Fast and Slim JavaScript

Toon Verwaest

Google

https://v8.dev | @tverwaes

Peak performance!
Load time!
Memory!
Latency!
Security!

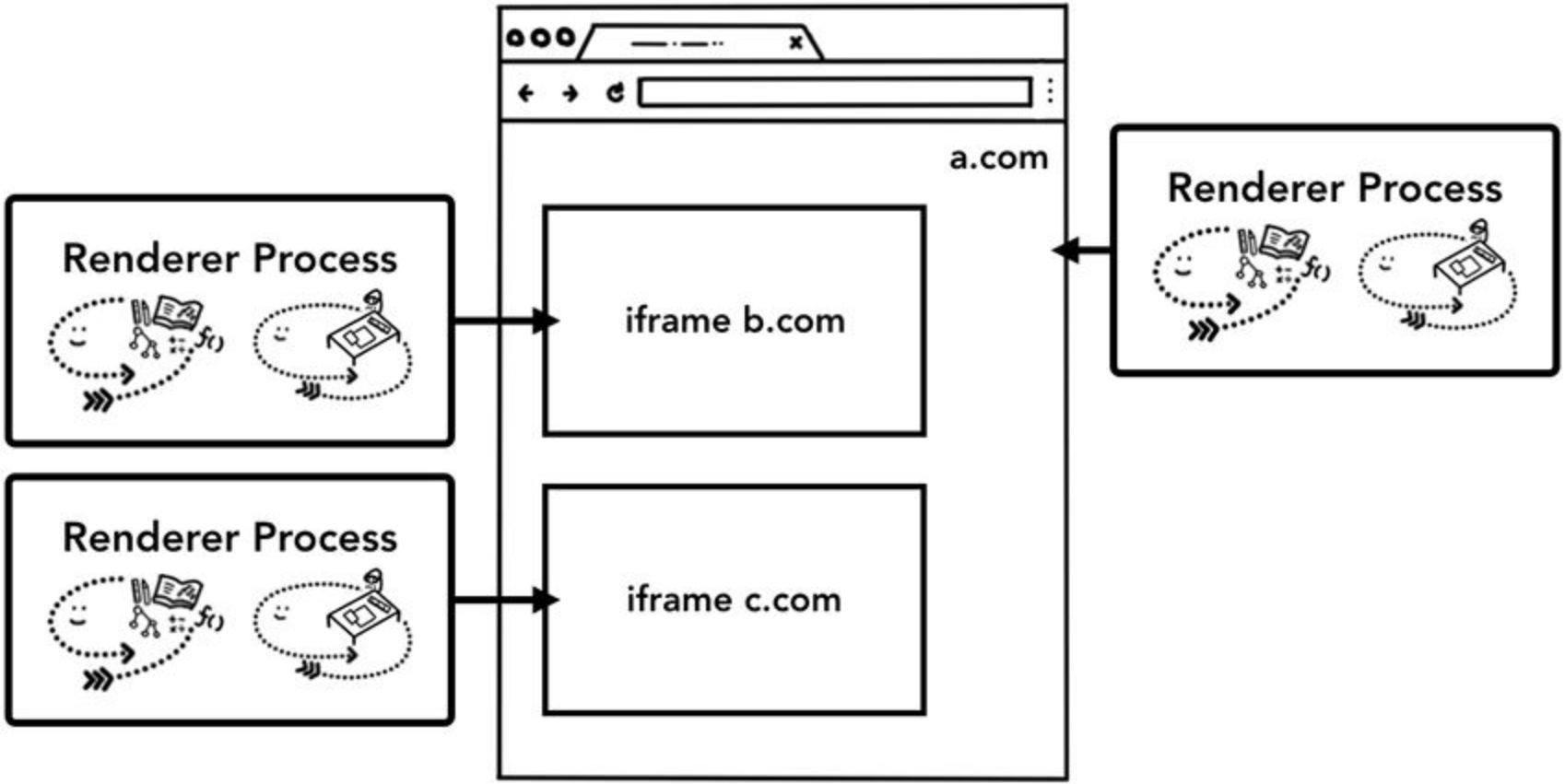
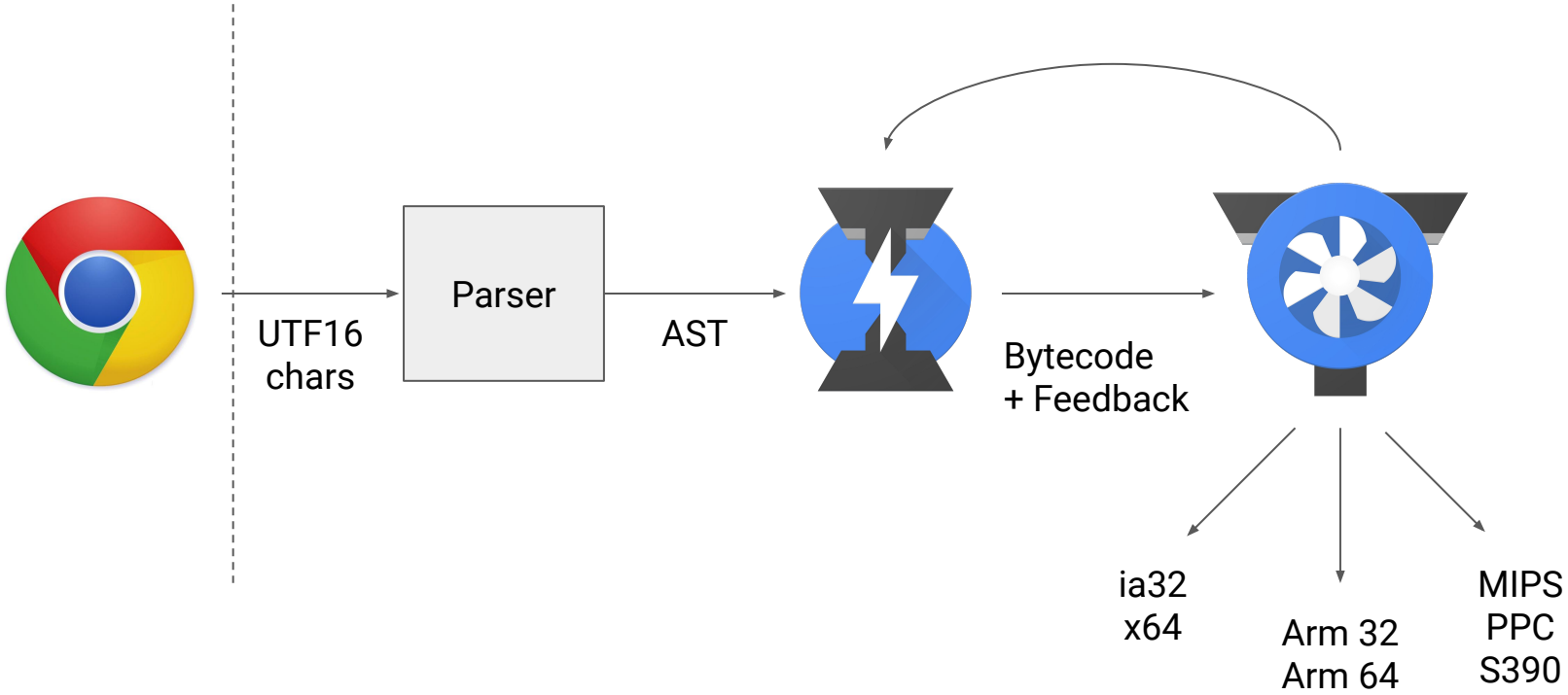
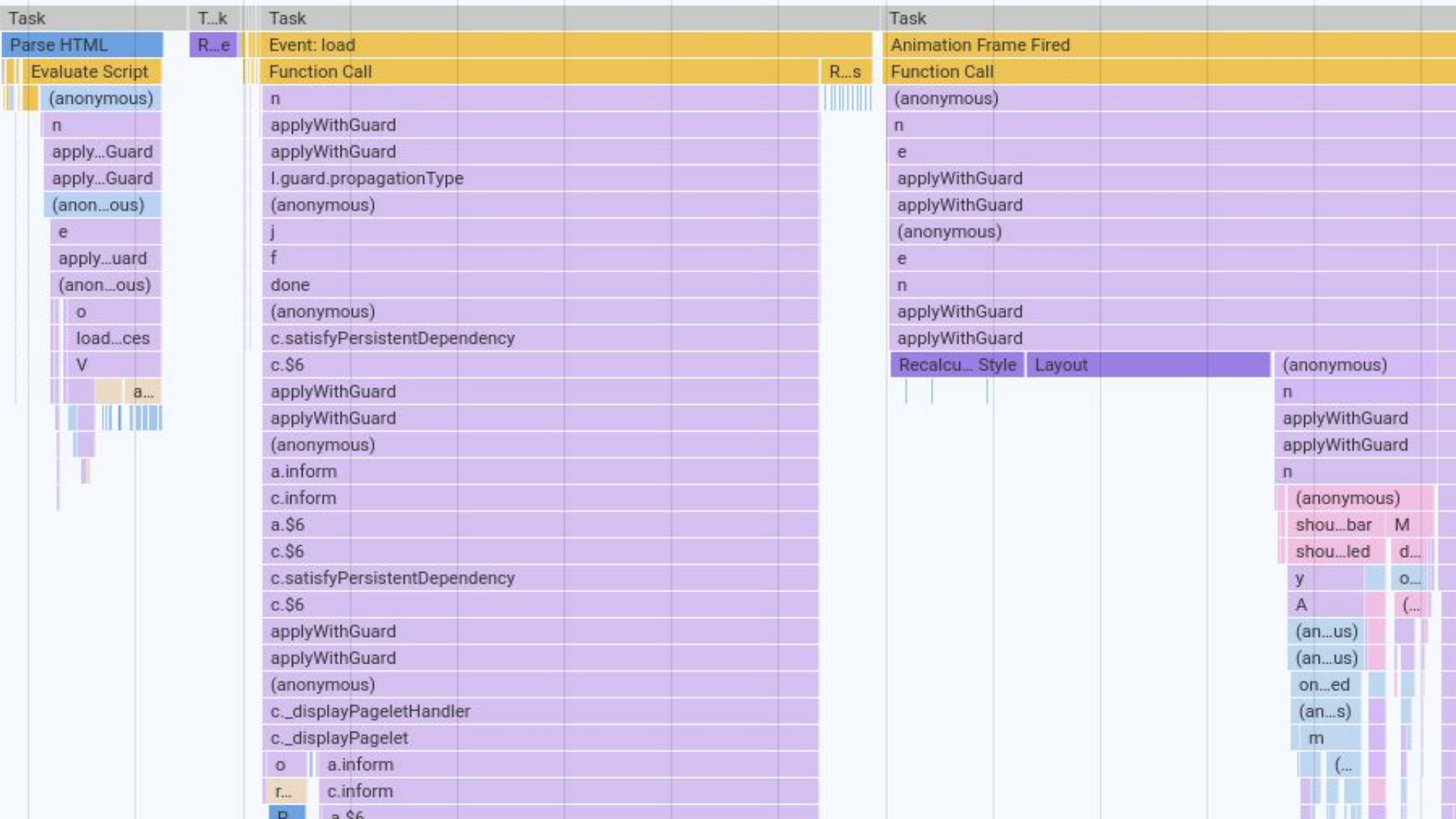


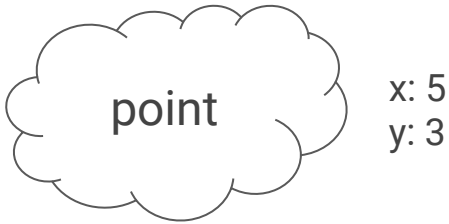
Diagram by Mariko Kosaka


UTF16
chars
Parser
AST
Bytecode
+ Feedback
ia32
x64
Arm 32
Arm 64
MIPS
PPC
S390


Task
T…k
Task
Task
Parse HTML
R…e
Event: load
Animation Frame Fired
Evaluate Script
Function Call
R…s
Function Call
(anonymous)
n
(anonymous)
n
applyWithGuard
n
apply…Guard
applyWithGuard
e
apply…Guard
l.guard.propagationType
applyWithGuard
(anon…ous)
(anonymous)
applyWithGuard
e
j
(anonymous)
apply…uard
f
e
(anon…ous)
done
n
o
(anonymous)
applyWithGuard
load…ces
c.satisfyPersistentDependency
applyWithGuard
V
c.$6
Recalcu… Style
Layout
(anonymous)
a…
applyWithGuard
n
applyWithGuard
applyWithGuard
(anonymous)
applyWithGuard
n
a.inform
(anonymous)
c.inform
shou…bar
M
a.$6
shou…led
d…
c.$6
y
o…
c.satisfyPersistentDependency
A
(…
c.$6
(an…us)
applyWithGuard
(an…us)
applyWithGuard
on…ed
(anonymous)
(an…s)
c._displayPageletHandler
m
c._displayPagelet
(…
o
a.inform
r…
c.inform

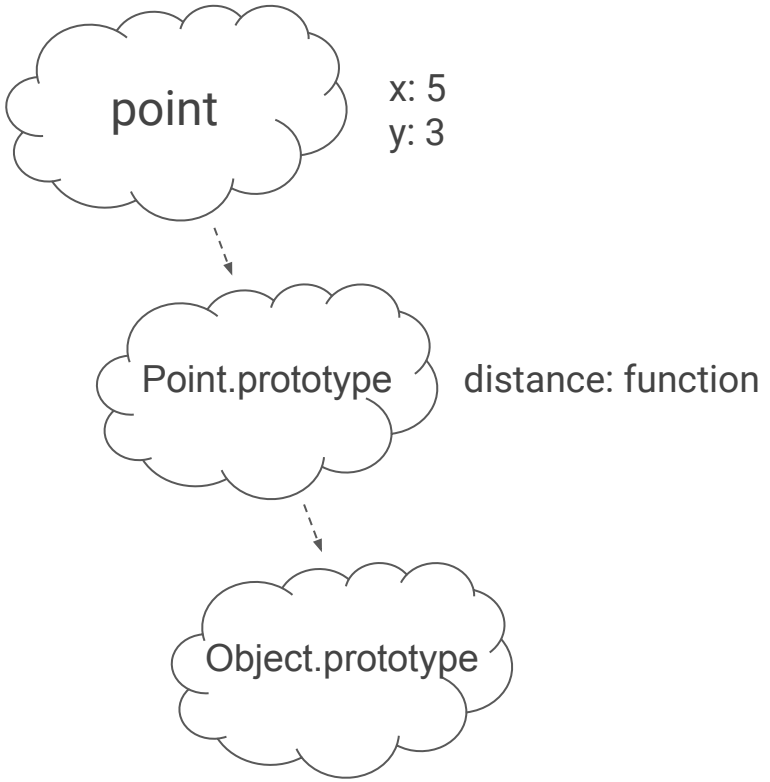
Objects are dictionaries...

```
function Point(x, y) {
  this.x = x;
  this[“y”] = y;
}

const point = new Point(5, 3);
```



