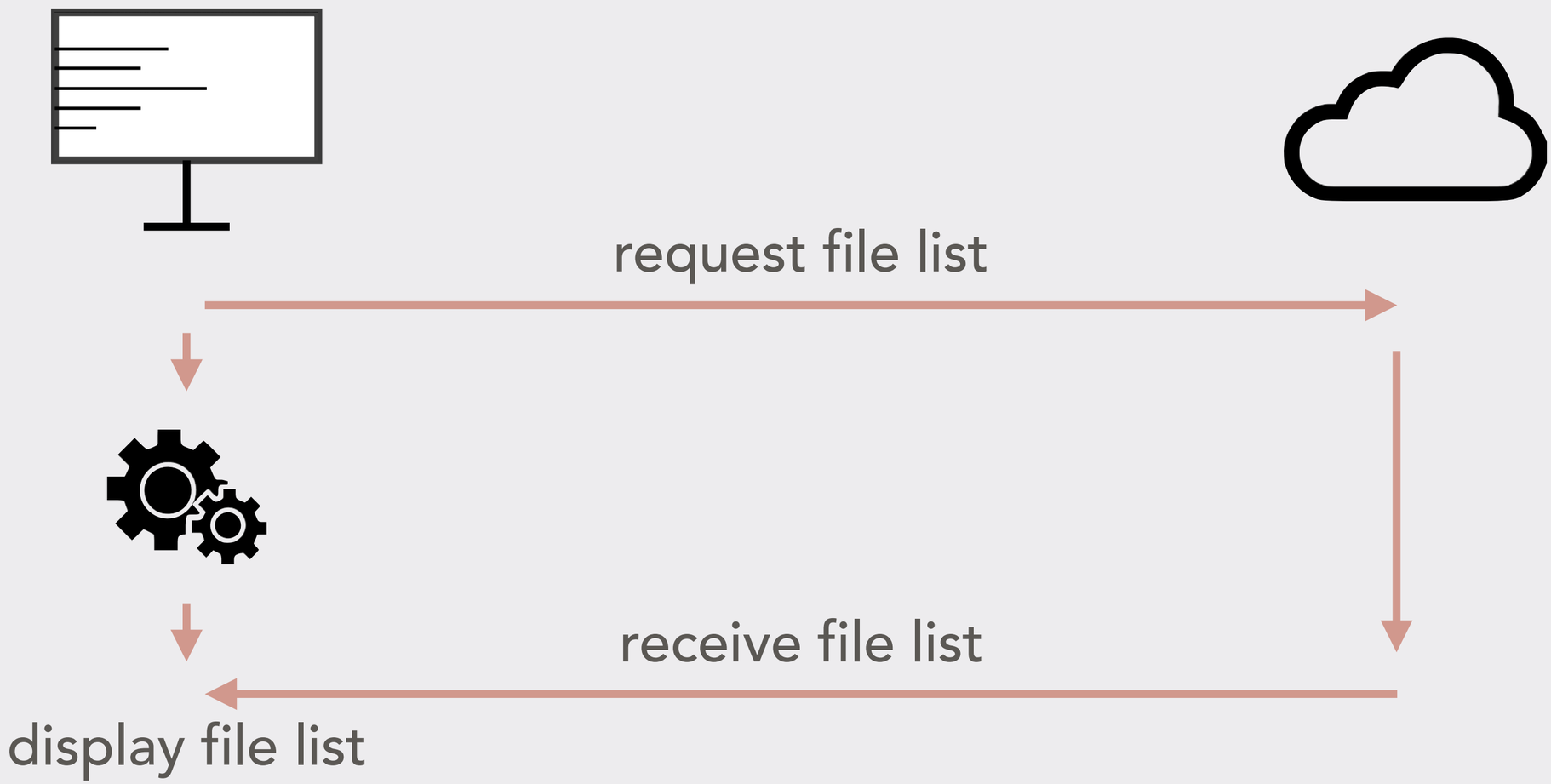
A PROMISING APPROACH TO DEBUGGING REMOTE PROMISES

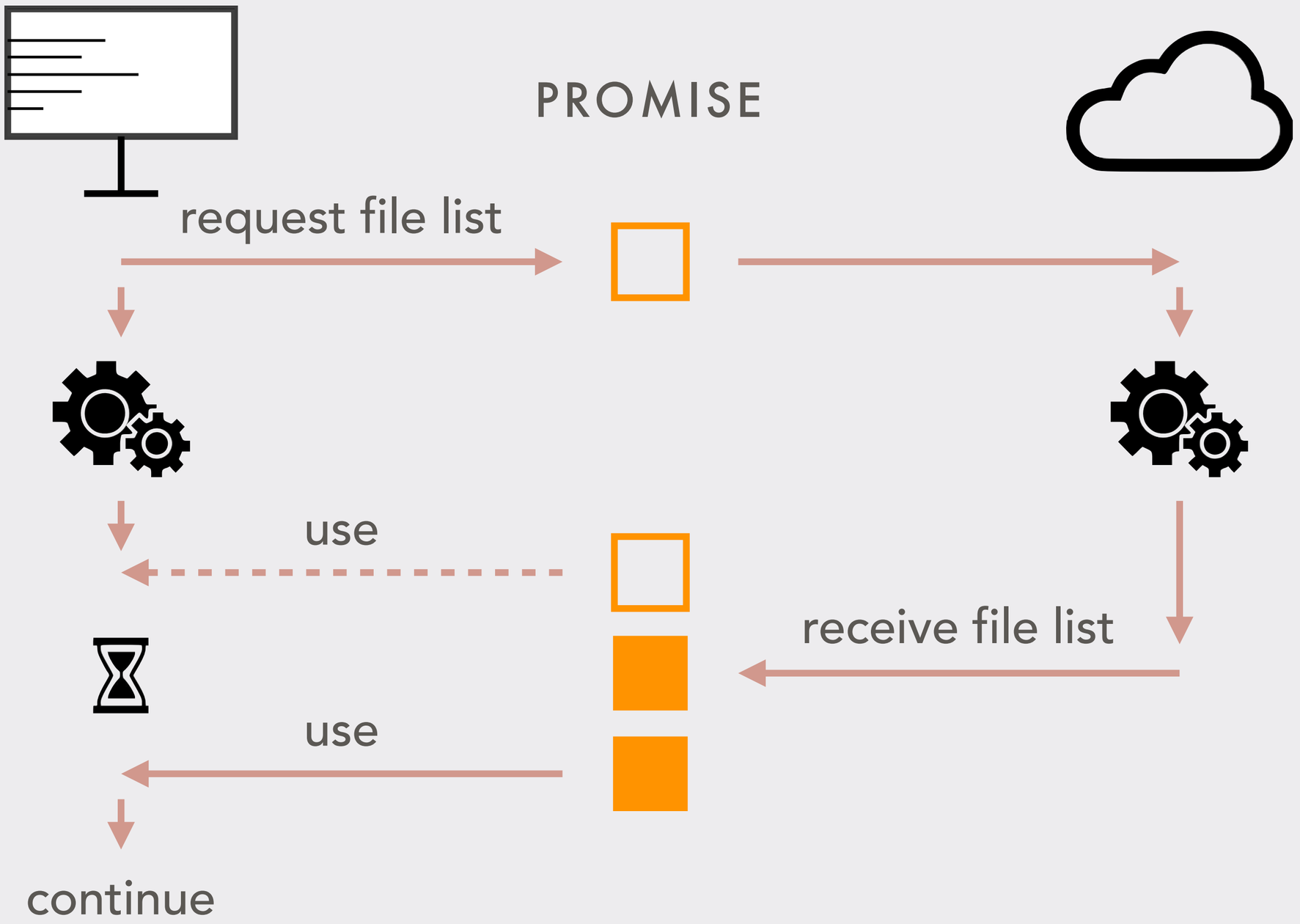
2016
Max Leske
Andrei Chiș
Oscar Nierstrasz

