Blockchains in the new era of participatory media experience

HORIZON 2020

762091 – BLOOMEN - H2020-ICT-2016-2
ICT-19-2017 Media and content convergence

D3.5 Anonymous Personalization Services - 2nd cycle

Version:	1.0
Date:	31/05/2019
Authors:	ICCS of NTUA
Type	Demonstrator
Dissemination level	Public

Co-funded by the Horizon 2020 programme
of the European Union

Table of Contents

1 Introduction and Description of the Prototype

This deliverable is the second iteration outcome of the Task 3.2 "Anonymous personalization over open, trustless media platforms" and it follows the results and description of the first iteration deliverable D3.4. It is the second out of three iterations and as a software prototype it contains the Bloomen related services for the privacy preserving techniques, personalization and identity management over blockchain enabled media delivery platforms. As described in the Description of Action, this task focuses on the research issues and the development of the identity management framework, including the anonymity and personalization functionalities.

Given the decentralized nature of the P2P architecture proposed by the project, the task focuses on how to distribute and decentralize the identity management functionality that is present in current centralized media sharing frameworks, including the options for users and other third parties to set this functionality themselves as opposed to depending on central systems administrators. Second, the task addresses how to distribute the identity management functionality across the different levels of the P2P architecture with a blockchain level, a middleware layer and an application level. Finally, the task investigates how, with the complexity induced by both the decentralization and the multi-layered architecture, identity management can be made to be universal, i.e., how it can extend across very dissimilar use cases inside the Bloomen project. The final outcome of the development work will be a set of software modules that provide the desired functionality of identity management and control over P2P networks.

Moreover, the Task 3.2 will research and implement all necessary data privacy enhancing mechanisms that are necessary for the Bloomen framework. This mechanism will reflect the whole lifecycle of the data privacy management including the collection, storage, sharing and deleting of the data in the Bloomen P2P network. In particular the mechanism will ensure anonymization of data when necessary, separation of data (what needs to be pushed on blockchains and what to remain on user device) and additional confidentiality measures (such as encryption, separation of data from context etc.). Data access control (authentication / authorization) and privacy policies will be implemented as well.

2 Solution Concept in the second iteration of prototype

2.1 Anonymous Personalization and related blockchain features

In the first iteration of the anonymous personalization component for the core blockchain implementation, Hyperledger was used. Although Hyperledger was a stable and well-established solution, with features that enabled the privacy and security features needed for the project, in the second iteration of the component a new blockchain solution was incorporated. In the new version of the Anonymous personalization, an Ethereum based blockchain was integrated, namely Quorum, which is an enterprise- and consortium- focused Ethereum blockchain. Since Quorum is based on Ethereum many of the advantages of Ethereum are inherited to Quorum. Some of the main key features of Quorum related to privacy and identity management, that needed in the frame of the project are listed below:

- Usability: Hyperledger is ideal for Business to Business transactions since the participation is very strict with only permissioned actions. Ethereum is a more generic purposed blockchain that supports both public and private platforms. This feature makes it more ideal to Business to Consumer transactions which is one of the key features needed for the use cases of the project (i.e WebTV use case as described in D2.1, or the Photo use case which implies also Know Your Customer - KYC features).
- Tokens: Hyperledger does not have a build in cryptocurrency. Although this might be a good feature because designers have the freedom and flexibility to adopt a variety of cryptocurrencies, having a well-established and popular cryptocurrency makes the job of creating applications and making more production ready application easier. Tokens, as a vital financial of value exchange over blockchains plays a significant role when considering identity management and trust in business scenarios as in the case of Bloomen. When there is exchange of value involved between a business and a customer it promotes honest, reputable and trusted businesses. So, in this instance, customers can be assured that they maintain full control over their personal data and preferences.
- Transactions: Ledgers in Hyperledger are not public a feature that seems very appropriate for the component. Ethereum gives access to the ledger to all the participants, a feature which the quorum solution enhances with security and privacy mechanisms in order to achieve anonymity and privacy for specific parties involved, making it more flexible than having complete privacy even when it is not needed.

