

DEBUGGING ASYNCHRONOUS PROCESSES

2016

Max Leske

Andrei Chiş

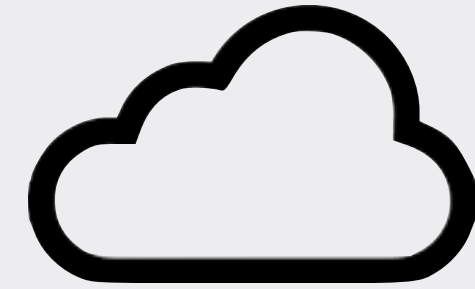
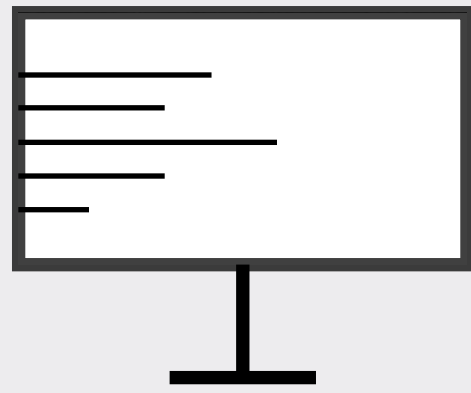
motivation

demo

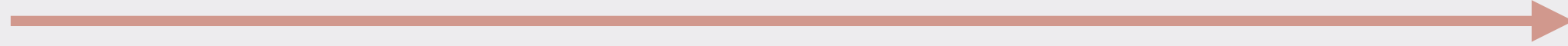
how it works

future work

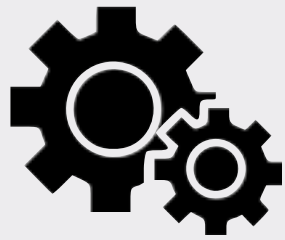
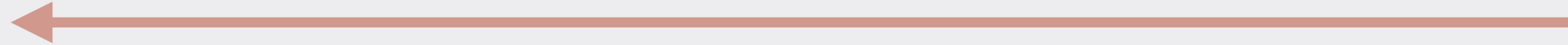
**WORKING IN
PARALLEL**



