OptiAux
Technologies Pvt.Ltd

AuxCert

Table on Contents

Introduction AuxCert

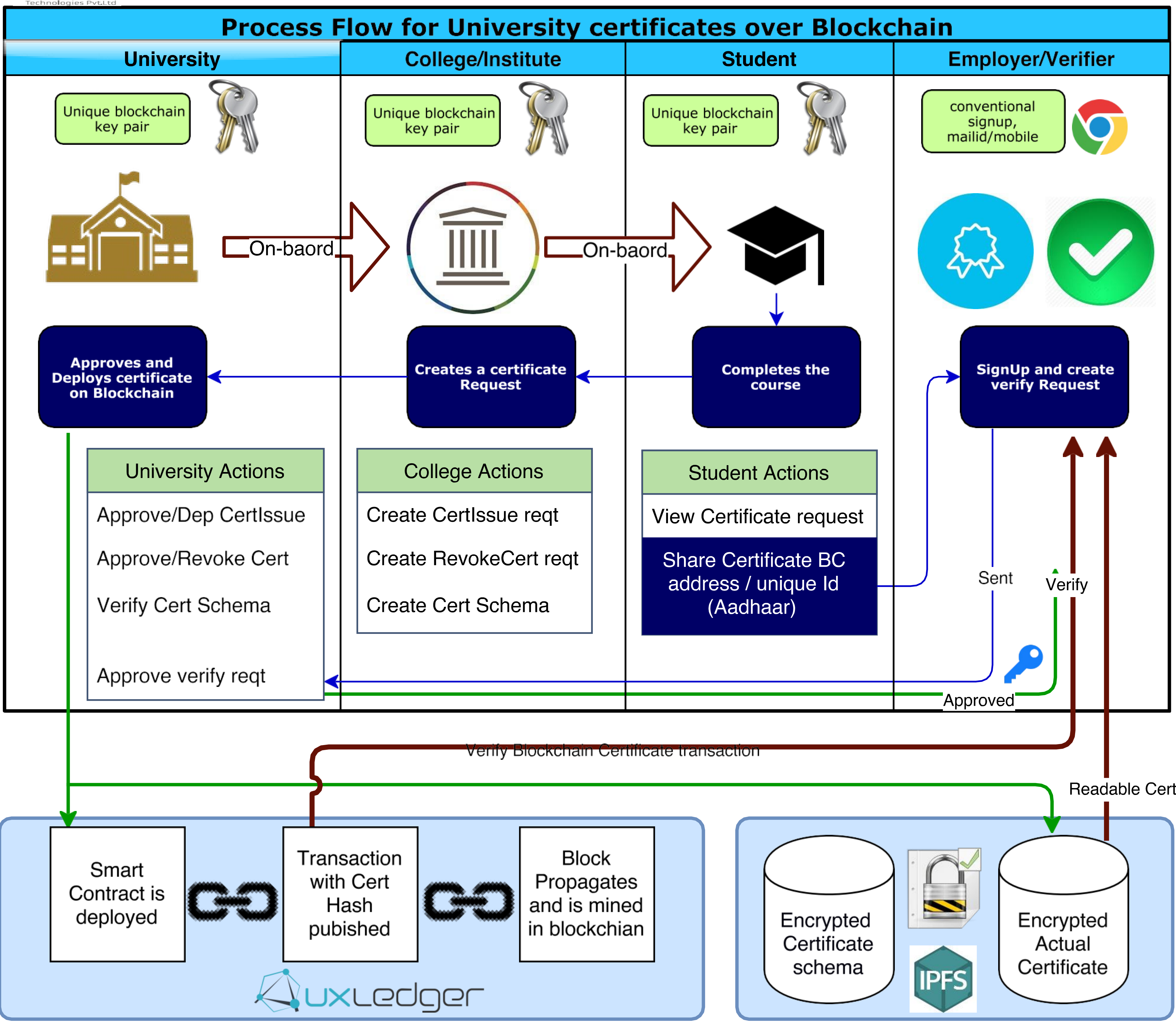
The Certificates over blockchain platform uses MEAN application stack for application layer with distributed layer implemented with Auxledger as the blockchain and IPFS as the file storage system to store encrypted certificates. This platform provides the interface so that the regulatory bodies
Below diagram and process flow depicts the user flow in which a certificate is deployed/verified over blockchain and

How the process works

Step1- The platform admin creates a university login, blockchain based key-pair is generated and shared with university admin.
Step 3- University admin on-boards a college, the on-boarding form/details are sent by college via a operational process.
Blockchain based key-pair is generated and shared with college admin.
Step 4- College on-boards a student, blockchain based key-pair is generated and shared with student.
Step 5- Student completes the course and college creates his certificate and send the approval request to university.
Step 6- University approves the certificate request and deployes a smart contract over blockchain having all the certificate details(like certificate hash, student id, college id, active status)
Step 7- The generated BC certificate smart contract address is sent to University, College and updated in student wallet/view.
Step 8- The student goes to a employer and shares BC certificate smart contract address.
Step 9- The verifier uses a open portal to and completes the login using email/mobile signup. Puts the BC certificate smart contract address and requests the permission to view from university.
Step 10- Once university approves the request the verifier view is directly updated with data from blockchain and the smart contract details as well as the transaction cross-verifying/showing the certificate hash and the time and details of the smart contract deployment.