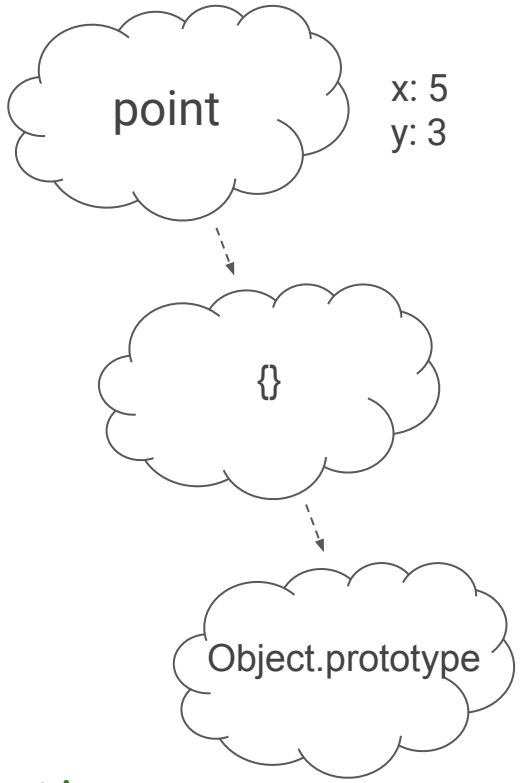
...with prototypes...

```
function Point(x, y) {
  this.x = x;
  this[“y”] = y;
}

Point.prototype.distance = function(other)  {
  return Math.sqrt((this.x - other.x)**2 +
                   (this.y - other.y)**2);
}

const point = new Point(5, 3);
point.distance({x:9, y:6});   // Returns: 5
```



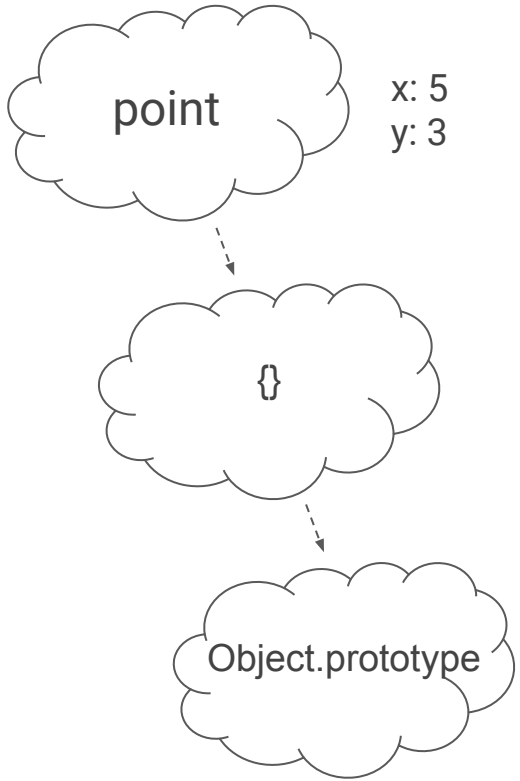
...that can change...

```
function Point(x, y) {
  this.x = x;
  this[“y”] = y;
}

Point.prototype.distance = function(other)  {
  return Math.sqrt((this.x - other.x)**2 +
                   (this.y - other.y)**2);
}

const point = new Point(5, 3);
point.__proto__ = {};
point.distance({x:9, y:6});
// Throws: TypeError: point.distance is not a function
```



...even if they are Classes

```
class Point {
  constructor(x, y) {
    this.x = x;
    this.y = y;
  }

  distance (other)  {
    return Math.sqrt((this.x - other.x)**2 +
                     (this.y - other.y)**2);
  }
};
const point = new Point(5, 3);
point.__proto__ = {};
point.distance({x:9, y:6});
// Throws: TypeError: point.distance is not a function
```

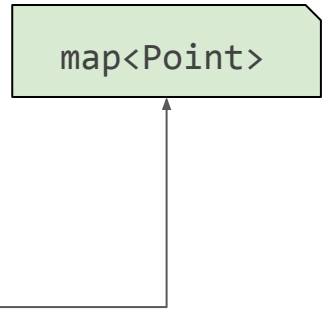


Hidden Classes

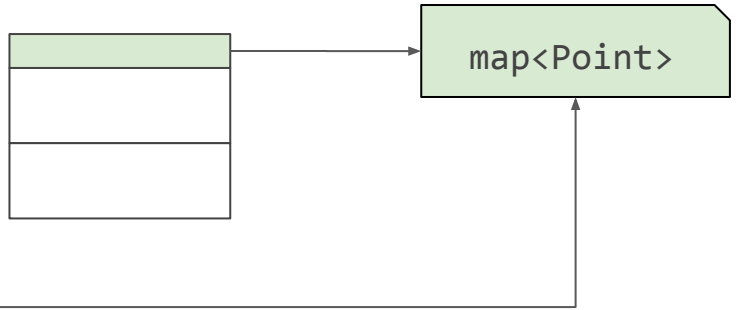
```
function Point(x, y) {
  this.x = x;
  this.y = y;
}
```

Hidden Classes

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}
```



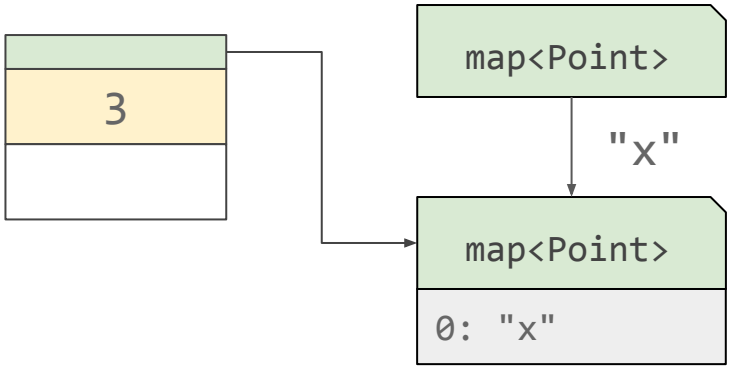
Hidden Classes




```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
```

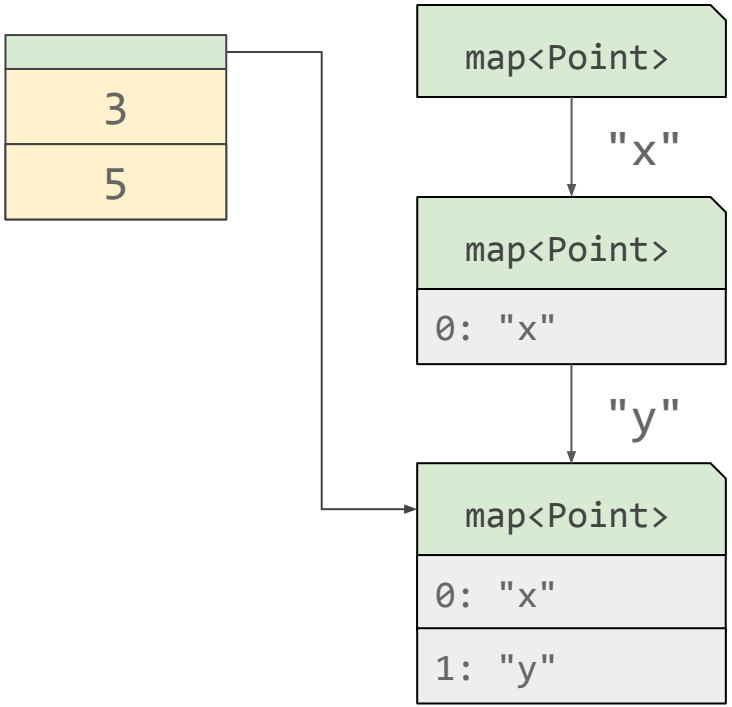
Hidden Classes

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
```



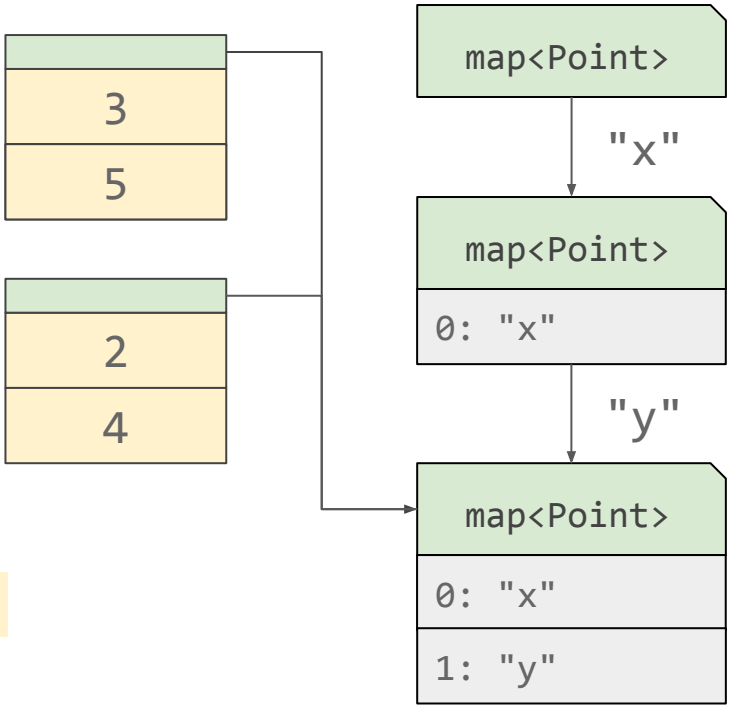
Hidden Classes

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
```



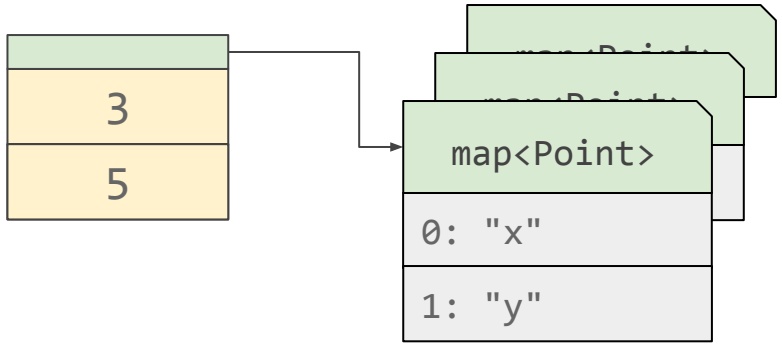
Hidden Classes

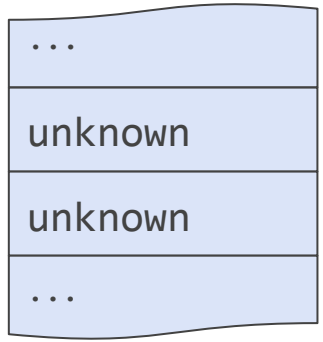
```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
const p2 = new Point(2, 4);
```



Type Feedback

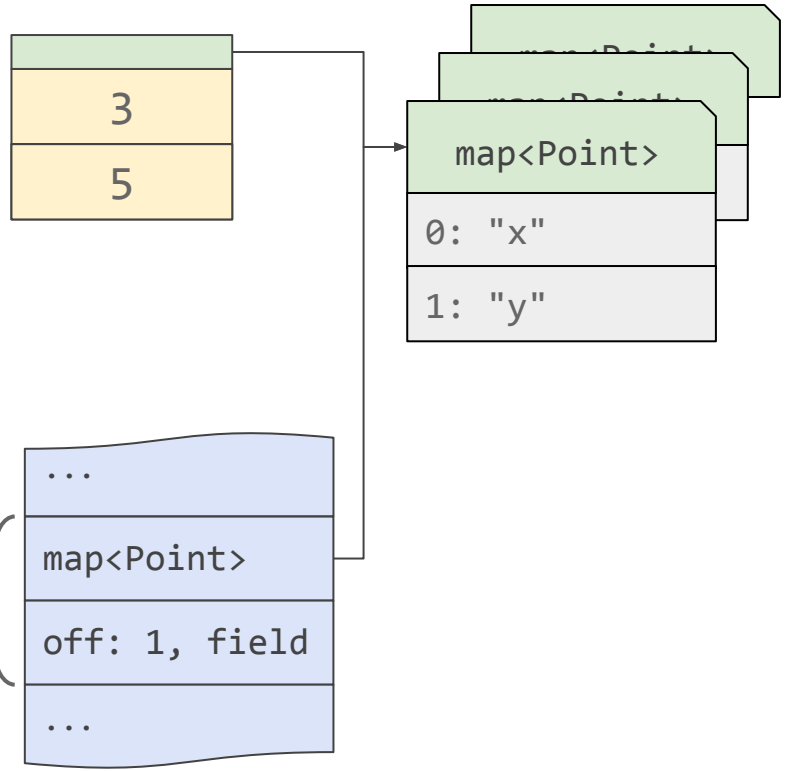
```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.y;
```





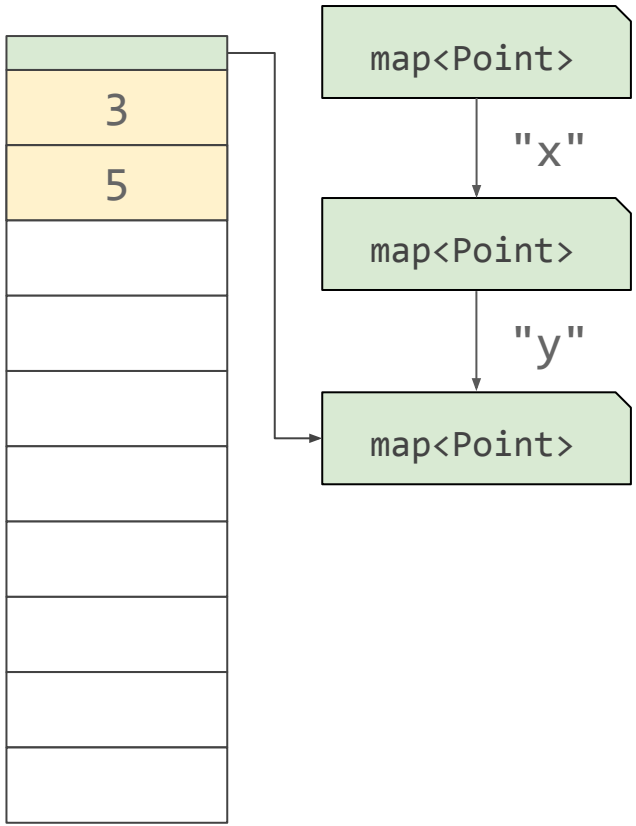
Type Feedback

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.y;
```



