Educhain

Endorsements of Open Badges with blockchain technology

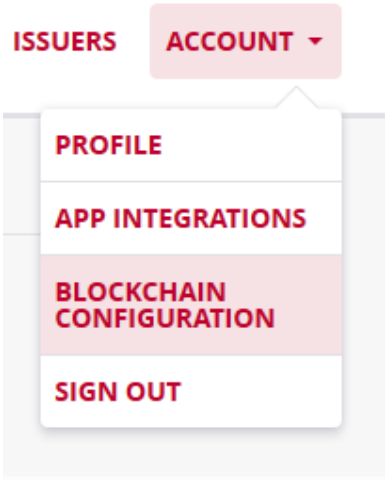
This short document marks the final deliverable from Coinversable B.V. to SURF for the proof of concept in which we have shown that blockchain can be used to store endorsements of Open Badges. We have extended Badgr (forked from Concentric Sky) with functionality to communicate with the open-source Validana.io blockchain.
Along with this document repositories with readme.md and source files have been delivered to SURF. All code written by Coinversable for this PoC is available under the AGPLv3 license as requested by SURF. On a later notice Coinversable will also commit / merge the endorsement code to a Git repository provided by SURF.

DEMO and walkthrough

As part of the final delivery Coinversable has installed a fully working version of Educhain on a server hosted by SURF. With the following credentials you can login as admin on the demo website.
https://dev.educhain.nl.

A full presentation of how the educhain environment works has been presented to the Open Badges core team of SURF. Below is a short introduction for newcomers. Please note that all provided public and private keys are entirely made up, and have no value of their own. They were created for demo purposes and should be replaced with actual private keys in any real world application.

Blockchain Configuration



The blockchain configuration page is new in Badgr. Here you can provide your private key. The private key you provide will never leave your browser. Transaction signatures are created within your own web browser!
Provide your private key in the field. You can use on of the following to experiment with:

- SURF (can add / withdraw educational institutes)
- Delft University (fictive, can add / withdraw educational entities)
- Delft EWI Faculty (fictive, can endorse badges / badgeclasses)

If you want to add more educational institutes to the demo, please login with the private key of SURF and add them by adding the corresponding public address. We follow the Bitcoin addressing protocol. You can use https://demo.coinversable.com to securely generate a private key in your web browser. After ‘creating an account’ you will be provided with the private and public key, which you can use. Repeat the process if you want more private/public key pairs or use any other bitcoin compatible wallet (WIF) private key.

Storing BadgeClass on the blockchain

If you are logged in with the private key (WIF) of an educational entity which is not withdrawn, you can store the badge class on the blockchain.

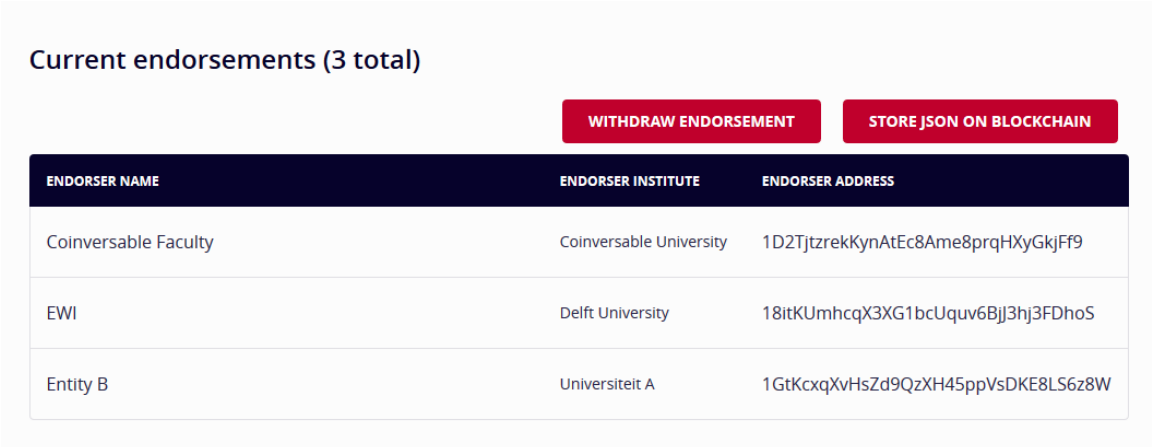
This process will happen automatically (and you will receive a notification in Badgr about it) if you have provided the WIF of an educational institute in the blockchain configuration page before you press ‘save’ in badgr upon creation of a new badge class.