OptiAux
Technologies Pvt.Ltd

Process Flow for University certificates over Blockchain



University

Unique blockchain key pair

On-baord

Approves and Deploys certificate on Blockchain

University Actions

Approve/Dep CertIssue

Approve/Revoke Cert

Verify Cert Schema

Approve verify reqt

College/Institute

Unique blockchain key pair

On-baord

Creates a certificate Request

College Actions

Create CertIssue reqt

Create RevokeCert reqt

Create Cert Schema

Student

Unique blockchain key pair

Completes the course

Student Actions

View Certificate request

Share Certificate BC address / unique Id (Aadhaar)

Employer/Verifier

conventional signup, mailid/mobile

SignUp and create verify Request

Sent

Verify

Approved

Verify Blockchain Certificate transaction

Readable Cert

Smart Contract is deployed

Transaction with Cert Hash pubished

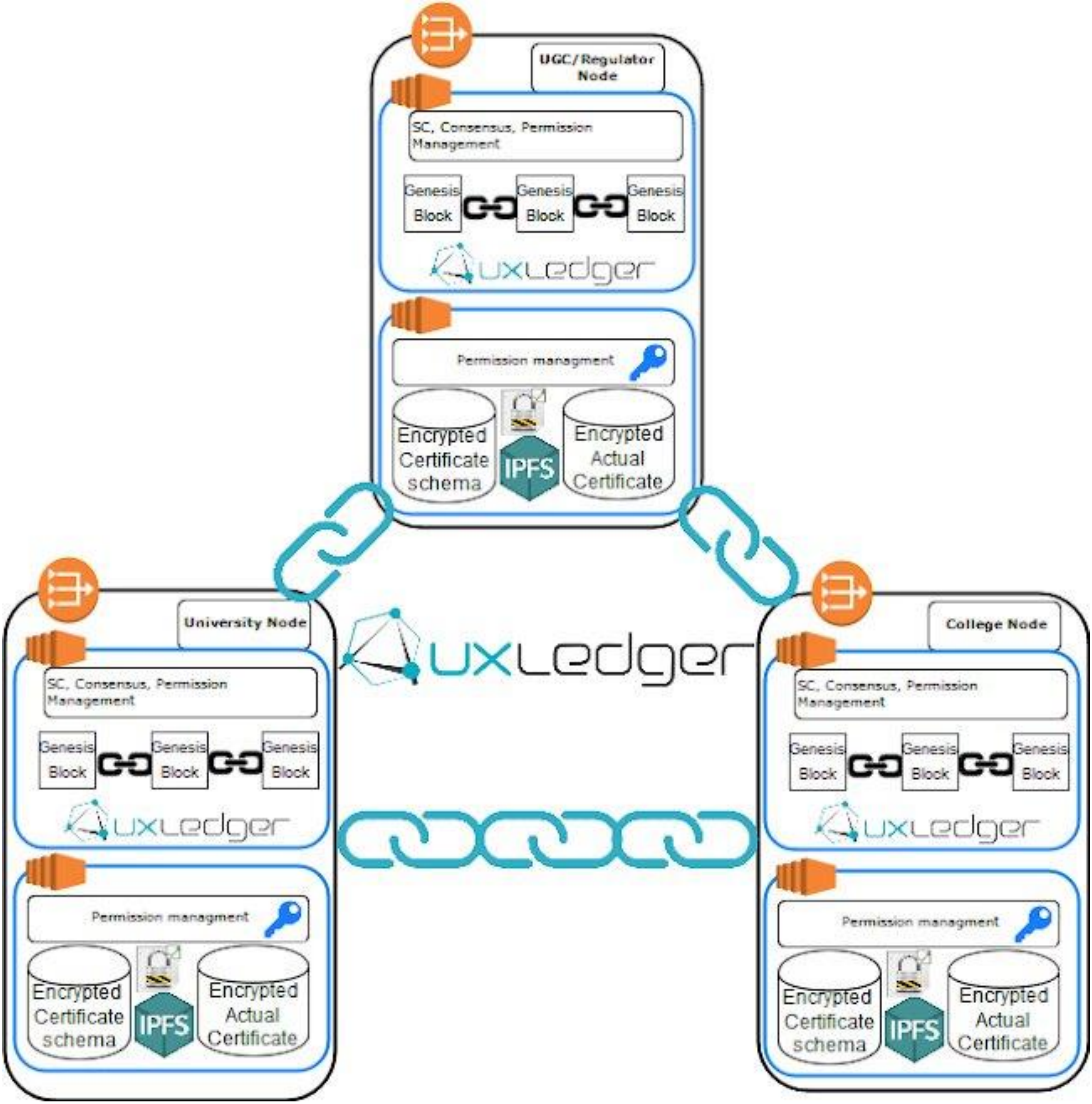
Block Propagates and is mined in blockchian

