DA51 Lab Session 7: Dapp Scientific Prepublication

Description:

Lab 7 aims to build a scientific prepublication platform using blockchain for document integrity and non-repudiation. The platform will utilize smart contracts for document upload and verification, with testing on a local blockchain environment.

Question 1:

Non-repudiation and integrity are essential in scientific publications to ensure the reliability, trustworthiness, and accountability of the research. Here's how each of these concepts contributes to scientific integrity:

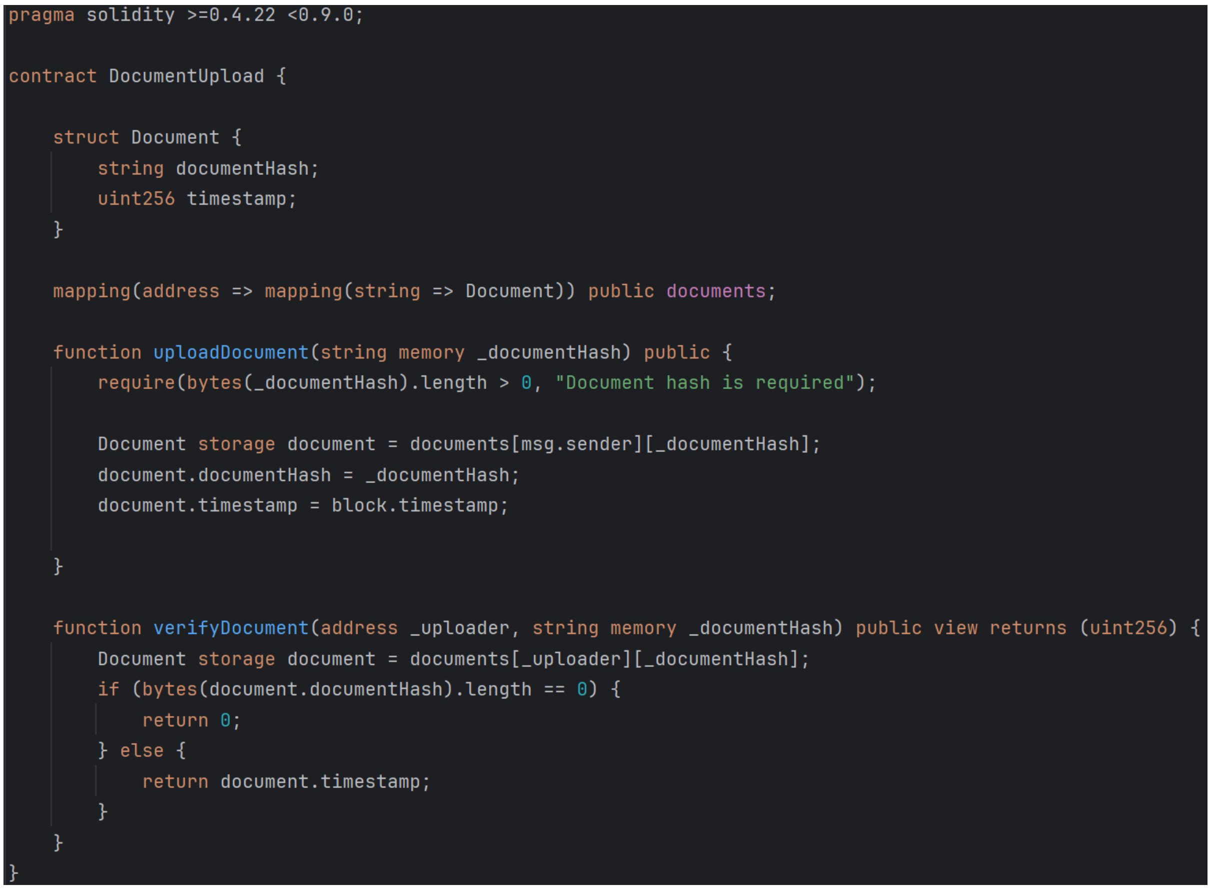
- Non-repudiation in scientific publishing ensures that authors cannot deny their involvement or claims made in their work.
- Integrity in scientific publishing refers to the trustworthiness and authenticity of the data, methods, and conclusions presented in research papers.

