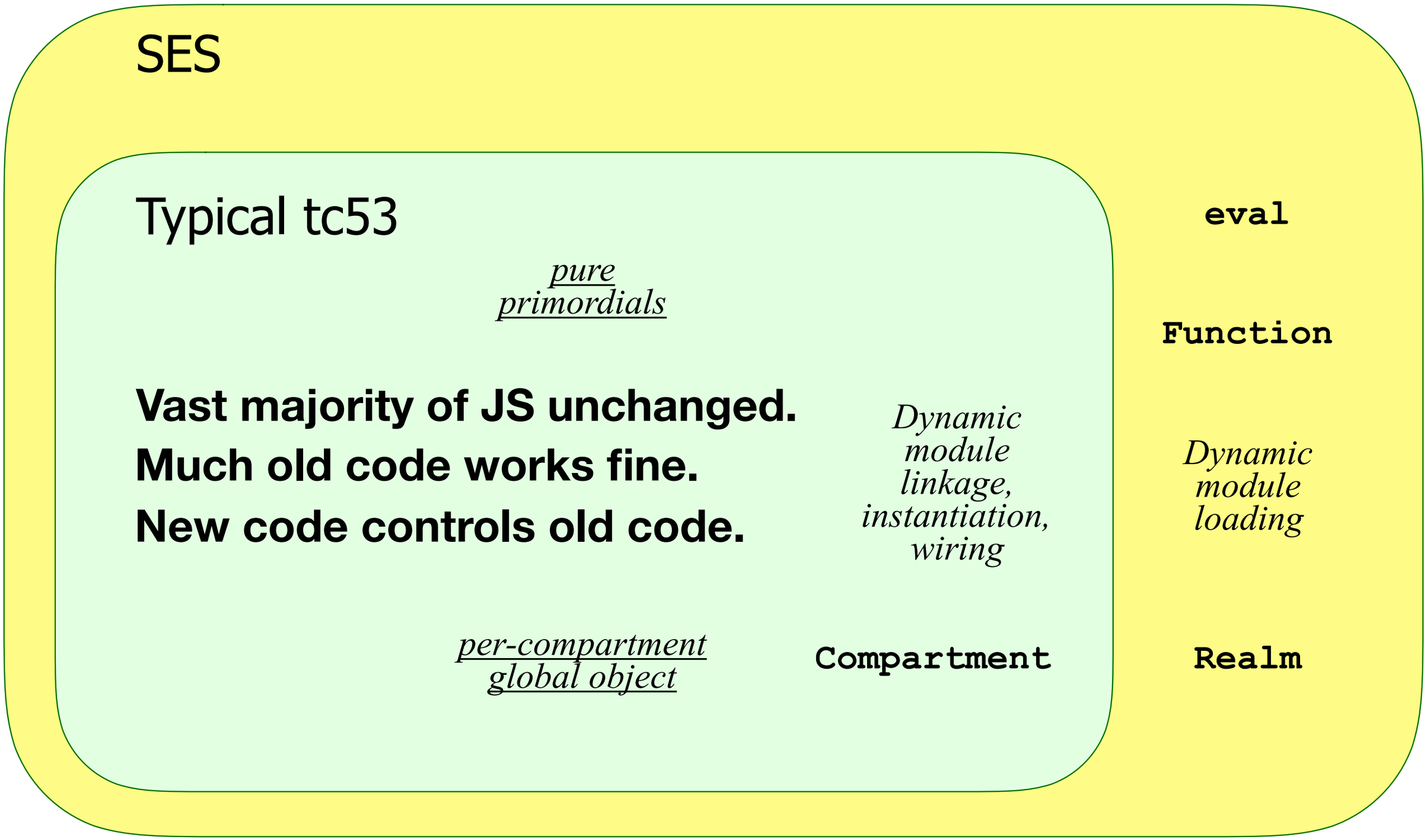
Min memory. Omit needless evaluation



Min memory. Omit needless evaluation



Min memory. Omit needless evaluation



Min memory. Omit needless evaluation

CSP

SES

Typical tc53

pure primordials

Vast majority of JS unchanged.
Much old code works fine.
New code controls old code.