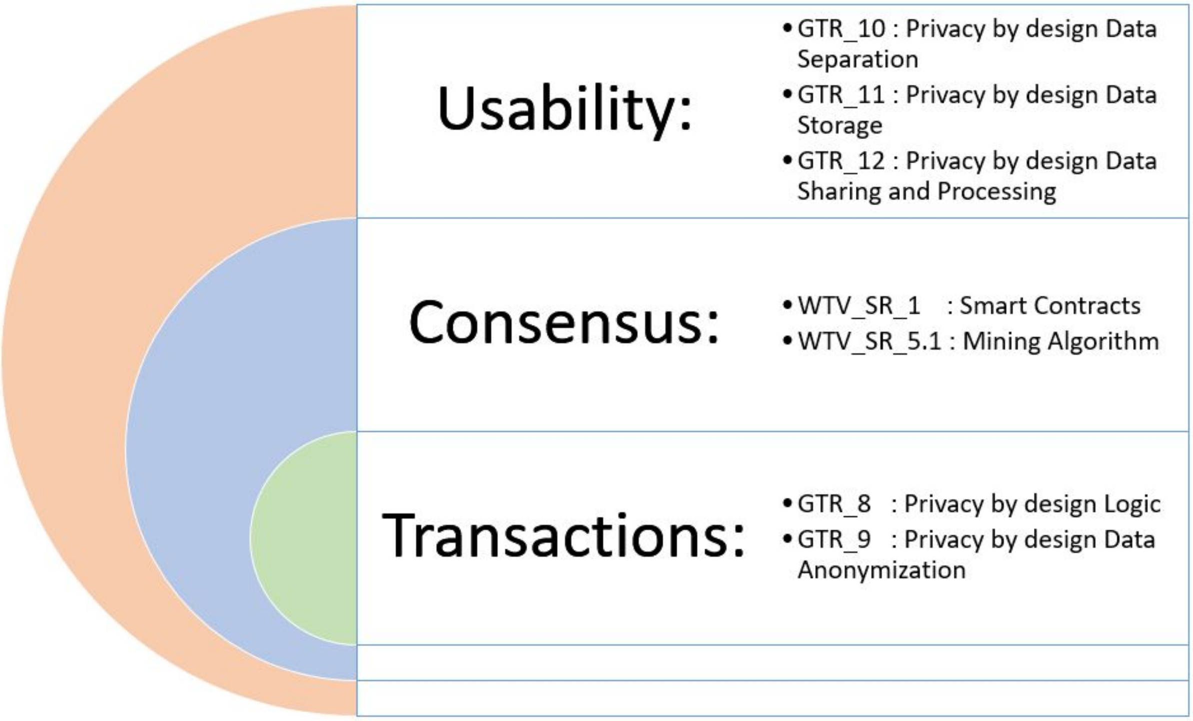

Usability:
• GTR_10 : Privacy by design Data Separation
• GTR_11 : Privacy by design Data Storage
• GTR_12 : Privacy by design Data Sharing and Processing
Consensus:
• WTV_SR_1 : Smart Contracts
• WTV_SR_5.1 : Mining Algorithm
Transactions:
• GTR_8 : Privacy by design Logic
• GTR_9 : Privacy by design Data Anonymization

Figure 1. *Ethereum features and requirement mapping*

It is important to understand that the Ethereum features described above aim to better address specific key requirements of the project. As depicted in the image above (Figure 1) there are specific requirements that shape the general architecture of the project these requirements were taken under strong consideration for the shift from Hyperledger to Ethereum based Blockchain.

Ethereum is among the foremost iterations of the bitcoin blockchain network. Being a public blockchain makes Ethereum not an ideal solution for enterprise purposes. Therefore, in the context of the project, a solution that incorporated all the good features of the Ethereum and enhances them with privacy and security features is needed, hence the Quorum.

One of the features of Quorum that are of great value for the component is the network and peer to peer permission management. This feature enables only the validated and authorized users to have access and be a part of the network. Also, Quorum provides an enhanced transaction and smart contract privacy features. Permission-based nature of the Quorum enables the constitution of private and public transaction getting the best of both worlds, open transactions are analogous to Ethereum but when it comes to the private transaction then it is confidential, and the data is not exposed to the public. Quorum adds privacy functions that allow for private transactions that are only visible to the transacting parties, while the other parties in the network would only see a hash. This is further explained in section 3. Finally, Quorum is considered to be very fast and being able to process even thousands of transactions per second[1], due to its efficient consensus mechanism which belongs to the family of Byzantine Fault Tolerance (BFT) mechanisms[2].

An important comparison that was taken under consideration was between the security features of Quorum and Hyperledger. In the previous paragraph, the advantages of using an Ethereum based solution were clarified. Now the security advantages of Quorum versus Hyperledger were considered for the decision.

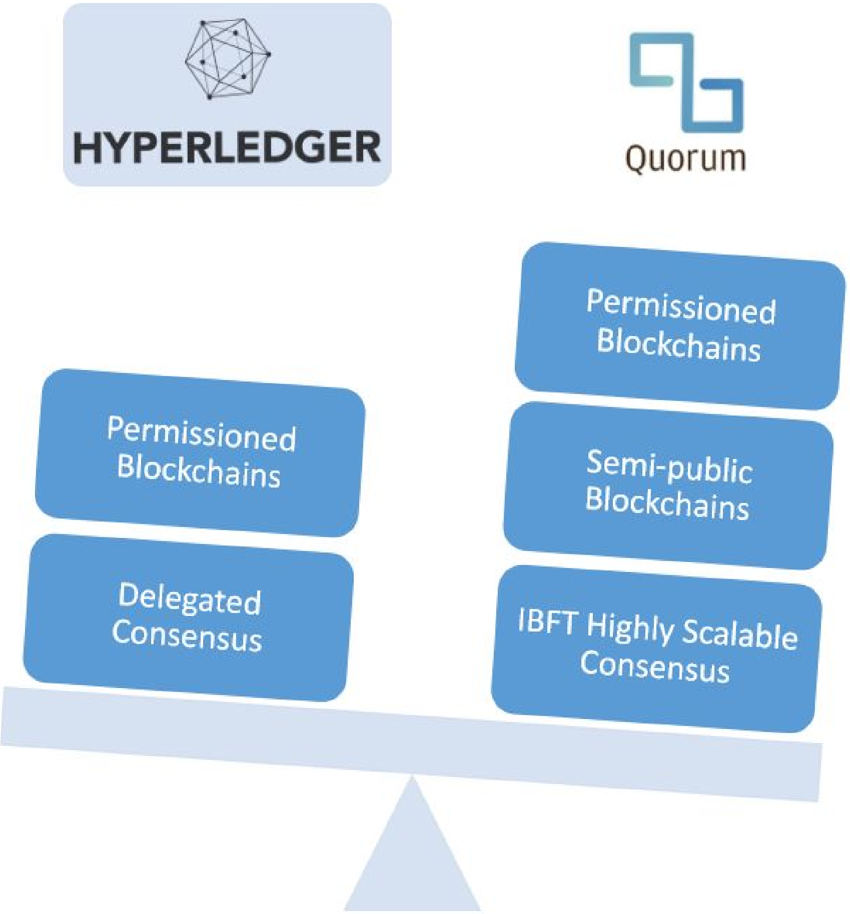


Figure 2. *Features that weighted in favor of Quorum in the second iteration of prototype*