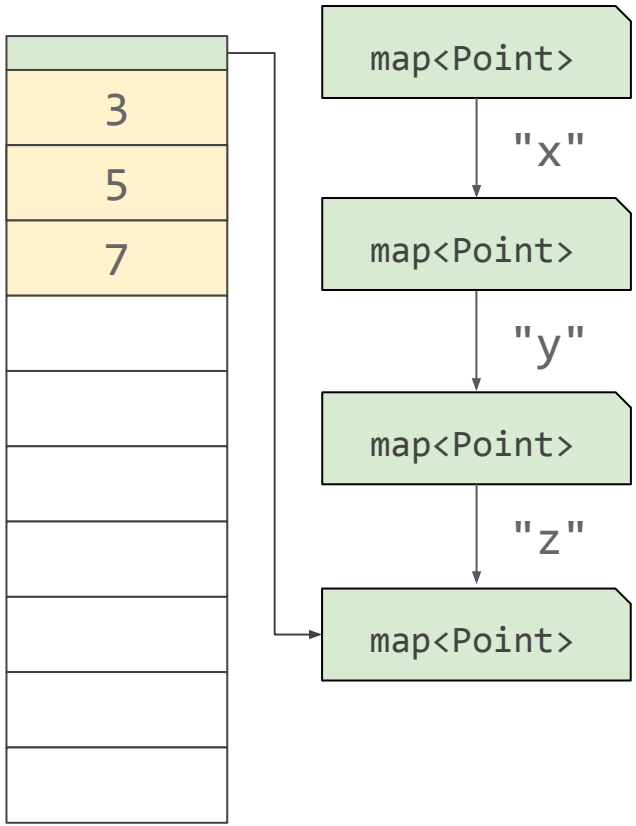
In-object Slack Tracking

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
```



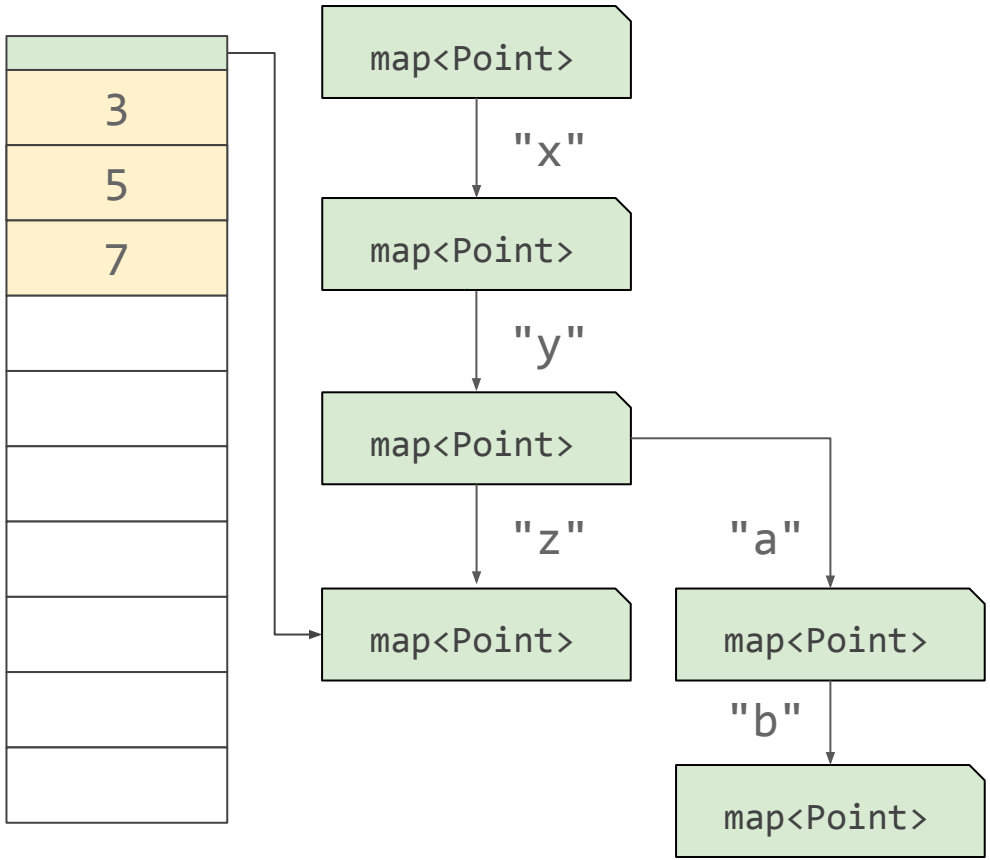
In-object Slack Tracking

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;
```



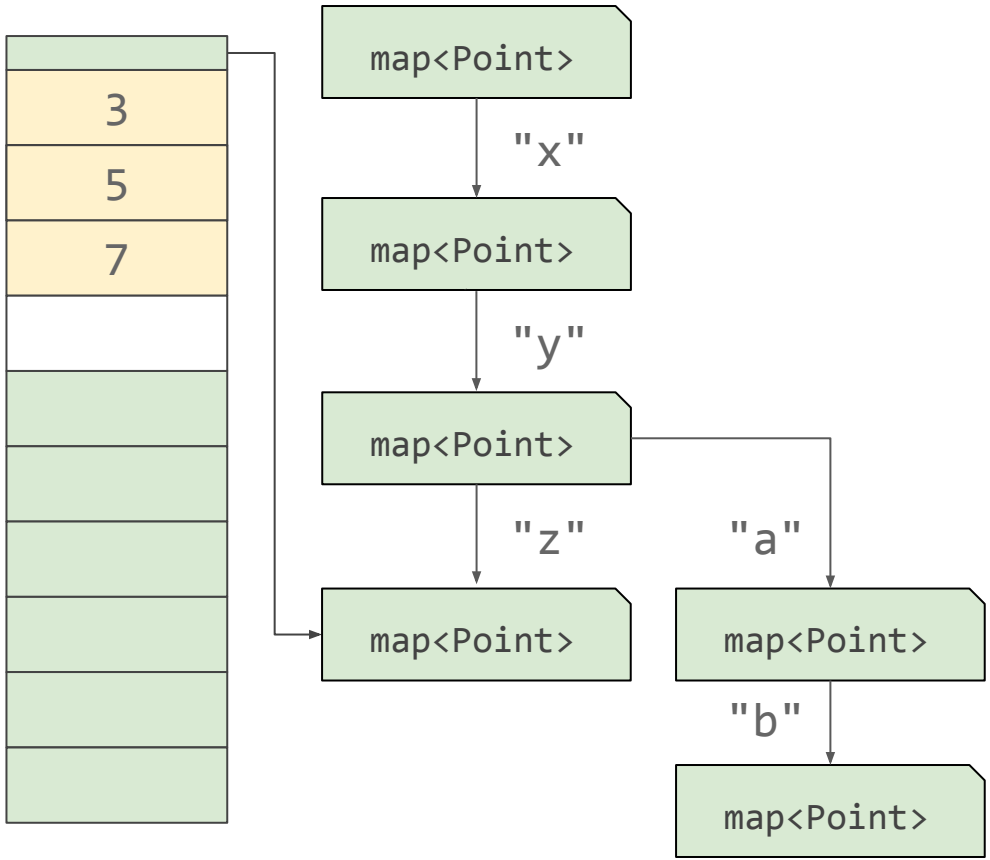
In-object Slack Tracking

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;

const p2 = new Point(4, 2);
p2.a = 9;
p2.b = 15;
```



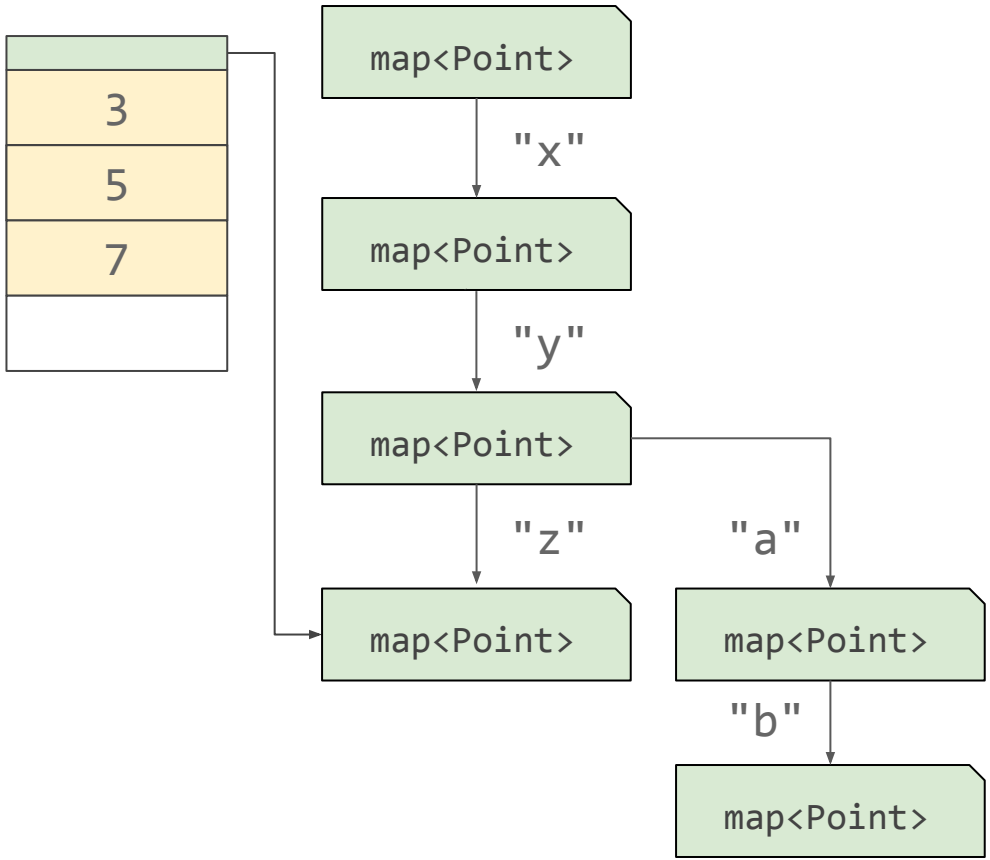
In-object Slack Tracking

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;

const p2 = new Point(4, 2);
p2.a = 9;
p2.b = 15;
```



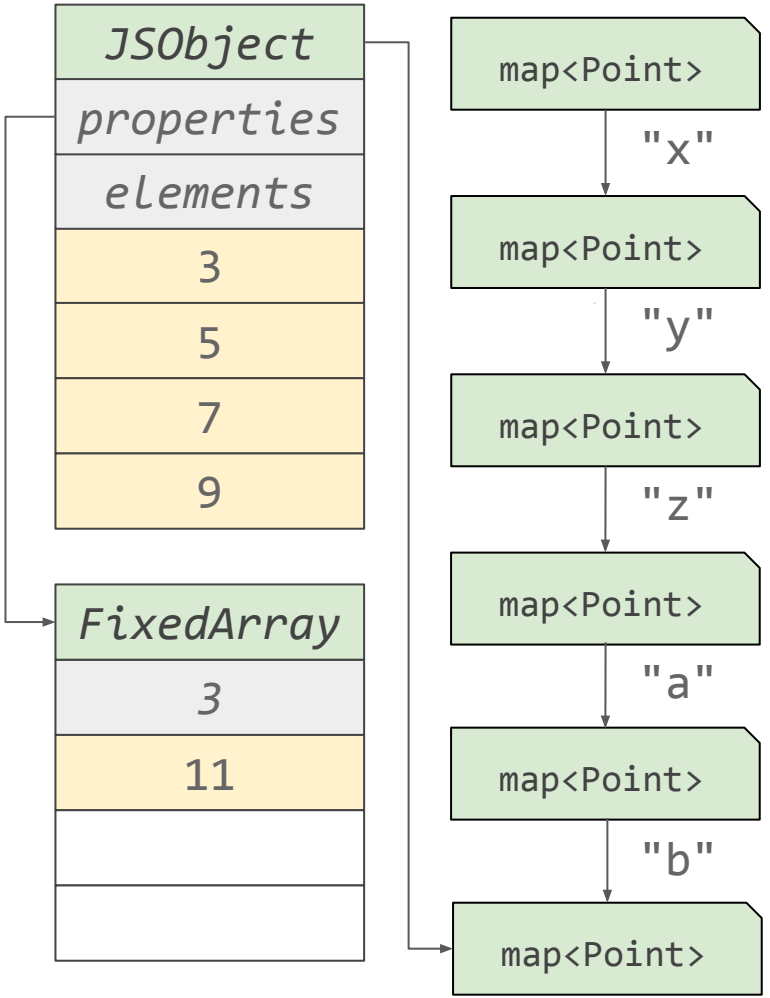
In-object Slack Tracking

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;

const p2 = new Point(4, 2);
p2.a = 9;
p2.b = 15;
```



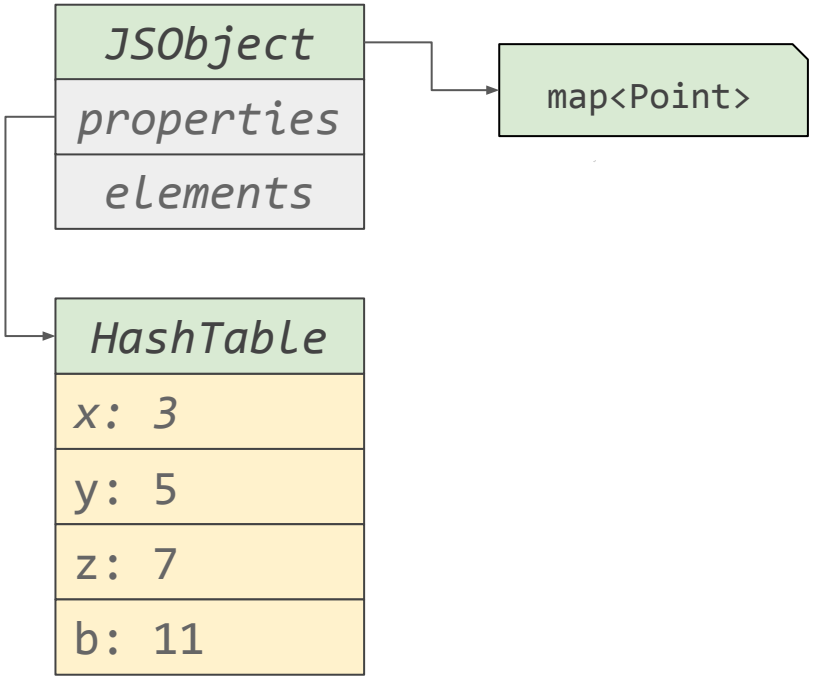
Properties Backing store

