

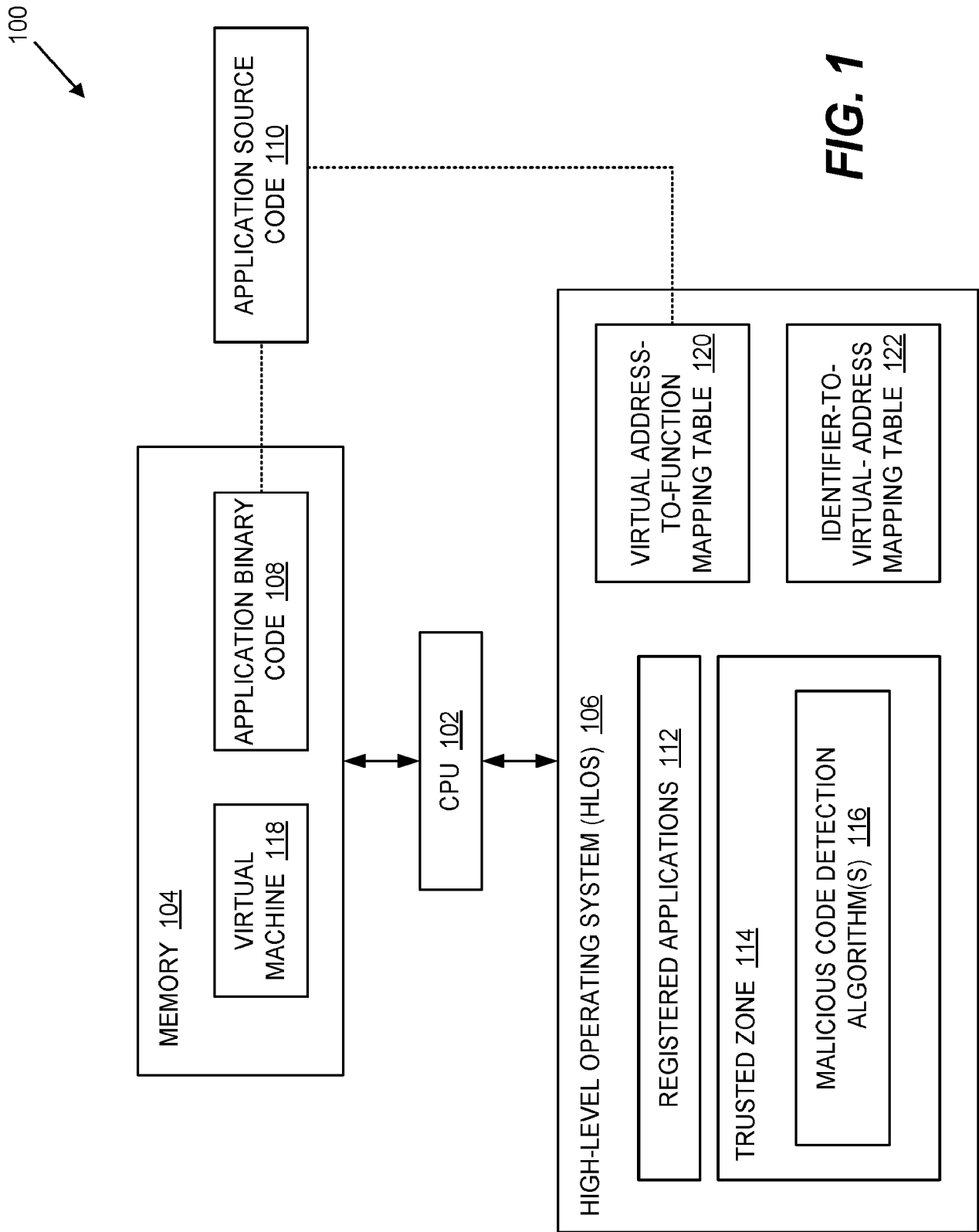


FIG. 1

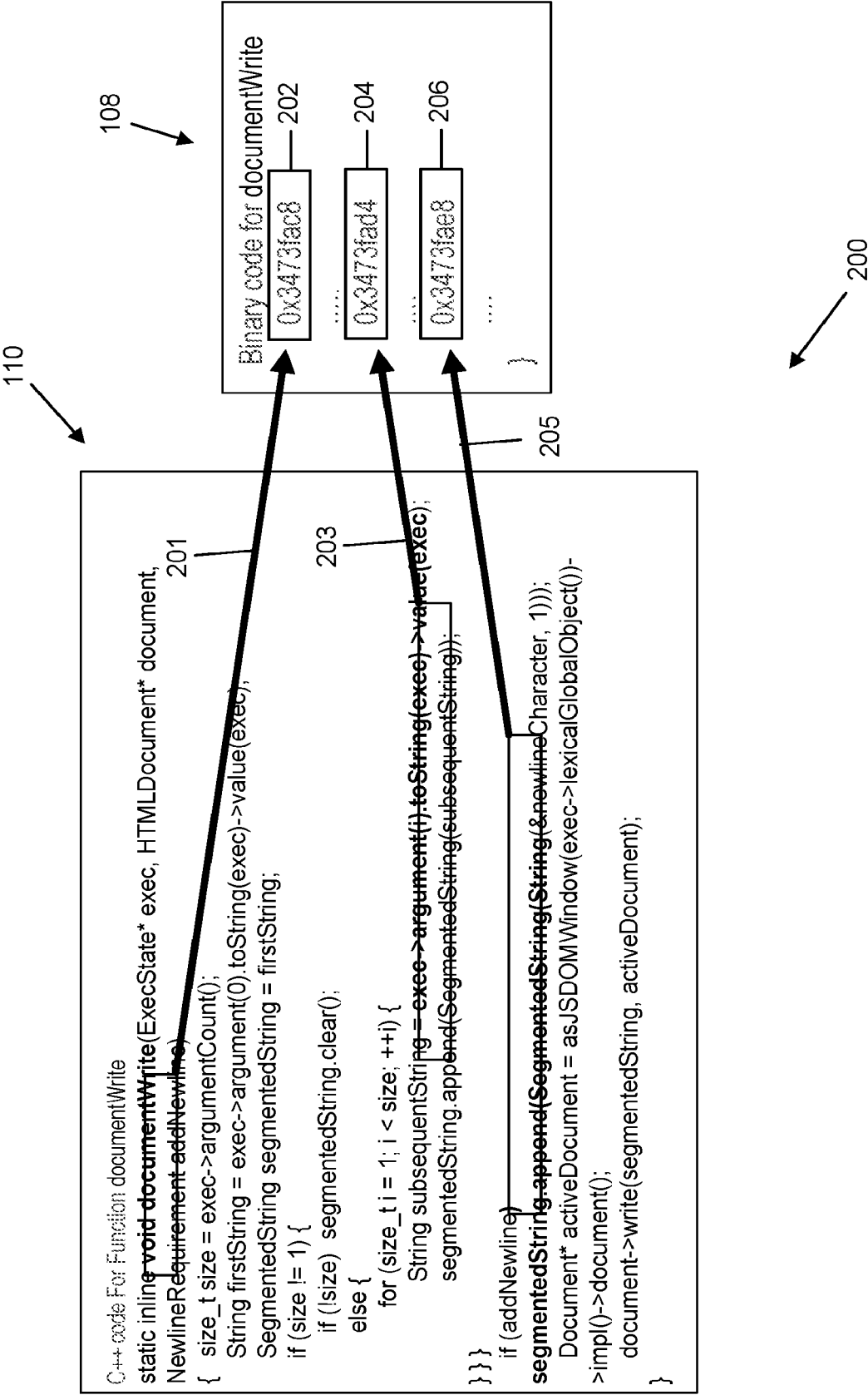


FIG. 2

300

VIRTUAL ADDRESS-TO-FUNCTION MAPPING TABLE <u>120</u>	
VIRTUAL ADDRESS <u>302</u>	FUNCTIONAL POINT OF INTEREST <u>304</u>
0x3273fa94	EVAL_FUNCTION
0x3473fac8	DOCUMENT_WRITE_FUNCTION_START
0x3473fad4	DOCUMENT_WRITE_1
0x3473fae8	DOCUMENT_WRITE_2
0x29b93420	ONCLICK_FUNCTION
0x59d782b4	DOCUMENT_COOKIE_FUNCTION
0x59d78264	SETTIMEOUT_FUNCTION_START

