Bloomen

Blockchains in the new era of participatory media experience

HORIZON 2020

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

D2.3 Initial Bloomen overall architecture

Version:	**1.0**
Date:	**31/08/2018**
Authors:	**ICCS/NTUA (Lead), ATC, Worldline, Kendraio**
Type:	**Report**
Dissemination level:	**Public**

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

Table of Contents

1 Executive Summary

As stated in Bloomen's Description of Action, this deliverable document describes the initial approach of the overall Bloomen architecture. This first version will be the main reference guide to feed with the appropriate specifications the development of the Bloomen frameworks and enablers.

This architecture document consists of another four sections. The introductory section, the "Bloomen Architecture", the "Bloomen Modules and Requirements Mapping" and the "Conclusion". On the "Introduction", a brief summary of the requirements analyzed on deliverable "D2.2 Bloomen Requirements Analysis" is given together with their possible association with the architecture conception and design. Furthermore, the initial Bloomen overall architecture is described in section 3, "Bloomen Architecture". All the different layers and modules are analyzed and explained in detail according to the *Figure 1* which represents this architecture. Afterwards, in section 4, a clearer mapping of the requirements of deliverable D2.2 to the core modules of the architecture is presented.

2 Introduction

2.1 Requirements Summary & Overview

According to the deliverable "D2.2 Bloomen Requirements Analysis", the General Technical Requirements (GTR) are summarized in the list below. These requirements are considered necessary and applicable to the whole Bloomen Blockchain Platform without being use case specific.

- GTR_1 : Blockchain Administrator Login
- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support
- GTR_15 : User Digital Wallet (User Register & Login)
- GTR_16 : Anonymous Personalization
- GTR_17 : Incentive Consensus Algorithm

2.2 From requirements to architectural considerations

Requirements constitute the foundation of an architectural representation. When architecture has to be designed, project requirements are there to determine the designing technique and the structure goal. Architecture is putting all the pieces together to form a system.

The General Technical Requirements (GTR) of the deliverable "D2.2 Bloomen Requirements Analysis" are the core requirements which form the whole Bloomen architecture. All three architecture layers consist of modules with specific functionalities that derive as a necessary capabilities from the above requirements. When put together, the GTR clarify the architecture's main format and objective.

A structure of the architecture with the corresponding modules emerges from the GTR and it can be found in *Figure 1* in the next section. More on this structure is explained in detail throughout the document.

3 Bloomen Architecture

3.1 Overview

The following figure represents the Initial Bloomen overall architecture with the three core layers - "Application Layer", "Middleware Layer", "Blockchain Layer" -, the Bloomen API that implements the integration of the first two layers and the Security and Identity Management Module named "Anonymous Personalization Module" that interacts with the whole architecture.