```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;
p.a = 9;
p.b = 11;
```

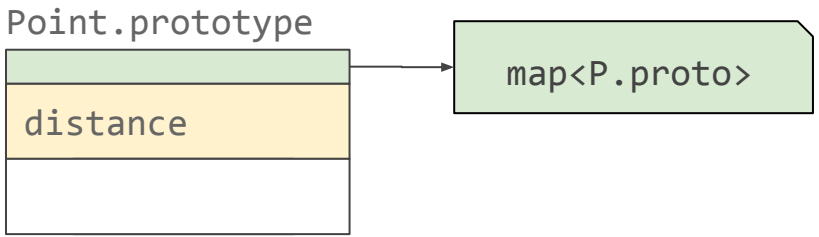


Properties Backing store

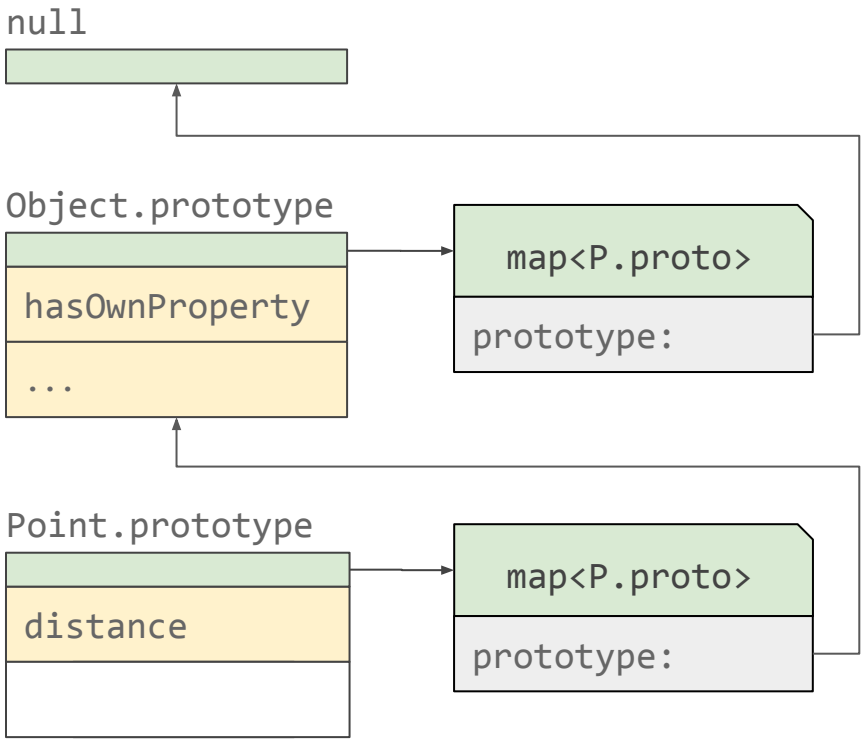
```
function Point(x, y) {
  this.x = x;
  this.y = y;
}

const p = new Point(3, 5);
p.z = 7;
p.a = 9;
p.b = 11;
delete p.a;
```



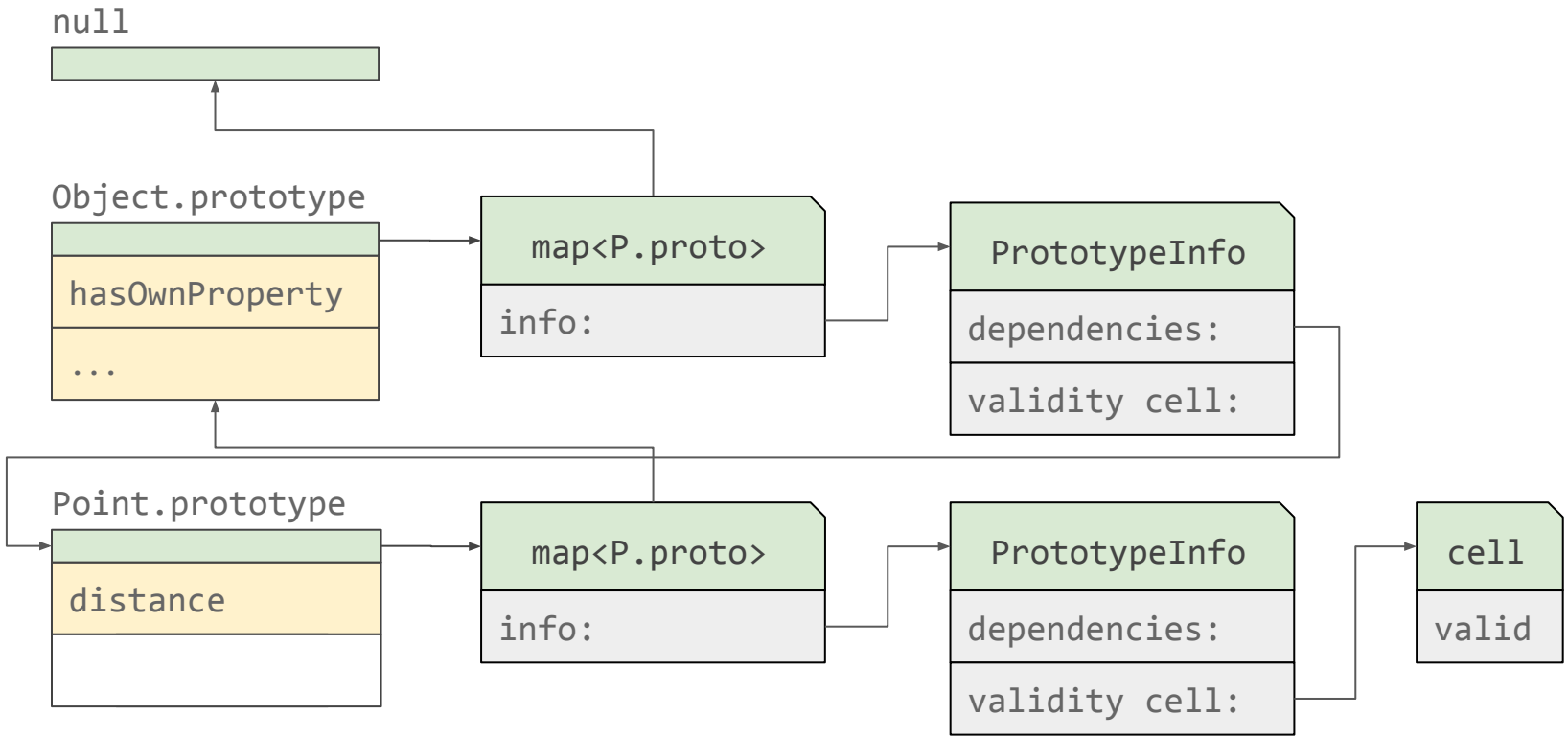
Prototypes



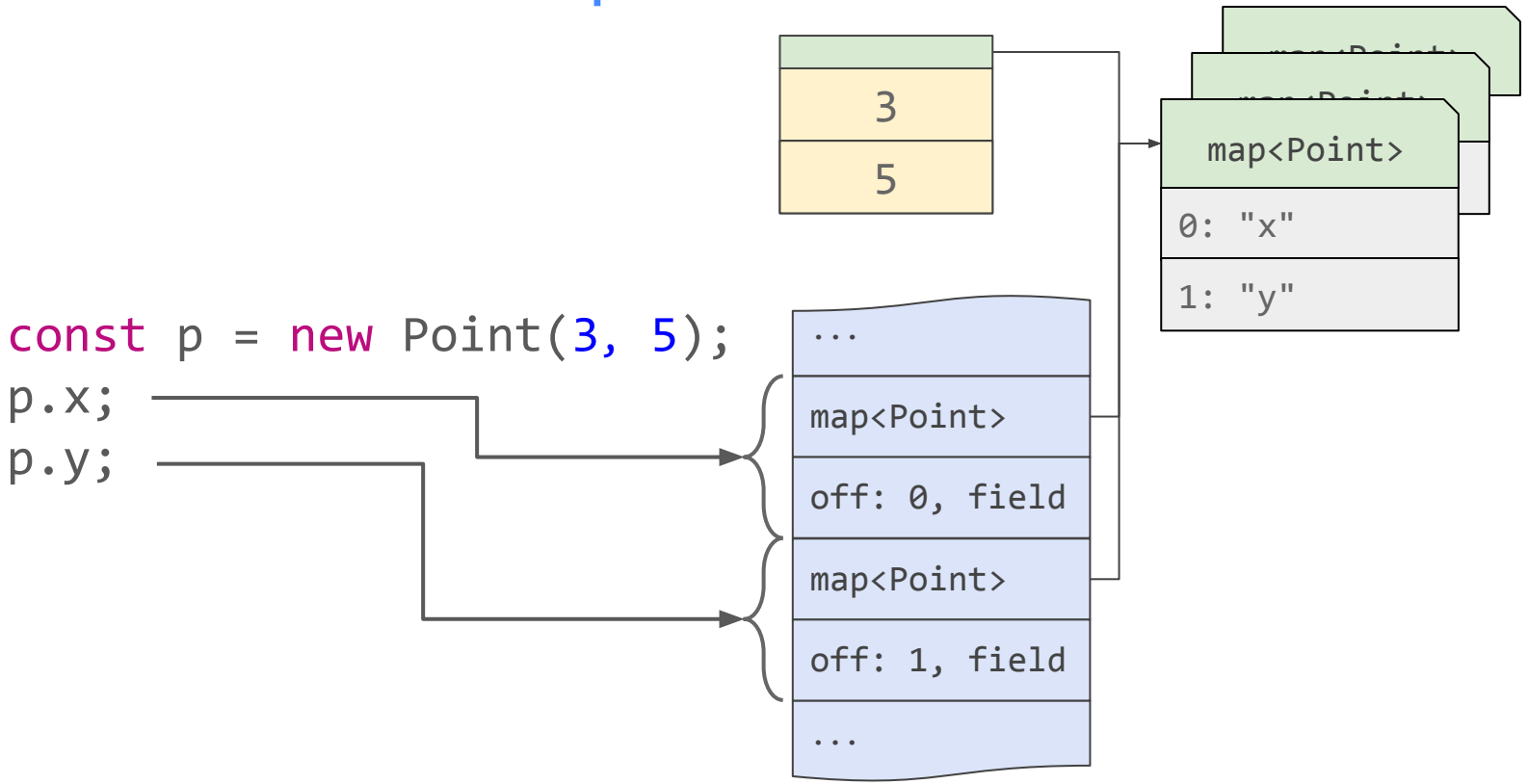
Prototypes


null
Object.prototype
hasOwnProperty
...
map<P.proto>
prototype:
Point.prototype
distance
map<P.proto>
prototype:

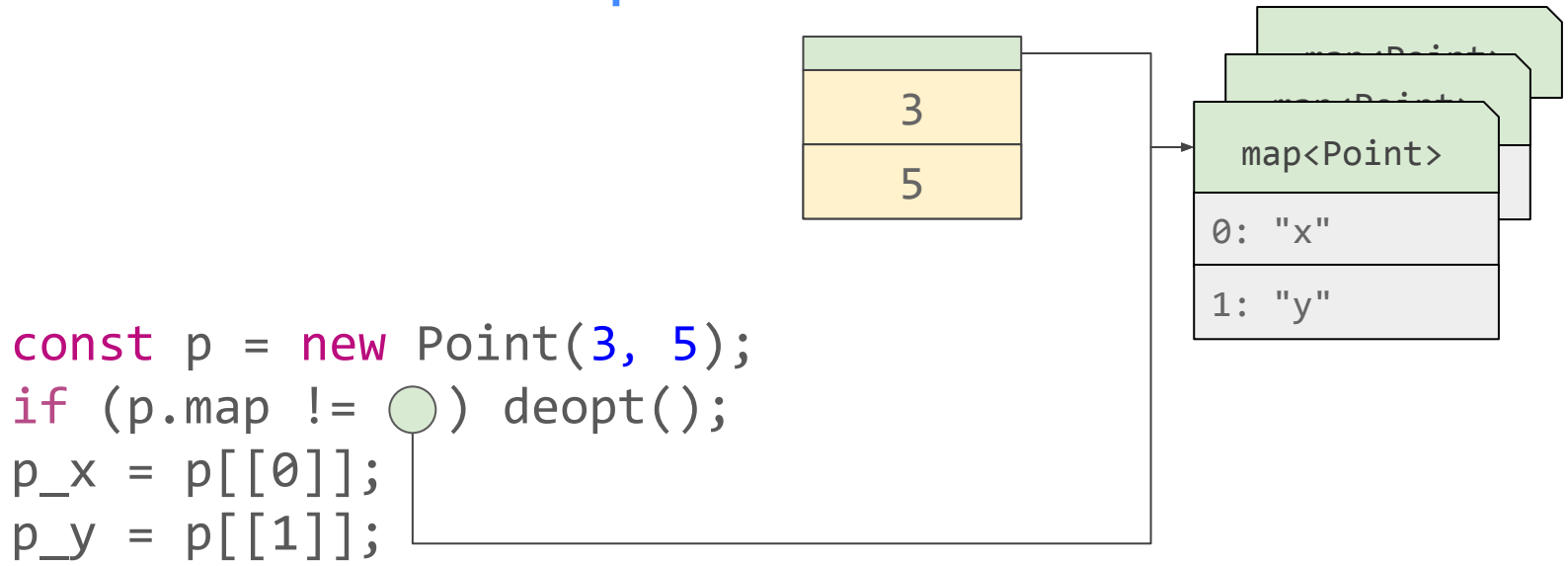
Prototypes


null
Object.prototype
hasOwnProperty
...
map<P.proto>
info:
PrototypeInfo
dependencies:
validity cell:
Point.prototype
distance
map<P.proto>
info:
PrototypeInfo
dependencies:
validity cell:
cell
valid

Feedback-Based Optimization



Feedback-Based Optimization

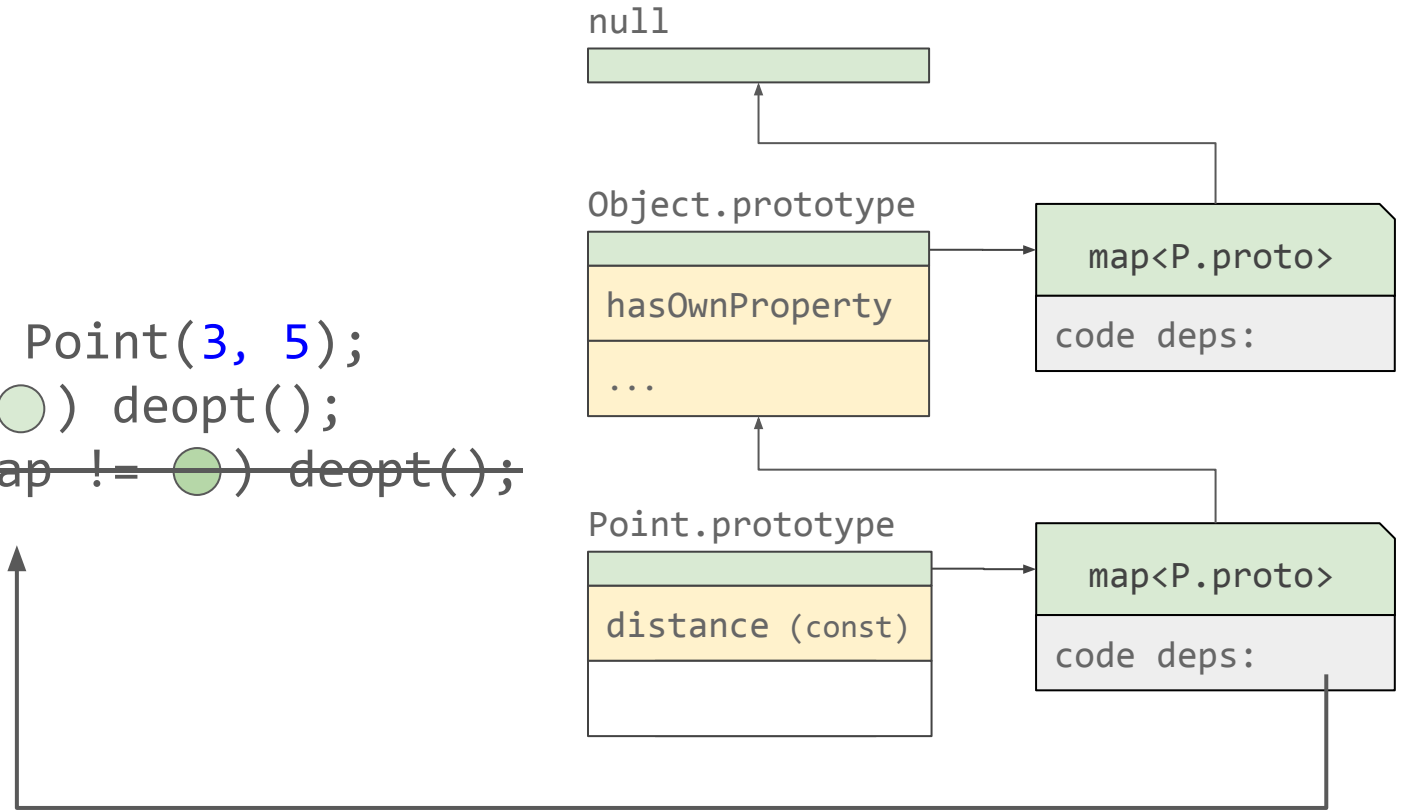


Code Dependencies

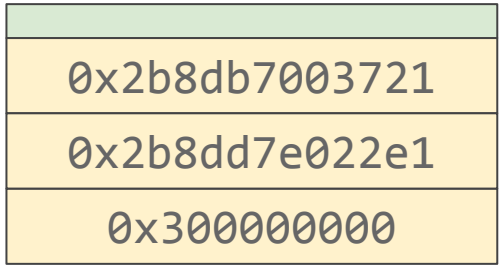
```
const p = new Point(3, 5);
if (p.map != ◯) deopt();
if (p_proto.map != ◯) deopt();
```

Code Dependencies

