

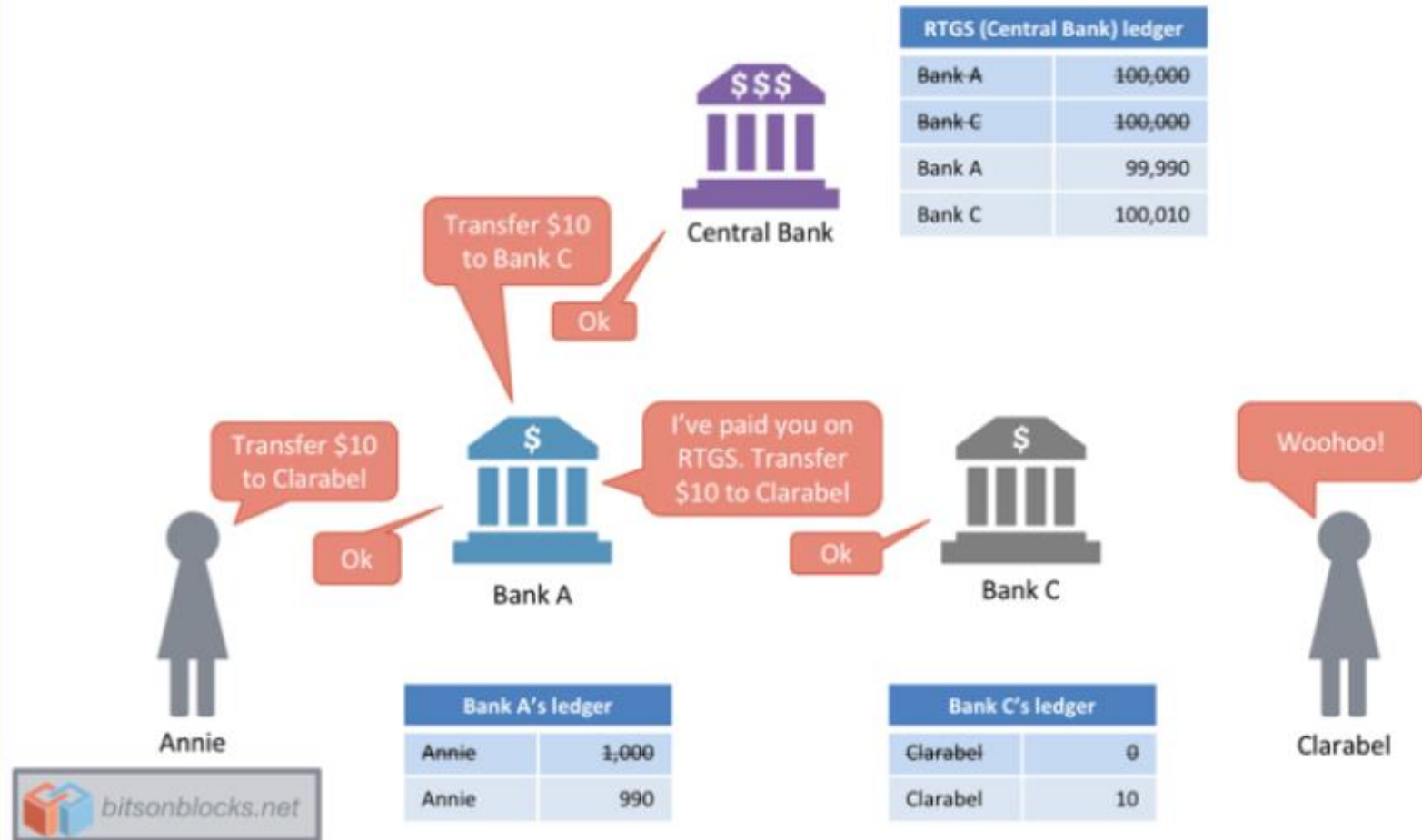
# BLOCKCHAIN HOMEWORK

## INTERBANK TRANSFER

---

# RTGS (CHATS in Hong Kong)

Another solution: Central bank real time gross settlement (RTGS) systems



# Functional Specification

- Deploy a smart contract to simulate a Central Bank RTGS
- Deploy multiple smart contracts to simulate Banks

## Use Cases:

- *registerRTGS* register bank at RTGS and initialize account balance
- *ledgerRTGS* display ledger of RTGS
- *openAccount*: create account at bank if account does not exist
- *deposit*: deposit x value from user's wallet if account exists
- *withdraw*: withdraw y value from user's balance
- *balance*: output the user's current balance
- *ledger*: display ledger of bank
- *transfer*: with an account at Bank1, transfer x value to another account at Bank2  
need to update RTGS accounts
- *closeBank*: bank owner can return all funding back to their users (including owner)