DISCLAIMER

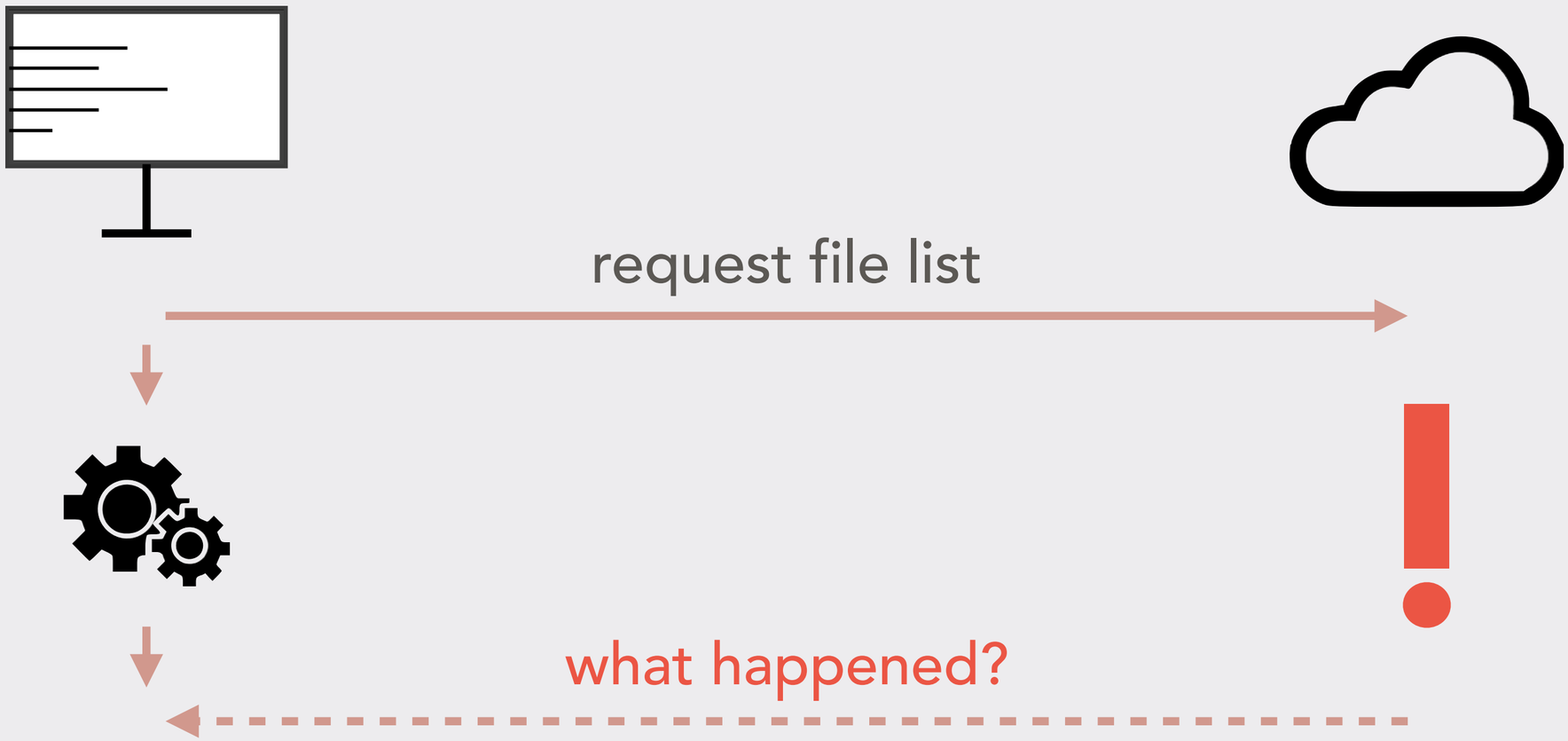
MOTIVATION

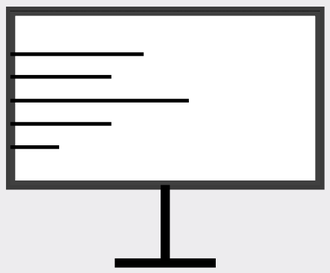

request file list
receive file list
display file list


request file list
receive file list
display file list


PROMISE
request file list
use
use
receive file list
continue

```
fileListPromise := [ self getFileList ] promise.
...
fileListPromise value.
```


request file list
what happened?



what happened?

