User Identity Verification & Authentication

- Team name :

Problem Statement

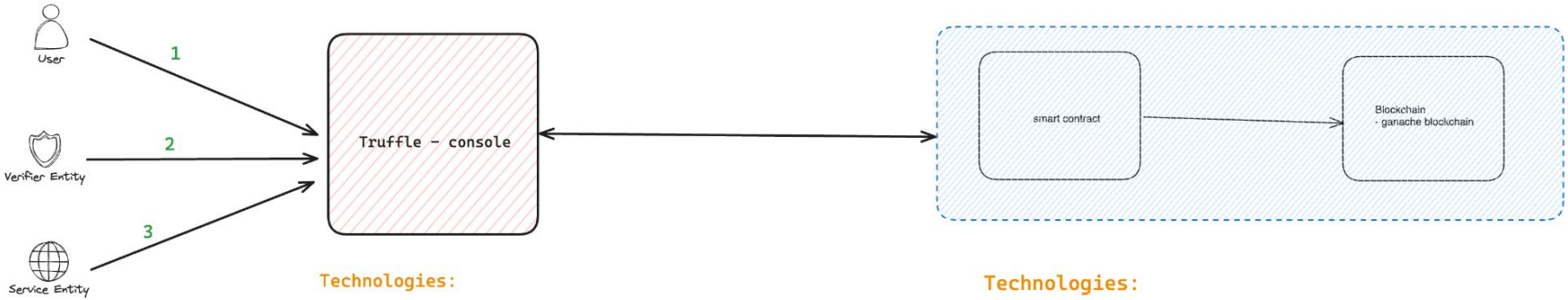
Use case

- **1 Identity Verification Contract**
  - Stores users' hashed identity information.
    - (currently we are storing whole user data in blockchain)
  - Functions to add, update, and retrieve identity information.
  - Functions to request and grant access to identity information.

- **2 Consent Management Contract**
  - Manages consents given by users to institutions to access their identity information.
  - Functions to request consent, grant consent, and check consent status.

- **3 User Consent Mechanism**
  - **Request Consent:** Financial institutions or service providers initiate a consent request through the Consent Management Contract when they need to access a user's identity information. (future use case)
  - **Grant Consent:** Users grant consent by interacting with the Consent Management Contract. This could involve signing a transaction that records their consent on the blockchain.
  - **Check Consent:** Before accessing a user's identity information, institutions must check the consent status through the Consent Management Contract to ensure that consent has been granted.

High Level Architecture Diagram - Phase 1



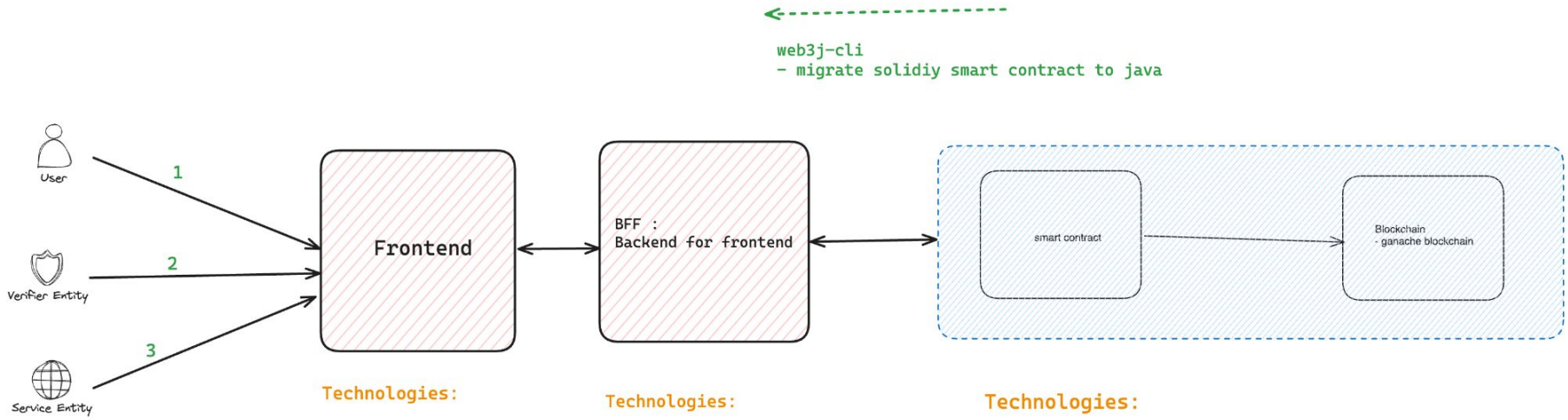
Technologies:

- Truffle-console CLI

Technologies:

- blockchain : ganache local blockchain
- smart contract : truffle, solidity language

High Level Architecture Diagram - Phase 2



Technologies:

- Vaadin framework

Technologies:

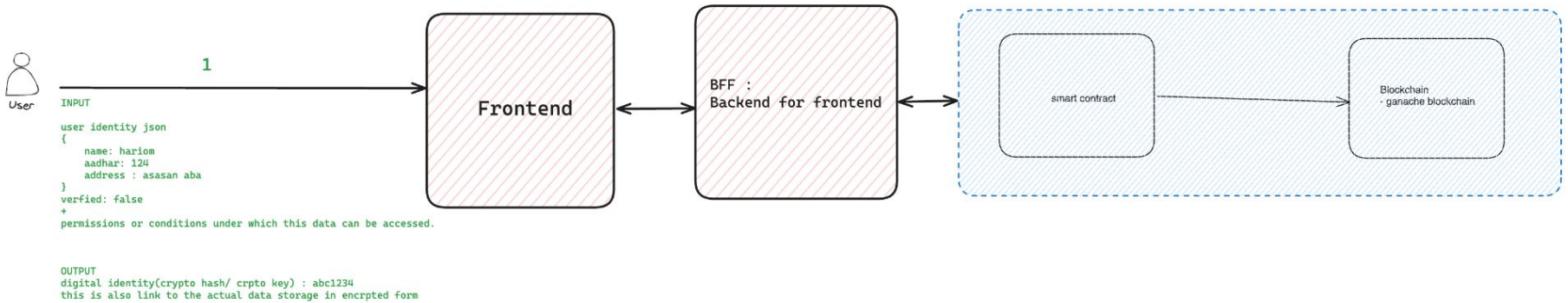
- Spring boot
- java 21

- important dependencies
  - webj for smart contract
  - vaadin for frontend

Technologies:

- blockchain : ganache local blockchain
- smart contract : truffle, solidity language

Use case - step 1

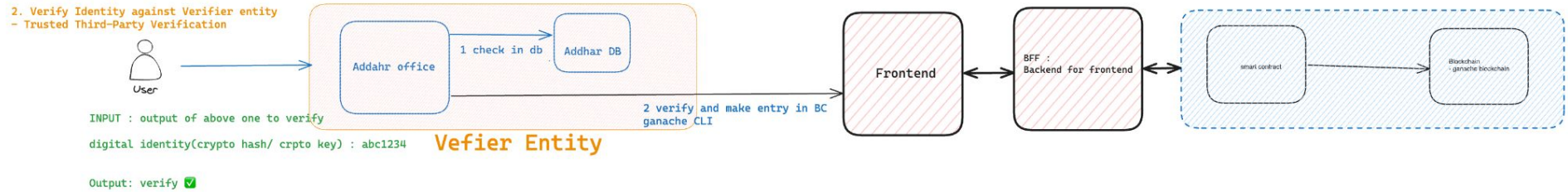


INPUT

user identity json
{
    name: hariom
    aadhar: 124
    address : asasan aba
}
verfied: false
+
permissions or conditions under which this data can be accessed.

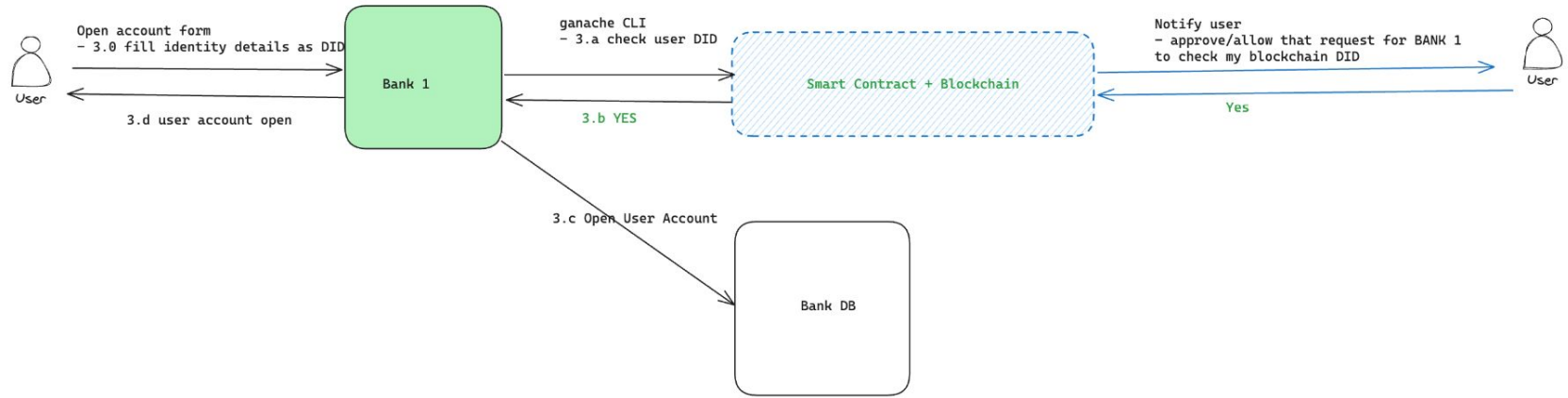
OUTPUT
digital identity(crypto hash/ crpto key) : abc1234
this is also link to the actual data storage in encrpted form