per-compartment global object

Dynamic module linkage, instantiation, wiring

`Compartment`

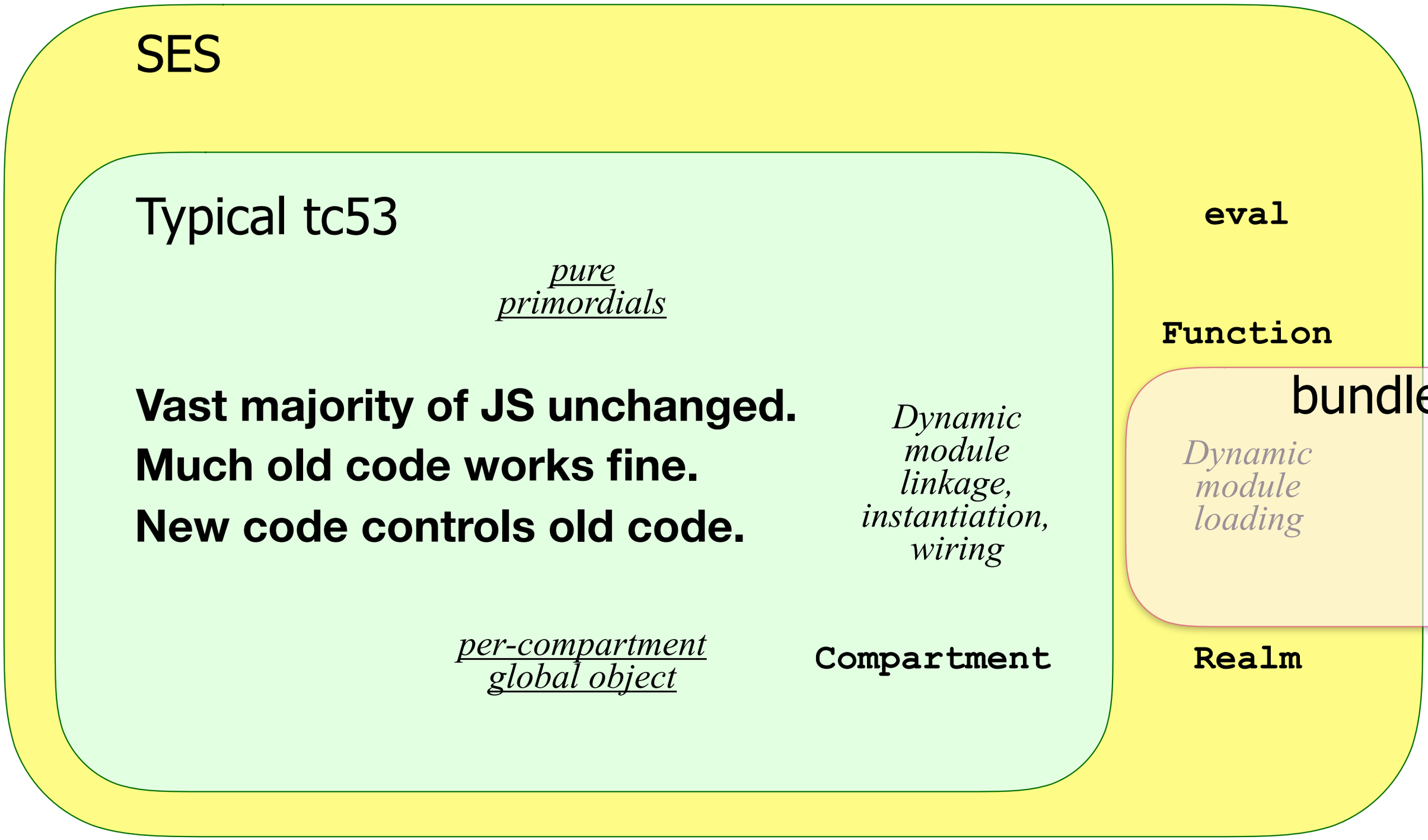
`eval`

`Function`

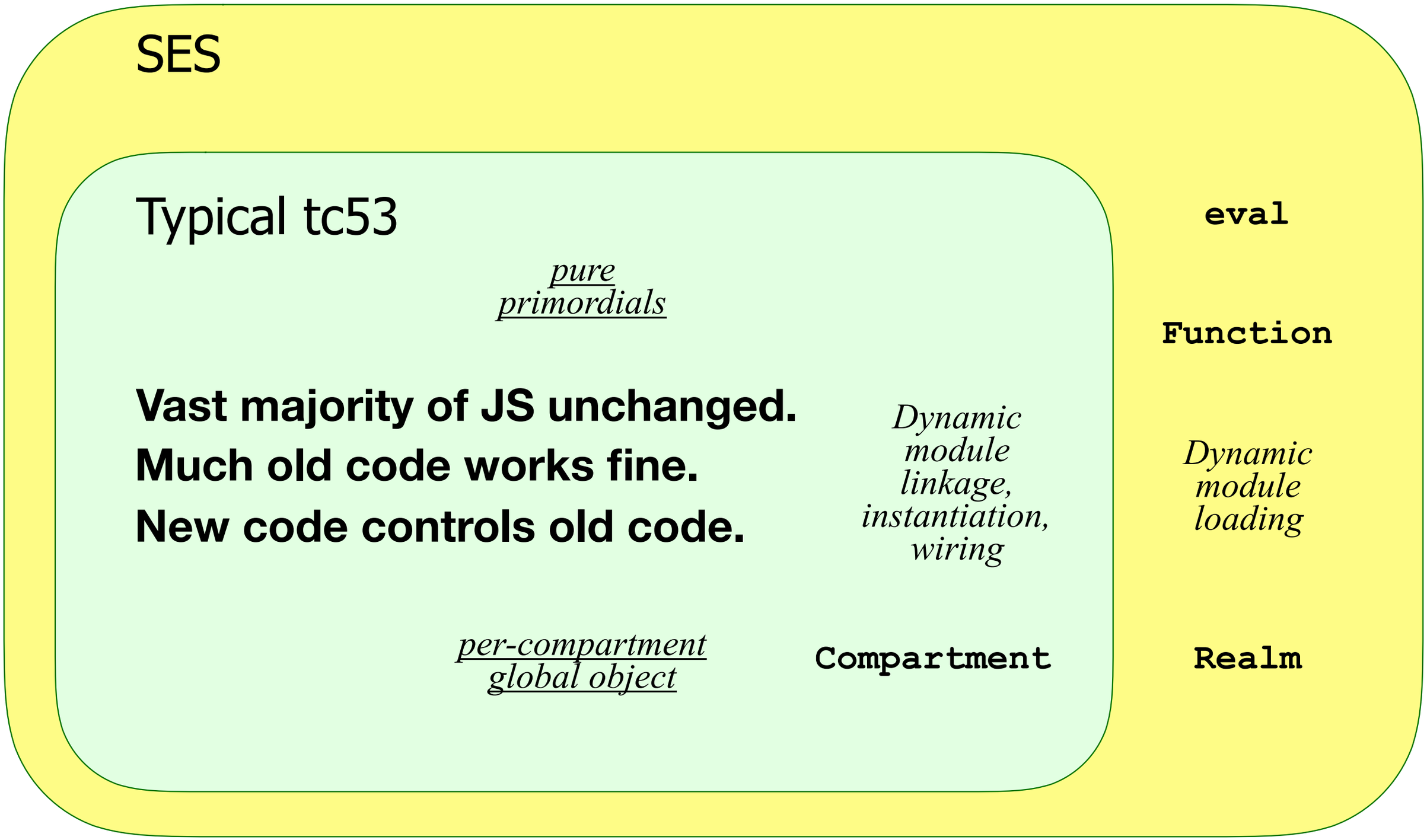
Dynamic module loading

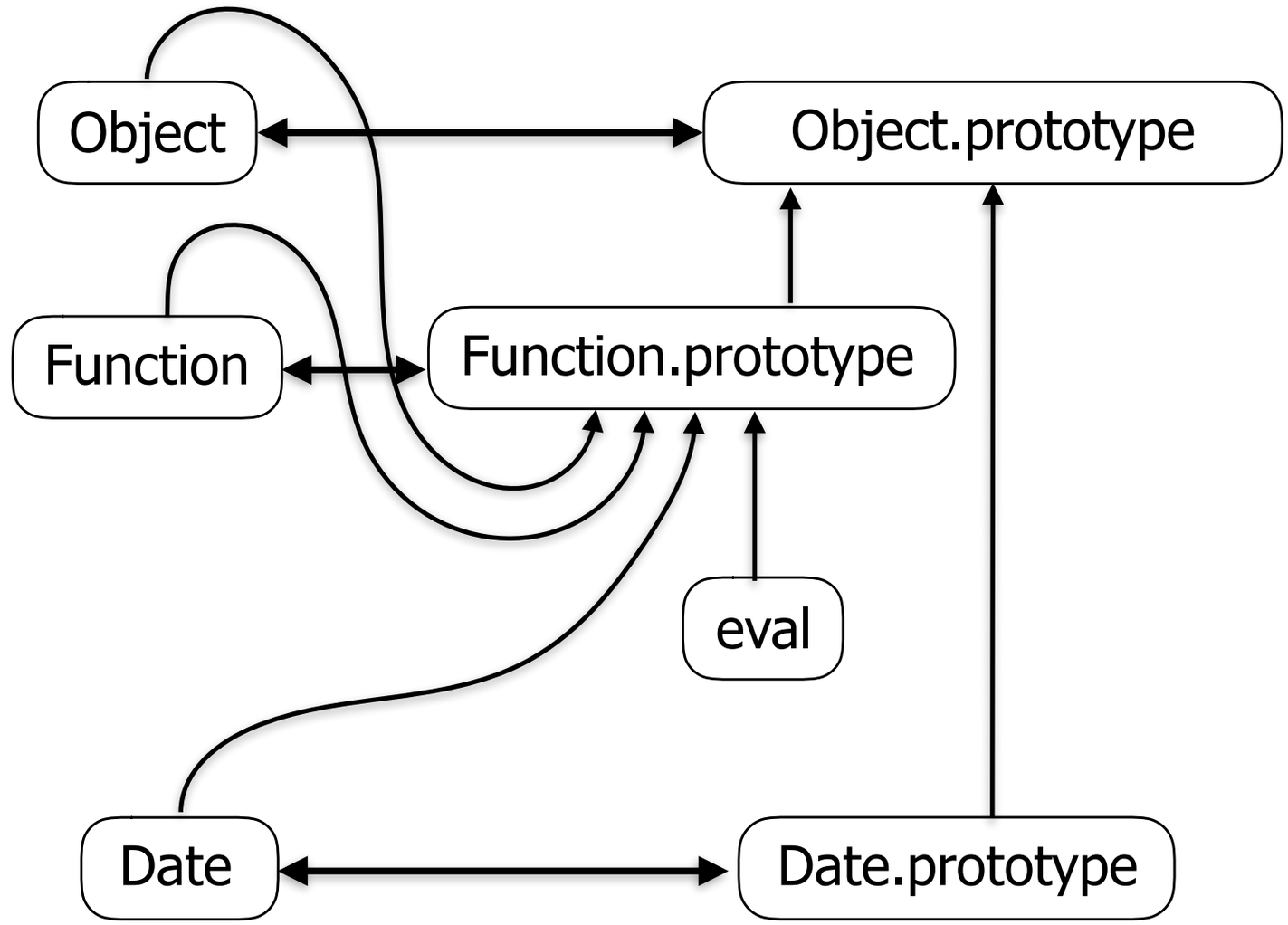
`Realm`

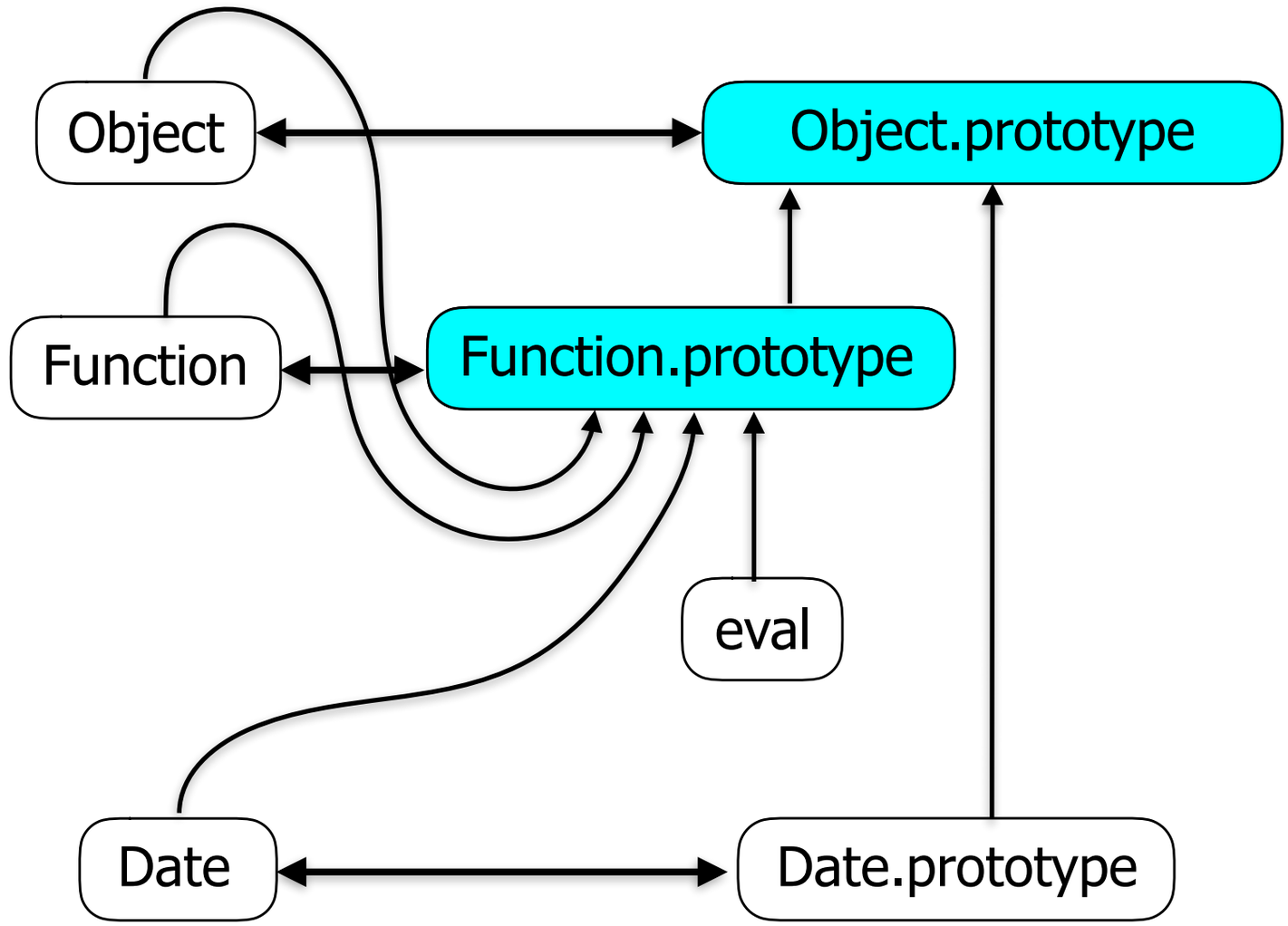
Min memory. Omit needless evaluation

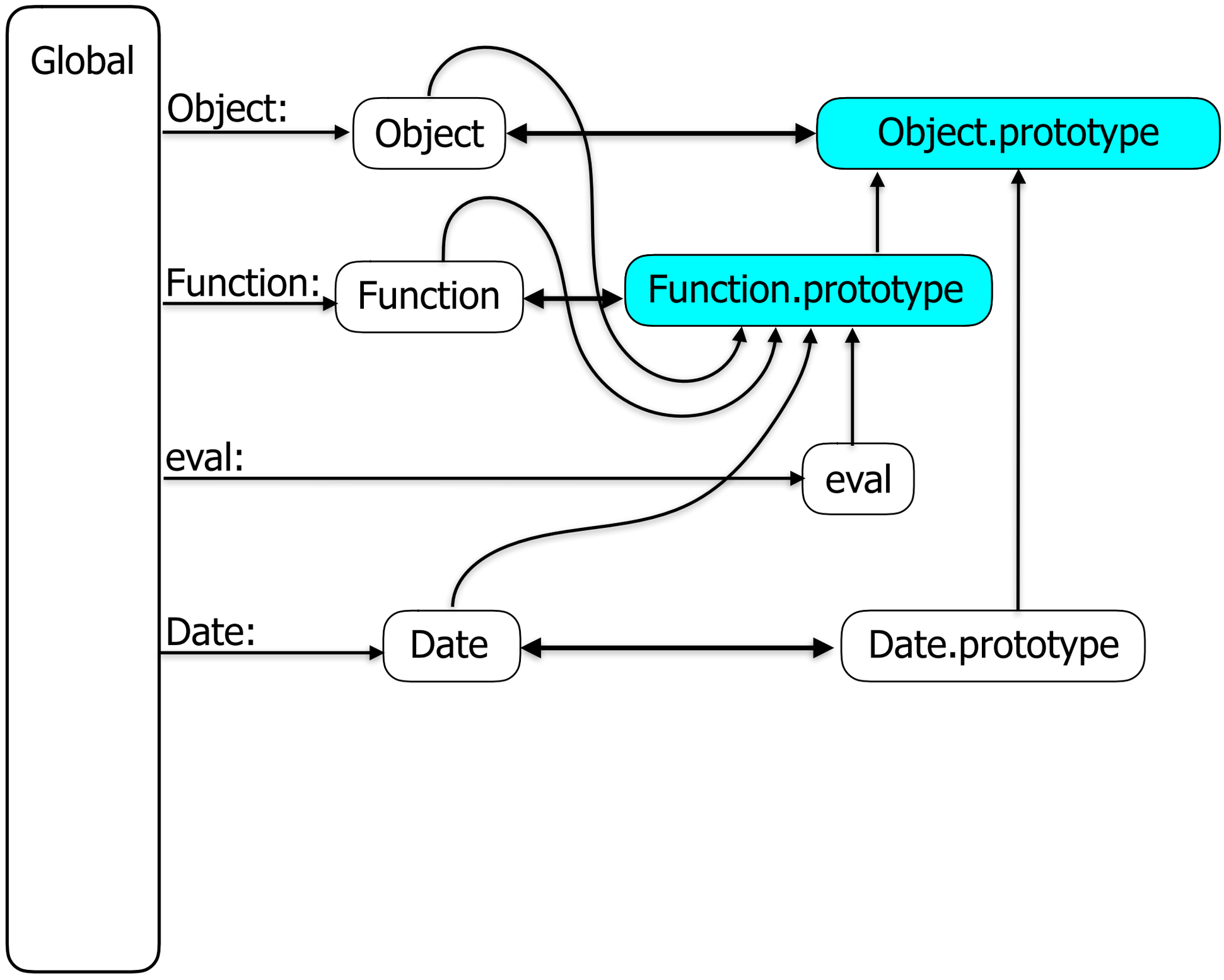


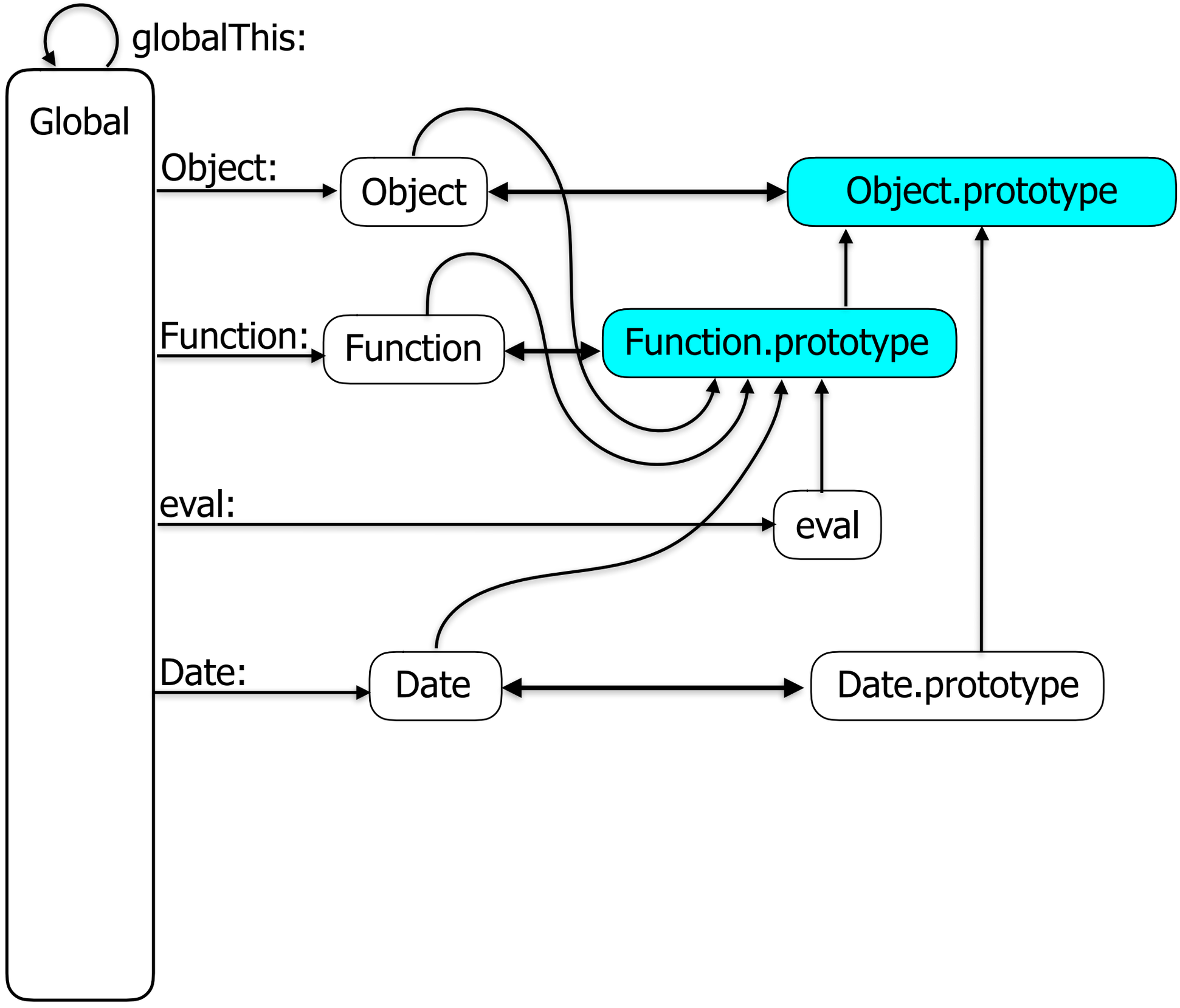
Min memory. Omit needless evaluation

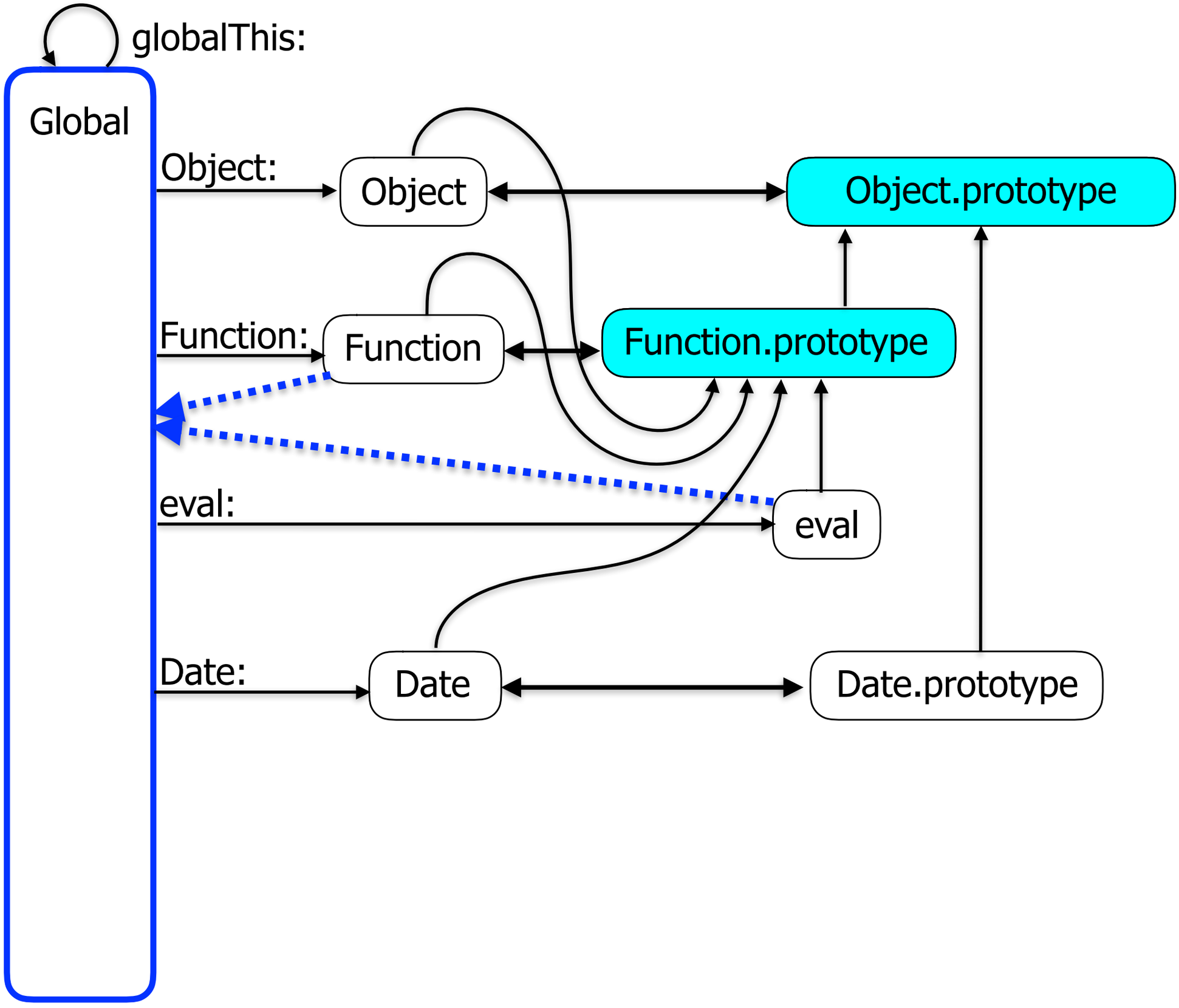


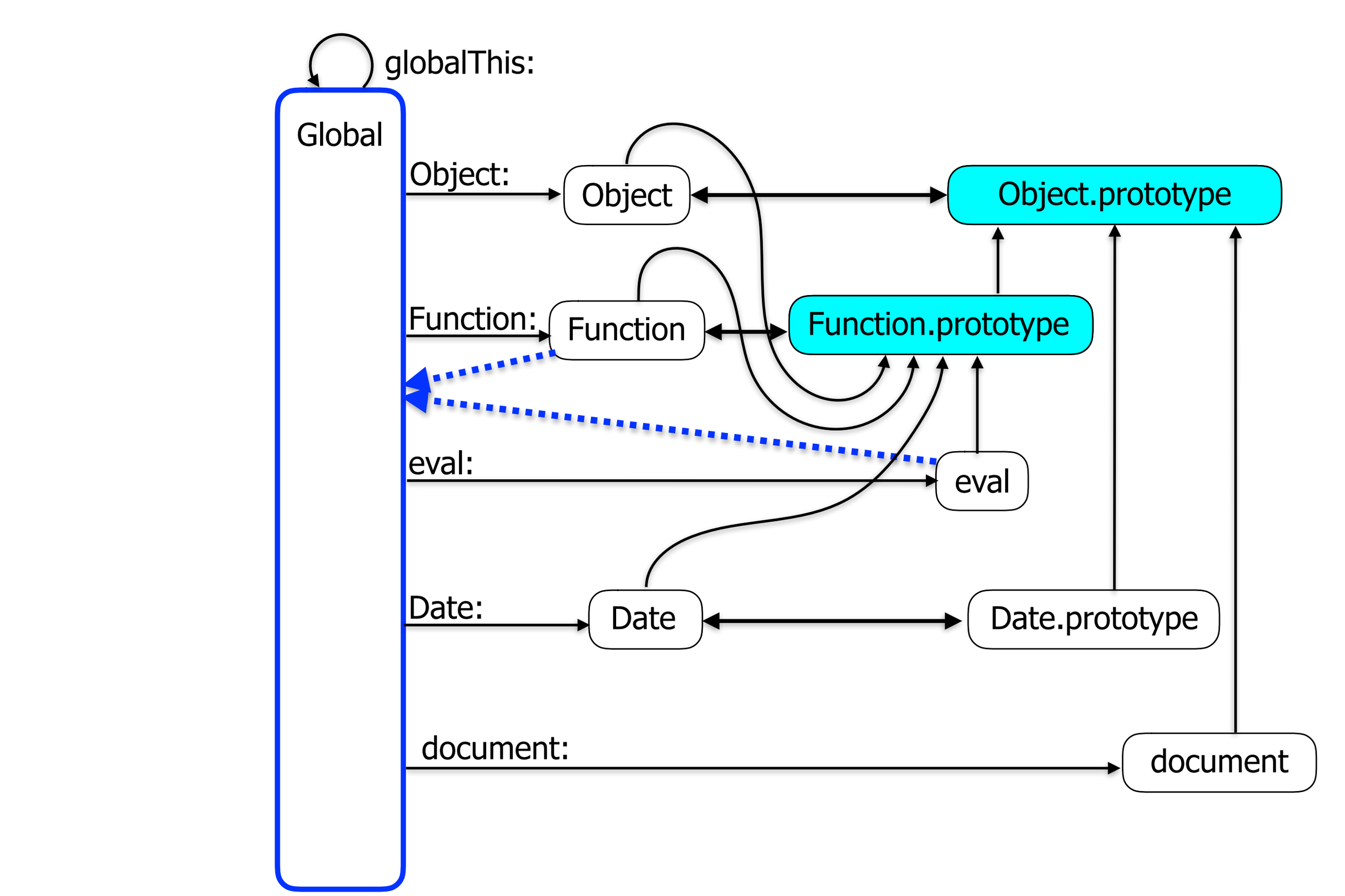

Object
Object.prototype
Function
Function.prototype
eval
Date
Date.prototype

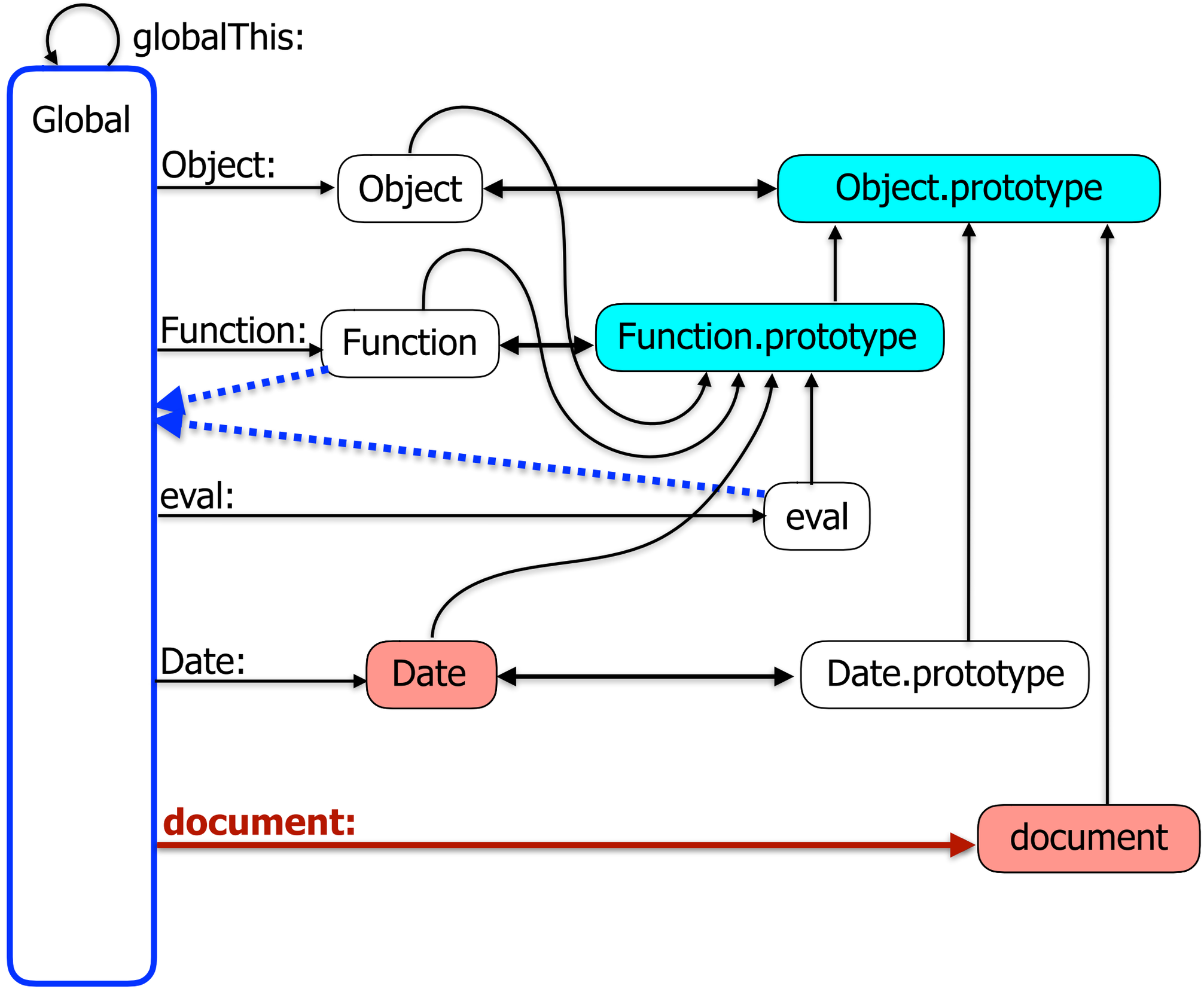

Object
Object.prototype
Function
Function.prototype
eval