108

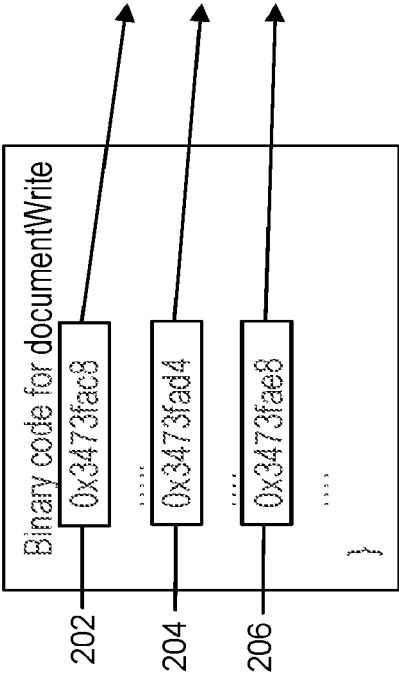
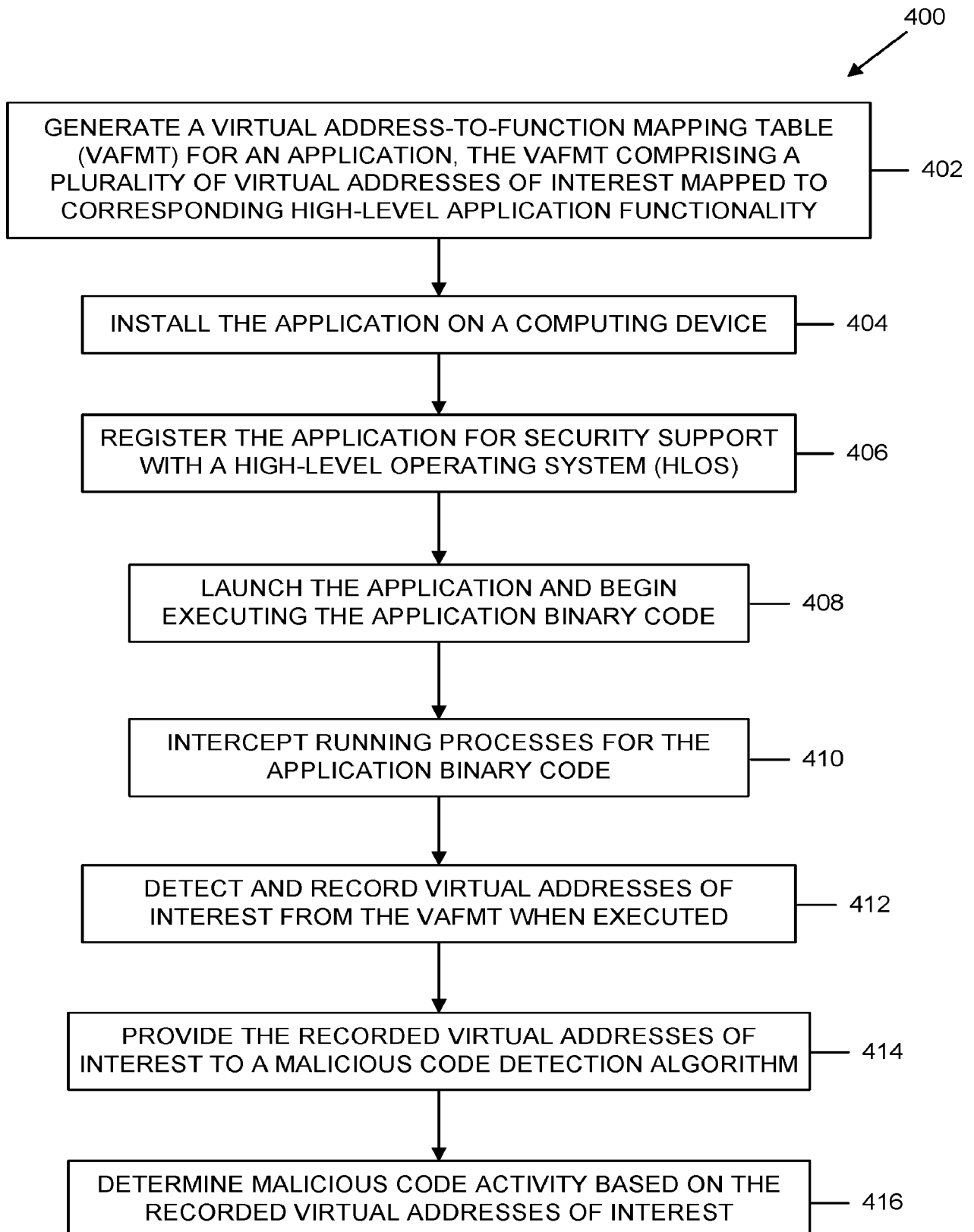
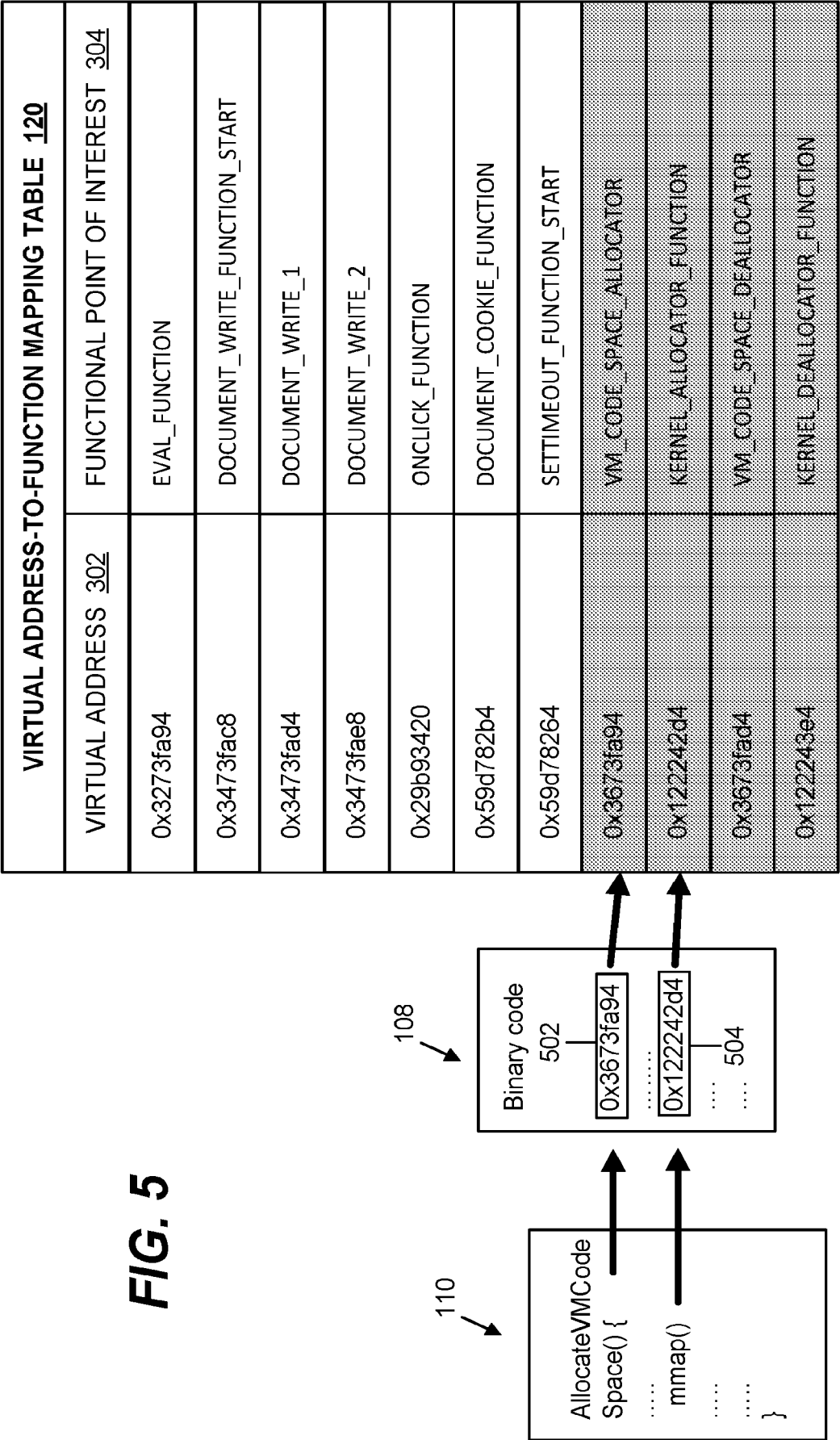