As depicted in the image above Quorum has more flexible blockchain solution with privacy features that can be tailored to the needs of the system that implements the Blockchain technologies. Also, the Consensus mechanism of Quorum is much faster

[1] Baliga, A., Subhod, I., Kamat, P., & Chatterjee, S. (2018). Performance Evaluation of the Quorum Blockchain Platform. *CoRR, abs/1809.03421*. https://arxiv.org/pdf/1809.03421.pdf

[2] Vukolić M. (2016) The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In: Camenisch J., Kesdoğan D. (eds) Open Problems in Network Security. iNetSec 2015. Lecture Notes in Computer Science, vol 9591. Springer, Cham

and more scalable due to Proof of Work algorithms implemented. The security and privacy features of Hyperledger were fully adapted in the project but one key factor besides the technical comparison was taken under strong consideration for the change to Quorum. Quorum is implemented by Alastria[3] which is a non-profit association that promoted the digital economy. In the context of the project, Worldline which is involved in the Alastria blockchain gave access and permission to use this blockchain. This is a huge opportunity to implement the solution of the Bloomen in an Industrial Blockchain with many security and privacy features. Since Bloomen aims at a more production-ready solutions having a production blockchain as a component will aim for a final Technology Readiness Level (TRL) to the scale of TRL 6-7 of the Anonymous personalization component by the end of the project.

2.2 Off-chain storage and access control for anonymization and privacy

The current demonstrator focuses on both the WebTV use case and the Photo use case and particularly on the scenario where streaming platforms and media content consumers are involved in a core B2C relationship. According to the latter, the underlying blockchain platform and smart contracts will allow trustworthy communications, which will execute transactions and deliver content under a cryptocurrency or utility-token based financial model. However, the proposed approach can be applied to any kind of media use case where an end-user is willing to purchase content that is made available by the provider. To this end, the goal is to design a blockchain-based decentralized content marketplace, which enables trustless disintermediation between providers and consumers. Using a cryptocurrency for payments, a consumer can buy videos on the marketplace without involving a marketplace intermediary. In the context of T3.2, this section refers to the research and development of data privacy-enhancing mechanisms along with data access control and privacy policies that are necessary for the Bloomen framework. Moreover, it deals with the separation of data, meaning to identify what needs to be pushed on blockchain and what to remain off-chain, a decision that is always critical when designing blockchain platforms.

Having in mind the scenario described above, we assume that a content provider creates a video and intends to offer it via the Bloomen platform, relying on smart contracts to encode the ownership and clearly define the business model, through which all the contributors receive their reward. However, the video and any kind of media content in general, is too expensive to store it on-chain due to its size. This is where the Content-Addressable Storage (CAS) technology kicks in. The metadata that describe the video and/or the data itself, are placed inside a CAS system, which stores files by their hashes. The latter are also stored in the smart contract, pointing to the externally stored video (see Figure 3). Consumers are then able to retrieve the hash from the contract and use it to query the storage system. In order to validate the correctness, the client can simply hash the result.

[3] https://alastria.io/en/

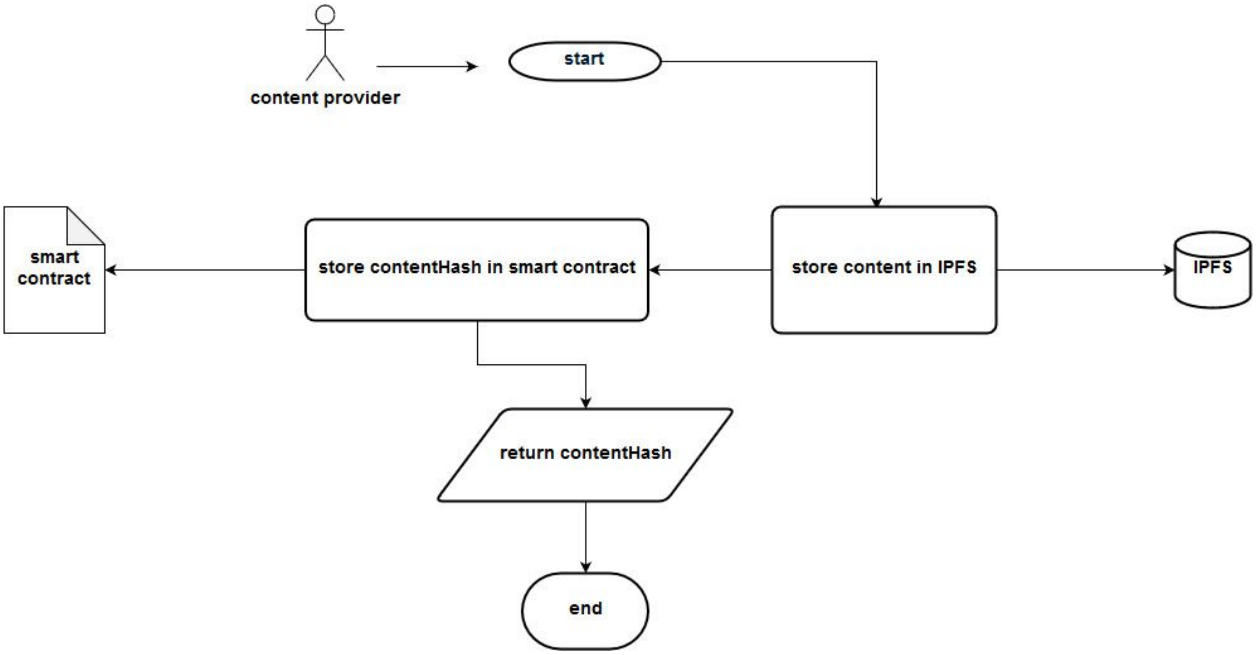


Figure 3. *Storing content by its hash*

The CAS system is implemented using the InterPlanetary File System (IPFS) [4], a peer-to-peer version controlled file system that combines a distributed Hash table, an incentivized block exchange and a self-certifying namespace. Although, there have been developed many P2P file-sharing systems all over the world, they cannot cover all the use cases when compared with HTTP. Actually, this is one of the most successful distributed file systems, so that many programs have adopted the browser/server architecture instead of client/server one due to the progressive development of browsers and enormous impact of HTTP. However, IPFS aims to replace it seeking to connect all computing devices within the same system of files, in order to take advantage of dozens of novel file distribution techniques invented recently.