Otherwise navigate to your existing badge classes and press ‘Store JSON on blockchain’. You can press this button multiple times, as the badge class JSON can be edited afterwards in badgr. Each press will submit the current version to the blockchain.

The first entity to submit the JSON to the blockchain becomes the ‘owner’ of that badge class on the blockchain, meaning that other entities which press ‘store JSON on blockchain’ will get a message warning them that the JSON cannot be stored.

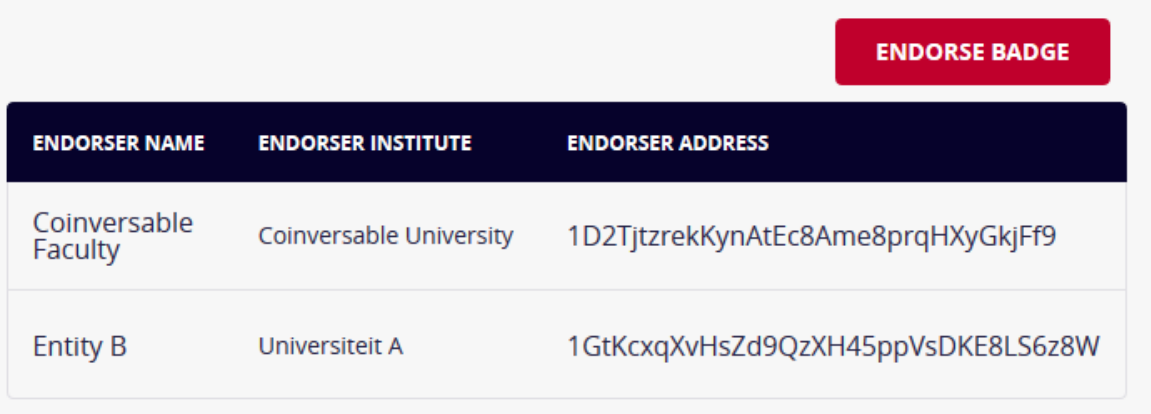
Endorsing a badge class

In Badgr navigate to a specific badge class. On the bottom you will find the current endorsements. Click the red button to endorse a badge class. Click it again to withdraw your endorsement.

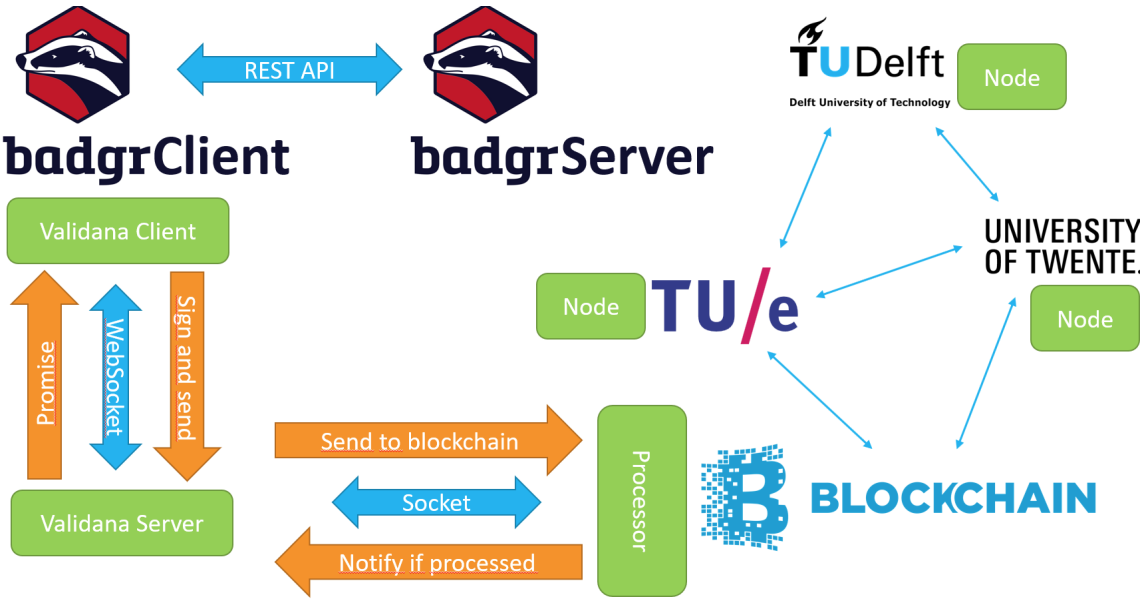


Endorsing a badge

A specific badge can also be endorsed. Please navigate to the specific badge and press the red button to endorse. An endorsement of a badge can not be withdrawn as per specification.



