Figure 1

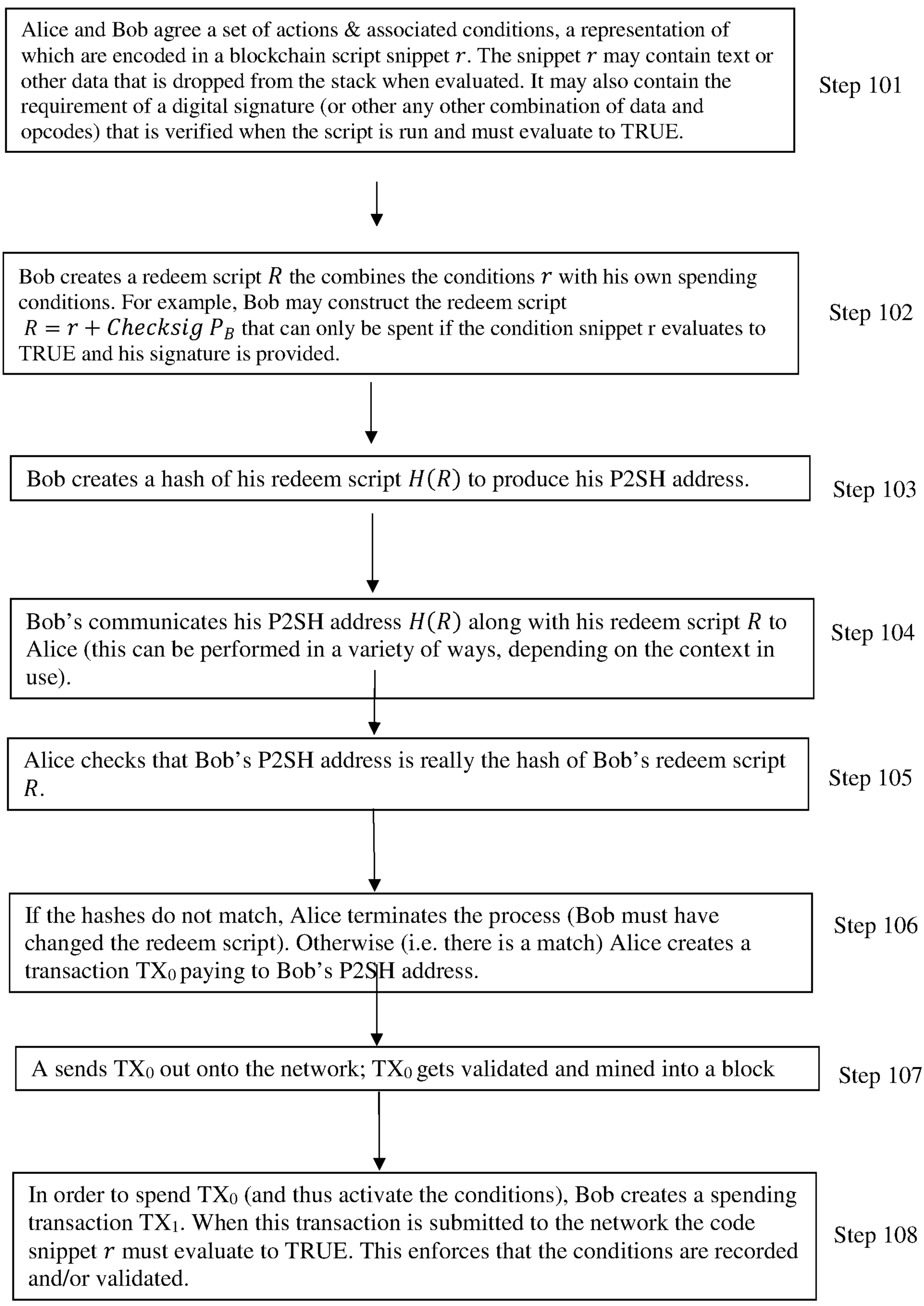
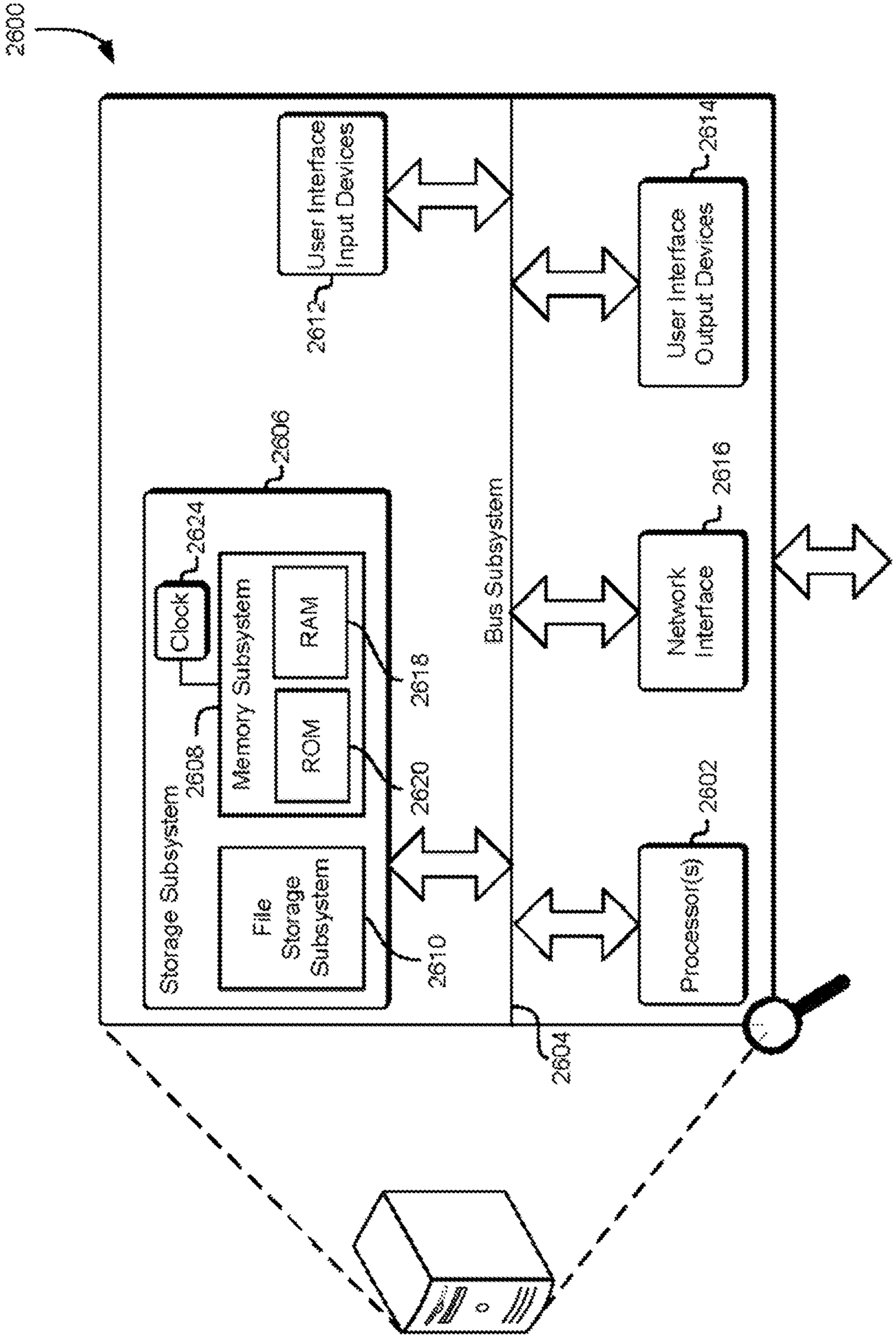

