DISTRIBUTED LEDGERS TECHNOLOGY

Research Area

DIGITIZE ACADEMIC CERTIFICATES USING BLOCKCHAIN TO CURB FRAUD:

The case of a Local University in Kenya

Agenda

- Problem definition
- Research Objective
- Proposed Solution
- Blockchain Features
- Design Features for Bitcoin Blockchain
- Proposed Approach

Problem definition and justification

- Worrying rate of the prevalence of fake academic certificates in the country
- The universities' role of issuing, storing and validating certificates is slow and expensive
- Detected during vetting for jobs especially in the public office
- Corruption has contributed to the spread of this malpractice

Research objective

- To digitize university academic certificates in an immutable, non-repudiable and easily verifiable way

Towards the solution

- The learning institutions have to take charge by:
  - Implementing systems that ensure ease of issuing, maintaining, tracking and retrieval of their graduate records and,
  - Allow for effective verification of the authenticity of these records

Proposed Solution

- Using Blockchain technology
  - Brief introduction to blockchain technology
  - Blockchain is a trusted distributed ledger with shared business processes
  - It allows for the creation of permissioned and immutable records

Features of Blockchain that allow the achievement of our goals

- It allows for the creation of permissioned and immutable records
  - Permissioned
  - Distributed (Shared)
  - Provenance and
  - Immutability

With these capabilities we can:

- Created digital certificates
  - Sharable and verifiable online
- Verify existing certificates to ascertain their authenticity
  - Physical and digital (Using the stored hash)
- Expedite reference of peoples credentials and qualifications online
  - From anywhere
  - No need to contact the institution directly – incase the institution closes down

Methodology

- **Agile implementation**

The Agile software methodology was proposed by the Agile team in 2001. It allows for a team of software developers to deliver working software to users at regular short intervals called sprints (Dingsøyr, Nerur, Balijepally, & Moe, 2012).

Blockchain platform decision:

- User base/ popularity
  - Bitcoin
  - Hyperledger
  - Ethereum
- Privacy/ Public
  - Hyperledger vs Ethereum
  - Other smaller platforms not so widely used: e.g. Gravity, Tendermint, Belrium, Hydrachain, IOTA etc

Other Considerations:

- Suitability for certificates/ Document maintenance
- Other institutions using it
- Consensus algorithm
- Supported languages
- Compatibility with different systems
  - Blockcerts

Design decisions for the A Blockchain platform:

- The process will involve no other products but a blockchain
- The process will allow anyone to authenticate a Strathmore University certificate without having to contact Strathmore.
- The whole process is based on document hashes and leverages the merkle root hash principle

Design decisions for the Bitcoin Blockchain:

- The record is stored on the bitcoin blockchain as an unspendable transaction.
- For self verification to work, extra metadata is included along with the certificate containing the merkle root hash, merkle proof, transaction identifier and public address used.
- Validity can be confirmed by comparing the hash of the certificate and the hash stored in the OP_RETURN code in the blockchain transaction at a certain time using an explorer such as https://www.blockchain.com/explorer

Blockcerts

- Blockcerts is an open standard for building apps that issue and verify blockchain-based official records. These may include certificates for civic records, academic credentials, professional licenses, workforce development, and more.
- Opensource
- Versatile - Hashes can be posted to Bitcoin, Ethereum and Hyperledger blockchain platforms [platform lock-up eliminated]
- The above platforms defined on Blockcerts, then the credentials are pushed to the blockchain of choice.
- Other institutions using it: Mit, University of Melbourne, UNIC, Gvt of Malta etc