FIG. 3

**FIG. 4**

500

FIG. 5



600

FIG. 6

IDENTIFIER-TO-VIRTUAL ADDRESS MAPPING	
TABLE 122	
MACRO MEANING 604	VIRTUAL ADDRESSES 602
VM_CODE_SPACE_START	0x2fa2ca08
VM_CODE_SPACE_END	0x3473fa80
VM_LARGE_OBJ_SPACE_START	
VM_LARGE_OBJ_SPACE_END	
VM_NEWSPACE_START	0x3663fa80
VM_NEWSPACE_END	0x3663fbb8

VIRTUAL ADDRESS-TO-FUNCTION MAPPING TABLE 120	
VA 302	FUNCTIONAL POINT OF INTEREST 304
0x3273fa94	EVAL_FUNCTION
0x3473fac8	DOCUMENT_WRITE_FUNCTION_START
0x3473fad4	DOCUMENT_WRITE_1
0x3473fae8	DOCUMENT_WRITE_2
0x29b93420	ONCLICK_FUNCTION
0x59d782b4	DOCUMENT_COOKIE_FUNCTION
0x59d78264	SETTIMEOUT_FUNCTION_START
0x3673fa94	VM_CODE_SPACE_ALLOCATOR
0x122242d4	KERNEL_ALLOCATOR_FUNCTION
0x3673fad4	VM_CODE_SPACE_DEALLOCATOR
0x122243e4	KERNEL_DEALLOCATOR_FUNCTION

