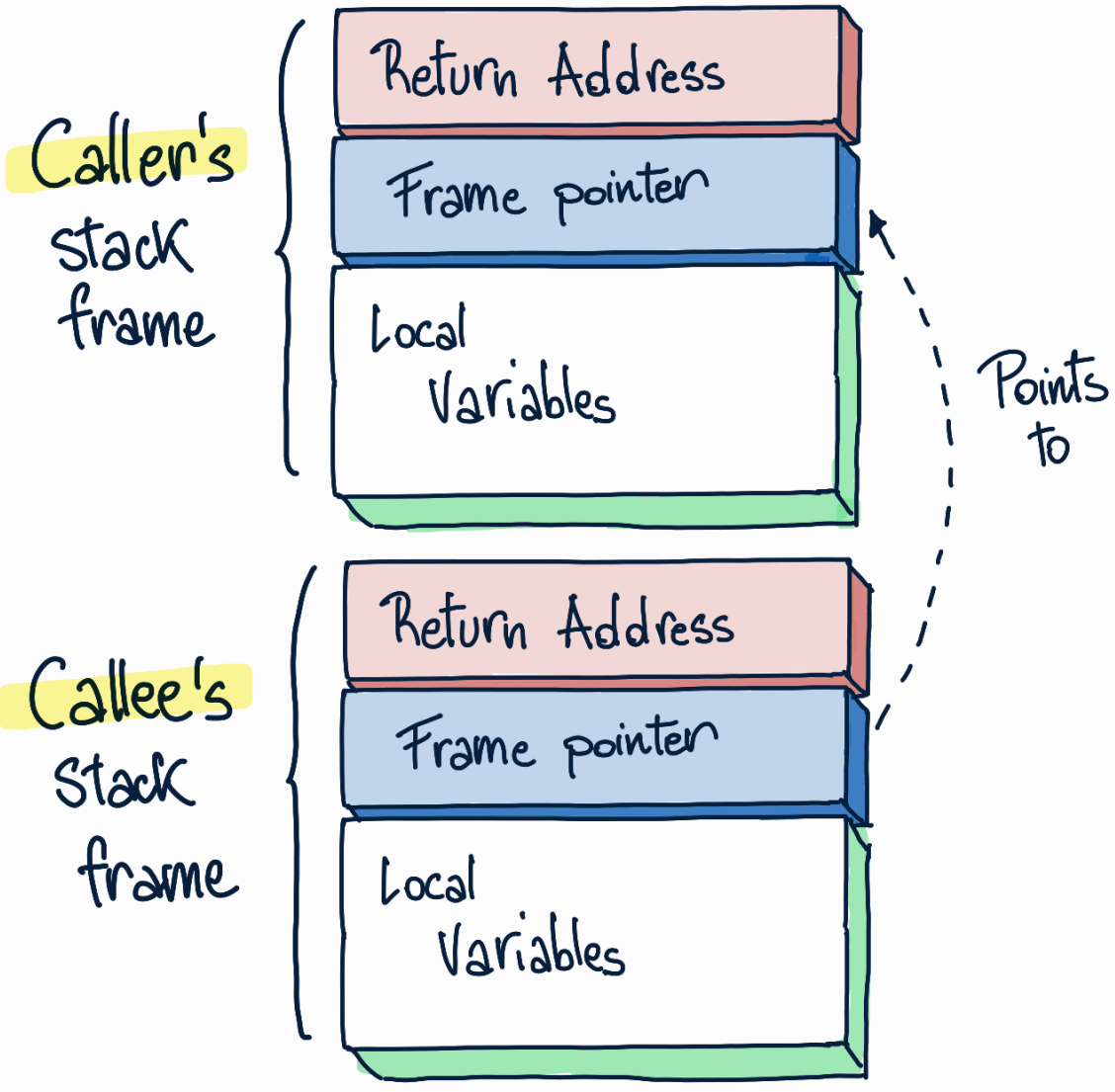