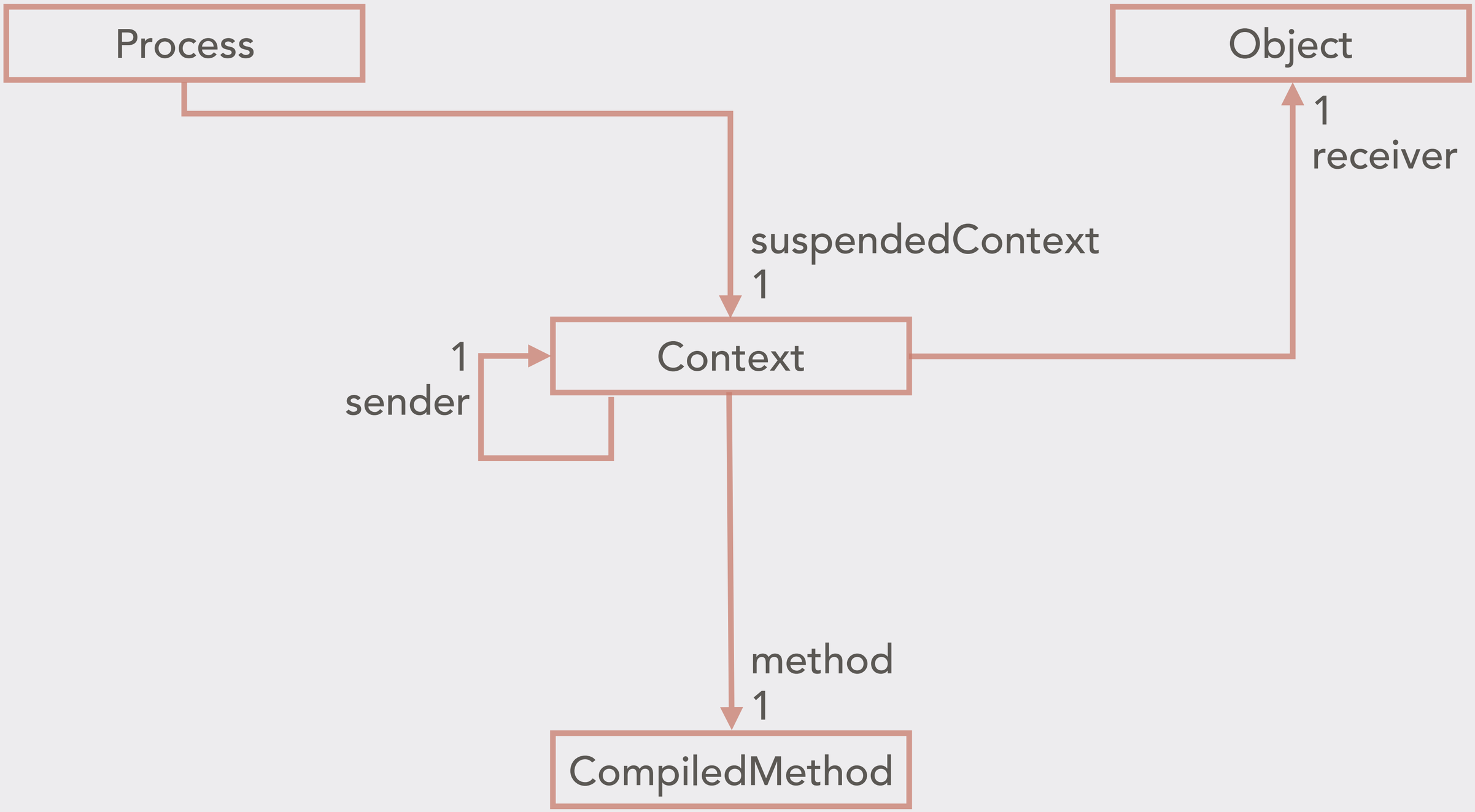
~~NULL~~

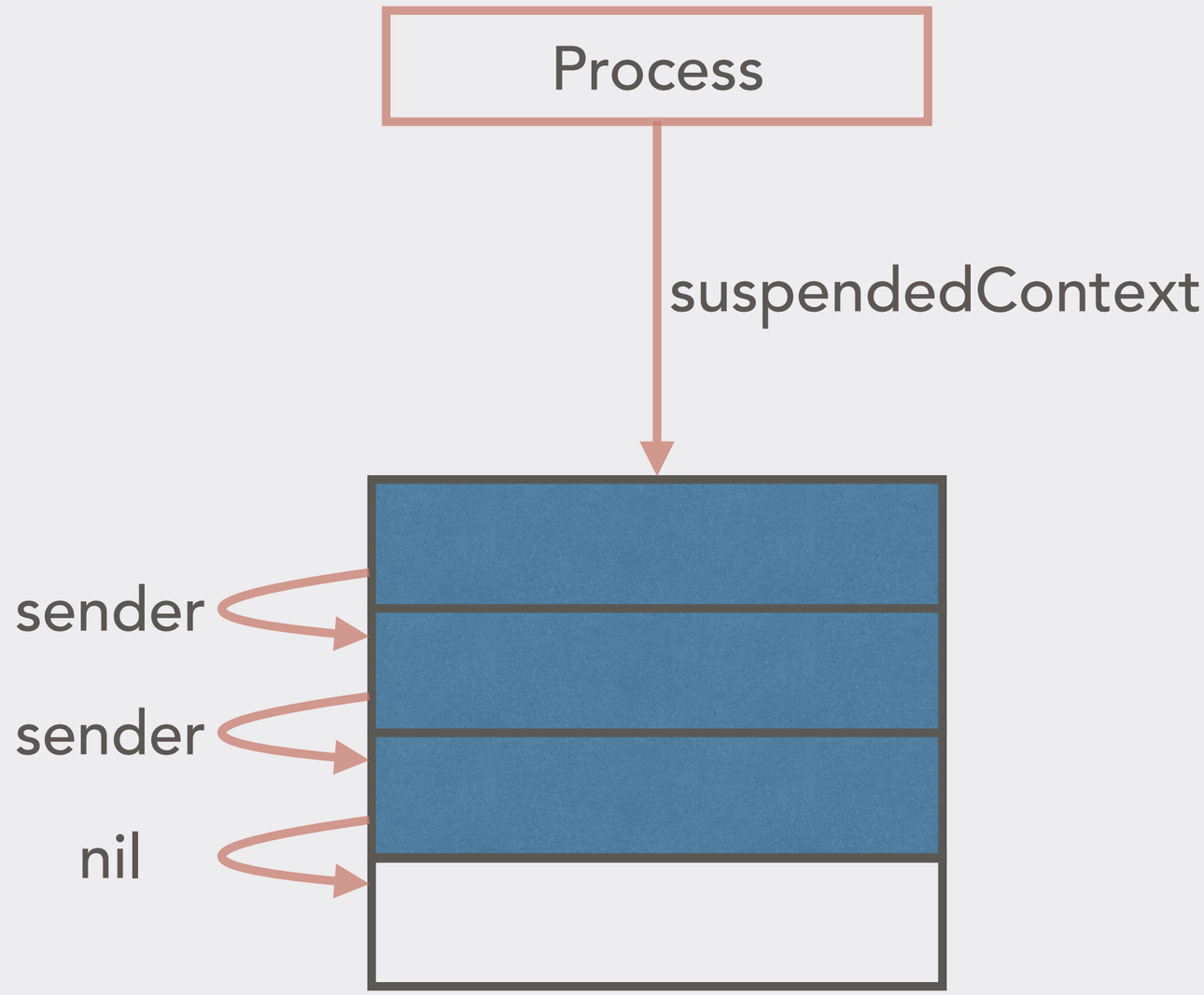
~~description~~

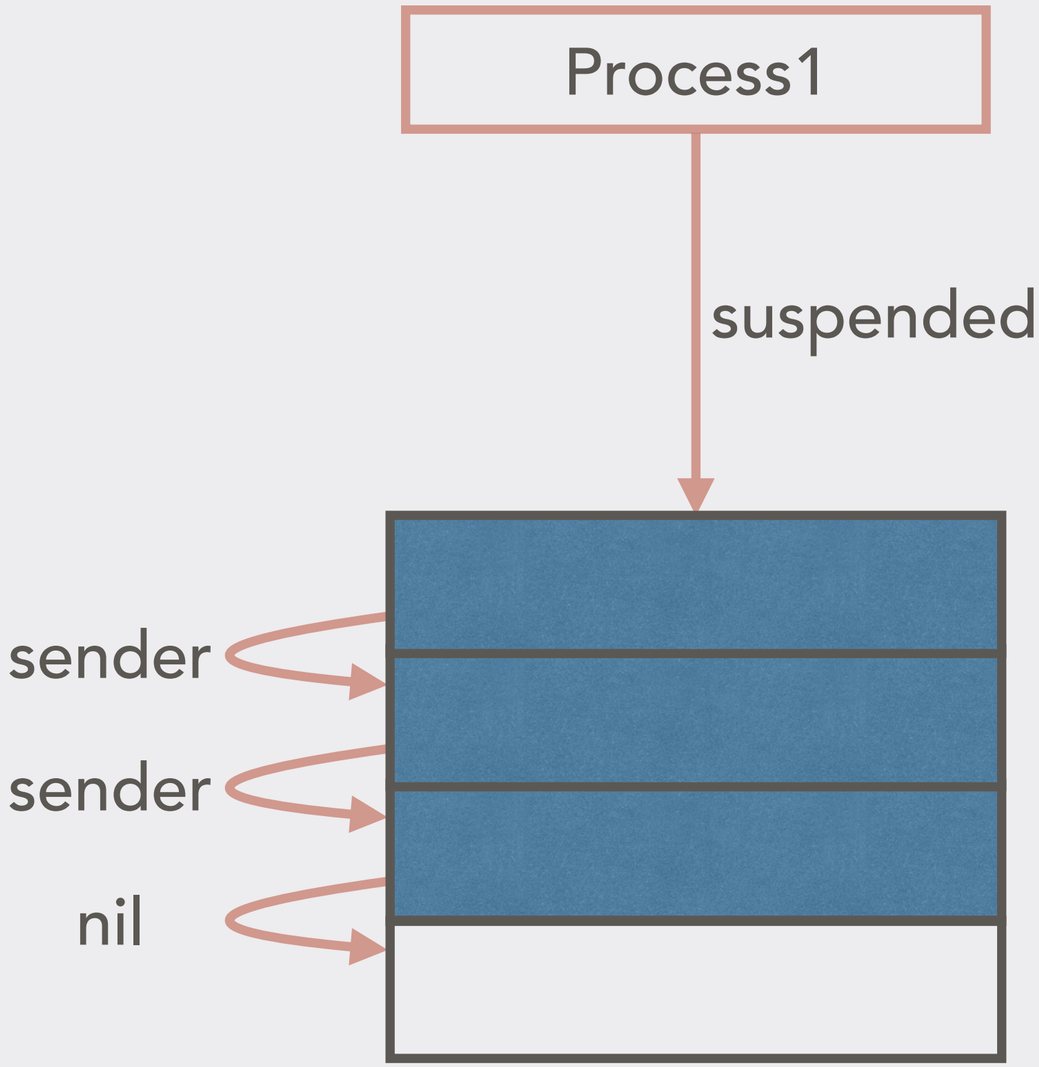
~~stack trace~~

