IMPROVING LIVE DEBUGGING OF CONCURRENT THREADS

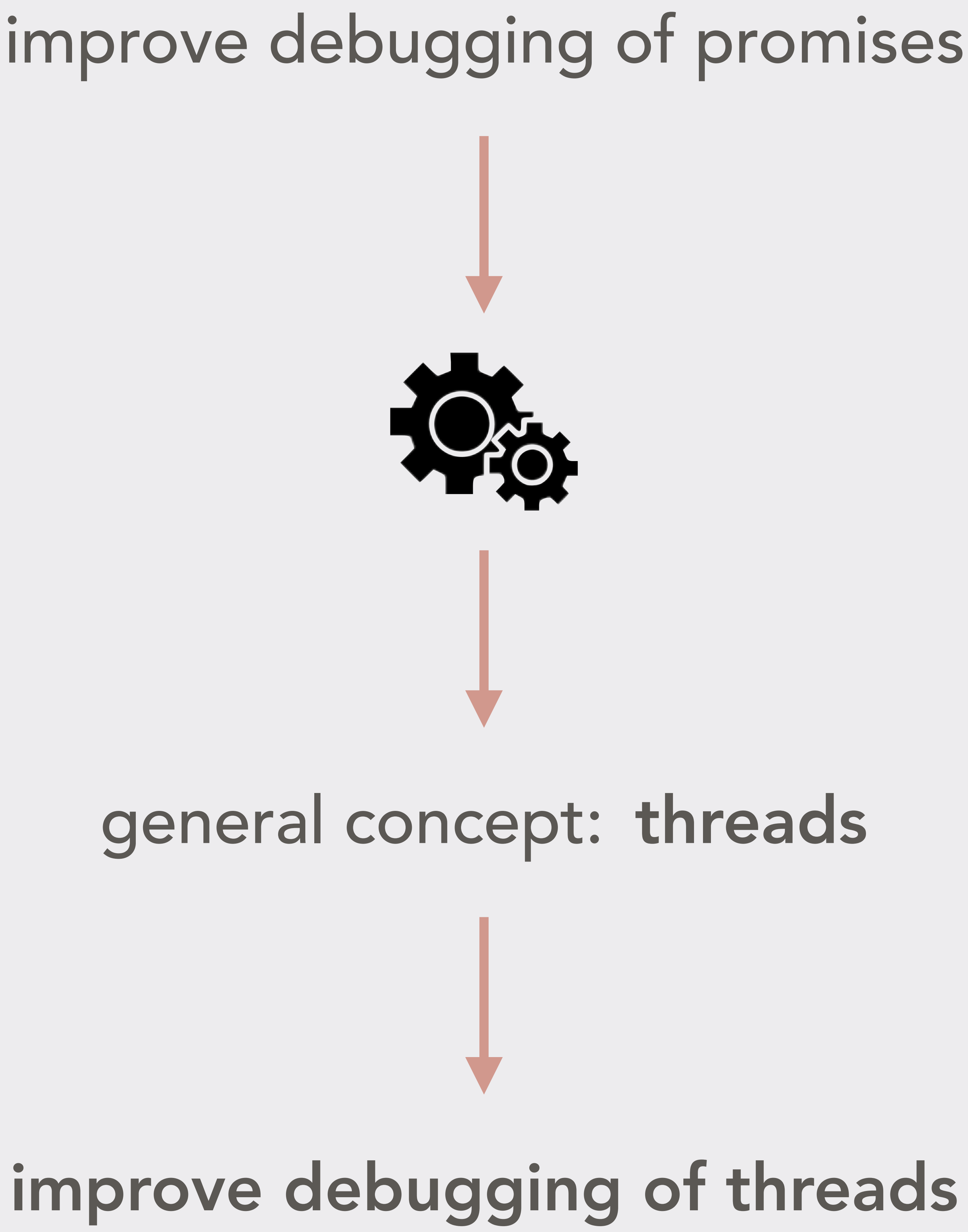
2016
Max Leske
Andrei Chiș

recap

idea

memory & performance

future work

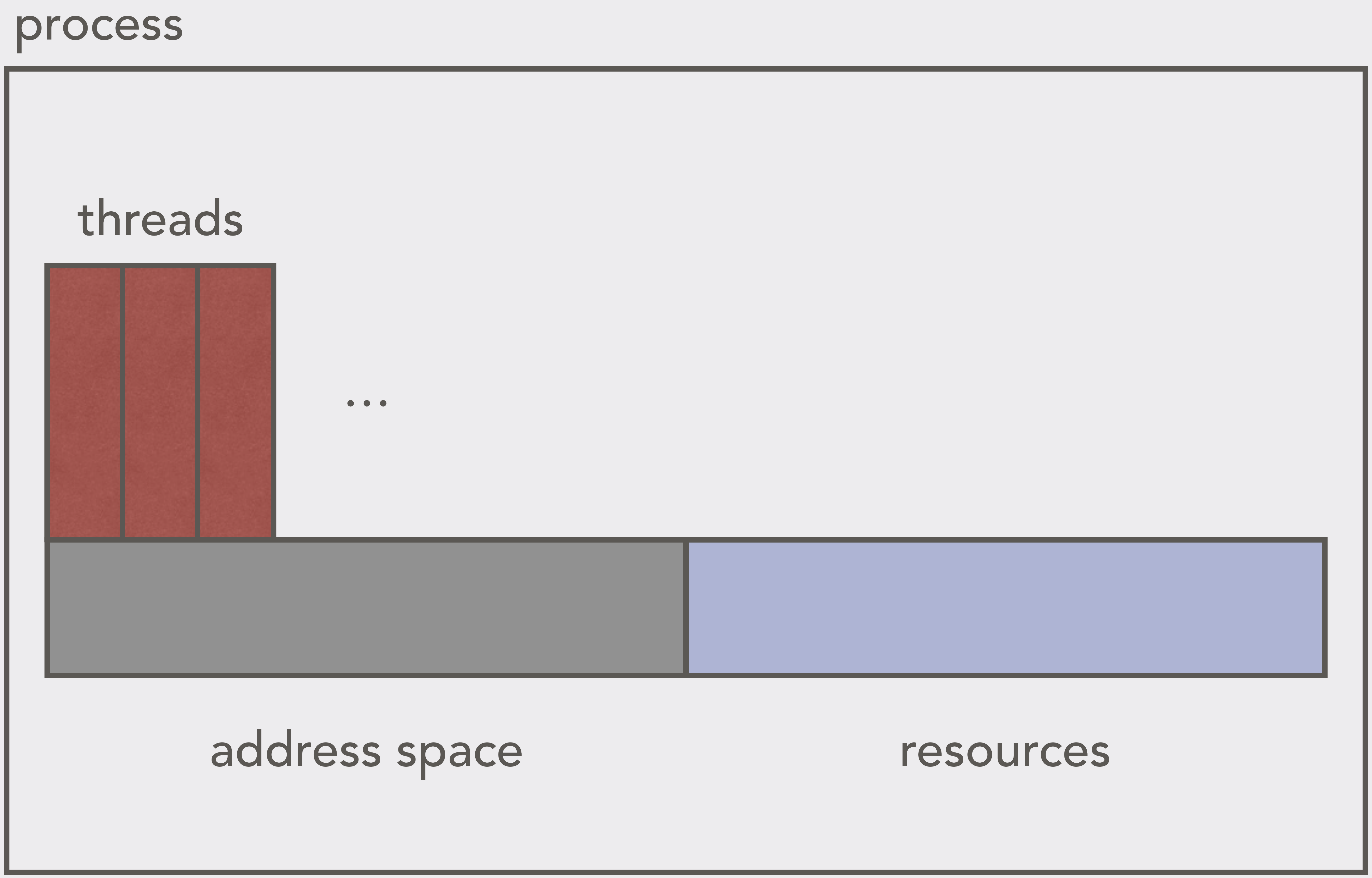


improve debugging of promises

general concept: **threads**

improve debugging of threads

IDEA



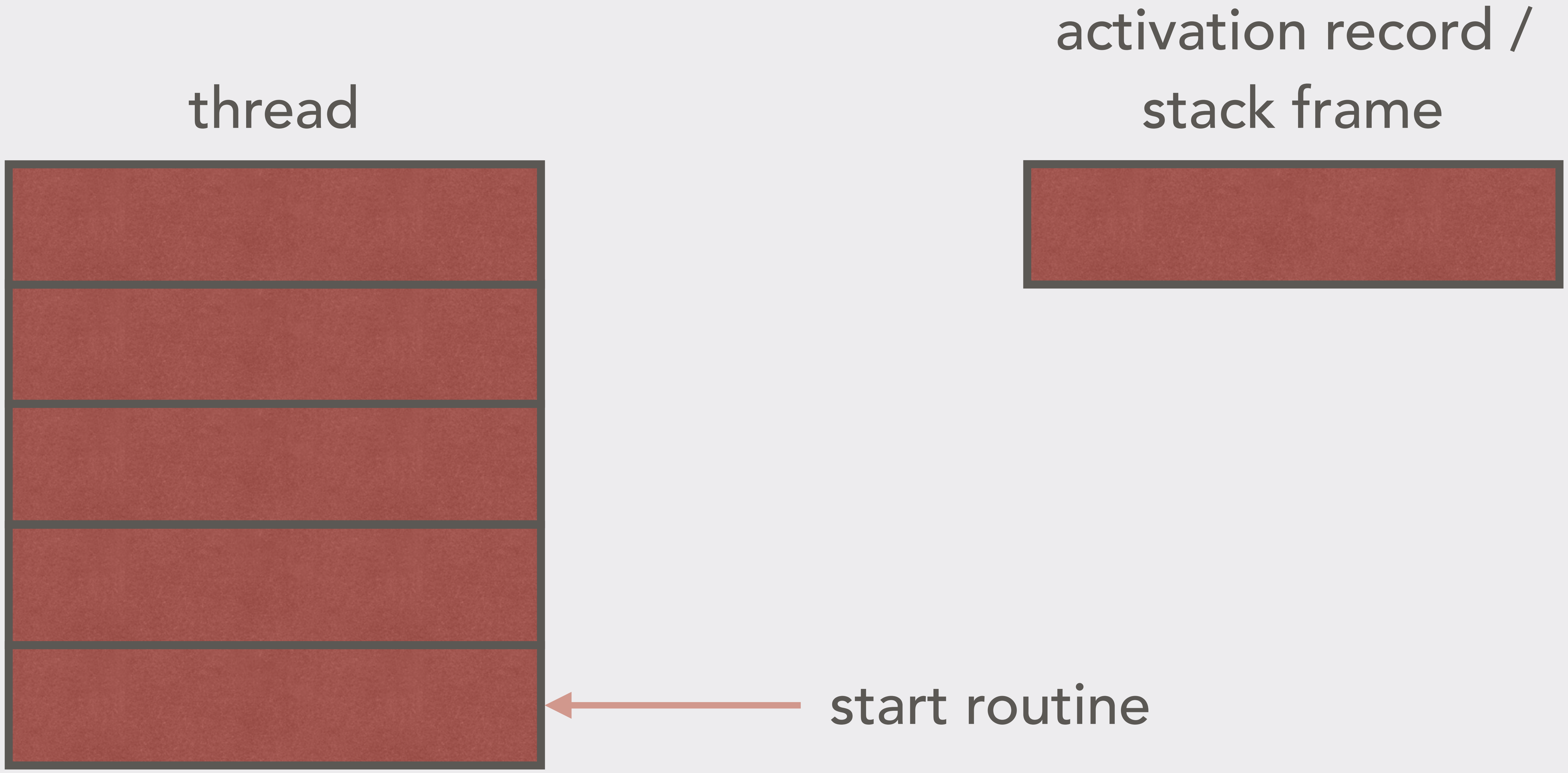
process

threads

...