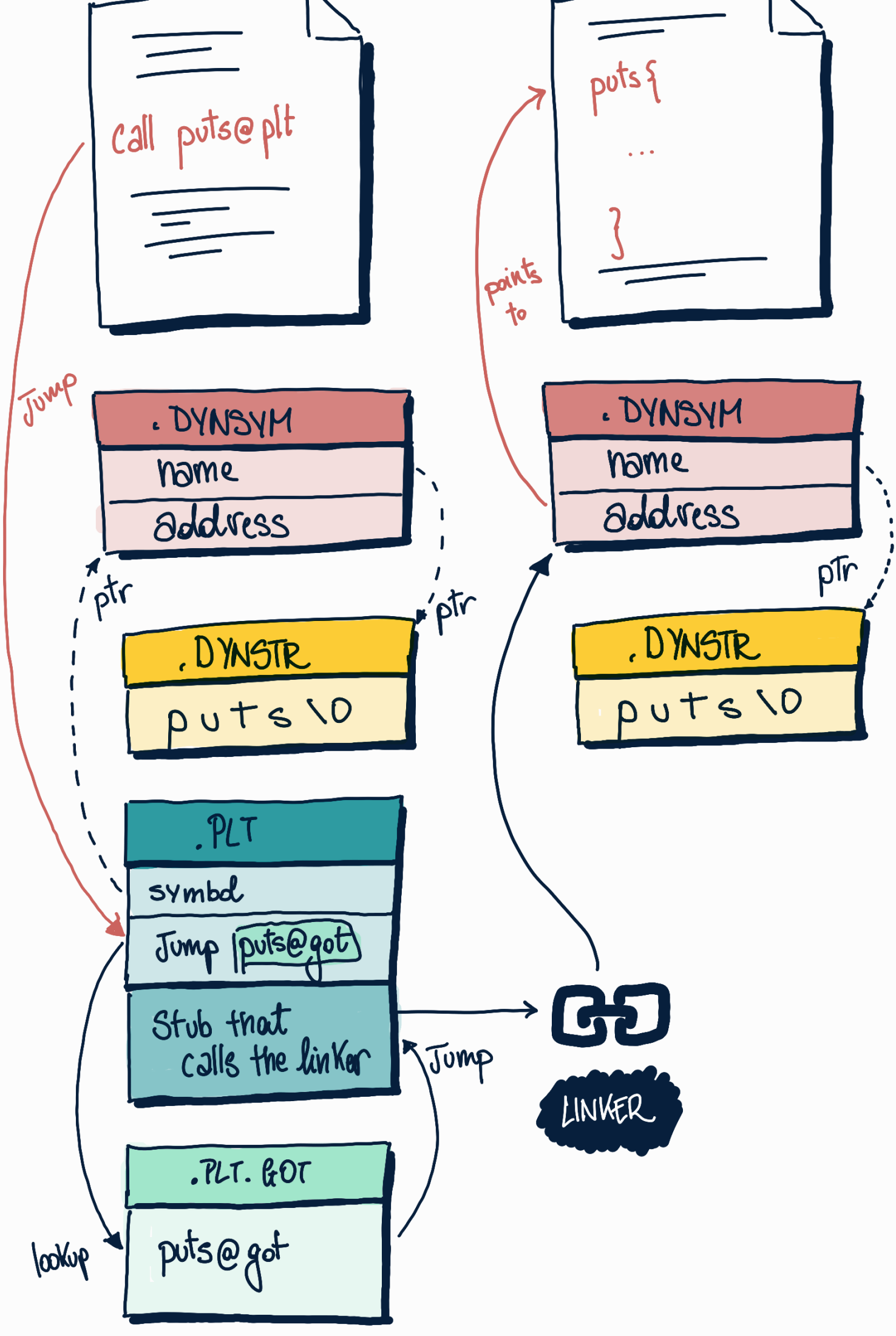
STACK FRAMES