The involvement of IPFS allows the trustless outsourcing of data to an off-chain storage system, since even a slight modification in the data would immediately alter its address and thus invalidate its references. By adopting this approach, an application’s storage cost can be greatly reduced, and content that originally could not be stored on-chain, can now be referenced without introducing trust. Furthermore, as the data retrieval is performed on the consumer’s side from an external storage system, privacy features might/should be implemented by adding access control to the whole architecture. In contradiction to traditional access control mechanisms that are based on centralized databases containing user identities and their access rights, various implementations are nowadays using blockchain as an access control manager for distributed systems.

[4] https://ipfs.io/

Given the adoption of Quorum, an Ethereum –based blockchain, the idea is to exploit the full potential that the smart contacts can offer. Indeed, they allow code to be executed and the results to be stored with the same assurances as financial transactions can be made on the blockchain. Ethereum is stateful, so that the code executed on the blockchain is able to change the state of the system. This allows the implementation of a smart contract based access control system where, in addition, removal and changes in permissions are being recorded on the blockchain and thus forming part of Ethereum's (and thus Quorum's) state. The benefits of using a smart contract for access control are the same as the benefits for financial transactions. The decentralization ensures that no entity has control over the code execution, thus making it trusted, with any resulting changes being agreed upon by consensus. The code cannot be tampered with or changed, given that Ethereum/Quorum blockchain is appended only due to consensus and proof of work. The following flow charts depict functions that are included in the access control smart contract. Through those functions, a content provider can store his/her content(storeContent) in the access control list and then make it available (allowAccess) to a potential consumer. Finally, the latter is able to retrieve the content using its hash (getContentLink).

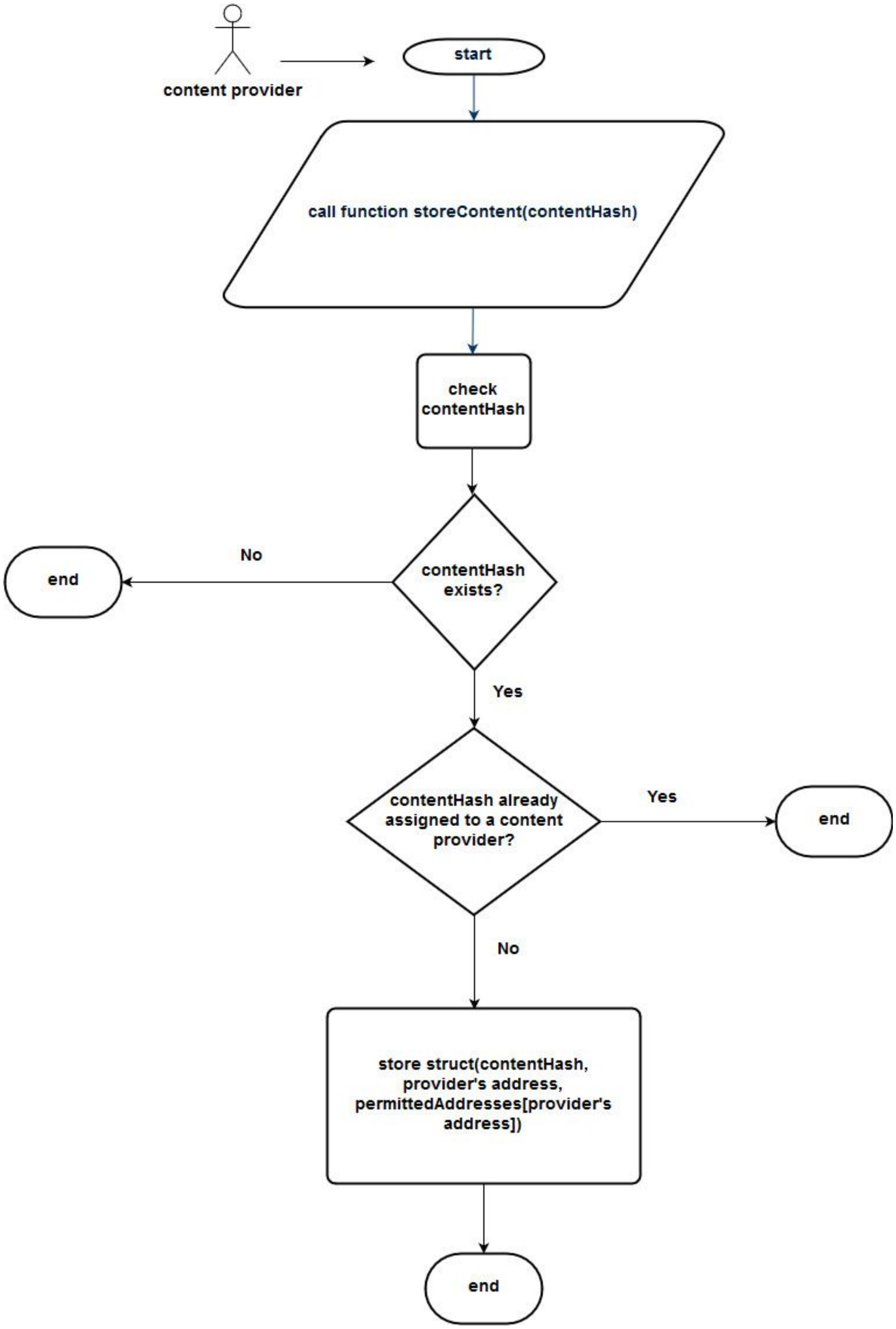

content provider
start
call function storeContent(contentHash)
check contentHash
contentHash exists?
No
end
Yes
contentHash already assigned to a content provider?
Yes
end
No
store struct(contentHash, provider's address, permittedAddresses[provider's address])
end

