ONCE UPON A TIME IN THE VM

A JOURNEY FROM SMALLTALK TO NATIVE CODE

CLEMENT BERA
CAMILLO BRUNI

1 The Standard:
From Smalltalk to Bytecode

2 The once Handsome:
The Cog

3 The Good:
The NativeBoost & The Mate

Source Code

AST

IR

Byte Code

AST

OPTIMIZE

IR

OPTIMIZE

ByteCode

DEMO

1.1 Bytecode Compiler
1.2 Bytecode Optimization
1.3 Connecting the VM

Smalltalk

Bytecode

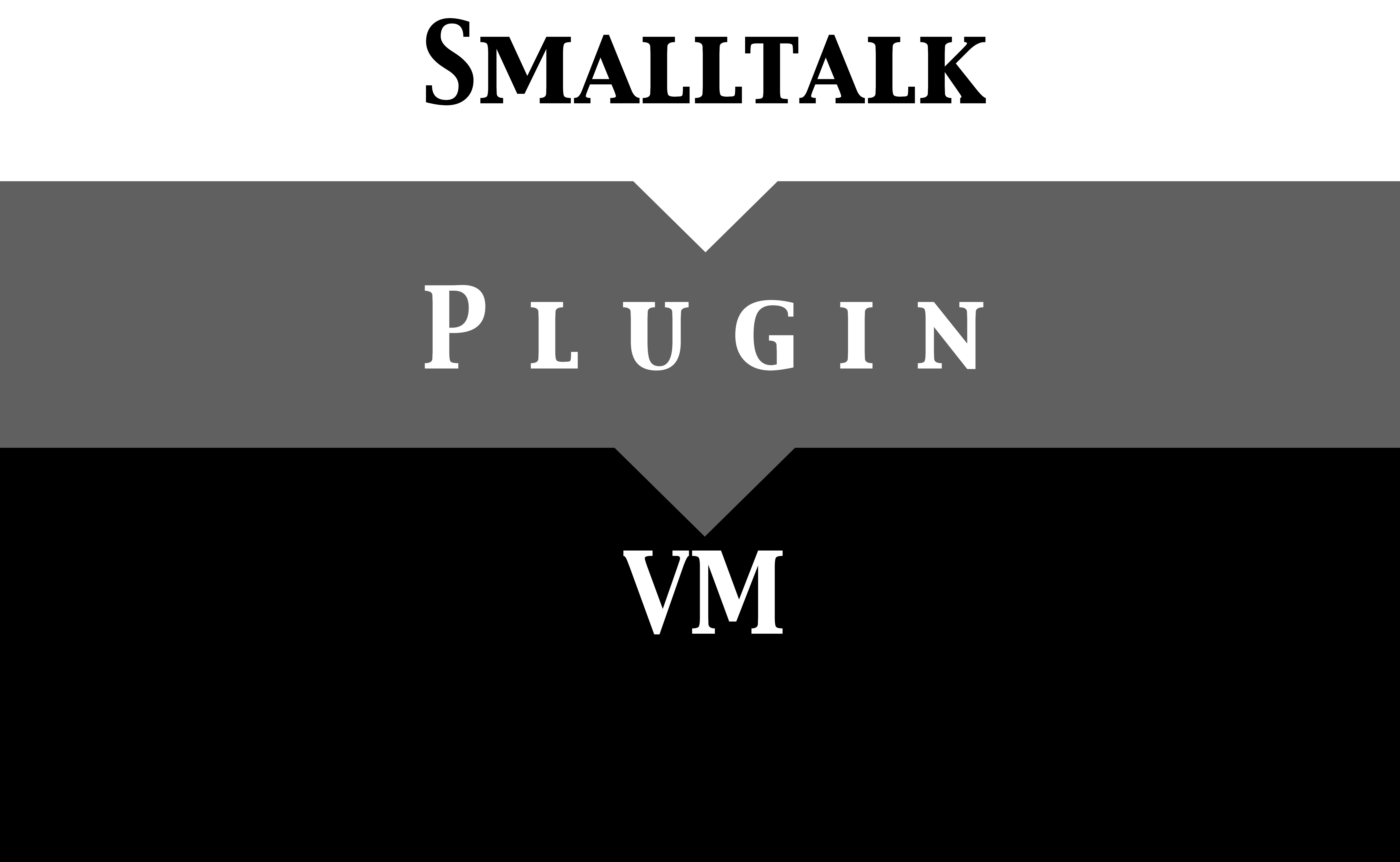
VM

Smalltalk

Primitive

VM

DEMO


Smalltalk
Plugin
VM

DEMO

Smalltalk

Special Objects

VM

Smalltalk

Bytecode
Primitives
Special Objects

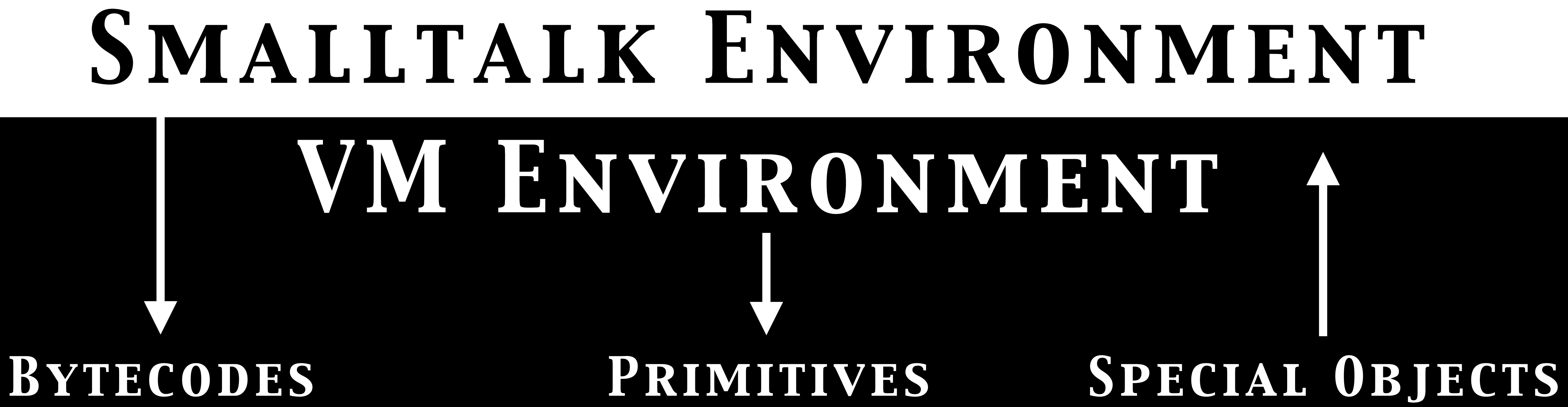
VM

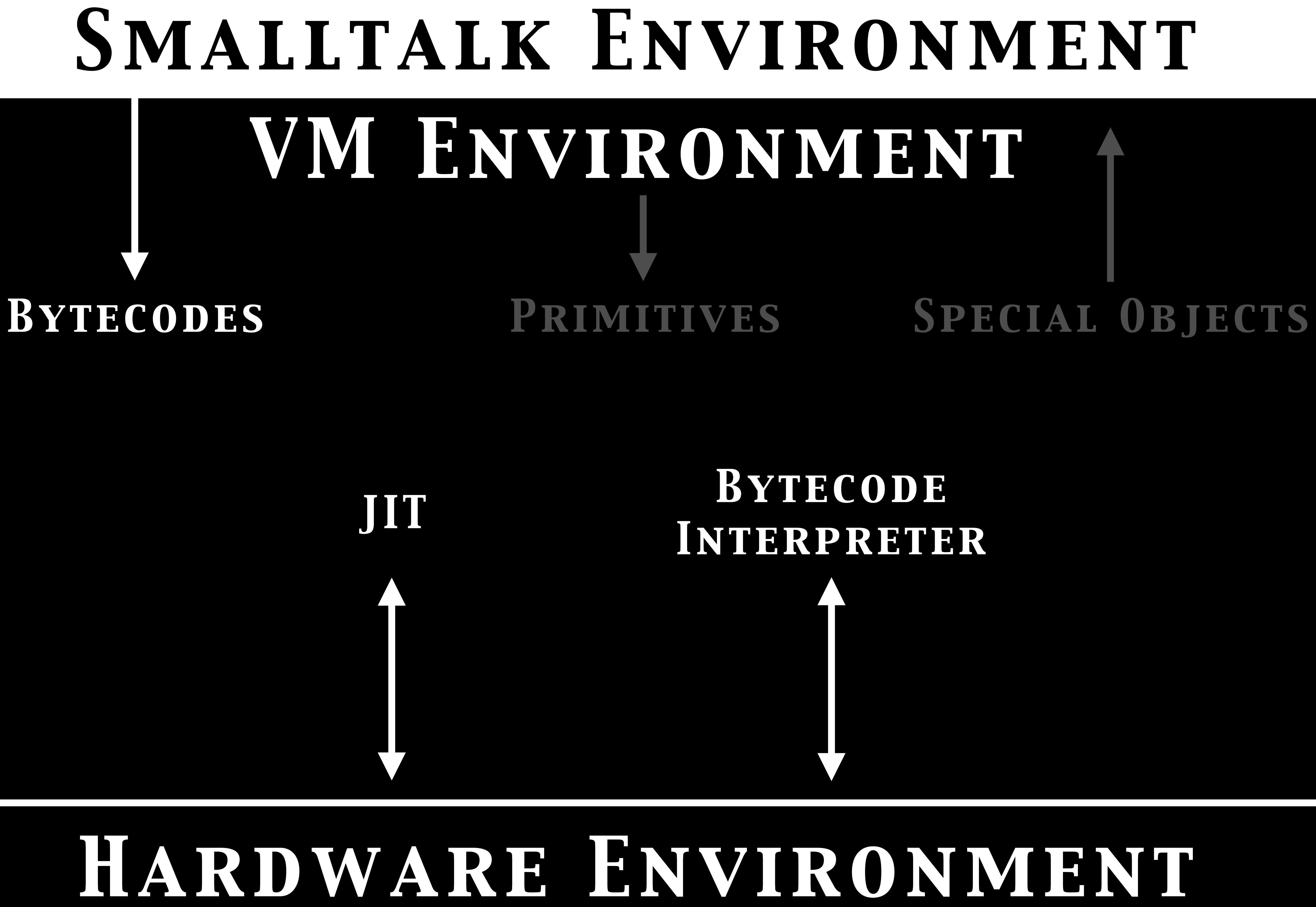
1 The Standard:
From Smalltalk to Bytecode