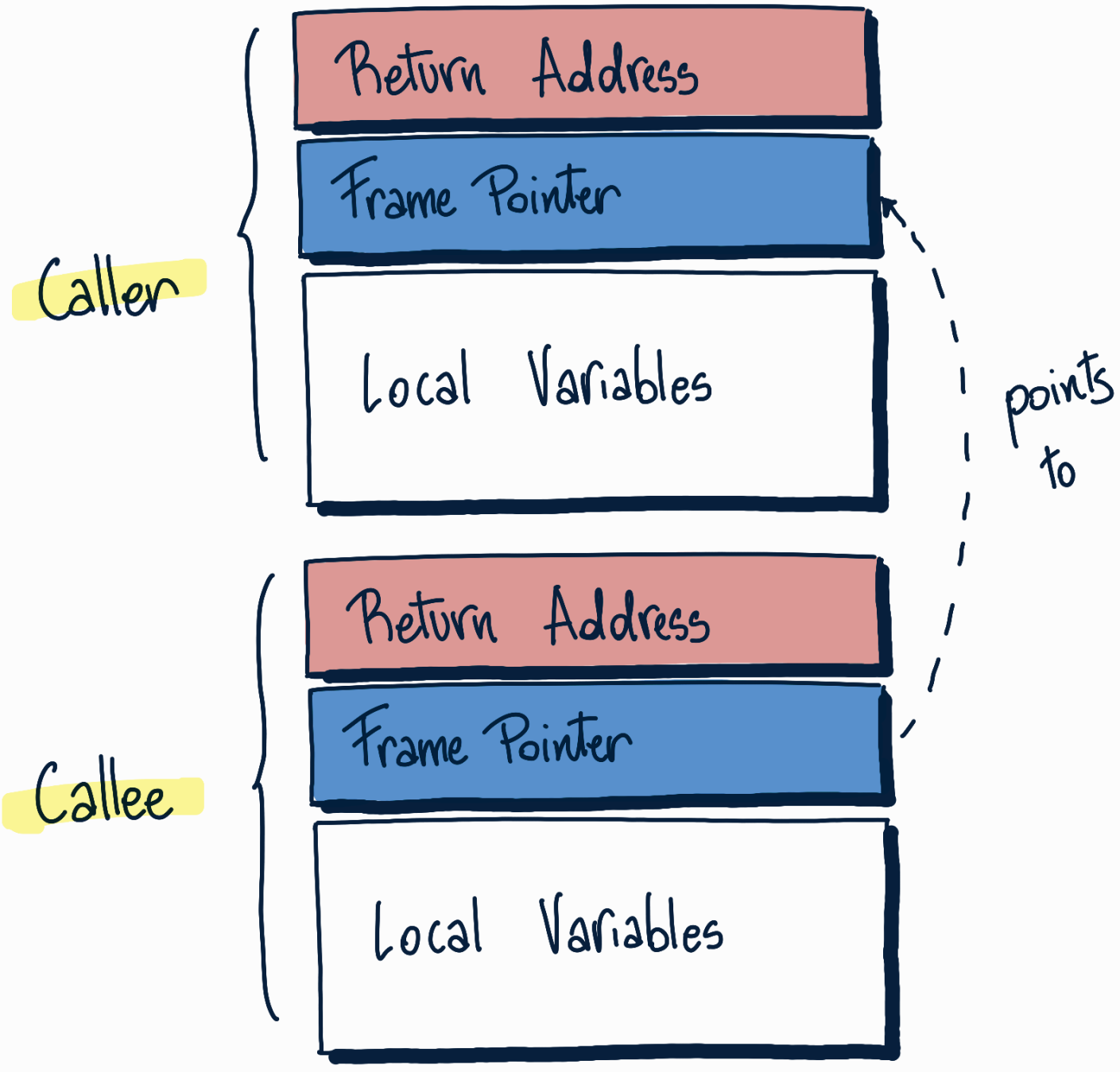


DYNAMIC LINKING