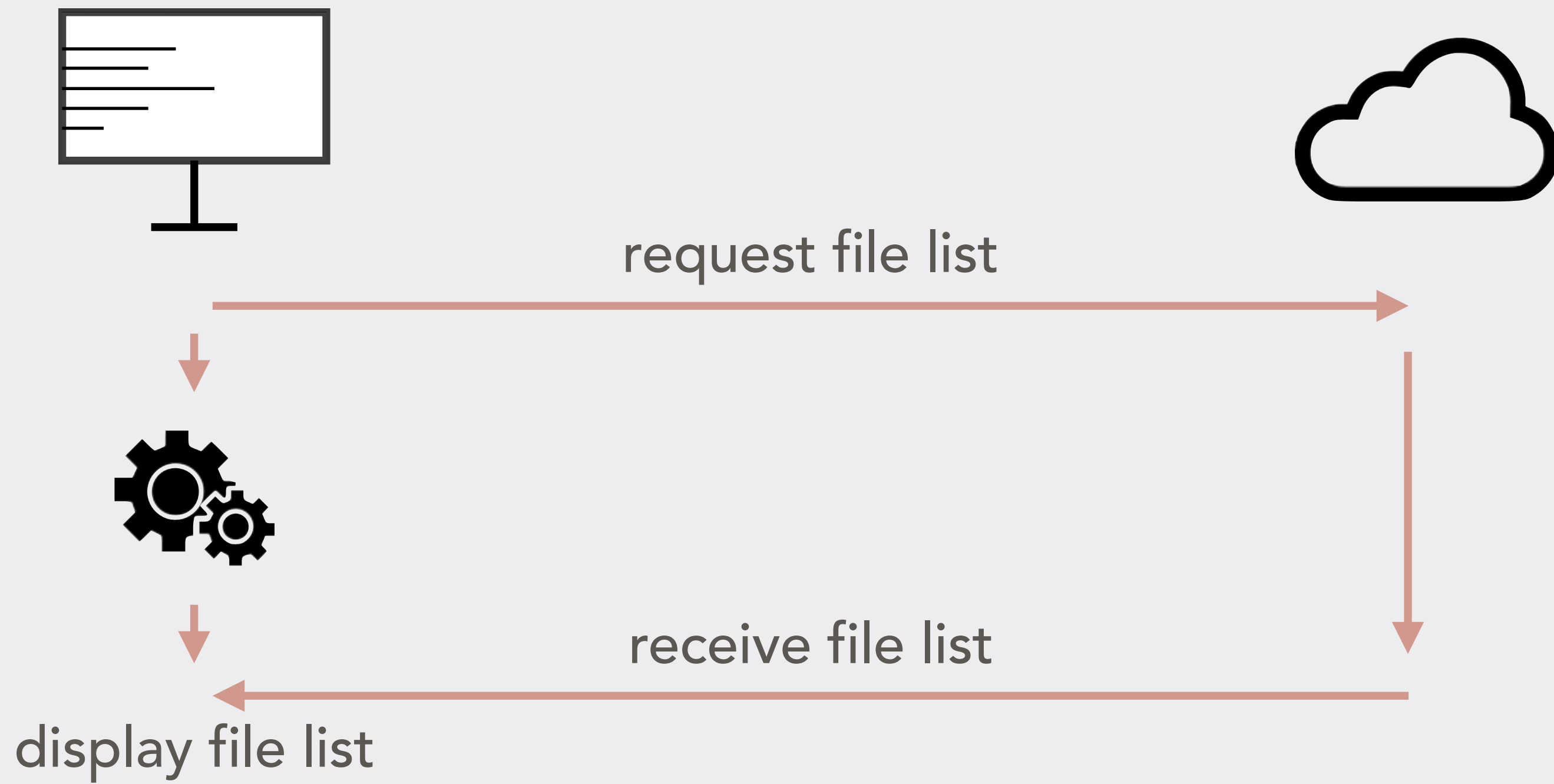
request file list

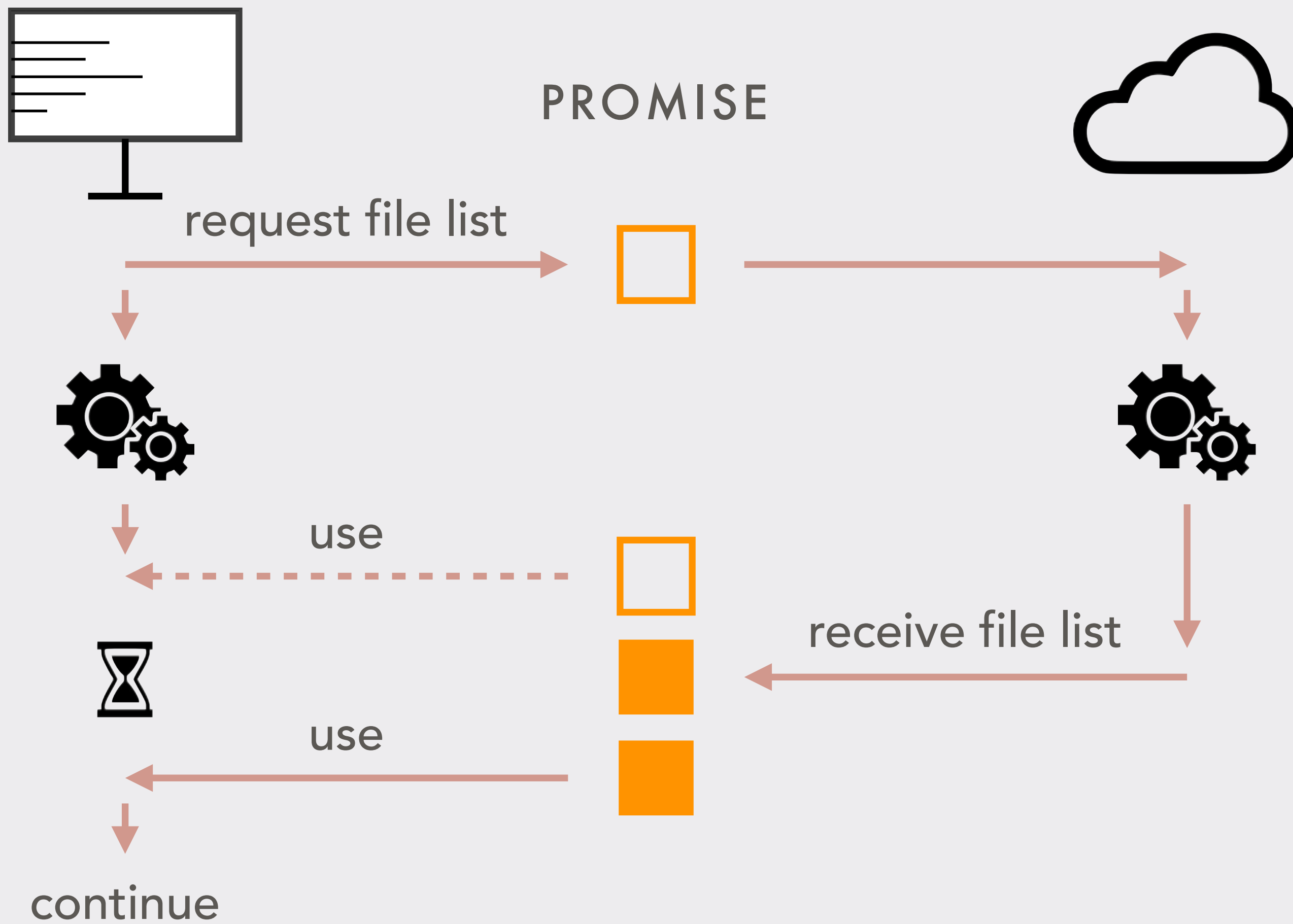


receive file list



display file list

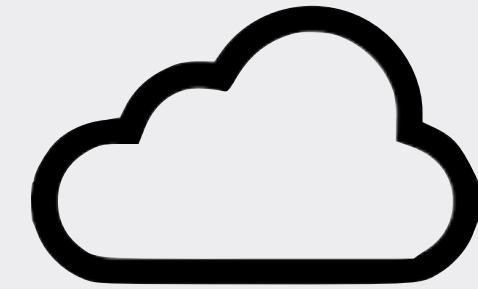
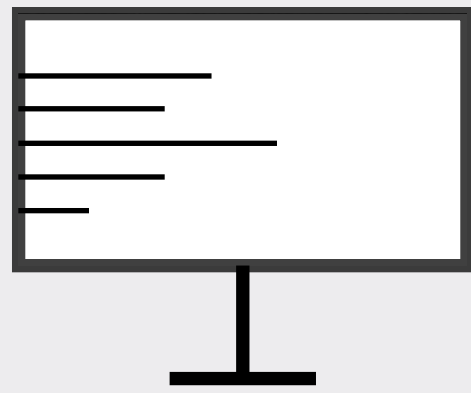




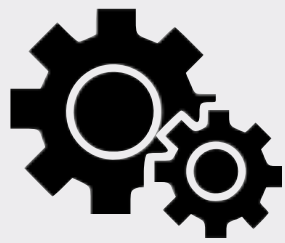