Use case - step 2

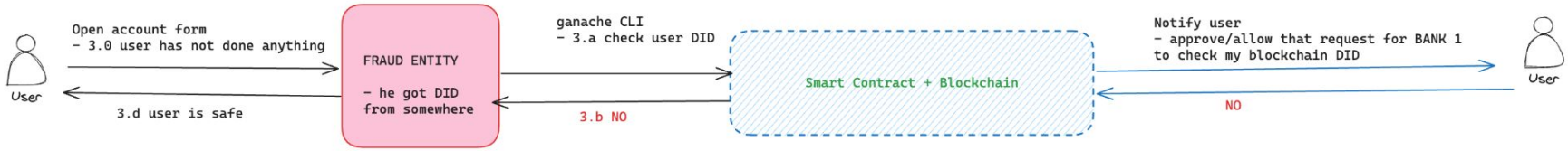


Use case

3. User uses verified DID different place like BANK

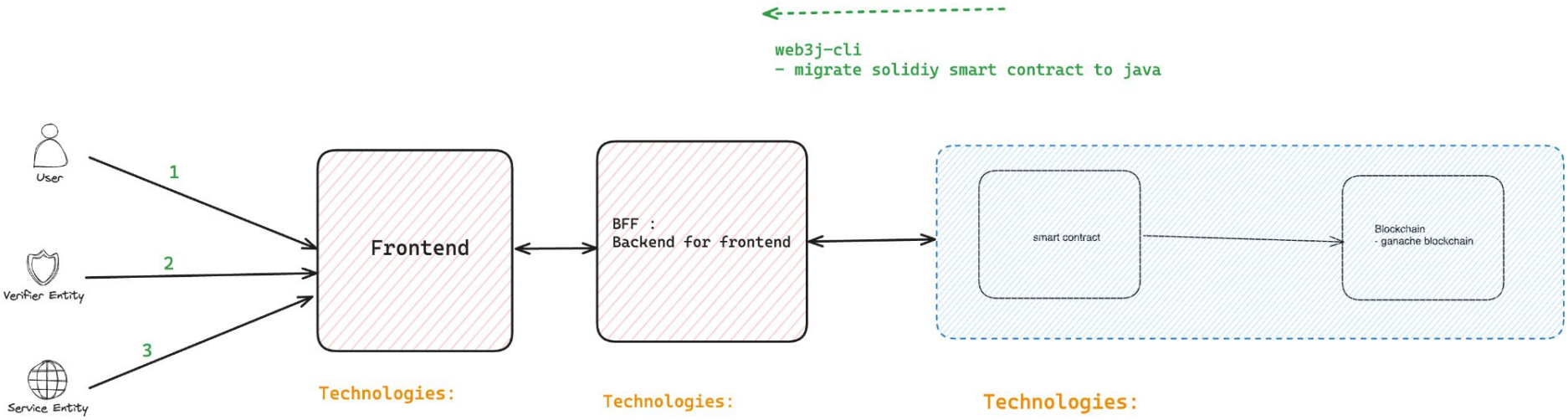


Use case


User
Open account form
- 3.0 user has not done anything
3.d user is safe
FRAUD ENTITY
- he got DID
from somewhere
ganache CLI
- 3.a check user DID
3.b NO
Smart Contract + Blockchain
Notify user
- approve/allow that request for BANK 1
to check my blockchain DID
NO
User

Demo

End to End Integration & Code Implementation



Q & A