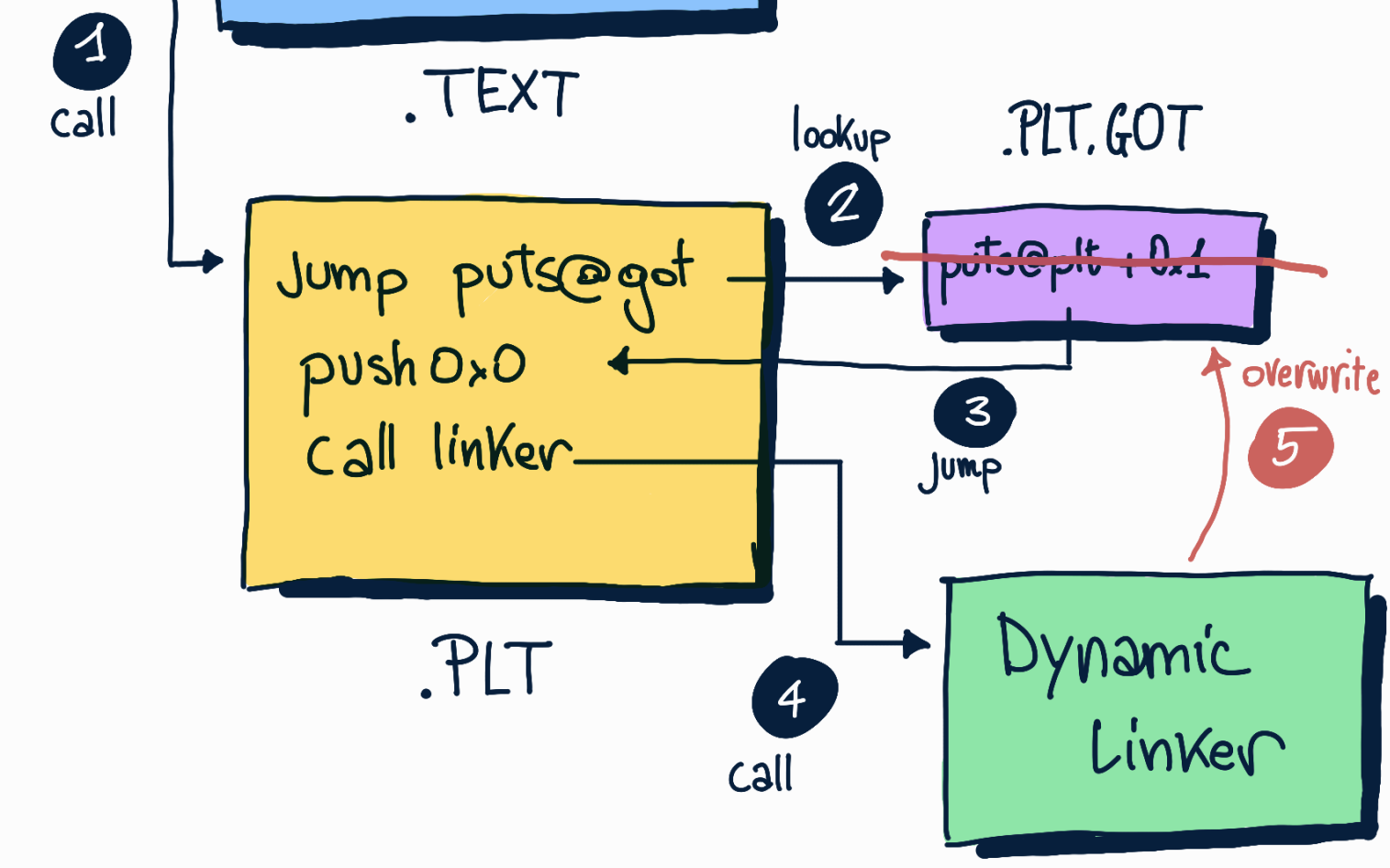
(ELF SECTIONS)