address space

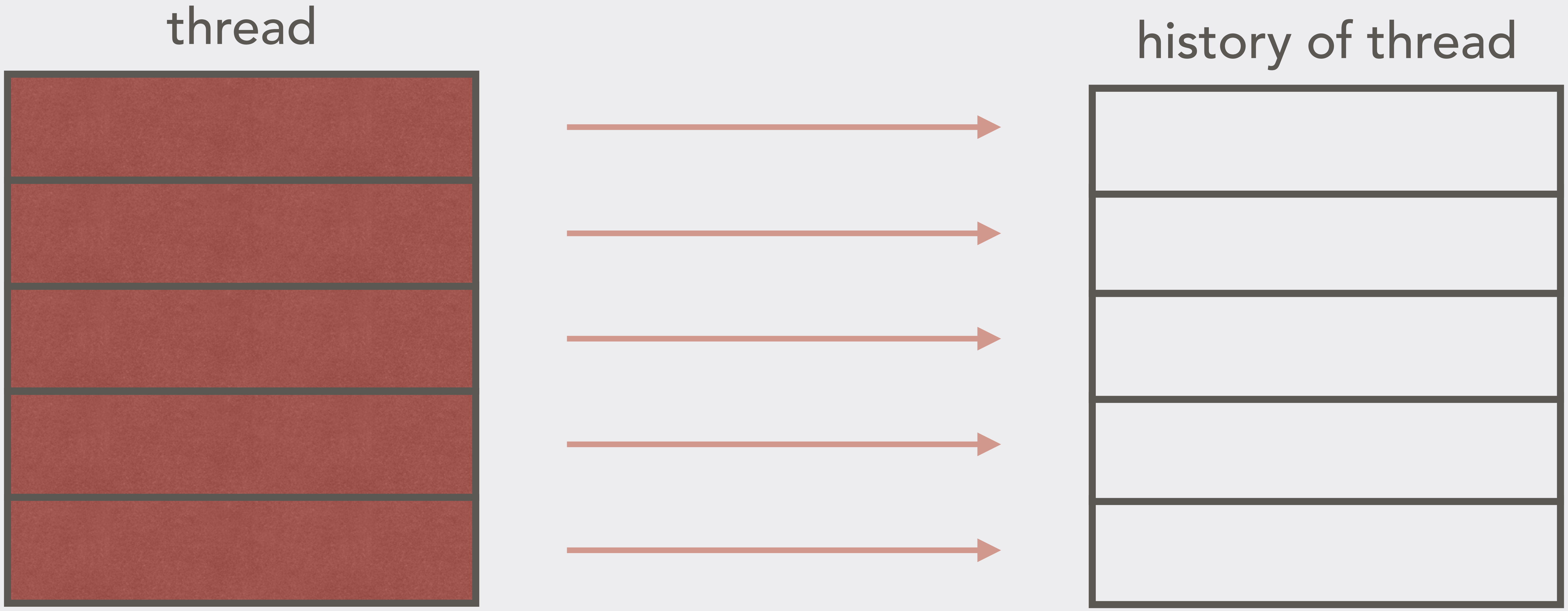
resources



thread

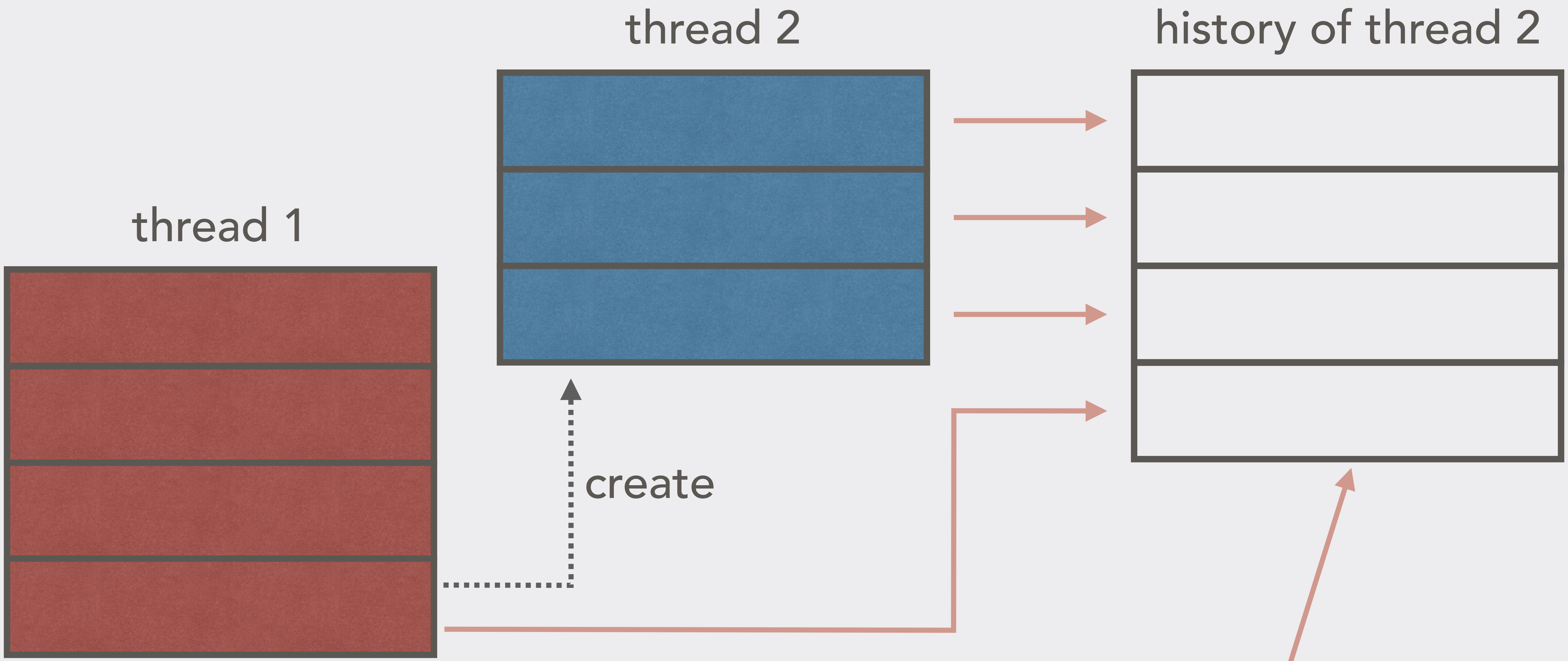
activation record / stack frame

start routine

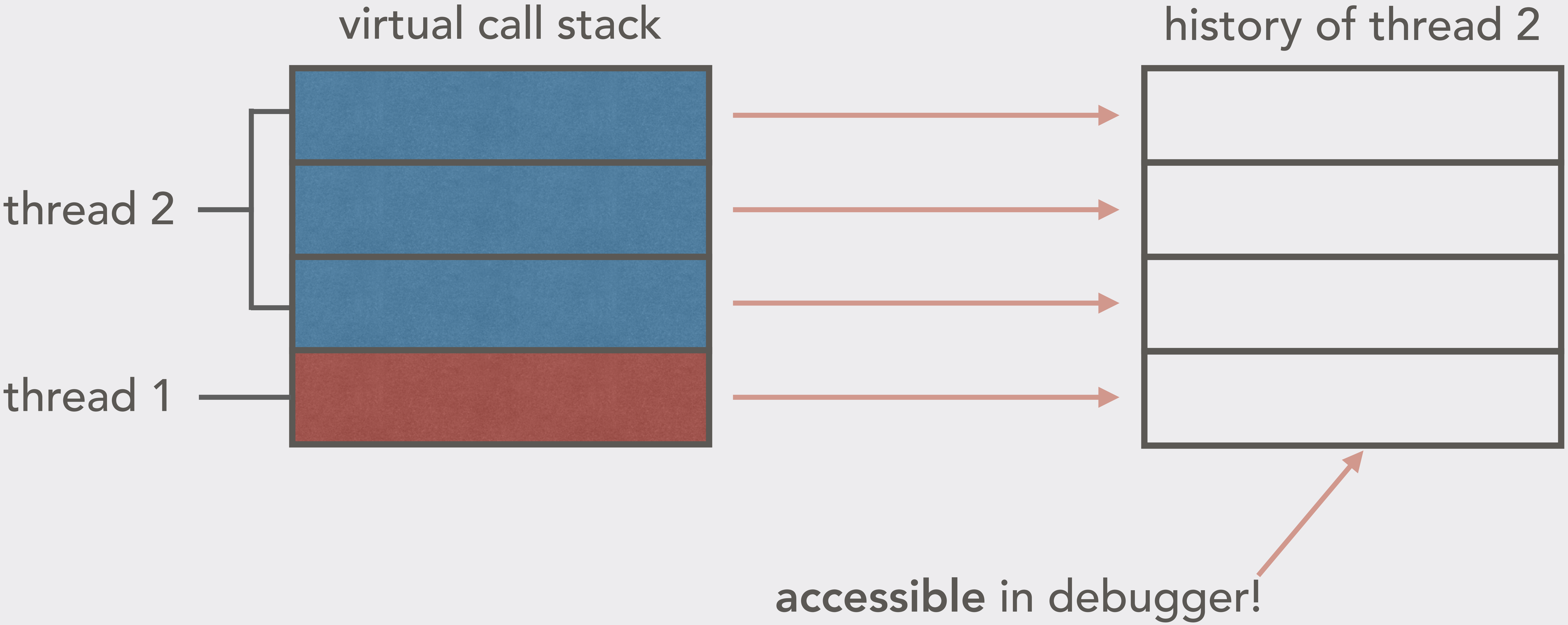