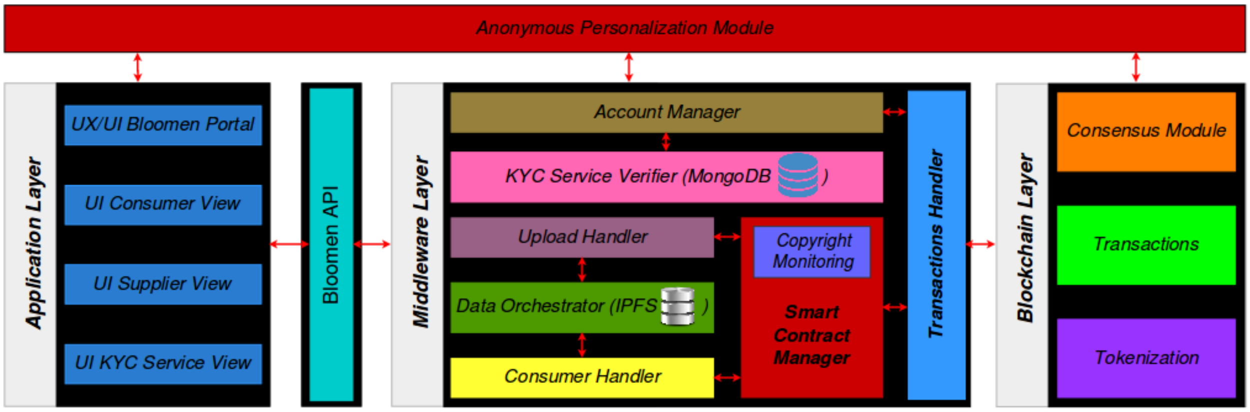


Figure 1. The top layer, "*Application Layer*", on the left, the *API* next, the "*Middleware Layer*" follows, the "*Blockchain Layer*" on the right (architecture bottom) and the "*Anonymous Personalization Module*" interacts with the whole architecture from top to bottom.

3.2 Description of Modules [bottom-up approach]

3.2.1 Blockchain Layer

This layer is the bottom layer of the Bloomen Architecture and describes the integration of the blockchain data structure and the Bloomen platform. Permissioned blockchain technologies such as Quorum, Hyperledger Fabric and others are recommended.

3.2.1.1 Consensus Module

Module Description

The Consensus Module is responsible for the general agreement and transaction verification among the peers of the Bloomen Blockchain Platform.

Interaction with other modules

This module is directly related with the other two modules of the Blockchain Layer (Transactions and Tokenization) and they all together form the blockchain data structure. The three modules interact with each other and as a whole communicate with the Middleware Layer.

Requirements mapping

- GTR_17 : Incentive Consensus Algorithm

3.2.1.2 Transactions

Module Description

The Transaction Module of this layer is responsible for storing the transaction data on the blockchain ledger.

Interaction with other modules

This module is directly related with the other two modules of the Blockchain Layer (Consensus Module and Tokenization) and they all together form the blockchain data structure. The three modules interact with each other and as a whole communicate with the Middleware Layer.

Requirements mapping

- GTR_1 : Blockchain Administrator Login
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support

- GTR_15 : User Digital Wallet
- GTR_16 : Anonymous Personalization

3.2.1.3 Tokenization

Module Description

The Tokenization module is responsible for providing token mechanisms needed for transactions. A new token can be created through a smart contract by following the Ethereum ERC20 standard.

Interaction with other modules

This module is directly related with the other two modules of the Blockchain Layer (Consensus Module and Transactions Module) and they all together form the blockchain data structure. The three modules interact with each other and as a whole communicate with the Middleware Layer.

Requirements mapping

- WTV_SR_1 : Smart Contracts
- WTV_SR_5.1 : Mining Algorithm
- WTV_SR_5.2 : Earn Tokens

3.2.2 Bloomen Middleware Services

Bloomen Middleware is a core layer of the Bloomen architecture as it is the connecting link between the Application and the Blockchain layer and it implements the core functionalities of the platform.

Bloomen Middleware layer will provide capabilities to:

- account access and managing by retrieving the corresponding data from the blockchain data structure and forwarding them to the Application layer.
- orchestrating content through corresponding handlers that interact with smart contracts
- transaction monitoring inside the blockchain P2P network

These capabilities are grouped in the following modules:

- Account Manager
- Upload Handler
- Data Orchestrator

- Consumer Handler
- KYC Service Verifier
- Smart Contract Manager
- Transaction Handler

3.2.2.1 Account Manager

Module Description

The Account Manager will be a service or daemon offering functionalities such as register, login, monitor on user account data and allowing the blockchain administrator to distribute permissions, issues certificates and others.

Interaction with other modules

This module is used to regulate accounts by interacting with the KYC Service Verifier and Transaction Handler. When a user connects to the Bloomen platform, their data are sent encrypted to the KYC Service Verifier module for validation; next, they are registered to the blockchain through the Transaction Handler along with every new user action.

Requirements mapping

- GTR_1 : Blockchain Administrator Login
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_15 : User Digital Wallet (User Register & Login)
- GTR_16 : Anonymous Personalization

3.2.2.2 Upload Handler

Module Description

The Upload Handler module is responsible for the initial uploading of media content on the corresponding distributed data storage.