Certificate tools module on blockcerts

blockchain-certificates / cert-tools

Watch 8 | Star 15 | Fork 33

Code | Issues 2 | Pull requests 0 | Projects 0 | Insights

Branch: master | cert-tools / cert_tools / | Create new file | Find file | History

gulix77 fix displayHtml — Latest commit [illegible] on Jun 28

..		
__init__.py	bump cert-core	4 months ago
create_issuer.py	fix paths	a year ago
create_revocation_addresses.py	uncompressed addresses no longer default. Add pycoin to requirements ...	2 years ago
create_v1_2_certificate_template.py	fix paths	a year ago
create_v2_certificate_template.py	fix displayHtml	3 months ago
create_v2_issuer.py	fix paths	a year ago
extract_links.py	V2 updates (#12)	2 years ago
helpers.py	fix paths	a year ago
instantiate_v1_2_certificate_batch.py	fix paths	a year ago
instantiate_v2_certificate_batch.py	human readable filenames, don't regenerate certs (#22)	11 months ago
jsonpath_helpers.py	V2 updates (#12)	2 years ago

Workflow depicting the actors in the system:

- Issuer
- Recipient
- Verifier

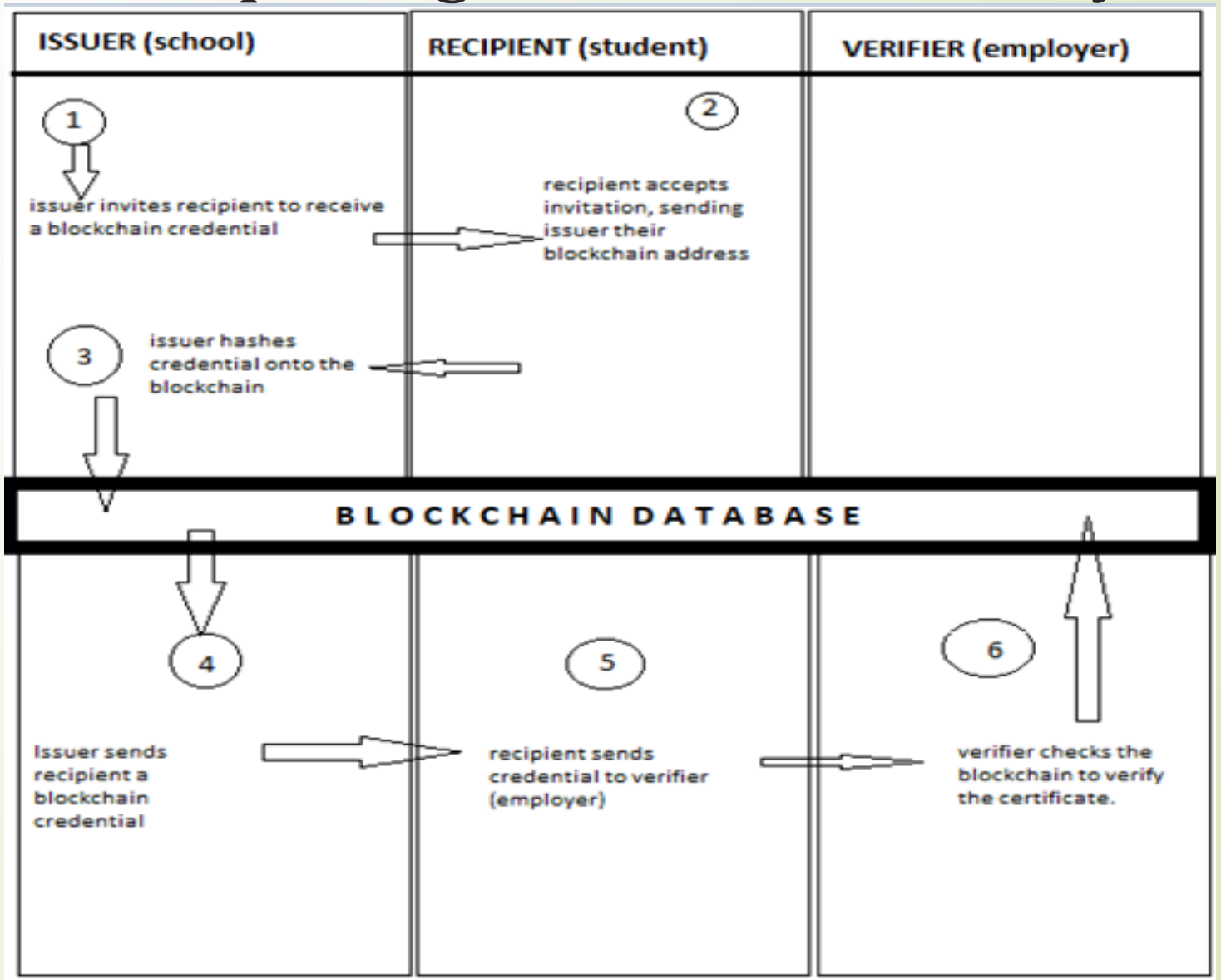
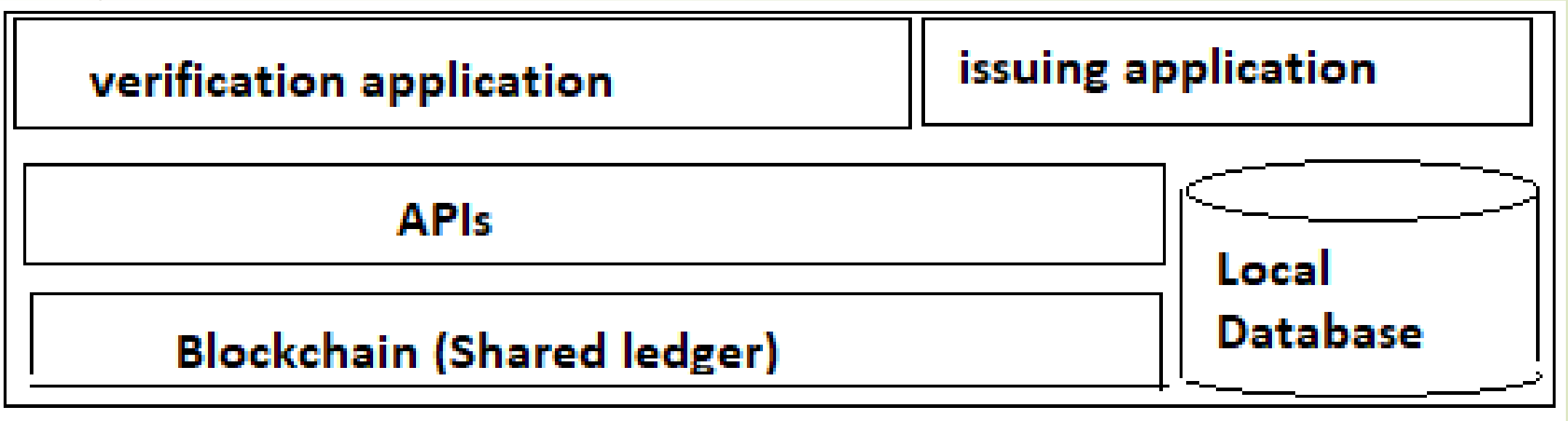