Figure 4.*Content upload flow by the content provider - preserving privacy and access rights (workflow for the initial upload and contentHash assignment).*

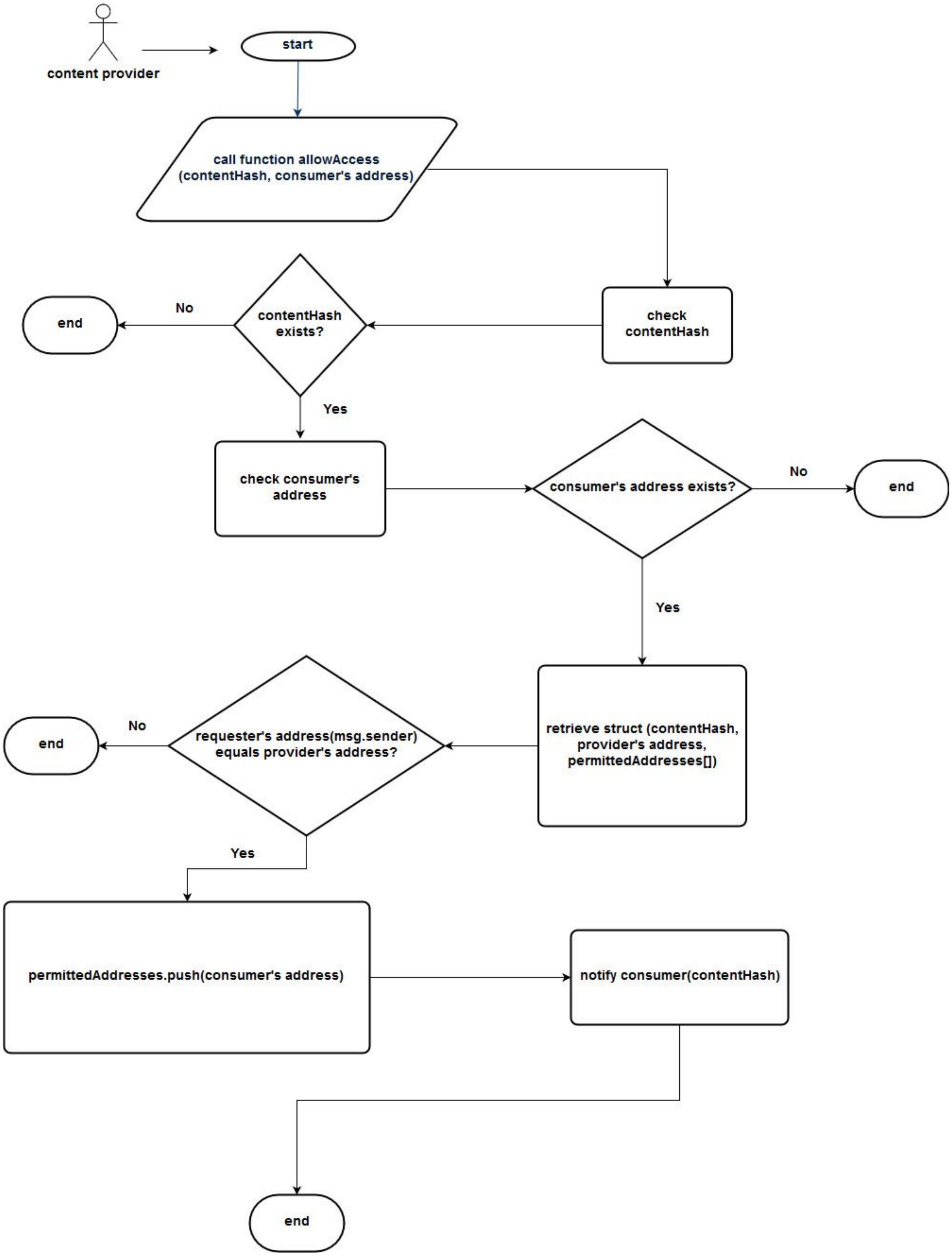

content provider
start
call function allowAccess (contentHash, consumer's address)
check contentHash
contentHash exists?
No
end
Yes
check consumer's address
consumer's address exists?
No
end
Yes
retrieve struct (contentHash, provider's address, permittedAddresses[])
requester's address(msg.sender) equals provider's address?
No
end
Yes
permittedAddresses.push(consumer's address)
notify consumer(contentHash)
end