Interaction with other modules

This module channels media content into distributed storage, like IPFS https://ipfs.io, and interacts with the Data Orchestrator and Smart Contract Manager modules. In order to upload content a smart contract is created and deployed through the latter module and then the content is guided through the Data Orchestrator to the respective distributed data storage.

Requirements mapping

- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_7 : Blockchain Administrator Right Protection
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing

3.2.2.3 Data Orchestrator

Module Description

The Data Orchestrator module is responsible for storing efficiently media content with low latency and in distributed way; IPFS-like distributed storage is recommended.

Interaction with other modules

This module retrieves and stores securely the content data provided from the Upload Handler and the Consumer Handler.

Requirements mapping

- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing

3.2.2.4 Consumer Handler

Module Description

The Consumer Handler module is responsible for the purchasing of media content from the corresponding distributed data storage.

Interaction with other modules

This module retrieves media content from the distributed storage, like IPFS https://ipfs.io, and interacts with the Data Orchestrator and Smart Contract Manager modules. In order to consume content an already deployed smart contract is triggered through the latter module and then the content is derived through the Data Orchestrator.

Requirements mapping

- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_7 : Blockchain Administrator Right Protection
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing

3.2.2.5 KYC Service Verifier

Module Description

The KYC Service Verifier module is used off-chain by the Media Suppliers in order to identify and permit users (Media Consumers or other Media Suppliers) to purchase their content. A document-oriented database like MongoDB for personal data storage is recommended.

Interaction with other modules

This module allows the Media Suppliers to validate the identity of their customers and for this reason interacts with the Account Manager module. KYC Service Verifier module is conducted off-chain in order to protect user privacy while the on-chain data will be encrypted with hash algorithms like SHA-256.

Requirements mapping

- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_8 : Privacy by design Logic

3.2.2.6 Smart Contract Manager

Module Description

The Smart Contract Manager module is responsible for the automatic and dynamic creation, deployment and triggering of smart contracts.

Interaction with other modules

This module is a core functionality module and interacts with the Upload, Consumer and Transaction Handlers. A blockchain administrator can create and deploy automatically any kind of smart contract through an up and running service (to grant permissions to users, to issue certificates, to do CRUD functionalities). When a user is uploading content, this module processes the corresponding inputs (metadata, asset value and payment requirements, supplier's public address and others) and creates a new smart contract. Then, the module deploys it to the blockchain through the Transaction Handler; when a user is purchasing content, this smart contract is triggered and emits a new transaction captured by the Transaction Handler and the purchase is complete.

Requirements mapping

- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_14 : Smart Contract Support
- GTR_16 : Anonymous Personalization

COPYRIGHT MONITORING

A copyright module may be required at the middleware layer. Copyright management is defined at the application layer (see below) as part of the Kendraio App. From our experience, we expect that some extra functionality will be required at the middleware layer to provide back office functionality for the API interactions from the Kendraio App.

3.2.2.7 Transaction Handler

Module Description

The Transaction Handler module regulates transaction managing and is responsible for all the Middleware and Blockchain layer interactions.

Interaction with other modules

This module manages every blockchain transaction emitted from the Middleware Layer and interacts with the Account Manager and Smart Contract Manager modules. When a new user registers or logs in, a new transaction is created and sent to the Transaction Handler which records it in the blockchain data structure. Also, every smart contract related functionality creates a new transaction which is received by the Transaction Handler and forwarded to the blockchain layer.

Requirements mapping

- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support
- GTR_16 : Anonymous Personalization

3.2.3 Bloomen REST API

Description

The Bloomen REST API provides RESTful services that are responsible for the efficient integration of the Application Layer with the Middleware Layer. This REST API will make available to the Application Layer all Middleware Layer functionalities providing to the Bloomen Blockchain Platform end-users a smooth UX experience. Technologies such as Java Spring or Python Flask frameworks are recommended.