Figure 1 Workflow of the blockchain based credential prototype

System and database architecture

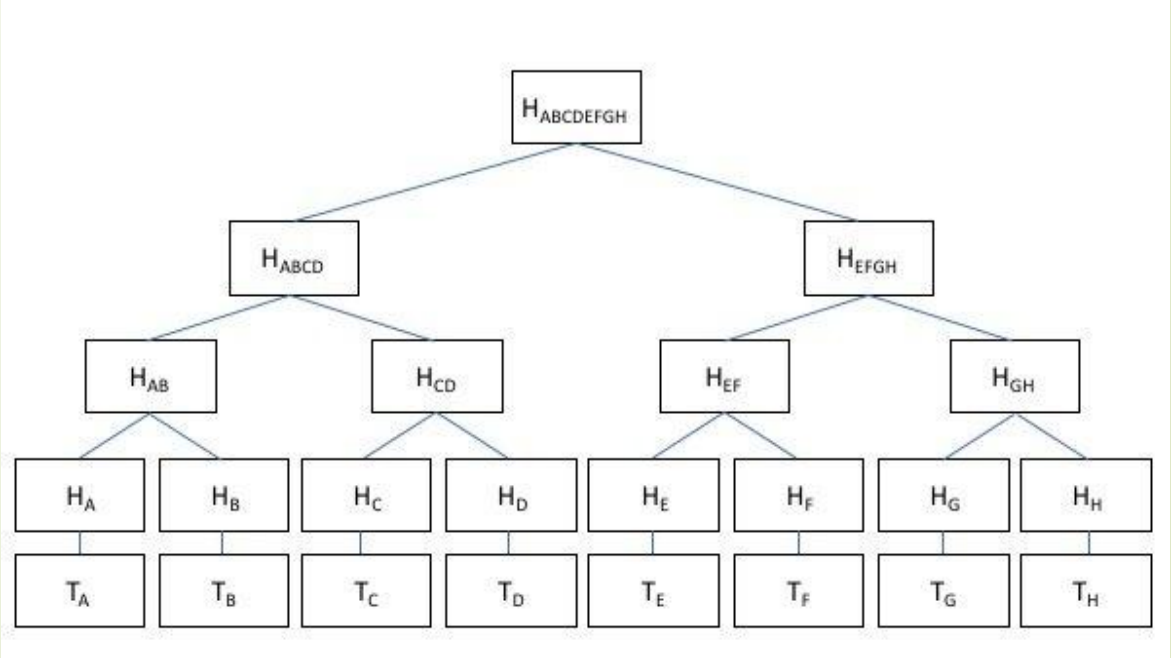


Creating the certificate

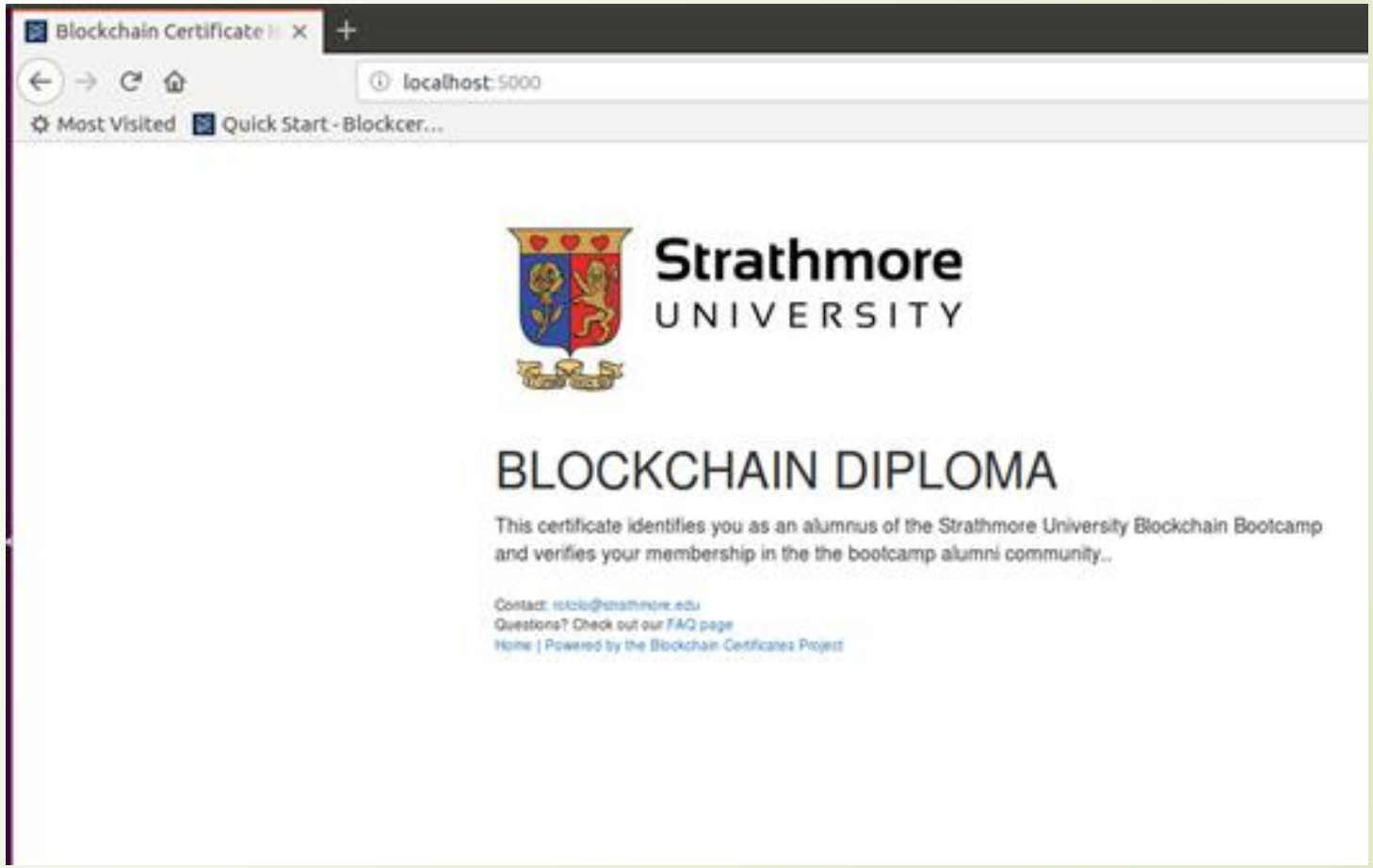
	A	B	C
1	name	pubkey	identity
2	Eularia Landroth	ecdsa-koblitz-pubkey:mtr98kany9G1XYNU74pRnfBQmaCg2FZLmc	eularia@landroth.org
3	Richard Otolo	ecdsa-koblitz-pubkey:mkwntSiQmc14H65YxwckLenxY3DsEpvFbe	rotolo@strathmore.edu
4	Eunice Maingi	ecdsa-koblitz-pubkey:maingiSiQmc14H65YxwckLenxY3DsEpvFb	emaingi@strathmore.edu