uxLedger

Encrypted Certificate schema

IPFS

Encrypted Actual Certificate


UGC/Regulator Node
SC, Consensus, Permission Management
Genesis Block
Genesis Block
Genesis Block
uxLedger
Permission managment
Encrypted Certificate schema
IPFS
Encrypted Actual Certificate
uxLedger
University Node
SC, Consensus, Permission Management
Genesis Block
Genesis Block
Genesis Block
uxLedger
Permission managment
Encrypted Certificate schema
IPFS
Encrypted Actual Certificate
College Node
SC, Consensus, Permission Management
Genesis Block
Genesis Block
Genesis Block
uxLedger
Permission managment
Encrypted Certificate schema
IPFS
Encrypted Actual Certificate

Application Components

Certificates over blockchain platform

1. **University view**
   Using this application login and role functions specific to university like on-boarding a college, approving/revoking a certificate are actioned.
   Every university has a unique blockchain key-pair through which it changes blockchain state and adds or accesses data to/from blockchain.

2. **College view**
   Using this application login and role functions specific to college like on-boarding a student, requesting a certificate and creating a certificate schema can be actioned.

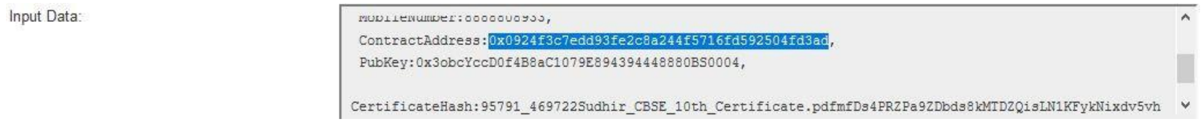
   2.1. **Certificate Schema module**
   This module is accessed by College role to create new certificate for different courses within the college based on Open Badges Standard*(https://openbadges.org/).
   Every college has a unique blockchain key-pair through which it accesses data from blockchain.

3. **Student view/wallet**
   Student can access his details via this module or wallet.
   Student can share his certificate Hash and certificate smart contract address(which is his certificate over blockchain).
   A verifier verify certificate smart contract address and certificate hash in a particular blockchain transaction which has immutable data.
   Every college has a unique blockchain key-pair through which it accesses data from blockchain.

4. **Verifier view**

Verifier can verify the smart contract address and certificate hash by checking these details in Input Data field of a Blockchain transaction, as depicted below.

0xe3f0162ff7e23e78ac7600c31eb04b9a7gfj98218t1f517ec9d48t12 4t154889



Input Data:

```
MobileNumber:8888808933,
ContractAddress:0x0924f3c7edd93fe2c8a244f5716fd592504fd3ad,
PubKey:0x3obcYccD0f4B8aC1079E894394448880BS0004,

CertificateHash:95791_469722Sudhir_CBSE_10th_Certificate.pdfmfDs4PRZPa9ZDbds8kMTDZQisLN1KFykNixdv5vh
```

Fig1

5. **Certificate registration platform**

This app component is on private subnet and is the core of the platform, it has all the business logic and can access DB to provide the functionality for different roles. This component is connected to AWS HSM to process key related activities. It is also connected Blockchain API layer which in-turns communicates with the Blockchain node and the IPFS node.

6. **Blockchain Integration API layer**

We have added this layer so that the core application doesn't hold any blockchain communication related logic. All the blockchain communication is taken care by this API layer.

Distributed application Components

Auxledger blockchain layer

A private permissioned blockchain implementation will be done for this platform. All the blockchain components Smart contract, consensus and permissioning of different nodes will be done by Auxledger.

For every degree certificate issued there will be 2 blockchain transactions:

1st- the Smart contract which will have the specific certificate, student, college details and ABI (Application binary interface) access to this smart will be to university and regulator body like UGC.

There will be only 2 functions allowed via roles-based ABI, 1st to check the details in the Smart Contract and 2nd revoke the certificate.

In case of revocation of certificate, the status in Smart Contract will change to not in effect.

The certificate check ABI can be made open (irrespective of the role) so that any party can check the status of the certificate.

2nd- Once a Smart contract for a certificate is deployed first time, then subsequently a transaction over blockchain is made showing high level data items accessible to all via explorer in human readable format to verify that the certificate hash and the student details do match which is shared by the student.

IPFS distributed storage layer

Private network of IPFS is implemented so that the whole platform one day can be moved to Web3.0 standards in which all the elements of a web application should be distributed.

IPFS stores encrypted actual certificate. The decryption of this certificate is possible only with the permission and key from university.

Also 1 college can have different courses for which it needs to create different certificate schema, these schema's will be stored in IPFS with decryption available by college and university.

IPFS provides a distributed file system to store the certificates so that if 1 node is down still the other nodes can provide the needed data.

These 2 distributed components has to be executed by every participant university and college to participate and support in the distributed consensus mechanism and provide unbiased system as well as storage.