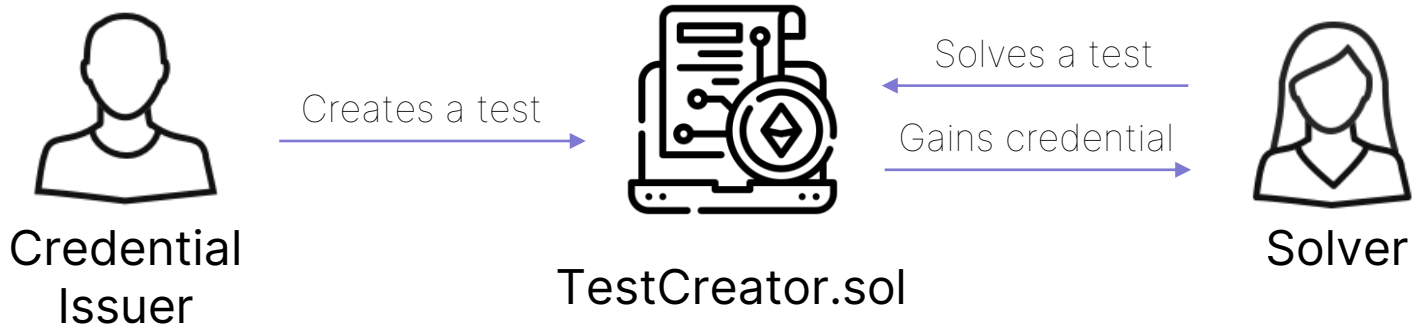
bq

Block Qualified

What is Block Qualified?

An open, decentralized and fully on-chain education platform, where **you** create your learning experience



Why is it useful?

When you solve a test, you prove your knowledge of the solution, *but without revealing it*

Privacy is preserved, cheating becomes unfeasible, thanks to zk-proofs

Who can use it?

As a permissionless protocol, anyone can create and solve tests

You can prove you have the necessary knowledge/credentials, without revealing anything else about yourself

By moving exams on-chain, we fully decentralize credentials

How can it be used?

Block Qualified supports three kinds of tests:

- Multiple choice, where answers are part of a given set
- Open answer, where answers can be any value
- Mixed tests, containing both

Where can it be used?

Potential use cases:

- Web3 learning sites

- Decentralized contests

- Privacy-preserving job platform

How to use Block Qualified

Block Qualified is fully open-source, and distributed under an MIT license

You can choose to deploy your own instance, or work on top of another

The *bqSDK* allows you to easily interact with any deployed contracts, abstracting away the zk-proof generation

Example: solving a test

```
// Defining a new bqTest object in solve mode
const solveModeTest = await bqTest.solveMode(
    '1',
    ethersProvider,  // valid ethers.providers.JsonRpcProvider
    '0x403E6BBCB3Ddbe3487c09E8827e5dEf058FE6db4'
)

// Generating a proof for our solution
const solutionProof = await solveModeTest.generateSolutionProof({
    recipient = '0x1B02c971d0322DB82170FB7950D45D26Efc5853B',
    openAnswers: new Array(64).fill("deenz"),
    multipleChoiceAnswers: Array.from({length: 64}, (_, i) => 1)
})

// Sending a solving transaction and gaining a credential
await solveModeTest.sendSolutionTransaction(
    signer,  // valid ethers.Signer
    solutionProof
)
```

Additional features

The recipient of the credential is embedded inside the proof, this means tx relayers can be set up for users to avoid gas fees

Both tests and credentials work as non-transferable ERC-721 tokens: they are bound to the address that received them

The credential issuer can set time/credential limits, and also invalidate credentials they own

What’s next?

An integration with the learning platform useWeb3.xyz, by @wslyvh from the Ethereum Foundation, is on the works

As this will be the first live implementation, we’ll look into gathering feedback to improve the scope of Block Qualified

We expect to study a potential integration of Semaphore (part of the EF’s Privacy & Scaling Explorations) into the protocol

Links and resources

Github repository: github.com/0xdeenz/bq-core

Documentation: deenz.gitbook.io/bq-core

Previous implementation: blockqualified.netlify.app

Reach out on twitter: @0xdeenz