# Some Checking Required

- Require initial funding of bank 5 ether (parameter)
- Bank register RTGS with initial funding 3 ether (parameter)
- Ensure there is enough funding before any transfer
- Every deposit, withdraw will emit AuditLog
- Only bank creator can close the bank
- Throw error when parameters do not pass checking
- Assume maximum of 10 accounts are allowed

# IDE Environment

- Remix - <https://remix.ethereum.org/>

The screenshot displays the Remix IDE interface, which is used for developing and deploying smart contracts. The interface is divided into several panels:

- Left Panel (Tools):** Contains icons for various tools including a compiler, debugger, and file explorer.
- Top Left Panel (Deploy & Run Transactions):**
  - ENVIRONMENT:** Set to JavaScript VM.
  - ACCOUNT:** 0x5B3...edc4 (99.999999% balance).
  - GAS LIMIT:** 3000000.
  - VALUE:** 0 wei.
  - CONTRACT:** MyContract - browser/Add.sol.
  - Buttons:** Deploy, Publish to IPFS, and At Address (Load contract from Address).
  - Transactions recorded:** 1.
  - Deployed Contracts:** MYCONTRACT AT 0XD91...39138 (MEMORY).
  - Low level interactions:** Includes a CALLDATA field and a Transact button.
- Top Right Panel (Code Editor):** Displays the Solidity code for the contract:

```
1 pragma solidity ^0.5.11;
2
3 contract MyContract {
4     event Log(address addr);
5
6     constructor() public {
7         emit Log(address(this));
8     }
9
10    function add(uint256 a, uint256 b) public payable returns (uint256) {
11        return a + b;
12    }
13 }
14
```
- Bottom Panel (Terminal):** Shows the Remix version (0.10.5) and provides instructions for using the terminal, such as checking transactions and executing JavaScript scripts.

# Deploy Example

DEPLOY & RUN TRANSACTIONS

Environment: JavaScript VM

Account: 0xCA3...a733c (100 ether)

Gas limit: 3000000

Value: 5 ether

BlockBank - browser/BlockBank.sol

Deploy 5000000000000000000

or

At Address Load contract from Address

Transactions recorded: 0

Deployed Contracts

Currently you have no contract instances to interact with.

**status** 0x1 Transaction mined and execution succeed

**transaction hash** 0xf90d09ea7ff1d3bf14d51be57e16ac533bdf9f38e8e86cbab706cff5c0592b5f

**contract address** 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a

**from** 0xca35b7d915458ef540ade6068dfe2f44e8fa733c

**to** BlockBank.(constructor)

**gas** 3000000 gas

**transaction cost** 1014402 gas

**execution cost** 715630 gas

```
[
  {
    "from": "0x692a70d2e424a56d2c6c27aa97d1a86395877b3a",
    "topic": "0xc11ff3038529d93428d78e9b5e9db25891866213e7bfe9dde0f7bf79397c2edf",
    "event": "AuditLog",
    "args": {
      "0": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
      "1": "5000000000000000000",
      "accountAddress": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
      "amount": "5000000000000000000",
      "length": 2
    }
  }
]
```

**logs**

**value** 5000000000000000000 wei

# Deposit Example

DEPLOY & RUN TRANSACTIONS

Environment: JavaScript VM

Account: 0x4B0...4D2dB (100 ether)

Gas limit: 3000000

Value: 2 ether

BlockBank - browser/BlockBank.sol

Deploy 5000000000000000000

At Address Load contract from Address

Transactions recorded: 1

Deployed Contracts

BlockBank at 0x692...77b3A (memory)

(fallback)

borrow uint256\_borrow\_amount

closeBank

deposit

withdraw uint256\_withdraw\_amount

DEPLOY & RUN TRANSACTIONS

Environment: JavaScript VM

Account: 0x4B0...4D2dB (97.99999999999999 ether)

Gas limit: 3000000

Value: 0 ether

BlockBank - browser/BlockBank.sol

Deploy 5000000000000000000

At Address Load contract from Address

Transactions recorded: 2

Deployed Contracts

BlockBank at 0x692...77b3A (memory)

(fallback)

borrow uint256\_borrow\_amount

closeBank

deposit

withdraw uint256\_withdraw\_amount

balance

0: int256: 2000000000000000000

```
[
  {
    "from": "0x692a70d2e424a56d2c6c27aa97d1a86395877b3a",
    "topic": "0xc11ff3038529d93428d78e9b5e9db25891866213e7bfe9dde0f7bf79397c2edf",
    "event": "AuditLog",
    "args": {
      "0": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
      "1": "2000000000000000000",
      "accountAddress": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
      "amount": "2000000000000000000",
      "length": 2
    }
  }
]
```