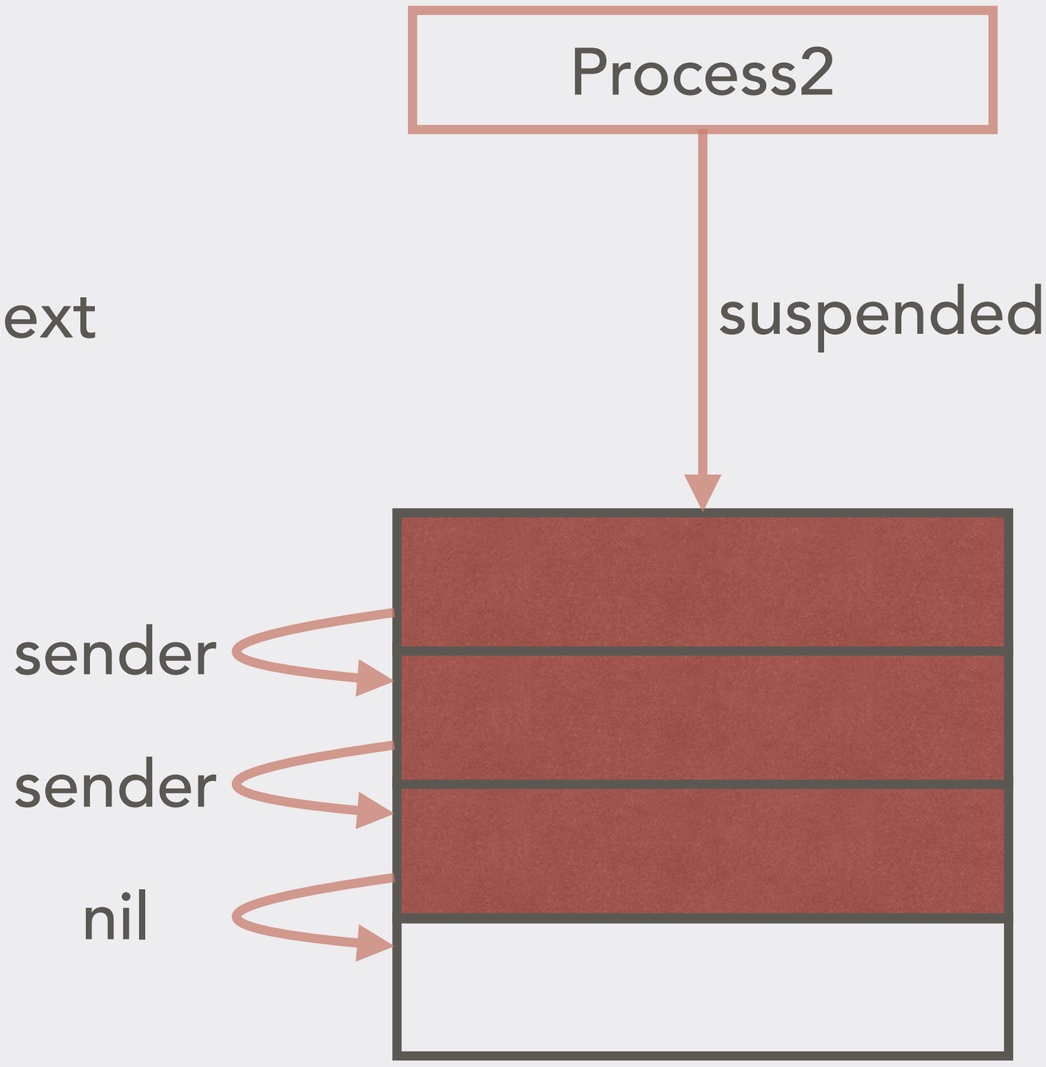
exception object

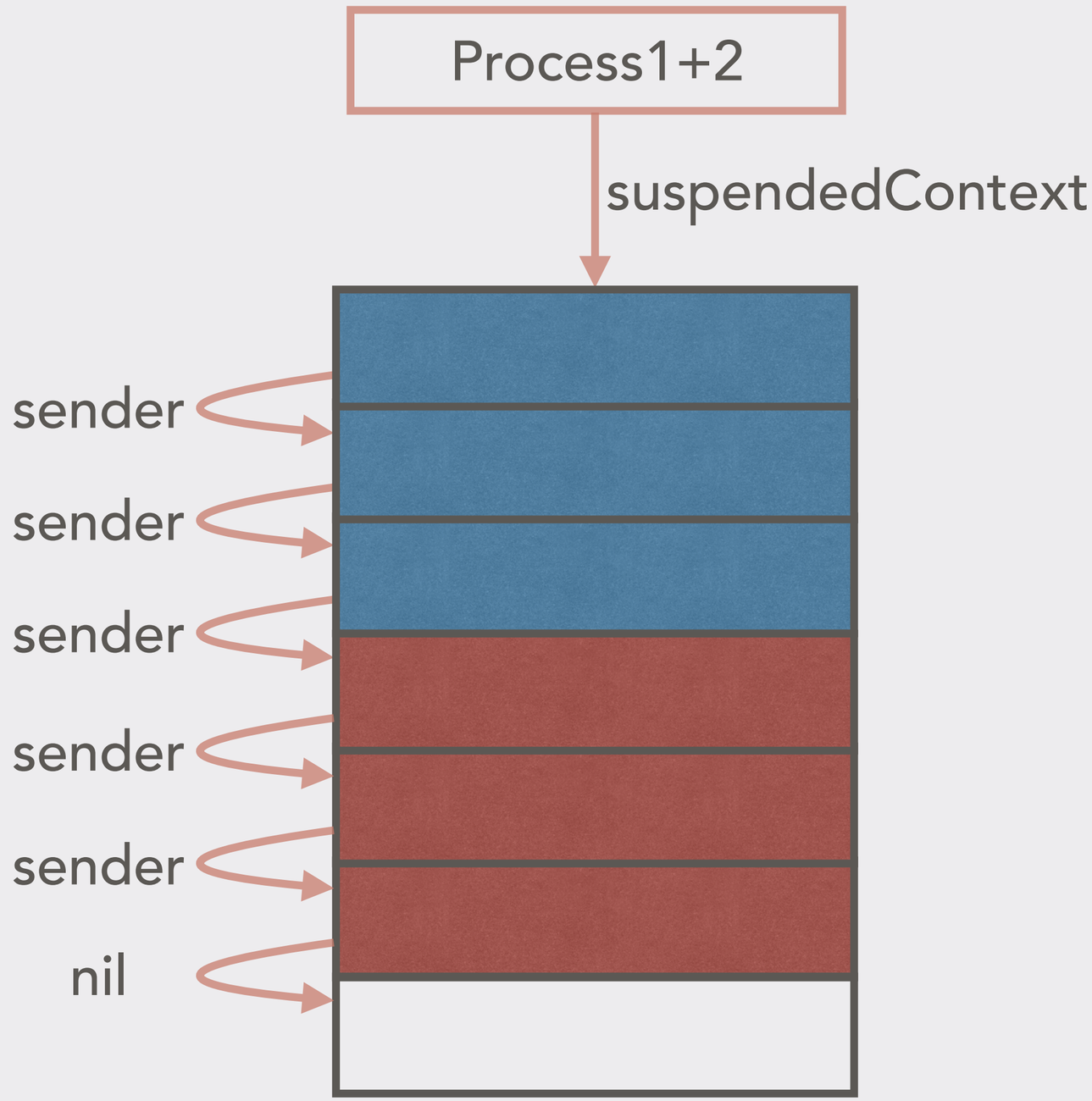
IDEA


Process
Object
1
receiver
suspendedContext
1
1
sender
Context
method
1
CompiledMethod