Date
Date.prototype

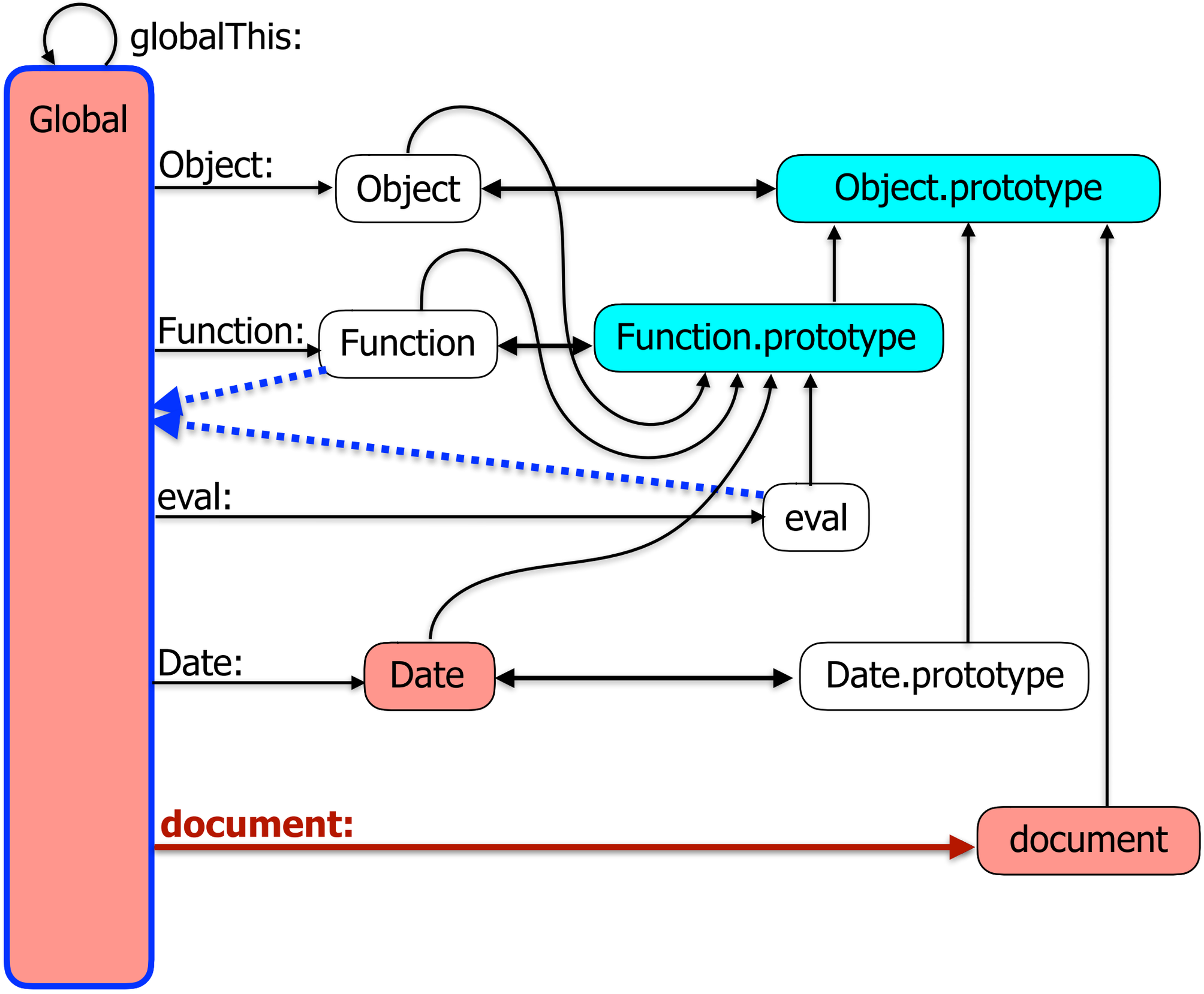

Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype

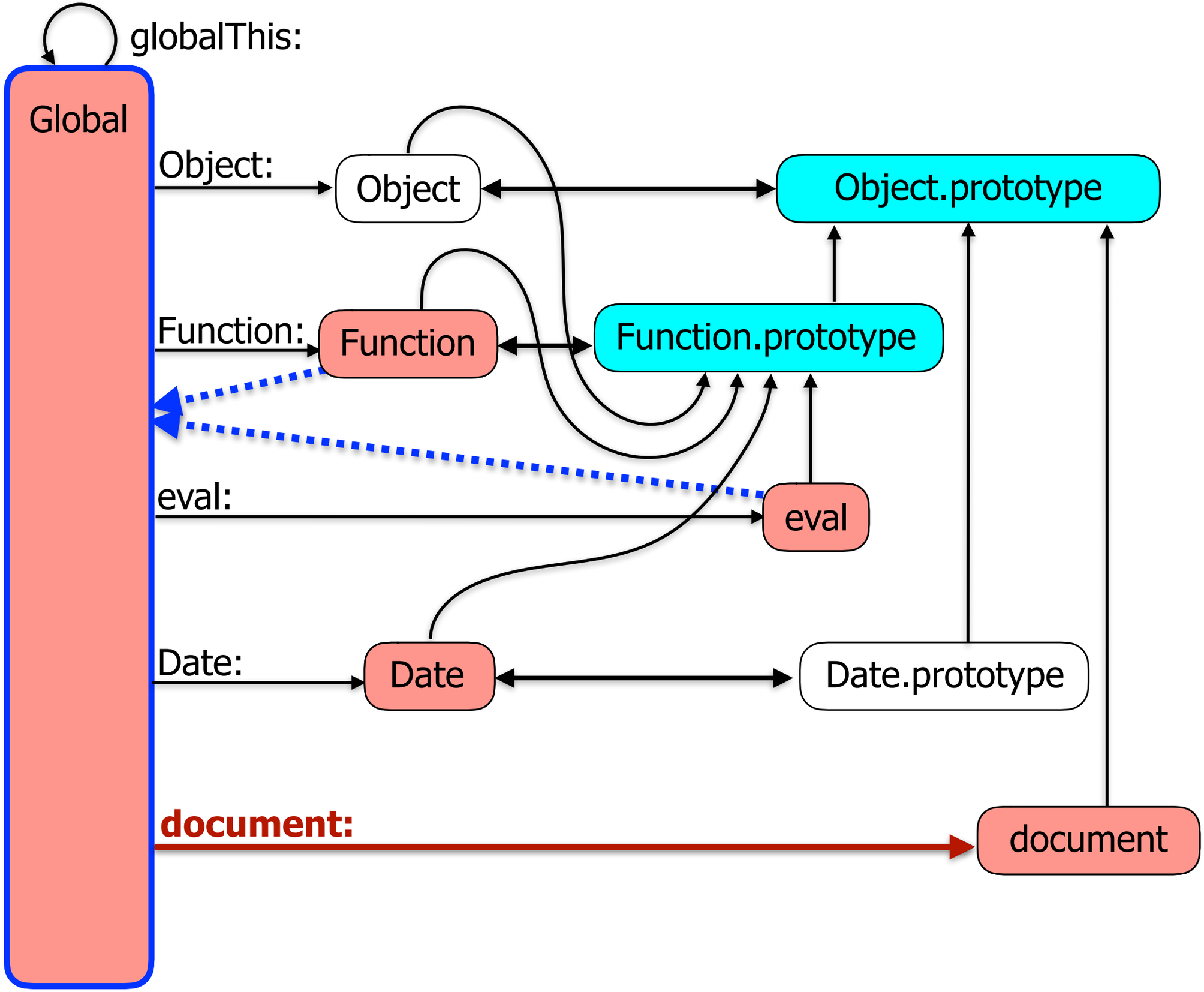

globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype

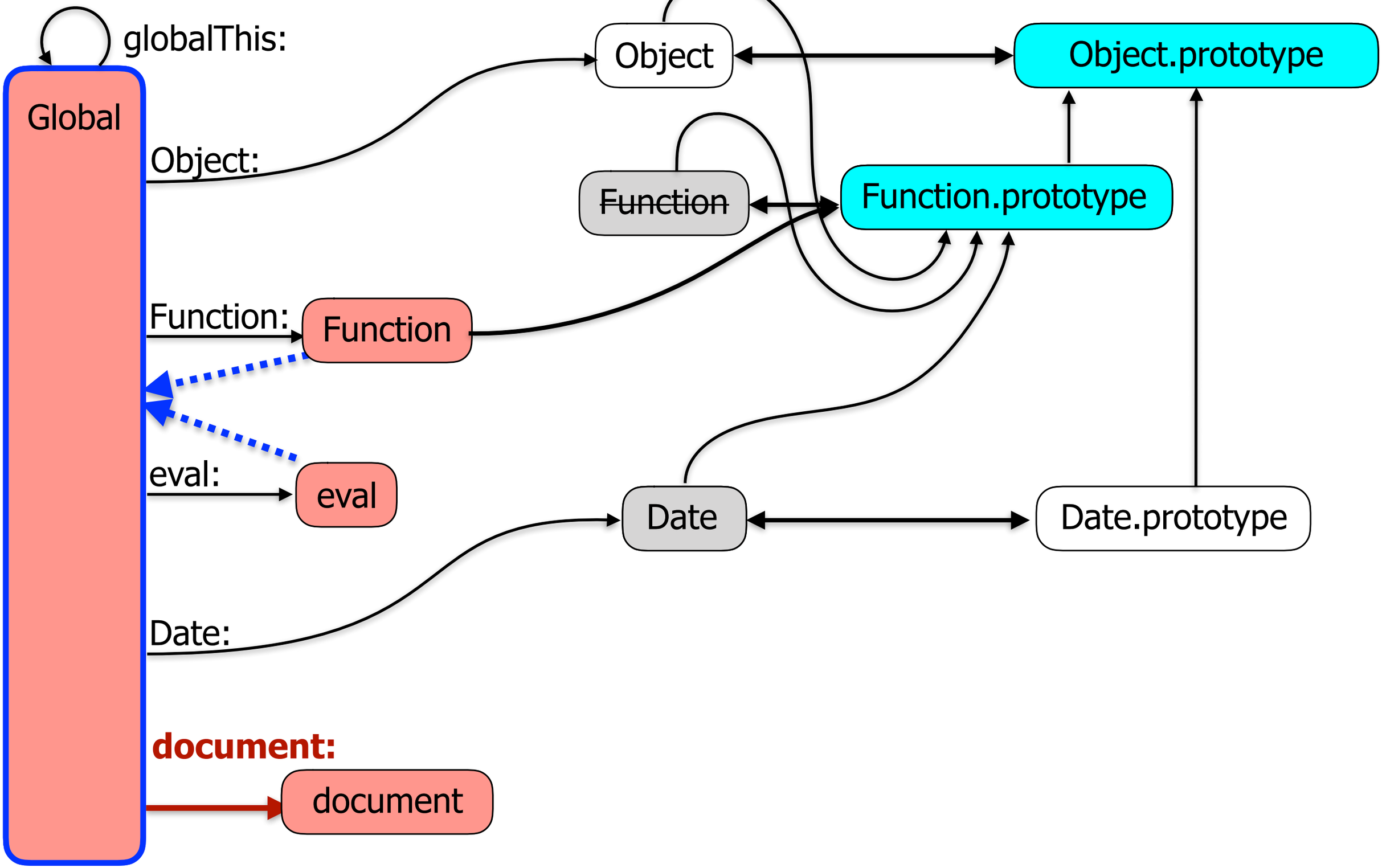

globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype


globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype
document:
document


globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype
document:
document


globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype
document:
document


globalThis:
Global
Object:
Object
Object.prototype
Function:
Function
Function.prototype
eval:
eval
Date:
Date
Date.prototype
document:
document


globalThis:
Global
Object:
Function:
eval:
Date:
document:
Object
Function
Function
eval
Date
document
Object.prototype
Function.prototype
Date.prototype

Start Compartment

globalThis:

Global

Object:

Function:

Function

eval:

eval

Date:

document:

document

Frozen Shared Intrinsics

Object

Object.prototype

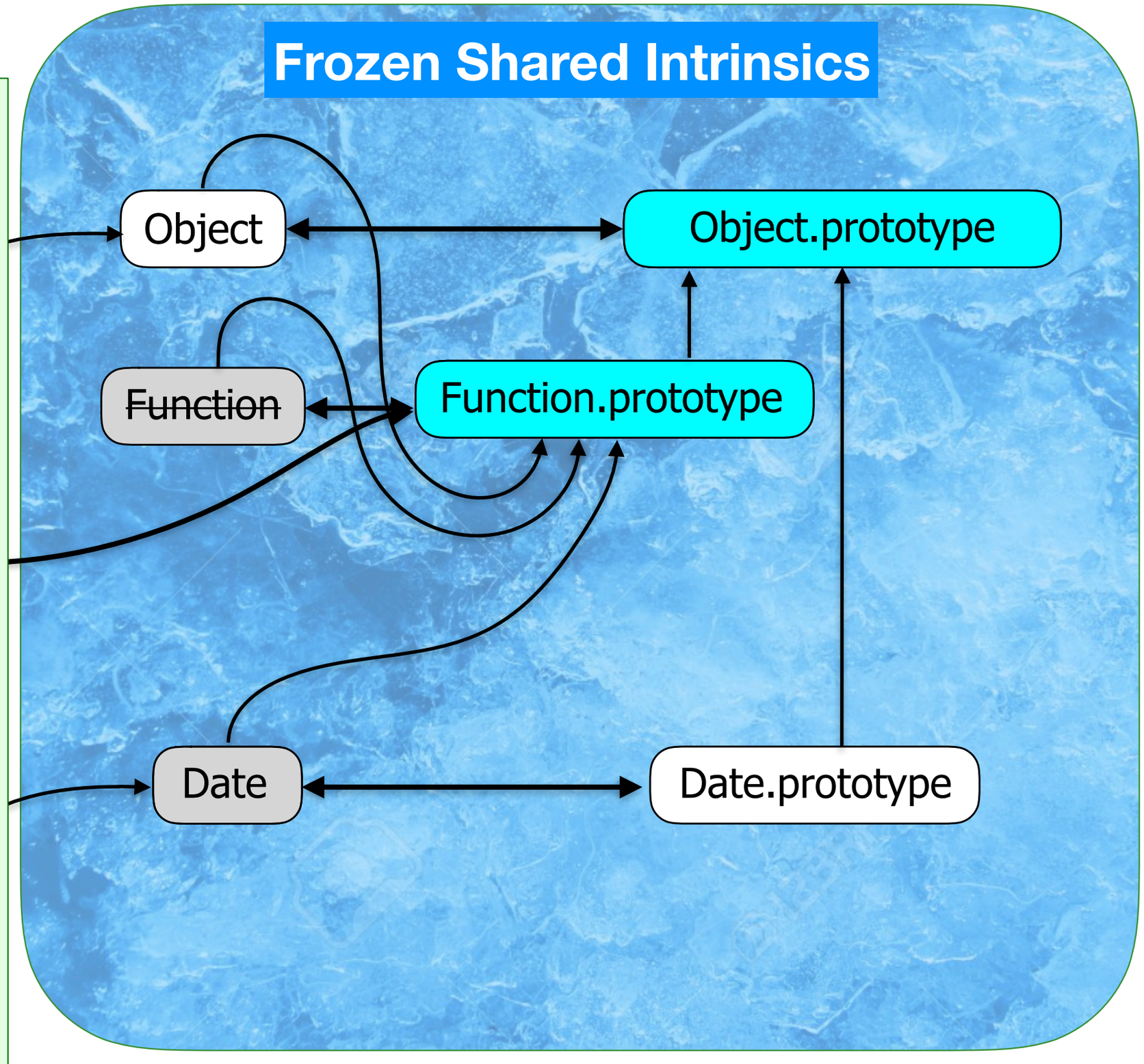
~~Function~~

Function.prototype

Date

Date.prototype

Compartment



Compartment

globalThis:

Global

Object:

Function:

Function

eval:

eval

Date:

Frozen Shared Intrinsics

Object

Object.prototype

~~Function~~

Function.prototype

Date

Date.prototype

Compartment

globalThis:

Global

Object:

Function:

Function

eval:

eval

Date:

bulb:

bulb

Frozen Shared Intrinsics

Object

Object.prototype

~~Function~~

Function.prototype

Date

Date.prototype

Compartment

globalThis:

Global

Object:

Function:

Function

eval:

eval

Date:

Compartment:

Compartment

Frozen Shared Intrinsics

Object

Object.prototype

~~Function~~

Function.prototype

Date

Date.prototype

~~Compartment~~

Compartment.prototype

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?         // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // implementation hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Global variables

Module imports

Host hooks

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // implementation host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Global variables