For each Use Case pilot a different REST API would be exposed, based on the corresponding specific blockchain business workflow. As it is implied, the API depends mostly on the blockchain platform. For example, if the same blockchain platform is used for the three use case pilots, the same REST API could be used; whereas if the three pilots use a different blockchain, then three different APIs would be required.

3.2.4 Application Layer

Description

This layer will contain applications (UX / UI) made on top of functionalities exposed by the components of the lower architectural layers such as the Middleware Layer and Blockchain Layer. The main scope of this layer is to provide the Software as a Service approach to the BBP end users.

Application Layers Sub-Modules

UX / UI Bloomen Portal, UI Consumer View, UI Supplier View, UI KYC View and others

Mobile Clients & Digital Wallet

The Mobile Client & Digital Wallet will be a piece of software designed in such a way that an end user can interact with the different DAPPs (distributed applications) developed in Bloomen.

The main idea of this component is to provide a common application that allows us to explore and access the different DAPPs from a single point of entry for Bloomen.

This type of software has the mission to facilitate the creation of new DAPPs within the Bloomen ecosystem by simplifying the access of final customers to new projects.

The functionalities incorporated in the application will be:

1. Join to a DAPP.
2. Consult status.
3. Search for contents.
4. Create transactions, purchases.
5. See previous transactions.
6. Transfer balance.
7. Make requests for balance loading through prepaid cards.
8. Export the identity to other formats.

In **Annex I** a first version of the listed functions corresponding to the wireframes is attached to this document.

The application will be customized by the different DAPPs according to established parameters such as:

1. Look & Feel, logo, DAPP description and corporate colors.
2. Activate / Deactivate functionalities.
3. Offer an API for the interaction with the supplier for the actions of balance acquisition and content search (pilots code).

This module will be compatible with Android, IOS, and WEB mobile systems.

Kendraio App

Module Description

The purpose of the Kendraio App module within the Application layer is to allow the creation and curation of resource metadata, with a special focus on assertion and management of rights.

Interaction with other modules

Distribute rights metadata to facilitate claim assertion and conflict resolution, allow for automated support of contract negotiation, collaboration and automated payments.

Content distributors all have different standards for holding rights metadata. In order to help content creators to manage and distribute this metadata we can provide a system to consolidate and normalise rights across those standards, allowing content creators to have a central repository of their metadata that can be transformed to the correct target schema.

3.2.5 Security & Identity Management

The Security and Identity Management interacts with the whole Bloomen architecture which means with all three core layers. This layer implements the logic of Bloomen Blockchain Platform to protect user privacy from unwilling public exposure or exploitation and to anonymize sensitive data when necessary. The scope of this layer is to encrypt user data and de-associate it with personalized identities offering at the same time an Anonymous ID which is used for personalization of the user in an anonymous way. The objective is that a user's real identity and any private information will not be visible to the blockchain network unless the user chooses so.

Anonymous Personalization Module

Module Description

The purpose of the Anonymous Personalization Service is allowing users to purchase assets without revealing their real identity while at the same time the identification of the users as a legal entities is done off-chain. Also, the user is able to choose what kind of personal information can be exposed on the blockchain network. In this case, any content provider will be able to provide personalized services to the user; only knowing the revealed information.

Interaction with other modules

The Anonymous Personalization module is the one coping with all security, privacy and reliability concerns for the entire Bloomen Blockchain Platform. This module provides its functionalities to all the other ones. By itself, it does not initiate any interaction, but its use is systematically integrated in the chain of calls implying any other functional module of the platform.

Anonymous Personalization module is cross to the whole architecture and provides capabilities to ensure that security requirements of Bloomen are satisfied. In particular it provides necessary functionalities to check and authorize the access to the modules of the platform in a secure way for the end users allowing personalized services.

Requirements mapping

Includes all the requirements, as this specific module interacts with the whole Bloomen Architecture.