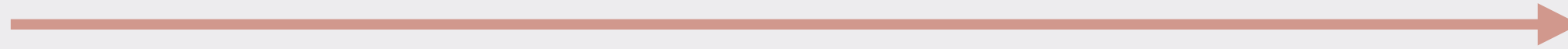
```
fileListPromise := [ self getFileList ] promise.
```

```
...
```

```
fileListPromise value.
```

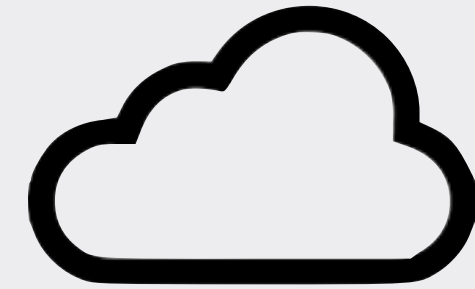
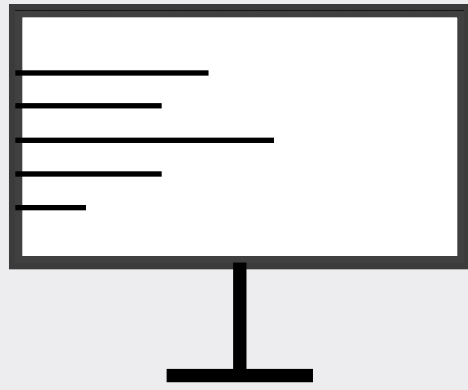


request file list



what happened?





what happened?