```
const p = new Point(3, 5);
if (p.map != ◯) deopt();
if (p_proto.map != ◯) deopt();
```



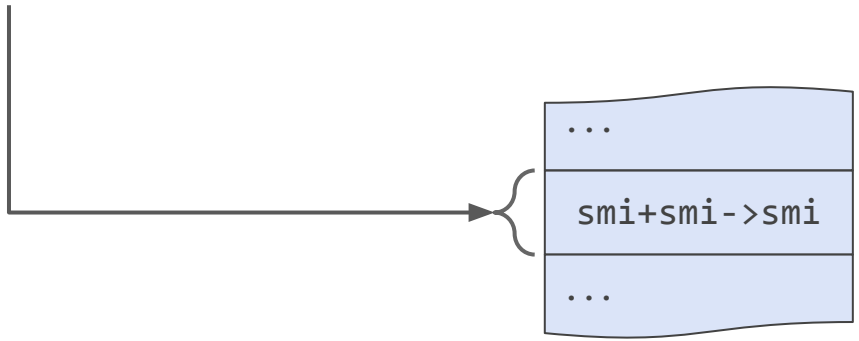
Tagged Pointers




```
0b...01: Heap object pointer + 0b01
0b...11: Weak object pointer + 0b11
0b....0: Small Integer << 32
         (<< 1 on 32bit)
```

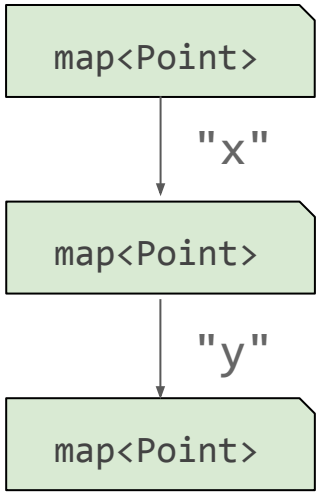
Field Representation

```
const p = new Point(3, 5);
p.x = p.x + p.y
```



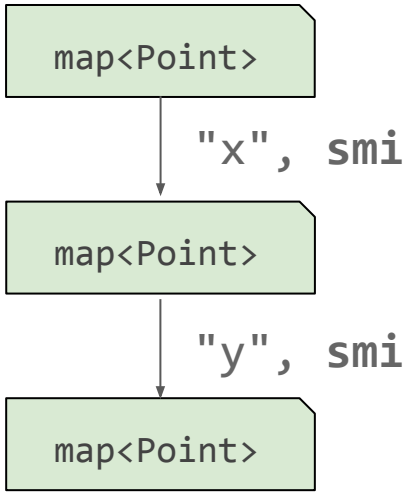
Field Representation

```
const p = new Point(3, 5);
if (p.map != ◯) deopt();

p_x = p[[0]];
p_y = p[[1]];

if (!IsSmi(p_x)) deopt();
if (!IsSmi(p_y)) deopt();

p[[0]] =
  SmiTag(SmiUntag(p_x) +
         SmiUntag(p_y));
```

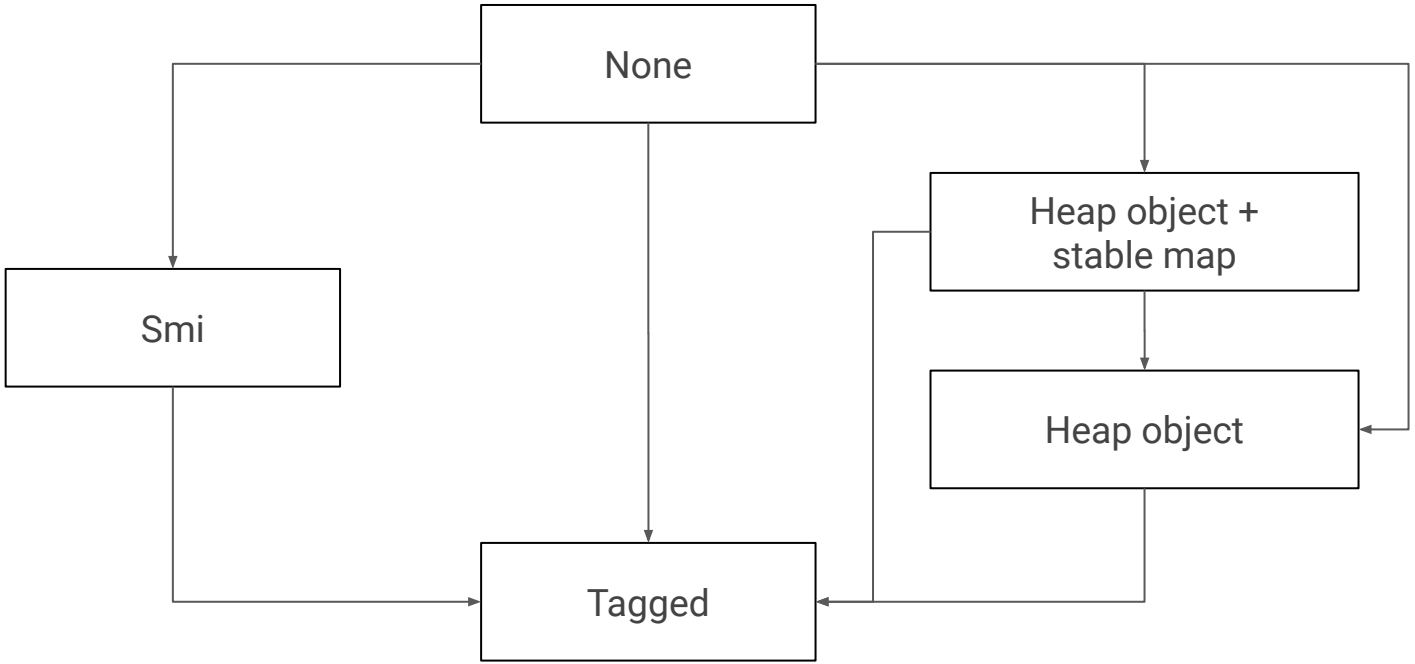


Field Representation

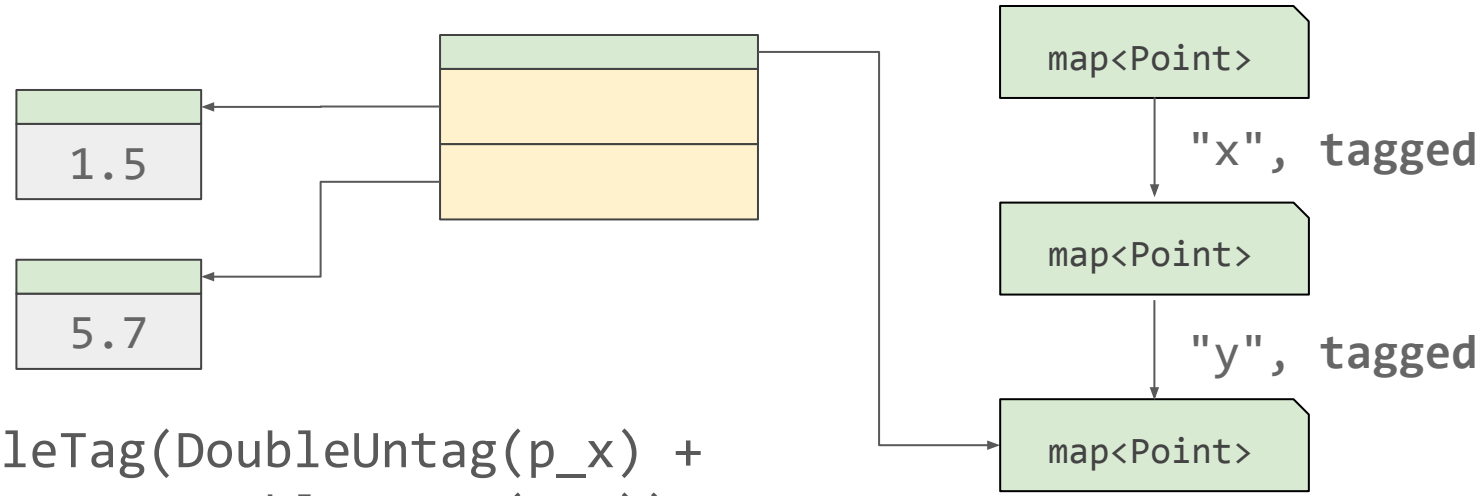
```
const p = new Point(3, 5);
if (p.map != ◯) deopt();

p_x = p[[0]];
p_y = p[[1]];

~~if (!IsSmi(p_x)) deopt();~~
~~if (!IsSmi(p_y)) deopt();~~

p[[0]] =
  SmiTag(SmiUntag(p_x) +
         SmiUntag(p_y));
```



Field Representation



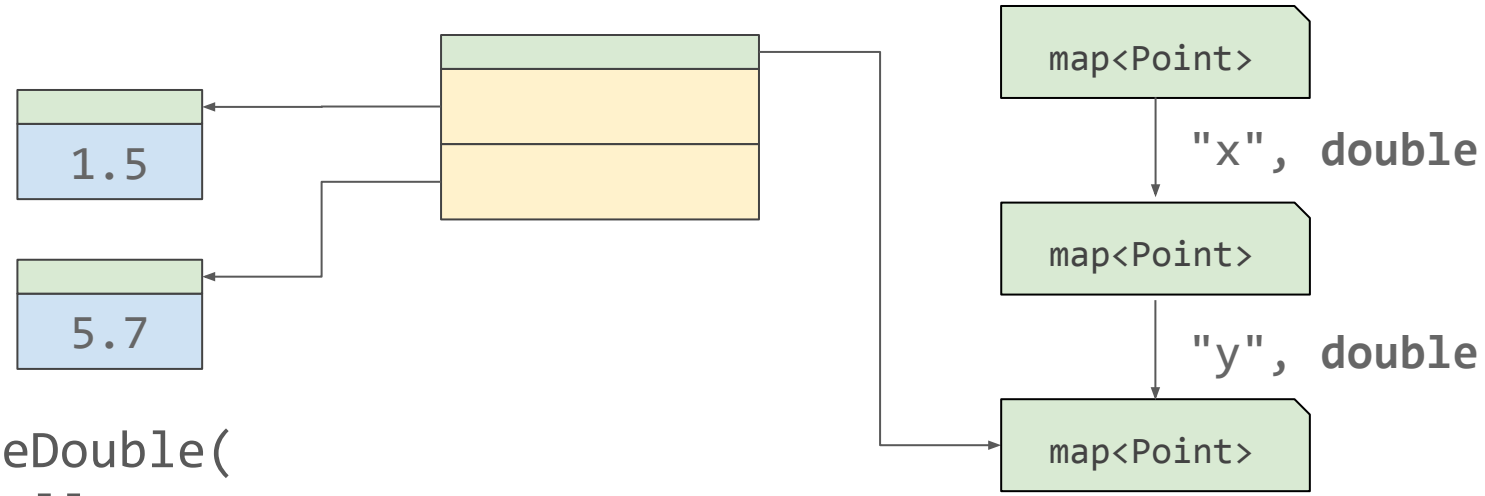
Doubles