thread

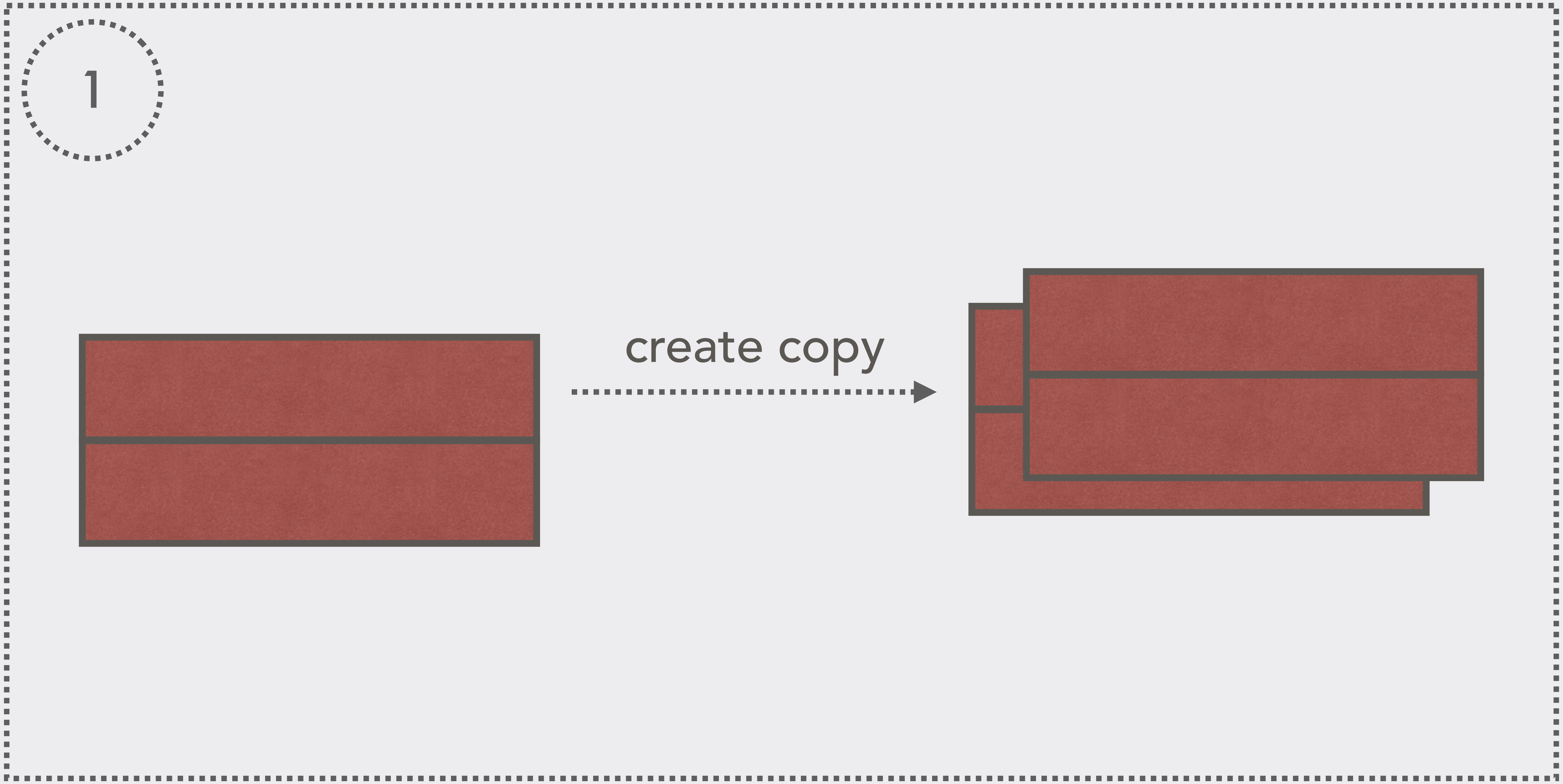
history of thread

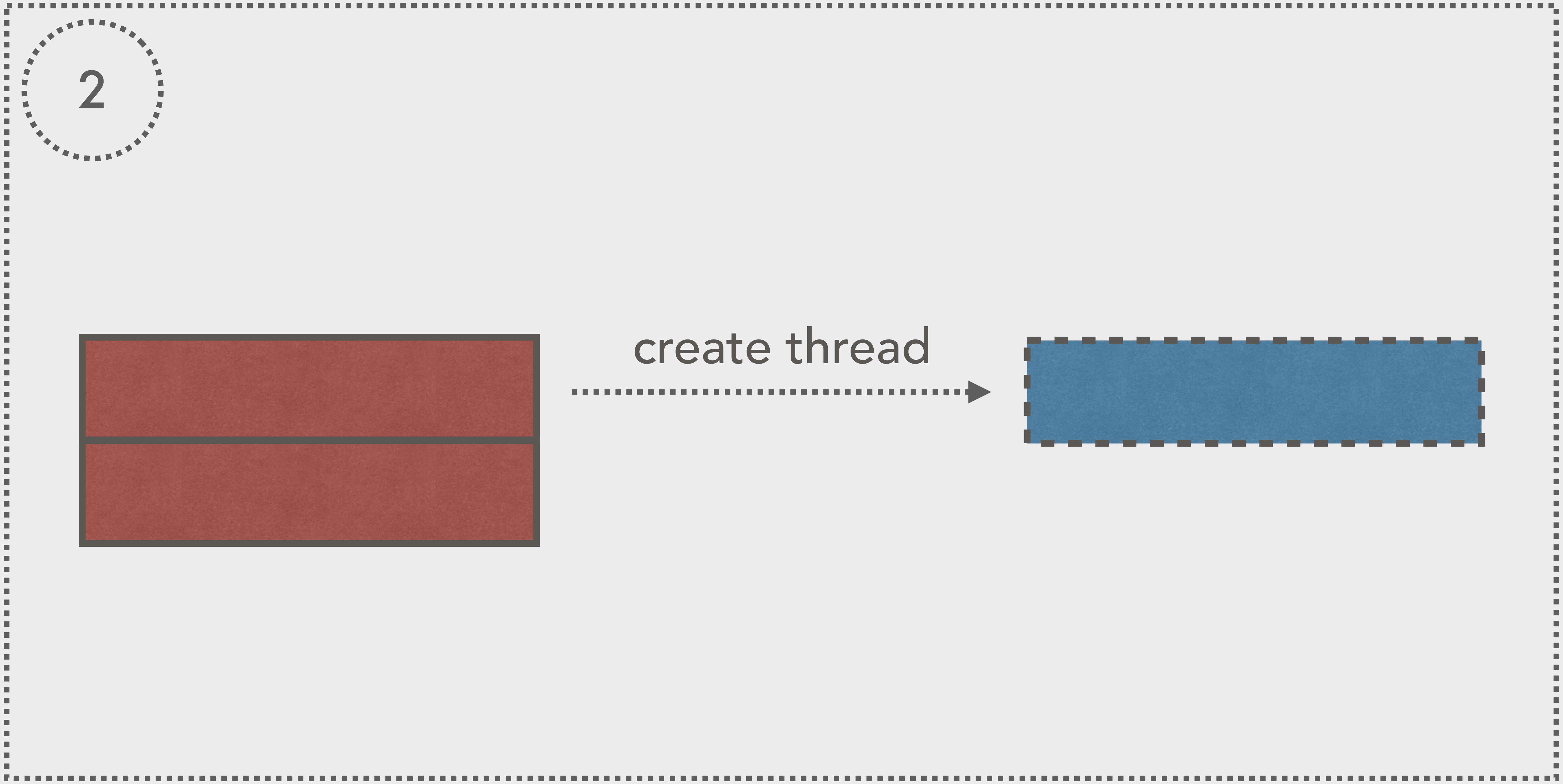


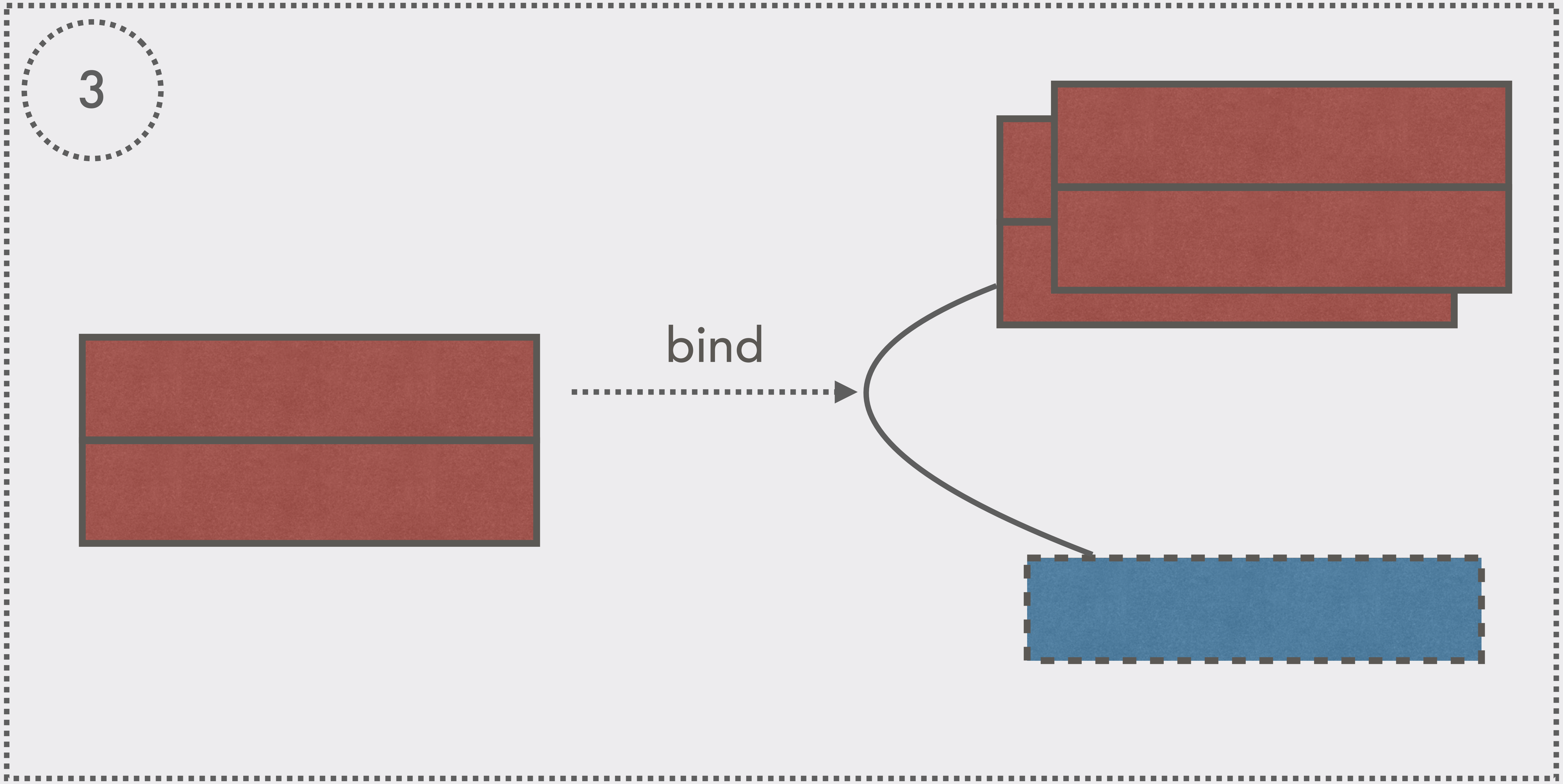
not accessible in debugger!