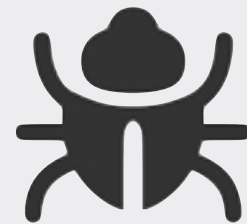


NULL

description

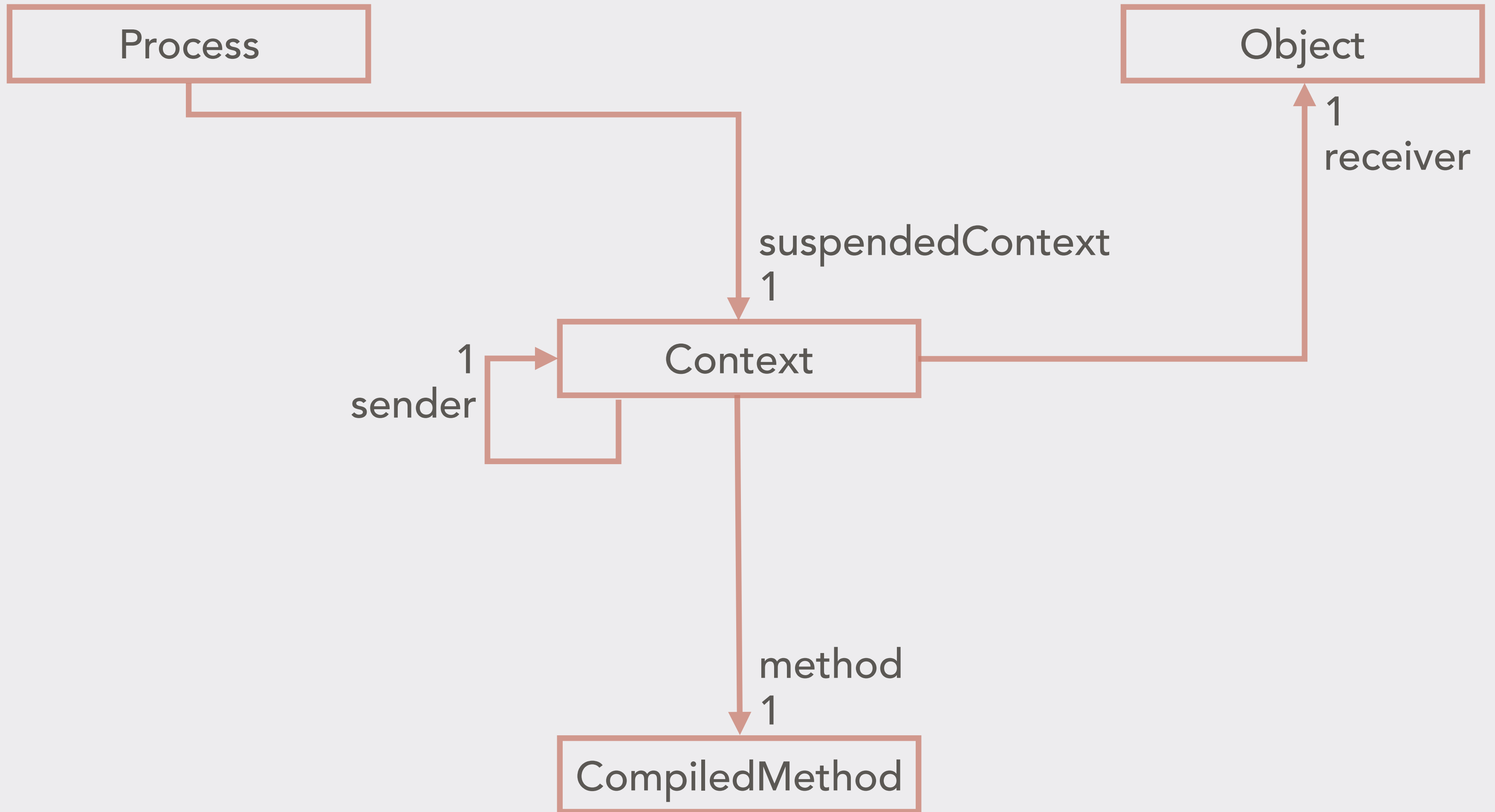
stack trace