Process
suspendedContext
sender
sender
nil

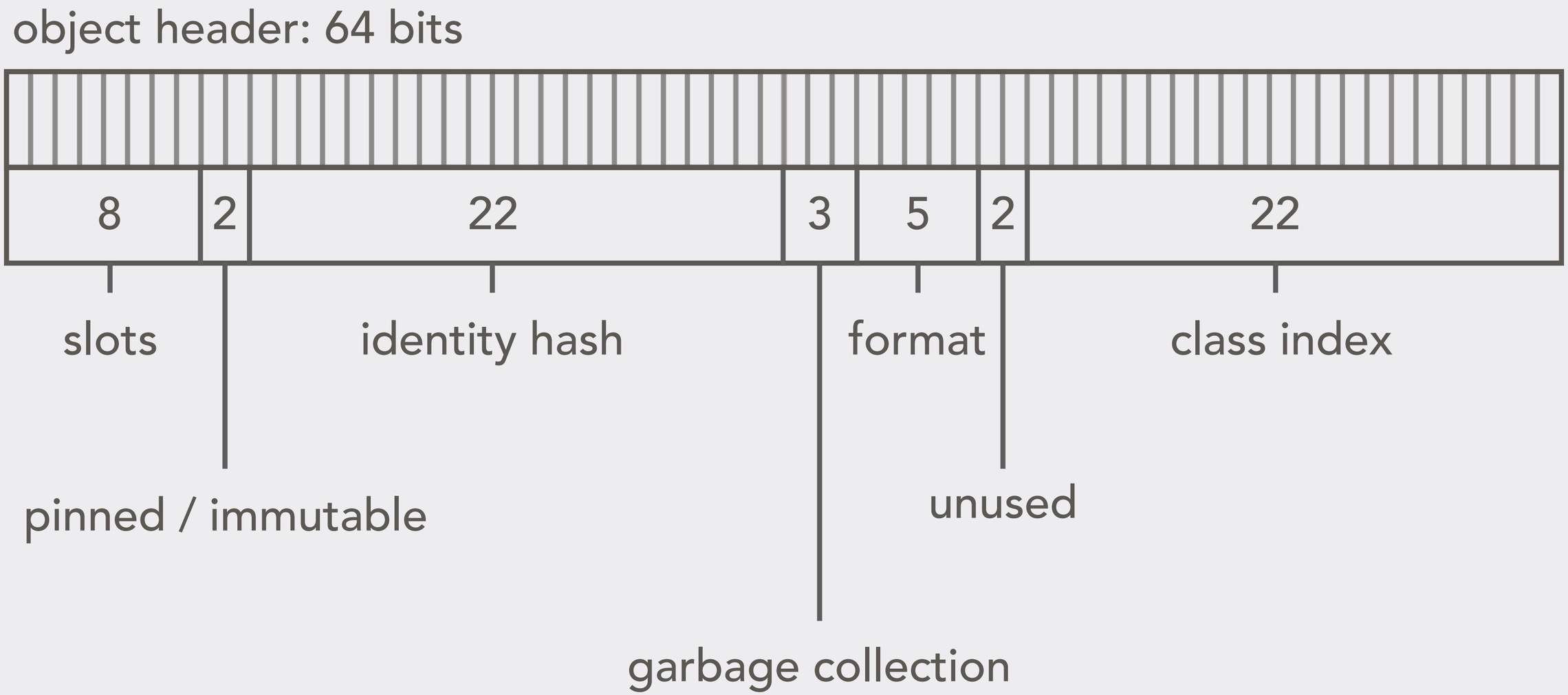

Process1
suspendedContext
sender
sender
nil

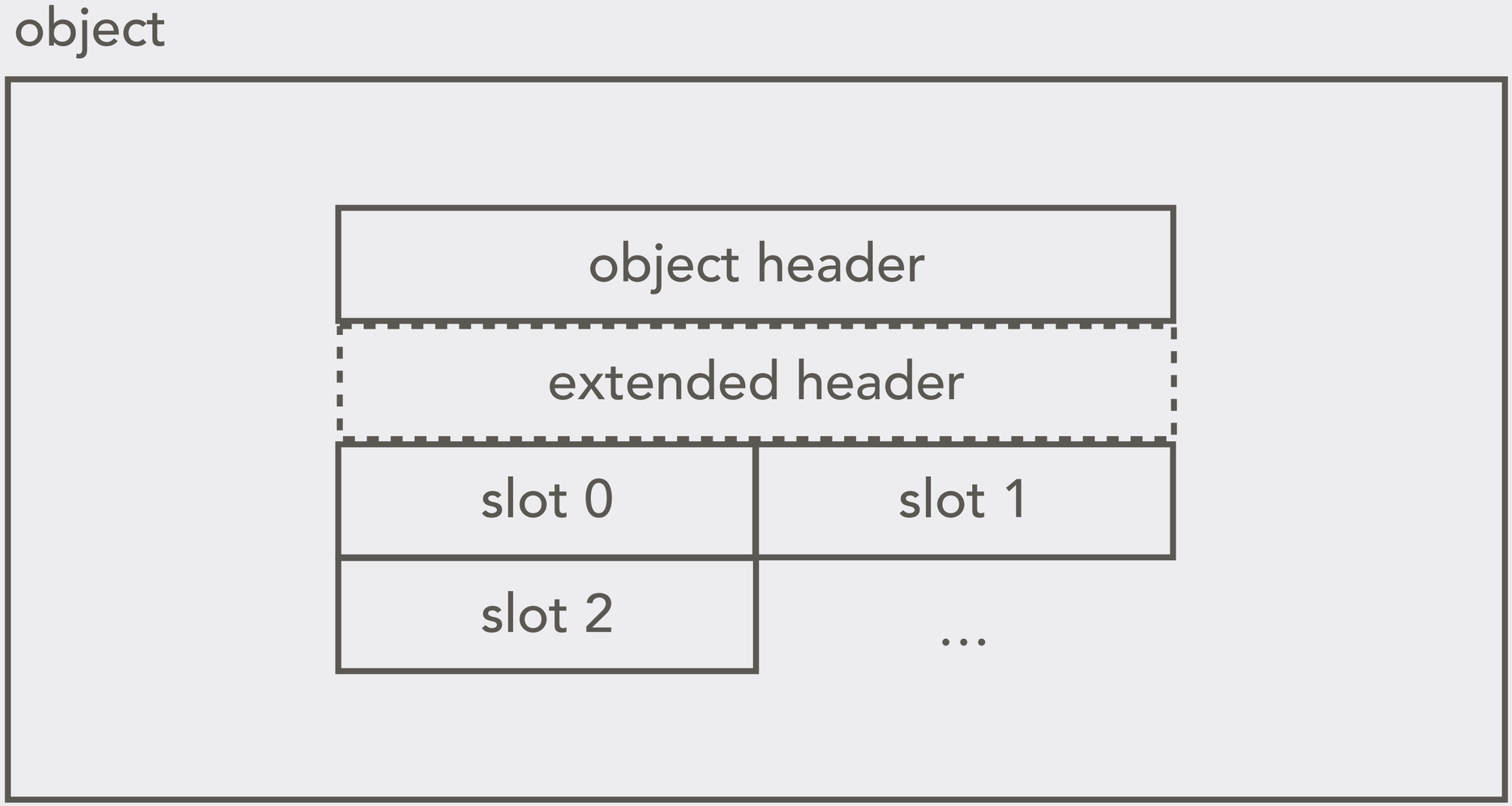

Process2
suspendedContext
sender
sender
nil

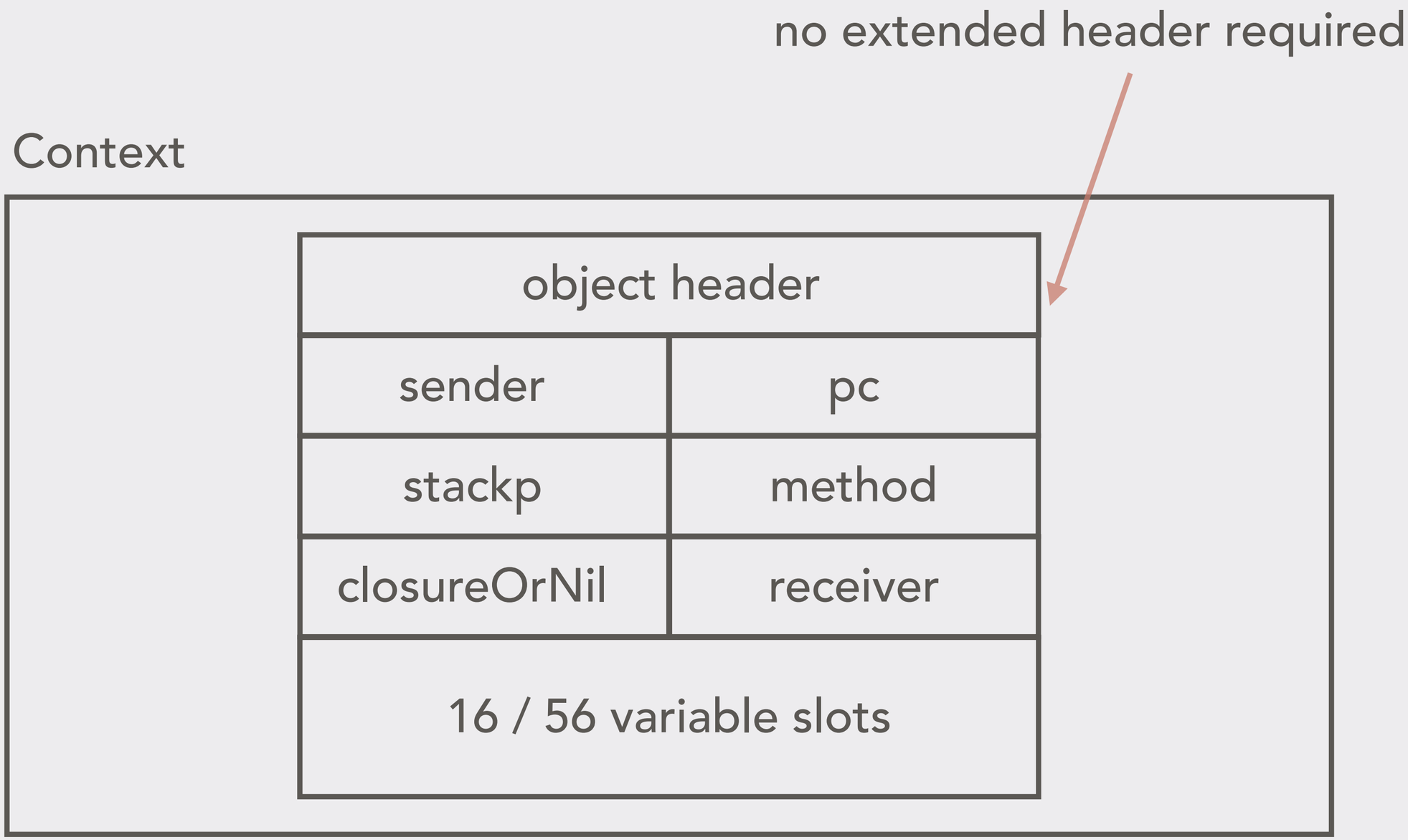

Process1+2
suspendedContext