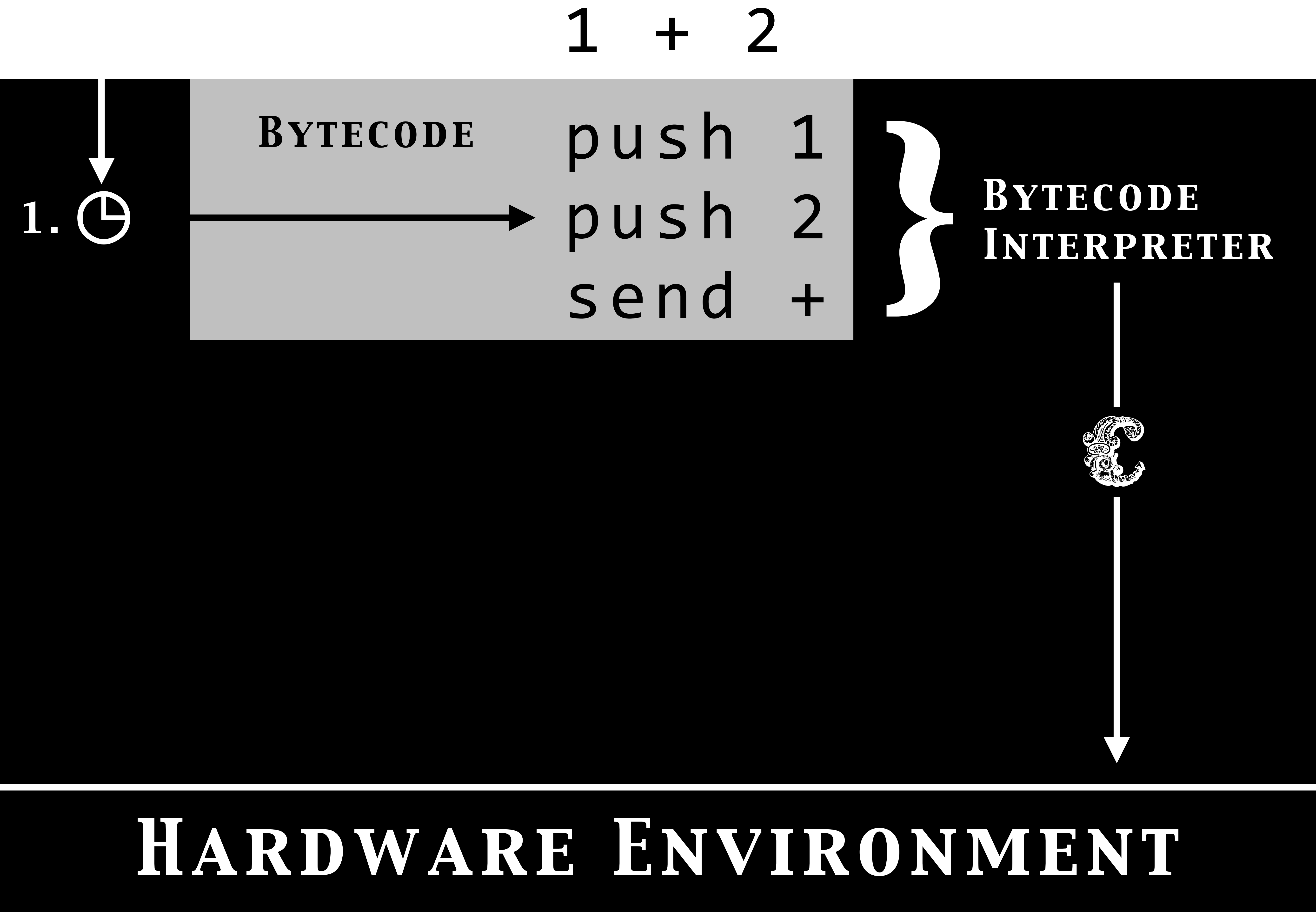
2 The once Handsome:
The Cog

3 The Good:
The NativeBoost & The Mate

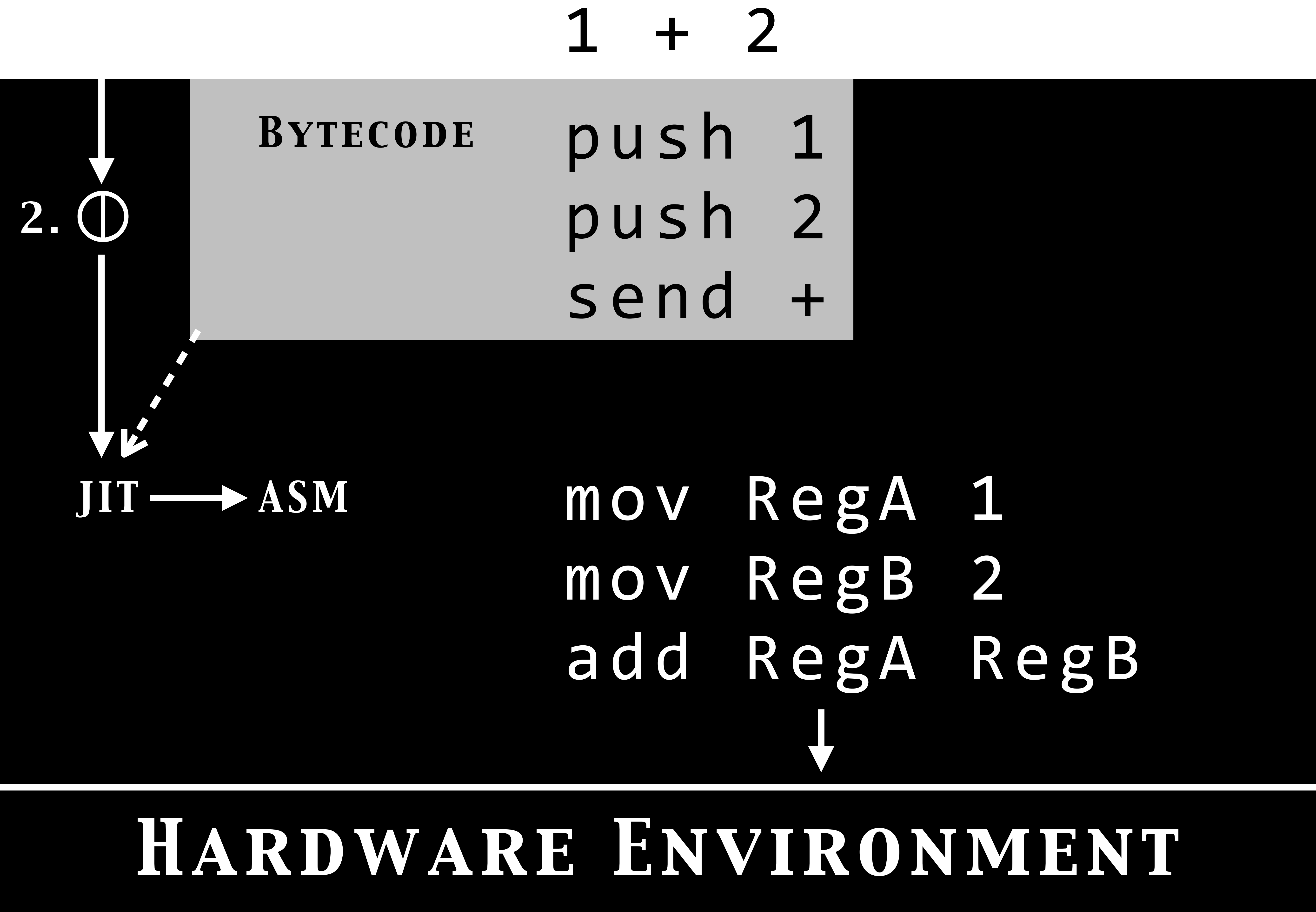
Smalltalk Environment


Smalltalk Environment
VM Environment
Bytecodes
Primitives
Special Objects


Smalltalk Environment
VM Environment
Bytecodes
Primitives
Special Objects
JIT
Bytecode Interpreter
Hardware Environment



```
push 1
push 2
send +
```