Figure 5. *Allowing access and permitted addresses for viewing content*

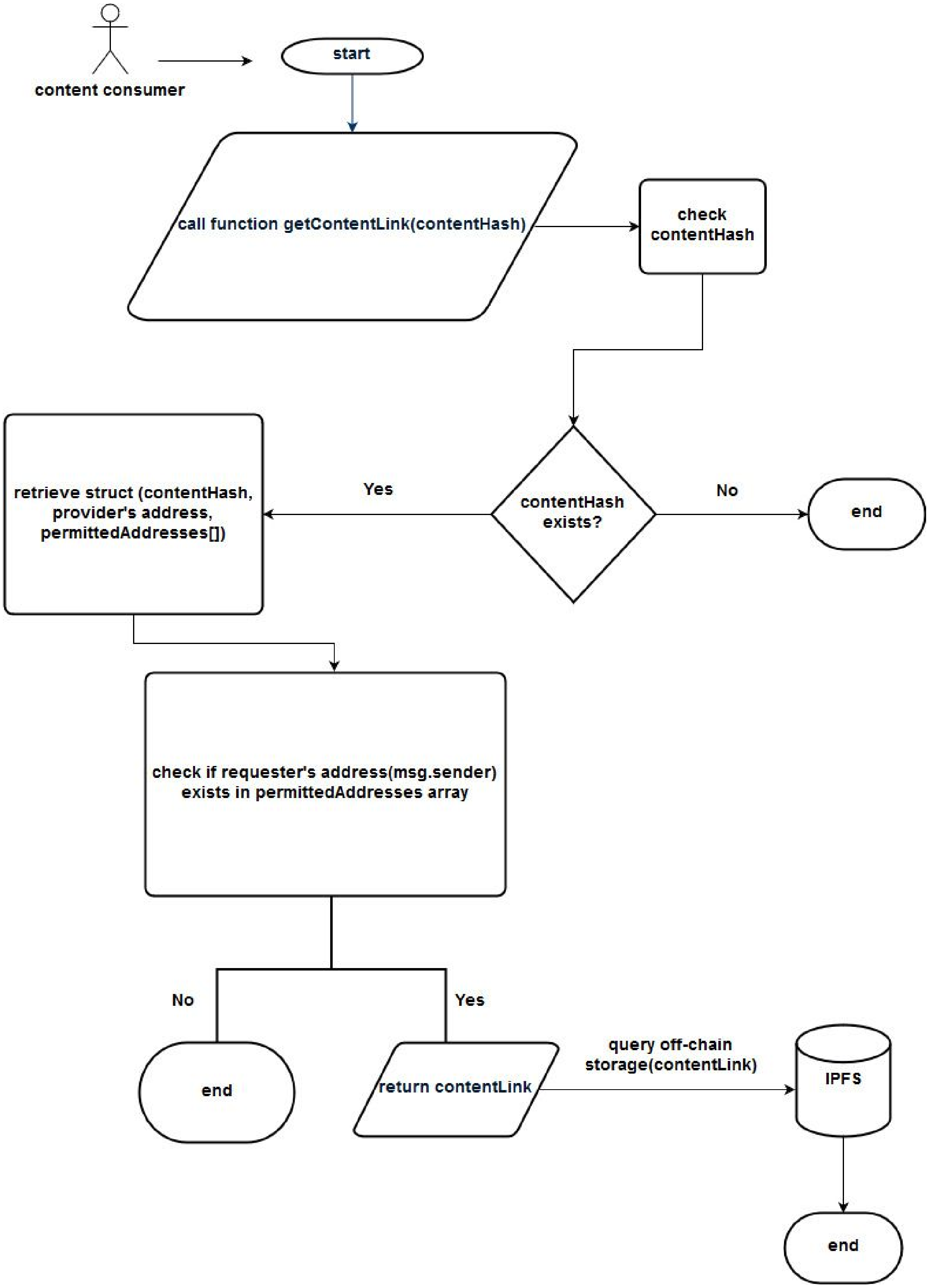
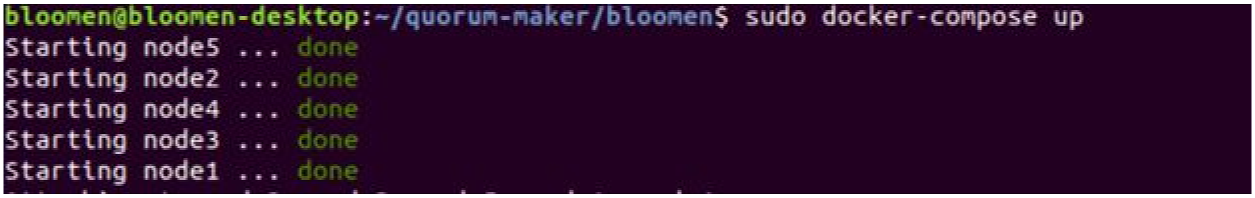

content consumer
start
call function getContentLink(contentHash)
check contentHash
contentHash exists?
Yes
No
end
retrieve struct (contentHash, provider's address, permittedAddresses[])
check if requester's address(msg.sender) exists in permittedAddresses array
No
Yes
end
return contentLink
query off-chain storage(contentLink)
IPFS
end

Figure 6. *Retrieving content flow for the content viewer*

3 Quorum private transactions and contracts

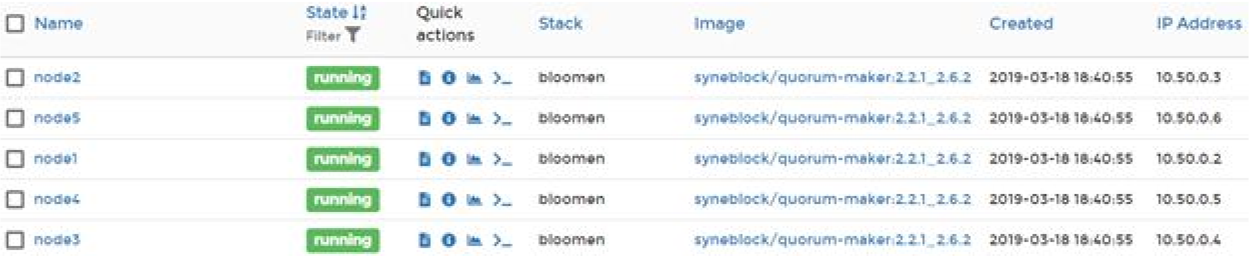
As described in the previous chapter(2.1) one of the main and most important features of the Quorum is the private transaction mechanism. Transaction privacy is achieved by using the Ethereum Transaction Model and enhancing in with new parameters that specify the nodes in which the transactions should be published. The Constellation layer of Quorum that contains the transaction Manager and Enclave module is responsible for the private transaction handling. All the public transactions follow the already established p2p Ethereum network flow.

