Here is the current version of the deposit contract

github.com

ethereum/eth2.0-specs/blob/9467d492b173b8d55633a16cc734ae0a26c04cf7/deposit_contract/contracts/validator_registration.vy#L93

1. # Emit DepositEvent log
2. amount: bytes[8] = self.to_little_endian_64(deposit_amount)
3. log.DepositEvent(pubkey, withdrawal_credentials, amount, signature, self.to_little_endian_64(self.deposit_count))
4.
5. # Compute deposit data root (DepositData hash tree root)
6. zero_bytes32: bytes32 = 0x0000000000000000000000000000000000000000000000000000000000000000
7. pubkey_root: bytes32 = sha256(concat(pubkey, slice(zero_bytes32, start=0, len=64 - PUBKEY_LENGTH)))
8. signature_root: bytes32 = sha256(concat(
9. sha256(slice(signature, start=0, len=64)),
10. sha256(concat(slice(signature, start=64, len=SIGNATURE_LENGTH - 64), zero_bytes32)),
11.))
12. node: bytes32 = sha256(concat(
13. sha256(concat(pubkey_root, withdrawal_credentials)),
14. sha256(concat(amount, slice(zero_bytes32, start=0, len=32 - AMOUNT_LENGTH), signature_root)),
15.))
16. # Verify computed and expected deposit data roots match
17. assert node == deposit_data_root
18.
19. # Add deposit data root to Merkle tree (update a single branch node)
20. self.deposit_count += 1
21. size: uint256 = self.deposit_count

The question becomes, if the deposit contract needs to be modified in case a two-way bridge is implemented.

The first downside of the current implementation, is that money gets stuck in the contract forever. There is no way to get it out. Therefore, to move it back from ETH2 it will need to be printed again. This means that ETH1 will need to provide and unlimited money printing privileges to ETH2. A hack of ETH2 will kill ETH1.

The second downside, is that mining rewards printed on the ETH2 network need to be accepted to unlimited extent on ETH1. This again leads to ETH1 having to fully rely on ETH2 security.

A much more secure alternative in my view would be to print ETH2 mining rewards on ETH1 and modify the deposit contract in such a way that deposits could be withdrawn.

In such a case the security of the ETH1 network would be fully protected and an attacker would not be able to compromise

ETH1 through ETH2.