sender
sender
sender
sender
sender
nil

DEMO

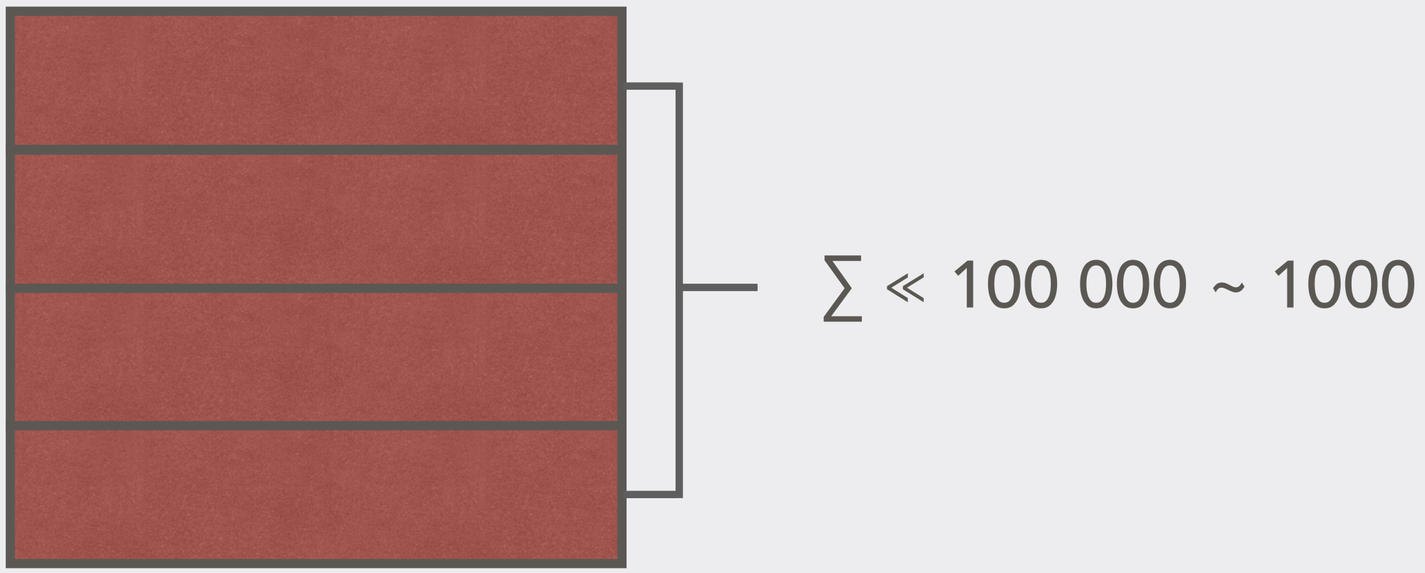
MEMORY


object header: 64 bits
8
2
22
3
5
2
22
slots
identity hash
format
class index
pinned / immutable
garbage collection
unused


object
object header
extended header
slot 0
slot 1
slot 2
...