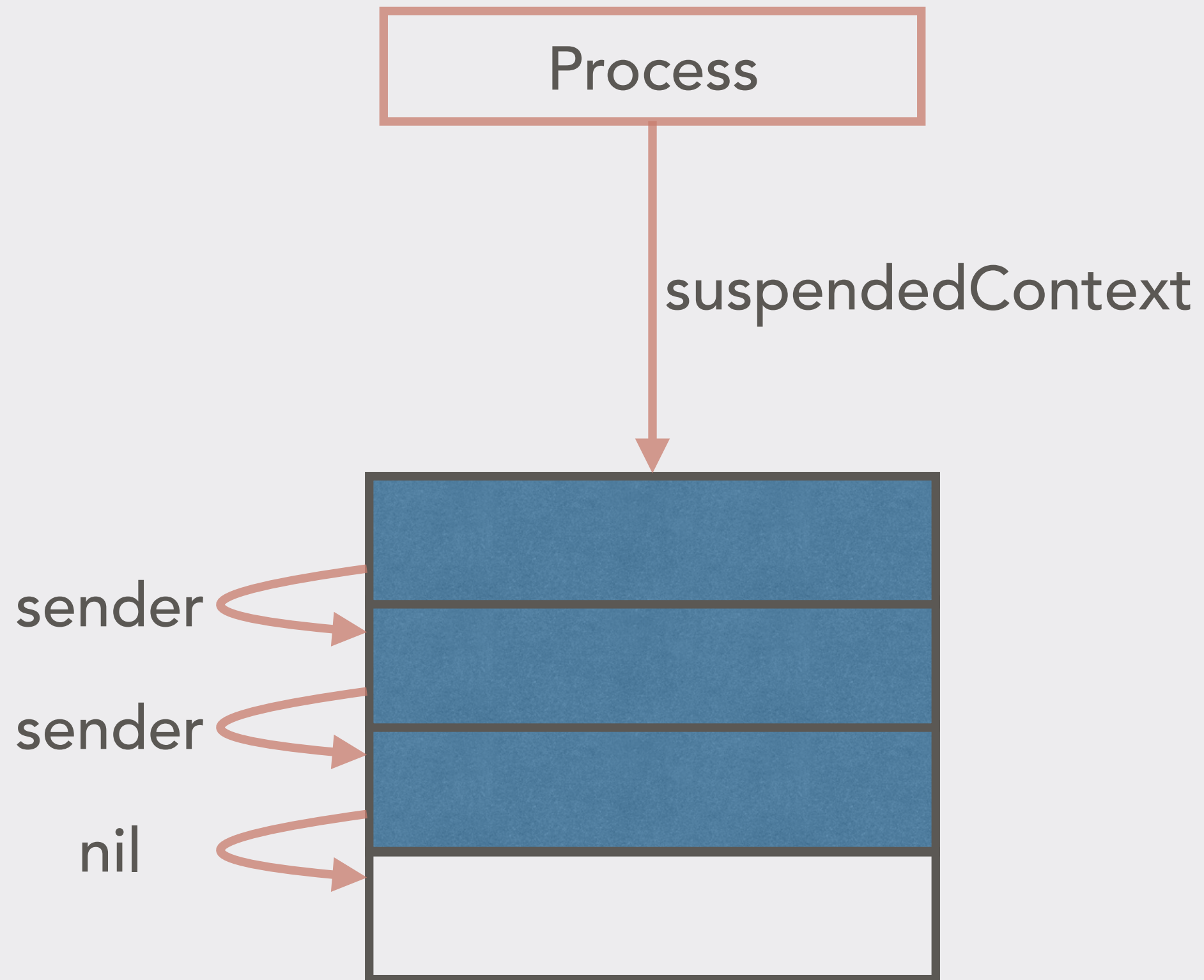
exception object

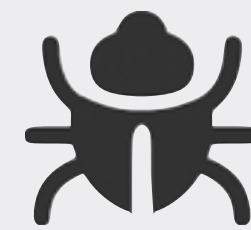
