2. Helping the Network Scale

with **Secret HEAPS**

Secret HEAPS

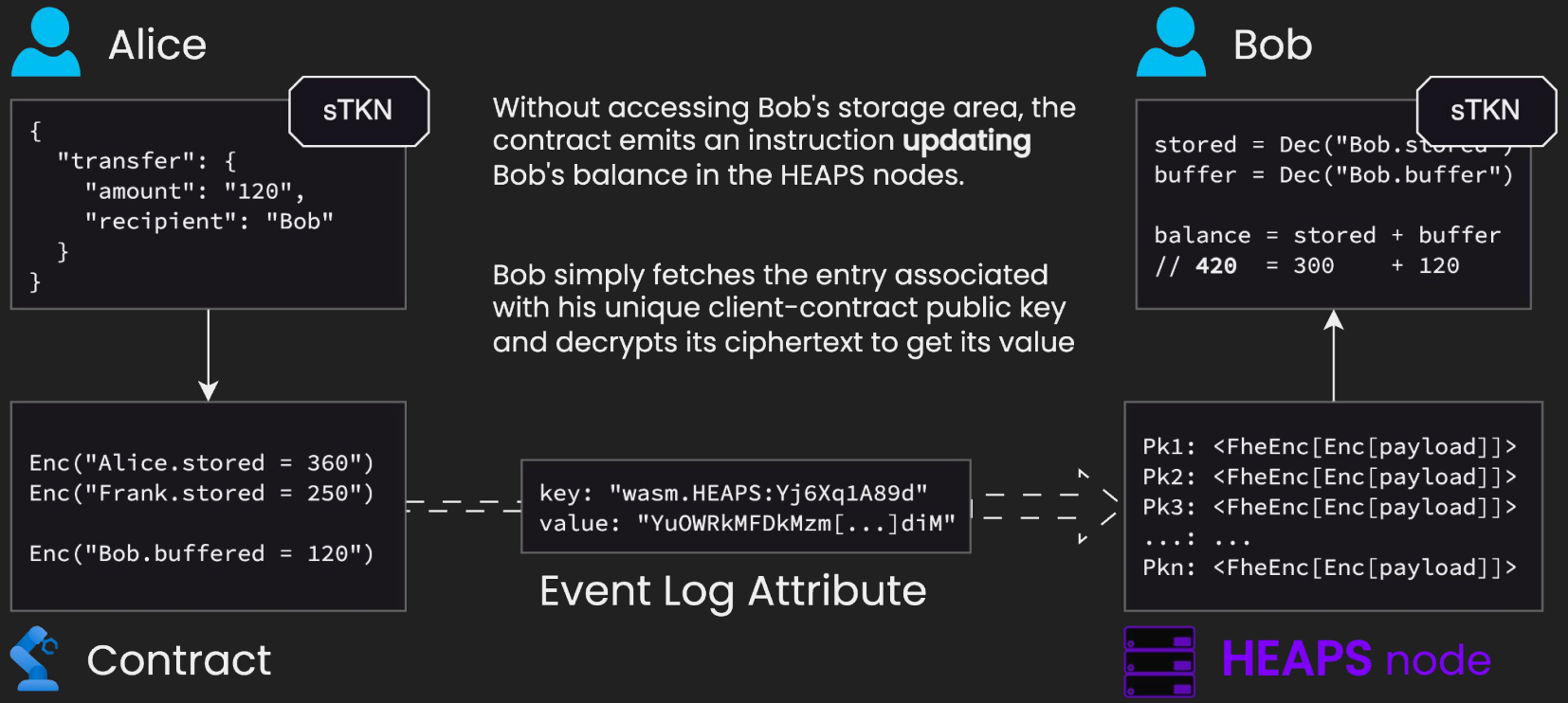
H.E.A.P.S. *- (Fully) **H**omomorphically-**E**ncrypted & **A**uthenticated **P**rivate **S**torage*

An L2 storage mechanism for Secret contracts that is:

- **private** - capable of storing sensitive user data
- **unidirectional** - contracts write, clients read
- **authenticated** - impossible for HEAPS nodes to deceive clients
- **oblivious** - free of storage access patterns

HEAPS provides an alternative source from which clients can retrieve private data that is associated with their account, instead of using SGX nodes to query the contract.

Secret HEAPS



Secret HEAPS

FHE simply guarantees **no storage access patterns**. Each instruction only changes a single entry's "plaintext" value, but affects all ciphertexts.

Data **integrity** is preserved w/ ChaCha20-Poly1305, ensuring that HEAPS nodes **cannot** possibly deceive clients, e.g., by running false circuits, replaying instructions, etc.

Storage retrieval is **quick**; simple key lookup followed by decryption.

HEAPS schema example
for buffered private balances

The FHE layer prevents storage access patterns.
The inner AEAD layer protects the private data.

Bob

```
[storedC, bufferedC] = FheDec(value, secretKey)

balance = Dec(storedC.data, seed, storedC.aad)
        + Dec(bufferedC.data, seed, bufferedC.aad)
```

Recipient is only party able to decrypt their data.
Client cryptographically verifies that the contract stated the given balance and buffer at the given block height and transaction hash.

starshell.net