call puts@plt
puts {
...
}
points to
Jump
.DYNSYM
name
address
.DYNSYM
name
address
ptr
ptr
ptr
.DYNSTR
puts\0
.DYNSTR
puts\0
.PLT
symbol
Jump puts@got
Stub that calls the linker
Jump
LINKER
.PLT.GOT
puts@got
lookup

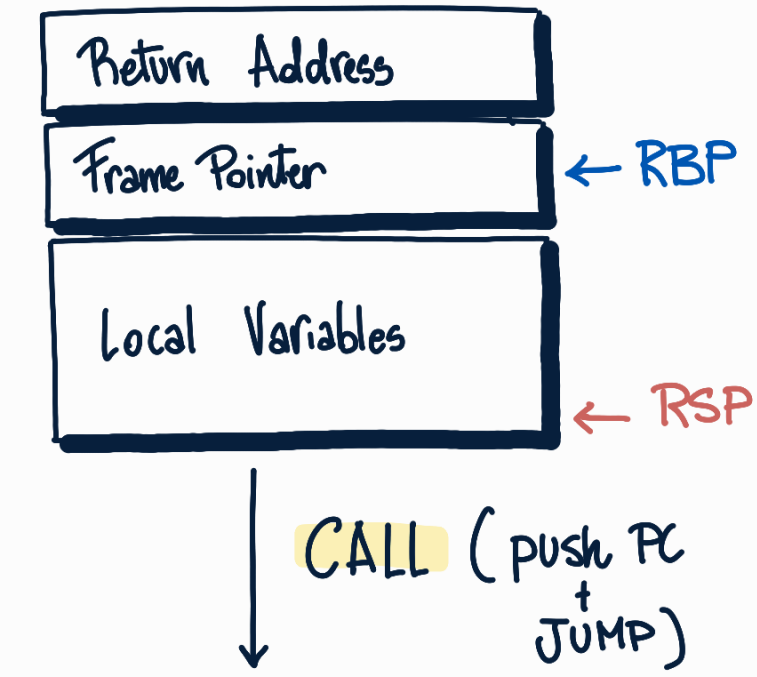
STACK FRAMES

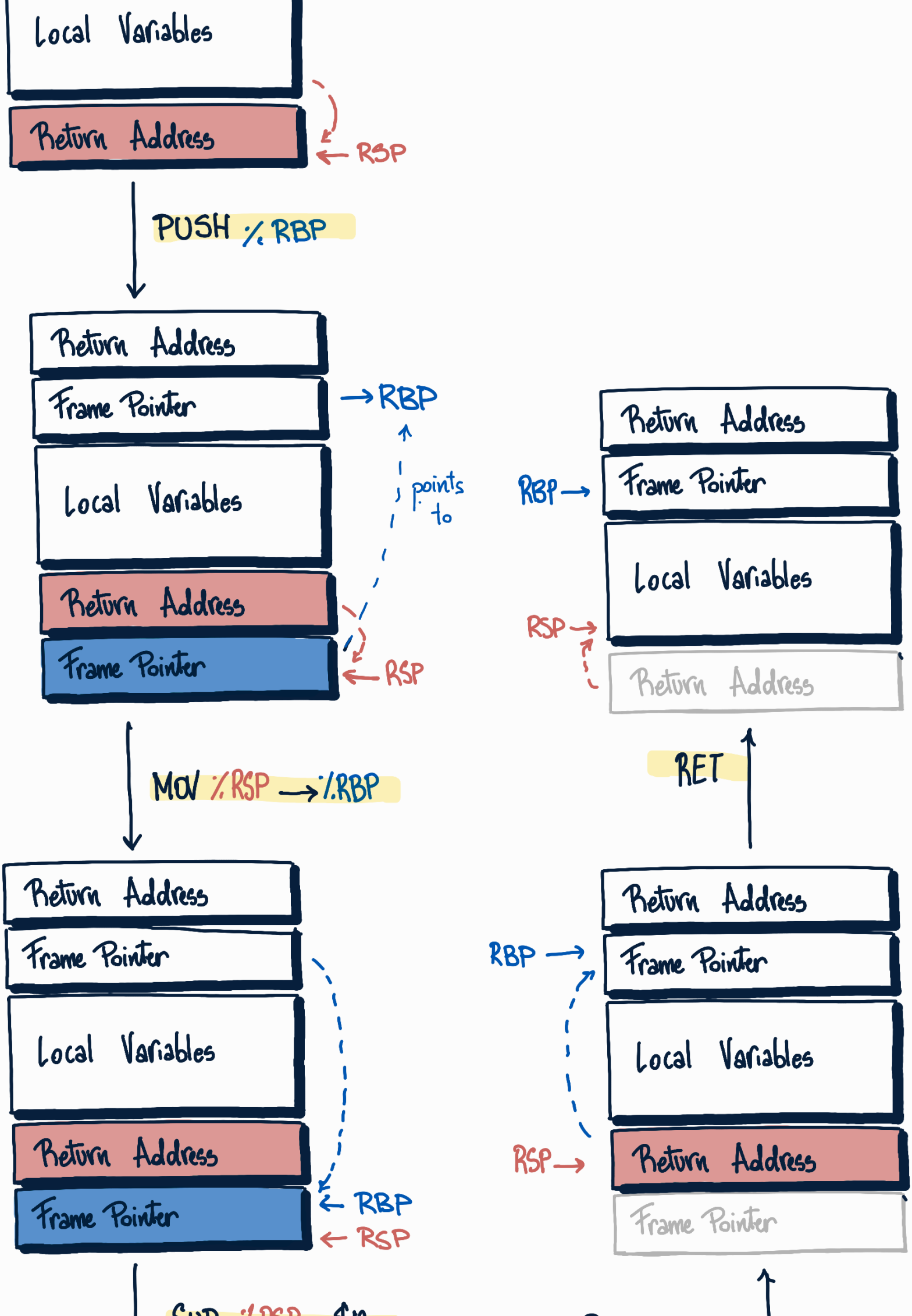


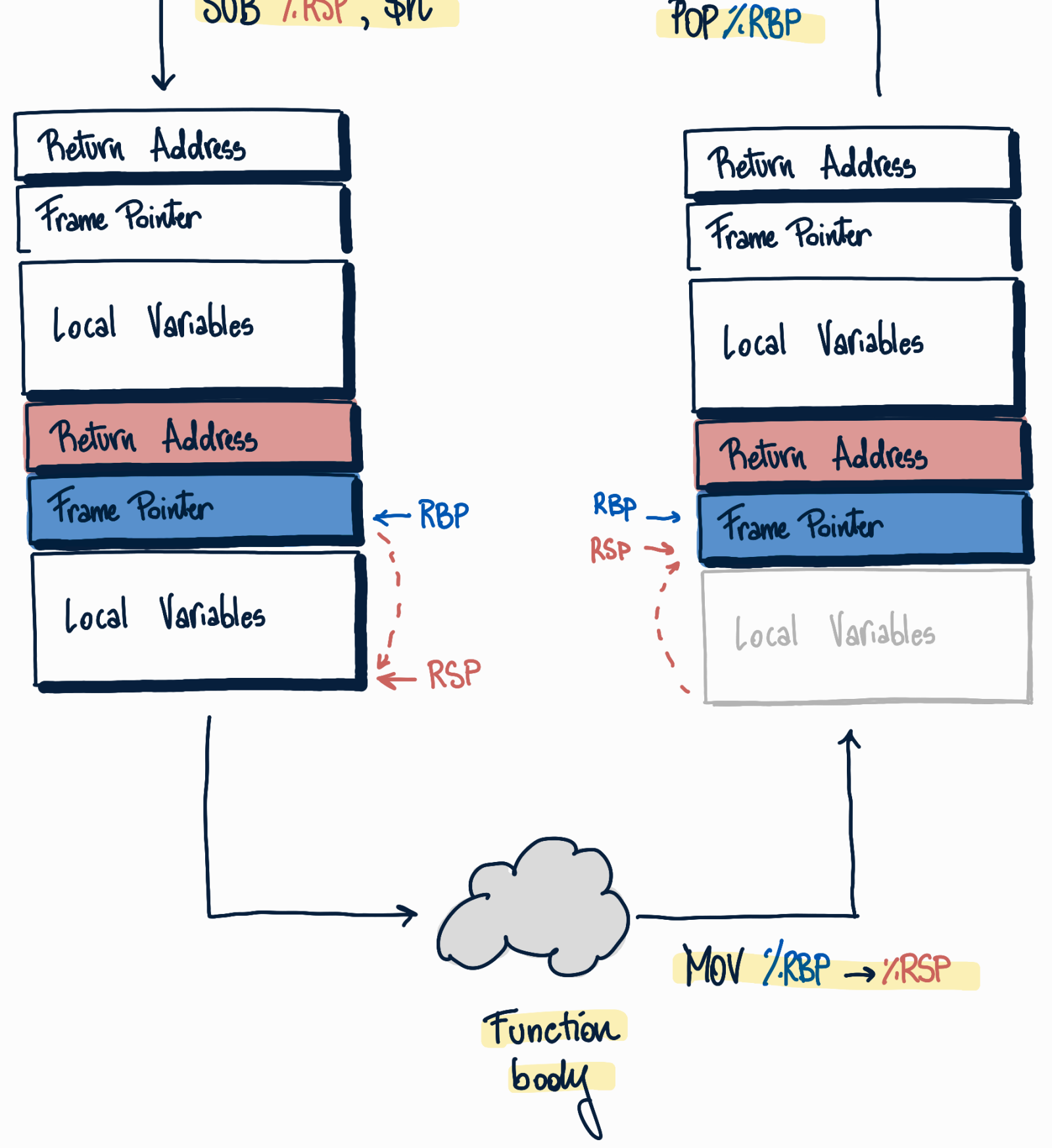
RELOCATIONS
(LAZY LOADING)