- GTR_1 : Blockchain Administrator Login
- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support
- GTR_15 : User Digital Wallet
- GTR_16 : Anonymous Personalization
- GTR_17 : Incentive Consensus Algorithm

4 Bloomen Modules and Requirements Mapping

4.1 Blockchain Layer requirements mapping

The following requirements as defined in D2.2 are mapped to the Blockchain Layer:

- GTR_1 : Blockchain Administrator Login
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support
- GTR_15 : User Digital Wallet
- GTR_16 : Anonymous Personalization
- GTR_17 : Incentive Consensus Algorithm
- WTV_SR_1 : Smart Contracts
- WTV_SR_5.1 : Mining Algorithm
- WTV_SR_5.2 : Earn Tokens

4.2 Bloomen Middleware Services requirements mapping

The following requirements as defined in "D2.2 Bloomen Requirements Analysis" are mapped to the Middleware Layer:

- GTR_1 : Blockchain Administrator Login
- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation

- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_15 : User Digital Wallet (User Register & Login)
- GTR_16 : Anonymous Personalization

4.3 Security & Identity Management requirements mapping

All General Technical Requirements as defined in "D2.2 Bloomen Requirements Analysis" are mapped to the Security and Identity Management which interacts with the whole Bloomen Blockchain Platform:

- GTR_1 : Blockchain Administrator Login
- GTR_2 : Blockchain Administrator Creates Smart Contracts
- GTR_3 : Blockchain Administrator Deploys Smart Contracts
- GTR_4 : Blockchain Administrator Issues Certificates
- GTR_5 : Blockchain Administrator Grants Permissions
- GTR_6 : Blockchain Administrator CRUD
- GTR_7 : Blockchain Administrator Right Protection
- GTR_8 : Privacy by design Logic
- GTR_9 : Privacy by design Data Anonymization
- GTR_10 : Privacy by design Data Separation
- GTR_11 : Privacy by design Data Storage
- GTR_12 : Privacy by design Data Sharing and Processing
- GTR_13 : Privacy by design Blockchain Timestamping
- GTR_14 : Smart Contract Support
- GTR_15 : User Digital Wallet
- GTR_16 : Anonymous Personalization
- GTR_17 : Incentive Consensus Algorithm

5 Conclusions

This deliverable described the Initial Bloomen overall architecture. The concept of the architecture consists of three core layers, the Bloomen API and the Security and Identity Management Module. The three layers are named "Application Layer", "Middleware Layer" and "Blockchain Layer". The Bloomen API is used in order to integrate the Application with the Middleware layer. Furthermore, the Security and Identity Management is currently represented by the "Anonymous Personalization Module" which interacts with all the three layers of the architecture.

The architecture of the deliverable is a simple approach which displays how media industry architecture could integrate with blockchain transactions and smart contracts. Modern blockchain platforms offer dozens of advantages that can transform and benefit not only media industry but other enterprise use cases as well. When using them wisely and efficiently a new path for technology evolution, scientific research and business expansion and growth is opening.

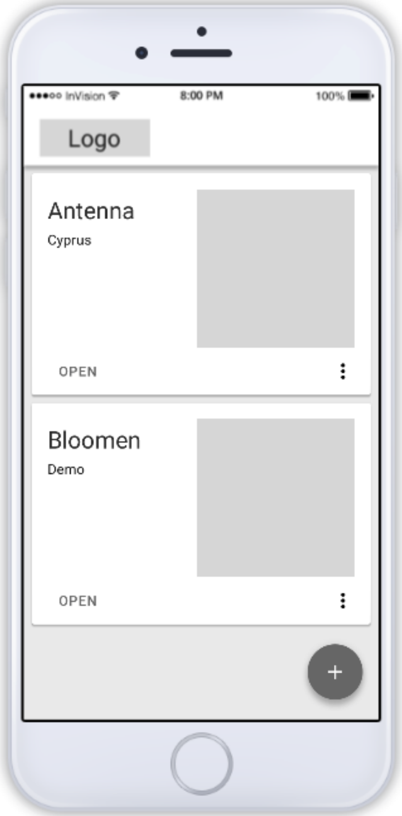
APPENDIX I: MOBILE APP WIREFRAMES

The mobile app wireframes can be viewed Online on the following link:

- **Url:** https://invis.io/FPNK3VSD28R
- **Password:** bloomen

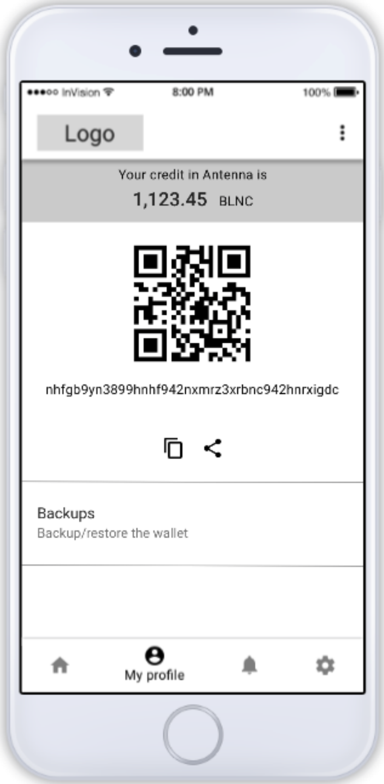
For your convenience, the wireframe screens can also be seen below:

1. Join to DAPP



DAPP enrollment

2. Check status



Your credit in Antenna is

1,123.45 BLNC

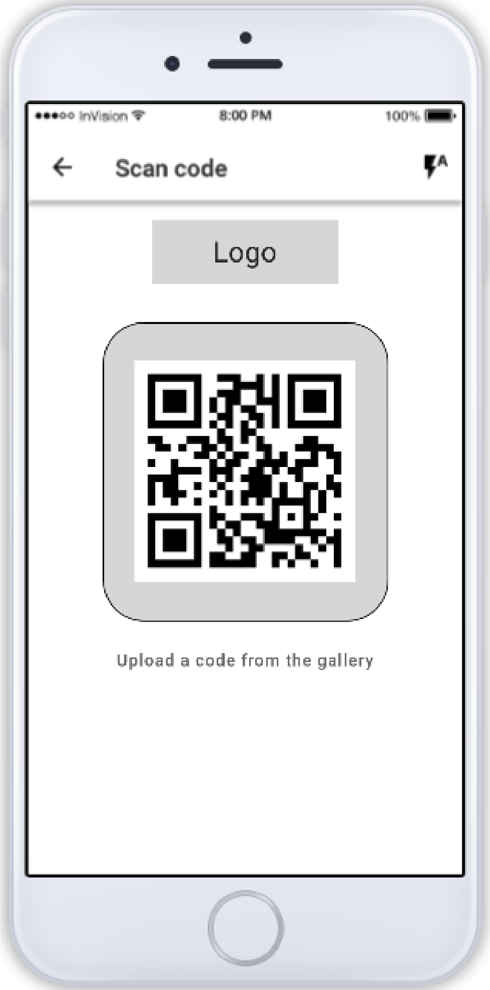
nhfgb9yn3899hnhf942nxmrz3xrbnc942hnrxigdc