A five node Quorum network was established in order to implement and demonstrate the private transaction features of quorum. More specifically the quorum maker which is an open Git project[5] was implemented that allows us to set up Dockerized quorum nodes.



```
bloomen@bloomen-desktop:~/quorum-maker/bloomen$ sudo docker-compose up
Starting node5 ... done
Starting node2 ... done
Starting node4 ... done
Starting node3 ... done
Starting node1 ... done
```

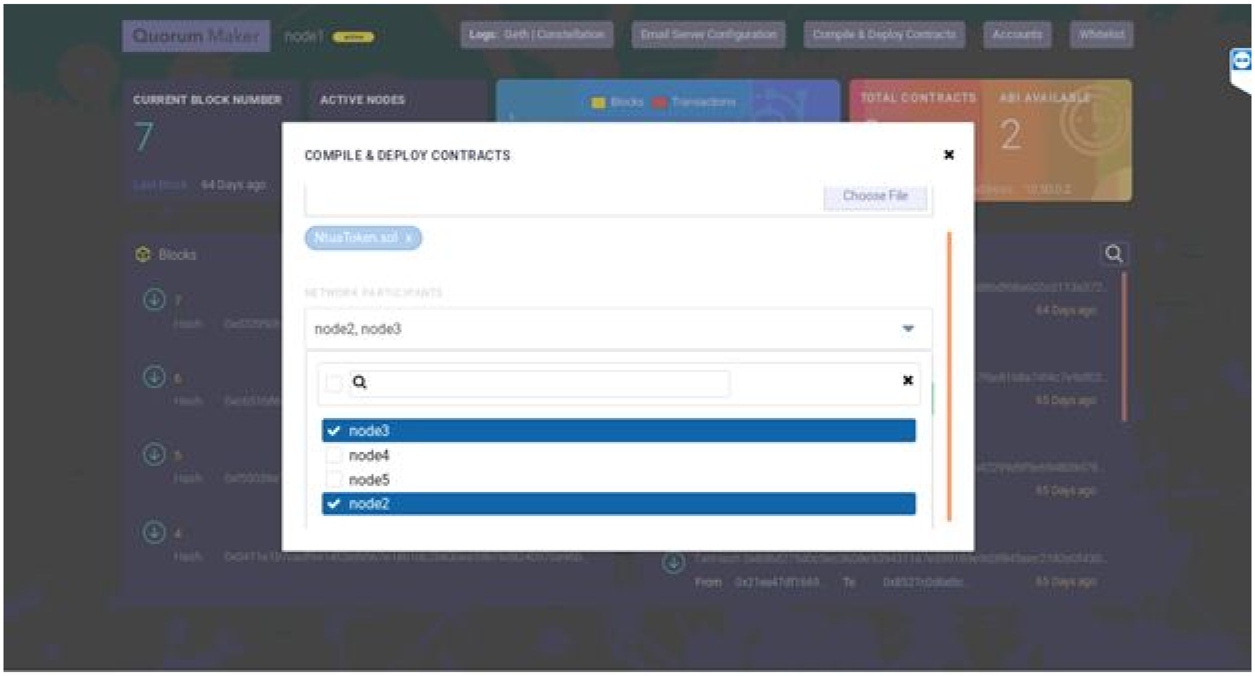
After starting up all the nodes in the Quorum network using the Docker compose command we can see that all the Dockerized nodes are up and running in different internal IP addresses in order to simulate a real network.



□ Name	State ↓↑ Filter	Quick actions	Stack	Image	Created	IP Address
□ node2	running		bloomen	syneblock/quorum-maker:2.2.1_2.6.2	2019-03-18 18:40:55	10.50.0.3
□ node5	running		bloomen	syneblock/quorum-maker:2.2.1_2.6.2	2019-03-18 18:40:55	10.50.0.6
□ node1	running		bloomen	syneblock/quorum-maker:2.2.1_2.6.2	2019-03-18 18:40:55	10.50.0.2
□ node4	running		bloomen	syneblock/quorum-maker:2.2.1_2.6.2	2019-03-18 18:40:55	10.50.0.5
□ node3	running		bloomen	syneblock/quorum-maker:2.2.1_2.6.2	2019-03-18 18:40:55	10.50.0.4

Using the UI provided by the Quorum maker we can now Compile and Deploy the NtuaToken. Using any web browser, we access the admin dashboard of the first node that is running (IP of the node:22004/dashboard).