96 / 256 bytes per instance



estimated upper bounds

large contexts: 256 kB

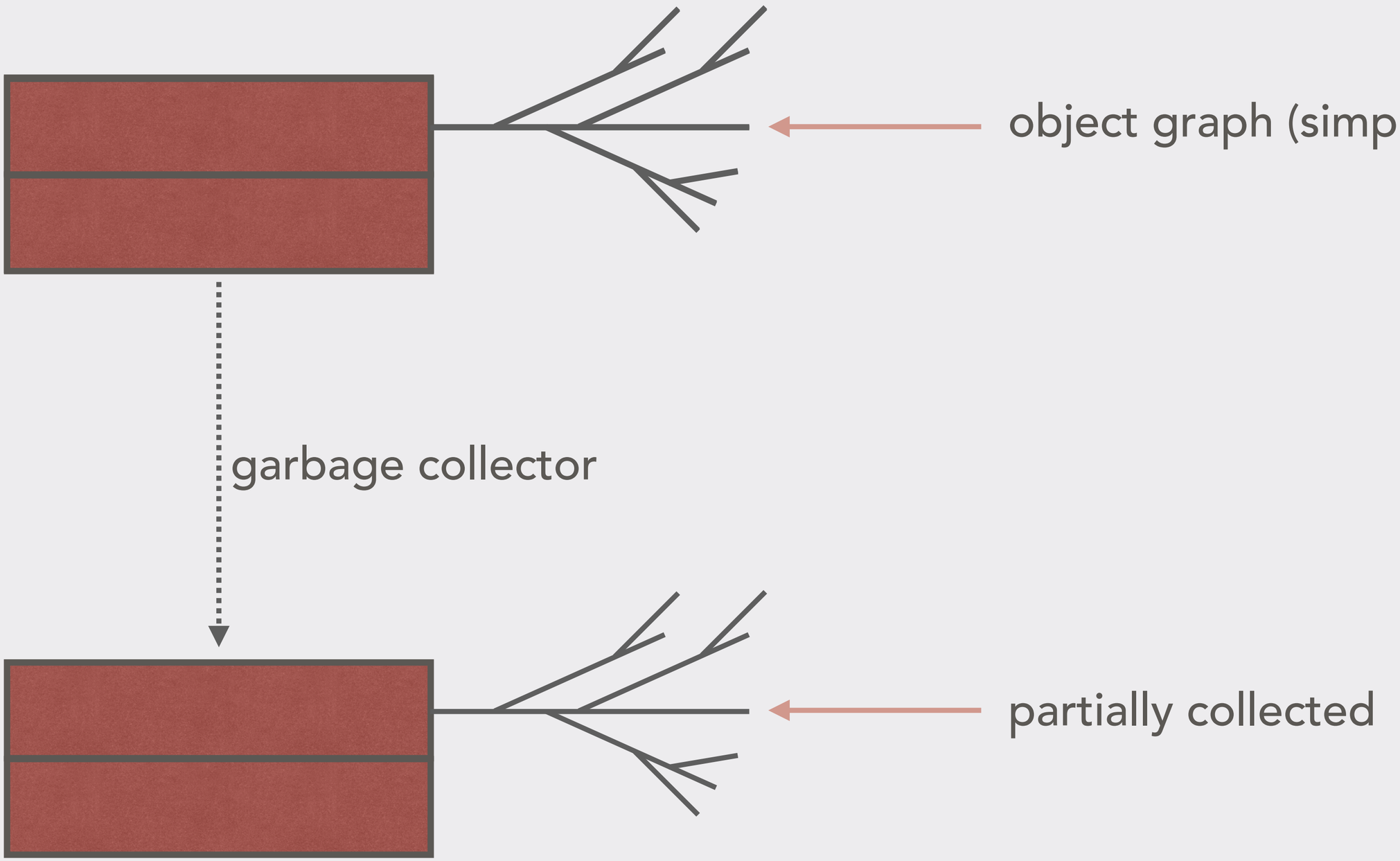
small contexts: 96 kB

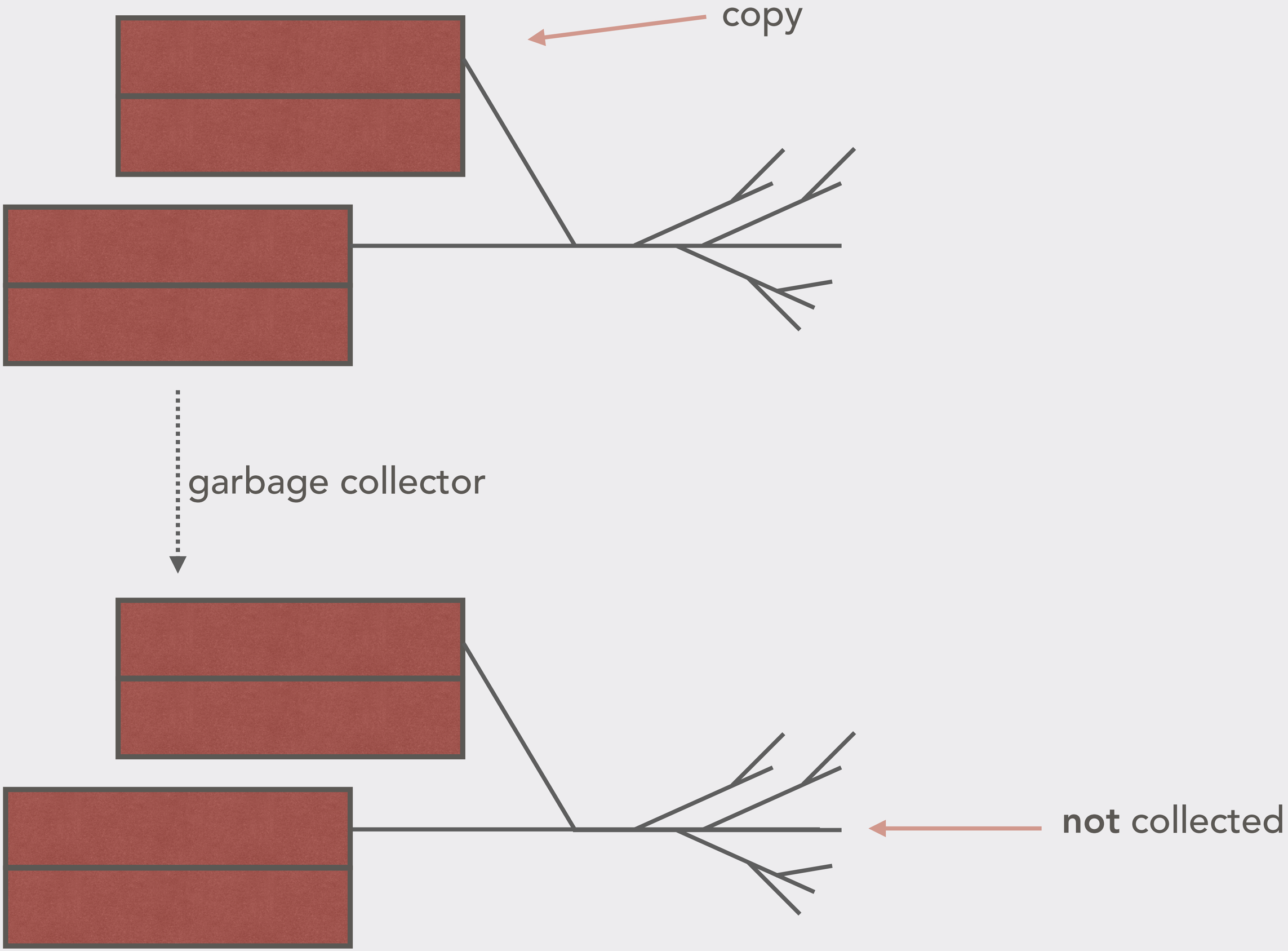
* 2 (reification)

large contexts: 512 kB

small contexts: 192 kB

memory consumption of contexts:


object graph (simplified)
garbage collector
partially collected


copy
garbage collector
not collected

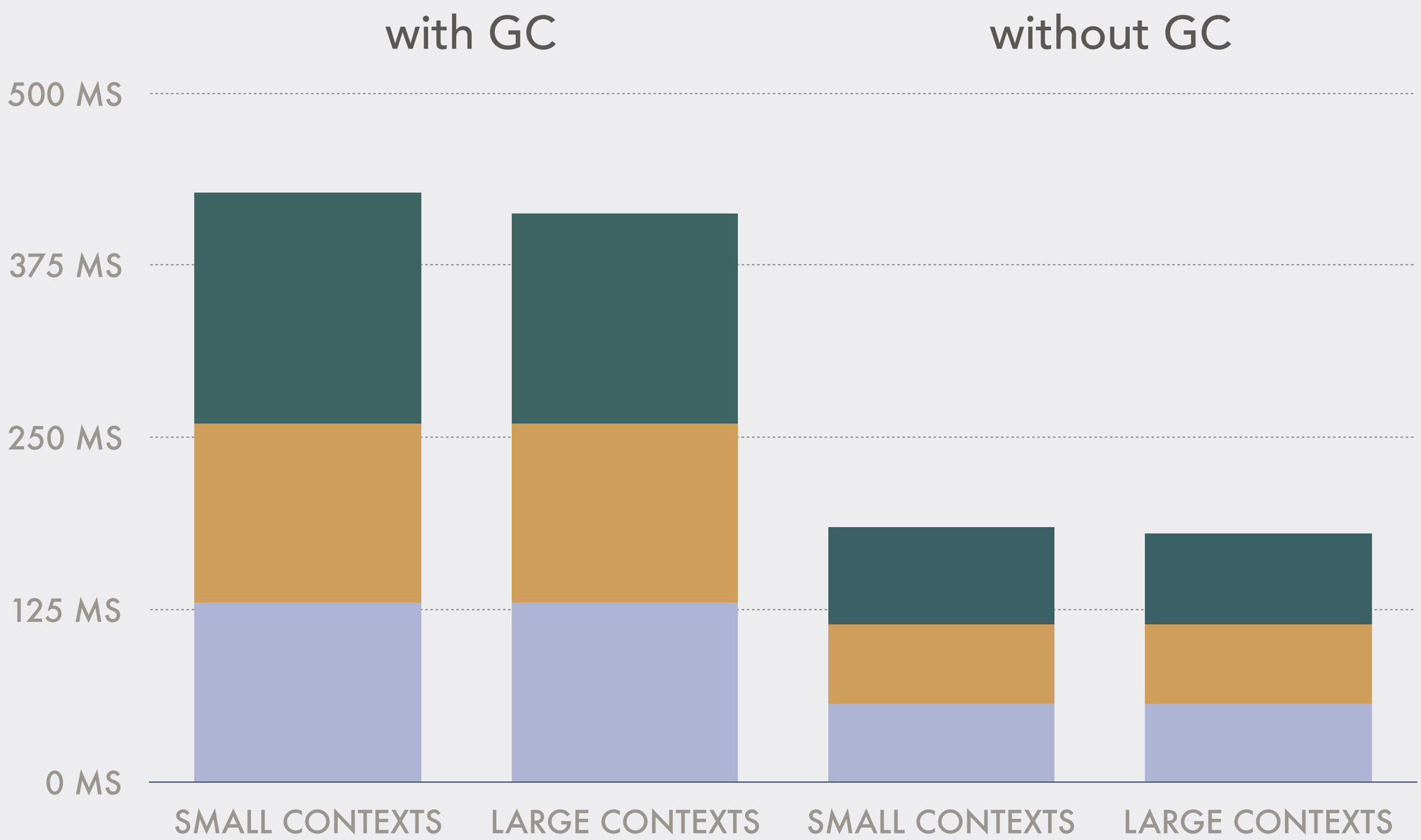
memory consumption of object graph: ?