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

User-mode instructions

standard EcmaScript language

Trapping

Global scope lookup

Devices

System-mode instructions

Browser internals

User

Net

Process

Process

System-call interface

OS Kernel

Devices

Realm

Global scope execution context

Host objects

Browser behavior

User

Net

Process

Process

System-call interface

Devices

OS Kernel

Frozen Realm

Compartment

Compartment

Host objects

Jser

Browser behavior

Net

Process

Process

Trap trapping system call

Devices

OS Kernel

Frozen Realm

Compartment

Compartment

SES API

Browser behavior

User

Net

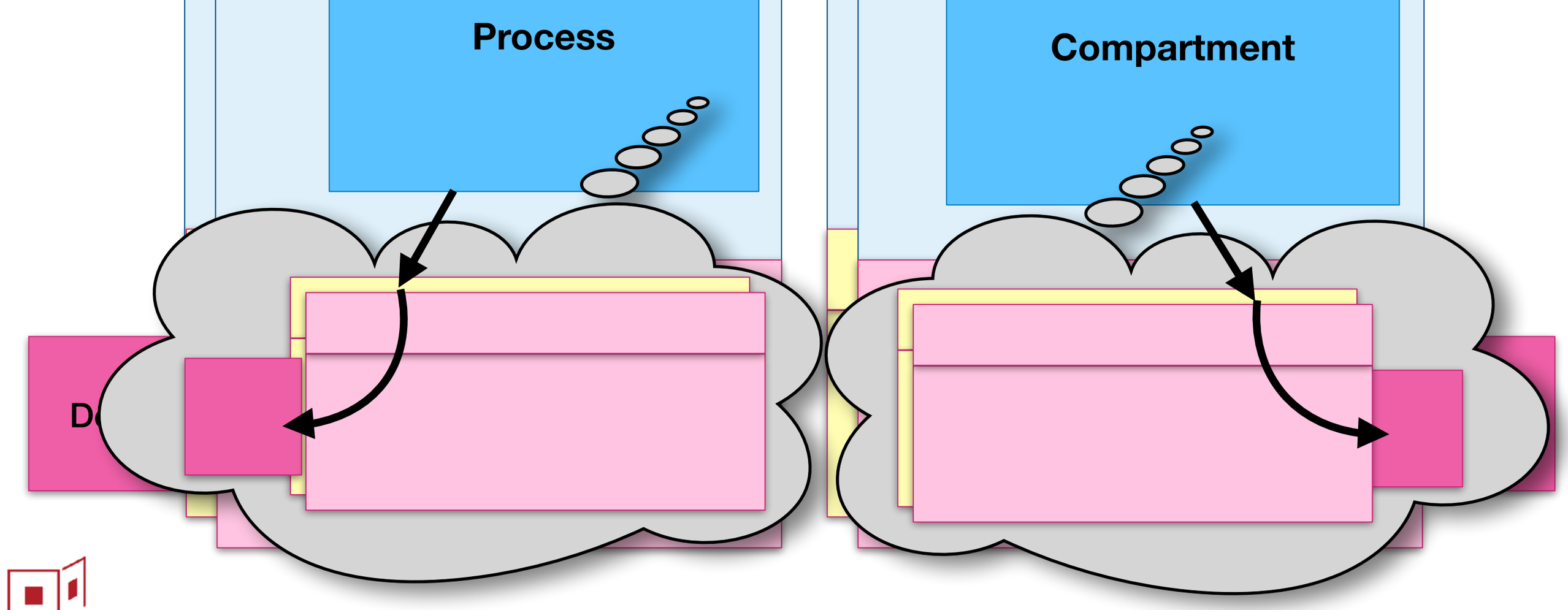
Process

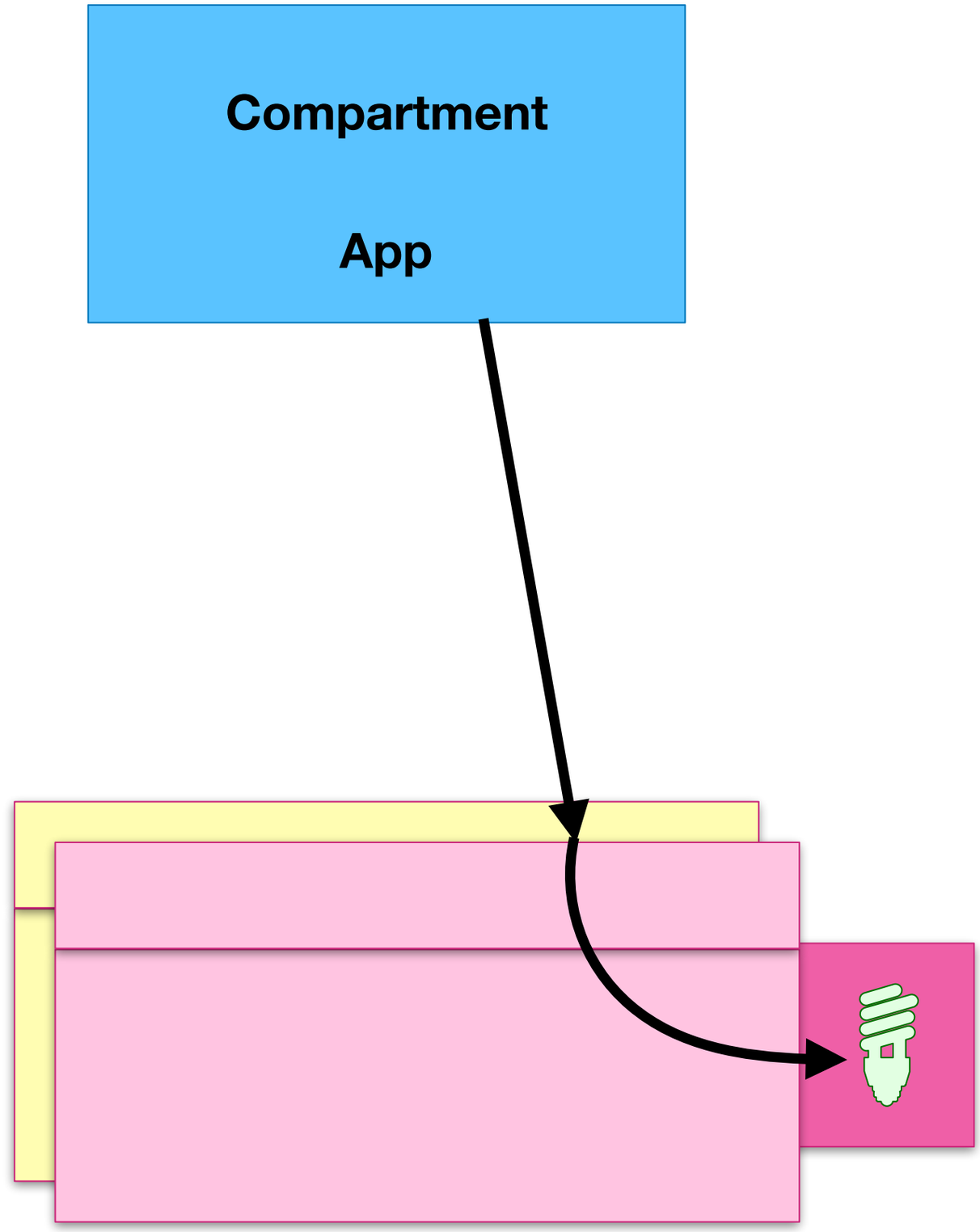
Process

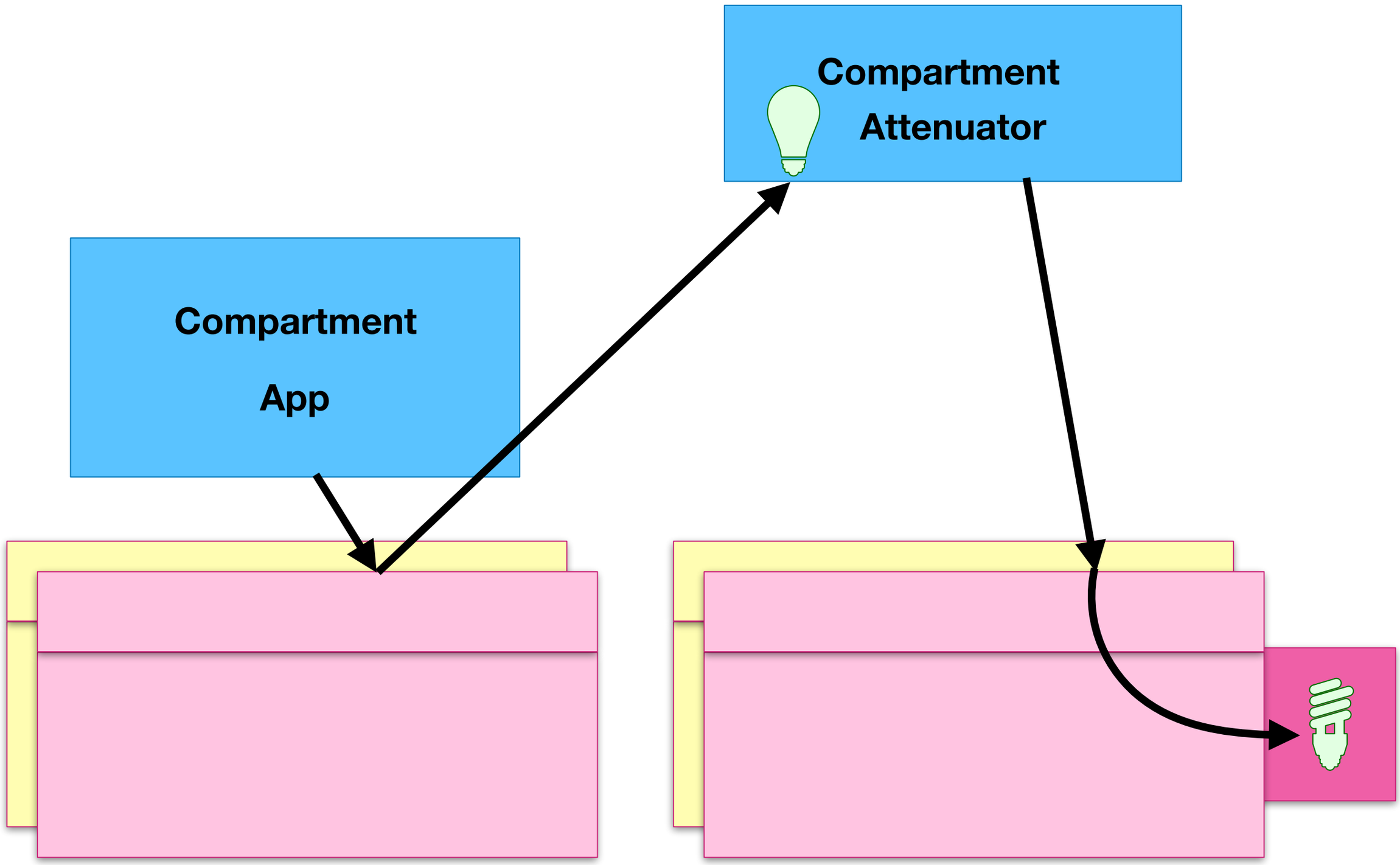
Frozen Realm

Compartment

Compartment




Compartment
App


Compartment
Attenuator
Compartment
App

Compartment

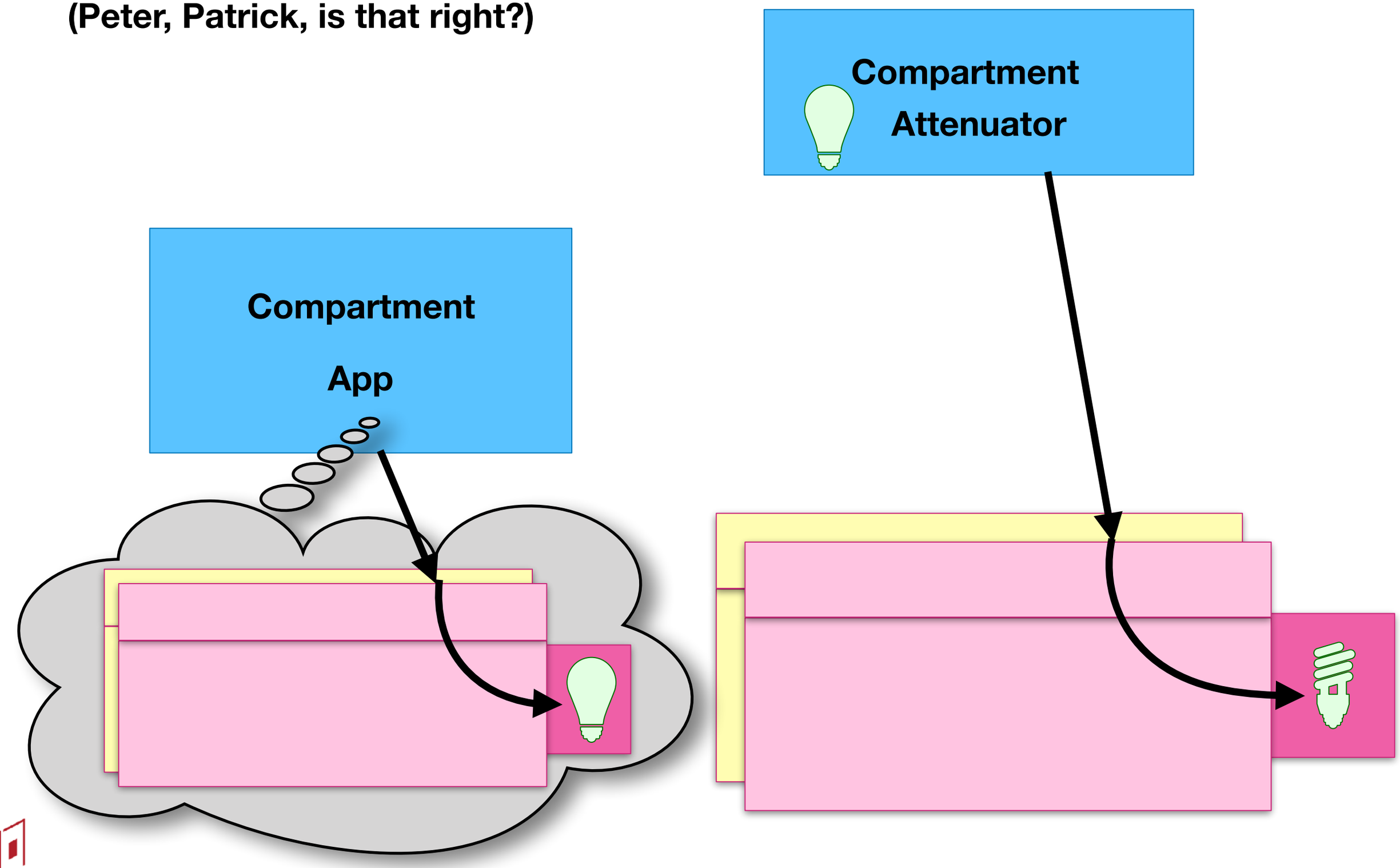
App

Compartment

Attenuator

40K RAM
1M ROM

(Peter, Patrick, is that right?)



```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // implementation host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Global variables

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object
```