[5] https://github.com/synechron-finlabs/quorum-maker/wiki



COMPILE & DEPLOY CONTRACTS

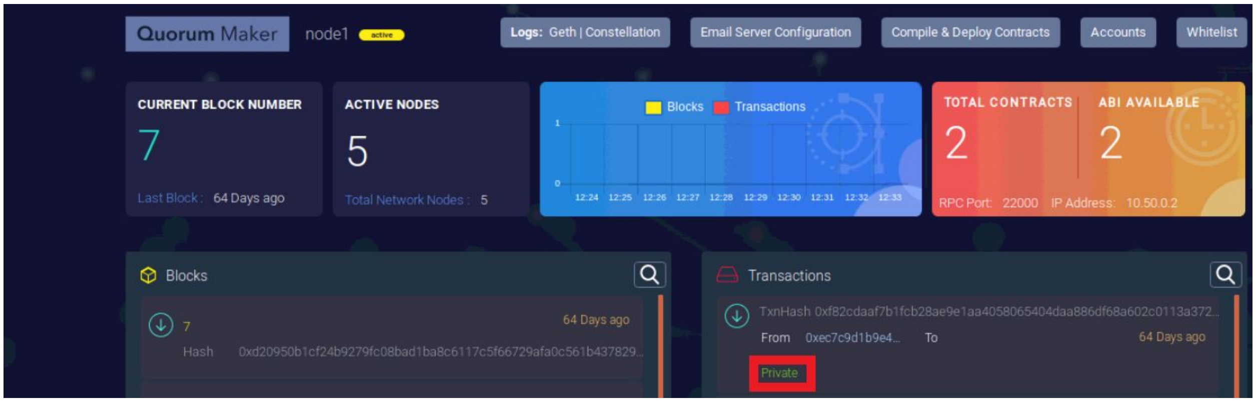
NtuaToken.sol x

NETWORK PARTICIPANTS

node2, node3

- ☑ node3
- □ node4
- □ node5
- ☑ node2

We can see that we can deploy a contract that will be accessible by specific nodes that we dictate. In the image above we can see that we select only nodes two and three to be participants in the Smart Contract. After the successful deployment of the contract we can see in the nodes the appropriate transaction established.



CURRENT BLOCK NUMBER 7 Last Block: 64 Days ago

ACTIVE NODES 5 Total Network Nodes: 5

TOTAL CONTRACTS 2 ABI AVAILABLE 2

RPC Port: 22000 IP Address: 10.50.0.2

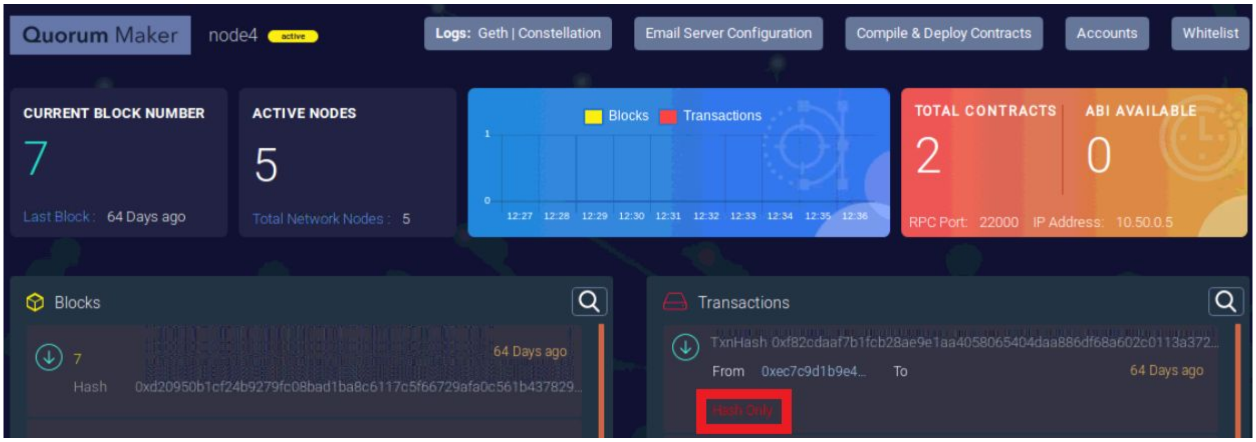
Blocks

7 64 Days ago
Hash 0xd20950b1cf24b9279fc08bad1ba8c6117c5f66729afa0c561b437829...

Transactions

TxnHash 0xf82cdaaf7b1fcb28ae9e1aa4058065404daa886df68a602c0113a372...
From 0xec7c9d1b9e4... To 64 Days ago
Private

If we access a node that was not a participant in this transaction (i.e. node4), we can see that only the hash for validation purposes is exposed to the specific node.



CURRENT BLOCK NUMBER 7

Last Block: 64 Days ago

ACTIVE NODES 5

Total Network Nodes: 5

TOTAL CONTRACTS 2

ABI AVAILABLE 0

RPC Port: 22000 IP Address: 10.50.0.5

Blocks

7 64 Days ago

Hash 0xd20950b1cf24b9279fc08bad1ba8c6117c5f66729afa0c561b437829...

Transactions

TxnHash 0xf82cdaaf7b1fcb28ae9e1aa4058065404daa886df68a602c0113a372...

From 0xec7c9d1b9e4... To 64 Days ago

4 Licensing

Quorum implementation comes licensed under the Apache License[6], Version 2.0 while IPFS is subject to a MIT License[7] which is a short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

5 Online Demonstrator

A demonstrator of the described prototype can be found online at demo5.bloomen.io with access credentials username: bloomen and password: secret.
The code is submitted in the project's Github github.com/bloomenio/demo5.
The front-end part is similar (with minor changes) compared to the previous prototype version.

6 Plan for Next Iteration - Conclusions

Anonymous personalization over open, trustless media platforms focuses on the way that identity management is going to be regulated inside business blockchain networks. Blockchain networks are not anonymised by default but with the right use of

[6] https://www.apache.org/licenses/LICENSE-2.0
[7] https://opensource.org/licenses/MIT

permissioned blockchains together with proper identity handling anonymisation inside business blockchain networks becomes feasible.

The plan for the next iteration is mostly changing the technical implementation of the demonstrator so that it is closer to a production environment. The off-chain part based on MongoDB and IPFS database and file system respectively will be further rolled out in order to be able to support a full scale synergy between the Quorum implementation on Alastria for all the pilots and use cases defined in the project. This way more data such as users and products will be able to flow into the demonstrator and simulate better real circumstances. Moreover, having synergies with the Know Your Customer framework and the user rights management a further integration and functionalities adaptation will be considered for achieving a seamless privacy enhancing experience for the users. Last, it is considered to implement also additional functionalities that would increase the anonymization, privacy and personalization features of the Bloomen system. Such an example is a ring based hashing of the user IDs that will allow for a perpetual masking of the user ID so that the unique hash ID that is related to the user will change in predefined intervals hindering thus the reveal of his identification and personalization features.