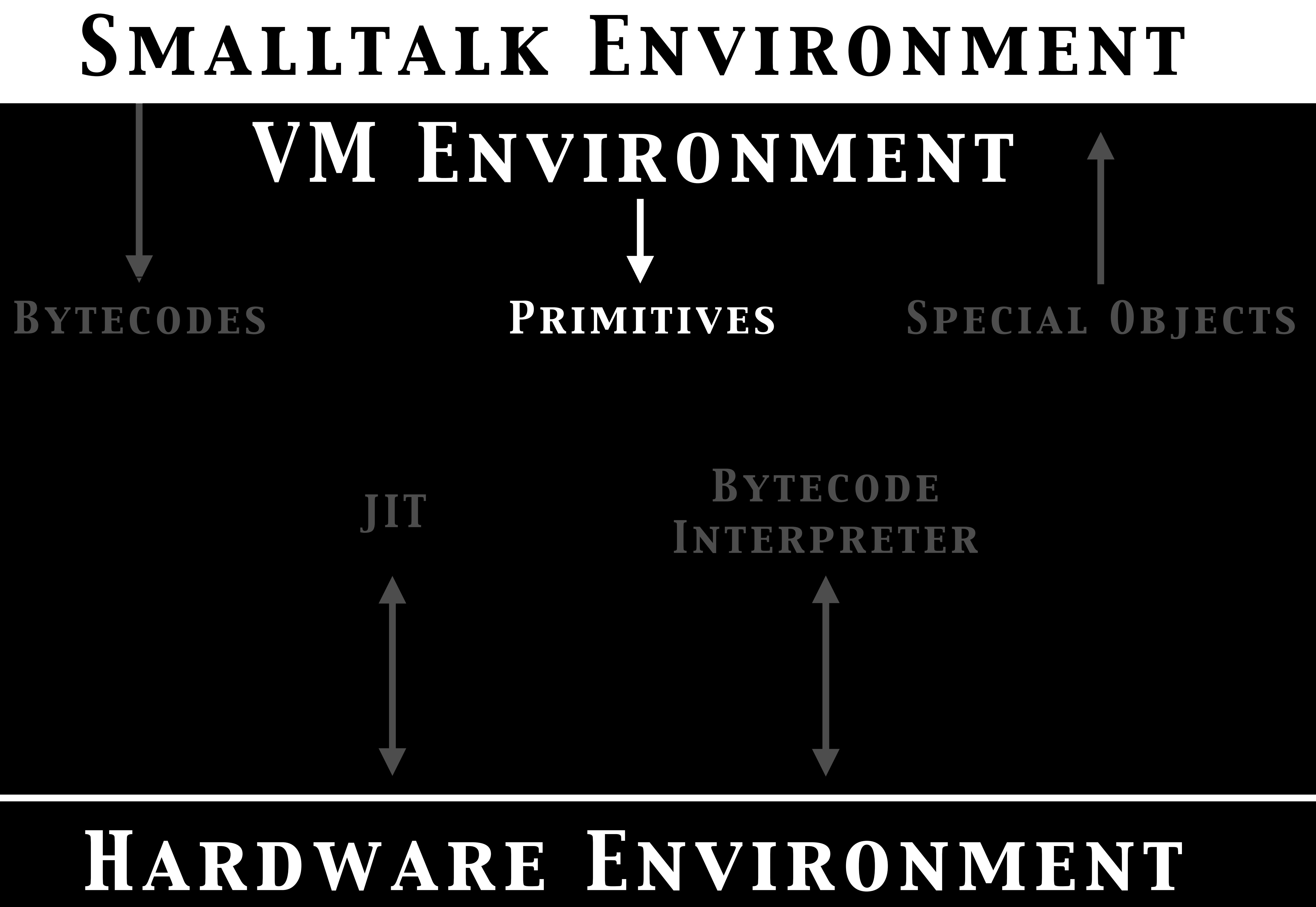


```
1 + 2
```

Bytecode

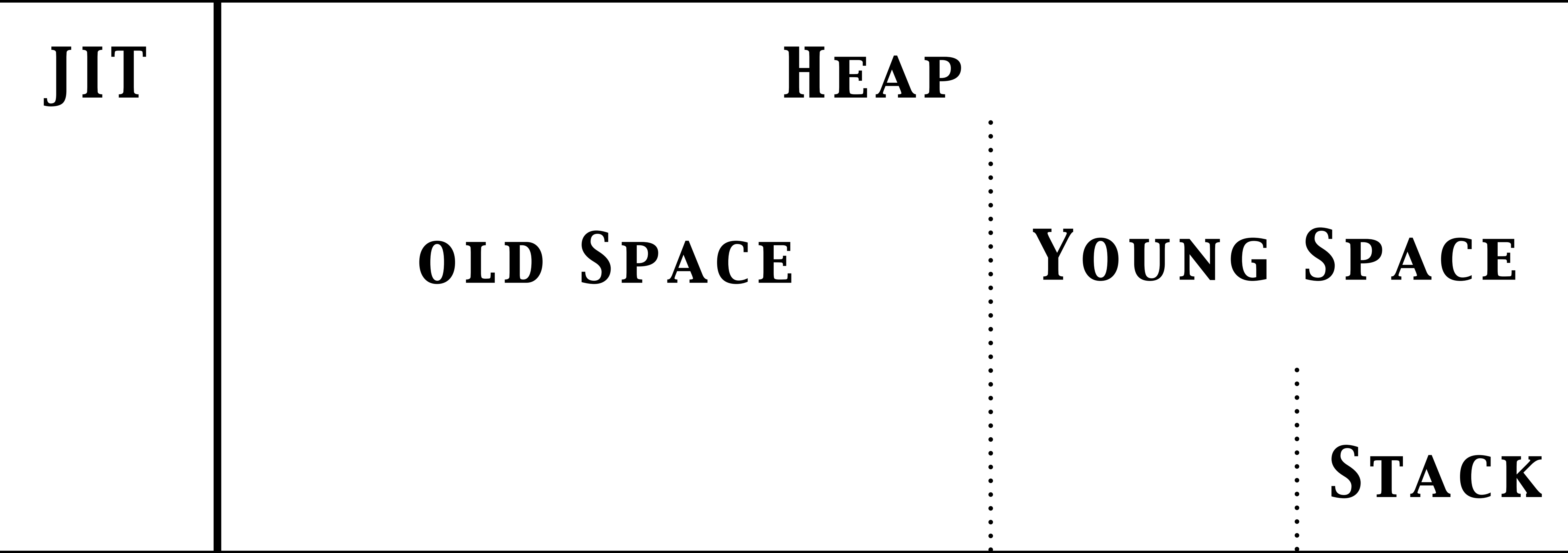
```
push 1
push 2
send +
```

```
mov RegA 1
mov RegB 2
add RegA RegB
```

