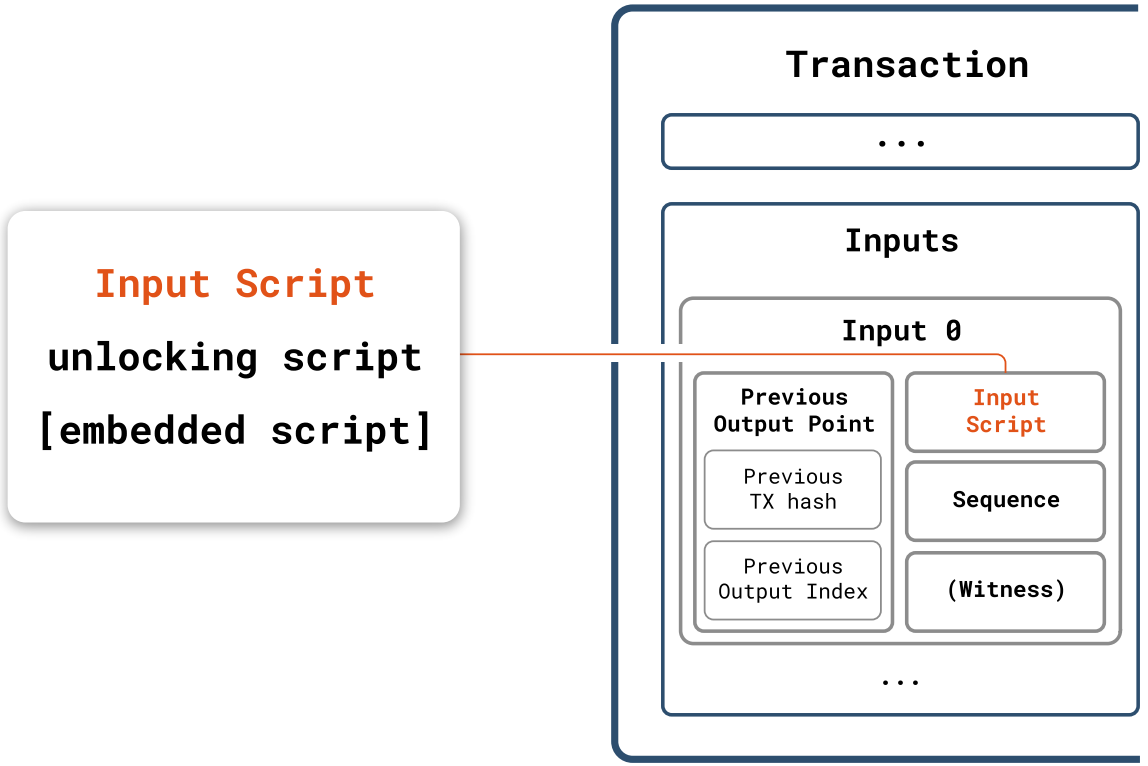
Problems with **complex output scripts**

- Bare multisig has long locking script, with many variable parts (pubkeys)
  - UTXO set bloat
  - Non-addressable
- Sender shouldn't have to know about receiver's spending conditions
- Sender shouldn't need to pay fees for complex spending conditions
- Privacy
- Solution: Pay to an undisclosed script: Pay-to-script-hash
- BIP 16: Activated 2012 to much controversy