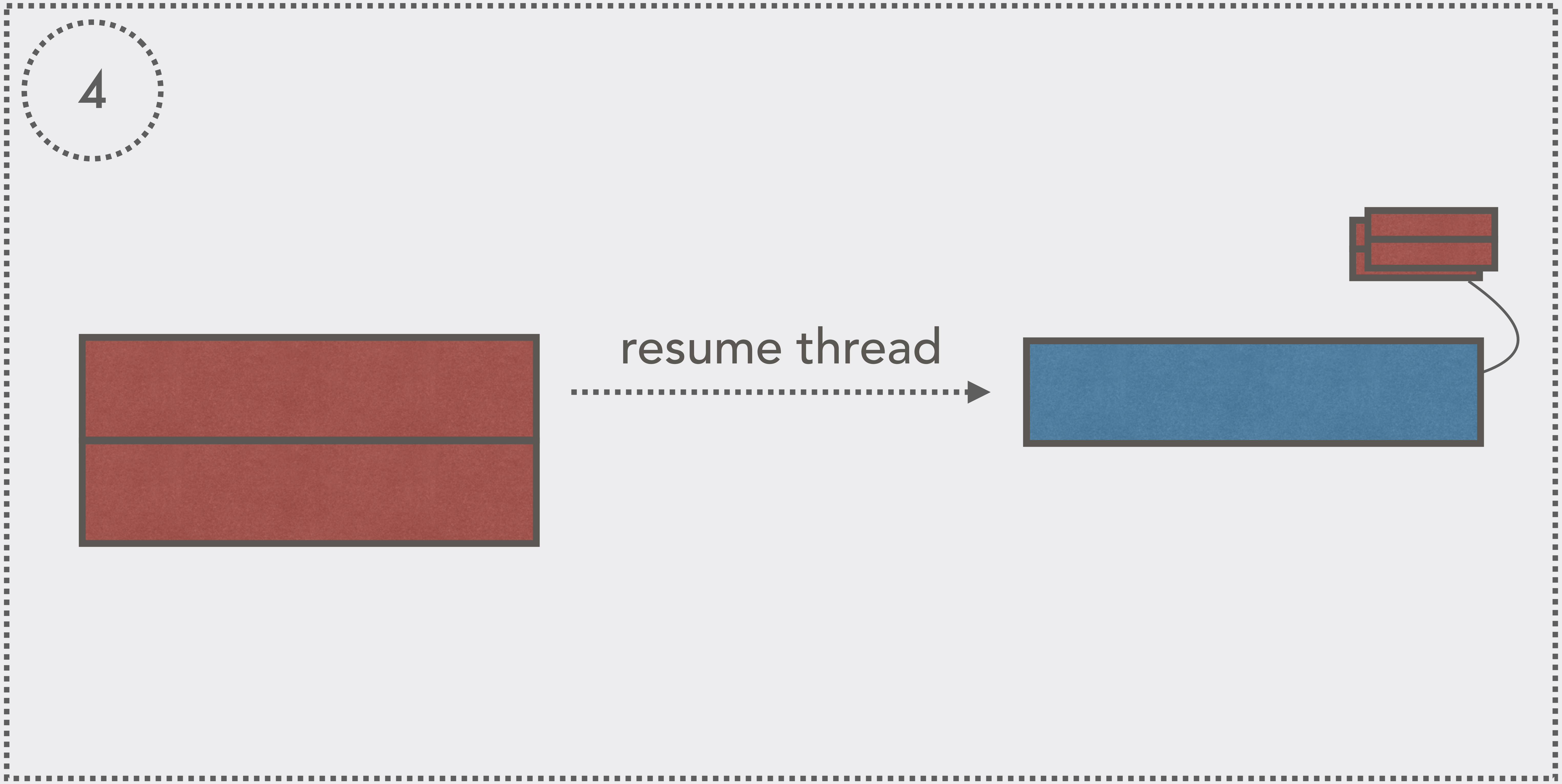

virtual call stack
history of thread 2
thread 2
thread 1
accessible in debugger!

Creating a new thread


1
create copy


2
create thread


3
bind



4

resume thread

solvable as special cases:

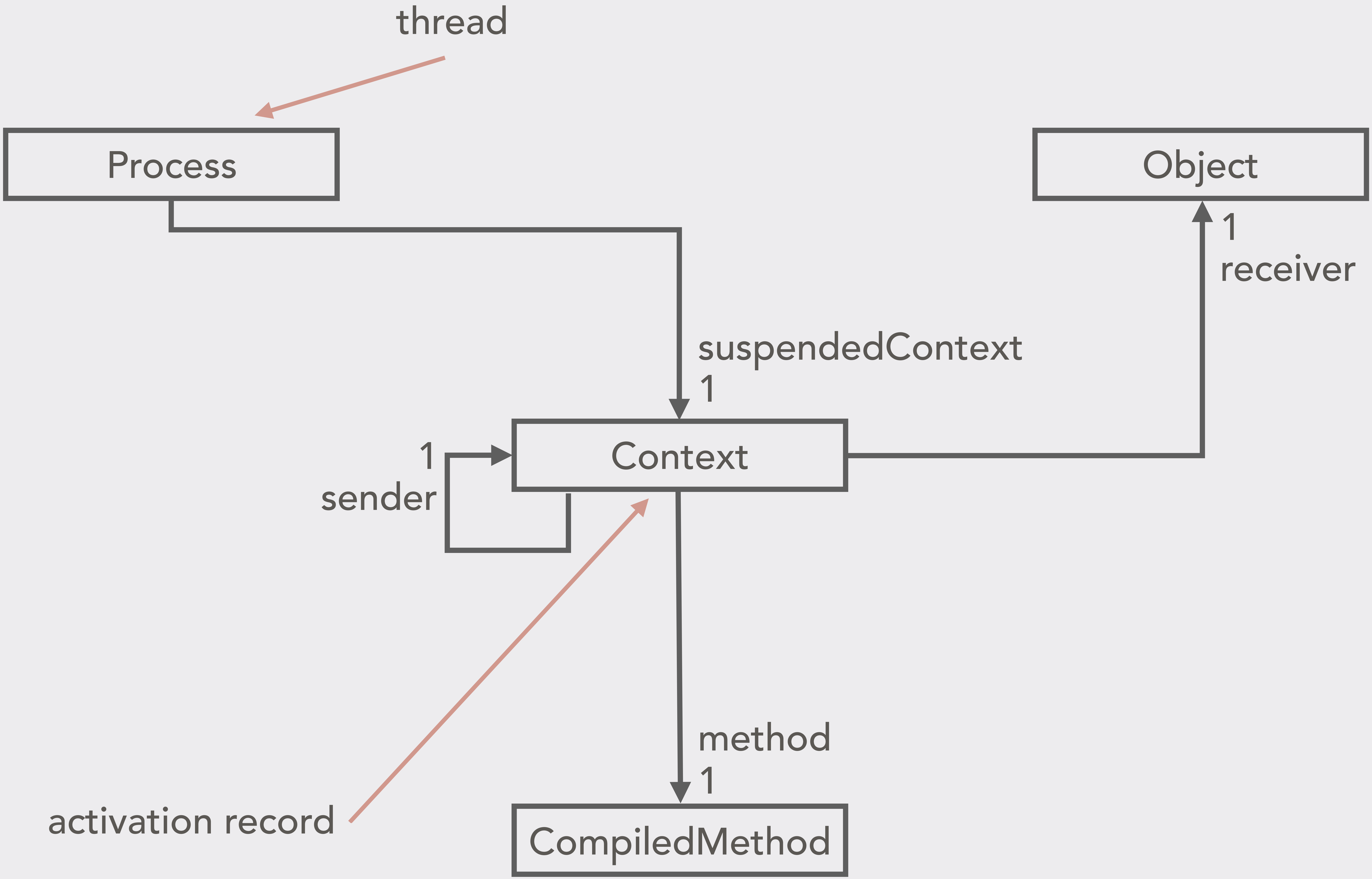
promises

events

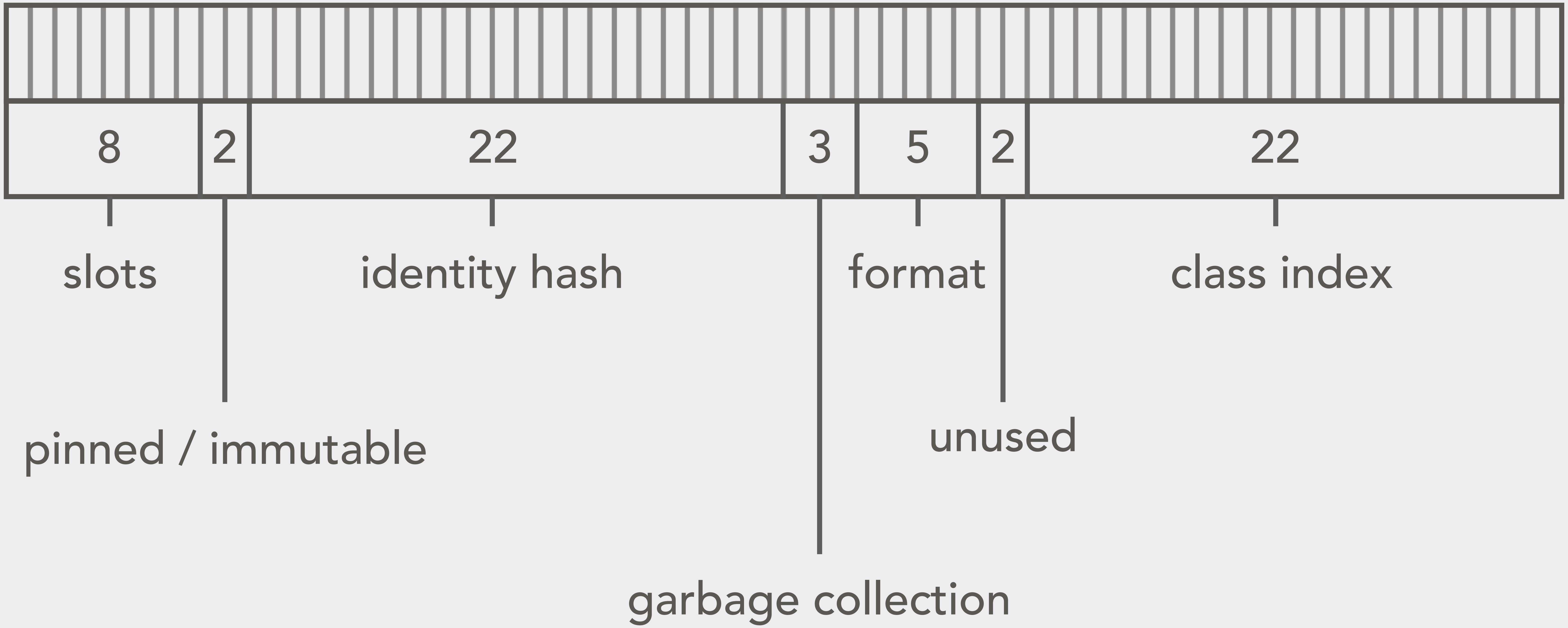
asynchronous network communication

messages (actor model)

MEMORY


thread
Process
Object
1
receiver
suspendedContext
1
1
sender
Context
method
1
activation record
CompiledMethod