Question 2:

Blockchain technology can significantly enhance both non-repudiation and integrity in scientific publications by providing a secure, decentralized, and immutable record of data, authorship, and publication history.

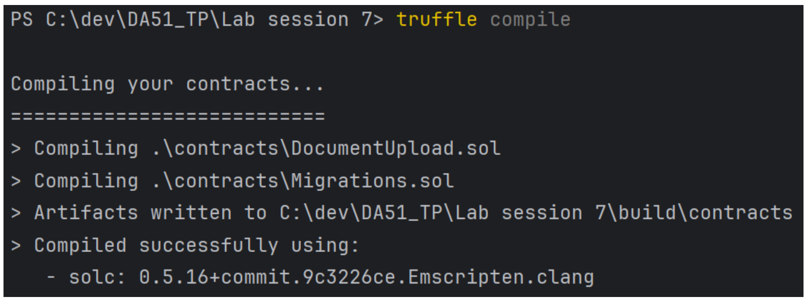
Question 4:

Smart Contract:



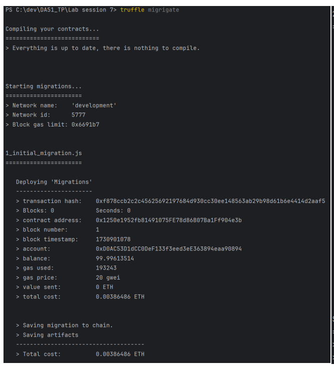
```
pragma solidity >=0.4.22 <0.9.0;

contract DocumentUpload {

    struct Document {
        string documentHash;
        uint256 timestamp;
    }

    mapping(address => mapping(string => Document)) public documents;

    function uploadDocument(string memory _documentHash) public {
        require(bytes(_documentHash).length > 0, "Document hash is required");

        Document storage document = documents[msg.sender][_documentHash];
        document.documentHash = _documentHash;
        document.timestamp = block.timestamp;

    }

    function verifyDocument(address _uploader, string memory _documentHash) public view returns (uint256) {
        Document storage document = documents[_uploader][_documentHash];
        if (bytes(document.documentHash).length == 0) {
            return 0;
        } else {
            return document.timestamp;
        }
    }
}
```

Compile:

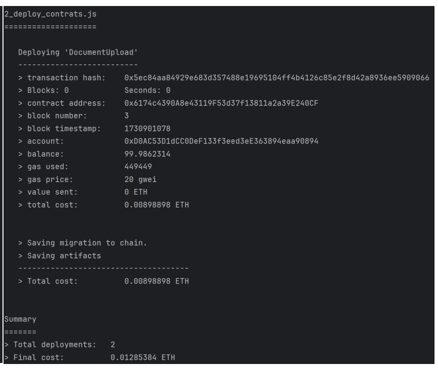


```
PS C:\dev\DA51_TP\Lab session 7> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\DocumentUpload.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\dev\DA51_TP\Lab session 7\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

Migrate:



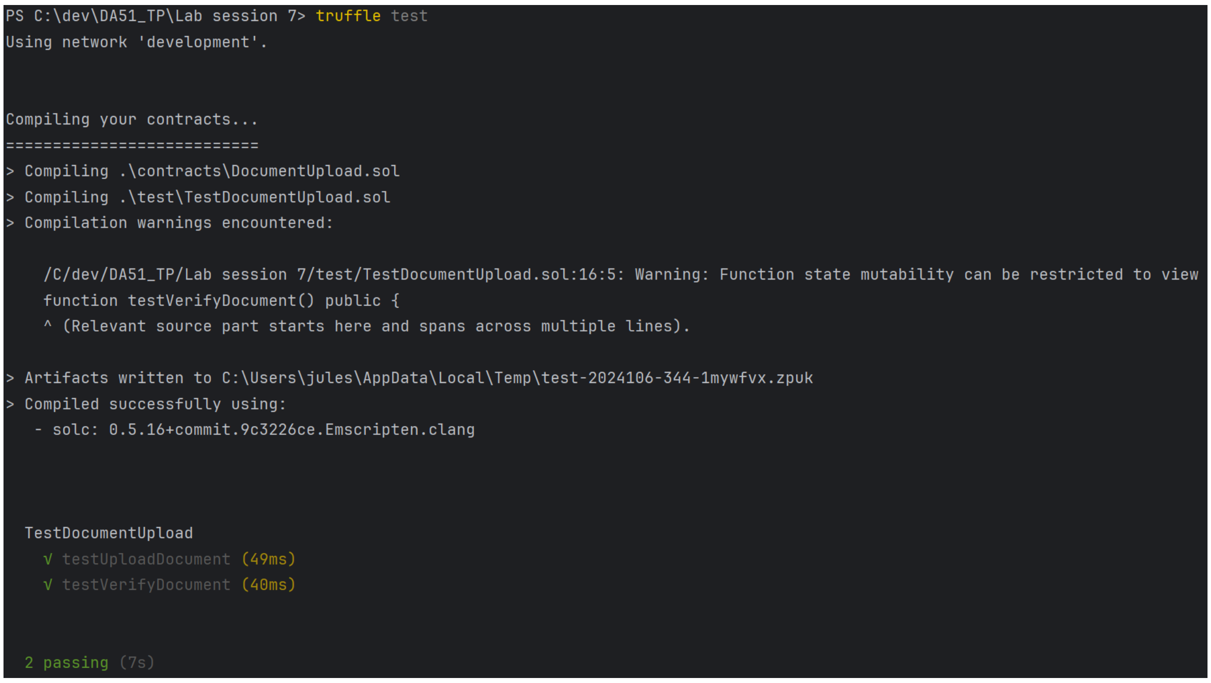
```
PS C:\dev\DA51_TP\Lab session 7> truffle migrigate

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 0x6691b7


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0xf878ccb2c2c45625692197684d930cc30ee148563ab29b98d61b6e4414d2aaf5
   > Blocks: 0            Seconds: 0
   > contract address:    0x1250e1952fb81491075FE78d86B07Ba1Ff904e3b
   > block number:        1
   > block timestamp:     1730901078
   > account:             0xD0AC53D1dCC0DeF133f3eed3eE363894eaa90894
   > balance:             99.99613514
   > gas used:            193243
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00386486 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00386486 ETH
```