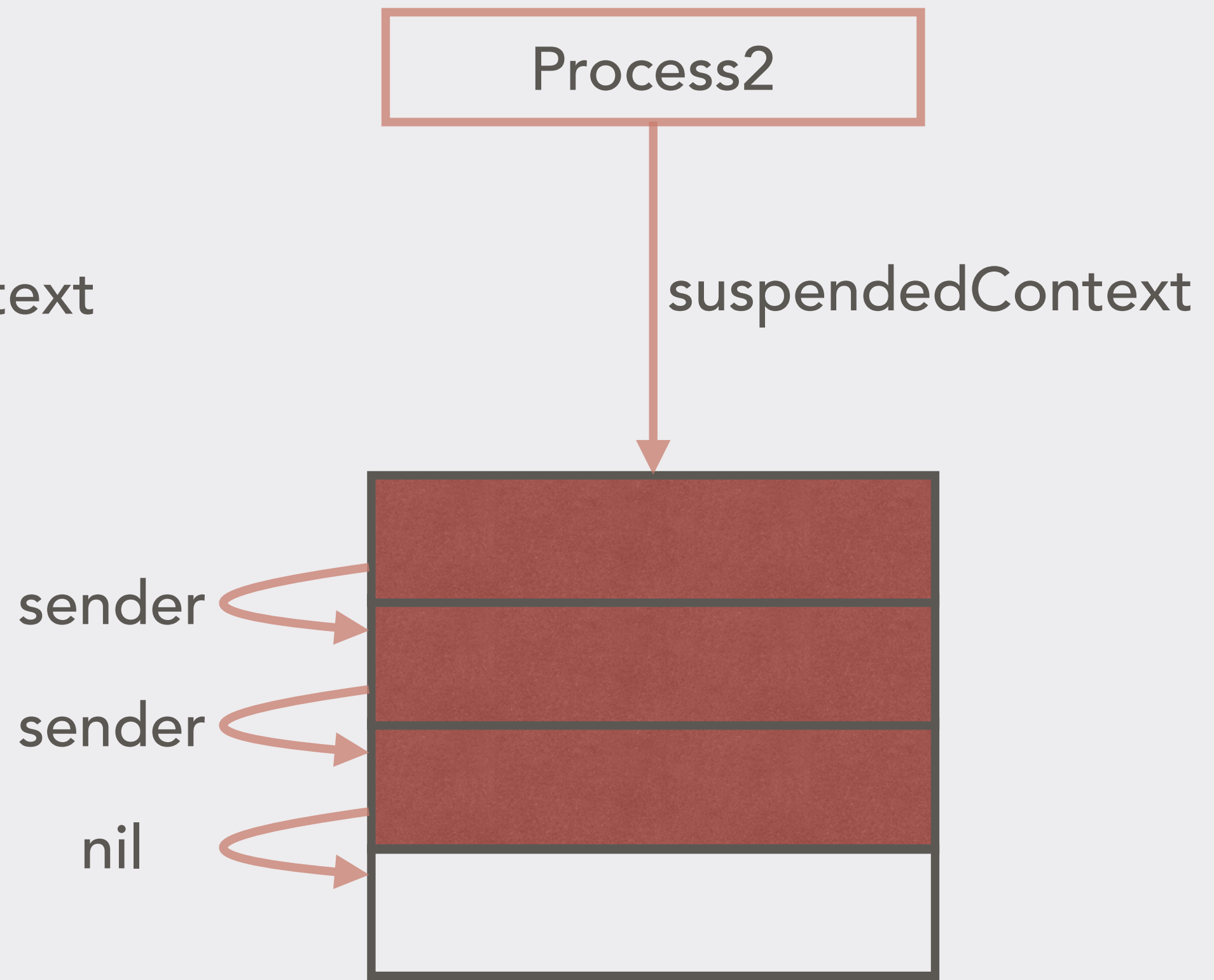
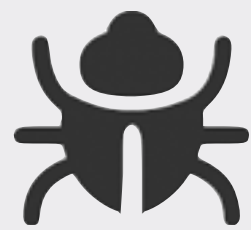
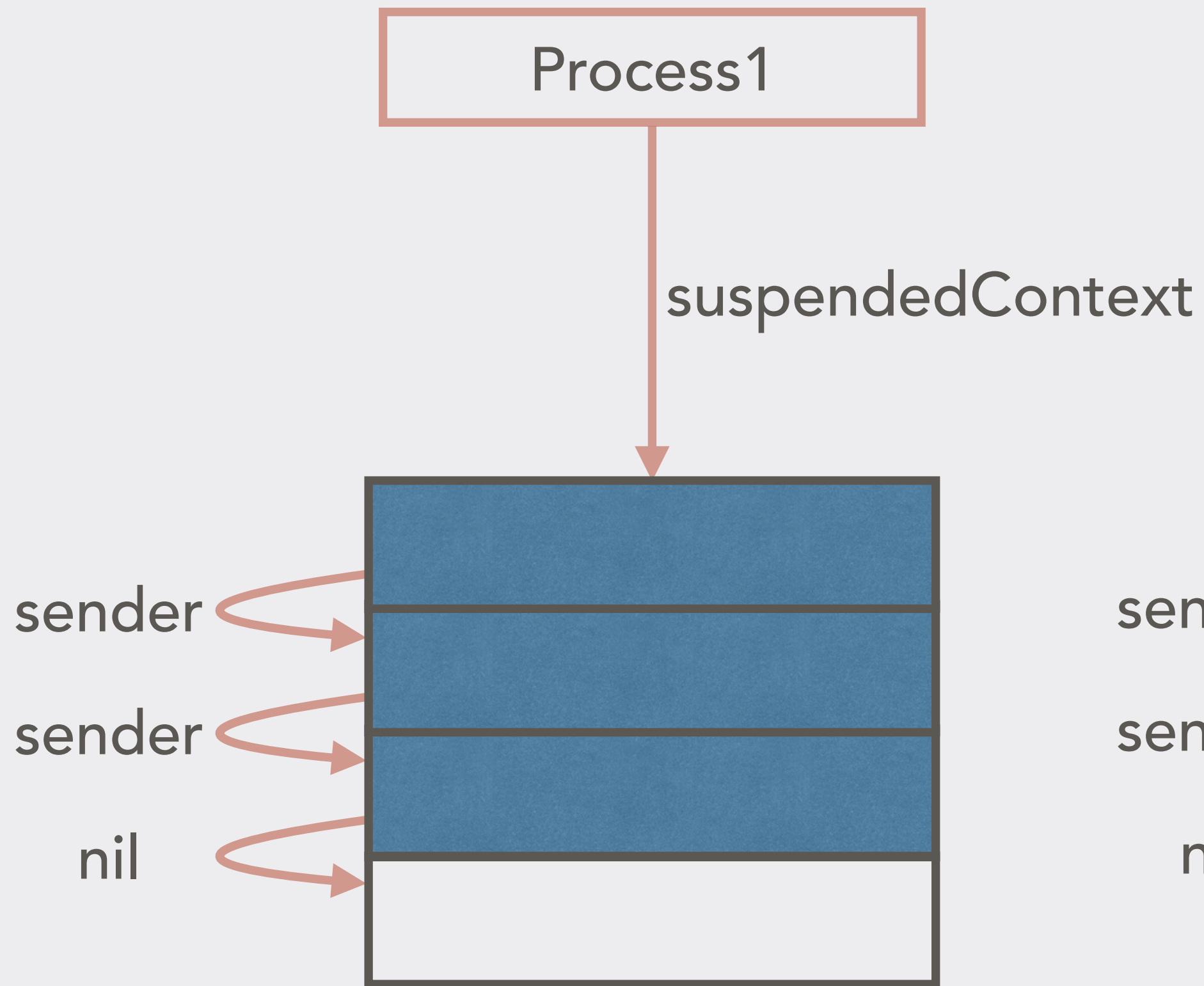