PERFORMANCE

```
fileListPromise := [ self getFileList ] promise.
```

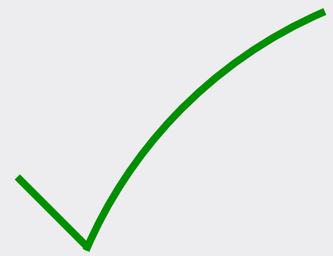

```
...

fileListPromise value.
```



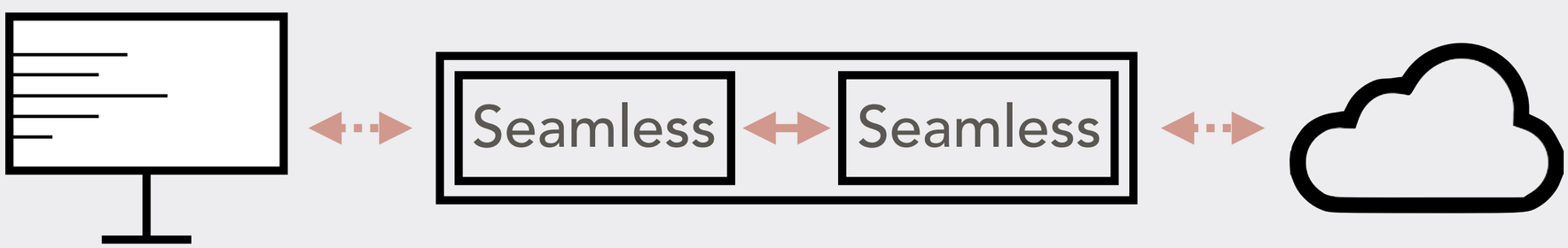
copying stack of 100 000 frames

performance:

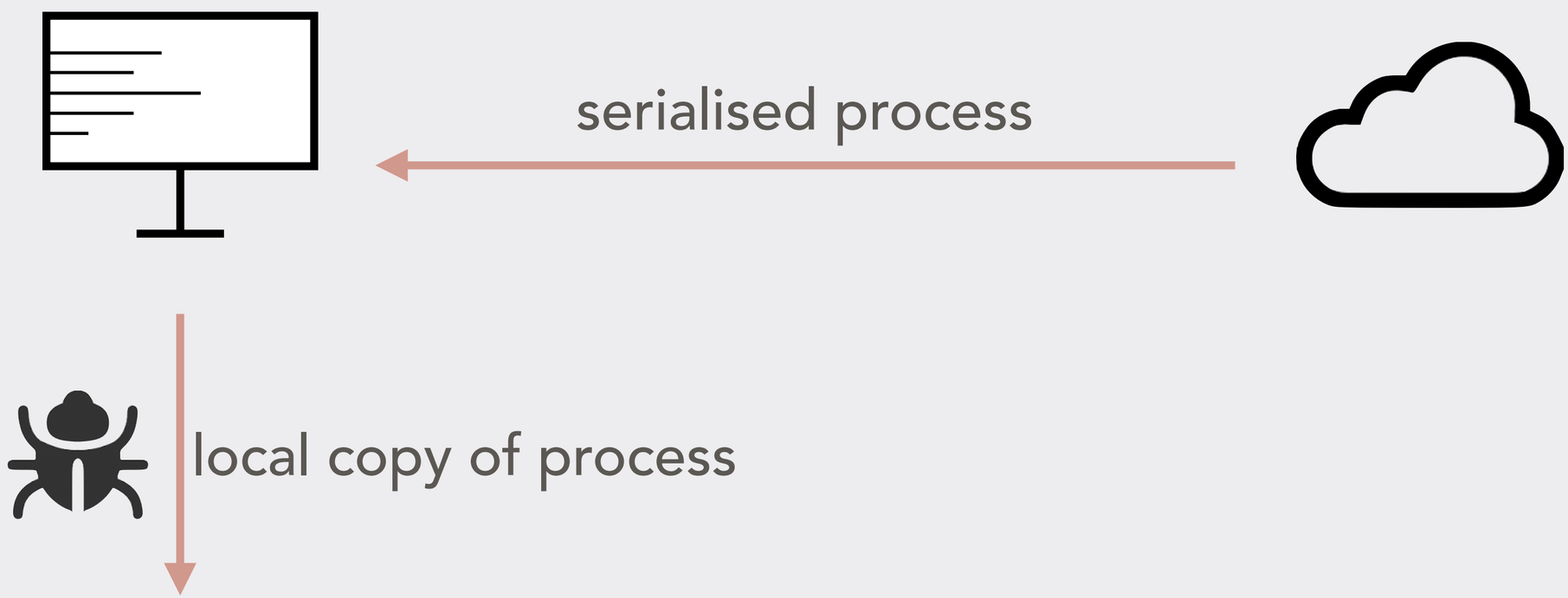


REMOTE COMMUNICATION


protocol?


Seamless
Seamless

DEBUGGING



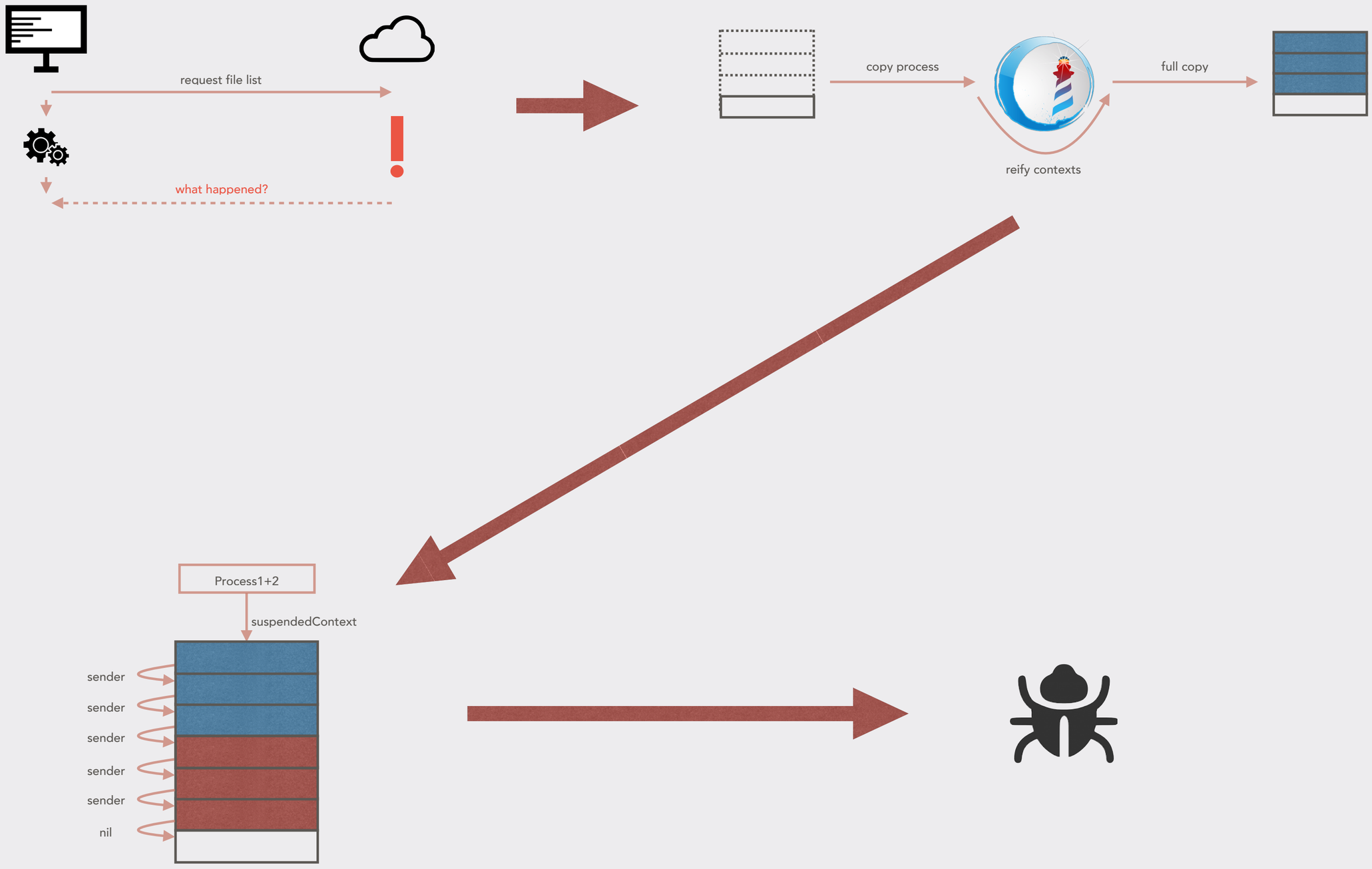
asynchronous network requests

promises

THREADS

asynchronous messages (actors)

events


request file list
what happened?
copy process
full copy
reify contexts
Process1+2
suspendedContext
sender
sender
sender
sender
sender
nil