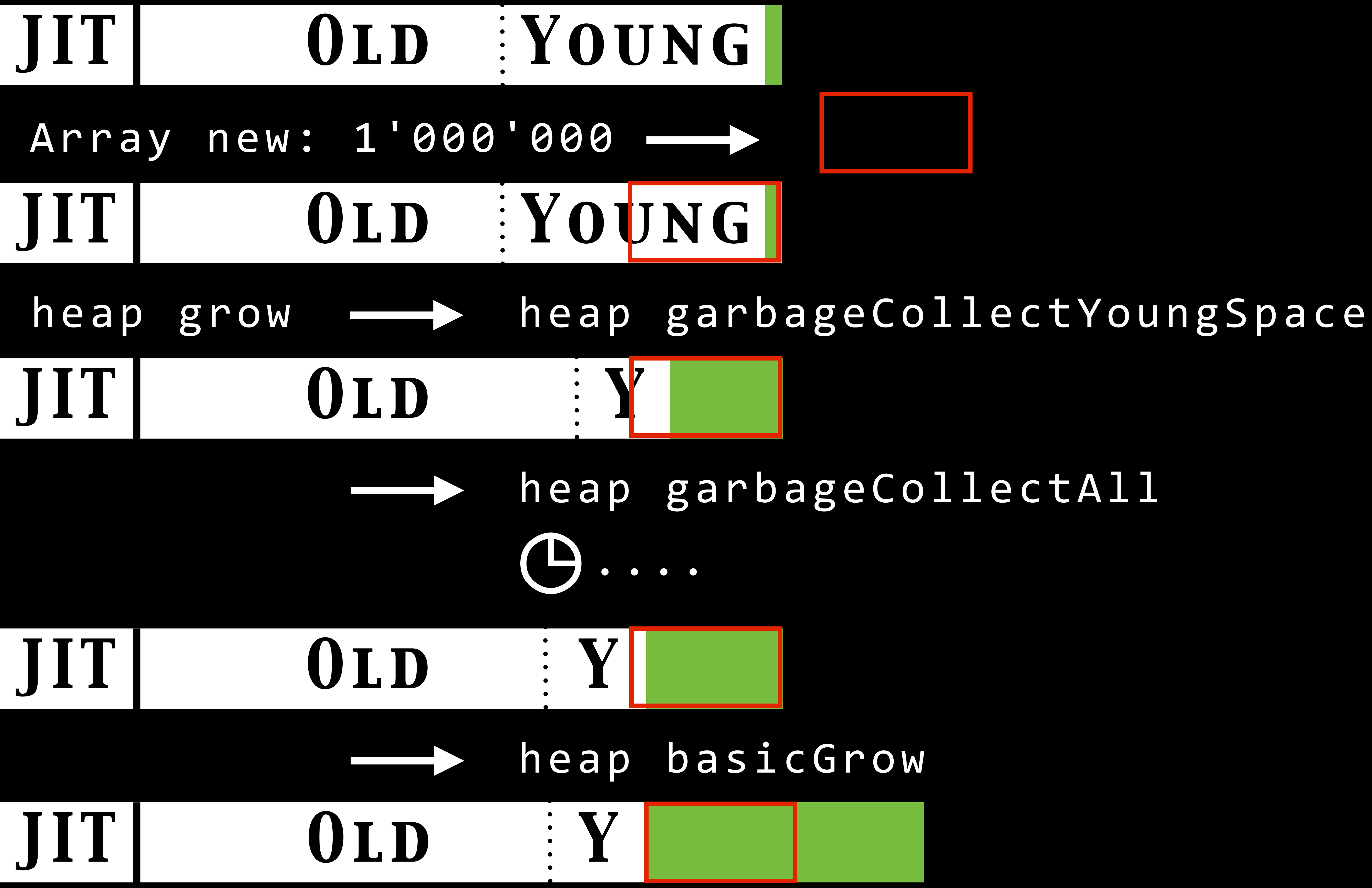

Smalltalk Environment
VM Environment
Bytecodes
Primitives
Special Objects
JIT
Bytecode Interpreter
Hardware Environment

```
Object basicNew
<primitive: 70>

objectSize := Object instanceSize.

(heap hasEnoughSpace: objectSize)
    ifFalse: [ heap grow ].

raw := heap allocate: objectSize.

raw initializeObjectStructure.

^ raw asObject
```

Memory Management



Garbage Collection



```
Array new: 1'000'000 →
heap grow → heap garbageCollectYoungSpace
→ heap garbageCollectAll
....
→ heap basicGrow
```

Object basicNew

```
<primitive: 70>

objectSize := Object instanceSize.

(heap hasEnoughSpace: objectSize)
    ifFalse: [ heap grow ].

raw := heap allocate: objectSize.

raw initializeObjectStructure.

^ raw asObject
```