```
2_deploy_contrats.js
====================

   Deploying 'DocumentUpload'
   --------------------------
   > transaction hash:    0x5ec84aa84929e683d357488e19695104ff4b4126c85e2f8d42a8936ee5909066
   > Blocks: 0            Seconds: 0
   > contract address:    0x6174c4390A8e43119F53d37f13811a2a39E240CF
   > block number:        3
   > block timestamp:     1730901078
   > account:             0xD0AC53D1dCC0DeF133f3eed3eE363894eaa90894
   > balance:             99.9862314
   > gas used:            449449
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00898898 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00898898 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.01285384 ETH
```

Question 5:

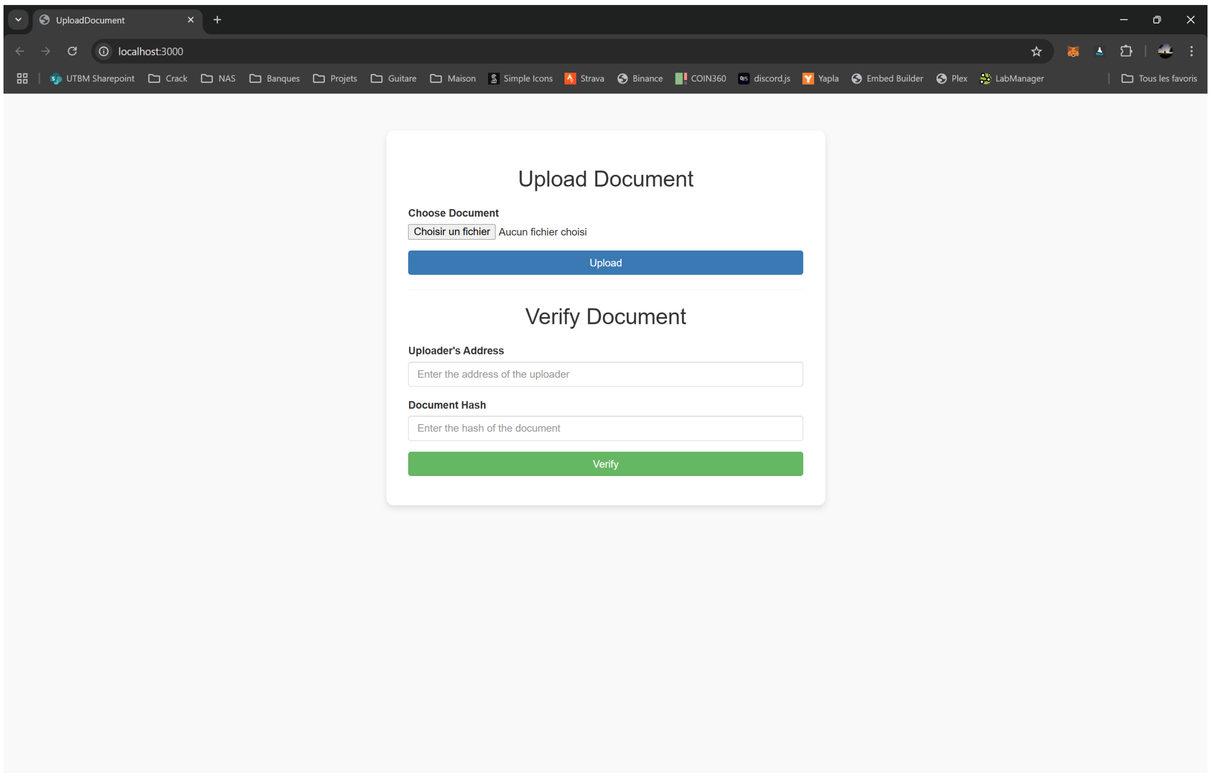
Tests:



```
PS C:\dev\DA51_TP\Lab session 7> truffle test
Using network 'development'.


Compiling your contracts...
===========================
> Compiling .\contracts\DocumentUpload.sol
> Compiling .\test\TestDocumentUpload.sol
> Compilation warnings encountered:

    /C/dev/DA51_TP/Lab session 7/test/TestDocumentUpload.sol:16:5: Warning: Function state mutability can be restricted to view
    function testVerifyDocument() public {
    ^ (Relevant source part starts here and spans across multiple lines).

> Artifacts written to C:\Users\jules\AppData\Local\Temp\test-2024106-344-1mywfvx.zpuk
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang



  TestDocumentUpload
    √ testUploadDocument (49ms)
    √ testVerifyDocument (40ms)