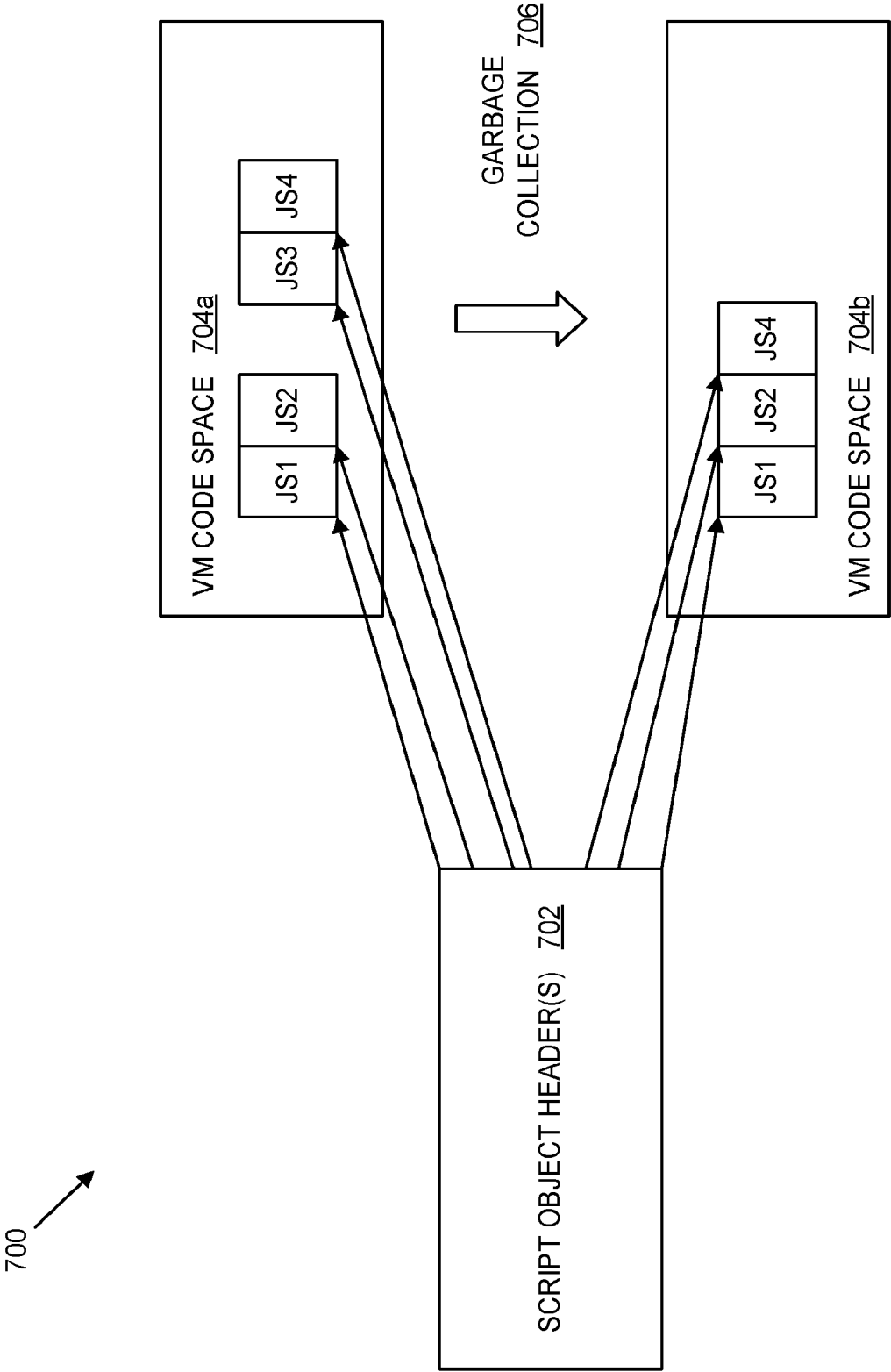


FIG. 7