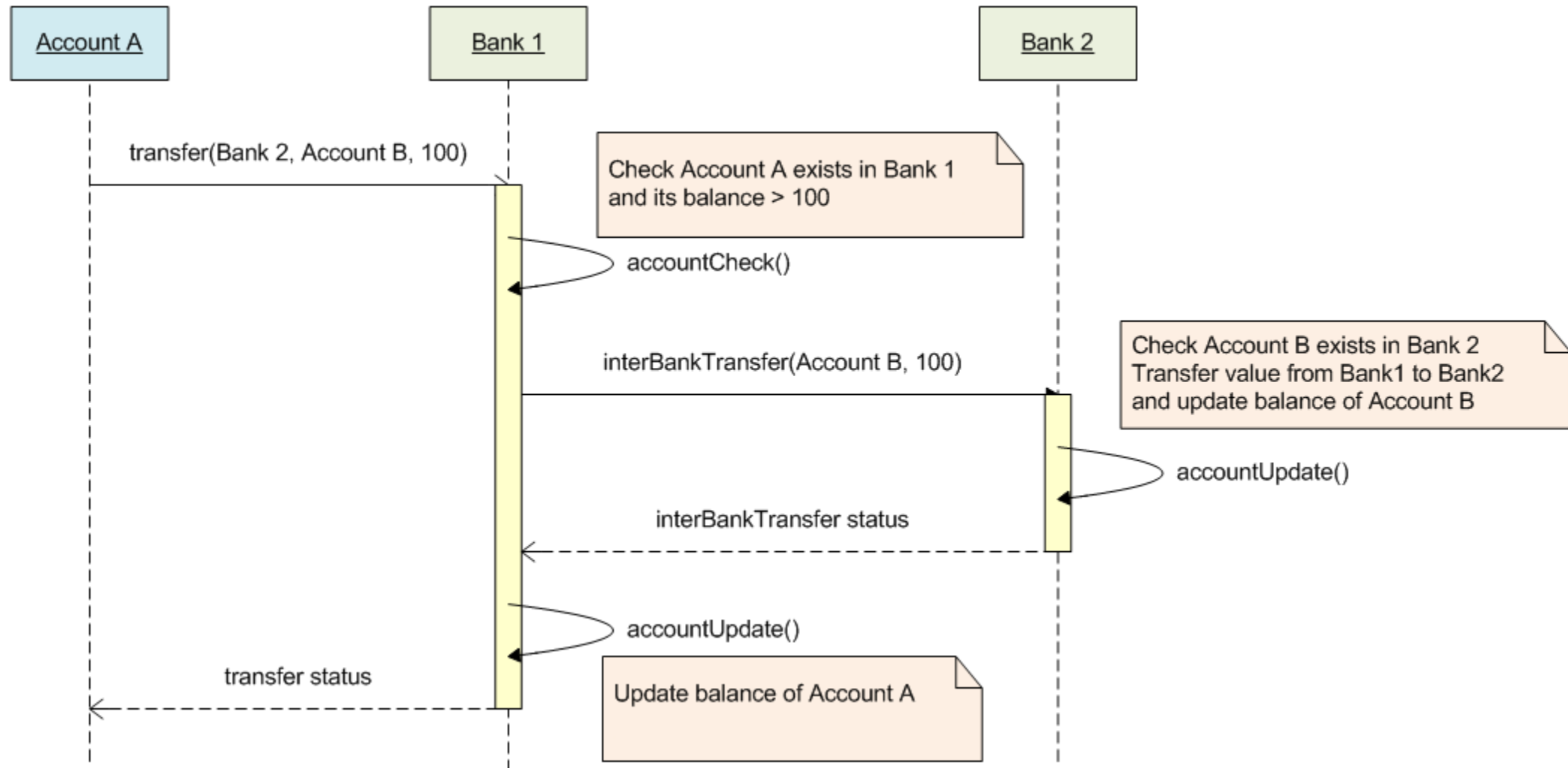
# Withdraw Example

```
[
{
  "from": "0x692a70d2e424a56d2c6c27aa97d1a86395877b3a",
  "topic": "0xc11ff3038529d93428d78e9b5e9db25891866213e7bfe9dde0f7bf79397c2edf",
  "event": "AuditLog",
  "args": {
    "0": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
    "1": "10000000000000000000",
    "accountAddress": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB",
    "amount": "10000000000000000000",
    "length": 2
  }
}
]
```



# Direct Bank Transfer Example

## Note that you need RTGS in this Homework



# Submissions

- The .sol file containing smart contracts
  - Inline documents expected
- A Word file clearly illustrating (with screen caps) of the following scenario:
  - Deploy RTGS
  - Deploy Bank1, Bank2
  - Register Bank1 and Bank2 with RTGS
  - Show ledger of RTGS
  - Open Account1 and Account2 with Bank1 and Bank2 respectively
  - Deposit in Bank1 and Bank2
  - Show ledgers of Bank1 and Bank2
  - Transfer money from Account1 to Account2
  - Show ledger of Bank1 and Bank2
  - Show ledger of RTGS