  2 passing (7s)
```

Question 6:

The platform:



Upload Document

Choose Document

Choisir un fichier | Aucun fichier choisi

Upload

Verify Document

Uploader's Address

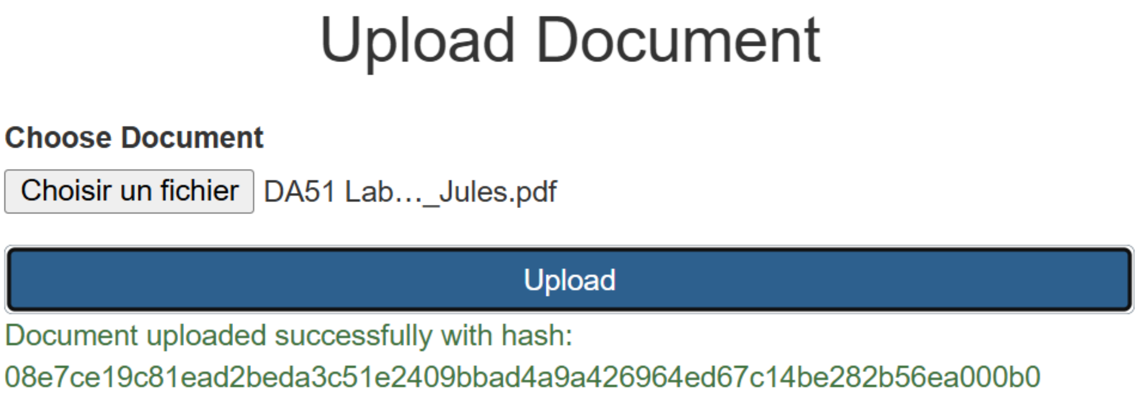
Enter the address of the uploader

Document Hash

Enter the hash of the document

Verify

Upload Document:

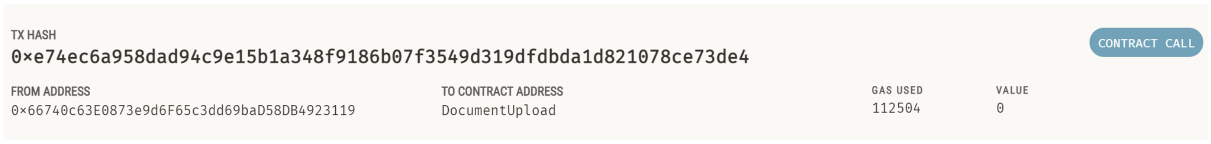


Upload Document

Choose Document

Document uploaded successfully with hash: 08e7ce19c81ead2beda3c51e2409bbad4a9a426964ed67c14be282b56ea000b0

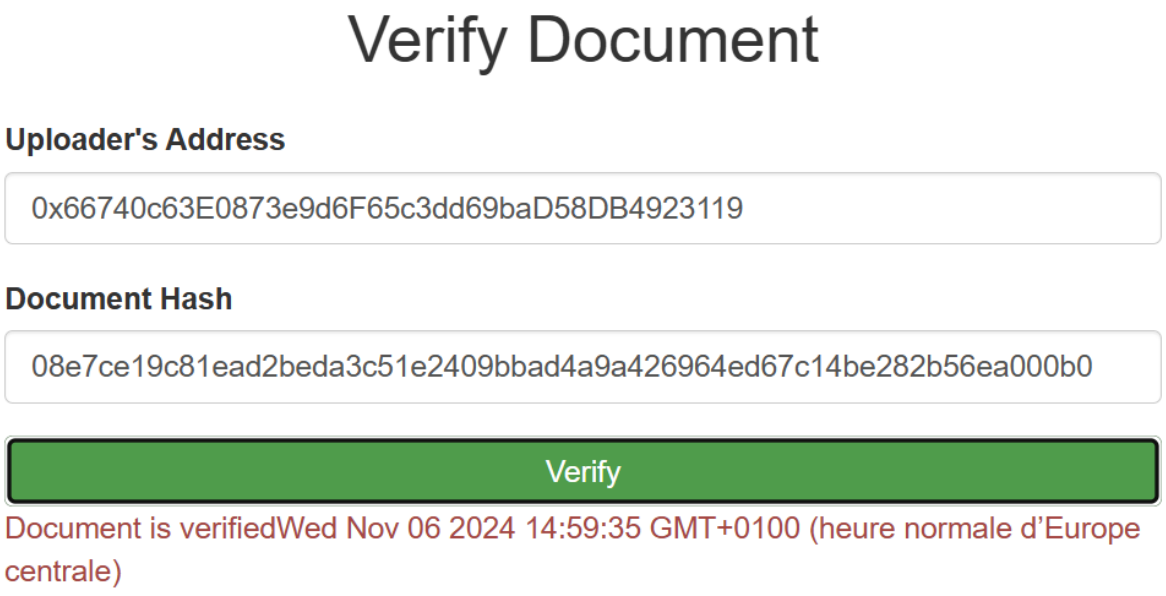
In Ganache:



TX HASH
0×e74ec6a958dad94c9e15b1a348f9186b07f3549d319dfdbda1d821078ce73de4

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0×66740c63E0873e9d6F65c3dd69baD58DB4923119	DocumentUpload	112504	0

Verification:



Verify Document

Uploader's Address: 0x66740c63E0873e9d6F65c3dd69baD58DB4923119

Document Hash: 08e7ce19c81ead2beda3c51e2409bbad4a9a426964ed67c14be282b56ea000b0

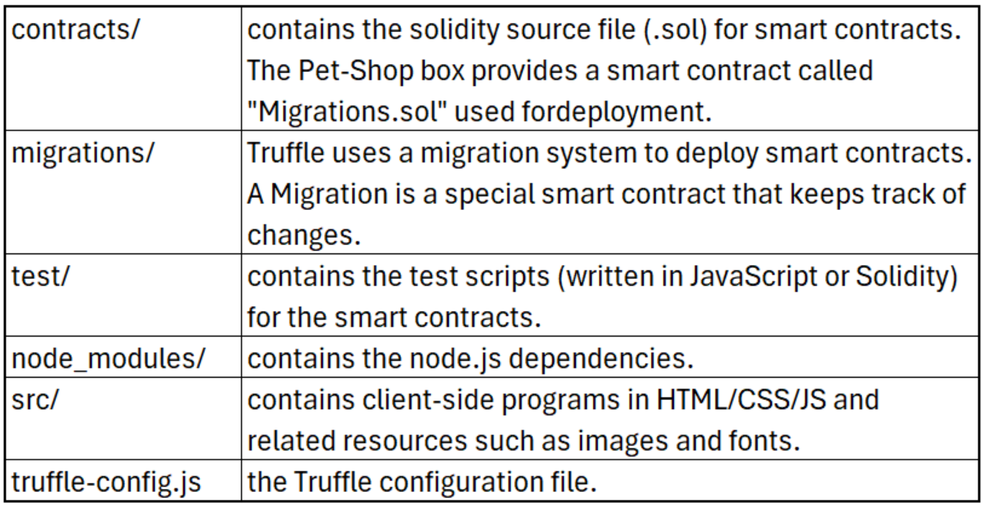
Document is verifiedWed Nov 06 2024 14:59:35 GMT+0100 (heure normale d’Europe centrale)

Question 9:

Using blockchain in scientific prepublication has several advantages and challenges. Prepublication typically involves sharing research data, findings, and methodologies before formal peer-reviewed publication, and blockchain can enhance this process. Here is a list:

- Permanent, Time-Stamped Records
- Improved Data Integrity
- Facilitating Open Science and Collaboration
- Enhanced Peer Review Process
- Protection of Intellectual Property

Question 10:



contracts/	contains the solidity source file (.sol) for smart contracts. The Pet-Shop box provides a smart contract called "Migrations.sol" used fordeployment.
migrations/	Truffle uses a migration system to deploy smart contracts. A Migration is a special smart contract that keeps track of changes.
test/	contains the test scripts (written in JavaScript or Solidity) for the smart contracts.
node_modules/	contains the node.js dependencies.
src/	contains client-side programs in HTML/CSS/JS and related resources such as images and fonts.
truffle-config.js	the Truffle configuration file.