Architecture



This picture provides a schematic overview of how educhain works

The badgr-client and badgr-server packages were forked from concentric sky. The badgr-server was left untouched within the educhain development. This means that all transactional logic with the blockchain is done from within the client, from within the web browser of the end user. This provides us with two mayor benefits. Firstly we are instantaneous compatible with all existing badgr-server installations. Secondly, private keys used for signing transactions will never leave the web browser of the client.

The Validana-client is implemented in Badgr, along with all custom page and handling logic. Everything is put in the ‘endorsement-api’ folder in the badgr-ui repository. The client signs transactions and sends them to the Validana-server. The server in turn forwards it to the blockchain processor, and keeps checking if the transaction is processed. If so it notifies the client via a Promise resolve.

The Validana-processor can best be compared to the ‘single’ miner in the blockchain. The processor manages which smart contracts live on the blockchain and allows or rejects incoming transactions accordingly. The Validana-nodes receive a full copy of the chain, including the smart contracts, and check the work of the processor independently. A block is created every 5 seconds. It is up to SURF to decide if everyone can become a node, or just a select view. For this demo two docker containers are created on the server of SURF mimicking the nodes for the University of Twente and TU Delft.

Nodes can always check if the processor did its work well. If a node is for instance a back-end application of the university, submitting a transaction to the blockchain itself, it should be able to verify that the transaction succeeded and expects the processor to process the transaction without failure. If the processor fails to do so this becomes immediately known to the nodes. The processor itself can also not modify transactions because they are signed by the client, and modification will break the signature.

Recommendations and enhancements

We have shown that blockchain can successfully be used with Open Badges in the educhain PoC. However before we can go into production or even into a real-life pilot with universities we suggest a few enhancements. This list is not exhaustive and possibly there are more user features.

Validation service
The validation service can be extended to also show which endorsements there are on the blockchain.

Endorsements with data
The blockchain currently only stores which entities endorsed which ID’s of badges / badge classes. Better would be to follow the specification on endorsements and also allow extra information such as a descriptive field. We could use hashing to prevent privacy sensitive information from landing on the blockchain, whilst maintaining the ability to prove that the badge class and metadata were in deed unmodified.

Timeline
When an entity endorses a badge class, and later withdraws that badge class, no ‘history’ is shown in the Badgr environment. Therefore if a student obtained a badge from an endorsed badge class, and the endorsement is withdrawn, one can not easily verify that the badge class was indeed endorsed at the time of acquisition. It would be convenient for the student to prove that the badge class was endorsed at that point in time via the user interface. Now you have to become a node and use the blockchain ledger to prove it.

History of names
SURF is able to change the name of an educational institute by submitting another request to add an institute for the same public address. This only changes the human friendly name on the blockchain. However we can expect end-users to not check the public addresses of the institutes themselves. Therefore it would be wise to show which previous names were used for a institute. The same holds for educational entities, which change name often.

Roles / permissions via blockchain
In the current PoC there already is a chain of trust on the blockchain, namely SURF -> Institute -> Entity. We could extend this functionality to also store on the blockchain which participants can be an issuer in badgr. For instance we could allow everyone to be able to create an issure, but display a ‘green lock’ icon next to the issuers which are inside the chain of trust. Or we simple disable the functionality in Badgr entirely when the user does not have the right permission on the blockchain. This functionality would be similar to the ‘endorse’ and ‘withdraw’ buttons currently found in the UI.

Signed badges
With blockchain everything is signed. We can use this to also sign badges and badge classes. If implemented well it would make it possible for organisations to check the validity of a badge without having to use the validation service. This way badges are verifiable even if the university website / validation service is offline.

Identity and Key management (badges without email addresses)
The current PoC does not help end-users with storing their Private Key. We could extend the PoC by allowing SURF or the university to manage the private keys. This functionality is optional. If the user wants to keep his own private key we can ‘transfer’ (new smart contract) all badges of the old user to the new user, via a three way signature (old account, university, new account). We should then provide the public address of the user in the badge recipient field, and the user can transfer badges after he left the university to a new account.

Store everything on the blockchain
We could create a version of the badgr-ui where all information is stored on the blockchain. Therefore the badgr-server package would become entirely obsolete. We should then rethink the definition of an account within badgr-ui, and create a possibility for end-users to allow a third-party (surf, university, DUO) to store their private key if preferred. Also everything would be stored on the blockchain. We should identify which information is private and which can be made public. We can also keep a simple local database and place hash values of privacy sensitive information on the blockchain. Upon validation the hash is calculated and checked with the blockchain.