Module imports

```
  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

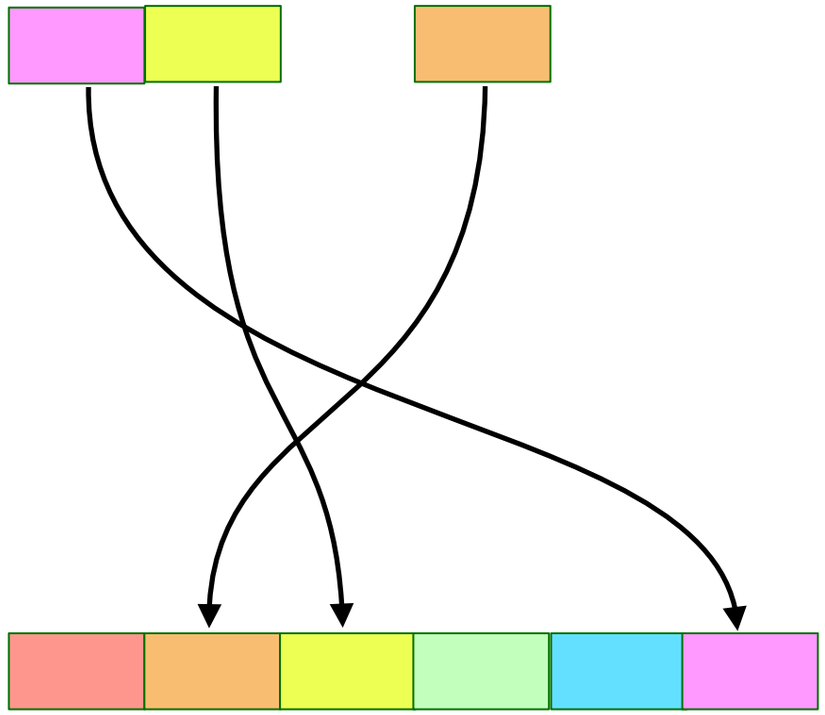
Intermission

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

virtual address

MMU

physical address



physical page

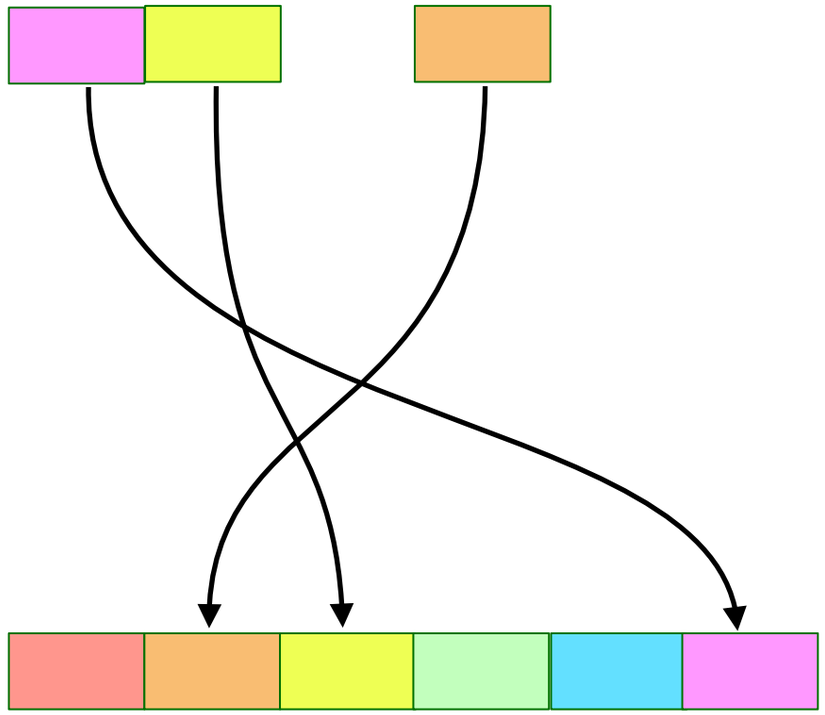
map map map

virtual page trap trap trap

child specifier

Module Map

parent specifier

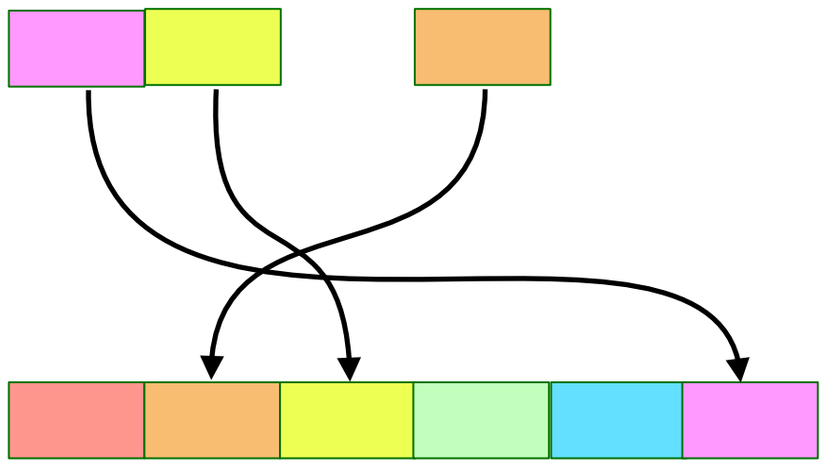


loaded static module

map map map

Module import hook trap trap trap

child specifier



parent specifier

loaded static module

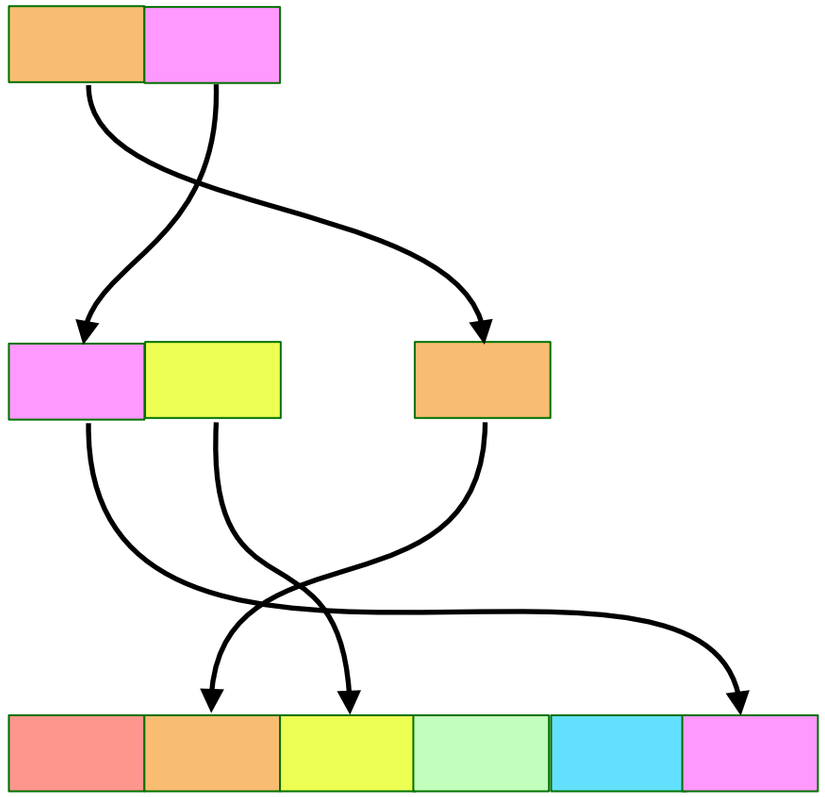
Module import hook

trap trap trap

grandchild specifier

child specifier

parent specifier



loaded static module

map map map

Module import hook trap trap trap

ModuleRecord

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

[[Context]]

ModuleRecord

[[RealmRecord]] : RealmRecord?

[[Environment]] : LexicalEnvironment?

[[Namespace]] : ModuleNamespace?

[[HostDefined]] : Any?

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]] : String[]

[[Status]] : unlinked | linking | linked | evaluating | evaluated

[[EvalError]] : AbruptCompletion?

[[DFSIndex]] : Integer?

[[DFSAncestorIndex]] : Integer?

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]] : ParseNode

[[ImportEntries]] : ImportEntry[]

[[*ExportEntries]] : ExportEntry[]

[[Context]] : ExecutionContext

ModuleRecord

[[RealmRecord]] : RealmRecord?

[[Environment]] : LexicalEnvironment?

[[Namespace]] : ModuleNamespace?

[[HostDefined]] : Any?

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]] : String[]

[[Status]] : unlinked | linking | linked | evaluating | evaluated

[[EvalError]] : AbruptCompletion?

[[DFSIndex]] : Integer?

[[DFSAncestorIndex]] : Integer?

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]] : ParseNode

[[ImportEntries]] : ImportEntry[]

[[*ExportEntries]] : ExportEntry[]

[[Context]] : ExecutionContext

Mixes concerns.

Messy.

Let’s refactor.

ModuleRecord

[[RealmRecord]] : RealmRecord?

[[Environment]] : LexicalEnvironment?

[[Namespace]] : ModuleNamespace?

[[HostDefined]] : Any?

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]] : String[]

[[Status]] : unlinked | linking | linked | evaluating | evaluated

[[EvalError]] : AbruptCompletion?

[[DFSIndex]] : Integer?

[[DFSAncestorIndex]] : Integer?

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]] : ParseNode

[[ImportEntries]] : ImportEntry[]

[[*ExportEntries]] : ExportEntry[]

[[Context]] : ExecutionContext

ModuleRecord

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

[[Context]]

ModuleRecord

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

CyclicModuleRecord

extends ModuleRecord

[[RequestedModules]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

SourceTextModuleRecord

extends CyclicModuleRecord

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

[[Context]]

StaticModuleRecord

[[RequestedModules]]

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ModuleInitialization

[[ModuleInstance]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

[[Context]]

StaticModuleRecord

[[RequestedModules]]

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ModuleInitialization

[[ModuleInstance]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

[[Context]]

StaticModuleRecord

[[RequestedModules]]

[[ECMAScriptCode]]

[[ImportEntries]]

[[*ExportEntries]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ModuleInitialization

[[ModuleInstance]]

[[Status]]

[[EvalError]]

[[DFSIndex]]

[[DFSAncestorIndex]]

[[Context]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

RealmRecord

[[Intrinsics]] : IntrinsicName —> Object

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

RealmRecord

[[Intrinsics]] : IntrinsicName —> Object

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any

So far, only pure refactoring.

No observable changes.

No extra state.

Clearer.

Sets the stage for…

ModuleInstance

[[StaticModuleRecord]]

[[RealmRecord]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[RealmRecord]]

[[Environment]]

[[HostDefined]]

RealmRecord

[[Intrinsics]] : IntrinsicName —> Object

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

ModuleInstance

[[StaticModuleRecord]]

[[Compartment]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[Compartment]]

[[Environment]]

[[HostDefined]]

Compartment

[[Intrinsics]] : IntrinsicName —> Object

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

ModuleInstance

[[StaticModuleRecord]]

[[Compartment]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[Compartment]]

[[Environment]]

[[HostDefined]]

Compartment

[[SharedIntrinsics]] : IntrinsicName —> Object // per-Realm

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

ModuleInstance

[[StaticModuleRecord]]

[[Compartment]]

[[Environment]]

[[Namespace]]

[[HostDefined]]

ScriptInstance

[[ECMAScriptCode]]

[[Compartment]]

[[Environment]]

[[HostDefined]]

Compartment

[[SharedIntrinsics]] : IntrinsicName —> Object // per-Realm

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

[[ModuleMap]] : name —> (StaticModuleRecord | ModuleInstance)

[[Hooks]] : Virtual Host Hooks

Compartment

[[SharedIntrinsics]] : IntrinsicName —> Object // per-Realm

[[GlobalObject]] : Object

[[GlobalEnv]] : LexicalEnvironment

[[TemplateMap]] : ParseNode —> Template

[[HostDefined]] : Any?

[[ModuleMap]] : name —> (StaticModuleRecord | ModuleInstance)

[[Hooks]] : Virtual Host Hooks

Compartment

- [[SharedIntrinsics]]
- [[GlobalObject]]
- [[GlobalEnv]]
- [[TemplateMap]]
- [[HostDefined]]
- [[ModuleMap]]
- [[Hooks]]

Compartment

[[SharedIntrinsics]]

[[GlobalObject]]

[[GlobalEnv]]

[[TemplateMap]]

[[HostDefined]]

[[ModuleMap]]

[[Hooks]]

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                // strict indirect eval
    src: stringable,
    options: object?       // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Compartment

[[SharedIntrinsics]]

[[GlobalObject]]

[[GlobalEnv]]

[[TemplateMap]]

[[HostDefined]]

[[ModuleMap]]

[[Hooks]]

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                // strict indirect eval
    src: stringable,
    options: object?       // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Compartment

[[SharedIntrinsics]]

[[GlobalObject]]

[[GlobalEnv]]

[[TemplateMap]]

[[HostDefined]]

[[ModuleMap]]

[[Hooks]]

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                // strict indirect eval
    src: stringable,
    options: object?       // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Compartment

[[SharedIntrinsics]]

[[GlobalObject]]

[[GlobalEnv]]

[[TemplateMap]]

[[HostDefined]]

[[ModuleMap]]

[[Hooks]]

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?         // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object
```