object header: 64 bits



8
2
22
3
5
2
22

slots

pinned / immutable

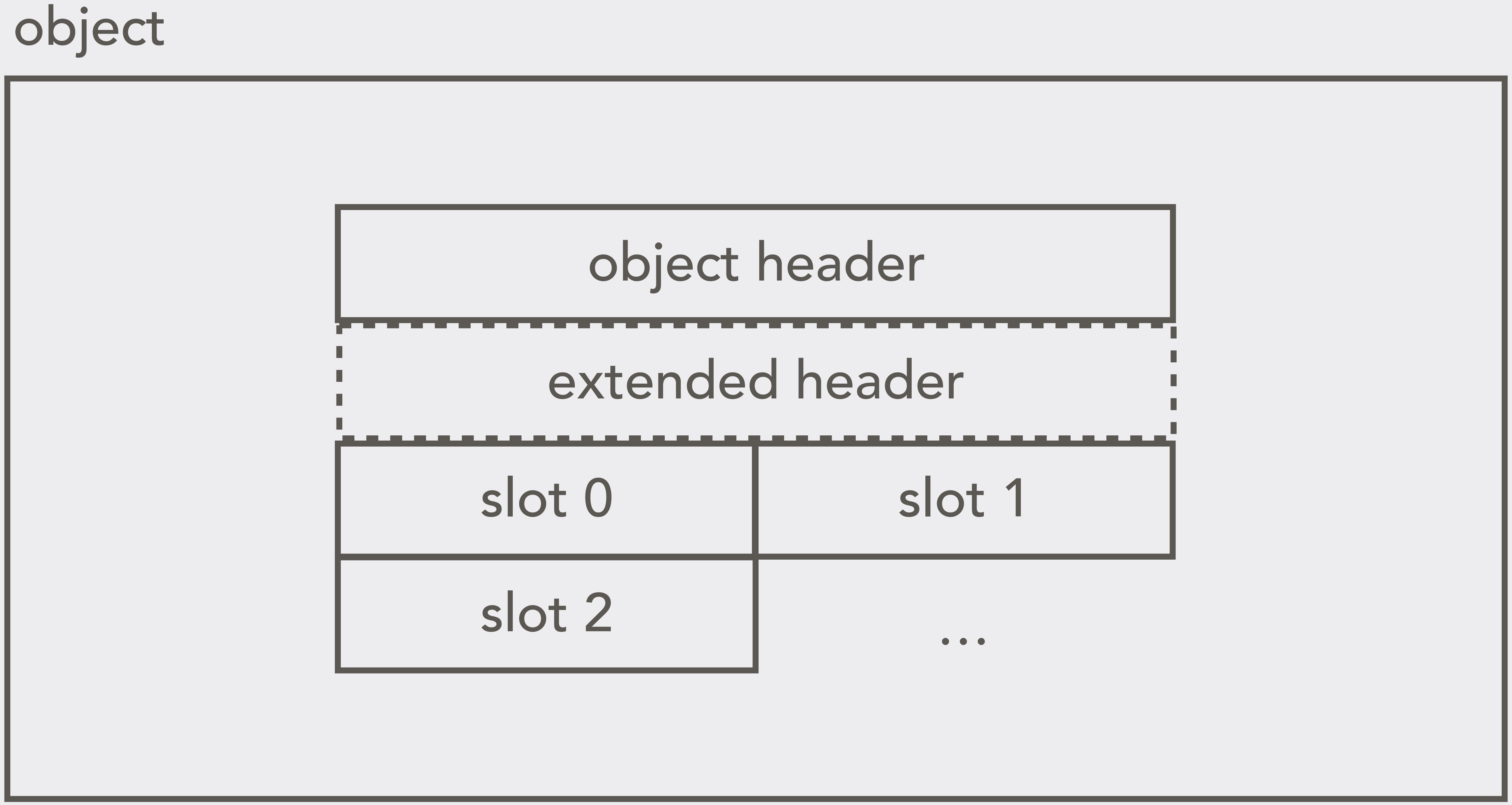
identity hash

garbage collection

format

unused

class index



object

object header

extended header

slot 0

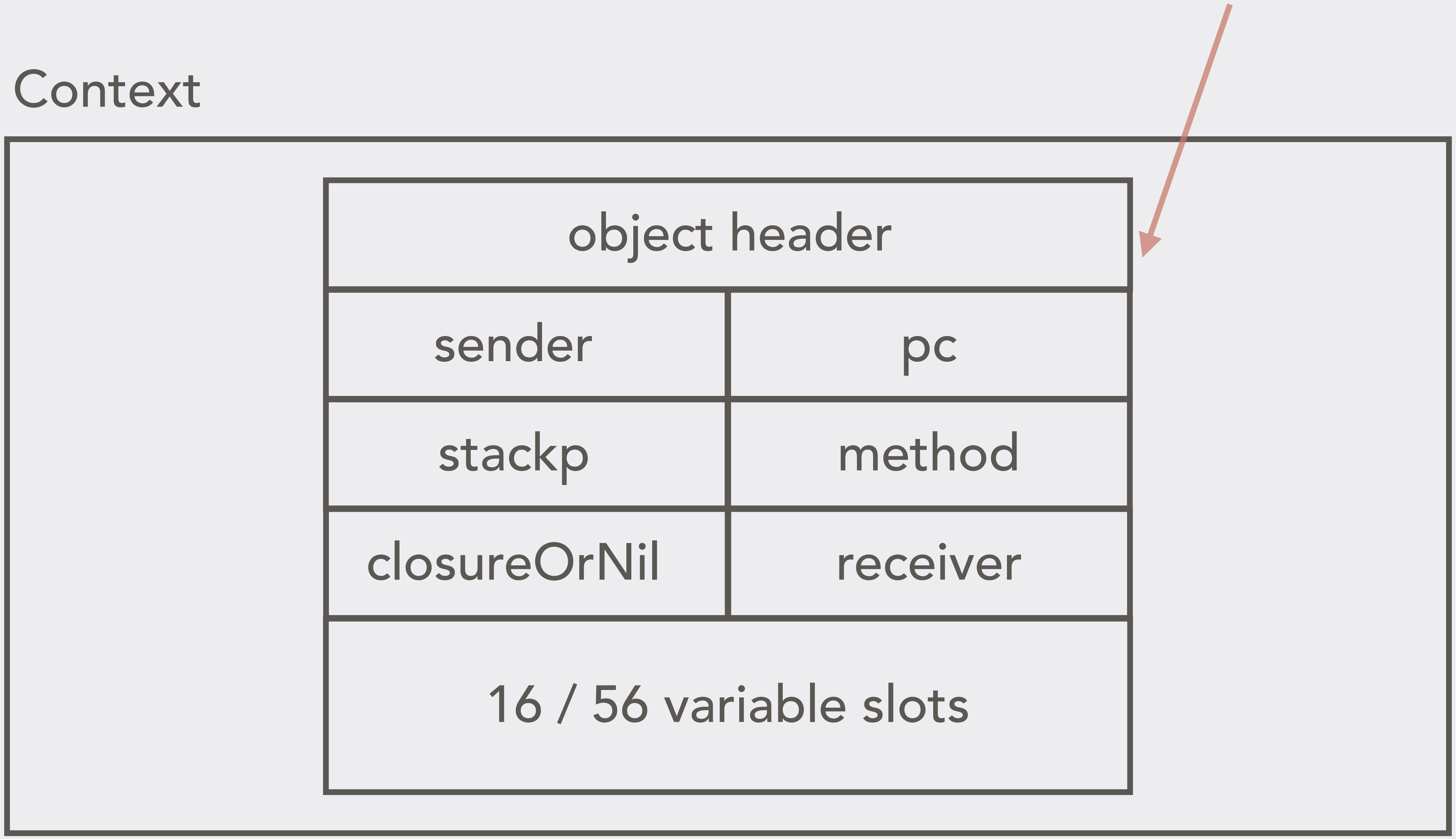
slot 1

slot 2

...

no extended header required

Context