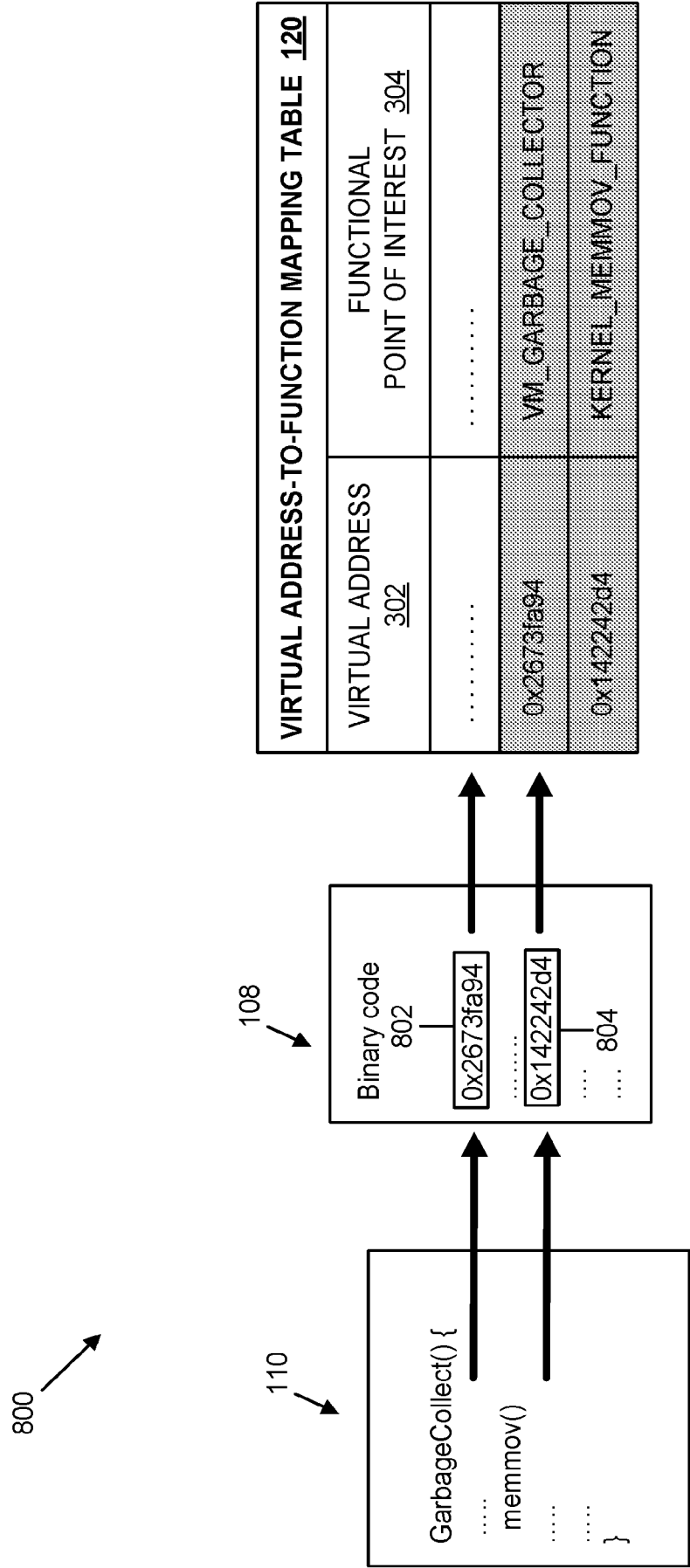


FIG. 8

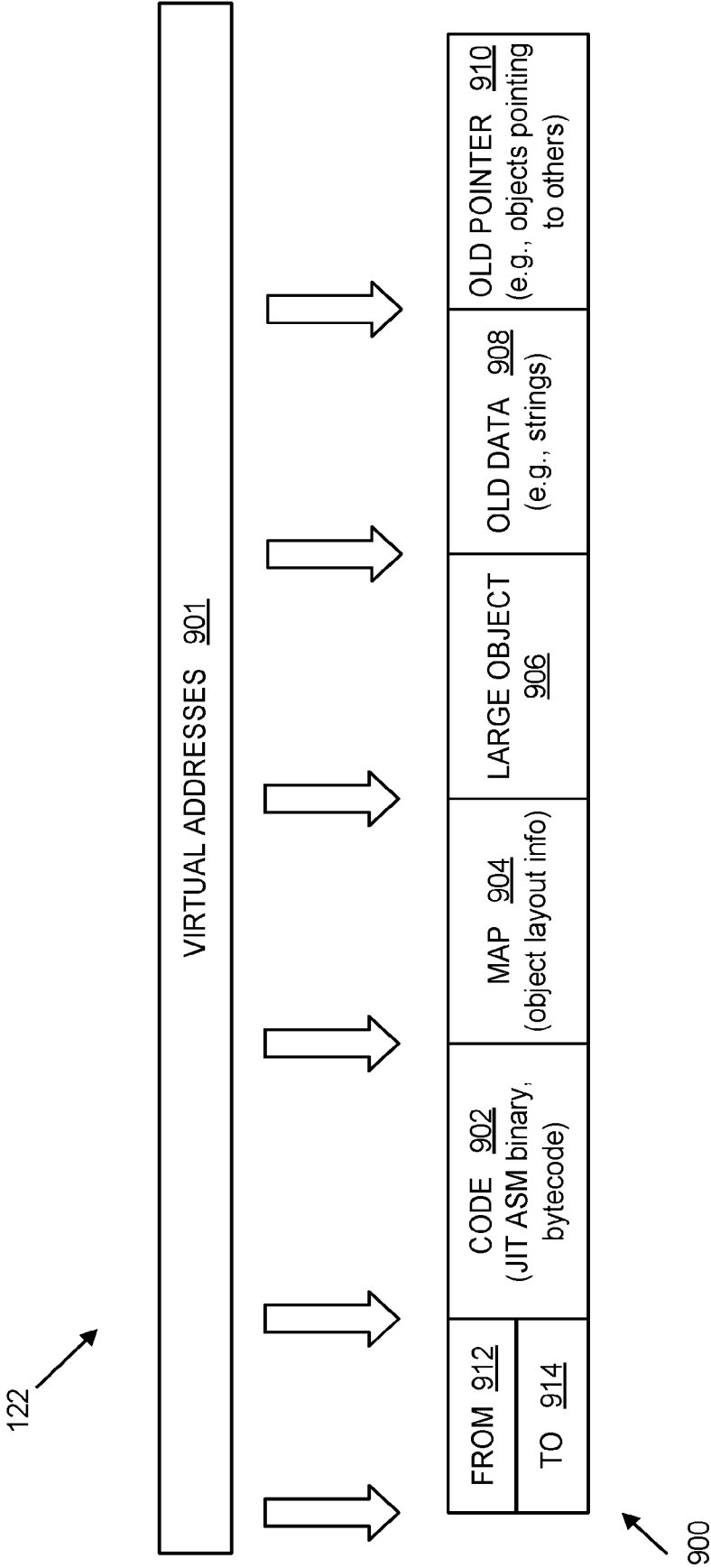