- Create a CSV file with the details of the certificates and their recipients - for a batch.
- Hash of hashes

  (Merkle tree)
- A note on privacy

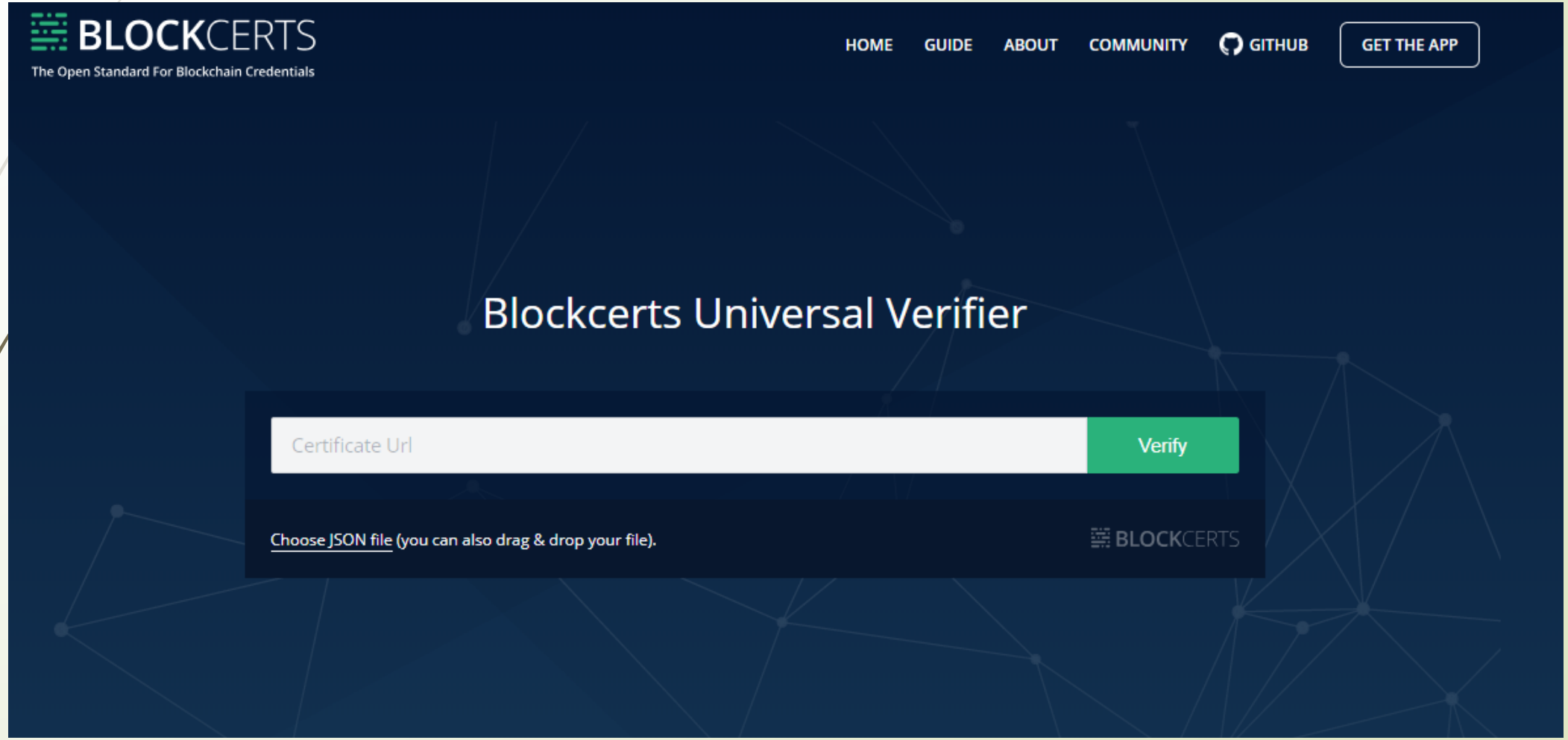


The results:


Blockchain Certificate
localhost:5000
Most Visited
Quick Start - Blockcer...
Strathmore
UNIVERSITY
BLOCKCHAIN DIPLOMA
This certificate identifies you as an alumnus of the Strathmore University Blockchain Bootcamp and verifies your membership in the the bootcamp alumni community..
Contact:
Questions? Check out our FAQ page
Home | Powered by the Blockchain Certificates Project

Verification of the certificate:

Sample verification portal



Legitimate certificate:

```
(venv) novice@blockchain:~/BLOCKCERTS/kenet_project/cert-verifier/cert_verifier$ python verifier.py cert.json
cert.json
Checking certificate has not been tampered with,passed
Checking certificate has not expired,passed
Checking not revoked by issuer,passed
Checking authenticity,passed
Validation,passed
```

Fake certificate:

```
(venv) novice@blockchain:~/BLOCKCERTS/kenet_project/cert-verifier/cert_verifier$ python verifier.py ca0f6165-0f8c-41fb-8
83a-35234a242e2e.json
ca0f6165-0f8c-41fb-883a-35234a242e2e.json
ERROR:root:Certificate has been modified
Traceback (most recent call last):
  File "/home/novice/BLOCKCERTS/kenet project/venv/local/lib/python2.7/site-packages/cert verifier/checks.py", line 111,
 in do execute
    detect unmapped fields=self.detect unmapped fields)
  File "/home/novice/BLOCKCERTS/kenet project/venv/local/lib/python2.7/site-packages/cert schema/jsonld helpers.py", lin
e 184, in normalize jsonld
    'There are some fields in the certificate that do not correspond to the expected schema. This has likely been tamper
ed with. Unmapped fields are: ' + error string)
BlockcertValidationError: There are some fields in the certificate that do not correspond to the expected schema. This h
as likely been tampered with. Unmapped fields are: <http://fallback.org/displayHtml> "<h1>Well done! Well done!</h1>
ERROR:root:Verification step VerificationGroup failed!
ERROR:root:Verification step VerificationGroup failed!
Checking certificate has not been tampered with,failed
Checking certificate has not expired,not started
Checking not revoked by issuer,not started
Validation,failed
```

Future project scaling

- Replicate to all the courses in the university
- Scaling to other institutions

Thank you

Questions, Comments, Suggestions