HOW IT WORKS

HOW IT WORKS







Process1+2

suspendedContext

sender

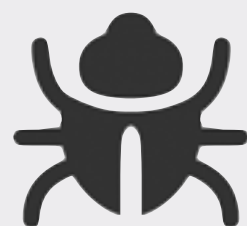
sender

sender

sender

sender

nil



CHALLENGES

COMPUTATIONAL OVERHEAD

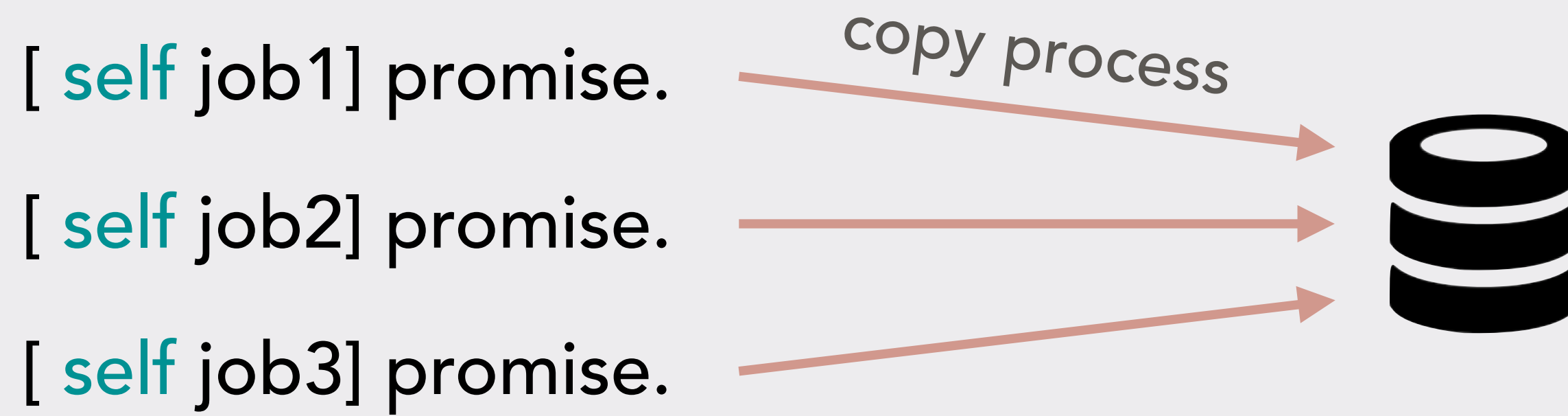
`fileListPromise := [self getFileList] promise.`



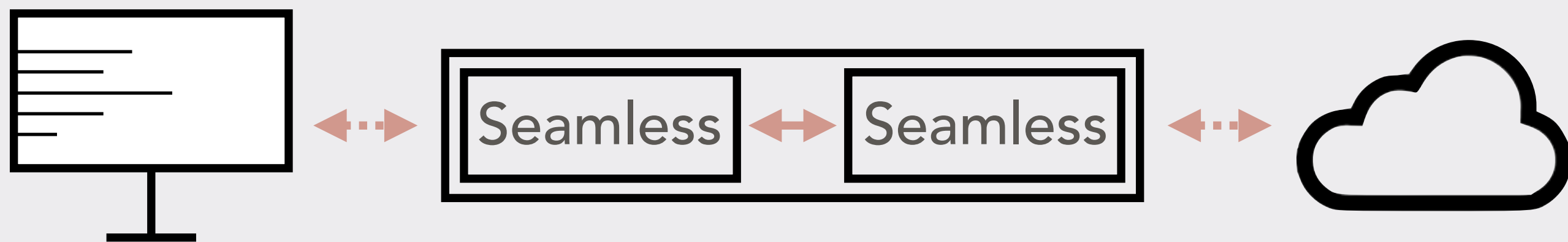
...