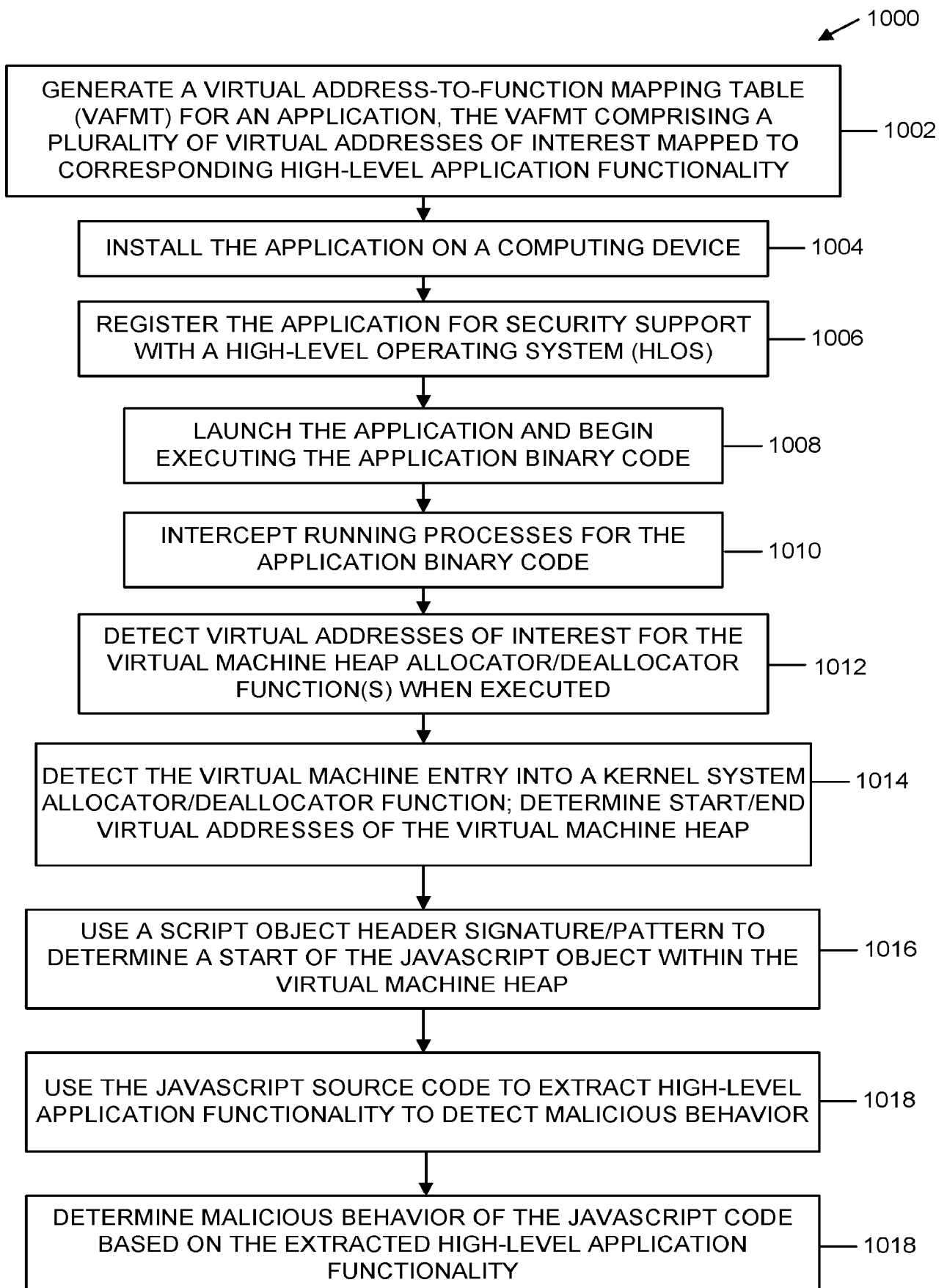