STACK FRAMES

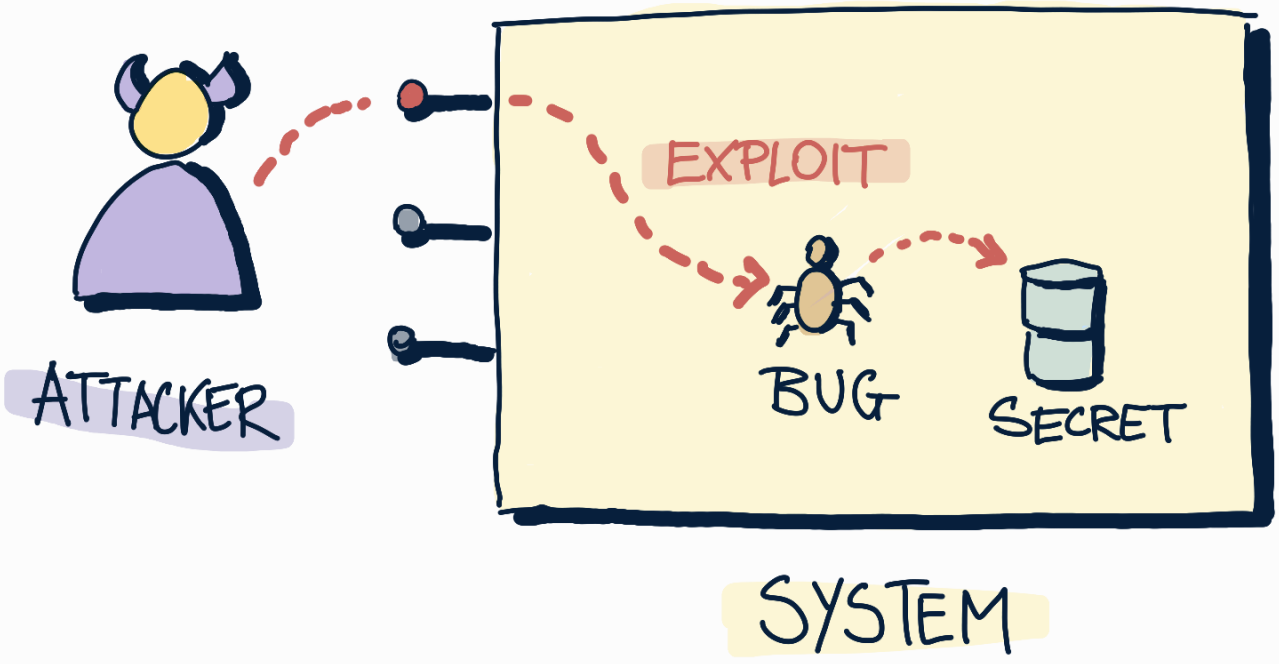


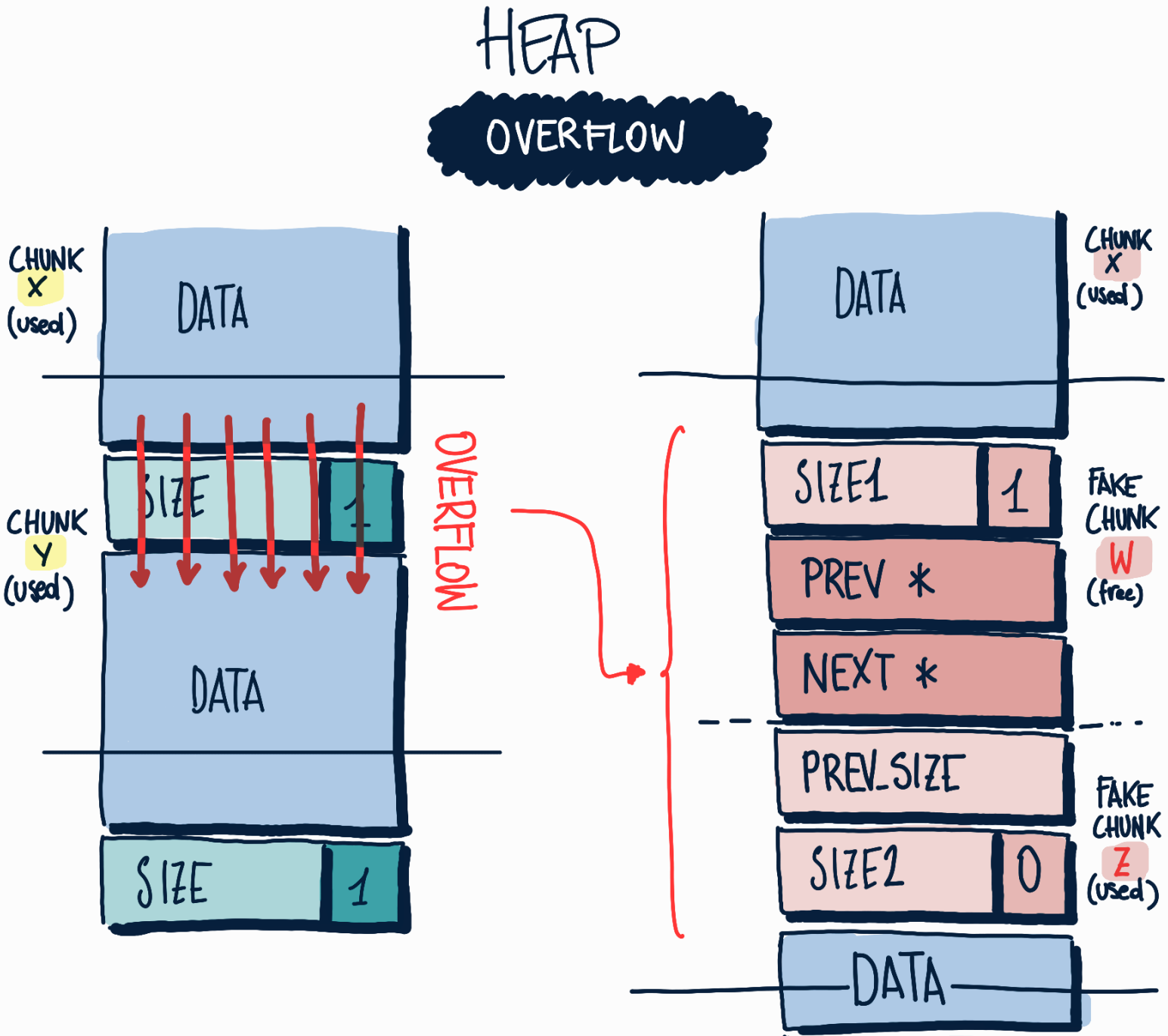

Local Variables
Return Address
RSP
PUSH %RBP
Return Address
Frame Pointer
RBP
Local Variables
points to
Return Address
Frame Pointer
RSP
MOV %RSP → %RBP
Return Address
Frame Pointer
Local Variables
Return Address
Frame Pointer
RBP
RSP
RET
Return Address
Frame Pointer
Local Variables
RBP
Return Address
RSP
Frame Pointer