P2SH Transaction

Spending TX



** Only partial TX shown*

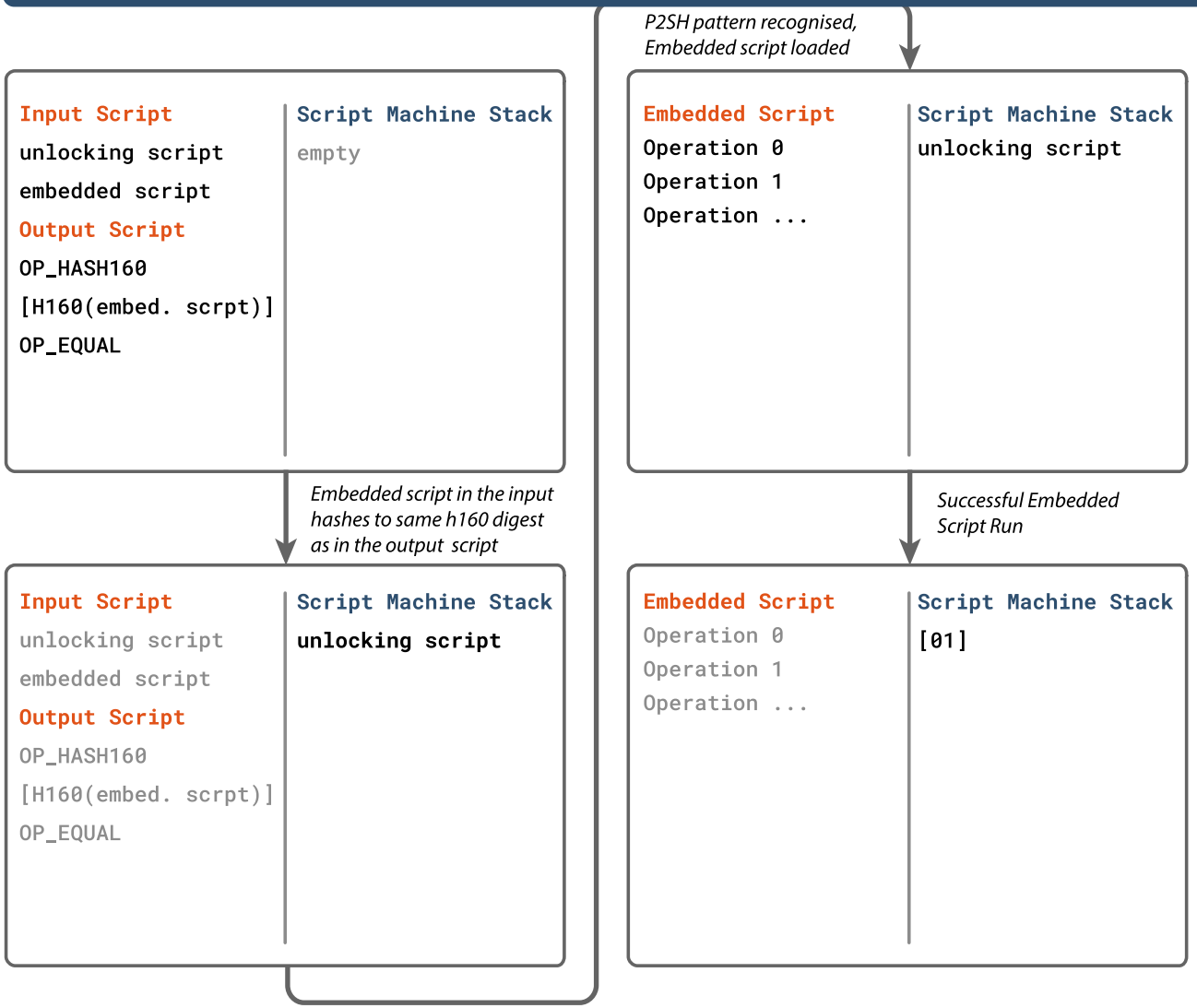
Pay-to-Script-Hash output commits to a specific embedded script.

- Any output script can be embedded into a P2SH output (no P2SH though :)).
- The embedded script must be supplied during spending, but is not disclosed beforehand.

P2SH output is spendable by the embedded script ("redeem script") and its unlocking script

- Embedded script in input is verified against the hash-digest in the output script.
- If preimage verification successful: Embedded script is run separately with the unlocking script operations loaded to the script machine stack.

P2SH Script Run



1) Input & Output Scripts are run

- Embedded script must hash to hash digest in P2SH script.

2) P2SH pattern is recognised

- If embedded script hashes correctly, it is now loaded and run.

3) Embedded script run

- The stack with the remaining unlocking script operations is carried over and the embedded script is run.

4) Final stack evaluation

P2SH addresses

- base58-encode: `[1-byte version][20-byte hash][4-byte checksum]`
- Version prefix (Mainnet/Testnet)
  - 0x05/0xC4
  - 3N5i3Vs9UMyjYbBCFNQqU3ybSuDepX7oT3
  - 2MzQwSSnBHWHqSAqtTVQ6v47XtaisrJa1Vc