object header

sender

pc

stackp

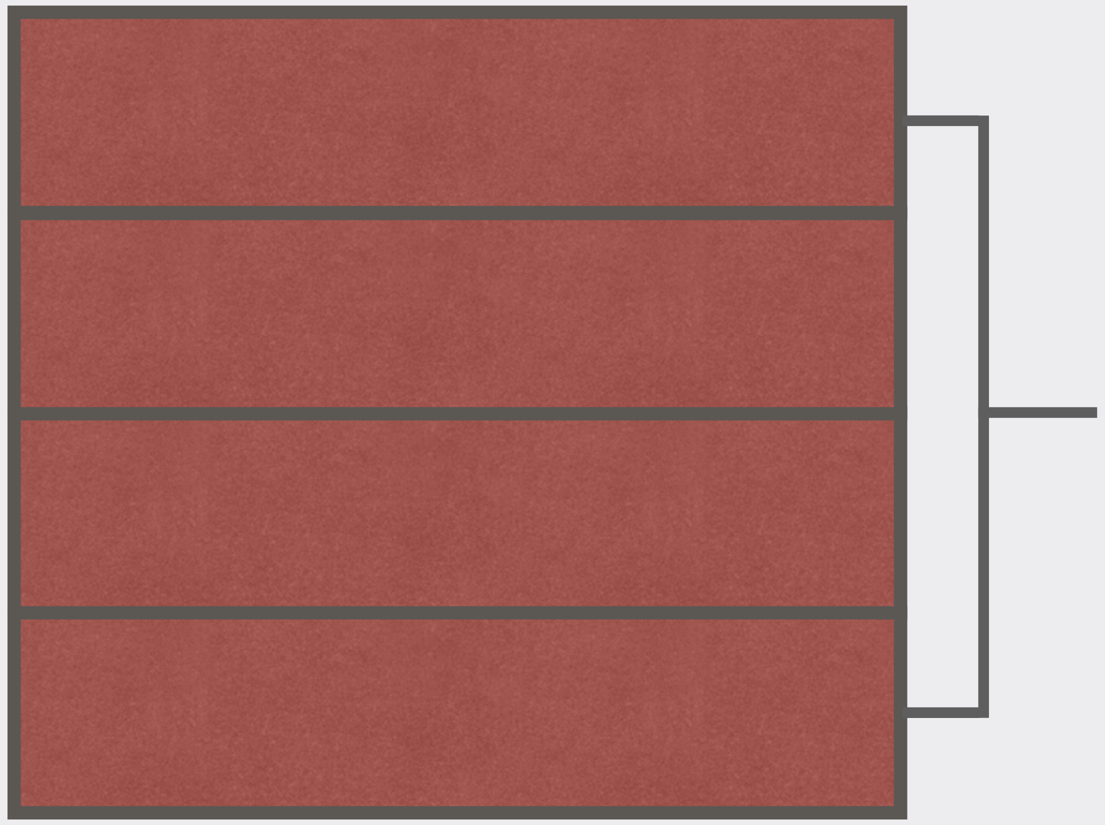
method

closureOrNil

receiver

16 / 56 variable slots

96 / 256 bytes per instance



$\sum \ll 100\,000 \sim 1000$

estimated upper bounds

large contexts: 256 kB

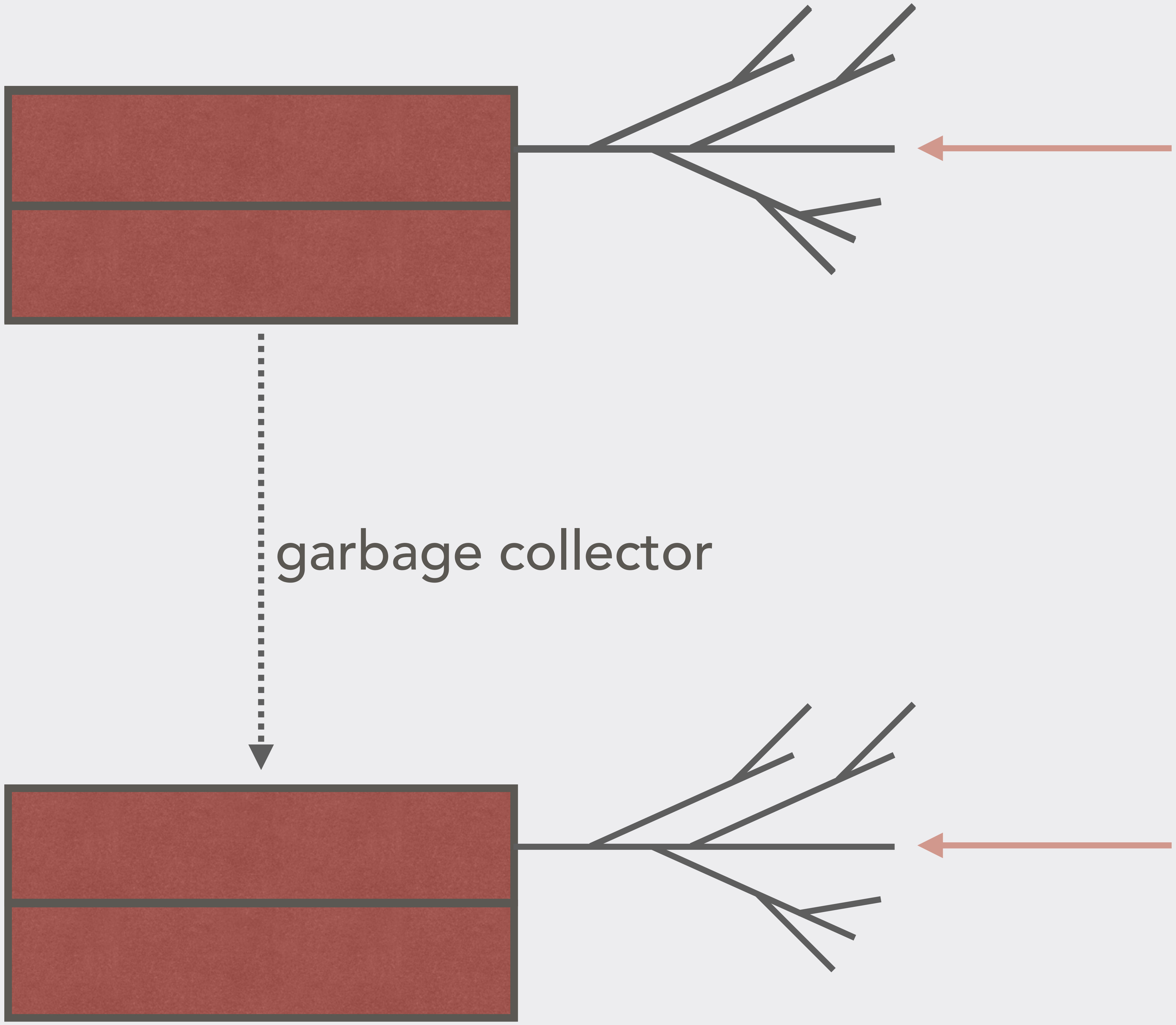
small contexts: 96 kB

* 2 (reification)