`fileListPromise` value.

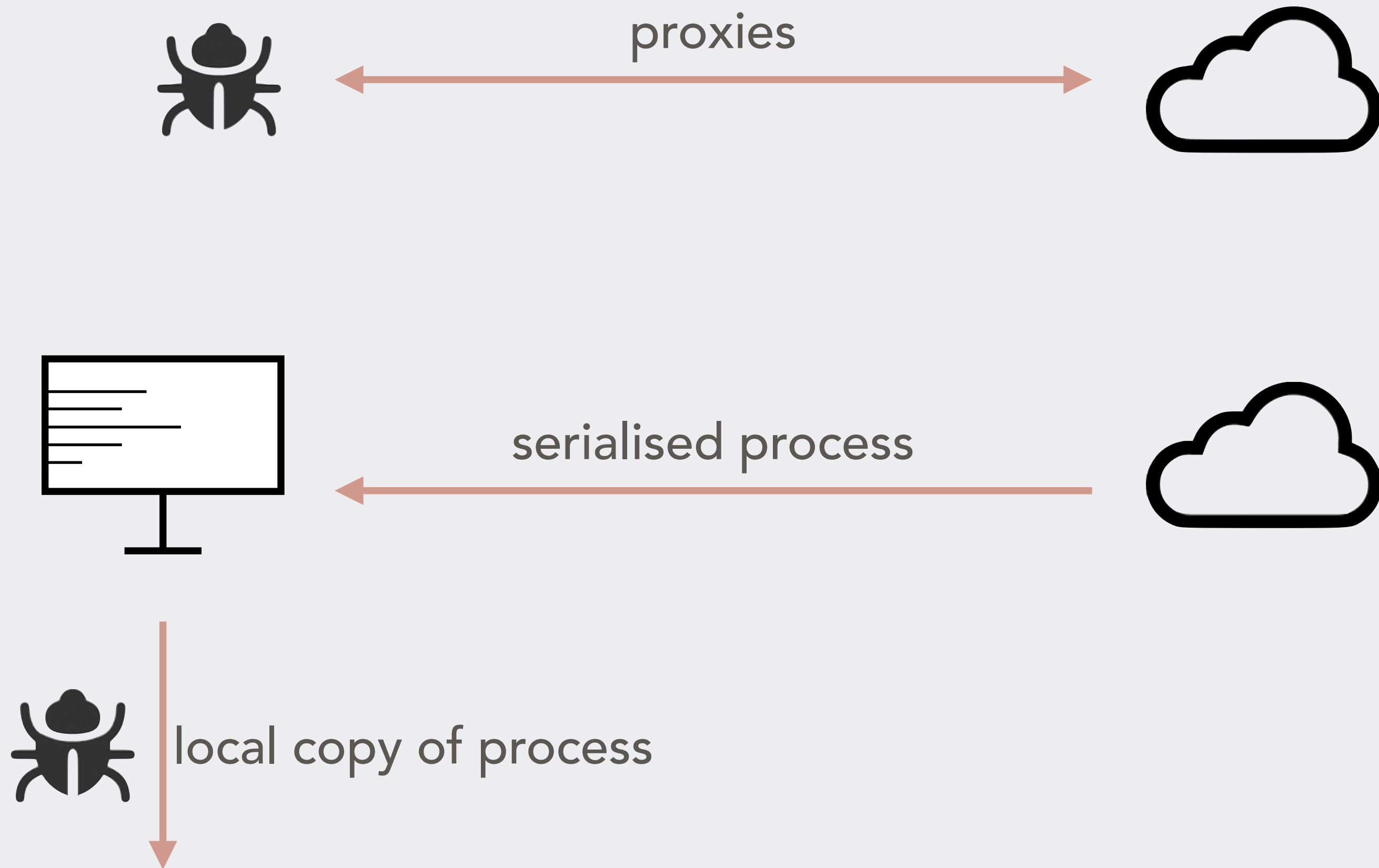
MEMORY CONSUMPTION



REMOTE COMMUNICATION



DEBUGGING



ROADMAP

improve implementation

applicability

IWST