```
DoubleTag(DoubleUntag(p_x) +
          DoubleUntag(p_y));
```

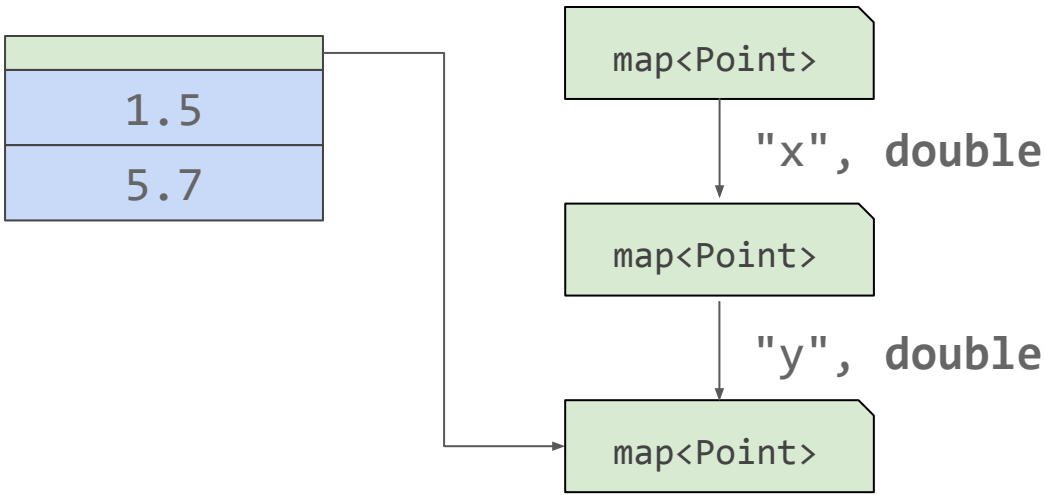
Allocation!

Mutable Double Boxes



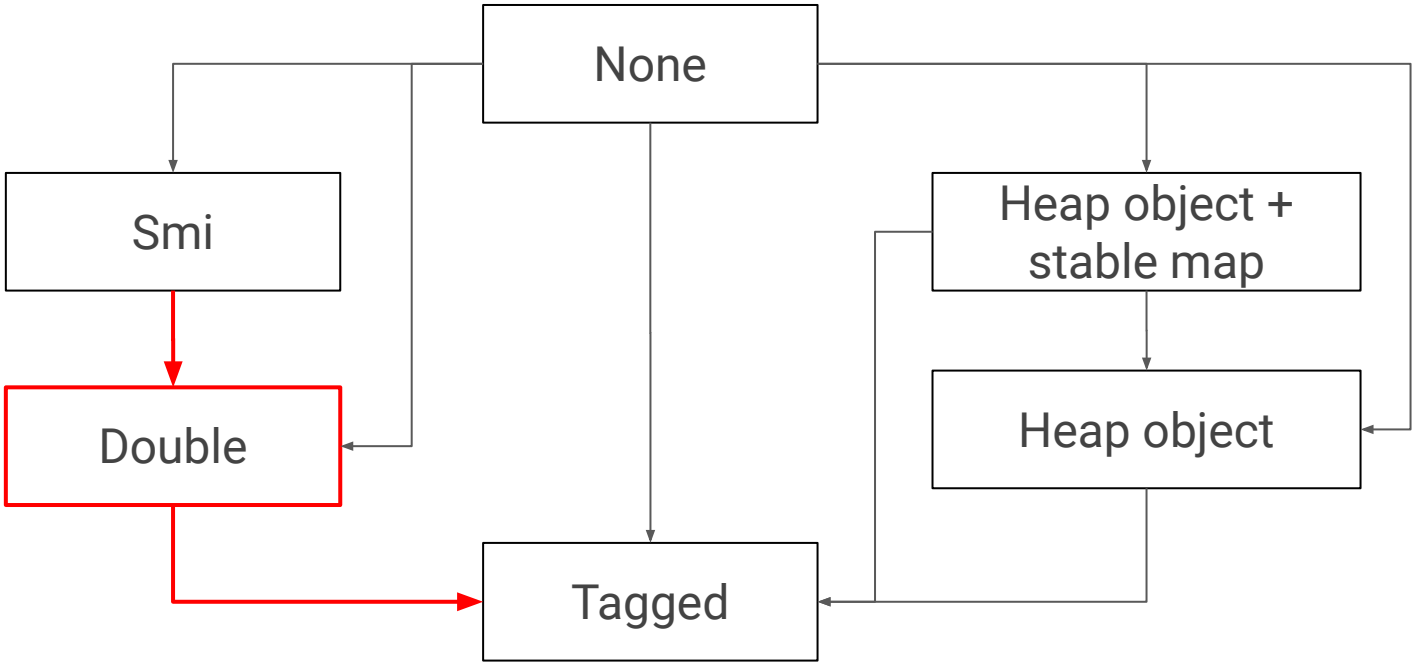

```
WriteDouble(
 p[[0]],
 ReadDouble(p[[0]]) +
 ReadDouble(p[[1]]));
```

Unboxed Double Fields

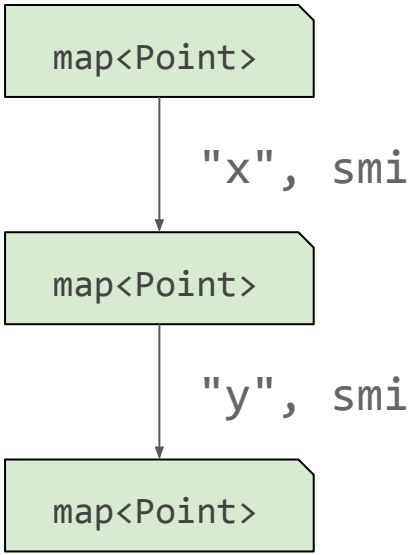