large contexts: 512 kB

small contexts: 192 kB

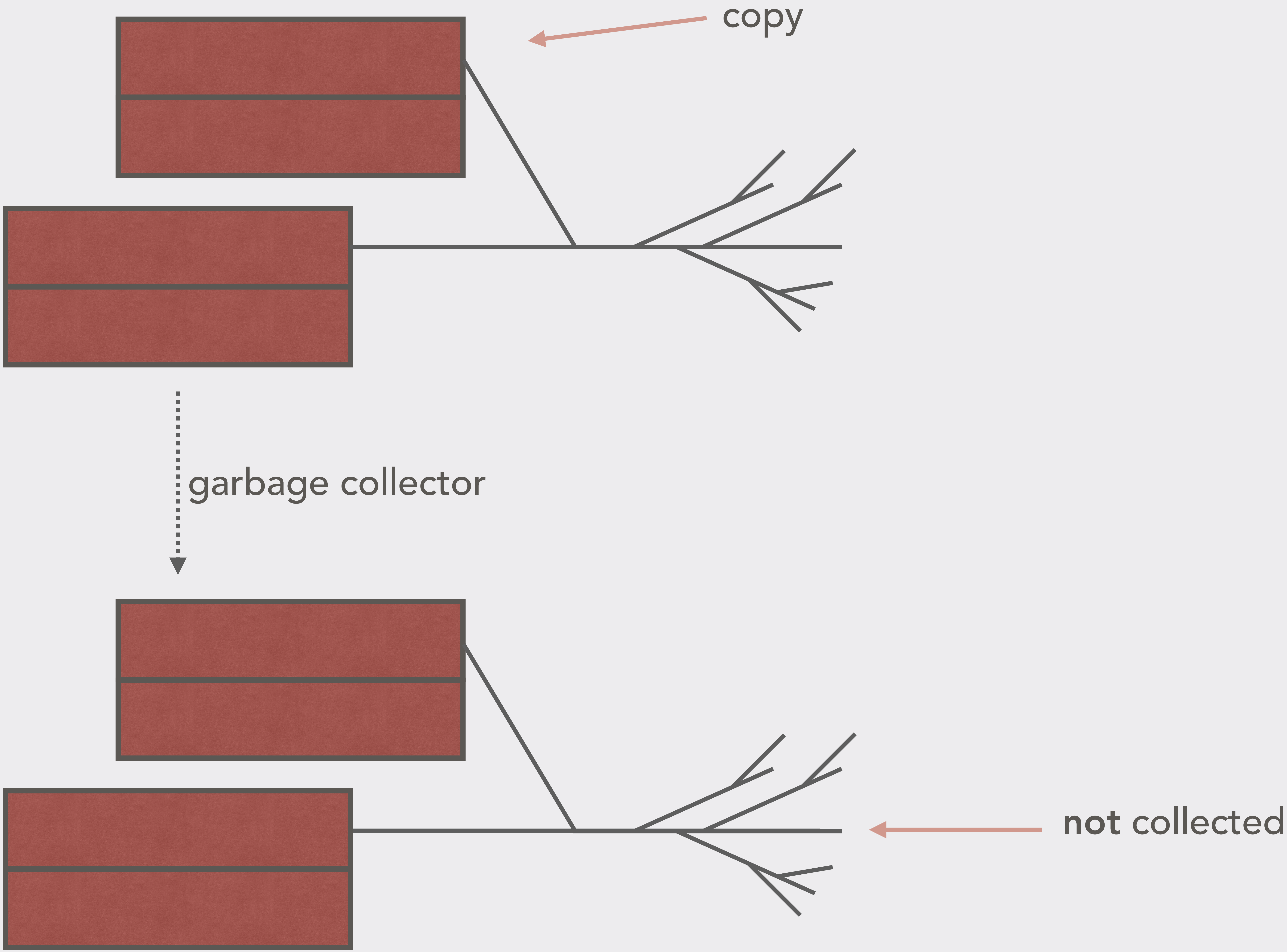
memory consumption of contexts: ✓



object graph (simplified)

garbage collector

partially collected


copy
garbage collector
not collected

memory consumption of object graph: ?

PERFORMANCE

	small context (96 bytes)			large context (256 bytes)		
garbage collection time	average	median	max	average	median	max
included	130.211	130	167	130.148	130	152
excluded	57.127	57	71	57.112	57	66

copying stack of 100 000 frames

performance: ✓

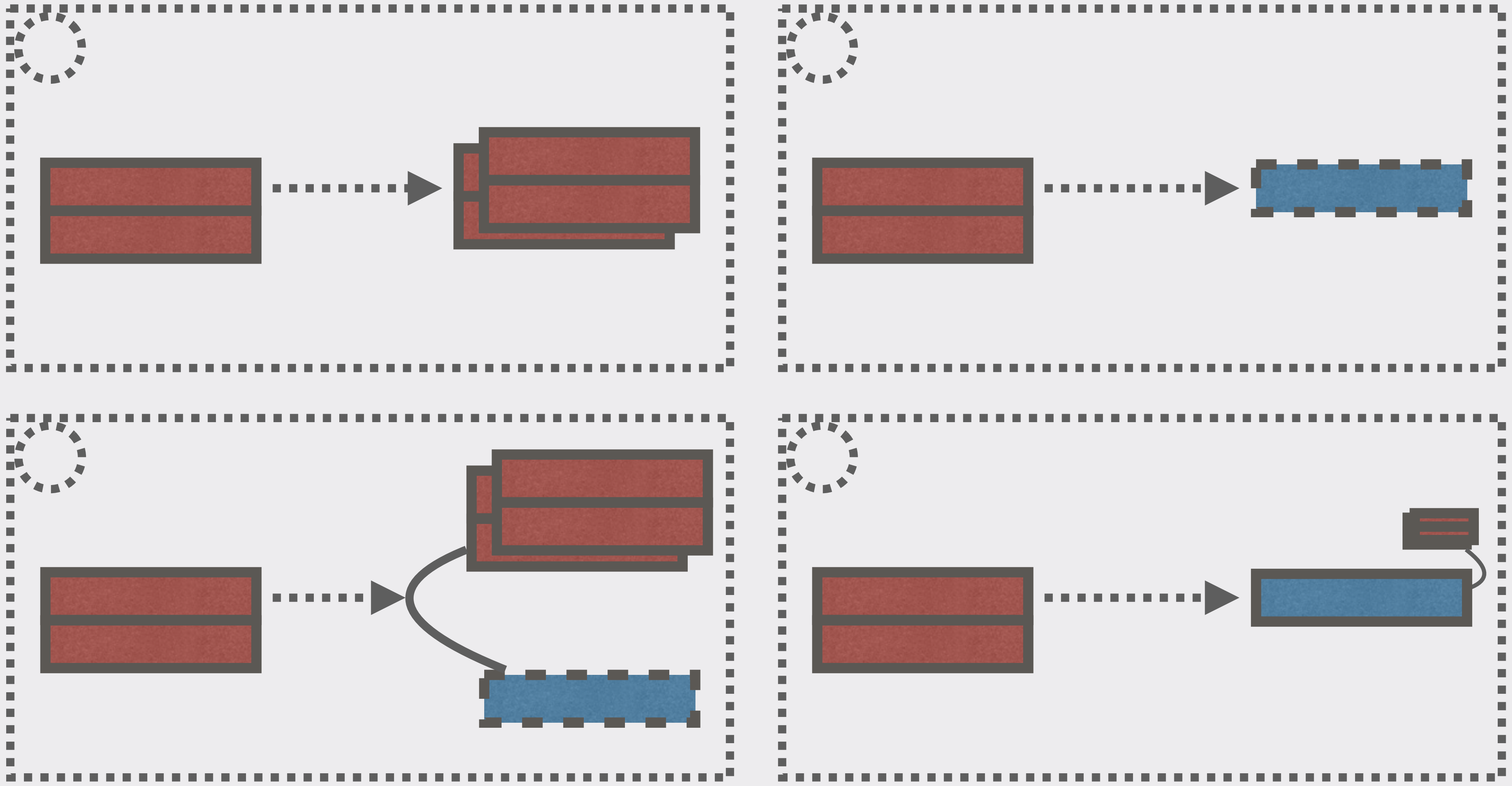
FUTURE WORK

VM support

memory consumption

threads -> processes

user interface



memory consumption
of contexts: ✓

performance: ✓

memory consumption
of object graph: ?