Backups

Backup/restore the wallet

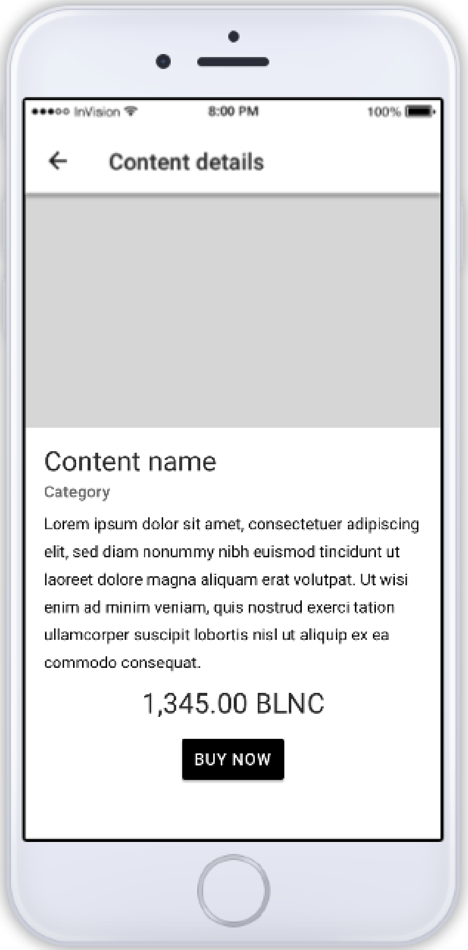
DAPP status

3. Content search



DAPP search content

4. Create transactions, purchases.



Content name

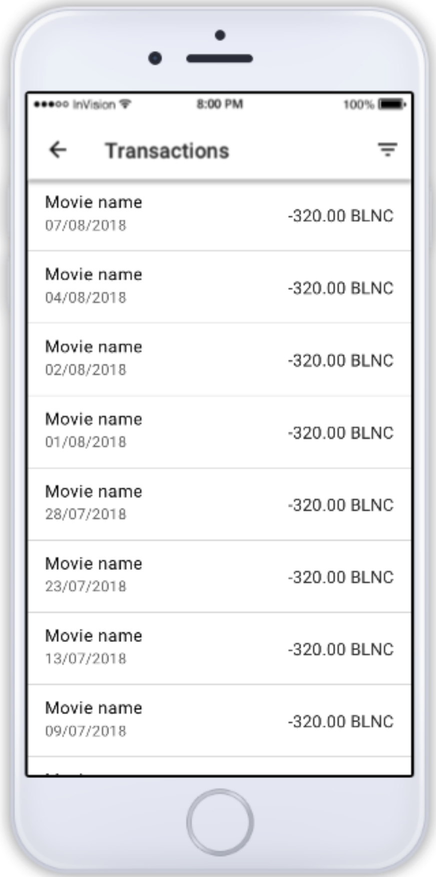
Category

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

1,345.00 BLNC

DAPP buy

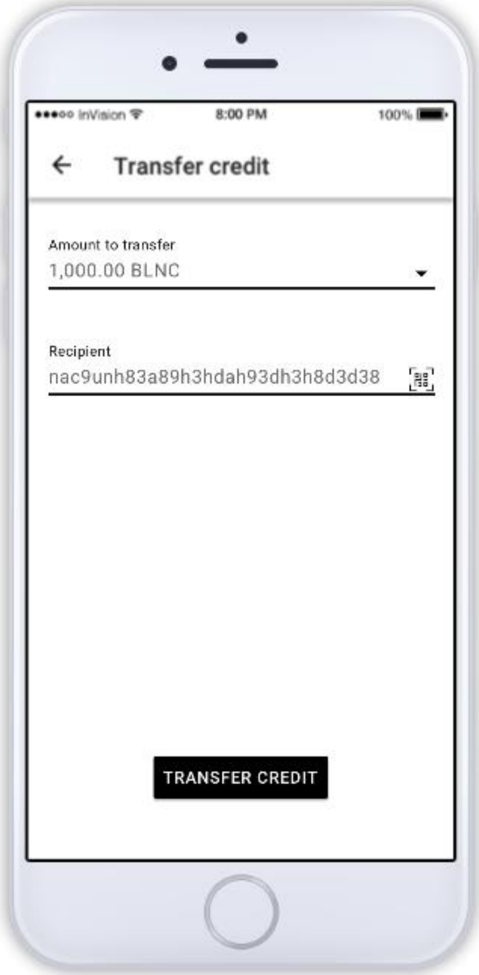
5. See previous transactions.