```
WriteDouble(
 p[[0]],
 ReadDouble(p[[0]]) +
 ReadDouble(p[[1]]));
```

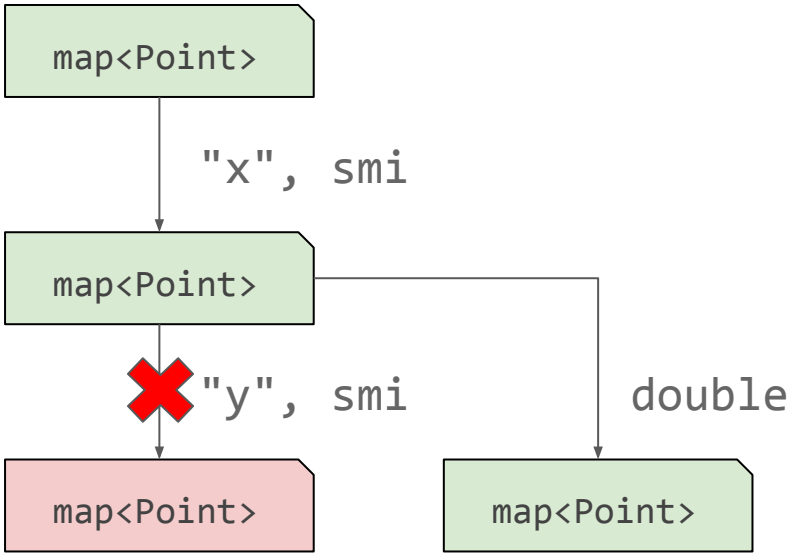
Field Representation



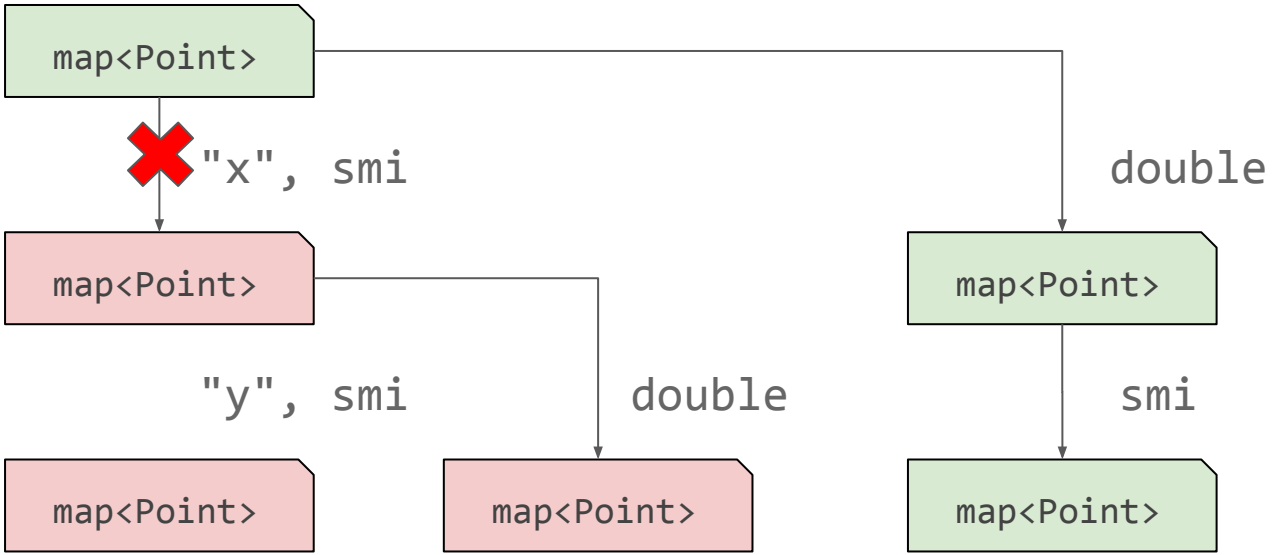
Mutable and Unboxed Double Fields



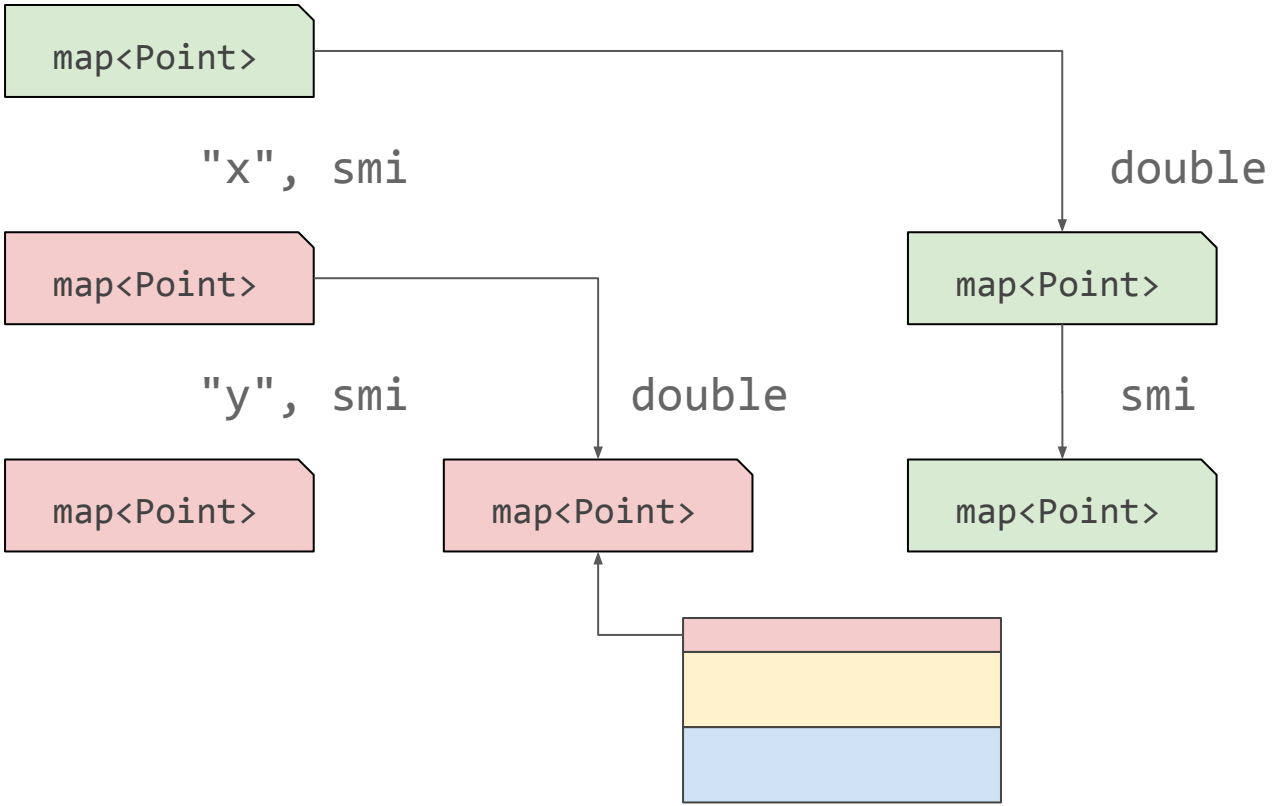
Mutable and Unboxed Double Fields



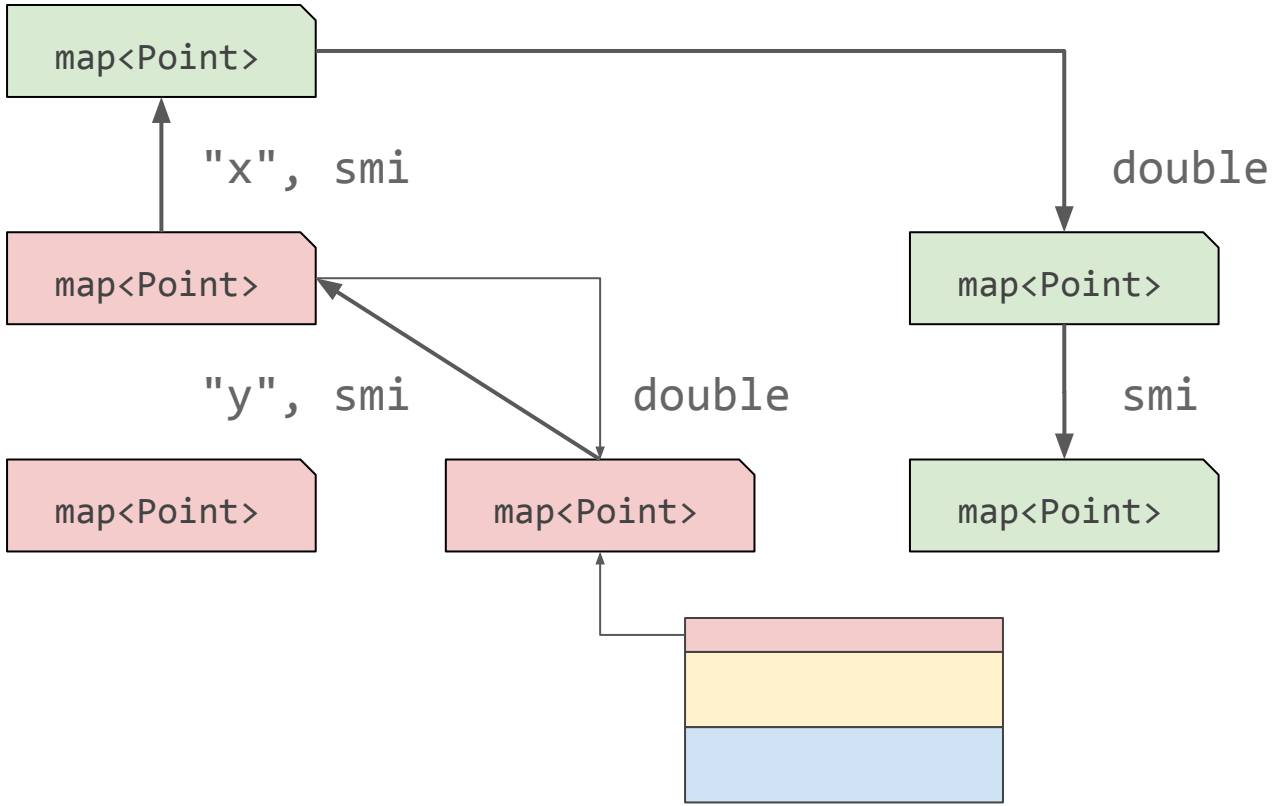
Mutable and Unboxed Double Fields



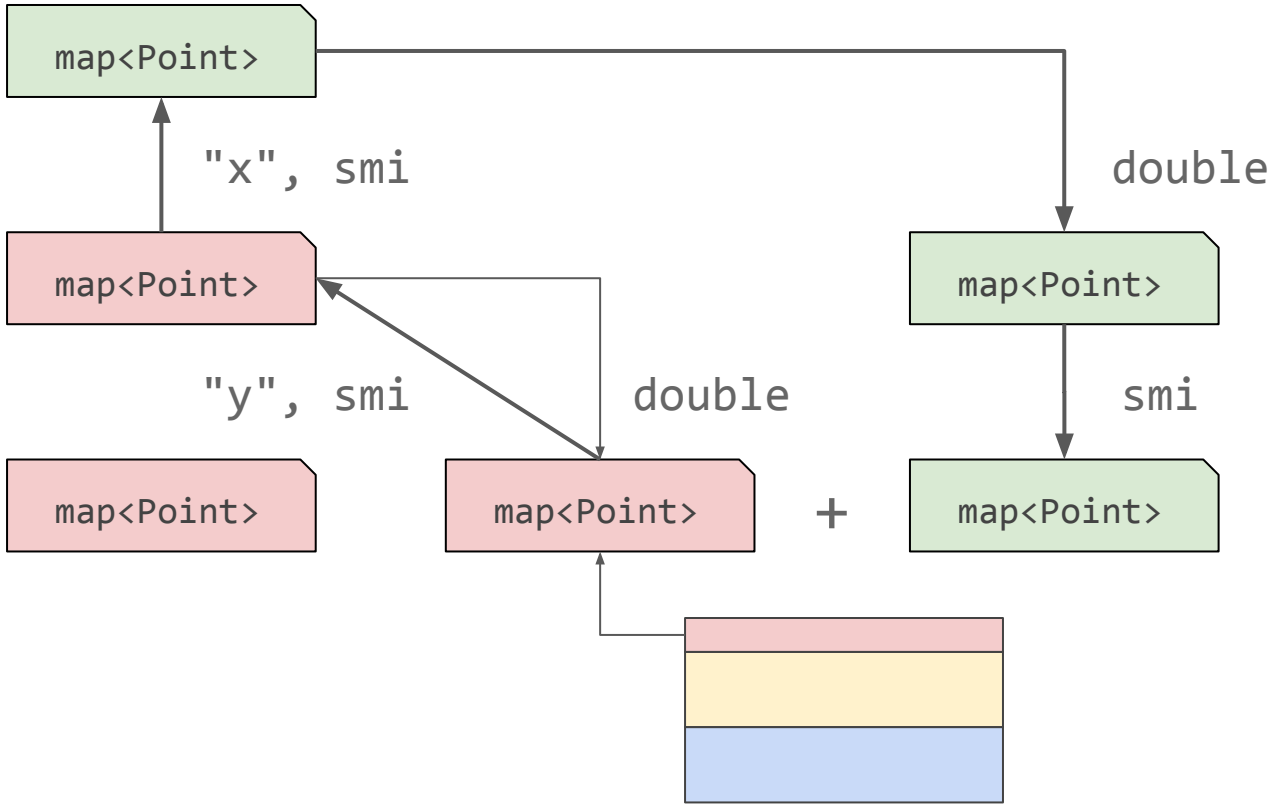
Mutable and Unboxed Double Fields



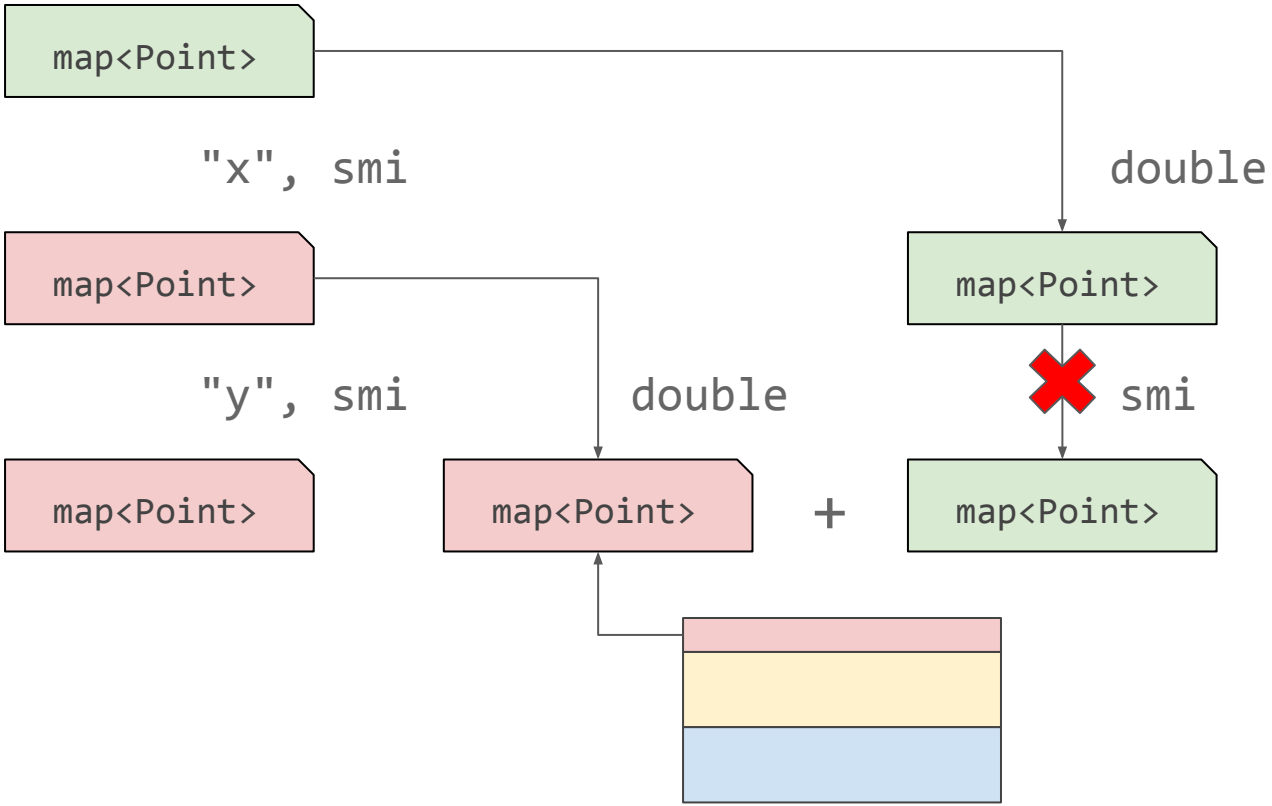
Mutable and Unboxed Double Fields



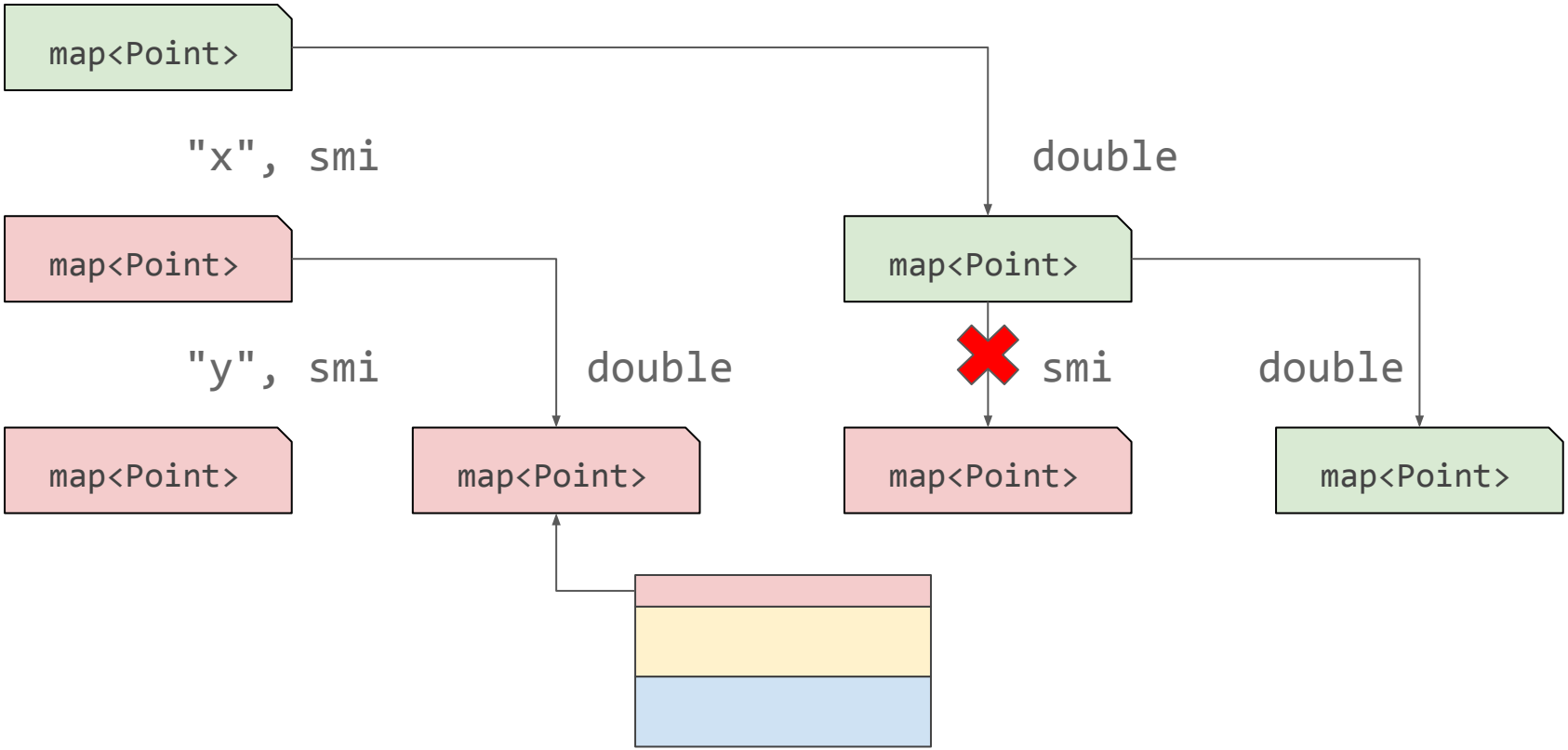
Mutable and Unboxed Double Fields



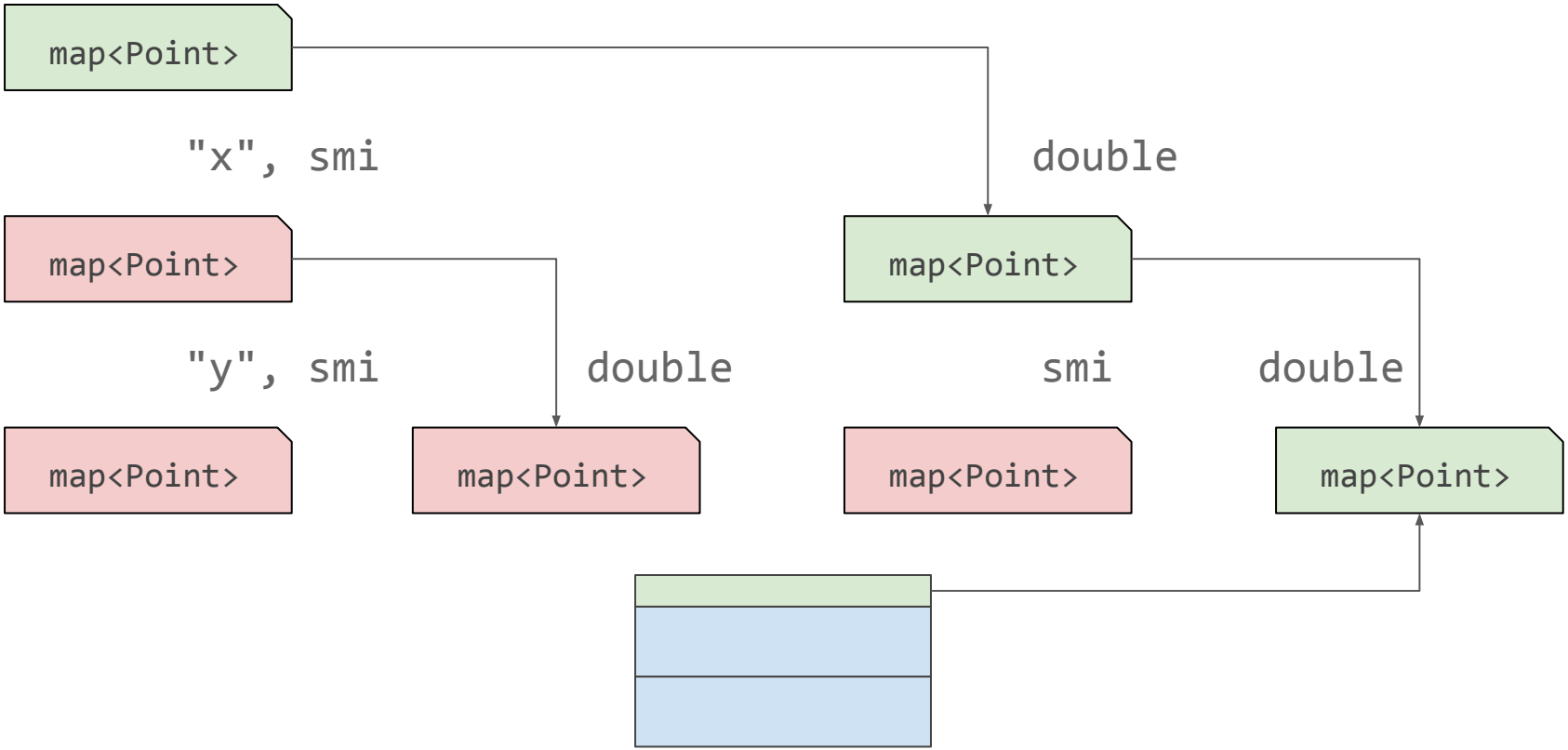
Mutable and Unboxed Double Fields

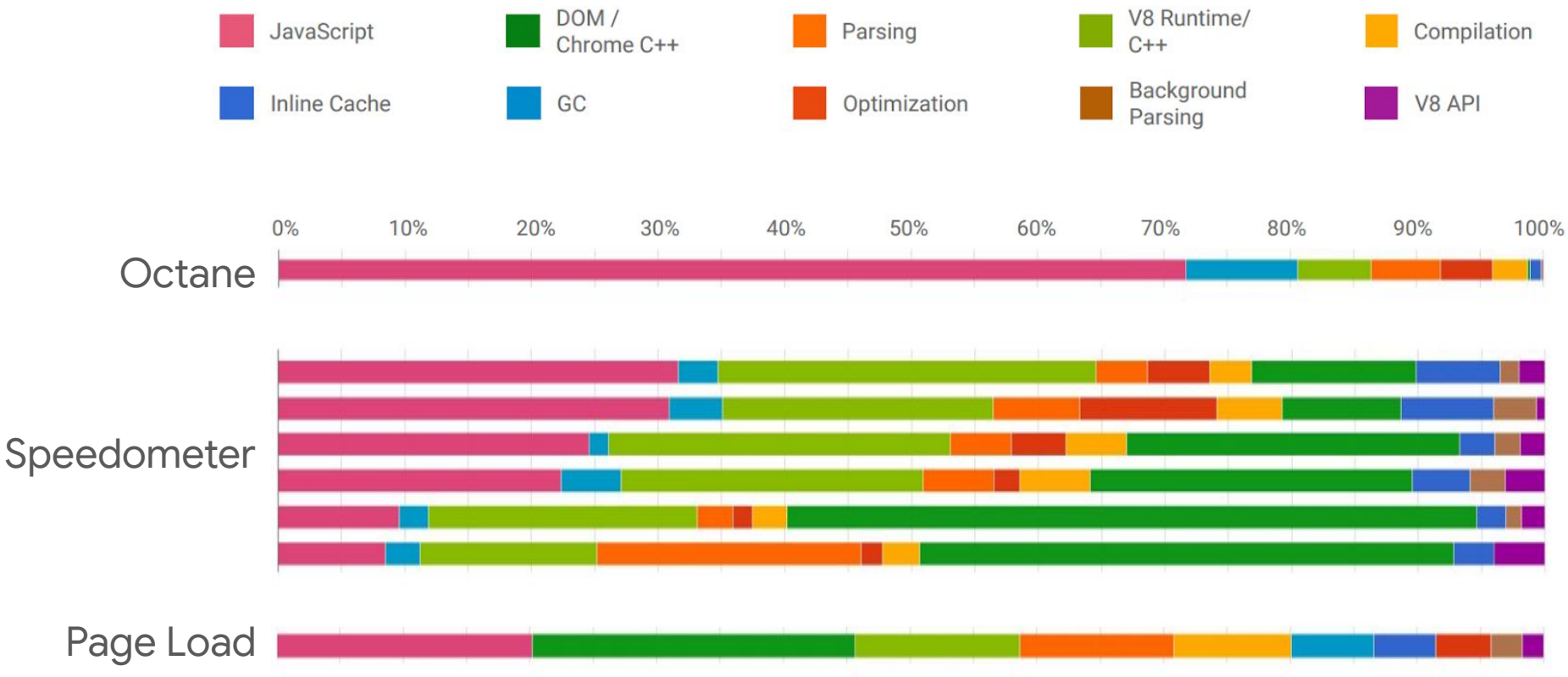


Mutable and Unboxed Double Fields



Mutable and Unboxed Double Fields




JavaScript
DOM / Chrome C++
Parsing
V8 Runtime/ C++
Compilation
Inline Cache
GC
Optimization
Background Parsing
V8 API
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Octane
Speedometer
Page Load

Lazy Compilation

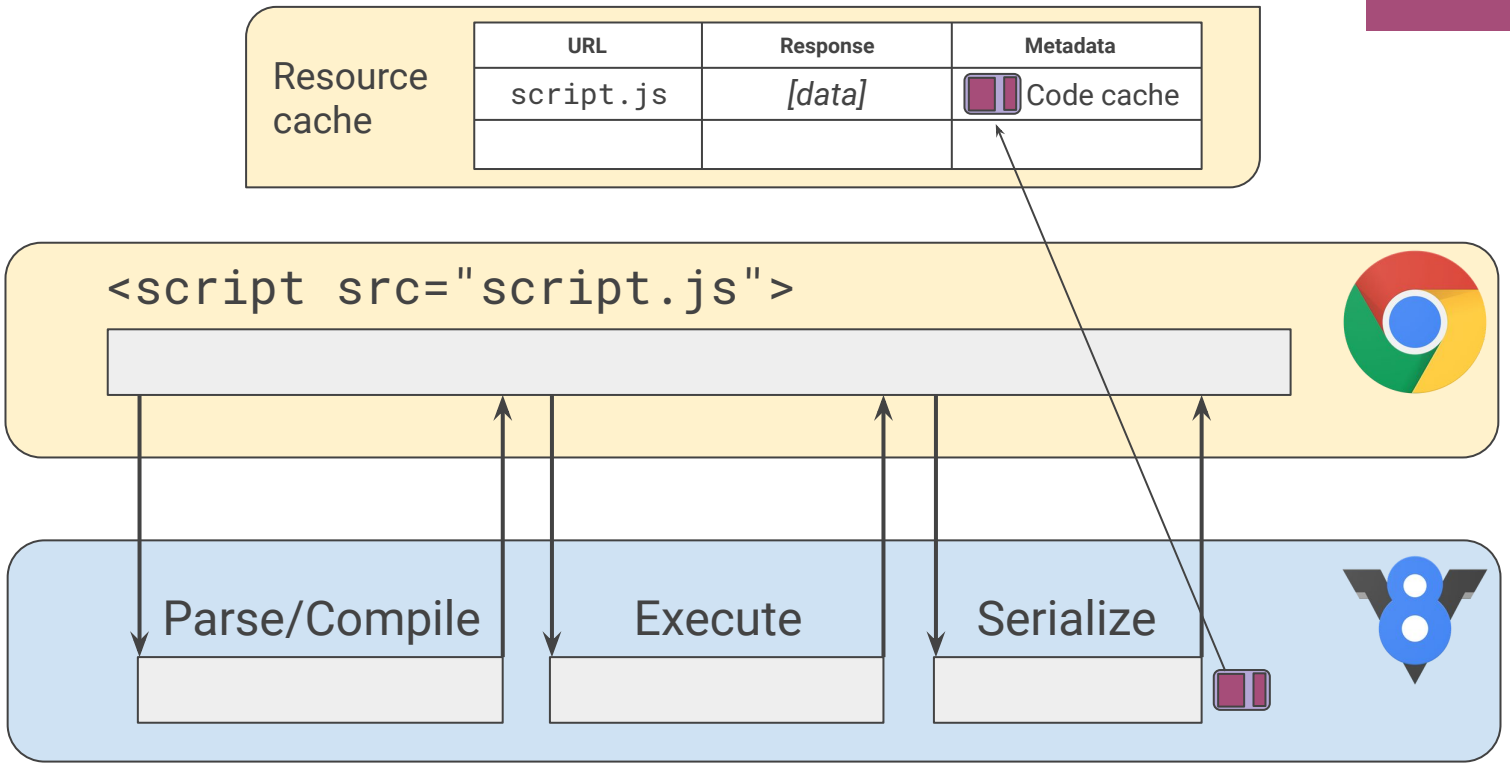
```
var TypeScript;
function (TypeScript) {
    var ASTSpan = (function () {
        function ASTSpan() {
            this.minChar = -1;
            this.limChar = -1;
        }
        return ASTSpan;
    })();
    TypeScript.ASTSpan = ASTSpan;
    var AST = (function (_super) {
        __extends(AST, _super);
        function AST(nodeType) {
                _super.call(this);
            this.nodeType = nodeType;
            this.type = null;
            this.flags = TypeScript.ASTFlags.Writeable;