Module imports

```
  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                    // strict indirect eval
    src: stringable,
    options: object?           // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

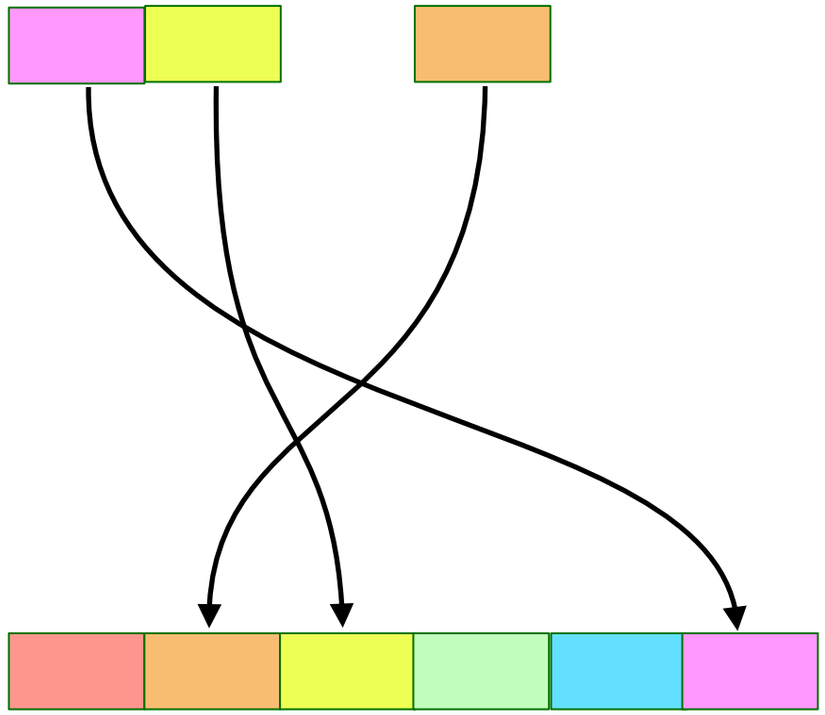
Host hooks

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?         // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

virtual address

MMU

physical address



physical page

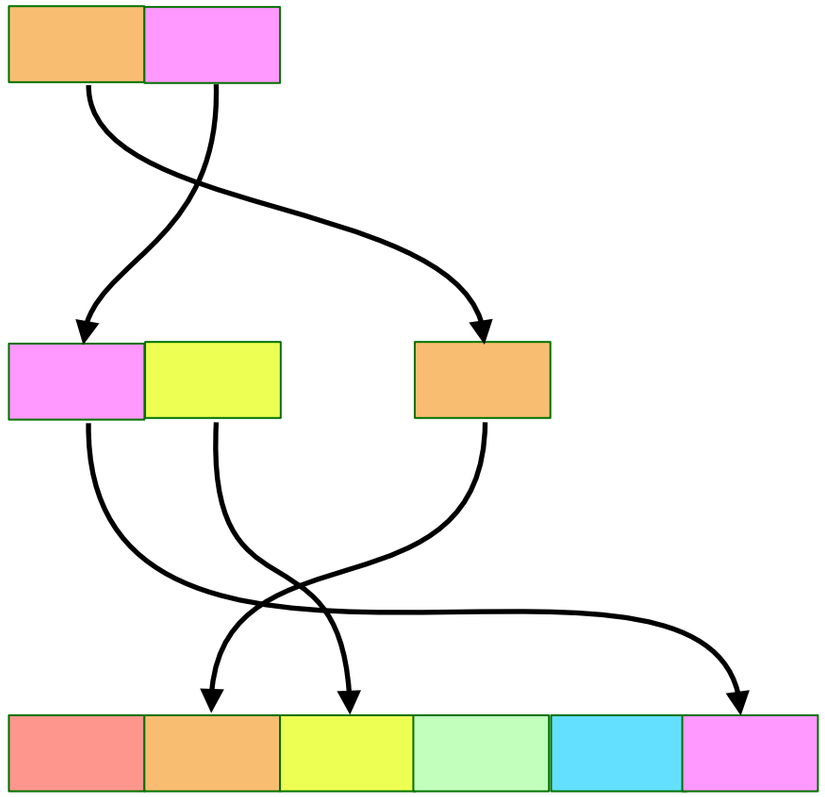
map map map

virtual page trap trap trap

grandchild specifier

child specifier

parent specifier



loaded static module

map map map

Module import hook trap trap trap

```
importNowHook(specifier, referrer): StaticModuleRecord;

importHook(specifier, referrer): Promise<StaticModuleRecord>;

...
```

Compartment

- [[SharedIntrinsics]]
- [[GlobalObject]]
- [[GlobalEnv]]
- [[TemplateMap]]
- [[HostDefined]]
- [[ModuleMap]]
- **[[Hooks]]**

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?     // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Compartment

- [[SharedIntrinsics]]
- [[GlobalObject]]
- [[GlobalEnv]]
- [[TemplateMap]]
- [[HostDefined]]
- **[[ModuleMap]]**
- **[[Hooks]]**

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?     // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

```
class Compartment {
  constructor: (
    endowments: object?, // extra globals
    moduleMap: (name -> name | ModuleNamespace)?,
    options: object?     // including host hooks
  ) -> object

  get globalThis -> object

  evaluate(                  // strict indirect eval
    src: stringable,
    options: object?     // per-evaluation
  ) -> any

  importNow(name) -> ModuleNamespace
  async import(name) -> promise<ModuleNamespace>
}
```

Compartment

[[SharedIntrinsics]]

[[GlobalObject]]

[[GlobalEnv]]

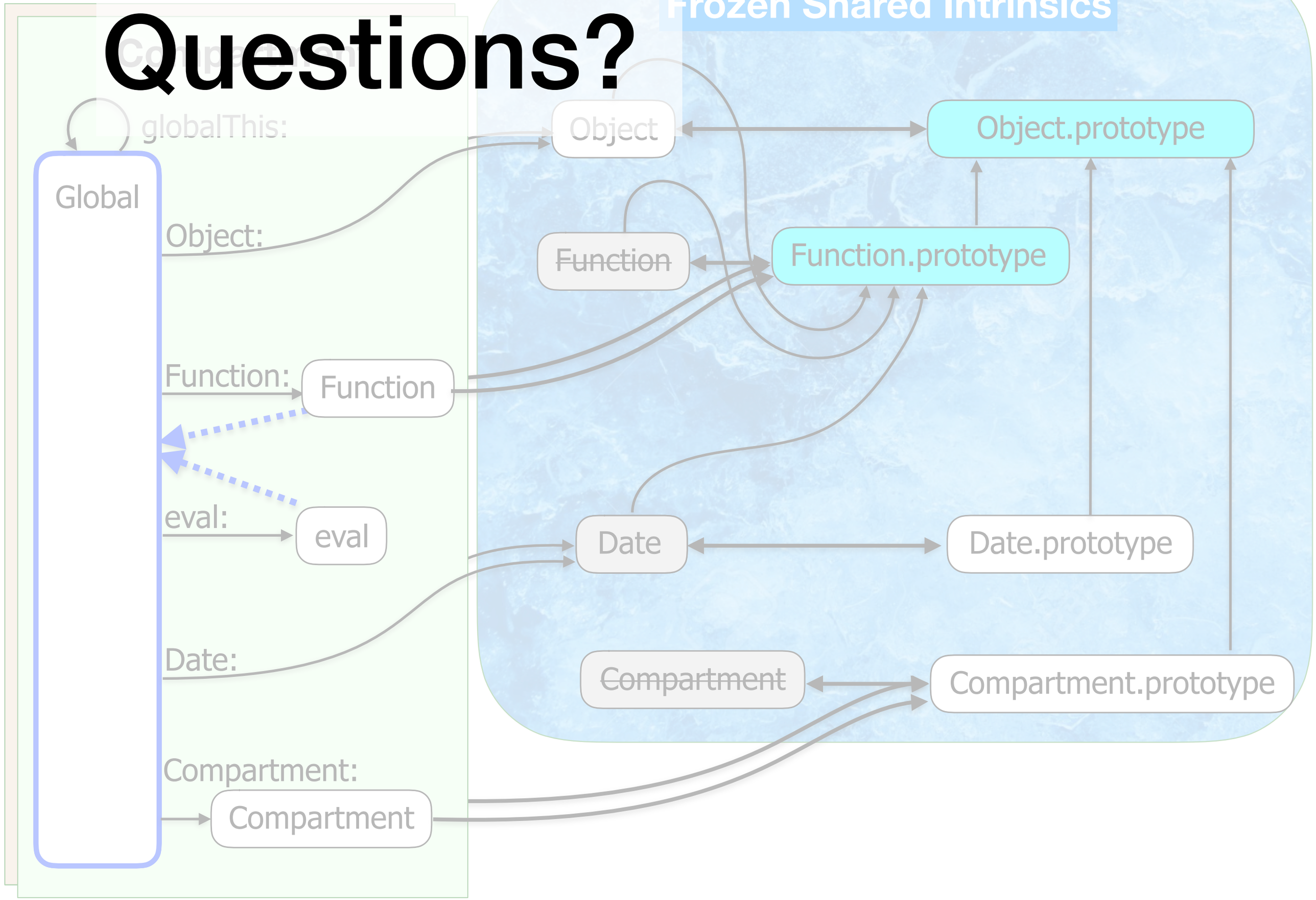
[[TemplateMap]]

[[HostDefined]]

[[ModuleMap]] : name —> (StaticModuleRecord | ModuleInstance)

[[Hooks]] { importNowHook(specifier, referrer) —> StaticModuleRecord, … }

Questions?


Compartment
globalThis:
Global
Object:
Function:
Function
eval:
eval
Date:
Compartment:
Compartment
Frozen Shared Intrinsics
Object
Object.prototype
Function
Function.prototype
Date
Date.prototype
Compartment
Compartment.prototype