Movie name 07/08/2018	-320.00 BLNC
Movie name 04/08/2018	-320.00 BLNC
Movie name 02/08/2018	-320.00 BLNC
Movie name 01/08/2018	-320.00 BLNC
Movie name 28/07/2018	-320.00 BLNC
Movie name 23/07/2018	-320.00 BLNC
Movie name 13/07/2018	-320.00 BLNC
Movie name 09/07/2018	-320.00 BLNC

DAPP transaction history

6. Transfer balance.



●●●○○ InVision 8:00 PM 100%

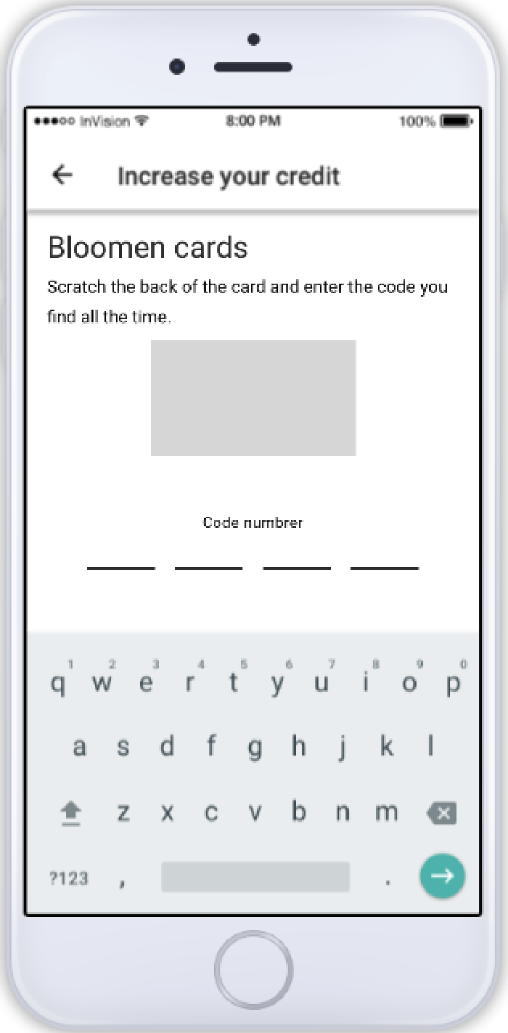
Transfer credit

Amount to transfer
1,000.00 BLNC

Recipient
nac9unh83a89h3hdah93dh3h8d3d38

DAPP send tokens

7. Make requests to load the balance using prepaid cards.



Increase your credit

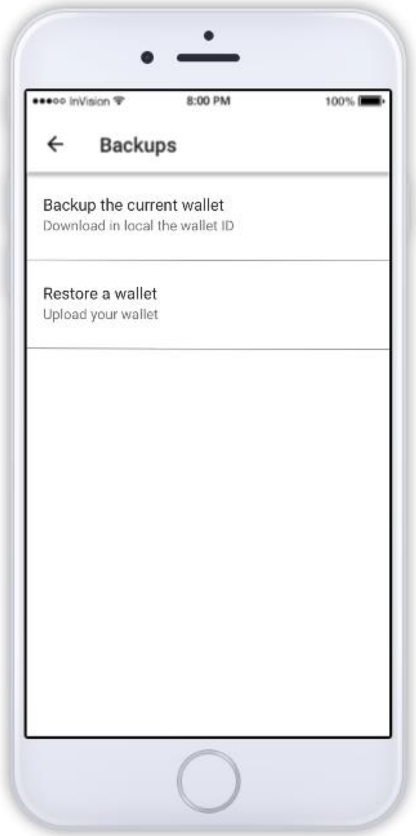
Bloomen cards

Scratch the back of the card and enter the code you find all the time.

Code numbrer

DAPP ask for tokens

8. Export the identity to other formats.



Backups

Backup the current wallet
Download in local the wallet ID

Restore a wallet
Upload your wallet

DAPP export / import identity