...


    })(ASTSpan);
...

}(TypeScript || (TypeScript = {}));
```

Parse Compile

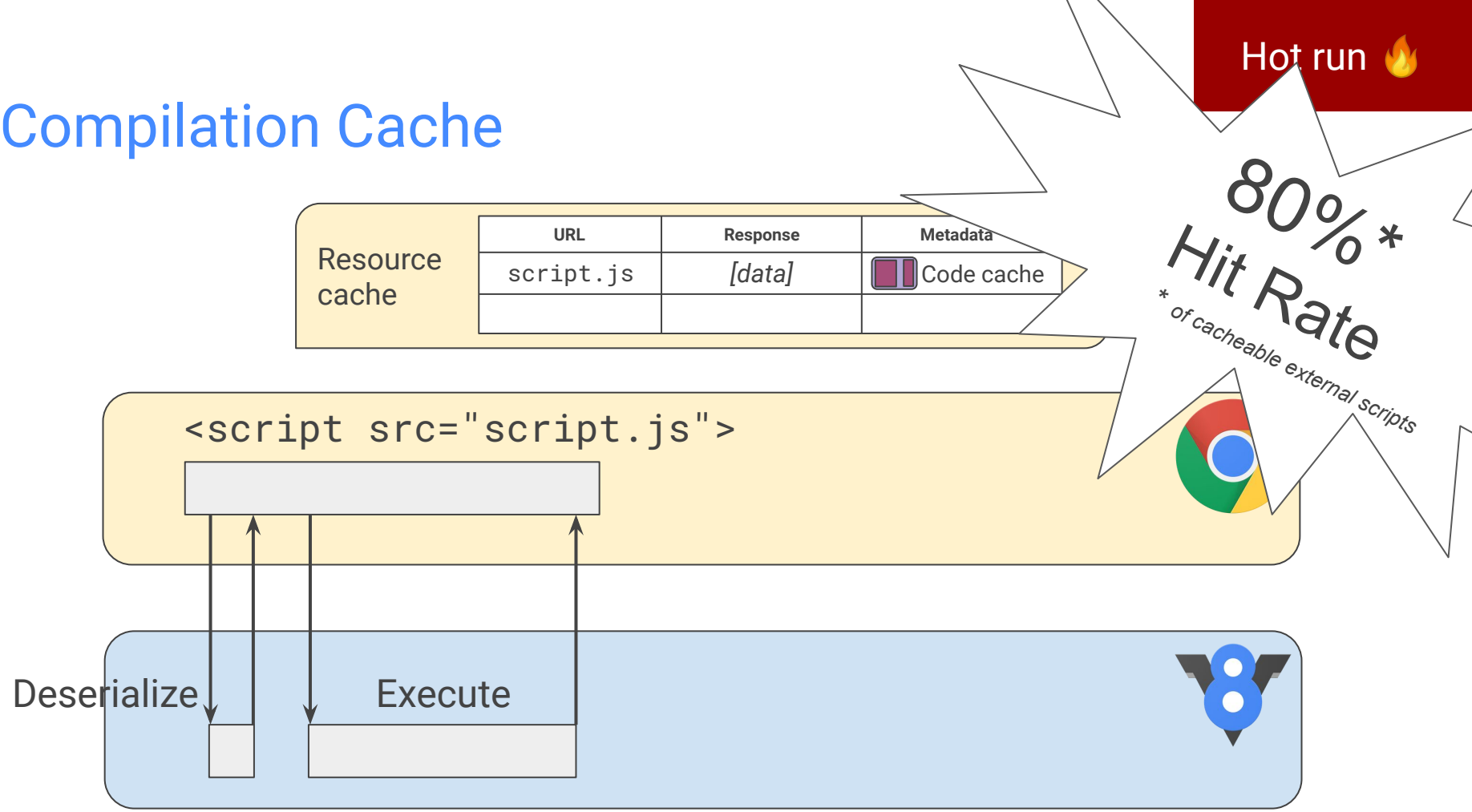
Compilation Cache

Warm run

<72 hours

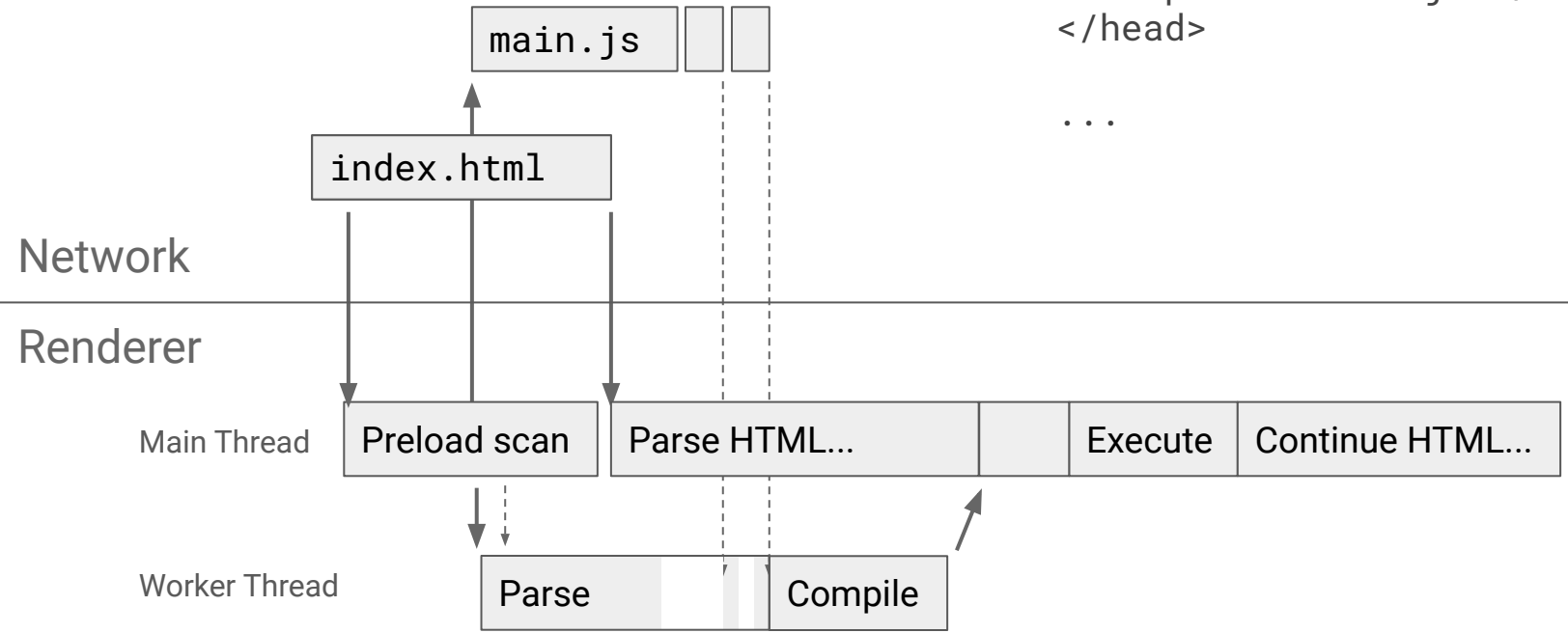


Compilation Cache



Streaming Compilation

```
<html>

<head>
<script src=main.js></script>
</head>

...
```




aiG8hk2PSok.js (static.xx.fbcdn.net)
PGqayaTRb7p.js (static.xx.fbcdn.net)
HzIsj7nHEcW.js (static.xx.fbcdn.net)
38Pls4nVjcc.js (static.xx.fbcdn.net)
p_IsObiLWC3.js (static.xx.fbcdn.net)
D9XH2sOeRXX.js (static.xx.fbcdn.net)
W2wcvygRqxO.js (static.xx.fbcdn.net)
FH-T6q5T4uh.js (static.xx.fbcdn.net)
Main — https://www.facebook.com/shakira
Task
Task
Task
Ti...d
Fu...ll
Eval...ript
(ano...us)
a.__d
ap...d
ap...d
e
(a...s)
appl...ard
appl...ard
$.root
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolForegroundWorker
ThreadPoolServiceThread

Peak performance!
Load time!
Memory!
Latency!
Security!