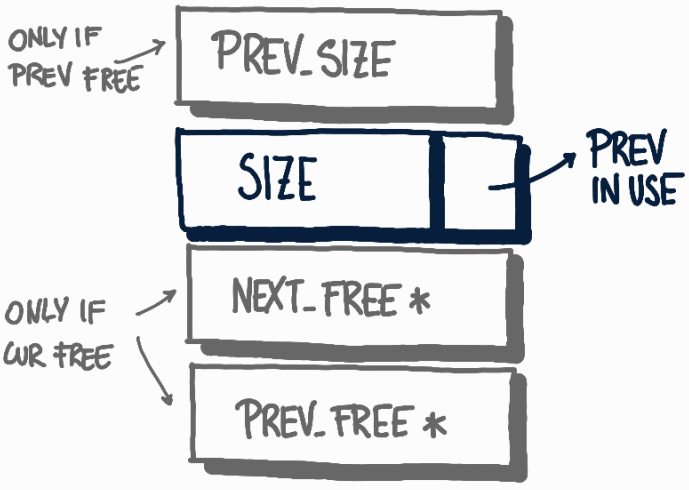
INTRO

SURFACE


EXPLOIT
ATTACKER
BUG
SECRET
SYSTEM


HEAP
OVERFLOW
CHUNK X (used)
DATA
SIZE
1
CHUNK Y (used)
DATA
SIZE
1
OVERFLOW
DATA
CHUNK X (used)
SIZE1
1
FAKE CHUNK W (free)
PREV *
NEXT *
PREV_SIZE
FAKE CHUNK Z (used)
SIZE2
0
DATA

METADATA

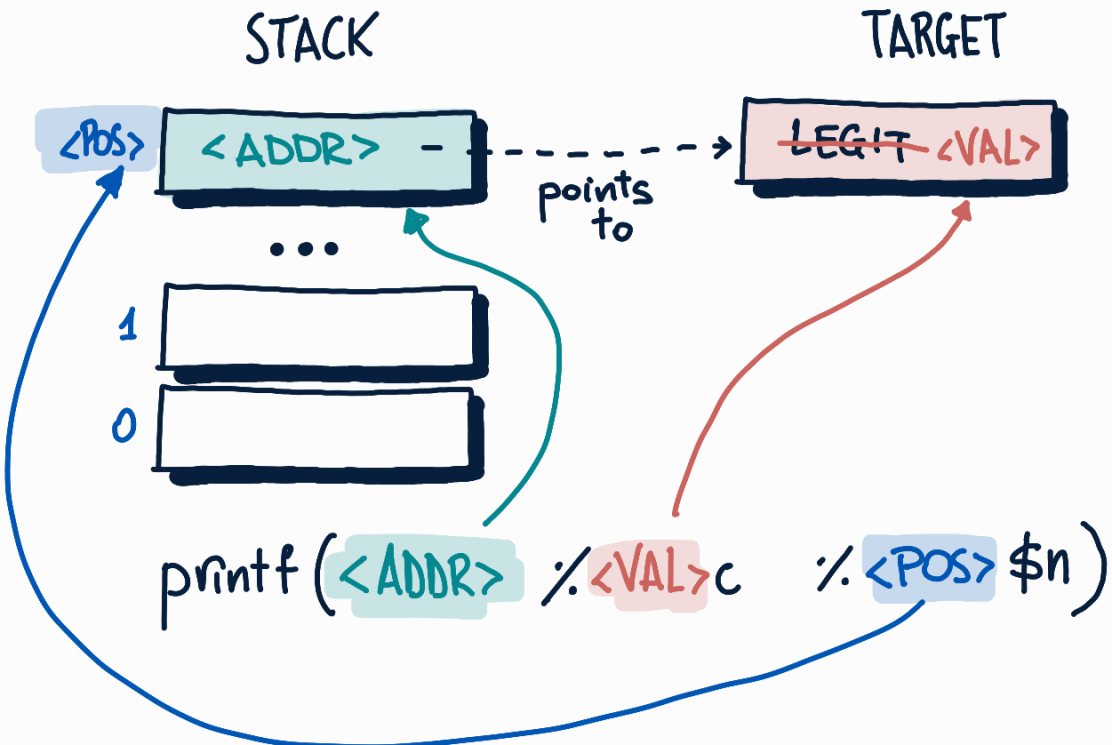


EXPLOITATION

1. OVERFLOW X
2. CALL FREE(X)
   - W is free ⇒ MERGE(X,W)
     - Calls UNLINK(W)

W.NEXT→PREV = PREV
W.PREV→NEXT = NEXT

FORMAT STRINGS



USE-AFTER-FREE