VM Maker

SLANG

VM Maker

SLANG

VM Maker

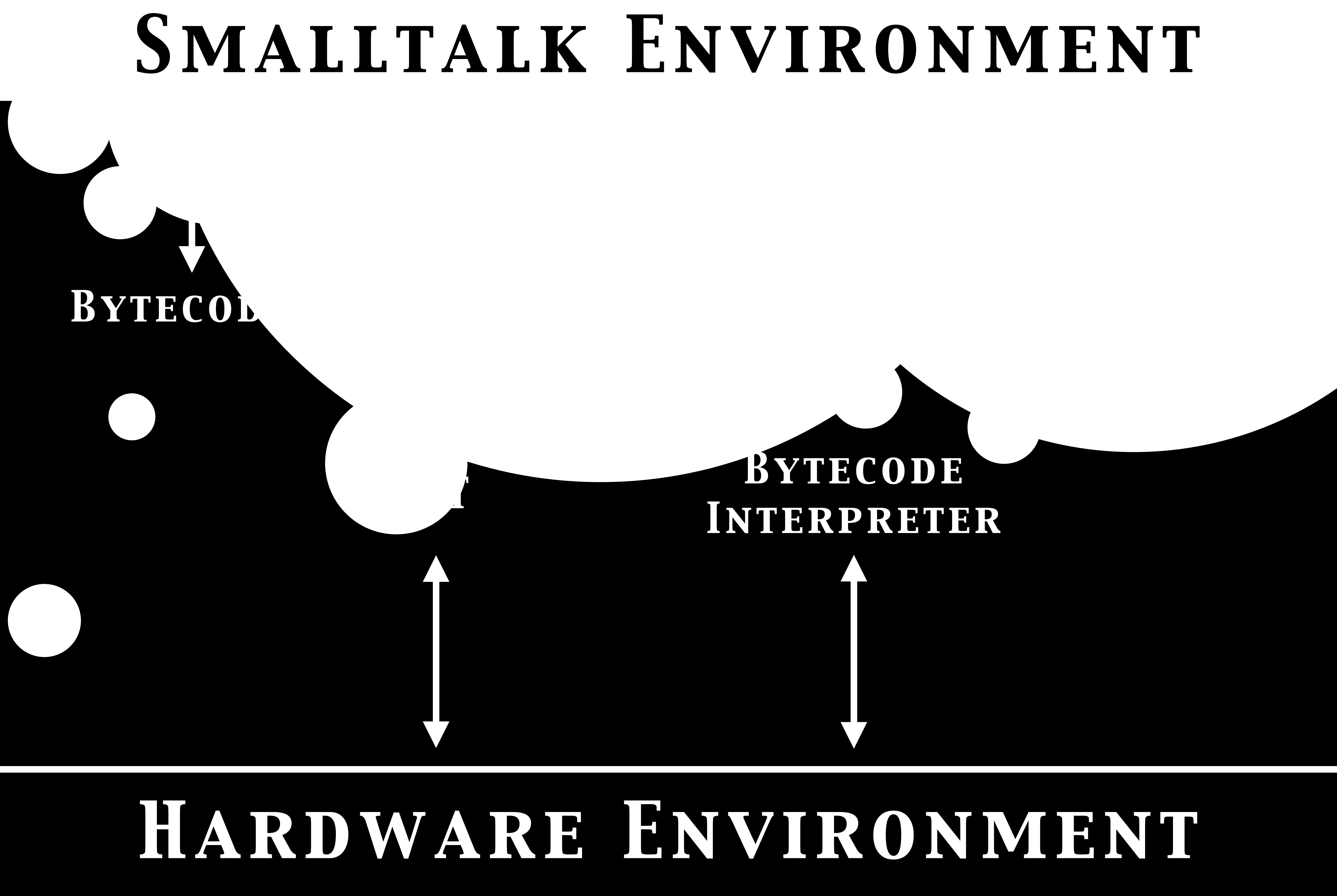
SLANG

1. The Standard:
   From Smalltalk to Bytecode

2. The once Handsome:
   The Cog

3. **The Good:**
   The NativeBoost & The Mate

VM Maker


Smalltalk Environment
Bytecod
Bytecode
Interpreter
Hardware Environment

Smalltalk

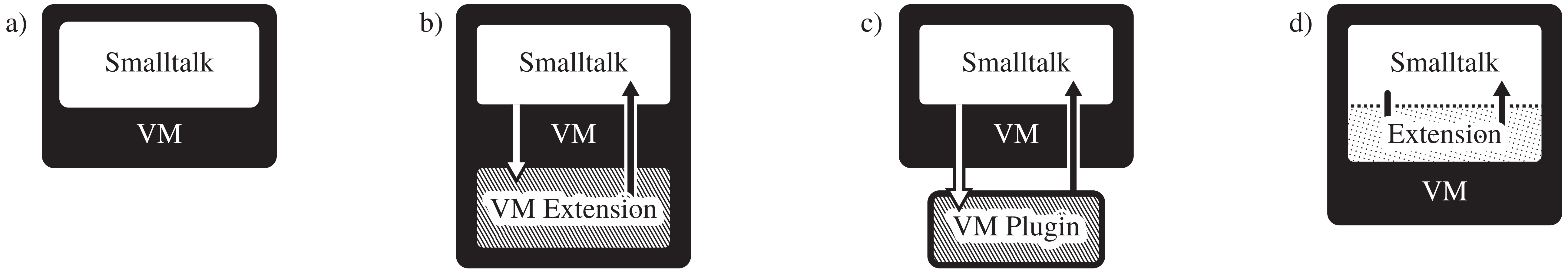
NativeBoost

Hardware

NativeBoost

- Call native Code from the language side
- Replace Plugins
- Replace Primitives
- Replace JIT Compiler

VM Evolution



Drawbacks

- C
- Complex Architecture
- Needs cleaning
- Needs documentation

The Mate

Bringing finest Reflection to VMs

The Goal

- ST-like high-level VM
- Metacircular
- Strong Reflection
- Control down to the Metal
- Maintainability over Performance

The Benefits

- Smalltalkish Development Process
  - Immediate Feedback
  - Live Debugger
  - Live Inspection
- C-Independent
- Easy to Understand
- Easy to Maintain

The Status

- Memory Management Simulator
- Hazelnut Bootstrap in Mate
- AST Interpreter
- NativeBoost
- Native Compilation Infrastructure
- Simple GC Implementation

DEMO

Break with old habits to explore new possibilities