FIG. 9

**FIG. 10**

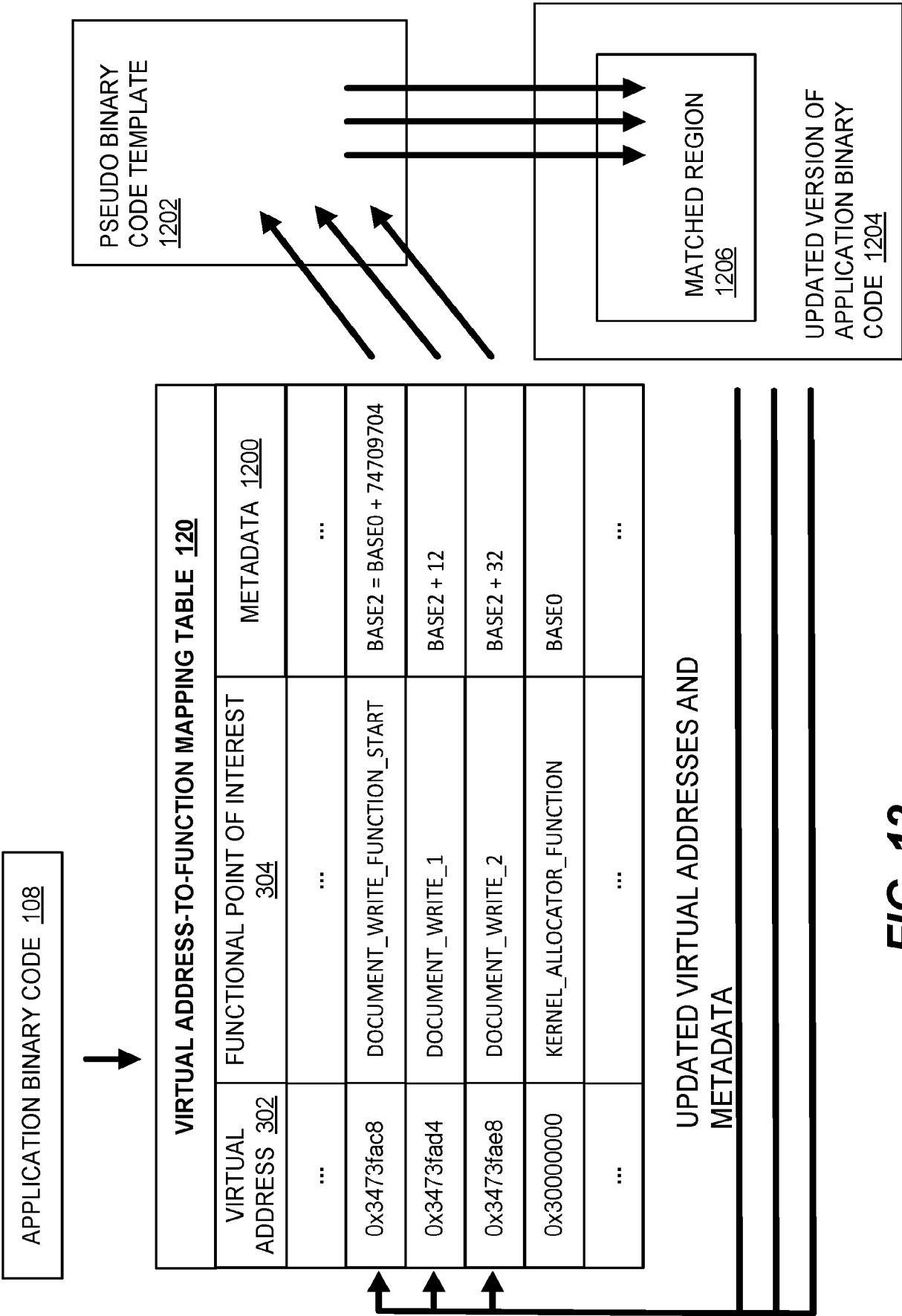
1104

1112

120

Custom Virtual Address Mapping Table with offset and length						
Virtual Address for the Web/ Browser Application function doing Buffer or data structure allocation	Virtual Address for the "kernel system allocators" invoked from Application functions doing Buffer or data structure allocation (Offline/ Statically Determined & OTA Updated) <u>1102</u>	Buffer Start Virtual Addresses (Dynamically determined) <u>1106</u>	Buffer end Virtual Addresses (Dynamically determined) <u>1108</u>	member field/ pointer offset within allocated buffer data structure (Offline/ Statically Determined & OTA Updated) (optional) <u>1110</u>	member Field/ pointer length in bytes within allocated buffer data structure (Offline/ Statically Determined & OTA Updated) (optional) <u>1114</u>	Is member field a pointer (Offline/ Statically Determined & OTA Updated) (optional) <u>1114</u>
0x3473fa94	0x4473fa94	0x2fa2ca08	0x3473fa80			YES/NO
0x3473fac8	0x4473fa94	0x3663fa80	0x3663fbb8			YES/NO
.....						
.....						

FIG. 11



VIRTUAL ADDRESS-TO-FUNCTION MAPPING TABLE 120		
VIRTUAL ADDRESS 302	MACRO REPRESENTING FUNCTIONAL MEANING 304	VIRTUAL-ADDRESS FREE 1200 REPRESENTATION USING BYTE-OFFSET
...	...	...
0x3133b61c	DOCUMENT_WRITE_FUNCTION_START	BASE2 = BASE0 + 74709000
0x3133b62c	DOCUMENT_WRITE_1	BASE2 + 16
0x3133b640	DOCUMENT_WRITE_2	BASE2 + 36
0x2cbfbe14	KERNEL_ALLOCATOR_FUNCTION	BASE0
...	...	...

FIG. 13

<u>Pseudo Binary Code Template 1202</u>	
<u>Push {callSave0, CallSave1, ReturnReg}</u>	
AddWord reg0, ProgCounter, #Const8bits	
LoadWord reg1, [reg0]	
LoadWord reg2, [reg1]	
.....	
<u>LogicalShiftLeft reg1, reg0, #16</u>	
AddWord reg2, reg1, #4	
LoadWord reg3, [reg2]	
AddWord reg4, reg3, ProgCounter	
LoadByte ArgumentReg0, [reg4]	
AddWord ReturnReg, ProgCounter, #8	
BranchDir (ProgCounter + #Const20bits)	
.....	
<u>AddWord reg0, ProgCounter, #Const8bits</u>	
LoadWord reg2, [reg0]	
AddWord reg1, reg2, ProgCounter	
LoadWord ArgumentReg0, [reg1]	
AddWord ReturnReg, ProgCounter, #8	
BranchDir (ProgCounter + #Const20bits)	
Compare ReturnValueReg0, #0	
BranchEQ (ProgCounter + #12)	
Compare ReturnValueReg0, #1	
BranchNE (ProgCounter + #Const5Bits)	
.....	

<u>VIRTUAL ADDRESS-TO-FUNCTION MAPPING TABLE</u> <u>120</u>	
<u>VIRTUAL ADDRESS 302</u>	<u>MACRO REPRESENTING FUNCTIONAL MEANING 304</u>
0x3473fac8	DOCUMENT_WRITE_FUNCTION_START
0x3473fad4	DOCUMENT_WRITE_1
0x3473fae8	DOCUMENT_WRITE_2
0x30000000	KERNEL_ALLOCATOR_FUNCTION

FIG. 14