Alice and Bob agree a set of actions & associated conditions, a representation of which are encoded in a blockchain script snippet r. The snippet r may contain text or other data that is dropped from the stack when evaluated. It may also contain the requirement of a digital signature (or other any other combination of data and opcodes) that is verified when the script is run and must evaluate to TRUE.
Step 101
Bob creates a redeem script R the combines the conditions r with his own spending conditions. For example, Bob may construct the redeem script $R = r + Checksig\ P_B$ that can only be spent if the condition snippet r evaluates to TRUE and his signature is provided.
Step 102
Bob creates a hash of his redeem script $H(R)$ to produce his P2SH address.
Step 103
Bob's communicates his P2SH address $H(R)$ along with his redeem script R to Alice (this can be performed in a variety of ways, depending on the context in use).
Step 104
Alice checks that Bob's P2SH address is really the hash of Bob's redeem script R.
Step 105
If the hashes do not match, Alice terminates the process (Bob must have changed the redeem script). Otherwise (i.e. there is a match) Alice creates a transaction TX_0 paying to Bob's P2SH address.
Step 106
A sends TX_0 out onto the network; TX_0 gets validated and mined into a block
Step 107
In order to spend TX_0 (and thus activate the conditions), Bob creates a spending transaction TX_1. When this transaction is submitted to the network the code snippet r must evaluate to TRUE. This enforces that the conditions are recorded and/or validated.
Step 108

Figure 2


2600
User Interface Input Devices
2612
2606
Clock
2624
Storage Subsystem
2608
Memory Subsystem
RAM
ROM
2618
File Storage Subsystem
2610
2620
Bus Subsystem
User Interface Output Devices
2614
Network Interface
2616
Processor(s)
2602
2604